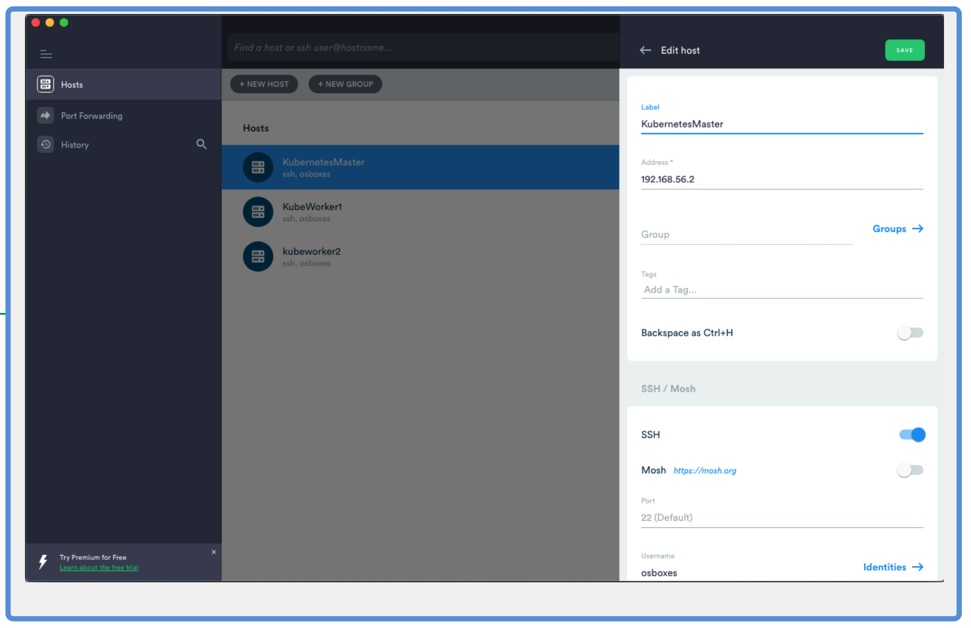
K8s Master Installation

1



Label: KubernetesMaster

Address *: 192.168.56.2

Group

Tags: Add a Tag...

Backspace as Ctrl+H

SSH / Mosh

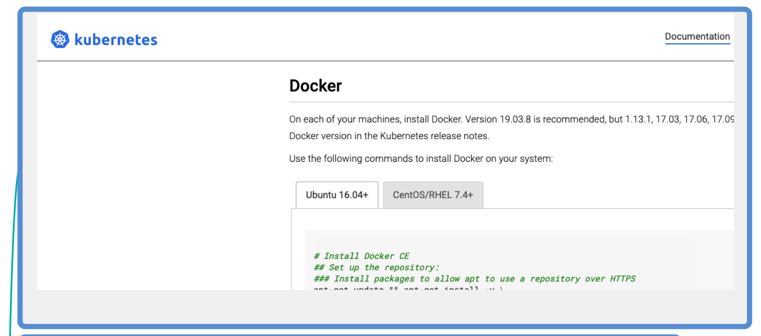
SSH

Mosh https://mosh.org

Port: 22 (Default)

Username: osboxes

2



Docker

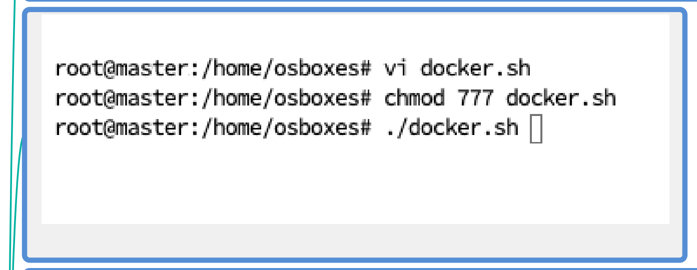
On each of your machines, install Docker. Version 19.03.8 is recommended, but 1.13.1, 17.03, 17.06, 17.09

Docker version in the Kubernetes release notes.

