

## Experiment No. 3

**Aim:** To understand the Kubernetes Cluster Architecture, install and Spin Up a Kubernetes Cluster on a Linux Machine/ Cloud Platforms. (LO1, LO2)

### Theory:

#### What is Kubernetes Cluster?

A Kubernetes cluster is a set of nodes that run containerized applications. Containerizing applications packages an app with its dependences and some necessary services. They are more lightweight and flexible than virtual machines. In this way, Kubernetes clusters allow for applications to be more easily developed, moved and managed.

Kubernetes clusters allow containers to run across multiple machines and environments: virtual, physical, cloud-based, and on-premises. Kubernetes containers are not restricted to a specific operating system, unlike virtual machines. Instead, they are able to share operating systems and run anywhere.

Kubernetes clusters are comprised of one master node and a number of worker nodes. These nodes can either be physical computers or virtual machines, depending on the cluster.

The master node controls the state of the cluster; for example, which applications are running and their corresponding container images. The master node is the origin for all task assignments. It coordinates processes such as:

- Scheduling and scaling applications
- Maintaining a cluster's state
- Implementing updates

The worker nodes are the components that run these applications. Worker nodes perform tasks assigned by the master node. They can either be virtual machines or physical computers, all operating as part of one system.

Kubernetes provides you with:

- **Service discovery and load balancing:** Kubernetes can expose a container using the DNS name or using their own IP address. If traffic to a container is high, Kubernetes is able to load balance and distribute the network traffic so that the deployment is stable.
- **Storage orchestration:** Kubernetes allows you to automatically mount a storage system of your choice, such as local storages, public cloud providers, and more.
- **Automated rollouts and rollbacks:** You can describe the desired state for your deployed containers using Kubernetes, and it can change the actual state to the desired state at a controlled rate. For example, you can automate Kubernetes to create new containers for your deployment, remove existing containers and adopt all their resources to the new container.
- **Automatic bin packing:** You provide Kubernetes with a cluster of nodes that it can use to run containerized tasks. You tell Kubernetes how much CPU and memory (RAM) each container needs. Kubernetes can fit containers onto your nodes to make the best use of your resources.

