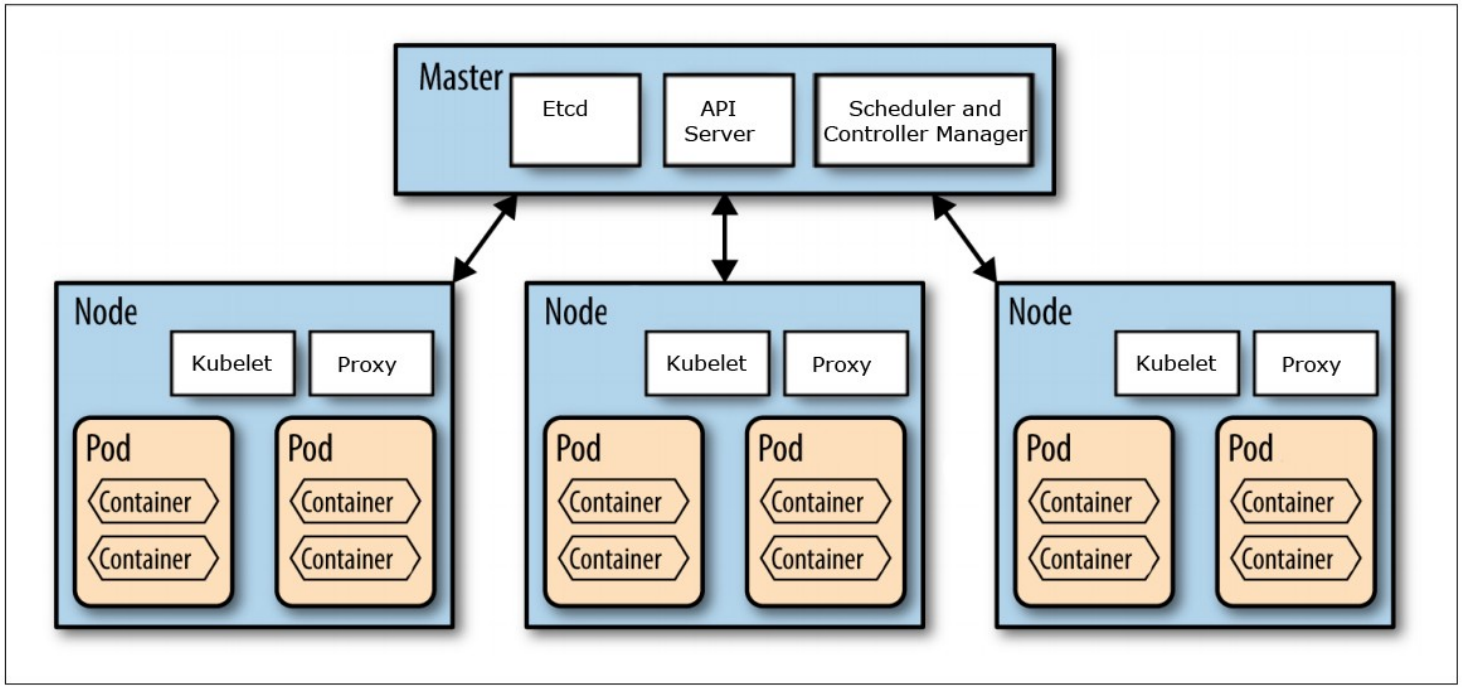
How to install Kubernetes(k8) in RHEL or Centos in just 7 steps

I had tough time to install K8 in Centos. So made this document to help others who want to install K8 on RHEL/Centos

Aravind G V Follow
Aug 25, 2018 · 3 min read



The need for RHEL is more because of lot of Enterprise company's use RHEL than Debian systems.

Here I will not be explaining k8 architecture or concepts. There are tons of documents available.

