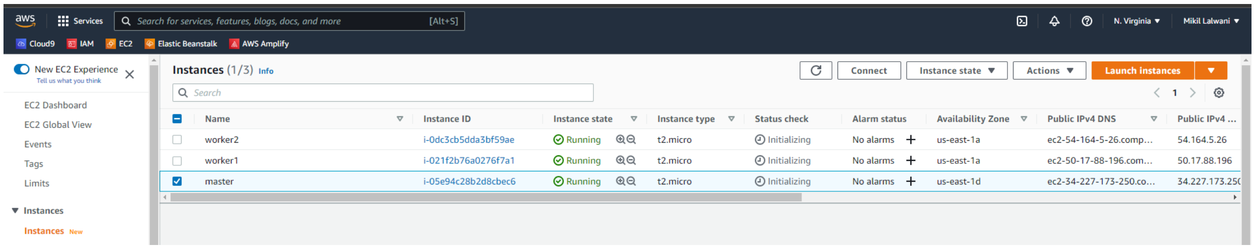
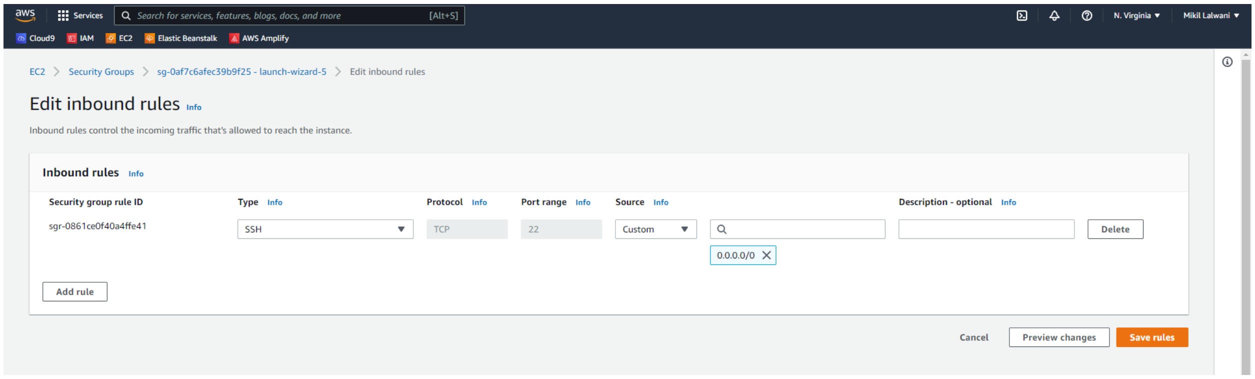
1. Create 3 EC2 instances i.e master, worker1, and worker2.



Instances (1/3)

☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
☐	worker2	i-0dc3cb5dda3bf59ae	✓ Running	t2.micro	Initializing	No alarms	us-east-1a	ec2-54-164-5-26.comp...	54.164.5.26
☐	worker1	i-021f2b76a0276f7a1	✓ Running	t2.micro	Initializing	No alarms	us-east-1a	ec2-50-17-88-196.com...	50.17.88.196
☑	master	i-05e94c28b2d8cbec6	✓ Running	t2.micro	Initializing	No alarms	us-east-1d	ec2-34-227-173-250.co...	34.227.173.250

2. Change network setting.



EC2 > Security Groups > sg-0af7c6afec39b9f25 - launch-wizard-5 > Edit inbound rules

Edit inbound rules

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules

Security group rule ID	Type	Protocol	Port range	Source		Description - optional
sgr-0861ce0f40a4ffe41	SSH	TCP	22	Custom	0.0.0.0/0	

