

Name: Hitesh Tanwani

Class: D15B

Roll: 62

Experiment 4

Aim: To install Kubectl and execute Kubectl commands to manage the Kubernetes cluster and deploy Your First Kubernetes Application.

Theory:

Kubernetes, originally developed by Google, is an open-source container orchestration platform. It automates the deployment, scaling, and management of containerized applications, ensuring high availability and fault tolerance. Kubernetes is now the industry standard for container orchestration and is governed by the **Cloud Native Computing Foundation (CNCF)**, with contributions from major cloud and software providers like Google, AWS, Microsoft, IBM, Intel, Cisco, and Red Hat.

Kubernetes Deployment: Is a resource in Kubernetes that provides declarative updates for Pods and ReplicaSets. With a Deployment, you can define how many replicas of a pod should run, roll out new versions of an application, and roll back to previous versions if necessary. It ensures that the desired number of pod replicas are running at all times.

Necessary Requirements:

- **EC2 Instance:** The experiment required launching a t2.medium EC2 instance with 2 CPUs, as Kubernetes demands sufficient resources for effective functioning.
- **Minimum Requirements:**
 - **Instance Type:** t2.medium
 - **CPUs:** 2
 - **Memory:** Adequate for container orchestration.

This ensured that the Kubernetes cluster had the necessary resources to function smoothly.

Note:

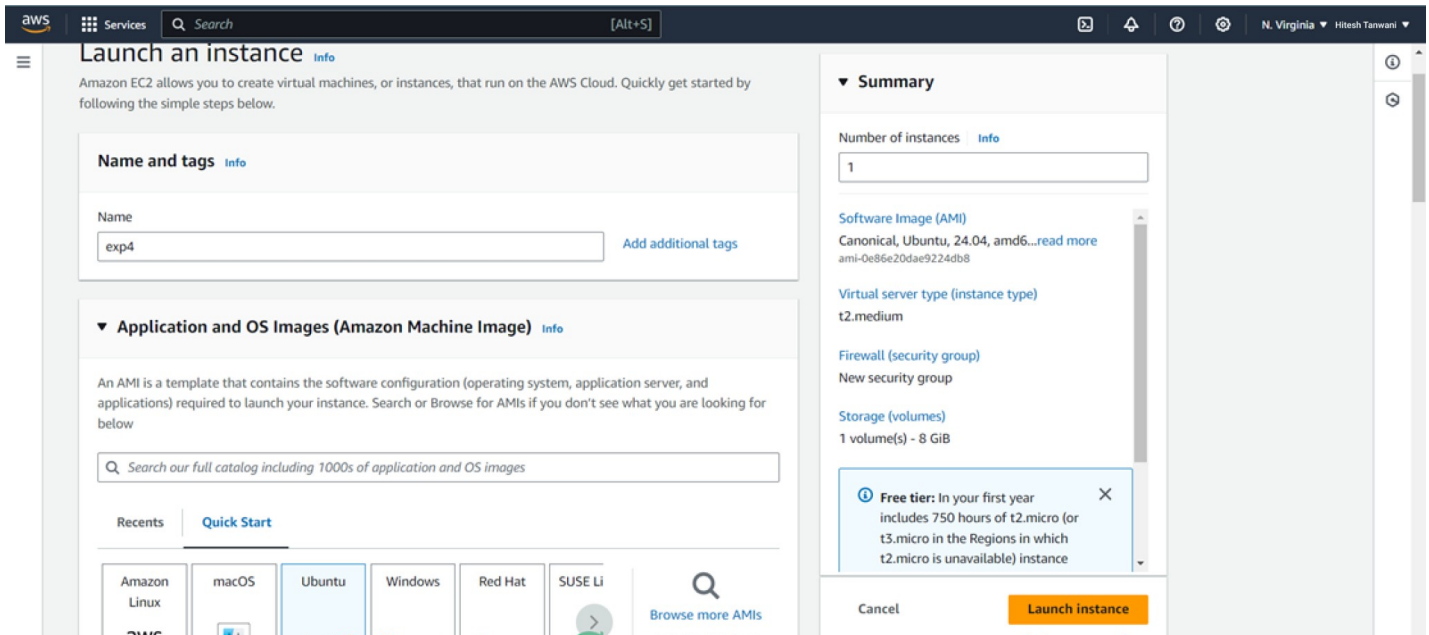
AWS Personal Account is preferred but we can also perform it on AWS Academy(adding some ignores in the command if any error occurs in below as the below experiment is performed on Personal Account).