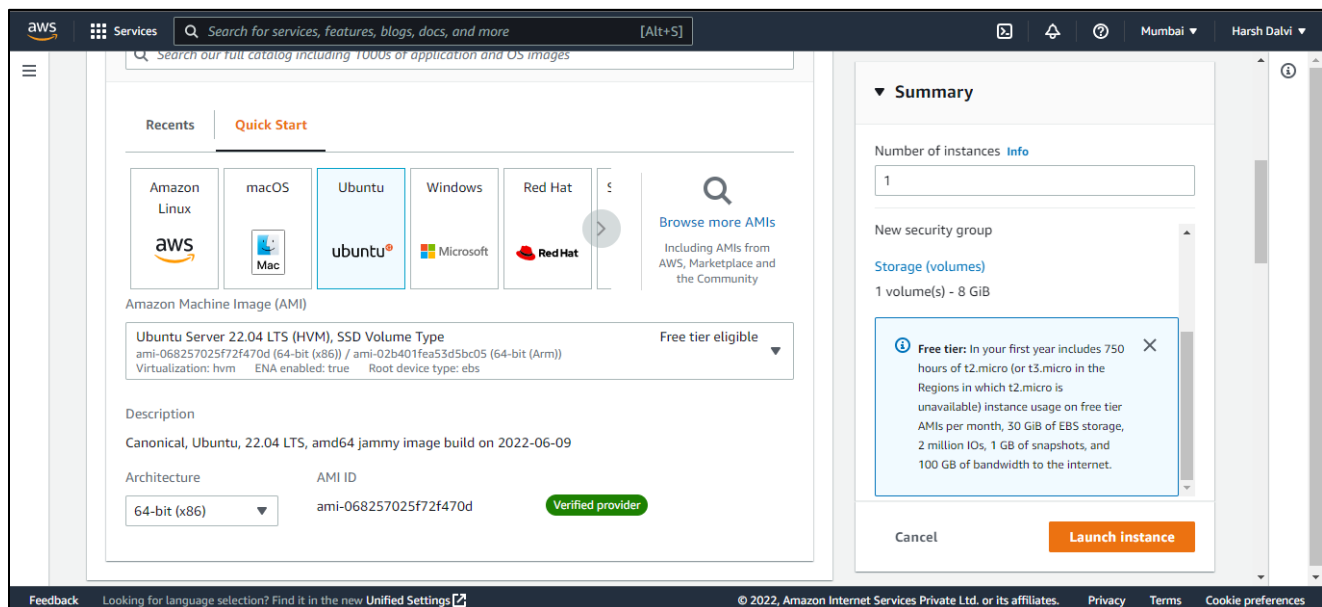
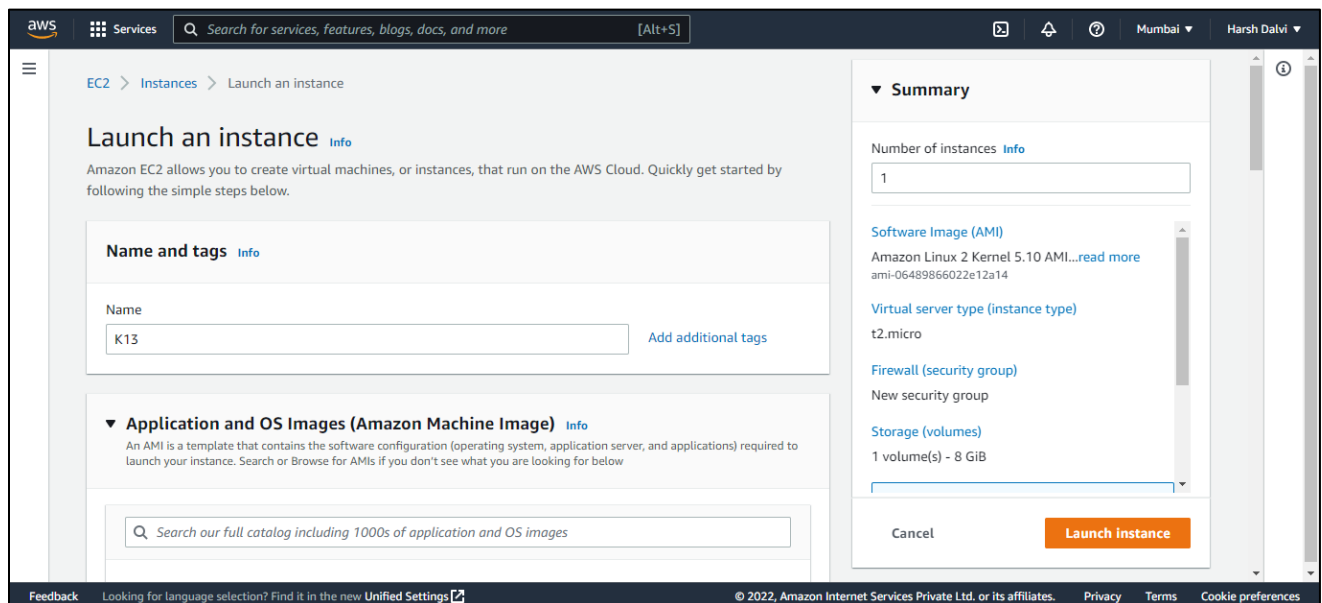
Name: Harsh Dalvi  
Roll No: 13

Subject: Advance DevOps , Sem: SEM V  
Class / Batch: TE-IT / Batch B

- **Self-healing:** Kubernetes restarts containers that fail, replaces containers, kills containers that don't respond to your user-defined health check, and doesn't advertise them to clients until they are ready to serve.
- **Secret and configuration management:** Kubernetes lets you store and manage sensitive information, such as passwords, OAuth tokens, and SSH keys. You can deploy and update secrets and application configuration without rebuilding your container images, and without exposing secrets in your stack configuration.

