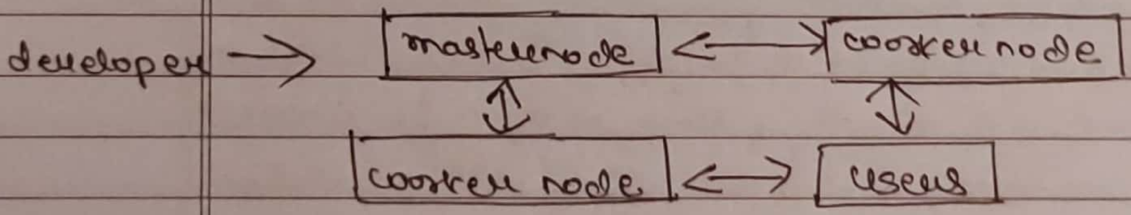
Mayuri Yerande
D15B
70

EXPERIMENT – 3

AIM: To understand the kubernetes cluster architecture, install and spin up a kuberenetes cluster on Linux machines/Cloud Platforms.

THEORY:

Kubernetes is an open source platform for managing container technologies such as Docker. Docker lets you create containers for a pre-configured image and application. Kubernetes provides the next step allowing you to balance loads between containers and run multiple containers across multiple system. Kubernetes makes it easy to deploy and operate application in a microservice architecture. It does so by creating an abstraction layer on top of a group of hosts so that development teams can deploy their applications and let kubernetes manage other activities.



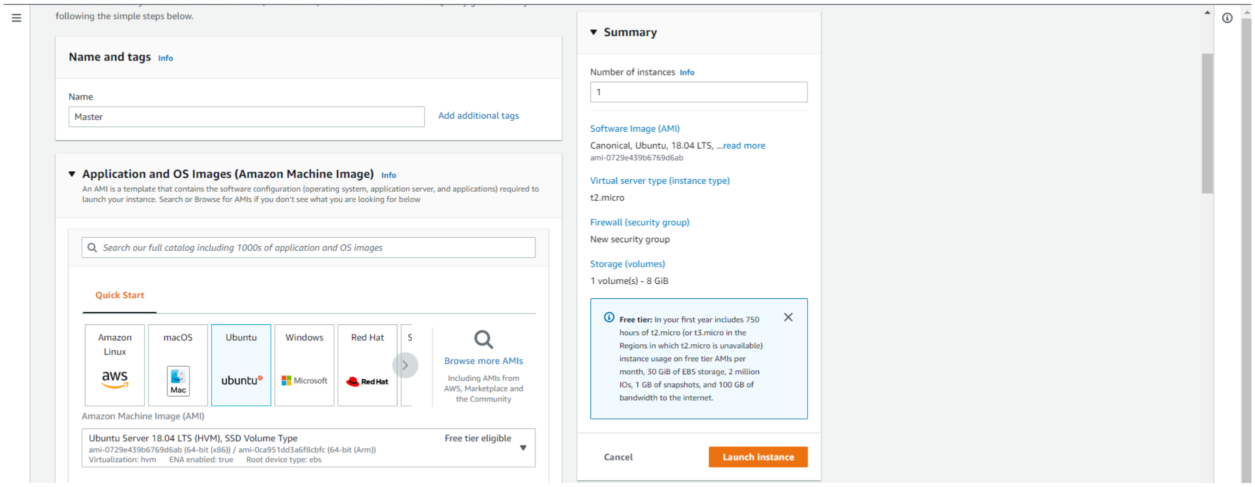
Kubernetes continously monitors the elements of the cluster to make sure the current state of application does not vary from desired state.

Kubernetes Installation on Ubuntu 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)

First Create three EC-2 Instances on AWS with Ubuntu 18.0

Create Master



following the simple steps below.

Name and tags Info

Name: Master

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

Amazon Machine Image (AMI)

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-0729e439b6769d6ab (64-bit (x86)) / ami-0ca951dd3a6f8cbfc (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Summary

Number of instances Info: 1

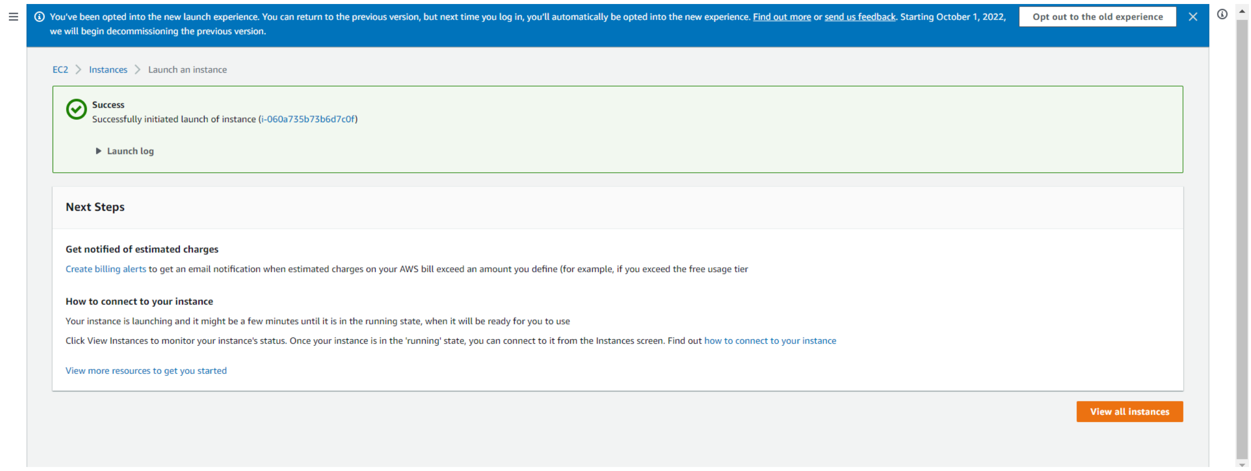
Software Image (AMI)
Canonical, Ubuntu, 18.04 LTS, ...read more
ami-0729e439b6769d6ab

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 AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.



You've been opted into the new launch experience. You can return to the previous version, but next time you log in, you'll automatically be opted into the new experience. Find out more or send us feedback. Starting October 1, 2022, we will begin decommissioning the previous version.

Opt out to the old experience
EC2 > Instances > Launch an instance

Success
Successfully initiated launch of instance (i-060a735b73b6d7c0f)

Next Steps

Get notified of estimated charges

Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier

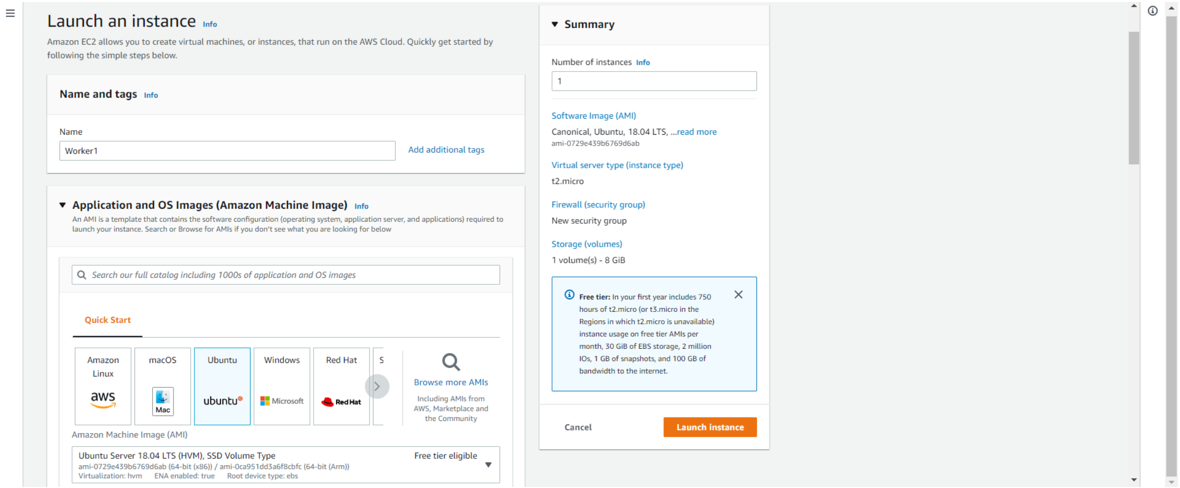
How to connect to your instance

Your instance is launching and it might be a few minutes until it is in the running state, when it will be ready for you to use

Click View Instances to monitor your instance's status. Once your instance is in the 'running' state, you can connect to it from the Instances screen. Find out how to connect to your instance

View more resources to get you started

Create worker 1



Launch an instance

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags

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

Amazon Machine Image (AMI)

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-0729e439b6769d6ab (64-bit (x86)) / ami-0ca951dd3a6f8cbfc (64-bit (Arm))
Virtualization: hvm   ENA enabled: true   Root device type: ebs

▼ Summary

Number of instances: 1

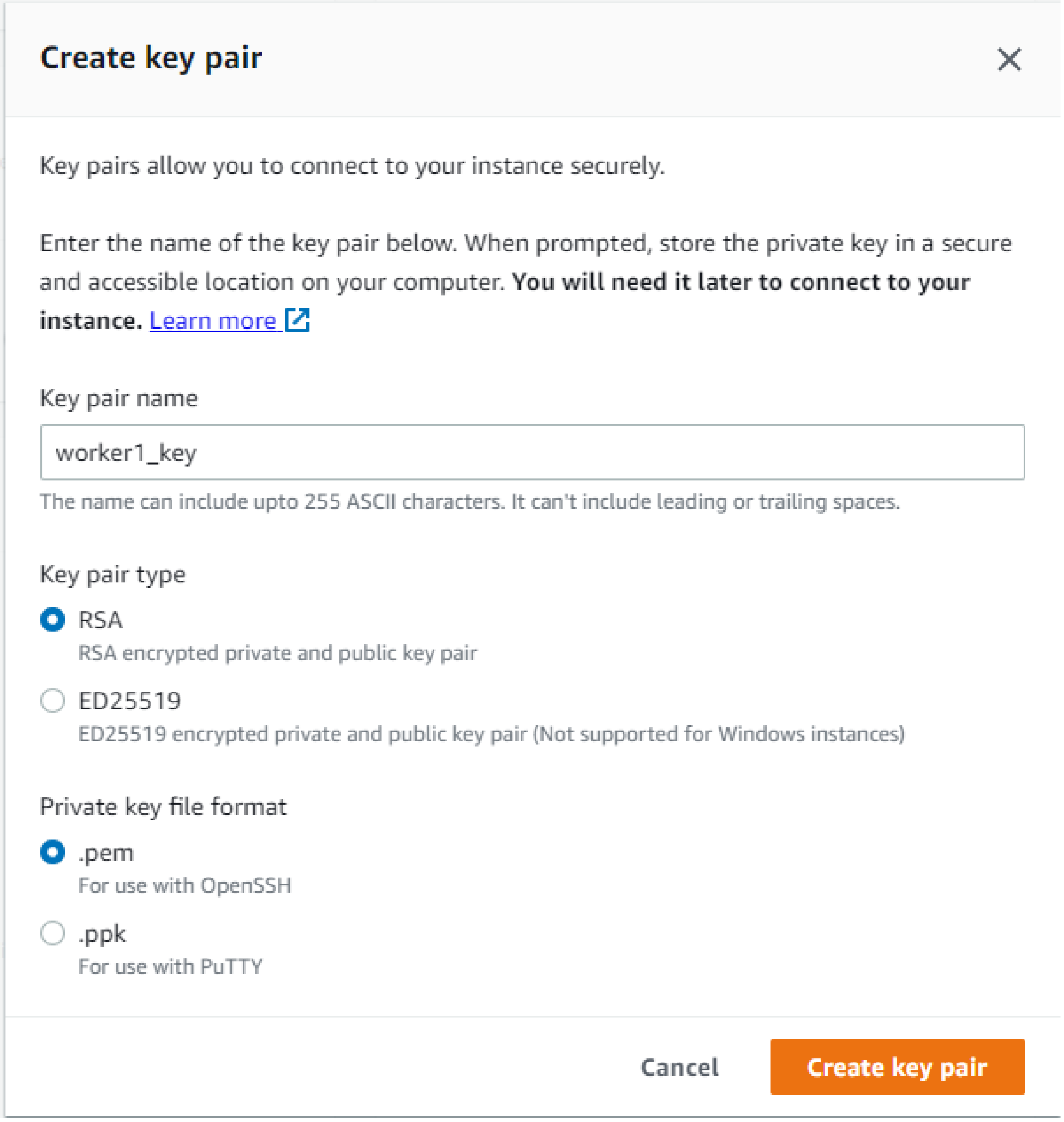
Software Image (AMI)
Canonical, Ubuntu, 18.04 LTS, ...read more
ami-0729e439b6769d6ab

Virtual server type (instance type)
t2.micro

Firewall (security group)
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 AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.



Create key pair

Key pairs allow you to connect to your instance securely.

