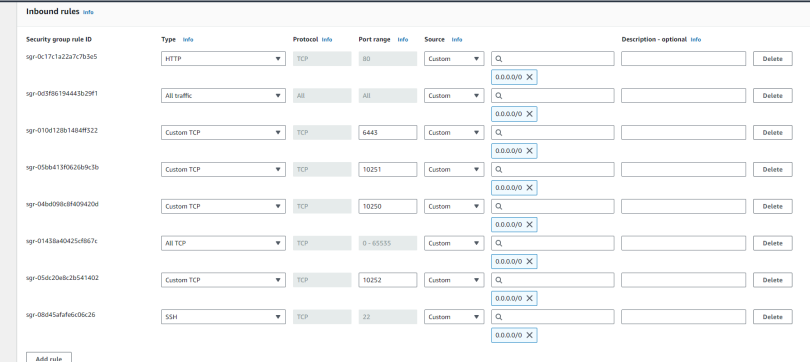
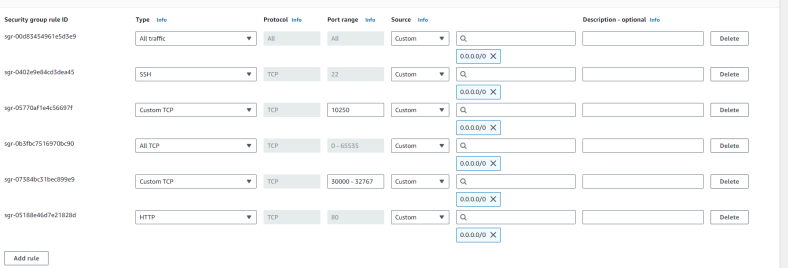
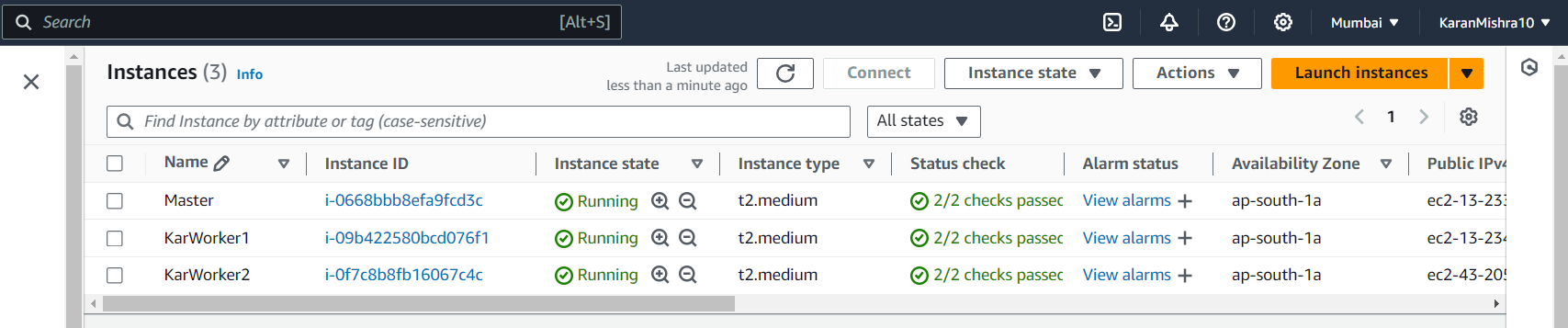
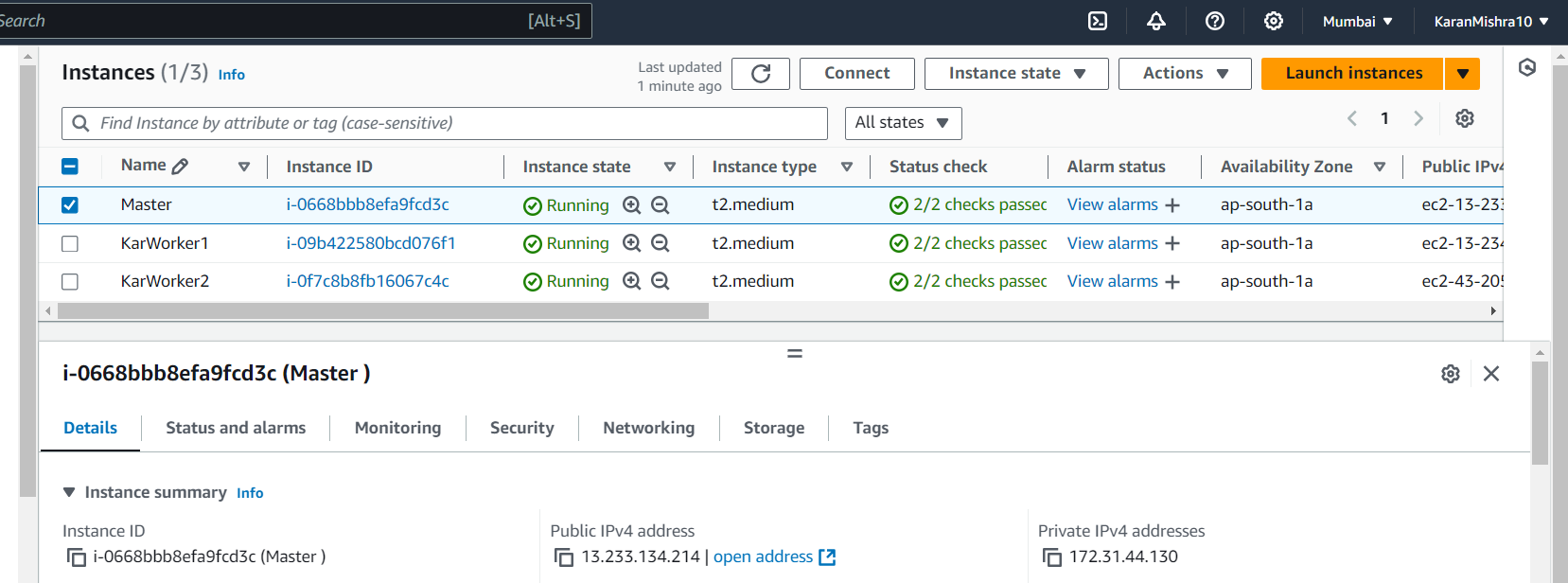
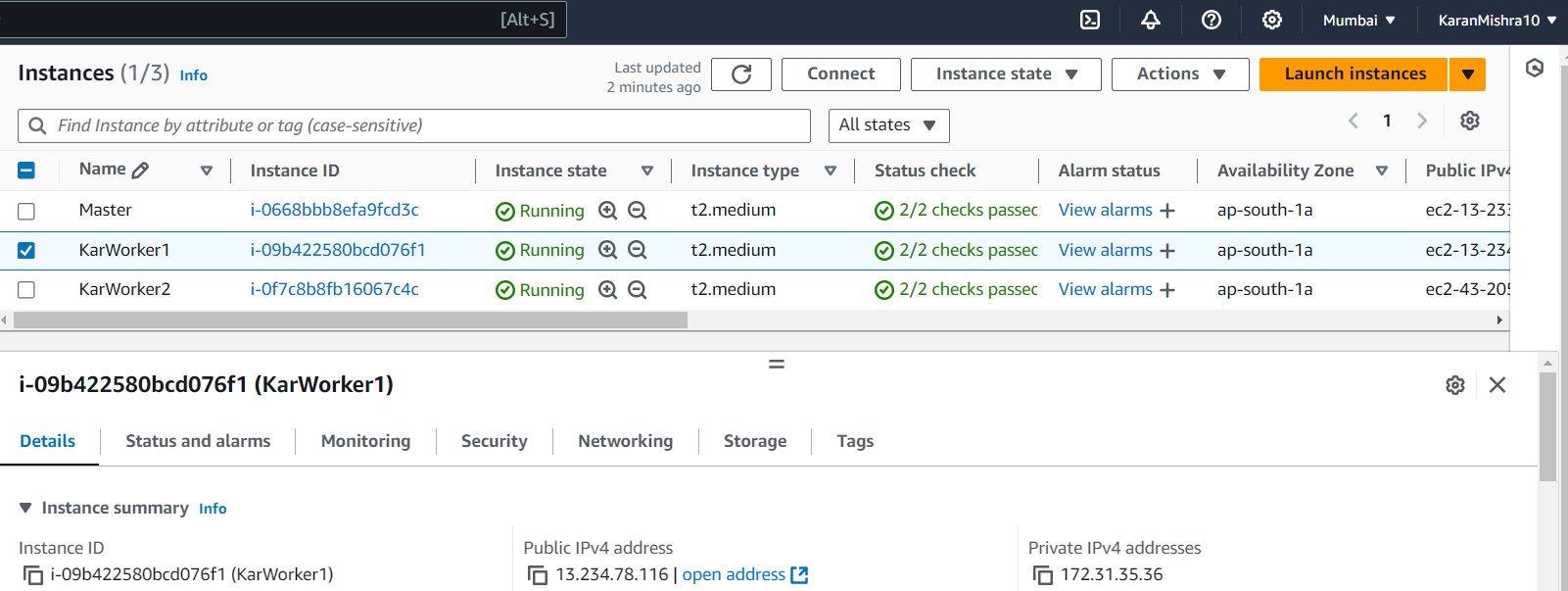
