Name:Atharva Shinde Roll No.-54 Div:D15C

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

Theory

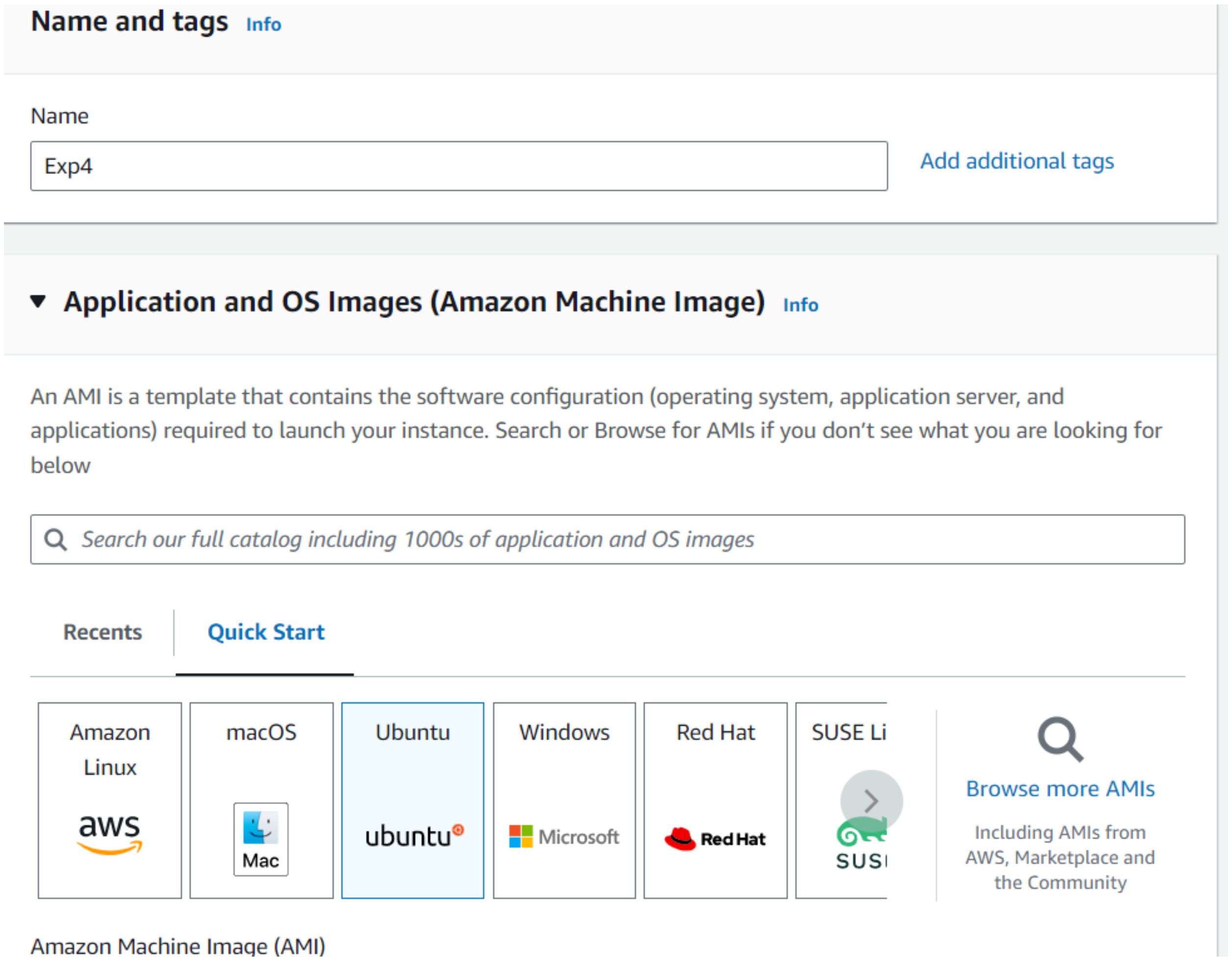
`kubectl` is the command-line tool used to interact with Kubernetes clusters. It serves as the primary interface for managing and orchestrating containers in a Kubernetes environment. By sending commands to the Kubernetes API server, `kubectl` allows you to control clusters, manage workloads, and inspect resource states.

To begin using Kubernetes, installing `kubectl` is essential. The installation process varies based on the operating system (Linux, Windows, or macOS). After installing, `kubectl` connects to the Kubernetes cluster using the `kubeconfig` file, which stores details like cluster name, server address, and access credentials. With this connection established, you can use `kubectl` to perform a variety of operations, such as creating, updating, scaling, and deleting applications.

When deploying your first application with Kubernetes, the process involves defining the application in a configuration file (usually YAML) that specifies its requirements, such as images, replicas, and networking settings. `kubectl` interprets this configuration and relays it to the Kubernetes cluster, which then manages the lifecycle of the application.

Name:Atharva Shinde Roll No.-54 Div:D15C

Step 1:Create an EC2 instance use ubuntu application and select t2 .medium category in instance type create a new key rsa type save it in local machine in an folder:



Name and tags Info

Name

Exp4

Add additional tags

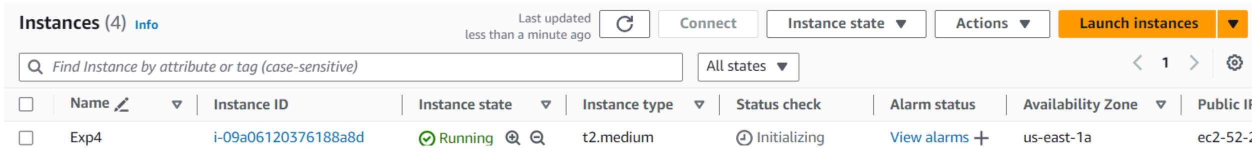
Application and OS Images (Amazon Machine Image) Info

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

Search our full catalog including 1000s of application and OS images
Recents
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Li
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Step 2:Click create to create the instance:



