PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)

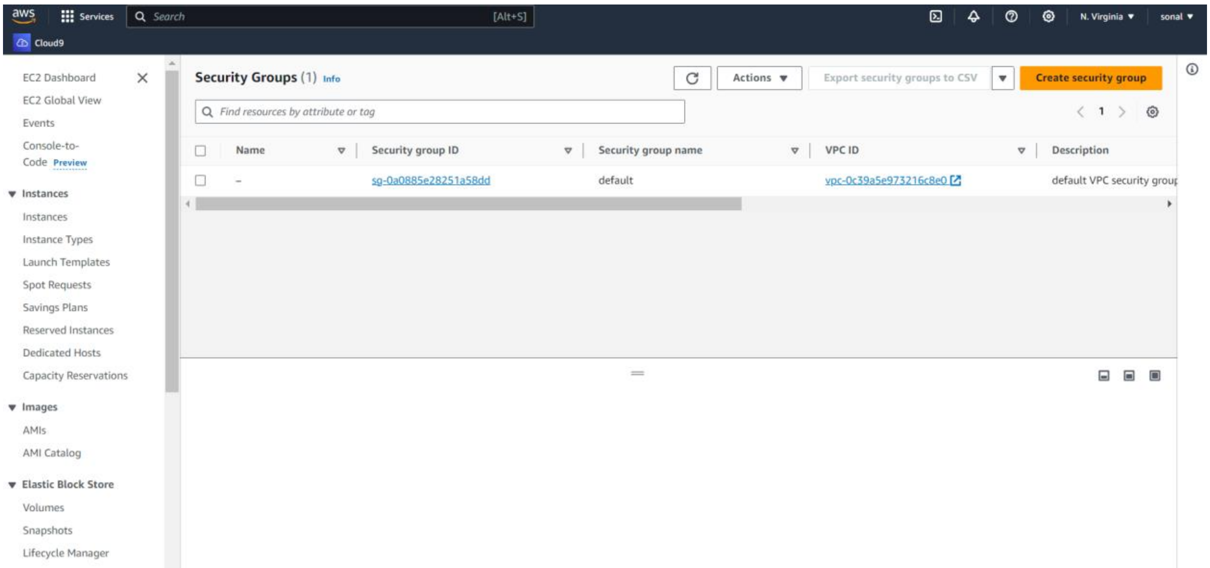
EXPERIMENT NO. 03

Aim: To understand the Kubernetes Cluster Architecture, install and Spin Up a Kubernetes Cluster on Linux Machines/Cloud Platforms.

Aim: To install Kubectl and execute Kubectl commands to manage the Kubernetes cluster and deploy YourFirst Kubernetes

STEP 1:

Check security group,delete all SG only keep default

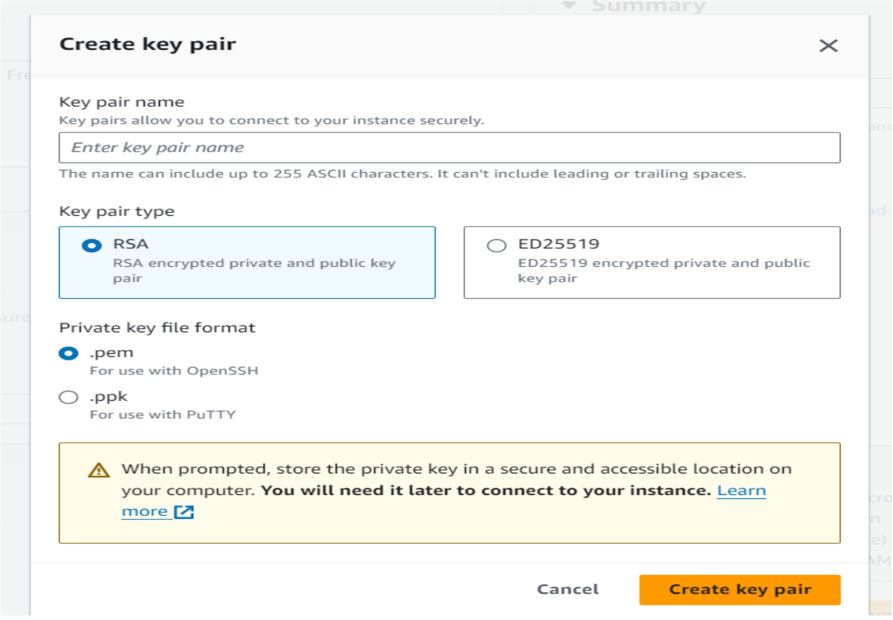


Security Groups (1) Info

□	Name	Security group ID	Security group name	VPC ID	Description
□	–	sg-0a0885e28251a58dd	default	vpc-0c39a5e973216c8e0	default VPC security group

Create 2 instance

Create key pair



Create key pair

Key pair name
Key pairs allow you to connect to your instance securely.

Enter key pair name

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

● RSA
RSA encrypted private and public key pair

○ ED25519
ED25519 encrypted private and public key pair

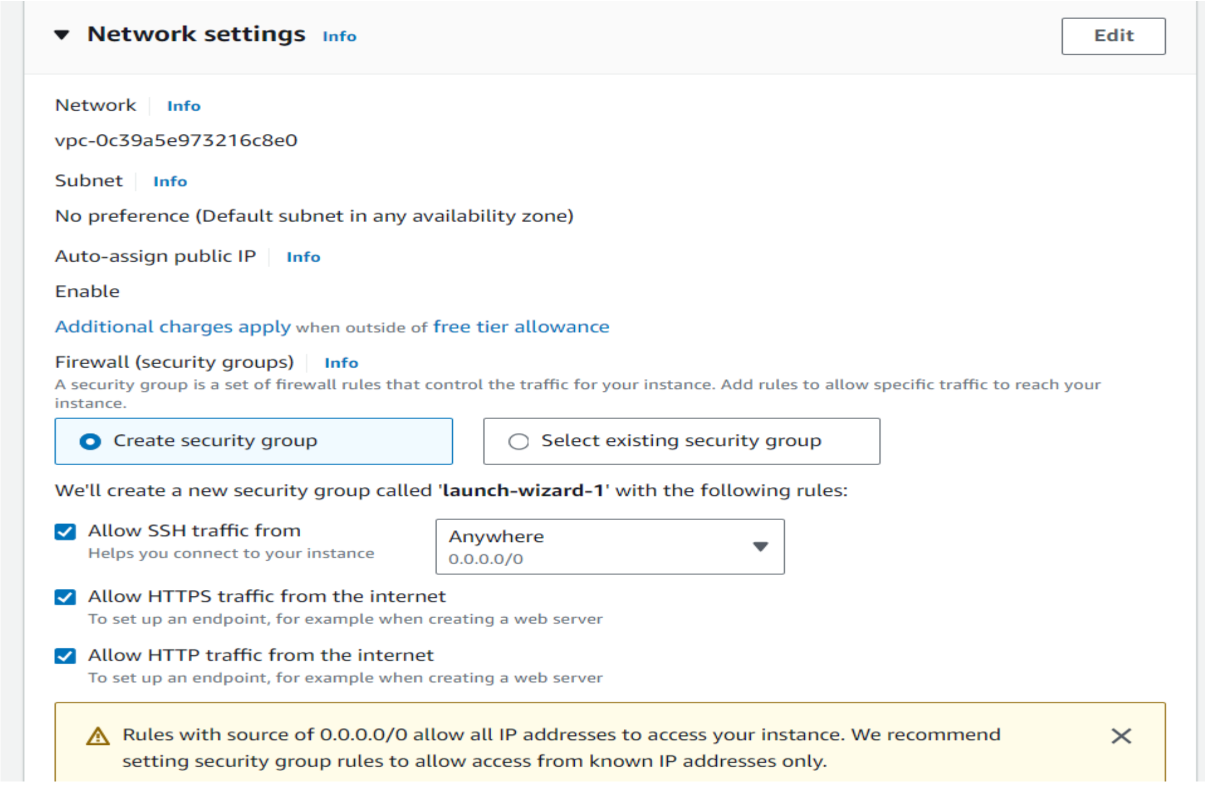
Private key file format

● .pem
For use with OpenSSH

○ .ppk
For use with PuTTY

⚠ When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** Learn more

Create security group and allow traffic



Network settings Info

Network Info
vpc-0c39a5e973216c8e0

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

We'll create a new security group called '**launch-wizard-1**' with the following rules:

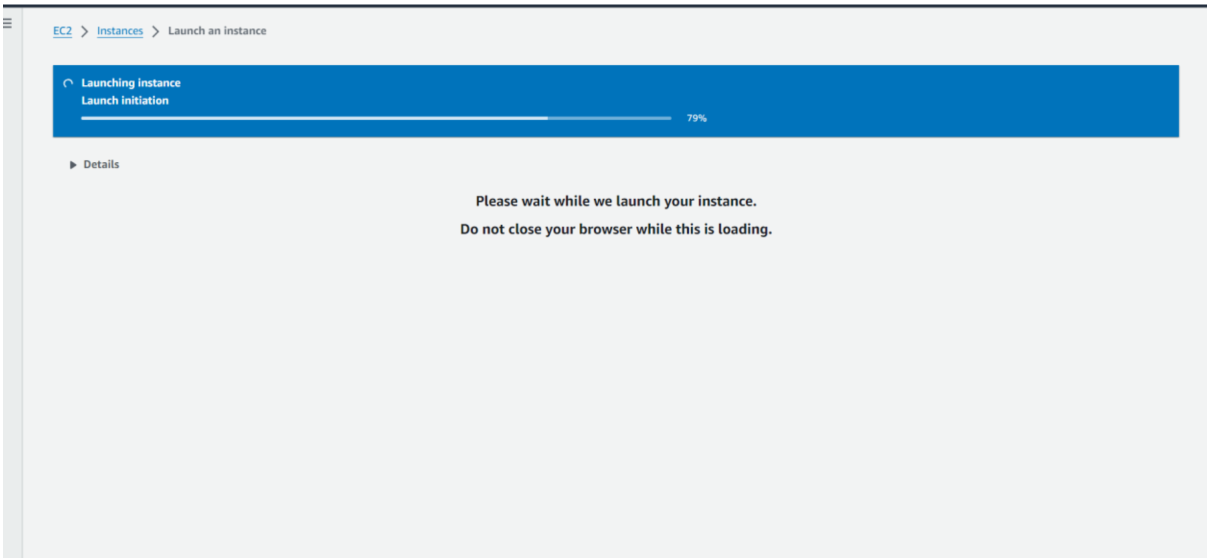
☑ Allow SSH traffic from
Helps you connect to your instance
Anywhere
0.0.0.0/0

☑ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

☑ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Launch Instance

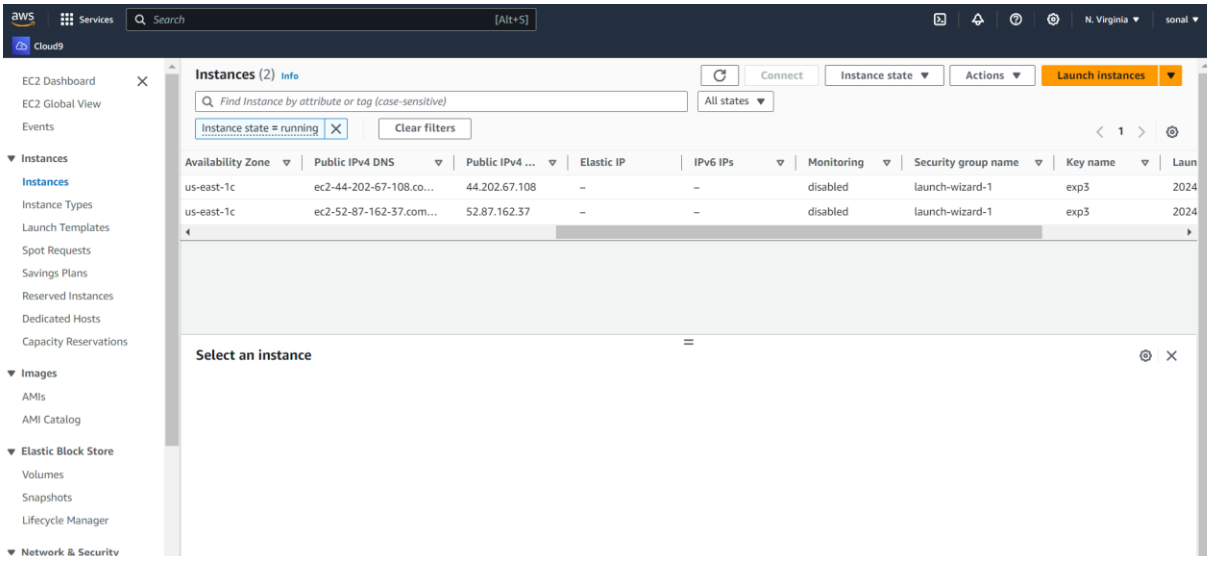


Launching instance
Launch initiation
79%

Please wait while we launch your instance.

