

Name - Mitil Lalwani

D15B/37

Advance DevOps Lab

Experiment 3

Aim -

To understand the Kubernetes Cluster Architecture, install and spin up a Kubernetes cluster on linux machines / cloud platforms.

Theory -

Kubernetes is a software that automatically manages, scales and maintains multi-container workloads in desired states.

Modern software is increasingly run as fleets of containers called as microservices.

Features of Kubernetes:

- 1) Standard services like local DNS and basic load balancing that most applications need, and are easy to use.
- 2) Standard behaviors (restart container if it dies) that are easy to invoke and do most of the work to keep containers running, available and performant.
- 3) A standard API that applications can call to easily enable more sophisticated behaviors, making it much easier to create applications that manage other applications.

Prerequisites

2 or more Linux servers running Ubuntu 18.04 /20.04 on Virtual box or you can use EC2 free tier instances choose the ubuntu 20.04 AMI free tier

Access to a user account on each system with sudo or root privileges

The apt package manager, included by default

Command-line/terminal window (Ctrl-Alt-T)

Steps to Install Kubernetes on Ubuntu

Set up Docker

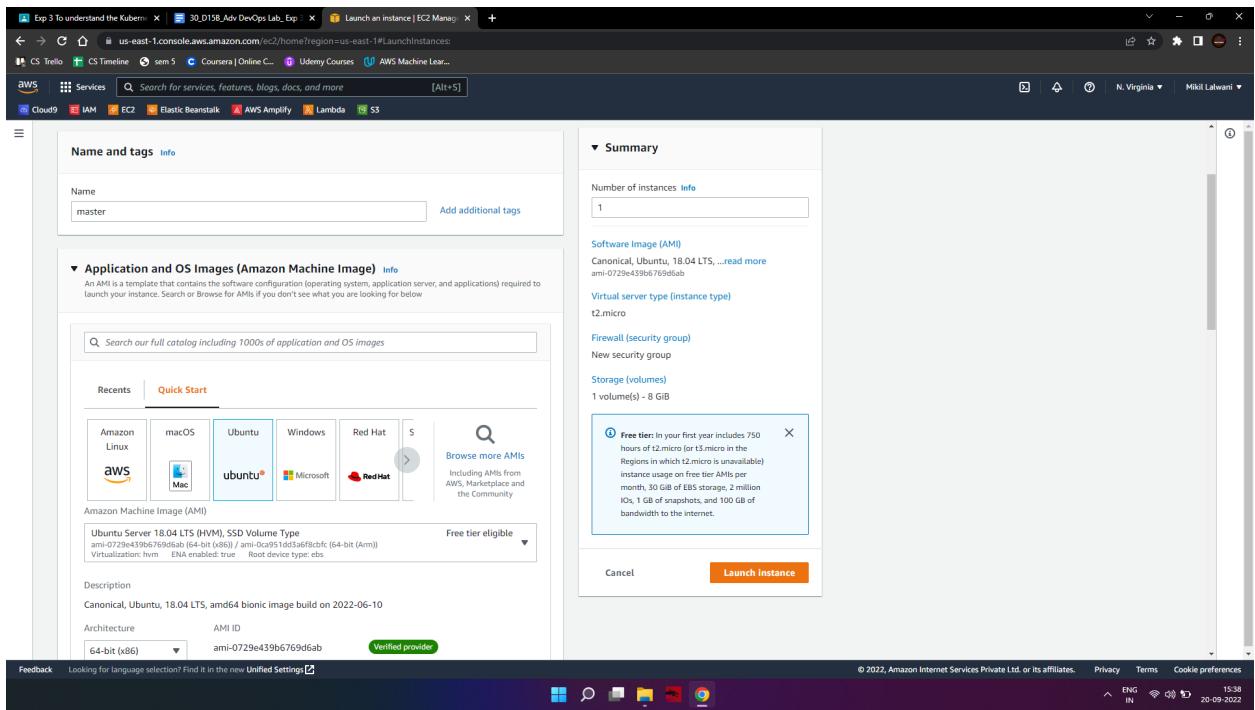
Step 1: Install Docker

Kubernetes requires an existing Docker installation. If you already have Docker installed, skip ahead to Step 2.

If you do not have Kubernetes, install it by following these steps:

1. Update the package list with the command:

```
on-master&slave$sudo apt-get update
```



Screenshot of the AWS EC2 Management Console showing the creation of an instance named "worker1".

Name and tags

Name: worker1

Application and OS Images (Amazon Machine Image)

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

Recent AMIs: Amazon Linux, macOS, Ubuntu, Windows, Red Hat

Browse more AMIs Including AMIs from AWS, Marketplace and the Community

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type

Description: Canonical, Ubuntu, 18.04 LTS, amd64 bionic image build on 2022-06-10

Architecture: 64-bit (x86) AMI ID: ami-0729e439b6769d6ab Verified provider

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 18.04 LTS, ...read more

ami-0729e439b6769d6ab

Virtual server type (instance type): t2.micro

Firewall (security group): launch-wizard-1

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 usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GiB of bandwidth to the internet.

