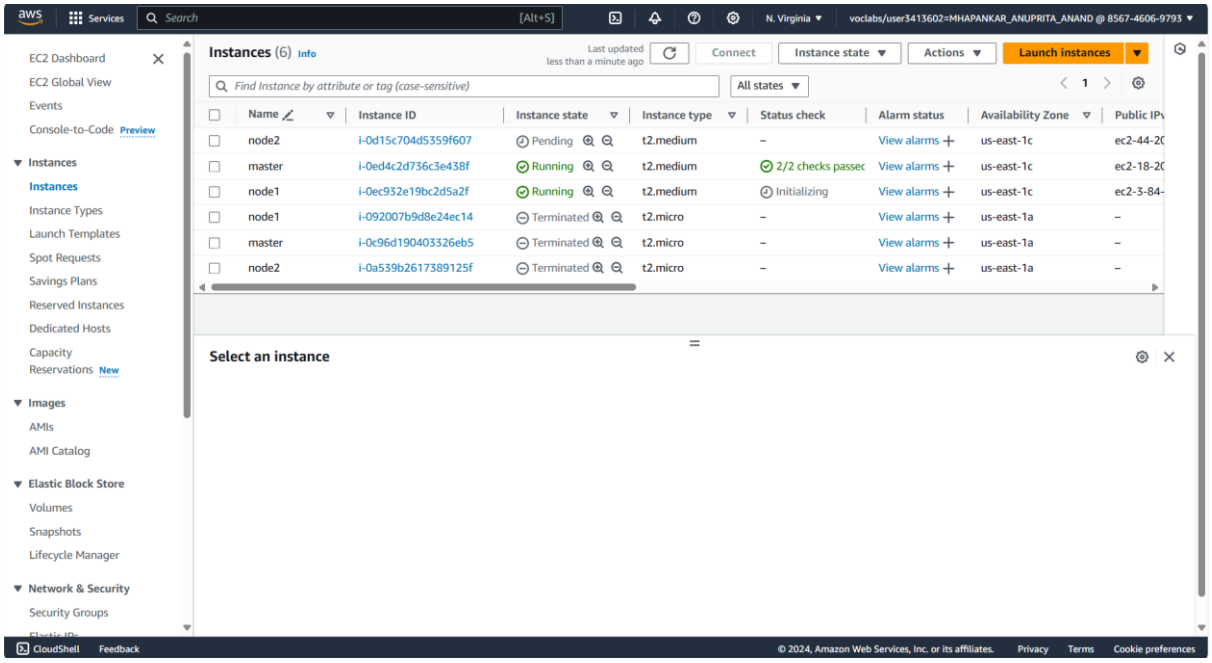
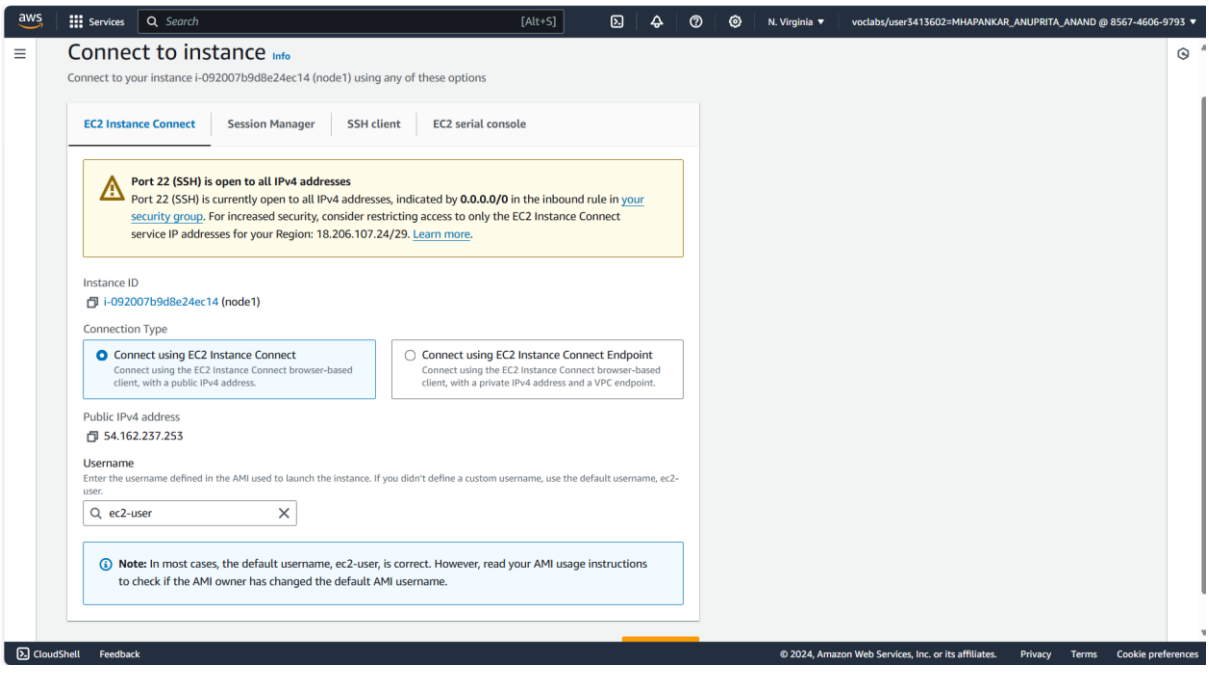
Advanced DevOps Experiment 3

