

Module 7: Kubernetes Assignment - 1

Assignment Submitted By:-Hitesh Chauhan

Course Offered: -Advanced Cloud Computing and Devops

Assignment By: -Intellipaat

Trainer: -Kumar

Date Of Submission: -09/03/2025

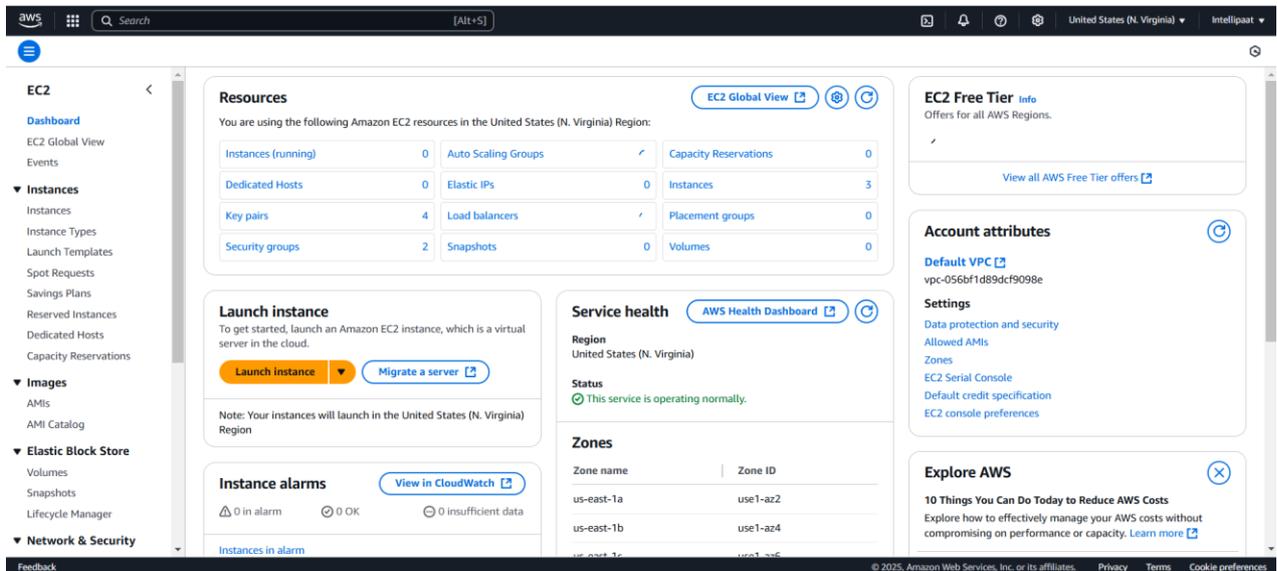
Tasks To Be Performed:

1. Deploy a Kubernetes cluster for 3 nodes
2. Create a NGINX deployment of 3 replicas

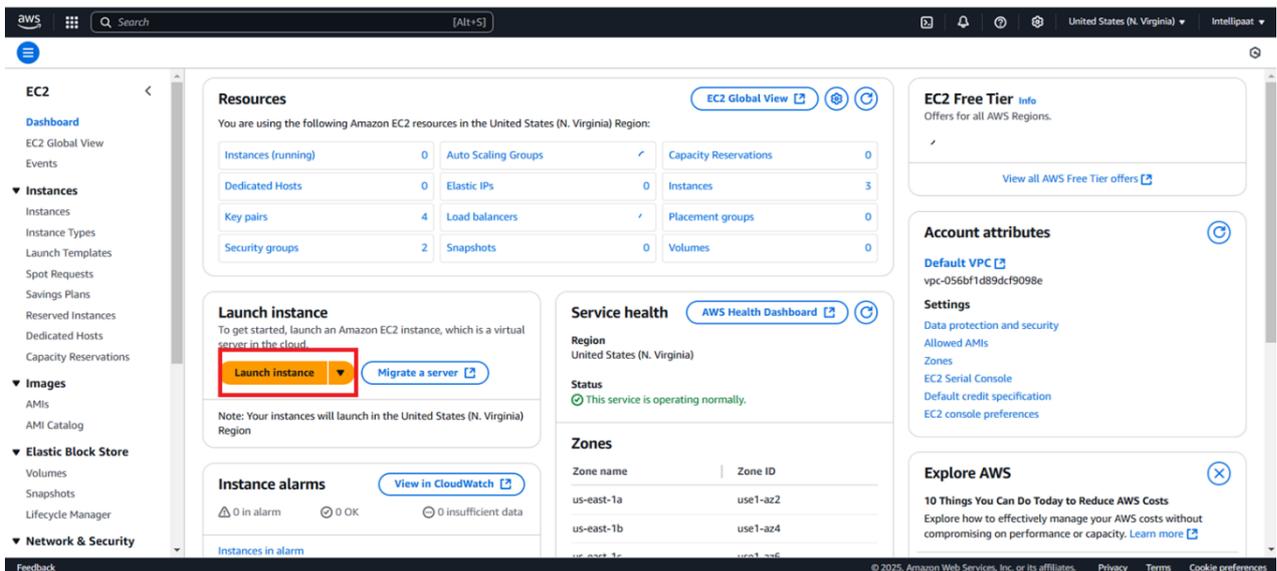
Solution:

First Go to aws management console.

