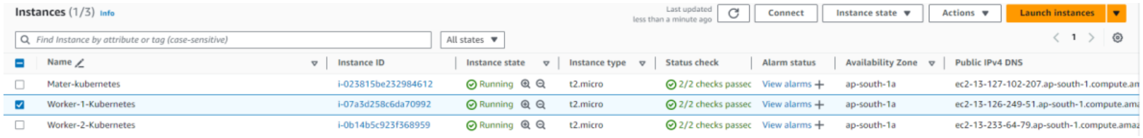
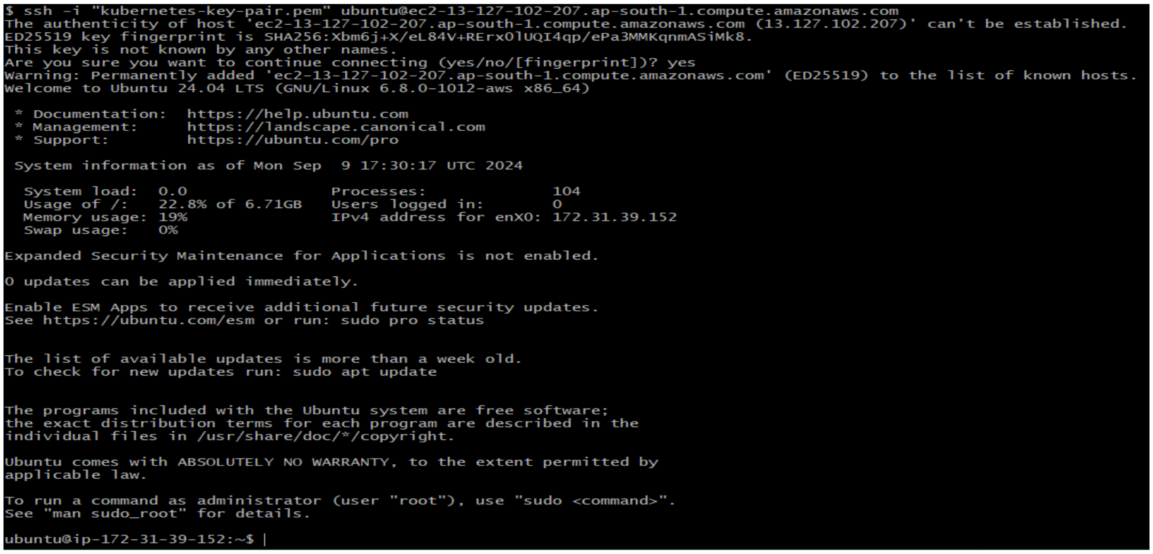
Documentation for creating the master and node architecture on the aws ec2.

1.Create three ec2 instances with 2vcpu,4 gb ram for master and t2.micro for worker node. Or you can use t2.micro as well.


Instances (1/3) Info
Last updated less than a minute ago
Connect
Instance state ▼
Actions ▼
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states ▼

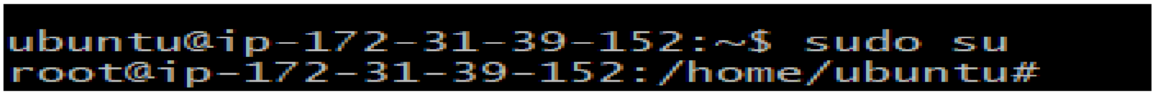
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
□	Mater-kubernetes	i-023815be232984612	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	ap-south-1a	ec2-13-127-102-207.ap-south-1.compute.am
☑	Worker-1-Kubernetes	i-07a3d258c6da70992	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	ap-south-1a	ec2-13-126-249-51.ap-south-1.compute.ama
□	Worker-2-Kubernetes	i-0b14b5c923f368959	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	ap-south-1a	ec2-13-233-64-79.ap-south-1.compute.amaz

2.Now take access of the ec2 instances(nodes) one by one.