Do not close your browser while this is loading.

Check security group of both instances

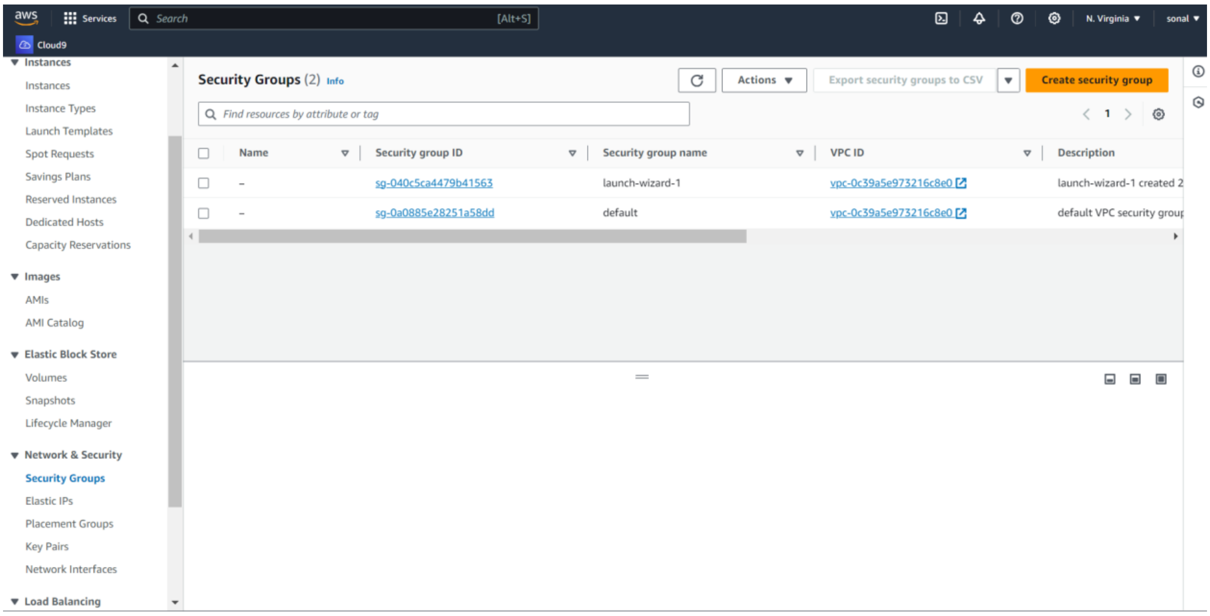


Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs	Monitoring	Security group name	Key name	Laun
us-east-1c	ec2-44-202-67-108.co...	44.202.67.108	–	–	disabled	launch-wizard-1	exp3	2024
us-east-1c	ec2-52-87-162-37.com...	52.87.162.37	–	–	disabled	launch-wizard-1	exp3	2024

Here security group is launch wizard-1

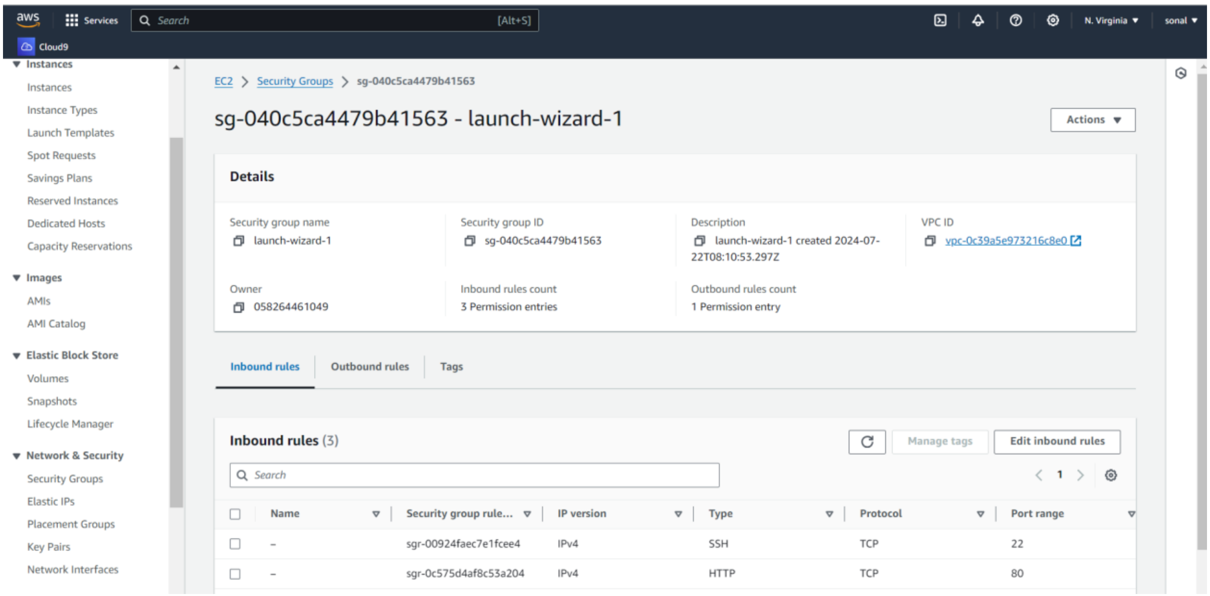
Now go to security group from left pane

Click on Security group id of Launch wizard-1



Security Groups (2)

□	Name	Security group ID	Security group name	VPC ID	Description
□	–	sg-040c5ca4479b41563	launch-wizard-1	vpc-0c39a5e973216c8e0	launch-wizard-1 created 2
□	–	sg-0a0885e28251a58dd	default	vpc-0c39a5e973216c8e0	default VPC security grour



EC2 > Security Groups > sg-040c5ca4479b41563

sg-040c5ca4479b41563 - launch-wizard-1

Details

Security group name: launch-wizard-1

Security group ID: sg-040c5ca4479b41563

Description: launch-wizard-1 created 2024-07-22T08:10:53.297Z

VPC ID: vpc-0c39a5e973216c8e0

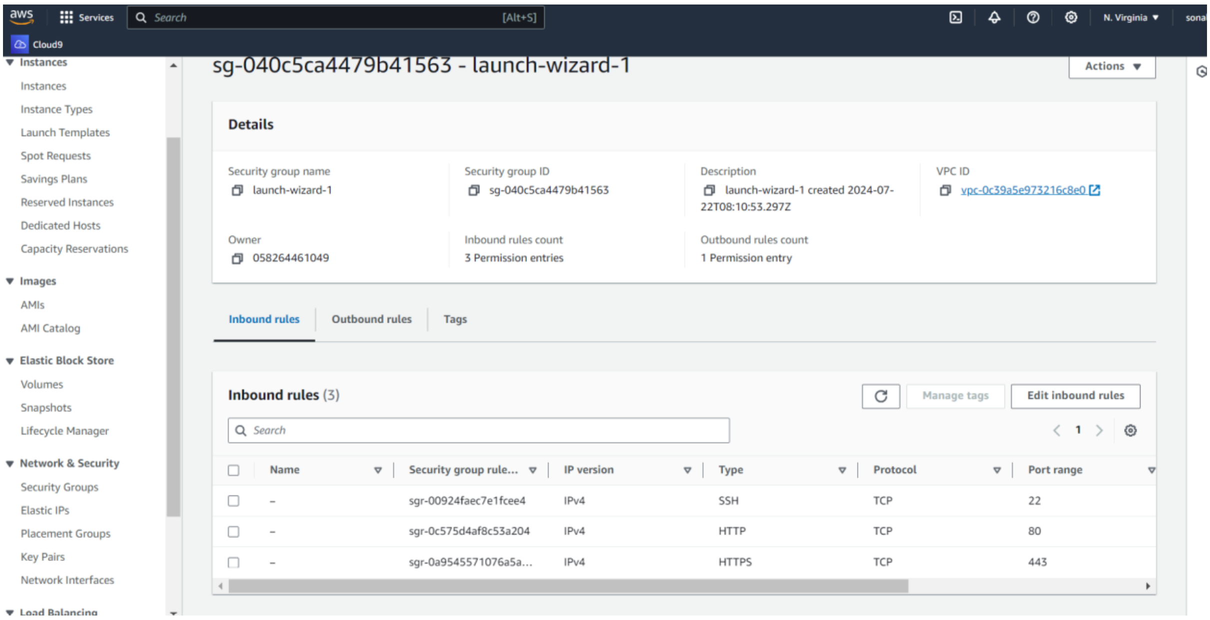
Owner: 058264461049

Inbound rules count: 3 Permission entries

Outbound rules count: 1 Permission entry

Inbound rules (3)

