Name – Nandan Shailesh Kasat

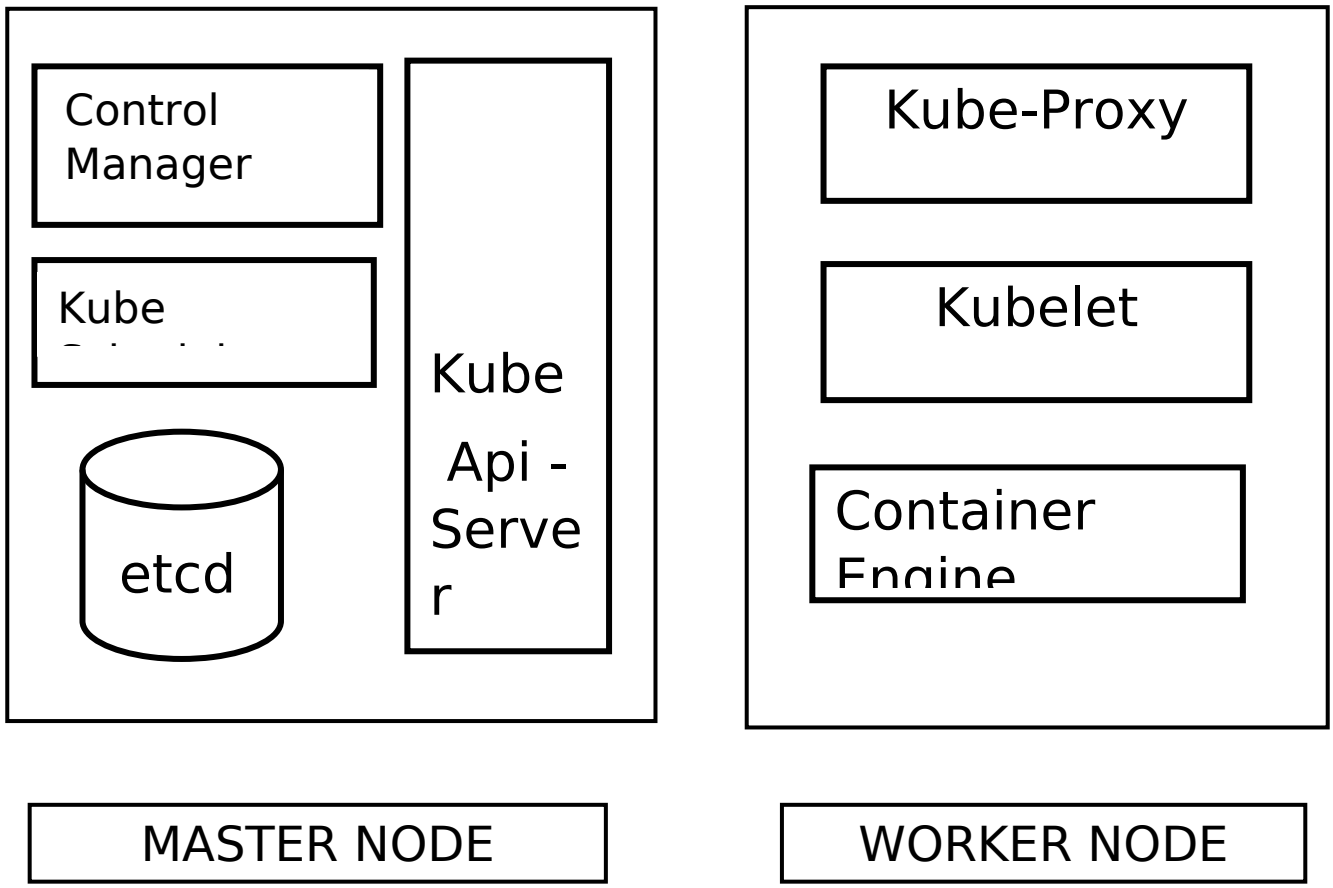
Kubernetes Task 1

Q1. Write a note on Kubernetes Architecture. Explain about each component of Kubernetes cluster.

