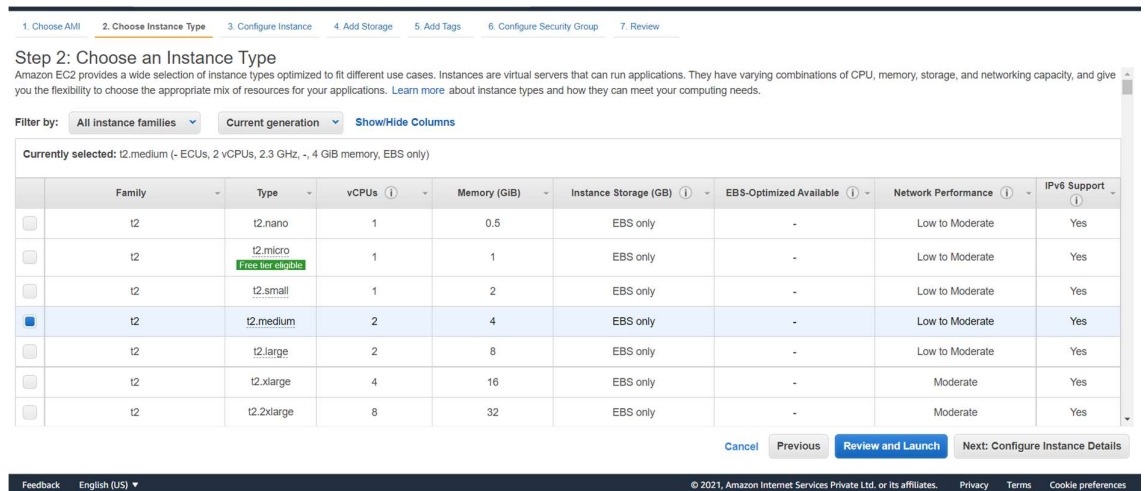
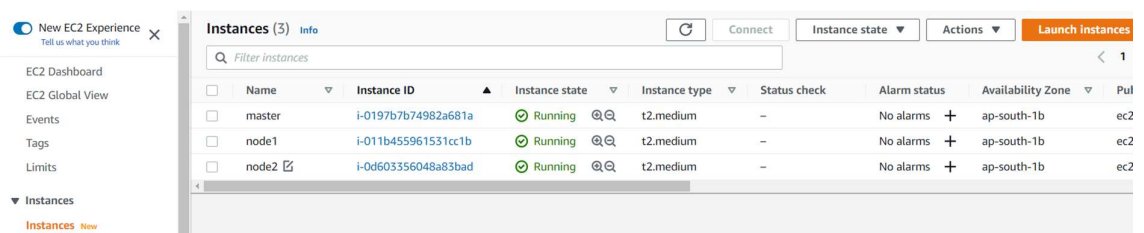


- Create 3 instance, jisme sein eak instance ko master banaunga aur 2 ko node
- Master banane ke liye minimum **2 virtual cpu** hona chahiye aur **4 gb ram** minimum hona chahiye humare machine mein isliye aws ki t2 micro kaam nahi aayega yaha, toh isliye liye hum **t2 medium** le lenge yaha.



- Now 1 master and 2 node, abhi mene 3 instance bana ke in bas usme sein eak decide kar diya ki eak master hai baki 2 node hai.



Master =

```
cmd Select root@ip-172-31-10-149: /home/ubuntu

C:\Users\user\OneDrive\Desktop>ssh -i satyam.pem ec2-user@3.109.212.92
ec2-user@3.109.212.92: Permission denied (publickey).

C:\Users\user\OneDrive\Desktop>ssh -i satyam.pem ubuntu@3.109.212.92
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1020-aws x86_64)

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

System information as of Wed Nov 24 12:49:03 UTC 2021
System load:  0.23          Processes:    114
Usage of /:   17.7% of 7.69GB Users logged in: 0
Memory usage: 5%          IPv4 address for eth0: 172.31.10.149
Swap usage:   0%

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

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-172-31-10-149:~$ sudo su
root@ip-172-31-10-149:/home/ubuntu#
```

Node 1 = Grey terminal

```
cmd ubuntu@ip-172-31-8-219: ~
Microsoft Windows [Version 10.0.19043.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user\OneDrive\Desktop>ssh -i satyam.pem ubuntu@13.234.75.250
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1020-aws x86_64)

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

System information as of Wed Nov 24 13:21:00 UTC 2021
System load:  0.0          Processes:    109
Usage of /:   17.9% of 7.69GB Users logged in: 0
Memory usage: 5%          IPv4 address for eth0: 172.31.8.219
Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

  https://ubuntu.com/aws/pro

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

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

Last login: Wed Nov 24 12:53:36 2021 from 122.160.34.213
ubuntu@ip-172-31-8-219:~$
```