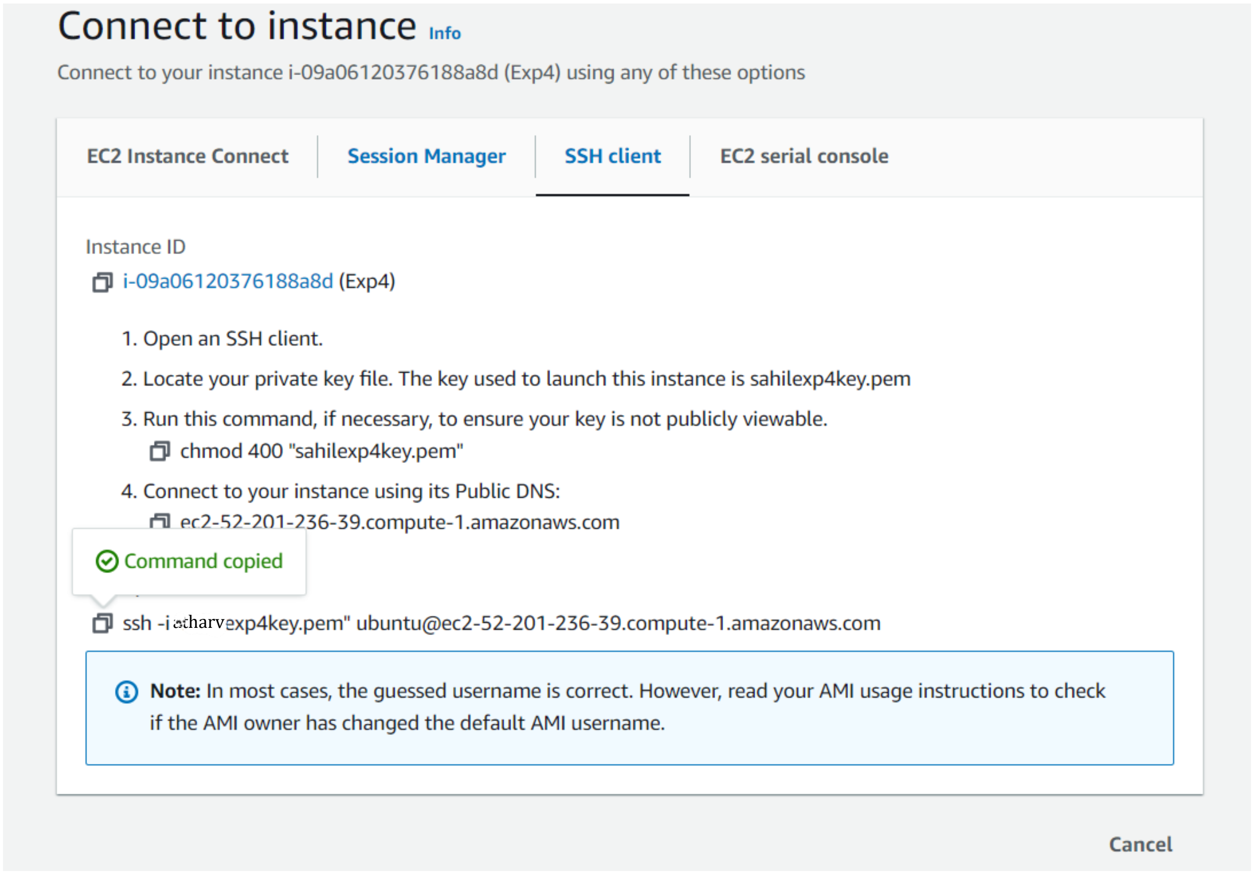
Instances (4) Info

Last updated less than a minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
1

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
Exp4	i-09a06120376188a8d	Running	t2.medium	Initializing	View alarms	us-east-1a	ec2-52-

Name:Atharva Shinde Roll No.-54 Div:D15C

Step 3:
Navigate to ssh client copy the key:



Connect to instance Info

Connect to your instance i-09a06120376188a8d (Exp4) using any of these options

Instance ID

i-09a06120376188a8d (Exp4)

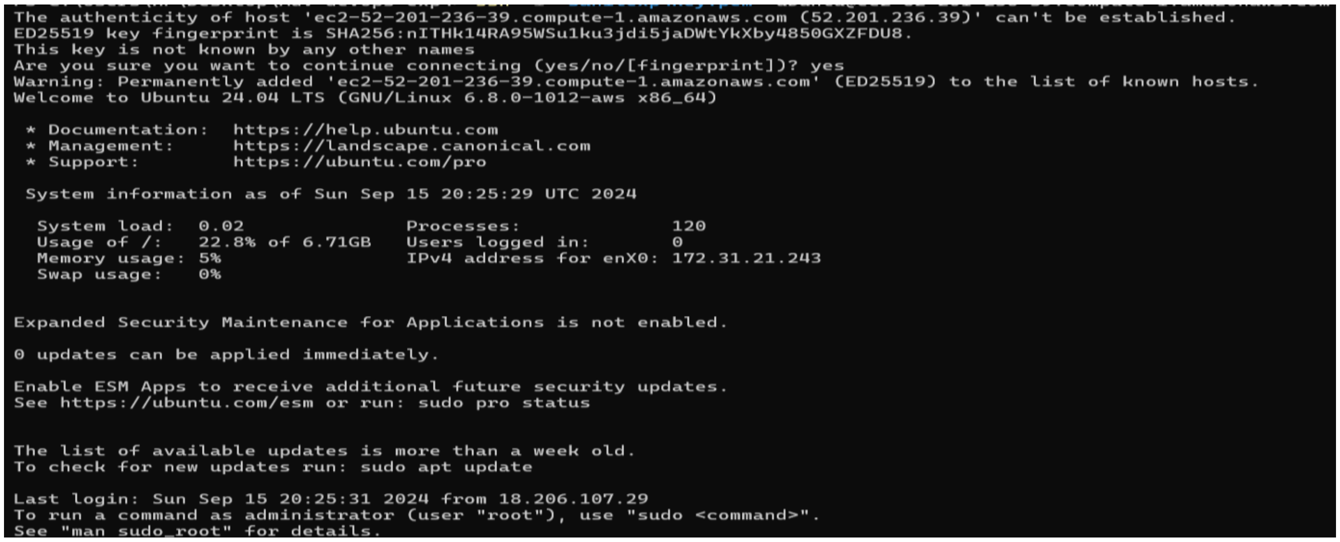
1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is sahilexp4key.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   chmod 400 "sahilexp4key.pem"
4. Connect to your instance using its Public DNS:
   ec2-52-201-236-39.compute-1.amazonaws.com

Command copied

ssh -i atharvexp4key.pem" ubuntu@ec2-52-201-236-39.compute-1.amazonaws.com

Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Step 4:navigate to the folder open the terminal and paste the ssh command:
ssh -i "atharvexp4key.pem" ubuntu@ec2-52-201-236-39.compute-1.amazonaws.com