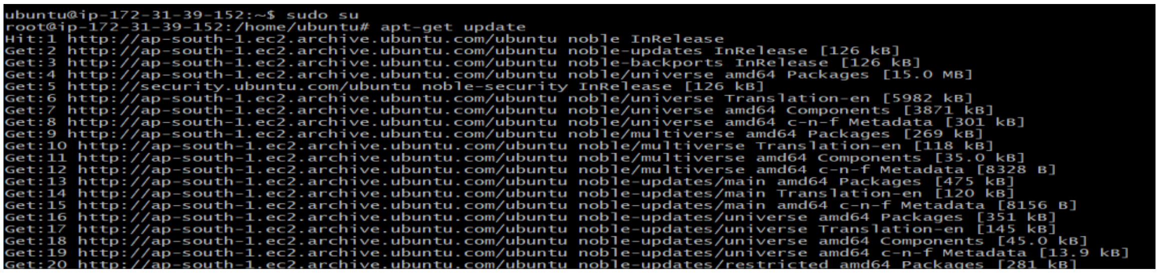
```
$ ssh -i "kubernetes-key-pair.pem" ubuntu@ec2-13-127-102-207.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-13-127-102-207.ap-south-1.compute.amazonaws.com (13.127.102.207)' can't be established.
ED25519 key fingerprint is SHA256:Xbm6j+X/eL84V+RErx0lUQI4qp/ePa3MMKqnmASiMk8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-127-102-207.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Mon Sep  9 17:30:17 UTC 2024

  System load:  0.0               Processes:             104
  Usage of /:   22.8% of 6.71GB   Users logged in:       0
  Memory usage: 19%               IPv4 address for enX0: 172.31.39.152
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-39-152:~$ |
```

sudo su (Repeated same below steps in the worker nodes 1,2)



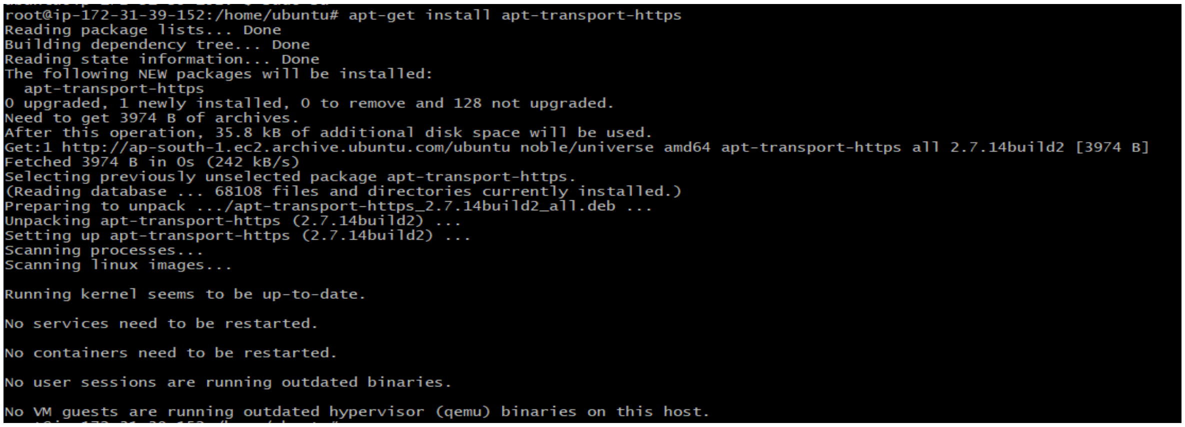
```
ubuntu@ip-172-31-39-152:~$ sudo su
root@ip-172-31-39-152:/home/ubuntu#
```

apt-get update



```
ubuntu@ip-172-31-39-152:~$ sudo su
root@ip-172-31-39-152:/home/ubuntu# apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [475 kB]
Get:14 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [120 kB]
Get:15 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [8156 B]
Get:16 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [351 kB]
Get:17 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [145 kB]
Get:18 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:19 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [13.9 kB]
Get:20 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [281 kB]
```