Node 2 = yellow terminal

```
cmd Select ubuntu@ip-172-31-10-149: ~
C:\Users\user\OneDrive\Desktop>ssh -i satyam.pem ubuntu@3.109.212.92
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1020-aws x86_64)

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

System information as of Wed Nov 24 13:15:31 UTC 2021
System load:  0.0          Processes:    110
Usage of /:   17.9% of 7.69GB Users logged in: 0
Memory usage: 5%          IPv4 address for eth0: 172.31.10.149
Swap usage:   0%

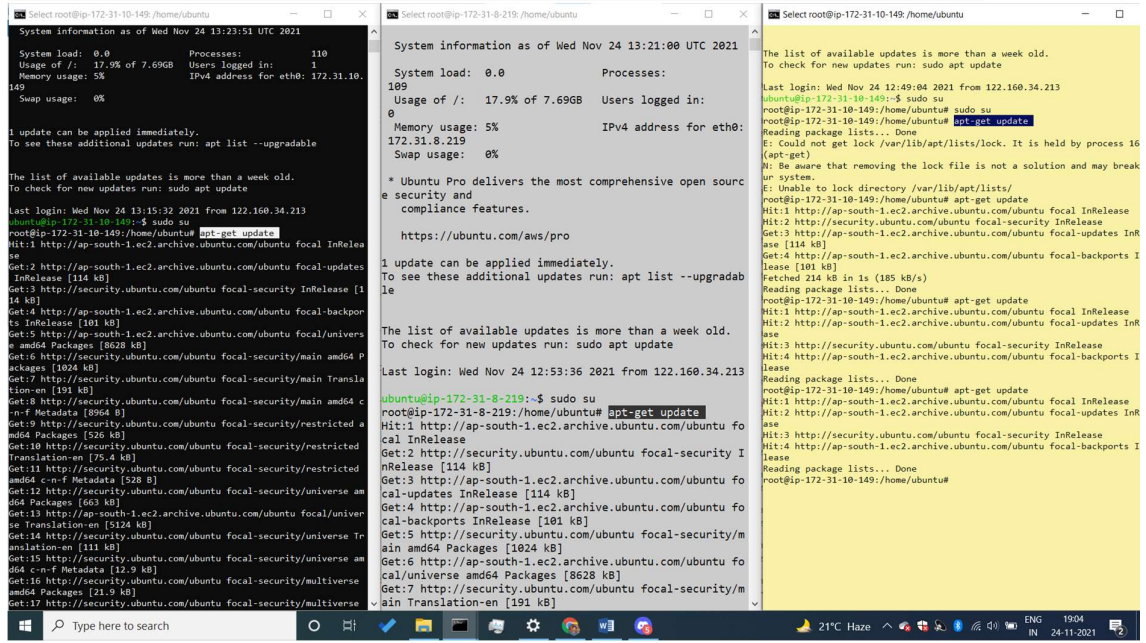
1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

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

Last login: Wed Nov 24 12:49:04 2021 from 122.160.34.213
ubuntu@ip-172-31-10-149:~$
```


➔ apt-get update

Master



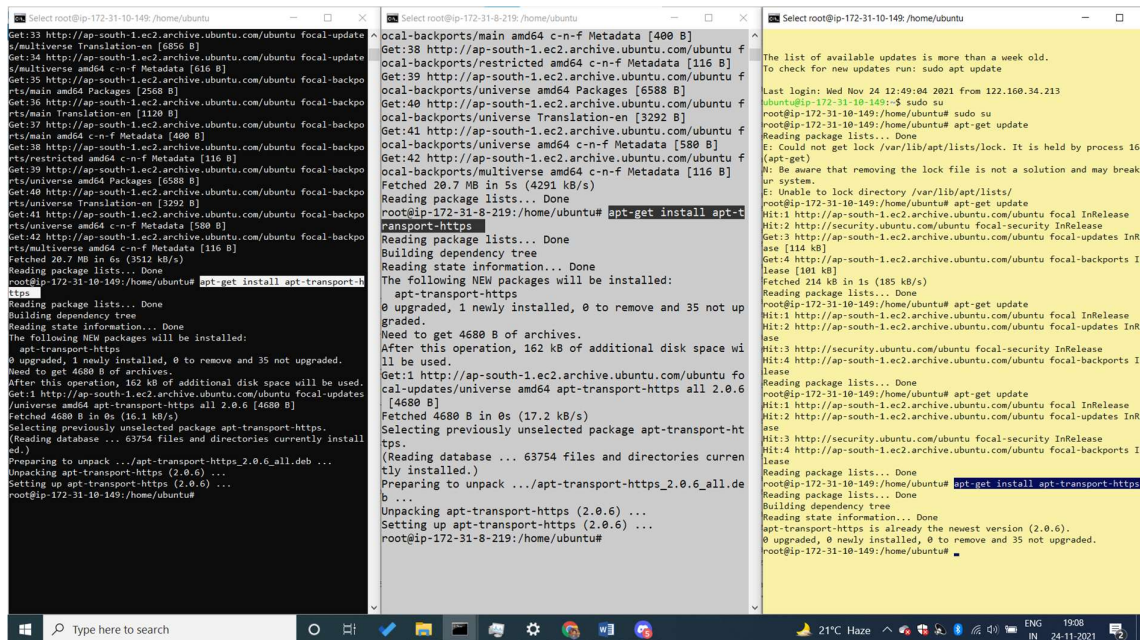
```
➔ apt-get install apt-transport-https
```

