

Experiment 4

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

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

Name

experiment4

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

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

Recents

Quick Start

Amazon Linux

aws

macOS

Mac

Ubuntu

ubuntu

Windows

Microsoft

Red Hat

Red Hat

SUSE Li

SUSE

Q

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type

ami-04cdc91e49cb06165 (64-bit (x86)) / ami-02b7539372433cf6b (64-bit (Arm))

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

Free tier eligible ▼

▼ **Instance type** [Info](#) | [Get advice](#)

Instance type

t3.medium

Family: t3 2 vCPU 4 GiB Memory Current generation: true
On-Demand RHEL base pricing: 0.072 USD per Hour
On-Demand Linux base pricing: 0.0432 USD per Hour
On-Demand Windows base pricing: 0.0616 USD per Hour
On-Demand SUSE base pricing: 0.0995 USD per Hour

☒ All generations

[Compare instance types](#)

[Additional costs apply for AMIs with pre-installed software](#)

▼ **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*

worker_key

 [Create new key pair](#)

[EC2](#) > [Instances](#) > Launch an instance

 **Success**

Successfully initiated launch of instance (i-09bc04278935d87f4)

► [Launch log](#)

Connect to instance [Info](#)

Connect to your instance i-09bc04278935d87f4 (experiment4) using any of these options

EC2 Instance Connect



Session Manager

SSH client


EC2 serial console


Instance ID

 i-09bc04278935d87f4 (experiment4)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is worker_key.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
 `chmod 400 "worker_key.pem"`
4. Connect to your instance using its Public DNS:
 `ec2-13-60-30-82.eu-north-1.compute.amazonaws.com`

Example:

 `ssh -i "worker_key.pem" ubuntu@ec2-13-60-30-82.eu-north-1.compute.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 3: Now open the folder in the terminal where our .pem key is stored and paste the Example command

```
PS C:\Users\siddi\downloads> cd exp4
PS C:\Users\siddi\downloads\exp4> ssh -i "worker_key.pem" ubuntu@ec2-13-60-30-82.eu-north-1.compute.amazonaws.com
The authenticity of host 'ec2-13-60-30-82.eu-north-1.compute.amazonaws.com (64:ff9b::d3c:1e52)' can't be established.
ED25519 key fingerprint is SHA256:poqkTnWc7IrjE4zQbuKp8iPtPdLEF+pM5aZMfdtY9Lo.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-60-30-82.eu-north-1.compute.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 Tue Sep 24 18:53:42 UTC 2024

System load:  0.11           Temperature:   -273.1 C
Usage of /:   22.8% of 6.71GB Processes:    112
Memory usage: 5%            Users logged in: 0
Swap usage:   0%            IPv4 address for ens5: 172.31.37.243
```

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 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-172-31-37-243:~$ 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"
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
-----BEGIN PGP PUBLIC KEY BLOCK-----
```

```
mQINBFit2ioBEADhWpZ8/wvZ6hUTiXOWQHxMAlaFHcPH9hAtr4F1y2+OYdbtMuth
lqqwp028AqyY+PRfVMTsYMBjuQuu5byyKR01BbqYhuS3jtqQmLjZ/bJvXqnmVXh
38UuLa+z077PxyxQhu5BbqntTPQMfiyqEiU+BKbq2WmANUKQf+1AmZY/IruOXbnq
L4C1+gJ8vfmXQt99npCaxEjaNRVYfOS8QcixNzHUYnb6emjLANyEVLZzeqo7XKL7
UrwV5inawTSzWNvtjEjj4nJL8NsLwscplPQUhTQ+7BbQXAwAmeHCUTQIvvWXqw0N
cmhh4HgeQscQHYgOJjjDVfoY5MucvglbIgCqfzAHW9jxmRL4qbMZj+b1XoePEtht
ku4bIQN1X5P07fNWzlgARL5Z4POXDDZTLIQ/El58j9kp4bnWRCJW0lya+f8ocodo
vZZ+Doi+fy4D5ZGrL4XEcIQP/Lv5uFyf+kQtL/94VFYVJ0leAv8W92KdgDkhTcTD
G7c0tIkVEKNUq48b3aQ64N0ZQW7fVjfoKwEZd0qPE72Pa45jrZzvUFxSpdiNk2tZ
XYukHjLxxEgBdC/J3cMMNRE1F4NCA3ApfV1Y7/hTeOnmDuDYwr9/obA8t016Yljj
```