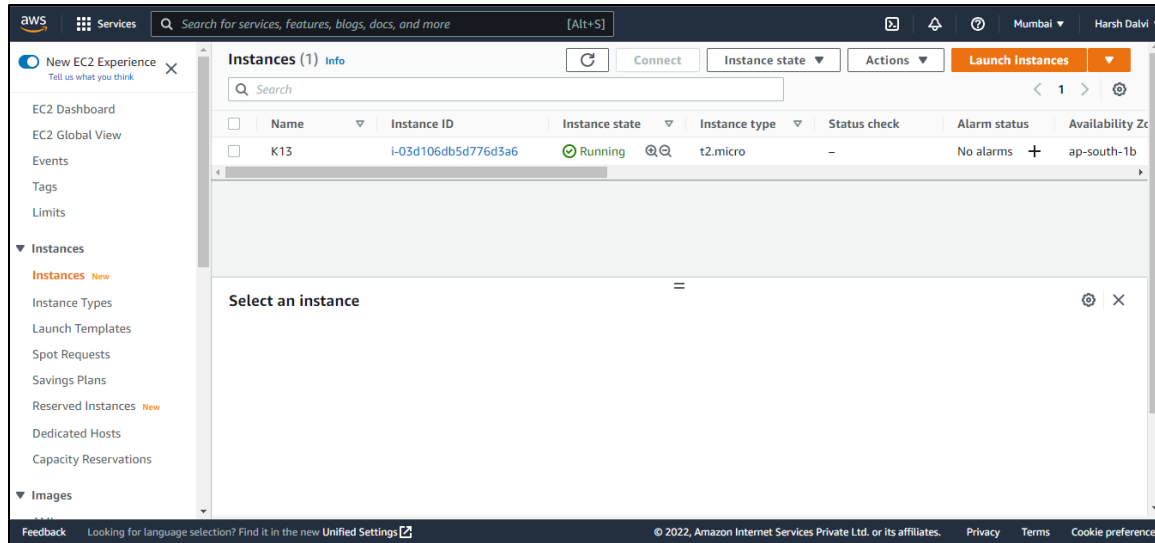
## Steps to perform the experiment:

### Step 1: Create an Instance of Linux Machine

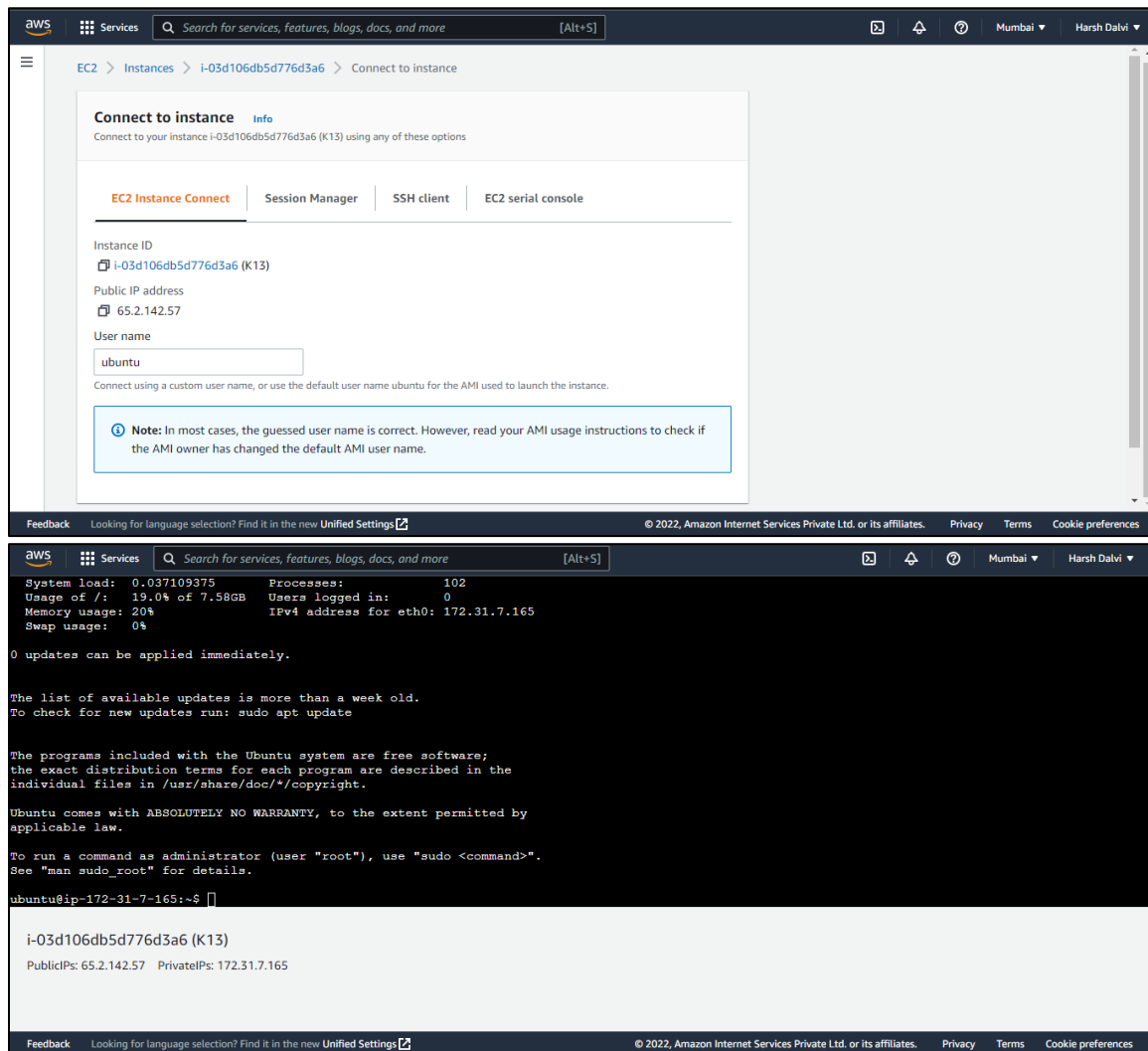


Name: Harsh Dalvi  
Roll No: 13

Subject: Advance DevOps , Sem: SEM V  
Class / Batch: TE-IT / Batch B



## Step 2: Connect Instance with the help of putty



Name: Harsh Dalvi  
Roll No: 13

Subject: Advance DevOps , Sem: SEM V  
Class / Batch: TE-IT / Batch B

```
ubuntu@ip-172-31-7-165:~$ sudo su -
root@ip-172-31-7-165:~# sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [295 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [69.1 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [3924 B]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [274 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:11 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [42.3 kB]
Get:12 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 c-n-f Metadata [524 B]
Get:13 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [121 kB]
Get:14 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [41.5 kB]
Get:15 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [2408 B]
Get:16 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [4192 B]
```

```
Get:24 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [128 kB]
Get:25 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [8136 B]
Get:26 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [306 kB]
Get:27 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [47.5 kB]
Get:28 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [524 B]
Get:29 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [254 kB]
Get:30 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [83.3 kB]
Get:31 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [4404 B]
Get:32 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [7000 B]
Get:33 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [2264 B]
Get:34 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [420 B]
Get:35 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [3008 B]
Get:36 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [1432 B]
Get:37 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [272 B]
Get:38 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]
Get:39 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [6724 B]
Get:40 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [9216 B]
Get:41 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [352 B]
Get:42 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]
Fetched 23.0 MB in 6s (3970 kB/s)
Reading package lists... Done
root@ip-172-31-7-165:~#
```