1. Deploy a Kubernetes cluster for 3 nodes

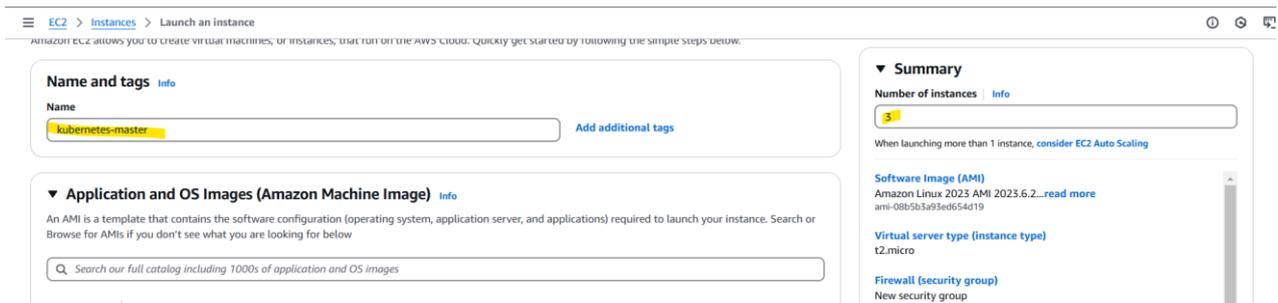


Click Launch Instance

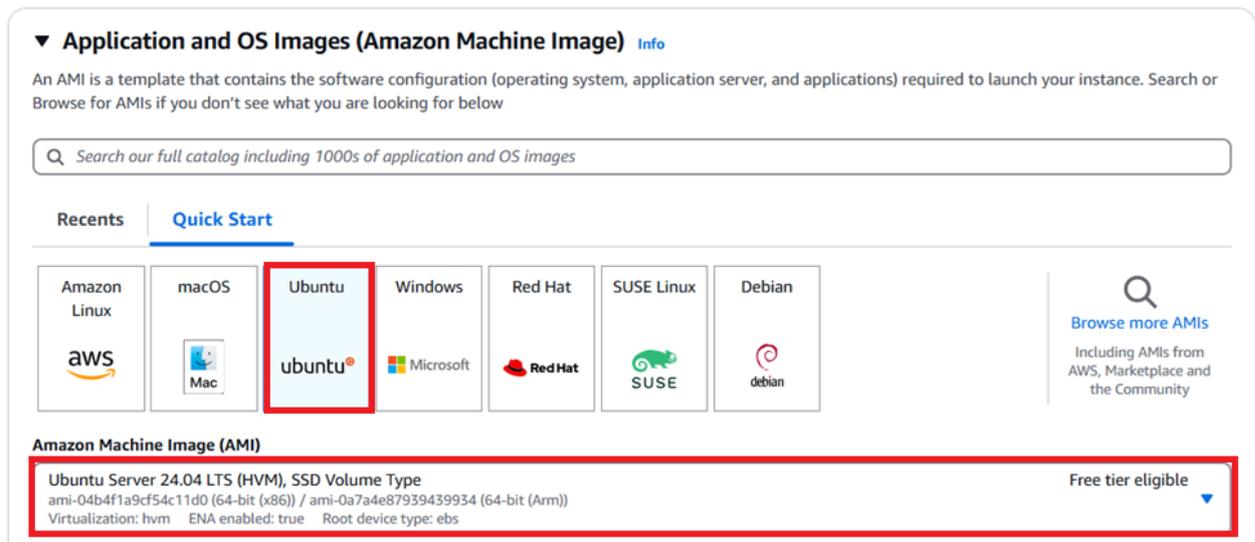


After Launch Instance

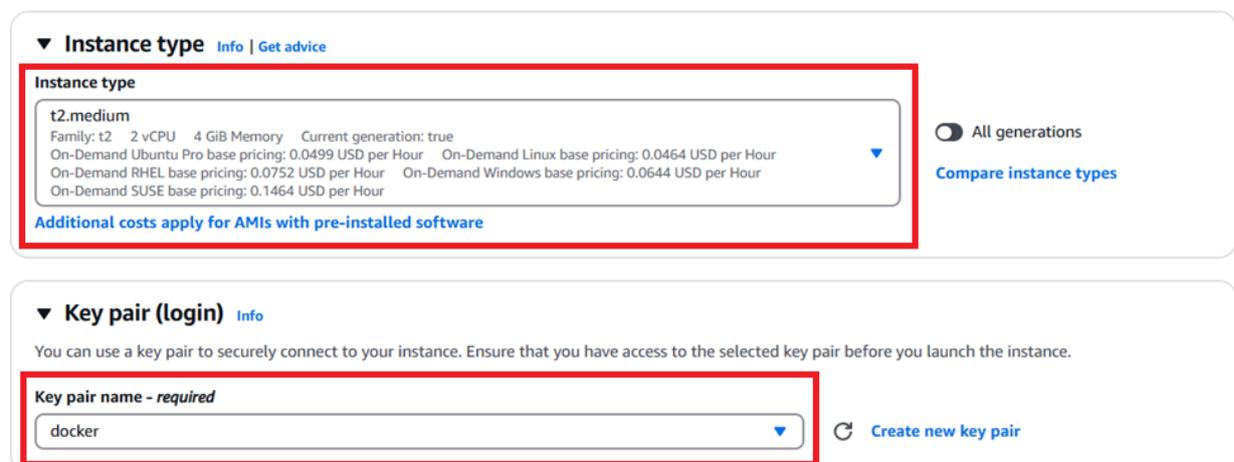
Assign the Name is require because if not assign then got confused.in my case I name assign like Kubernetes-master and summary you will see the number instances you need choose 3.in my case I choosed as 3 instance.



