

Seat no - IT5B110

Roll No - 37

Aim-

Install Kubectl and execute Kubectl commands to manage the Kubernetes cluster and deploy your first Kubernetes application.

Prerequisites

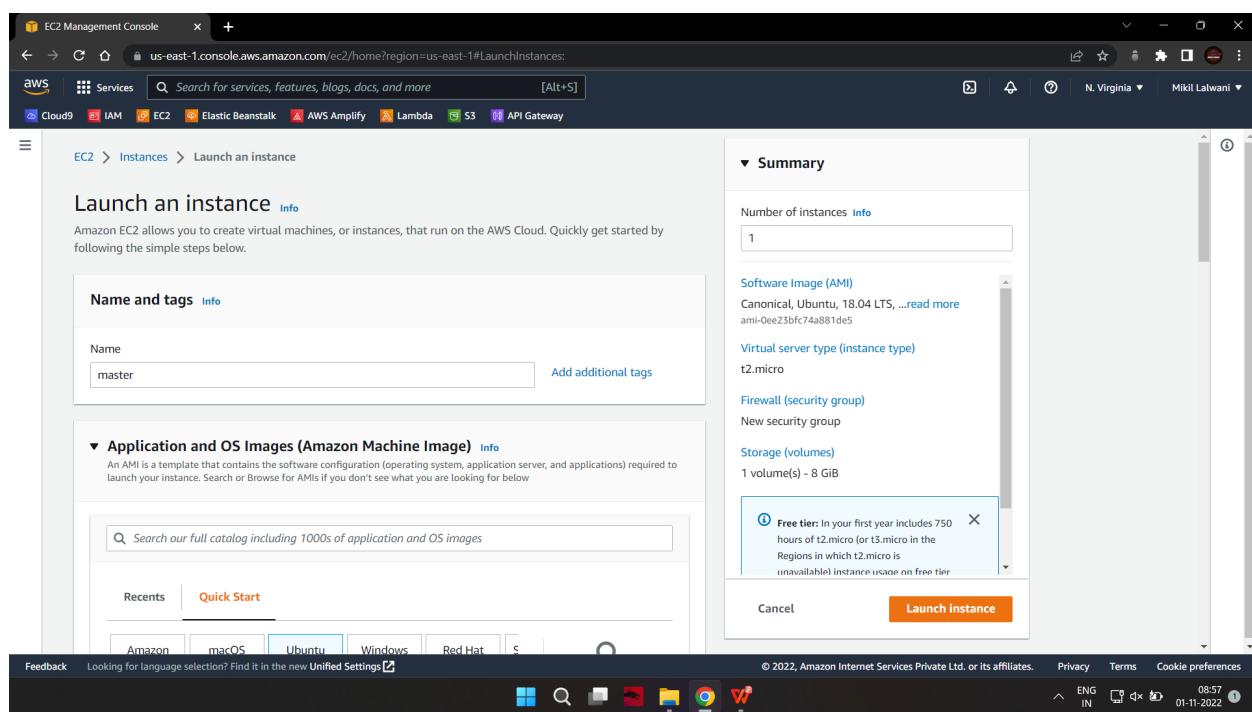
2 or more Linux servers running Ubuntu 18.04 /20.04 on Virtual box or you can use EC2 free tier instances choose the ubuntu 20.04 AMI free tier

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



EC2 Management Console

Services Search for services, features, blogs, docs, and more [Alt+S]

Cloud9 IAM EC2 Elastic Beanstalk AWS Amplify Lambda S3 API Gateway

Recents Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat

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

Amazon Machine Image (AMI)

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type ami-0ee23bfc74a881de5 (64-bit (x86)) / ami-0351643488963af72 (64-bit (Arm)) Virtualization: hvm ENA enabled: true Root device type: ebs

Description Canonical, Ubuntu, 18.04 LTS, amd64 bionic image build on 2022-09-01

Architecture AMI ID

64-bit (x86) ami-0ee23bfc74a881de5 Verified provider

Instance type t2.micro

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

Summary

Number of instances Info 1

Software Image (AMI) Canonical, Ubuntu, 18.04 LTS, ...read more ami-0ee23bfc74a881de5

Virtual server type (instance type) t2.micro

Firewall (security group) New security group

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

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier

Cancel Launch instance

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences ENG IN 08:57 01-11-2022

EC2 Management Console

Services Search for services, features, blogs, docs, and more [Alt+S]

Cloud9 IAM EC2 Elastic Beanstalk AWS Amplify Lambda S3 API Gateway

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 Proceed without a key pair (Not recommended) Default value Create new key pair

Network settings

Network Info vpc-06f870b50074eb753

Subnet Info No preference (Default subnet in any availability zone)

Auto-assign public IP Info

Enable

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

We'll create a new security group called 'launch-wizard-3' with the following rules:

Allow SSH traffic from Anywhere 0.0.0.0/0

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

Summary

Number of instances Info 1

Software Image (AMI) Canonical, Ubuntu, 18.04 LTS, ...read more ami-0ee23bfc74a881de5

Virtual server type (instance type) t2.micro

Firewall (security group) New security group

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

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier

Cancel Launch instance

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences ENG IN 08:57 01-11-2022

EC2 Management Console

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:

New EC2 Experience Tell us what you think

Services Search for services, features, blogs, docs, and more [Alt+S]

Cloud9 IAM EC2 Elastic Beanstalk AWS Amplify Lambda S3 API Gateway

EC2 Dashboard EC2 Global View Events Tags Limits Instances Instances New Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Scheduled Instances Capacity Reservations Images AMIs New AMI Catalog