If You are using AWS Academy Account Errors you will face in kubeadm init command so you have to add some ignores with this command.

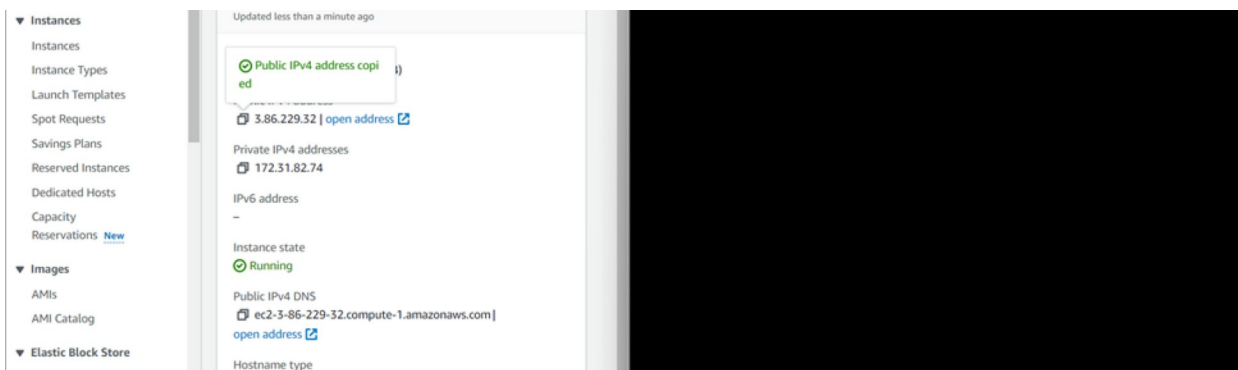
Step 1: Log in to your AWS Academy/personal account and launch a new Ec2 Instance. Select

Ubuntu as AMI and **t2.medium** as Instance Type, create a key of type RSA with .pem extension, and move the downloaded key to the new folder.

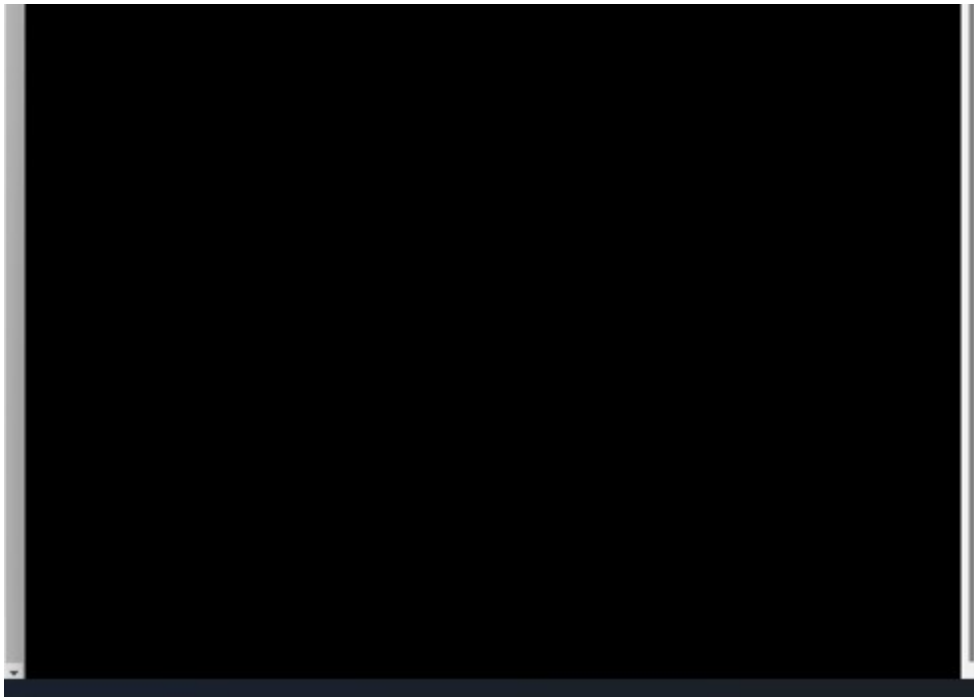
Note: A minimum of 2 CPUs are required so Please select t2.medium and do not forget to stop the instance after the experiment because it is not available in the free tier.



