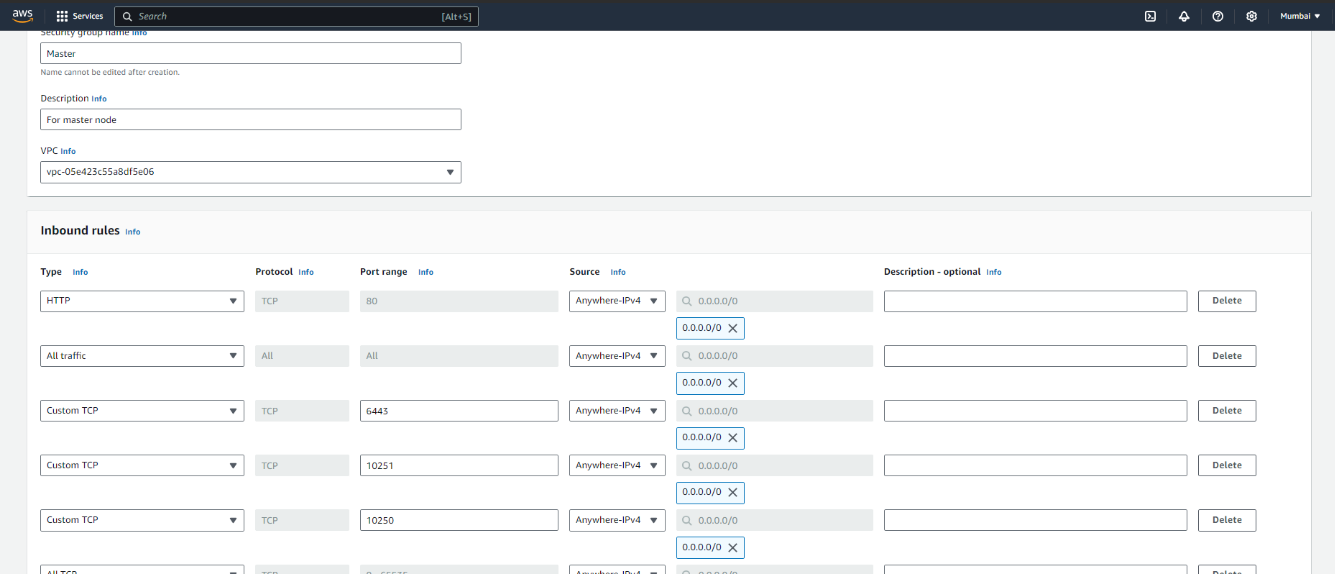
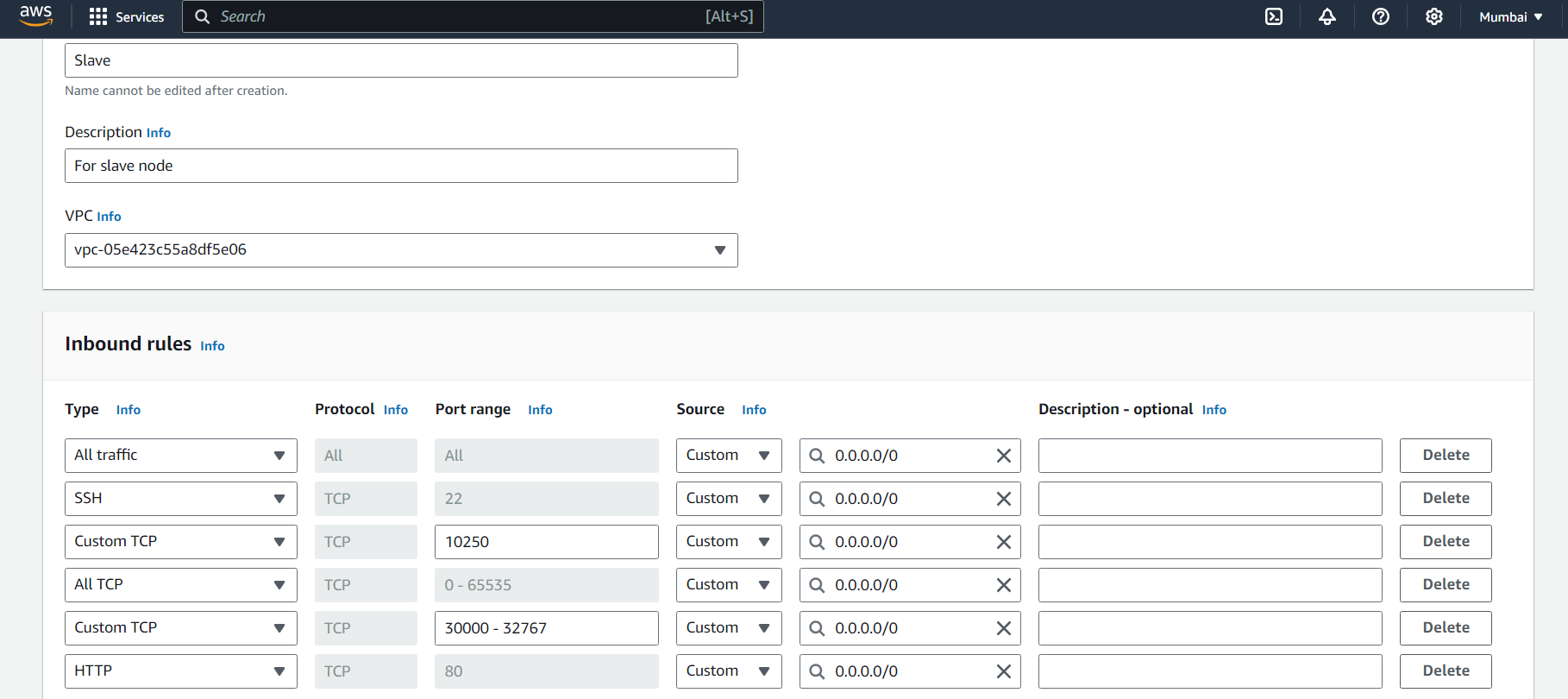
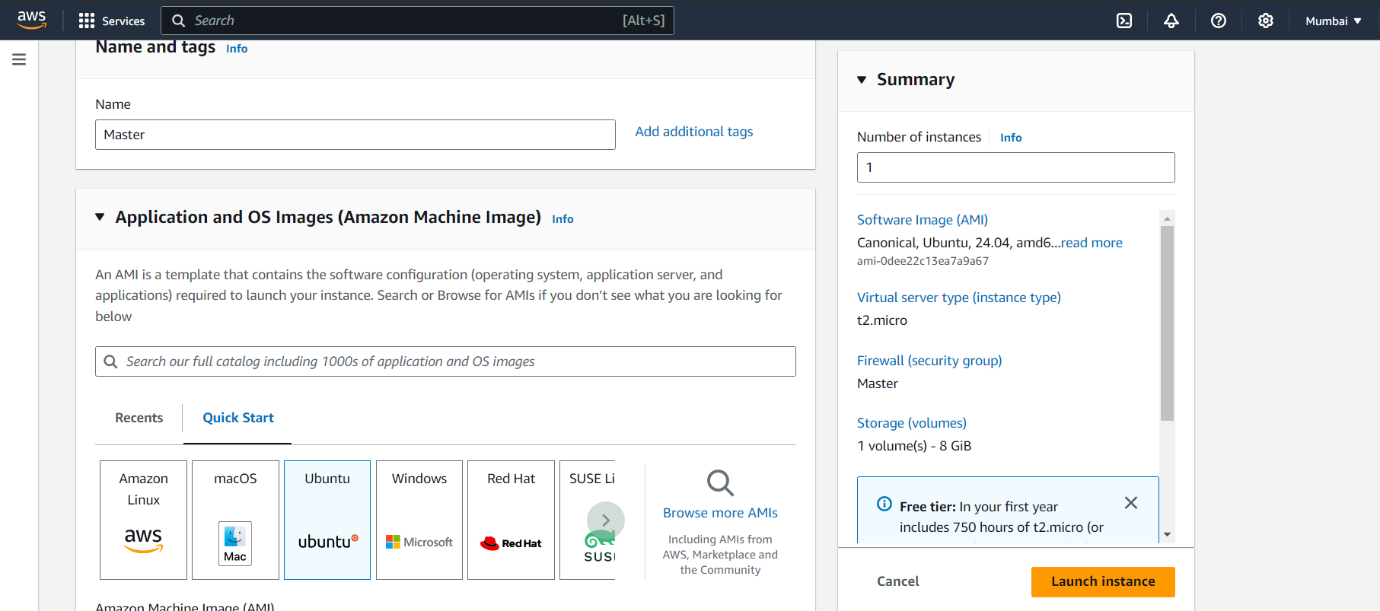
**Step 1:** Log in to your AWS Academy/personal account and launch 3 new Ec2 Instances.

