

St. Francis Institute of Technology, Mumbai-400 103
Department Of Information Technology

A.Y. 2023-2024

Class: TE-ITA/B, Semester: V

Subject: **Advanced DevOps Lab**

Experiment –5: To understand Kubernetes cluster architecture, install and spin up a Kubernetes cluster on a linux machine or cloud platform.

1. Aim: To understand Kubernetes Cluster Architecture and its installation.

2. Objectives: Aim of this experiment is that, the students will learn:

- Kubernetes concepts
- Installation of Kubernetes architecture.
- Creating instances of client server architecture on EC2
- Use kubectl to deploy resources into an EKS cluster
- Work with and configure commonly used Kubernetes resources

3. Lab objective mapped : ITL504.2: To deploy single and multiple container applications and manage application deployments with rollouts in Kubernetes

4. Prerequisite:

- Basic Linux command line administration
- Basic Kubernetes and Container-based concepts

5. Requirements: AWS account, browser, Personal Computer, Windows operating system, Internet Connection, AWS CLI, kubectl, Required IAM permissions,

6. Pre-Experiment Exercise:

Brief Theory:

Amazon EKS Amazon Elastic Kubernetes Service (Amazon EKS) is a managed service that you can use to run Kubernetes on AWS without needing to install, operate, and maintain your own Kubernetes control plane or nodes. Kubernetes is an open-source system for automating the deployment, scaling, and management of containerized applications. Amazon EKS:

Runs and scales the Kubernetes control plane across multiple AWS Availability Zones to ensure high availability.

Automatically scales control plane instances based on load, detects and replaces unhealthy control plane instances, and it provides automated version updates and patching for them.

Is integrated with many AWS services to provide scalability and security for your applications, including the following capabilities:

Amazon ECR for container images
Elastic Load Balancing for load distribution
IAM for authentication
Amazon VPC for isolation

How does Amazon EKS work?

Create an Amazon EKS cluster in the AWS Management Console or with the AWS CLI or one of the AWS SDKs.

Launch managed or self-managed Amazon EC2 nodes, or deploy your workloads to AWS Fargate.

When your cluster is ready, you can configure your favorite Kubernetes tools, such as kubectl, to communicate with your cluster.

Deploy and manage workloads on your Amazon EKS cluster the same way that you would with any other Kubernetes environment. You can also view information about your workloads using the AWS Management Console.

Amazon EKS pricing You pay \$0.10 per hour for each Amazon EKS cluster that you create. You can use a single EKS cluster to run multiple applications by taking advantage of Kubernetes namespaces and IAM security policies. You can run EKS on AWS using either Amazon Elastic Compute Cloud (Amazon EC2) or AWS Fargate, and on-premises using AWS Outposts.

1 Clusters x 0.10 USD per hour x 730 hours per month = 73.00 USD
EKS Total Cost (monthly): 73.00 USD

Kubectl is a command line tool that you use to communicate with the Kubernetes API server. The kubectl binary is available in many operating system package managers.

CLI – A command line tool for working with AWS services, including Amazon EKS.

Features:

1. Auto-scaling. Automatically scale containerized applications and their resources up or down based on usage.
2. Lifecycle management. Automate deployments and updates with the ability to Rollback to previous versions and Pause and continue a deployment.
3. Declarative model. Declare the desired state, and K8s work in the background to maintain that state and recover from any failures.
4. Resilience and self-healing. Auto placement, auto restart, auto replication and auto-scaling provide application self-healing.
5. Persistent storage. Ability to mount and add storage dynamically.
6. Load balancing. Kubernetes supports various internal and external load balancing options to address diverse needs.
7. DevSecOps support. DevSecOps is an advanced approach to security that simplifies and automates container operations across clouds, integrates security throughout the container lifecycle, and enables teams to deliver secure, high-quality software more quickly. Combining DevSecOps practices and Kubernetes improves developer productivity.