Successfully terminated i-0b8fad27ffec1706b

Instances (1/2) Info Connect Instance state Actions Launch Instances

Name Instance ID Instance state Instance type Status check Alarm status Availability Zone Public IPv4

master	i-0b8fad27ffec1706b	Terminated	t2.micro	-	No alarms	us-east-1d	-
master	i-04e06c15187e92897	Pending	t2.micro	-	No alarms	us-east-1d	ec2-54-86-39-

Instance: i-04e06c15187e92897 (master)

Security groups sg-03517dd81c71cbc0c (launch-wizard-3)

Inbound rules

Security group rule ID	Port range	Protocol	Source	Security groups
sgr-0589fdf64cdccb8fc	22	TCP	0.0.0.0/0	launch-wizard-3

Outbound rules

https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#SecurityGroup:securityGroup... © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences ENG IN 08:57 01-11-2022

EC2 Management Console

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#SecurityGroup:securityGroupId=sg-03517dd81c71cbc0c

New EC2 Experience Tell us what you think

Services Search for services, features, blogs, docs, and more [Alt+S]

Cloud9 IAM EC2 Elastic Beanstalk AWS Amplify Lambda S3 API Gateway

EC2 Dashboard EC2 Global View Events Tags Limits Instances Instances New Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Scheduled Instances Capacity Reservations Images AMIs New AMI Catalog

Details

Security group name launch-wizard-3 Security group ID sg-03517dd81c71cbc0c Description launch-wizard-3 created 2022-11-01T03:26:50.382Z VPC ID vpc-06f870b50074eb753

Owner 689179663899 Inbound rules count 1 Permission entry Outbound rules count 1 Permission entry

Inbound rules Outbound rules Tags

You can now check network connectivity with Reachability Analyzer Run Reachability Analyzer

Inbound rules (1/1)

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sgr-0589fdf64cdccb8fc	IPv4	SSH	TCP	22

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 08:57 01-11-2022

EC2 Management Console x

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#ModifyInboundSecurityGroupRules:securityGroupId=sg-03517dd81c71cbc0c

Services Search for services, features, blogs, docs, and more [Alt+S]

Cloud9 IAM EC2 Elastic Beanstalk AWS Amplify Lambda S3 API Gateway

EC2 > Security Groups > sg-03517dd81c71cbc0c - launch-wizard-3 > Edit inbound rules

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info
sgr-0589fdf64cdccb8fc	SSH	TCP	22	Anywhere Info	<input type="text" value="0.0.0.0/0"/> X
-	All traffic	All	All	Custom Info	<input type="text" value="sg-03517dd81c71cbc0c"/> X

Add rule

Cancel Preview changes Save rules

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 08:58 01-11-2022

EC2 Management Console x

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:

Services Search for services, features, blogs, docs, and more [Alt+S]

Cloud9 IAM EC2 Elastic Beanstalk AWS Amplify Lambda S3 API Gateway

Name Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

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

Recents Quick Start

Amazon Linux	macOS	Ubuntu	Windows	Red Hat

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

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type
ami-0ee23bfc74a881de5 (64-bit (x86)) / ami-0551643488963af72 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible ?

Number of instances Info

Software Image (AMI)
Canonical, Ubuntu, 18.04 LTS, ... read more
ami-0ee23bfc74a881de5

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

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

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier ?

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 08:59 01-11-2022

The screenshot shows the AWS EC2 Management Console interface. A modal window titled "Launch Instance" is open, specifically the "Summary" step. It displays the following configuration:

- Number of instances:** 1
- Software Image (AMI):** Canonical, Ubuntu, 18.04 LTS, ami-0ee23bfc74a881de5
- Virtual server type (instance type):** t2.micro
- Firewall (security group):** launch-wizard-3
- Storage (volumes):** 1 volume(s) - 8 GiB

A tooltip for the "Free tier" is visible, stating: "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". At the bottom right of the modal is a large orange "Launch instance" button.

```
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added 'ec2-54-86-247-32.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1084-aws x86_64)

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

 System information as of Tue Nov  1 03:32:49 UTC 2022

 System load:  0.75      Processes:         96
 Usage of /:   16.1% of 7.57GB   Users logged in:     0
 Memory usage: 19%
 Swap usage:   0%

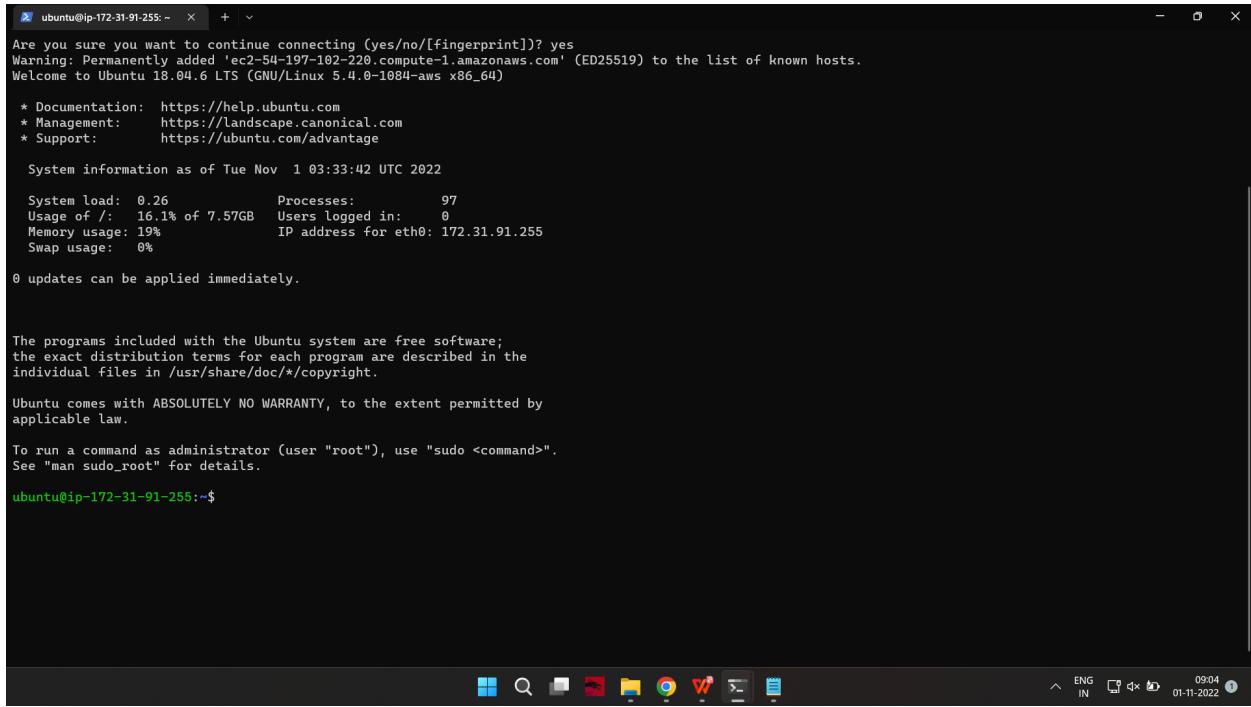
 0 updates can be applied immediately.

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-172-31-95-250:~$ |
```



```
ubuntu@ip-172-31-91-255: ~ + | 
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-197-102-220.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1084-aws x86_64)

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

System information as of Tue Nov  1 03:33:42 UTC 2022

System load:  0.26      Processes:         97
Usage of /:  16.1% of 7.57GB  Users logged in:    0
Memory usage: 19%          IP address for eth0: 172.31.91.255
Swap usage:   0%

0 updates can be applied immediately.

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-172-31-91-255:~$
```

