

Name:Sujal.S.Tekwani  
Class:D15B  
Roll no:63

## Adv DevOps Lab3

The screenshot shows the 'Launch an instance' wizard in the AWS EC2 console. The summary section indicates 3 instances will be launched. The software image (AMI) selected is Canonical, Ubuntu, 24.04, amd64, ami-0866a3c8d6eae9ba. The virtual server type (instance type) is t2.medium, and the firewall (security group) is exp3. Storage (volumes) is set to 1 volume(s) - 8 GiB. A tooltip for the Free tier is displayed, stating it includes 750 hours of t2.micro (or t3.micro in regions where t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, and 30 GiB of EBS storage, 2 million IOPS. The 'Launch instance' button is highlighted.

The screenshot shows the 'Create security group' wizard in the AWS EC2 console. The basic details section includes a security group name (exp3), a description (exp3), and a VPC (vpc-064c555fd93118b55). The inbound rules section shows two entries: one for All traffic (Protocol: All, Port range: All, Source: Custom, Description: sg-0e9a2293c22b74437) and another for SSH (Protocol: TCP, Port range: 22, Source: Anywhere-..., Description: sg-0e9a2293c22b74437). The 'CloudShell' and 'Feedback' buttons are at the bottom.

AWS CloudWatch Metrics Dashboard showing metrics for EC2 instances. The dashboard includes a search bar, a legend for metric types (CloudWatch Metrics, CloudWatch Metrics Insights, CloudWatch Metrics Metrics Insights), and a table of metrics. Metrics include CPU Utilization, Network In, Network Out, and Memory Utilization.

Metric	Value	Unit	Period
CloudWatch Metrics CPU Utilization	~0.00%	%	1 minute
CloudWatch Metrics Network In	~0.00 B/s	B/s	1 minute
CloudWatch Metrics Network Out	~0.00 B/s	B/s	1 minute
CloudWatch Metrics Metrics Insights CPU Utilization	~0.00%	%	1 minute
CloudWatch Metrics Metrics Insights Network In	~0.00 B/s	B/s	1 minute
CloudWatch Metrics Metrics Insights Network Out	~0.00 B/s	B/s	1 minute
CloudWatch Metrics Metrics Insights Memory Utilization	~0.00%	%	1 minute

AWS CloudShell session showing the AWS CLI command `aws ec2 describe-instances --instance-ids i-0b0fe5ee9bfbcc316d` running on an Ubuntu 22.04 LTS instance. The output shows the instance details, including its public IP address (3.87.219.157) and private IP address (172.31.81.74).

```
ubuntu@ip-172-31-93-58:~$ aws ec2 describe-instances --instance-ids i-0b0fe5ee9bfbcc316d
{
    "Reservations": [
        {
            "Instances": [
                {
                    "InstanceId": "i-0b0fe5ee9bfbcc316d",
                    "InstanceType": "t2.medium",
                    "InstanceState": "running",
                    "PublicIpAddress": "3.87.219.157",
                    "PrivateIpAddress": "172.31.81.74"
                }
            ]
        }
    ]
}
```

The terminal also displays a message from the system about available updates:

```
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
```

```

root@ip-172-31-92-66:/home/ubuntu
ubuntu@ip-172-31-92-66:~$ Read from remote host 54.210.246.214: Connection reset by peer
Connection to 54.210.246.214 closed.
client_loop: send disconnect: Connection reset by peer
j
sujal@LAPTOP-2UFUKR2C MINGW64 ~/Downloads (user)
$ sudo su
bash: sudo: command not found

sujal@LAPTOP-2UFUKR2C MINGW64 ~/Downloads (user)
$ ssh -i "master.pem" ubuntu@54.210.246.214
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 Fri Oct 11 11:20:14 UTC 2024

System load: 0.0          Processes:           117
Usage of /: 23.1% of 6.71GB  Users logged in:      1
Memory usage: 5%          IPv4 address for enX0: 172.31.92.66
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: Fri Oct 11 11:14:29 2024 from 110.224.118.109
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-92-66:~$ sudo su
root@ip-172-31-92-66:/home/ubuntu# 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 a pt-key(8)).
OK
root@ip-172-31-92-66:/home/ubuntu# 

root@ip-172-31-92-66:~$ 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 a pt-key(8)).
OK
root@ip-172-31-92-66:/home/ubuntu# 