Enter the name of the key pair below. When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** Learn more

Key pair name

worker1_key

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

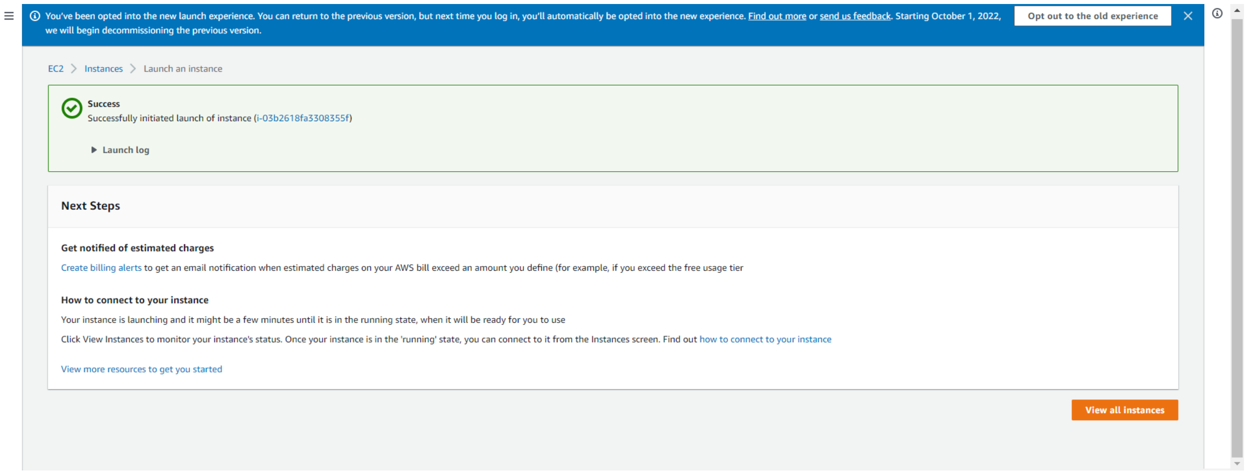
● RSA
RSA encrypted private and public key pair

○ ED25519
ED25519 encrypted private and public key pair (Not supported for Windows instances)

Private key file format

● .pem
For use with OpenSSH

○ .ppk
For use with PuTTY



You've been opted into the new launch experience. You can return to the previous version, but next time you log in, you'll automatically be opted into the new experience. Find out more or send us feedback. Starting October 1, 2022, we will begin decommissioning the previous version.

Opt out to the old experience

EC2 > Instances > Launch an instance

Success
Successfully initiated launch of instance (i-03b2618fa3308355f)

Next Steps

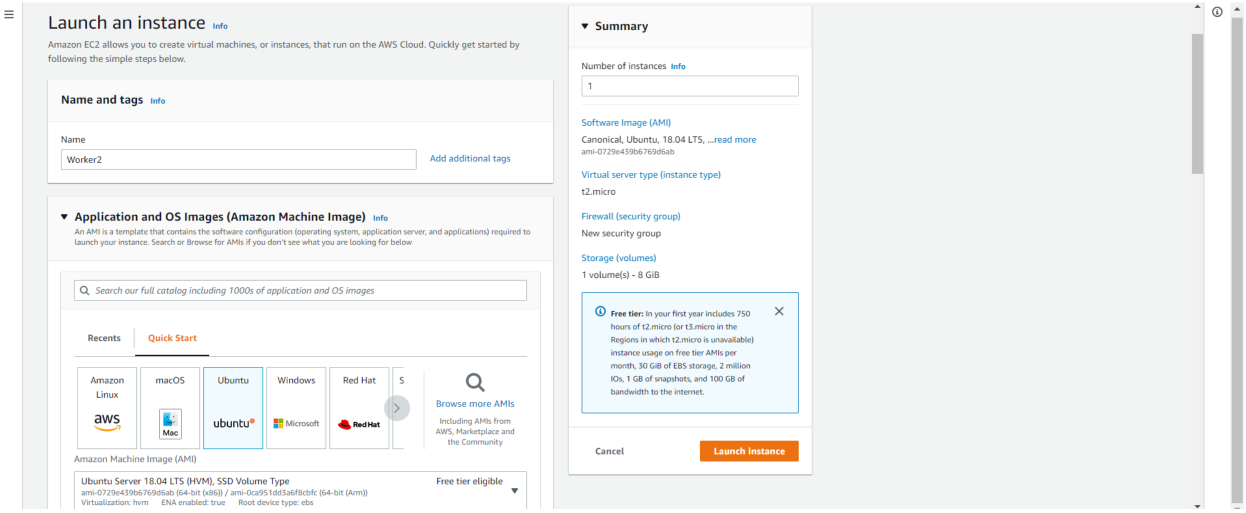
Get notified of estimated charges
Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier

How to connect to your instance
Your instance is launching and it might be a few minutes until it is in the running state, when it will be ready for you to use

Click View Instances to monitor your instance's status. Once your instance is in the 'running' state, you can connect to it from the Instances screen. Find out how to connect to your instance

View more resources to get you started

Create worker 2



Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: Worker2

▼ 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

Amazon Machine Image (AMI)
Ubuntu Server 18.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-0729e439b6769d6ab (64-bit (x86)) / ami-0ca951dd3a6f8cbfc (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

▼ Summary

Number of instances Info: 1

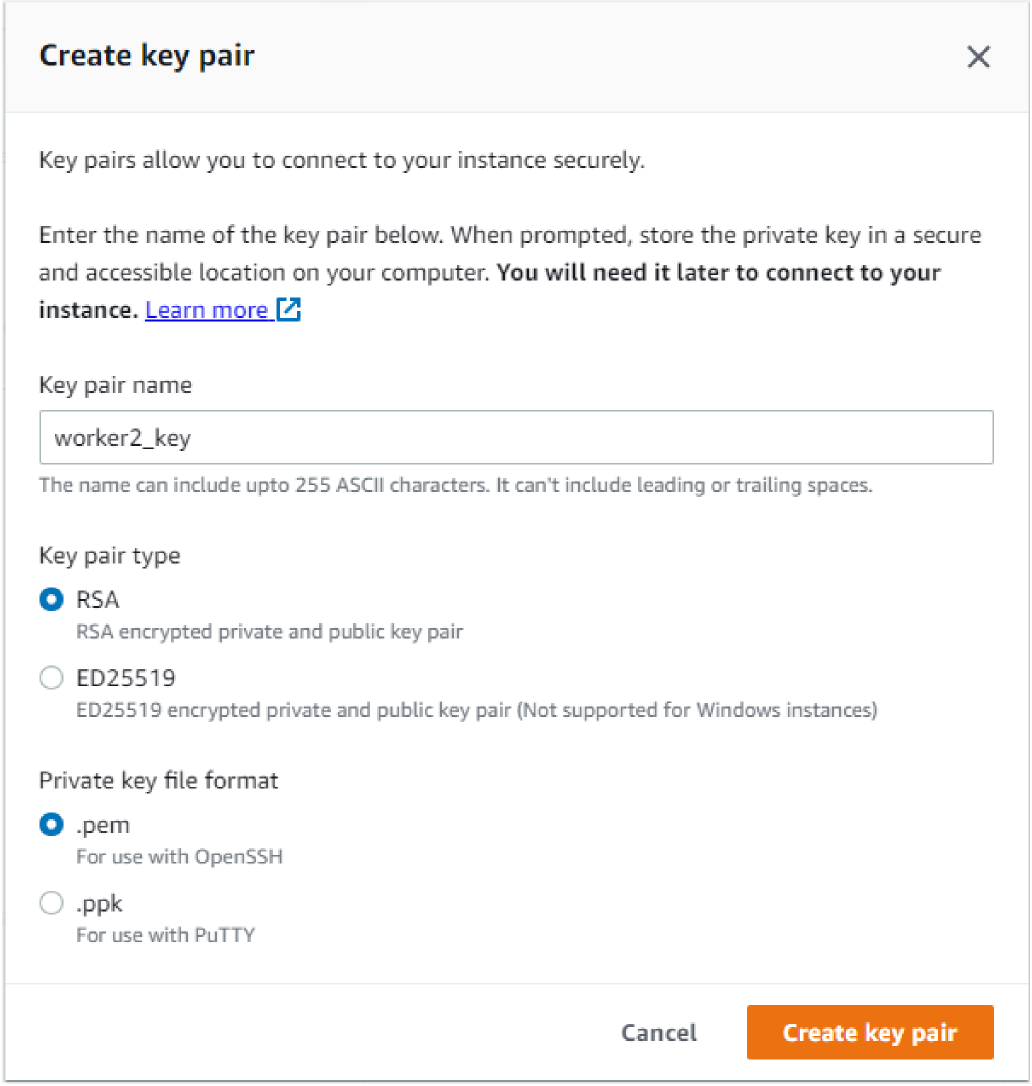
Software Image (AMI)
Canonical, Ubuntu, 18.04 LTS, ...read more
ami-0729e439b6769d6ab

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 AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.



Create key pair

Key pairs allow you to connect to your instance securely.

Enter the name of the key pair below. When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** Learn more

Key pair name

worker2_key

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

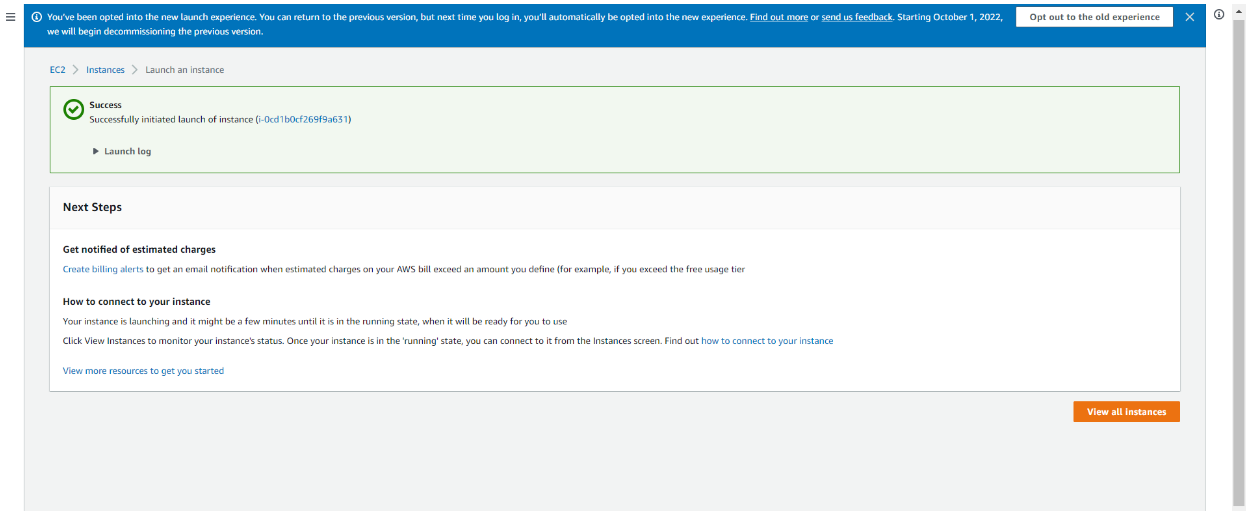
● RSA
RSA encrypted private and public key pair

○ ED25519
ED25519 encrypted private and public key pair (Not supported for Windows instances)

Private key file format

● .pem
For use with OpenSSH

○ .ppk
For use with PuTTY



You've been opted into the new launch experience. You can return to the previous version, but next time you log in, you'll automatically be opted into the new experience. Find out more or send us feedback. Starting October 1, 2022, we will begin decommissioning the previous version.

EC2 > Instances > Launch an instance

Success
Successfully initiated launch of instance (i-0cd1b0cf269f9a631)

▶ Launch log

Next Steps

Get notified of estimated charges

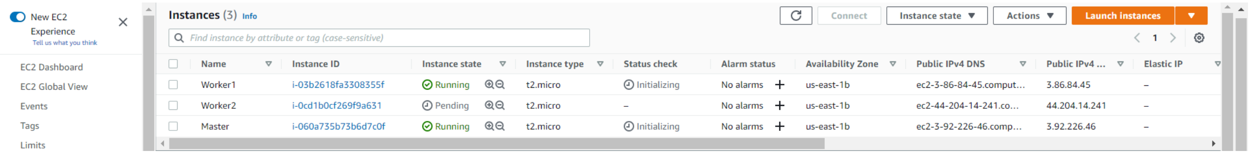
Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier

How to connect to your instance

Your instance is launching and it might be a few minutes until it is in the running state, when it will be ready for you to use

Click View Instances to monitor your instance's status. Once your instance is in the 'running' state, you can connect to it from the Instances screen. Find out how to connect to your instance

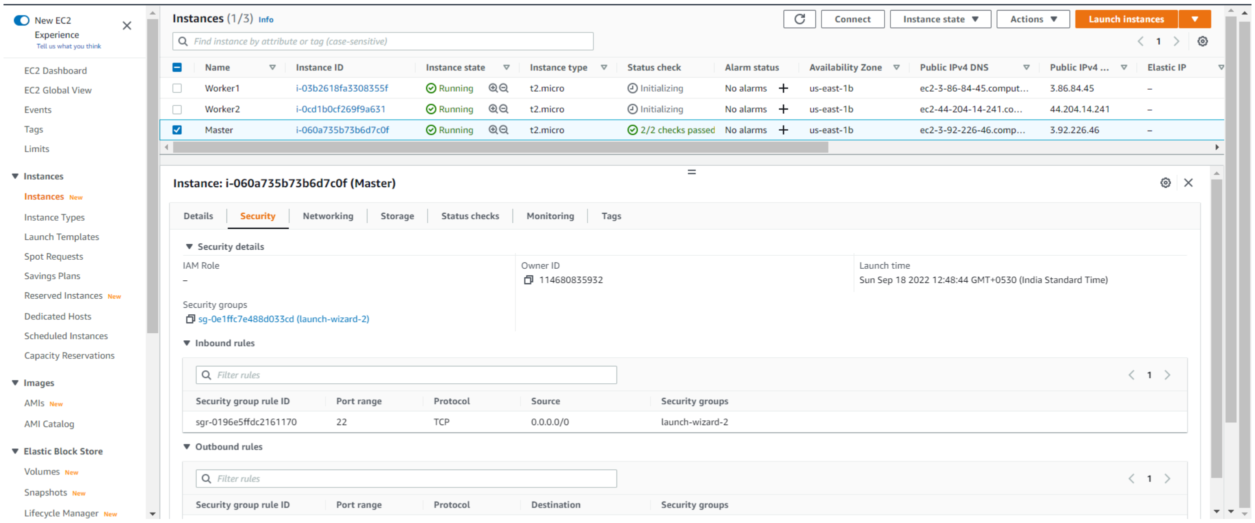
View more resources to get you started



Instances (3) Info

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
□	Worker1	i-03b2618fa3308355f	Running	t2.micro	Initializing	No alarms	us-east-1b	ec2-3-86-84-45.comput...	3.86.84.45	–
□	Worker2	i-0cd1b0cf269f9a631	Pending	t2.micro	–	No alarms	us-east-1b	ec2-44-204-14-241.co...	44.204.14.241	–
□	Master	i-060a735b73b6d7c0f	Running	t2.micro	Initializing	No alarms	us-east-1b	ec2-3-92-226-46.comp...	3.92.226.46	–