➤ agar humko secure level par kuch karna hai toh uske liye humein **https** install karna hoga.

Master

Node1

Node2



➔ apt install docker.io -y

➤ docker install (humane master aur worker node mein sab mein docker install kar diya hai)

Master	Node1	Node2
<pre>root@ip-172-31-10-149:/home/ubuntu# apt install docker.io -y Reading package lists... Done Building dependency tree Reading state information... Done The following additional packages will be installed: bridge-utils containerd dns-root-data dnsmasq-base libidn11 pigz runc ubuntu-fan Suggested packages: ifupdown aufs-tools cgroupfs-mount cgroup-lite debotstrap docker-doc rinse zfs-fuse zfsutils The following NEW packages will be installed: bridge-utils containerd dns-root-data dnsmasq-base docker.io libidn11 pigz runc ubuntu-fan 0 upgraded, 9 newly installed, 0 to remove and 35 not upgraded. Need to get 74.5 MB of archives. After this operation, 361 MB of additional disk space will be used. Do you want to continue? [Y/n] y Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 pigz amd64 2.4-1 [57.4 kB] Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 bridge-utils amd64 1.6-2ubuntu1 [30.5 kB] Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 runc amd64 1.0.1-0ubuntu2-20.04.1 [4155 kB] Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 containerd amd64 1.5.5-0ubuntu3-20.04.1 [33.0 MB] Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 dns-root-data all 2019052802 [5300 B] Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 libidn11 amd64 1.33-2.ubuntu2 [46.2 kB] Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 dnsmasq-base amd64 2.80-1.1ubuntu1.4 [315 kB] Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 docker.io amd64 20.10.7-0ubuntu5-20.04.2 [36.9 MB] Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 ubuntu-fan all 0.12-13 [34.5 kB] Fetched 74.5 MB in 16s (4551 kB/s) Preconfiguring packages ... Selecting previously unselected package pigz. (Reading database ... 63758 files and directories currently installed.) Preparing to unpack .../0-pigz_2.4-1_amd64.deb ... Unpacking pigz (2.4-1) ... Selecting previously unselected package bridge-utils. Preparing to unpack .../1-bridge-utils_1.6-2ubuntu1_amd64.deb ... Unpacking bridge-utils (1.6-2ubuntu1) ... Selecting previously unselected package runc. Preparing to unpack .../2-runc_1.0.1-0ubuntu2-20.04.1_amd64.deb ... Unpacking runc (1.0.1-0ubuntu2-20.04.1) ... Selecting previously unselected package containerd. Preparing to unpack .../3-containerd_1.5.5-0ubuntu3-20.04.1_amd64.deb ... Unpacking containerd (1.5.5-0ubuntu3-20.04.1) ... Preconfiguring packages ...</pre>	<pre>root@ip-172-31-8-219:/home/ubuntu# apt install docker.io -y Reading package lists... Done Building dependency tree Reading state information... Done The following additional packages will be installed: bridge-utils containerd dns-root-data dnsmasq-base libidn11 pigz runc ubuntu-fan Suggested packages: ifupdown aufs-tools cgroupfs-mount cgroup-lite debotstrap docker-doc rinse zfs-fuse zfsutils The following NEW packages will be installed: bridge-utils containerd dns-root-data dnsmasq-base docker.io libidn11 pigz runc ubuntu-fan 0 upgraded, 9 newly installed, 0 to remove and 35 not up graded. Need to get 74.5 MB of archives. After this operation, 361 MB of additional disk space wi ll be used. Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu fo cal/universe amd64 pigz amd64 2.4-1 [57.4 kB] Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu fo cal/main amd64 bridge-utils amd64 1.6-2ubuntu1 [30.5 kB] Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu fo cal-updates/main amd64 runc amd64 1.0.1-0ubuntu2-20.04.1 [4155 kB] Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu fo cal-updates/main amd64 containerd amd64 1.5.5-0ubuntu3-2 0.04.1 [33.0 MB] Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu fo cal/main amd64 dns-root-data all 2019052802 [5300 B] Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu fo cal/main amd64 libidn11 amd64 1.33-2.ubuntu2 [46.2 kB] Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu fo cal-updates/main amd64 dnsmasq-base amd64 2.80-1.1ubunt u 1.4 [315 kB] Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu fo cal-updates/universe amd64 docker.io amd64 20.10.7-0ubun tu5-20.04.2 [36.9 MB] Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu fo cal/main amd64 ubuntu-fan all 0.12.13 [34.5 kB] Fetched 74.5 MB in 7s (10.5 MB/s) Preconfiguring packages ...</pre>	<pre>root@ip-172-31-10-149:/home/ubuntu# apt install docker.io -y Reading package lists... Done Building dependency tree Reading state information... Done The following additional packages will be installed: bridge-utils containerd dns-root-data dnsmasq-base libidn11 pigz runc ubuntu-fan Suggested packages: ifupdown aufs-tools cgroupfs-mount cgroup-lite debotstrap docker-doc rinse zfs-fuse zfsutils The following NEW packages will be installed: bridge-utils containerd dns-root-data dnsmasq-base docker.io libidn11 pigz runc ubuntu-fan 0 upgraded, 9 newly installed, 0 to remove and 35 not upgraded. Need to get 74.5 MB of archives. After this operation, 361 MB of additional disk space will be used. Do you want to continue? [Y/n] y Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 pigz amd64 2.4-1 [57.4 kB] Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 bridge-utils amd64 1.6-2ubuntu1 [30.5 kB] Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 runc amd64 1.0.1-0ubuntu2-20.04.1 [4155 kB] Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 containerd amd64 1.5.5-0ubuntu3-20.04.1 [33.0 MB] Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 dns-root-data all 2019052802 [5300 B] Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 libidn11 amd64 1.33-2.ubuntu2 [46.2 kB] Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 dnsmasq-base amd64 2.80-1.1ubuntu1.4 [315 kB] Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 docker.io amd64 20.10.7-0ubuntu5-20.04.2 [36.9 MB] Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 ubuntu-fan all 0.12.13 [34.5 kB] Fetched 74.5 MB in 7s (10.5 MB/s) Preconfiguring packages ...</pre>