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
```

```

ubuntu@ip-172-31-95-250:~ + | 
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-172-31-95-250:~$ 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 [83.3 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 [2785 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [513 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [980 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted Translation-en [136 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [1853 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe Translation-en [401 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [30.0 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse Translation-en [7532 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [53.2 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/main Translation-en [14.5 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [18.1 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe Translation-en [8668 kB]
Get:21 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [2444 kB]
Get:22 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [422 kB]
Get:23 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [950 kB]
Get:24 http://security.ubuntu.com/ubuntu bionic-security/restricted Translation-en [132 kB]
Get:25 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [1239 kB]
Get:26 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [284 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 kB]
Fetched 26.3 MB in 6s (4758 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-95-250:~$ |

```

^ ENG IN 09:05 01-11-2022

2.Next, install Docker with the command:
on-master&slave\$ sudo apt-get install docker.io

```

ubuntu@ip-172-31-95-250:~ + | 
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [980 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted Translation-en [136 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [1853 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe Translation-en [401 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [30.0 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse Translation-en [7532 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [53.2 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/main Translation-en [14.5 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [18.1 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe Translation-en [8668 kB]
Get:21 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [2444 kB]
Get:22 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [422 kB]
Get:23 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [950 kB]
Get:24 http://security.ubuntu.com/ubuntu bionic-security/restricted Translation-en [132 kB]
Get:25 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [1239 kB]
Get:26 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [284 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 kB]
Fetched 26.3 MB in 6s (4758 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-95-250:~$ 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 containedr 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 containedr docker.io pigz runc ubuntu-fan
0 upgraded, 6 newly installed, 0 to remove and 59 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 containedr 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]
53% [5 docker.io 0 B/36.9 MB 0%]

```

^ ENG IN 09:05 01-11-2022

```

ubuntu@ip-172-31-91-255:~$ sudo apt-get update
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe Translation-en [401 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [39.0 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse Translation-en [7532 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [53.2 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/main Translation-en [14.5 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [18.1 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe Translation-en [8668 kB]
Get:21 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [2444 kB]
Get:22 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [422 kB]
Get:23 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [958 kB]
Get:24 http://security.ubuntu.com/ubuntu bionic-security/restricted Translation-en [132 kB]
Get:25 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [1239 kB]
Get:26 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [284 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 kB]
Fetched 26.3 MB in 5s (4815 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-91-255:~$ sudo apt-get install docker.io -y
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 59 not upgraded.
Need to get 74.2 MB of archives.
After this operation, 360 MB of additional disk space will be used.
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 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 main amd64 ubuntu-fan all 0.12.10 [34.7 kB]
Fetched 74.2 MB in 2s (39.0 MB/s)
Preconfiguring packages...
Selecting previously unselected package pigz.
(Reading database ... 65%

```

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

```

ubuntu@ip-172-31-95-256:~$ docker --version
Docker version 20.10.7, build 20.10.7-0ubuntu5~18.04.3
ubuntu@ip-172-31-95-256:~$ |

```

```

Docker version 20.10.7, build 20.10.7-0ubuntu5~18.04.3
ubuntu@ip-172-31-91-255:~$ |

```

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-172-31-95-256:~$ sudo systemctl enable docker
ubuntu@ip-172-31-95-256:~$ |

```

```

ubuntu@ip-172-31-91-255:~$ sudo systemctl enable docker
ubuntu@ip-172-31-91-255:~$ |

