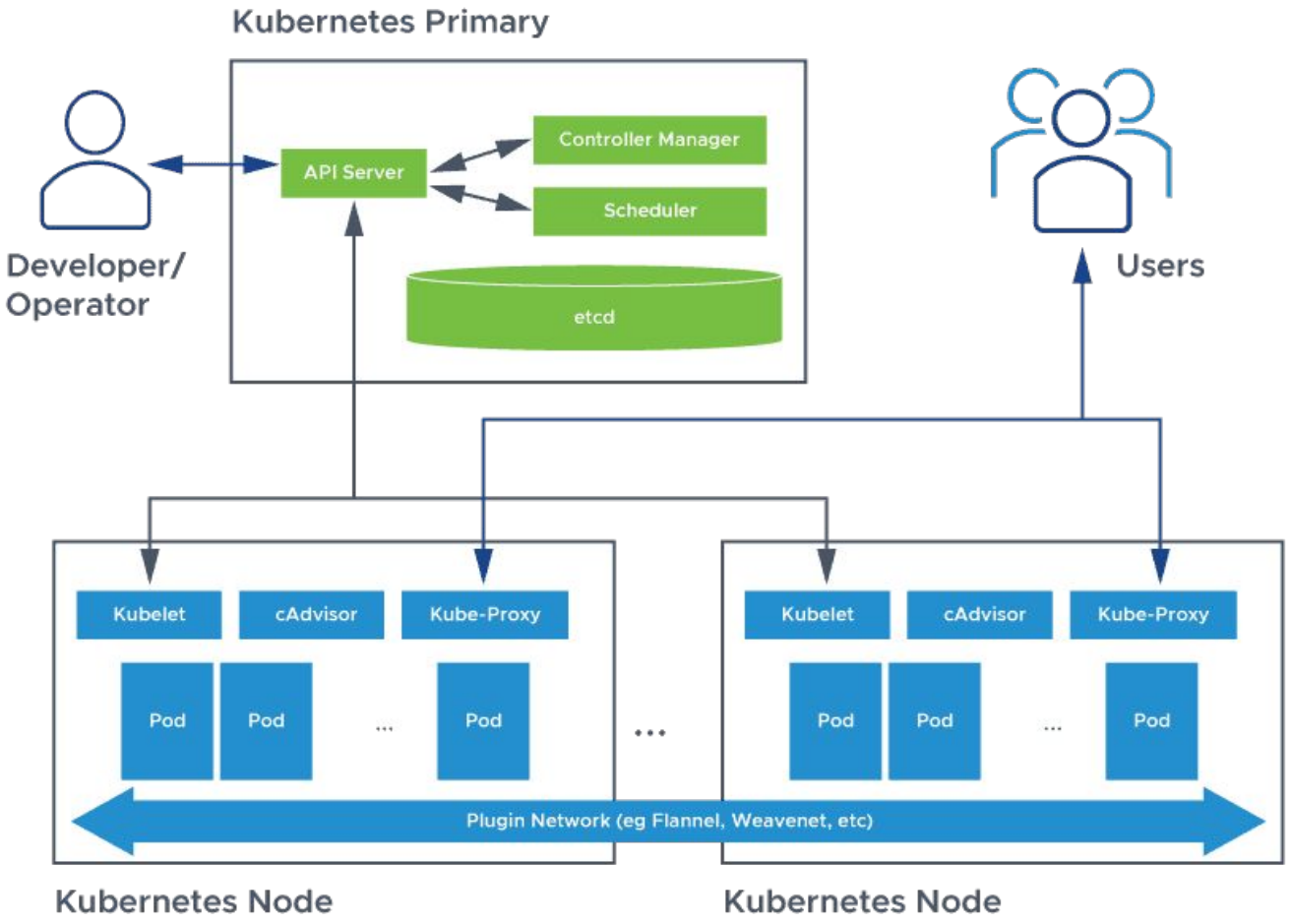
CREATING A KUBERNETES CLUSTER WITH KUBEADM


Kubernetes Primary
Developer/
Operator
API Server
Controller Manager
Scheduler
etcd
Users
Kubelet
cAdvisor
Kube-Proxy
Pod
Pod
...
Pod
Kubelet
cAdvisor
Kube-Proxy
Pod
Pod
...
Pod
Plugin Network (eg Flannel, Weavenet, etc)
Kubernetes Node
Kubernetes Node