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

Step 1: Go to AWS Academia in services select EC2 and create 3 instance with instance type t2.medium and names as node1, node2 and master





Step 2: Select and connect each instance and run the following commands inside the console of each instance.

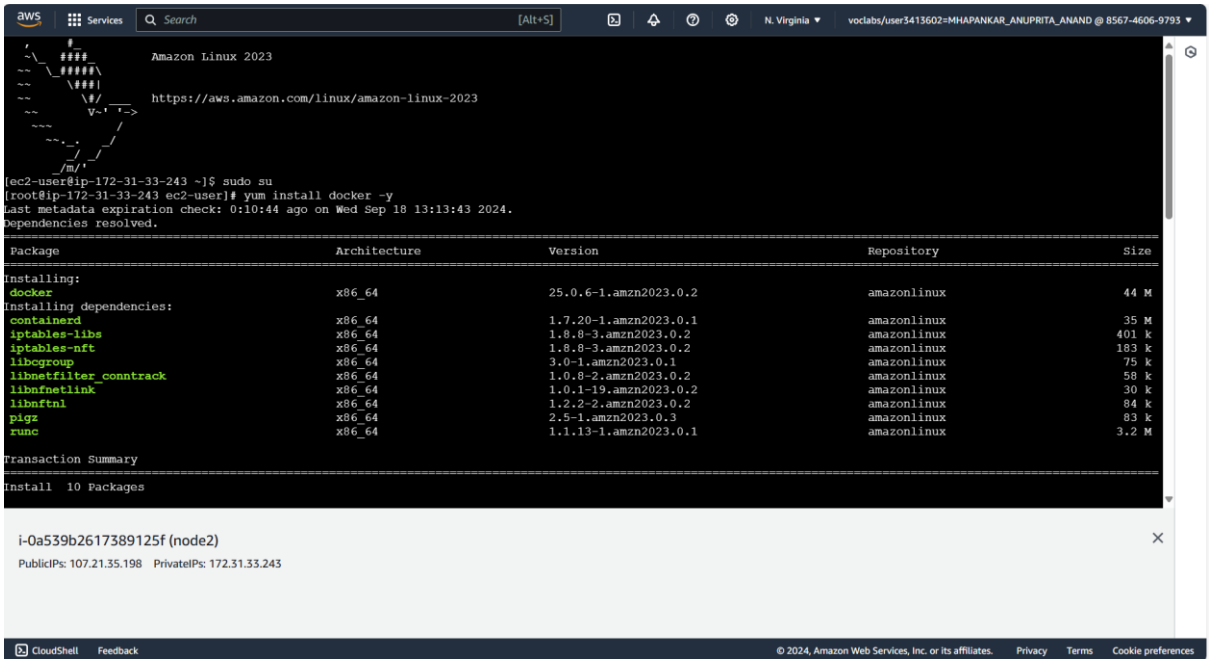
sudo su

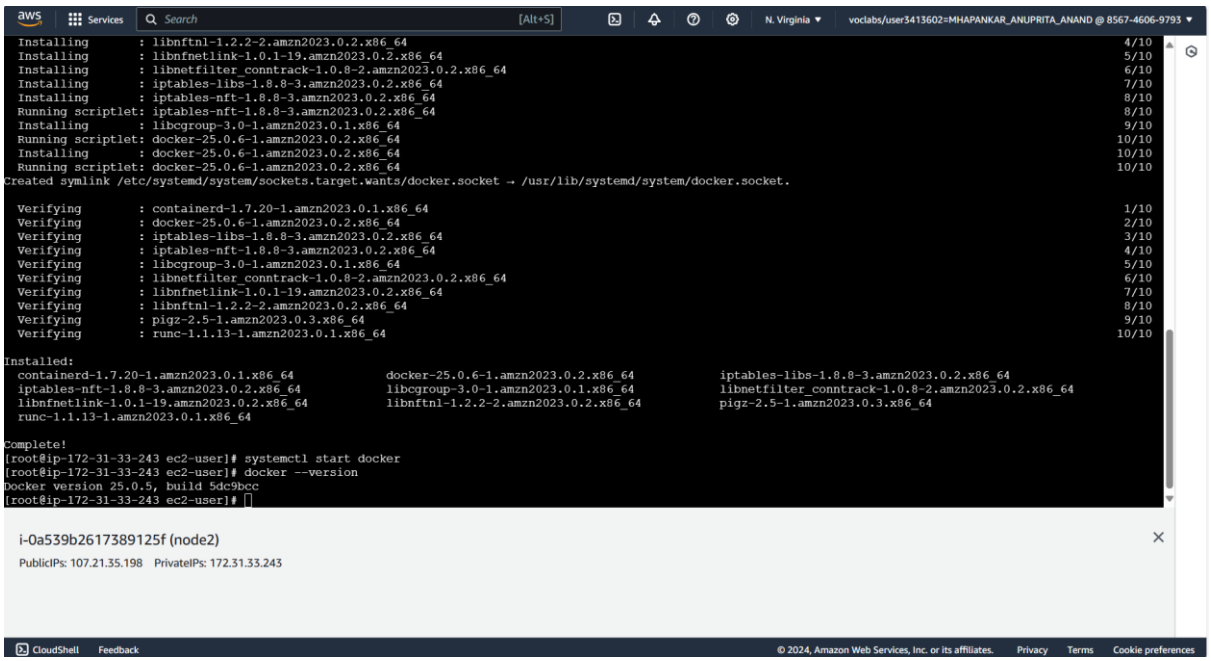
yum install docker -y

systemctl start docker

docker –version

yum repolist





Step 3: Now, go to the following link https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/ and scroll down and select Red-Hat based distributions tab copy all the commands on by one in each console of instance.

kubernetes Documentation Kubernetes Blog Training Partners Community Case Studies Versions English Search this site

Search this site

- Documentation
- Getting started
  - Learning environment
  - Production environment
    - Container Runtimes
    - Installing Kubernetes with deployment tools
      - Bootstrapping clusters with kubeadm
        - **Installing kubeadm**
        - Troubleshooting kubeadm
        - Creating a cluster with kubeadm
        - Customizing components with the kubeadm API
        - Options for Highly Available Topology

Debian-based distributions | Red Hat-based distributions | Without a package manager

1. Set SELinux to `permissive` mode:

   These instructions are for Kubernetes 1.31.

```
# Set SELinux in permissive mode (effectively disabling it)
sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
```

Caution:

- Setting SELinux in permissive mode by running `setenforce 0` and `sed ...` effectively disables it. This is required to allow containers to access the host filesystem; for example, some cluster network plugins require that. You have to do this until SELinux support is improved in the kubelet.
- You can leave SELinux enabled if you know how to configure it but it may require settings that are not supported by kubeadm.