□	Name	Security group rule...	IP version	Type	Protocol	Port range
□	–	sgr-00924faec7e1fcee4	IPv4	SSH	TCP	22
□	–	sgr-0c575d4af8c53a204	IPv4	HTTP	TCP	80



sg-040c5ca4479b41563 - launch-wizard-1

Details

Security group name: launch-wizard-1

Security group ID: sg-040c5ca4479b41563

Description: launch-wizard-1 created 2024-07-22T08:10:53.297Z

VPC ID: vpc-0c39a5e973216c8e0

Owner: 058264461049

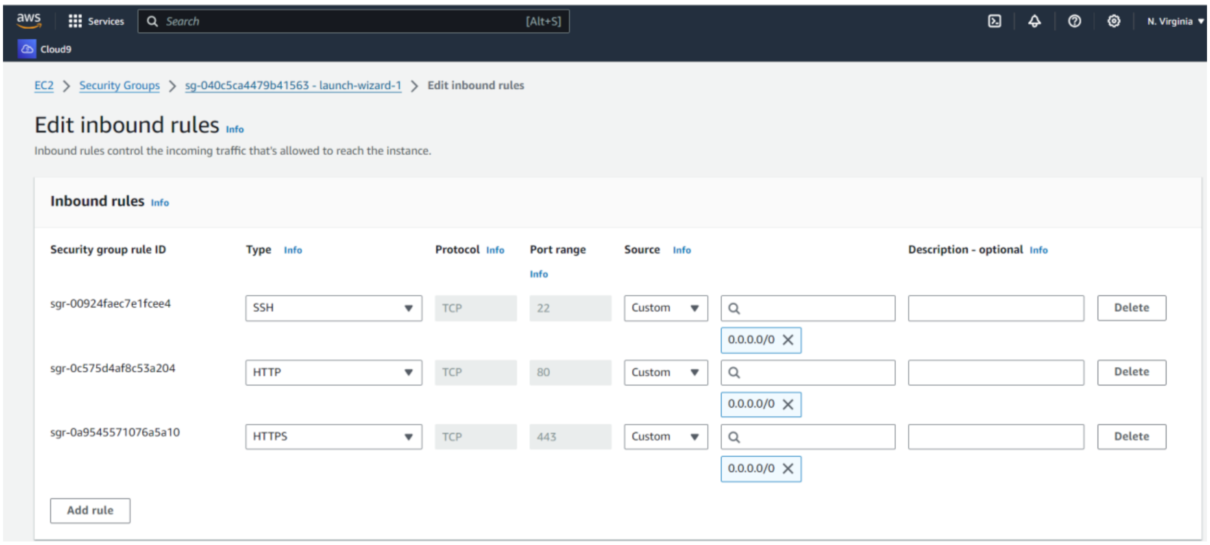
Inbound rules count: 3 Permission entries

Outbound rules count: 1 Permission entry

Inbound rules (3)

Name	Security group rule...	IP version	Type	Protocol	Port range
–	sgr-00924faec7e1fcee4	IPv4	SSH	TCP	22
–	sgr-0c575d4af8c53a204	IPv4	HTTP	TCP	80
–	sgr-0a9545571076a5a...	IPv4	HTTPS	TCP	443

Edit inbound rule



EC2 > Security Groups > sg-040c5ca4479b41563 - launch-wizard-1 > Edit inbound rules

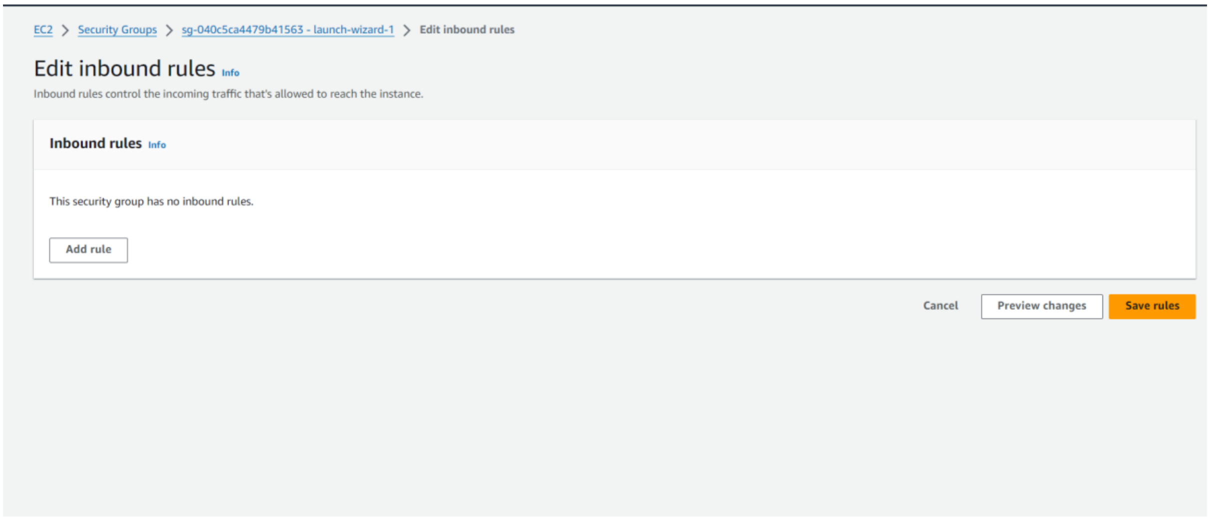
Edit inbound rules

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-00924faec7e1fcee4	SSH	TCP	22	Custom	0.0.0.0/0		Delete
sgr-0c575d4af8c53a204	HTTP	TCP	80	Custom	0.0.0.0/0		Delete
sgr-0a9545571076a5a10	HTTPS	TCP	443	Custom	0.0.0.0/0		Delete

Delete all rules



EC2 > Security Groups > sg-040c5ca4479b41563 - launch-wizard-1 > Edit inbound rules

Edit inbound rules Info

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

Inbound rules Info

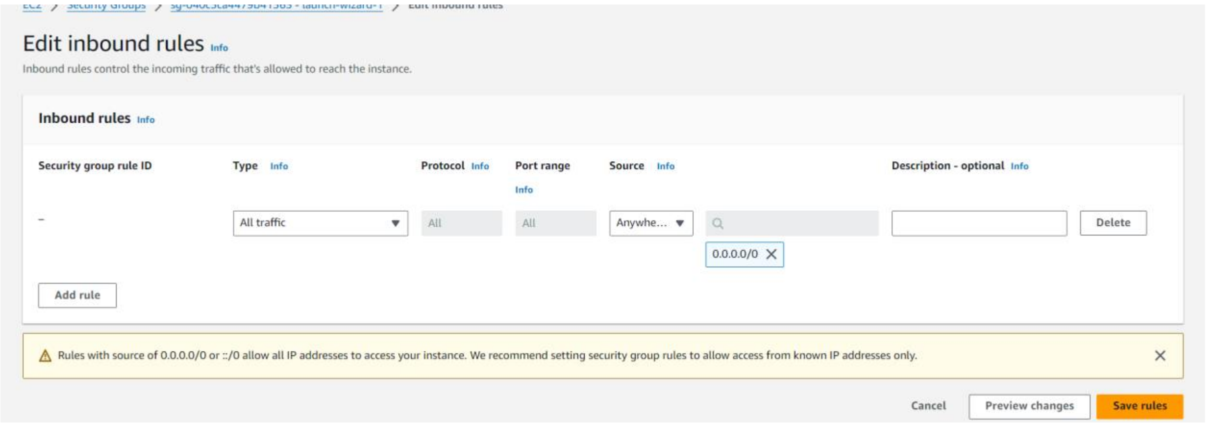
This security group has no inbound rules.

Add new rule

Select

ALL traffic

Anywhere IPV4



EC2 > Security Groups > sg-040c5ca4479b41563 - launch-wizard-1 > Edit inbound rules

Edit inbound rules Info

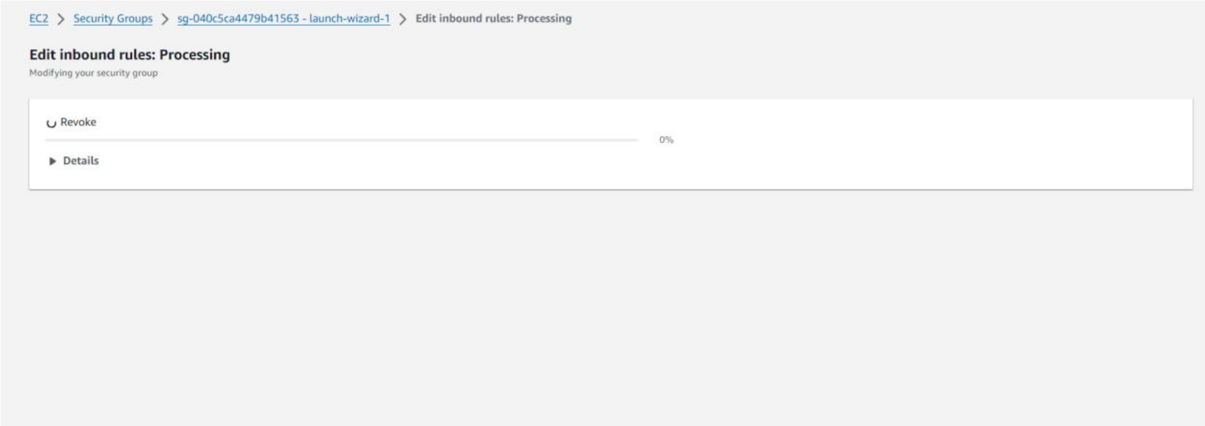
Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
–	All traffic	All	All	Anywhe...	0.0.0.0/0		Delete

Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Now save rules

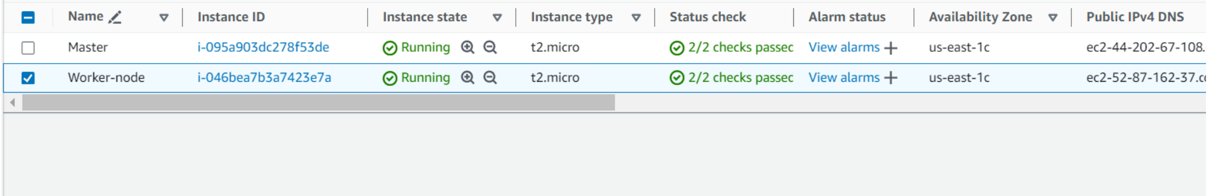


EC2 > Security Groups > sg-040c5ca4479b41563 - launch-wizard-1 > Edit inbound rules: Processing

Edit inbound rules: Processing

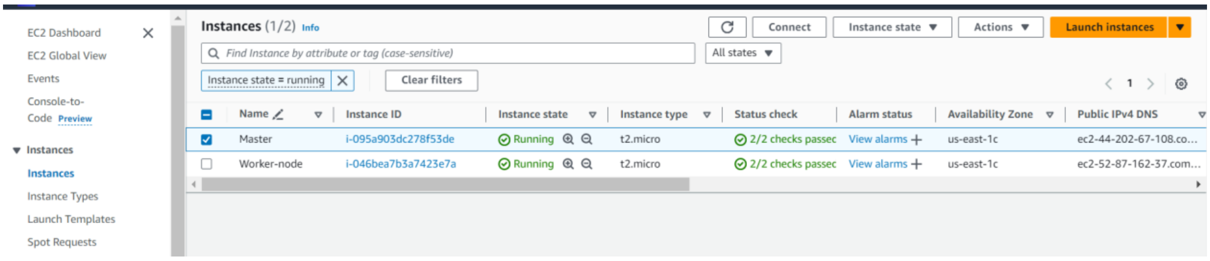
Modifying your security group

Name the instances



☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
☐	Master	i-095a903dc278f53de	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1c	ec2-44-202-67-108.
☑	Worker-node	i-046bea7b3a7423e7a	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1c	ec2-52-87-162-37.c