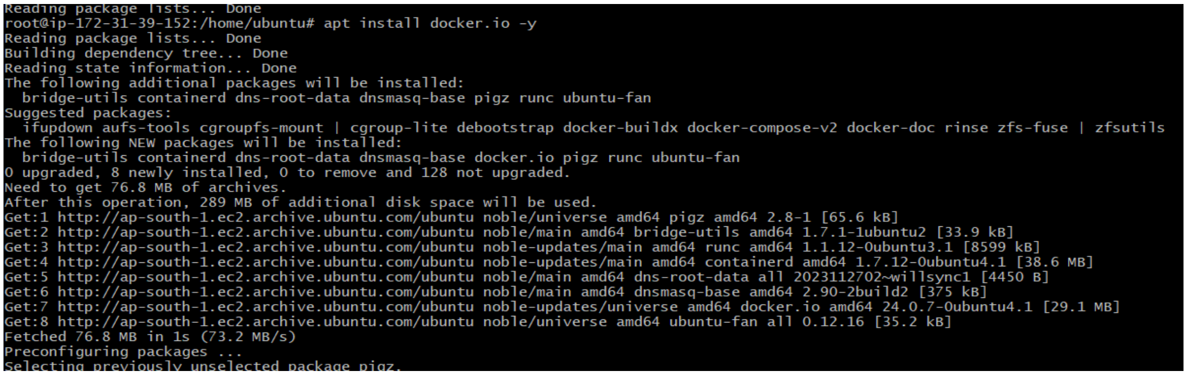
apt-get install apt-transport-https(for the secure communication between the master and worker nodes).



```
root@ip-172-31-39-152:/home/ubuntu# apt-get install apt-transport-https
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  apt-transport-https
0 upgraded, 1 newly installed, 0 to remove and 128 not upgraded.
Need to get 3974 B of archives.
After this operation, 35.8 kB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 apt-transport-https all 2.7.14build2 [3974 B]
Fetched 3974 B in 0s (242 kB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 68108 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.7.14build2_all.deb ...
Unpacking apt-transport-https (2.7.14build2) ...
Setting up apt-transport-https (2.7.14build2) ...
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.
```

.Now install docker in the three ec2 one by one

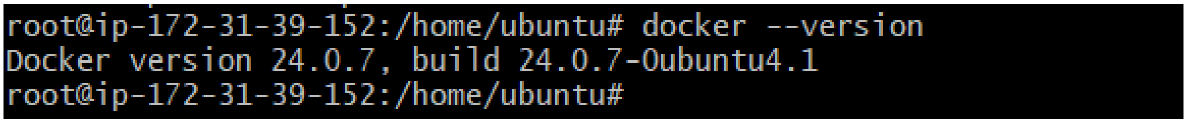
By hitting "apt install docker.io -y"



```
Reading package lists... Done
root@ip-172-31-39-152:/home/ubuntu# apt install docker.io -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-buildx docker-compose-v2 docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 128 not upgraded.
Need to get 76.8 MB of archives.
After this operation, 289 MB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 bridge-utils amd64 1.7.1-1ubuntu2 [33.9 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.1.12-0ubuntu3.1 [8599 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.12-0ubuntu4.1 [38.6 MB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 dns-root-data all 2023112702~willsync1 [4450 B]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 dnsmasq-base amd64 2.90-2build2 [375 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 docker.io amd64 24.0.7-0ubuntu4.1 [29.1 MB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 ubuntu-fan all 0.12.16 [35.2 kB]
Fetched 76.8 MB in 1s (73.2 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
```

ensure whether docker is installed or not by hitting

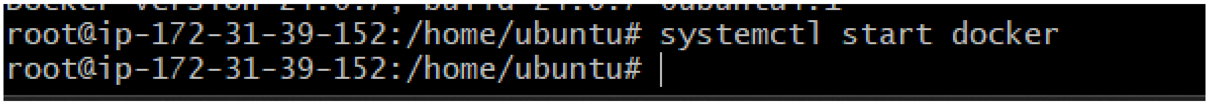
Docker --version



```
root@ip-172-31-39-152:/home/ubuntu# docker --version
Docker version 24.0.7, build 24.0.7-0ubuntu4.1
root@ip-172-31-39-152:/home/ubuntu#
```

