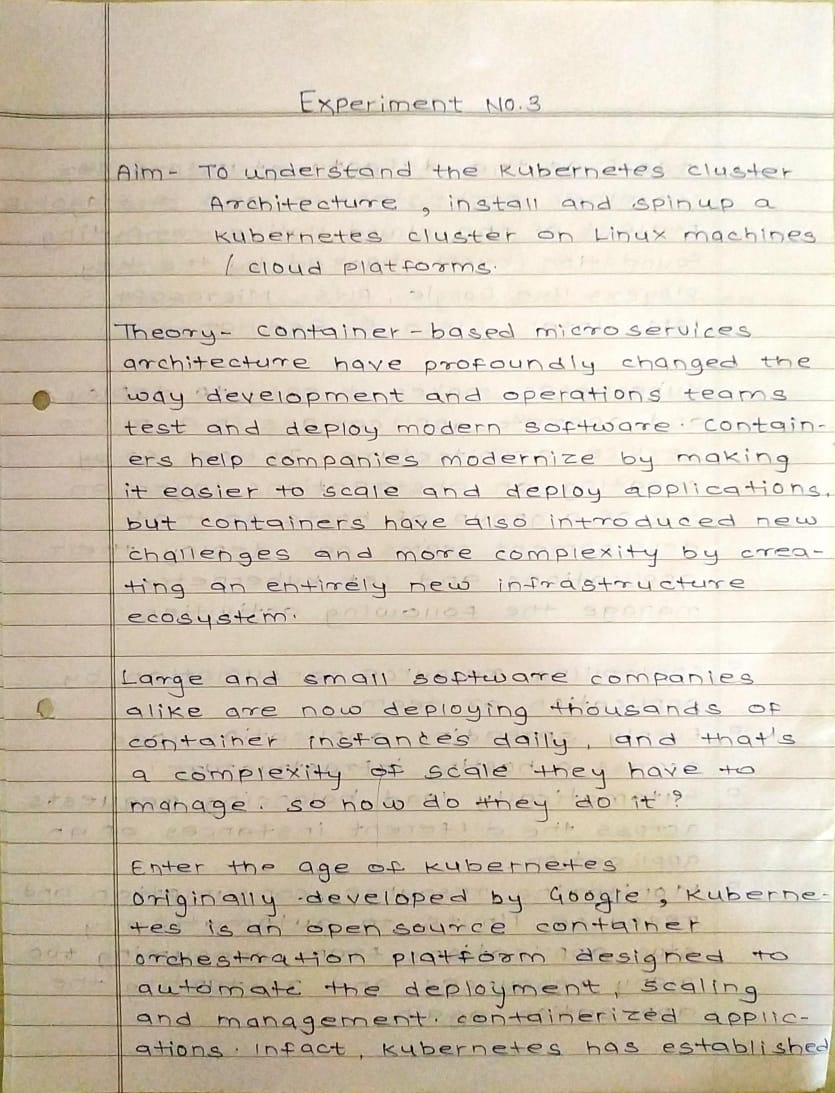
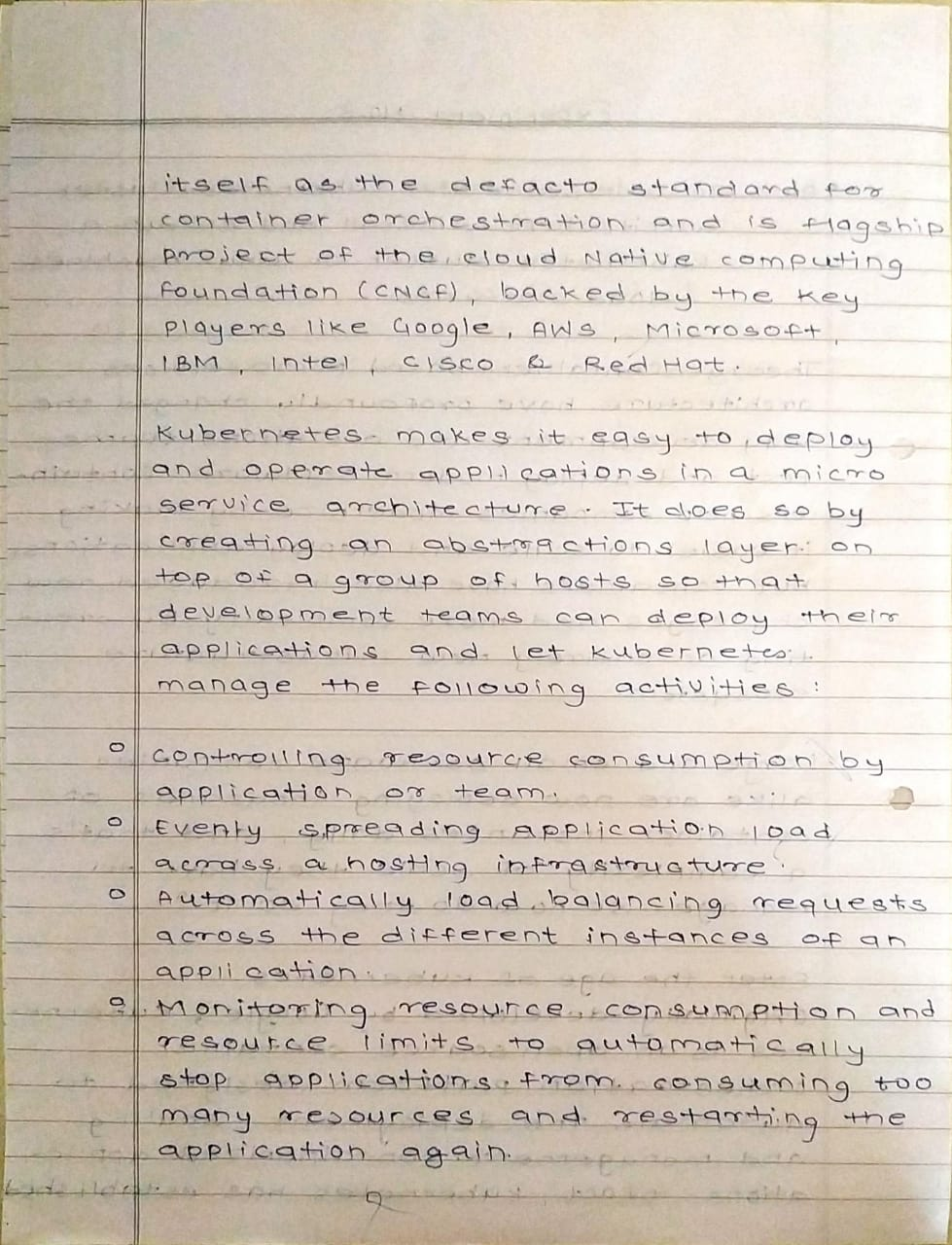
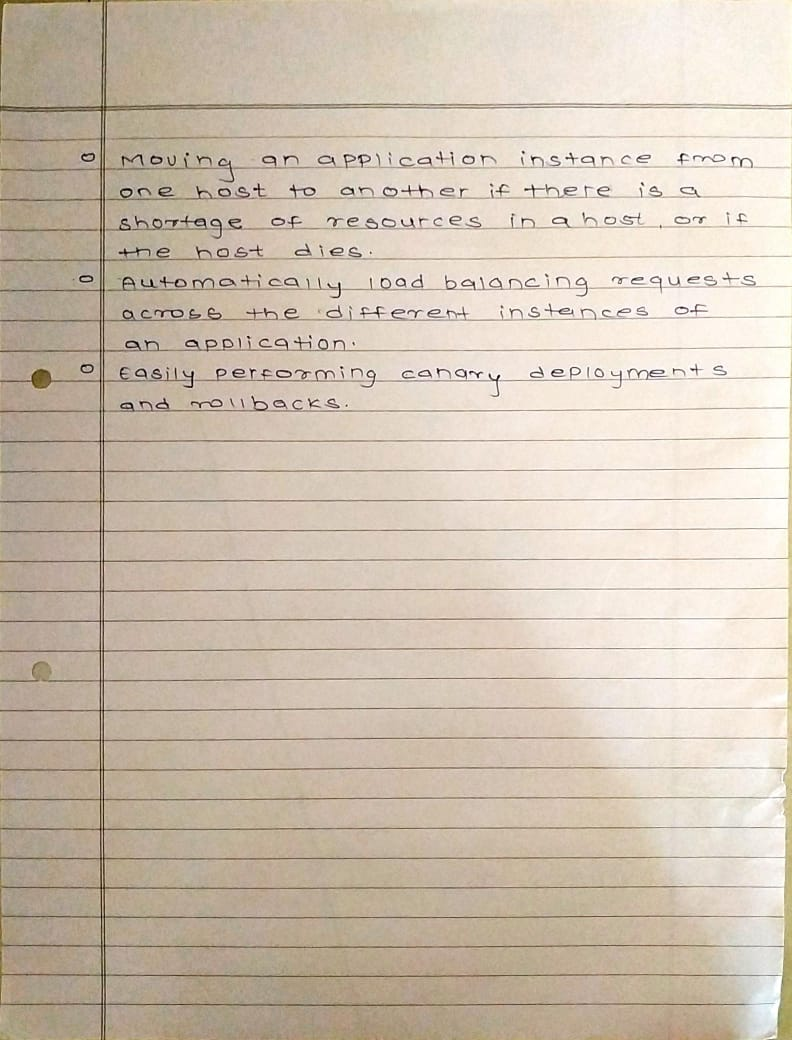
**Adv.DevOps Exp 03**