Cancel Launch Instance

Feedback: Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences ENG IN 15:39 20-09-2022

Screenshot of the AWS EC2 Management Console showing the creation of an instance named "worker2".

Name and tags

Name: worker2

Application and OS Images (Amazon Machine Image)

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

Recent AMIs: Amazon Linux, macOS, Ubuntu, Windows, Red Hat

Browse more AMIs Including AMIs from AWS, Marketplace and the Community

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type

Description: Canonical, Ubuntu, 18.04 LTS, amd64 bionic image build on 2022-06-10

Architecture: 64-bit (x86) AMI ID: ami-0729e439b6769d6ab Verified provider

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 18.04 LTS, ...read more

ami-0729e439b6769d6ab

Virtual server type (instance type): t2.micro

Firewall (security group): launch-wizard-1

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 usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GiB of bandwidth to the internet.

Cancel Launch Instance

Feedback: Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences ENG IN 15:39 20-09-2022

Screenshot of the AWS EC2 Management Console showing the Instances page and a detailed view of a security group.

Instances Page

The Instances page displays three running t2.micro instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 IP	Elastic IP
worker2	i-03a454a56d799f787	Running	t2.micro	Initializing	No alarms	us-east-1a	ec2-54-208-121-49.co...	54.208.121.49	-
worker1	i-06e954f95c4336f0b	Running	t2.micro	Initializing	No alarms	us-east-1d	ec2-44-211-142-52.co...	44.211.142.52	-
master	i-032ca59d70b51f311	Running	t2.micro	Initializing	No alarms	us-east-1d	ec2-3-86-253-100.com...	3.86.253.100	-

Security Group Details

The security group details page for sg-04e78149a25aa7f3d shows the following information:

Details			
Security group name: launch-wizard-1	Security group ID: sg-04e78149a25aa7f3d	Description: launch-wizard-1 created 2022-09-20T10:08:10.608Z	VPC ID: vpc-06f870b50074eb753
Owner: 689179665899	Inbound rules count: 1 Permission entry	Outbound rules count: 1 Permission entry	

Inbound rules (1/1)

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
-	sgr-0cb44745cbcd41b70	IPv4	SSH	TCP	22	0.0.0.0/0	-

The screenshot shows the AWS EC2 Management Console with the URL <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#ModifyInboundSecurityGroupRulessecurityGroupId=sg-04e78149a25aa7f3d>. The page is titled "Edit inbound rules" under "Security Groups". It lists two rules:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
-	SSH	TCP	22	Anywhere-1...	
-	All traffic	All	All	Custom	sg-04e78149a25aa7f3d

Buttons at the bottom include "Add rule", "Cancel", "Preview changes", and "Save rules". The status bar at the bottom right shows "Feedback", "Looking for language selection? Find it in the new Unified Settings", "© 2022, Amazon Internet Services Private Ltd. or its affiliates.", "Privacy", "Terms", "Cookie preferences", "ENG IN", "15:43", and "20-09-2022".

The screenshot shows a terminal window with the command `sudo apt-get update` running. The output is as follows:

```
ubuntu@ip-172-31-82-227:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [8570 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe Translation-en [4941 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [151 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/multiverse Translation-en [108 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [2729 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [503 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [913 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted Translation-en [126 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [1842 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe Translation-en [399 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [24.9 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse Translation-en [6012 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [10.8 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/main Translation-en [5016 B]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [11.6 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe Translation-en [5864 B]
Get:21 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [2388 kB]
Get:22 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [414 kB]
Get:23 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [884 kB]
Get:24 http://security.ubuntu.com/ubuntu bionic-security/restricted Translation-en [122 kB]
Get:25 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [1228 kB]
Get:26 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [282 kB]
Get:27 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [19.0 kB]
Get:28 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [3836 B]
```

```
ubuntu@ip-172-31-25-41:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [8570 kB]
Get:5 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe Translation-en [4941 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [151 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/multiverse Translation-en [108 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [2729 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [503 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [913 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted Translation-en [126 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [1842 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe Translation-en [399 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [24.9 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse Translation-en [6012 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [10.8 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/main Translation-en [5016 B]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [11.6 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe Translation-en [5864 B]
Get:21 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [2388 kB]
Get:22 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [414 kB]
Get:23 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [884 kB]
Get:24 http://security.ubuntu.com/ubuntu bionic-security/restricted Translation-en [122 kB]
Get:25 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [1228 kB]
Get:26 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [282 kB]
Get:27 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [19.0 kB]
Get:28 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [3836 B]
```

```
ubuntu@ip-172-31-90-183:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [8570 kB]
Get:5 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe Translation-en [4941 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [151 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/multiverse Translation-en [108 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [2729 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [503 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [913 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted Translation-en [126 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [1842 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe Translation-en [399 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [24.9 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse Translation-en [6012 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [10.8 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/main Translation-en [5016 B]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [11.6 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe Translation-en [5864 B]
Get:21 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [2388 kB]
Get:22 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [414 kB]
Get:23 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [884 kB]
Get:24 http://security.ubuntu.com/ubuntu bionic-security/restricted Translation-en [122 kB]
Get:25 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [1228 kB]
Get:26 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [282 kB]
Get:27 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [19.0 kB]
Get:28 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [3836 B]
```