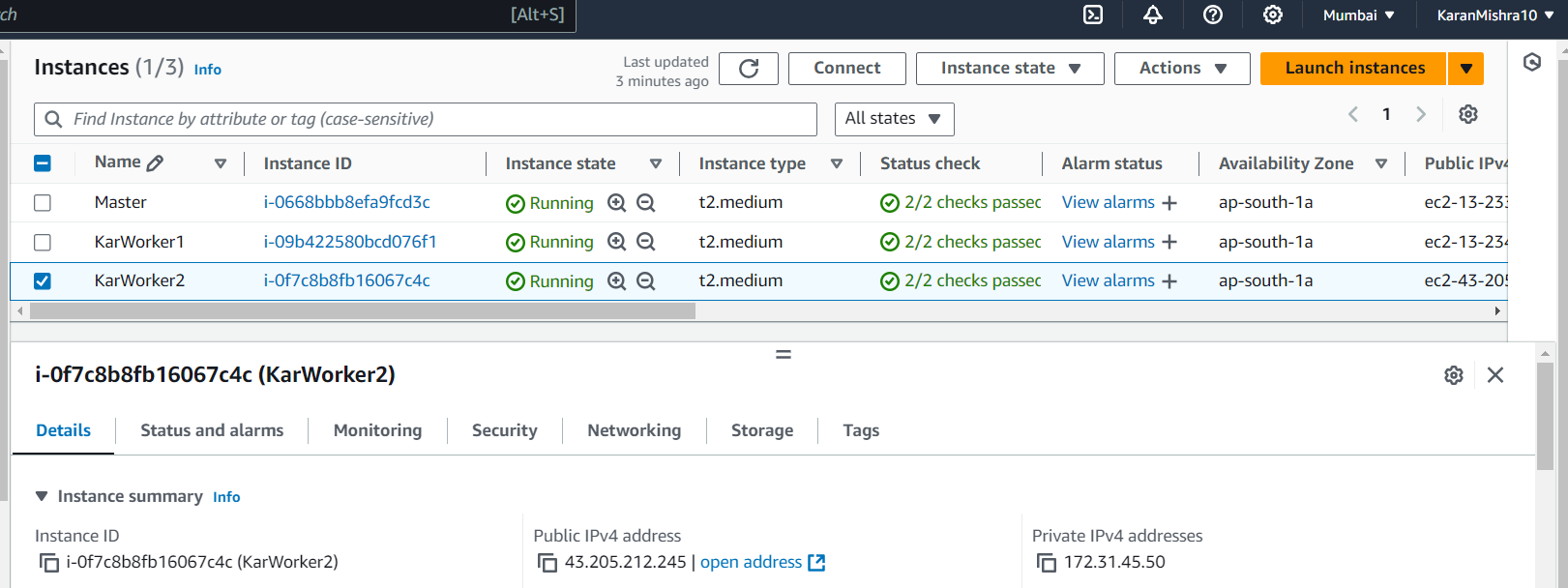
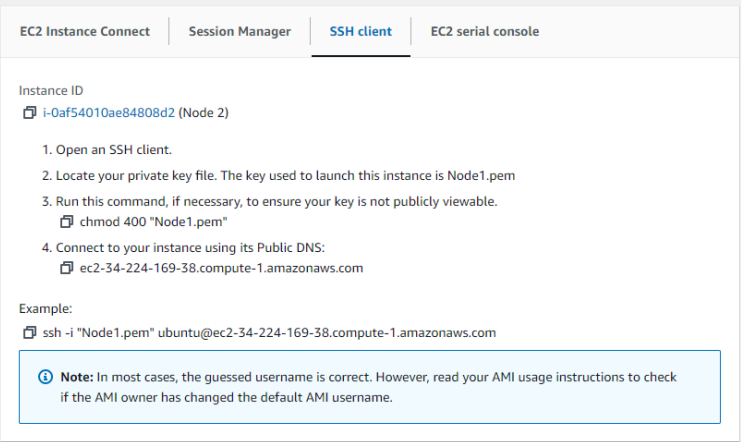
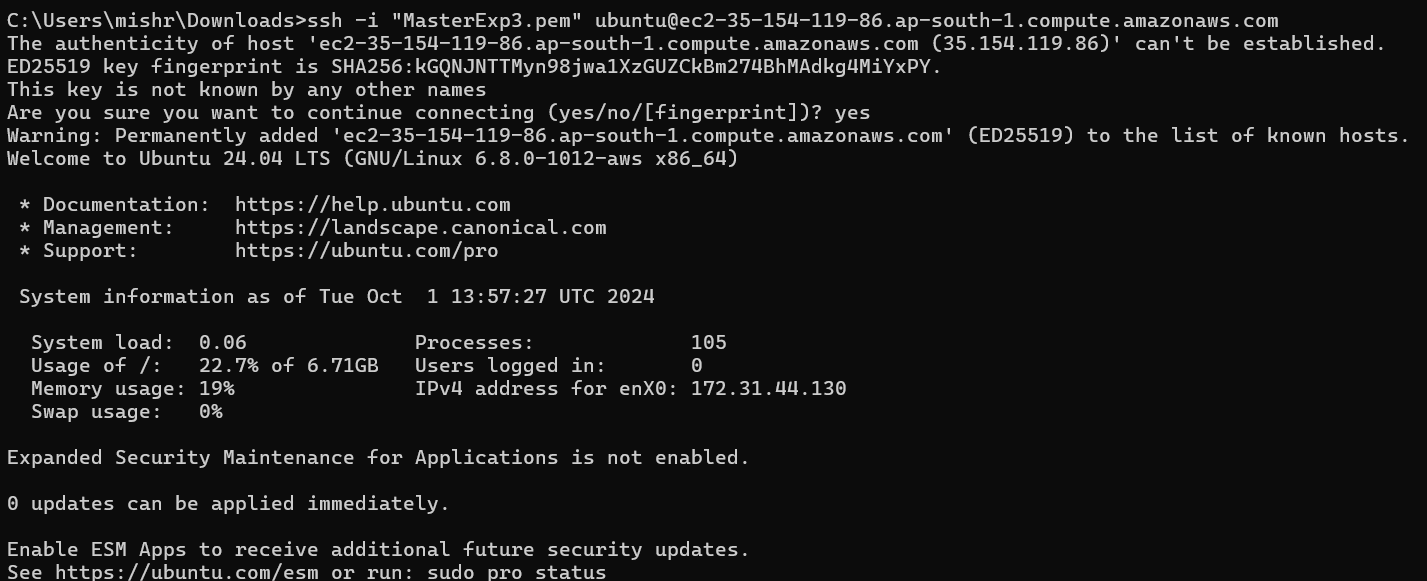
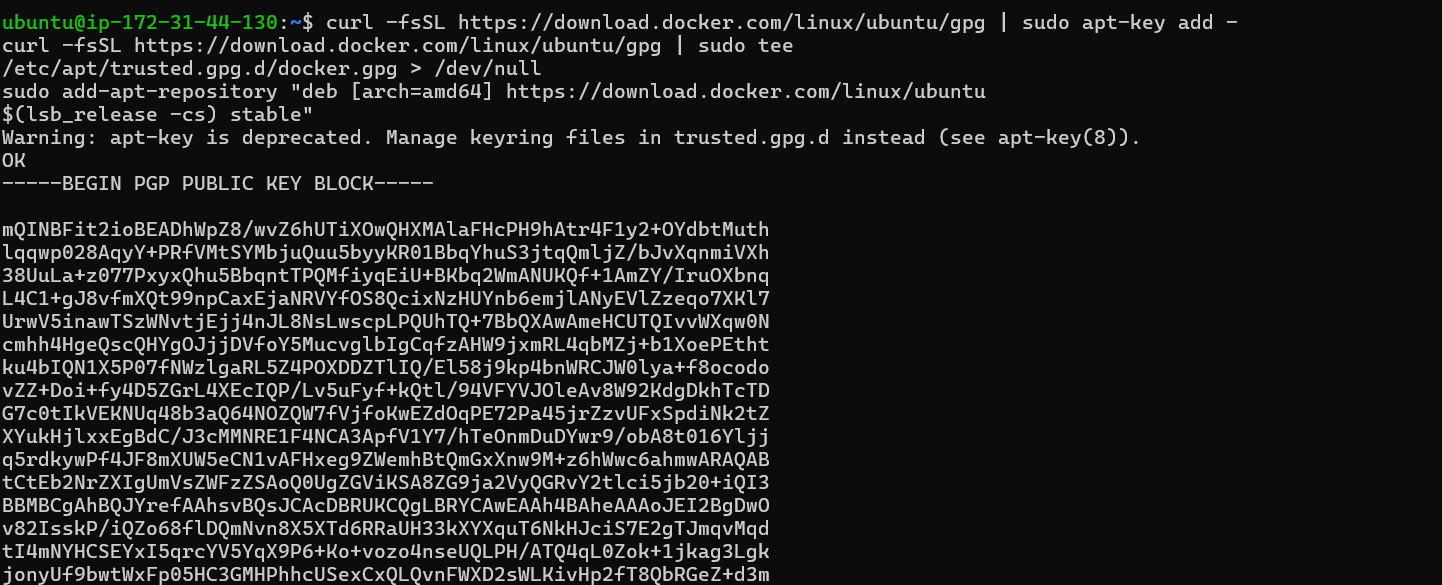