```
on-en [2808 B]
Get:51 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Com
ponents [208 B]
Get:52 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n
-f Metadata [344 B]
Fetched 29.1 MB in 5s (5866 kB/s)
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is st
ored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATI
ON section in apt-key(8) for details.
ubuntu@ip-172-31-37-243:~$ |
```

```
sudo apt-get update
sudo apt-get install -y docker-ce
```

```
ubuntu@ip-172-31-37-243:~$ sudo apt-get update
sudo apt-get install -y docker-ce
Hit:1 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates InReleas
e
Hit:3 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRele
ase
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 st
ored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATI
ON section in apt-key(8) for details.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

```
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 → /us
r/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.
ubuntu@ip-172-31-37-243:~$ |
```

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

```
ubuntu@ip-172-31-37-243:~$ sudo mkdir -p /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json
{
"exec-opts": ["native.cgroupdriver=systemd"]
}
EOF
{
"exec-opts": ["native.cgroupdriver=systemd"]
}
ubuntu@ip-172-31-37-243:~$ |
```

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

```
ubuntu@ip-172-31-37-243:~$ sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart 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-172-31-37-243:~$ |
```

Step 5: Run the below command to install Kubernets.

```
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-172-31-37-243:~$ 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
gpg: missing argument for option "-o"
-bash: /etc/apt/keyrings/kubernetes-apt-keyring.gpg: No such file or directory
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /
ubuntu@ip-172-31-37-243:~$ |
```

Error:

```
ubuntu@ip-172-31-37-243:~$ sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
E: Malformed entry 1 in list file /etc/apt/sources.list.d/kubernetes.list (URI)
E: The list of sources could not be read.
E: Malformed entry 1 in list file /etc/apt/sources.list.d/kubernetes.list (URI)
E: The list of sources could not be read.
E: Malformed entry 1 in list file /etc/apt/sources.list.d/kubernetes.list (URI)
E: The list of sources could not be read.
```

```

ubuntu@ip-172-31-37-243:~$ sudo mkdir -p /etc/apt/keyrings
ubuntu@ip-172-31-37-243:~$ 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
gpg: missing argument for option "-o"
-bash: /etc/apt/keyrings/kubernetes-apt-keyring.gpg: No such file or directory
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /
ubuntu@ip-172-31-37-243:~$ sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
E: Malformed entry 1 in list file /etc/apt/sources.list.d/kubernetes.list (URI)
E: The list of sources could not be read.
E: Malformed entry 1 in list file /etc/apt/sources.list.d/kubernetes.list (URI)
E: The list of sources could not be read.
E: Malformed entry 1 in list file /etc/apt/sources.list.d/kubernetes.list (URI)
E: The list of sources could not be read.
ubuntu@ip-172-31-37-243:~$ sudo mkdir -p /etc/apt/keyrings
ubuntu@ip-172-31-37-243:~$ 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-172-31-37-243:~$ 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-172-31-37-243:~$ sudo apt-get update
Hit:1 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease

```



```

ubuntu@ip-172-31-37-243:~$ 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 139 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://eu-north-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 (78.5 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.
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.
ubuntu@ip-172-31-37-243:~$ sudo apt-mark hold kubelet kubeadm kubectl
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
ubuntu@ip-172-31-37-243:~$ |

```

```

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

```



```

ubuntu@ip-172-31-37-243:~$ sudo systemctl enable --now kubelet
sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
W0924 19:06:55.141347 4241 checks.go:1080] [preflight] WARNING: Couldn't
create the interface used for talking to the container runtime: failed to cr
eate new CRI runtime service: validate service connection: validate CRI v1 r
untime API for endpoint "unix:///var/run/containerd/containerd.sock": rpc er
ror: 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 confi
g images pull'
error execution phase preflight: [preflight] Some fatal errors occurred:
failed to create new CRI runtime service: validate service connection: valid
ate CRI v1 runtime API for endpoint "unix:///var/run/containerd/containerd.s
ock": rpc error: code = Unimplemented desc = unknown service runtime.v1.Runt
imeService[preflight] If you know what you are doing, you can make a check n
on-fatal with '--ignore-preflight-errors=...'
To see the stack trace of this error execute with --v=5 or higher
ubuntu@ip-172-31-37-243:~$ |