2. Next, install Docker with the command:

on-master&slave\$ sudo apt-get install docker.io

```
ubuntu@ip-172-31-82-227:~$ sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd docker.io pigz runc ubuntu-fan
0 upgraded, 6 newly installed, 0 to remove and 56 not upgraded.
Need to get 74.2 MB of archives.
After this operation, 360 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe amd64 pigz amd64 2.4-1 [57.4 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 bridge-utils amd64 1.5-15ubuntu1 [30.1 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 runc amd64 1.0.1-0ubuntu2~18.04.1
[4155 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 containerd amd64 1.5.5-0ubuntu3~18
.04.2 [33.0 MB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 docker.io amd64 20.10.7-0ubuntu5~1
8.04.3 [36.9 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 ubuntu-fan all 0.12.10 [34.7 kB]
Fetched 74.2 MB in 2s (37.0 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 58033 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.4-1_amd64.deb ...
Unpacking pigz (2.4-1) ...
```

```
Reading package lists... Done
ubuntu@ip-172-31-25-41:~$ sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd docker.io pigz runc ubuntu-fan
0 upgraded, 6 newly installed, 0 to remove and 56 not upgraded.
Need to get 74.2 MB of archives.
After this operation, 360 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe amd64 pigz amd64 2.4-1 [57.4 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 bridge-utils amd64 1.5-15ubuntu1 [30.1 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 runc amd64 1.0.1-0ubuntu2~18.04.1
[4155 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 containerd amd64 1.5.5-0ubuntu3~18
.04.2 [33.0 MB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 docker.io amd64 20.10.7-0ubuntu5~1
8.04.3 [36.9 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 ubuntu-fan all 0.12.10 [34.7 kB]
Fetched 74.2 MB in 2s (39.7 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 58033 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.4-1_amd64.deb ...
Unpacking pigz (2.4-1) ...
```

```
ubuntu@ip-172-31-90-183:~$ sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd docker.io pigz runc ubuntu-fan
0 upgraded, 6 newly installed, 0 to remove and 56 not upgraded.
Need to get 74.2 MB of archives.
After this operation, 360 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe amd64 pigz amd64 2.4-1 [57.4 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 bridge-utils amd64 1.5-15ubuntu1 [30.1 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 runc amd64 1.0.1-0ubuntu2~18.04.1 [4155 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 containerd amd64 1.5.5-0ubuntu3~18.04.2 [33.0 MB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 docker.io amd64 20.10.7-0ubuntu5~18.04.3 [36.9 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 ubuntu-fan all 0.12.10 [34.7 kB]
Fetched 74.2 MB in 2s (47.5 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 58833 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.4-1_amd64.deb ...
Unpacking pigz (2.4-1) ...
```

3. Repeat the process on each server that will act as a node.

Check the installation (and version) by entering the following:

on-master&slave\$docker --version

```
Processing triggers for libltdl3:1.0.20-21 ...  
ubuntu@ip-172-31-82-227:~$ docker --version  
Docker version 20.10.7, build 20.10.7-0ubuntu5~18.04.3  
ubuntu@ip-172-31-82-227:~$
```

```
ubuntu@ip-172-31-25-41:~$ docker --version  
Docker version 20.10.7, build 20.10.7-0ubuntu5~18.04.3  
ubuntu@ip-172-31-25-41:~$
```

```
ubuntu@ip-172-31-90-183:~$ docker --version  
Docker version 20.10.7, build 20.10.7-0ubuntu5~18.04.3  
ubuntu@ip-172-31-90-183:~$
```

Step 2: Start and Enable Docker

1. Set Docker to launch at boot by entering the following:

on-master&slave\$sudo systemctl enable docker

```
ubuntu@ip-172-31-82-227:~$ sudo systemctl enable docker  
ubuntu@ip-172-31-82-227:~$
```

```
ubuntu@ip-172-31-25-41:~$ sudo systemctl enable docker  
ubuntu@ip-172-31-25-41:~$
```

```
ubuntu@ip-172-31-90-183:~$ sudo systemctl enable docker  
ubuntu@ip-172-31-90-183:~$
```