```

2. Verify Docker is running:

on-master&slave\$sudo systemctl status docker

```
ubuntu@ip-172-31-95-250:~ % + | 
Unpacking ubuntu-fan (0.12.10) ...
Setting up runc (1.0.1-0ubuntu2~18.04.1) ...
Setting up containerd (1.5.5-0ubuntu3~18.04.2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
Setting up bridge-utils (1.5-15ubuntu1) ...
Setting up ubuntu-fan (0.12.10) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /lib/systemd/system/ubuntu-fan.service.
Setting up pigz (2.4-1) ...
Setting up docker.io (20.10.7-0ubuntu5~18.04.3) ...
Adding group 'docker' (GID 115) ...
Done.
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/multi-user.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Processing triggers for systemd (237-3ubuntu0.53) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
ubuntu@ip-172-31-95-250:~$ docker --version
Docker version 20.10.7, build 20.10.7-0ubuntu5~18.04.3
ubuntu@ip-172-31-95-250:~$ sudo systemctl enable docker
ubuntu@ip-172-31-95-250:~$ 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 2022-11-01 03:36:07 UTC; 1min 35s ago
       Docs: https://docs.docker.com
 Main PID: 2914 (dockerd)
    Tasks: 8
   CGroup: /system.slice/docker.service
           └─2914 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Nov 01 03:36:06 ip-172-31-95-250 dockerd[2914]: time="2022-11-01T03:36:06.900398948Z" level=warning msg="Your kernel does not support CPU realtime scheduler"
Nov 01 03:36:06 ip-172-31-95-250 dockerd[2914]: time="2022-11-01T03:36:06.900578958Z" level=warning msg="Your kernel does not support cgroup blkio weight"
Nov 01 03:36:06 ip-172-31-95-250 dockerd[2914]: time="2022-11-01T03:36:06.900718675Z" level=warning msg="Your kernel does not support cgroup blkio weight_de"
Nov 01 03:36:06 ip-172-31-95-250 dockerd[2914]: time="2022-11-01T03:36:06.9001056474Z" level=info msg="Loading containers: start."
Nov 01 03:36:07 ip-172-31-95-250 dockerd[2914]: time="2022-11-01T03:36:07.073138582Z" level=info msg="Default bridge (docker0) is assigned with an IP address"
Nov 01 03:36:07 ip-172-31-95-250 dockerd[2914]: time="2022-11-01T03:36:07.145953838Z" level=info msg="Loading containers: done."
Nov 01 03:36:07 ip-172-31-95-250 dockerd[2914]: time="2022-11-01T03:36:07.228042050Z" level=info msg="Docker daemon" commit="20.10.7-0ubuntu5~18.04.3" graph
Nov 01 03:36:07 ip-172-31-95-250 dockerd[2914]: time="2022-11-01T03:36:07.228423988Z" level=info msg="Daemon has completed initialization"
Nov 01 03:36:07 ip-172-31-95-250 systemd[1]: Started Docker Application Container Engine.
Nov 01 03:36:07 ip-172-31-95-250 dockerd[2914]: time="2022-11-01T03:36:07.280381571Z" level=info msg="API listen on /var/run/docker.sock"
lines 1-19/19 (END)
```

```
ubuntu@ip-172-31-91-255:~ % + | 
Unpacking ubuntu-fan (0.12.10) ...
Setting up runc (1.0.1-0ubuntu2~18.04.1) ...
Setting up containerd (1.5.5-0ubuntu3~18.04.2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
Setting up bridge-utils (1.5-15ubuntu1) ...
Setting up ubuntu-fan (0.12.10) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /lib/systemd/system/ubuntu-fan.service.
Setting up pigz (2.4-1) ...
Setting up docker.io (20.10.7-0ubuntu5~18.04.3) ...
Adding group 'docker' (GID 115) ...
Done.
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Processing triggers for systemd (237-3ubuntu0.53) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
ubuntu@ip-172-31-91-255:~$ docker --version
Docker version 20.10.7, build 20.10.7-0ubuntu5~18.04.3
ubuntu@ip-172-31-91-255:~$ sudo systemctl enable docker
ubuntu@ip-172-31-91-255:~$ 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 2022-11-01 03:36:24 UTC; 1min 23s ago
       Docs: https://docs.docker.com
 Main PID: 2927 (dockerd)
    Tasks: 8
   CGroup: /system.slice/docker.service
           └─2927 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Nov 01 03:36:23 ip-172-31-91-255 dockerd[2927]: time="2022-11-01T03:36:23.758736733Z" level=warning msg="Your kernel does not support CPU realtime scheduler"
Nov 01 03:36:23 ip-172-31-91-255 dockerd[2927]: time="2022-11-01T03:36:23.758883102Z" level=warning msg="Your kernel does not support cgroup blkio weight"
Nov 01 03:36:23 ip-172-31-91-255 dockerd[2927]: time="2022-11-01T03:36:23.759027465Z" level=warning msg="Your kernel does not support cgroup blkio weight_de"
Nov 01 03:36:23 ip-172-31-91-255 dockerd[2927]: time="2022-11-01T03:36:23.759357453Z" level=info msg="Loading containers: start."
Nov 01 03:36:24 ip-172-31-91-255 dockerd[2927]: time="2022-11-01T03:36:24.021160961Z" level=info msg="Default bridge (docker0) is assigned with an IP address"
Nov 01 03:36:24 ip-172-31-91-255 dockerd[2927]: time="2022-11-01T03:36:24.098826650Z" level=info msg="Loading containers: done."
Nov 01 03:36:24 ip-172-31-91-255 dockerd[2927]: time="2022-11-01T03:36:24.197153623Z" level=info msg="Docker daemon" commit="20.10.7-0ubuntu5~18.04.3" graph
Nov 01 03:36:24 ip-172-31-91-255 dockerd[2927]: time="2022-11-01T03:36:24.197550008Z" level=info msg="Daemon has completed initialization"
Nov 01 03:36:24 ip-172-31-91-255 systemd[1]: Started Docker Application Container Engine.
Nov 01 03:36:24 ip-172-31-91-255 dockerd[2927]: time="2022-11-01T03:36:24.259082902Z" level=info msg="API listen on /var/run/docker.sock"
lines 1-19/19 (END)
```

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

3.Repeat on all the other nodes.