2. Add the Kubernetes `yum` repository. The `exclude` parameter in the repository definition ensures that the packages related to Kubernetes are not upgraded upon running `yum update` as there's a special procedure that must be followed for upgrading Kubernetes. Please note that this repository have packages only for Kubernetes 1.31; for other Kubernetes minor versions, you need to change the Kubernetes minor version in the URL to match your desired minor version (you should also check that you are reading the documentation for the version of Kubernetes that you plan to install).

Edit this page
Create child page
Create documentation issue
Print entire section

Before you begin
Verify the MAC address and product_uuid are unique for every node
Check network adapters
Check required ports
Swap configuration
Installing a container runtime
Installing kubeadm, kubelet and kubectl
Configuring a cgroup driver
Troubleshooting
What's next

aws Services Search [Alt+S] N. Virginia voclabs/user3413602=MHAPANKAR_ANUPRITA_ANAND @ 8567-4606-9793

```
[root@ip-172-31-33-243 ec2-user]# yum repolist
repo id                                          repo name
amazonlinux                                      Amazon Linux 2023 repository
kernel-livepatch                                 Amazon Linux 2023 Kernel Livepatch repository
[root@ip-172-31-33-243 ec2-user]# sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
[root@ip-172-31-33-243 ec2-user]# cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repodata/repomd.xml.key
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
[root@ip-172-31-33-243 ec2-user]# sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
Kubernetes                                                          54 kB/s | 9.4 kB     00:00
Dependencies resolved.
==============================================================================================
 Package              Architecture       Version                      Repository        Size
==============================================================================================
Installing:
 kubeadm              x86_64             1.31.1-150500.1.1            kubernetes        11 M
 kubectl              x86_64             1.31.1-150500.1.1            kubernetes        11 M
 kubelet              x86_64             1.31.1-150500.1.1            kubernetes        15 M
Installing dependencies:
 conntrack-tools      x86_64             1.4.6-2.amzn2023.0.2         amazonlinux      208 k
```

i-0a539b2617389125f (node2)

PublicIPs: 107.21.35.198 PrivateIPs: 172.31.33.243

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws Services Search [Alt+S] N. Virginia voclabs/user3413602=MHAPANKAR_ANUPRITA_ANAND @ 8567-4606-9793

