

**Now we need to install Kubernetes in all the machines using this command-**

# disable swap

sudo swapoff -a

# Create the .conf file to load the modules at bootup

cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf

overlay

br\_netfilter

EOF

sudo modprobe overlay

sudo modprobe br\_netfilter

# sysctl params required by setup, params persist across reboots

cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf

net.bridge.bridge-nf-call-iptables = 1

net.bridge.bridge-nf-call-ip6tables = 1

net.ipv4.ip\_forward = 1

EOF

# Apply sysctl params without reboot

sudo sysctl --system

## Install CRIO Runtime

sudo apt-get update -y

sudo apt-get install -y software-properties-common curl apt-transport-https ca-certificates gpg

sudo curl -fsSL https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/cri-o-apt-keyring.gpg

echo "deb [signed-by=/etc/apt/keyrings/cri-o-apt-keyring.gpg] https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/ /" | sudo tee /etc/apt/sources.list.d/cri-o.list

sudo apt-get update -y

sudo apt-get install -y cri-o

sudo systemctl daemon-reload

sudo systemctl enable crio --now

sudo systemctl start crio.service

sudo apt install docker.io -y

echo "CRI runtime installed successfully"

# Add Kubernetes APT repository and install required packages

curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.29/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:/v1.29/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list

sudo apt-get update -y

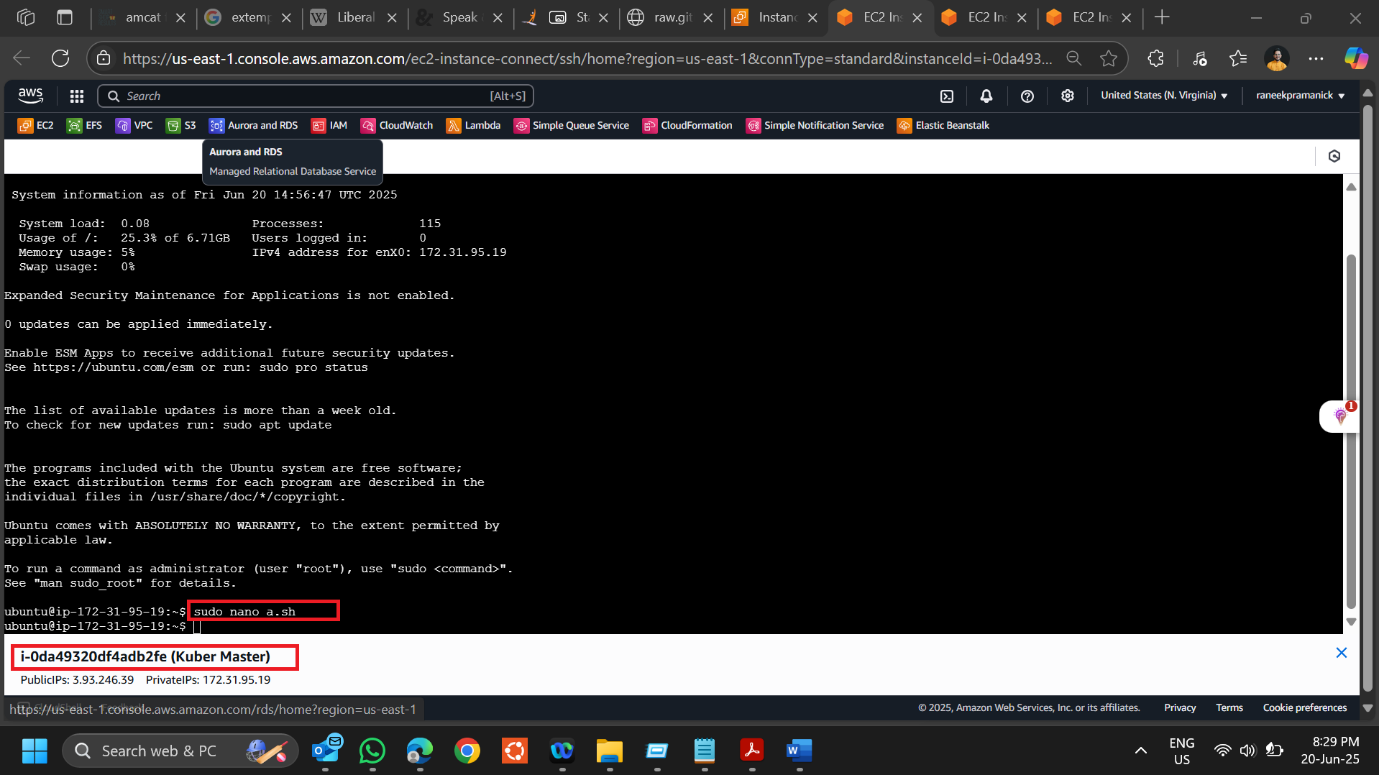
sudo apt-get install -y kubelet="1.29.0-\*" kubectl="1.29.0-\*" kubeadm="1.29.0-\*"