Edit security group setting to allow ssh



Instances (1/3) Info

☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
□	Worker1	i-03b2618fa3308355f	Running	t2.micro	Initializing	No alarms	us-east-1b	ec2-3-86-84-45.comput...	3.86.84.45	–
□	Worker2	i-0cd1b0cf269f9a631	Running	t2.micro	Initializing	No alarms	us-east-1b	ec2-44-204-14-241.co...	44.204.14.241	–
☑	Master	i-060a735b73b6d7c0f	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-3-92-226-46.comp...	3.92.226.46	–

Instance: i-060a735b73b6d7c0f (Master)

Security details

IAM Role	Owner ID	Launch time
–	114680835932	Sun Sep 18 2022 12:48:44 GMT+0530 (India Standard Time)

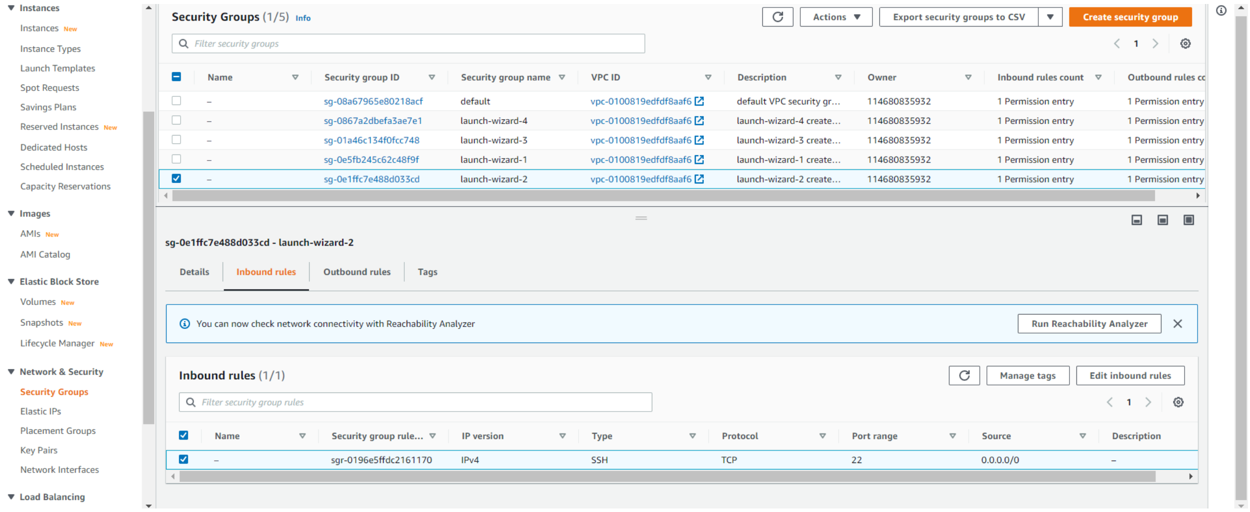
Security groups
sg-0e1ffc7e488d033cd (launch-wizard-2)

Inbound rules

Security group rule ID	Port range	Protocol	Source	Security groups
sgr-0196e5ffdc2161170	22	TCP	0.0.0.0/0	launch-wizard-2

Outbound rules

Security group rule ID	Port range	Protocol	Destination	Security groups



Security Groups (1/5) Info

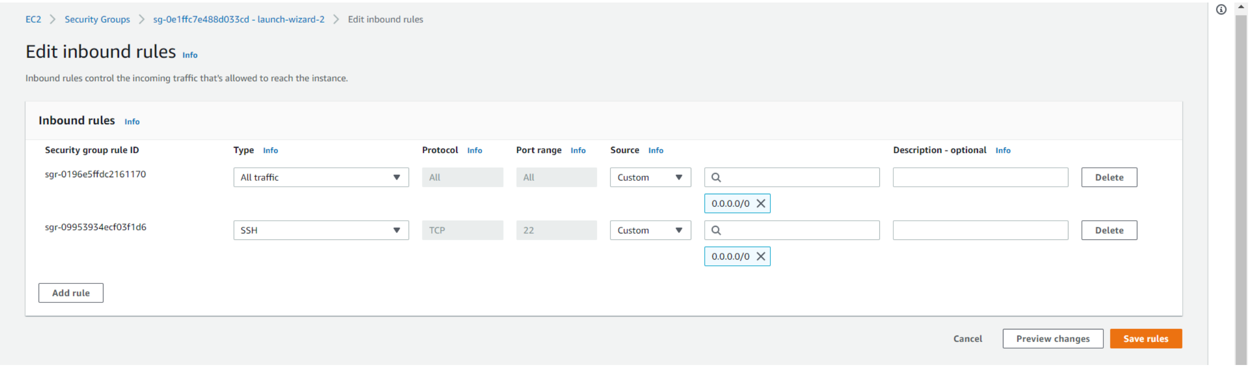
☐	Name	Security group ID	Security group name	VPC ID	Description	Owner	Inbound rules count	Outbound rules cc
□	–	sg-08a67965e80218acf	default	vpc-0100819edfdf8aaf6	default VPC security gr...	114680835932	1 Permission entry	1 Permission entry
□	–	sg-0867a2dbefa3ae7e1	launch-wizard-4	vpc-0100819edfdf8aaf6	launch-wizard-4 create...	114680835932	1 Permission entry	1 Permission entry
□	–	sg-01a46c134f0fcc748	launch-wizard-3	vpc-0100819edfdf8aaf6	launch-wizard-3 create...	114680835932	1 Permission entry	1 Permission entry
□	–	sg-0e5fb245c62c48f9f	launch-wizard-1	vpc-0100819edfdf8aaf6	launch-wizard-1 create...	114680835932	1 Permission entry	1 Permission entry
☑	–	sg-0e1ffc7e488d033cd	launch-wizard-2	vpc-0100819edfdf8aaf6	launch-wizard-2 create...	114680835932	1 Permission entry	1 Permission entry

sg-0e1ffc7e488d033cd - launch-wizard-2

You can now check network connectivity with Reachability Analyzer

Inbound rules (1/1)

☑	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
☑	–	sgr-0196e5ffdc2161170	IPv4	SSH	TCP	22	0.0.0.0/0	–



EC2 > Security Groups > sg-0e1ffc7e488d033cd - launch-wizard-2 > Edit inbound rules

Edit inbound rules Info

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

Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
sgr-0196e5ffdc2161170	All traffic	All	All	Custom	0.0.0.0/0		Delete
sgr-09953934ecf03f1d6	SSH	TCP	22	Custom	0.0.0.0/0		Delete

Steps to Install Kubernetes on Ubuntu

- Set up Docker

Step 1: Install Docker Kubernetes requires an existing Docker installation.

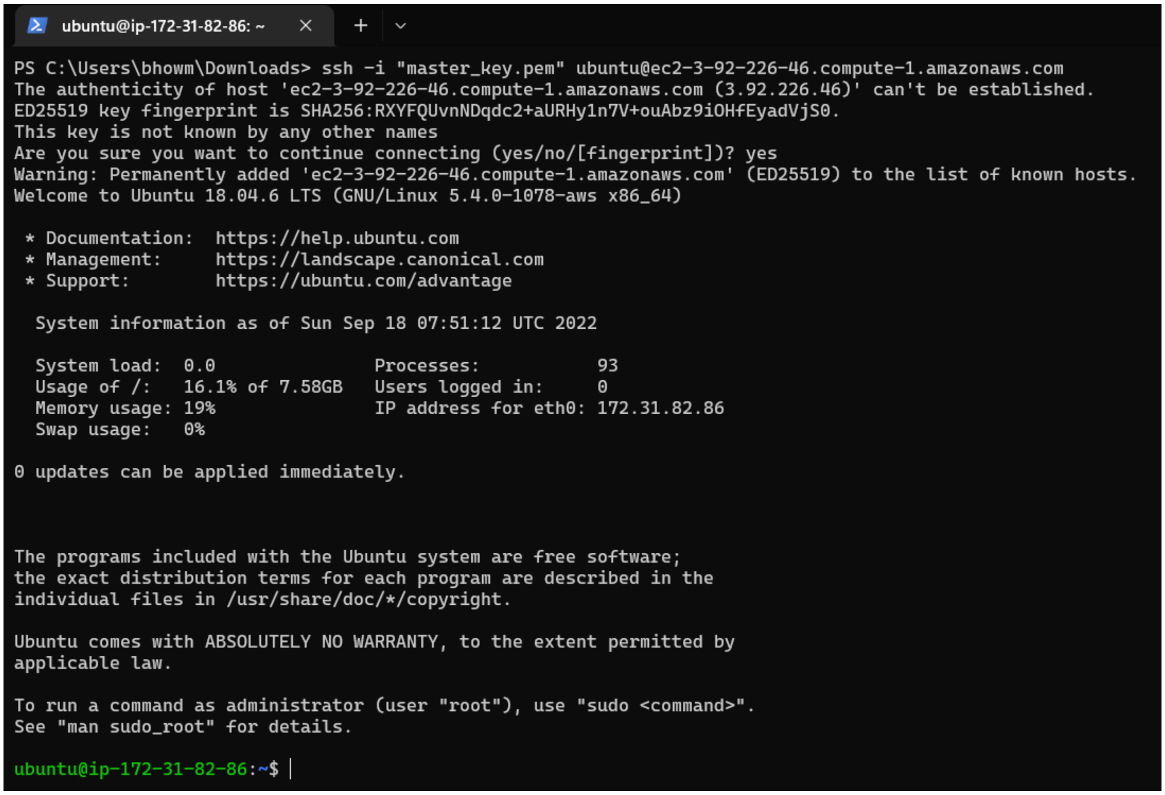
Step 2: If you do not have Kubernetes, install it by following these 1.

Update the package list with the command:

