


Procedure -

Setting up Kubernetes Cluster

Create 3 EC2 Ubuntu Instances on AWS. Name one as Master, the other two as worker1 and worker2.

 **Ubuntu Server 20.04 LTS (HVM), SSD Volume Type** - ami-0c1a7f89451184c8b (64-bit x86) / ami-0d18acc6e813fd2e0 (64-bit Arm)
Ubuntu Server 20.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).
Free tier eligible Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Instances (5) [Info](#) Refresh Connect Instance state ▼ Action

<input type="checkbox"/>	Name ▼	Instance ID	Instance state ▲	Instance type ▼	Status check	Alarm status
<input type="checkbox"/>	master	i-03687b8cd6b8b9dd8	Running	t2.micro	Initializing	No alarms +
<input type="checkbox"/>	worker1	i-0a10ddb7ca339e2	Running	t2.micro	Initializing	No alarms +
<input type="checkbox"/>	worker2	i-0fe27ef6f87be2c00	Running	t2.micro	Initializing	No alarms +

Edit the Security Group Inbound Rules to allow SSH

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-020053614c1e4f482	All traffic ▼	All	All	Custom ▼ <input type="text" value="sg-d9a40da6"/>	<input type="text"/> Delete
-	SSH ▼	TCP	22	Anywh... ▼ <input type="text" value="0.0.0.0/0"/>	<input type="text"/> Delete

Add rule

Cancel Preview changes Save rules

Establish connection with all three machines using SSH, using the following command:

```
ssh -i <keyname>.pem ubuntu@<public_ip_address>
```

```

ubuntu@ip-172-31-10-100: ~
Microsoft Windows [Version 10.0.19043.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\vkris>cd Downloads

C:\Users\vkris\Downloads>ssh -i keypair111.pem ubuntu@13.233.119.158
The authenticity of host '13.233.119.158 (13.233.119.158)' can't be established.
ECDSA key fingerprint is SHA256:HsaFFde8w3o7UtvYtU1SQRsJX0/oNW7Yj113vFNAdI.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.233.119.158' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-1045-aws x86_64)

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

System information as of Sun Oct  3 14:50:48 UTC 2021

System load:  0.0          Processes:            100
Usage of /:   16.4% of 7.69GB   Users logged in:    0
Memory usage: 22%          IPv4 address for eth0: 172.31.10.100
Swap usage:   0%

```

```

ubuntu@ip-172-31-12-218: ~
Microsoft Windows [Version 10.0.19043.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\vkris>cd Downloads

C:\Users\vkris\Downloads>ssh -i keypair111.pem ubuntu@3.109.155.147
The authenticity of host '3.109.155.147 (3.109.155.147)' can't be established.
ECDSA key fingerprint is SHA256:ufxZirY0JTnemOTkAag70IrvGKu2yRrpsm9ttFYHyBY.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.109.155.147' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-1045-aws x86_64)

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

System information as of Sun Oct  3 14:51:32 UTC 2021

System load:  0.56          Processes:            100
Usage of /:   16.4% of 7.69GB   Users logged in:    0
Memory usage: 22%          IPv4 address for eth0: 172.31.12.218
Swap usage:   0%

```

```

ubuntu@ip-172-31-4-243: ~
Usage of /:   16.4% of 7.69GB   Users logged in:    0
Memory usage: 22%          IPv4 address for eth0: 172.31.4.243
Swap usage:   0%

```

1 update can be applied immediately.
To see these additional updates run: `apt list --upgradable`