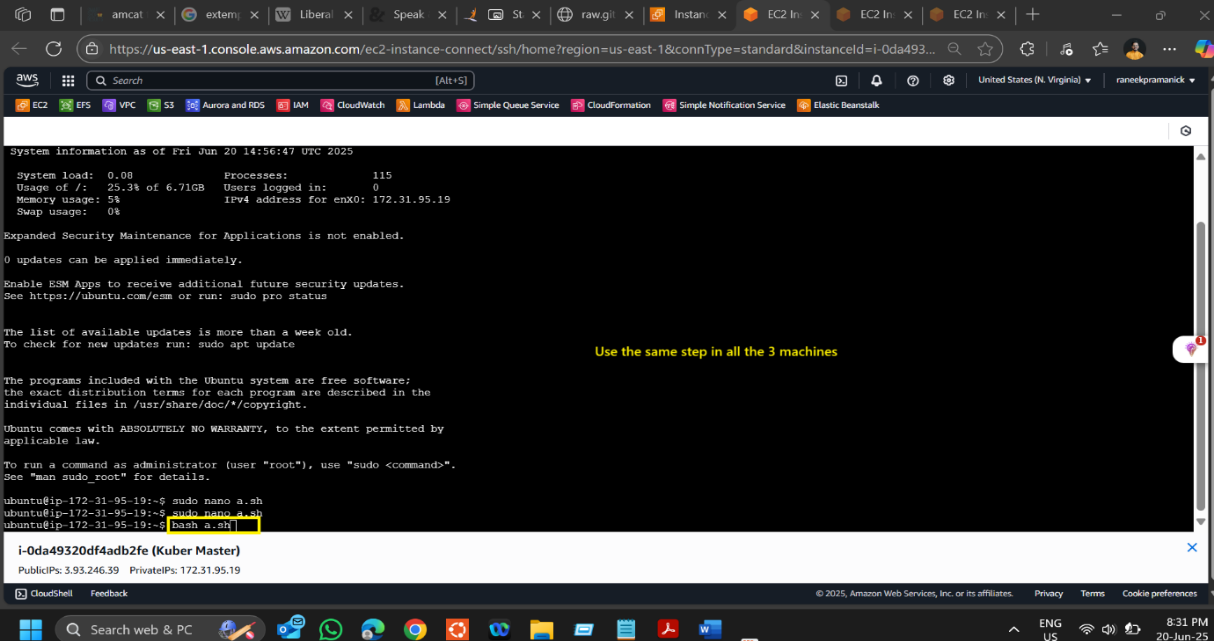
sudo apt-get update -y

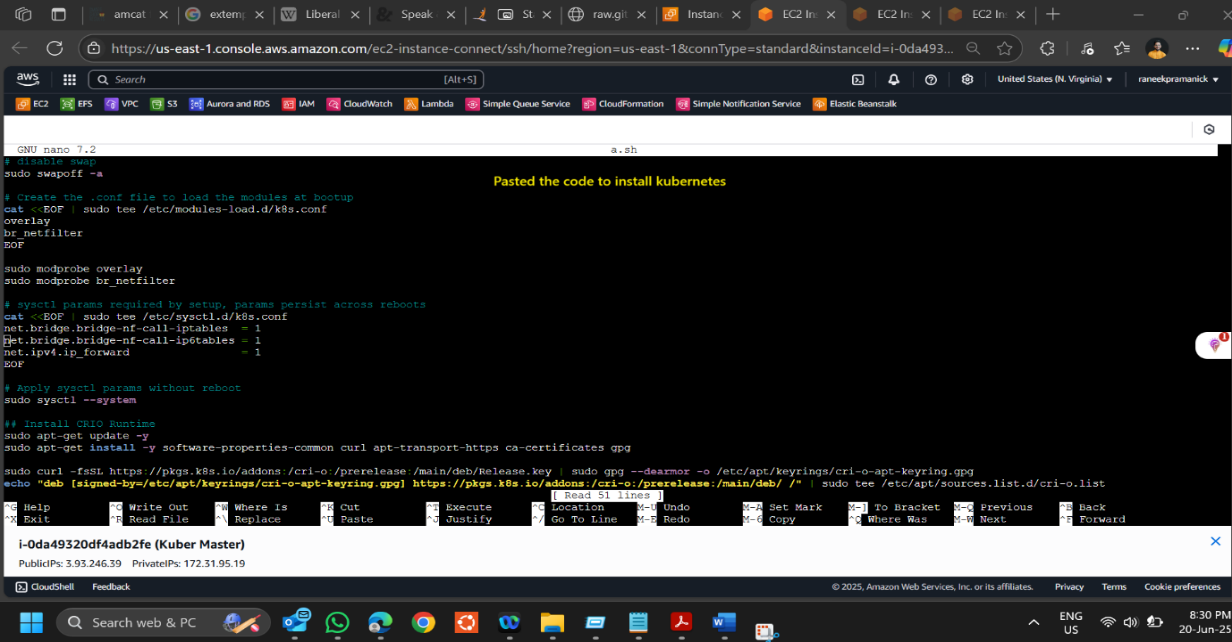
sudo apt-get install -y jq

sudo systemctl enable --now kubelet