Kubernetes works with Amazon EC2, Azure Container Service, Rack space, GCE, IBM Software, and other clouds. And it works with bare-metal (using CoreOS), Docker, and vSphere.

Kubernetes is used by Google, Spotify, The New York Times, Pinterest,

Adidas, Tinder, Capital One, etc.

7. Laboratory Exercise :

Steps to install and spin up a Kubernetes cluster on Linux machine/cloud platforms. (attach SS) AWS

1. Launch two instances of Linux machine
2. Rename one of the instances as worker and keep the name of the other one as master. 3. Connect both instances using EC2 Instance Connect.
4. Switch to sudo user and get update

Commands:Both Master and Worker

1. sudo su
2. apt-get update
5. Install docker and get its version and status.

Commands:Both Master and Worker

1. apt-get install docker -y
2. docker --version
3. systemctl enable docker
4. systemctl status docker
6. Add repository.

Commands:Both Master and Worker

1. curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add
2. sudo apt-add-repository "deb http://apt.kubernetes.io/kubernetes-xenial main"
7. Install Kuber admin, set the hold on it and check its version

Commands:Both Master and Worker

1. sudo apt-get install kubeadm kubelet kubectl
2. sudo apt-mark hold kubeadm kubelet kubectl
3. kubeadm version

8. Post-Experiments Exercise

A. Extended Theory:

Nil

B. Questions:

Nil

C. Conclusion:(write in hand)

- 1. Write what was performed in the experiment
- 2. Mention few applications of what was studied.
- 3. Write the significance of the studied topic

C. References:

A. <https://kubernetes.io/case-studies/>

B. <https://cloudacademy.com/lab/eks-voteapp/>

Steps & Screenshots:

Step1: Create a new Instance

1. Goto EC2 console and click on Launch Instance

Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Instances (running)	0	Dedicated Hosts	0	Elastic IPs	0
Instances	0	Key pairs	6	Load balancers	0
Placement groups	0	Security groups	7	Snapshots	0
Volumes	0				

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Service health

Region: US East (N. Virginia)

Status: ✔ This service is operating normally

Account attributes

Supported platforms

- VPC

Default VPC

vpc-0326be2a8620a30ce

Settings

- EBS encryption
- Zones
- EC2 Serial Console
- Default credit specification
- Console experiments

Explore AWS

Enable Best Price-Performance with AWS Graviton2

AWS Graviton2 powered EC2 instances enable up to 40% better price performance for a broad spectrum of cloud workloads. [Learn more](#)

Get Up to 40% Better Price Performance

T4g instances deliver the best price performance for burstable general purpose workloads in Amazon EC2.

2. Give the name to your instance

Launch an instance

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags

Name: KA_Clusters

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 22.04 LTS, ...read more

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

3. Choose Ubuntu OS

Application and OS Images (Amazon Machine Image)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Quick Start

Search our full catalog including 1000s of application and OS images

Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type

ami-052efd3df9dad4825 (64-bit (x86)) / ami-070650c005c4e203 (64-bit (ARM))

Virtualization: hvm, ENA enabled: true, Root device type: ebs

Description

Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2022-06-09

Architecture: 64-bit (x86)

AMI ID: ami-052efd3df9dad4825

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 22.04 LTS, ...read more

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 1 million IOPS of EBS storage, and

Launch instance

4. Click on Create new key pair

Instance type Info

Instance type: t2.micro (Free tier eligible) [Compare instance types](#)

Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required: [Select](#) [Create new key pair](#)

Network settings Info [Edit](#)

Network: vpc-0326be2a8620a30ce

Subnet: No preference (Default subnet in any availability zone)

Auto-assign public IP

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 22.04 LTS, ...[read more](#)

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, ...[read more](#)

[Cancel](#) [Launch instance](#)

5. Give the Key pair name and select .ppk file format and then click on Create key pair

Create key pair [×](#)

Key pairs allow you to connect to your instance securely.