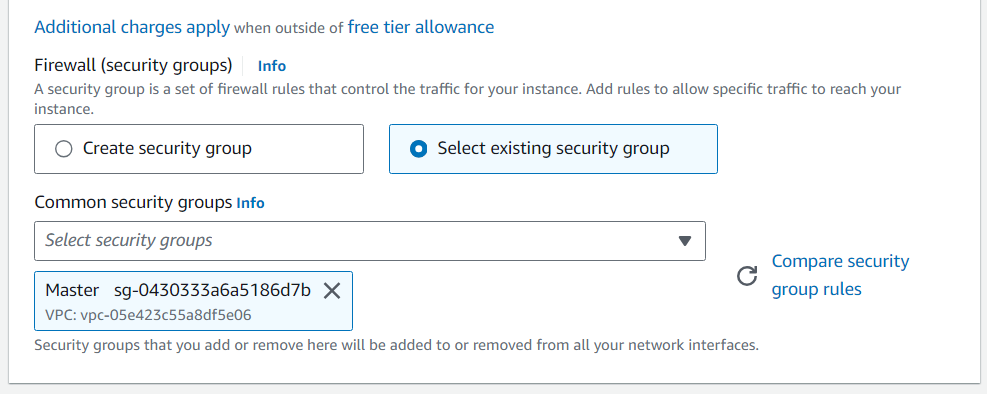
Select Ubuntu as AMI and t2.medium as Instance Type and create a key of type RSA with .pem extension and move the downloaded key to the new folder.We can use 3 Different keys or 1 common key also.





