Vivekanand Education Society's

Institute of Technology

An Autonomous Institute Affiliated to University of Mumbai,, Approved by AICTE & Recognized by Govt. of Maharashtra
Hashu Advani Memorial Complex, Collector Colony, Chembur East, Mumbai - 400074.

V.E.S.
Since 1962

Department of Information Technology

A.Y. 2024-25

Advance DevOps Lab

Experiment 04

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

Roll No.	43
Name	HARSH PRAMOD PADYAL
Class	D15B
Subject	Advance DevOps Lab
LO Mapped	LO1: To understand the fundamentals of Cloud Computing and be fully proficient with Cloud based DevOps solution deployment options to meet your business requirements. LO2: To deploy single and multiple container applications and manage application deployments with rollouts in Kubernetes
Grade:	

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 interface (CLI) tool that allows users to interact with a Kubernetes cluster. As a central component of Kubernetes, **kubectl** provides the functionality needed to manage applications, inspect cluster resources, and perform administrative tasks through simple commands executed in a terminal.

Importance of Kubectl in Kubernetes Management

kubectl is essential for effective Kubernetes management for several reasons:

1. User-Friendly Interface: **kubectl** offers a command-line interface that simplifies complex operations, making it accessible for developers and administrators.

2. Resource Management: Users can create, update, and delete Kubernetes resources such as pods, deployments, services, and namespaces with straightforward commands.

3. Deployment and Scaling: **kubectl** facilitates the deployment of containerized applications and allows users to easily scale them up or down based on current demands.

4. Monitoring and Troubleshooting: The tool enables users to monitor the health and status of applications running in the cluster. It provides commands to view logs, describe resources, and check the current state of pods and services, which aids in troubleshooting issues.

5. Configuration Management: **kubectl** supports YAML configuration files that define the desired state of applications and resources, allowing users to apply changes consistently and repeatedly across different environments.

Key Features of Kubectl

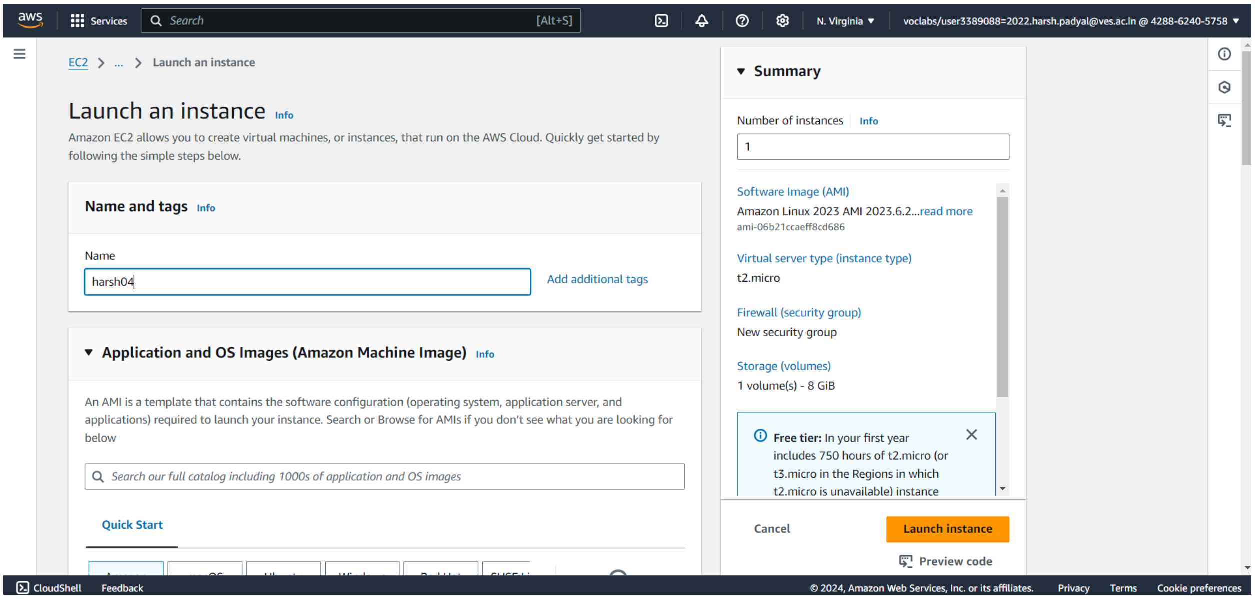
- Resource Discovery: **kubectl** can list all resources in a Kubernetes cluster, providing an overview of what is running and its current status.

- Detailed Resource Descriptions: The tool can display detailed information about specific resources, including configuration, current state, events, and resource utilization.

- Access to Container Logs: Users can view the logs generated by application containers, helping diagnose issues and understand application behavior.

Namespace Management: **kubectl** allows for the management of namespaces, which help organize resources and provide isolation within a cluster.

Launch an EC2 Instance


aws
Services
Search [Alt+S]
N. Virginia ▾
voclabs/user3389088=2022.harsh.padyal@ves.ac.in @ 4288-6240-5758 ▾
EC2 > ... > Launch an instance
Cancel
Launch instance
Preview code
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Launch an instance Info

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 Info

Name: harsh04 — 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

Quick Start

▾ Summary

Number of instances Info: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.6.2...read more
ami-06b21ccaeff8cd686

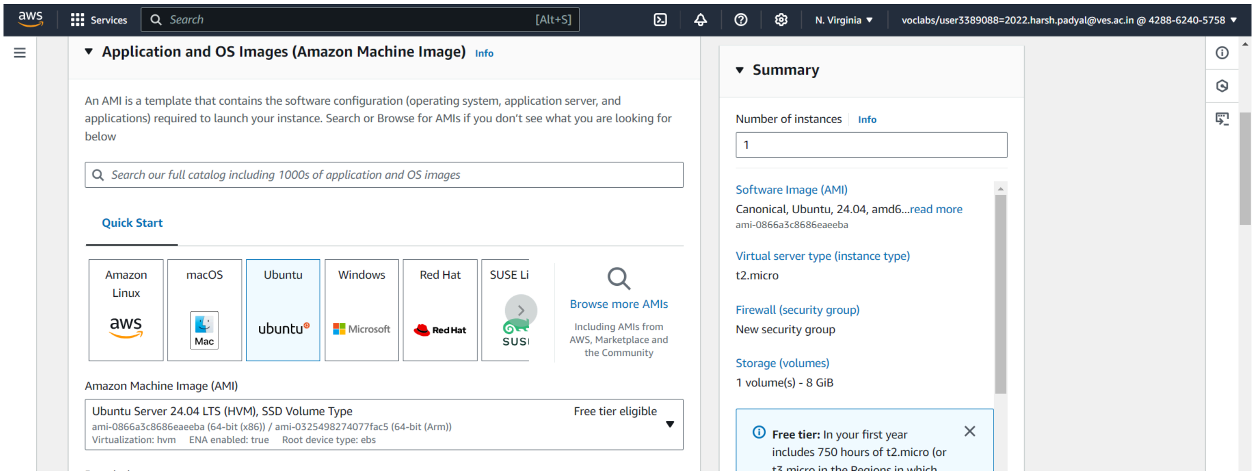
Virtual server type (instance type): t2.micro

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

Choose Ubuntu Server 20.04 LTS (HVM), SSD Volume Type as your AMI.



▾ 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

Quick Start

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-0866a3c8686eaeeba (64-bit (x86)) / ami-0325498274077fac5 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

▾ Summary

Number of instances Info: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd6...read more
ami-0866a3c8686eaeeba

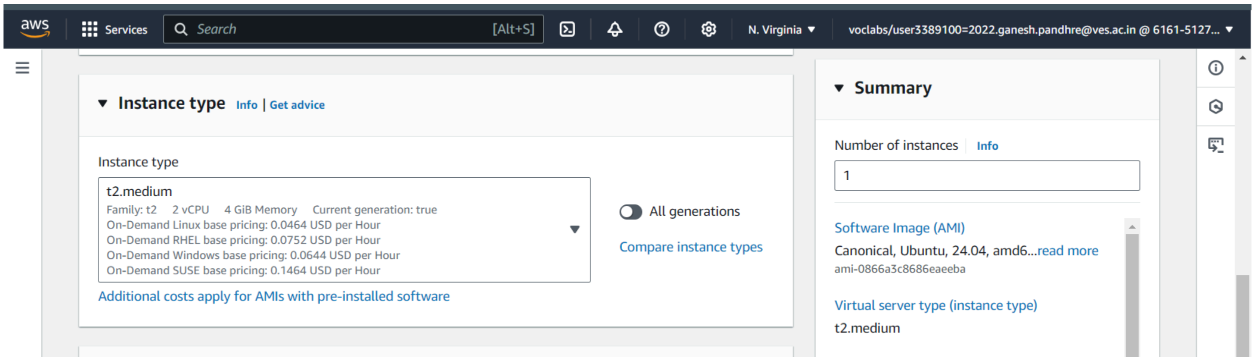
Virtual server type (instance type): t2.micro

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

Select t2.medium as the instance type (2 CPUs).



▾ Instance type Info | Get advice

Instance type

t2.medium
Family: t2 2 vCPU 4 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0464 USD per Hour
On-Demand RHEL base pricing: 0.0752 USD per Hour
On-Demand Windows base pricing: 0.0644 USD per Hour
On-Demand SUSE base pricing: 0.1464 USD per Hour

Additional costs apply for AMIs with pre-installed software

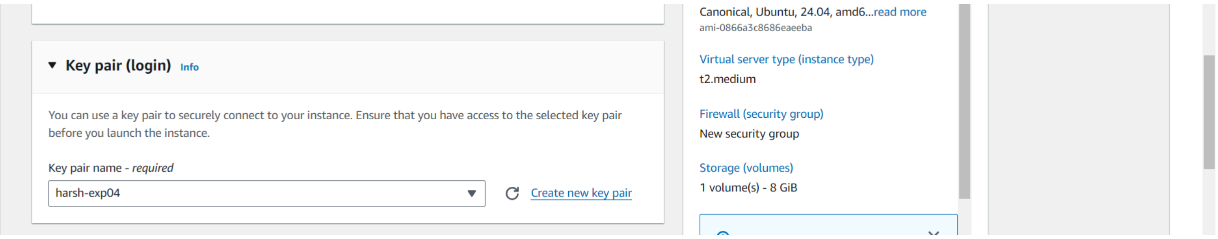
▾ Summary

Number of instances Info: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd6...read more
ami-0866a3c8686eaeeba

Virtual server type (instance type): t2.medium

Select Create a new key pair, name it (e.g., harsh-exp04), and click Download Key Pair. This will download a .pem file to your computer.