**Name- Ishika Devare RollNo- 14**

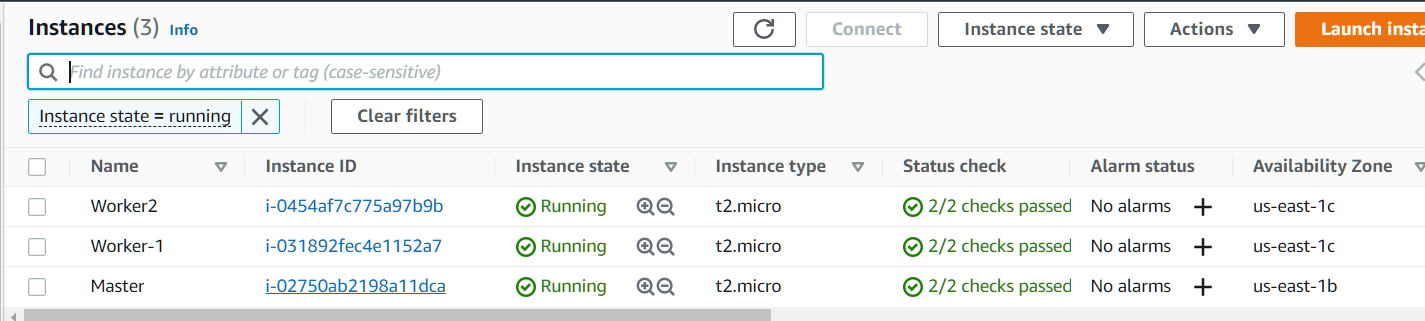
****

****

****

# Implementation:

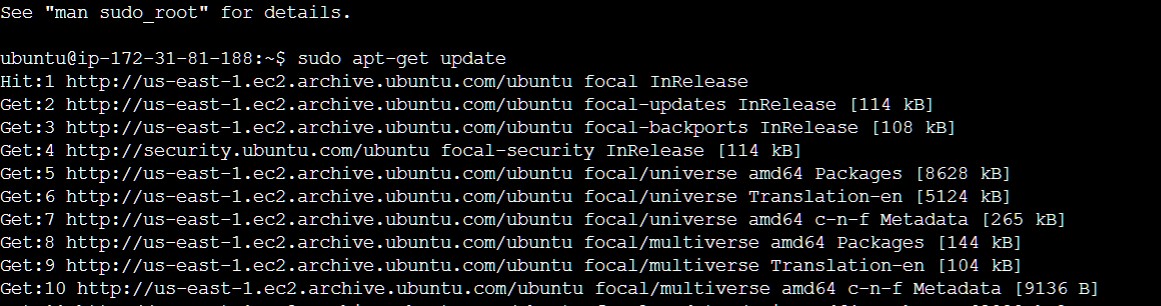
Create 3 EC2 Ubuntu Instances of Ubuntu version 20.04 and keep all the instances in the same security group on AWS.