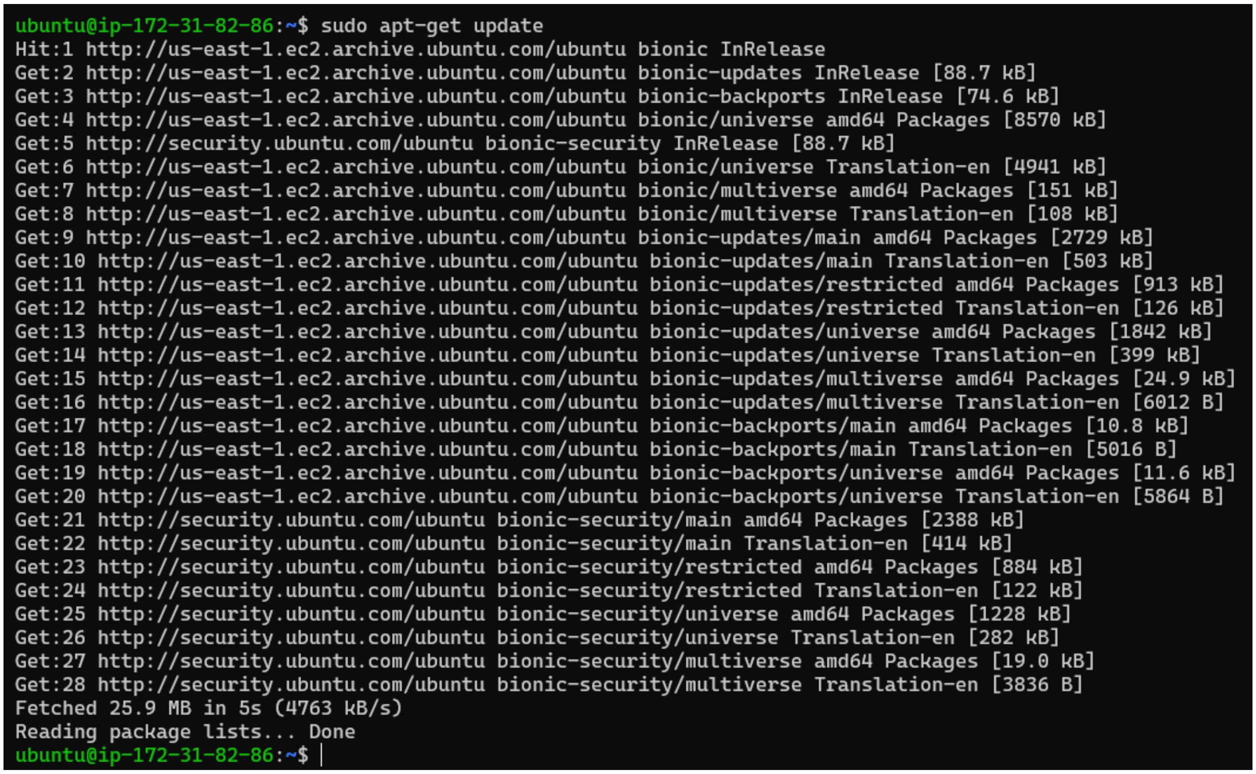
on-master&slave

$ sudo apt-get update

Master -

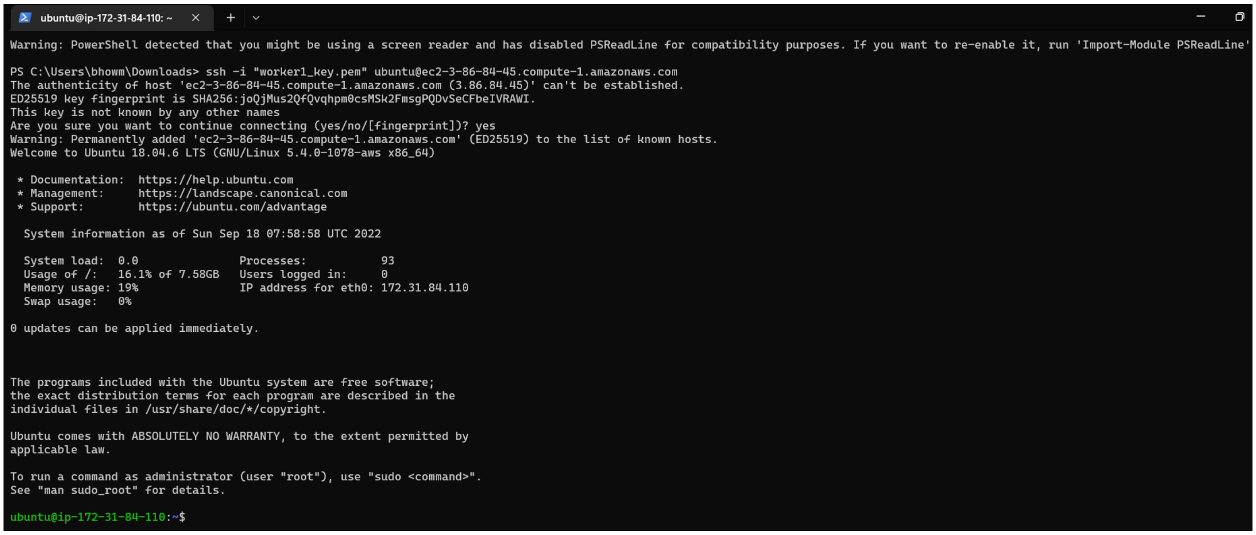


```
PS C:\Users\bhowm\Downloads> ssh -i "master_key.pem" ubuntu@ec2-3-92-226-46.compute-1.amazonaws.com
The authenticity of host 'ec2-3-92-226-46.compute-1.amazonaws.com (3.92.226.46)' can't be established.
ED25519 key fingerprint is SHA256:RXYFQUvnNDqdc2+aURHy1n7V+ouAbz9iOHfEyadVjS0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-92-226-46.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1078-aws x86_64)

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

  System information as of Sun Sep 18 07:51:12 UTC 2022

  System load:  0.0               Processes:           93
  Usage of /:   16.1% of 7.58GB   Users logged in:     0
  Memory usage: 19%               IP address for eth0: 172.31.82.86
  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-172-31-82-86:~$ 
```

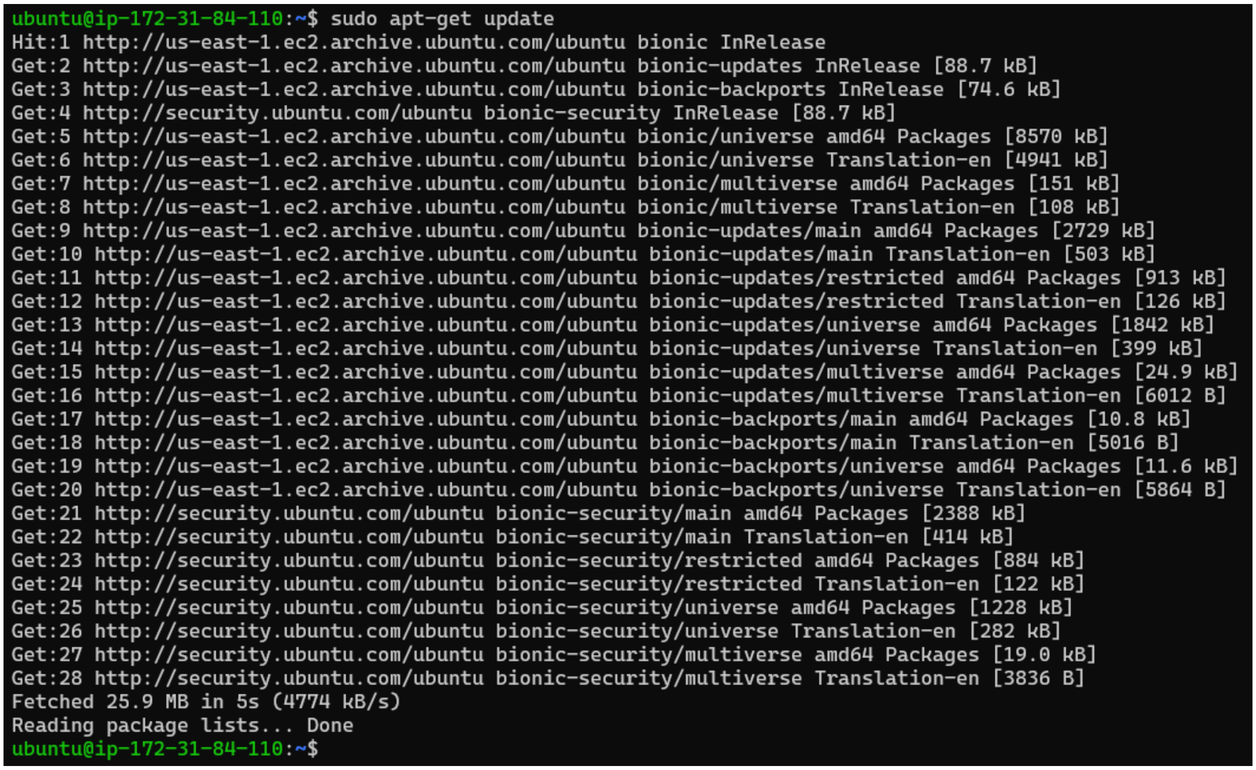


```
ubuntu@ip-172-31-82-86:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:4 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]
Fetched 25.9 MB in 5s (4763 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-82-86:~$ 
```

Worker 1

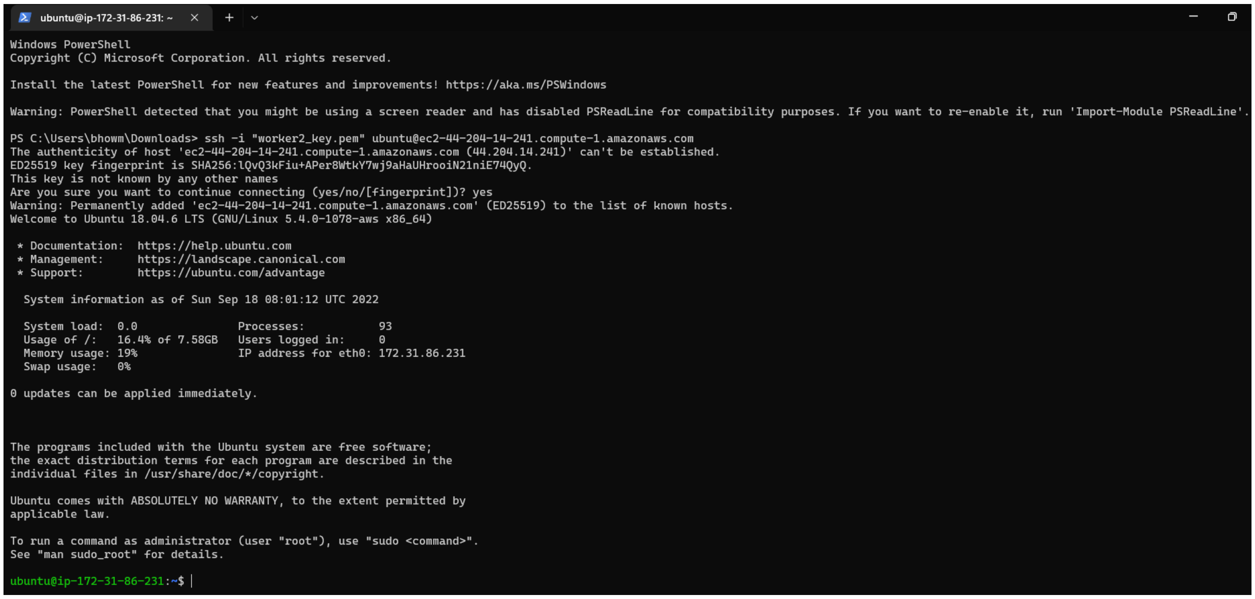


```
Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. If you want to re-enable it, run 'Import-Module PSReadLine'

PS C:\Users\bhowm\Downloads> ssh -i "worker1_key.pem" ubuntu@ec2-3-86-84-45.compute-1.amazonaws.com
The authenticity of host 'ec2-3-86-84-45.compute-1.amazonaws.com (3.86.84.45)' can't be established.
ED25519 key fingerprint is SHA256:joQjMus2QfQvqhpm0csMSk2FmsgPQDvSeCFbeIVRAWI.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-86-84-45.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1078-aws x86_64)

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

  System information as of Sun Sep 18 07:58:58 UTC 2022

  System load:  0.0               Processes:           93
  Usage of /:   16.1% of 7.58GB   Users logged in:     0
  Memory usage: 19%               IP address for eth0: 172.31.84.110
  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-172-31-84-110:~$
```

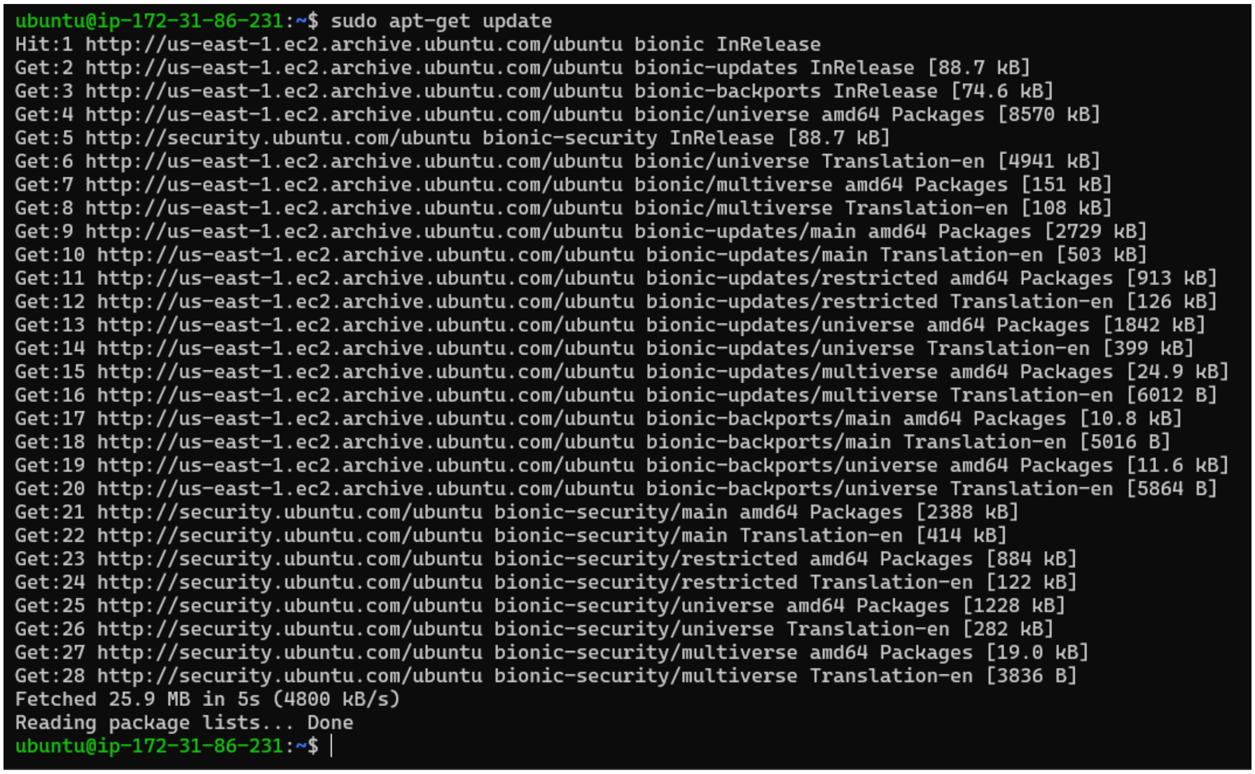


```
ubuntu@ip-172-31-84-110:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:4 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]
Fetched 25.9 MB in 5s (4774 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-84-110:~$
```

Worker 2



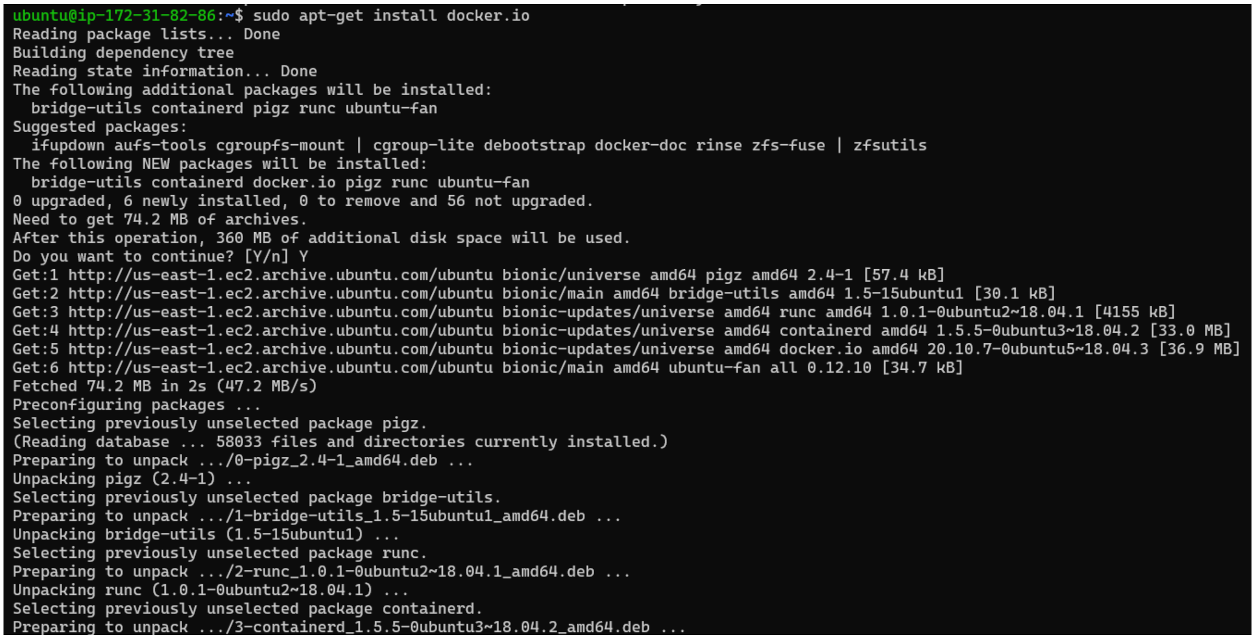
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. If you want to re-enable it, run 'Import-Module PSReadLine'.