Ans:

Kubernetes follows a client-server architecture. It's designed as a set of loosely coupled and highly modular components, which makes the system flexible, extensible, and easy to manage.
A master and worker nodes constitute a Kubernetes cluster.

Overview of the Kubernetes Architecture:



Master Node: The master node is responsible for the management of the Kubernetes cluster. It is the entry point for all administrative tasks.

The master node manages worker nodes and pods in the cluster. The components of the master node include:

API Server (kube-apiserver): Acts as the frontend for the Kubernetes control plane.

It exposes the Kubernetes API and is the main management point of the entire cluster.

Controller Manager (kube-controller-manager): Runs controllers, which are background threads that handle routine tasks in the cluster.

Scheduler (kube-scheduler): Distributes work or containers across multiple nodes. It looks for newly created pods and assigns them to nodes.

etcd: A consistent and highly-available key value store used as Kubernetes' backing store for all cluster data.

Worker Nodes: Worker nodes are the machines (VMs, physical servers, etc.) that run your applications and cloud workflows.

The components of a worker node include:

Kubelet: An agent that runs on each node in the cluster. It makes sure that containers are running in a pod1.

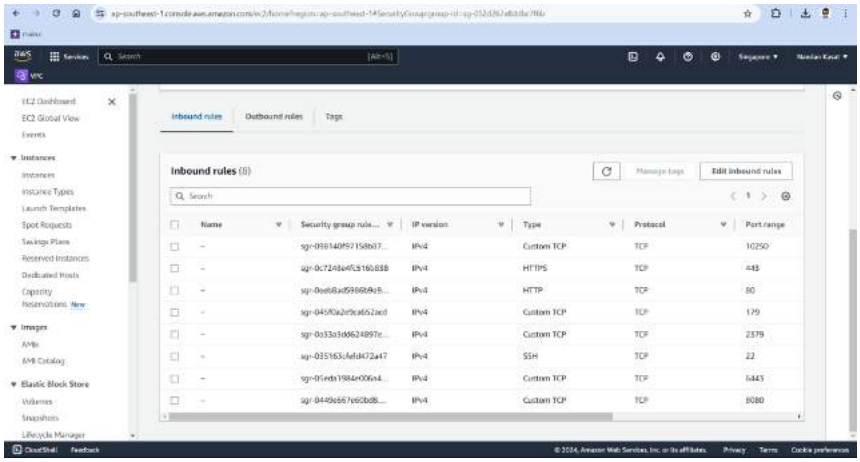
Kube Proxy (kube-proxy): A network proxy that runs on each node in your cluster, implementing part of the Kubernetes Service concept.

Pods (Container Engine): The smallest and simplest unit in the Kubernetes object model that you create or deploy.

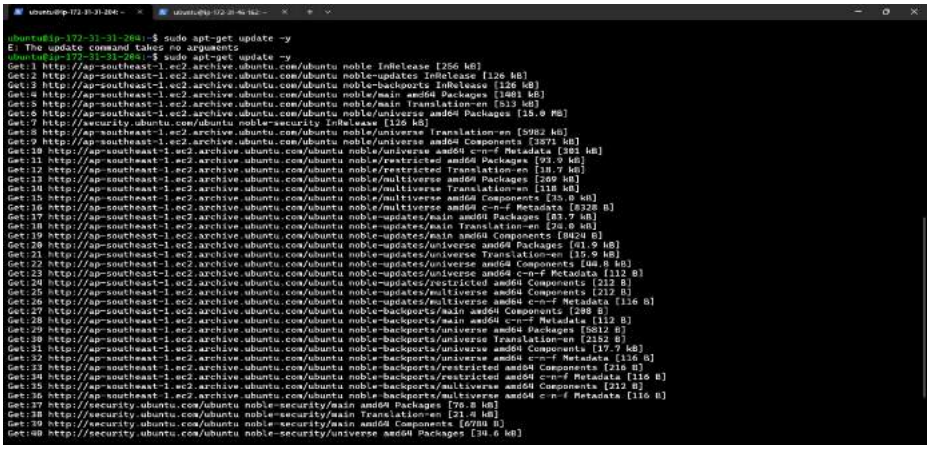
Q2. Prepare a documentation on Kubernetes setup on Ubuntu. With screenshot of each command.