▾ 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*: harsh-exp04

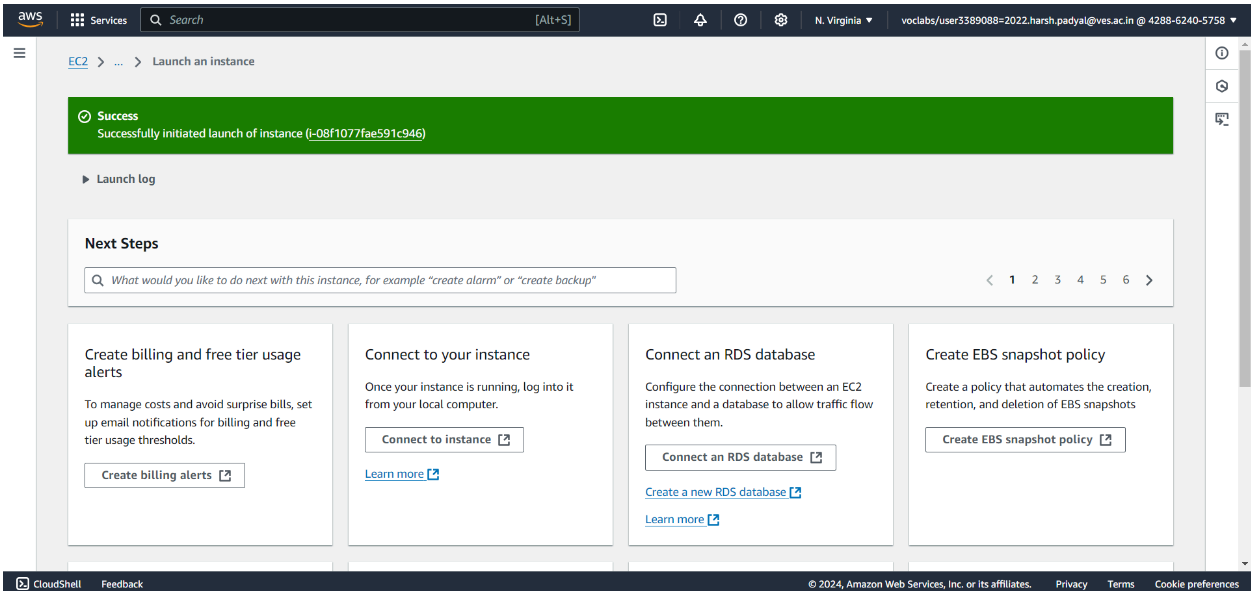
Canonical, Ubuntu, 24.04, amd6...read more
ami-0866a3c8686eaeeba

Virtual server type (instance type): t2.medium

Firewall (security group): New security group

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

Launch the EC2 instance.


aws
Services
Search [Alt+S]
N. Virginia
voclabs/user3389088=2022.harsh.padyal@ves.ac.in @ 4288-6240-5758
EC2 > ... > Launch an instance
Launch log
1 2 3 4 5 6
Create billing alerts
Connect to instance
Learn more
Connect an RDS database
Create a new RDS database
Learn more
Create EBS snapshot policy
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Success
Successfully initiated launch of instance (i-08f1077fae591c946)

Next Steps

What would you like to do next with this instance, for example "create alarm" or "create backup"

Create billing and free tier usage alerts

To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds.

Connect to your instance

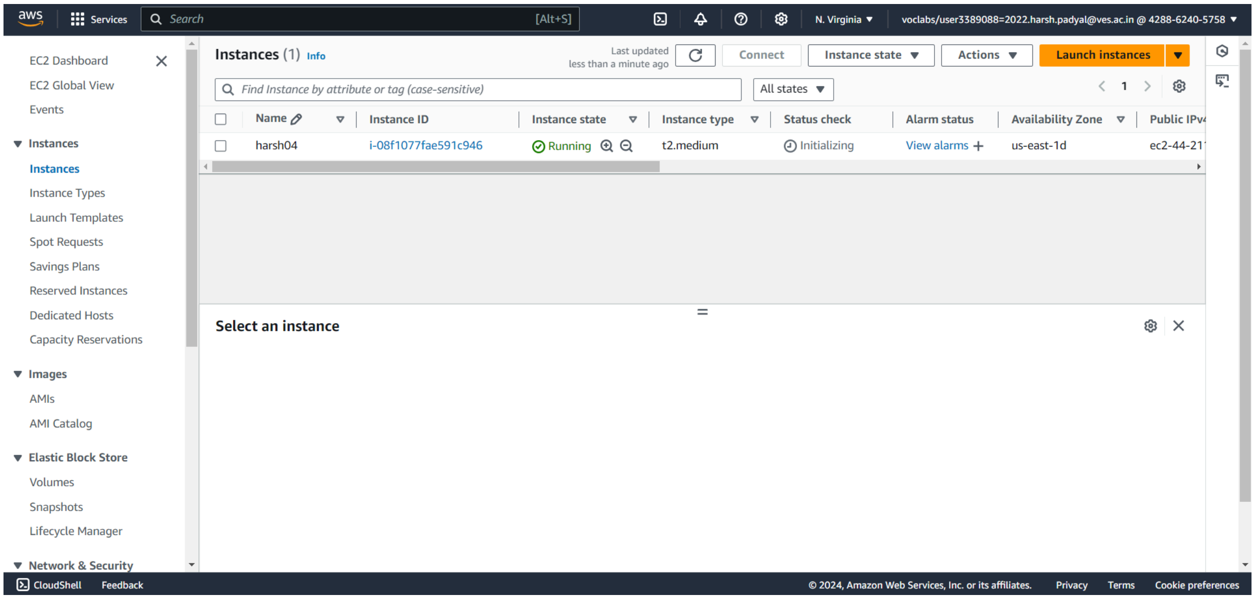
Once your instance is running, log into it from your local computer.

Connect an RDS database

Configure the connection between an EC2 instance and a database to allow traffic flow between them.

Create EBS snapshot policy

Create a policy that automates the creation, retention, and deletion of EBS snapshots



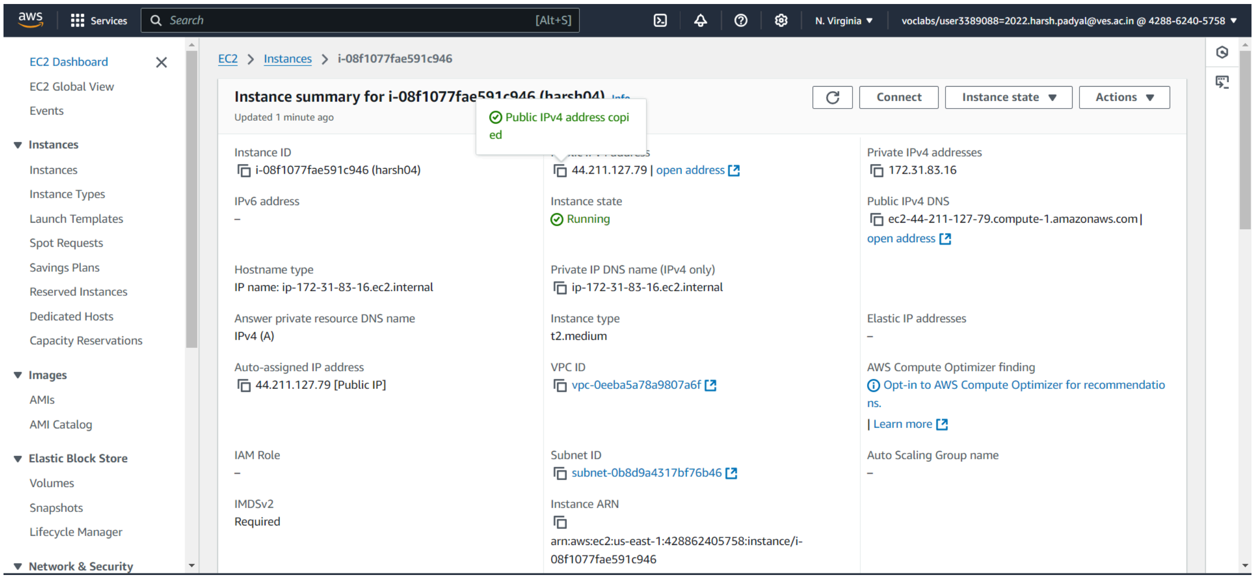
Instances (1)

Last updated less than a minute ago

Find Instance by attribute or tag (case-sensitive)

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv
□	harsh04	i-08f1077fae591c946	✓ Running	t2.medium	Initializing	View alarms +	us-east-1d	ec2-44-21

Click on the instance id of the newly created ec2 instance and copy the public url of it.



