Name :- Anjali Ram Punsi

Class :- D20B

Roll No :- 57

Experiment no 11

Aim :- To study and implement container orchestration using Kubernetes on AWS/Azure/Google cloud platform.

Theory :-

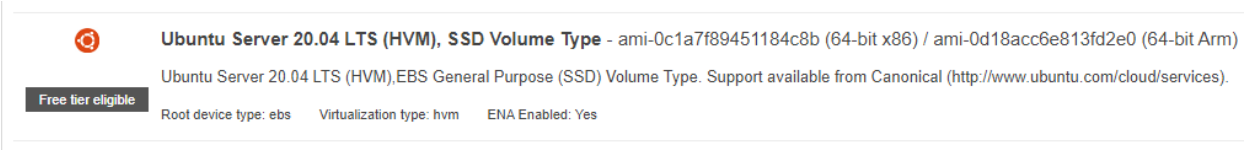
Container orchestration using Kubernetes is a methodical approach to managing and scaling containerized applications. Containers are lightweight, portable, and isolated environments that encapsulate an application and its dependencies. Kubernetes, an open-source platform, automates the deployment, scaling, and management of these containers.

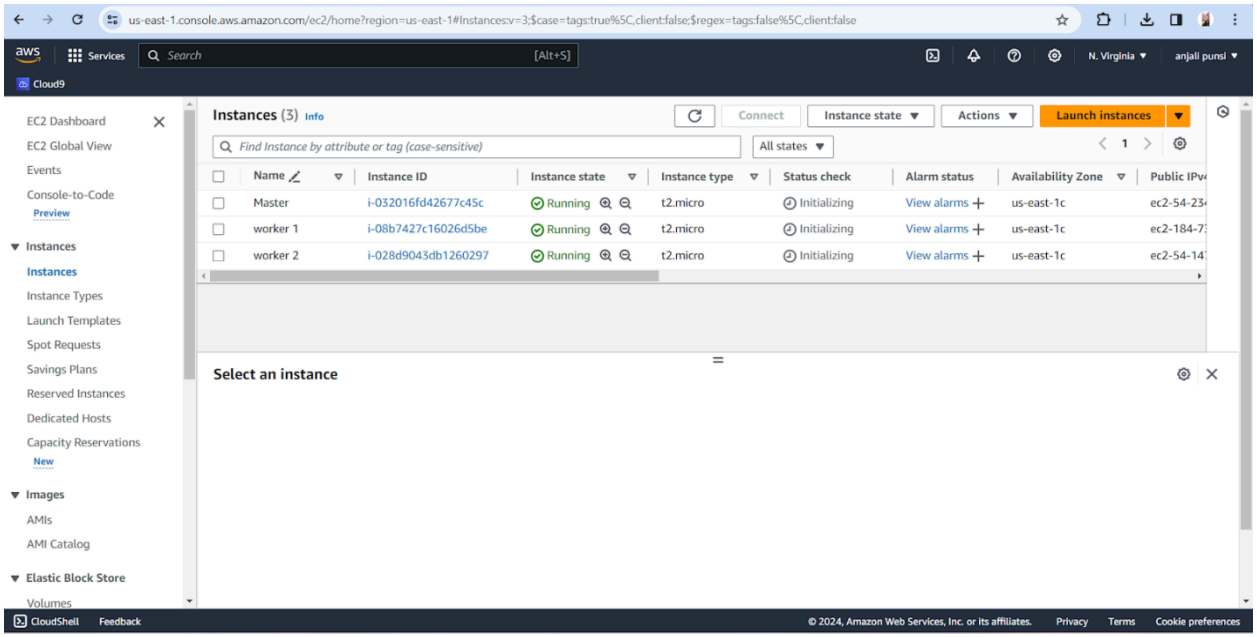
Kubernetes orchestrates containers by grouping them into logical units called pods, which can consist of one or more containers. It schedules pods to run on a cluster of machines, manages their lifecycle, and ensures that the desired state of the application is maintained. This includes scaling pods based on resource utilization, rolling out updates without downtime, and load balancing traffic across containers.

Key components of Kubernetes include the Master node, which manages the cluster and schedules pods, and Worker nodes, where containers are deployed and run. The Master node uses the kube-apiserver to handle API requests, the kube-scheduler to schedule pods, and the kube-controller-manager to manage cluster state. Worker nodes run the kubelet, which communicates with the Master node, and the kube-proxy, which manages network traffic.

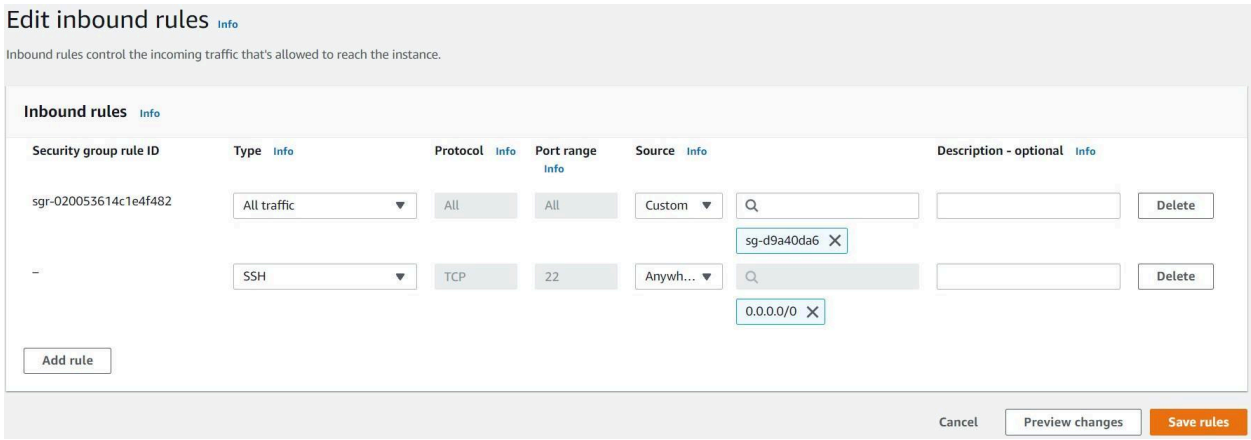
Steps and Output

Step 1 :- Setting up Kubernetes Cluster - Create 3 EC2 Ubuntu Instances on AWS. Name one as Master, the other two as worker1 and worker2.