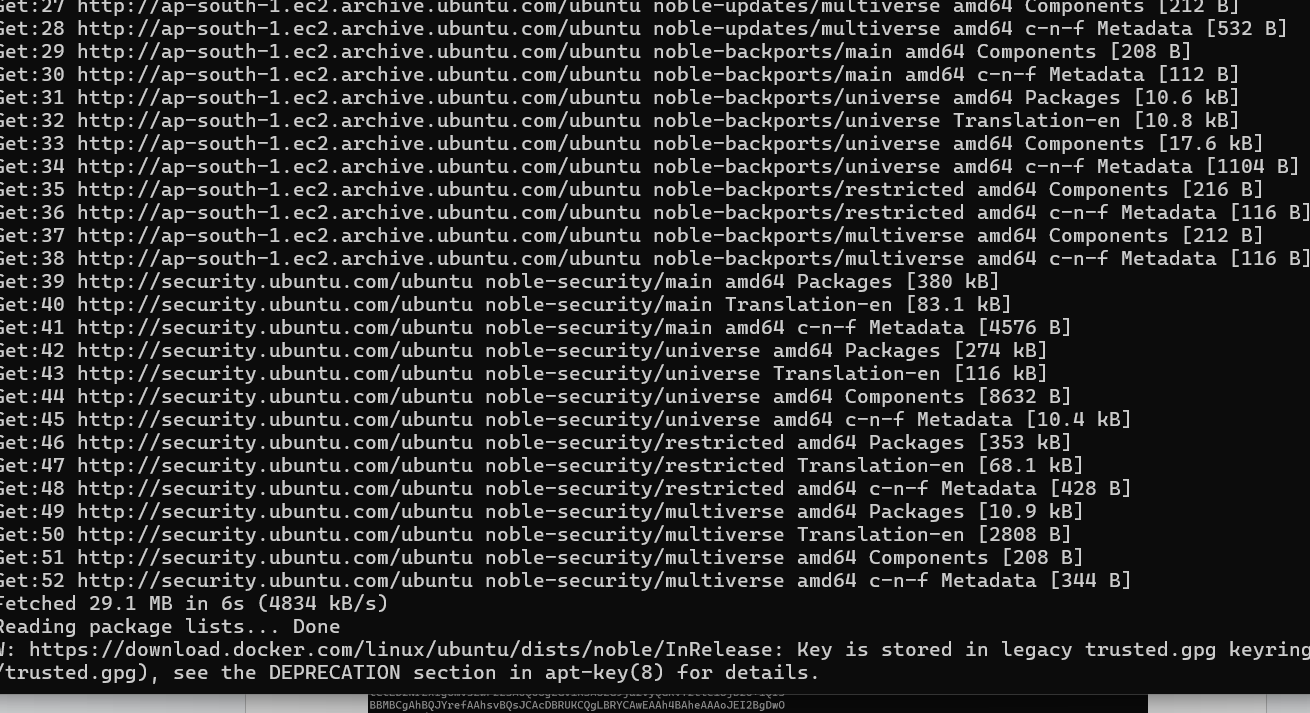
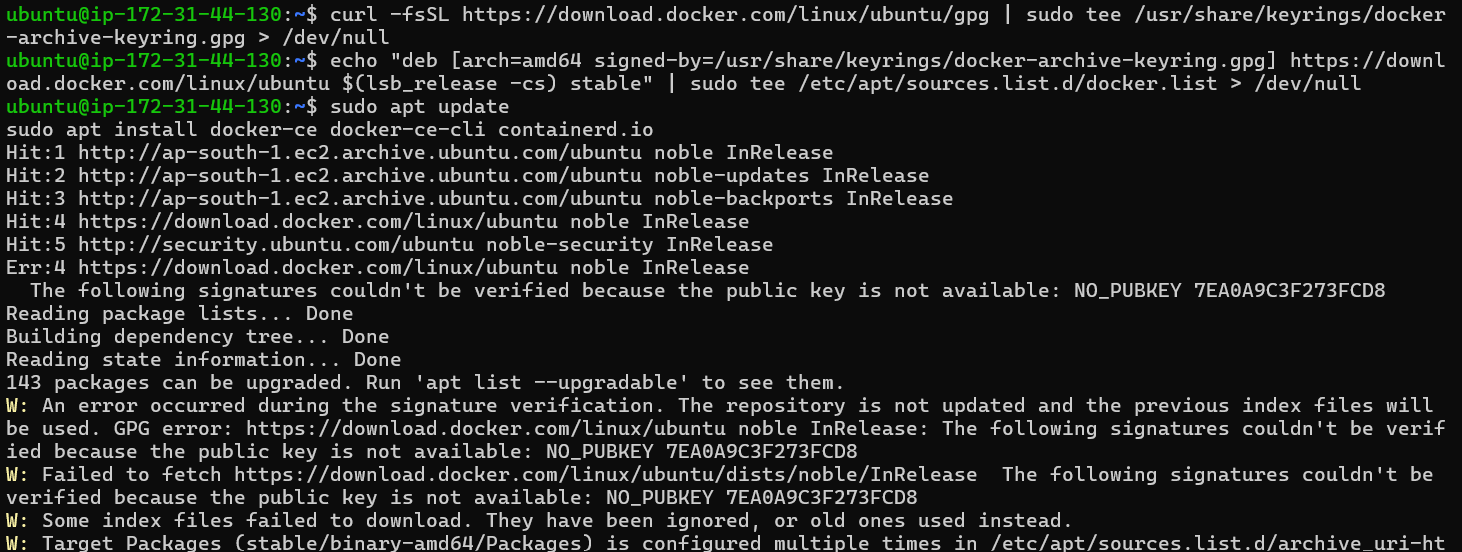
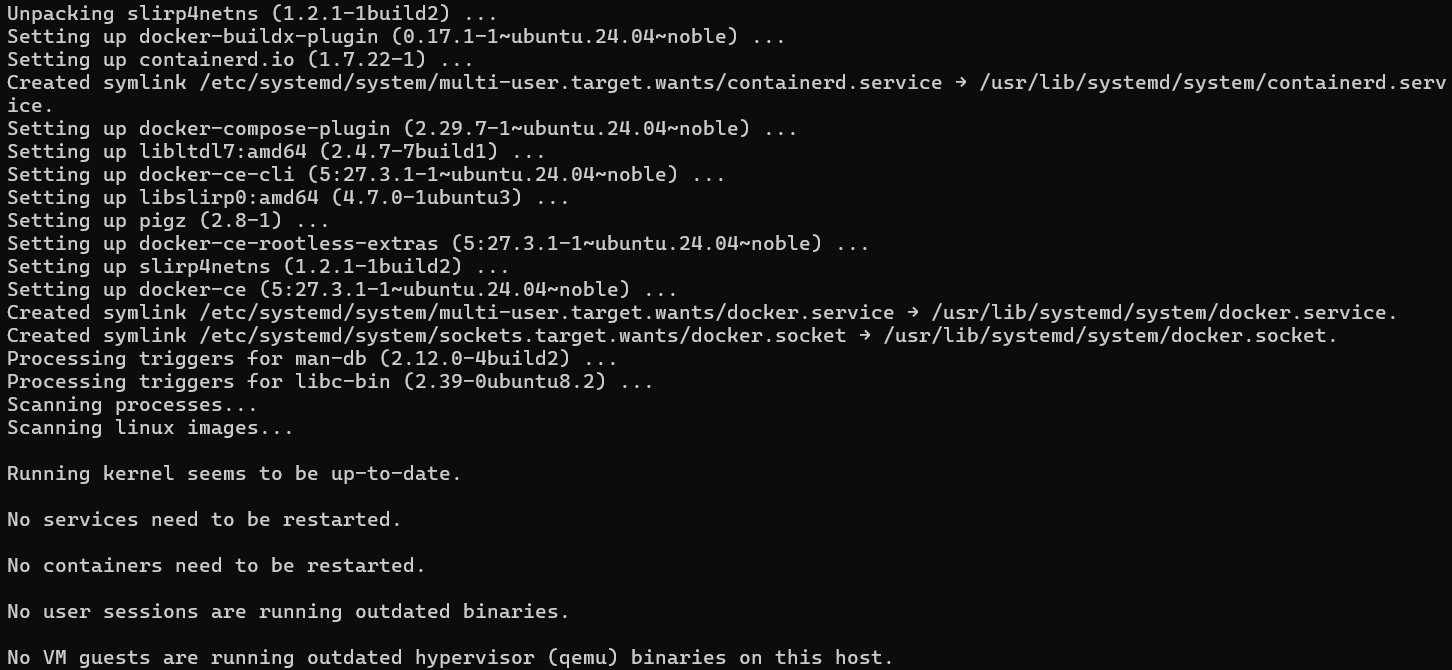
**NAME - KARAN MISHRA  
DIV - D15B  
ROLL NO - 36  
ADV.DEVOPS EXP 3  
  