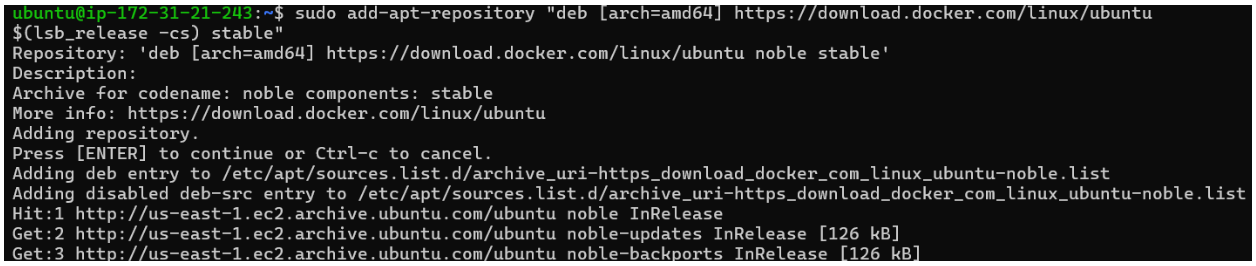


```
The authenticity of host 'ec2-52-201-236-39.compute-1.amazonaws.com (52.201.236.39)' can't be established.
ED25519 key fingerprint is SHA256:nITHk14RA95WSu1ku3jdi5jaDWtYkXby4850GXZFDU8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-52-201-236-39.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Sun Sep 15 20:25:29 UTC 2024

  System load:  0.02               Processes:             120
  Usage of /:   22.8% of 6.71GB    Users logged in:       0
  Memory usage: 5%                 IPv4 address for enX0: 172.31.21.243
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


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

Last login: Sun Sep 15 20:25:31 2024 from 18.206.107.29
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

Step 5:Install docker

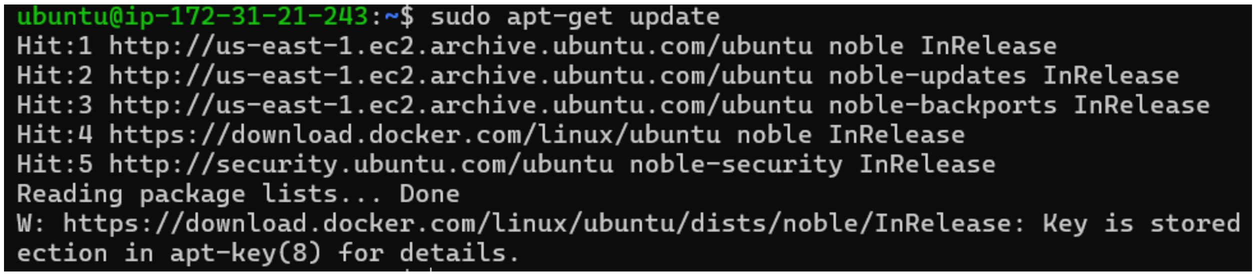
Name:Atharva Shinde Roll No.-54 Div:D15C

Use the commands given below to install docker
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add - curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee /etc/apt/trusted.gpg.d/docker.gpg > /dev/null sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"



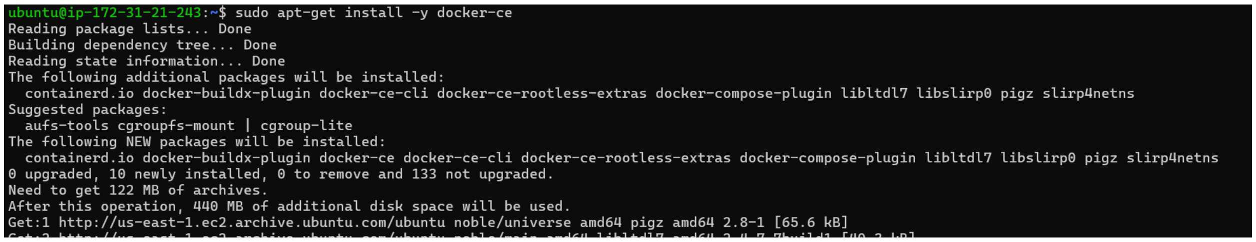
```
ubuntu@ip-172-31-21-243:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable"
Repository: 'deb [arch=amd64] https://download.docker.com/linux/ubuntu noble stable'
Description:
Archive for codename: noble components: stable
More info: https://download.docker.com/linux/ubuntu
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
```

Use:
sudo apt-get update



```
ubuntu@ip-172-31-21-243:~$ 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:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored
ection in apt-key(8) for details.
```

Use:
sudo apt-get install -y docker-ce

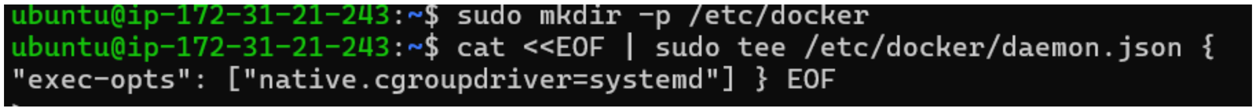


```
ubuntu@ip-172-31-21-243:~$ sudo apt-get install -y docker-ce
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz slirp4netns
0 upgraded, 10 newly installed, 0 to remove and 133 not upgraded.
Need to get 122 MB of archives.
After this operation, 440 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
```

Now the docker is installed;

Now lets enable the docker:

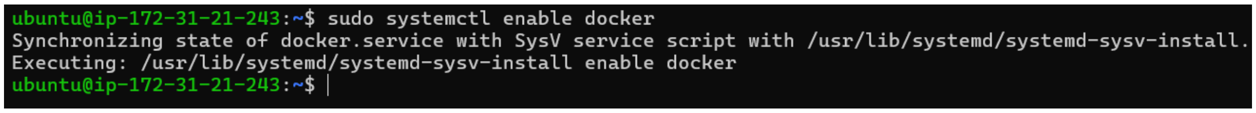
sudo mkdir -p /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json {
"exec-opts": ["native.cgroupdriver=systemd"] } EOF



```
ubuntu@ip-172-31-21-243:~$ sudo mkdir -p /etc/docker
ubuntu@ip-172-31-21-243:~$ cat <<EOF | sudo tee /etc/docker/daemon.json {
"exec-opts": ["native.cgroupdriver=systemd"] } EOF
```

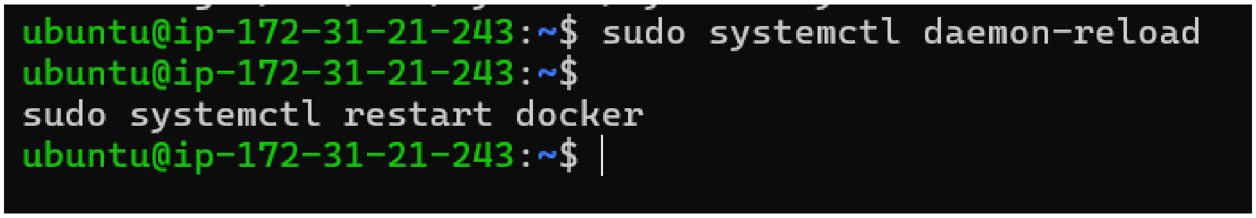
sudo systemctl enable docker

Name:Atharva Shinde Roll No.-54 Div:D15C



```
ubuntu@ip-172-31-21-243:~$ sudo systemctl enable docker
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
ubuntu@ip-172-31-21-243:~$ |
```

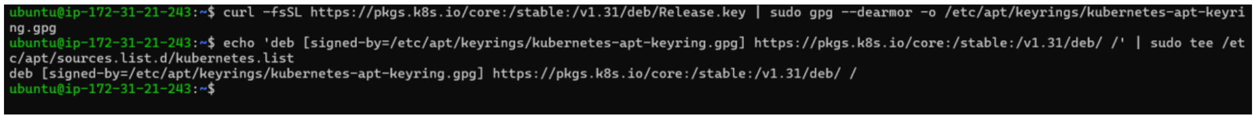
sudo systemctl daemon-reload
sudo systemctl restart docker