On Master:

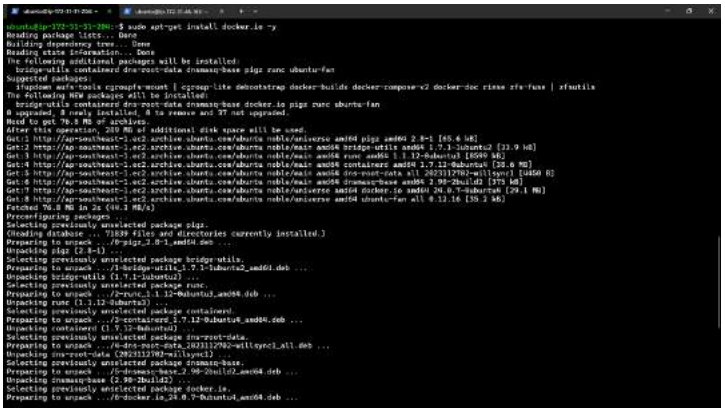
Add Rules 8080, 179, 6443, 10250, 2379 to Security Group of Master Node instance (kube_master).



1 sudo apt-get update –y

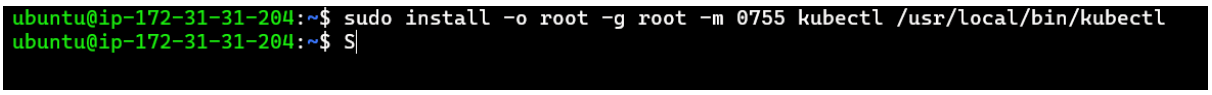


2 sudo apt-get install docker.io –y

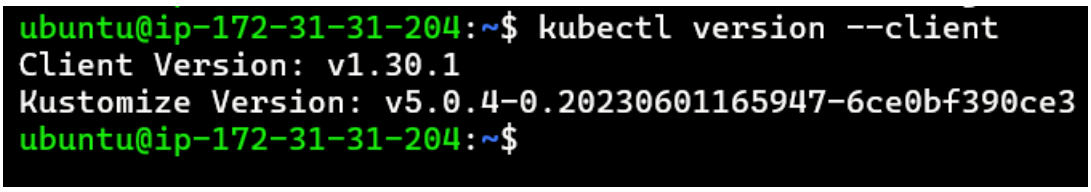


3 curl -LO “https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl.sha256”

4 sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl



5 kubectl version –client



6 sudo apt-get update

7 sudo apt-get install -y apt-transport-https ca-certificates curl

```
ubuntu@ip-172-31-31-204:~$ 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 34 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://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 apt-transport-https all 2.7.14build2 [3974 B]
Get:2 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 curl amd64 8.5.0-2ubuntu10.1 [227 kB]
Get:3 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libcurl4t64 amd64 8.5.0-2ubuntu10.1 [341 kB]
Get:4 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libcurl3t64-gnutls amd64 8.5.0-2ubuntu10.1 [333 kB]
Fetched 904 kB in 11s (83.2 kB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 72206 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.7.14build2_all.deb ...
Unpacking apt-transport-https (2.7.14build2) ...
Preparing to unpack .../curl_8.5.0-2ubuntu10.1_amd64.deb ...
Unpacking curl (8.5.0-2ubuntu10.1) over (8.5.0-2ubuntu10) ...
Preparing to unpack .../libcurl4t64_8.5.0-2ubuntu10.1_amd64.deb ...
Unpacking libcurl4t64:amd64 (8.5.0-2ubuntu10.1) over (8.5.0-2ubuntu10) ...
Preparing to unpack .../libcurl3t64-gnutls_8.5.0-2ubuntu10.1_amd64.deb ...
Unpacking libcurl3t64-gnutls:amd64 (8.5.0-2ubuntu10.1) over (8.5.0-2ubuntu10) ...
Setting up apt-transport-https (2.7.14build2) ...
Setting up libcurl4t64:amd64 (8.5.0-2ubuntu10.1) ...
Setting up libcurl3t64-gnutls:amd64 (8.5.0-2ubuntu10.1) ...
Setting up curl (8.5.0-2ubuntu10.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8) ...
Scanning processes...
Scanning candidates...
Scanning linux images...

Running kernel seems to be up-to-date.
```