Enter the name of the key pair below. When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** [Learn more](#)

Key pair name:

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

☒ RSA
RSA encrypted private and public key pair

☐ ED25519
ED25519 encrypted private and public key pair (Not supported for Windows instances)

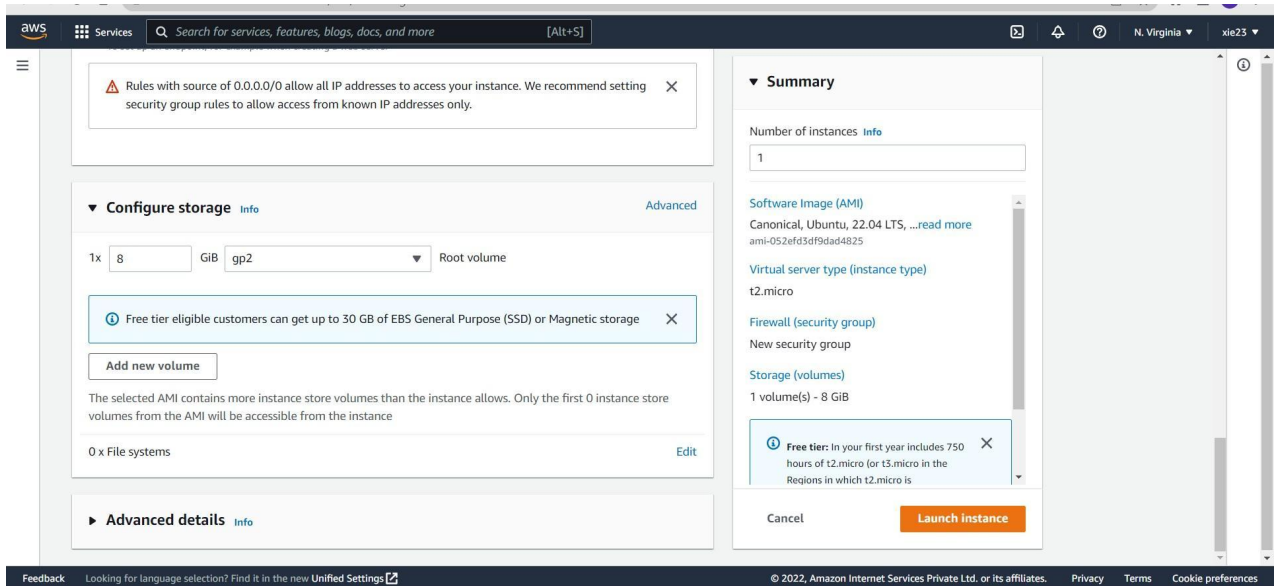
Private key file format

☐ .pem
For use with OpenSSH

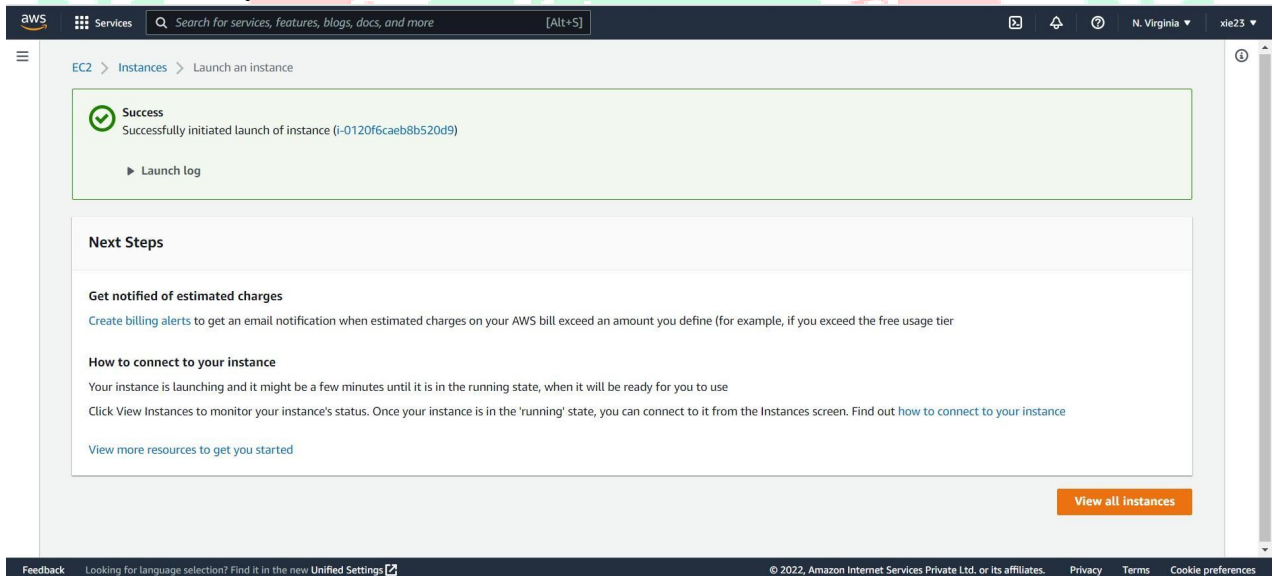
☒ .ppk
For use with PuTTY

[Cancel](#) [Create key pair](#)