```
ubuntu@ip-172-31-21-243:~$ sudo systemctl daemon-reload
ubuntu@ip-172-31-21-243:~$
sudo systemctl restart docker
ubuntu@ip-172-31-21-243:~$ |
```

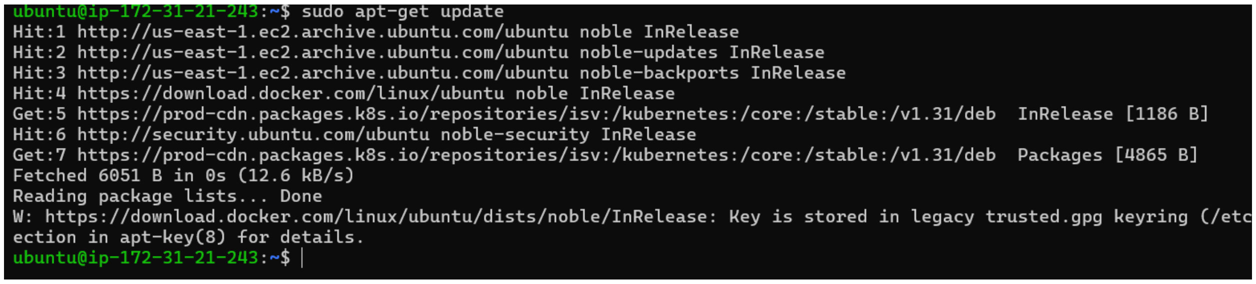
Step 6:Now lets install kubernetes;
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg

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



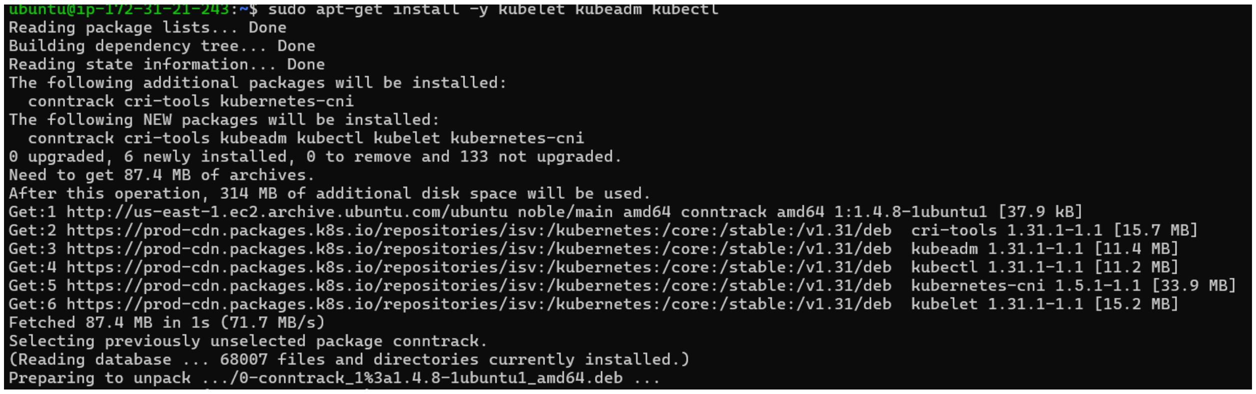
```
ubuntu@ip-172-31-21-243:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyri
ng.gpg
ubuntu@ip-172-31-21-243:~$ echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee /et
c/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/ /
ubuntu@ip-172-31-21-243:~$
```

sudo apt-get update



```
ubuntu@ip-172-31-21-243:~$ 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
Get:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  InRelease [1186 B]
Hit:6 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:7 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.31/deb  Packages [4865 B]
Fetched 6051 B in 0s (12.6 kB/s)
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc
ection in apt-key(8) for details.
ubuntu@ip-172-31-21-243:~$ |
```

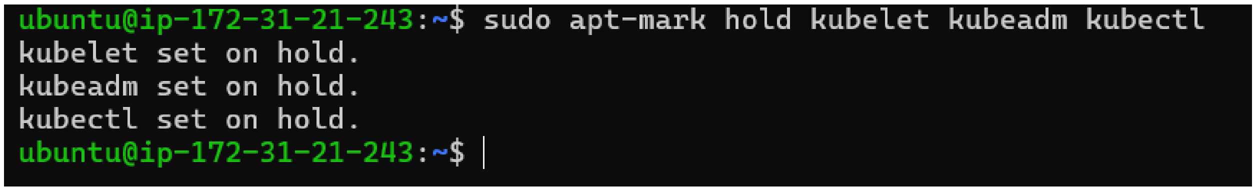
sudo apt-get install -y kubelet kubeadm kubectl



```
ubuntu@ip-172-31-21-243:~$ sudo apt-get install -y kubelet kubeadm kubectl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools kubernetes-cni
The following NEW packages will be installed:
  conntrack cri-tools kubeadm kubectl kubelet kubernetes-cni
0 upgraded, 6 newly installed, 0 to remove and 133 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 (71.7 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 ...
```

sudo apt-mark hold kubelet kubeadm kubectl

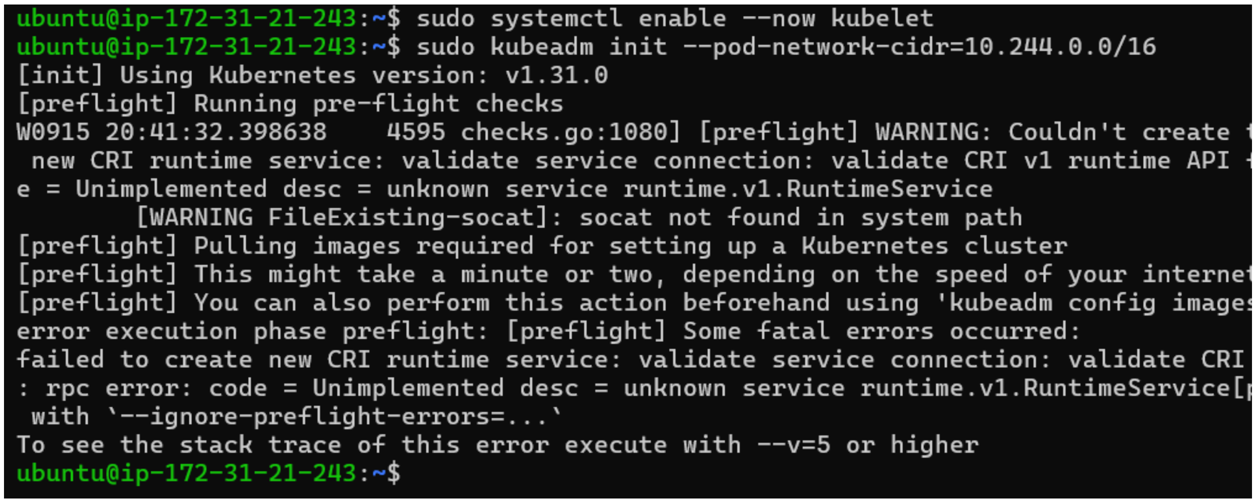
Name:Atharva Shinde Roll No.-54 Div:D15C