➔ systemctl start docker

(mene docker install karne ke baad docker ko start kiya hai)

Master	Node1	Node2
<pre>root@ip-172-31-10-149:/home/ubuntu# systemctl start docker</pre>	<pre>root@ip-172-31-8-219:/home/ubuntu# systemctl start docker</pre>	<pre>root@ip-172-31-10-149:/home/ubuntu# systemctl start docker</pre>

➔ systemctl enable docker

(docker ko enable kar diya hai)

Master	Node1	Node2
<pre>root@ip-172-31-10-149:/home/ubuntu# systemctl enable docker</pre>	<pre>root@ip-172-31-8-219:/home/ubuntu# systemctl enable docker</pre>	<pre>root@ip-172-31-10-149:/home/ubuntu# systemctl enable docker</pre>

- ➔ GPG KEY = Master ko jab bhi node sein baat karna ho toh dono kein pass GPG key hona chahiye jisase NODE ko ye confirm ho ki jo samne vala banda matlab master hai uske pass bhi same key hai authentication kein liye agar key match ho rahi toh conversastion ho jayegi master aur worker node mein,
- ➔ GPG key ka use karne kein liye humko gpg key kein package ko install karna hoga .
- ➔ GPG-key ko install karne kein liye hum ye command use karenge , hum link kein andar dekhenge ki eak package hai jo ki hum google.com sen uthaa rahe hai , aur humane directory bhi bata diya hai ki iss directory mein package rakha hai yaha sein jaake uthaa lo, aur last mein mene sudo apt-key add isliye likha hai ki meri ye key master par bhi laga do aur worker node par bhi attach kar do .

➔ `sudo curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add`

The image shows three terminal windows side-by-side, each representing a different Ubuntu node. The first window (left) shows the command `sudo curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add` being executed successfully. The second window (middle) shows the same command being executed on a node with IP 172-31-8-219. The third window (right) shows the command being executed on a node with IP 172-31-10-149, with the output of the command visible in the background.

Install kubernetes

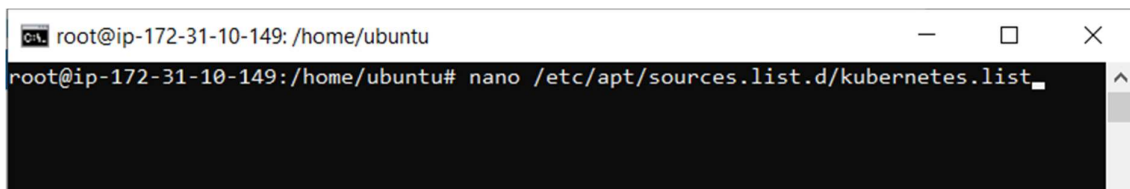
- ➔ Kubernetes ko install karane kein liye sabse pehale mene eak file banayi hai nano ki help sein, us file kein andar mene link dala hai jaha sein humko kubernetes ka package download karna hai, us link kein through humko us file kein andar download karna hai kubernetes kein package
- ➔ nano command hum tab use karte hai jab humko koi file bannai hai aur us file kein andar humko agar kuch edit karna hai toh vaha hum nano command use karte hai , aur hum command mein dekhenge ki humane eak path diya hai aur path kein andar eak source.list.d kein naam sein eak file hogi uske andar hum jaake **kubernetes.list** kein naam sein eak file create karenge .