3. Set up Docker

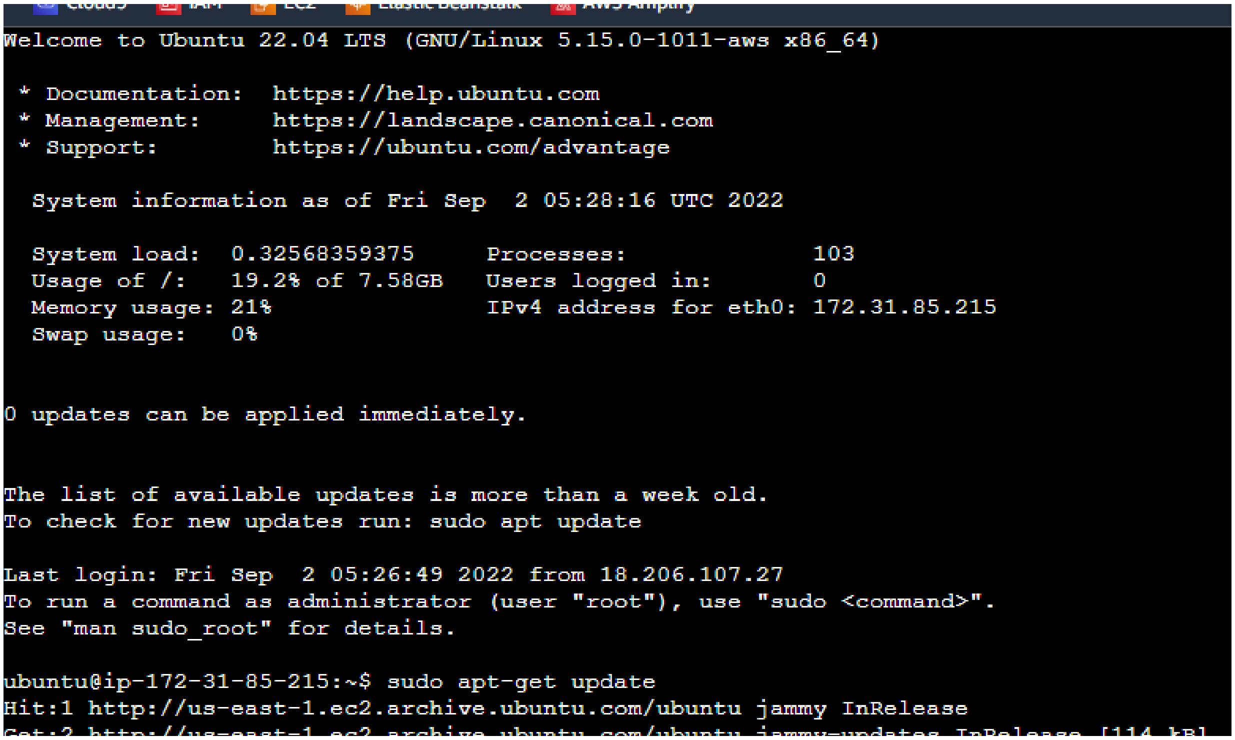
Install Docker

Kubernetes requires an existing Docker installation. If you already have Docker installed, skip ahead to Step 4.

If you do not have Kubernetes, install it by following these steps:

Update the package list with the command:

```
on-master&slave$sudo apt-get update
```



```
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-1011-aws x86_64)

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

  System information as of Fri Sep  2 05:28:16 UTC 2022

  System load:  0.32568359375      Processes:             103
  Usage of /:   19.2% of 7.58GB   Users logged in:       0
  Memory usage: 21%                IPv4 address for eth0: 172.31.85.215
  Swap usage:   0%


0 updates can be applied immediately.


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

Last login: Fri Sep  2 05:26:49 2022 from 18.206.107.27
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-85-215:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
```

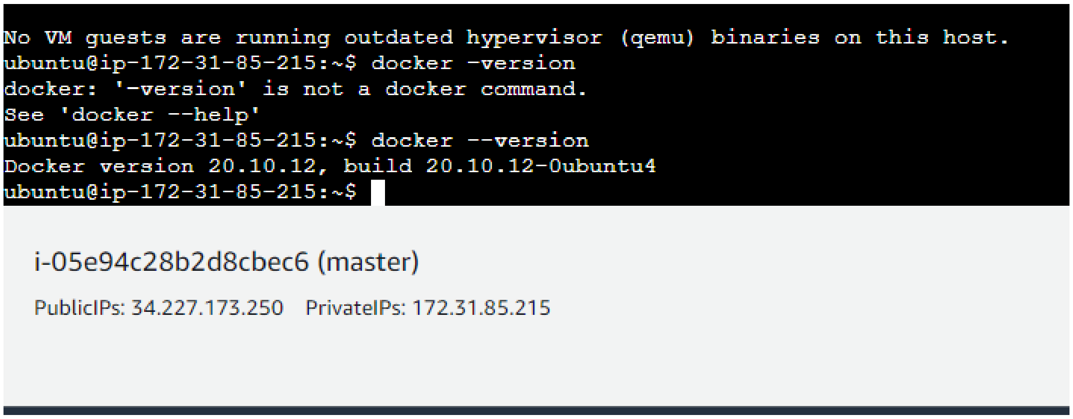
Next, install Docker with the command:

on-master&slave$sudo apt-get install docker.io

Repeat the process on each server that will act as a node.

Check the installation (and version) by entering the following:

on-master&slave$docker --version



```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-85-215:~$ docker -version
docker: '-version' is not a docker command.
See 'docker --help'
ubuntu@ip-172-31-85-215:~$ docker --version
Docker version 20.10.12, build 20.10.12-0ubuntu4
ubuntu@ip-172-31-85-215:~$
```

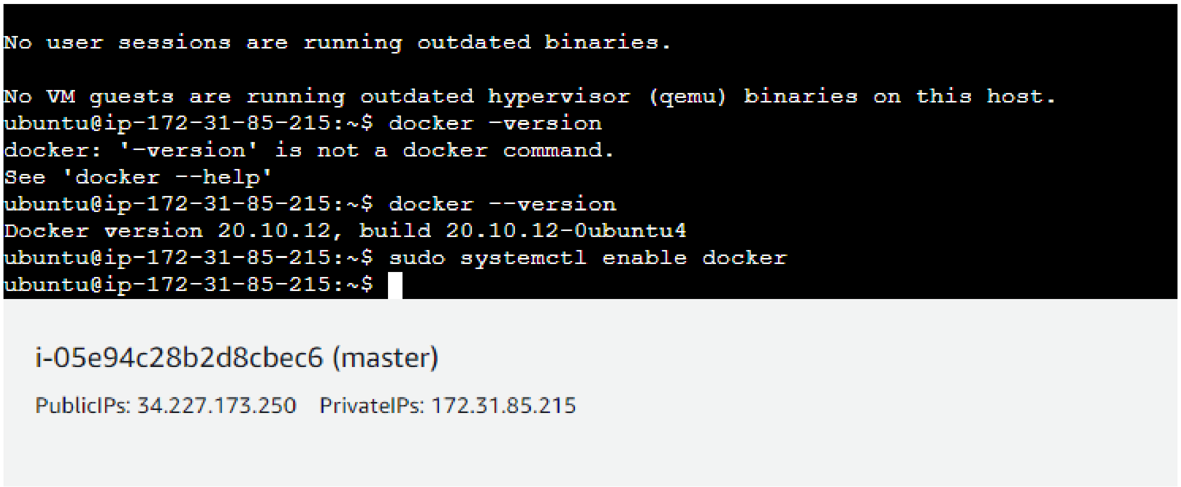
i-05e94c28b2d8cbec6 (master)