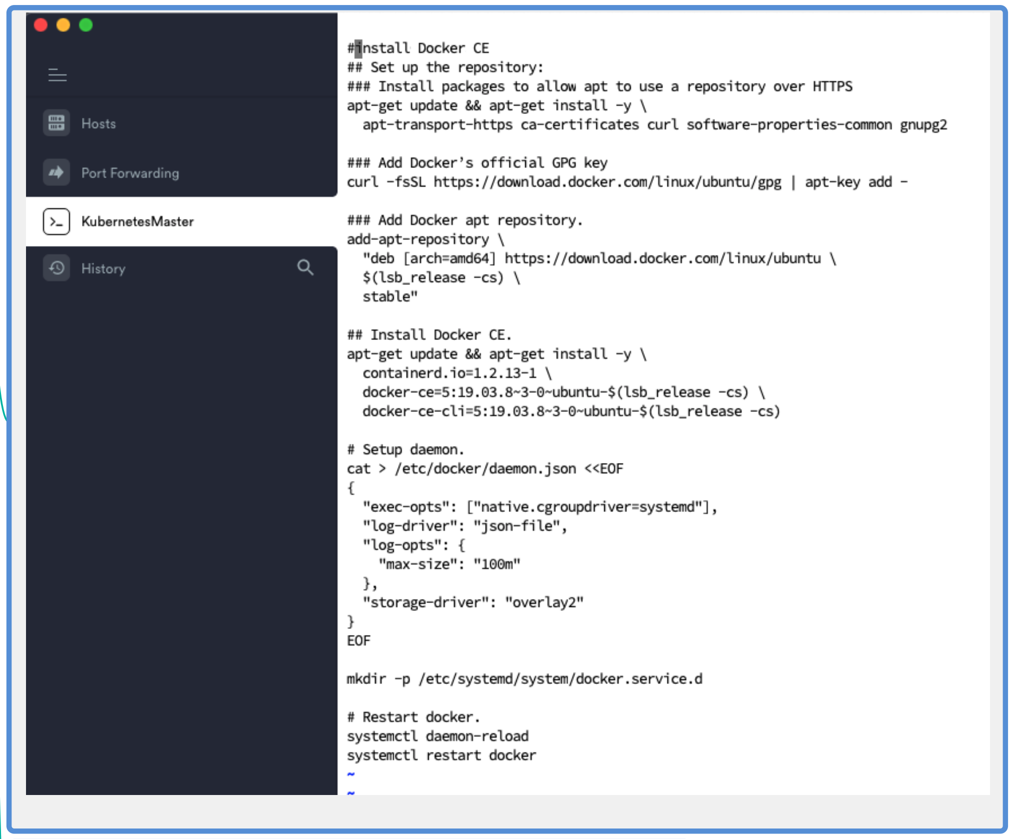
Use the following commands to install Docker on your system:

Ubuntu 16.04+ | CentOS/RHEL 7.4+

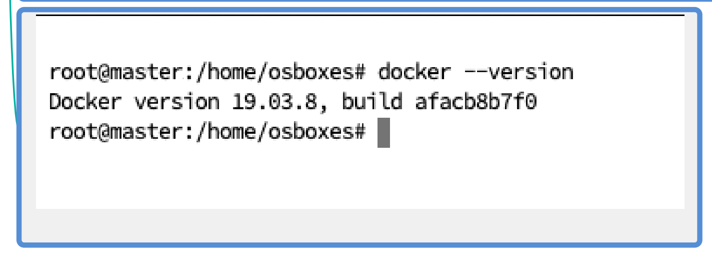
```
# Install Docker CE
## Set up the repository:
### Install packages to allow apt to use a repository over HTTPS
```



```
root@master:/home/osboxes# vi docker.sh
root@master:/home/osboxes# chmod 777 docker.sh
root@master:/home/osboxes# ./docker.sh
```