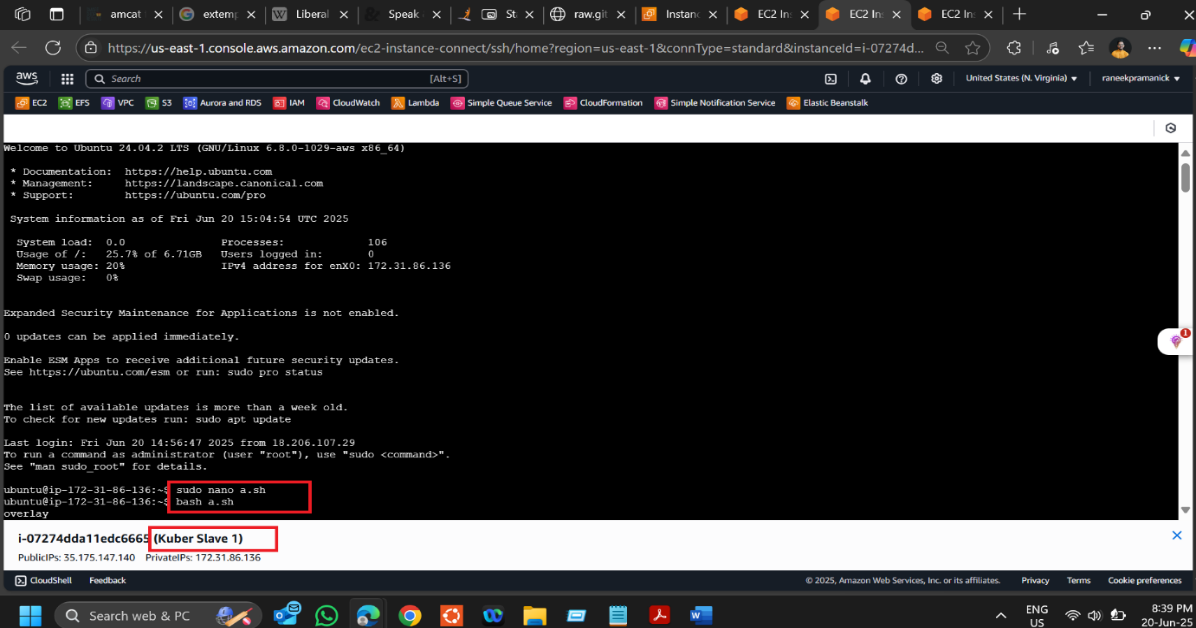
sudo systemctl start kubelet

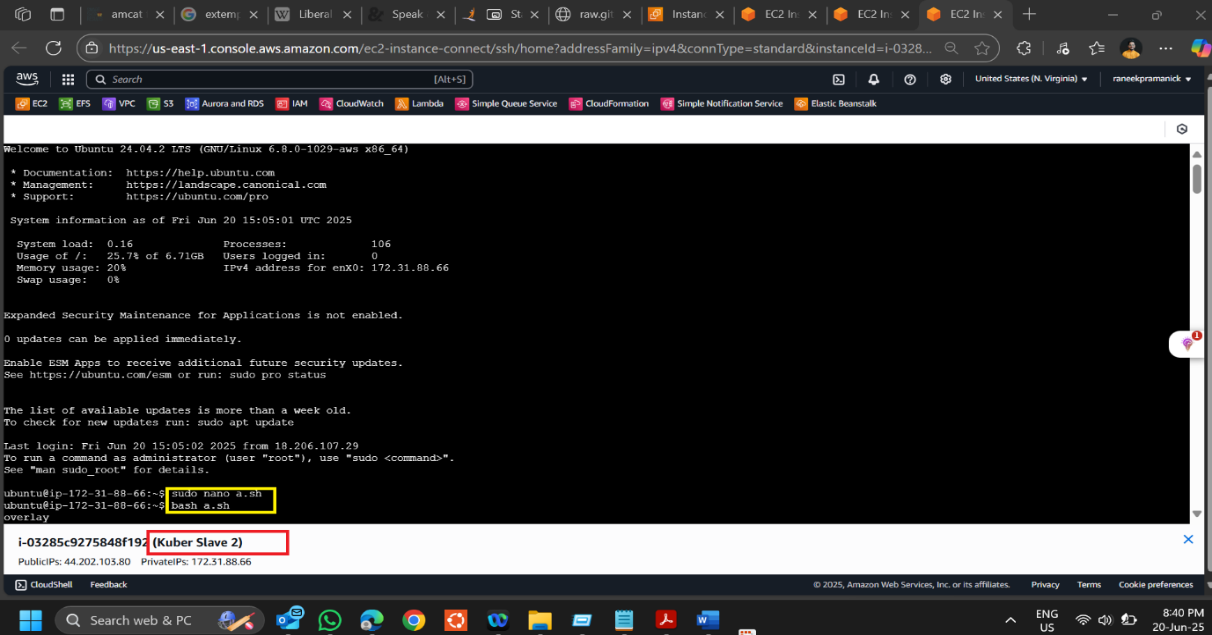






**Now we need to use the same in all the 3 machines to