PublicIPs: 34.227.173.250   PrivateIPs: 172.31.85.215

4. Start and Enable Docker

Set Docker to launch at boot by entering the following:

on-master&slave$sudo systemctl enable docker



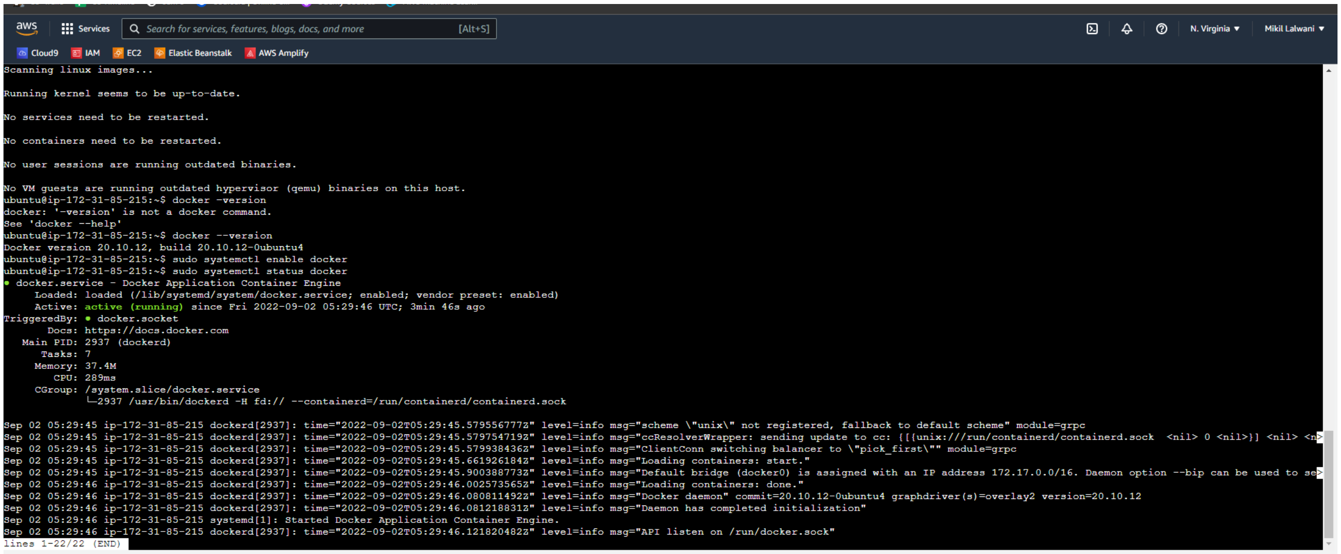
```
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-85-215:~$ docker -version
docker: '-version' is not a docker command.
See 'docker --help'
ubuntu@ip-172-31-85-215:~$ docker --version
Docker version 20.10.12, build 20.10.12-0ubuntu4
ubuntu@ip-172-31-85-215:~$ sudo systemctl enable docker
ubuntu@ip-172-31-85-215:~$ 
```

i-05e94c28b2d8cbec6 (master)

PublicIPs: 34.227.173.250   PrivateIPs: 172.31.85.215

Verify Docker is running:

on-master&slave$sudo systemctl status docker To start Docker if it’s not running:
on-master&slave$sudo systemctl start docker



```
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-85-215:~$ docker -version
docker: '-version' is not a docker command.
See 'docker --help'
ubuntu@ip-172-31-85-215:~$ docker --version
Docker version 20.10.12, build 20.10.12-0ubuntu4
ubuntu@ip-172-31-85-215:~$ sudo systemctl enable docker
ubuntu@ip-172-31-85-215:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
     Active: active (running) since Fri 2022-09-02 05:29:46 UTC; 3min 46s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 2937 (dockerd)
      Tasks: 7
     Memory: 37.4M
        CPU: 289ms
     CGroup: /system.slice/docker.service
             └─2937 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Sep 02 05:29:45 ip-172-31-85-215 dockerd[2937]: time="2022-09-02T05:29:45.579556777Z" level=info msg="scheme \"unix\" not registered, fallback to default scheme" module=grpc
Sep 02 05:29:45 ip-172-31-85-215 dockerd[2937]: time="2022-09-02T05:29:45.579754719Z" level=info msg="ccResolverWrapper: sending update to cc: {[{unix:///run/containerd/containerd.sock  <nil> 0 <nil>}] <nil> <n>
Sep 02 05:29:45 ip-172-31-85-215 dockerd[2937]: time="2022-09-02T05:29:45.579938436Z" level=info msg="ClientConn switching balancer to \"pick_first\"" module=grpc
Sep 02 05:29:45 ip-172-31-85-215 dockerd[2937]: time="2022-09-02T05:29:45.661926184Z" level=info msg="Loading containers: start."
Sep 02 05:29:45 ip-172-31-85-215 dockerd[2937]: time="2022-09-02T05:29:45.900388773Z" level=info msg="Default bridge (docker0) is assigned with an IP address 172.17.0.0/16. Daemon option --bip can be used to se>
Sep 02 05:29:46 ip-172-31-85-215 dockerd[2937]: time="2022-09-02T05:29:46.002573565Z" level=info msg="Loading containers: done."
Sep 02 05:29:46 ip-172-31-85-215 dockerd[2937]: time="2022-09-02T05:29:46.080811492Z" level=info msg="Docker daemon" commit=20.10.12-0ubuntu4 graphdriver(s)=overlay2 version=20.10.12
Sep 02 05:29:46 ip-172-31-85-215 dockerd[2937]: time="2022-09-02T05:29:46.081218831Z" level=info msg="Daemon has completed initialization"
Sep 02 05:29:46 ip-172-31-85-215 systemd[1]: Started Docker Application Container Engine.
Sep 02 05:29:46 ip-172-31-85-215 dockerd[2937]: time="2022-09-02T05:29:46.121820482Z" level=info msg="API listen on /run/docker.sock"
lines 1-22/22 (END)
```