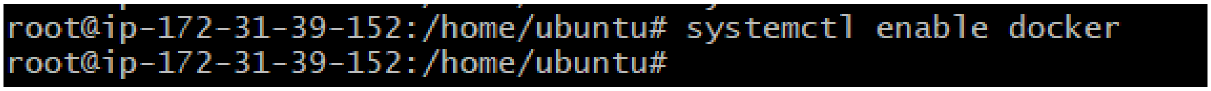
start docker by hitting

“systemctl start docker “



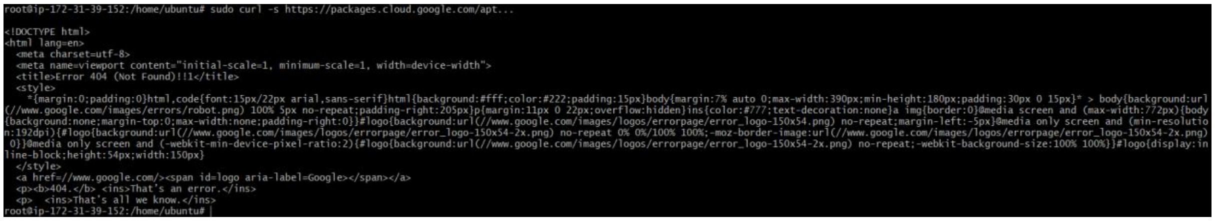
```
root@ip-172-31-39-152:/home/ubuntu# systemctl start docker
root@ip-172-31-39-152:/home/ubuntu# |
```

Enable the docker engine by hitting
“systemctl enable docker”



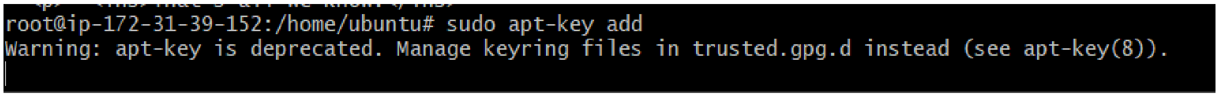
```
root@ip-172-31-39-152:/home/ubuntu# systemctl enable docker
root@ip-172-31-39-152:/home/ubuntu#
```

sudo curl -s https://packages.cloud.google.com/apt…



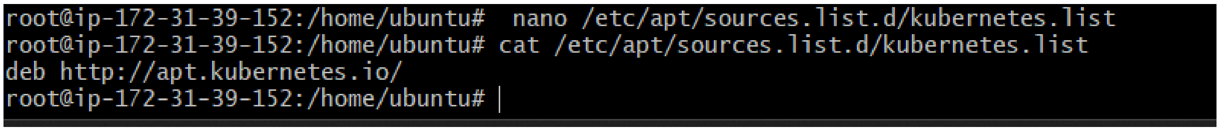
```
root@ip-172-31-39-152:/home/ubuntu# sudo curl -s https://packages.cloud.google.com/apt...

<!DOCTYPE html>
<html lang=en>
  <meta charset=utf-8>
  <meta name=viewport content="initial-scale=1, minimum-scale=1, width=device-width">
  <title>Error 404 (Not Found)!!1</title>
  <style>
    *{margin:0;padding:0}html,code{font:15px/22px arial,sans-serif}html{background:#fff;color:#222;padding:15px}body{margin:7% auto 0;max-width:390px;min-height:180px;padding:30px 0 15px}* > body{background:url
(//www.google.com/images/errors/robot.png) 100% 5px no-repeat;padding-right:205px}p{margin:11px 0 22px;overflow:hidden}ins{color:#777;text-decoration:none}a img{border:0}@media screen and (max-width:772px){body
{background:none;margin-top:0;max-width:none;padding-right:0}}#logo{background:url(//www.google.com/images/logos/errorpage/error_logo-150x54.png) no-repeat;margin-left:-5px}@media only screen and (min-resolutio
n:192dpi){#logo{background:url(//www.google.com/images/logos/errorpage/error_logo-150x54-2x.png) no-repeat 0% 0%/100% 100%;-moz-border-image:url(//www.google.com/images/logos/errorpage/error_logo-150x54-2x.png)
 0}}@media only screen and (-webkit-min-device-pixel-ratio:2){#logo{background:url(//www.google.com/images/logos/errorpage/error_logo-150x54-2x.png) no-repeat;-webkit-background-size:100% 100%}}#logo{display:in
line-block;height:54px;width:150px}
  </style>
  <a href=//www.google.com/><span id=logo aria-label=Google></span></a>
  <p><b>404.</b> <ins>That's an error.</ins>
  <p>  <ins>That's all we know.</ins>
root@ip-172-31-39-152:/home/ubuntu# |
```