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

-
- ➔ aur nano file jab open ho jaye toh uske andar hum yein command daal denge , link mein **deb** ka matlab hai **debian family** ka use kare aur deb kein baad mene kubernetes ki **official website** di hai, aur us link par jane kein baaad mene eak package ka naam likha hai jiska naam hai **kubernetes-xenial** download kar le aur uske package kein andar bhi huko **main** vali file download karna hai, matlab main vala package download karna hai, command likhane kein baad file ko save kar dena hai , file save karne kein liye **ctrl+x => ctrl+y => enter**

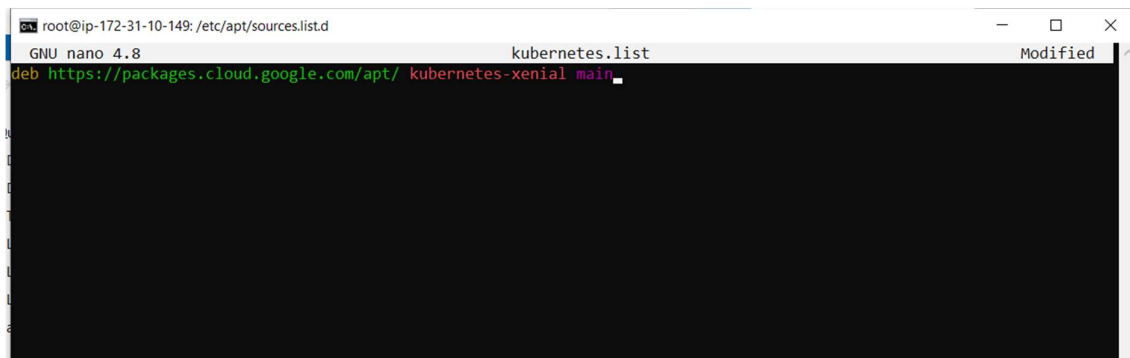
- **deb** [https://packages.cloud.google.com/apt/](https://packages.cloud.google.com/apt/kubernetes-xenial) **kubernetes-xenial main**

master



```
root@ip-172-31-10-149: /home/ubuntu
root@ip-172-31-10-149:/home/ubuntu# nano /etc/apt/sources.list.d/kubernetes.list
```

Master



```
root@ip-172-31-10-149: /etc/apt/sources.list.d
GNU nano 4.8 kubernetes.list Modified
deb https://packages.cloud.google.com/apt/ kubernetes-xenial main
```

Node1

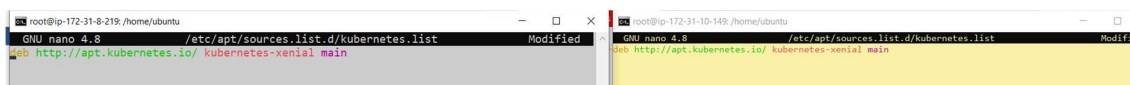
Node2



```
root@ip-172-31-8-219:/home/ubuntu nano /etc/apt/sources.list.d/kubernetes.list
root@ip-172-31-10-149:/home/ubuntu nano /etc/apt/sources.list.d/kubernetes.list
```

Node1

Node2



```
root@ip-172-31-8-219:/home/ubuntu GNU nano 4.8 /etc/apt/sources.list.d/kubernetes.list Modified
deb http://apt.kubernetes.io/ kubernetes-xenial main
root@ip-172-31-10-149:/home/ubuntu GNU nano 4.8 /etc/apt/sources.list.d/kubernetes.list Modified
deb http://apt.kubernetes.io/ kubernetes-xenial main
```

-
- ➔ Now we want to update our packages, iska matlab jo humane upar **kubernetes-xenial** naam ka package download kiya thaa vo install ho jayega
- ➔ apt-get update hum yaha isliye likhe hai ki upar mene jo kubernetes-xenial file dali hai usko unzip kar do.