Repeat on all the other nodes.

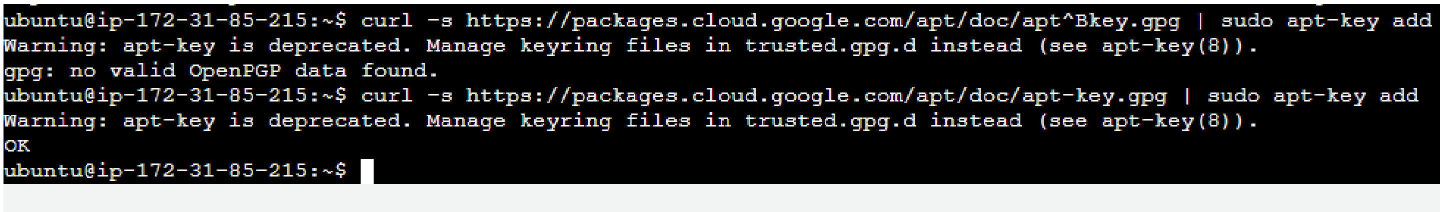
Install Kubernetes

5. Add Kubernetes Signing Key

Since you are downloading Kubernetes from a non-standard repository, it is essential to ensure that the software is authentic. This is done by adding a signing key.

Enter the following to add a signing key:

on-master&slave$curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add



```
ubuntu@ip-172-31-85-215:~$ curl -s https://packages.cloud.google.com/apt/doc/apt^Bkey.gpg | sudo apt-key add
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
gpg: no valid OpenPGP data found.
ubuntu@ip-172-31-85-215:~$ curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
ubuntu@ip-172-31-85-215:~$
```

If you get an error that curl is not installed, install it with:

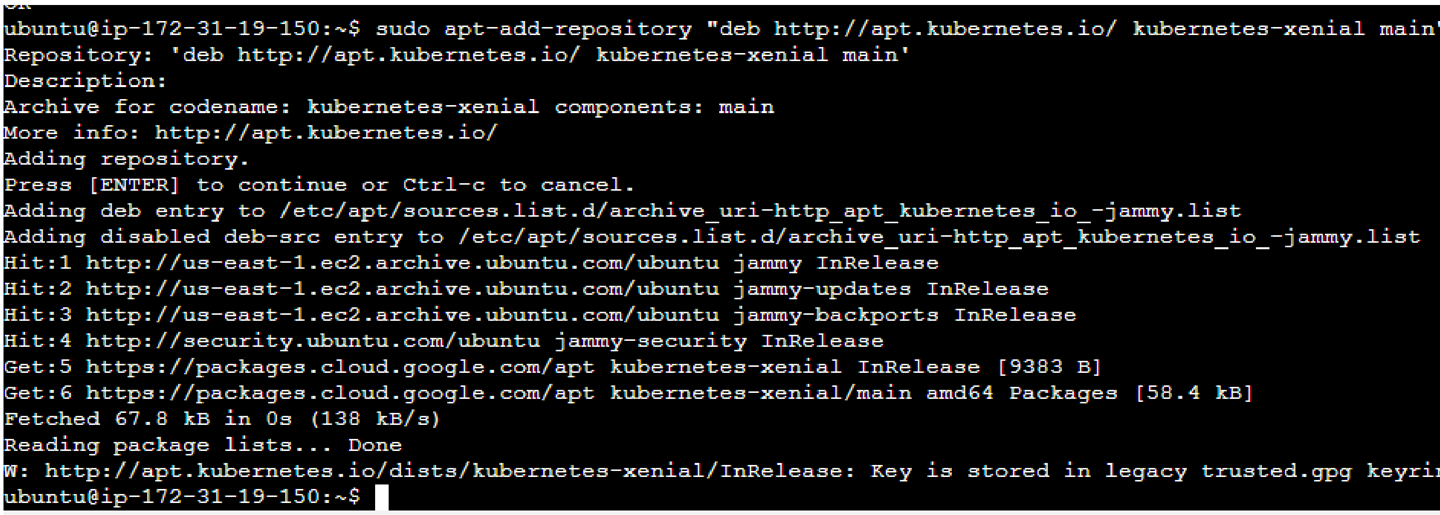
on-master&slave$sudo apt-get install curl

Then repeat the previous command to install the signing keys. Repeat for each server node.

