Ratul

Install and Configure Kubernetes Cluster on Ubuntu 16.04

05 Jun 2018

Download PDF

> By completing this article we'll learn about Docker orchestration- Kubernetes

Create 2 VMs and set the hostname

- kube-master
- kube-worker

In Master Node

```
sudo hostnamectl set-hostname kube-master
```

In Worker Node

```
sudo hostnamectl set-hostname kube-worker
```

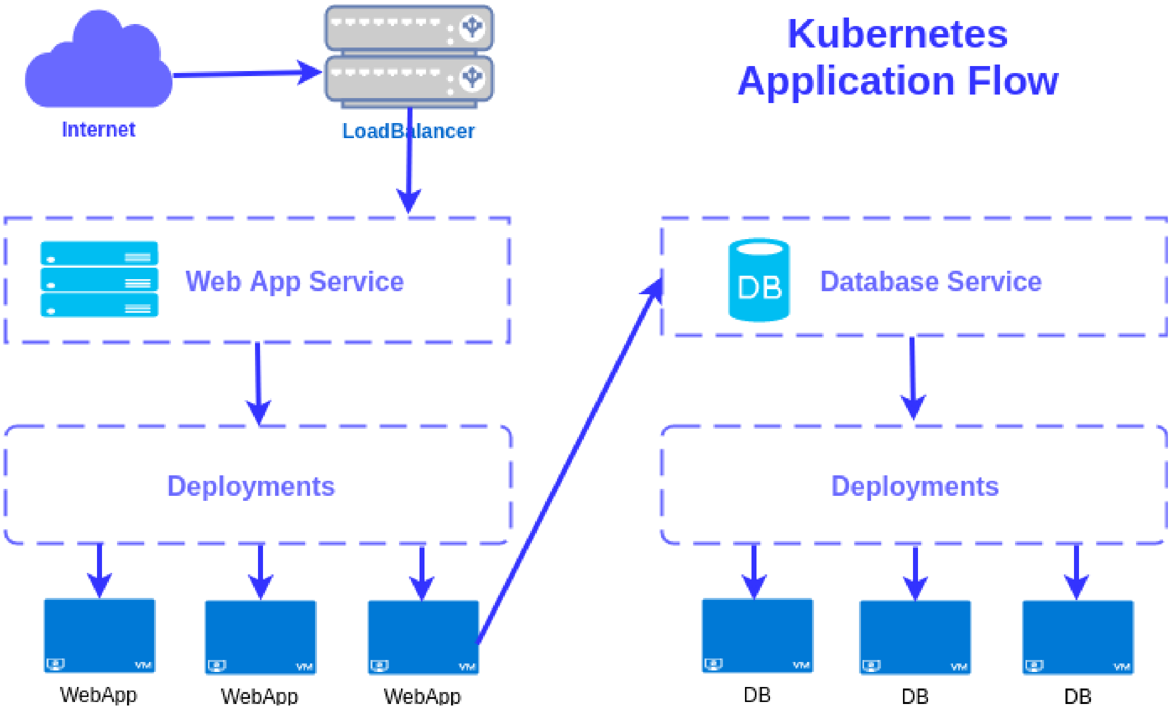
Update the hosts file in both nodes

```
sudo vim /etc/hosts
master_private_ip   kube-master
worker_private_ip   kube-worker
```

Install Docker in both nodes:

```
curl -fsSL https://download.docker.com/linux/ubun
tu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] http
```

```
s://download.docker.com/linux/ubuntu $(lsb_releas
e -cs) stable"
sudo apt-get update
sudo apt-get update && sudo apt-get install -y do
cker-ce=$(apt-cache madison docker-ce | grep 17.0
3 | head -1 | awk '{print $3}')
```



Install Kubernetes in both nodes: Installing Kubeadm, kubelet and kubectl. You will install these packages on all of your machines: **kubeadm:** the command to bootstrap the cluster. **kubelet:** the component that runs on all of the machines in your cluster and does things like starting pods and containers. **kubectl:** the command line util to talk to your cluster.

```
sudo apt-get update && sudo apt-get install -y ap
t-transport-https curl
curl -s https://packages.cloud.google.com/apt/do
c/apt-key.gpg | apt-key add -
cat **<<EOF >**/etc/apt/sources.list.d/kubernetes
.list
deb http://apt.kubernetes.io/ kubernetes-xenial m
ain
**EOF**
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
```

In Master Node

Make sure that the cgroup driver used by kubelet is the same as the one used by Docker. Verify that your Docker cgroup driver matches the kubelet config:

```
systemctl restart kubelet
```

Initialize Cluster The master is the machine where the control plane components run, including etcd (the cluster database) and the API server (which the kubectl CLI communicates with).

```
sudo kubeadm init --pod-network-cidr=10.244.0.0/16 --apiserver-advertise-address=**kube-master’s_private_ip**
```

To make kubectl work for your non-root user, you might want to run these commands (which is also a part of the kubeadm init output):

```
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 could run this:

```
export KUBECONFIG=/etc/kubernetes/admin.conf
```

Now, install a pod network add-on so that your pods can communicate with each other. It is a must…

```
kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/v0.10.0/Documentation/kube-flannel.yml
```

Install metrics-server

```
git clone https://github.com/kubernetes-incubator/metrics-server.git

cd metrics-server
kubectl apply --filename deploy/1.8+/
```

Install heapster

```
kubectl create -f https://raw.githubusercontent.com/kubernetes/kops/master/addons/monitoring-standalone/v1.7.0.yaml
```

If API Aggregator no enabled then follow this https://kubernetes.io/docs/tasks/access-kubernetes-api/configure-aggregation-layer/

```
kubectl get pods --all-namespaces
kubectl get nodes -o wide
```

In Minion Node

Joining Worker

```
sudo kubeadm join 192.168.33.10:6443 --token ujbvgu.vxvk2vml3xkcl6q4 --discovery-token-ca-cert-hash sha256:73fca98b91e8fd589f4e50e3f55f4889c9db1ee026ac647af7b6ea0af2f6c624
```

> **–token**: kube-master generated token and **–discovery-token-ca-cert-hash**: also generated by kube-master, output of (kubeadm init)

If you want to remove all configuration from any node:

```
kubeadm reset
```

Related Posts

Serving Private Content of S3 through CloudFront Signed URL 14 Jun 2018