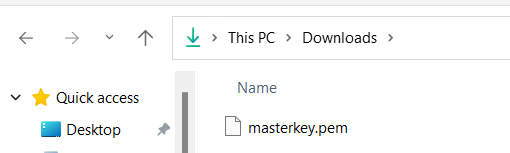
**Master:**





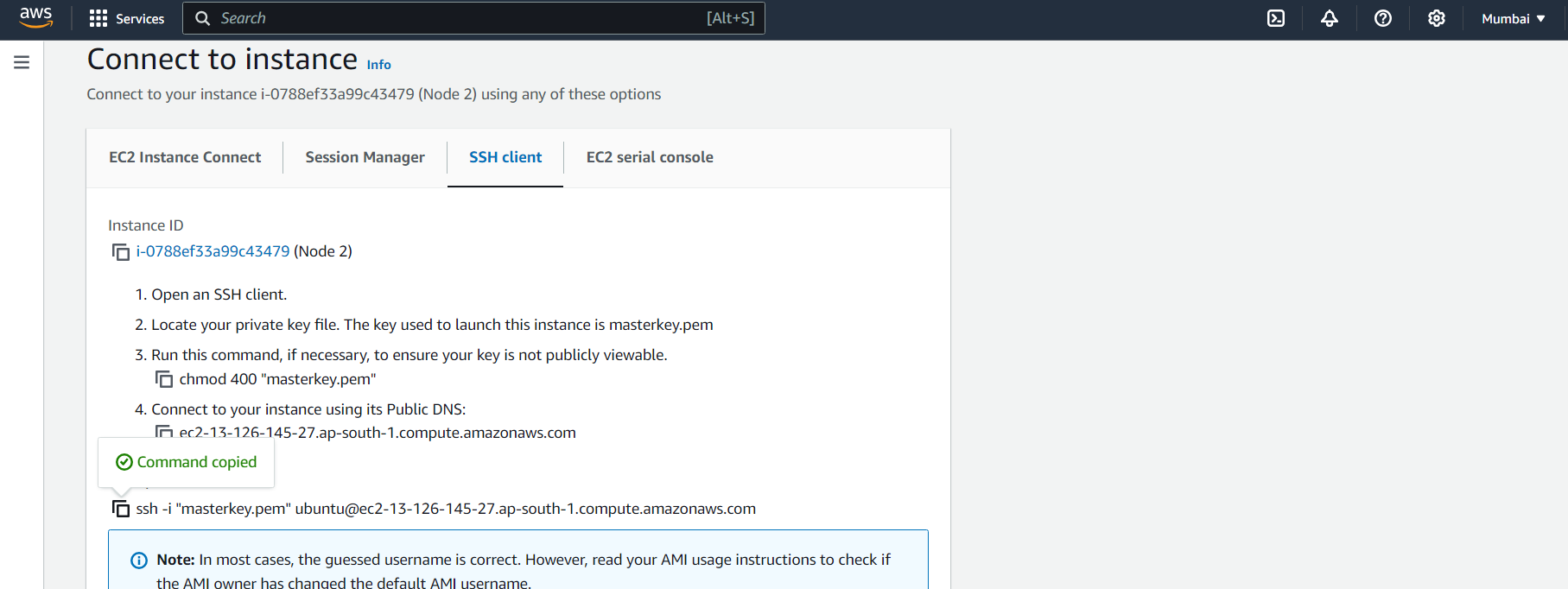
**Step 2:** After creating the instances click on Connect & connect all 3 instances and navigate to SSH Client.

