Basic Kubernetes Cluster with 2 VMs

I'm researching for my next Unix Tutorial Project, so today I'm spinning up two new Ubuntu 19.10 VMs and configuring them as Kubernetes Master and Kubernetes Node, accessible from my macOS desktop.

This is just a bunch of notes on the topic, the project will have the full procedure (and a YouTube video, if everything works out as I expect).

NOTE: I'm using install instructions supplied by Docker and Google for their packages install. Apparently, there's a few more ways and packaing systems to get similar results specifically on Ubuntu.

Step 1: Spin Up the 2 VMs

I chose Ubuntu, but any Linux capable of running Docker should be fine.

The plan is to have one VM as Kubernetes Master (k8master) and another VM as Kubernetes Node (k8node1) – that's where our containers will actually run.

I'm using Parallels on macOS, it has this express install which means you just point it at an ISO and it does unattended

Ubuntu install. Really cool!

Step 2: Configure Host-Only Networking

Now, we need to shut down both VMs and then add host-only network adapter to each – this will be used for cluster communication and for remotely managing k8s.

IMPORTANT: this is an additional interface. The primary one is separate and used for Internet access (cause we need it to install software).

Step 3: Install Docker and Kubernetes on both VMs

Pretty standard procedure, following official install guides. This needs to be done on both Ubuntu 19.10 VMs.

The only thing is that Ubuntu 19.10 (code name Eoan in repo URLs) is not supported properly, so I ended up using previous distro codename (disco) for Docker URL:

```
$ echo "deb [arch=amd64] https://download.docker.com/linux/ubuntu disco stable" | sudo tee -a /etc/apt/sources.list.d/docker.list
$ echo "deb https://apt.kubernetes.io/ kubernetes-eoan main" | sudo tee -a /etc/apt/sources.list.d/kubernetes.list
```

```
$ sudo apt-get update
```

Install Docker

```
$ sudo apt-get install docker-ce
Reading package lists… Done
Building dependency tree
Reading state information… Done
The following additional packages will be installed:
aufs-tools cgroupfs-mount containerd.io docker-ce-cli git git-man liberror-perl pigz
Suggested packages:
git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
aufs-tools cgroupfs-mount containerd.io docker-ce docker-ce-cli git git-man liberror-perl pigz
0 upgraded, 9 newly installed, 0 to remove and 175 not upgraded.
Need to get 90.5 MB of archives.
After this operation, 418 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Install k8s (Kubernetes)

```
$ sudo apt-get install -y kubelet kubeadm kubectl
 Reading package lists… Done
 Building dependency tree
 Reading state information… Done
 The following additional packages will be installed:
   conntrack cri-tools ebtables ethtool kubernetes-cni socat
 Suggested packages:
   nftables
 The following NEW packages will be installed:
    conntrack cri-tools ebtables ethtool kubeadm kubectl
```

```
kubelet kubernetes-cni socat
 0 upgraded, 9 newly installed, 0 to remove and 175 not upgraded.
 Need to get 51.8 MB of archives.
 After this operation, 273 MB of additional disk space will be used.
```

Step 4: Setup Kubernetes Cluster

This means starting a master node:

greys@k8master:~$ **sudo kubeadm init --pod-network-cidr=10.37.0.0/16 --apiserver-advertise-address=10.37.129.3**

and then using the command from the output in previous step, join the Kubernetes node 1 to the cluster:

root@k8node1:/# **kubeadm join 10.37.129.3:6443 --token 2cqf5d.ykwwsripsqe0s530 --discovery-token-ca-cert-hash sha256:418c2ef3a98d73cddaea8b3470d7b710c384f1644b87785e7e907e8ef44a2193**

This is how we'd verify status:

```
greys@k8master:~$ kubectl get pods –all-namespaces
NAMESPACE NAME READY STATUS RESTARTS AGE
kube-system coredns-6955765f44-r567t 0/1 Pending 0 6m38s
kube-system coredns-6955765f44-wj4vw 0/1 Pending 0 6m38s
kube-system etcd-k8master 1/1 Running 0 6m50s
kube-system kube-apiserver-k8master 1/1 Running 0 6m50s
kube-system kube-controller-manager-k8master 1/1 Running 0
```

```
6m50s
kube-system kube-proxy-g5ntg 1/1 Running 0 5m4s
kube-system kube-proxy-m24gq 1/1 Running 0 6m38s
kube-system kube-scheduler-k8master 1/1 Running 0 6m50s
```

Confirm status of nodes:

```
greys@k8master:~$ kubectl get nodes
NAME       STATUS     ROLES    AGE   VERSION
k8master   NotReady   master   12m   v1.17.3
k8node1    NotReady      10m   v1.17.3
```

Step 5: Connect to Kubernetes Master from my desktop

This step will need some consideration because I already have local Kubernetes setup on my desktop (via docker-desktop), which I'd like to keep. So I'll need to add another configuration, I suspect.

That's all for now!

See Also

- Docker