```

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

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

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

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

ubuntu@ip-172-31-93-58:~\$ sudo su
root@ip-172-31-93-58:/home/ubuntu# 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 a pt-key(8)).
OK
root@ip-172-31-93-58:/home/ubuntu#

```

root@ip-172-31-92-66:/home/ubuntu
Unpacking docker-compose-plugin (2.29.7-1~ubuntu.24.04-nobie) ...
Selecting previously unselected package libltdl7:amd64.
Preparing to unpack .../libltdl7_2.4.7-1~ubuntu.24.04-nobie.deb ...
Unpacking libltdl7:amd64 (2.4.7-1~ubuntu.24.04-nobie) ...
Selecting previously unselected package libslirp0:amd64.
Preparing to unpack .../8-libslirp0_4.7.0~ubuntu3_amd64.deb ...
Unpacking libslirp0:amd64 (4.7.0~ubuntu3) ...
Selecting previously unselected package slirp4netns.
Preparing to unpack .../9-slirp4netns_1.2.1-1build2_amd64.deb ...
Unpacking slirp4netns (1.2.1-1build2) ...
Setting up docker-buildx-plugin (0.17.1-1~ubuntu.24.04-nobie) ...
Setting up containerd.io (1.7.22-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/lib/systemd/system/containerd.service.
Setting up docker-compose-plugin (2.29.7-1~ubuntu.24.04-nobie) ...
Setting up libltdl7:amd64 (2.4.7-1~build1) ...
Setting up docker-ce-clnt (5:27.3.1-1~ubuntu.24.04-nobie) ...
Setting up libltdl7:amd64 (4.7.0~ubuntu3) ...
Setting up libltdl7:amd64 (2.4.7-1~build2) ...
Setting up docker-ce-rootless-extras (5:27.3.1-1~ubuntu.24.04-nobie) ...
Setting up slirp4netns (1.2.1-1build2) ...
Setting up docker-ce (5:27.3.1-1~ubuntu.24.04-nobie) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/lib/systemd/system/docker.socket.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
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-172-31-93-58:~$ echo sujal
sujal
ubuntu@ip-172-31-93-58:~$ 

ubuntu@ip-172-31-93-58:~$ 

```

Setting up docker-ce (5:27.3.1-1~ubuntu.24.04-nobie) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/lib/systemd/system/docker.socket.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
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-172-31-81-74:~\$ echo sujal
sujal
ubuntu@ip-172-31-81-74:~\$

```

Starting up libltdl17:amd64 (2.4.7-7build1) ...
Starting up docker-ce-cli (5:27.3.1-1~ubuntu.24.04-noble) ...
Starting up libslirp0:amd64 (4.7.0-1ubuntu3) ...
Starting up pigz (2.8-1) ...
Starting up docker-ce-rootless-extras (5:27.3.1-1~ubuntu.24.04-noble) ...
Starting up slirp4nets (1.2.1-1~build2) ...
Starting up docker-ce (5:27.3.1-1~ubuntu.24.04-noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
Managing processes...
Managing Linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-92-66:/home/ubuntu# echo sujal
sujal
root@ip-172-31-92-66:/home/ubuntu# echo sujal
sujal
root@ip-172-31-92-66:/home/ubuntu# echo sujal
sujal
root@ip-172-31-92-66:/home/ubuntu# sudo mkdir -p /etc/docker
cat <>EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}
EOF
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}

root@ip-172-31-92-66:/home/ubuntu# sudo systemctl enable docker
|do systemctl daemon-reload
|do 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
root@ip-172-31-92-66:/home/ubuntu# |

```

No user sessions are running outdated binaries.  
 No VM guests are running outdated hypervisor (qemu) binaries on this host.  
 ubuntu@ip-172-31-93-58:~\$ echo sujal  
 sujal  
 ubuntu@ip-172-31-93-58:~\$ 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-172-31-93-58:~\$ sudo systemctl enable docker  
|do systemctl daemon-reload
|do 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-172-31-93-58:~\$ |

No user sessions are running outdated binaries.  
 No VM guests are running outdated hypervisor (qemu) binaries on this host.  
 ubuntu@ip-172-31-81-74:~\$ echo sujal  
 sujal  
 ubuntu@ip-172-31-81-74:~\$ 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-172-31-81-74:~\$ sudo systemctl enable docker  
|do systemctl daemon-reload
|do 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-172-31-81-74:~\$ |

```

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-92-66:/home/ubuntu# echo sujal
sujal
root@ip-172-31-92-66:/home/ubuntu# echo sujal
sujal
root@ip-172-31-92-66:/home/ubuntu# echo sujal
sujal
root@ip-172-31-92-66:/home/ubuntu# sudo mkdir -p /etc/docker
cat <>EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}
EOF
{
  "exec-opts": ["native.cgroupdriver=systemd"]
}

root@ip-172-31-92-66:/home/ubuntu# sudo systemctl enable docker
|do systemctl daemon-reload
|do 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
root@ip-172-31-92-66:/home/ubuntu# curl -fsSL https://pkgs.k8s.io/core/stable:v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
ubuntu@ip-172-31-93-58:~$ echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]' > /etc/apt/sources.list.d/kubernetes.list
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/
root@ip-172-31-92-66:/home/ubuntu# |