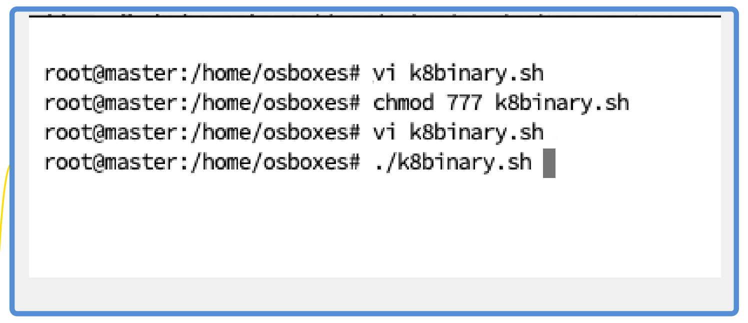


```
#Install Docker CE
## Set up the repository:
### Install packages to allow apt to use a repository over HTTPS
apt-get update && apt-get install -y \
  apt-transport-https ca-certificates curl software-properties-common gnupg2

### Add Docker's official GPG key
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | apt-key add -

### Add Docker apt repository.
add-apt-repository \
  "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) \
  stable"

## Install Docker CE.
apt-get update && apt-get install -y \
  containerd.io=1.2.13-1 \
  docker-ce=5:19.03.8~3-0~ubuntu-$(lsb_release -cs) \
  docker-ce-cli=5:19.03.8~3-0~ubuntu-$(lsb_release -cs)

# Setup daemon.
cat > /etc/docker/daemon.json <<EOF
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF

mkdir -p /etc/systemd/system/docker.service.d

# Restart docker.
systemctl daemon-reload
systemctl restart docker
~
~
```

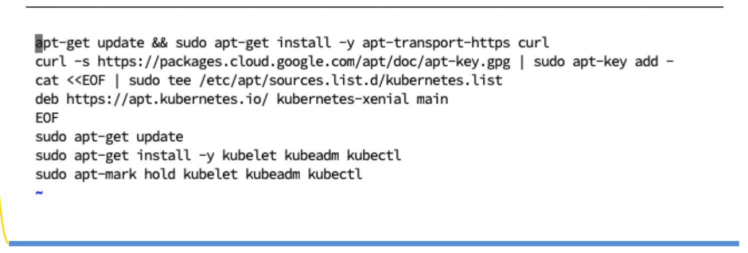


```
root@master:/home/osboxes# docker --version
Docker version 19.03.8, build afacb8b7f0
root@master:/home/osboxes#
```

3

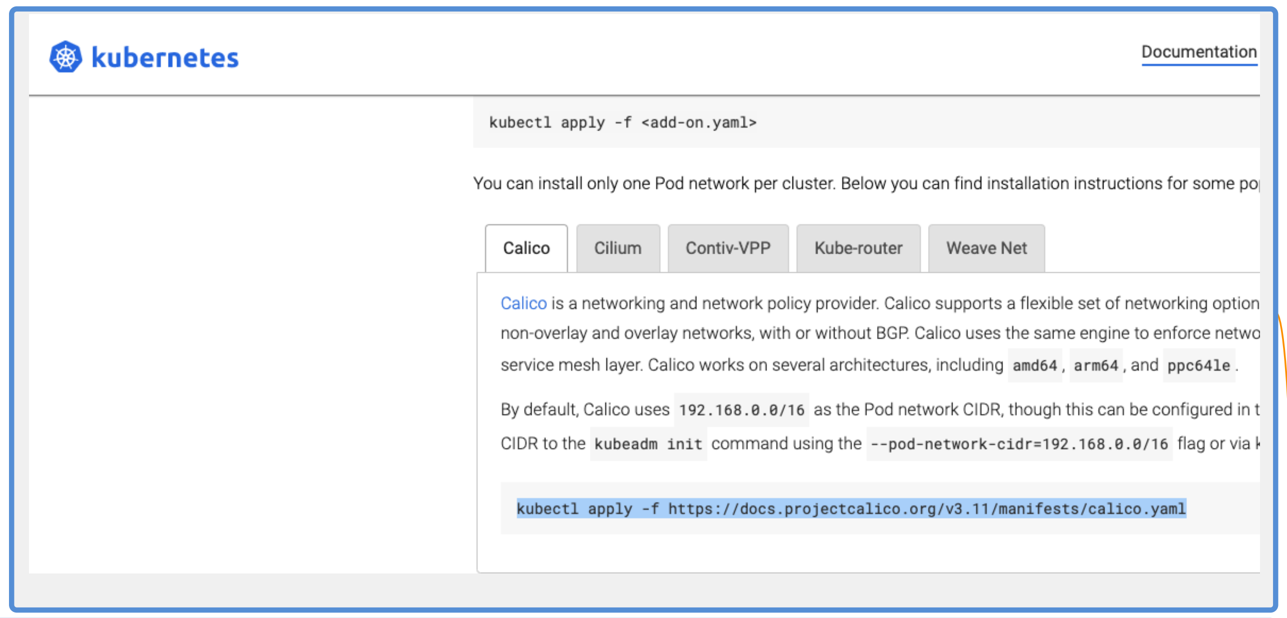


```
root@master:/home/osboxes# vi k8binary.sh
root@master:/home/osboxes# chmod 777 k8binary.sh
root@master:/home/osboxes# vi k8binary.sh
root@master:/home/osboxes# ./k8binary.sh
```