Then in Application and OS Images need to choose ubuntu and Amazon Machine Image(AMI) Will be Ubuntu Server 24.04 LTS Version



Then instance Type Will be t2.medium and keypair is docker



NOTE:-The t2.medium is not a free instance.it is chargeable instance you will got charge per hour 0.0499 USD in ubuntu Pro base. Now we need to convert the doller to Indian rupees 4.35/-.

1 United States Dollar equals

87.15 Indian Rupee

7 Mar, 11:58 pm UTC · Disclaimer

0.0499	United States Dollar
4.35	Indian Rupee

Now Select The Network Settings

▼ Network settings [Info](#) Edit

Network | [Info](#)
vpc-056bf1d89dcf9098e

Subnet | [Info](#)
No preference (Default subnet in any availability zone)

Auto-assign public IP | [Info](#)
Enable
[Additional charges apply](#) when outside of [free tier allowance](#)

Firewall (security groups) | [Info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

Common security groups | [Info](#)

×
VPC: vpc-056bf1d89dcf9098e

[Compare security group rules](#)

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Select the storage And Click the launch instance.

▼ Configure storage [Info](#) Advanced

1x GiB Root volume, 3000 IOPS, Not encrypted

×

[Add new volume](#)

The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance.

[Click refresh to view backup information](#)
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems Edit

▶ Advanced details [Info](#)

Software image (AMI)
Canonical, Ubuntu, 24.04, amd64...[read more](#)
ami-04b4f1a9cf54c11d0

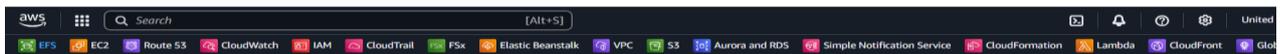
Virtual server type (instance type)
t2.medium

Firewall (security group)
launch-wizard-1

Storage (volumes)
1 volume(s) - 8 GiB

[Free tier](#): In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (for t2.micro where t2.micro isn't available) when used.

[Cancel](#) [Launch instance](#) [Preview code](#)



Launching instance
Launch initiation 77%

▶ Details

Please wait while we launch your instance.
Do not close your browser while this is loading.

Success
 Successfully initiated launch of instances (i-0a5d2681e624a0d94, i-0a5d592bd2e9d1b13, i-08b49f64ba9d92b22)

Instances (1/3) Info Last updated less than a minute ago

Find Instance by attribute or tag (case-sensitive) Running

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
kubernetes-master	i-0a5d2681e624a0d94	Running	t2.medium	Initializing	View alarms +	us-east-1b	ec2-52-23-164-240.co...
kubernetes-slave1	i-08b49f64ba9d92b22	Running	t2.medium	Initializing	View alarms +	us-east-1b	ec2-3-84-110-207.com...
kubernetes-slave2	i-0a5d592bd2e9d1b13	Running	t2.medium	Initializing	View alarms +	us-east-1b	ec2-3-90-102-239.com...

Session settings

SSH Telnet Rsh Xdmcp RDP VNC FTP SFTP Serial File Shell Browser Mosh Aws S3 WSL

Basic SSH settings

Remote host * 52.23.164.240 Specify username ubuntu Port 22

Advanced SSH settings Terminal settings Network settings Bookmark settings

X11-Forwarding Compression Remote environment: Interactive shell

Execute command: Do not exit after command ends

SSH-browser type: SFTP protocol Follow SSH path (experimental)

Use private key D:\Data\Desktop\docker.ppk

Execute macro at session start: <none>

OK Cancel

```
5. 52.23.164.240 (ubuntu)
> For more info, ctrl+click on help or visit our website.

Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1021-aws x86_64)

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

System information as of Sun Mar  9 17:56:42 UTC 2025

System load: 0.11          Processes:            119
Usage of /:  24.9% of 6.71GB Users logged in:      0
Memory usage: 6%          IPv4 address for enX0: 172.31.30.75
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.

/usr/bin/xauth: file /home/ubuntu/.Xauthority does not exist
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-30-75:~$
```

Session settings

SSH Telnet Rsh Xdmcp RDP VNC FTP SFTP Serial File Shell Browser Mosh Aws S3 WSL

Basic SSH settings

Remote host * 3.84.110.207 Specify username ubuntu Port 22

Advanced SSH settings Terminal settings Network settings Bookmark settings

X11-Forwarding Compression Remote environment: Interactive shell

Execute command: Do not exit after command ends

SSH-browser type: SFTP protocol Follow SSH path (experimental)

Use private key D:\Data\Desktop\docker.ppk

Execute macro at session start: <none>

```
5. 52.23.164.240 (ubuntu) 6. 3.84.110.207 (ubuntu)
> For more info, ctrl+click on help or visit our website.

Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1021-aws x86_64)

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

System information as of Sun Mar 9 17:57:46 UTC 2025

System load: 0.03          Processes: 120
Usage of /: 24.9% of 6.71GB Users logged in: 0
Memory usage: 5%          IPv4 address for enX0: 172.31.19.163
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.

/usr/bin/xauth: file /home/ubuntu/.Xauthority does not exist
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-19-163:~$
```