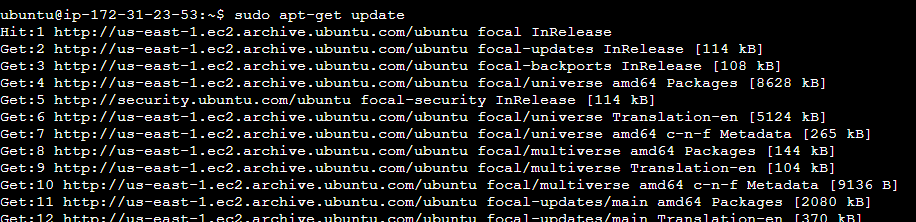
(Name 1 as Master, the other 2 as worker-1 and worker-2)

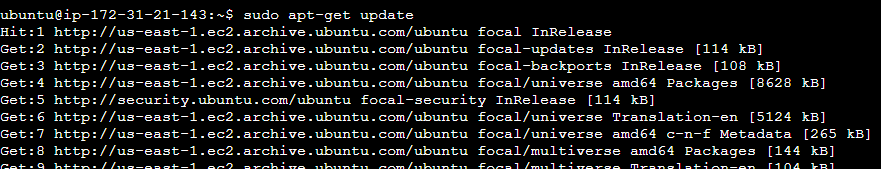
**Steps to Install Kubernetes on Ubuntu Step 1:** Install Docker

1. Update the package list with the command:

**$sudo apt-get update**

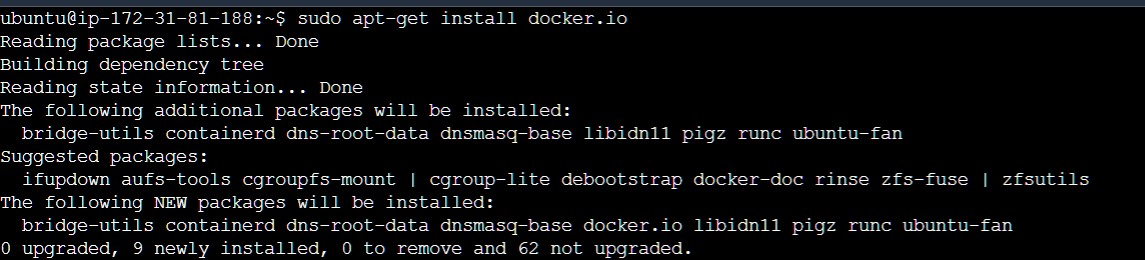
Master

Worker-1

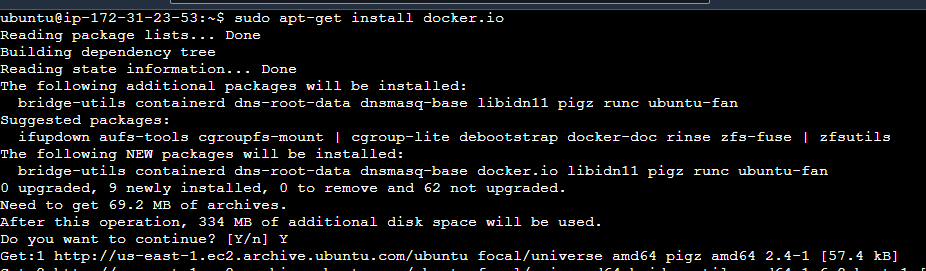


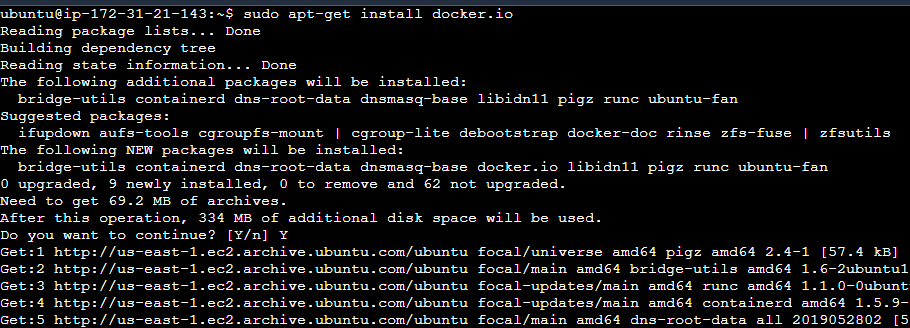
1. Next, **install Docker** with the command:

**$sudo apt-get install docker.io**

Master

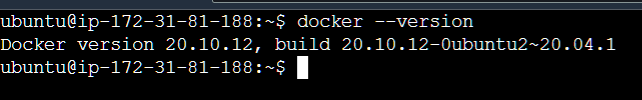
1. Repeat the process on each server that will act as a node.

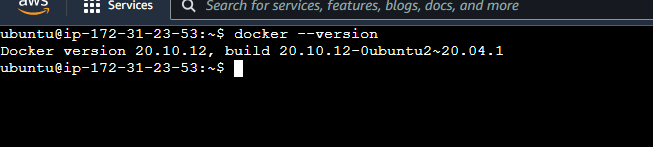
Worker 1



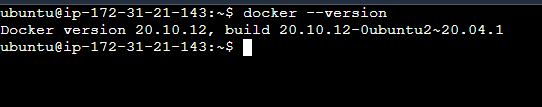
1. Check the installation (and version) by entering the following:

$docker --version

Master

Worker1

Worker2

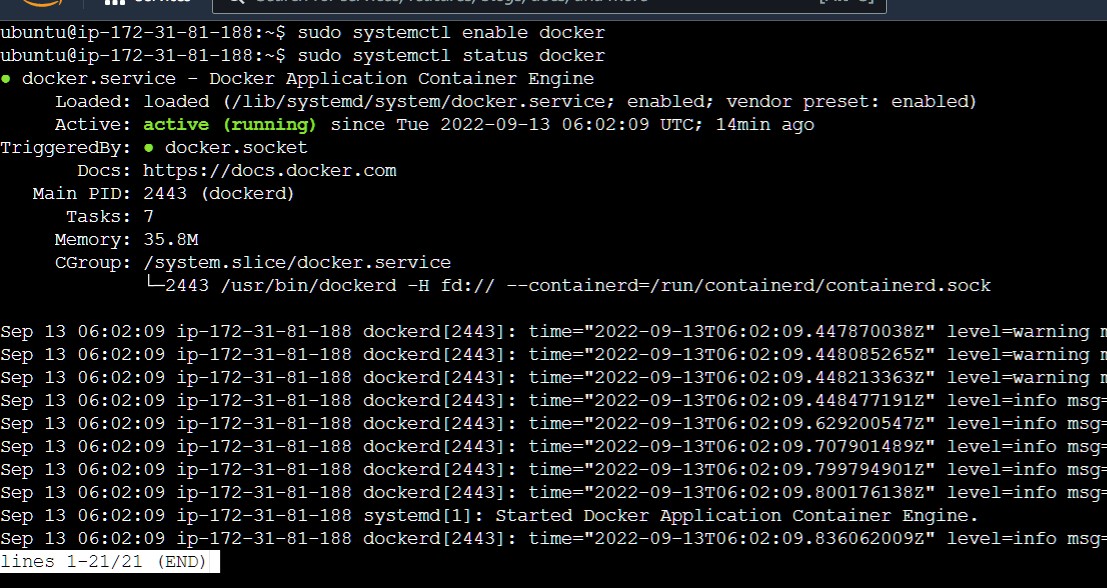


1. Set Docker to launch at boot by entering the following:

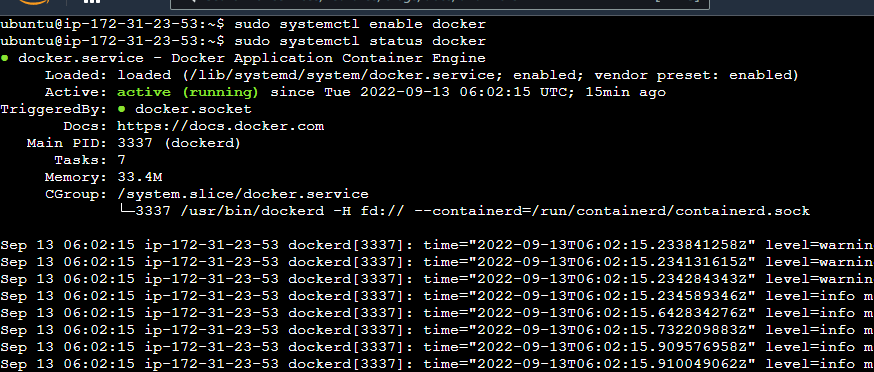
**$sudo systemctl enable docker**

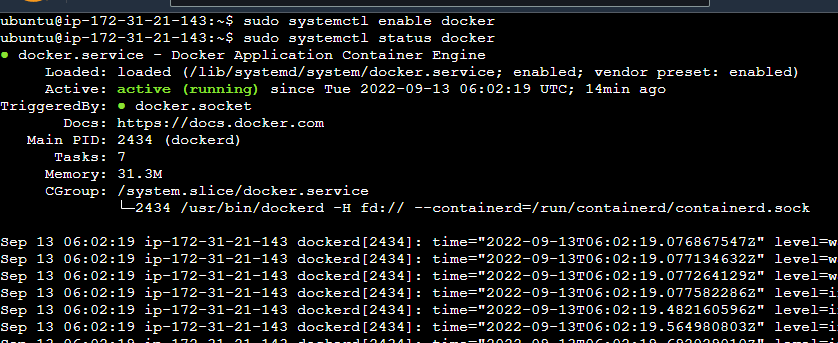
1. Verify Docker is running:

**$sudo systemctl status docker**

Master

Worker 1

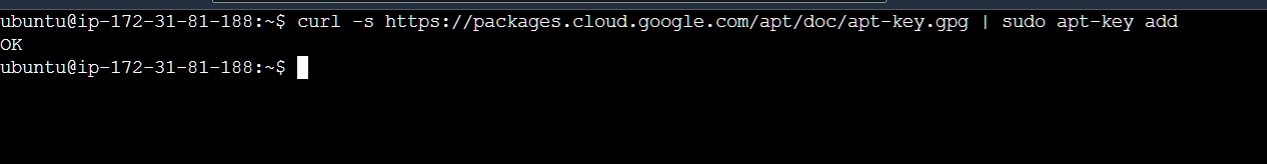


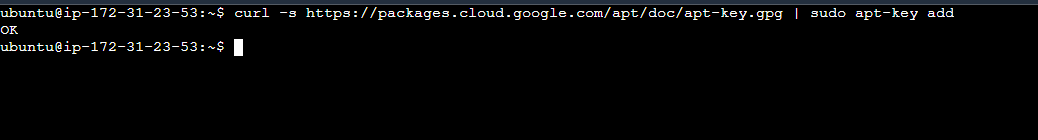


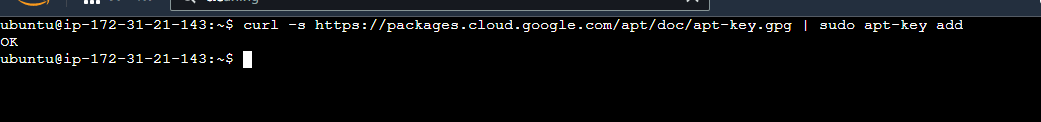
**Step 3:** Add Kubernetes Signing Key

1. Enter the following to add a signing key:

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

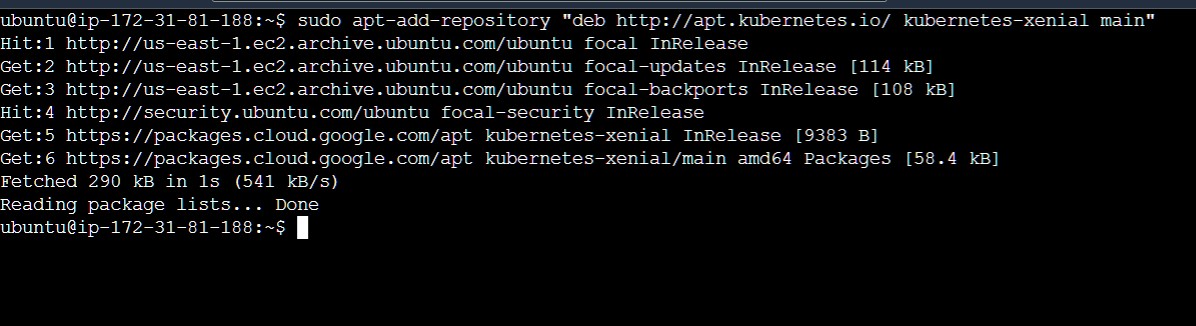
Master

Worker1

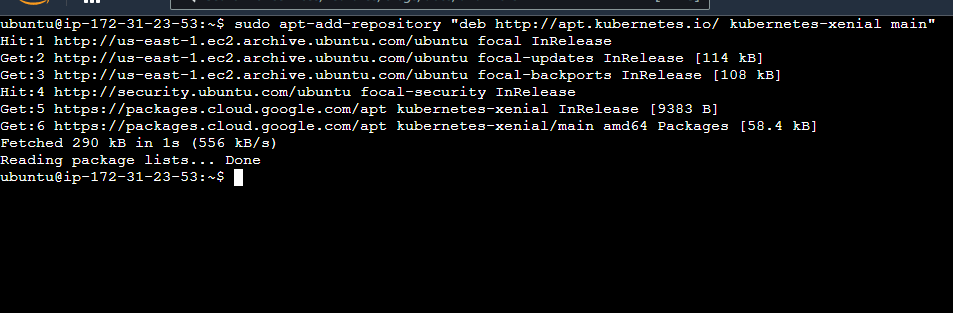
Worker2

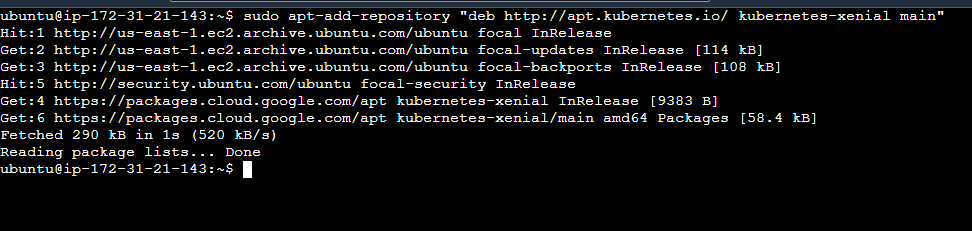
Kubernetes is not included in the default repositories. To add them, enter the following:

**$sudo apt-add-repository "deb** [**http://apt.kubernetes.io/**](http://apt.kubernetes.io/) **kubernetes-xenial main"**

Master

Worker1

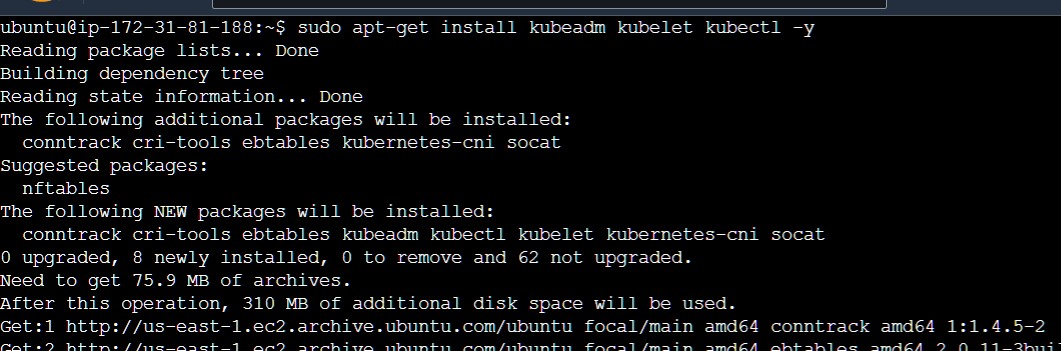


Worker2

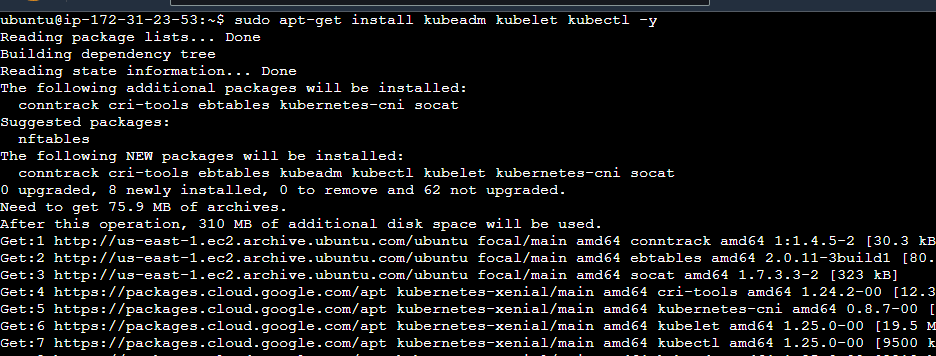
**Step 5:** Kubernetes Installation Tools

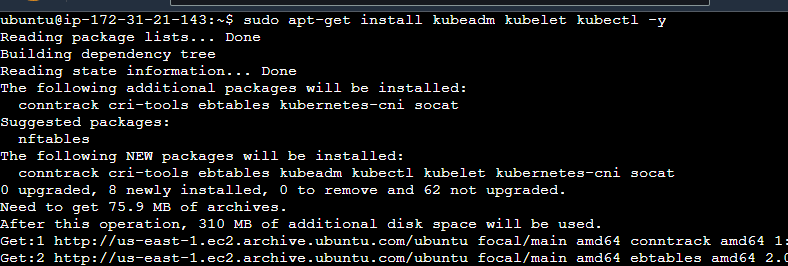
1. Install Kubernetes tools with the command:

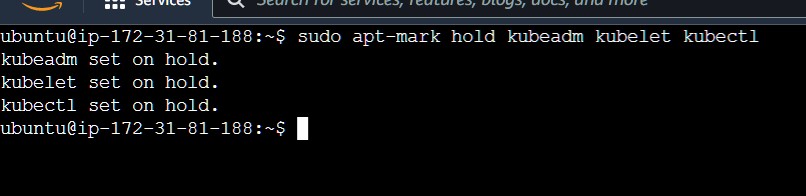
**$sudo apt-get install kubeadm kubelet kubectl -y**