Select Master and connect

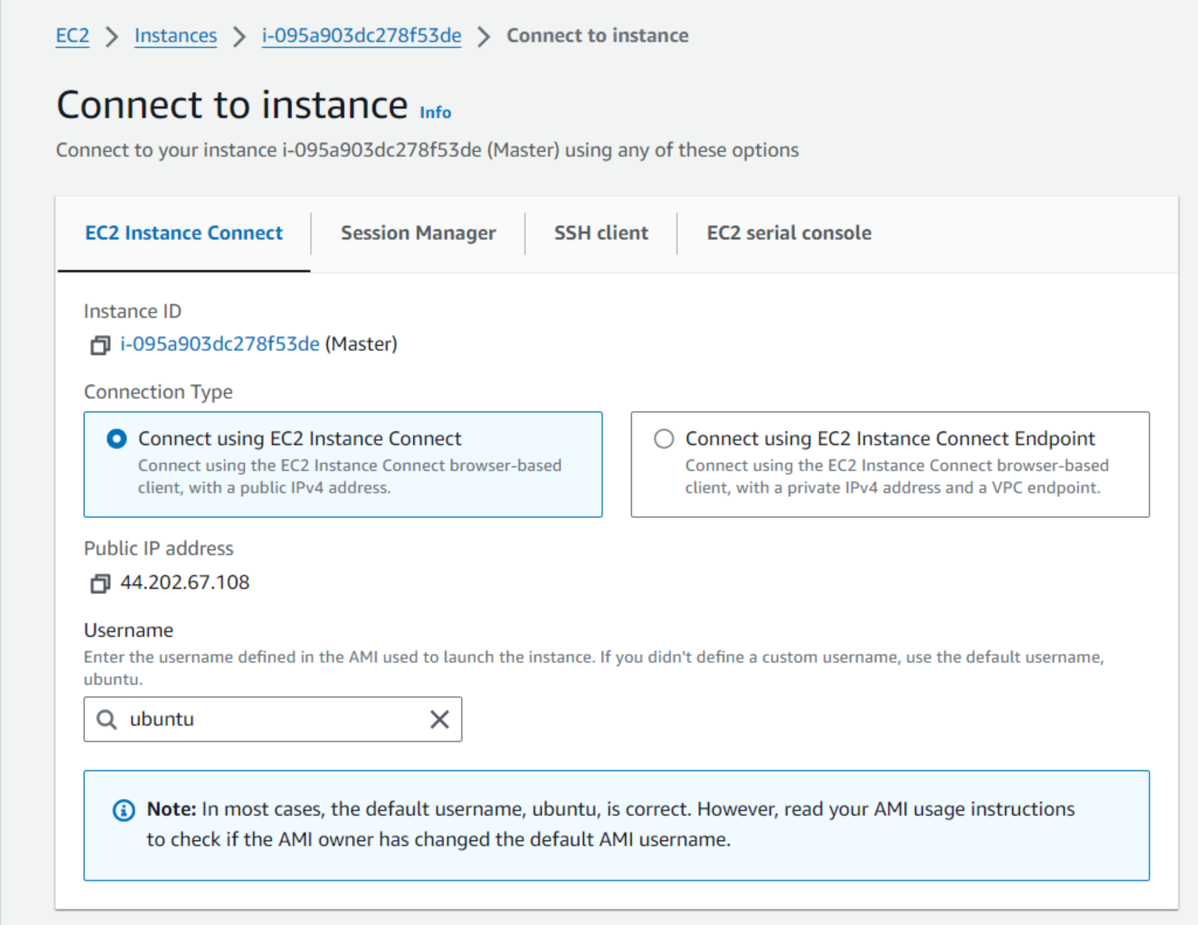


Instances (1/2) Info

Instance state = running

☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
☑	Master	i-095a903dc278f53de	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1c	ec2-44-202-67-108.co...
☐	Worker-node	i-046bea7b3a7423e7a	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1c	ec2-52-87-162-37.com...

Click on connect



EC2 > Instances > i-095a903dc278f53de > Connect to instance

Connect to instance Info

Connect to your instance i-095a903dc278f53de (Master) using any of these options

Instance ID

i-095a903dc278f53de (Master)

Connection Type

● Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.

○ Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address

44.202.67.108

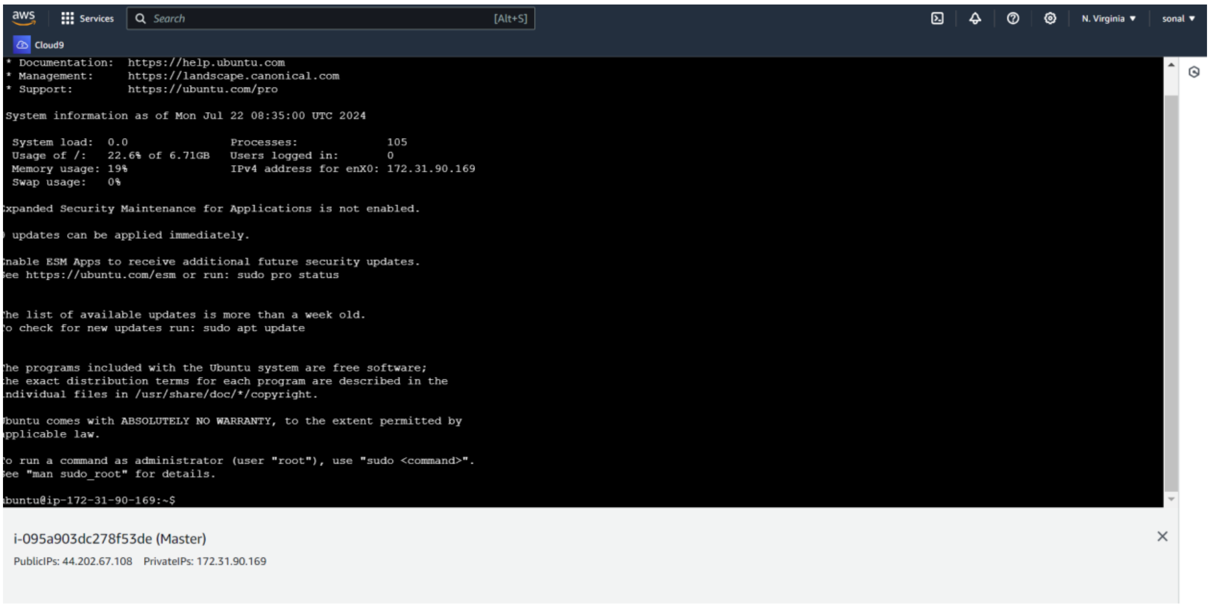
Username

Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.

ubuntu

Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

After Connecting

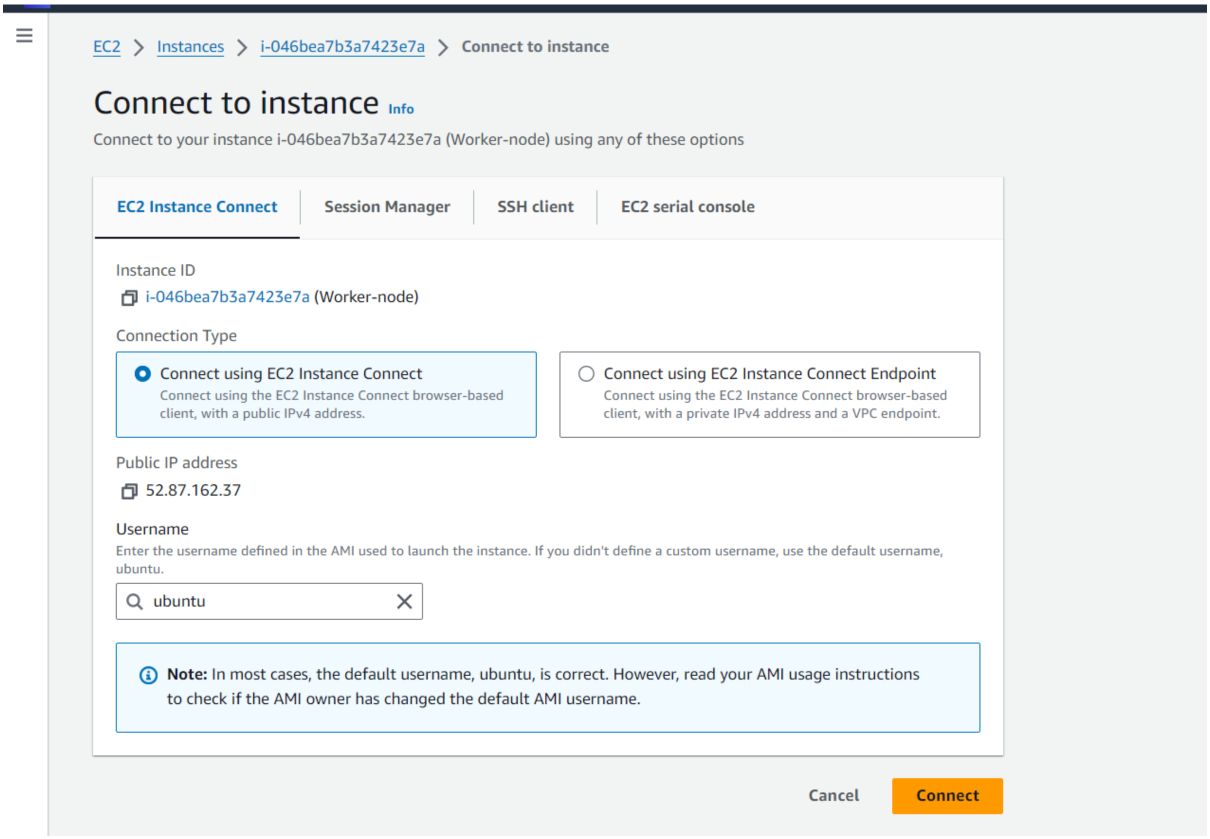


```
* Documentation:  https://help.ubuntu.com
* Management:     https://landscape.canonical.com
* Support:        https://ubuntu.com/pro

System information as of Mon Jul 22 08:35:00 UTC 2024

 System load:  0.0               Processes:             105
 Usage of /:   22.6% of 6.71GB   Users logged in:       0
 Memory usage: 19%               IPv4 address for enX0: 172.31.90.169
 Swap usage:   0%

xpanded Security Maintenance for Applications is not enabled.

 updates can be applied immediately.

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


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


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

buntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
pplicable law.

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

buntu@ip-172-31-90-169:~$
```

i-095a903dc278f53de (Master)

PublicIPs: 44.202.67.108   PrivateIPs: 172.31.90.169

Sameway connect to worker-node



Connect to instance

Connect to your instance i-046bea7b3a7423e7a (Worker-node) using any of these options

Instance ID

i-046bea7b3a7423e7a (Worker-node)

Connection Type

● Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.

○ Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address

52.87.162.37

Username

Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.

ubuntu

Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

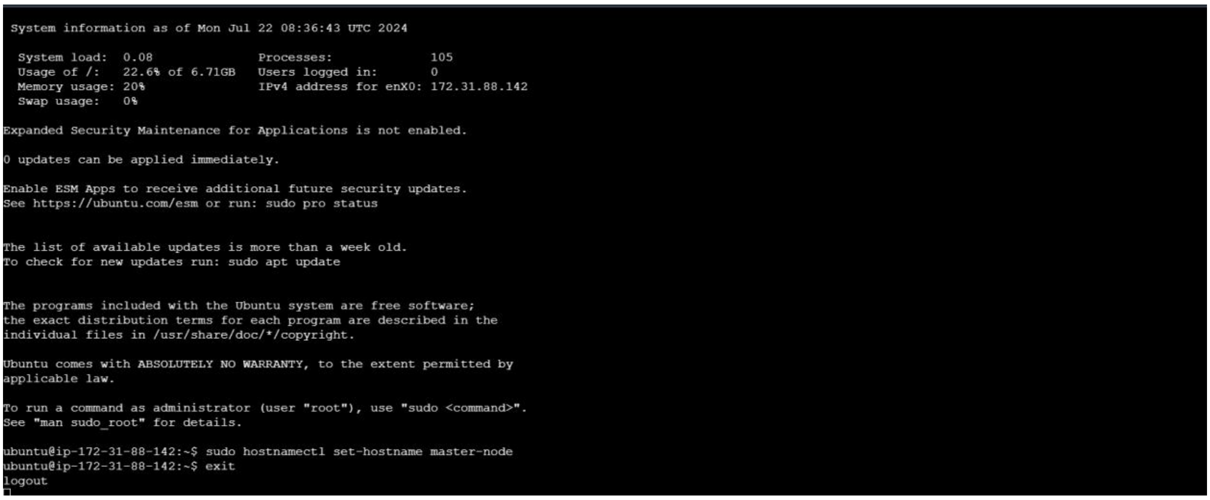
Step 2:

Assign Unique Hostname for Each Server Node

$ sudo hostnamectl set-hostname master-node

Than exit

Refresh



```
 System information as of Mon Jul 22 08:36:43 UTC 2024

  System load:  0.08              Processes:             105
  Usage of /:   22.6% of 6.71GB   Users logged in:       0
  Memory usage: 20%               IPv4 address for enX0: 172.31.88.142
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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


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


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

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

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

ubuntu@ip-172-31-88-142:~$ sudo hostnamectl set-hostname master-node
ubuntu@ip-172-31-88-142:~$ exit
logout
```

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

$ sudo hostnamectl set-hostname worker1

STEP 3:

On both master and worker1

$ sudo apt-get update

STEP 4:

On both master and worker1

Install docker

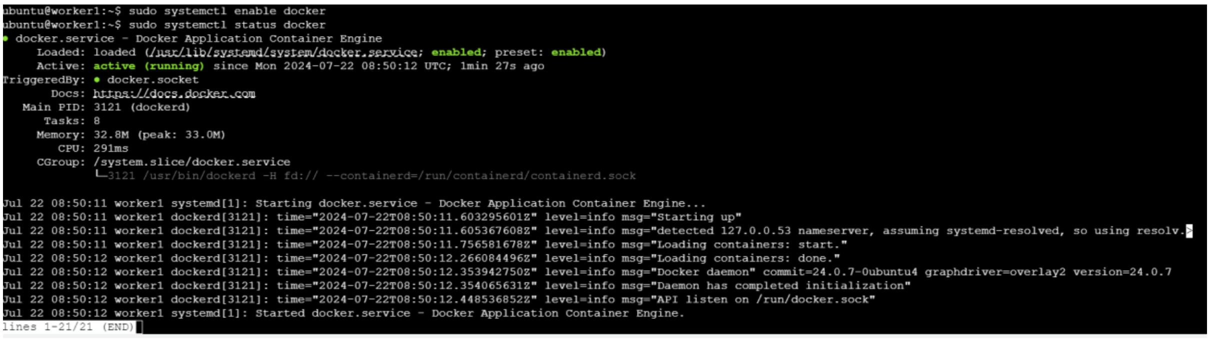
sudo apt-get install docker.io

STEP 5 : Start and Enable Docker

Set Docker to launch at boot by entering the following:

$ sudo systemctl enable docker

$ sudo systemctl status docker



```
ubuntu@worker1:~$ sudo systemctl enable docker
ubuntu@worker1:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
     Active: active (running) since Mon 2024-07-22 08:50:12 UTC; 1min 27s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 3121 (dockerd)
      Tasks: 8
     Memory: 32.8M (peak: 33.0M)
        CPU: 291ms
     CGroup: /system.slice/docker.service
             └─3121 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Jul 22 08:50:11 worker1 systemd[1]: Starting docker.service - Docker Application Container Engine...
Jul 22 08:50:11 worker1 dockerd[3121]: time="2024-07-22T08:50:11.603295601Z" level=info msg="Starting up"
Jul 22 08:50:11 worker1 dockerd[3121]: time="2024-07-22T08:50:11.605367608Z" level=info msg="detected 127.0.0.53 nameserver, assuming systemd-resolved, so using resolv.>
Jul 22 08:50:11 worker1 dockerd[3121]: time="2024-07-22T08:50:11.756581678Z" level=info msg="Loading containers: start."
Jul 22 08:50:12 worker1 dockerd[3121]: time="2024-07-22T08:50:12.266084496Z" level=info msg="Loading containers: done."
Jul 22 08:50:12 worker1 dockerd[3121]: time="2024-07-22T08:50:12.353942750Z" level=info msg="Docker daemon" commit=24.0.7-0ubuntu4 graphdriver=overlay2 version=24.0.7
Jul 22 08:50:12 worker1 dockerd[3121]: time="2024-07-22T08:50:12.354065631Z" level=info msg="Daemon has completed initialization"
Jul 22 08:50:12 worker1 dockerd[3121]: time="2024-07-22T08:50:12.448536852Z" level=info msg="API listen on /run/docker.sock"
Jul 22 08:50:12 worker1 systemd[1]: Started docker.service - Docker Application Container Engine.
lines 1-21/21 (END)
```

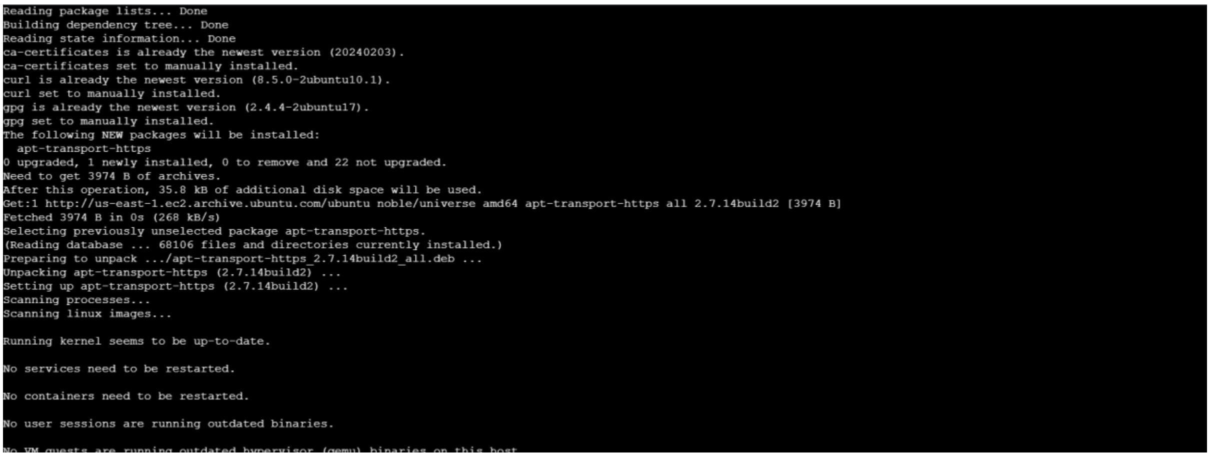
Ctrl+c

Clear

sudo systemctl start docker

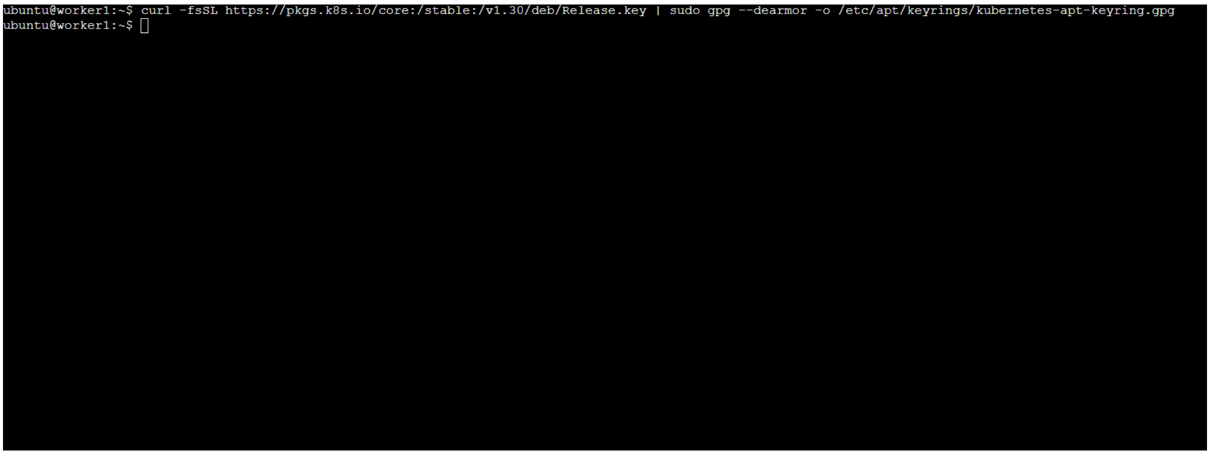
STEP 6 Install Kubernetes

https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/

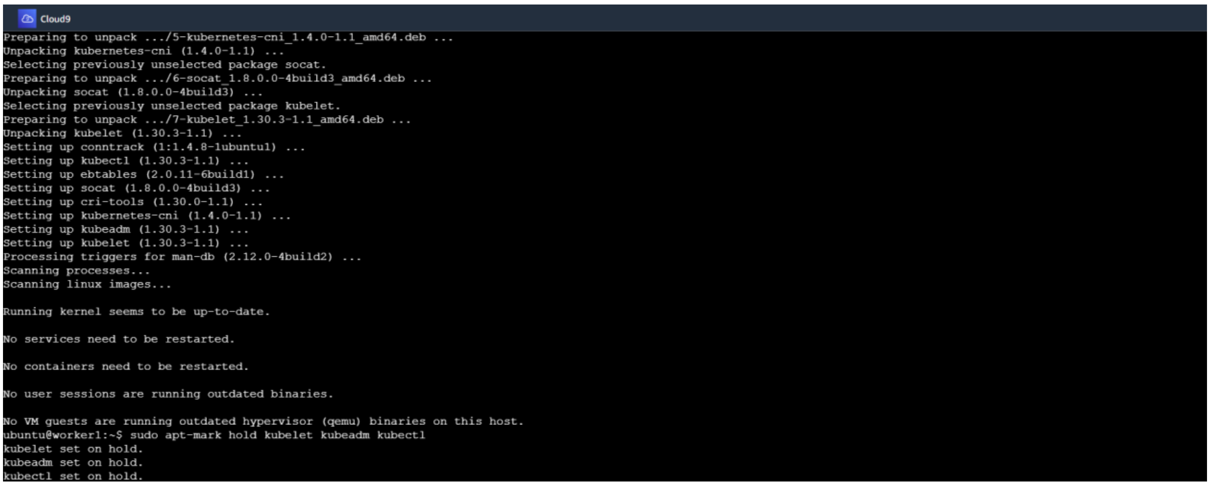


```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203).
ca-certificates set to manually installed.
curl is already the newest version (8.5.0-2ubuntu10.1).
curl set to manually installed.
gpg is already the newest version (2.4.4-2ubuntu17).
gpg set to manually installed.
The following NEW packages will be installed:
  apt-transport-https
0 upgraded, 1 newly installed, 0 to remove and 22 not upgraded.
Need to get 3974 B of archives.
After this operation, 35.8 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 apt-transport-https all 2.7.14build2 [3974 B]
Fetched 3974 B in 0s (268 kB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 68106 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.7.14build2_all.deb ...
Unpacking apt-transport-https (2.7.14build2) ...
Setting up apt-transport-https (2.7.14build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

Signining key



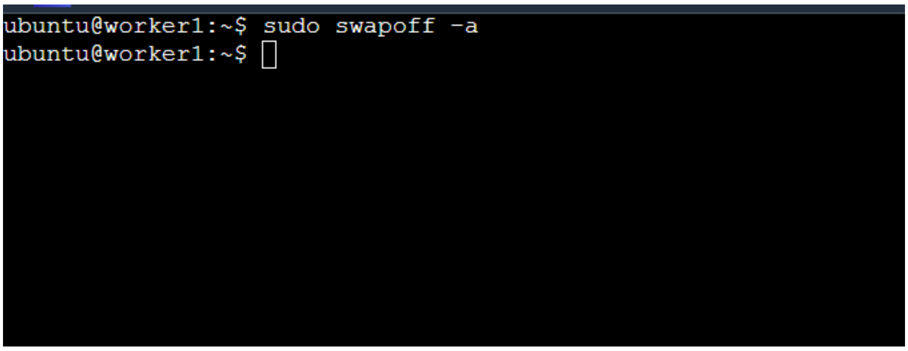
```
ubuntu@worker1:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.30/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
ubuntu@worker1:~$ 
```