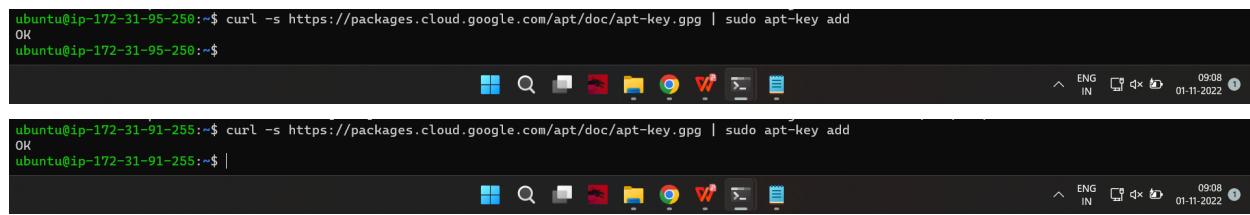
Install Kubernetes

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-172-31-95-250:~$ curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add
OK
ubuntu@ip-172-31-95-250:~$
```

```
ubuntu@ip-172-31-91-255:~$ curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add
OK
ubuntu@ip-172-31-91-255:~$
```

If you get an error that curl is not installed, install it with:

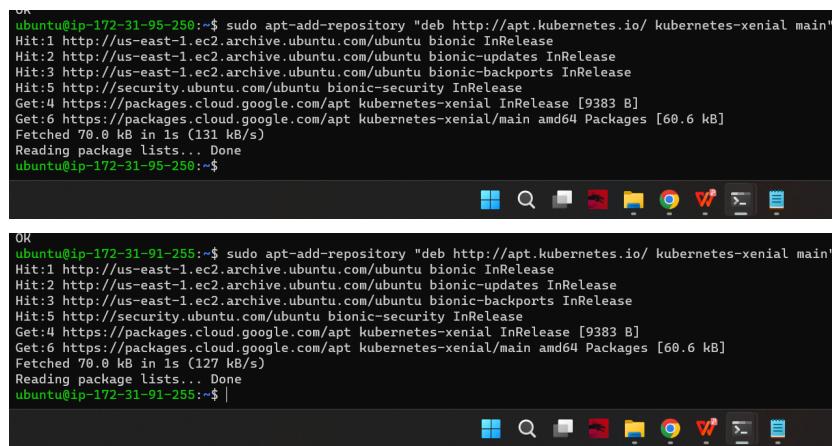
```
on-master&slave$sudo apt-get install curl
```

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"
```



```
ubuntu@ip-172-31-95-250:~$ 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 [60.6 kB]
Fetched 70.0 kB in 1s (131 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-95-250:~$
```

```
OK
ubuntu@ip-172-31-91-255:~$ 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 [60.6 kB]
Fetched 70.0 kB in 1s (127 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-91-255:~$
```

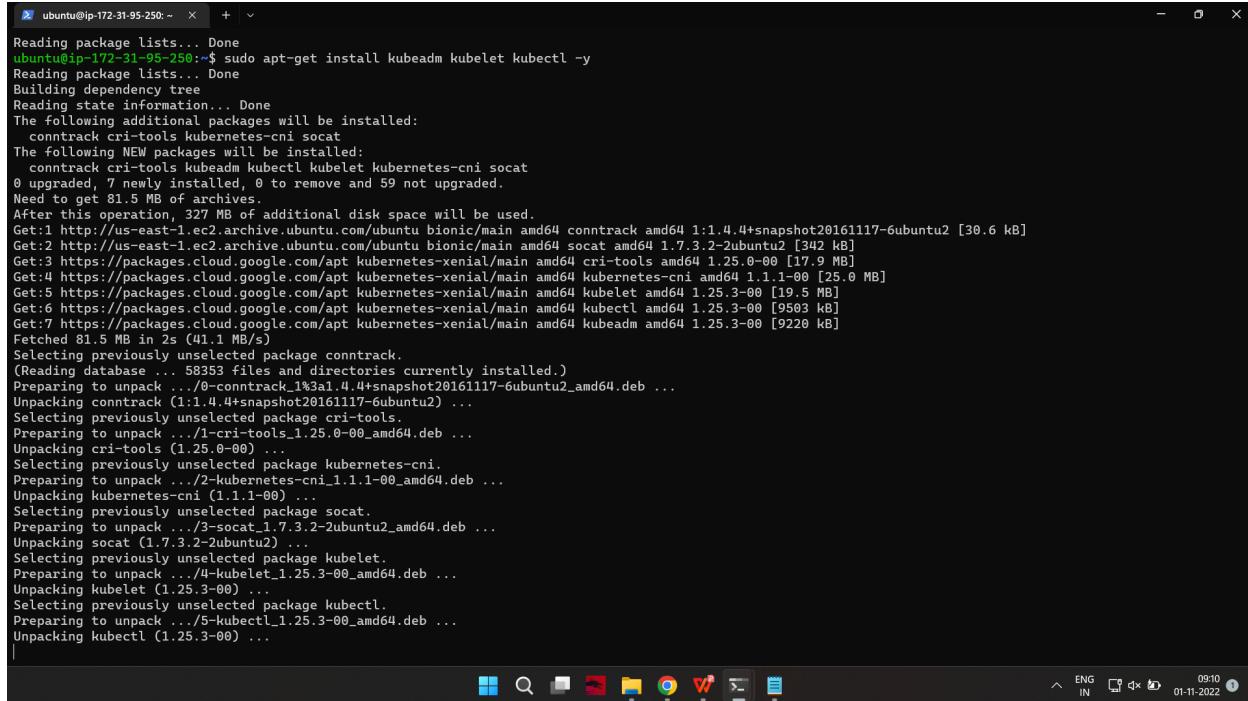
Repeat on each server node.