```
ubuntu@ip-172-31-21-243:~$ sudo apt-mark hold kubelet kubeadm kubectl
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
ubuntu@ip-172-31-21-243:~$ |
```

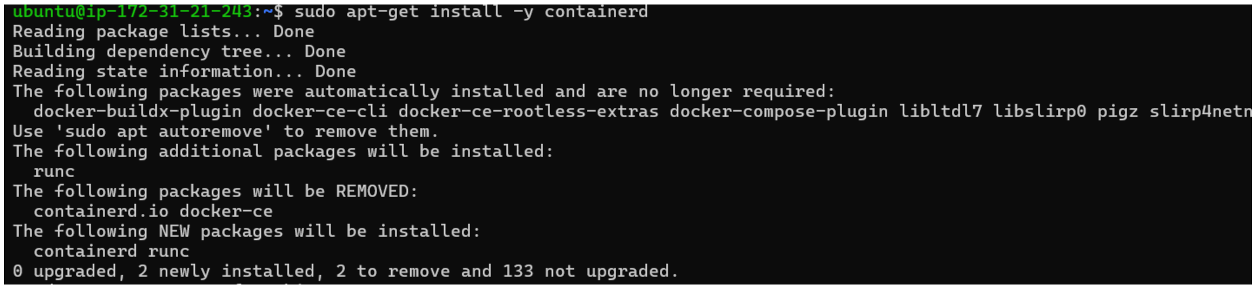
sudo systemctl enable --now kubelet

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



```
ubuntu@ip-172-31-21-243:~$ sudo systemctl enable --now kubelet
ubuntu@ip-172-31-21-243:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
W0915 20:41:32.398638    4595 checks.go:1080] [preflight] WARNING: Couldn't create t
 new CRI runtime service: validate service connection: validate CRI v1 runtime API f
e = 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
[preflight] You can also perform this action beforehand using 'kubeadm config images
error execution phase preflight: [preflight] Some fatal errors occurred:
failed to create new CRI runtime service: validate service connection: validate CRI
: rpc error: code = Unimplemented desc = unknown service runtime.v1.RuntimeService[p
 with `--ignore-preflight-errors=...`
To see the stack trace of this error execute with --v=5 or higher
ubuntu@ip-172-31-21-243:~$
```

sudo apt-get install -y containerd

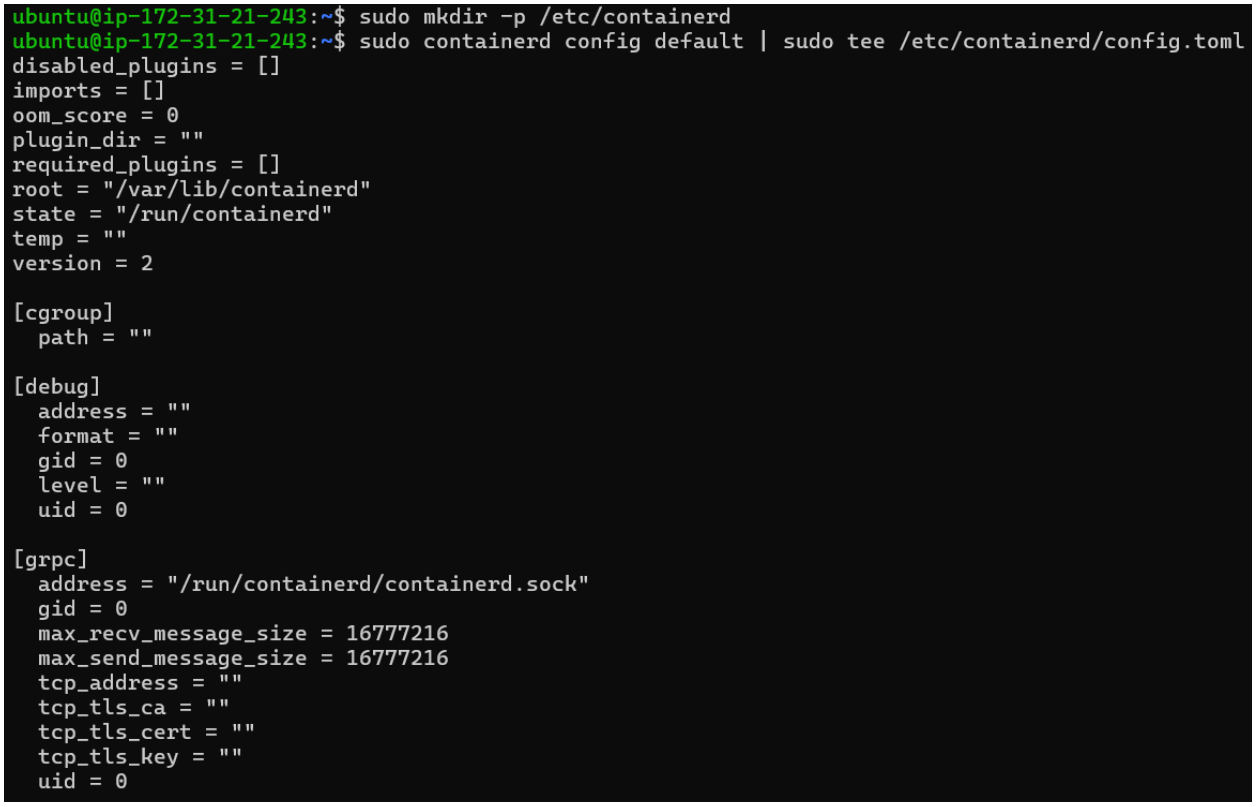


```
ubuntu@ip-172-31-21-243:~$ sudo apt-get install -y containerd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz slirp4netn
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  runc
The following packages will be REMOVED:
  containerd.io docker-ce
The following NEW packages will be installed:
  containerd runc
0 upgraded, 2 newly installed, 2 to remove and 133 not upgraded.
```

sudo mkdir -p /etc/containerd

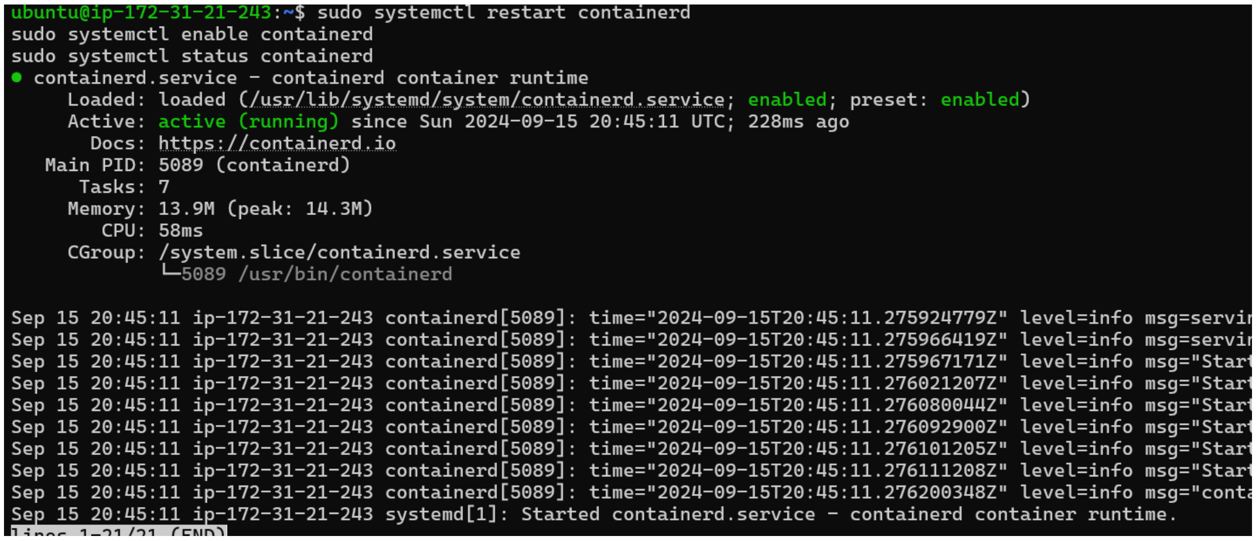
sudo containerd config default | sudo tee /etc/containerd/config.toml

Name:Atharva Shinde Roll No.-54 Div:D15C