```

EOF  
{  
 "exec-opts": ["native.cgroupdriver=systemd"]
}  
ubuntu@ip-172-31-93-58:~\$ sudo systemctl enable docker  
|do systemctl daemon-reload
|do 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-172-31-93-58:~\$ |

EOF  
{  
 "exec-opts": ["native.cgroupdriver=systemd"]
}  
ubuntu@ip-172-31-81-74:~\$ sudo systemctl enable docker  
|do systemctl daemon-reload
|do 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-172-31-81-74:~\$ |

EOF  
{  
 "exec-opts": ["native.cgroupdriver=systemd"]
}  
ubuntu@ip-172-31-81-74:~\$ sudo mkdir -p /etc/apt/keyrings
ubuntu@ip-172-31-93-58:~\$ 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-172-31-93-58:~\$ echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]' > /etc/apt/sources.list.d/kubernetes.list
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/
root@ip-172-31-92-66:/home/ubuntu# |

```

1.31/deb kubernetes-cni 1.5.1-1.1 [33.9 MB]
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes/core/stable:/v
1.31/deb kubelet 1.31.1-1.1 [15.2 MB]
Fetched 87.4 MB in 1s (84.8 MB/s)
Selecting previously unselected package conntrack.
(Reading database ... 68102 files and directories currently installed.)
Preparing to unpack .../0-conntrack_1%3a1.4.8-ubuntu1_amd64.deb ...
Unpacking conntrack (1:1.4.8-ubuntu1) ...
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 kubelet.
Preparing to unpack .../3-kubelet_1.31.1-1.1_amd64.deb ...
Unpacking kubelet (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-ubuntu1) ...
Setting up kubelet (1.31.1-1.1) ...
Setting up cri-tools (1.31.1-1.1) ...
Setting up kubernetes-cni (1.5.1-1.1) ...
Setting up kubeadm (1.31.1-1.1) ...
Setting up kubelet (1.31.1-1.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-92-66:/home/ubuntu# sudo apt-mark hold kubelet kubeadm kubectl
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
ubuntu@ip-172-31-93-58:~ |
```

```

endpoint = ""
insecure = false
protocol = ""

[[plugins."io.containerd.transfer.v1.local"]
config_path = ""
max_concurrent_downloads = 3
max_concurrent_uploaded_layers = 3

[[plugins."io.containerd.transfer.v1.local".unpack_config]]
differ = ""
platform = "linux/amd64"
snapshotter = "overlayfs"

[proxy_plugins]

[stream_processors]

[[stream_processors."io.containerd.ocicrypt.decoder.v1.tar"]]
accepts = ["application/vnd.oci.image.layer.v1.tar+gzip"]
args = ["--decryption-keys-path", "/etc/containerd/ocicrypt/keys"]
env = ["OCICRYPT_KEYPROVIDER_CONFIG=/etc/containerd/ocicrypt/ocicrypt_keyprovide
r.conf"]
path = "ctd-decoder"
returns = "application/vnd.oci.image.layer.v1.tar"

[[stream_processors."io.containerd.ocicrypt.decoder.v1.tar.gzip"]]
accepts = ["application/vnd.oci.image.layer.v1.tar+gzip+encrypted"]
args = ["--decryption-keys-path", "/etc/containerd/ocicrypt/keys"]
env = ["OCICRYPT_KEYPROVIDER_CONFIG=/etc/containerd/ocicrypt/ocicrypt_keyprovide
r.conf"]
path = "ctd-decoder"
returns = "application/vnd.oci.image.layer.v1.tar+gzip"

[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-172-31-93-58:~ |
```

```

returns = "application/vnd.oci.image.layer.v1.tar"
[stream_processors, "io.containerd.ocicrypt.decoder.v1.tar.gzip"]
accepts = ["application/vnd.oci.image.layer.v1.tar+gzip+encrypted"]
env = ["OCICRYPT_KEYPROVIDER_CONFIG=/etc/containerd/ocicrypt/keys"]
path = "ctd-decoder"
returns = "application/vnd.oci.image.layer.v1.tar+gzip"

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

ttrpc]
address = ""
gid = 0
uid = 0
ostripip=172-31-92-66:/home/ubuntu# sudo systemctl restart containerd
udo systemctl enable containerd
udo systemctl status containerd
containerd.service - containerd container runtime
   Loaded: loaded (/usr/lib/systemd/system/containerd.service; enabled; preset: en