Instance summary for i-08f1077fae591c946 (harsh04)

Updated 1 minute ago

✓ Public IPv4 address copied

Instance ID: i-08f1077fae591c946 (harsh04)

Public IPv4 address: 44.211.127.79 | open address

Private IPv4 addresses: 172.31.83.16

IPv6 address: –

Instance state: ✓ Running

Public IPv4 DNS: ec2-44-211-127-79.compute-1.amazonaws.com | open address

Hostname type: IP name: ip-172-31-83-16.ec2.internal

Private IP DNS name (IPv4 only): ip-172-31-83-16.ec2.internal

Answer private resource DNS name: IPv4 (A)

Instance type: t2.medium

Elastic IP addresses: –

Auto-assigned IP address: 44.211.127.79 [Public IP]

VPC ID: vpc-0eeba5a78a9807a6f

AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more

IAM Role: –

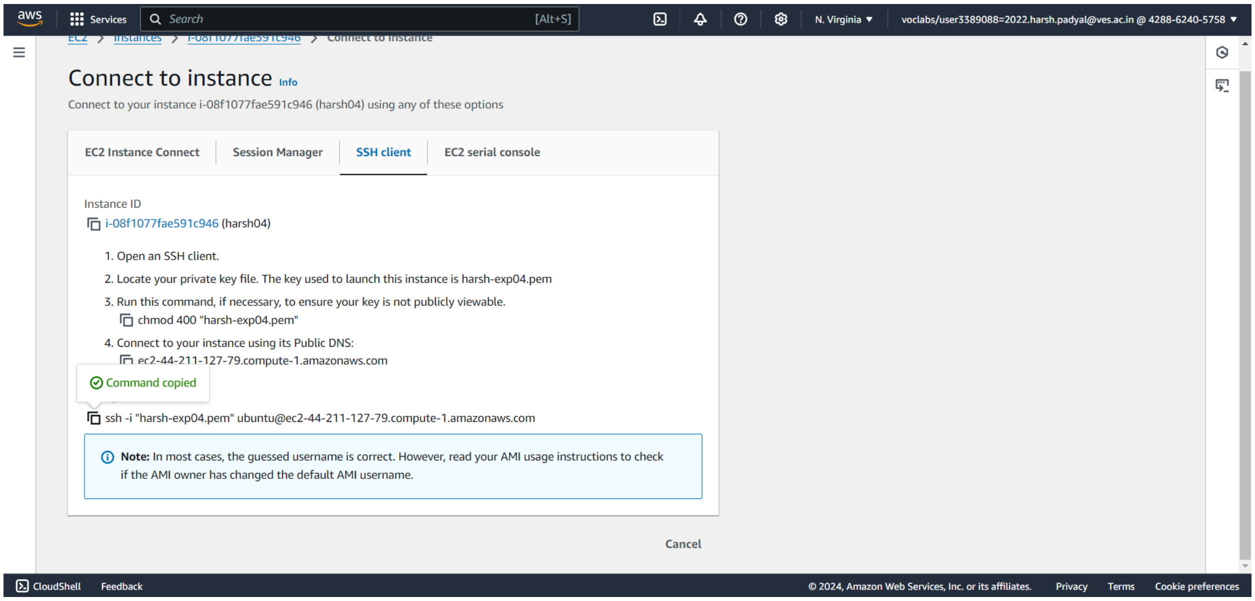
Subnet ID: subnet-0b8d9a4317bf76b46

Auto Scaling Group name: –

IMDSv2: Required

Instance ARN: arn:aws:ec2:us-east-1:428862405758:instance/i-08f1077fae591c946

Click on connect and copy the command as shown


aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3389088=2022.harsh.padyal@ves.ac.in @ 4288-6240-5758
EC2 > Instances > i-08f1077fae591c946 > Connect to instance
Info
EC2 Instance Connect
Session Manager
SSH client
EC2 serial console
Command copied
Cancel
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Connect to instance

Connect to your instance i-08f1077fae591c946 (harsh04) using any of these options

Instance ID

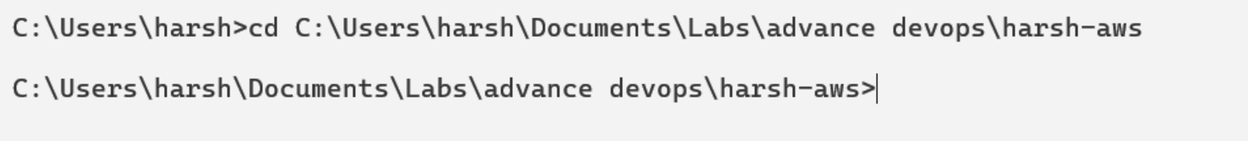
i-08f1077fae591c946 (harsh04)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is harsh-exp04.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   chmod 400 "harsh-exp04.pem"
4. Connect to your instance using its Public DNS:
   ec2-44-211-127-79.compute-1.amazonaws.com

ssh -i "harsh-exp04.pem" ubuntu@ec2-44-211-127-79.compute-1.amazonaws.com

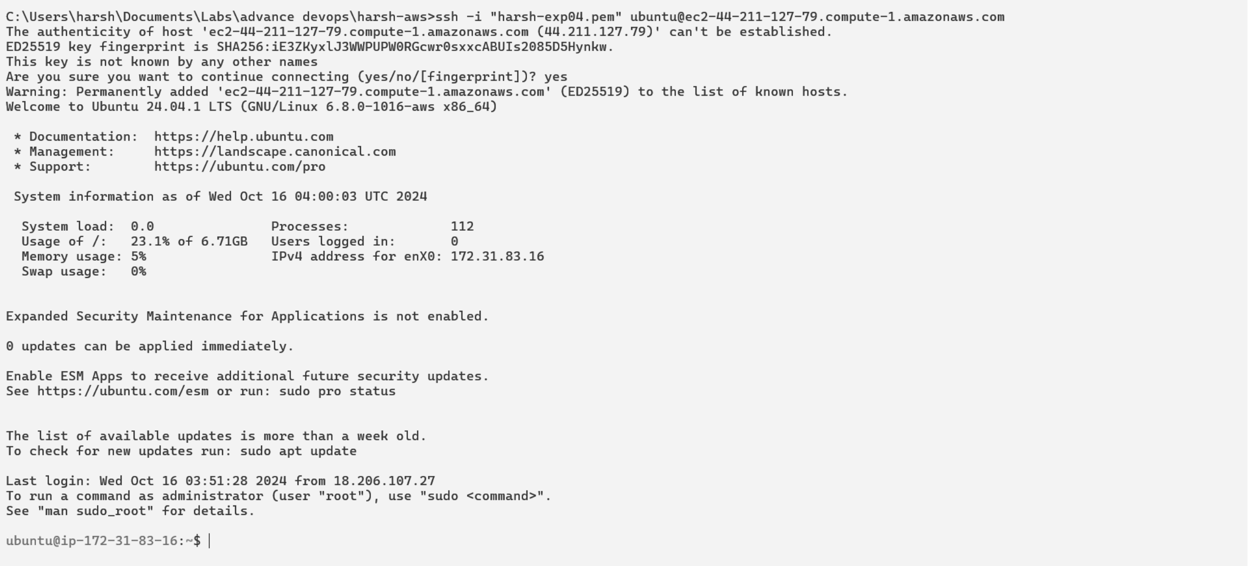
Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

If you are using Windows, you might need a terminal like **Git Bash** or **PuTTY**.
Use the cd command to navigate to the folder where your downloaded key is located.



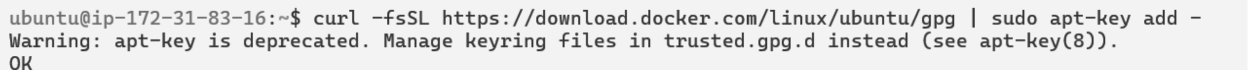
```
C:\Users\harsh>cd C:\Users\harsh\Documents\Labs\advance devops\harsh-aws

C:\Users\harsh\Documents\Labs\advance devops\harsh-aws>
```

Run the following command, replacing the placeholder with your actual EC2 public DNS:
ssh -i "harsh-exp04.pem" ubuntu@ec2-44-211-127-79.compute-1.amazonaws.com



```
C:\Users\harsh\Documents\Labs\advance devops\harsh-aws>ssh -i "harsh-exp04.pem" ubuntu@ec2-44-211-127-79.compute-1.amazonaws.com
The authenticity of host 'ec2-44-211-127-79.compute-1.amazonaws.com (44.211.127.79)' can't be established.
ED25519 key fingerprint is SHA256:iE3ZKyxlJ3WWPUPW0RGcwr0sxxcABUIs2085D5Hynkw.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-211-127-79.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1016-aws x86_64)

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

 System information as of Wed Oct 16 04:00:03 UTC 2024

  System load:  0.0               Processes:             112
  Usage of /:   23.1% of 6.71GB   Users logged in:       0
  Memory usage: 5%                IPv4 address for enX0: 172.31.83.16
  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: Wed Oct 16 03:51:28 2024 from 18.206.107.27
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-83-16:~$ |
```

To install Docker, Run the Following Commands:
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

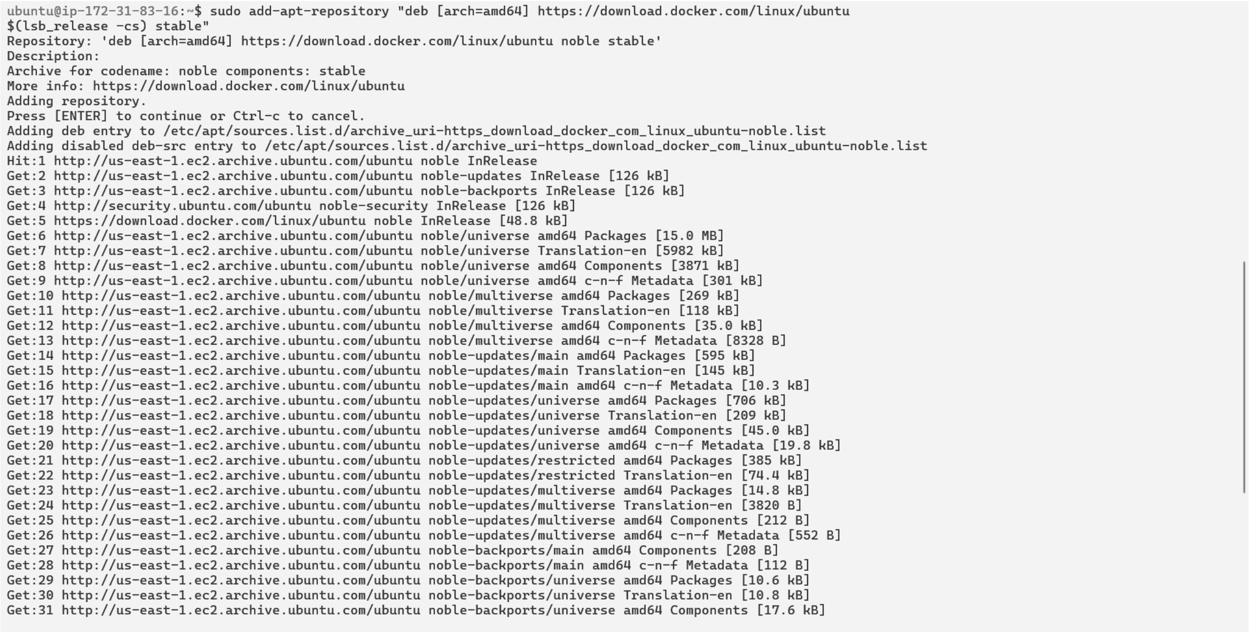


```
ubuntu@ip-172-31-83-16:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
```