insatall Kubernetes using** sudo nano a.sh and to run it use bash a.sh





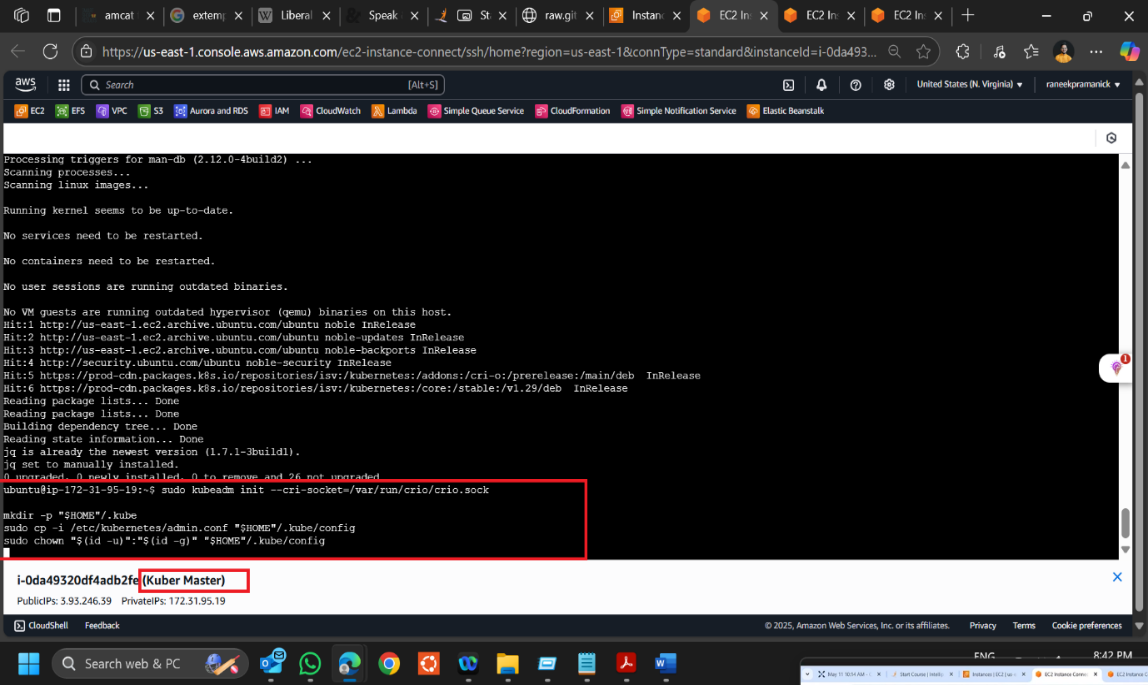
**Now we need to Execute this step ONLY on "Master Node"**