Session settings

SSH Telnet Rsh Xdmpc RDP VNC FTP SFTP Serial File Shell Browser Mosh Aws S3 WSL

Basic SSH settings

Remote host * 3.90.102.239 Specify username ubuntu Port 22

Advanced SSH settings Terminal settings Network settings Bookmark settings

X11-Forwarding Compression Remote environment: Interactive shell

Execute command: Do not exit after command ends

SSH-browser type: SFTP protocol Follow SSH path (experimental)

Use private key D:\Data\Desktop\docker.ppk

Execute macro at session start: <none>

OK Cancel

```
5. 52.23.164.240 (ubuntu) 6. 3.84.110.207 (ubuntu) 7. 3.90.102.239 (ubuntu)
> For more info, ctrl+click on help or visit our website.

Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1021-aws x86_64)

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

System information as of Sun Mar 9 17:59:00 UTC 2025

System load: 0.01          Processes: 119
Usage of /: 24.9% of 6.71GB Users logged in: 0
Memory usage: 5%          IPv4 address for enX0: 172.31.19.153
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.

/usr/bin/xauth: file /home/ubuntu/.Xauthority does not exist
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-19-153:~$
```

```
5. 52.23.164.240 (ubuntu) 6. 3.84.110.207 (ubuntu) 7. 3.90.102.239 (ubuntu)
GNU nano 7.2 install.sh *

##Update the system's package list and install necessary things:

sudo apt-get update
sudo apt install apt-transport-https curl -y

## Install containerd

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" > /etc/apt/sources.list.d/docker.list
sudo apt-get update
sudo apt-get install containerd.io -y

## Create containerd configuration

sudo mkdir -p /etc/containerd
sudo containerd config default | sudo tee /etc/containerd/config.toml

sudo sed -i -e 's/SystemdCgroup = false/SystemdCgroup = true/g' /etc/containerd/config.toml

sudo systemctl restart containerd

## Install Kubernetes

curl -fsSL https://pkgs.k8s.io/core/stable/v1.30/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
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
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
sudo systemctl enable --now kubelet

## Disable swap
```

After connected the ec2 instances we need to install Kubernetes in all 3 nodes

First I take master node.

In master node I need to create one script file.in this scripts there is Kubernetes repository.

```
ubuntu@ip-172-31-30-75:~$ sudo nano install.sh
ubuntu@ip-172-31-30-75:~$ sudo bash install.sh
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://us-east-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://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [916 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [206 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [151 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1036 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [260 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [364 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [19.9 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [754 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [151 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [30.1 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [5884 B]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [552 B]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [14.2 kB]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [12.1 kB]
Get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [20.0 kB]
Get:32 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1104 B]
Get:33 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:35 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
```

Same as it is I need run same script run in slave1 and slave 2 as like master node.

After **sudo kubeadm init --pod-network-cidr=10.244.0.0/16** this command in master node.

```
ubuntu@ip-172-31-30-75:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
I0309 18:09:32.017131 5767 version.go:256] remote version is much newer: v1.32.2; falling back to: stable-1.30
[init] Using Kubernetes version: v1.30.10
[preflight] Running pre-flight checks
[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'
W0309 18:09:32.256871 5767 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.
```

```
[kubelet-check] The kubelet is healthy after 1.501531519s
[api-check] Waiting for a healthy API server. This can take up to 4m0s
[api-check] The API server is healthy after 5.001515215s
[upload-config] Storing the configuration used in ConfigMap "kubeadm-config" in the "kube-system" Namespace
[kubelet] Creating a ConfigMap "kubelet-config" 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-172-31-30-75 as control-plane by adding the labels: [node-role.kubernetes.io/control-plane node.kubernetes.io/exclude-from-external-load-balancers]
[mark-control-plane] Marking the node ip-172-31-30-75 as control-plane by adding the taints [node-role.kubernetes.io/control-plane:NoSchedule]
[bootstrap-token] Using token: nqfkr2.x6zhy6wgkhp5hsv
[bootstrap-token] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to get nodes
[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 csrapprover controller automatically approve CSRs from a Node Bootstrap Token
[bootstrap-token] Configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstrap-token] Creating the "cluster-info" 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 172.31.30.75:6443 --token nqfkr2.x6zhy6wgkhp5hsv \
--discovery-token-ca-cert-hash sha256:338fd511687f18fc6d27b0a70f0269a0d977573a76f0887c97d807ac22edc75
ubuntu@ip-172-31-30-75:~$
```

Now We need to run this command in 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
```

Then need to install the network plugins

#Network plugin

kubectl apply -f <https://github.com/flannel-io/flannel/releases/latest/download/kube-flannel.yml>

```
--discovery-token-ca-cert-hash sha256:338fd511687f18fc6d27b0a70f0269a0d977573a76f0887c97d807ac22edc75
ubuntu@ip-172-31-30-75:~$ mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-30-75:~$ kubectl apply -f https://github.com/flannel-io/flannel/releases/latest/download/kube-flannel.yml
namespace/kube-flannel created
serviceaccount/flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
ubuntu@ip-172-31-30-75:~$
```

Now we need to perform the pre-flight checks in worker node.