PS C:\Users\bhowm\Downloads> ssh -i "worker2_key.pem" ubuntu@ec2-44-204-14-241.compute-1.amazonaws.com
The authenticity of host 'ec2-44-204-14-241.compute-1.amazonaws.com (44.204.14.241)' can't be established.
ED25519 key fingerprint is SHA256:lQvQ3kFiu+APer8WtkY7wj9aHaUHrooiN21niE74QyQ.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-204-14-241.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1078-aws x86_64)

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

  System information as of Sun Sep 18 08:01:12 UTC 2022

  System load:  0.0               Processes:           93
  Usage of /:   16.4% of 7.58GB   Users logged in:     0
  Memory usage: 19%               IP address for eth0: 172.31.86.231
  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-172-31-86-231:~$ |
```



```
ubuntu@ip-172-31-86-231:~$ 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]
Fetched 25.9 MB in 5s (4800 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-86-231:~$ |
```

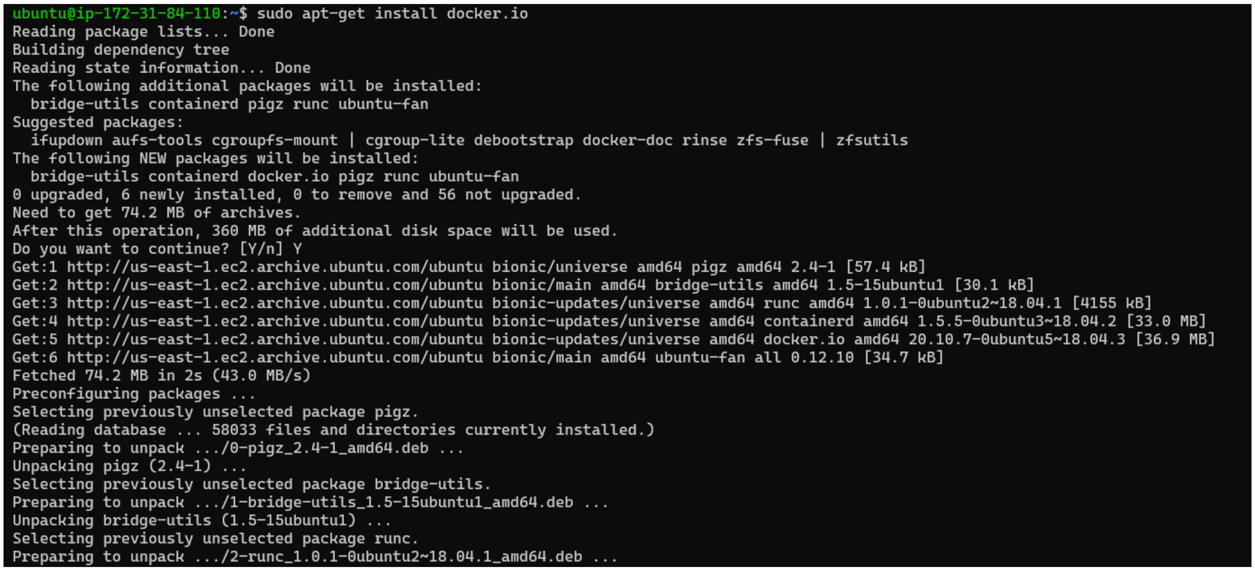
2. Next, install Docker with the command:
on-master&slave
$ sudo apt-get install docker.io

Master -



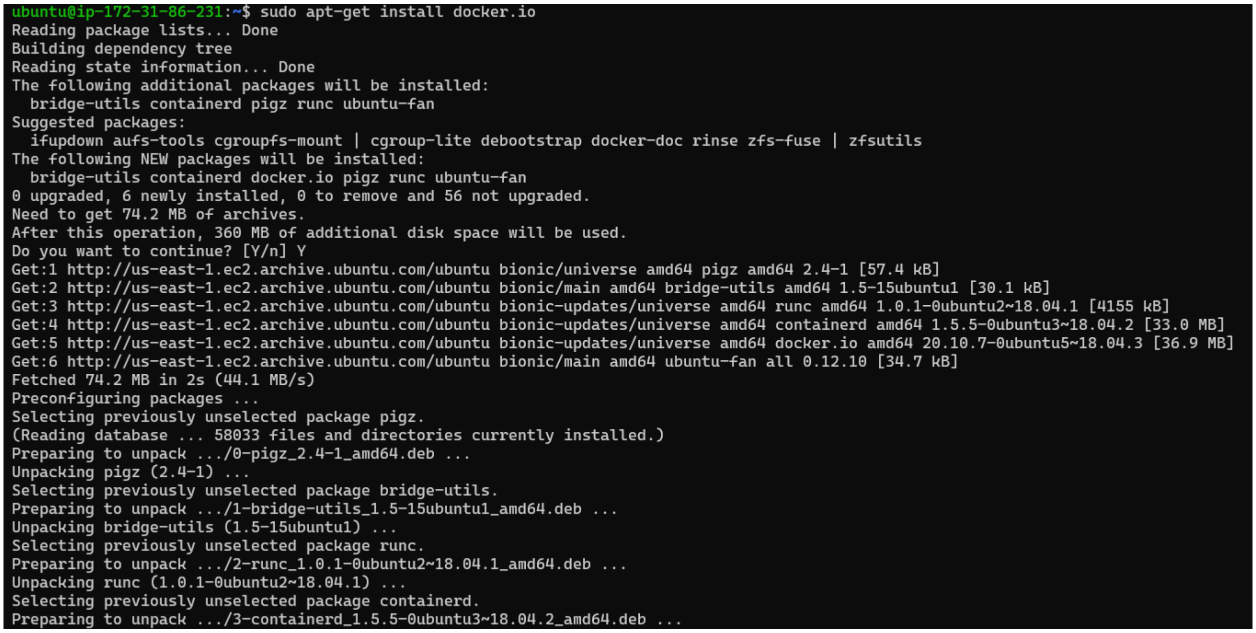
```
ubuntu@ip-172-31-82-86:~$ sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | 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 (47.2 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) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.5-15ubuntu1_amd64.deb ...
Unpacking bridge-utils (1.5-15ubuntu1) ...
Selecting previously unselected package runc.
Preparing to unpack .../2-runc_1.0.1-0ubuntu2~18.04.1_amd64.deb ...
Unpacking runc (1.0.1-0ubuntu2~18.04.1) ...
Selecting previously unselected package containerd.
Preparing to unpack .../3-containerd_1.5.5-0ubuntu3~18.04.2_amd64.deb ...
```

Worker 1



```
ubuntu@ip-172-31-84-110:~$ 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 (43.0 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) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.5-15ubuntu1_amd64.deb ...
Unpacking bridge-utils (1.5-15ubuntu1) ...
Selecting previously unselected package runc.
Preparing to unpack .../2-runc_1.0.1-0ubuntu2~18.04.1_amd64.deb ...
```

Worker 2

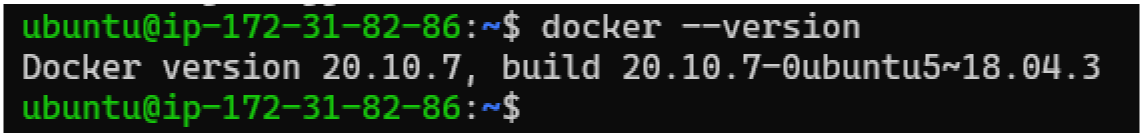


```
ubuntu@ip-172-31-86-231:~$ sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | 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 (44.1 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) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.5-15ubuntu1_amd64.deb ...
Unpacking bridge-utils (1.5-15ubuntu1) ...
Selecting previously unselected package runc.
Preparing to unpack .../2-runc_1.0.1-0ubuntu2~18.04.1_amd64.deb ...
Unpacking runc (1.0.1-0ubuntu2~18.04.1) ...
Selecting previously unselected package containerd.
Preparing to unpack .../3-containerd_1.5.5-0ubuntu3~18.04.2_amd64.deb ...
```

4. Check the installation (and version) by entering the following:
on-master&slave

$ docker --version

Master -



```
ubuntu@ip-172-31-82-86:~$ docker --version
Docker version 20.10.7, build 20.10.7-0ubuntu5~18.04.3
ubuntu@ip-172-31-82-86:~$
```

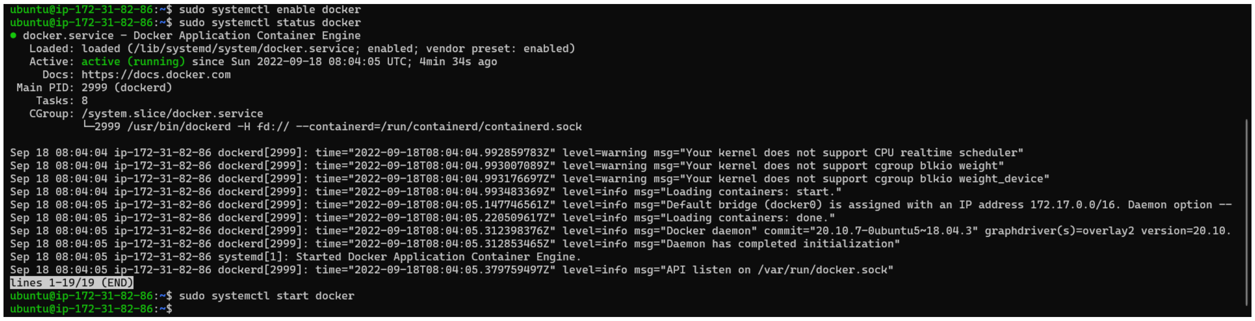
Step 3: Start and Enable Docker

1. Set Docker to launch at boot by entering the following: on-master&slave

$ sudo systemctl enable docker

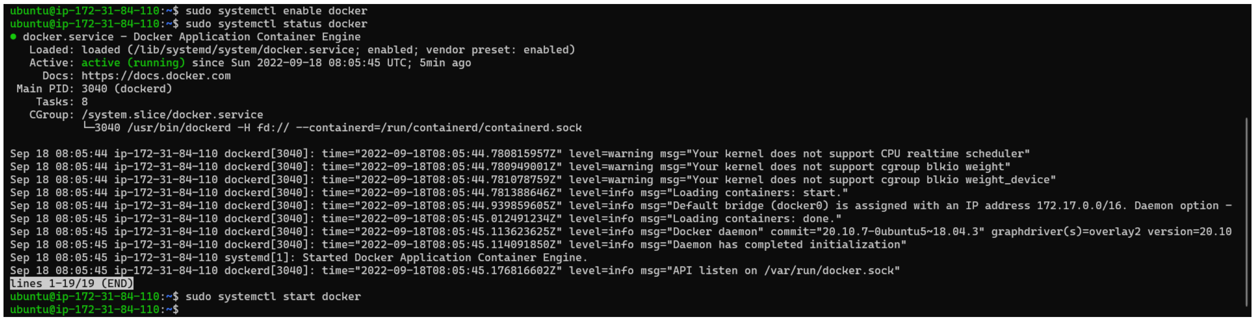
2. Verify Docker is running: on-master&slave$ sudo systemctl status docker
3. Start Docker if it’s not running: on-master&slave$ sudo systemctl start docker
4. Repeat on all the other nodes.

Master-



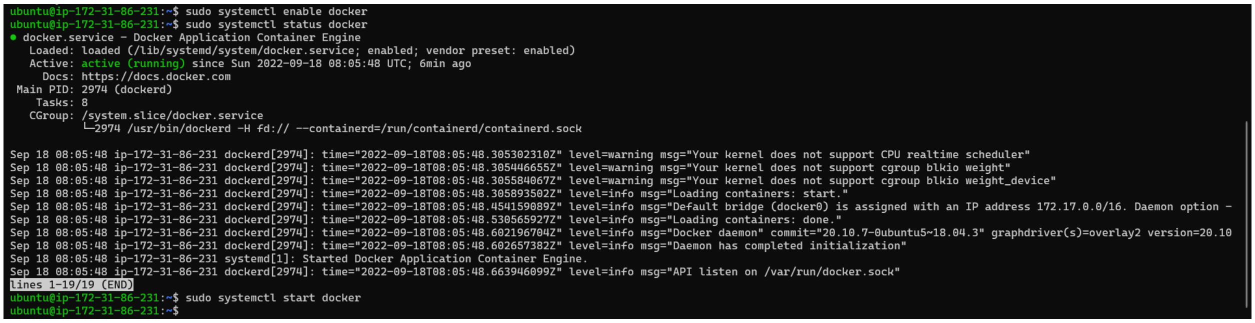
```
ubuntu@ip-172-31-82-86:~$ sudo systemctl enable docker
ubuntu@ip-172-31-82-86:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2022-09-18 08:04:05 UTC; 4min 34s ago
     Docs: https://docs.docker.com
 Main PID: 2999 (dockerd)
    Tasks: 8
   CGroup: /system.slice/docker.service
           └─2999 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Sep 18 08:04:04 ip-172-31-82-86 dockerd[2999]: time="2022-09-18T08:04:04.992859783Z" level=warning msg="Your kernel does not support CPU realtime scheduler"
Sep 18 08:04:04 ip-172-31-82-86 dockerd[2999]: time="2022-09-18T08:04:04.993007089Z" level=warning msg="Your kernel does not support cgroup blkio weight"
Sep 18 08:04:04 ip-172-31-82-86 dockerd[2999]: time="2022-09-18T08:04:04.993176697Z" level=warning msg="Your kernel does not support cgroup blkio weight_device"
Sep 18 08:04:04 ip-172-31-82-86 dockerd[2999]: time="2022-09-18T08:04:04.993483369Z" level=info msg="Loading containers: start."
Sep 18 08:04:05 ip-172-31-82-86 dockerd[2999]: time="2022-09-18T08:04:05.147746561Z" level=info msg="Default bridge (docker0) is assigned with an IP address 172.17.0.0/16. Daemon option --
Sep 18 08:04:05 ip-172-31-82-86 dockerd[2999]: time="2022-09-18T08:04:05.220509617Z" level=info msg="Loading containers: done."
Sep 18 08:04:05 ip-172-31-82-86 dockerd[2999]: time="2022-09-18T08:04:05.312398376Z" level=info msg="Docker daemon" commit="20.10.7-0ubuntu5~18.04.3" graphdriver(s)=overlay2 version=20.10.
Sep 18 08:04:05 ip-172-31-82-86 dockerd[2999]: time="2022-09-18T08:04:05.312853465Z" level=info msg="Daemon has completed initialization"
Sep 18 08:04:05 ip-172-31-82-86 systemd[1]: Started Docker Application Container Engine.
Sep 18 08:04:05 ip-172-31-82-86 dockerd[2999]: time="2022-09-18T08:04:05.379759497Z" level=info msg="API listen on /var/run/docker.sock"
lines 1-19/19 (END)
ubuntu@ip-172-31-82-86:~$ sudo systemctl start docker
ubuntu@ip-172-31-82-86:~$
```

Worker 1