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

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

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

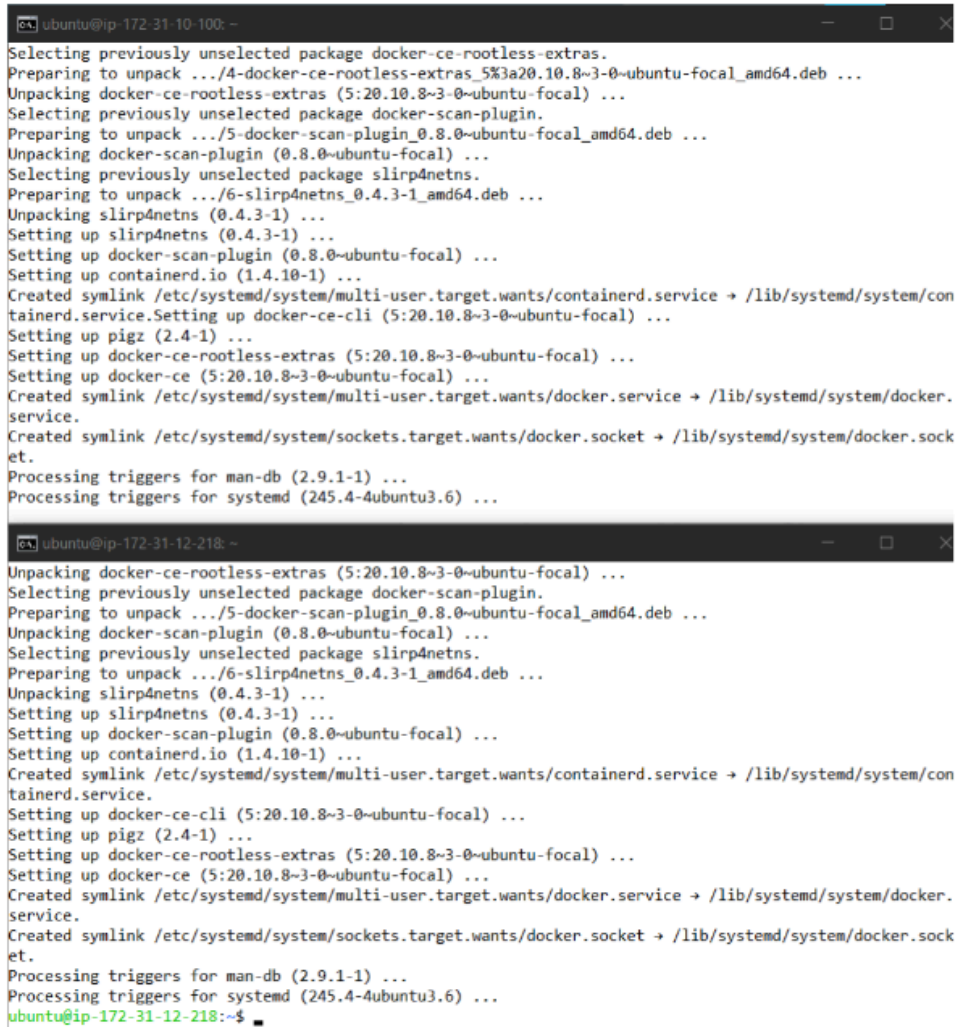
To run a command as administrator (user "root"), use "`sudo <command>`".
See "`man sudo_root`" for details.

```
ubuntu@ip-172-31-4-243:~$
```

Perform all these steps at the same time on all 3 machines, unless specified otherwise.

Install Docker:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -  
sudo add-apt-repository "deb [arch=amd64]  
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"  
sudo apt-get update  
sudo apt-get install -y docker-ce
```



```
ubuntu@ip-172-31-10-100: ~  
Selecting previously unselected package docker-ce-rootless-extras.  
Preparing to unpack .../4-docker-ce-rootless-extras_5%3a20.10.8~3-0~ubuntu-focal_amd64.deb ...  
Unpacking docker-ce-rootless-extras (5:20.10.8~3-0~ubuntu-focal) ...  
Selecting previously unselected package docker-scan-plugin.  
Preparing to unpack .../5-docker-scan-plugin_0.8.0~ubuntu-focal_amd64.deb ...  
Unpacking docker-scan-plugin (0.8.0~ubuntu-focal) ...  
Selecting previously unselected package slirp4netns.  
Preparing to unpack .../6-slirp4netns_0.4.3-1_amd64.deb ...  
Unpacking slirp4netns (0.4.3-1) ...  
Setting up slirp4netns (0.4.3-1) ...  
Setting up docker-scan-plugin (0.8.0~ubuntu-focal) ...  
Setting up containerd.io (1.4.10-1) ...  
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/con  
tainerd.service.  
Setting up docker-ce-cli (5:20.10.8~3-0~ubuntu-focal) ...  
Setting up pigz (2.4-1) ...  
Setting up docker-ce-rootless-extras (5:20.10.8~3-0~ubuntu-focal) ...  
Setting up docker-ce (5:20.10.8~3-0~ubuntu-focal) ...  
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.sock  
et.  
Processing triggers for man-db (2.9.1-1) ...  
Processing triggers for systemd (245.4-4ubuntu3.6) ...  
  