```
sudo kubeadm reset pre-flight checks
```

Worker Node 1

```
ubuntu@ip-172-31-19-163:~$ sudo kubeadm reset pre-flight checks
W0309 18:14:27.483322 6011 preflight.go:56] [reset] WARNING: Changes made to this host by 'kubeadm init' or 'kubeadm join' will be reverted.
[reset] Are you sure you want to proceed? [y/N]: y
[preflight] Running pre-flight checks
W0309 18:14:29.367833 6011 removeetcdmember.go:106] [reset] No kubeadm config, using etcd pod spec to get data directory
[reset] Deleted contents of the etcd data directory: /var/lib/etcd
[reset] Stopping the kubelet service
[reset] Unmounting mounted directories in "/var/lib/kubelet"
[reset] Deleting contents of directories: [/etc/kubernetes/manifests /var/lib/kubelet /etc/kubernetes/pki]
[reset] Deleting files: [/etc/kubernetes/admin.conf /etc/kubernetes/super-admin.conf /etc/kubernetes/kubelet.conf /etc/kubernetes/bootstrap-kubelet.conf /etc/kubernetes/controller-manager.conf /etc/kubernetes/scheduler.conf]

The reset process does not clean CNI configuration. To do so, you must remove /etc/cni/net.d

The reset process does not reset or clean up iptables rules or IPVS tables.
If you wish to reset iptables, you must do so manually by using the "iptables" command.

If your cluster was setup to utilize IPVS, run ipvsadm --clear (or similar)
to reset your system's IPVS tables.

The reset process does not clean your kubeconfig files and you must remove them manually.
Please, check the contents of the $HOME/.kube/config file.
ubuntu@ip-172-31-19-163:~$
```

Worker Node 2

```
ubuntu@ip-172-31-19-153:~$ sudo kubeadm reset pre-flight checks
W0309 18:14:34.123560 3742 preflight.go:56] [reset] WARNING: Changes made to this host by 'kubeadm init' or 'kubeadm join' will be reverted.
[reset] Are you sure you want to proceed? [y/N]: y
[preflight] Running pre-flight checks
W0309 18:14:36.090718 3742 removeetcdmember.go:106] [reset] No kubeadm config, using etcd pod spec to get data directory
[reset] Deleted contents of the etcd data directory: /var/lib/etcd
[reset] Stopping the kubelet service
[reset] Unmounting mounted directories in "/var/lib/kubelet"
[reset] Deleting contents of directories: [/etc/kubernetes/manifests /var/lib/kubelet /etc/kubernetes/pki]
[reset] Deleting files: [/etc/kubernetes/admin.conf /etc/kubernetes/super-admin.conf /etc/kubernetes/kubelet.conf /etc/kubernetes/bootstrap-kubelet.conf /etc/kubernetes/controller-manager.conf /etc/kubernetes/scheduler.conf]

The reset process does not clean CNI configuration. To do so, you must remove /etc/cni/net.d

The reset process does not reset or clean up iptables rules or IPVS tables.
If you wish to reset iptables, you must do so manually by using the "iptables" command.

If your cluster was setup to utilize IPVS, run ipvsadm --clear (or similar)
to reset your system's IPVS tables.

The reset process does not clean your kubeconfig files and you must remove them manually.
Please, check the contents of the $HOME/.kube/config file.
ubuntu@ip-172-31-19-153:~$
```

Now Worker Node 1 and 2 we need join to master node.

Paste the join command you got from the master node and append `--v=5` at the end but first use sudo su command to become root (avoid using sudo your-token).

```
kubeadm join 172.31.30.75:6443 --token nqfkr2.x6zhy6wgkhp5hsv \
--discovery-token-ca-cert-hash
sha256:338fdf511687f18fc6d27b0a70f0269a0d977573a76f0887c97d80
7ac22edc75 --v=5
```

This Command need to run in both worker nodes.

Worker Node 1

