



System Mining

[Follow](#)

Mar 7, 2017 · 3 min read

Setup kubernetes cluster on ubuntu 16.04 with kubeadm

Problems

I has been tried to setup kubernetes on ubuntu 16.04 with kubeadm

Steps to setup:

1. Install a secure Kubernetes cluster on your machines
2. Install a pod network on the cluster so that application components (pods) can talk to each other
3. Install a sample microservices application (a socks shop) on the cluster

Setup

1. Install a secure Kubernetes cluster on your machines

Add and prepare kubeadm package

- Install dependency library

```
root@system-mining:~$ apt-get update && apt-get install -y  
apt-transport-https
```

- Add key for new repository and add repository

```
root@system-mining:~$ curl -s  
https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-  
key add -
```

```
root@system-mining:~$ cat <<EOF >  
/etc/apt/sources.list.d/kubernetes.list  
deb http://apt.kubernetes.io/ kubernetes-xenial main  
EOF
```

- Update repository list

```
root@system-mining:~$ apt-get update
```

- Install docker.io

If you have been install docker, Ignore docker.io packages

```
# Install docker if you don't have it already.  
root@system-mining:~$ apt-get install -y docker.io
```

- Install kubelet & kubeadm *If you have been install docker, Ignore docker.io packages*

```
root@system-mining:~$ apt-get install -y kubelet kubeadm  
kubect1 kubernetes-cni
```

Notes: You have to install kubernetes-cni to enable cni network on your machine. if not, kubernetes network will not working.

You need to do this step on your all of machines that you want to run kubernetes

2. Init your master cluster *All thing you need has been installed, now you need to initialize your master cluster*

```
sminer@system-mining:~$ kubeadm init
```

```
[preflight] Running pre-flight checks  
[init] Using Kubernetes version: v1.5.3  
[tokens] Generated token: "858698.51d1418b0490485a"  
[certificates] Generated Certificate Authority key and certificate.  
[certificates] Generated API Server key and certificate  
[certificates] Generated Service Account signing keys  
[certificates] Created keys and certificates in  
"/etc/kubernetes/pki"  
[kubeconfig] Wrote KubeConfig file to disk:  
"/etc/kubernetes/kubelet.conf"  
[kubeconfig] Wrote KubeConfig file to disk:
```

```
"/etc/kubernetes/admin.conf"
[apiclient] Created API client, waiting for the control
plane to become ready
[apiclient] All control plane components are healthy after
116.296344 seconds
[apiclient] Waiting for at least one node to register and
become ready
[apiclient] First node is ready after 4.004781 seconds
[apiclient] Creating a test deployment
[apiclient] Test deployment succeeded
[token-discovery] Created the kube-discovery deployment,
waiting for it to become ready
[token-discovery] kube-discovery is ready after 10.004425
seconds
[addons] Created essential addon: kube-proxy
[addons] Created essential addon: kube-dns
```

Your Kubernetes master has initialized successfully!

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the
options listed at:
<http://kubernetes.io/docs/admin/addons/>

You can now join any number of machines by running the
following on each node:

```
kubeadm join --token=858698.51d1418b0490485a 192.168.0.13
```

Okay, your master machine has been setup, `kubeadm join --token=858698.51d1418b0490485a 192.168.0.13` is the command you need to remember to setup your node machine to join into cluster now we need to setup network on your cluster.

You can check your current node status with command

```
sminer@system-mining:~$ kubectl get node
NAME          STATUS    AGE
masterserver  Ready,master  10m
```

3. Setup kubernetes network

Kubeadm is only support CNI network in this moment, we need to install an cni network in the master machine to help pod in cluster can communicate with each other, more infor about kubernetes pod network you can read [here](#)

In this example, I'll use [weave-kube] kubectl apply -f

<https://git.io/weave-kube>(<https://github.com/weaveworks-experiments/weave-kube>) as pod networks plugin

```
kubectl apply -f https://git.io/weave-kube
```

4. Enable master node run pod [optional]

As default, kubernetes does not allow run pod on the master node, if you want to allow it, run command bellow

```
kubectl taint nodes --all dedicated-
```

5. Setup kubernetes on other node

After finish step 4, you has been completed setup master node of your kubernetes cluster. To setup other machine to join into your cluster

- Prepare your machine as step 1
- Run command `kubeadm join` with params is the secret key of your kubernetes cluser and your master node ip

```
# This is the output of command *kubeadm init* on your  
master node  
kubeadm join --token=858698.51d1418b0490485a 192.168.0.13
```

After run command, you can check to ensure that your node has been joined into cluster by run command `kubectl get node`

```
root@system-mining: kubectl get node  
NAME           STATUS    AGE  
master,master   Ready     10m  
sm_node 1       Ready     1m
```

If you forgot the cluster token, you can generate a new one with command:

```
root@system-mining:~# kubeadm token generate
206b7b.a815ac87abb0ea03
```

Now, you have been completed setup kubernetes cluster, That is useful if you have small number of machines running kubernetes, on this tutorial we have two, one master and one node.

Thanks for your reading, in the next post, i'll list out all of useful tutorial, documents that i used to getting started with kubernetes

<http://blog.system-mining.xyz/setup-kubernetes-cluster-on-ubuntu-16-04-with-kubeadm/>