     Active: active (running) since Fri 2024-10-11 11:45:29 UTC; 215ms ago
       Docs: https://containerd.io
    Main PID: 4699 (containerd)
      Tasks: 8
     Memory: 13.9M (peak: 14.6M)
        CPU: 52ms
       CGroup: /system.slice/containerd.service
           └─4699 /usr/bin/containerd

Oct 11 11:45:33 ip-172-31-93-58 containerd[4644]: time="2024-10-11T11:45:33.2995009"
Oct 11 11:45:33 ip-172-31-93-58 containerd[4644]: time="2024-10-11T11:45:33.2995285"
Oct 11 11:45:33 ip-172-31-93-58 containerd[4644]: time="2024-10-11T11:45:33.2995927"
Oct 11 11:45:33 ip-172-31-93-58 containerd[4644]: time="2024-10-11T11:45:33.2996566"
Oct 11 11:45:33 ip-172-31-93-58 containerd[4644]: time="2024-10-11T11:45:33.2996749"
Oct 11 11:45:33 ip-172-31-93-58 containerd[4644]: time="2024-10-11T11:45:33.2996836"
Oct 11 11:45:33 ip-172-31-93-58 containerd[4644]: time="2024-10-11T11:45:33.2996905"
Oct 11 11:45:33 ip-172-31-93-58 containerd[4644]: time="2024-10-11T11:45:33.2997395"
Oct 11 11:45:33 ip-172-31-93-58 containerd[4644]: time="2024-10-11T11:45:33.2997449"
lines 1-21

ubuntu@ip-172-31-81-74: ~
● containerd.service - containerd container runtime
   Loaded: loaded (/usr/lib/systemd/system/containerd.service; enabled; preset: en
     Active: active (running) since Fri 2024-10-11 11:45:36 UTC; 211ms ago
       Docs: https://containerd.io
    Main PID: 4558 (containerd)
      Tasks: 7
     Memory: 13.5M (peak: 14.3M)
        CPU: 65ms
       CGroup: /system.slice/containerd.service
           └─4558 /usr/bin/containerd

Oct 11 11:45:36 ip-172-31-81-74 containerd[4558]: time="2024-10-11T11:45:36.5166178"
Oct 11 11:45:36 ip-172-31-81-74 containerd[4558]: time="2024-10-11T11:45:36.5166468"
Oct 11 11:45:36 ip-172-31-81-74 containerd[4558]: time="2024-10-11T11:45:36.5166931"
Oct 11 11:45:36 ip-172-31-81-74 containerd[4558]: time="2024-10-11T11:45:36.5167016"
Oct 11 11:45:36 ip-172-31-81-74 containerd[4558]: time="2024-10-11T11:45:36.5167093"
Oct 11 11:45:36 ip-172-31-81-74 containerd[4558]: time="2024-10-11T11:45:36.5167156"
Oct 11 11:45:36 ip-172-31-81-74 containerd[4558]: time="2024-10-11T11:45:36.5168334"
Oct 11 11:45:36 ip-172-31-81-74 containerd[4558]: time="2024-10-11T11:45:36.5168666"
Oct 11 11:45:36 ip-172-31-81-74 containerd[4558]: time="2024-10-11T11:45:36.5184964"
Oct 11 11:45:36 ip-172-31-81-74 containerd[4558]: time="2024-10-11T11:45:36.5184964"
Oct 11 11:45:36 ip-172-31-81-74 systemd[1]: Started containerd.service - containerd
lines 1-21

```

```

[aintains [node-role.kubernetes.io/control-plane:NoSchedule]
[bootstrap-token] Using token: 5djwwc.5kfpqrqql9xq7e1
[bootstrap-token] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to get nodes
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to post CSRs
in order for nodes to get long term certificate credentials
[bootstrap-token] Configured RBAC rules to allow the csapprover controller automatically
approve CSRs from a Node Bootstrap Token
[bootstrap-token] Configured RBAC rules to allow certificate rotation for all node client
certificates in the cluster
[bootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client
certificate and key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

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:
root@ip-172-31-92-66:/home/ubuntu#
  kubeadm join 172.31.92.66:6443 --token 5djwwc.5kfpqrqql9xq7e1 \
    --discovery-token-ca-cert-hash sha256:e354581053520065bef022a9afcafde5d65421
786182911057643fc05cb3c7b8
root@ip-172-31-92-66:/home/ubuntu#
  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config
root@ip-172-31-92-66:/home/ubuntu# kubectl get nodes
NAME          STATUS    ROLES      AGE     VERSION
ip-172-31-92-66  NotReady  control-plane  74s   v1.31.1
root@ip-172-31-92-66:/home/ubuntu# |