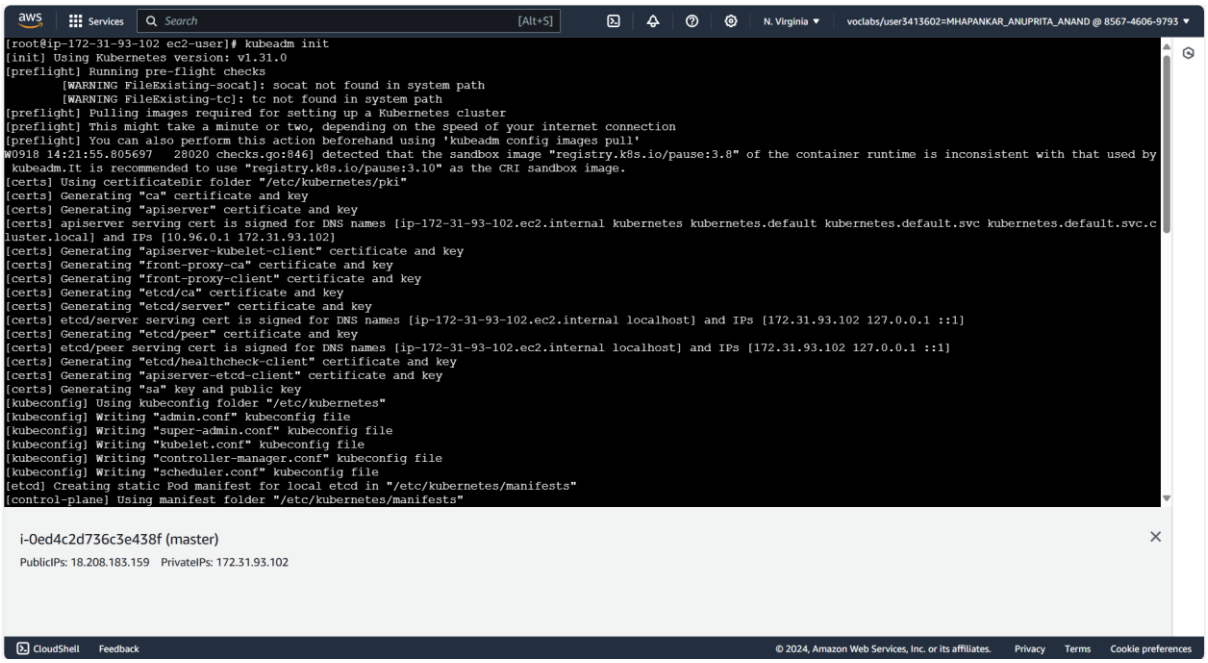
```
Transaction test succeeded.
Running transaction
  Preparing        :                                                           1/1
  Installing       : kubernetes-cni-1.5.1-150500.1.1.x86_64                    1/9
  Installing       : cri-tools-1.31.1-150500.1.1.x86_64                        2/9
  Installing       : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64            3/9
  Installing       : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64       4/9
  Installing       : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64        5/9
  Installing       : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64               6/9
  Running scriptlet: conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64               6/9
  Installing       : kubelet-1.31.1-150500.1.1.x86_64                          7/9
  Running scriptlet: kubelet-1.31.1-150500.1.1.x86_64                          7/9
  Installing       : kubeadm-1.31.1-150500.1.1.x86_64                          8/9
  Installing       : kubectl-1.31.1-150500.1.1.x86_64                          9/9
  Running scriptlet: kubectl-1.31.1-150500.1.1.x86_64                          9/9
  Verifying        : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64               1/9
  Verifying        : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64        2/9
  Verifying        : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64       3/9
  Verifying        : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64            4/9
  Verifying        : cri-tools-1.31.1-150500.1.1.x86_64                        5/9
  Verifying        : kubeadm-1.31.1-150500.1.1.x86_64                          6/9
  Verifying        : kubectl-1.31.1-150500.1.1.x86_64                          7/9
  Verifying        : kubelet-1.31.1-150500.1.1.x86_64                          8/9
  Verifying        : kubernetes-cni-1.5.1-150500.1.1.x86_64                    9/9

Installed:
  conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64     cri-tools-1.31.1-150500.1.1.x86_64                     kubeadm-1.31.1-150500.1.1.x86_64
  kubectl-1.31.1-150500.1.1.x86_64                kubelet-1.31.1-150500.1.1.x86_64                       kubernetes-cni-1.5.1-150500.1.1.x86_64
  libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64   libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64   libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64

Complete!
[root@ip-172-31-33-243 ec2-user]# sudo systemctl enable --now kubelet
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /usr/lib/systemd/system/kubelet.service.
[root@ip-172-31-33-243 ec2-user]#
```

i-0a539b2617389125f (node2)