6. Add Software Repositories

Kubernetes is not included in the default repositories. To add them, enter the following:

on-master&slave$sudo apt-add-repository "deb http://apt.kubernetes.io/ kubernetes-xenial main"



```
OK
ubuntu@ip-172-31-19-150:~$ sudo apt-add-repository "deb http://apt.kubernetes.io/ kubernetes-xenial main"
Repository: 'deb http://apt.kubernetes.io/ kubernetes-xenial main'
Description:
Archive for codename: kubernetes-xenial components: main
More info: http://apt.kubernetes.io/
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
Adding deb entry to /etc/apt/sources.list.d/archive_uri-http_apt_kubernetes_io_-jammy.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-http_apt_kubernetes_io_-jammy.list
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:5 https://packages.cloud.google.com/apt kubernetes-xenial InRelease [9383 B]
Get:6 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 Packages [58.4 kB]
Fetched 67.8 kB in 0s (138 kB/s)
Reading package lists... Done
W: http://apt.kubernetes.io/dists/kubernetes-xenial/InRelease: Key is stored in legacy trusted.gpg keyri
ubuntu@ip-172-31-19-150:~$
```

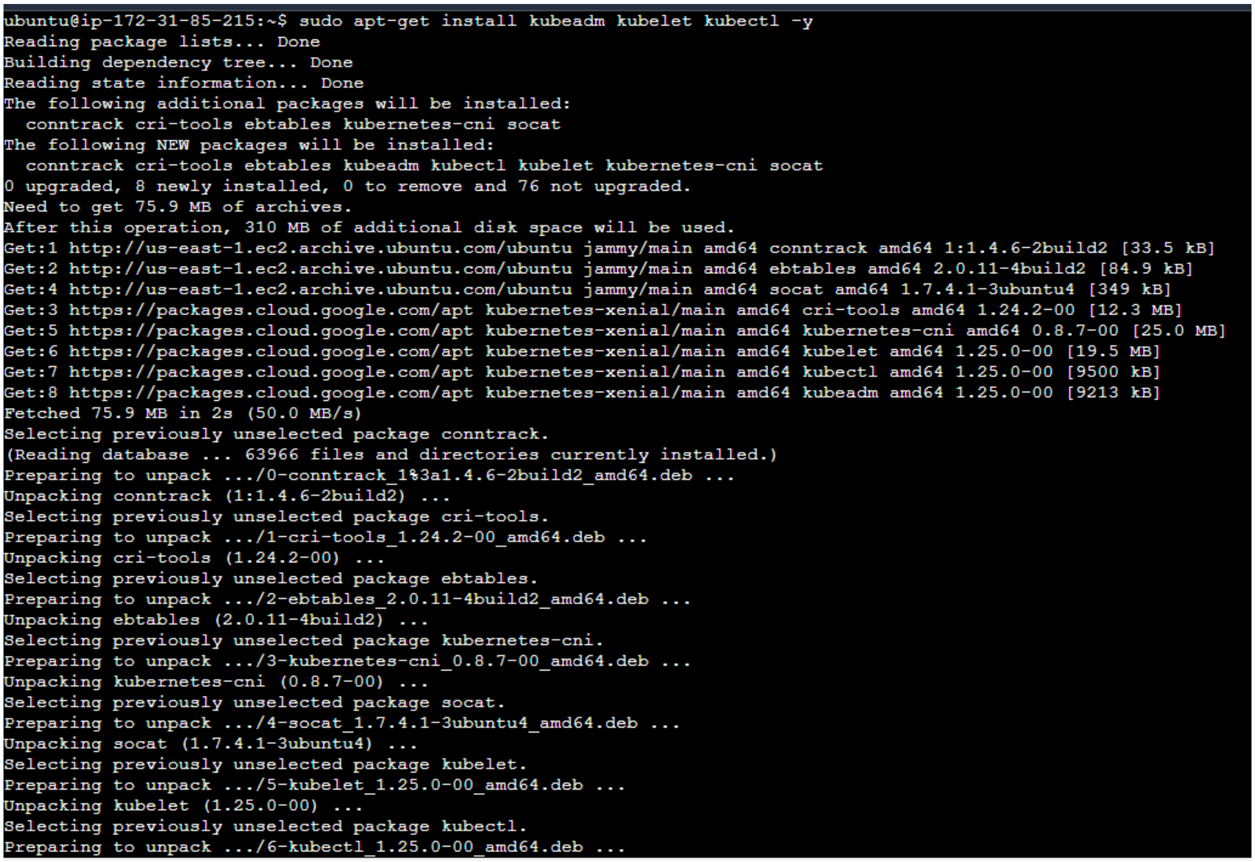
Repeat on each server node.

7. Kubernetes Installation Tools

Kubeadm (Kubernetes Admin) is a tool that helps initialize a cluster. It fast-tracks setup by using community-sourced best practices. Kubelet is the work package, which runs on every node and starts containers. The tool gives you command-line access to clusters.

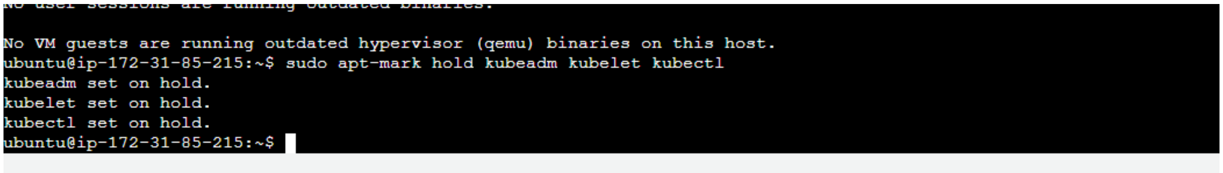
Install Kubernetes tools with the command:

on-master&slave$sudo apt-get install kubeadm kubelet kubectl -y



```
ubuntu@ip-172-31-85-215:~$ sudo apt-get install kubeadm kubelet kubectl -y
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 76 not upgraded.
Need to get 75.9 MB of archives.
After this operation, 310 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 conntrack amd64 1:1.4.6-2build2 [33.5 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 ebtables amd64 2.0.11-4build2 [84.9 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 socat amd64 1.7.4.1-3ubuntu4 [349 kB]
Get:3 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 cri-tools amd64 1.24.2-00 [12.3 MB]
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 1.25.0-00 [19.5 MB]
Get:7 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubectl amd64 1.25.0-00 [9500 kB]
Get:8 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubeadm amd64 1.25.0-00 [9213 kB]
Fetched 75.9 MB in 2s (50.0 MB/s)
Selecting previously unselected package conntrack.
(Reading database ... 63966 files and directories currently installed.)
Preparing to unpack .../0-conntrack_1%3a1.4.6-2build2_amd64.deb ...
Unpacking conntrack (1:1.4.6-2build2) ...
Selecting previously unselected package cri-tools.
Preparing to unpack .../1-cri-tools_1.24.2-00_amd64.deb ...
Unpacking cri-tools (1.24.2-00) ...
Selecting previously unselected package ebtables.
Preparing to unpack .../2-ebtables_2.0.11-4build2_amd64.deb ...
Unpacking ebtables (2.0.11-4build2) ...
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) ...
Selecting previously unselected package socat.
Preparing to unpack .../4-socat_1.7.4.1-3ubuntu4_amd64.deb ...
Unpacking socat (1.7.4.1-3ubuntu4) ...
Selecting previously unselected package kubelet.
Preparing to unpack .../5-kubelet_1.25.0-00_amd64.deb ...
Unpacking kubelet (1.25.0-00) ...
Selecting previously unselected package kubectl.
Preparing to unpack .../6-kubectl_1.25.0-00_amd64.deb ...
```

on-master&slave$sudo apt-mark hold kubeadm kubelet kubectl

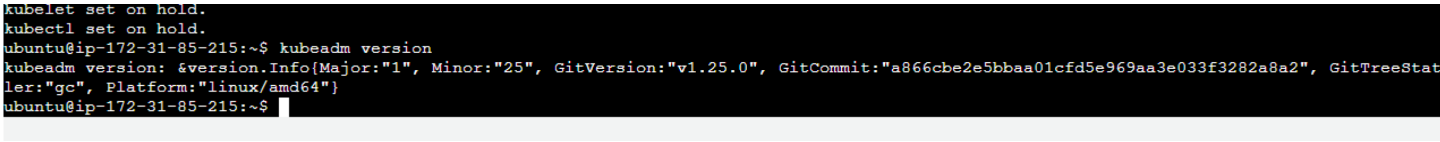


```
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-85-215:~$ sudo apt-mark hold kubeadm kubelet kubectl
kubeadm set on hold.
kubelet set on hold.
kubectl set on hold.
ubuntu@ip-172-31-85-215:~$
```

Allow the process to complete.

Verify the installation with:

on-master&slave$kubeadm version



```
kubelet set on hold.
kubectl set on hold.
ubuntu@ip-172-31-85-215:~$ kubeadm version
kubeadm version: &version.Info{Major:"1", Minor:"25", GitVersion:"v1.25.0", GitCommit:"a866cbe2e5bbaa01cfd5e969aa3e033f3282a8a2", GitTreeStat
ler:"gc", Platform:"linux/amd64"}
ubuntu@ip-172-31-85-215:~$
```

Repeat for each server node.

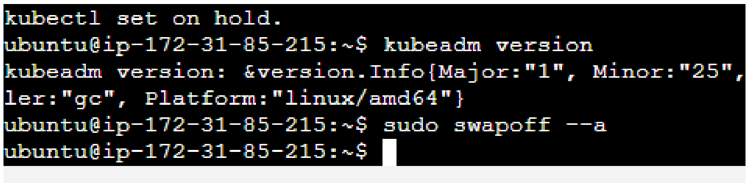
Note: Make sure you install the same version of each package on each machine. Different versions can create instability. Also, this process prevents apt from automatically updating Kubernetes. For update instructions, please see the developers' instructions.

Kubernetes Deployment

8. Begin Kubernetes Deployment

Start by disabling the swap memory on each server:

on-master&slave$sudo swapoff --a

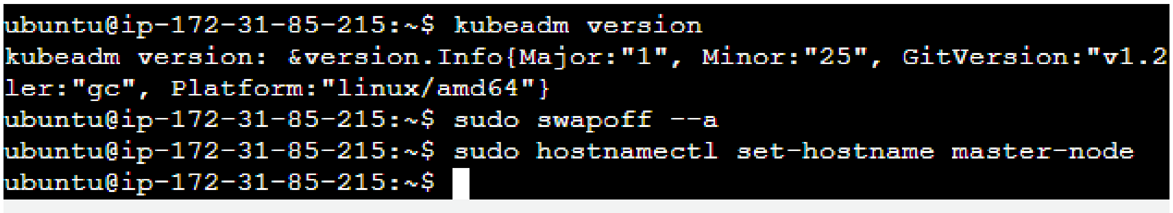


```
kubectl set on hold.
ubuntu@ip-172-31-85-215:~$ kubeadm version
kubeadm version: &version.Info{Major:"1", Minor:"25",
ler:"gc", Platform:"linux/amd64"}
ubuntu@ip-172-31-85-215:~$ sudo swapoff --a
ubuntu@ip-172-31-85-215:~$
```