2. Verify Docker is running:

on-master&slave\$ sudo systemctl status docker

```
ubuntu@ip-172-31-82-227:~$ sudo systemctl status docker  
● docker.service - Docker Application Container Engine  
  Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)  
  Active: active (running) since Tue 2022-09-20 10:19:10 UTC; 1min 26s ago  
    Docs: https://docs.docker.com  
   Main PID: 2899 (dockerd)  
     Tasks: 8  
    CGroup: /system.slice/docker.service  
           └─2899 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock  
  
Sep 20 10:19:09 ip-172-31-82-227 dockerd[2899]: time="2022-09-20T10:19:09.925990222Z" level=warning msg="Your kernel  
Sep 20 10:19:09 ip-172-31-82-227 dockerd[2899]: time="2022-09-20T10:19:09.926171168Z" level=warning msg="Your kernel  
Sep 20 10:19:09 ip-172-31-82-227 dockerd[2899]: time="2022-09-20T10:19:09.926350026Z" level=warning msg="Your kernel  
Sep 20 10:19:09 ip-172-31-82-227 dockerd[2899]: time="2022-09-20T10:19:09.926717856Z" level=info msg="Loading contain  
Sep 20 10:19:10 ip-172-31-82-227 dockerd[2899]: time="2022-09-20T10:19:10.198675310Z" level=info msg="Default bridge  
Sep 20 10:19:10 ip-172-31-82-227 dockerd[2899]: time="2022-09-20T10:19:10.274468973Z" level=info msg="Loading contain  
Sep 20 10:19:10 ip-172-31-82-227 dockerd[2899]: time="2022-09-20T10:19:10.3541721997Z" level=info msg="Docker daemon"  
Sep 20 10:19:10 ip-172-31-82-227 dockerd[2899]: time="2022-09-20T10:19:10.354560036Z" level=info msg="Daemon has comp  
Sep 20 10:19:10 ip-172-31-82-227 systemd[1]: Started Docker Application Container Engine.  
Sep 20 10:19:10 ip-172-31-82-227 dockerd[2899]: time="2022-09-20T10:19:10.412402390Z" level=info msg="API listen on /  
ubuntu@ip-172-31-82-227:~$
```

```
ubuntu@ip-172-31-25-41:~$ sudo systemctl status docker  
● docker.service - Docker Application Container Engine  
  Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)  
  Active: active (running) since Tue 2022-09-20 10:19:14 UTC; 1min 26s ago  
    Docs: https://docs.docker.com  
   Main PID: 2972 (dockerd)  
     Tasks: 8  
    CGroup: /system.slice/docker.service  
           └─2972 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock  
  
Sep 20 10:19:14 ip-172-31-25-41 dockerd[2972]: time="2022-09-20T10:19:14.437064763Z" level=warning msg="Your kernel d  
Sep 20 10:19:14 ip-172-31-25-41 dockerd[2972]: time="2022-09-20T10:19:14.437216874Z" level=warning msg="Your kernel d  
Sep 20 10:19:14 ip-172-31-25-41 dockerd[2972]: time="2022-09-20T10:19:14.437350043Z" level=warning msg="Your kernel d  
Sep 20 10:19:14 ip-172-31-25-41 dockerd[2972]: time="2022-09-20T10:19:14.437658488Z" level=info msg="Loading contain  
Sep 20 10:19:14 ip-172-31-25-41 dockerd[2972]: time="2022-09-20T10:19:14.738513805Z" level=info msg="Default bridge (br  
Sep 20 10:19:14 ip-172-31-25-41 dockerd[2972]: time="2022-09-20T10:19:14.817701471Z" level=info msg="Loading contain  
Sep 20 10:19:14 ip-172-31-25-41 dockerd[2972]: time="2022-09-20T10:19:14.930885179Z" level=info msg="Docker daemon" c  
Sep 20 10:19:14 ip-172-31-25-41 dockerd[2972]: time="2022-09-20T10:19:14.931404245Z" level=info msg="Daemon has compl  
Sep 20 10:19:15 ip-172-31-25-41 systemd[1]: Started Docker Application Container Engine.  
Sep 20 10:19:15 ip-172-31-25-41 dockerd[2972]: time="2022-09-20T10:19:15.006636091Z" level=info msg="API listen on /v  
ubuntu@ip-172-31-25-41:~$
```

```
ubuntu@ip-172-31-90-183:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
  Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
  Active: active (running) since Tue 2022-09-20 10:19:15 UTC; 1min 29s ago
    Docs: https://docs.docker.com
   Main PID: 2926 (dockerd)
     Tasks: 8
    CGroup: /system.slice/docker.service
            └─2926 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Sep 20 10:19:15 ip-172-31-90-183 dockerd[2926]: time="2022-09-20T10:19:15.406246033Z" level=warning msg="Your kernel
Sep 20 10:19:15 ip-172-31-90-183 dockerd[2926]: time="2022-09-20T10:19:15.406437757Z" level=warning msg="Your kernel
Sep 20 10:19:15 ip-172-31-90-183 dockerd[2926]: time="2022-09-20T10:19:15.406619488Z" level=warning msg="Your kernel
Sep 20 10:19:15 ip-172-31-90-183 dockerd[2926]: time="2022-09-20T10:19:15.406989239Z" level=info msg="Loading contain
Sep 20 10:19:15 ip-172-31-90-183 dockerd[2926]: time="2022-09-20T10:19:15.592615851Z" level=info msg="Default bridge
Sep 20 10:19:15 ip-172-31-90-183 dockerd[2926]: time="2022-09-20T10:19:15.683912831Z" level=info msg="Loading contain
Sep 20 10:19:15 ip-172-31-90-183 dockerd[2926]: time="2022-09-20T10:19:15.769870324Z" level=info msg="Docker daemon"
Sep 20 10:19:15 ip-172-31-90-183 dockerd[2926]: time="2022-09-20T10:19:15.770318024Z" level=info msg="Daemon has comp
Sep 20 10:19:15 ip-172-31-90-183 systemd[1]: Started Docker Application Container Engine.
Sep 20 10:19:15 ip-172-31-90-183 dockerd[2926]: time="2022-09-20T10:19:15.829178976Z" level=info msg="API listen on /
ubuntu@ip-172-31-90-183:~$
```

To start Docker if it's not running:
on-master&slave\$ sudo systemctl start docker

3. Repeat on all the other nodes.

Install Kubernetes

Step 3: Add Kubernetes Signing Key

Since you are downloading Kubernetes from a non-standard repository, it is essential to ensure that the software is authentic. This is done by adding a signing key.