```
Preparing to unpack .../5-kubernetes-cni_1.4.0-1.1_amd64.deb ...
Unpacking kubernetes-cni (1.4.0-1.1) ...
Selecting previously unselected package socat.
Preparing to unpack .../6-socat_1.8.0.0-4build3_amd64.deb ...
Unpacking socat (1.8.0.0-4build3) ...
Selecting previously unselected package kubelet.
Preparing to unpack .../7-kubelet_1.30.3-1.1_amd64.deb ...
Unpacking kubelet (1.30.3-1.1) ...
Setting up conntrack (1:1.4.8-1ubuntu1) ...
Setting up kubectl (1.30.3-1.1) ...
Setting up ebtables (2.0.11-6build1) ...
Setting up socat (1.8.0.0-4build3) ...
Setting up cri-tools (1.30.0-1.1) ...
Setting up kubernetes-cni (1.4.0-1.1) ...
Setting up kubeadm (1.30.3-1.1) ...
Setting up kubelet (1.30.3-1.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@worker1:~$ sudo apt-mark hold kubelet kubeadm kubectl
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
```

Step 7 : Begin Kubernetes Deployment

$ sudo swapoff –a



```
ubuntu@worker1:~$ sudo swapoff -a
ubuntu@worker1:~$ 
```

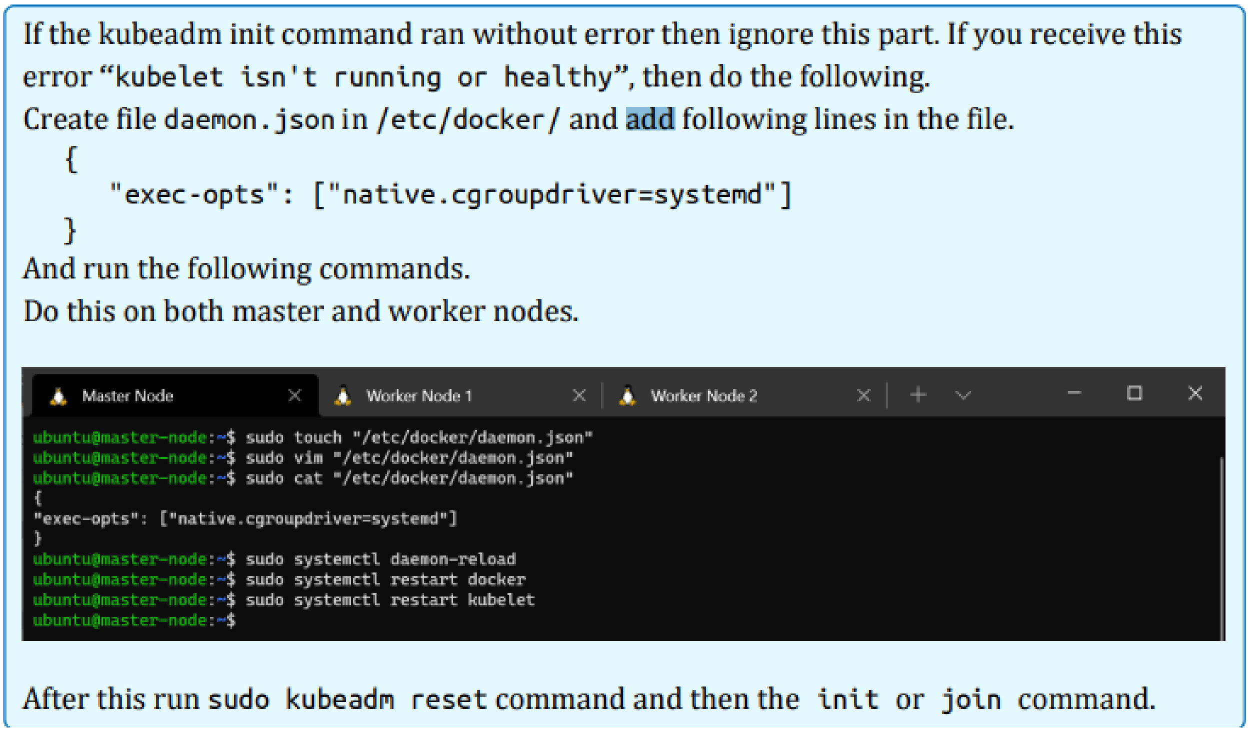
STEP 8:

Initialize Kubernetes on Master Node

$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=all

If getting error

Run below code on both



If the kubeadm init command ran without error then ignore this part. If you receive this error "`kubelet isn't running or healthy`", then do the following.

Create file `daemon.json` in `/etc/docker/` and add following lines in the file.

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

And run the following commands.

Do this on both master and worker nodes.

```
ubuntu@master-node:~$ sudo touch "/etc/docker/daemon.json"
ubuntu@master-node:~$ sudo vim "/etc/docker/daemon.json"
ubuntu@master-node:~$ sudo cat "/etc/docker/daemon.json"
{
"exec-opts": ["native.cgroupdriver=systemd"]
}
ubuntu@master-node:~$ sudo systemctl daemon-reload
ubuntu@master-node:~$ sudo systemctl restart docker
ubuntu@master-node:~$ sudo systemctl restart kubelet
ubuntu@master-node:~$
```

After this run `sudo kubeadm reset` command and then the `init` or `join` command.

$ sudo touch "/etc/docker/daemon.json"

$ sudo nano "/etc/docker/daemon.json"

ADD code

CTRL+ O enter CTRL X

$ sudo cat "/etc/docker/daemon.json"

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

$ sudo systemctl daemon-reload

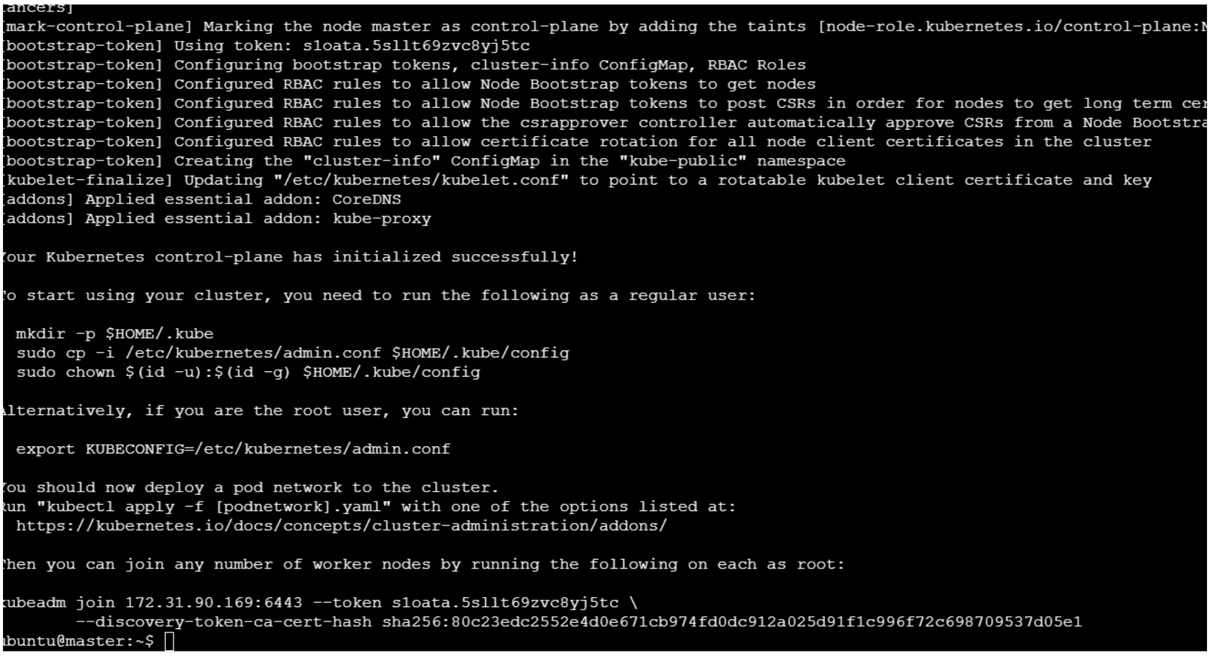
$ sudo systemctl restart docker

$ sudo systemctl restart kubelet

$ sudo kubeadm reset

STEP 9 on master node

sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=all



```
ancers]
[mark-control-plane] Marking the node master as control-plane by adding the taints [node-role.kubernetes.io/control-plane:N
[bootstrap-token] Using token: s1oata.5sllt69zvc8yj5tc
[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 cer
[bootstrap-token] Configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstra
[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

our Kubernetes control-plane has initialized successfully!

o 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

lternatively, if you are the root user, you can run:

 export KUBECONFIG=/etc/kubernetes/admin.conf

ou should now deploy a pod network to the cluster.
un "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
 https://kubernetes.io/docs/concepts/cluster-administration/addons/

hen you can join any number of worker nodes by running the following on each as root:

ubeadm join 172.31.90.169:6443 --token s1oata.5sllt69zvc8yj5tc \
       --discovery-token-ca-cert-hash sha256:80c23edc2552e4d0e671cb974fd0dc912a025d91f1c996f72c698709537d05e1
buntu@master:~$ 
```

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

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

copy

```
kubeadm join 172.31.90.169:6443 --token s1oata.5sllt69zvc8yj5tc \
        --discovery-token-ca-cert-hash sha256:80c23edc2552e4d0e671cb974fd0dc912a025d91f1c996f72c698709537d05e1
```

Makeit as and copy in worker after flannel is created on master(after step 10)