8 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-172-31-31-204:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.30/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyri
ng.gpg
ubuntu@ip-172-31-31-204:~$
```

9 sudo chmod 644 /etc/apt/keyrings/kubernetes-apt-keyring.gpg

```
ubuntu@ip-172-31-31-204:~$ sudo chmod 644 /etc/apt/keyrings/kubernetes-apt-keyring.gpg
ubuntu@ip-172-31-31-204:~$
```

10 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

```
ubuntu@ip-172-31-31-204:~$ echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.30/deb/ /' | sudo tee /et
c/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-172-31-31-204:~$
```

11 sudo chmod 644 /etc/apt/sources.list.d/kubernetes.list

```
ubuntu@ip-172-31-31-204:~$ sudo chmod 644 /etc/apt/sources.list.d/kubernetes.list
ubuntu@ip-172-31-31-204:~$
```

12 sudo apt-get update

```
ubuntu@ip-172-31-31-204:~$ sudo apt-get update
Hit:1 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:4 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.30/deb  InRelease [1186 B]
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.30/deb  Packages [3957 B]
Fetched 5143 B in 1s (6668 B/s)
Reading package lists... Done
ubuntu@ip-172-31-31-204:~$
```

13sudo apt-get install -y kubectl kubeadm kubelet

```
ubuntu@ip-172-31-31-204:~$ 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 34 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://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 conntrack amd64 1:1.4.8-1ubuntu1 [37.9 kB]
Get:2 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 ebtables amd64 2.0.11-6build1 [88.4 kB]
Get:3 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 socat amd64 1.8.0.0-4build3 [374 kB]
Get:4 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.30/deb  cri-tools 1.30.0-1.1 [21.3 MB]
Get:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.30/deb  kubeadm 1.30.1-1.1 [10.4 MB]
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.30/deb  kubectl 1.30.1-1.1 [10.8 MB]
Get:7 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.30/deb  kubernetes-cni 1.4.0-1.1 [32.9 MB]
Get:8 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.30/deb  kubelet 1.30.1-1.1 [18.1 MB]
Fetched 93.9 MB in 2s (59.0 MB/s)
```

14sudo systemctl enable kubelet && sudo systemctl start kubelet

```
ubuntu@ip-172-31-31-204:~$ sudo systemctl enable kubelet &&  sudo systemctl start kubelet
ubuntu@ip-172-31-31-204:~$
```

///// 16 sudo kubeadm init --pod-network-cidr=192.168.0.0/16

17 sudo kubeadm init --pod-network-cidr=192.168.0.0/16 --ignore-preflight-errors=all

```
ubuntu@ip-172-31-31-204:~$ sudo kubeadm init --pod-network-cidr=192.168.0.0/16  --ignore-preflight-errors=all
[init] Using Kubernetes version: v1.30.1
[preflight] Running pre-flight checks
        [WARNING NumCPU]: the number of available CPUs 1 is less than the required 2
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
W0529 14:34:02.930289    3398 checks.go:844] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that
used by kubeadm.It is recommended to use "registry.k8s.io/pause:3.9" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-31-204 kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster
.local] and IPs [10.96.0.1 172.31.31.204]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [ip-172-31-31-204 localhost] and IPs [172.31.31.204 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-31-204 localhost] and IPs [172.31.31.204 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
```

18 mkdir -p $HOME/.kube