9. Assign Unique Hostname for Each Server Node

Decide which server to set as the master node. Then enter the command:

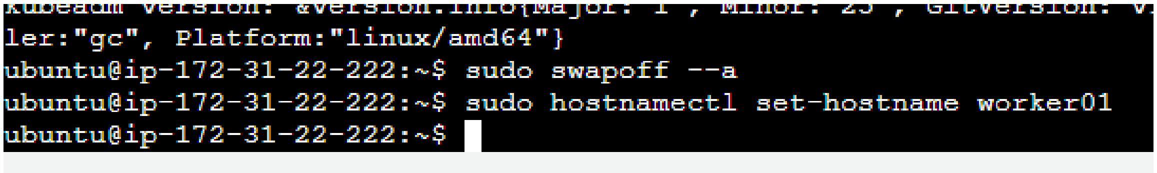
on-master$sudo hostnamectl set-hostname master-node



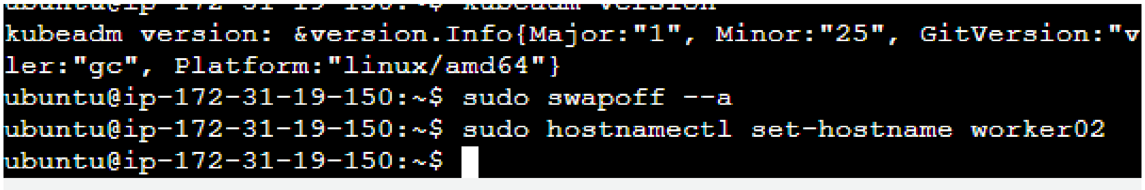
```
ubuntu@ip-172-31-85-215:~$ kubeadm version
kubeadm version: &version.Info{Major:"1", Minor:"25", GitVersion:"v1.2
ler:"gc", Platform:"linux/amd64"}
ubuntu@ip-172-31-85-215:~$ sudo swapoff --a
ubuntu@ip-172-31-85-215:~$ sudo hostnamectl set-hostname master-node
ubuntu@ip-172-31-85-215:~$
```

Next, set a worker node hostname by entering the following on the worker server:

on-slave$sudo hostnamectl set-hostname worker01\



```
kubeadm version: &version.Info{Major:"1", Minor:"25", GitVersion:"v
ler:"gc", Platform:"linux/amd64"}
ubuntu@ip-172-31-22-222:~$ sudo swapoff --a
ubuntu@ip-172-31-22-222:~$ sudo hostnamectl set-hostname worker01
ubuntu@ip-172-31-22-222:~$
```



```
ubuntu@ip-172-31-19-150:~$ kubeadm version
kubeadm version: &version.Info{Major:"1", Minor:"25", GitVersion:"v
ler:"gc", Platform:"linux/amd64"}
ubuntu@ip-172-31-19-150:~$ sudo swapoff --a
ubuntu@ip-172-31-19-150:~$ sudo hostnamectl set-hostname worker02
ubuntu@ip-172-31-19-150:~$
```

If you have additional worker nodes, use this process to set a unique hostnsame on each.

10. Initialize Kubernetes on Master Node

Switch to the master server node, and enter the following:

on-master$sudo kubeadm init --pod-network-cidr=10.244.0.0/16

If you are trying to run this on EC2 you'll get an error message saying less cpu and memory to override the error run the above command with --ignore-preflight-errors=all

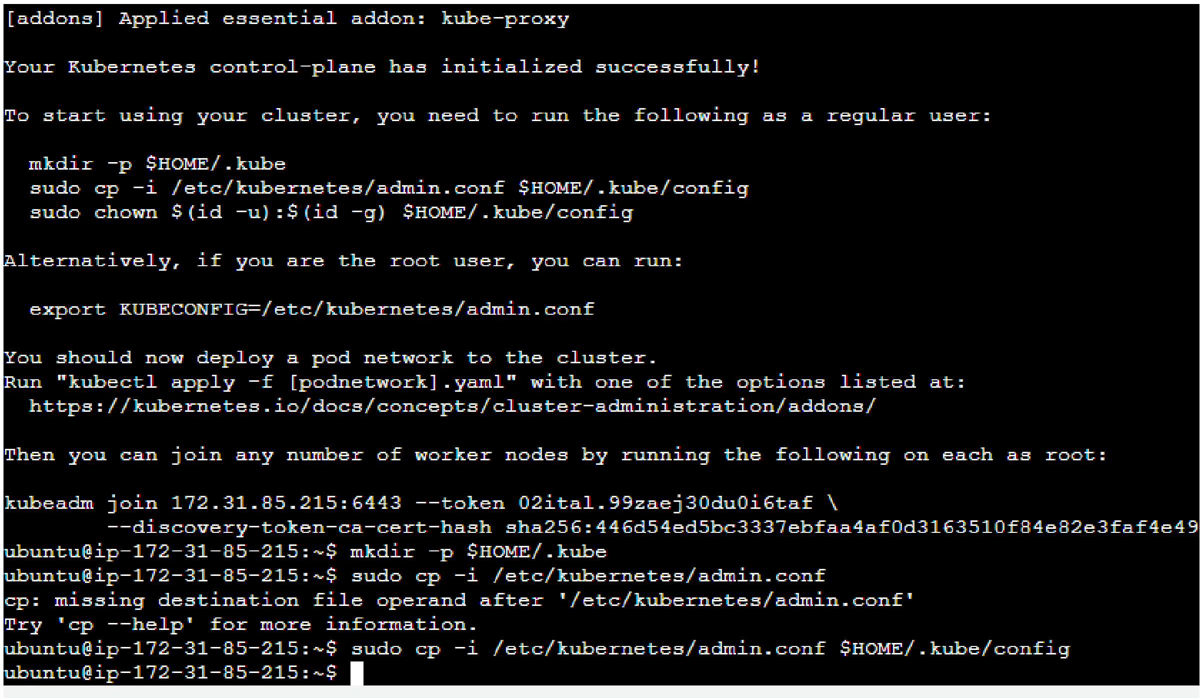
For eg: on-master$sudo kubeadm init --pod-network-cidr=10.244.0.0/16
--ignore-preflight-errors=all