Step 2: After creating the instance click on Connect the instance and navigate to SSH Client.



Step 3: Now open the folder in the terminal where our .pem key is stored and paste the Example command (starting with ssh -i) in the terminal.(ssh -i "Master_Ec2_Key.pem" ubuntu@ec2-54-196-129-215.compute-1.amazonaws.com)



Step 4: Run the below commands to install and setup Docker.

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add  
- curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --  
dearmor -o /etc/apt/trusted.gpg.d/docker.gpg
```

```
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu  
$(lsb_release -cs) stable"
```

```
root@ip-172-31-82-74: /home/ubuntu
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [378 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [82.0 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [4528 B]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [271 kB]
Get:14 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [115 kB]
Get:15 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]
Get:16 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [10.1 kB]
Get:17 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [353 kB]
Get:18 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [68.1 kB]
Get:19 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [428 B]
Get:20 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.9 kB]
Get:21 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2808 B]
Get:22 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:23 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [344 B]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [531 kB]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [129 kB]
Get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [8600 B]
Get:32 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [374 kB]
Get:33 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [154 kB]
Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:35 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [14.6 kB]
Get:36 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [353 kB]
Get:37 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [68.1 kB]
Get:38 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c-n-f Metadata [424 B]
Get:39 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [14.4 kB]
Get:40 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3608 B]
Get:41 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [212 B]
Get:42 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [532 B]
Get:43 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:44 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
Get:45 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [10.6 kB]
Get:46 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.8 kB]
Get:47 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [17.6 kB]
Get:48 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1104 B]
Get:49 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:50 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:51 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:52 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
Fetched 29.1 MB in 4s (7907 kB/s)
Reading package lists... Done
root@ip-172-31-82-74: /home/ubuntu# |
```

sudo apt-get update

sudo apt-get install -y docker-ce

```
root@ip-172-31-82-74: /home/ubuntu
Preparing to unpack .../1-containerd.io_1.7.22-1_amd64.deb ...
Unpacking containerd.io (1.7.22-1) ...
Selecting previously unselected package docker-buildx-plugin.
Preparing to unpack .../2-docker-buildx-plugin_0.17.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-buildx-plugin (0.17.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce-cli.
Preparing to unpack .../3-docker-ce-cli_5%3a27.3.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce-cli (5:27.3.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce.
Preparing to unpack .../4-docker-ce_5%3a27.3.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce (5:27.3.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce-rootless-extras.
Preparing to unpack .../5-docker-ce-rootless-extras_5%3a27.3.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce-rootless-extras (5:27.3.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-compose-plugin.
Preparing to unpack .../6-docker-compose-plugin_2.29.7-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-compose-plugin (2.29.7-1~ubuntu.24.04~noble) ...
Selecting previously unselected package libltdl7:amd64.
Preparing to unpack .../7-libltdl7_2.4.7-7build1_amd64.deb ...
Unpacking libltdl7:amd64 (2.4.7-7build1) ...
Selecting previously unselected package libslirp0:amd64.
Preparing to unpack .../8-libslirp0_4.7.0-1ubuntu3_amd64.deb ...
Unpacking libslirp0:amd64 (4.7.0-1ubuntu3) ...
Selecting previously unselected package slirp4netns.
Preparing to unpack .../9-slirp4netns_1.2.1-1build2_amd64.deb ...
Unpacking slirp4netns (1.2.1-1build2) ...
Setting up docker-buildx-plugin (0.17.1-1~ubuntu.24.04~noble) ...
Setting up containerd.io (1.7.22-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.service.
Setting up docker-compose-plugin (2.29.7-1~ubuntu.24.04~noble) ...
Setting up libltdl7:amd64 (2.4.7-7build1) ...
Setting up docker-ce-cli (5:27.3.1-1~ubuntu.24.04~noble) ...
Setting up libslirp0:amd64 (4.7.0-1ubuntu3) ...
Setting up pigz (2.8-1) ...
Setting up docker-ce-rootless-extras (5:27.3.1-1~ubuntu.24.04~noble) ...
Setting up slirp4netns (1.2.1-1build2) ...
Setting up docker-ce (5:27.3.1-1~ubuntu.24.04~noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.2) ...
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.
```