### Step 3: Check the Docker Version

```
apt install podman-docker # version 3.4.4+ds1-1ubuntu1
root@ip-172-31-7-165:~# apt install docker.io -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 77 not upgraded.
Need to get 65.6 MB of archives.
After this operation, 283 MB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63.6 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 bridge-utils amd64 1.7-1ubuntu3 [34.4 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 runc amd64 1.1.0-0ubuntu1 [4087 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 containerd amd64 1.5.9-0ubuntu3 [27.0 MB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 dns-root-data all 2021011101 [5256 B]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 dnsmasq-base amd64 2.86-1.1ubuntu0.1 [354 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 docker.io amd64 20.10.12-0ubuntu4 [34.0 MB]
94% [7 docker.io 32.8 MB/34.0 MB 97%] 10.1 MB/s 0s
```

```
root@ip-172-31-7-165:~# docker --version
Docker version 20.10.12, build 20.10.12-0ubuntu4
root@ip-172-31-7-165:~#
```

### Step 4: To enable the docker

```
root@ip-172-31-7-165:~# sudo systemctl enable docker
root@ip-172-31-7-165:~#
```

### Step 5 : To check the status of the docker

```
root@ip-172-31-7-165:~# sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2022-09-07 08:48:00 UTC; 2min 1s ago
   TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 2562 (dockerd)
      Tasks: 7
     Memory: 39.4M
        CPU: 266ms
    CGroup: /system.slice/docker.service
            └─2562 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Sep 07 08:47:59 ip-172-31-7-165 dockerd[2562]: time="2022-09-07T08:47:59.771741544Z" level=info msg="scheme \"unix\" not registered, fallback to default scheme"
Sep 07 08:47:59 ip-172-31-7-165 dockerd[2562]: time="2022-09-07T08:47:59.771905861Z" level=info msg="ccResolverWrapper: sending update to cc: {[{unix://var/run/docker.sock}]}
Sep 07 08:47:59 ip-172-31-7-165 dockerd[2562]: time="2022-09-07T08:47:59.772059483Z" level=info msg="ClientConn switching balancer to \"pick_first\""
Sep 07 08:47:59 ip-172-31-7-165 dockerd[2562]: time="2022-09-07T08:47:59.823344381Z" level=info msg="Loading containers: start."
Sep 07 08:47:59 ip-172-31-7-165 dockerd[2562]: time="2022-09-07T08:47:59.997693067Z" level=info msg="Default bridge (docker0) is assigned with a IP address 172.17.0.1/16
Sep 07 08:48:00 ip-172-31-7-165 dockerd[2562]: time="2022-09-07T08:48:00.098991919Z" level=info msg="Loading containers: done."
Sep 07 08:48:00 ip-172-31-7-165 dockerd[2562]: time="2022-09-07T08:48:00.167660219Z" level=info msg="Docker daemon commit=20.10.12-0ubuntu4 graphdriver=overlay2
Sep 07 08:48:00 ip-172-31-7-165 dockerd[2562]: time="2022-09-07T08:48:00.168134575Z" level=info msg="Daemon has completed initialization"
Sep 07 08:48:00 ip-172-31-7-165 system[1]: Started Docker Application Container Engine.
Sep 07 08:48:00 ip-172-31-7-165 dockerd[2562]: time="2022-09-07T08:48:00.226393618Z" level=info msg="API listen on /run/docker.sock"

lines 1-22
```

### Step 6: Using Curl Command

```
root@ip-172-31-7-165:~# curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
  % Total    % Received    Xferd  Average Speed   Time    Time     Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0     0      0     0      0     0      0     0      0      0      0      0      0  0 --:--:-- --:--:-- --:--:-- Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
100 2537 100 2537  0     0 12217    0 --:--:-- --:--:-- --:--:-- 12256
OK
root@ip-172-31-7-165:~#
```