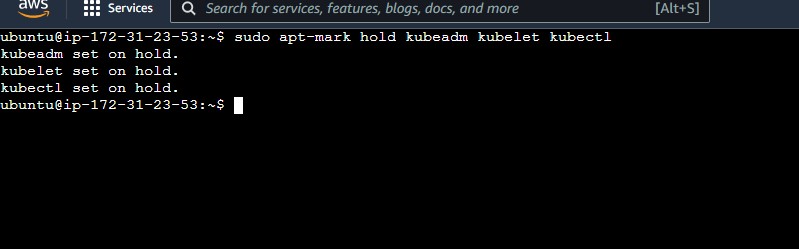
Master

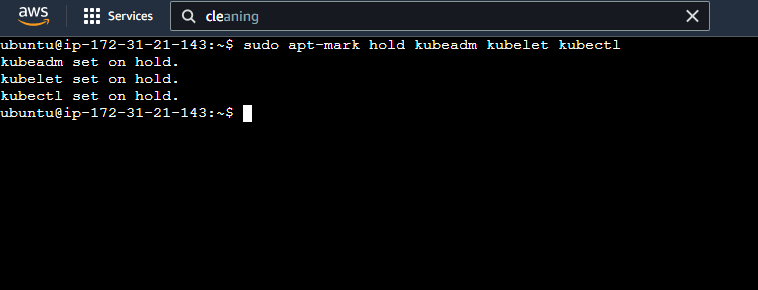
Worker1



Worker2

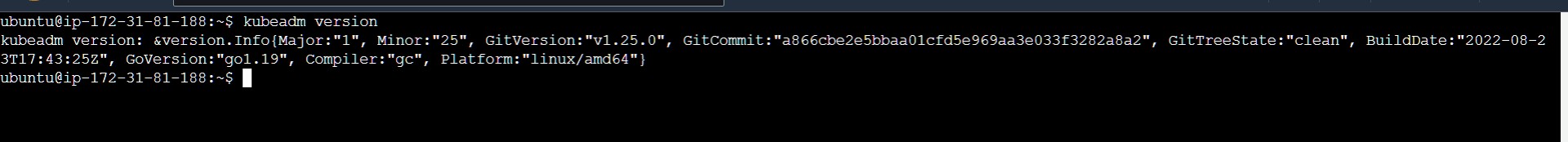
Master

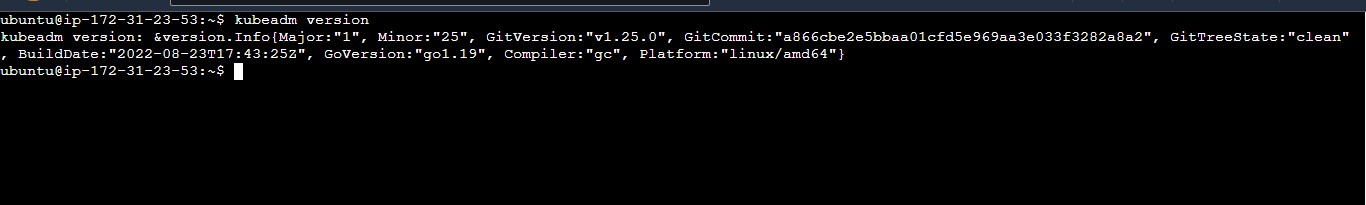
Worker1

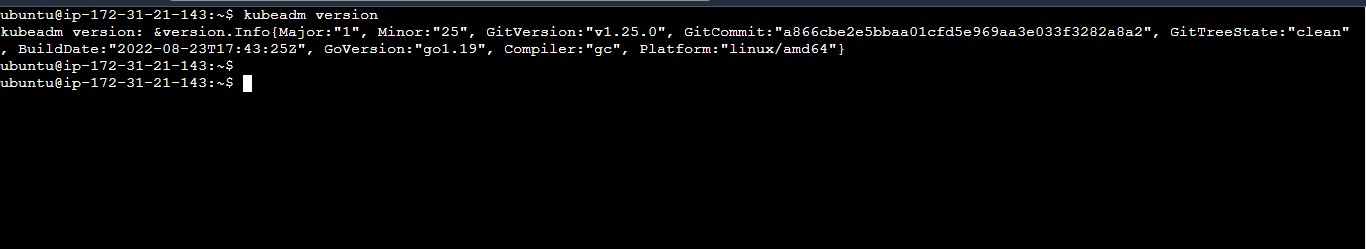
Worker2

1. Verify the installation with:

**$kubeadm version**



Worker1

Worker2

# Kubernetes Deployment

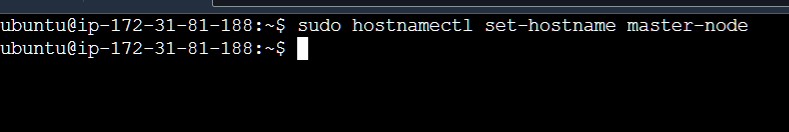
**Step 6:** Begin Kubernetes Deployment

Start by disabling the swap memory on each machine:

**$sudo swapoff --a**

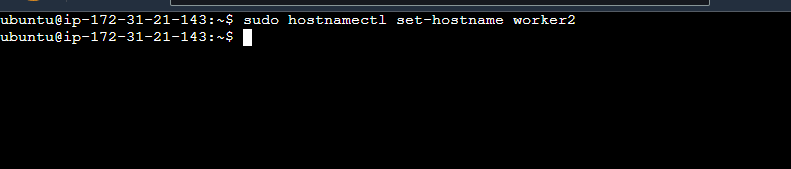
**Step 7:** Assign Unique Hostname for Each Server Node

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

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

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

**$sudo hostnamectl set-hostname worker1**

$sudo hostnamectl set-hostname worker2

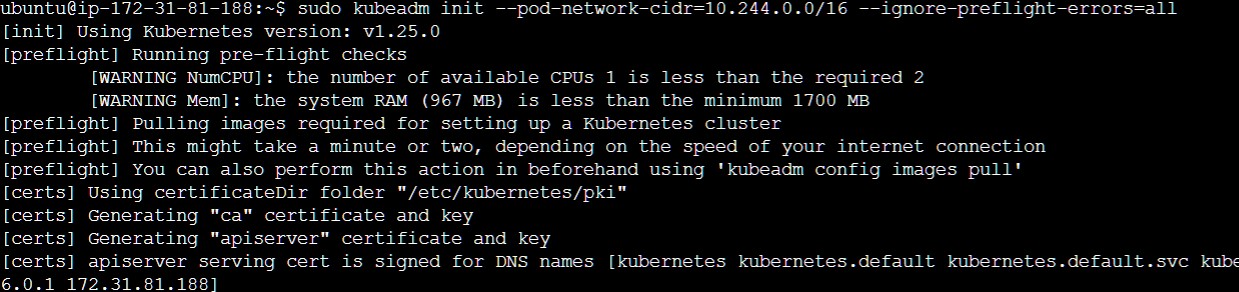
# Note - Perform the next two steps i.e. Step 8 and Step 9 only on the Master machine.