```

sudo apt-get install -y containerd

```

ubuntu@ip-172-31-37-243:~$ 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 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 139 not upgraded.
Need to get 47.2 MB of archives.
After this operation, 53.1 MB disk space will be freed.
Get:1 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.1.12-0ubuntu3.1 [8599 kB]
Get:2 http://eu-north-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 (81.0 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...

```

```
sudo mkdir -p /etc/containerd
```

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

```
ubuntu@ip-172-31-37-243:~$ 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

[timeouts]
  "io.containerd.timeout.bolt.open" = "0s"
  "io.containerd.timeout.metrics.shimstats" = "2s"
  "io.containerd.timeout.shim.cleanup" = "5s"
  "io.containerd.timeout.shim.load" = "5s"
  "io.containerd.timeout.shim.shutdown" = "3s"
  "io.containerd.timeout.task.state" = "2s"

[ttrpc]
  address = ""
  gid = 0
  uid = 0
ubuntu@ip-172-31-37-243:~$
```

```
sudo systemctl restart containerd
```

```
sudo systemctl enable containerd
```

```
sudo systemctl status containerd
```

```

ubuntu@ip-172-31-37-243:~$ 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; p>
   Active: active (running) since Tue 2024-09-24 19:09:13 UTC; 325ms ago
     Docs: https://containerd.io
   Main PID: 4718 (containerd)
      Tasks: 8
    Memory: 13.2M (peak: 13.8M)
       CPU: 73ms
    CGroup: /system.slice/containerd.service
            └─4718 /usr/bin/containerd

Sep 24 19:09:13 ip-172-31-37-243 containerd[4718]: time="2024-09-24T19:09:13"
Sep 24 19:09:13 ip-172-31-37-243 containerd[4718]: time="2024-09-24T19:09:13"
Sep 24 19:09:13 ip-172-31-37-243 containerd[4718]: time="2024-09-24T19:09:13"
Sep 24 19:09:13 ip-172-31-37-243 containerd[4718]: time="2024-09-24T19:09:13"
Sep 24 19:09:13 ip-172-31-37-243 containerd[4718]: time="2024-09-24T19:09:13"
Sep 24 19:09:13 ip-172-31-37-243 containerd[4718]: time="2024-09-24T19:09:13"
Sep 24 19:09:13 ip-172-31-37-243 containerd[4718]: time="2024-09-24T19:09:13"
Sep 24 19:09:13 ip-172-31-37-243 containerd[4718]: time="2024-09-24T19:09:13"
Sep 24 19:09:13 ip-172-31-37-243 systemd[1]: Started containerd.service - c
Sep 24 19:09:13 ip-172-31-37-243 containerd[4718]: time="2024-09-24T19:09:13"
lines 1-21/21 (END)
● containerd.service - containerd container runtime
   Loaded: loaded (/usr/lib/systemd/system/containerd.service; enabled; preset: enabled)
   Active: active (running) since Tue 2024-09-24 19:09:13 UTC; 325ms ago
     Docs: https://containerd.io
   Main PID: 4718 (containerd)
      Tasks: 8
    Memory: 13.2M (peak: 13.8M)
       CPU: 73ms
    CGroup: /system.slice/containerd.service
            └─4718 /usr/bin/containerd

```

sudo apt-get install -y socat

```

ubuntu@ip-172-31-37-243:~$ 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 libltdl7 libsllp0 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 139 not upgraded.
Need to get 374 kB of archives.
After this operation, 1649 kB of additional disk space will be used.
Get:1 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 socat amd64 1.8.0-4build3 [374 kB]
Fetched 374 kB in 0s (22.1 MB/s)
Selecting previously unselected package socat.
(Reading database ... 68108 files and directories currently installed.)
Preparing to unpack .../socat_1.8.0-4build3_amd64.deb ...
Unpacking socat (1.8.0-4build3) ...
Setting up socat (1.8.0-4build3) ...
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-172-31-37-243:~$ |