## Step 7: Adding Repository

```
root@ip-172-31-7-163:~# sudo apt-add-repository "deb http://apt.kubernetes.io/ kubernetes-xenial main"
Repository: 'deb http://apt.kubernetes.io/ kubernetes-xenial main'
Description:
Archive for codename: kubernetes-xenial components: main
More info: http://apt.kubernetes.io/
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
Adding deb entry to /etc/apt/sources.list.d/archive uri=http_apt_kubernetes_io_jammy.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive uri=http_apt_kubernetes_io_jammy.list
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:2 https://packages.cloud.google.com/apt kubernetes-xenial InRelease [9383 B]
Get:6 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 Packages [58.4 kB]
Fetched 67.8 kB in 1s (57.7 kB/s)
Reading package lists... Done
W: http://apt.kubernetes.io/dists/kubernetes-xenial/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION
section in apt-key(8) for details.
```

## Step 8: Install Kubernetes Admin

```
root@ip-172-31-7-165:~# sudo apt-get install kubeadm kubelet kubect1
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools ebttables kubernetes-cni socat
The following NEW packages will be installed:
  conntrack cri-tools ebttables kubeadm kubect1 kubelet kubernetes-cni socat
0 upgraded, 8 newly installed, 0 to remove and 77 not upgraded.
Need to get 75.9 MB of archives.
After this operation, 310 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 conntrack amd64 1:1.4.6-2build2 [33.5 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 ebttables amd64 2.0.11-4build2 [84.9 kB]
Get:2 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 cri-tools amd64 1.24.2-00 [12.3 MB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 socat amd64 1.7.4.1-3ubuntu4 [349 kB]
Get:4 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubernetes-cni amd64 0.8.7-00 [25.0 MB]
```

Name: Harsh Dalvi  
Roll No: 13

Subject: Advance DevOps , Sem: SEM V  
Class / Batch: TE-IT / Batch B

```
Setting up kubect1 (1.25.0-00) ...  
Setting up ebtaltes (2.0.11-4build2) ...  
Setting up socat (1.7.4.1-3ubuntu4) ...  
Setting up cri-tools (1.24.2-00) ...  
Setting up kubernetes-cni (0.8.7-00) ...  
Setting up kubelet (1.25.0-00) ...  
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /lib/systemd/system/kubelet.service.  
Setting up kubeadm (1.25.0-00) ...  
Processing triggers for man-db (2.10.2-1) ...  
Scanning processes...  
Scanning linux images...  
  
Running kernel seems to be up-to-date.  
  
No services need to be restarted.  
  
No containers need to be restarted.  
  
No user sessions are running outdated binaries.  
  
No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

Step 9: To set on hold

```
root@ip-172-31-7-165:~# sudo apt-mark hold kubeadm kubelet kubect1  
kubeadm set on hold.  
kubelet set on hold.  
kubect1 set on hold.  
root@ip-172-31-7-165:~#
```

Step 10: To check the Admin version

```
root@ip-172-31-7-165:~# kubeadm version  
kubeadm version: {version:Info{Major:"1", Minor:"25", GitVersion:"v1.25.0", GitCommit:"a866cbe2e5bbaa01cfd5e969aa3e033f3282a8a2", GitTreeState:"clean", BuildDate:"2022-08-23T17:43:25Z", GoVersion:"go1.19", Compiler:"gc", Platform:"linux/amd64"}}
```

**Conclusion:** From this experiment it is concluded that, we have learnt about the Kubernetes Cluster and its Architecture. Also, we have install and spin up the Kubernetes on the Linux Machine on Ubuntu and used the Cloud Platforms such as AWS. Hence, we have successfully achieved the Lab Outcome 1 and 2 (LO1 and LO2).