```
sudo mkdir -p /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}
```

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-82-74:/home/ubuntu# sudo mkdir -p /etc/docker
root@ip-172-31-82-74:/home/ubuntu# cat <<EOF | sudo tee /etc/docker/daemon.json
{
"exec-opts": ["native.cgroupdriver=systemd"]
}
EOF
{
"exec-opts": ["native.cgroupdriver=systemd"]
}
root@ip-172-31-82-74:/home/ubuntu# |
```

sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker

```
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
root@ip-172-31-82-74:/home/ubuntu# sudo systemctl daemon-reload
root@ip-172-31-82-74:/home/ubuntu# sudo systemctl restart docker
root@ip-172-31-82-74:/home/ubuntu# |
```

Step 5: Run the below command to install Kubernetes.

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

```
root@ip-172-31-82-74:/home/ubuntu# curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key |
sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
root@ip-172-31-82-74:/home/ubuntu# |
```

sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl


```
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /
root@ip-172-31-82-74:/home/ubuntu# sudo apt-get update
E: Malformed entry 1 in list file /etc/apt/sources.list.d/kubernetes.list (URI)
E: The list of sources could not be read.
root@ip-172-31-82-74:/home/ubuntu# ^C
root@ip-172-31-82-74:/home/ubuntu# deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://
/pkgs.k8s.io/core:/stable:/v1.31/deb/ /
Command 'deb' not found, did you mean:
  command 'den' from snap den (1.2.0-0)
  command 'dub' from snap dub (1.19.0)
  command 'dab' from deb bsdgames (2.17-30)
  command 'dub' from deb dub (1.34.0-1)
  command 'edb' from deb edb-debugger (1.3.0-2.1)
  command 'debi' from deb devscripts (2.23.7)
  command 'debc' from deb devscripts (2.23.7)
  command 'dex' from deb dex (0.9.0-2)
  command 'deb3' from deb quilt (0.67+really0.67-4)
  command 'dcb' from deb iproute2 (6.1.0-1ubuntu2)
  command 'derb' from deb icu-devtools (74.2-1ubuntu3.1)
See 'snap info <snapname>' for additional versions.
root@ip-172-31-82-74:/home/ubuntu# sudo nano /etc/apt/sources.list.d/kubernetes.list
root@ip-172-31-82-74:/home/ubuntu# sudo apt-get update
E: Malformed entry 1 in list file /etc/apt/sources.list.d/kubernetes.list (URI)
E: The list of sources could not be read.
root@ip-172-31-82-74:/home/ubuntu# sudo nano /etc/apt/sources.list.d/kubernetes.list
root@ip-172-31-82-74:/home/ubuntu# sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu noble InRelease
Hit:6 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb InRelease
[1186 B]
Get:7 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb Packages
[4865 B]
Fetched 6051 B in 1s (11.5 kB/s)
Reading package lists... Done
root@ip-172-31-82-74:/home/ubuntu#
```

```
root@ip-172-31-82-74: /home/ubuntu
The following NEW packages will be installed:
  conntrack cri-tools kubeadm kubectl kubelet kubernetes-cni
0 upgraded, 6 newly installed, 0 to remove and 142 not upgraded.
Need to get 87.4 MB of archives.
After this operation, 314 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 conntrack amd64 1:1.4.8-1ubuntu1 [37.9 kB]
Get:2 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb cri-tools 1.31.1-1.1 [15.7 MB]
Get:3 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb kubeadm 1.31.1-1.1 [11.4 MB]
Get:4 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb kubectl 1.31.1-1.1 [11.2 MB]
Get:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb kubernetes-cni 1.5.1-1.1 [33.9 MB]
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb kubelet 1.31.1-1.1 [15.2 MB]
Fetched 87.4 MB in 1s (85.4 MB/s)
Selecting previously unselected package conntrack.
(Reading database ... 68007 files and directories currently installed.)
Preparing to unpack .../0-conntrack_1%3a1.4.8-1ubuntu1_amd64.deb ...
Unpacking conntrack (1:1.4.8-1ubuntu1) ...