ubuntu@ip-172-31-12-218: ~  
Unpacking docker-ce-rootless-extras (5:20.10.8~3-0~ubuntu-focal) ...  
Selecting previously unselected package docker-scan-plugin.  
Preparing to unpack .../5-docker-scan-plugin_0.8.0~ubuntu-focal_amd64.deb ...  
Unpacking docker-scan-plugin (0.8.0~ubuntu-focal) ...  
Selecting previously unselected package slirp4netns.  
Preparing to unpack .../6-slirp4netns_0.4.3-1_amd64.deb ...  
Unpacking slirp4netns (0.4.3-1) ...  
Setting up slirp4netns (0.4.3-1) ...  
Setting up docker-scan-plugin (0.8.0~ubuntu-focal) ...  
Setting up containerd.io (1.4.10-1) ...  
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/con  
tainerd.service.  
Setting up docker-ce-cli (5:20.10.8~3-0~ubuntu-focal) ...  
Setting up pigz (2.4-1) ...  
Setting up docker-ce-rootless-extras (5:20.10.8~3-0~ubuntu-focal) ...  
Setting up docker-ce (5:20.10.8~3-0~ubuntu-focal) ...  
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.sock  
et.  
Processing triggers for man-db (2.9.1-1) ...  
Processing triggers for systemd (245.4-4ubuntu3.6) ...  
ubuntu@ip-172-31-12-218:~$
```

```

ubuntu@ip-172-31-4-243: ~
Unpacking docker-ce-rootless-extras (5:20.10.8~3-0~ubuntu-focal) ...
Selecting previously unselected package docker-scan-plugin.
Preparing to unpack .../5-docker-scan-plugin_0.8.0~ubuntu-focal_amd64.deb ...
Unpacking docker-scan-plugin (0.8.0~ubuntu-focal) ...
Selecting previously unselected package slirp4netns.
Preparing to unpack .../6-slirp4netns_0.4.3-1_amd64.deb ...
Unpacking slirp4netns (0.4.3-1) ...
Setting up slirp4netns (0.4.3-1) ...
Setting up docker-scan-plugin (0.8.0~ubuntu-focal) ...
Setting up containerd.io (1.4.10-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/co
ntainerd.service.
Setting up docker-ce-cli (5:20.10.8~3-0~ubuntu-focal) ...
Setting up pigz (2.4-1) ...
Setting up docker-ce-rootless-extras (5:20.10.8~3-0~ubuntu-focal) ...
Setting up docker-ce (5:20.10.8~3-0~ubuntu-focal) ...
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.soc
ket.
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for systemd (245.4-4ubuntu3.6) ...
ubuntu@ip-172-31-4-243:~$

```

Then, configure cgroup in a daemon.json file.

```

cd /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json
{
    "exec-opts": ["native.cgroupdriver=systemd"],
    "log-driver": "json-file",
    "log-opts": {
        "max-size": "100m"
    },
    "storage-driver": "overlay2"
}
EOF
sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker

```

Install Kubernetes on all 3 machines

```

curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key
add -
cat << EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb https://apt.kubernetes.io/ kubernetes-xenial main
EOF
sudo apt-get update

```

sudo apt-get install -y kubelet kubeadm kubectl

```
ubuntu@ip-172-31-10-100: /etc/docker
Preparing to unpack .../4-socat_1.7.3.3-2_amd64.deb ...
Unpacking socat (1.7.3.3-2) ...
Selecting previously unselected package kubelet.
Preparing to unpack .../5-kubelet_1.22.2-00_amd64.deb ...
Unpacking kubelet (1.22.2-00) ...
Selecting previously unselected package kubectl.
Preparing to unpack .../6-kubectl_1.22.2-00_amd64.deb ...
Unpacking kubectl (1.22.2-00) ...
Selecting previously unselected package kubeadm.
Preparing to unpack .../7-kubeadm_1.22.2-00_amd64.deb ...
Unpacking kubeadm (1.22.2-00) ...
Setting up conntrack (1:1.4.5-2) ...
Setting up kubectl (1.22.2-00) ...
Setting up ebtables (2.0.11-3build1) ...
Setting up socat (1.7.3.3-2) ...
Setting up cri-tools (1.13.0-01) ...
Setting up kubernetes-cni (0.8.7-00) ...
Setting up kubelet (1.22.2-00) ...
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /lib/systemd/system/kubelet.service.
Setting up kubeadm (1.22.2-00) ...
Processing triggers for man-db (2.9.1-1) ...
ubuntu@ip-172-31-10-100: /etc/docker$
```