Security Groups:-  
Master:-  
  
Worker:-  
  
  
Creating 3 instances(1 Master, 2 Worker Nodes)(Use the same Key for connecting each instance through SSH to your command prompt)**  
  
  
  
  
  
Connect to your EC2 instance through SSH by running the command having such format:-  
ssh -i “<Your\_saved\_key>.pem” ubuntu@<your-instance-public-ip>.<the region of created instance>.compute.amazonaws.com  
Showed when you go to the SSH Client section when you select your instance and press Connect like here:-  
  
  
  
**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-ce-cli-containerd.io**  
  
**sudo mkdir -p /etc/docker**

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

**{**

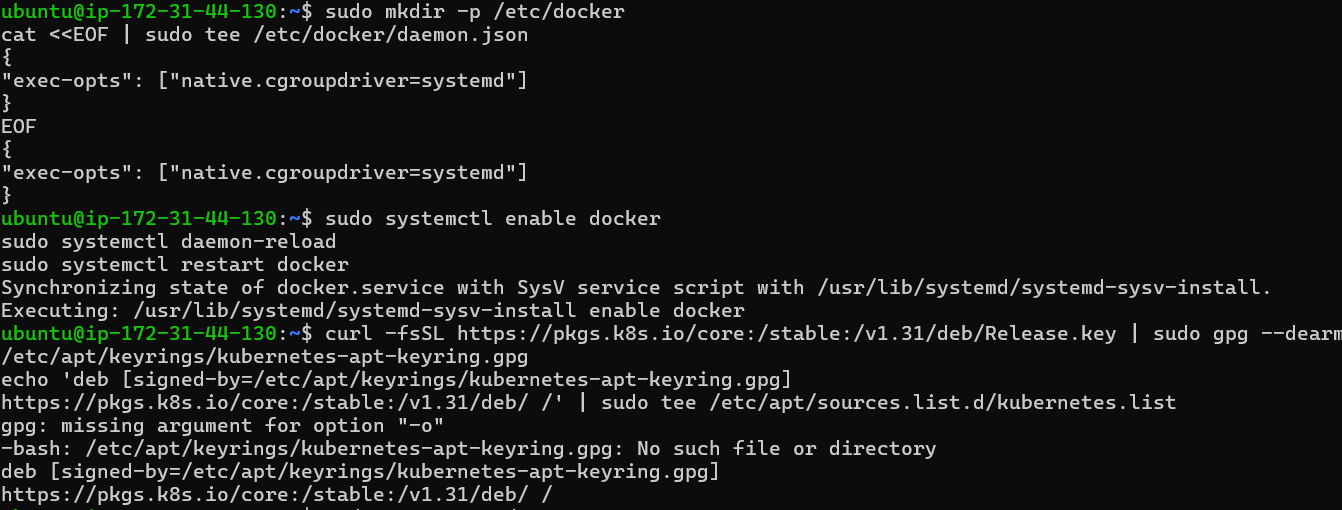
**"exec-opts": ["native.cgroupdriver=systemd"]**