Selecting previously unselected package cri-tools.
Preparing to unpack .../1-cri-tools_1.31.1-1.1_amd64.deb ...
Unpacking cri-tools (1.31.1-1.1) ...
Selecting previously unselected package kubeadm.
Preparing to unpack .../2-kubeadm_1.31.1-1.1_amd64.deb ...
Unpacking kubeadm (1.31.1-1.1) ...
Selecting previously unselected package kubectl.
Preparing to unpack .../3-kubectl_1.31.1-1.1_amd64.deb ...
Unpacking kubectl (1.31.1-1.1) ...
Selecting previously unselected package kubernetes-cni.
Preparing to unpack .../4-kubernetes-cni_1.5.1-1.1_amd64.deb ...
Unpacking kubernetes-cni (1.5.1-1.1) ...
Selecting previously unselected package kubelet.
Preparing to unpack .../5-kubelet_1.31.1-1.1_amd64.deb ...
Unpacking kubelet (1.31.1-1.1) ...
Setting up conntrack (1:1.4.8-1ubuntu1) ...
Setting up kubectl (1.31.1-1.1) ...
Setting up cri-tools (1.31.1-1.1) ...
Setting up kubernetes-cni (1.5.1-1.1) ...
Setting up kubeadm (1.31.1-1.1) ...
Setting up kubelet (1.31.1-1.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
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.
root@ip-172-31-82-74: /home/ubuntu#
```

```
sudo systemctl enable --now kubelet
sudo kubeadm init --pod-network-cidr=10.244.0.0/16
```



```

root@ip-172-31-82-74:/home/ubuntu# sudo apt-mark hold kubelet kubeadm kubectl
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
root@ip-172-31-82-74:/home/ubuntu# sudo systemctl enable --now kubelet
root@ip-172-31-82-74:/home/ubuntu# sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
W0926 08:45:24.774275 4285 checks.go:1080] [preflight] WARNING: Couldn't create the interface used for
or talking to the container runtime: failed to create new CRI runtime service: validate service connect
ion: validate CRI v1 runtime API for endpoint "unix:///var/run/containerd/containerd.sock": rpc error:
code = Unimplemented desc = unknown service runtime.v1.RuntimeService
[WARNING FileExisting-socat]: socat not found in system path
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
error execution phase preflight: [preflight] Some fatal errors occurred:
failed to create new CRI runtime service: validate service connection: validate CRI v1 runtime API for
endpoint "unix:///var/run/containerd/containerd.sock": rpc error: code = Unimplemented desc = unknown s
ervice runtime.v1.RuntimeService[preflight] If you know what you are doing, you can make a check non-fa
tal with '--ignore-preflight-errors=...'
To see the stack trace of this error execute with --v=5 or higher
root@ip-172-31-82-74:/home/ubuntu#

```

Now We have got an error.

So we have to perform some additional commands as follow.

sudo apt-get install -y containerd

```

The following NEW packages will be installed:
  containerd runc
0 upgraded, 2 newly installed, 2 to remove and 142 not upgraded.
Need to get 47.2 MB of archives.
After this operation, 53.1 MB disk space will be freed.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.1.12-0ubuntu
3.1 [8599 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.12-0
ubuntu4.1 [38.6 MB]
Fetched 47.2 MB in 1s (78.9 MB/s)
(Reading database ... 68064 files and directories currently installed.)
Removing docker-ce (5:27.3.1-1~ubuntu.24.04~noble) ...
Removing containerd.io (1.7.22-1) ...
Selecting previously unselected package runc.
(Reading database ... 68044 files and directories currently installed.)
Preparing to unpack .../runc_1.1.12-0ubuntu3.1_amd64.deb ...
Unpacking runc (1.1.12-0ubuntu3.1) ...
Selecting previously unselected package containerd.
Preparing to unpack .../containerd_1.7.12-0ubuntu4.1_amd64.deb ...
Unpacking containerd (1.7.12-0ubuntu4.1) ...
Setting up runc (1.1.12-0ubuntu3.1) ...
Setting up containerd (1.7.12-0ubuntu4.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
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.
root@ip-172-31-82-74:/home/ubuntu#