```

Step 6: Initialize the Kubecluster

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

```
ubuntu@ip-172-31-37-243:~$ 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 confi
g images pull'
W0924 19:12:15.863783    5009 checks.go:846] detected that the sandbox image
"registry.k8s.io/pause:3.8" of the container runtime is inconsistent with t
hat 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-172-31-37-243 kub
ernetes.kubernetes.default.kubernetes.default.svc.kubernetes.default.svc.clu
ster.local] and IPs [10.96.0.1 172.31.37.243]
[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-172-31-37-243 l
ocalhost] and IPs [172.31.37.243 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-37-243 loc
alhost] and IPs [172.31.37.243 127.0.0.1 ::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/manif
ests"
```

```

Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user
:

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

Alternatively, if you are the root user, you can run:

export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 172.31.37.243:6443 --token 83t146.gii8h4xivxue1bio \
--discovery-token-ca-cert-hash sha256:d3833aa042f888a0a506ff97a41023c5524cd0e0b533ba00adb635a5eff723d9
ubuntu@ip-172-31-37-243:~$ |

```

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

```

```

ubuntu@ip-172-31-37-243:~$ mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-37-243:~$ |

```

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

```

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

```

```

ubuntu@ip-172-31-37-243:~$ 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-172-31-37-243:~$ |

```

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

```

```
ubuntu@ip-172-31-37-243:~$ kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
ubuntu@ip-172-31-37-243:~$ |
```

kubectl get pods

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

kubectl port-forward \$POD_NAME 8081:80

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

untainted

kubectl get nodes

```
ubuntu@ip-172-31-37-243:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-ptclv    0/1     Pending   0           55s
nginx-deployment-d556bf558-rsgxw    0/1     Pending   0           55s
ubuntu@ip-172-31-37-243:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath="{.items[0].metadata.name}")
ubuntu@ip-172-31-37-243:~$ kubectl port-forward $POD_NAME 8080:80
error: unable to forward port because pod is not running. Current status=Pending
ubuntu@ip-172-31-37-243:~$ 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-172-31-37-243:~$ kubectl get nodes
NAME                                STATUS   ROLES    AGE   VERSION
ip-172-31-37-243                    Ready    control-plane   10m   v1.31.1
ubuntu@ip-172-31-37-243:~$ kubectl taint nodes --all node-role.kubernetes.io/control-plane-node/ip-172-31-37-243 untainted
ubuntu@ip-172-31-37-243:~$ kubectl get nodes
NAME                                STATUS   ROLES    AGE   VERSION
ip-172-31-37-243                    Ready    control-plane   10m   v1.31.1
ubuntu@ip-172-31-37-243:~$ |
```

```
ubuntu@ip-172-31-37-243:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-ptclv    1/1     Running   0           8m32s
nginx-deployment-d556bf558-rsgxw    1/1     Running   0           8m32s
ubuntu@ip-172-31-37-243:~$ |
```

```
ubuntu@ip-172-31-37-243:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath="{.items[0].metadata.name}")
```

```
ubuntu@ip-172-31-37-243:~$ kubectl port-forward $POD_NAME 8081:80
Forwarding from 127.0.0.1:8081 -> 80
Forwarding from [::1]:8081 -> 80
Forwarding from 127.0.0.1:8081 -> 80
Forwarding from [::1]:8081 -> 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

```
PS C:\Users\siddi\downloads\exp4> ssh -i "worker_key.pem" ubuntu@ec2-13-60-30-82.eu-north-1.compute.amazonaws.com
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 Tue Sep 24 19:52:47 UTC 2024

System load:  0.01               Temperature:   -273.1 C
Usage of /:   55.5% of 6.71GB    Processes:    149
Memory usage: 20%               Users logged in: 1
Swap usage:   0%                IPv4 address for ens5: 172.31.37.243

Expanded Security Maintenance for Applications is not enabled.

143 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: Tue Sep 24 19:41:15 2024 from 152.57.249.67
ubuntu@ip-172-31-37-243:~$ curl --head http://127.0.0.1:8080
HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Tue, 24 Sep 2024 19:53:04 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-172-31-37-243:~$ |
```