sudo kubeadm init --cri-socket=/var/run/crio/crio.sock

mkdir -p "$HOME"/.kube

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

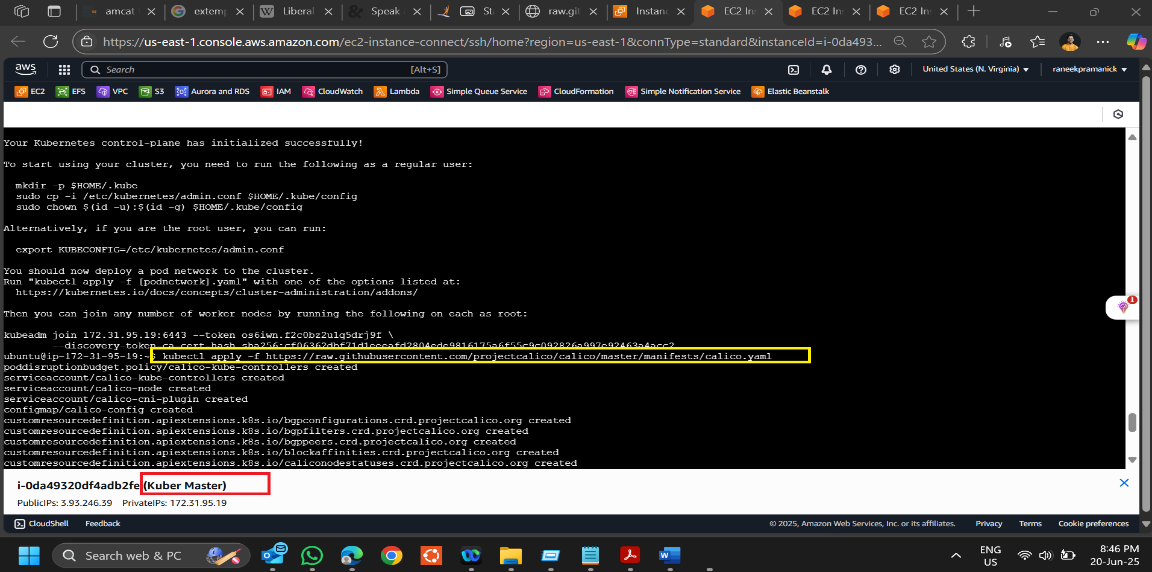
sudo chown "$(id -u)":"$(id -g)" "$HOME"/.kube/config