```

sudo mkdir -p /etc/containerd

sudo containerd config default | sudo tee /etc/containerd/config.toml

```
root@ip-172-31-82-74:/home/ubuntu# sudo mkdir -p /etc/containerd
sudo containerd config default | sudo tee /etc/containerd/config.toml
disabled_plugins = []
imports = []
oom_score = 0
plugin_dir = ""
required_plugins = []
root = "/var/lib/containerd"
state = "/run/containerd"
temp = ""
version = 2

[cgroup]
  path = ""

[debug]
  address = ""
  format = ""
  gid = 0
  level = ""
  uid = 0

[grpc]
  address = "/run/containerd/containerd.sock"
  gid = 0
  max_recv_message_size = 16777216
  max_send_message_size = 16777216
  tcp_address = ""
  tcp_tls_ca = ""
  tcp_tls_cert = ""
  tcp_tls_key = ""
  uid = 0

[metrics]
  address = ""
  grpc_histogram = false

[plugins]

[plugins."io.containerd.gc.v1.scheduler"]
  deletion_threshold = 0
  mutation_threshold = 100
  pause_threshold = 0.02
  schedule_delay = "0s"
  startup_delay = "100ms"

[plugins."io.containerd.grpc.v1.cri"]
  cdi_spec_dirs = ["/etc/cdi", "/var/run/cdi"]
  device_ownership_from_security_context = false
  disable_apparmor = false
  disable_cgroup = false
  disable_hugetlb_controller = true
  disable_proc_mount = false
  disable_tcp_service = true
  drain_exec_sync_io_timeout = "0s"
```

...

sudo systemctl restart containerd

sudo systemctl enable containerd

sudo systemctl status containerd

```

Tasks: 7
Memory: 13.8M (peak: 14.3M)
CPU: 111ms
CGroup: /system.slice/containerd.service
└─4682 /usr/bin/containerd

Sep 26 08:47:02 ip-172-31-82-74 containerd[4682]: time="2024-09-26T08:47:02.292205693Z" level=info msg>
Sep 26 08:47:02 ip-172-31-82-74 containerd[4682]: time="2024-09-26T08:47:02.292247111Z" level=info msg>
Sep 26 08:47:02 ip-172-31-82-74 containerd[4682]: time="2024-09-26T08:47:02.292297409Z" level=info msg>
Sep 26 08:47:02 ip-172-31-82-74 containerd[4682]: time="2024-09-26T08:47:02.292314172Z" level=info msg>
Sep 26 08:47:02 ip-172-31-82-74 containerd[4682]: time="2024-09-26T08:47:02.292322854Z" level=info msg>
Sep 26 08:47:02 ip-172-31-82-74 containerd[4682]: time="2024-09-26T08:47:02.292329036Z" level=info msg>
Sep 26 08:47:02 ip-172-31-82-74 containerd[4682]: time="2024-09-26T08:47:02.292225859Z" level=info msg>
Sep 26 08:47:02 ip-172-31-82-74 containerd[4682]: time="2024-09-26T08:47:02.292401004Z" level=info msg>
Sep 26 08:47:02 ip-172-31-82-74 containerd[4682]: time="2024-09-26T08:47:02.292461784Z" level=info msg>
Sep 26 08:47:02 ip-172-31-82-74 systemd[1]: Started containerd.service - containerd container runtime.
lines 1-21/21 (END)