**}**

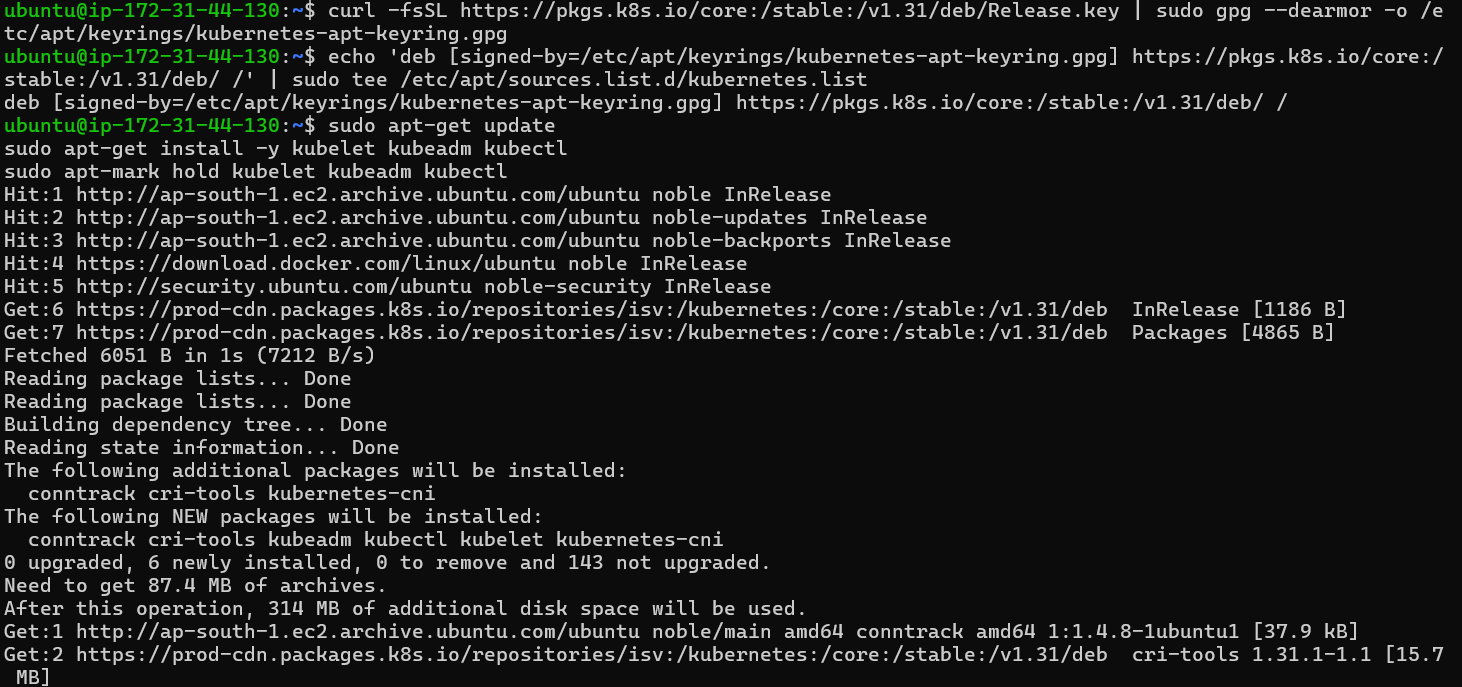
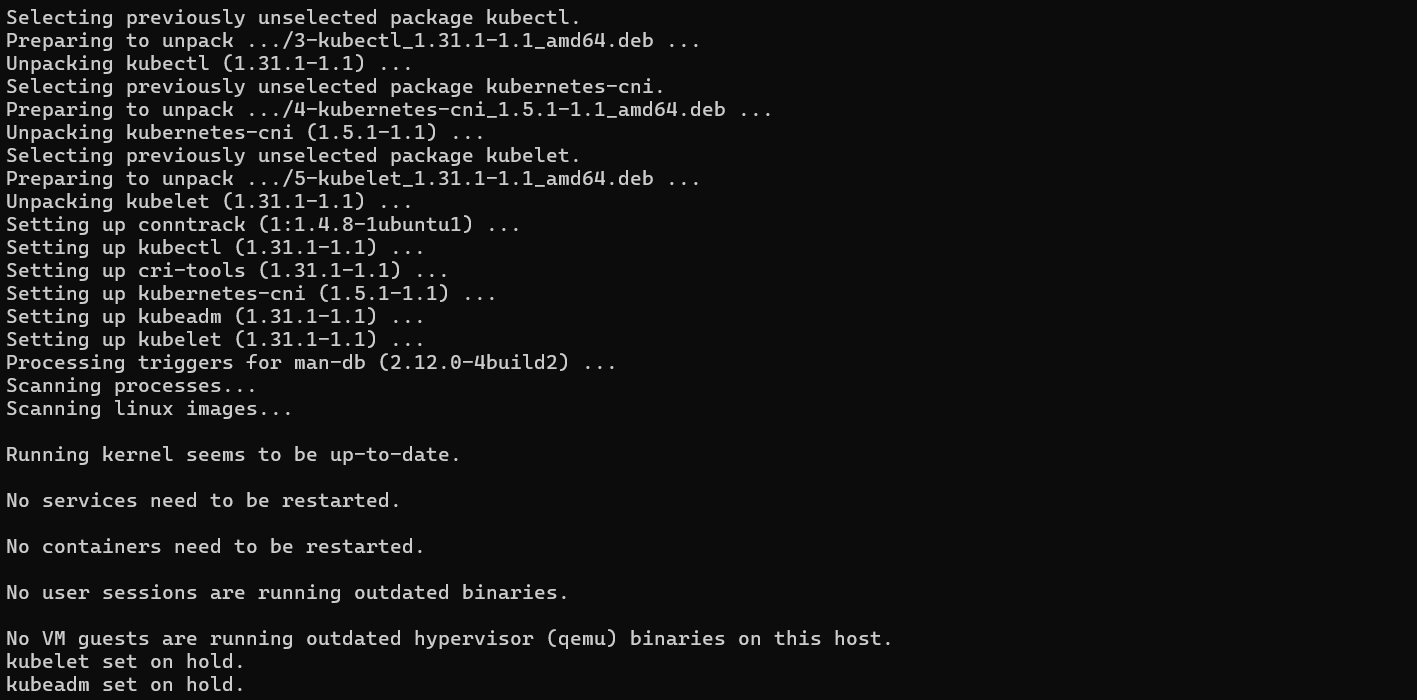
**EOF**