apt-get update

```
root@ip-172-31-10-149:/home/ubuntu# apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal InRelease [114 kB]
Get:2 https://packages.cloud.google.com/apt/kubernetes-xenial InRelease [9383 B]
Hit:3 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:5 https://packages.cloud.google.com/apt/kubernetes-xenial InRelease [9383 B]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:7 https://packages.cloud.google.com/apt/kubernetes-xenial InRelease [9383 B]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [876 kB]
Fetched 1265 kB in 2s (718 kB/s)
Reading package lists... Done
root@ip-172-31-10-149:/home/ubuntu#

root@ip-172-31-8-219:/home/ubuntu# apt-get update
Hit:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal InRelease [114 kB]
Hit:3 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [876 kB]
Get:7 https://packages.cloud.google.com/apt/kubernetes-xenial InRelease [9383 B]
Get:8 https://packages.cloud.google.com/apt/kubernetes-xenial/main amd64 Packages [51.6 kB]
Fetched 1265 kB in 2s (798 kB/s)
Reading package lists... Done
root@ip-172-31-8-219:/home/ubuntu#

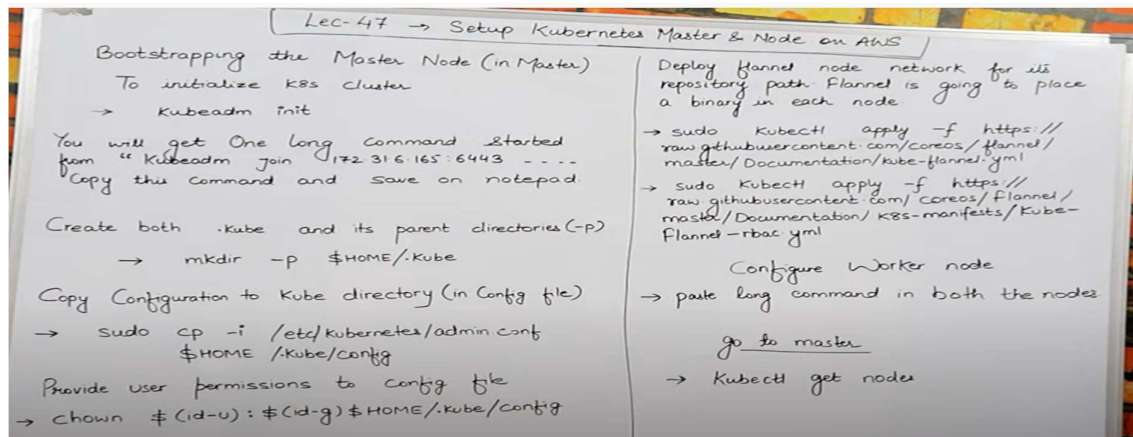
root@ip-172-31-10-149:/home/ubuntu# apt-get update
Hit:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal InRelease [114 kB]
Hit:3 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [876 kB]
Get:7 https://packages.cloud.google.com/apt/kubernetes-xenial InRelease [9383 B]
Get:8 https://packages.cloud.google.com/apt/kubernetes-xenial/main amd64 Packages [51.6 kB]
Fetched 1265 kB in 2s (798 kB/s)
Reading package lists... Done
root@ip-172-31-10-149:/home/ubuntu#
```

➔ Jab humara package unzip ho jayega uske andar bhi humko bahut saree packages hai jo ki install karna jaruri hai

➔ apt-get install -y kubelet kubeadm kubectl kubernetes-cni

```
root@ip-172-31-10-149:/home/ubuntu# apt-get install -y kubelet kubeadm kubectl kubernetes-cni
Reading package lists... Done
Building dependency tree
Reading state information... Done
kubeadm is already the newest version (1.22.4-00).
kubectl is already the newest version (1.22.4-00).
kubernetes-cni is already the newest version (0.8.7-00).
0 upgraded, 0 newly installed, 0 to remove and 35 not upgraded.
root@ip-172-31-10-149:/home/ubuntu#

root@ip-172-31-8-219:/home/ubuntu# apt-get install -y kubelet kubeadm kubectl kubernetes-cni
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools ebtables socat
Suggested packages:
  nftables
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 35 not upgraded.
Need to get 73.5 MB of archives.
After this operation, 318 MB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 conntrack amd64 1:1.4.5-2 [30.3 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 cri-tools amd64 1.19.0-00 [11.2 MB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 ebtables amd64 2.0.11-3build1 [80.3 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 socat amd64 1.7.3-3.2 [323 kB]
Get:5 https://packages.cloud.google.com/apt/kubernetes-xenial/main amd64 kubernetes-cni amd64 0.8.7-00 [25.0 MB]
Get:6 https://packages.cloud.google.com/apt/kubernetes-xenial/main amd64 kubelet amd64 1.22.4-00 [19.1 MB]
Get:7 https://packages.cloud.google.com/apt/kubernetes-xenial/main amd64 kubectl amd64 1.22.4-00 [3957 kB]
Get:8 https://packages.cloud.google.com/apt/kubernetes-xenial/main amd64 kubeadm amd64 1.22.4-00 [8732 kB]
Fetched 73.5 MB in 4s (17.2 MB/s)
Selecting previously unselected package conntrack.
(Reading database ... 64112 files and directories currently installed.)
Preparing to unpack .../0-conntrack_1:1.4.5-2_amd64.deb ...
Unpacking conntrack (1:1.4.5-2) ...
Selecting previously unselected package cri-tools.
Preparing to unpack .../1-cri-tools_1.19.0-00_amd64.deb ...
Unpacking cri-tools (1.19.0-00) ...
Selecting previously unselected package ebtables.
Preparing to unpack .../2-ebtables_2.0.11-3build1_amd64.deb ...
Unpacking ebtables (2.0.11-3build1) ...
Selecting previously unselected package kubernetes-cni.
Preparing to unpack .../3-kubernetes-cni_0.8.7-00_amd64.deb ...
Unpacking kubernetes-cni (0.8.7-00) ...
```