**(Downloded Key)**

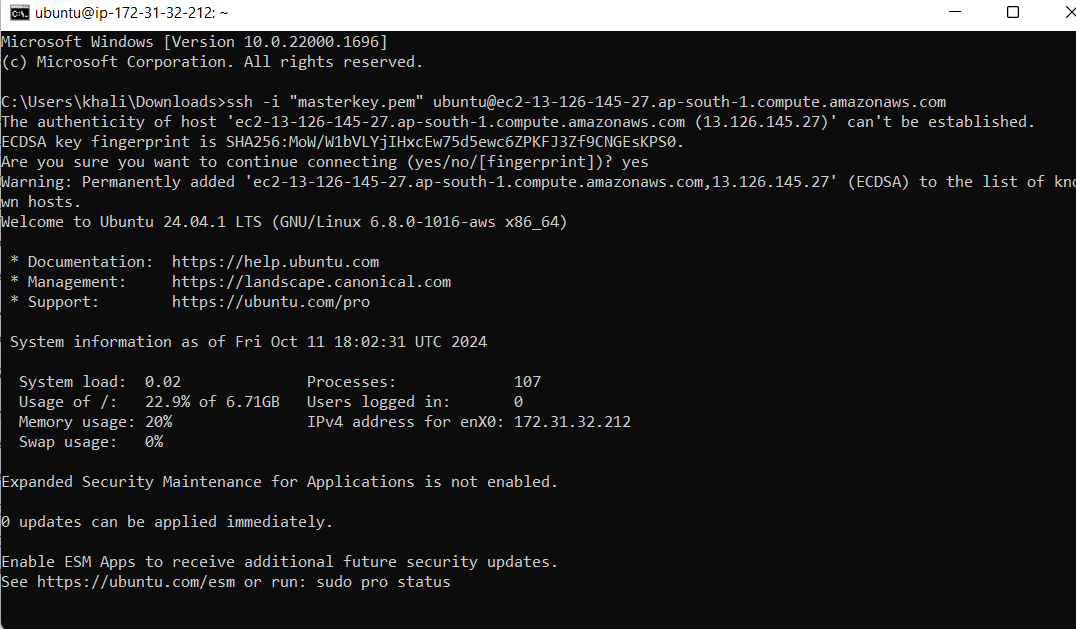
****

**Step 3:** Now open the folder in the terminal 3 times for Master, Node1& Node 2 where our .pem key is stored and paste the Example command (starting with ssh -i …..) in the terminal

Master:



Successful Connection:



**Step 4:** Run on Master,Node 1,and Node 2 the below commands to install and setup Docker in Master, Node1, and Node2.

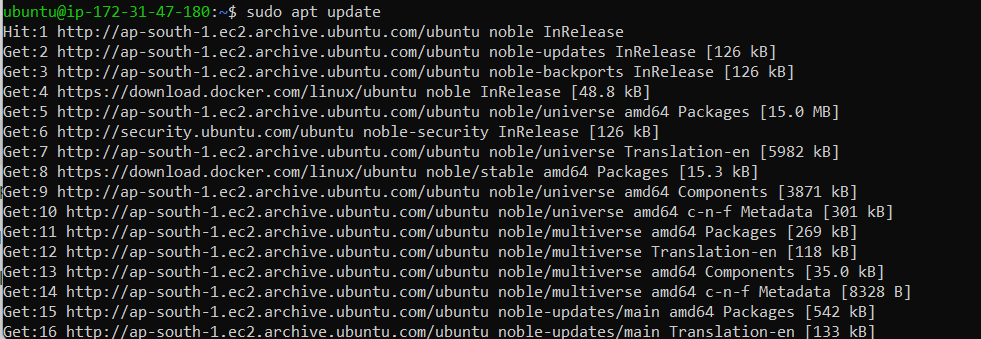
**curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add - curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee**

**/etc/apt/trusted.gpg.d/docker.gpg > /dev/null**

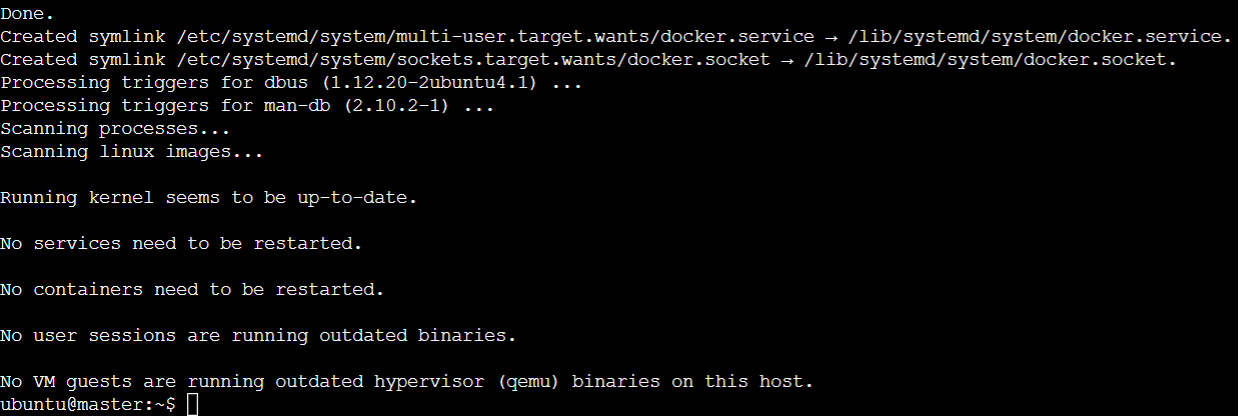
**sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu**

**$(lsb\_release -cs) stable"**

**sudo apt-get update**



**sudo apt-get install -y docker-c**



**sudo mkdir -p /etc/docker**

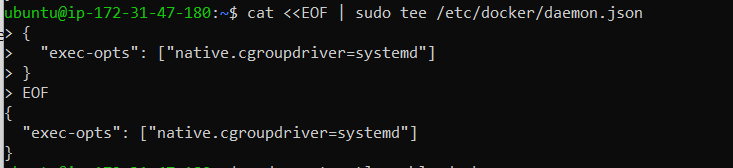
**cat <<EOF | sudo tee /etc/docker/daemon.json**