```
ubuntu@ip-172-31-84-110:~$ sudo systemctl enable docker
ubuntu@ip-172-31-84-110:~$ 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 Sun 2022-09-18 08:05:45 UTC; 5min ago
     Docs: https://docs.docker.com
 Main PID: 3040 (dockerd)
    Tasks: 8
   CGroup: /system.slice/docker.service
           └─3040 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Sep 18 08:05:44 ip-172-31-84-110 dockerd[3040]: time="2022-09-18T08:05:44.780815957Z" level=warning msg="Your kernel does not support CPU realtime scheduler"
Sep 18 08:05:44 ip-172-31-84-110 dockerd[3040]: time="2022-09-18T08:05:44.780949001Z" level=warning msg="Your kernel does not support cgroup blkio weight"
Sep 18 08:05:44 ip-172-31-84-110 dockerd[3040]: time="2022-09-18T08:05:44.781078759Z" level=warning msg="Your kernel does not support cgroup blkio weight_device"
Sep 18 08:05:44 ip-172-31-84-110 dockerd[3040]: time="2022-09-18T08:05:44.781388646Z" level=info msg="Loading containers: start."
Sep 18 08:05:44 ip-172-31-84-110 dockerd[3040]: time="2022-09-18T08:05:44.939859605Z" level=info msg="Default bridge (docker0) is assigned with an IP address 172.17.0.0/16. Daemon option -
Sep 18 08:05:45 ip-172-31-84-110 dockerd[3040]: time="2022-09-18T08:05:45.012491234Z" level=info msg="Loading containers: done."
Sep 18 08:05:45 ip-172-31-84-110 dockerd[3040]: time="2022-09-18T08:05:45.113623625Z" level=info msg="Docker daemon" commit="20.10.7-0ubuntu5~18.04.3" graphdriver(s)=overlay2 version=20.10
Sep 18 08:05:45 ip-172-31-84-110 dockerd[3040]: time="2022-09-18T08:05:45.114091850Z" level=info msg="Daemon has completed initialization"
Sep 18 08:05:45 ip-172-31-84-110 systemd[1]: Started Docker Application Container Engine.
Sep 18 08:05:45 ip-172-31-84-110 dockerd[3040]: time="2022-09-18T08:05:45.176816602Z" level=info msg="API listen on /var/run/docker.sock"
lines 1-19/19 (END)
ubuntu@ip-172-31-84-110:~$ sudo systemctl start docker
ubuntu@ip-172-31-84-110:~$
```

Worker 2



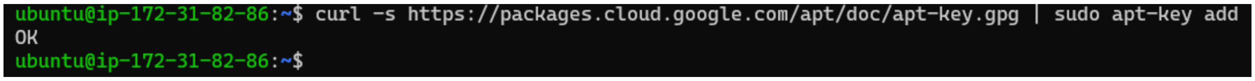
```
ubuntu@ip-172-31-86-231:~$ sudo systemctl enable docker
ubuntu@ip-172-31-86-231:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2022-09-18 08:05:48 UTC; 6min ago
     Docs: https://docs.docker.com
 Main PID: 2974 (dockerd)
    Tasks: 8
   CGroup: /system.slice/docker.service
           └─2974 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Sep 18 08:05:48 ip-172-31-86-231 dockerd[2974]: time="2022-09-18T08:05:48.305302310Z" level=warning msg="Your kernel does not support CPU realtime scheduler"
Sep 18 08:05:48 ip-172-31-86-231 dockerd[2974]: time="2022-09-18T08:05:48.305446655Z" level=warning msg="Your kernel does not support cgroup blkio weight"
Sep 18 08:05:48 ip-172-31-86-231 dockerd[2974]: time="2022-09-18T08:05:48.305584067Z" level=warning msg="Your kernel does not support cgroup blkio weight_device"
Sep 18 08:05:48 ip-172-31-86-231 dockerd[2974]: time="2022-09-18T08:05:48.305893502Z" level=info msg="Loading containers: start."
Sep 18 08:05:48 ip-172-31-86-231 dockerd[2974]: time="2022-09-18T08:05:48.454159089Z" level=info msg="Default bridge (docker0) is assigned with an IP address 172.17.0.0/16. Daemon option -
Sep 18 08:05:48 ip-172-31-86-231 dockerd[2974]: time="2022-09-18T08:05:48.530565927Z" level=info msg="Loading containers: done."
Sep 18 08:05:48 ip-172-31-86-231 dockerd[2974]: time="2022-09-18T08:05:48.602196704Z" level=info msg="Docker daemon" commit="20.10.7-0ubuntu5~18.04.3" graphdriver(s)=overlay2 version=20.10
Sep 18 08:05:48 ip-172-31-86-231 dockerd[2974]: time="2022-09-18T08:05:48.602657382Z" level=info msg="Daemon has completed initialization"
Sep 18 08:05:48 ip-172-31-86-231 systemd[1]: Started Docker Application Container Engine.
Sep 18 08:05:48 ip-172-31-86-231 dockerd[2974]: time="2022-09-18T08:05:48.663946099Z" level=info msg="API listen on /var/run/docker.sock"
lines 1-19/19 (END)
ubuntu@ip-172-31-86-231:~$ sudo systemctl start docker
ubuntu@ip-172-31-86-231:~$
```

Install Kubernetes

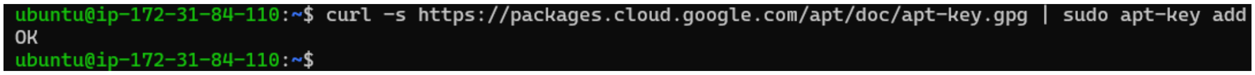
Step 4: Add Kubernetes Signing Key Since we are downloading Kubernetes from a non-standard repository, it is essential to ensure that the software is authentic. This is done by adding a signing key.

1. Enter the following to add a signing key: on-master&slave

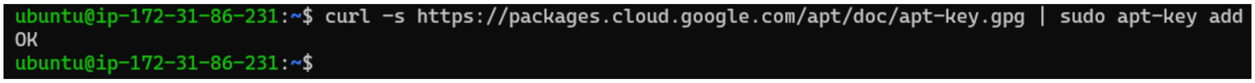
$ curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add



```
ubuntu@ip-172-31-82-86:~$ curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add
OK
ubuntu@ip-172-31-82-86:~$
```



```
ubuntu@ip-172-31-84-110:~$ curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add
OK
ubuntu@ip-172-31-84-110:~$
```



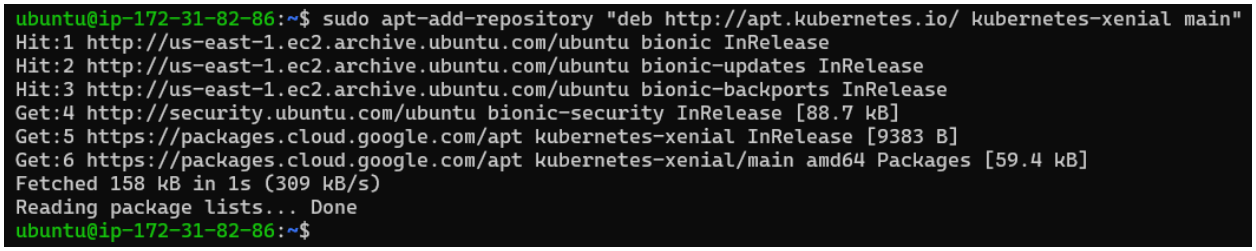
```
ubuntu@ip-172-31-86-231:~$ curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add
OK
ubuntu@ip-172-31-86-231:~$
```

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

Step 5: Add Software Repositories Kubernetes is not included in the default repositories. To add them, enter the following: on-master&slave

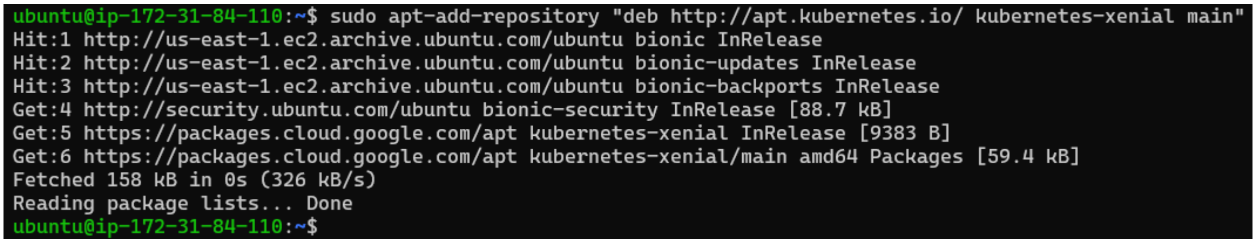
$ sudo apt-add-repository "deb http://apt.kubernetes.io/ kubernetes-xenial main"

Master



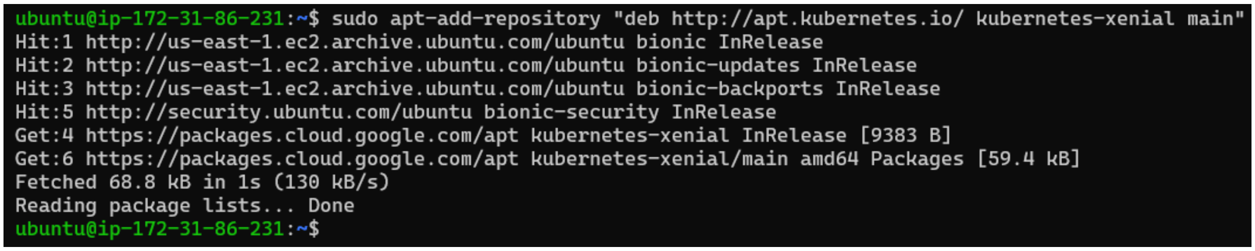
```
ubuntu@ip-172-31-82-86:~$ sudo apt-add-repository "deb http://apt.kubernetes.io/ kubernetes-xenial main"
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
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 158 kB in 1s (309 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-82-86:~$
```

Worker 1



```
ubuntu@ip-172-31-84-110:~$ 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
Get:4 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
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 158 kB in 0s (326 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-84-110:~$
```

Worker 2