```
ubuntu@ip-172-31-12-218: /etc/docker
Preparing to unpack .../4-socat_1.7.3.3-2_amd64.deb ...
Unpacking socat (1.7.3.3-2) ...
Selecting previously unselected package kubelet.
Preparing to unpack .../5-kubelet_1.22.2-00_amd64.deb ...
Unpacking kubelet (1.22.2-00) ...
Selecting previously unselected package kubectl.
Preparing to unpack .../6-kubectl_1.22.2-00_amd64.deb ...
Unpacking kubectl (1.22.2-00) ...
Selecting previously unselected package kubeadm.
Preparing to unpack .../7-kubeadm_1.22.2-00_amd64.deb ...
Unpacking kubeadm (1.22.2-00) ...
Setting up conntrack (1:1.4.5-2) ...
Setting up kubectl (1.22.2-00) ...
Setting up ebtables (2.0.11-3build1) ...
Setting up socat (1.7.3.3-2) ...
Setting up cri-tools (1.13.0-01) ...
Setting up kubernetes-cni (0.8.7-00) ...
Setting up kubelet (1.22.2-00) ...
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /lib/systemd/system/kubelet.service.
Setting up kubeadm (1.22.2-00) ...
Processing triggers for man-db (2.9.1-1) ...
ubuntu@ip-172-31-12-218: /etc/docker$
```

```
ubuntu@ip-172-31-4-243: /etc/docker
MB[Get:7 https://packages.cloud.google.com/apt/kubernetes-xenial/main amd64 kubectl amd64 1.22.2-00 [90
38 kB]Get:8 https://packages.cloud.google.com/apt/kubernetes-xenial/main amd64 kubeadm amd64 1.22.2-00
[8718 kB]Fetched 73.8 MB in 4s (16.5 MB/s)
Selecting previously unselected package conntrack.
(Reading database ... 60400 files and directories currently installed.)
Preparing to unpack .../0-conntrack_1:1.4.5-2_amd64.deb ...
Unpacking conntrack (1:1.4.5-2) ...
Selecting previously unselected package cri-tools.
Preparing to unpack .../1-cri-tools_1.13.0-01_amd64.deb ...
Unpacking cri-tools (1.13.0-01) ...
Selecting previously unselected package ebtables.
Preparing to unpack .../2-ebtables_2.0.11-3build1_amd64.deb ...
Unpacking ebtables (2.0.11-3build1) ...
Selecting previously unselected package kubernetes-cni.
Preparing to unpack .../3-kubernetes-cni_0.8.7-00_amd64.deb ...
Unpacking kubernetes-cni (0.8.7-00) ...
Selecting previously unselected package socat.
Preparing to unpack .../4-socat_1.7.3.3-2_amd64.deb ...
Unpacking socat (1.7.3.3-2) ...
Selecting previously unselected package kubelet.
Preparing to unpack .../5-kubelet_1.22.2-00_amd64.deb ...
Unpacking kubelet (1.22.2-00) ...
Selecting previously unselected package kubectl.
Preparing to unpack .../6-kubectl_1.22.2-00_amd64.deb ...
Unpacking kubectl (1.22.2-00) ...
Selecting previously unselected package kubeadm.
Preparing to unpack .../7-kubeadm_1.22.2-00_amd64.deb ...
Unpacking kubeadm (1.22.2-00) ...
Setting up conntrack (1:1.4.5-2) ...
Setting up kubectl (1.22.2-00) ...
Setting up ebtables (2.0.11-3build1) ...
Setting up socat (1.7.3.3-2) ...
Setting up cri-tools (1.13.0-01) ...
Setting up kubernetes-cni (0.8.7-00) ...
Setting up kubelet (1.22.2-00) ...
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /lib/systemd/system/kubelet.service.
Setting up kubeadm (1.22.2-00) ...
Processing triggers for man-db (2.9.1-1) ...
ubuntu@ip-172-31-4-243: /etc/docker$
```


After installing Kubernetes, we need to configure internet options to allow bridging.