PublicIPs: 107.21.35.198 PrivateIPs: 172.31.33.243

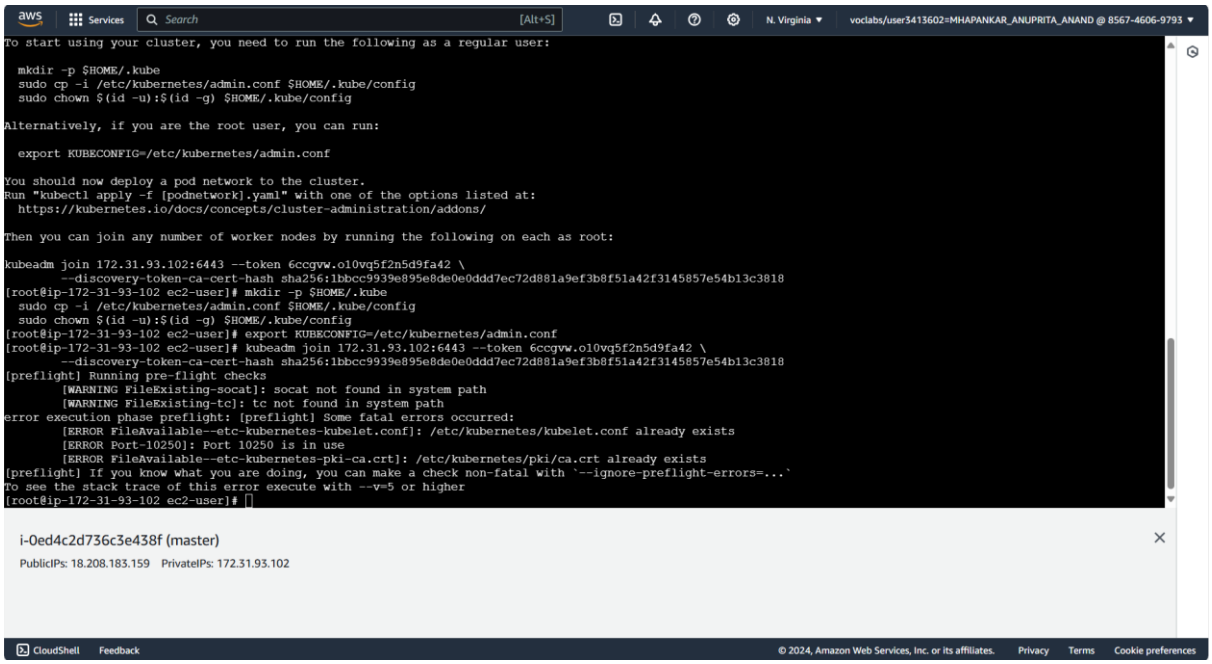
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 4: Now, run the following command in the mater instance -
kubeadm init