And  
  
**sudo systemctl enable docker**

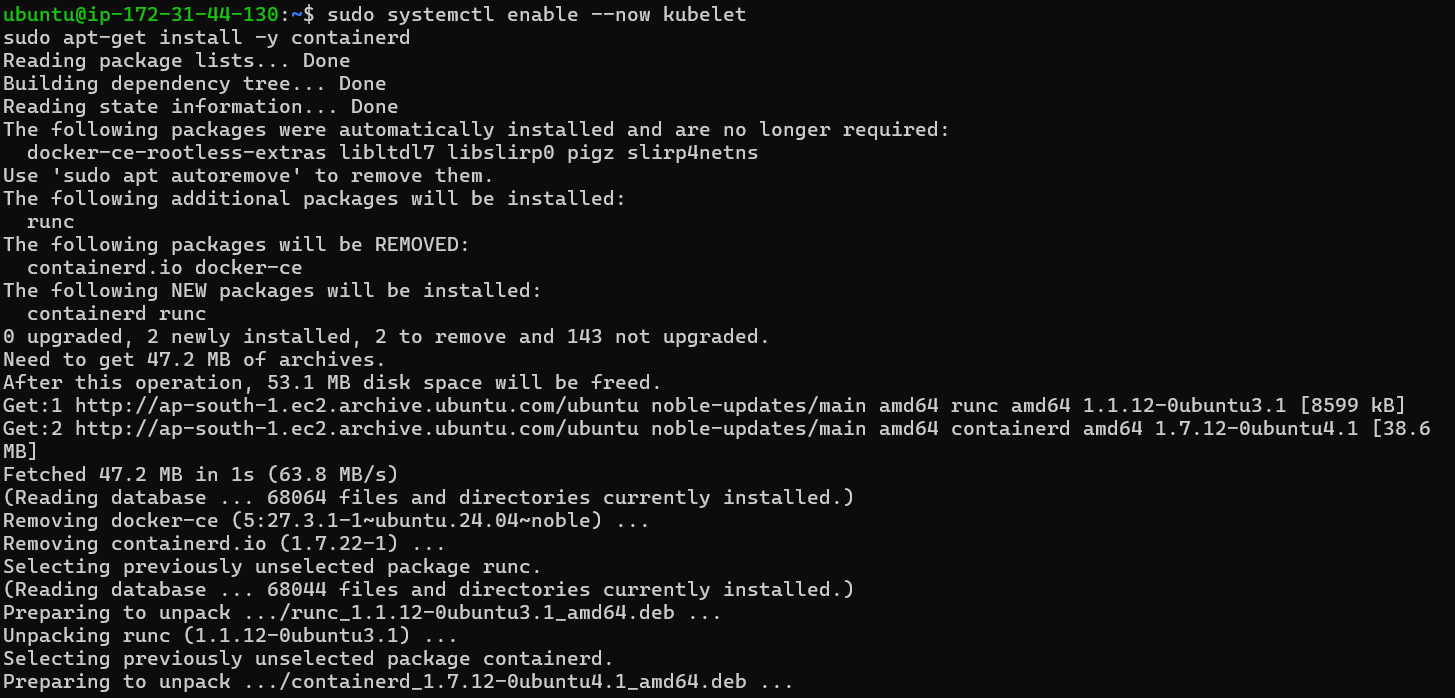
**sudo systemctl daemon-reload**

**sudo systemctl restart docker**   


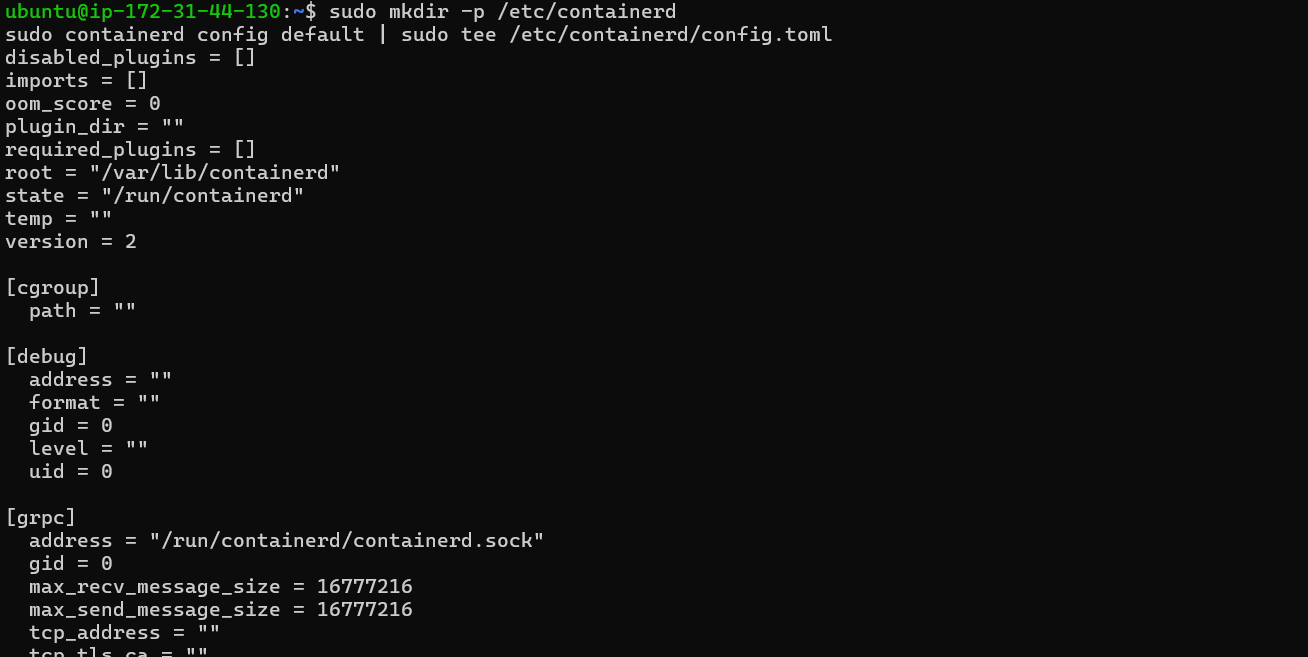
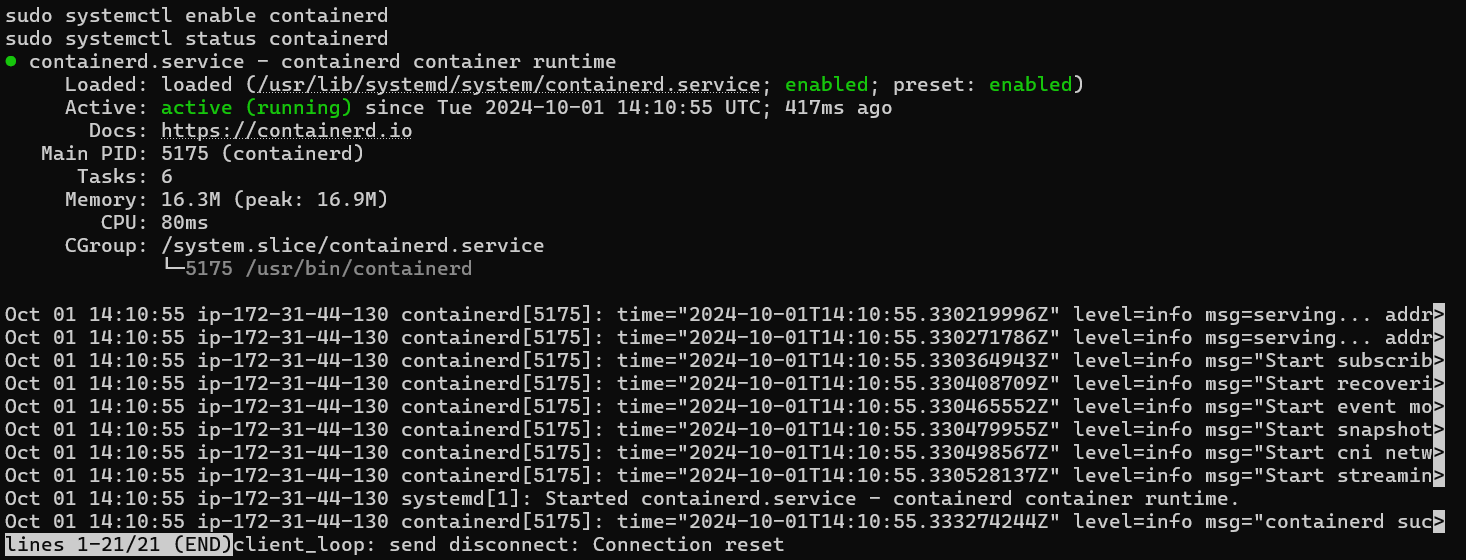
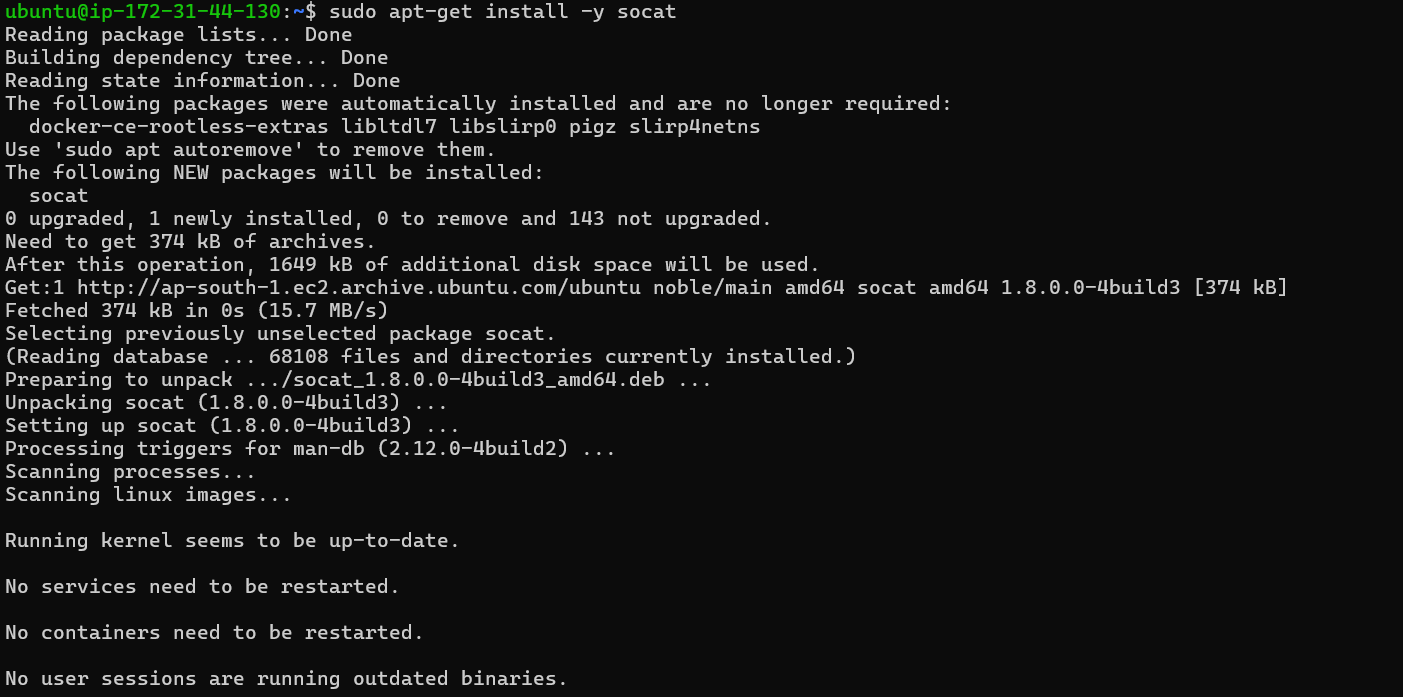
**Step 5: Run the below command to install Kubernets. curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg**

**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   
  
And  
sudo apt-get update   
sudo apt-get install -y kubelet kubeadm kubectl   
sudo apt-mark hold kubelet kubeadm kubectl   
  
If any errors faced here please refer to:-**[**https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/**](https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/) **And**[**https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/**](https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/)  


**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  
  
And  
  
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   
  
If any errors in this command , run:-  
1.) Enable IP Forwarding by running the following command:  
sudo sysctl -w net.ipv4.ip\_forward=1**