step 2 :- Edit the Security Group Inbound Rules to allow SSH



step 3 :- Establish connection with all three machines using SSH, using the following command:
ssh -i <keyname>.pem ubuntu@<public_ip_address>

my is ssh -i <keyname>.pem ubuntu@<54.234.61.197>

```
C:\Users\Anjali>cd downloads
```

```
C:\Users\Anjali\Downloads>ssh -i Master1.pem ubuntu@3.84.241.113
The authenticity of host '3.84.241.113 (3.84.241.113)' can't be established.
ECDSA key fingerprint is SHA256:ExQKHbs0qSGZ7G1zVBsBt3bDguz8vKKXVJMSP9xlXGw.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.84.241.113' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1014-aws x86_64)

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

  System information as of Tue Mar 26 06:23:12 UTC 2024

  System load:  0.0               Processes:             96
  Usage of /:   20.4% of 7.57GB   Users logged in:       0
  Memory usage: 19%               IPv4 address for eth0: 172.31.17.39
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
```

Step 4 :- Perform all these steps at the same time on all 3 machines, unless specified otherwise.
Install Docker:

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

```
ubuntu@ip-172-31-17-39:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
ubuntu@ip-172-31-17-39:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_relea
se -cs) stable"
Repository: 'deb [arch=amd64] https://download.docker.com/linux/ubuntu jammy stable'
Description:
Archive for codename: jammy components: stable
More info: https://download.docker.com/linux/ubuntu
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-jammy.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-jammy.list
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 https://download.docker.com/linux/ubuntu jammy InRelease [48.8 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1505 kB]
```

```
ubuntu@ip-172-31-30-214:~$ sudo get-apt update
sudo: get-apt: command not found
ubuntu@ip-172-31-30-214:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:5 https://download.docker.com/linux/ubuntu jammy InRelease
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/jammy/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
ubuntu@ip-172-31-30-214:~$ sudo apt-get install -y docker-ce
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz slirp4netns
0 upgraded, 10 newly installed, 0 to remove and 19 not upgraded.
Need to get 120 MB of archives.
After this operation, 429 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63.6 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libltdl7 amd64 2.4.6-15build2 [39.6 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libslirp0 amd64 4.6.1-1build1 [61.5 kB]
Get:4 https://download.docker.com/linux/ubuntu jammy/stable amd64 containerd.io amd64 1.6.28-2 [29.7 MB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 slirp4netns amd64 1.0.1-2 [28.2 kB]
Get:6 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-buildx-plugin amd64 0.13.1-1~ubuntu.22.04~jammy [29.5 MB]
Get:7 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce-cli amd64 5:26.0.0-1~ubuntu.22.04~jammy [13.8 MB]
Get:8 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce amd64 5:26.0.0-1~ubuntu.22.04~jammy [25.1 MB]
Get:9 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce-rootless-extras amd64 5:26.0.0-1~ubuntu.22.04~jammy [9320 kB]
Get:10 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-compose-plugin amd64 2.25.0-1~ubuntu.22.04~jammy [12.1 MB]
Fetched 120 MB in 3s (37.8 MB/s)
Selecting previously unselected package pigz.
```

Step 5 :- Then, configure cgroup in a daemon.json file.
Step 1 in this :- cd /etc/docker
Next step :- cat <<EOF | sudo tee /etc/docker/daemon.json
In which we have to add

{
"exec-opts": ["native.cgroupdriver=systemd"], "log-driver": "json-file",
"log-opts": {
"max-size": "100m"
},
"storage-driver": "overlay2"
}

Ctrl+o then enter ctrl+x then to save file

All these steps now
EOF
sudo systemctl
enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker

step 6 :- now next :- 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

next 7 :- step 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/kubele
t.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/kubele
t.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%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/kubel
et.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$
```

Step 8 :- After installing Kubernetes, we need to configure internet options to allow bridging.
sudo swapoff -a after that
echo "net.bridge.bridge-nf-call-iptables=1" | sudo tee -a /etc/sysctl.conf 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 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/kubele
t.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 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/kubele
t.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/kubel
et.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:~$
```

Step 9 :- 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:d4a8ffbf3d1ccbb755aa60cf45e0837c65c6504348858a8163cf564b86e59697
```

Step 10 :- 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 flamel file as mentioned in the code.

kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-fl annel.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:d4a8ffbf3d1ccbb755aa60cf45e0837c65c6504348858a8163cf564b86e59697
ubuntu@ip-172-31-10-100:~$ mkdir -p $HOME/.kube
d -u):$(id -g) $HOME/.kube/configubuntu@ip-172-31-10-100:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/cont
ig
ubuntu@ip-172-31-10-100:~$ 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
```

step 11 :- 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

```
ubuntu@ip-172-31-10-100: ~
Every 2.0s: kubectl get nodes                                   ip-172-31-10-100: Sun Oct  3 15:09:50 2021

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.

Conclusion :- Kubernetes provides a robust framework for managing containerized applications, offering features for automation, scalability, and reliability. By abstracting away the complexities of container management, Kubernetes enables developers to focus on building and deploying applications more efficiently.