sudo apt-key add



```
root@ip-172-31-39-152:/home/ubuntu# sudo apt-key add
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
|
```

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



```
root@ip-172-31-39-152:/home/ubuntu#  nano /etc/apt/sources.list.d/kubernetes.list
root@ip-172-31-39-152:/home/ubuntu# cat /etc/apt/sources.list.d/kubernetes.list
deb http://apt.kubernetes.io/
root@ip-172-31-39-152:/home/ubuntu# |
```

kubernetes-xenial main You need to follow the https://kubernetes.io/blog/2023/08/15/pkgs-k8s-io-introduction/ doc for downloading the Kubernetes Community-Owned Package RepositoriesTo install the kubernetes components.For Debian, Ubuntu, and operating systems using apt/apt-get

1. Replace the apt repository definition so that apt points to the new repository instead of the Google-hosted repository. Make sure to replace the Kubernetes minor version in the command below with the minor version that you're currently using:

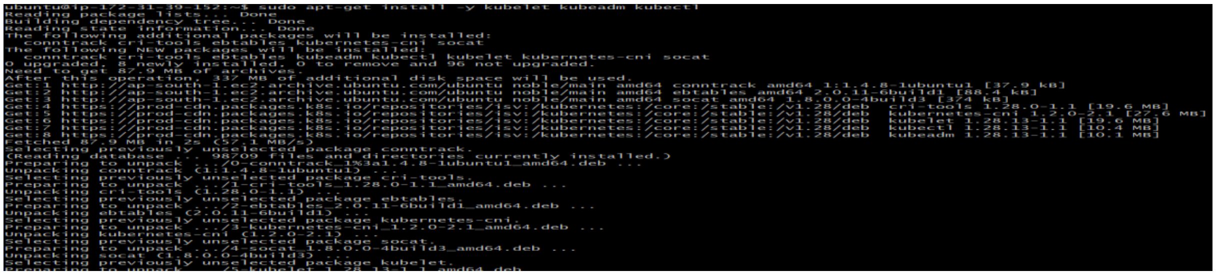
```
echo "deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.28/deb/ /" | sudo tee /etc/apt/sources.list.d/kubernetes.list
```

2. 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:/v1.28/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg*Update: In releases older than Debian 12 and Ubuntu 22.04, the folder /etc/apt/keyrings does not exist by default, and it should be created before the curl command.*

3. Update the apt package index:

```
sudo apt-get update
```

4.now install the kubernetes components by hitting “apt-get install -y kubelet kubeadm kubectl kubernetes-cni" you need to repeat all this steps in the rest 2 nodes as well.