1. Enter the following to add a signing key:

on-master&slave\$ curl -s
https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add

```
ubuntu@ip-172-31-82-227:~$ curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add
OK
ubuntu@ip-172-31-82-227:~$
```

```
ubuntu@ip-172-31-25-41:~$ curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add
OK
ubuntu@ip-172-31-25-41:~$
```

```
ubuntu@ip-172-31-90-183:~$ curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add
OK
ubuntu@ip-172-31-90-183:~$
```

If you get an error that curl is not installed, install it with:
on-master&slave\$ sudo apt-get install curl

2. Then repeat the previous command to install the signing keys. Repeat for each server node.

Step 4: Add Software Repositories

Kubernetes is not included in the default repositories. To add them, enter the following:

```
on-master&slave$ sudo apt-add-repository "deb http://apt.kubernetes.io/
kubernetes-xenial main"
```

```
OK
ubuntu@ip-172-31-82-227:~$ sudo apt-add-repository "deb http://apt.kubernetes.io/ kubernetes-xenial main"
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:5 http://security.ubuntu.com/ubuntu bionic-security InRelease
Get:4 https://packages.cloud.google.com/apt kubernetes-xenial InRelease [9383 B]
Get:6 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 Packages [59.4 kB]
Fetched 68.8 kB in 0s (142 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-82-227:~$
```

```
ubuntu@ip-172-31-25-41:~$ sudo apt-add-repository "deb http://apt.kubernetes.io/ kubernetes-xenial main"
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu bionic-security InRelease
Get:5 https://packages.cloud.google.com/apt kubernetes-xenial InRelease [9383 B]
Get:6 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 Packages [59.4 kB]
Fetched 68.8 kB in 0s (145 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-25-41:~$
```

```
ubuntu@ip-172-31-90-183:~$ sudo apt-add-repository "deb http://apt.kubernetes.io/ kubernetes-xenial main"
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu bionic-security InRelease
Get:5 https://packages.cloud.google.com/apt kubernetes-xenial InRelease [9383 B]
Get:6 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 Packages [59.4 kB]
Fetched 68.8 kB in 0s (144 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-90-183:~$
```

Repeat on each server node.

Step 5: Kubernetes Installation Tools

Kubeadm (Kubernetes Admin) is a tool that helps initialize a cluster. It fast-tracks setup by using community-sourced best practices. Kubelet is the work package,

which runs on every node and starts containers. The tool gives you command-line access to clusters.