```
ubuntu@ip-172-31-31-204:~$ mkdir -p $HOME/.kube
ubuntu@ip-172-31-31-204:~$
```

19 sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

```
ubuntu@ip-172-31-31-204:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
ubuntu@ip-172-31-31-204:~$
```

20 sudo chown $(id -u):$(id -g) $HOME/.kube/config

```
ubuntu@ip-172-31-31-204:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-31-204:~$
```

21 sudo su

export KUBECONFIG=/etc/kubernetes/admin.conf

```
root@ip-172-31-31-204:/home/ubuntu# export KUBECONFIG=/etc/kubernetes/admin.conf
root@ip-172-31-31-204:/home/ubuntu# exit
exit
ubuntu@ip-172-31-31-204:~$
```

Exit

22 kubectl create -f https://raw.githubusercontent.com/projectcalico/calico/v3.27.3/manifests/tigera-operator.yaml

```
ubuntu@ip-172-31-31-204:~$ 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
customresourcedefinition.apiextensions.k8s.io/tigerastatuses.operator.tigera.io created
serviceaccount/tigera-operator created
clusterrole.rbac.authorization.k8s.io/tigera-operator created
clusterrolebinding.rbac.authorization.k8s.io/tigera-operator created
deployment.apps/tigera-operator created
ubuntu@ip-172-31-31-204:~$
```

23 kubectl create -f https://raw.githubusercontent.com/projectcalico/calico/v3.27.3/manifests/custom-resources.yaml

```
ubuntu@ip-172-31-31-204:~$ kubectl create -f https://raw.githubusercontent.com/projectcalico/calico/v3.27.3/manifests/custom-resources.yaml
installation.operator.tigera.io/default created
apiserver.operator.tigera.io/default created
ubuntu@ip-172-31-31-204:~$
```

24 kubectl get nodes

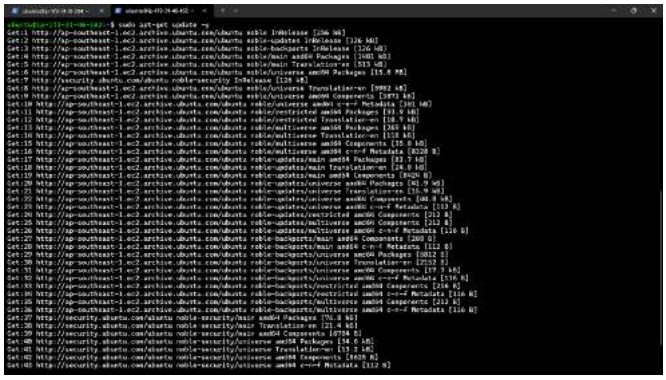
```
ubuntu@ip-172-31-31-204:~$ kubectl get nodes
NAME                STATUS   ROLES           AGE    VERSION
ip-172-31-24-174    Ready    <none>          62s    v1.30.1
ip-172-31-31-204    Ready    control-plane   153m   v1.30.1
ubuntu@ip-172-31-31-204:~$
```

25 kubectl get componentstatus