```
ubuntu@ip-172-31-86-231:~$ 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 1s (130 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-86-231:~$
```

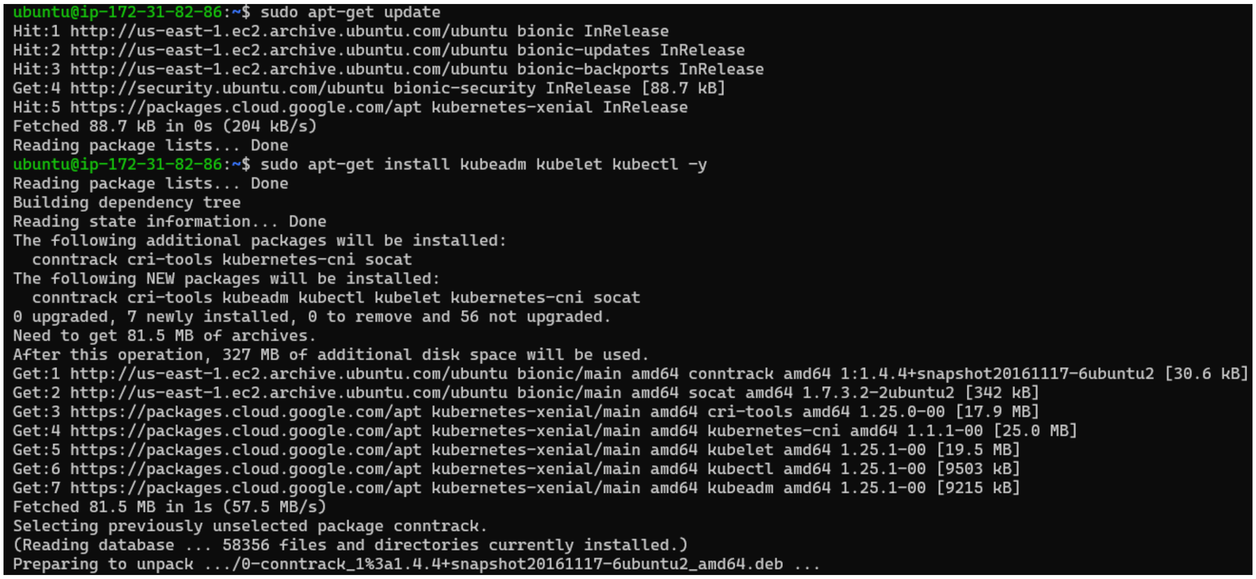
Step 6: Kubernetes Installation Tools Kubeadm (Kubernetes Admin) is a tool that helps initialize a cluster. It fast-tracks setup by using community-sourced best practices. Kubelet is the work package, which runs on every node and starts containers. The tool gives you command-line access to clusters.

1. Install Kubernetes tools with the command:

on-master&slave

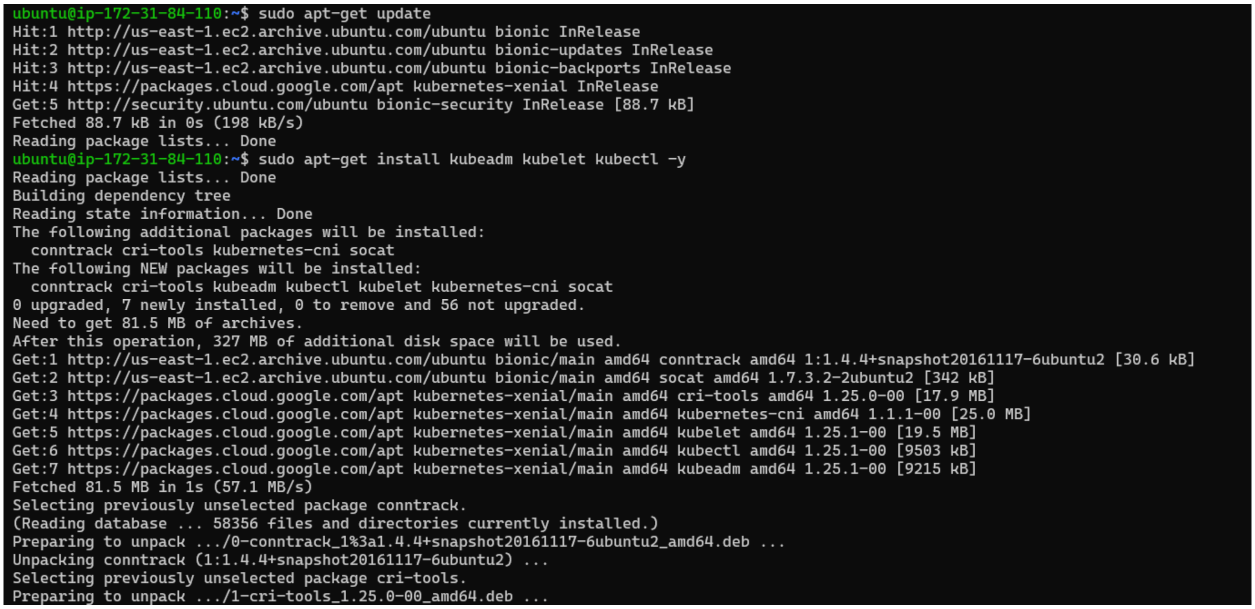
$ sudo apt-get install kubeadm kubelet kubectl -y

Master



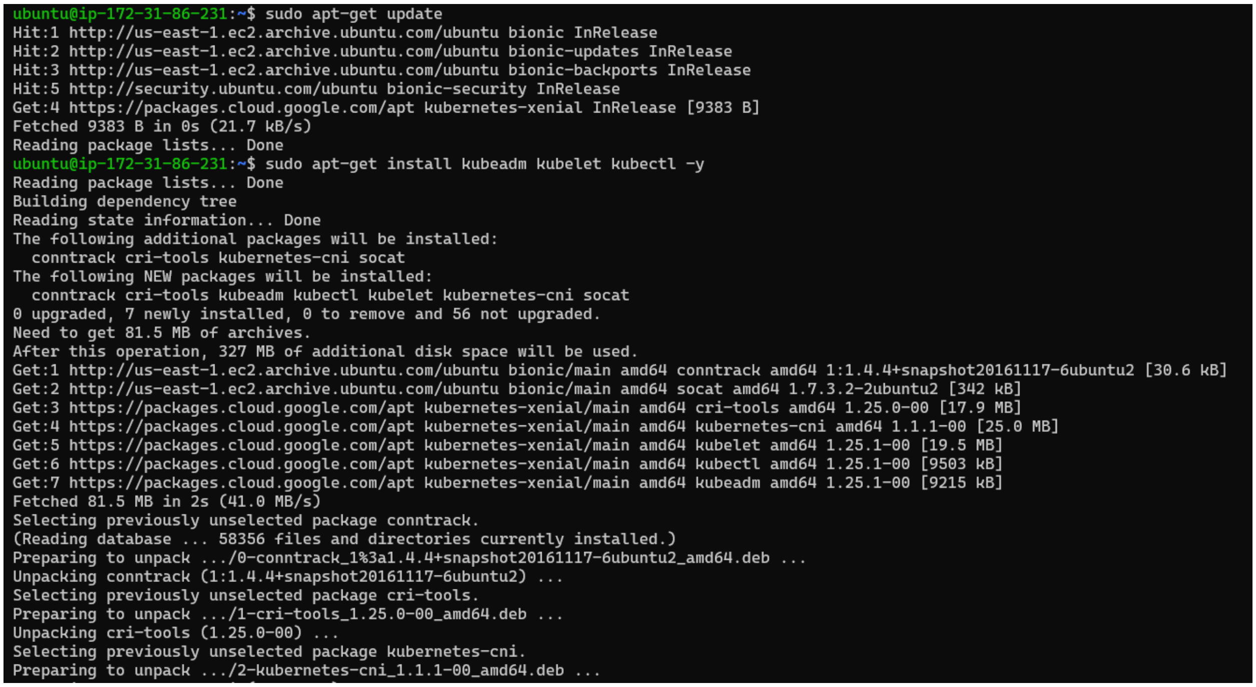
```
ubuntu@ip-172-31-82-86:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Hit:5 https://packages.cloud.google.com/apt kubernetes-xenial InRelease
Fetched 88.7 kB in 0s (204 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-82-86:~$ 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 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 1s (57.5 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 ...
```

Worker 1



```
ubuntu@ip-172-31-84-110:~$ sudo apt-get update
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 https://packages.cloud.google.com/apt kubernetes-xenial InRelease
Get:5 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Fetched 88.7 kB in 0s (198 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-84-110:~$ 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 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 1s (57.1 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 ...
```

Worker 2

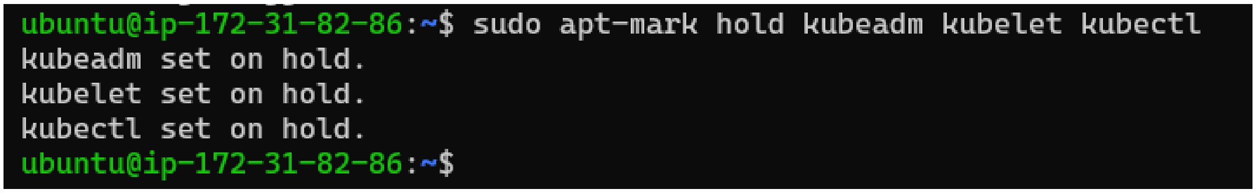


```
ubuntu@ip-172-31-86-231:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:5 http://security.ubuntu.com/ubuntu bionic-security InRelease
Get:4 https://packages.cloud.google.com/apt kubernetes-xenial InRelease [9383 B]
Fetched 9383 B in 0s (21.7 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-86-231:~$ 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 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 (41.0 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.
Preparing to unpack .../2-kubernetes-cni_1.1.1-00_amd64.deb ...
```

on-master&slave

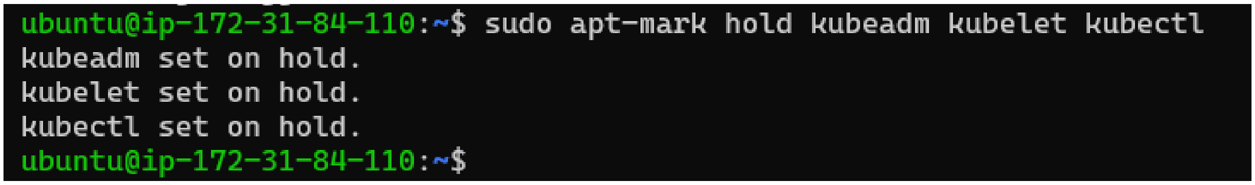
$ sudo apt-mark hold kubeadm kubelet kubectl

Master



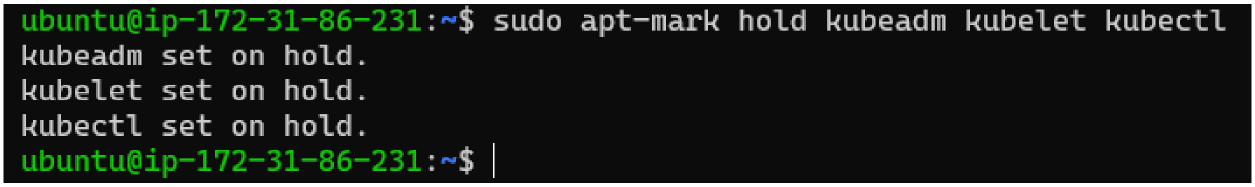
```
ubuntu@ip-172-31-82-86:~$ sudo apt-mark hold kubeadm kubelet kubectl
kubeadm set on hold.
kubelet set on hold.
kubectl set on hold.
ubuntu@ip-172-31-82-86:~$
```

Worker 1



```
ubuntu@ip-172-31-84-110:~$ sudo apt-mark hold kubeadm kubelet kubectl
kubeadm set on hold.
kubelet set on hold.
kubectl set on hold.
ubuntu@ip-172-31-84-110:~$
```

Worker 2



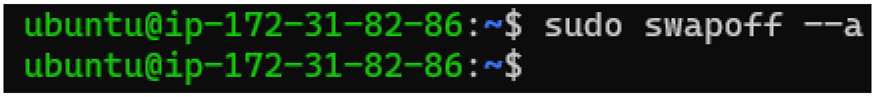
```
ubuntu@ip-172-31-86-231:~$ sudo apt-mark hold kubeadm kubelet kubectl
kubeadm set on hold.
kubelet set on hold.
kubectl set on hold.
ubuntu@ip-172-31-86-231:~$
```

Kubernetes Deployment

Step 7: Begin Kubernetes Deployment Start by disabling the swap memory on each server: on-master&slave

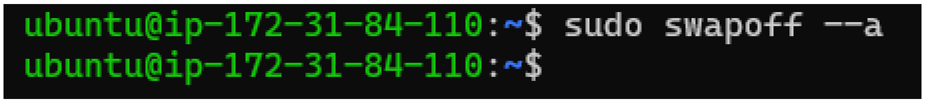
$ sudo swapoff --a

Master



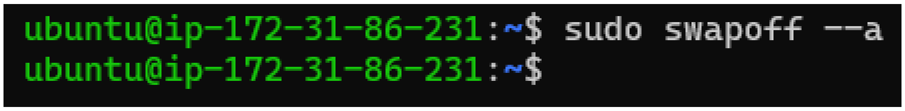
```
ubuntu@ip-172-31-82-86:~$ sudo swapoff --a
ubuntu@ip-172-31-82-86:~$
```

Worker 1



```
ubuntu@ip-172-31-84-110:~$ sudo swapoff --a
ubuntu@ip-172-31-84-110:~$
```

Worker 2

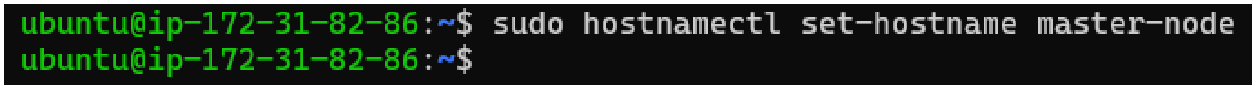


```
ubuntu@ip-172-31-86-231:~$ sudo swapoff --a
ubuntu@ip-172-31-86-231:~$
```

Step 8: 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

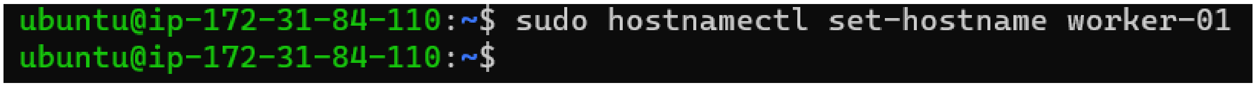


```
ubuntu@ip-172-31-82-86:~$ sudo hostnamectl set-hostname master-node
ubuntu@ip-172-31-82-86:~$
```

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

on-slave

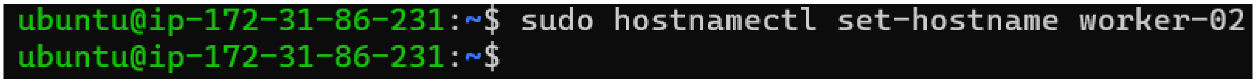
$ sudo hostnamectl set-hostname worker-01