Once this command finishes, it will display a kubeadm join message at the end. Make a note of the whole entry. This will be used to join the worker nodes to the cluster.

```
kubeadm join 172.31.38.116:6443 --token lhu178.98f4hafj9qdn4pve \
        --discovery-token-ca-cert-hash
sha256:8f55371cd3d53a5d72faa368e93e0a99d2953016706de871fcdb27c83e5b995c
```

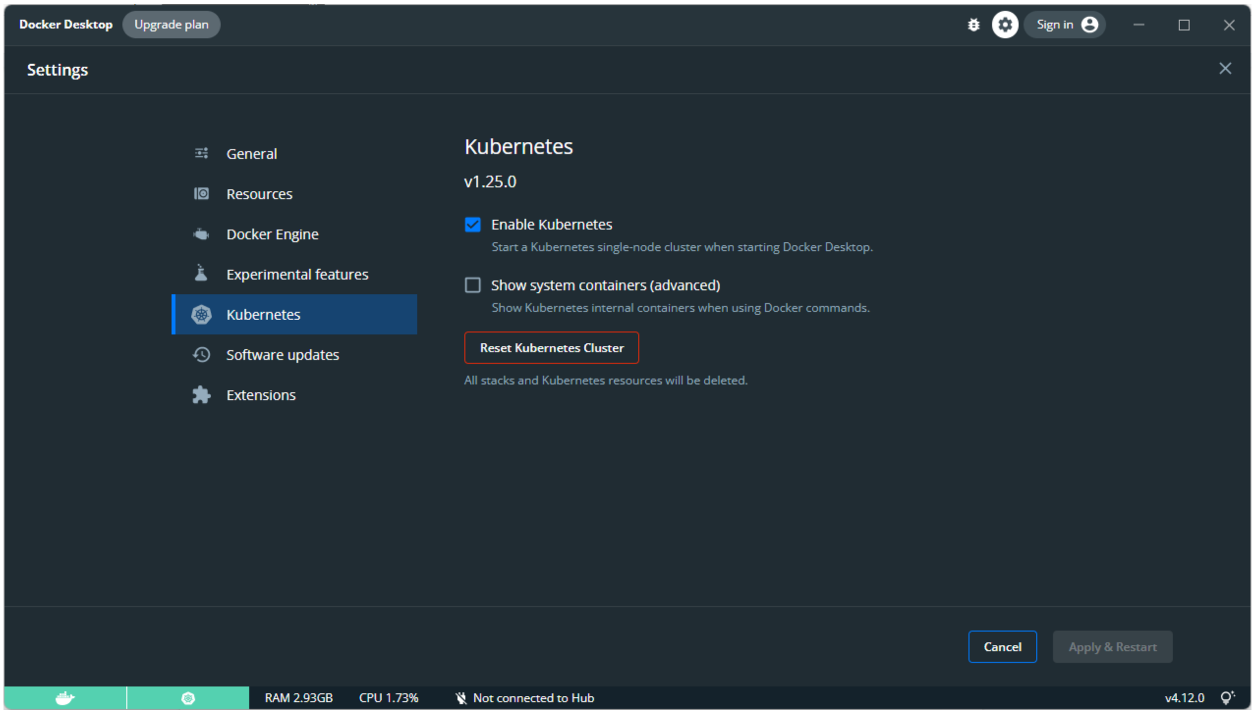
Next, enter the following to create a directory for the cluster:

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



```
[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 172.31.85.215:6443 --token 02ital.99zaej30du0i6taf \
        --discovery-token-ca-cert-hash sha256:446d54ed5bc3337ebfaa4af0d3163510f84e82e3faf4e49
ubuntu@ip-172-31-85-215:~$ mkdir -p $HOME/.kube
ubuntu@ip-172-31-85-215:~$ sudo cp -i /etc/kubernetes/admin.conf
cp: missing destination file operand after '/etc/kubernetes/admin.conf'
Try 'cp --help' for more information.
ubuntu@ip-172-31-85-215:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
ubuntu@ip-172-31-85-215:~$
```

11. Install Docker Desktop.
12. Enable Kubernetes in settings.



Kubernetes

v1.25.0

☑ Enable Kubernetes
Start a Kubernetes single-node cluster when starting Docker Desktop.

☐ Show system containers (advanced)
Show Kubernetes internal containers when using Docker commands.

All stacks and Kubernetes resources will be deleted.

13. Run this commands

    ```
    kubectl create deployment httpenv --image=bretfisher/httpenv
    kubectl scale deployment/httpenv --replicas=3
    kubectl expose deployment/httpenv --port 8888
    kubectl get service
    kubectl run tmp-shell --rm -it --image bretfisher/netshoot -- bash
    ```

14. To delete pods run kubectl delete pod <pod-name>