**Step 8:** Initialize Kubernetes only on Master Node

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

**$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.

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

**$ mkdir -p $HOME/.kube**

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

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

**Step 9:** Deploy Pod Network to Cluster

A Pod Network is a way to allow communication between different nodes in the cluster. This tutorial uses the flannel virtual network.

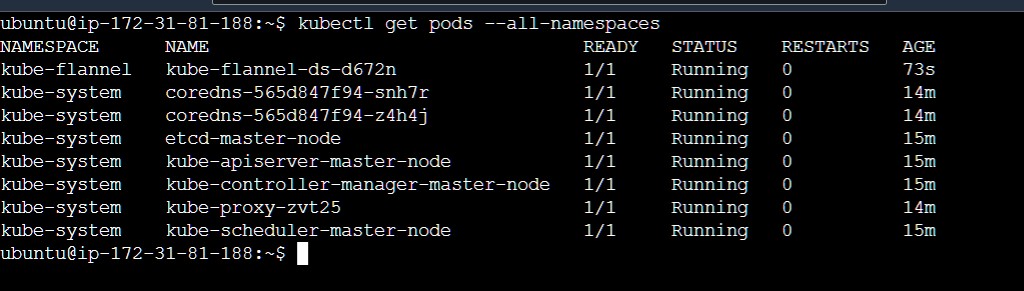
Enter the following:

$ **kubectl apply -f** [**https://raw.githubusercontent.com/coreos/flannel/master/Documentation/ku**](https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml)[**be-flannel.yml**](https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml)

Allow the process to complete.

Verify that everything is running and communicating:

**$ kubectl get pods --all-namespaces**



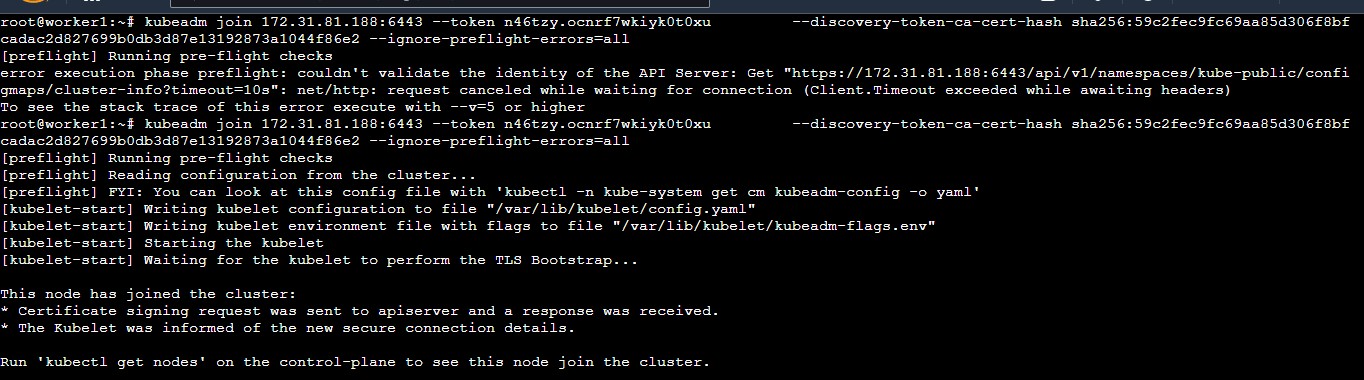
**Do this step only on the worker nodes. Step 10:** Join Worker Node to Cluster

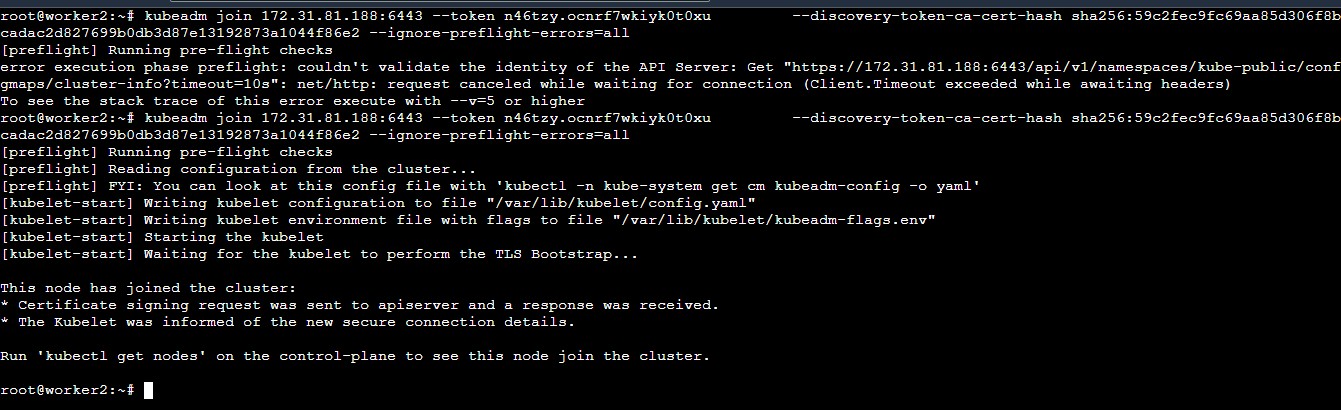
As indicated in Step 8, enter the kubeadm join command on each worker node to connect to the cluster. Switch to the **root user** of your worker system and enter the command you noted from Step 8.