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

4



```
kubectl apply -f <add-on.yaml>
```

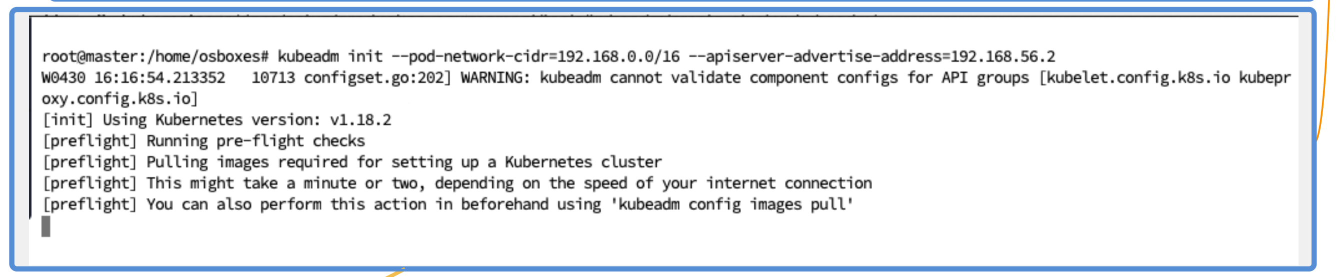
You can install only one Pod network per cluster. Below you can find installation instructions for some po

Calico | Cilium | Contiv-VPP | Kube-router | Weave Net

Calico is a networking and network policy provider. Calico supports a flexible set of networking option non-overlay and overlay networks, with or without BGP. Calico uses the same engine to enforce netwo service mesh layer. Calico works on several architectures, including `amd64`, `arm64`, and `ppc64le`.

By default, Calico uses `192.168.0.0/16` as the Pod network CIDR, though this can be configured in t CIDR to the `kubeadm init` command using the `--pod-network-cidr=192.168.0.0/16` flag or via k

```
kubectl apply -f https://docs.projectcalico.org/v3.11/manifests/calico.yaml
```



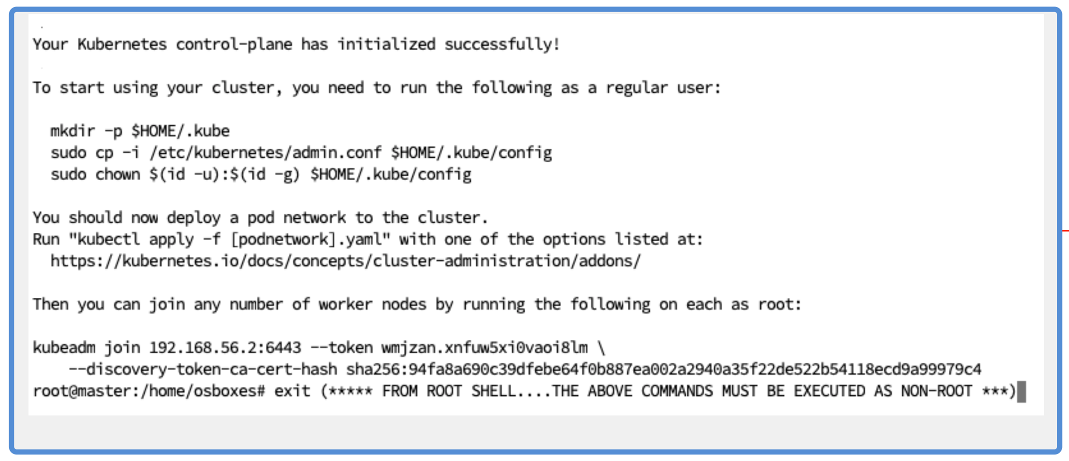
```
root@master:/home/osboxes# kubeadm init --pod-network-cidr=192.168.0.0/16 --apiserver-advertise-address=192.168.56.2
W0430 16:16:54.213352   10713 configset.go:202] WARNING: kubeadm cannot validate component configs for API groups [kubelet.config.k8s.io kubepr
oxy.config.k8s.io]
[init] Using Kubernetes version: v1.18.2
[preflight] Running pre-flight checks
[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'
```