```
sudo swapoff -a
```

```
echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl
```

```
sudo sysctl -p
```

```
ubuntu@ip-172-31-10-100: ~  
Selecting previously unselected package kubeadm.  
Preparing to unpack .../7-kubeadm_1.22.2-00_amd64.deb ...  
Unpacking kubeadm (1.22.2-00) ...  
Setting up conntrack (1:1.4.5-2) ...  
Setting up kubect1 (1.22.2-00) ...  
Setting up ebtables (2.0.11-3build1) ...  
Setting up socat (1.7.3.3-2) ...  
Setting up cri-tools (1.13.0-01) ...  
Setting up kubernetes-cni (0.8.7-00) ...  
Setting up kubelet (1.22.2-00) ...  
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /lib/systemd/system/kubelet.service.  
Setting up kubeadm (1.22.2-00) ...  
Processing triggers for man-db (2.9.1-1) ...  
ubuntu@ip-172-31-10-100:/etc/docker$ cd /home/ubuntu  
ubuntu@ip-172-31-10-100:~$ sudo swapoff -a  
-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf  
sudo sysctl -p  
ubuntu@ip-172-31-10-100:~$ echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf  
net.bridge.bridge-nf-call-iptables=1  
ubuntu@ip-172-31-10-100:~$ sudo sysctl -p  
net.bridge.bridge-nf-call-iptables = 1  
ubuntu@ip-172-31-10-100:~$
```

```
ubuntu@ip-172-31-12-218: ~  
Selecting previously unselected package kubeadm.  
Preparing to unpack .../7-kubeadm_1.22.2-00_amd64.deb ...  
Unpacking kubeadm (1.22.2-00) ...  
Setting up conntrack (1:1.4.5-2) ...  
Setting up kubect1 (1.22.2-00) ...  
Setting up ebtables (2.0.11-3build1) ...  
Setting up socat (1.7.3.3-2) ...  
Setting up cri-tools (1.13.0-01) ...  
Setting up kubernetes-cni (0.8.7-00) ...  
Setting up kubelet (1.22.2-00) ...  
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /lib/systemd/system/kubelet.service.  
Setting up kubeadm (1.22.2-00) ...  
Processing triggers for man-db (2.9.1-1) ...  
ubuntu@ip-172-31-12-218:/etc/docker$ cd /home/ubuntu  
ubuntu@ip-172-31-12-218:~$ sudo swapoff -a  
et.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf  
sudo sysctl -p  
ubuntu@ip-172-31-12-218:~$ echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf  
net.bridge.bridge-nf-call-iptables=1  
ubuntu@ip-172-31-12-218:~$ sudo sysctl -p  
net.bridge.bridge-nf-call-iptables = 1  
ubuntu@ip-172-31-12-218:~$
```

```

ubuntu@ip-172-31-4-243: ~
38 kB]Get:8 https://packages.cloud.google.com/apt/kubernetes-xenial/main amd64 kubeadm amd64 1.22.2-00
[8718 kB]Fetched 73.8 MB in 4s (16.5 MB/s)
Selecting previously unselected package conntrack.
(Reading database ... 60400 files and directories currently installed.)
Preparing to unpack .../0-conntrack_1%3a1.4.5-2_amd64.deb ...
Unpacking conntrack (1:1.4.5-2) ...
Selecting previously unselected package cri-tools.
Preparing to unpack .../1-cri-tools_1.13.0-01_amd64.deb ...
Unpacking cri-tools (1.13.0-01) ...
Selecting previously unselected package ebtables.
Preparing to unpack .../2-ebtables_2.0.11-3build1_amd64.deb ...
Unpacking ebtables (2.0.11-3build1) ...
Selecting previously unselected package kubernetes-cni.
Preparing to unpack .../3-kubernetes-cni_0.8.7-00_amd64.deb ...
Unpacking kubernetes-cni (0.8.7-00) ...
Selecting previously unselected package socat.
Preparing to unpack .../4-socat_1.7.3.3-2_amd64.deb ...
Unpacking socat (1.7.3.3-2) ...
Selecting previously unselected package kubelet.
Preparing to unpack .../5-kubelet_1.22.2-00_amd64.deb ...
Unpacking kubelet (1.22.2-00) ...
Selecting previously unselected package kubectl.
Preparing to unpack .../6-kubectl_1.22.2-00_amd64.deb ...
Unpacking kubectl (1.22.2-00) ...
Selecting previously unselected package kubeadm.
Preparing to unpack .../7-kubeadm_1.22.2-00_amd64.deb ...
Unpacking kubeadm (1.22.2-00) ...
Setting up conntrack (1:1.4.5-2) ...
Setting up kubectl (1.22.2-00) ...
Setting up ebtables (2.0.11-3build1) ...
Setting up socat (1.7.3.3-2) ...
Setting up cri-tools (1.13.0-01) ...
Setting up kubernetes-cni (0.8.7-00) ...
Setting up kubelet (1.22.2-00) ...
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /lib/systemd/system/kubelet.service.
Setting up kubeadm (1.22.2-00) ...
Processing triggers for man-db (2.9.1-1) ...
ubuntu@ip-172-31-4-243:/etc/docker$ cd home/ubuntu
-bash: cd: home/ubuntu: No such file or directory
ubuntu@ip-172-31-4-243:/etc/docker$ cd /home/ubuntu
ubuntu@ip-172-31-4-243:~$ sudo swapoff -a
-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
sudo sysctl -p
ubuntu@ip-172-31-4-243:~$ echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf
net.bridge.bridge-nf-call-iptables=1
ubuntu@ip-172-31-4-243:~$ sudo sysctl -p
net.bridge.bridge-nf-call-iptables = 1
ubuntu@ip-172-31-4-243:~$

```

Perform this ONLY on the Master machine:

Initialize the Kubecluster

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