```
ubuntu@ip-172-31-21-243:~$ sudo mkdir -p /etc/containerd
ubuntu@ip-172-31-21-243:~$ 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
```

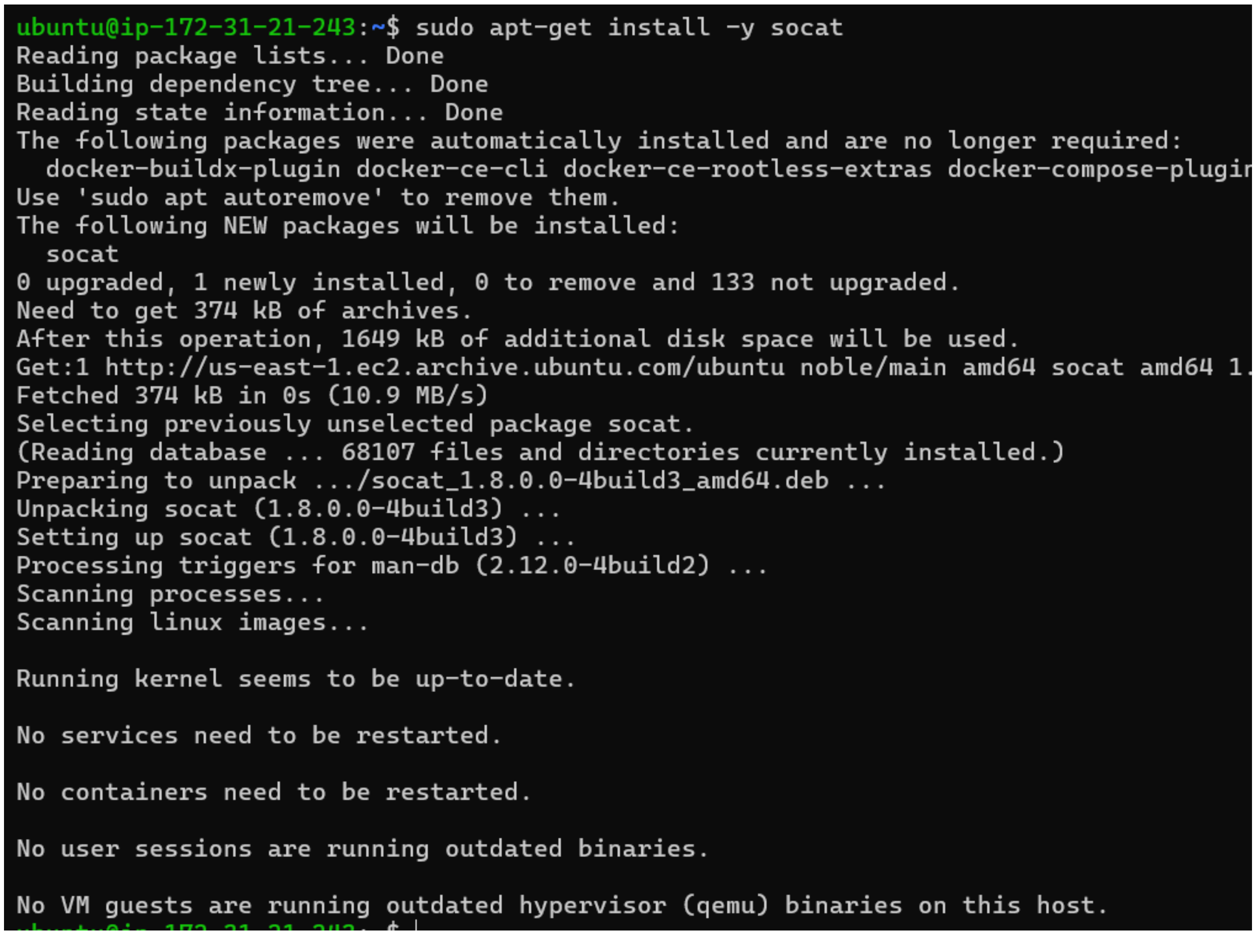
sudo systemctl restart containerd
sudo systemctl enable containerd
sudo systemctl status containerd



```
ubuntu@ip-172-31-21-243:~$ sudo systemctl restart containerd
sudo systemctl enable containerd
sudo systemctl status containerd
● containerd.service - containerd container runtime
     Loaded: loaded (/usr/lib/systemd/system/containerd.service; enabled; preset: enabled)
     Active: active (running) since Sun 2024-09-15 20:45:11 UTC; 228ms ago
       Docs: https://containerd.io
   Main PID: 5089 (containerd)
      Tasks: 7
     Memory: 13.9M (peak: 14.3M)
        CPU: 58ms
     CGroup: /system.slice/containerd.service
             └─5089 /usr/bin/containerd