Install Kubernetes tools with the command:

```
on-master&slave$ sudo apt-get install kubeadm kubelet kubectl -y
```

```
on-master&slave$ sudo apt-mark hold kubeadm kubelet kubectl
```

```
Fetched 68.8 kB in 0s (142 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-82-227:~$ sudo apt-get install kubeadm kubelet kubectl -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools kubernetes-cni socat
The following NEW packages will be installed:
  conntrack cri-tools kubeadm kubelet kubernetes-cni socat
0 upgraded, 7 newly installed, 0 to remove and 56 not upgraded.
Need to get 81.5 MB of archives.
After this operation, 327 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 conntrack amd64 1:1.4.4+snapshot20161117-6ubuntu2 [30.6 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 socat amd64 1.7.3.2-2ubuntu2 [342 kB]
Get:3 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 cri-tools amd64 1.25.0-00 [17.9 MB]
Get:4 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubernetes-cni amd64 1.1.1-00 [25.0 MB]
Get:5 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubelet amd64 1.25.1-00 [19.5 MB]
Get:6 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubectl amd64 1.25.1-00 [9503 kB]
Get:7 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubeadm amd64 1.25.1-00 [9215 kB]
Fetched 81.5 MB in 2s (46.7 MB/s)
Selecting previously unselected package conntrack.
(Reading database ... 58356 files and directories currently installed.)
Preparing to unpack .../0-conntrack_1%3a1.4.4+snapshot20161117-6ubuntu2_amd64.deb ...
Unpacking conntrack (1:1.4.4+snapshot20161117-6ubuntu2) ...
Selecting previously unselected package cri-tools.
Preparing to unpack .../1-cri-tools_1.25.0-00_amd64.deb ...
Unpacking cri-tools (1.25.0-00) ...
Selecting previously unselected package kubernetes-cni.
```

```
ubuntu@ip-172-31-25-41:~$ sudo apt-get install kubeadm kubelet kubectl -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools kubernetes-cni socat
The following NEW packages will be installed:
  conntrack cri-tools kubeadm kubelet kubernetes-cni socat
0 upgraded, 7 newly installed, 0 to remove and 56 not upgraded.
Need to get 81.5 MB of archives.
After this operation, 327 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 conntrack amd64 1:1.4.4+snapshot20161117-6ubuntu2 [30.6 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 socat amd64 1.7.3.2-2ubuntu2 [342 kB]
Get:3 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 cri-tools amd64 1.25.0-00 [17.9 MB]
Get:4 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubernetes-cni amd64 1.1.1-00 [25.0 MB]
Get:5 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubelet amd64 1.25.1-00 [19.5 MB]
Get:6 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubectl amd64 1.25.1-00 [9503 kB]
Get:7 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubeadm amd64 1.25.1-00 [9215 kB]
Fetched 81.5 MB in 2s (43.2 MB/s)
Selecting previously unselected package conntrack.
(Reading database ... 58356 files and directories currently installed.)
Preparing to unpack .../0-conntrack_1%3a1.4.4+snapshot20161117-6ubuntu2_amd64.deb ...
Unpacking conntrack (1:1.4.4+snapshot20161117-6ubuntu2) ...
Selecting previously unselected package cri-tools.
Preparing to unpack .../1-cri-tools_1.25.0-00_amd64.deb ...
Unpacking cri-tools (1.25.0-00) ...
Selecting previously unselected package kubernetes-cni.
```

```

ubuntu@ip-172-31-90-183:~$ sudo apt-get install kubeadm kubelet kubectl -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools kubernetes-cni socat
The following NEW packages will be installed:
  conntrack cri-tools kubeadm kubelet kubernetes-cni socat
0 upgraded, 7 newly installed, 0 to remove and 56 not upgraded.
Need to get 81.5 MB of archives.
After this operation, 327 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 conntrack amd64 1:1.4.4+snapshot20161117-6ubuntu2 [30.6 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 socat amd64 1.7.3.2-2ubuntu2 [342 kB]
Get:3 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 cri-tools amd64 1.25.0-00 [17.9 MB]
Get:4 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubernetes-cni amd64 1.1.1-00 [25.0 MB]
Get:5 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubelet amd64 1.25.1-00 [19.5 MB]
Get:6 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubectl amd64 1.25.1-00 [9503 kB]
Get:7 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubeadm amd64 1.25.1-00 [9215 kB]
Fetched 81.5 MB in 3s (24.8 MB/s)
Selecting previously unselected package conntrack.
(Reading database ... 58356 files and directories currently installed.)
Preparing to unpack .../0-conntrack_1%3a1.4.4+snapshot20161117-6ubuntu2_amd64.deb ...
Unpacking conntrack (1:1.4.4+snapshot20161117-6ubuntu2) ...
Selecting previously unselected package cri-tools.
Preparing to unpack .../1-cri-tools_1.25.0-00_amd64.deb ...
Unpacking cri-tools (1.25.0-00) ...
Selecting previously unselected package kubernetes-cni.

ubuntu@ip-172-31-82-227:~$ sudo apt-mark hold kubeadm kubelet kubectl
kubeadm set on hold.
kubelet set on hold.
kubectl set on hold.
ubuntu@ip-172-31-82-227:~$

ubuntu@ip-172-31-25-41:~$ sudo apt-mark hold kubeadm kubelet kubectl
kubeadm set on hold.
kubelet set on hold.
kubectl set on hold.
ubuntu@ip-172-31-25-41:~$

Processing triggers for man-db (2.7.0.2-1ubuntu0.1) ...
ubuntu@ip-172-31-90-183:~$ sudo apt-mark hold kubeadm kubelet kubectl
kubeadm set on hold.
kubelet set on hold.
kubectl set on hold.
ubuntu@ip-172-31-90-183:~$
```

Allow the process to complete.

Verify the installation with:

on-master&slave\$**kubeadm version**

Repeat for each server node.

Note: Make sure you install the same version of each package on each machine.

Different versions can create instability. Also, this process prevents apt from automatically updating Kubernetes. For update instructions, please see the developers' instructions.

Kubernetes Deployment

Step 6: Begin Kubernetes Deployment

Start by disabling the swap memory on each server:

on-master&slave\$**sudo swapoff --a**

```
kubectl set on hold.  
kubectl set on hold.  
ubuntu@ip-172-31-82-227:~$ sudo swapoff --a  
kubectl set on hold.  
kubectl set on hold.  
ubuntu@ip-172-31-25-41:~$ sudo swapoff --a  
kubectl set on hold.  
kubectl set on hold.  
ubuntu@ip-172-31-90-183:~$ sudo swapoff --a
```

Step 7: Assign Unique Hostname for Each Server Node

Decide which server to set as the master node. Then enter the command:
on-master\$ sudo hostnamectl set-hostname master-node

```
ubuntu@ip-172-31-82-227:~$ sudo hostnamectl set-hostname master-node  
ubuntu@ip-172-31-82-227:~$  
  
ubuntu@ip-172-31-25-41:~$ sudo hostnamectl set-hostname worker01  
ubuntu@ip-172-31-25-41:~$  
  
ubuntu@ip-172-31-90-183:~$ sudo hostnamectl set-hostname worker02  
ubuntu@ip-172-31-90-183:~$
```

Next, set a worker node hostname by entering the following on the worker server:
on-slave\$ sudo hostnamectl set-hostname worker01

If you have additional worker nodes, use this process to set a unique hostnsame on each.