```

```

sujal
root@ip-172-31-92-66:/home/ubuntu# curl -k https://172.31.92.66:6443
{
  "kind": "Status",
  "apiVersion": "v1",
  "metadata": {},
  "status": "Failure",
  "message": "forbidden: User \\"system:anonymous\\" cannot get path \"/\"",
  "reason": "Forbidden",
  "details": {},
  "code": 403
}root@ip-172-31-92-66:/home/ubuntu#sudo ufw allow 6443/tcp
sudo ufw reload
Rules updated
Rules updated (v6)
Firewall not enabled (skipping reload)
root@ip-172-31-92-66:/home/ubuntu# sudo systemctl status kubelet
● kubelet.service - kubelet: The Kubernetes Node Agent
   Loaded: loaded (/usr/lib/systemd/system/kubelet.service; enabled; preset: enabled)
   Drop-In: /usr/lib/systemd/system/kubelet.service.d
             └─10-kubeadm.conf
     Active: active (running) since Fri 2024-10-11 11:46:52 UTC; 10min ago
       Docs: https://kubernetes.io/docs/
   Main PID: 5529 (kubelet)
     Tasks: 10 (limit: 4676)
    Memory: 31.7M (peak: 33.1M)
      CPU: 9.142s
     CGroup: /system.slice/kubelet.service
             └─5529 /usr/bin/kubelet --bootstrap-kubeconfig=/etc/kubernetes/bootstrap...
Oct 11 11:56:48 ip-172-31-92-66 kubelet[5529]: E1011 11:56:48.547919      5529 pod_wor...
Oct 11 11:56:52 ip-172-31-92-66 kubelet[5529]: E1011 11:56:52.690831      5529 kubele...
Oct 11 11:56:57 ip-172-31-92-66 kubelet[5529]: E1011 11:56:57.691591      5529 kubele...
Oct 11 11:56:59 ip-172-31-92-66 kubelet[5529]: E1011 11:56:59.501636      5529 scope.>...
Oct 11 11:56:59 ip-172-31-92-66 kubelet[5529]: E1011 11:56:59.501746      5529 pod_wor...
Oct 11 11:57:02 ip-172-31-92-66 kubelet[5529]: E1011 11:57:02.692854      5529 kubele...
Oct 11 11:57:07 ip-172-31-92-66 kubelet[5529]: E1011 11:57:07.694345      5529 kubele...
Oct 11 11:57:12 ip-172-31-92-66 kubelet[5529]: E1011 11:57:12.695210      5529 kubele...
Oct 11 11:57:14 ip-172-31-92-66 kubelet[5529]: E1011 11:57:14.496757      5529 scope.>...
Oct 11 11:57:14 ip-172-31-92-66 kubelet[5529]: E1011 11:57:14.496872      5529 pod_wor...
root@ip-172-31-92-66:/home/ubuntu# sudo systemctl restart kubelet
root@ip-172-31-92-66:/home/ubuntu# kubectl get nodes
NAME           STATUS    ROLES          AGE   VERSION
ip-172-31-81-74  NotReady <none>        24s   v1.31.1
ip-172-31-92-66  NotReady control-plane  12m   v1.31.1
ip-172-31-93-58  NotReady <none>        26s   v1.31.1
root@ip-172-31-92-66:/home/ubuntu#

```

```

pxrfqb19xq7e] --discovery-token-ca-cert-hash sha256:e354581053520065bef022a9
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get c
m/kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubele
t/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. Th
is can take up to 4m0s
[kubelet-check] The kubelet is healthy after 502.317008ms
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap
This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
root@ip-172-31-93-58:/home/ubuntu# 

pxrfqb19xq7e] --discovery-token-ca-cert-hash sha256:e354581053520065bef022a9
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get c
m/kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubele
t/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. Th
is can take up to 4m0s
[kubelet-check] The kubelet is healthy after 505.070054ms
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap
This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
root@ip-172-31-81-74:/home/ubuntu# 

pxrfqb19xq7e] --discovery-token-ca-cert-hash sha256:e354581053520065bef022a9
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get c
m/kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubele
t/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. Th
is can take up to 4m0s
[kubelet-check] The kubelet is healthy after 505.070054ms
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap
This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
root@ip-172-31-81-74:/home/ubuntu# 

```