➔ Bootstrapping ka matlab hai master ko humare node kein sath connect karna, matlab master aur node kein beech mein connection karna aur path ko hi hum bootstrapping kehate hai.

➔ Create master node

➔ Sabse pehale master aur worker node ko jodane kein liye humko master node banana padega, aur master node banana kein liye hum master ko initialize karenge, aur ye command chalayenge, **aur initialize hone kein baad eak lamba sa code aayega jisko hum copy karke rakh lenge apane notepad mein**

➔ Master banana kein liye minimum 2 virtual cpu hona chahiye aur 4 gb ram minimum hona chahiye humare machine mein isliye aw ski t2 micro kaam nahi aayega yaha, toh iske liye hum t2 medium le lenge yaha

➤ kubeadm init :-

```
root@ip-172-31-10-87:/home/ubuntu# kubeadm init
[init] Using Kubernetes version: v1.22.4
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This may 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'
[certs] Using certificate 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-172-31-10-87 kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.1.0.1]
[certs] Generating "apiserver-kubelet-client" 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-172-31-10-87 localhost] and IPs [172.31.10.87 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-10-87 localhost] and IPs [172.31.10.87 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
[kubeconfig] Using kubeconfig folder /etc/kubernetes
[kubeconfig] Writing "admin.conf" kubeconfig file
[kubeconfig] Writing "kubeadm.conf" kubeconfig file
[kubeconfig] Writing "controller-manager.conf" kubeconfig file
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Starting the kubelet
[control-plane] Using manifest folder "/etc/kubernetes/manifests"
[control-plane] Creating static Pod manifest for "kube-apiserver"
[control-plane] Creating static Pod manifest for "kube-controller-manager"
[control-plane] Creating static Pod manifest for "kube-scheduler"
[etcd] Creating static Pod manifest for local etcd in "/etc/kubernetes/manifests"
[wait-control-plane] Waiting for the kubelet to boot up the control plane as static Pods from directory "/etc/kubernetes/manifests". This can take up to 4m0s
[upload-certs] Storing the configuration used in ConfigMap "kubeadm-config" in the "kube-system" Namespace
[mark-control-plane] Marking the node ip-172-31-10-87 as control-plane by adding the labels: [node-role.kubernetes.io/master(deprecated) node-role.kubernetes.io/control-plane]
[bootstrap-token] Creating the "kubeadm-bootstrap-token" ConfigMap in the "kube-public" namespace
[bootstrap-token] Creating the "kubeadm-kubelet-token" ConfigMap in the "kube-public" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key
[mark-master-done] Marking the node ip-172-31-10-87 as control-plane by adding the labels: [node-role.kubernetes.io/master(deprecated) node-role.kubernetes.io/control-plane]
[mark-master-done] Marking the node ip-172-31-10-87 as control-plane by adding the labels: [node-role.kubernetes.io/master(deprecated) node-role.kubernetes.io/control-plane]
```

➤ **key:-** ye jo key humko millegi isko hum save karke rakh lenge aur hum is key ko apane worker node mein dalenge hum and tab humari master jaa kein worker node sein connect ho jayegi.

```
[kubeadm] Creating a ConfigMap "kubeadm-config:1.22" in namespace kube-system with the configuration for the kubelets in the cluster
[upload-certs] Skipping phase, please see --upload-certs
[mark-control-plane] Marking the node ip-172-31-10-87 as control-plane by adding the labels: [node-role.kubernetes.io/master(deprecated) node-role.kubernetes.io/control-plane]
[bootstrap-token] Creating the "kubeadm-bootstrap-token" ConfigMap in the "kube-public" namespace
[bootstrap-token] Creating the "kubeadm-kubelet-token" ConfigMap in the "kube-public" namespace
[bootstrap-token] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
[bootstrap-token] configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate credentials
[bootstrap-token] configured RBAC rules to allow the kubelet to retrieve long term certificate credentials
[bootstrap-token] configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstrap-token] Creating the "kubeadm-kubelet-token" ConfigMap in the "kube-public" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key
[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 172.31.10.87:6443 --token 2h0exa:1u912fuf4e0y7o \
  --discovery-token-ca-cert-hash sha256:26a785c45f78304f2a09fe06810b753e2aade43d43903bf9abaf11d34e
root@ip-172-31-10-87:/home/ubuntu#
```