Step 8: Initialize Kubernetes on Master Node

Switch to the master server node, and enter the following:
on-master\$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16

```

ubuntu@ip-172-31-82-227:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=all
[init] Using Kubernetes version: v1.25.1
[preflight] Running pre-flight checks
  [WARNING NumCPU]: the number of available CPUs 1 is less than the required 2
  [WARNING Mem]: the system RAM (974 MB) is less than the minimum 1700 MB
[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 in beforehand using 'kubeadm config images pull'
[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 [kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local master-node] and IPs [10.96.0.1 172.31.82.227]
[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 [localhost master-node] and IPs [172.31.82.227 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [localhost master-node] and IPs [172.31.82.227 127.0.0.1 ::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 "kubelet.conf" kubeconfig file

```

If you are trying to run this on EC2 you'll get an error message saying less cpu and memory to override the error run the above command with
`--ignore-preflight-errors=all`

For eg: on-master\$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
`--ignore-preflight-errors=all`

Once this command finishes, it will display a kubeadm join message at the end.
 Make a note of the whole entry. This will be used to join the worker nodes to the cluster.

```

kubeadm join 172.31.82.227:6443 --token qiacv4.hcjm6j95j6ilt1tl \
  --discovery-token-ca-cert-hash
sha256:4c13f7d2b7dc771a0380df19969ea5e3fc920c24611072dee907269ece7e503
3

```

Next, enter the following to create a directory for the cluster:

```

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

```

```

ubuntu@ip-172-31-82-227:~$ mkdir -p $HOME/.kube
ubuntu@ip-172-31-82-227:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
ubuntu@ip-172-31-82-227:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-82-227:~$ 

```

Step 9: Deploy Pod Network to Cluster

A Pod Network is a way to allow communication between different nodes in the cluster. This tutorial uses the flannel virtual network.

Enter the following:

```
kubernetes-master:~$ sudo kubectl apply -f
```

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

```
ubuntu@ip-172-31-82-227:~$ sudo 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-172-31-82-227:~$
```

(if the above command is not working try without sudo command)

```
kubectl apply -f [podnetwork].yaml"
```

<https://kubernetes.io/docs/concepts/cluster-administration/addons/>

Allow the process to complete.

Verify that everything is running and communicating:

```
kubernetes-master:~$ kubectl get pods --all-namespaces
```

```
ubuntu@ip-172-31-82-227:~$ kubectl get pods --all-namespaces
NAMESPACE      NAME                READY   STATUS    RESTARTS   AGE
kube-flannel   kube-flannel-ds-7tsvw   1/1     Running   0          23s
kube-system    coredns-565d847f94-97lxp   1/1     Running   0          85s
kube-system    coredns-565d847f94-vr6wv   1/1     Running   0          85s
kube-system    etcd-master-node        1/1     Running   0          97s
kube-system    kube-apiserver-master-node 1/1     Running   0          97s
kube-system    kube-controller-manager-master-node 1/1     Running   0          99s
kube-system    kube-proxy-26bsz       1/1     Running   0          85s
kube-system    kube-scheduler-master-node 1/1     Running   0          97s
ubuntu@ip-172-31-82-227:~$
```

Step 10: Join Worker Node to Cluster

As indicated in step 7, you can enter the kubeadm join command on each worker node to connect it to the cluster.

Switch to the worker01 system and enter the command you noted from step 7 -

```
kubernetes-slave:~$ kubeadm join 172.31.82.227:6443 --token
qiacv4.hcjhm6j95j6ilt1tl \
```

```
--discovery-token-ca-cert-hash  
sha256:4c13f7d2b7dc771a0380df19969ea5e3fc920c24611072dee907269ece7e503  
3
```

```
ubuntu@ip-172-31-25-41:~$ sudo su  
root@worker01:/home/ubuntu# kubeadm join 172.31.82.227:6443 --token qiacv4.hcj6j95j6ilt1l \  
>     --discovery-token-ca-cert-hash sha256:4c13f7d2b7dc771a0380df19969ea5e3fc920c24611072dee907269ece7e5033  
[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 cm 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/kubelet/kubeadm-flags.env"  
[kubelet-start] Starting the kubelet  
[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@worker01:/home/ubuntu#
```

```
ubuntu@ip-172-31-90-183:~$ sudo su  
root@worker02:/home/ubuntu# kubeadm join 172.31.82.227:6443 --token qiacv4.hcj6j95j6ilt1l \  
>     --discovery-token-ca-cert-hash sha256:4c13f7d2b7dc771a0380df19969ea5e3fc920c24611072dee907269ece7e5033  
[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 cm 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/kubelet/kubeadm-flags.env"  
[kubelet-start] Starting the kubelet  
[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@worker02:/home/ubuntu#
```

ON EC2 make sure you open the port in security group ADVERTISED HERE:

Replace the alphanumeric codes with those from your master server. Repeat for each worker node on the cluster. Wait a few minutes; then you can check the status of the nodes.