```
ubuntu@ip-172-31-31-204:~$ kubectl get 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-172-31-31-204:~$
```

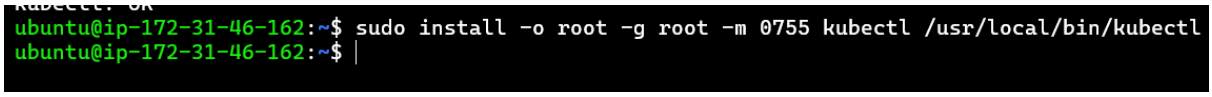
On Worker (kube_worker):

1 sudo apt-get update –y

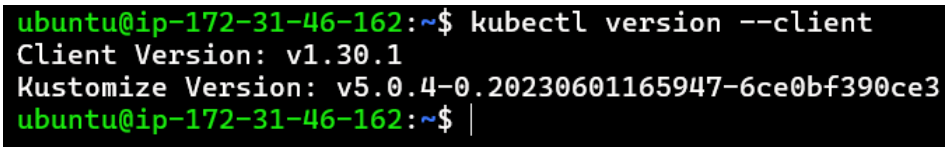


2 curl -LO “https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl.sha256”

3 sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl

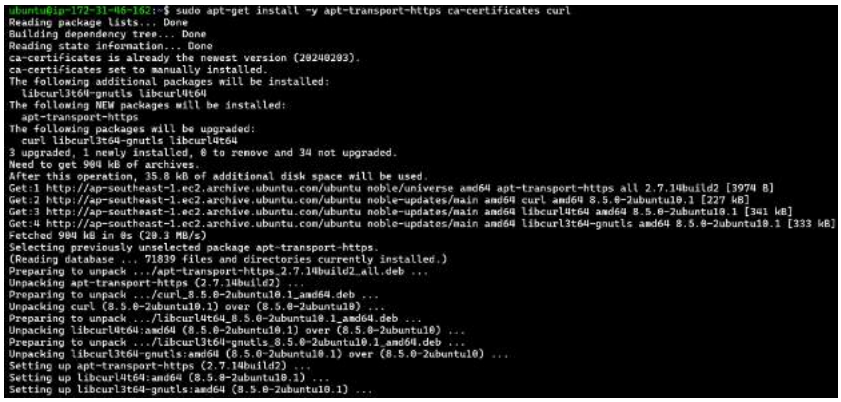


4 kubectl version –client

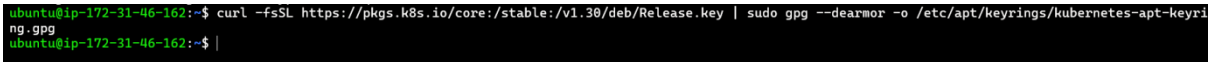


5 sudo apt-get update

6 sudo apt-get install -y apt-transport-https ca-certificates curl



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



8 sudo chmod 644 /etc/apt/keyrings/kubernetes-apt-keyring.gpg

```
ubuntu@ip-172-31-46-162:~$ sudo chmod 644 /etc/apt/keyrings/kubernetes-apt-keyring.gpg
```

9 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

```
ubuntu@ip-172-31-46-162:~$ echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.30/deb/ /' | sudo tee /et
c/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-172-31-46-162:~$
```

10 sudo chmod 644 /etc/apt/sources.list.d/kubernetes.list

```
ubuntu@ip-172-31-46-162:~$ sudo chmod 644 /etc/apt/sources.list.d/kubernetes.list
ubuntu@ip-172-31-46-162:~$
```

11 sudo apt-get update