**$ kubeadm join 172.31.81.188:6443 --token n46tzy.ocnrf7wkiyk0t0xu**

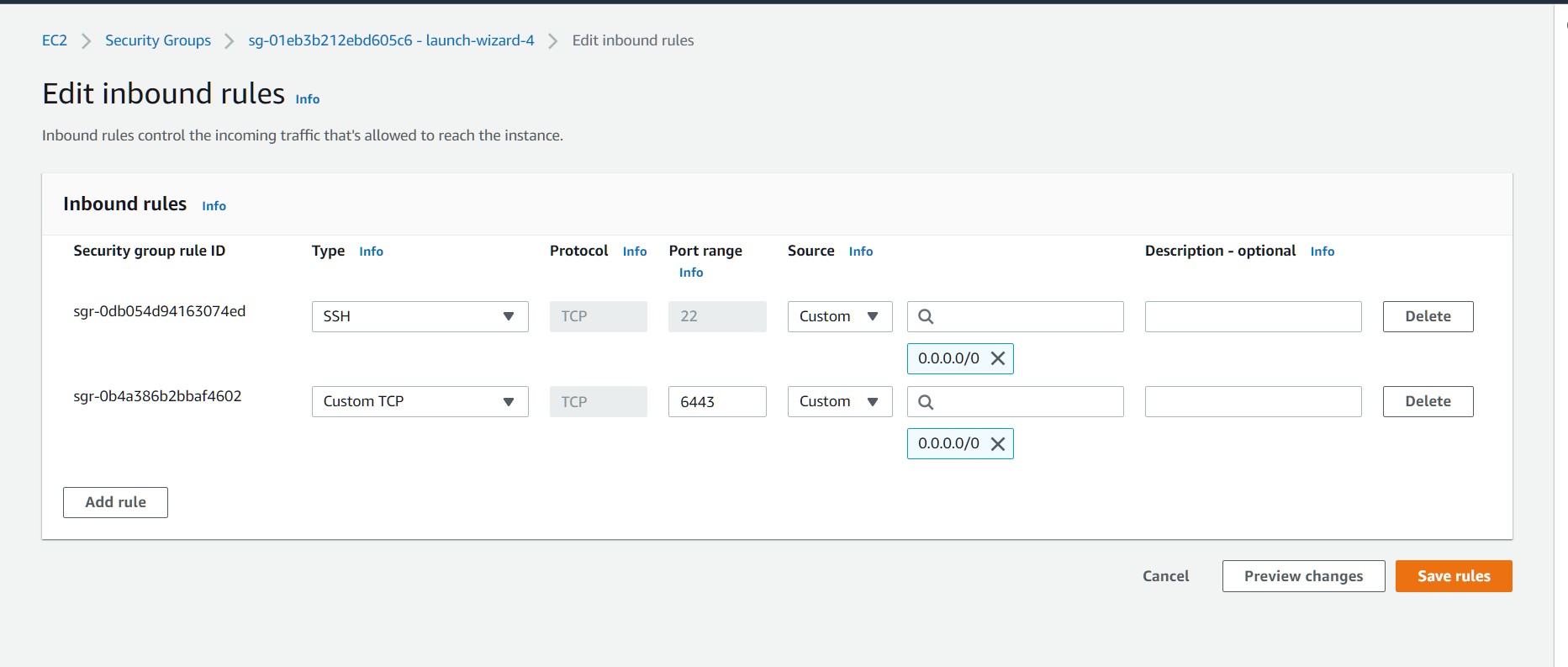
**--discovery-token-ca-cert-hash sha256:59c2fec9fc69aa85d306f8bfcadac2d827699b0db3d87e13192873a1 044f86e2 --ignore-preflight-errors=all**

Note - Join command is different for everyone please do not use this. Worker1

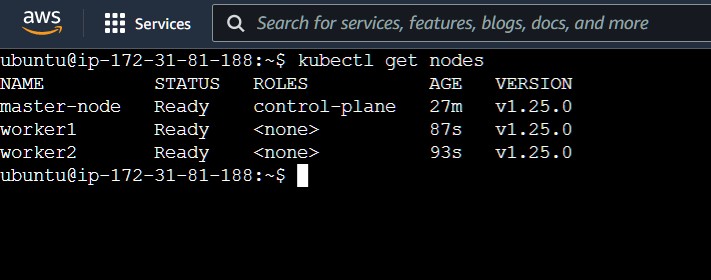


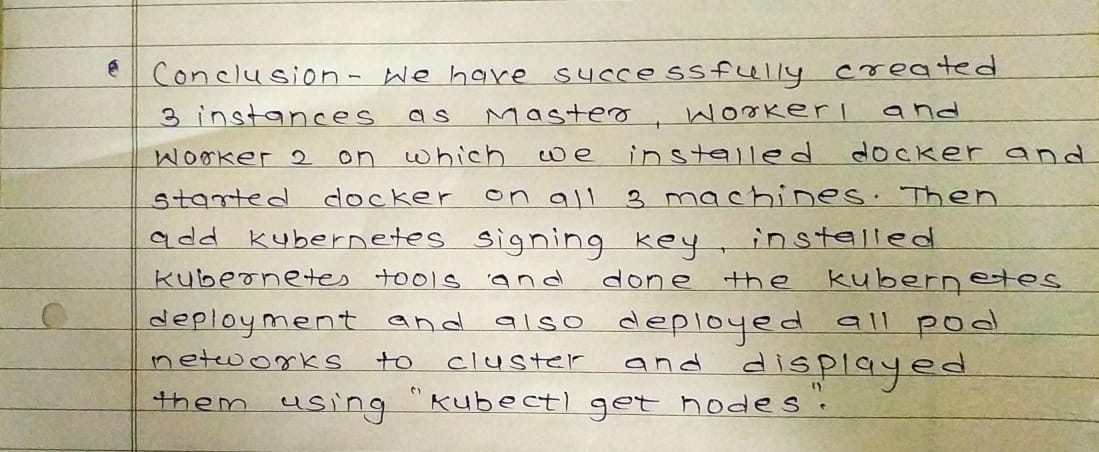
Worker 2

If you are getting an error of port “6443” while joining to master then edit the inbound rules of your security group and add port 6443.



Switch to the master server, and enter:

$ **kubectl get nodes**

****