```

sudo apt-get install -y socat

```

root@ip-172-31-82-74:/home/ubuntu# sudo apt-get install -y socat
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7
  libslirp0 pigz slirp4netns
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  socat
0 upgraded, 1 newly installed, 0 to remove and 142 not upgraded.
Need to get 374 kB of archives.
After this operation, 1649 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 socat amd64 1.8.0.0-4build3 [374
kB]
Fetched 374 kB in 0s (16.8 MB/s)
Selecting previously unselected package socat.
(Reading database ... 68108 files and directories currently installed.)
Preparing to unpack .../socat_1.8.0.0-4build3_amd64.deb ...
Unpacking socat (1.8.0.0-4build3) ...
Setting up socat (1.8.0.0-4build3) ...
Processing triggers for man-db (2.12.0-4build2) ...
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.
root@ip-172-31-82-74:/home/ubuntu#

```

Step 6: Initialize the Kubecluster

sudo kubeadm init --pod-network-cidr=10.244.0.0/16

```

[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [ip-172-31-82-74 localhost] and IPs [172.31.82.74 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-82-74 localhost] and IPs [172.31.82.74 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
[kubeconfig] Writing "admin.conf" kubeconfig file
[kubeconfig] Writing "super-admin.conf" kubeconfig file
[kubeconfig] Writing "kubelet.conf" kubeconfig file
[kubeconfig] Writing "controller-manager.conf" kubeconfig file
[kubeconfig] Writing "scheduler.conf" kubeconfig file
[etcd] Creating static Pod manifest for local etcd in "/etc/kubernetes/manifests"
[control-plane] Using manifest folder "/etc/kubernetes/manifests"
[control-plane] Creating static Pod manifest for "kube-apiserver"
[control-plane] Creating static Pod manifest for "kube-controller-manager"
[control-plane] Creating static Pod manifest for "kube-scheduler"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Starting the kubelet

```

```

root@ip-172-31-82-74:/home/ubuntu# mkdir -p $HOME/.kube
root@ip-172-31-82-74:/home/ubuntu# sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
root@ip-172-31-82-74:/home/ubuntu# sudo chown $(id -u):$(id -g) $HOME/.kube/config
root@ip-172-31-82-74:/home/ubuntu# |

```

Copy the mkdir and chown commands from the top and execute them.

```

mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config

```

```

root@ip-172-31-82-74:/home/ubuntu# mkdir -p $HOME/.kube
root@ip-172-31-82-74:/home/ubuntu# sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
root@ip-172-31-82-74:/home/ubuntu# sudo chown $(id -u):$(id -g) $HOME/.kube/config
root@ip-172-31-82-74:/home/ubuntu# kubectl apply -f
https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
error: flag needs an argument: 'f' in -f
See 'kubectl apply --help' for usage.
bash: https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml: No such file or directory
root@ip-172-31-82-74:/home/ubuntu# kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
root@ip-172-31-82-74:/home/ubuntu#

```

Add a common networking plugin called flannel as mentioned in the code.

```

sudo nano /etc/apt/sources.list.d/kubernetes.list

```

```

deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]

```

<https://pkgs.k8s.io/core:/stable:/v1.31/deb/>

kubectl apply -f

<https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml>

Step 7: Now that the cluster is up and running, we can deploy our nginx server on this cluster. Apply this deployment file using this command to create a deployment

kubectl apply -f <https://k8s.io/examples/application/deployment.yaml>

```
root@ip-172-31-82-74:/home/ubuntu# kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
root@ip-172-31-82-74:/home/ubuntu#
```

kubectl get pods

```
root@ip-172-31-82-74:/home/ubuntu# kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
root@ip-172-31-82-74:/home/ubuntu# kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-g4c1n    0/1     Pending   0           17s
nginx-deployment-d556bf558-z19p4    0/1     Pending   0           17s
root@ip-172-31-82-74:/home/ubuntu#
```

POD_NAME=\$(kubectl get pods -l app=nginx -o jsonpath="{.items[0].metadata.name}")

kubectl port-forward \$POD_NAME 8080:80

```
root@ip-172-31-82-74:/home/ubuntu# kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
root@ip-172-31-82-74:/home/ubuntu# kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
root@ip-172-31-82-74:/home/ubuntu# kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-g4c1n    0/1     Pending   0           17s
nginx-deployment-d556bf558-z19p4    0/1     Pending   0           17s
root@ip-172-31-82-74:/home/ubuntu# POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath="{.items[0].metadata.name}")
root@ip-172-31-82-74:/home/ubuntu# kubectl port-forward $POD_NAME 8080:80
error: unable to forward port because pod is not running. Current status=Pending
root@ip-172-31-82-74:/home/ubuntu#
```

Note : We have faced an error as pod status is pending so make it running run below commands then again run above 2 commands.