```
ubuntu@ip-172-31-46-162:~$ sudo apt-get update
Hit:1 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
```

12 sudo apt-get install -y kubectl kubeadm kubelet

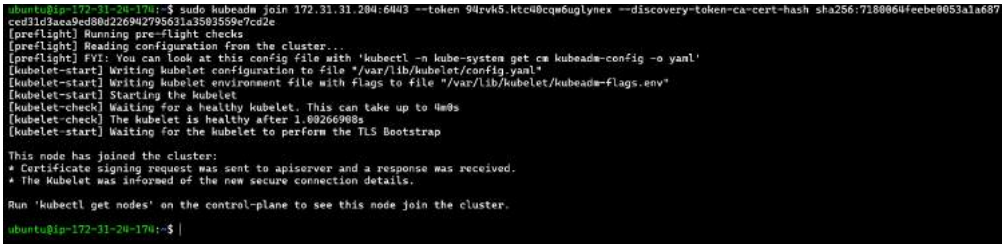
```
ubuntu@ip-172-31-46-162:~$ sudo apt-get install -y kubectl kubeadm kubelet
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools ebtables kubernetes-cni socat
The following NEW packages will be installed:
  conntrack cri-tools ebtables kubeadm kubectl kubelet kubernetes-cni socat
0 upgraded, 8 newly installed, 0 to remove and 34 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://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 conntrack amd64 1:1.4.8-1ubuntu1 [37.9 kB]
Get:2 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 ebtables amd64 2.0.11-6build1 [88.4 kB]
Get:3 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 socat amd64 1.8.0.0-4build3 [374 kB]
Get:4 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.30/deb  cri-tools 1.30.0-1.1 [21.3 MB]
Get:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.30/deb  kubeadm 1.30.1-1.1 [10.4 MB]
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.30/deb  kubectl 1.30.1-1.1 [10.8 MB]
Get:7 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.30/deb  kubernetes-cni 1.4.0-1.1 [32.9 MB]
Get:8 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.30/deb  kubelet 1.30.1-1.1 [18.1 MB]
Fetched 93.9 MB in 2s (60.3 MB/s)
```

13 sudo systemctl enable kubelet && sudo systemctl start kubelet

```
ubuntu@ip-172-31-46-162:~$ sudo systemctl enable kubelet &&  sudo systemctl start kubelet
ubuntu@ip-172-31-46-162:~$
```

14 sudo kubeadm join 172.31.31.204:6443 --token 94rvk5.ktc40cqw6uglynex --discovery-token-ca-cert-hash

sha256:7180064feebe0053a1a687ced31d3aea9ed80d226
942795631a3503559e7cd2e



Q3. 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.

Ans:

Step 1: Create a httpd images.

Step 2: Run httpd image:

```
root@ip-172-31-36-148:/home/ubuntu# docker run -d --name my-httpd httpd
71df14b0e615ee26e4a1164c4a699c3ef3b6ee4e103786b2274d20c99840a84c
root@ip-172-31-36-148:/home/ubuntu# docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
httpd        latest    356125da0595   8 weeks ago   147MB
root@ip-172-31-36-148:/home/ubuntu# docker ps
CONTAINER ID   IMAGE     COMMAND              CREATED              STATUS              PORTS     NAMES
71df14b0e615   httpd     "httpd-foreground"   About a minute ago   Up About a minute   80/tcp    my-httpd
```

Step 3: Go inside the httpd container and create your own pages**:**

docker exec -it 71df14b0e615 /bin/bash

```
root@ip-172-31-36-148:/home/ubuntu# docker exec -it 71df14b0e615 /bin/bash
root@71df14b0e615:/usr/local/apache2# cd htdocs/
```

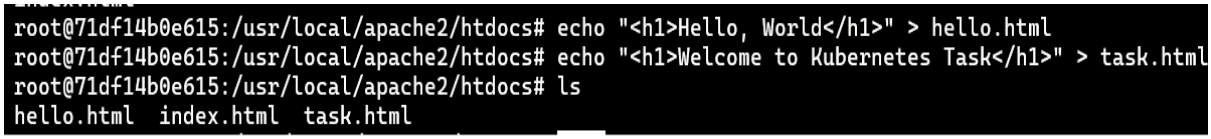
Step 4: Navigate to the htdocs directory


```
root@ip-172-31-36-148:/home/ubuntu# docker exec -it 71df14b0e615 /bin/bash
root@71df14b0e615:/usr/local/apache2# cd htdocs/
```

Step 6: Create HTML pages

root@71df14b0e615:/usr/local/apache2/htdocs# echo "<h1>Hello, World</h1>" > hello.html

root@71df14b0e615:/usr/local/apache2/htdocs# echo "<h1>Welcome to Kubernetes Task</h1>" > task.html



Step 7: Access the pages from inside the container**:**

root@71df14b0e615:~# curl localhost/task.html

<h1>Welcome to Kubernetes Task</h1>