Resources to create

- 1 ec2 instance control plane ubuntu 20.04 LTS t2.medium
- 1 ec2 instance node ubuntu 20.04 LTS t2.medium
- 1 sg for control plane
- 1 sg for nodes

Security group config control plane

6443/tcp	for Kubernetes API Server
2379-2380	for etcd server client API
6783/tcp,6784/udp	for Weavenet CNI
10248-10260	for Kubelet API, Kube-scheduler, Kube-controller-manager, Read-Only Kubelet API, Kubelet health
80,8080,443	Generic Ports
30000-32767	for NodePort Services

Security group config nodes

6783/tcp,6784/udp	for Weavenet CNI
10248-10260	for Kubelet API
30000-32767	for NodePort Services
80,8080,443	Generic Ports

Commands to run on control plane and worker node

//disable swap

swapoff -a
sed -i '/ swap / s/^\(.*\)$/#\1/g' /etc/fstab

// install containerd

wget https://github.com/containerd/containerd/releases/download/v1.6.16/containerd-1.6.16-linux-amd64.tar.gz

tar Cxzvf /usr/local containerd-1.6.16-linux-amd64.tar.gz

wget https://raw.githubusercontent.com/containerd/containerd/main/containerd.service

mkdir -p /usr/local/lib/systemd/system

mv containerd.service /usr/local/lib/systemd/system/containerd.service

systemctl daemon-reload

systemctl enable --now containerd

// install runc

```
wget https://github.com/opencontainers/runc/releases/download/v1.1.4/runc.amd64

install -m 755 runc.amd64 /usr/local/sbin/runc
```

// install CNI

```
wget https://github.com/containernetworking/plugins/releases/download/v1.2.0/cni-plugins-linux-amd64-v1.2.0.tgz

mkdir -p /opt/cni/bin

tar Cxzvf /opt/cni/bin cni-plugins-linux-amd64-v1.2.0.tgz
```

// install CRICTL

```
VERSION="v1.26.0" # check latest version in /releases page

wget https://github.com/kubernetes-sigs/cri-tools/releases/download/$VERSION/crictl-$VERSION-linux-amd64.tar.gz

sudo tar zxvf crictl-$VERSION-linux-amd64.tar.gz -C /usr/local/bin

rm -f crictl-$VERSION-linux-amd64.tar.gz
```

```
cat <<EOF | sudo tee /etc/crictl.yaml
runtime-endpoint: unix:///run/containerd/containerd.sock
image-endpoint: unix:///run/containerd/containerd.sock
timeout: 2
debug: false
pull-image-on-create: false
EOF
```

// Forwarding IPv4 and letting iptables see bridged traffic

```
cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
br_netfilter
EOF

sudo modprobe overlay
sudo modprobe br_netfilter

cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-iptables = 1
net.bridge.bridge-nf-call-ip6tables = 1
net.ipv4.ip_forward = 1
EOF
```

```
sudo sysctl --system
sysctl net.bridge.bridge-nf-call-iptables net.bridge.bridge-nf-call-ip6tables net.ipv4.ip_forward
modprobe br_netfilter
sysctl -p /etc/sysctl.conf
```

// install kubectl, kubelet and kubeadm

```
apt-get update && sudo apt-get install -y apt-transport-https curl

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

apt update -y
apt install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
```

Run on master

g images pull

control-plane-endpoint "PUBLIC_IP:6443"

ugin

https://github.com/weaveworks/weave/releases/download/v2.8.1/weave-daemonset-k8s.ya

Run on nodes

mmand obtained from kubeadm init output on all Workers nodes. Example

192.168.56.2:6443 --token … --discovery-token-ca-cert-hash sha256 . . . .

Connect to cluster from local

- Copy the content of the kubeconfig from ~/.kube/config
- And create same folder locally

  - for linux ~/.kube/config and paste the content inside

  - for windows C:\Users**user-name**\.kube\config where **user-name** is your windows user name

- run

Kubectl config use-context **context-name**

Where **context-name** in the name of the context in the .kube/config file