```
Select ubuntu@ip-172-31-10-100: ~
[addons] Applied essential addon: kube-proxy

Your Kubernetes control-plane has initialized successfully!

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

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

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

  export KUBECONFIG=/etc/kubernetes/admin.conf

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

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

kubeadm join 172.31.10.100:6443 --token m6nbfs.6zzc00t0m6p3y2q9 \
  --discovery-token-ca-cert-hash sha256:d4a8ffb3d1ccbb755aa60cf45e0837c65c6504348858a8163cf564b86e59697
```

Copy the join command and keep it in a notepad, we'll need it later.

Copy the mkdir and chown commands from the top and execute them.

Then, add a common networking plugin called flannel file as mentioned in the code.

`kubectl apply -f`

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

```
ubuntu@ip-172-31-10-100: ~
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 172.31.10.100:6443 --token m6nbfs.6zzc00t0m6p3y2q9 \
  --discovery-token-ca-cert-hash sha256:d4a8ffb3d1ccbb755aa60cf45e0837c65c6504348858a8163cf564b86e59697
ubuntu@ip-172-31-10-100:~$ mkdir -p $HOME/.kube
ubuntu@ip-172-31-10-100:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
ubuntu@ip-172-31-10-100:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-10-100:~$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
Warning: policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
podsecuritypolicy.policy/psp.flannel.unprivileged 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
```

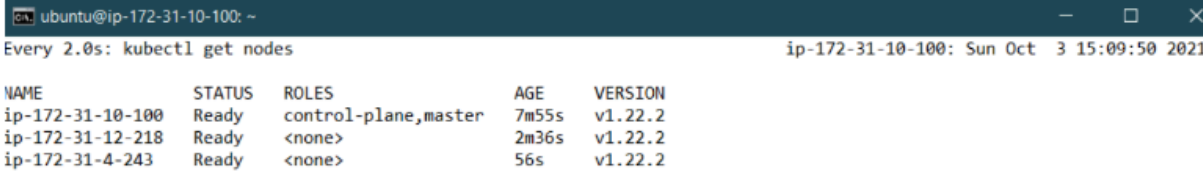
Check the created pod using this command. Now, keep a watch on all nodes using the following command:


```
watch kubectl get nodes
```

Perform this ONLY on the worker machines:

```
sudo kubeadm join <ip> --token <token> \ --discovery-token-ca-cert-hash  
<hash>
```

Now, notice the changes on the master terminal



The screenshot shows a terminal window titled 'ubuntu@ip-172-31-10-100: ~'. The command 'watch kubectl get nodes' is being executed, and the output is displayed every 2.0 seconds. The output shows three nodes in the cluster: a master node and two worker nodes. The master node is 'ip-172-31-10-100' with status 'Ready' and roles 'control-plane,master'. The two worker nodes are 'ip-172-31-12-218' and 'ip-172-31-4-243', both with status 'Ready' and roles '<none>'. All nodes are running Kubernetes version 'v1.22.2'.

NAME	STATUS	ROLES	AGE	VERSION
ip-172-31-10-100	Ready	control-plane,master	7m55s	v1.22.2
ip-172-31-12-218	Ready	<none>	2m36s	v1.22.2
ip-172-31-4-243	Ready	<none>	56s	v1.22.2

We now have a Kubernetes cluster running across 3 AWS EC2 Instances.

This cluster can be used to further deploy applications and their loads being distributed across these machines.