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

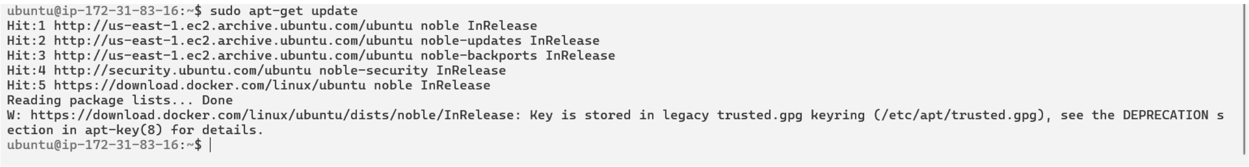
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu

$(lsb_release -cs) stable"



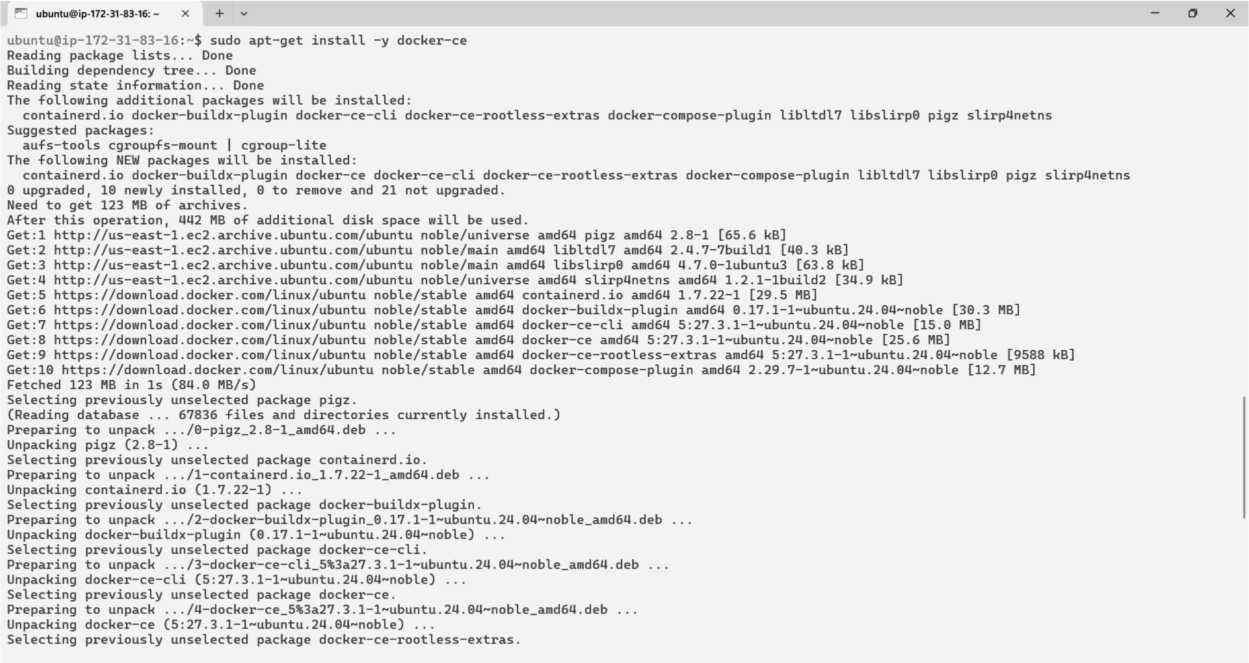
```
ubuntu@ip-172-31-83-16:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable"
Repository: 'deb [arch=amd64] https://download.docker.com/linux/ubuntu noble stable'
Description:
Archive for codename: noble components: stable
More info: https://download.docker.com/linux/ubuntu
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]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 https://download.docker.com/linux/ubuntu noble InRelease [48.8 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [595 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [145 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [10.3 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [706 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [209 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [19.8 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [385 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [74.4 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [14.8 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3820 B]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [212 B]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [552 B]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [10.6 kB]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.8 kB]
Get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [17.6 kB]
```

sudo apt-get update



```
ubuntu@ip-172-31-83-16:~$ 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 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 https://download.docker.com/linux/ubuntu noble InRelease
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION s
ection in apt-key(8) for details.
ubuntu@ip-172-31-83-16:~$
```

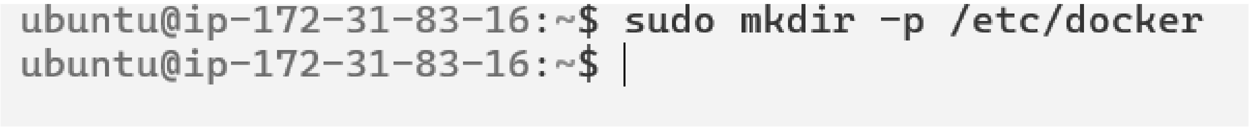
sudo apt-get install -y docker-ce



```
ubuntu@ip-172-31-83-16:~$ 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 21 not upgraded.
Need to get 123 MB of archives.
After this operation, 442 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]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libltdl7 amd64 2.4.7-7build1 [40.3 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libslirp0 amd64 4.7.0-1ubuntu3 [63.8 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 slirp4netns amd64 1.2.1-1build2 [34.9 kB]
Get:5 https://download.docker.com/linux/ubuntu noble/stable amd64 containerd.io amd64 1.7.22-1 [29.5 MB]
Get:6 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-buildx-plugin amd64 0.17.1-1~ubuntu.24.04~noble [30.3 MB]
Get:7 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce-cli amd64 5:27.3.1-1~ubuntu.24.04~noble [15.0 MB]
Get:8 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce amd64 5:27.3.1-1~ubuntu.24.04~noble [25.6 MB]
Get:9 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce-rootless-extras amd64 5:27.3.1-1~ubuntu.24.04~noble [9588 kB]
Get:10 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-compose-plugin amd64 2.29.7-1~ubuntu.24.04~noble [12.7 MB]
Fetched 123 MB in 1s (84.0 MB/s)
Selecting previously unselected package pigz.
(Reading database ... 67836 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...
Selecting previously unselected package containerd.io.
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.
```

Configure Docker

sudo mkdir -p /etc/docker

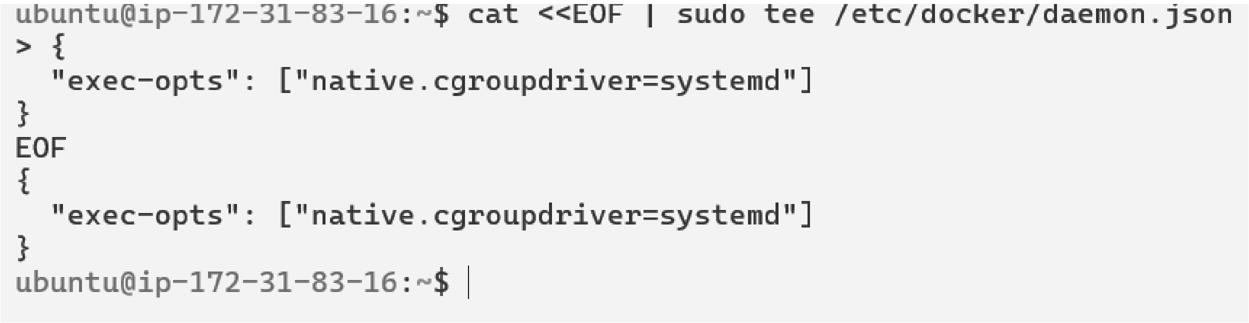


```
ubuntu@ip-172-31-83-16:~$ sudo mkdir -p /etc/docker
ubuntu@ip-172-31-83-16:~$
```

cat <<EOF | sudo tee /etc/docker/daemon.json

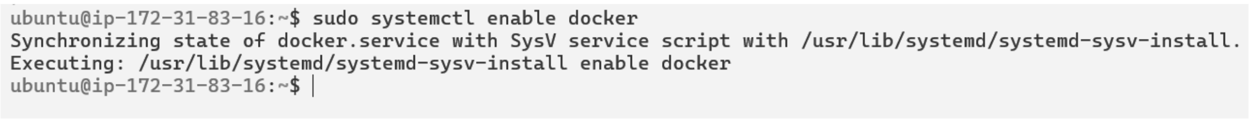
{

```
  "exec-opts": ["native.cgroupdriver=systemd"]
}
EOF
```



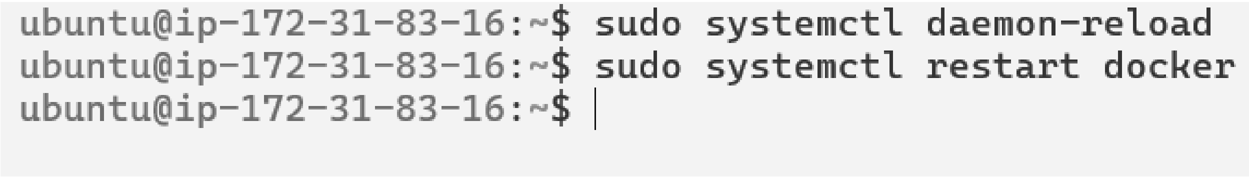
```
ubuntu@ip-172-31-83-16:~$ cat <<EOF | sudo tee /etc/docker/daemon.json
> {
  "exec-opts": ["native.cgroupdriver=systemd"]
}
EOF
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}
ubuntu@ip-172-31-83-16:~$
```

sudo systemctl enable docker



```
ubuntu@ip-172-31-83-16:~$ 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-172-31-83-16:~$
```