Switch to the master server, and enter:

```
kubernetes-master:~$ kubectl get nodes
```

```
ubuntu@ip-172-31-82-227:~$ kubectl get nodes  
NAME      STATUS  ROLES      AGE      VERSION  
master-node  Ready   control-plane  4m52s  v1.25.1  
worker01    Ready   <none>    2m10s  v1.25.1  
worker02    Ready   <none>    2m23s  v1.25.1  
ubuntu@ip-172-31-82-227:~$
```

The system should display the worker nodes that you joined to the cluster.

Output

NAME	STATUS	ROLES	AGE	VERSION
------	--------	-------	-----	---------

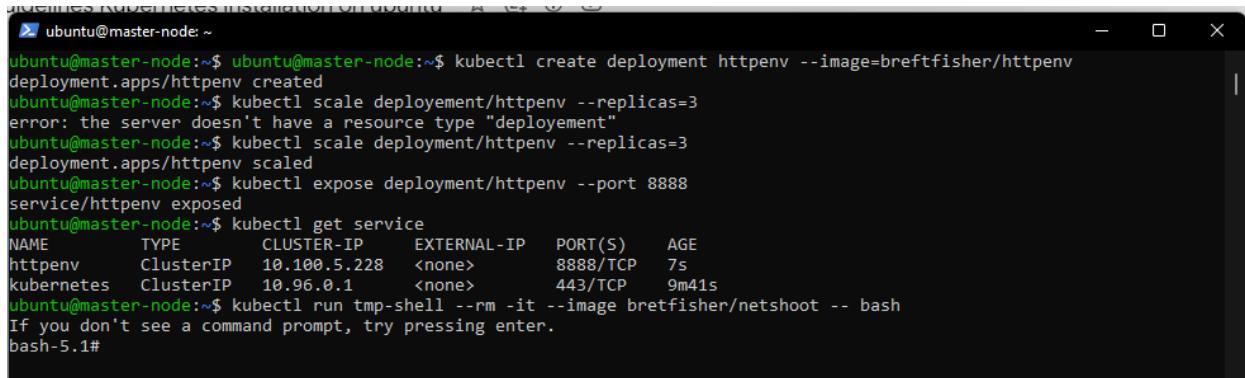
```
master     Readymaster    1d    v1.14.0
```

```
worker1 Ready <none> 1d v1.14.0
```

If all of your nodes have the value Ready for STATUS, it means that they're part of the cluster and ready to run workloads.

If, however, a few of the nodes have NotReady as the STATUS, it could mean that the worker nodes haven't finished their setup yet. Wait for around five to ten minutes before re-running kubectl get node and inspecting the new output. If a few nodes still have NotReady as the status, you might have to verify and re-run the commands in the previous steps.

```
kubectl create deployment httpenv --image=bretfisher/httpenv
kubectl scale deployment/httpenv --replicas=3
kubectl expose deployment/httpenv --port 8888
kubectl get service
kubectl run tmp-shell --rm -it --image bretfisher/netshoot -- bash
```



A terminal window showing the execution of several Kubernetes commands. The session starts with 'ubuntu@master-node:~\$' and ends with 'bash-5.1#'. The commands entered are:

```
ubuntu@master-node:~$ kubectl create deployment httpenv --image=bretfisher/httpenv
deployment.apps/httpenv created
ubuntu@master-node:~$ kubectl scale deployment/httpenv --replicas=3
error: the server doesn't have a resource type "deployment"
ubuntu@master-node:~$ kubectl scale deployment/httpenv --replicas=3
deployment.apps/httpenv scaled
ubuntu@master-node:~$ kubectl expose deployment/httpenv --port 8888
service/httpenv exposed
ubuntu@master-node:~$ kubectl get service
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
httpenv   ClusterIP  10.100.5.228  <none>          8888/TCP    7s
kubernetes  ClusterIP  10.96.0.1    <none>          443/TCP    9m41s
ubuntu@master-node:~$ kubectl run tmp-shell --rm -it --image bretfisher/netshoot -- bash
If you don't see a command prompt, try pressing enter.
bash-5.1#
```

Conclusion -

Thus we have successfully install and spin up a Kubernetes cluster on a Linux/Cloud machine.

Our application will now be able to automatically scale up or down based on the demand of the application. This makes it more efficient and cost effective.

In future we can add more nodes to the cluster and increase the available resources. This will help us to handle more traffic and provide better user experience.

This was a brief introduction to Kubernetes. In the next few articles, we will cover more advanced topics such as deployment strategies, service discovery, and persistent storage.

We hope you found this article informative and useful. If you have any questions or comments, please feel free to leave them in the comments section below.

Stay tuned for our next article where we will cover more advanced topics such as deployment strategies, service discovery, and persistent storage.

We hope you found this article informative and useful. If you have any questions or comments, please feel free to leave them in the comments section below.