6. Click on Launch Instance

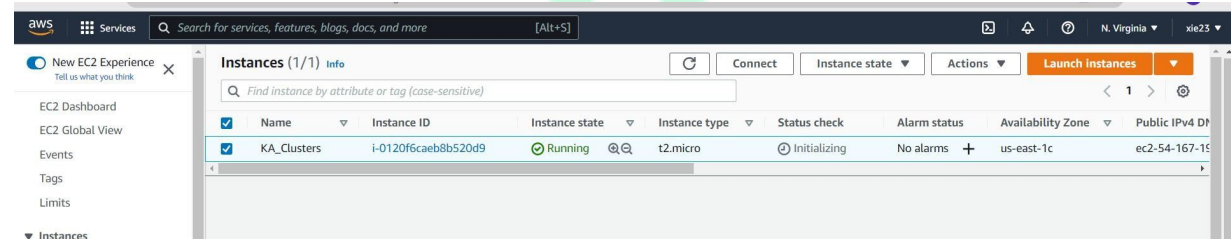


7. Successfully instance is launch



Step 2: Connection

1. Goto Instances and select the newly created instance and then click on connect

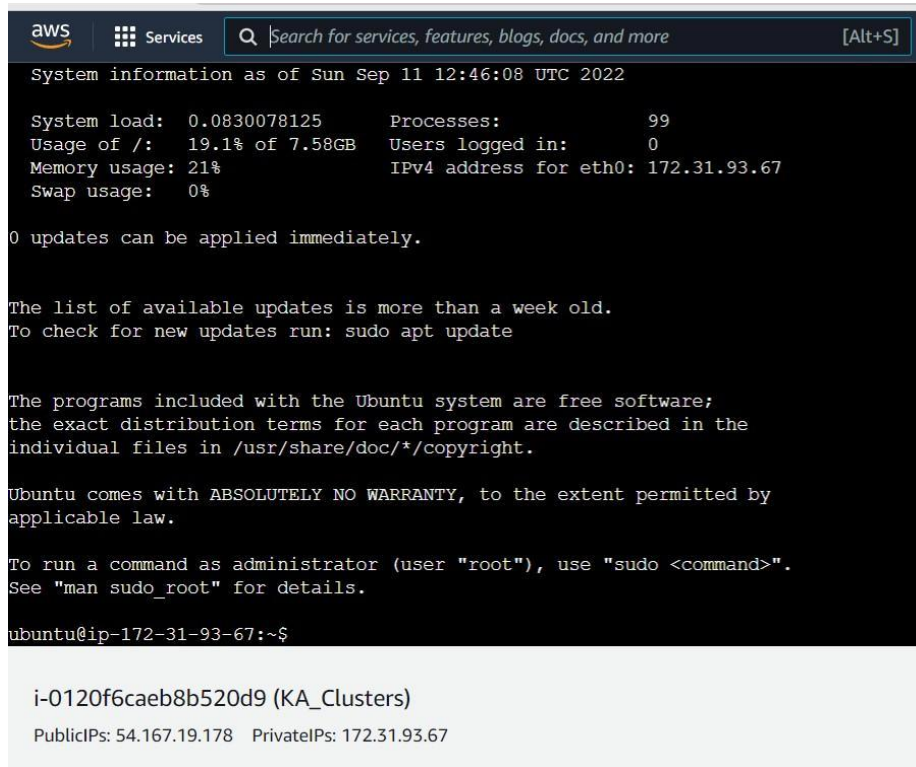


2. Click on Connect

The screenshot shows the AWS Management Console interface for connecting to an EC2 instance. The breadcrumb navigation at the top reads: EC2 > Instances > i-0120f6caeb8b520d9 > Connect to instance. The main heading is 'Connect to instance' with an 'Info' link. Below this, a subtitle states: 'Connect to your instance i-0120f6caeb8b520d9 (KA_Clusters) using any of these options'. There are four tabs: 'EC2 Instance Connect' (selected), 'Session Manager', 'SSH client', and 'EC2 serial console'. Under the 'EC2 Instance Connect' tab, the 'Instance ID' is 'i-0120f6caeb8b520d9 (KA_Clusters)' and the 'Public IP address' is '54.167.19.178'. The 'User name' field contains 'ubuntu'. A note below the field states: 'Connect using a custom user name, or use the default user name ubuntu for the AMI used to launch the instance.' A blue box contains a note: 'Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.' At the bottom right, there are 'Cancel' and 'Connect' buttons.

3. Establishing Connection

The screenshot shows the AWS Management Console interface during the connection establishment process. The top navigation bar includes the AWS logo, 'Services', a search bar, and the text '[Alt+S]'. The main content area is dark with a large white circular loading spinner in the center and the text 'Establishing Connection ...' below it. At the bottom, the instance details are displayed: 'i-0120f6caeb8b520d9 (KA_Clusters)' and 'Public IPs: 54.167.19.178 Private IPs: 172.31.93.67'. The footer contains a 'Feedback' link, a language selection prompt, and copyright information: '© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences'.



aws Services Search for services, features, blogs, docs, and more [Alt+S]