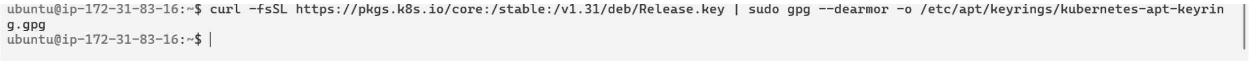
sudo systemctl daemon-reload
sudo systemctl restart docker



```
ubuntu@ip-172-31-83-16:~$ sudo systemctl daemon-reload
ubuntu@ip-172-31-83-16:~$ sudo systemctl restart docker
ubuntu@ip-172-31-83-16:~$
```

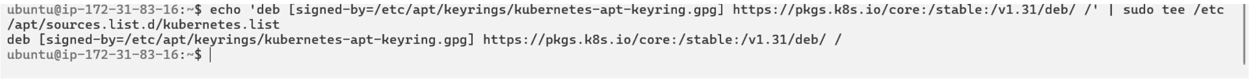
To Install Kubernetes, Add the Kubernetes Repository

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



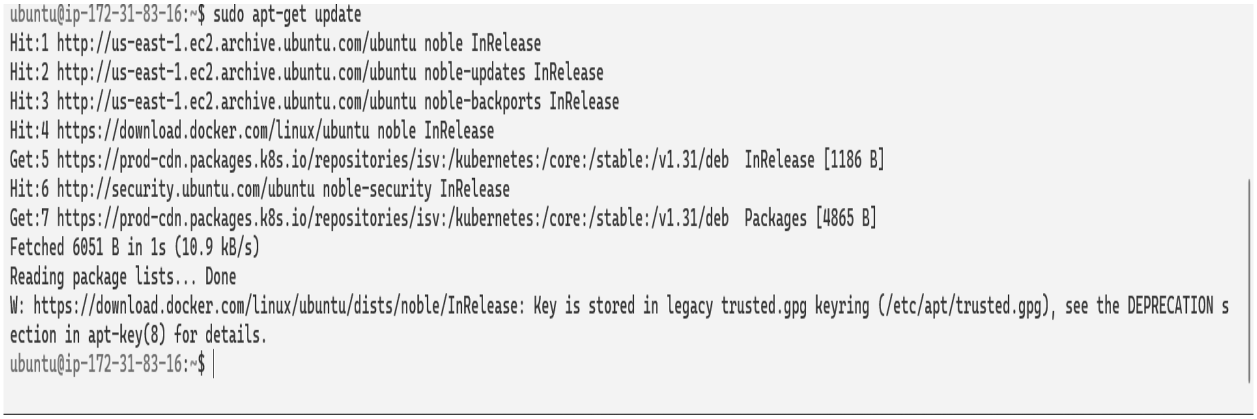
```
ubuntu@ip-172-31-83-16:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyrin
g.gpg
ubuntu@ip-172-31-83-16:~$
```

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



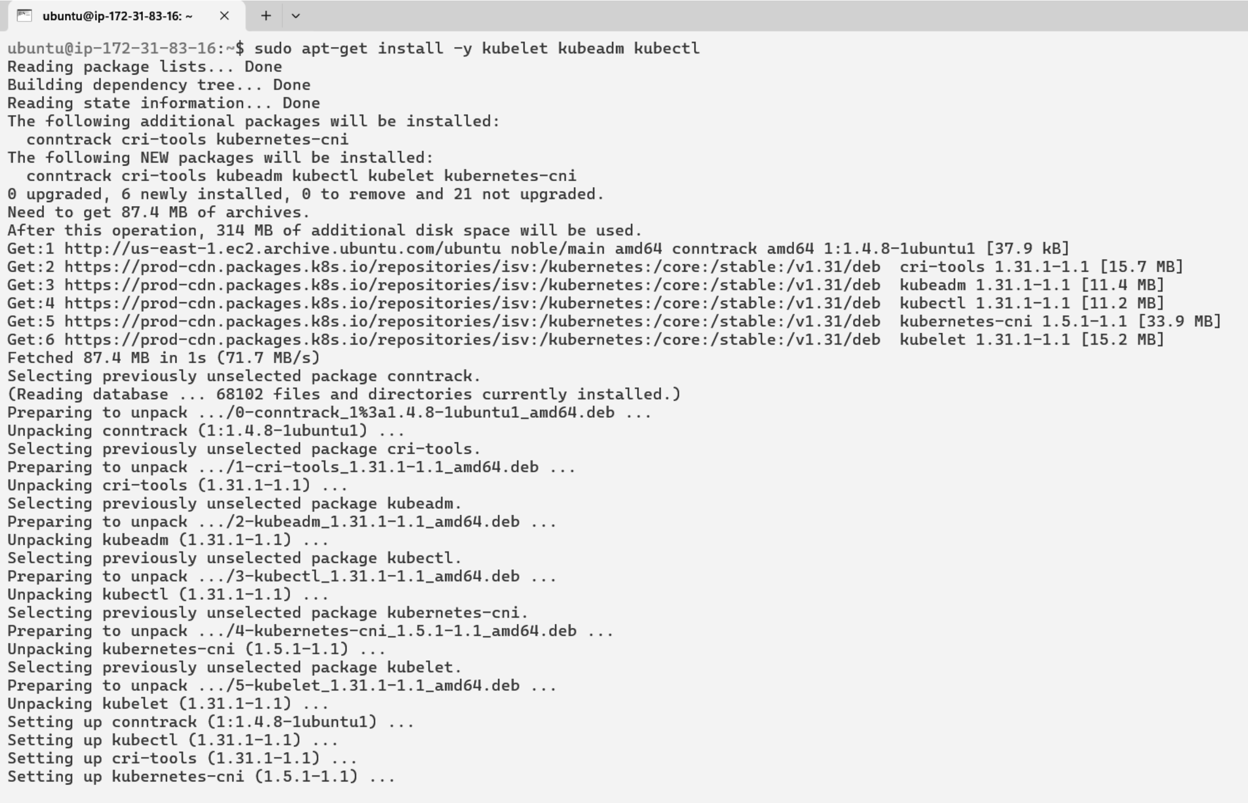
```
ubuntu@ip-172-31-83-16:~$ echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee /etc
/apt/sources.list.d/kubernetes.list
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /
ubuntu@ip-172-31-83-16:~$
```

sudo apt-get update



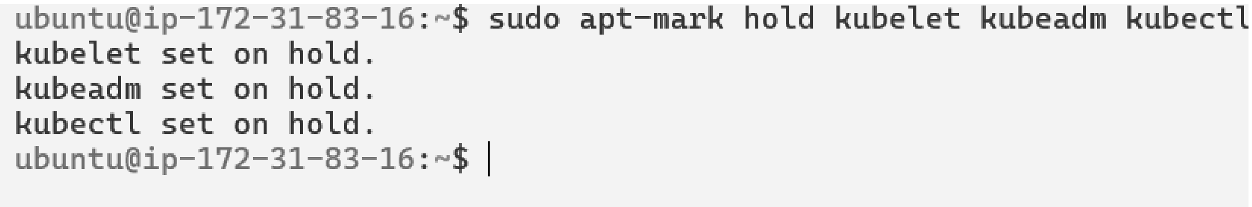
```
ubuntu@ip-172-31-83-16:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu 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 1s (10.9 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/apt/trusted.gpg), see the DEPRECATION s
ection in apt-key(8) for details.
ubuntu@ip-172-31-83-16:~$
```

sudo apt-get install -y kubelet kubeadm kubectl



```
ubuntu@ip-172-31-83-16:~$ 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 21 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 ... 68102 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) ...
```

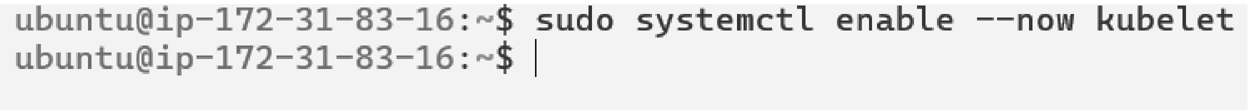
sudo apt-mark hold kubelet kubeadm kubectl



```
ubuntu@ip-172-31-83-16:~$ sudo apt-mark hold kubelet kubeadm kubectl
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
ubuntu@ip-172-31-83-16:~$
```

Enable and Start Kubelet:

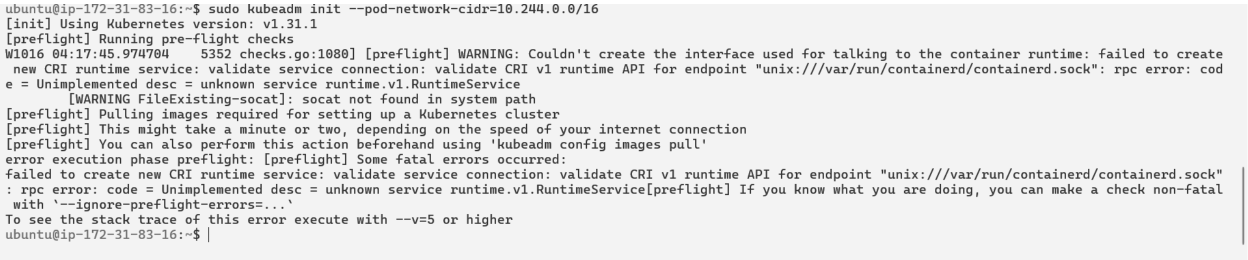
sudo systemctl enable --now kubelet



```
ubuntu@ip-172-31-83-16:~$ sudo systemctl enable --now kubelet
ubuntu@ip-172-31-83-16:~$
```