```
kubeadm join 172.31.90.169:6443 --token s1oata.5sllt69zvc8yj5tc --discovery-token-ca-cert-hash sha256:80c23edc2552e4d0e671cb974fd0dc912a025d91f1c996f72c698709537d05e1
```

it will give error

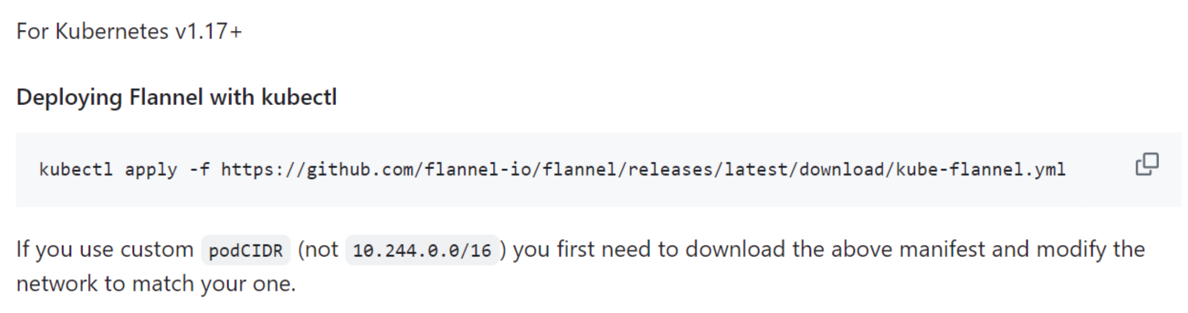
```
sudo kubeadm join 172.31.90.169:6443 --token s1oata.5sllt69zvc8yj5tc --discovery-token-ca-cert-hash sha256:80c23edc2552e4d0e671cb974fd0dc912a025d91f1c996f72c698709537d05e1 --ignore-preflight-errors=all
```

STEP 10 Copy weblink from masternode

https://kubernetes.io/docs/concepts/cluster-administration/addons/

goto flannel

copy this command and paste on master



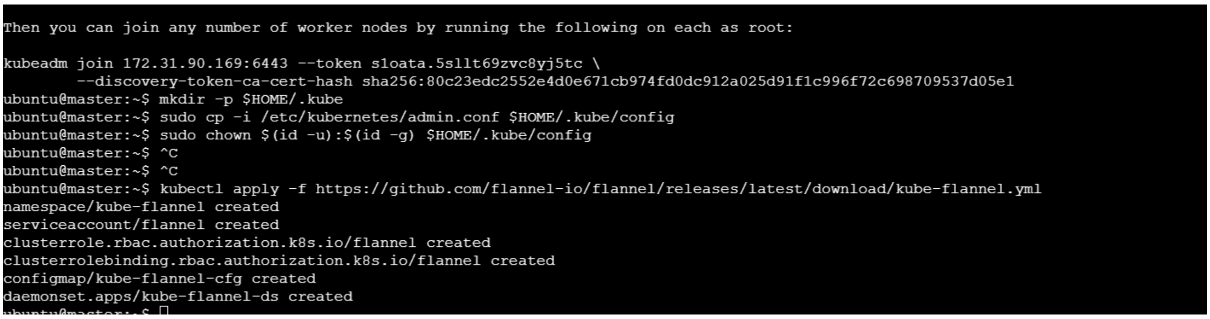
For Kubernetes v1.17+

Deploying Flannel with kubectl

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

If you use custom `podCIDR` (not `10.244.0.0/16`) you first need to download the above manifest and modify the network to match your one.

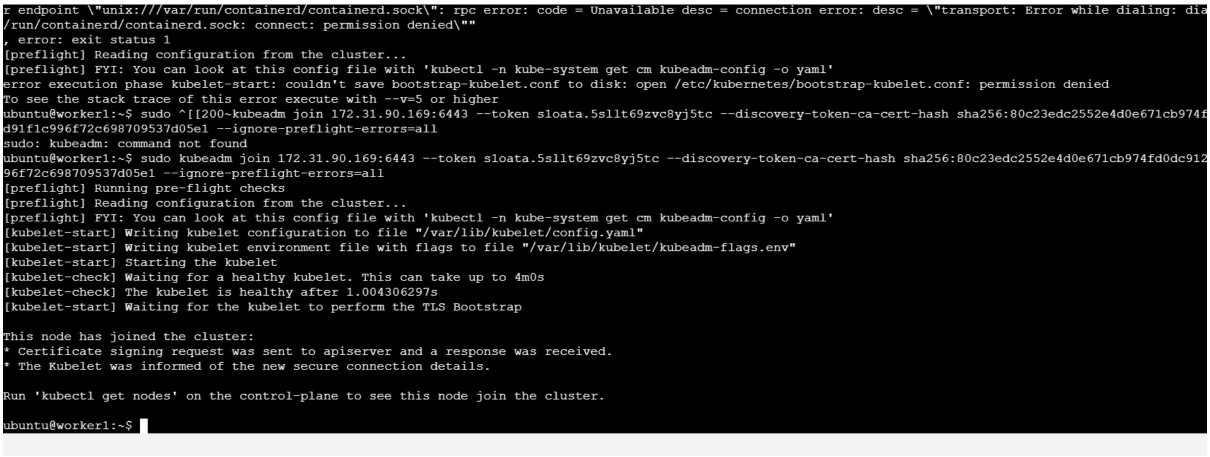
Flannel created



```
Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 172.31.90.169:6443 --token s1oata.5sllt69zvc8yj5tc \
        --discovery-token-ca-cert-hash sha256:80c23edc2552e4d0e671cb974fd0dc912a025d91f1c996f72c698709537d05e1
ubuntu@master:~$ mkdir -p $HOME/.kube
ubuntu@master:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
ubuntu@master:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@master:~$ ^C
ubuntu@master:~$ ^C
ubuntu@master:~$ 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@master:~$
```

Step 11:

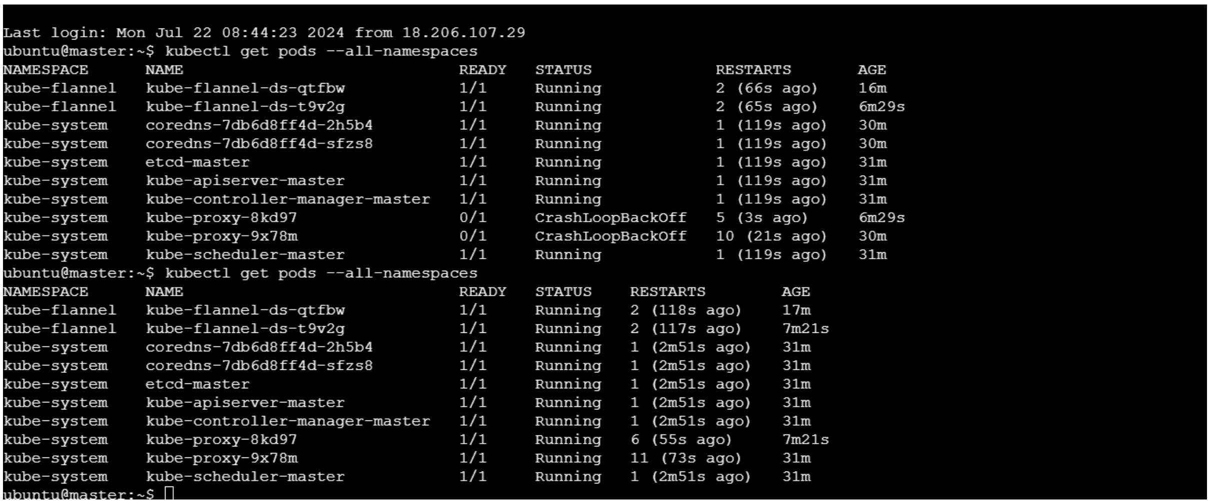
AFTER



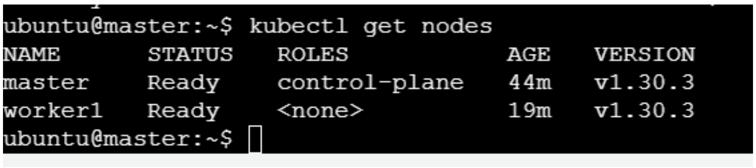
```
r endpoint \"unix:///var/run/containerd/containerd.sock\": rpc error: code = Unavailable desc = connection error: desc = \"transport: Error while dialing: dia
/run/containerd/containerd.sock: connect: permission denied\""
, error: exit status 1
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
error execution phase kubelet-start: couldn't save bootstrap-kubelet.conf to disk: open /etc/kubernetes/bootstrap-kubelet.conf: permission denied
To see the stack trace of this error execute with --v=5 or higher
ubuntu@worker1:~$ sudo ^[[200~kubeadm join 172.31.90.169:6443 --token s1oata.5sllt69zvc8yj5tc --discovery-token-ca-cert-hash sha256:80c23edc2552e4d0e671cb974f
d91f1c996f72c698709537d05e1 --ignore-preflight-errors=all
sudo: kubeadm: command not found
ubuntu@worker1:~$ sudo kubeadm join 172.31.90.169:6443 --token s1oata.5sllt69zvc8yj5tc --discovery-token-ca-cert-hash sha256:80c23edc2552e4d0e671cb974fd0dc912
96f72c698709537d05e1 --ignore-preflight-errors=all
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 1.004306297s
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

ubuntu@worker1:~$
```

Step 12:

Reboot both instances



```
Last login: Mon Jul 22 08:44:23 2024 from 18.206.107.29
ubuntu@master:~$ kubectl get pods --all-namespaces
NAMESPACE      NAME                              READY   STATUS             RESTARTS      AGE
kube-flannel   kube-flannel-ds-qtfbw             1/1     Running            2 (66s ago)   16m
kube-flannel   kube-flannel-ds-t9v2g             1/1     Running            2 (65s ago)   6m29s
kube-system    coredns-7db6d8ff4d-2h5b4          1/1     Running            1 (119s ago)  30m
kube-system    coredns-7db6d8ff4d-sfzs8          1/1     Running            1 (119s ago)  30m
kube-system    etcd-master                       1/1     Running            1 (119s ago)  31m
kube-system    kube-apiserver-master             1/1     Running            1 (119s ago)  31m
kube-system    kube-controller-manager-master    1/1     Running            1 (119s ago)  31m
kube-system    kube-proxy-8kd97                  0/1     CrashLoopBackOff   5 (3s ago)    6m29s
kube-system    kube-proxy-9x78m                  0/1     CrashLoopBackOff   10 (21s ago)  30m
kube-system    kube-scheduler-master             1/1     Running            1 (119s ago)  31m
ubuntu@master:~$ kubectl get pods --all-namespaces
NAMESPACE      NAME                              READY   STATUS    RESTARTS       AGE
kube-flannel   kube-flannel-ds-qtfbw             1/1     Running   2 (118s ago)   17m
kube-flannel   kube-flannel-ds-t9v2g             1/1     Running   2 (117s ago)   7m21s
kube-system    coredns-7db6d8ff4d-2h5b4          1/1     Running   1 (2m51s ago)  31m
kube-system    coredns-7db6d8ff4d-sfzs8          1/1     Running   1 (2m51s ago)  31m
kube-system    etcd-master                       1/1     Running   1 (2m51s ago)  31m
kube-system    kube-apiserver-master             1/1     Running   1 (2m51s ago)  31m
kube-system    kube-controller-manager-master    1/1     Running   1 (2m51s ago)  31m
kube-system    kube-proxy-8kd97                  1/1     Running   6 (55s ago)    7m21s
kube-system    kube-proxy-9x78m                  1/1     Running   11 (73s ago)   31m
kube-system    kube-scheduler-master             1/1     Running   1 (2m51s ago)  31m
ubuntu@master:~$
```



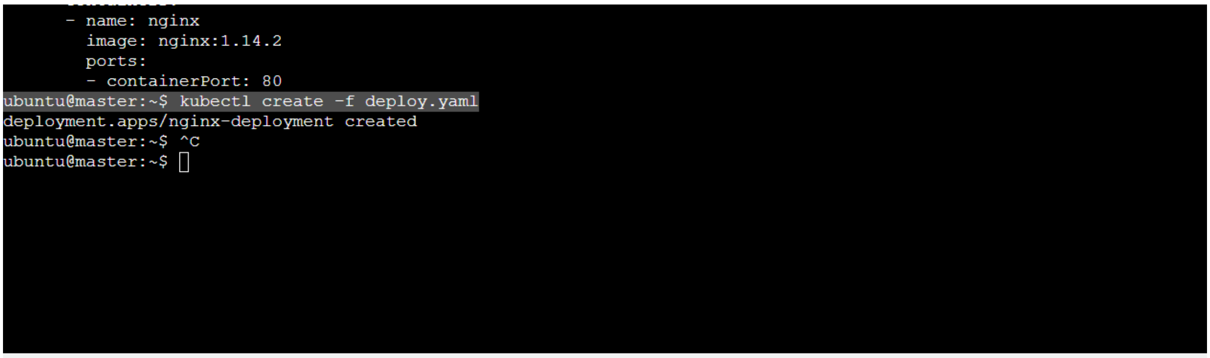
```
ubuntu@master:~$ kubectl get nodes
NAME      STATUS   ROLES           AGE   VERSION
master    Ready    control-plane   44m   v1.30.3
worker1   Ready    <none>          19m   v1.30.3
ubuntu@master:~$
```

Deploy service

On browser search for ngnix deployment yaml

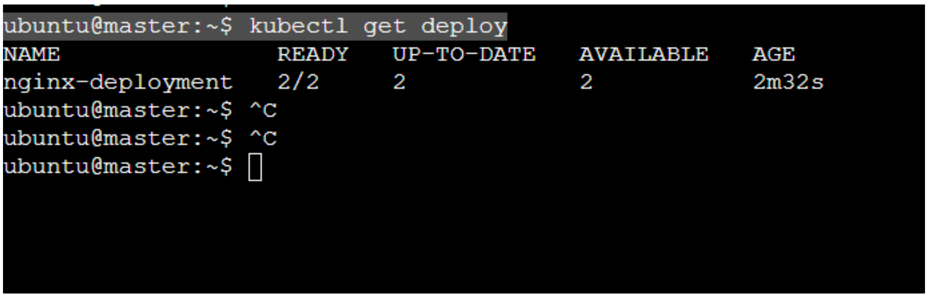
```
ubuntu@master:~$ sudo nano deploy.yaml
ubuntu@master:~$ sudo cat deploy.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 2 # tells deployment to run 2 pods matching the template
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
```

ubuntu@master:~$ kubectl create -f deploy.yaml