**2.) Make the Change Persistent (to ensure it remains active after a reboot):**

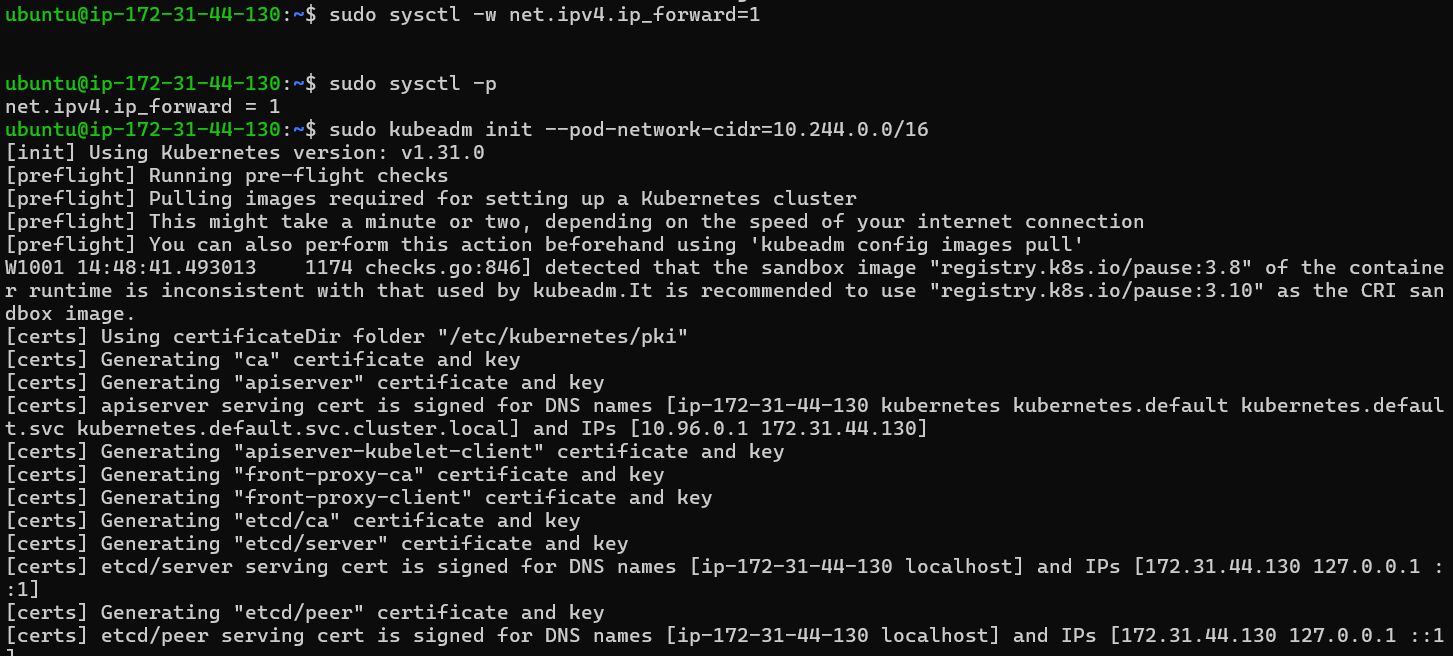
**Open the system control configuration file for editing:  
sudo nano /etc/sysctl.conf**

**3.) Add or modify the following line:  
net.ipv4.ip\_forward=1**

**4.) Save the file and exit the editor.(Click ctrl+X,then Yes and then Enter)**

**5.) Apply the Changes:  
sudo sysctl -p**

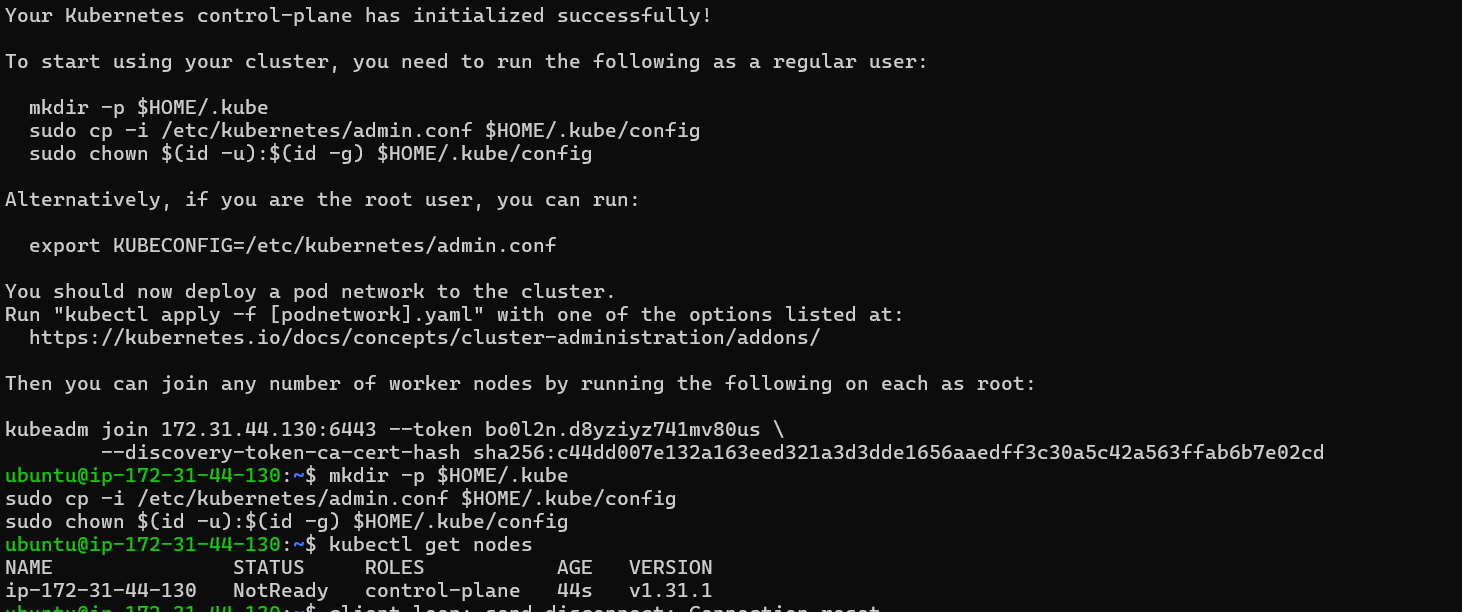
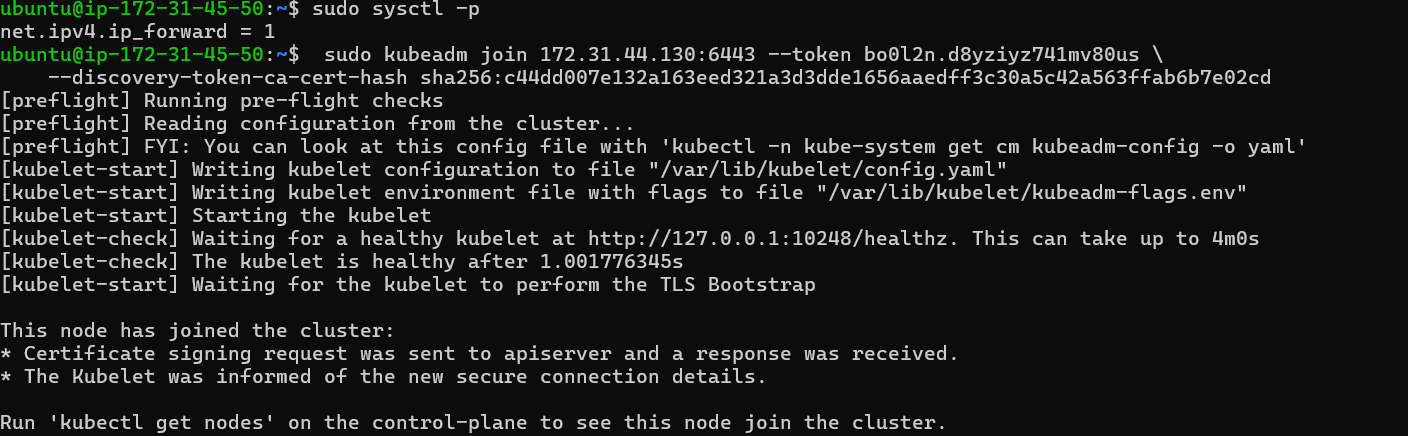
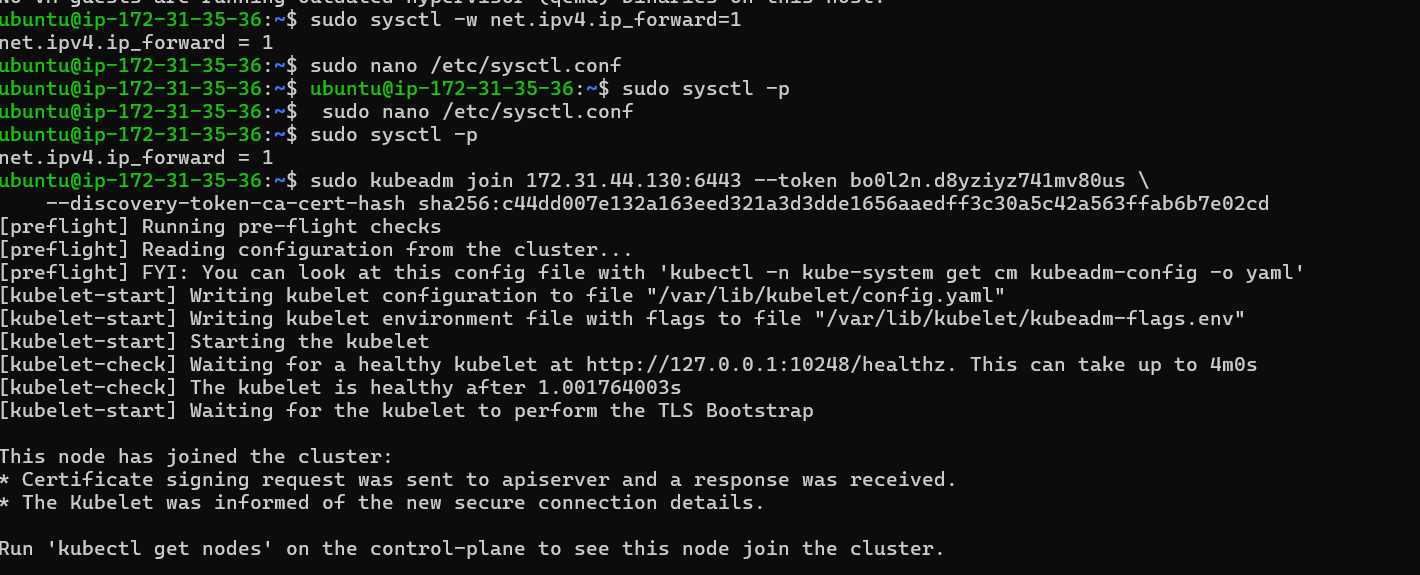
**6.) Re-run the kubeadm init command:  
sudo kubeadm init --pod-network-cidr=10.244.0.0/16**