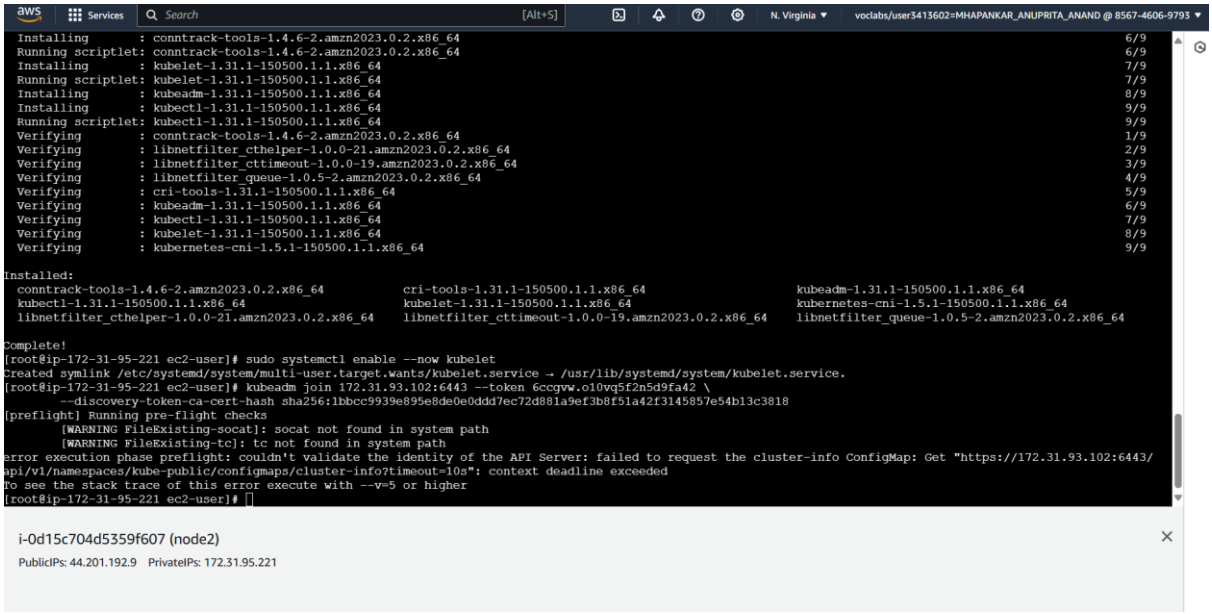


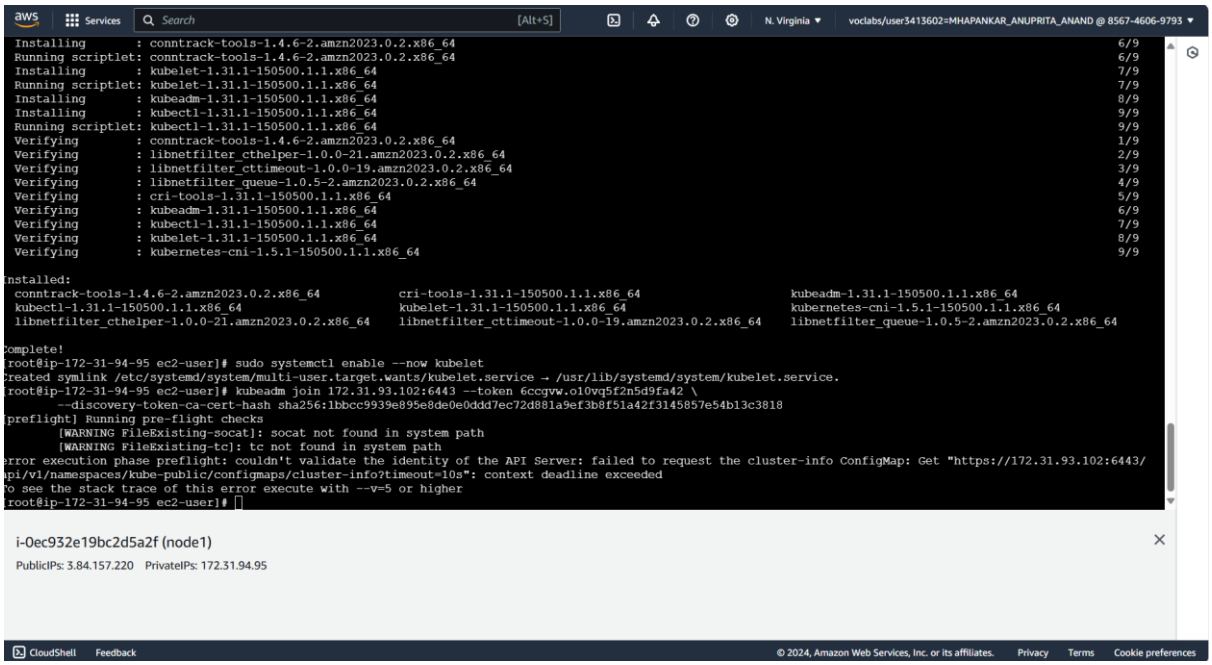
Step 5: Now, run the following commands in master instance's console –

a. mkdir -p $HOME/.kube
   sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
   sudo chown $(id -u):$(id -g) $HOME/.kube/config
b. export KUBECONFIG=/etc/kubernetes/admin.conf
c. kubeadm join 172.31.93.102:6443 --token 6ccgvw.o10vq5f2n5d9fa42 \
   --discovery-token-ca-cert-hash
   sha256:1bbcc9939e895e8de0e0ddd7ec72d881a9ef3b8f51a42f3145857e54b13c3818



Step 6: Run this command in node1 and node2 -
kubeadm join 172.31.93.102:6443 --token 6ccgvw.o10vq5f2n5d9fa42 \
--discovery-token-ca-cert-hash sha256:1bbcc9939e895e8de0e0ddd7ec72d881a9ef3b8f51a42f3145857e54b13c3818





Step 7: Run the following command in master instance console -
kubectl get nodes