To Initialize the Kubernetes Cluster, Run the Command

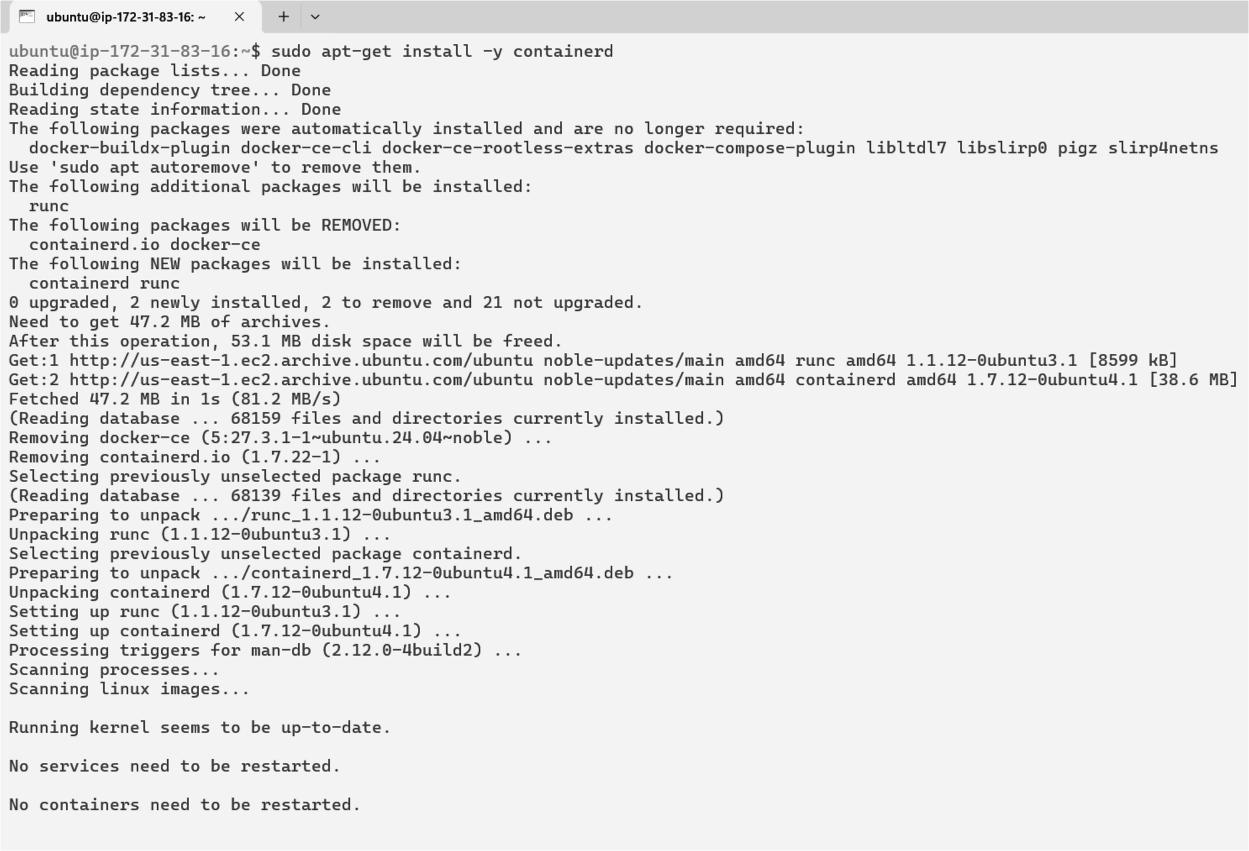
sudo kubeadm init --pod-network-cidr=10.244.0.0/16



```
ubuntu@ip-172-31-83-16:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.1
[preflight] Running pre-flight checks
W1016 04:17:45.974704    5352 checks.go:1080] [preflight] WARNING: Couldn't create the interface used for talking to the container runtime: 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: cod
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 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 service runtime.v1.RuntimeService[preflight] If you know what you are doing, you can make a check non-fatal
 with `--ignore-preflight-errors=...`
To see the stack trace of this error execute with --v=5 or higher
ubuntu@ip-172-31-83-16:~$
```

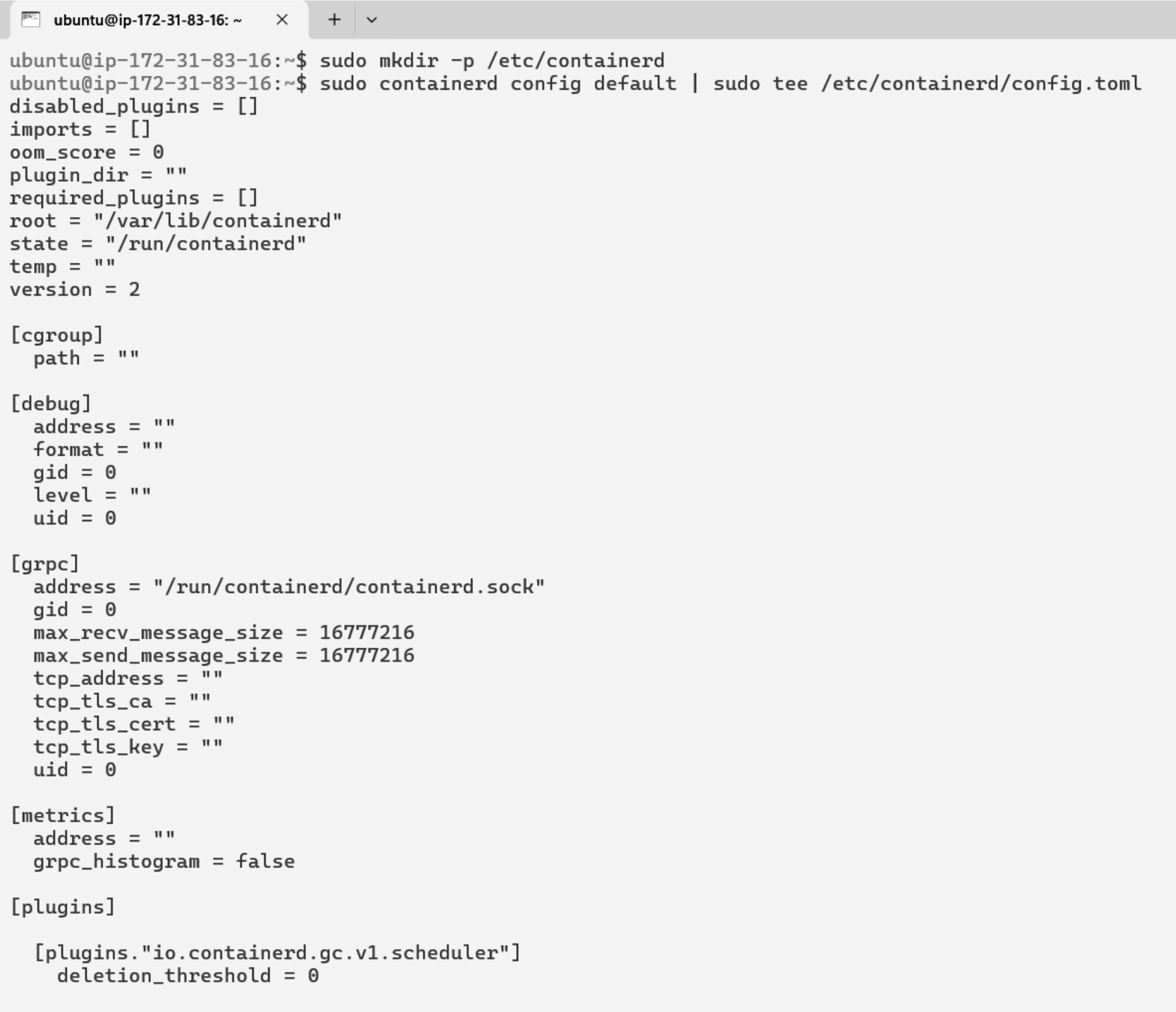
If you encounter errors, run the following commands to fix containerd issues:

sudo apt-get install -y containerd



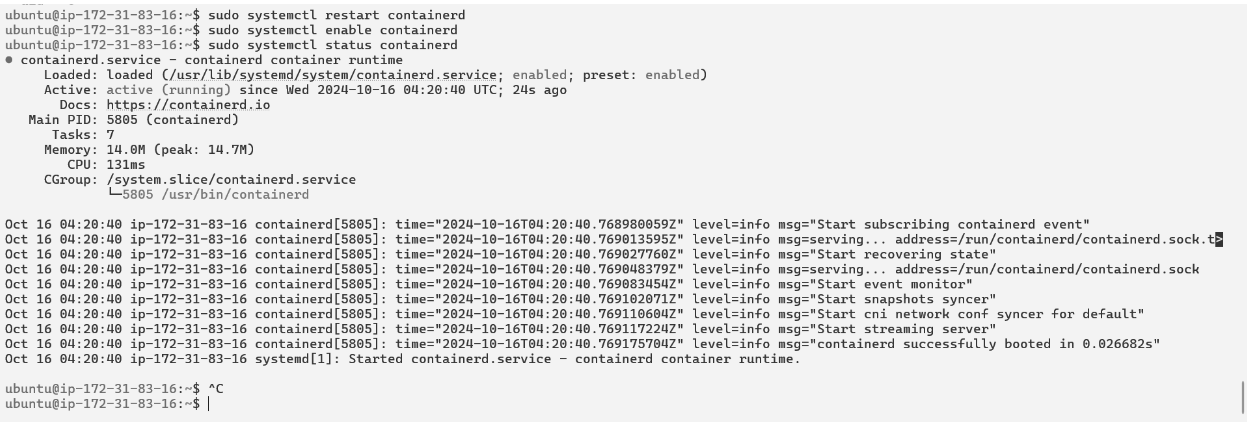
```
ubuntu@ip-172-31-83-16:~$ sudo apt-get install -y containerd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz slirp4netns
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 21 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 1.1.12-0ubuntu3.1 [8599 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.12-0ubuntu4.1 [38.6 MB]
Fetched 47.2 MB in 1s (81.2 MB/s)
(Reading database ... 68159 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 ... 68139 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.
```

sudo mkdir -p /etc/containerd
sudo containerd config default | sudo tee /etc/containerd/config.toml