```
ubuntu@ip-172-31-19-163:~$ sudo su
root@ip-172-31-19-163:/home/ubuntu# kubeadm join 172.31.30.75:6443 --token nqfkr2.x6zhy6wgkhp5hsv \
--discovery-token-ca-cert-hash sha256:338fdf511687f18fc6d27b0a70f0269a0d977573a76f0887c97d807ac22edc75 --v=5
I0309 18:18:59.318013 7109 join.go:417] [preflight] found NodeName empty; using OS hostname as NodeName
I0309 18:18:59.318204 7109 initconfiguration.go:122] detected and using CRI socket: unix:///var/run/containerd/containerd.sock
[preflight] Running pre-flight checks
I0309 18:18:59.319086 7109 preflight.go:93] [preflight] Running general checks
I0309 18:18:59.319122 7109 checks.go:278] validating the existence of file /etc/kubernetes/kubelet.conf
I0309 18:18:59.319133 7109 checks.go:278] validating the existence of file /etc/kubernetes/bootstrap-kubelet.conf
I0309 18:18:59.319141 7109 checks.go:102] validating the container runtime
I0309 18:18:59.339726 7109 checks.go:637] validating whether swap is enabled or not
I0309 18:18:59.339816 7109 checks.go:368] validating the presence of executable crictl
I0309 18:18:59.339868 7109 checks.go:368] validating the presence of executable conntrack
I0309 18:18:59.339914 7109 checks.go:368] validating the presence of executable ip
I0309 18:18:59.339956 7109 checks.go:368] validating the presence of executable iptables
I0309 18:18:59.340025 7109 checks.go:368] validating the presence of executable mount
I0309 18:18:59.340071 7109 checks.go:368] validating the presence of executable nsenter
I0309 18:18:59.340122 7109 checks.go:368] validating the presence of executable ethtool
I0309 18:18:59.340169 7109 checks.go:368] validating the presence of executable tc
I0309 18:18:59.340199 7109 checks.go:368] validating the presence of executable touch
I0309 18:18:59.340237 7109 checks.go:514] running all checks
I0309 18:18:59.355280 7109 checks.go:399] checking whether the given node name is valid and reachable using net.LookupHost
I0309 18:18:59.357424 7109 checks.go:603] validating kubelet version
I0309 18:18:59.405713 7109 checks.go:128] validating if the "kubelet" service is enabled and active
I0309 18:18:59.410924 7109 checks.go:201] validating availability of port 10250
I0309 18:18:59.420540 7109 checks.go:278] validating the existence of file /etc/kubernetes/pki/ca.crt
I0309 18:18:59.420564 7109 checks.go:428] validating if the connectivity type is via proxy or direct
I0309 18:18:59.420589 7109 checks.go:327] validating the contents of file /proc/sys/net/ipv4/ip_forward
I0309 18:18:59.420615 7109 join.go:536] [preflight] Discovering cluster-info
I0309 18:18:59.420632 7109 token.go:79] [discovery] Created cluster-info discovery client, requesting info from "172.31.30.75:6443"
I0309 18:18:59.420937 7109 token.go:210] [discovery] Waiting for the cluster-info ConfigMap to receive a JWS signature for token ID "nqfkr2"
I0309 18:18:59.431049 7109 token.go:117] [discovery] Requesting info from "172.31.30.75:6443" again to validate TLS against the pinned public key
I0309 18:18:59.431396 7109 token.go:210] [discovery] Waiting for the cluster-info ConfigMap to receive a JWS signature for token ID "nqfkr2"
I0309 18:18:59.437450 7109 token.go:134] [discovery] Cluster info signature and contents are valid and TLS certificate validates against pinned roots,
will use API Server "172.31.30.75:6443"
I0309 18:18:59.437471 7109 discovery.go:52] [discovery] Using provided TLSBootstrapToken as authentication credentials for the join process
I0309 18:18:59.437479 7109 join.go:550] [preflight] Fetching init configuration
I0309 18:18:59.437487 7109 join.go:596] [preflight] Retrieving KubeConfig objects
[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'
I0309 18:18:59.444763 7109 kubeproxy.go:55] attempting to download the KubeProxyConfiguration from ConfigMap "kube-proxy"
```

```
I0309 18:18:59.444763 7109 kubeproxy.go:55] attempting to download the KubeProxyConfiguration from ConfigMap "kube-proxy"
I0309 18:18:59.448128 7109 kubelet.go:74] attempting to download the KubeletConfiguration from ConfigMap "kubelet-config"
I0309 18:18:59.452781 7109 initconfiguration.go:114] skip CRI socket detection, fill with the default CRI socket unix:///var/run/containerd/containerd.sock
I0309 18:18:59.452910 7109 interface.go:432] Looking for default routes with IPv4 addresses
I0309 18:18:59.452933 7109 interface.go:437] Default route transits interface "enX0"
I0309 18:18:59.452997 7109 interface.go:209] Interface enX0 is up
I0309 18:18:59.453023 7109 interface.go:257] Interface "enX0" has 2 addresses :[172.31.19.163/20 fe80::8ff:e9ff:fe85:b28f/64].
I0309 18:18:59.453033 7109 interface.go:224] Checking addr 172.31.19.163/20.
I0309 18:18:59.453040 7109 interface.go:231] IP found 172.31.19.163
I0309 18:18:59.453057 7109 interface.go:263] Found valid IPv4 address 172.31.19.163 for interface "enX0".
I0309 18:18:59.453063 7109 interface.go:443] Found active IP 172.31.19.163
I0309 18:18:59.459278 7109 preflight.go:104] [preflight] Running configuration dependant checks
I0309 18:18:59.459298 7109 controlplaneprepare.go:225] [download-certs] Skipping certs download
I0309 18:18:59.459324 7109 kubelet.go:124] [kubelet-start] writing bootstrap kubelet config file at /etc/kubernetes/bootstrap-kubelet.conf
I0309 18:18:59.459745 7109 kubelet.go:139] [kubelet-start] writing CA certificate at /etc/kubernetes/pki/ca.crt
I0309 18:18:59.460213 7109 kubelet.go:160] [kubelet-start] Checking for an existing Node in the cluster with name "ip-172-31-19-163" and status "Ready"
I0309 18:18:59.462499 7109 kubelet.go:175] [kubelet-start] Stopping the kubelet
[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/kubelet-environment"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 503.36862ms
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap
I0309 18:19:00.230578 7109 kubelet.go:242] [kubelet-start] preserving the crisocket information for the node
I0309 18:19:00.230589 7109 patchnode.go:31] [patchnode] Uploading the CRI Socket information "unix:///var/run/containerd/containerd.sock" to the Node
API object "ip-172-31-19-163" as an annotation
I0309 18:19:00.230746 7109 cert_rotation.go:137] Starting client certificate rotation controller