**{**

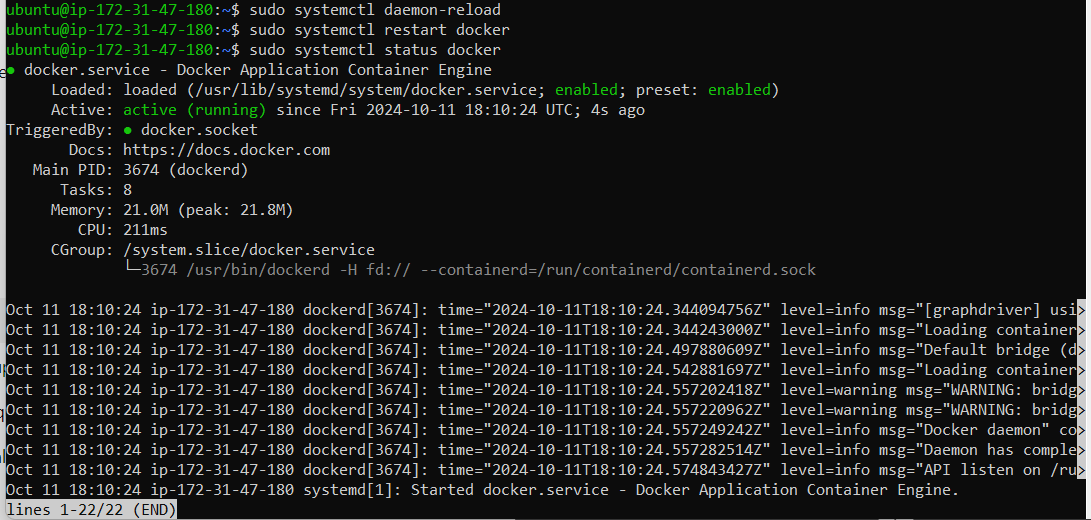
**"exec-opts": ["native.cgroupdriver=systemd"]**

**}**

**EOF**



**sudo systemctl enable docker sudo systemctl daemon-reload sudo systemctl restart docker**

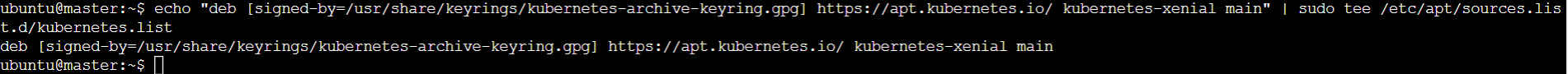


**Step 5:** Run the below command to install Kubernets.

**curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o**

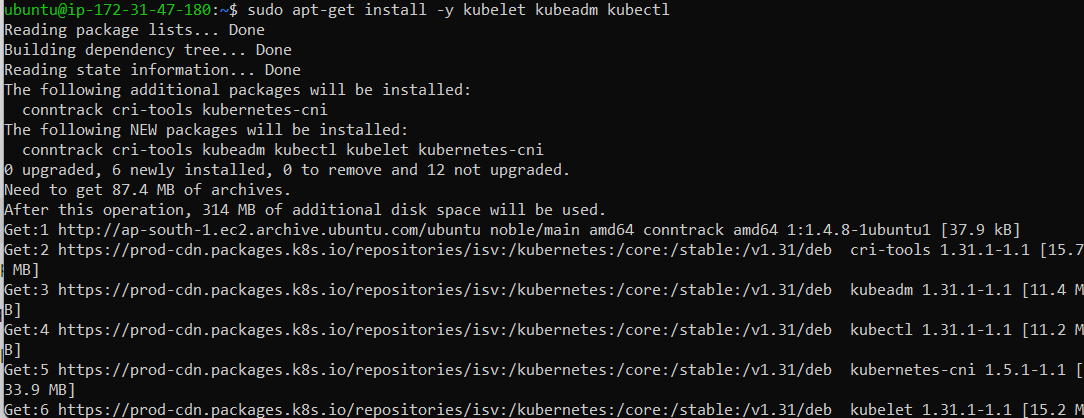
**/etc/apt/keyrings/kubernetes-apt-keyring.gpg**

**echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list**

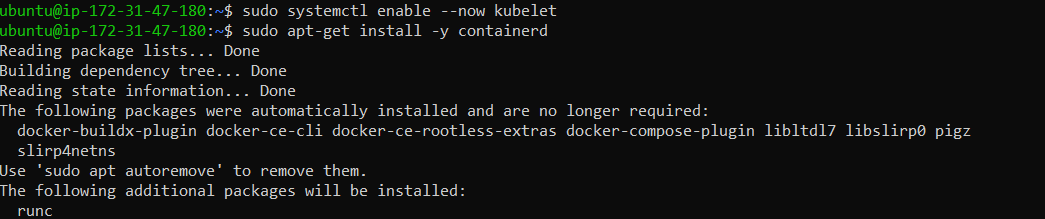
****

**sudo apt-get update**

**sudo apt-get install -y kubelet kubeadm kubectl sudoapt-mark hold kubelet kubeadm kubectl**



**sudo systemctl enable --now kubelet sudo apt-get install -y containerd**

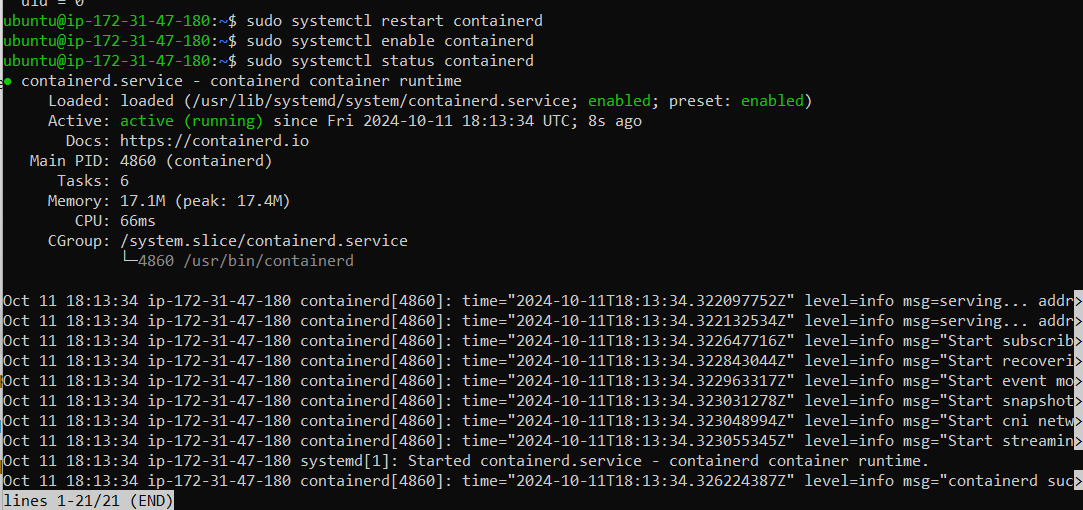


**sudo mkdir -p /etc/containerd**

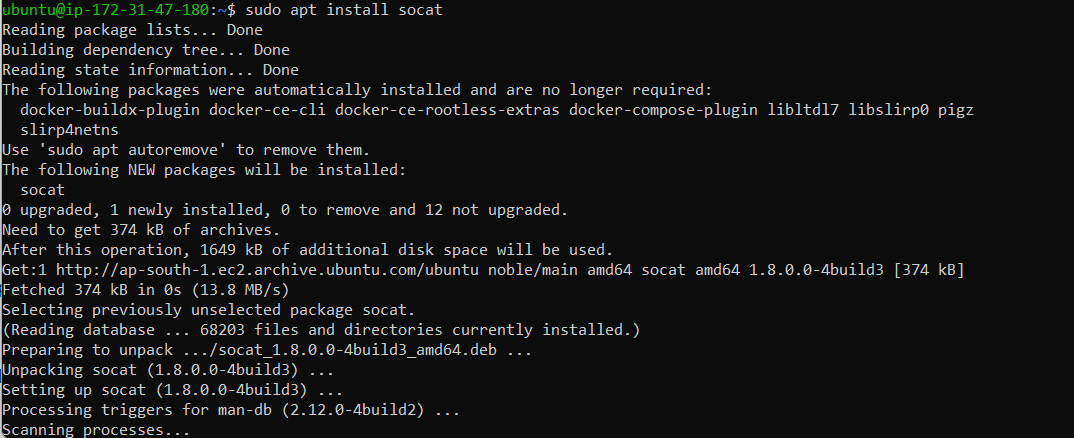
**sudo containerd config default | sudo tee /etc/containerd/config.toml**