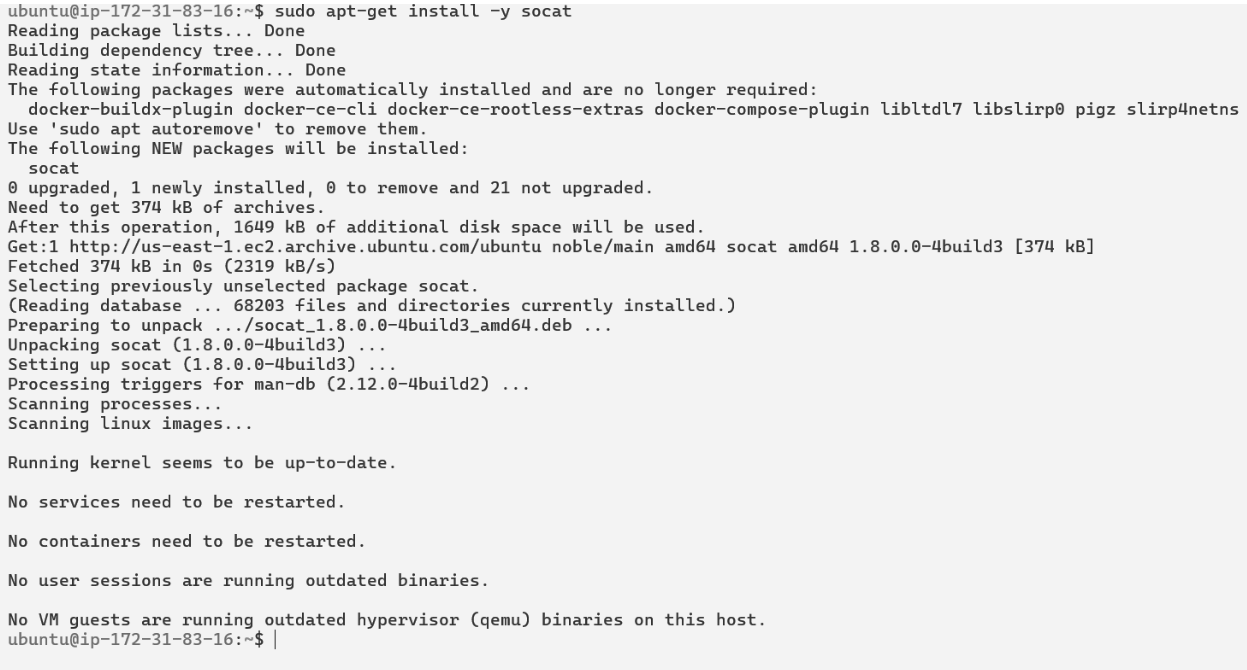
```
ubuntu@ip-172-31-83-16:~$ sudo mkdir -p /etc/containerd
ubuntu@ip-172-31-83-16:~$ sudo containerd config default | sudo tee /etc/containerd/config.toml
disabled_plugins = []
imports = []
oom_score = 0
plugin_dir = ""
required_plugins = []
root = "/var/lib/containerd"
state = "/run/containerd"
temp = ""
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
```

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



```
ubuntu@ip-172-31-83-16:~$ sudo systemctl restart containerd
ubuntu@ip-172-31-83-16:~$ sudo systemctl enable containerd
ubuntu@ip-172-31-83-16:~$ sudo systemctl status containerd
● containerd.service - containerd container runtime
     Loaded: loaded (/usr/lib/systemd/system/containerd.service; enabled; preset: enabled)
     Active: active (running) since Wed 2024-10-16 04:20:40 UTC; 24s ago
       Docs: https://containerd.io
   Main PID: 5805 (containerd)
      Tasks: 7
     Memory: 14.0M (peak: 14.7M)
        CPU: 131ms
     CGroup: /system.slice/containerd.service
             └─5805 /usr/bin/containerd

Oct 16 04:20:40 ip-172-31-83-16 containerd[5805]: time="2024-10-16T04:20:40.768980059Z" level=info msg="Start subscribing containerd event"
Oct 16 04:20:40 ip-172-31-83-16 containerd[5805]: time="2024-10-16T04:20:40.769013595Z" level=info msg=serving... address=/run/containerd/containerd.sock.t>
Oct 16 04:20:40 ip-172-31-83-16 containerd[5805]: time="2024-10-16T04:20:40.769027760Z" level=info msg="Start recovering state"
Oct 16 04:20:40 ip-172-31-83-16 containerd[5805]: time="2024-10-16T04:20:40.769048379Z" level=info msg=serving... address=/run/containerd/containerd.sock
Oct 16 04:20:40 ip-172-31-83-16 containerd[5805]: time="2024-10-16T04:20:40.769083454Z" level=info msg="Start event monitor"
Oct 16 04:20:40 ip-172-31-83-16 containerd[5805]: time="2024-10-16T04:20:40.769102071Z" level=info msg="Start snapshots syncer"
Oct 16 04:20:40 ip-172-31-83-16 containerd[5805]: time="2024-10-16T04:20:40.769110604Z" level=info msg="Start cni network conf syncer for default"
Oct 16 04:20:40 ip-172-31-83-16 containerd[5805]: time="2024-10-16T04:20:40.769117224Z" level=info msg="Start streaming server"
Oct 16 04:20:40 ip-172-31-83-16 containerd[5805]: time="2024-10-16T04:20:40.769175704Z" level=info msg="containerd successfully booted in 0.026682s"
Oct 16 04:20:40 ip-172-31-83-16 systemd[1]: Started containerd.service - containerd container runtime.

ubuntu@ip-172-31-83-16:~$ ^C
ubuntu@ip-172-31-83-16:~$ |
```

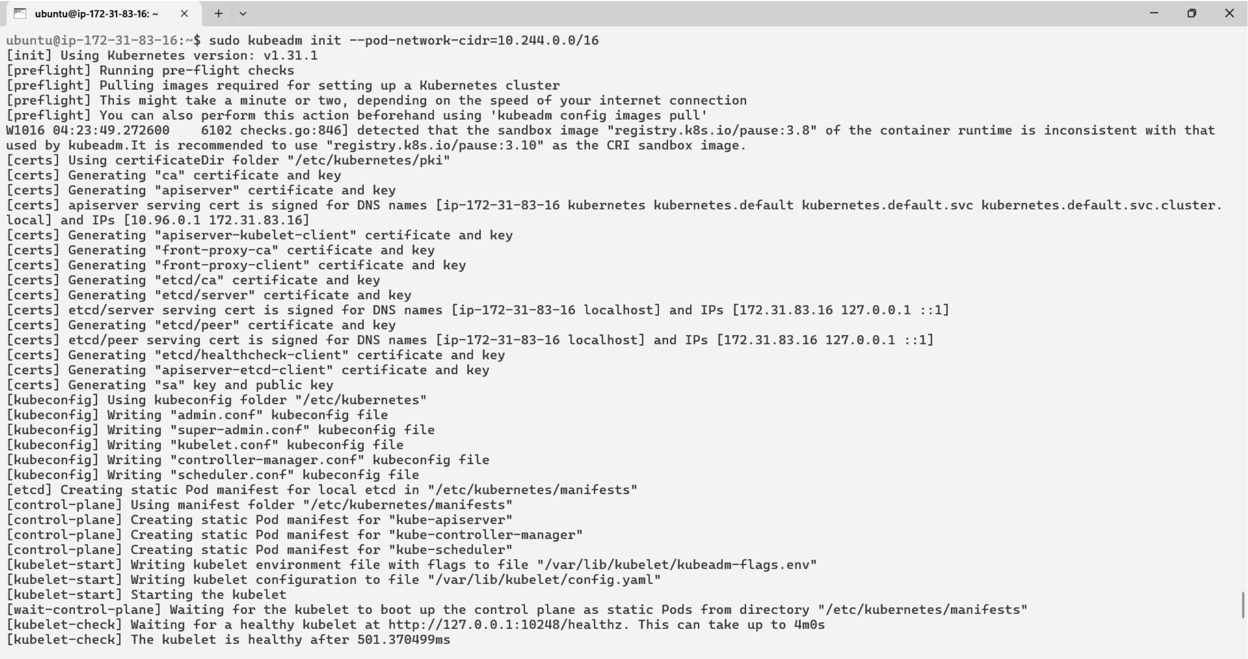
sudo apt-get install -y socat



```
ubuntu@ip-172-31-83-16:~$ 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 21 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 (2319 kB/s)
Selecting previously unselected package socat.
(Reading database ... 68203 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.
ubuntu@ip-172-31-83-16:~$
```

Re-run the Init Command:

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



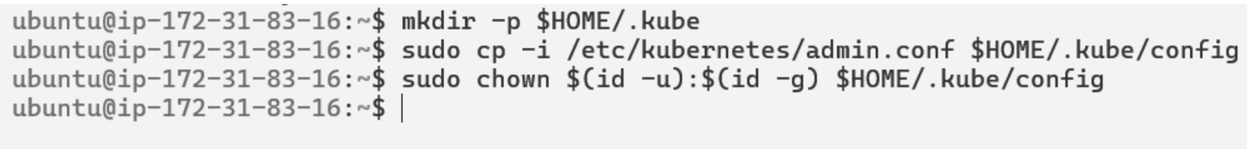
```
ubuntu@ip-172-31-83-16:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.1
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
W1016 04:23:49.272600    6102 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that
used by kubeadm.It is recommended to use "registry.k8s.io/pause:3.10" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-83-16 kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.
local] and IPs [10.96.0.1 172.31.83.16]
[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-172-31-83-16 localhost] and IPs [172.31.83.16 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-83-16 localhost] and IPs [172.31.83.16 127.0.0.1 ::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
[wait-control-plane] Waiting for the kubelet to boot up the control plane as static Pods from directory "/etc/kubernetes/manifests"
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 501.370499ms
```

To Configure kubectl, Set Up kubeconfig

mkdir -p $HOME/.kube

sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

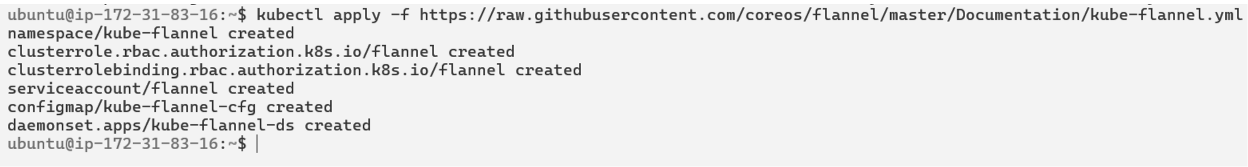
sudo chown $(id -u):$(id -g) $HOME/.kube/config



```
ubuntu@ip-172-31-83-16:~$ mkdir -p $HOME/.kube
ubuntu@ip-172-31-83-16:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
ubuntu@ip-172-31-83-16:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-83-16:~$
```

Install Flannel (a networking plugin):