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.

root@ip-172-31-19-163:/home/ubuntu#
```

Worker Node 2

```
ubuntu@ip-172-31-19-153:~$ sudo su
root@ip-172-31-19-153:/home/ubuntu# kubectl join 172.31.30.75:6443 --token nqfkr2.x6zhy6wgkhp5hsv \
--discovery-token-ca-cert-hash sha256:338fdf511687f18fc6d27b0a70f0269a0d97753a76f0887c97d807ac22edc75 --v=5
I0309 18:20:32.511151 4010 join.go:417] [preflight] found NodeName empty; using OS hostname as NodeName
I0309 18:20:32.511440 4010 initconfiguration.go:122] detected and using CRI socket: unix:///var/run/containerd/containerd.sock
[preflight] Running pre-flight checks
I0309 18:20:32.511487 4010 preflight.go:93] [preflight] Running general checks
I0309 18:20:32.511514 4010 checks.go:278] validating the existence of file /etc/kubernetes/kubelet.conf
I0309 18:20:32.511521 4010 checks.go:278] validating the existence of file /etc/kubernetes/bootstrap-kubelet.conf
I0309 18:20:32.511528 4010 checks.go:102] validating the container runtime
I0309 18:20:32.531505 4010 checks.go:637] validating whether swap is enabled or not
I0309 18:20:32.531561 4010 checks.go:368] validating the presence of executable crictl
I0309 18:20:32.531610 4010 checks.go:368] validating the presence of executable conntrack
I0309 18:20:32.531631 4010 checks.go:368] validating the presence of executable ip
I0309 18:20:32.531644 4010 checks.go:368] validating the presence of executable iptables
I0309 18:20:32.531702 4010 checks.go:368] validating the presence of executable mount
I0309 18:20:32.531715 4010 checks.go:368] validating the presence of executable nseter
I0309 18:20:32.531737 4010 checks.go:368] validating the presence of executable ethtool
I0309 18:20:32.531753 4010 checks.go:368] validating the presence of executable tc
I0309 18:20:32.531768 4010 checks.go:368] validating the presence of executable touch
I0309 18:20:32.531781 4010 checks.go:514] running all checks
I0309 18:20:32.545551 4010 checks.go:399] checking whether the given node name is valid and reachable using net.LookupHost
I0309 18:20:32.547130 4010 checks.go:603] validating kubelet version
I0309 18:20:32.593001 4010 checks.go:128] validating if the "kubelet" service is enabled and active
I0309 18:20:32.606810 4010 checks.go:201] validating availability of port 10250
I0309 18:20:32.606950 4010 checks.go:278] validating the existence of file /etc/kubernetes/pki/ca.crt
I0309 18:20:32.606960 4010 checks.go:428] validating if the connectivity type is via proxy or direct
I0309 18:20:32.606983 4010 checks.go:327] validating the contents of file /proc/sys/net/ipv4/ip_forward
I0309 18:20:32.607004 4010 join.go:536] [preflight] Discovering cluster-info
I0309 18:20:32.607021 4010 token.go:79] [discovery] Created cluster-info discovery client, requesting info from "172.31.30.75:6443"
I0309 18:20:32.607288 4010 token.go:210] [discovery] Waiting for the cluster-info ConfigMap to receive a JWS signature for token ID "nqfkr2"
I0309 18:20:32.618687 4010 token.go:117] [discovery] Requesting info from "172.31.30.75:6443" again to validate TLS against the pinned public key
I0309 18:20:32.618922 4010 token.go:210] [discovery] Waiting for the cluster-info ConfigMap to receive a JWS signature for token ID "nqfkr2"
I0309 18:20:32.625664 4010 token.go:134] [discovery] Cluster info signature and contents are valid and TLS certificate validates against pinned roots,
will use API Server "172.31.30.75:6443"
I0309 18:20:32.625759 4010 discovery.go:52] [discovery] Using provided TLSBootstrapToken as authentication credentials for the join process
I0309 18:20:32.625772 4010 join.go:550] [preflight] Fetching init configuration
I0309 18:20:32.625778 4010 join.go:596] [preflight] Retrieving KubeConfig objects
[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'
```