ISSUE : Network add on - CALICO did not start, PODs are not getting deployed.
Next Steps : Change the CIDR value -- Pod CIDR must be different from network/ host CIDR

For Eg : My local Network/Node CIDR is 192.168.0.0/16 , During initialization my Pod CIDR should be different (REFER RFC 1918 for private address)

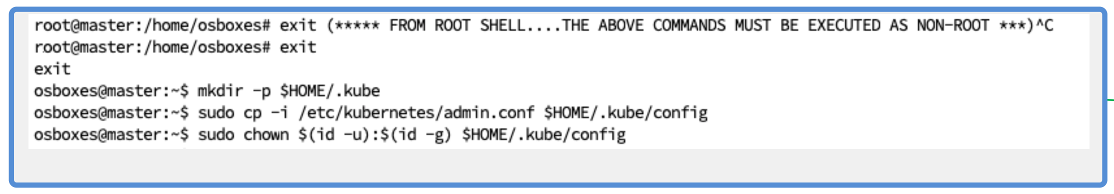
kubeadm init --pod-network-cidr=10.2.0.0/16 --apiserver-advertise-address=192.168.56.2

5



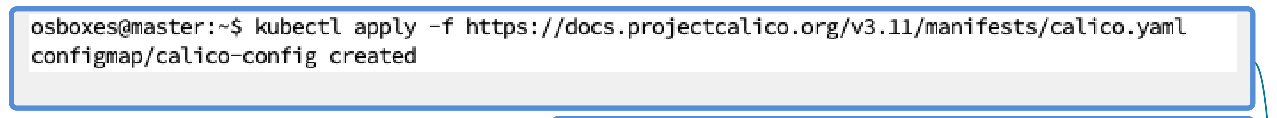
```
Your Kubernetes control-plane has initialized successfully!

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

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 192.168.56.2:6443 --token wmjzan.xnfuw5xi0vaoi8lm \
    --discovery-token-ca-cert-hash sha256:94fa8a690c39dfebe64f0b887ea002a2940a35f22de522b54118ecd9a99979c4
root@master:/home/osboxes# exit (***** FROM ROOT SHELL....THE ABOVE COMMANDS MUST BE EXECUTED AS NON-ROOT ***)
```

6

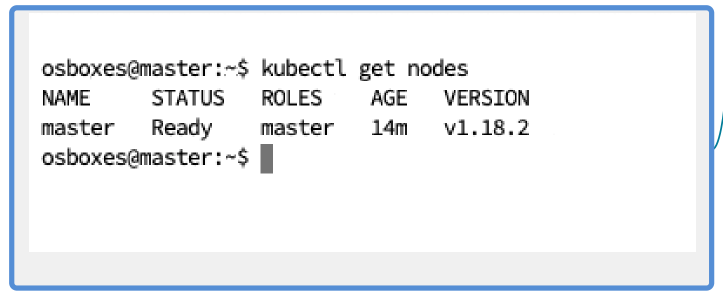


```
root@master:/home/osboxes# exit (***** FROM ROOT SHELL....THE ABOVE COMMANDS MUST BE EXECUTED AS NON-ROOT ***)^C
root@master:/home/osboxes# exit
exit
osboxes@master:~$ mkdir -p $HOME/.kube
osboxes@master:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
osboxes@master:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

8



```
osboxes@master:~$ kubectl apply -f https://docs.projectcalico.org/v3.11/manifests/calico.yaml
configmap/calico-config created
```



```
osboxes@master:~$ kubectl get nodes
NAME     STATUS   ROLES    AGE   VERSION
master   Ready    master   14m   v1.18.2
osboxes@master:~$
```