**sudo systemctl restart containerd sudo systemctl enable containerd sudo systemctl status containerd**

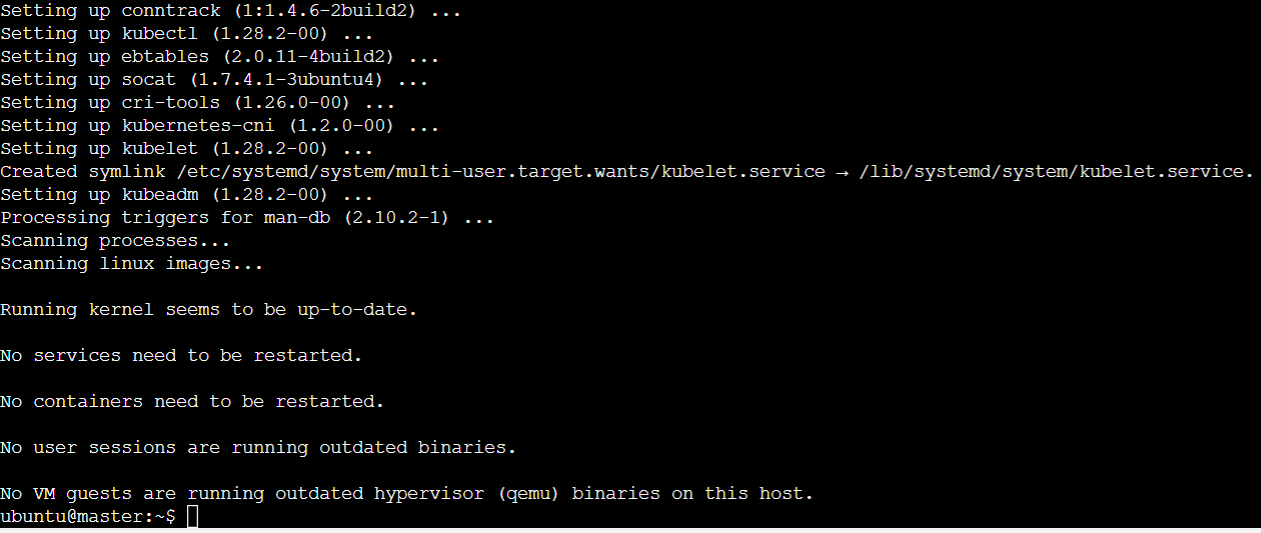


**sudo apt-get install -y socat**



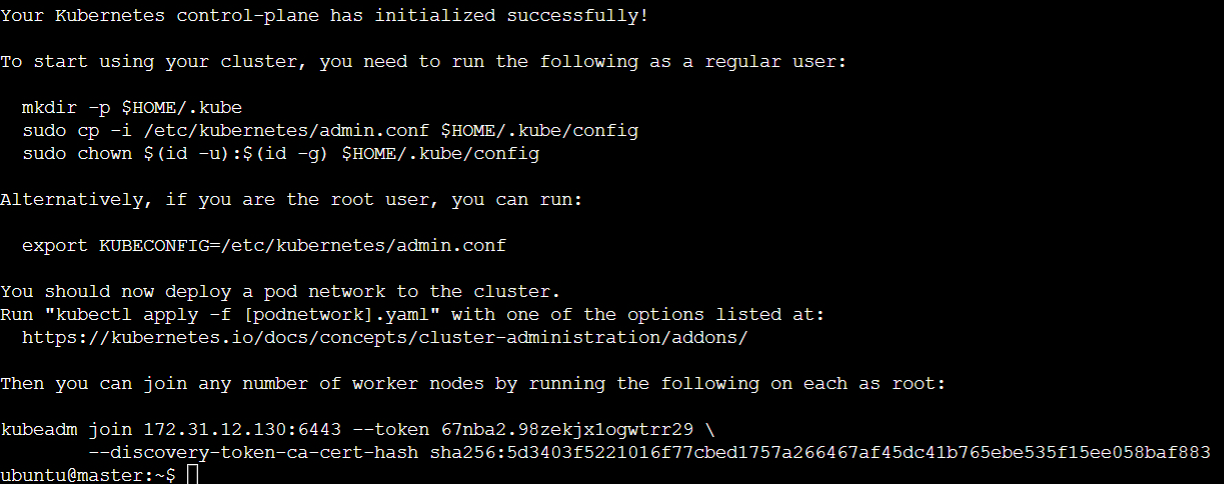
**Step 6:** Initialize the Kubecluster .Now Perform this Command only for Master.

**sudo kubeadm init --pod-network-cidr=10.244.0.0/16**

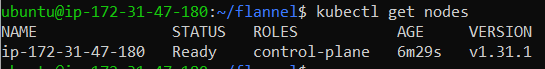
****

**Run this command on master and also copy and save the Join command from above. mkdir -p $HOME/.kube**

**sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config sudo chown $(id -u):$(id -g) $HOME/.kube/config**

****

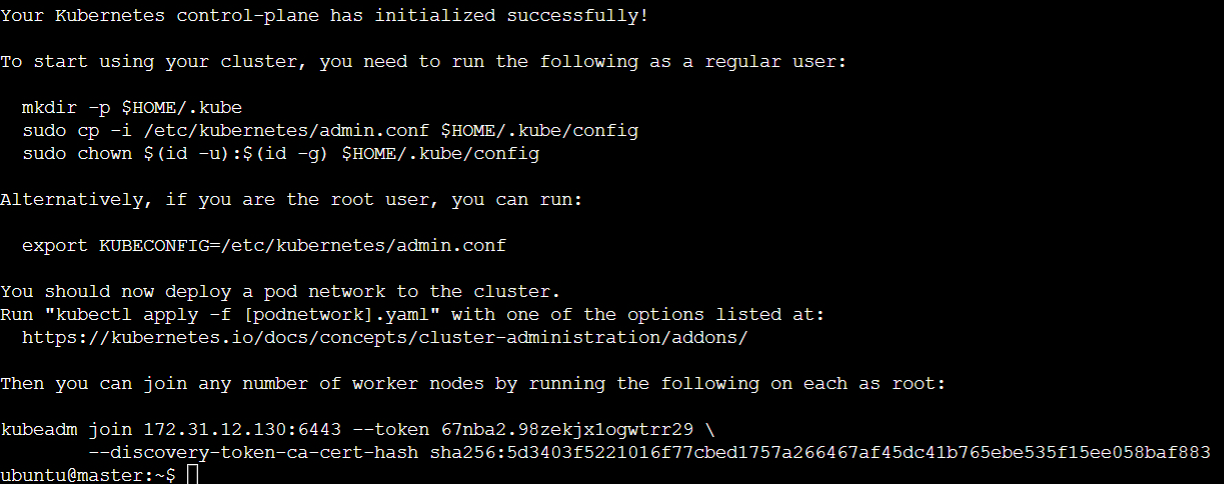
**Step 7: Now Run the command kubectl get nodes to see the nodes before executing Join command on nodes.**



**Step 8: Now Run the following command on Node 1 and Node 2 to Join to master. sudo kubeadm join 172.31.27.176:6443 --token ttay2x.n0sqeukjai8sgfg3 \**