Sep 15 20:45:11 ip-172-31-21-243 containerd[5089]: time="2024-09-15T20:45:11.275924779Z" level=info msg=servi
Sep 15 20:45:11 ip-172-31-21-243 containerd[5089]: time="2024-09-15T20:45:11.275966419Z" level=info msg=servi
Sep 15 20:45:11 ip-172-31-21-243 containerd[5089]: time="2024-09-15T20:45:11.275967171Z" level=info msg="Star
Sep 15 20:45:11 ip-172-31-21-243 containerd[5089]: time="2024-09-15T20:45:11.276021207Z" level=info msg="Star
Sep 15 20:45:11 ip-172-31-21-243 containerd[5089]: time="2024-09-15T20:45:11.276080044Z" level=info msg="Star
Sep 15 20:45:11 ip-172-31-21-243 containerd[5089]: time="2024-09-15T20:45:11.276092900Z" level=info msg="Star
Sep 15 20:45:11 ip-172-31-21-243 containerd[5089]: time="2024-09-15T20:45:11.276101205Z" level=info msg="Star
Sep 15 20:45:11 ip-172-31-21-243 containerd[5089]: time="2024-09-15T20:45:11.276111208Z" level=info msg="Star
Sep 15 20:45:11 ip-172-31-21-243 containerd[5089]: time="2024-09-15T20:45:11.276200348Z" level=info msg="conta
Sep 15 20:45:11 ip-172-31-21-243 systemd[1]: Started containerd.service - containerd container runtime.
lines 1-21/21 (END)
```

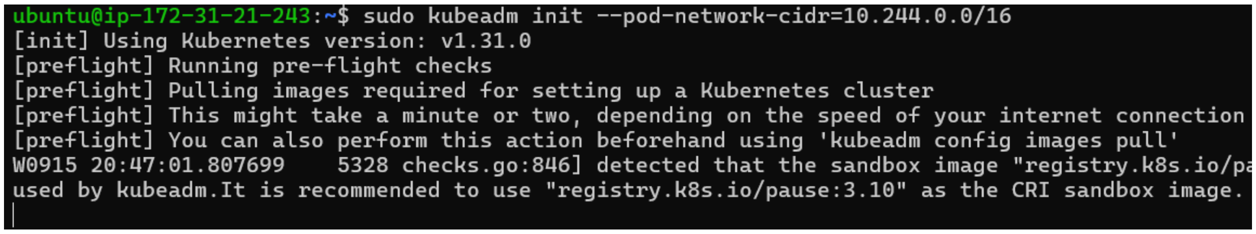
sudo apt-get install -y socat

Name:Atharva Shinde Roll No.-54 Div:D15C

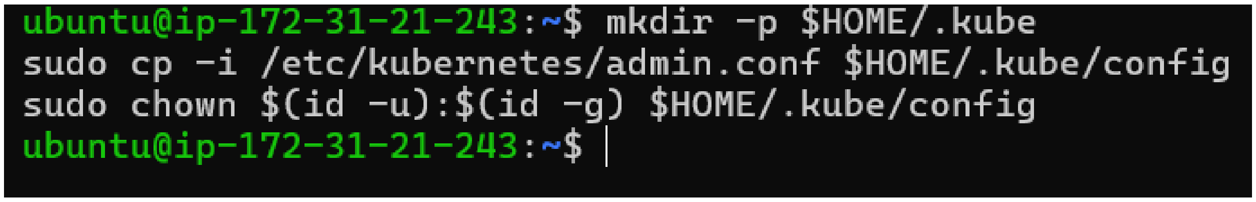


```
ubuntu@ip-172-31-21-243:~$ sudo apt-get install -y socat
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  socat
0 upgraded, 1 newly installed, 0 to remove and 133 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.
Fetched 374 kB in 0s (10.9 MB/s)
Selecting previously unselected package socat.
(Reading database ... 68107 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.
```

Step 7: Intitialize the kubernete:
sudo kubeadm init --pod-network-cidr=10.244.0.0/16



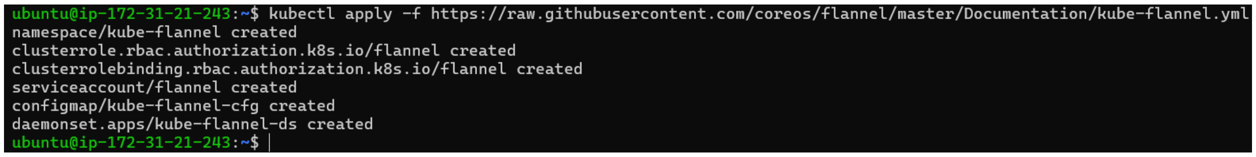
```
ubuntu@ip-172-31-21-243:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
W0915 20:47:01.807699    5328 checks.go:846] detected that the sandbox image "registry.k8s.io/pa
used by kubeadm.It is recommended to use "registry.k8s.io/pause:3.10" as the CRI sandbox image.
```



```
ubuntu@ip-172-31-21-243:~$ mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-21-243:~$
```

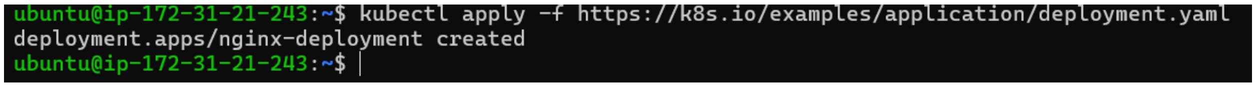
kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml

Name:Atharva Shinde   Roll No.-54   Div:D15C



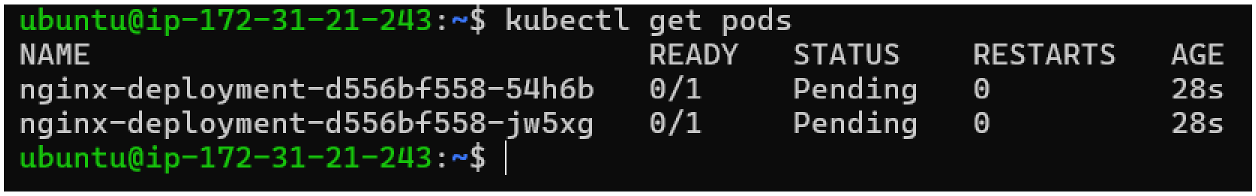
```
ubuntu@ip-172-31-21-243:~$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
ubuntu@ip-172-31-21-243:~$
```

Step 8:Now we can deploy our nginx server on this cluster using following steps:
kubectl apply -f https://k8s.io/examples/application/deployment.yaml



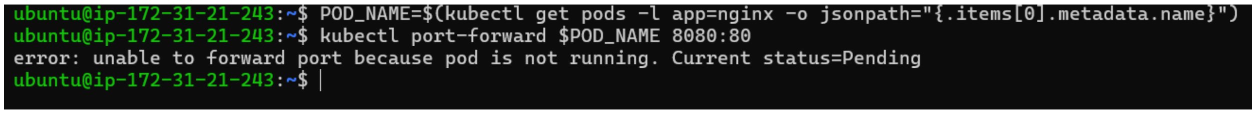
```
ubuntu@ip-172-31-21-243:~$ kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
ubuntu@ip-172-31-21-243:~$
```

kubectl get pods



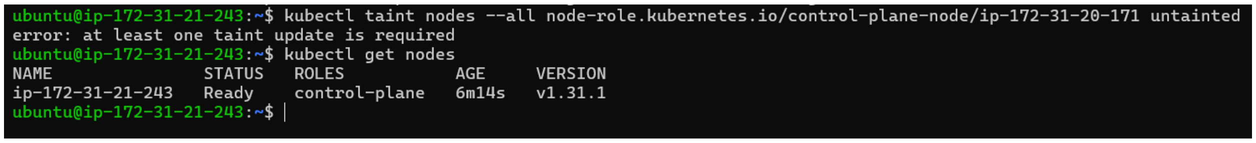
```
ubuntu@ip-172-31-21-243:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-54h6b    0/1     Pending   0          28s
nginx-deployment-d556bf558-jw5xg    0/1     Pending   0          28s
ubuntu@ip-172-31-21-243:~$
```

POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath="{.items[0].metadata.name}")
kubectl port-forward $POD_NAME 8080:80



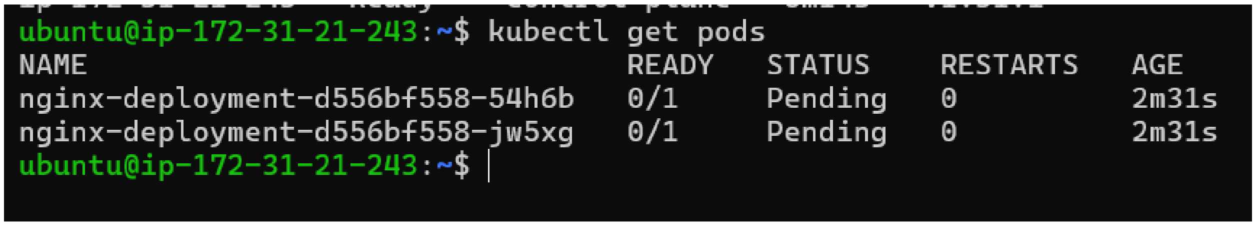
```
ubuntu@ip-172-31-21-243:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath="{.items[0].metadata.name}")
ubuntu@ip-172-31-21-243:~$ kubectl port-forward $POD_NAME 8080:80
error: unable to forward port because pod is not running. Current status=Pending
ubuntu@ip-172-31-21-243:~$
```

kubectl taint nodes --all node-role.kubernetes.io/control-plane-node/ip-172-31-20-171 untainted
kubectl get nodes



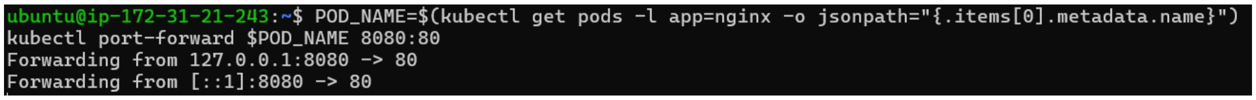
```
ubuntu@ip-172-31-21-243:~$ kubectl taint nodes --all node-role.kubernetes.io/control-plane-node/ip-172-31-20-171 untainted
error: at least one taint update is required
ubuntu@ip-172-31-21-243:~$ kubectl get nodes
NAME               STATUS   ROLES           AGE     VERSION
ip-172-31-21-243   Ready    control-plane   6m14s   v1.31.1
ubuntu@ip-172-31-21-243:~$
```

kubectl get pods