```
I0309 18:20:32.625772 4010 join.go:550] [preflight] Fetching init configuration
I0309 18:20:32.625778 4010 join.go:596] [preflight] Retrieving KubeConfig objects
[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'
I0309 18:20:32.632900 4010 kubeproxy.go:55] attempting to download the KubeProxyConfiguration from ConfigMap "kube-proxy"
I0309 18:20:32.636321 4010 kubelet.go:74] attempting to download the KubeletConfiguration from ConfigMap "Kubelet-config"
I0309 18:20:32.640078 4010 initconfiguration.go:114] skip CRI socket detection, fill with the default CRI socket unix:///var/run/containerd/containerd.sock
I0309 18:20:32.640229 4010 interface.go:432] Looking for default routes with IPv4 addresses
I0309 18:20:32.640243 4010 interface.go:437] Default route transits interface "enX0"
I0309 18:20:32.640309 4010 interface.go:209] Interface enX0 is up
I0309 18:20:32.640336 4010 interface.go:257] Interface "enX0" has 2 addresses :[172.31.19.153/20 fe80::8ff:c9ff:fecc:c53/64].
I0309 18:20:32.640352 4010 interface.go:224] Checking addr 172.31.19.153/20.
I0309 18:20:32.640363 4010 interface.go:231] IP found 172.31.19.153
I0309 18:20:32.640375 4010 interface.go:263] Found valid IPv4 address 172.31.19.153 for interface "enX0".
I0309 18:20:32.640380 4010 interface.go:443] Found active IP 172.31.19.153
I0309 18:20:32.644892 4010 preflight.go:104] [preflight] Running configuration dependant checks
I0309 18:20:32.644910 4010 controlplaneprepare.go:225] [download-certs] Skipping certs download
I0309 18:20:32.644922 4010 kubelet.go:124] [kubelet-start] writing bootstrap kubelet config file at /etc/kubernetes/bootstrap-kubelet.conf
I0309 18:20:32.645894 4010 kubelet.go:139] [kubelet-start] writing CA certificate at /etc/kubernetes/pki/ca.crt
I0309 18:20:32.646484 4010 kubelet.go:160] [kubelet-start] Checking for an existing Node in the cluster with name "ip-172-31-19-153" and status "Ready"
I0309 18:20:32.649500 4010 kubelet.go:175] [kubelet-start] Stopping the kubelet
[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 at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 1.00232247s
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap
I0309 18:20:33.910530 4010 cert rotation.go:137] Starting client certificate rotation controller
I0309 18:20:33.910658 4010 kubelet.go:242] [kubelet-start] preserving the crisocket information for the node
I0309 18:20:33.910693 4010 patchnode.go:31] [patchnode] Uploading the CRI Socket information "unix:///var/run/containerd/containerd.sock" to the Node API object "ip-172-31-19-153" as an annotation

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.

root@ip-172-31-19-153:/home/ubuntu#
```

Now we need to check the nodes in master server.

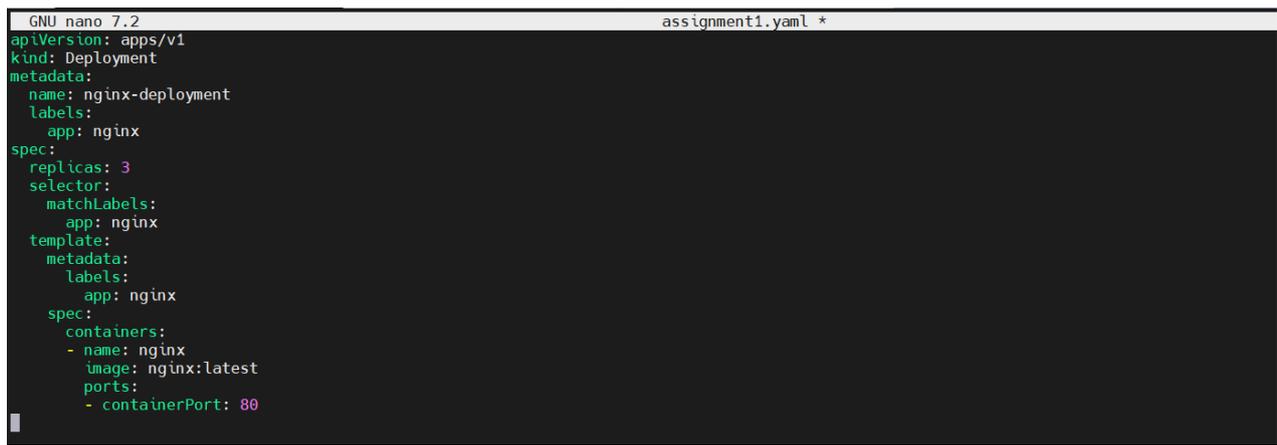
```
ubuntu@ip-172-31-30-75:~$ kubectl get nodes
NAME                 STATUS    ROLES    AGE     VERSION
ip-172-31-19-153    Ready    <none>   73s    v1.30.10
ip-172-31-19-163    Ready    <none>   2m46s  v1.30.10
ip-172-31-30-75     Ready    control-plane  11m    v1.30.10
ubuntu@ip-172-31-30-75:~$
```

Now Kubernetes installation has been done successfully.

2. Create a NGINX deployment of 3 replicas

In this we need to create the yaml file and apply below code in Kubernetes.create nano file
Sudo nano assignment1.yaml.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        ports:
        - containerPort: 80
```



```
GNU nano 7.2 assignment1.yaml *
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        ports:
        - containerPort: 80
```

```
ubuntu@ip-172-31-30-75:~$ cat assignment1.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        ports:
        - containerPort: 80
ubuntu@ip-172-31-30-75:~$
```

Now we need to apply this code so command is
kubectl apply -f assignment1.yaml

```
ubuntu@ip-172-31-30-75:~$ kubectl apply -f assignment1.yaml
deployment.apps/nginx-deployment created
ubuntu@ip-172-31-30-75:~$
```

The Deployment has been successfully

```
ubuntu@ip-172-31-30-75:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-576c6b7b6-jh55k    1/1     Running   0           38s
nginx-deployment-576c6b7b6-qwq7d    1/1     Running   0           38s
nginx-deployment-576c6b7b6-vql56    1/1     Running   0           38s
ubuntu@ip-172-31-30-75:~$
```

As per task we have completed the deployment.