```
ubuntu@ip-172-31-84-110:~$ sudo hostnamectl set-hostname worker-01
ubuntu@ip-172-31-84-110:~$
```

on-slave

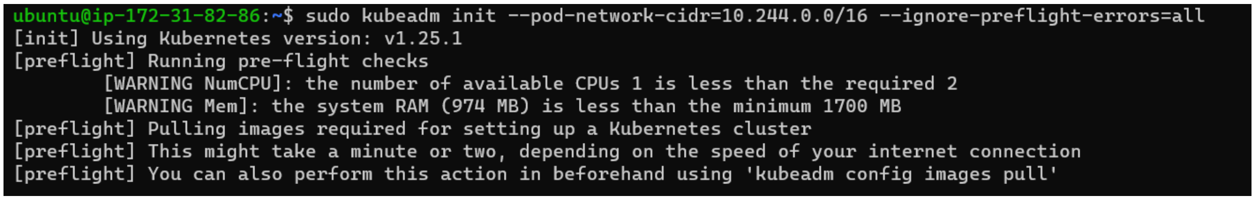
$ sudo hostnamectl set-hostname worker-02



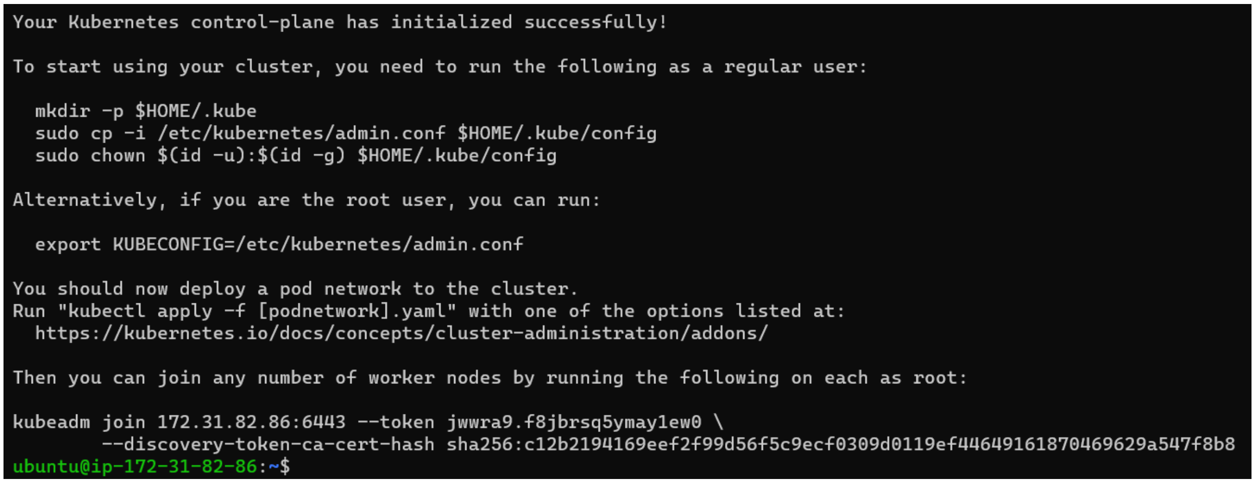
```
ubuntu@ip-172-31-86-231:~$ sudo hostnamectl set-hostname worker-02
ubuntu@ip-172-31-86-231:~$
```

Step 9: 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 –ignore-preflight-errors=all



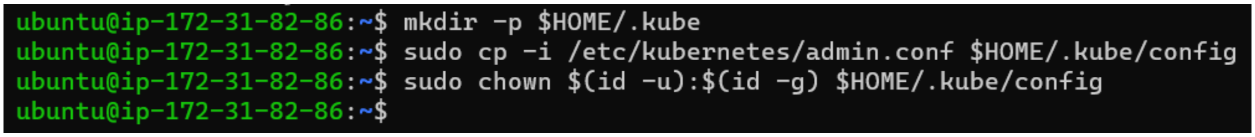
```
ubuntu@ip-172-31-82-86:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=all
[init] Using Kubernetes version: v1.25.1
[preflight] Running pre-flight checks
        [WARNING NumCPU]: the number of available CPUs 1 is less than the required 2
        [WARNING Mem]: the 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'
```



```
Your Kubernetes control-plane has initialized successfully!

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

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

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

  export KUBECONFIG=/etc/kubernetes/admin.conf

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

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

kubeadm join 172.31.82.86:6443 --token jwwra9.f8jbrsq5ymay1ew0 \
        --discovery-token-ca-cert-hash sha256:c12b2194169eef2f99d56f5c9ecf0309d0119ef446491618704696 29a547f8b8
ubuntu@ip-172-31-82-86:~$
```

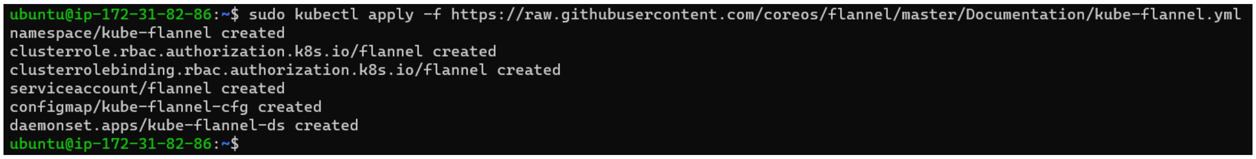
Once this command finishes, it will display a kubeadm join message at the end.
Make a note of the whole entry. This will be used to join the worker nodes to the cluster.
Next, enter the following to create a directory for the cluster:

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**



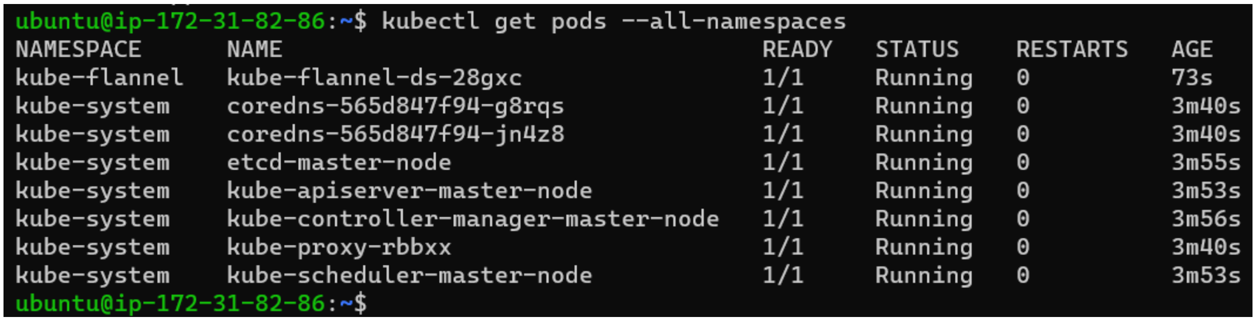
```
ubuntu@ip-172-31-82-86:~$ mkdir -p $HOME/.kube
ubuntu@ip-172-31-82-86:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
ubuntu@ip-172-31-82-86:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-82-86:~$
```

Step 10: 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-172-31-82-86:~$ 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-172-31-82-86:~$
```

Allow the process to complete. Verify that everything is running and communicating:
kubernetes-master:~$ **kubectl get pods --all-namespaces**

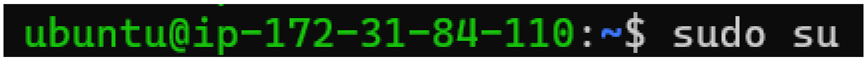


```
ubuntu@ip-172-31-82-86:~$ kubectl get pods --all-namespaces
NAMESPACE      NAME                                  READY   STATUS    RESTARTS   AGE
kube-flannel   kube-flannel-ds-28gxc                 1/1     Running   0          73s
kube-system    coredns-565d847f94-g8rqs              1/1     Running   0          3m40s
kube-system    coredns-565d847f94-jn4z8              1/1     Running   0          3m40s
kube-system    etcd-master-node                      1/1     Running   0          3m55s
kube-system    kube-apiserver-master-node            1/1     Running   0          3m53s
kube-system    kube-controller-manager-master-node   1/1     Running   0          3m56s
kube-system    kube-proxy-rbbxx                      1/1     Running   0          3m40s
kube-system    kube-scheduler-master-node            1/1     Running   0          3m53s
ubuntu@ip-172-31-82-86:~$
```

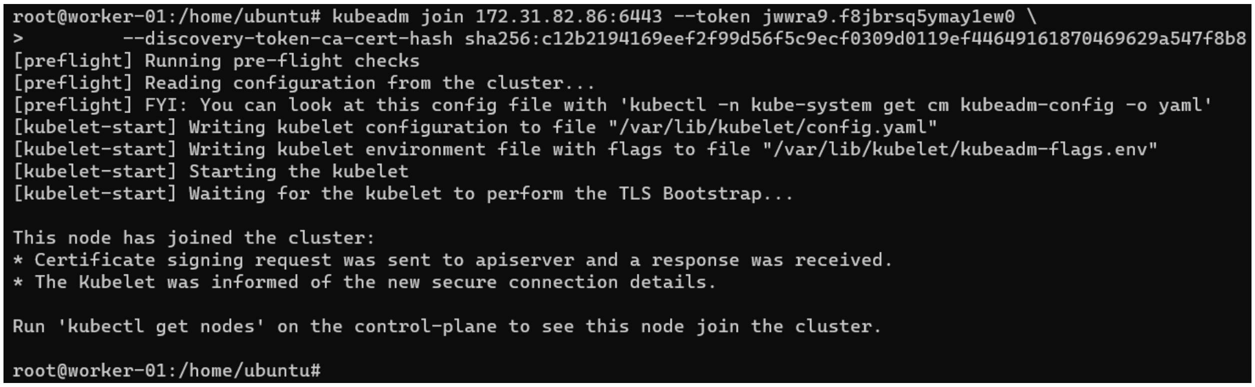
Step 11: Join Worker Node to Cluster As indicated in Step 8, you can enter the kubeadm join command on each worker node to connect it to the cluster. Switch to the worker01 system and enter the command you noted from

Step 7: (Example):*kubernetes-slave:*~$ **kubeadm join --discovery-token abcdef.1234567890abcdef**
--discovery-token-ca-cert-hash sha256:1234..cdef 1.2.3.4:6443
–ignore-preflight-errors=all

Worker 1

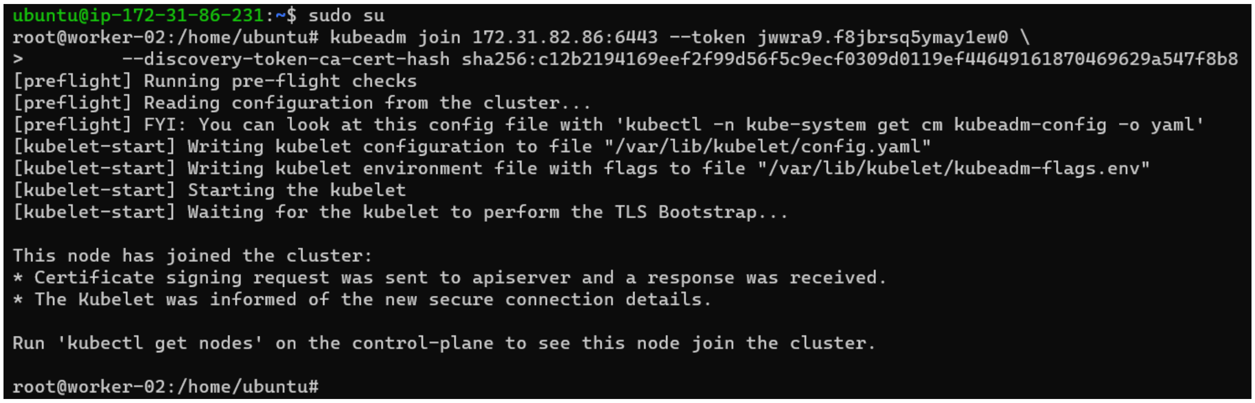


```
ubuntu@ip-172-31-84-110:~$ sudo su
```



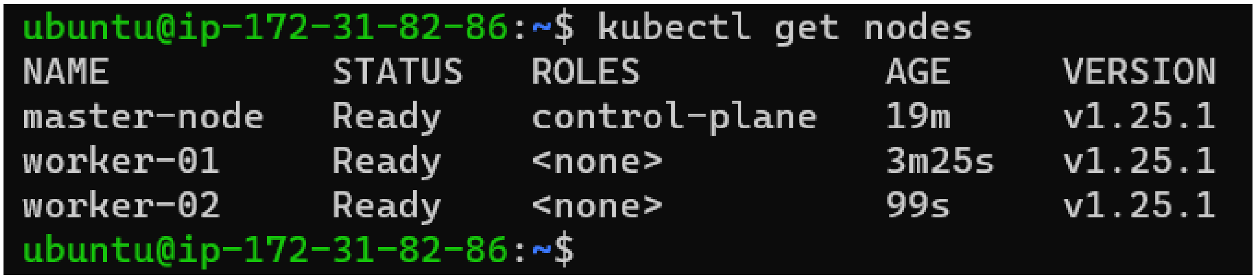
```
root@worker-01:/home/ubuntu# kubeadm join 172.31.82.86:6443 --token jwwra9.f8jbrsq5ymay1ew0 \
>         --discovery-token-ca-cert-hash sha256:c12b2194169eef2f99d56f5c9ecf0309d0119ef446491618704696 29a547f8b8
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[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-01:/home/ubuntu#
```

Worker 2



```
ubuntu@ip-172-31-86-231:~$ sudo su
root@worker-02:/home/ubuntu# kubeadm join 172.31.82.86:6443 --token jwwra9.f8jbrsq5ymay1ew0 \
>         --discovery-token-ca-cert-hash sha256:c12b2194169eef2f99d56f5c9ecf0309d0119ef44649161870469629a547f8b8
[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-02:/home/ubuntu#
```

Switch to the master server, and enter: kubernetes-master:~**$ kubectl get nodes**



```
ubuntu@ip-172-31-82-86:~$ kubectl get nodes
NAME          STATUS   ROLES           AGE     VERSION
master-node   Ready    control-plane   19m     v1.25.1
worker-01     Ready    <none>          3m25s   v1.25.1
worker-02     Ready    <none>          99s     v1.25.1
ubuntu@ip-172-31-82-86:~$
```

Conclusion:-

In this experiment, we learned how to install kubernetes, create a kubernetes cluster in AWS EC2 instances and get them to run and connect to eachother then delete it at the end.