System information as of Sun Sep 11 12:46:08 UTC 2022

```

System load:  0.0830078125    Processes:           99
Usage of /:   19.1% of 7.58GB  Users logged in:     0
Memory usage: 21%            IPv4 address for eth0: 172.31.93.67
Swap usage:   0%

```

0 updates can be applied immediately.

The list of available updates is more than a week old.
To check for new updates run: `sudo apt update`

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in `/usr/share/doc/*/copyright`.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-93-67:~\$

i-0120f6caeb8b520d9 (KA_Clusters)

PublicIPs: 54.167.19.178 PrivateIPs: 172.31.93.67

Step 3: Run the command “`sudo su -`” to goto root

```
ubuntu@ip-172-31-93-67:~$ sudo su -
root@ip-172-31-93-67:~#
```

Step 4: Install all the updates

```

root@ip-172-31-93-67:~# sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [544 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [129 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [8168 B]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [306 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [47.5 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [524 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [254 kB]

```


Step 5: Install Docker and check its version

```

root@ip-172-31-93-67:~# apt install docker.io
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 78 not upgraded.
Need to get 65.6 MB of archives.
After this operation, 283 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63.6 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 bridge-utils amd64 1.7-1ubuntu3 [34.4 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 runc amd64 1.1.0-0ubuntu1 [4087 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 containerd amd64 1.5.9-0ubuntu3 [27.0 MB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 dns-root-data all 2021011101 [5256 B]

```

```

root@ip-172-31-93-67:~# docker --version
Docker version 20.10.12, build 20.10.12-0ubuntu4
root@ip-172-31-93-67:~#