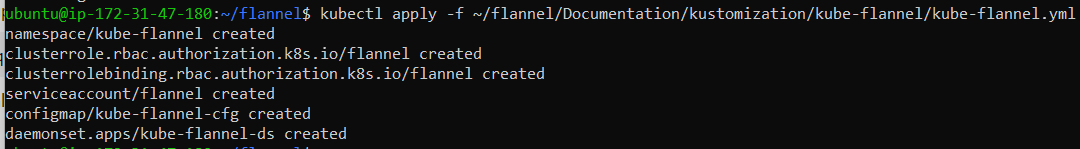
**--discovery-token-ca-cert-hash**

**sha256:d6fc5fb7e984c83e2807780047fec6c4f2acfe9da9184ecc028d77157608fbb6**

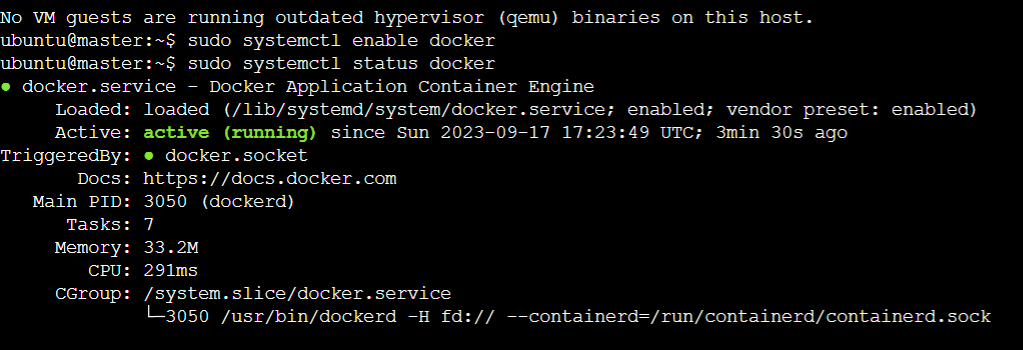
****

**Step 9: Now Run the command kubectl get nodes to see the nodes after executing Join command on nodes. Since Status is NotReady we have to add a network plugin. And also we have to give the name to the nodes.**

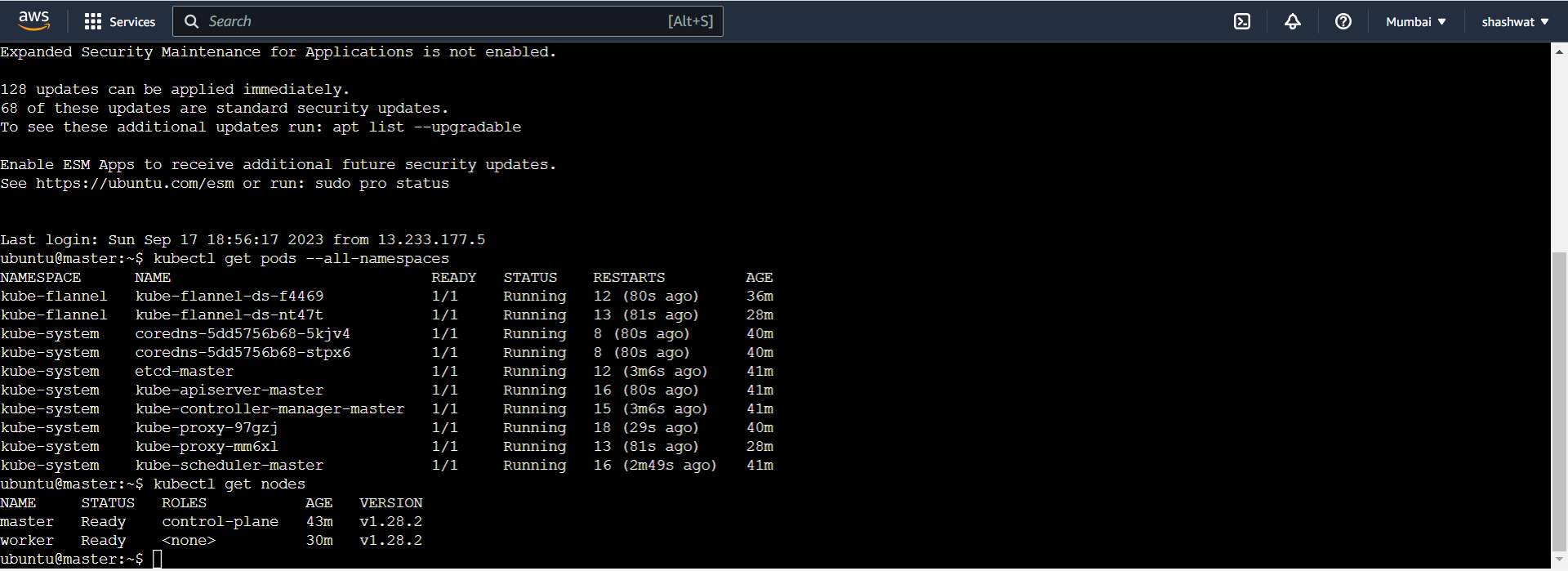
**kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml**



**sudo systemctl status kubelet**



**Step 10: Run command kubectl get nodes -o wide . And Hence we can see we have Successfully connected Node 1 and Node 2 to the Master.**

****