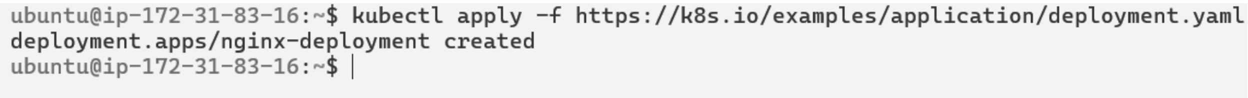
kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml



```
ubuntu@ip-172-31-83-16:~$ 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-172-31-83-16:~$ |
```

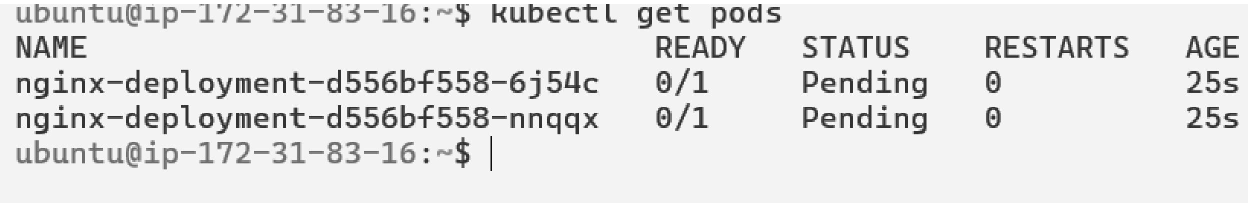
To Deploy Nginx Server, **Create a Deployment:**

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



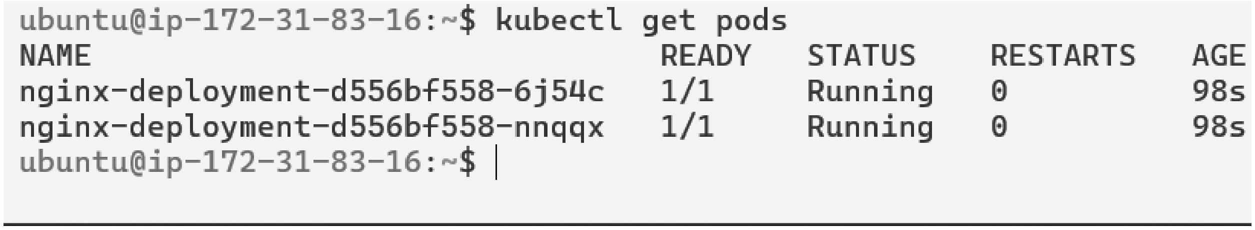
```
ubuntu@ip-172-31-83-16:~$ kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
ubuntu@ip-172-31-83-16:~$ |
```

Check Pods:
kubectl get pods



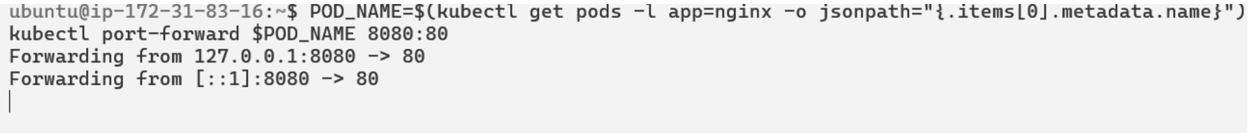
```
ubuntu@ip-172-31-83-16:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-6j54c   0/1     Pending   0          25s
nginx-deployment-d556bf558-nnqqx   0/1     Pending   0          25s
ubuntu@ip-172-31-83-16:~$ |
```

If the pod status is pending, you might need to remove the control-plane taint:
kubectl taint nodes --all node-role.kubernetes.io/control-plane-



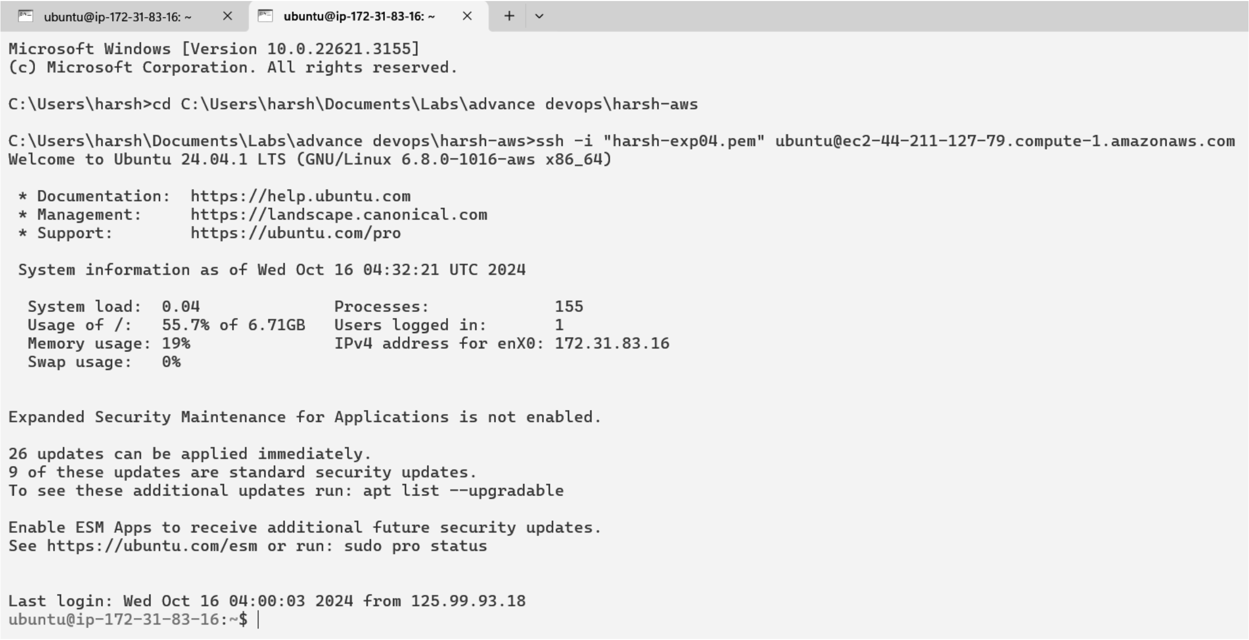
```
ubuntu@ip-172-31-83-16:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-6j54c   1/1     Running   0          98s
nginx-deployment-d556bf558-nnqqx   1/1     Running   0          98s
ubuntu@ip-172-31-83-16:~$ |
```

Port Forward to Access Nginx: Find the Pod name
POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath="{.items[0].metadata.name}")
kubectl port-forward $POD_NAME 8080:80



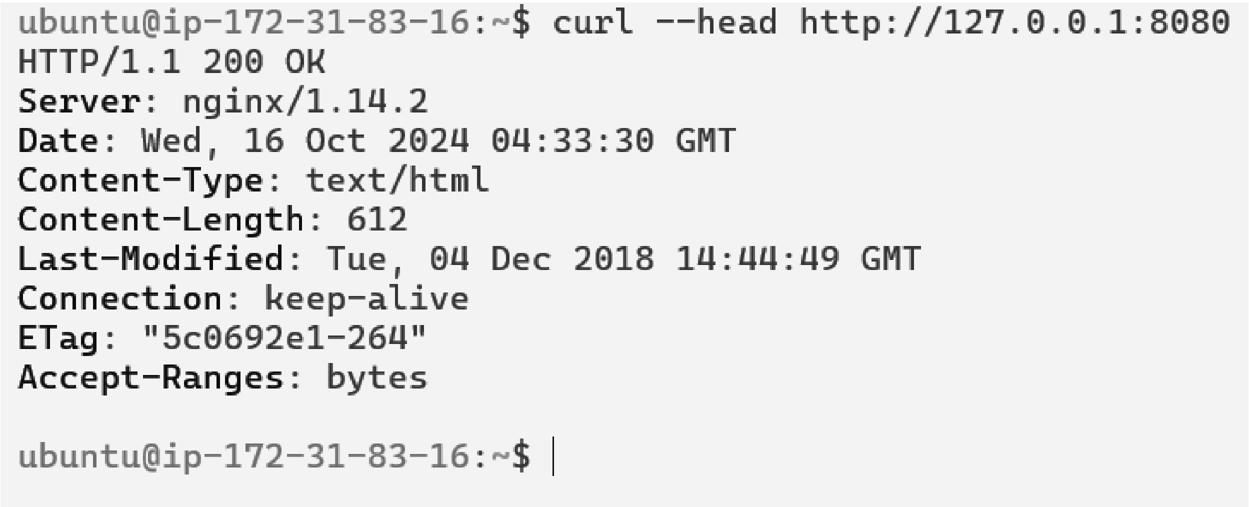
```
ubuntu@ip-172-31-83-16:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath="{.items[0].metadata.name}")
kubectl port-forward $POD_NAME 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
|
```

Open a New Terminal and SSH back into your EC2 instance.



```
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\harsh>cd C:\Users\harsh\Documents\Labs\advance devops\harsh-aws

C:\Users\harsh\Documents\Labs\advance devops\harsh-aws>ssh -i "harsh-exp04.pem" ubuntu@ec2-44-211-127-79.compute-1.amazonaws.com
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1016-aws x86_64)

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

 System information as of Wed Oct 16 04:32:21 UTC 2024

  System load:  0.04              Processes:             155
  Usage of /:   55.7% of 6.71GB   Users logged in:       1
  Memory usage: 19%               IPv4 address for enX0: 172.31.83.16
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

26 updates can be applied immediately.
9 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: Wed Oct 16 04:00:03 2024 from 125.99.93.18
ubuntu@ip-172-31-83-16:~$
```

Use Curl to Check Nginx:
curl --head http://127.0.0.1:8080



```
ubuntu@ip-172-31-83-16:~$ curl --head http://127.0.0.1:8080
HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Wed, 16 Oct 2024 04:33:30 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

ubuntu@ip-172-31-83-16:~$
```

If you see `200 OK`, your Nginx server is successfully running.

Conclusion :

Understanding **kubectl** is crucial for anyone working with Kubernetes, as it serves as the primary interface for managing applications and resources. Through `kubectl`, users can effectively deploy, monitor, and troubleshoot applications, ensuring that they run smoothly in a Kubernetes environment.