```

Step 6: Enable Docker and then check docker status

```

root@ip-172-31-93-67:~# sudo systemctl enable docker
root@ip-172-31-93-67:~#

```

```

root@ip-172-31-93-67:~# 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 Sun 2022-09-11 12:53:49 UTC; 3min 37s ago
 TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 2549 (dockerd)
       Tasks: 7
      Memory: 38.0M
         CPU: 269ms
    CGroup: /system.slice/docker.service
            └─2549 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Sep 11 12:53:49 ip-172-31-93-67 dockerd[2549]: time="2022-09-11T12:53:49.356235070Z" level=
Sep 11 12:53:49 ip-172-31-93-67 dockerd[2549]: time="2022-09-11T12:53:49.356399659Z" level=
Sep 11 12:53:49 ip-172-31-93-67 dockerd[2549]: time="2022-09-11T12:53:49.356555442Z" level=
Sep 11 12:53:49 ip-172-31-93-67 dockerd[2549]: time="2022-09-11T12:53:49.426023858Z" level=
Sep 11 12:53:49 ip-172-31-93-67 dockerd[2549]: time="2022-09-11T12:53:49.705280711Z" level=
Sep 11 12:53:49 ip-172-31-93-67 dockerd[2549]: time="2022-09-11T12:53:49.799925454Z" level=
Sep 11 12:53:49 ip-172-31-93-67 dockerd[2549]: time="2022-09-11T12:53:49.880040777Z" level=
Sep 11 12:53:49 ip-172-31-93-67 dockerd[2549]: time="2022-09-11T12:53:49.880496977Z" level=
Sep 11 12:53:49 ip-172-31-93-67 systemd[1]: Started Docker Application Container Engine.
Sep 11 12:53:49 ip-172-31-93-67 dockerd[2549]: time="2022-09-11T12:53:49.919670674Z" level=
lines 1-22/22 (END)

```

Step 7 : Curl

```

root@ip-172-31-93-67:~# curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
root@ip-172-31-93-67:~#

```

Step 8: Add repository

```

root@ip-172-31-93-67:~# 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://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
Hit:5 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:4 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 282 kB in 1s (519 kB/s)
Reading package lists... Done
W: http://apt.kubernetes.io/dists/kubernetes-xenial/InRelease: Key is stored in legacy trusted.gpg keyring
for details.
root@ip-172-31-93-67:~#

```

Step 9: Install Kubeadm

```

root@ip-172-31-93-67:~# 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 78 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://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 conntrack amd64 1:1.4.6-2build2 [33.5 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 ebttables amd64 2.0.11-4build2 [84.9 kB]
Get:3 http://us-east-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 cri-tools amd64 1.24.2-00 [12.3 MB]
Get:5 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubernetes-cni amd64 0.8.7-00 [25.0 MB]
Get:6 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubelet amd64 1.25.0-00 [19.5 MB]
Get:7 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubect1 amd64 1.25.0-00 [9500 kB]
Get:8 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubeadm amd64 1.25.0-00 [9213 kB]
Fetched 75.9 MB in 2s (46.8 MB/s)
Selecting previously unselected package conntrack.
(Reading database ... 63966 files and directories currently installed.)

```

Step 10: Set on hold

```

root@ip-172-31-93-67:~# sudo apt-mark hold kubeadm kubelet kubect1
kubeadm set on hold.
kubelet set on hold.
kubect1 set on hold.
root@ip-172-31-93-67:~#

```

Step 11: Check kubeadm version

```

root@ip-172-31-93-67:~# kubeadm version
kubeadm version: &version.Info{Major:"1", Minor:"25", GitVersion:"v1.25.0", GitCommit:"
a866cbe2e5bbaa01cfd5e969aa3e033f3282a8a2", GitTreeState:"clean", BuildDate:"2022-08-23T
17:43:25Z", GoVersion:"go1.19", Compiler:"gc", Platform:"linux/amd64"}
root@ip-172-31-93-67:~#

```