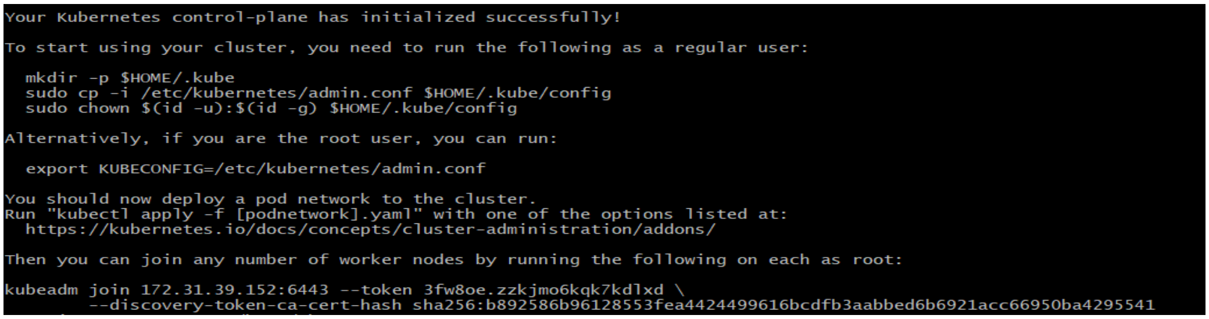


```
ubuntu@ip-172-31-39-152:~$ sudo apt-get install -y kubelet kubeadm kubectl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools ebtables kubernetes-cni socat
The following NEW packages will be installed:
  conntrack cri-tools ebtables kubeadm kubectl kubelet kubernetes-cni socat
0 upgraded, 8 newly installed, 0 to remove and 96 not upgraded.
Need to get 87.9 MB of archives.
After this operation, 337 MB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 conntrack amd64 1:1.4.8-1ubuntu1 [37.9 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 ebtables amd64 2.0.11-6build1 [88.4 kB]
Get:3 http://ap-south-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:/v1.28/deb  cri-tools 1.28.0-1.1 [19.6 MB]
Get:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.28/deb  kubernetes-cni 1.2.0-2.1 [27.6 MB]
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.28/deb  kubelet 1.28.13-1.1 [19.6 MB]
Get:7 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.28/deb  kubectl 1.28.13-1.1 [10.4 MB]
Get:8 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.28/deb  kubeadm 1.28.13-1.1 [10.1 MB]
Fetched 87.9 MB in 2s (57.1 MB/s)
Selecting previously unselected package conntrack.
(Reading database ... 98709 files and directories currently installed.)
Preparing to unpack .../0-conntrack_1%3a1.4.8-1ubuntu1_amd64.deb ...
Unpacking conntrack (1:1.4.8-1ubuntu1) ...
Selecting previously unselected package cri-tools.
Preparing to unpack .../1-cri-tools_1.28.0-1.1_amd64.deb ...
Unpacking cri-tools (1.28.0-1.1) ...
Selecting previously unselected package ebtables.
Preparing to unpack .../2-ebtables_2.0.11-6build1_amd64.deb ...
Unpacking ebtables (2.0.11-6build1) ...
Selecting previously unselected package kubernetes-cni.
Preparing to unpack .../3-kubernetes-cni_1.2.0-2.1_amd64.deb ...
Unpacking kubernetes-cni (1.2.0-2.1) ...
Selecting previously unselected package socat.
Preparing to unpack .../4-socat_1.8.0.0-4build3_amd64.deb ...
Unpacking socat (1.8.0.0-4build3) ...
Selecting previously unselected package kubelet.
Preparing to unpack .../5-kubelet_1.28.13-1.1_amd64.deb
```

3.BOOTSTRAPPING THE MASTER NODE (IN MASTER)

Insure your are using t2.medium ec2 type for master becuase it requires 2 vcpus.

Then hit below command in master.



```
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 172.31.39.152:6443 --token 3fw8oe.zzkjmo6kqk7kdlxd \
        --discovery-token-ca-cert-hash sha256:b892586b96128553fea4424499616bcdfb3aabbed6b6921acc66950ba4295541
```

Note:-you need to note hit those mentioned 3 commands to start using your cluster.

Note2:- hit this commnad in the node to connect your nodes with master
“kubeadm join 172.31.39.152:6443 --token 3fw8oe.zzkjmo6kqk7kdlxd \

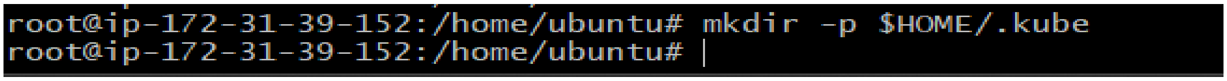
--discovery-token-ca-cert-hash"
sha256:b892586b96128553fea4424499616bcdfb3aabbed6b6921acc66950ba4295541”