<http://node-role.kubernetes.io/control-plane->

kubectl get nodes

```
root@ip-172-31-82-74:/home/ubuntu# ^C
root@ip-172-31-82-74:/home/ubuntu# kubectl taint nodes --all node-role.kubernetes.io/control-plane-
node/ip-172-31-82-74 untainted
root@ip-172-31-82-74:/home/ubuntu#
```

kubectl get pods


```
root@ip-172-31-82-74:/home/ubuntu# kubectl get nodes
NAME                STATUS    ROLES    AGE     VERSION
ip-172-31-82-74    Ready    control-plane  5m24s   v1.31.1
root@ip-172-31-82-74:/home/ubuntu# |
```

POD_NAME=\$(kubectl get pods -l app=nginx -o jsonpath="{.items[0].metadata.name}")

kubectl port-forward \$POD_NAME 8080:80

```
root@ip-172-31-82-74:/home/ubuntu# kubectl get nodes
NAME                STATUS    ROLES    AGE     VERSION
ip-172-31-82-74    Ready    control-plane  5m24s   v1.31.1
root@ip-172-31-82-74:/home/ubuntu# kubectl get pods
NAME                READY    STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-g4c1n  1/1      Running    0           3m27s
nginx-deployment-d556bf558-z19p4  1/1      Running    0           3m27s
root@ip-172-31-82-74:/home/ubuntu# POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath="{.items[0].metadata.name}")
root@ip-172-31-82-74:/home/ubuntu# kubectl port-forward $POD_NAME 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
```

Step 8: Verify your deployment

Open up a new terminal and ssh to your EC2 instance.

Then, use this curl command to check if the Nginx server is running.

curl --head http://127.0.0.1:8080

```
root@ip-172-31-82-74:/home/ubuntu# kubectl get nodes
NAME                STATUS    ROLES    AGE     VERSION
ip-172-31-82-74    Ready    control-plane  5m24s   v1.31.1
root@ip-172-31-82-74:/home/ubuntu# kubectl get pods
NAME                READY    STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-g4c1n  1/1      Running    0           3m27s
nginx-deployment-d556bf558-z19p4  1/1      Running    0           3m27s
root@ip-172-31-82-74:/home/ubuntu# POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath="{.items[0].metadata.name}")
root@ip-172-31-82-74:/home/ubuntu# kubectl port-forward $POD_NAME 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
```

The screenshot displays two terminal windows. The left window, titled 'ubuntu@ip-172-31-82-74: ~', shows an SSH session initiated from a Windows machine (MINGW64). The user runs 'ssh -i "exp4.pem" ubuntu@83.86.229.32', and the terminal shows the Ubuntu 24.04 LTS login banner and system information. A 'curl --head http://127.0.0.1:8080' command is executed, returning a 200 OK status with various headers. The right window, titled 'root@ip-172-31-82-74: /home/ubuntu', shows the installation of Kubernetes components using 'kubeadm init', 'kubeadm join', and 'kubectl apply'. It then shows the deployment of an Nginx pod with 'kubectl apply -f https://k8s.io/examples/application/deployment.yaml'. Finally, it shows the port-forwarding command 'kubectl port-forward \$POD_NAME 8080:80' and the execution of 'curl --head http://127.0.0.1:8080' which returns a 200 OK status.

If the response is 200 OK and you can see the Nginx server name, your deployment was

successful.

We have successfully deployed our Nginx server on our EC2 instance.

Conclusion:

In this experiment, we successfully installed Kubernetes on an EC2 instance and deployed an Nginx server using Kubectl commands. During the process, we encountered two main errors: the Kubernetes pod was initially in a pending state, which was resolved by removing the control-plane taint using `kubectl taint nodes --all`, and we also faced an issue with the missing `containerd` runtime, which was fixed by installing and starting containerd. We used a **t2.medium EC2 instance with 2 CPUs** to meet the necessary resource requirements for the Kubernetes setup and deployment.