**Now we need to use this command in the master node.**

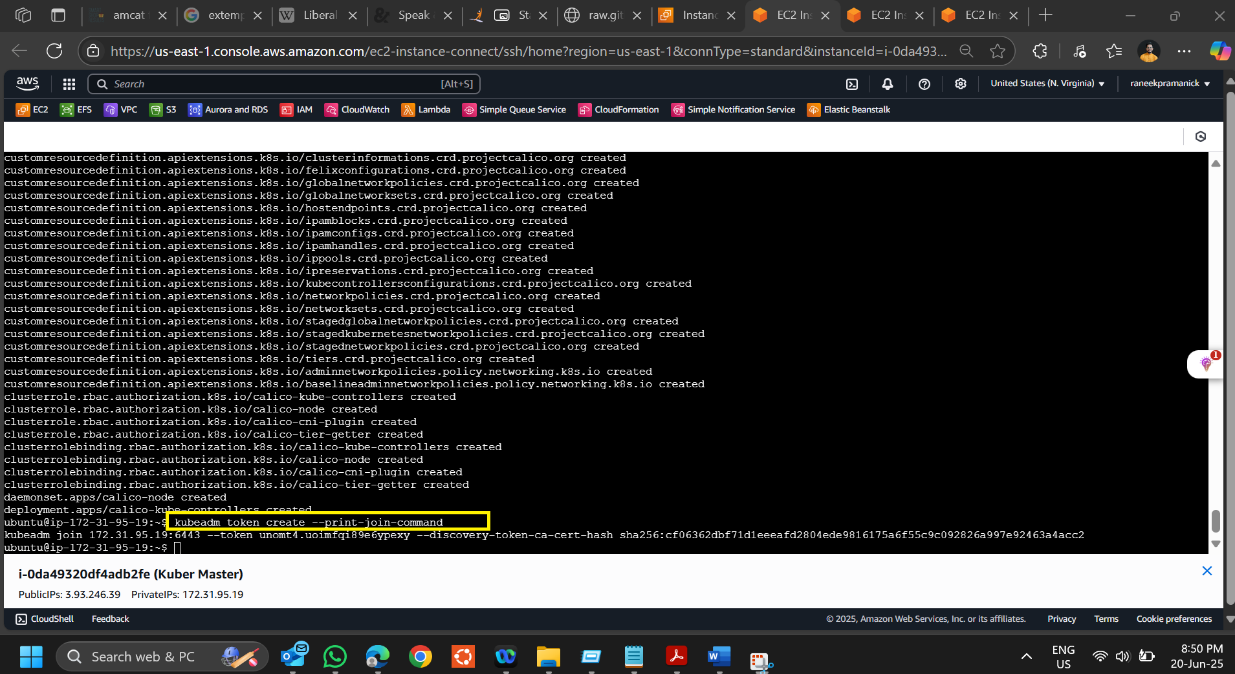
# Network Plugin = calico

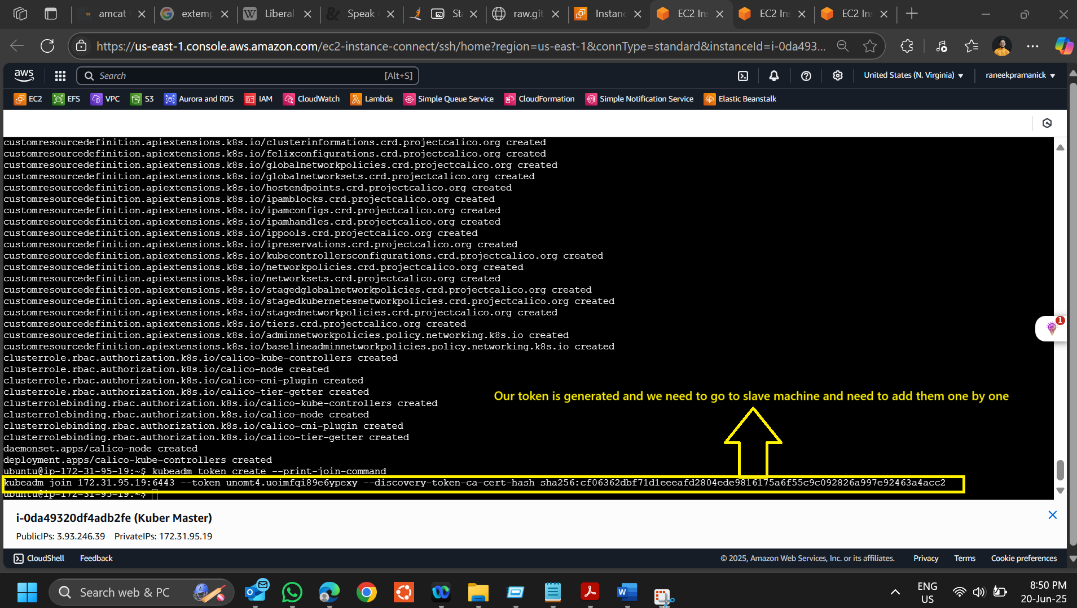
kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/master/manifests/calico.yaml



**Now we need to add slave machines to master, so for that we need to use this command in the master**

kubeadm token create --print-join-command





**1. Paste the join command you got from the master node.**

sudo <your-token> --cri-socket=/var/run/crio/crio.sock