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
```



```
ubuntu@ip-172-31-95-250: ~ + |  
Reading package lists... Done  
ubuntu@ip-172-31-95-250:~$ 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 kubectl kubernetes-cni socat  
0 upgraded, 7 newly installed, 0 to remove and 59 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 1.25.3-00 [19.5 MB]  
Get:6 https://packages.cloud.google.com/apt/kubernetes-xenial/main amd64 kubectl amd64 1.25.3-00 [9503 kB]  
Get:7 https://packages.cloud.google.com/apt/kubernetes-xenial/main amd64 kubeadm amd64 1.25.3-00 [9220 kB]  
Fetched 81.5 MB in 2s (41.1 MB/s)  
Selecting previously unselected package conntrack.  
(Reading database ... 58353 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.  
Preparing to unpack .../2-kubernetes-cni_1.1.1-00_amd64.deb ...  
Unpacking kubernetes-cni (1.1.1-00) ...  
Selecting previously unselected package socat.  
Preparing to unpack .../3-socat_1.7.3.2-2ubuntu2_amd64.deb ...  
Unpacking socat (1.7.3.2-2ubuntu2) ...  
Selecting previously unselected package kubelet.  
Preparing to unpack .../4-kubelet_1.25.3-00_amd64.deb ...  
Unpacking kubelet (1.25.3-00) ...  
Selecting previously unselected package kubectl.  
Preparing to unpack .../5-kubectl_1.25.3-00_amd64.deb ...  
Unpacking kubectl (1.25.3-00) ...
```

```

ubuntu@ip-172-31-91-255:~$ 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 kubectl kubernetes-cni socat
0 upgraded, 7 newly installed, 0 to remove and 59 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 kubeadm amd64 1.1.1-00 [25.0 MB]
Get:5 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubelet amd64 1.25.3-00 [19.5 MB]
Get:6 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubectl amd64 1.25.3-00 [9503 kB]
Get:7 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubeadm amd64 1.25.3-00 [9220 kB]
Fetched 81.5 MB in 2s (37.9 MB/s)
Selecting previously unselected package conntrack.
(Reading database ... 58353 files and directories currently installed.)
Preparing to unpack .../0-conntrack_1%3ai.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.
Preparing to unpack .../2-kubernetes-cni_1.1.1-00_amd64.deb ...
Unpacking kubernetes-cni (1.1.1-00) ...
Selecting previously unselected package socat.
Preparing to unpack .../3-socat_1.7.3.2-2ubuntu2_amd64.deb ...
Unpacking socat (1.7.3.2-2ubuntu2) ...
Selecting previously unselected package kubelet.
Preparing to unpack .../4-kubelet_1.25.3-00_amd64.deb ...
Unpacking kubelet (1.25.3-00) ...
Selecting previously unselected package kubectl.
Preparing to unpack .../5-kubectl_1.25.3-00_amd64.deb ...
Unpacking kubectl (1.25.3-00) ...
Selecting previously unselected package kubeadm.
Preparing to unpack .../6-kubeadm_1.25.3-00_amd64.deb ...

```

```

ubuntu@ip-172-31-95-256:~$ sudo apt-mark hold kubeadm kubelet kubectl
kubeadm set on hold.
kubelet set on hold.
kubectl set on hold.
ubuntu@ip-172-31-95-256:~$

```

```

ubuntu@ip-172-31-91-255:~$ sudo apt-mark hold kubeadm kubelet kubectl
kubeadm set on hold.
kubelet set on hold.
kubectl set on hold.
ubuntu@ip-172-31-91-255:~$

```

Allow the process to complete.

Verify the installation with:

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

```

ubuntu@ip-172-31-95-256:~$ kubeadm version
kubeadm version: &version.Info{Major:"1", Minor:"25", GitVersion:"v1.25.3", GitCommit:"434bfd82814af038ad94d62ebe59b133fc50506", GitTreeState:"clean", BuildDate:"2022-10-12T10:55:36Z", GoVersion:"go1.19.2", Compiler:"gc", Platform:"linux/amd64"}
ubuntu@ip-172-31-95-256:~|

```

```

ubuntu@ip-172-31-91-255:~$ kubeadm version
kubeadm version: &version.Info{Major:"1", Minor:"25", GitVersion:"v1.25.3", GitCommit:"434bfd82814af038ad94d62ebe59b133fc50506", GitTreeState:"clean", BuildDate:"2022-10-12T10:55:36Z", GoVersion:"go1.19.2", Compiler:"gc", Platform:"linux/amd64"}
ubuntu@ip-172-31-91-255:~$

```

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**

```
ubuntu@ip-172-31-91-255:~$ sudo swapoff --a  
ubuntu@ip-172-31-91-255:~$ sudo hostnamectl
```

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-172-31-91-255:~$ sudo hostnamectl set-hostname master-node  
ubuntu@ip-172-31-91-255:~$
```

Next, set a worker node hostname by entering the following on the worker server:

on-slave\$ sudo hostnamectl set-hostname worker01

```
ubuntu@ip-172-31-91-255:~$ sudo hostnamectl set-hostname worker  
ubuntu@ip-172-31-91-255:~$
```

If you have additional worker nodes, use this process to set a unique hostname 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

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 172.31.95.250:6443 --token h1ootv.7y591xc9d8faqqqr4 \  
--discovery-token-ca-cert-hash sha256:17d5d14955c522d906d7110a74cbe4ef54e60559e905985bcf9fde17d87302f3  
sha256:17d5d14955c522d906d7110a74cbe4ef54e60559e905985bcf9fde17d87302f3
```

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

```
kubeadm join 172.31.95.250:6443 --token h1ootv.7y591xc9d8faqqqr4 \  
--discovery-token-ca-cert-hash sha256:17d5d14955c522d906d7110a74cbe4ef54e60559e905985bcf9fde17d87302f3  
ubuntu@ip-172-31-95-250:~$ mkdir -p $HOME/.kube  
ubuntu@ip-172-31-95-250:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config  
ubuntu@ip-172-31-95-250:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config  
ubuntu@ip-172-31-95-250:~$
```

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.yaml
```