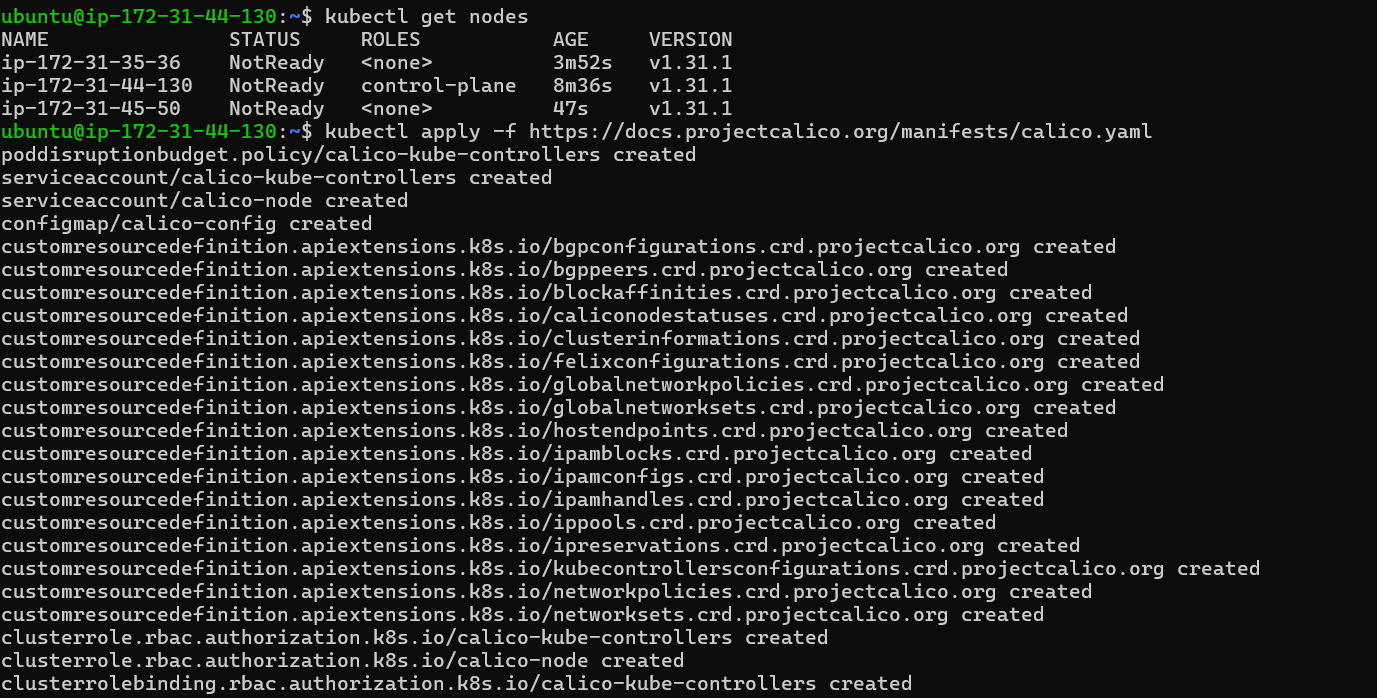
  
  
  
  
  
  
  
  
**Run this command on master and also copy and save the Join command from below.**

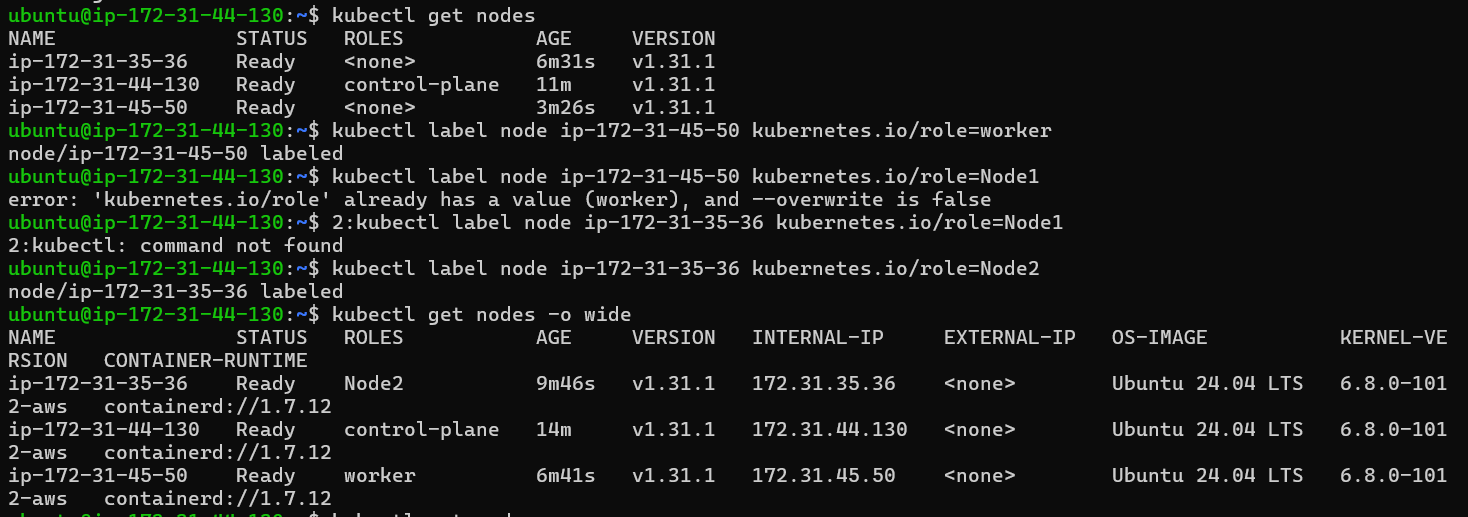
**mkdir -p $HOME/.kube**

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

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

  
**Join command:-**  
  
**Step 8: Now Run the following command on Node 1 and Node 2 to Join to master. sudo kubeadm join <EC2 Instance Ip> --token *<*randomly*\_*alloted\_token>\ --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.   
  
And   
  
Step 10: 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**  
  
  
  
**Now Run command kubectl get nodes -o wide  
Or   
kubectl get nodes so   
 we can see Status is ready.  
  
Renaming:-  
Rename to Node 1:kubectl label node ip-172-31-45-50 kubernetes.io/role=Node1**

**Rename to Node 2:kubectl label node ip-172-31-35-36 kubernetes.io/role=Node2**  
 **run kubectl get nodes**