Save key :-

```
kubeadm join 172.31.10.87:6443 --token e1imnk.nq72dd3nsshcg3c --discovery-token-ca-cert-hash sha256:d03f04cf494f6f03d72b98788fd41960536f58112952cad459a97bccb43b8e75
```

- hum jab key dalenge toh beech mein slash/ hata denge aur kewal eak space denge beech mein .
- Agar kubeadm init karne mein key generate nahi ho rahi aur error aa raha ho toh hum ye command chalayenge phir uske baad kubeadm init karenge.
- I am facing an error while performing lab and using the command kubeadm init. And solved the problem by using the following steps/commands:

`vim /etc/docker/daemon.json`

```
{  
  "exec-opts": ["native.cgroupdriver=systemd"]  
}
```

`sudo systemctl daemon-reload`

`sudo systemctl restart docker`

`sudo systemctl restart kubelet`

`sudo kubeadm init`

- Humara master node jab initialize hojaye toh uske baad hum kuch command chalayenge jo ki humko initialize hone ke baad waha dikhengei , humko ye 3 command dikhegi console par initialize hone ke baad

➤ 1 st command = `mkdir -p $HOME/.kube`

Is command mein hum parent directory create karenge isliye mene `-p` likha hai ki parent ke liye eak directory create kar do

```
root@ip-172-31-10-87:/home/ubuntu# mkdir -p $HOME/.kube  
root@ip-172-31-10-87:/home/ubuntu#
```

➤ 2 nd command = `cp -i /etc/kubernetes/admin.conf $HOME/.kube/config`

Is command mein hum jo upar jo `.kube` naam ki directory banaya hai uske andar hum copy karenge , matlab kubernetes ke andar jo `admin.conf` naam ki file hai `(/etc/kubernetes/admin.conf)` usko hum copy karenge is path par `(HOME/.kube/config)` matlab `.kube` mein jaake config ke andar copy ho jayegi `Admin.conf` aur `$HOME` ke pehle space bhi hoga.

```
root@ip-172-31-10-87:/home/ubuntu# sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config  
root@ip-172-31-10-87:/home/ubuntu#
```

➤ 3 rd command = `chown $(id -u):$(id -g) $HOME/.kube/config`

Provide user permission to config file, matlab agar humko user ko kya permission dena hai isliye mene likha hai starting mein **chown means change owner permission** iska matlab user kaun si permission change kar payega kaun sid nahi humko vo dikha dega jab command chalayenge aur vo by default kubernetes kar bhi deta hai. Aur ye 3 ki 3 sab command master mein hi run karenge.

```
root@ip-172-31-10-87:/home/ubuntu# sudo chown $(id -u):$(id -g) $HOME/.kube/config
root@ip-172-31-10-87:/home/ubuntu#
```

Run those 2 commands:-

kubectl apply -f <https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml>

```
root@ip-172-31-10-87:/home/ubuntu# kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
Warning: policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
podsecuritypolicy.policy/psp.flannel.unprivileged 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
```

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

```
root@ip-172-31-10-87:/home/ubuntu# kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/k8s-manifests/kube-flannel-rbac.yml
clusterrole.rbac.authorization.k8s.io/flannel configured
clusterrolebinding.rbac.authorization.k8s.io/flannel unchanged
root@ip-172-31-10-87:/home/ubuntu#
```

- ➔ Ab mene jo upar master initialize karte samaye jo code copy karke notepad mein rakha tha usko uthaunga aur us code ko le jake apne node mein daal dunga jisase humara master aur worker node sab eak sath connect ho jayenge, aur isko hi hum bootstrapping kehate hai .
- ➔ Aba agar mujhe check karna ho ki mera master kein sath mera worker node connect huva hai ya nahi uske liye hum ye command likhenge.

worker node :- run this command in all the node those you want to connect to master.sp

vim /etc/docker/daemon.json

```
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}
```

sudo systemctl daemon-reload

sudo systemctl restart docker

sudo systemctl restart kubelet

kubeadm join 172.31.10.87:6443 --token e1imnk.nq72dd3nsshhcg3c --discovery-token-ca-cert-hash sha256:d03f04cf494f6f03d72b98788fd41960536f58112952cad459a97bccb43b8e75

GO TO MASTER AND RUN THIS COMMAND :-

➤ **kubectl get nodes**

- ➔ aur hum master mein jake search karenge toh hum dekheneghe ki humare master kein sath mene jitani node connect ki thii humane vo sab node connected hai . aur unki private ip bhi mujhe show ho rahi hai.