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

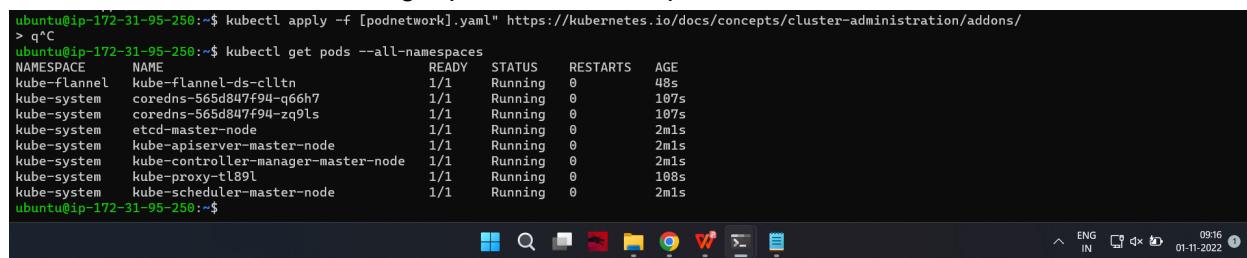
```
kubectl apply -f [podnetwork].yaml"
```

```
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-172-31-95-250:~$ kubectl apply -f [podnetwork].yaml" https://kubernetes.io/docs/concepts/cluster-administration/addons/
> q^C
ubuntu@ip-172-31-95-250:~$ kubectl get pods --all-namespaces
NAMESPACE      NAME                READY   STATUS    RESTARTS   AGE
kube-flannel   kube-flannel-ds-cltn   1/1     Running   0          48s
kube-system    coredns-565d847f94-q66h7   1/1     Running   0          107s
kube-system    coredns-565d847f94-zq9ls   1/1     Running   0          107s
kube-system    etcd-master-node       1/1     Running   0          2m1s
kube-system    kube-apiserver-master-node 1/1     Running   0          2m1s
kube-system    kube-controller-manager-master-node 1/1     Running   0          2m1s
kube-system    kube-proxy-tl89l       1/1     Running   0          108s
kube-system    kube-scheduler-master-node 1/1     Running   0          2m1s
ubuntu@ip-172-31-95-250:~$
```

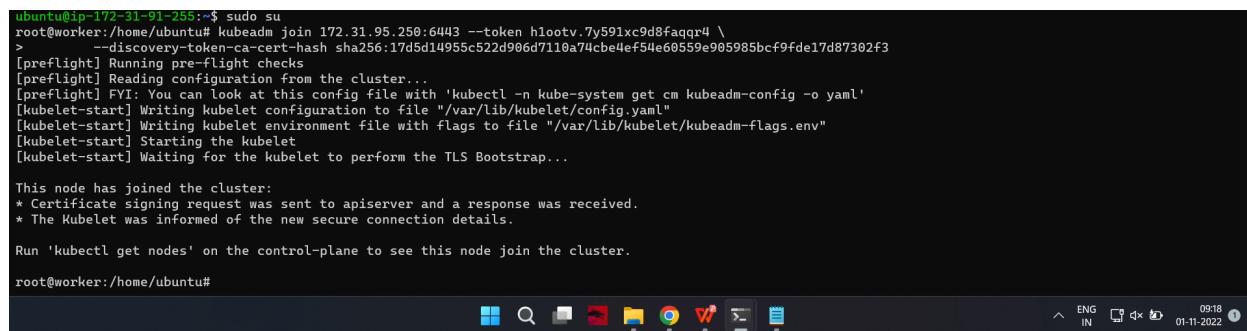
Step 10: Join Worker Node to Cluster

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.hcjim6j95j6ilt1tl \
--discovery-token-ca-cert-hash
```

```
sha256:4c13f7d2b7dc771a0380df19969ea5e3fc920c24611072dee907269ece7e5033
```



```
ubuntu@ip-172-31-91-255:~$ sudo su
root@worker:/home/ubuntu# kubeadm join 172.31.95.250:6443 --token hlootv.7y591xc9d8faqqr4 \
--discovery-token-ca-cert-hash sha256:17d5d14955c522d986d7110a74cbe4ef54e60559e905985bcf9fde17d87302f3
[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.

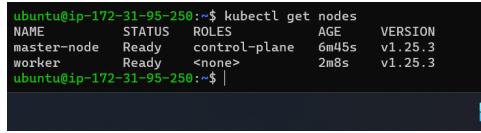
Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
root@worker:/home/ubuntu#
```

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-172-31-95-250:~$ kubectl get nodes
NAME     STATUS   ROLES      AGE     VERSION
master-node  Ready    control-plane   6m45s   v1.25.3
worker     Ready    <none>    2m8s   v1.25.3
ubuntu@ip-172-31-95-250:~$ |
```

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

Output

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

master	Ready	master	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
```

You can now deploy any containerized application to your cluster. To keep things familiar, let's deploy Nginx using Deployments and Services to see how this application can be deployed to the cluster. You can use the commands below for other containerized applications as well, provided you change the Docker image name and any relevant flags (such as ports and volumes).

Still within the master node, execute the following command to create a deployment named nginx:

```
kubernetes-master:~$kubectl create deployment nginx --image=nginx
```

```
ubuntu@master-node:~$ kubectl create deployment nginx --image=nginx
deployment.apps/nginx created
ubuntu@master-node:~$ kubectl expose deploy nginx --port 80 --target-port 80 --type NodePort
service/nginx exposed
```

A deployment is a type of Kubernetes object that ensures there's always a specified number of pods running based on a defined template, even if the pod crashes during the cluster's lifetime. The above deployment will create a pod with one container from the Docker registry's Nginx Docker Image.

Next, run the following command to create a service named nginx that will expose the app publicly. It will do so through a NodePort, a scheme that will make the pod accessible through an arbitrary port opened on each node of the cluster:

```
kubernetes-master:~$kubectl expose deploy nginx --port 80 --target-port 80 --type NodePort
```

```
ubuntu@master-node:~$ kubectl create deployment nginx --image=nginx
deployment.apps/nginx created
ubuntu@master-node:~$ kubectl expose deploy nginx --port 80 --target-port 80 --type NodePort
service/nginx exposed
```

Services are another type of Kubernetes object that expose cluster internal services to clients, both internal and external. They are also capable of load balancing requests to multiple pods, and are an integral component in Kubernetes, frequently interacting with other components.

Run the following command:

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

```
ubuntu@master-node:~$ kubectl get services
NAME      TYPE    CLUSTER-IP   EXTERNAL-IP  PORT(S)        AGE
kubernetes  ClusterIP  10.96.0.1    <none>        443/TCP      9m39s
nginx      NodePort   10.99.39.69  <none>        80:30654/TCP  23s
ubuntu@master-node:~$
```

This will output text similar to the following:

Output

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	1d
nginx	NodePort	10.109.228.209	<none>	80:nginx_port/TCP	40m

From the third line of the above output, you can retrieve the port that Nginx is running on. Kubernetes will assign a random port that is greater than 30000 automatically, while ensuring that the port is not already bound by another service.

Note: if you're running your setup on ec2 ensure the nginx_port is open under the inbound rules in the security groups.

To test that everything is working, visit

`http://worker_1_ip:nginx_port or
http://worker_2_ip:nginx_port`

through a browser on your local machine. You will see Nginx's familiar welcome page. To see the deployed container on the worker node switch to worker01

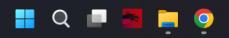
on-slave#`docker ps`

Output: you will see the container for nginx image running.

If you want to scale up the replicas for a deployment (nginx in our case) the use the following command:

kubernetes-master:~\$`kubectl scale --current-replicas=1 --replicas=2 deployment/nginx`

```
ubuntu@master-node:~$ kubectl scale --current-replicas=1 --replicas=2 deployment/nginx
deployment.apps/nginx scaled
ubuntu@master-node:~$ |
```



kubernetes-master:~\$`kubectl get pods`

```
ubuntu@master-node:~$ kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
nginx-76d6c9b8c-kg7x9  1/1     Running   0          4m12s
nginx-76d6c9b8c-nh2ws  1/1     Running   0          9s
ubuntu@master-node:~$ |
```



Output: you will see 2/2 as output in nginx deployment.

kubernetes-master:~\$`kubectl describe deployment/nginx`

```
ubuntu@master-node:~$ kubectl describe deployment/nginx
Name:           nginx
Namespace:      default
CreationTimestamp:  Tue, 01 Nov 2022 03:53:14 +0000
Labels:         app=nginx
Annotations:    deployment.kubernetes.io/revision: 1
Selector:       app=nginx
Replicas:      2 desired | 2 updated | 2 total | 2 available | 0 unavailable
StrategyType:   RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx
  Containers:
    nginx:
      Image:      nginx
      Port:       <none>
      Host Port:  <none>
      Environment: <none>
      Mounts:     <none>
      Volumes:    <none>
  Conditions:
    Type        Status  Reason
    ----        ----  -----
    Progressing  True    NewReplicaSetAvailable
    Available   True    MinimumReplicasAvailable
  OldReplicaSets: <none>
  NewReplicaSet:  nginx-76d6c9b8c (2/2 replicas created)
Events:
  Type        Reason     Age   From            Message
  ----        -----     --   --   -----
  Normal  ScalingReplicaSet  4m44s  deployment-controller  Scaled up replica set nginx-76d6c9b8c to 1
  Normal  ScalingReplicaSet  41s   deployment-controller  Scaled up replica set nginx-76d6c9b8c to 2 from 1
ubuntu@master-node:~$ |
```



Output: give details about the service deployed

If you would like to remove the Nginx application, first delete the nginx service from the master node:

```
kubernetes-master:~$kubectl delete service nginx
```

```
Normal  StartingHTTPBackend  4ms  deployment-controller  Scaled up ReplicaSet nginx-700dc98c to 2 from 1
ubuntu@master-node:~$ kubectl delete service nginx
service "nginx" deleted
ubuntu@master-node:~$ kubectl get services
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kubernetes  ClusterIP  10.96.0.1    <none>        443/TCP      15m
ubuntu@master-node:~$ |
```

Run the following to ensure that the service has been deleted:

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

You will see the following output:

```
Normal  StartingHTTPBackend  4ms  deployment-controller  Scaled up ReplicaSet nginx-700dc98c to 2 from 1
ubuntu@master-node:~$ kubectl delete service nginx
service "nginx" deleted
ubuntu@master-node:~$ kubectl get services
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kubernetes  ClusterIP  10.96.0.1    <none>        443/TCP      15m
ubuntu@master-node:~$ |
```

Output

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	1d

Then delete the deployment:

```
kubernetes-master:~$kubectl delete deployment nginx
```

```
ubuntu@master-node:~$ kubectl delete deployment nginx
deployment.apps "nginx" deleted
ubuntu@master-node:~$ kubectl get deployments
No resources found in default namespace.
ubuntu@master-node:~$ |
```

Run the following to confirm that this worked:

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

```
ubuntu@master-node:~$ kubectl delete deployment nginx
deployment.apps "nginx" deleted
ubuntu@master-node:~$ kubectl get deployments
No resources found in default namespace.
ubuntu@master-node:~$ |
```

Output

No resources found.