“kubeadm init”

Note:-COPY THE COMMAND TO RUN IN NODES & SAVE IN NOTEPAD

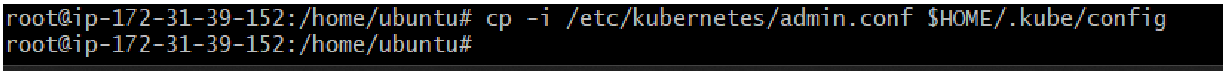
Now in master you need to run this below 3 command:-

1.mkdir -p $HOME/.kube



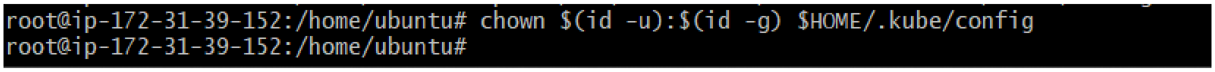
```
root@ip-172-31-39-152:/home/ubuntu# mkdir -p $HOME/.kube
root@ip-172-31-39-152:/home/ubuntu# |
```

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



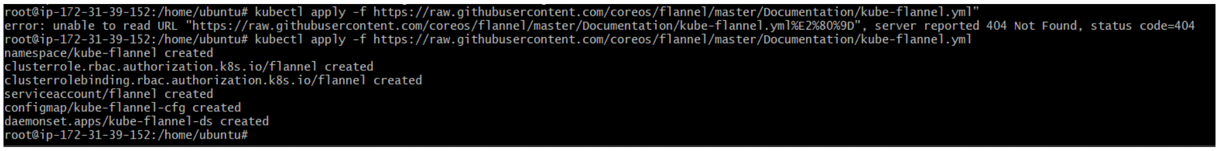
```
root@ip-172-31-39-152:/home/ubuntu# cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
root@ip-172-31-39-152:/home/ubuntu#
```

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



```
root@ip-172-31-39-152:/home/ubuntu# chown $(id -u):$(id -g) $HOME/.kube/config
root@ip-172-31-39-152:/home/ubuntu#
```

Now you need to hit below command to install flannel
“kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml”



```
root@ip-172-31-39-152:/home/ubuntu# kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml"
error: unable to read URL "https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml%E2%80%9D", server reported 404 Not Found, status code=404
root@ip-172-31-39-152:/home/ubuntu# kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
root@ip-172-31-39-152:/home/ubuntu#
```

kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/k8s-manifests/kube-flannel-rbac.yml

4.CONFIGURE WORKER NODES (IN NODES)

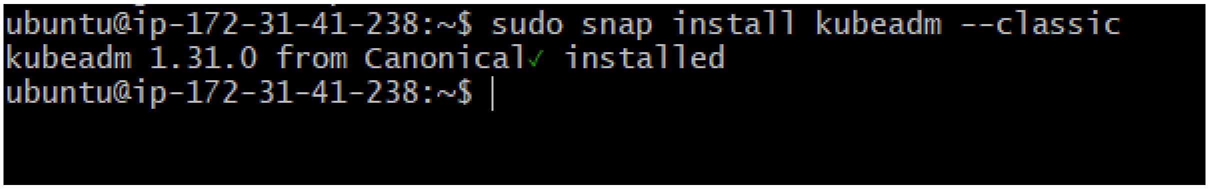
COPY LONG CODE PROVIDED BY MASTER IN NODE NOW LIKE CODE GIVEN BELOW

e.g- kubeadm join 172.31.6.165:6443 --token kl9fhu.co2n90v3rxtqllrs --discovery-token-ca-cert-hash sha256:b0f8003d23dbf445e0132a53d7aa1922bdef8d553d9eca06e65c928322b3e7c0

You need yo hit above command in the worker nodes to commnunicate your worker nodes with the master.

Note:-kudeadm is needed to be installed before proceeding further in the worker nodes.