```
ubuntu@ip-172-31-21-243:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-54h6b    0/1     Pending   0          2m31s
nginx-deployment-d556bf558-jw5xg    0/1     Pending   0          2m31s
ubuntu@ip-172-31-21-243:~$
```

POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath="{.items[0].metadata.name}")
kubectl port-forward $POD_NAME 8080:80

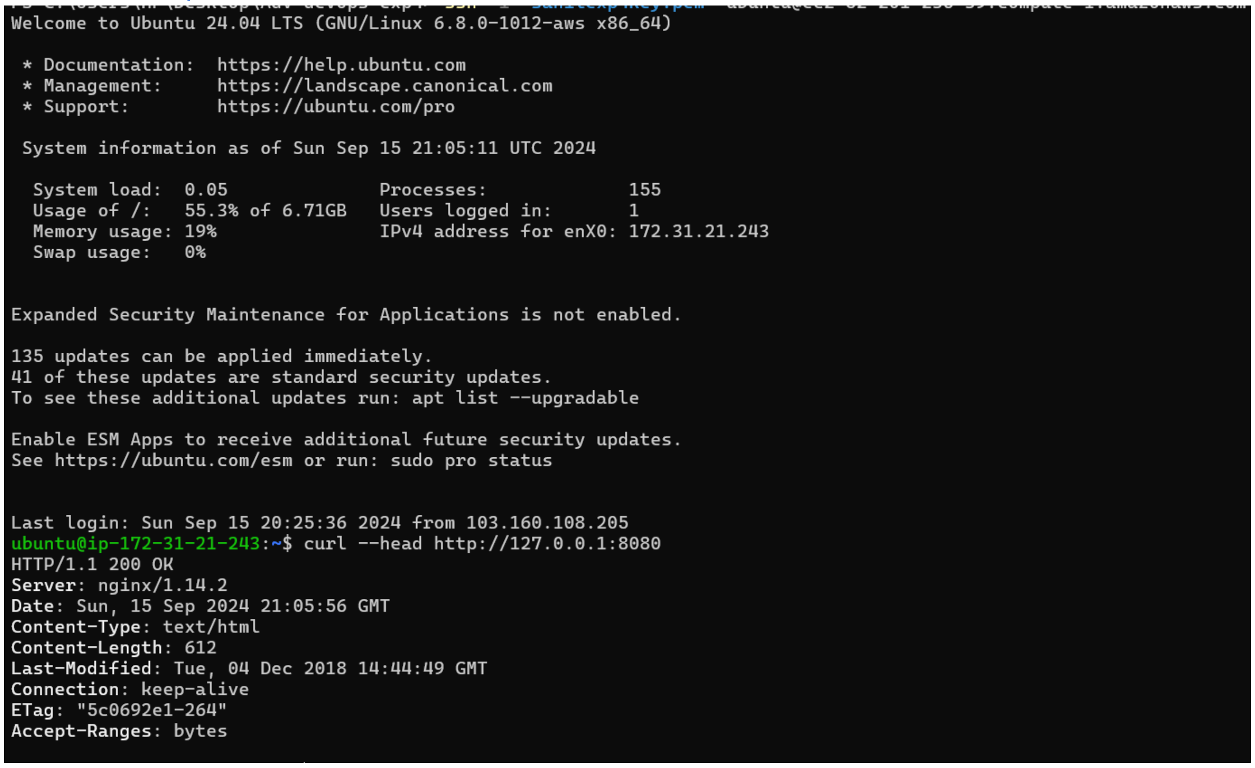


```
ubuntu@ip-172-31-21-243:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath="{.items[0].metadata.name}")
kubectl port-forward $POD_NAME 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
```

Step 9 :check deployment:
Open new terminal in folder,
Paste the ssh key,
Type

Name:Atharva Shinde Roll No.-54 Div:D15C

curl --head http://127.0.0.1:8080



```
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Sun Sep 15 21:05:11 UTC 2024

  System load:  0.05              Processes:             155
  Usage of /:   55.3% of 6.71GB   Users logged in:       1
  Memory usage: 19%               IPv4 address for enX0: 172.31.21.243
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

135 updates can be applied immediately.
41 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Sun Sep 15 20:25:36 2024 from 103.160.108.205
ubuntu@ip-172-31-21-243:~$ curl --head http://127.0.0.1:8080
HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Sun, 15 Sep 2024 21:05:56 GMT
Content-Type: text/html
Content-Length: 612
Last-Modified: Tue, 04 Dec 2018 14:44:49 GMT
Connection: keep-alive
ETag: "5c0692e1-264"
Accept-Ranges: bytes
```

Now we have successfully deployed our nginx server on our ec2 instance.

Conclusion

Installing `kubectl` and using it effectively is a crucial part of managing Kubernetes clusters. As the main interface to Kubernetes, `kubectl` empowers you to deploy, monitor, and troubleshoot applications, providing full control over cluster resources. Deploying an application for the first time serves as an introduction to Kubernetes' ability to orchestrate containers seamlessly. It showcases the power of declarative configurations and automated scaling, which are central to Kubernetes' efficiency in managing modern applications. Understanding how to install and operate `kubectl` lays the foundation for more advanced interactions with Kubernetes, enabling both developers and administrators to harness its full potential in creating resilient, scalable, and portable containerized applications. This knowledge is fundamental for any professional aiming to work within a Kubernetes-driven infrastructure.