```
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
ubuntu@master:~$ kubectl create -f deploy.yaml
deployment.apps/nginx-deployment created
ubuntu@master:~$ ^C
ubuntu@master:~$ 
```

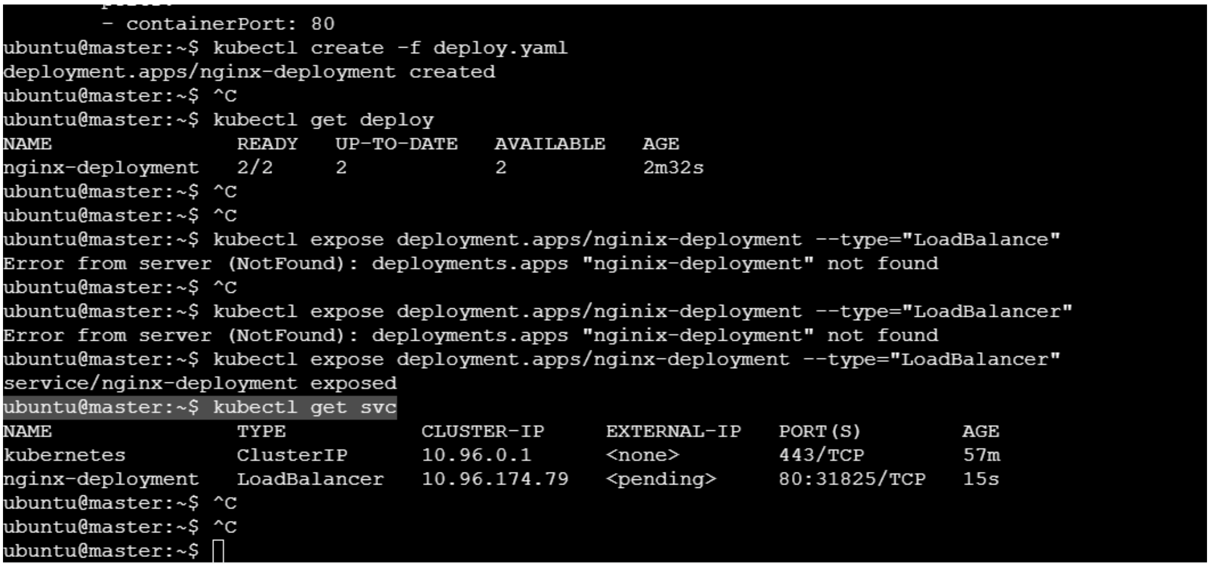
ubuntu@master:~$ kubectl get deploy



```
ubuntu@master:~$ kubectl get deploy
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2     2            2           2m32s
ubuntu@master:~$ ^C
ubuntu@master:~$ ^C
ubuntu@master:~$ 
```

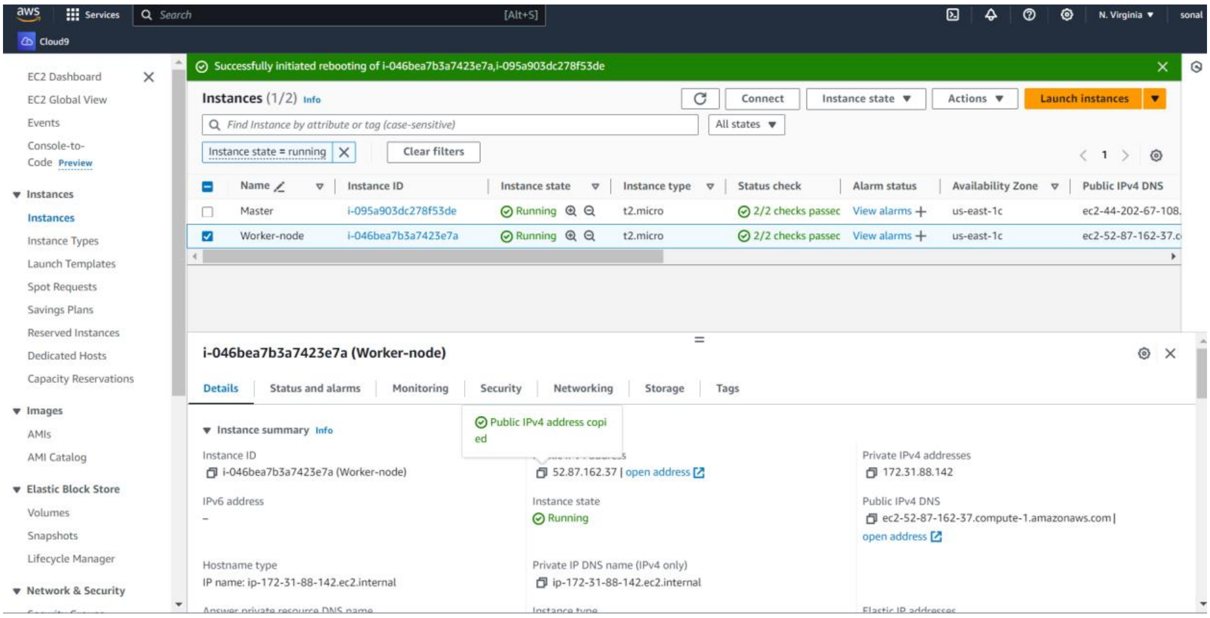
kubectl expose deployment.apps/nginix-deployment --type="LoadBalancer"

ubuntu@master:~$ kubectl get svc



```
        - containerPort: 80
ubuntu@master:~$ kubectl create -f deploy.yaml
deployment.apps/nginx-deployment created
ubuntu@master:~$ ^C
ubuntu@master:~$ kubectl get deploy
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2     2            2           2m32s
ubuntu@master:~$ ^C
ubuntu@master:~$ ^C
ubuntu@master:~$ kubectl expose deployment.apps/nginix-deployment --type="LoadBalance"
Error from server (NotFound): deployments.apps "nginix-deployment" not found
ubuntu@master:~$ ^C
ubuntu@master:~$ kubectl expose deployment.apps/nginix-deployment --type="LoadBalancer"
Error from server (NotFound): deployments.apps "nginix-deployment" not found
ubuntu@master:~$ kubectl expose deployment.apps/nginx-deployment --type="LoadBalancer"
service/nginx-deployment exposed
ubuntu@master:~$ kubectl get svc
NAME               TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
kubernetes         ClusterIP      10.96.0.1      <none>        443/TCP        57m
nginx-deployment   LoadBalancer   10.96.174.79   <pending>     80:31825/TCP   15s
ubuntu@master:~$ ^C
ubuntu@master:~$ ^C
ubuntu@master:~$ 
```

Go to instance,master select public ipv4



Successfully initiated rebooting of i-046bea7b3a7423e7a,i-095a903dc278f53de

Instances (1/2) Info

Find Instance by attribute or tag (case-sensitive)

Instance state = running

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
□	Master	i-095a903dc278f53de	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1c	ec2-44-202-67-108.
☑	Worker-node	i-046bea7b3a7423e7a	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1c	ec2-52-87-162-37.c

i-046bea7b3a7423e7a (Worker-node)

Instance summary Info

Public IPv4 address copied

Instance ID: i-046bea7b3a7423e7a (Worker-node)

52.87.162.37 | open address

Private IPv4 addresses: 172.31.88.142

IPv6 address: –

Instance state: Running

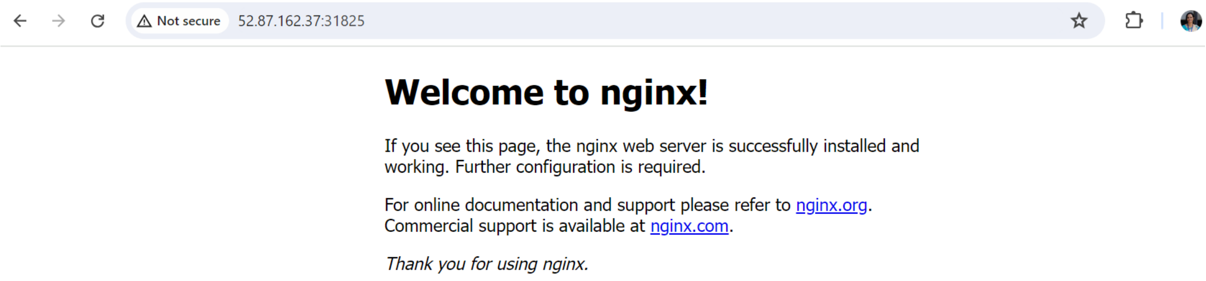
Public IPv4 DNS: ec2-52-87-162-37.compute-1.amazonaws.com | open address

Hostname type: IP name: ip-172-31-88-142.ec2.internal

Private IP DNS name (IPv4 only): ip-172-31-88-142.ec2.internal

Go to brower

Ipv4:portnumber



Not secure 52.87.162.37:31825

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.