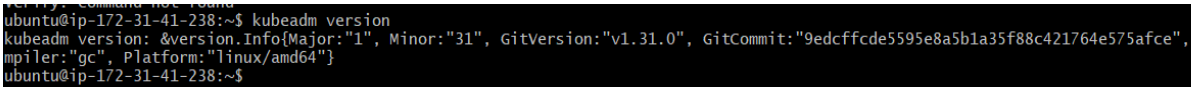
Install kubeadm with the --classic flag: (you need to follw below steps in the worker nodes).



```
ubuntu@ip-172-31-41-238:~$ sudo snap install kubeadm --classic
kubeadm 1.31.0 from Canonical✓ installed
ubuntu@ip-172-31-41-238:~$ |
```

Verify kubeadm installation:

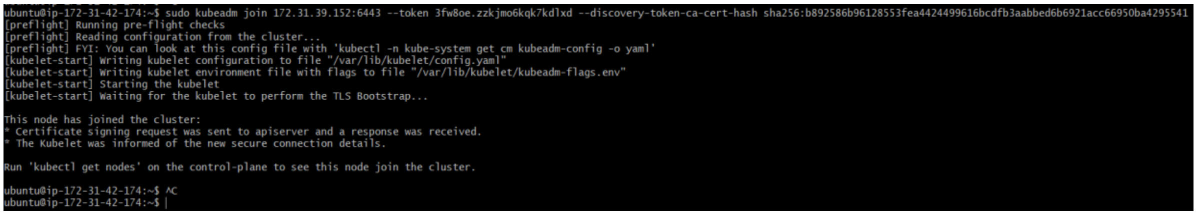
After installation, verify that kubeadm is correctly installed:



```
verify: command not found
ubuntu@ip-172-31-41-238:~$ kubeadm version
kubeadm version: &version.Info{Major:"1", Minor:"31", GitVersion:"v1.31.0", GitCommit:"9edcffcde5595e8a5b1a35f88c421764e575afce",
mpiler:"gc", Platform:"linux/amd64"}
ubuntu@ip-172-31-41-238:~$
```

Now as the kubeadm is been installed sucessfully on the worker nodes now you need to hit below command to communicate your master and worker nodes.

“kubeadm join 172.31.39.152:6443 --token 3fw8oe.zzkjmo6kqk7kdlxd --discovery-token-ca-cert-hash sha256: b892586b96128553fea4424499616bcdfb3aabbed6b6921acc66950ba4295541”



```
ubuntu@ip-172-31-42-174:~$ sudo kubeadm join 172.31.39.152:6443 --token 3fw8oe.zzkjmo6kqk7kdlxd --discovery-token-ca-cert-hash sha256:b892586b96128553fea4424499616bcdfb3aabbed6b6921acc66950ba4295541
[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-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-172-31-42-174:~$ ^C
ubuntu@ip-172-31-42-174:~$ |
```

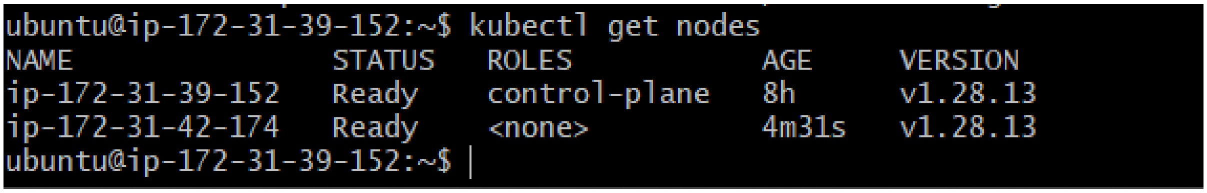
Above output denoted our worker node is now ready to communicate with the master node.

5.**Check the connections between worker and master nodes.**

You can cross check the same by hitting below command in the master node: -

Note: -you need to ensure whether the control plane components and kubelet is running in your master then you can check further.

“kubectl get nodes”



```
ubuntu@ip-172-31-39-152:~$ kubectl get nodes
NAME               STATUS   ROLES           AGE     VERSION
ip-172-31-39-152   Ready    control-plane   8h      v1.28.13
ip-172-31-42-174   Ready    <none>          4m31s   v1.28.13
ubuntu@ip-172-31-39-152:~$ |
```