We need 2 Servers/VM/Instance installed with RHEL-7 or Centos-7.
One will be called Master and a Node

Lets start with Master node and setup and configure before node install

First thing to do is install Docker.In RHEL we will not be getting regular OpenSource Docker it comes with Docker-EE which we don't want to use in this blog.So trick is to enable centos repo.

1.To add centos yum repo. Run below command

```
agv-master$ cat <<EOF > /etc/yum.repos.d/centos.repo
[centos]

name=CentOS-7

baseurl=http://ftp.heanet.ie/pub/centos/7/os/x86_64/

enabled=1

gpgcheck=1

gpgkey=http://ftp.heanet.ie/pub/centos/7/os/x86_64/RPM-GPG-
KEY-CentOS-7

#additional packages that may be useful

[extras]

name=CentOS-$releasever - Extras

baseurl=http://ftp.heanet.ie/pub/centos/7/extras/x86_64/

enabled=1

gpgcheck=0

EOF
```

2. As a standard religious practice run yum update and then install docker

```
agv-master$ yum -y update
agv-master$ yum -y install docker
agv-master$ systemctl enable docker
agv-master$ systemctl start docker
```

3. Now time to install Kubernetes packages, we need yum repo from google Also disable selinux as docker uses cgroups and other lib which

selinux falsely treats as threat.

```
agv-master$ cat <<EOF > /etc/yum.repos.d/kubernetes.repo

[kubernetes]

name=Kubernetes

baseurl=https://packages.cloud.google.com/yum/repos/kubernet
es-el7-x86_64

enabled=1

gpgcheck=1

repo_gpgcheck=1

gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg
https://packages.cloud.google.com/yum/doc/rpm-package-
key.gpg

EOF

agv-masrter$ setenforce 0

agv-master$ vi /etc/selinux/config
     SELINUX=permissive ##Change if it is enforceing

agv-master$ yum -y install kubelet kubeadm kubectl

agv-master$ systemctl start kubelet

agv-master$ systemctl enable kubelet
```

4. Con-grates you installed K8 and now some hacks and config's to enable cluster.

```
agv-master$ cat <<EOF >  /etc/sysctl.d/k8s.conf

net.bridge.bridge-nf-call-ip6tables = 1

net.bridge.bridge-nf-call-iptables = 1

EOF

agv-master$ sysctl --system
```

```
agv-master$ echo 1 > /proc/sys/net/ipv4/ip_forward
```

5. Configure and Enable Networking to the cluster.

```
agv-master$ kubeadm init --pod-network-cidr=10.244.0.0/16

----Output-of above command-------

kubeadm join 10.0.2.203:6443 --token 49ub6n.b97ie9hxthvfyjtx
--discovery-token-ca-cert-hash
sha256:09e35eb11e535c64171d50059a584ea209a8d2479d00de30c566f
47dbc7128cf

agv-master$ kubectl get nodes
NAME                      STATUS     ROLES       AGE
VERSION

js-master.js.com          NotReady   master      17h
v1.11.1
```

6. Run these commands as regular user to setup your profile and configure cluster. ***Also make note of token which will be used to configure nodes***.

```
ec2user@agv-master$ mkdir -p $HOME/.kube
ec2user@agv-master$ sudo cp -i /etc/kubernetes/admin.conf
$HOME/.kube/config

ec2user@agv-master$ sudo chown $(id -u):$(id -g)
$HOME/.kube/config
```

Now we will enable Kubernetes cluster and will use flannel to get the config in yaml. And this should be run only on Master node

```
ec2user@agv$ kubectl apply -f
https://raw.githubusercontent.com/coreos/flannel/v0.9.1/Docu
mentation/kube-flannel.yml
```

Verify the Cluster with below command

```
ec2user@agv$ kubectl get nodes
NAME                      STATUS     ROLES     AGE
VERSION

js-master.js.com          Ready      master    17h
v1.11.1
```

This means My Master node is successfully running and I am ready to join nodes to the cluster.

Lets add one node to this cluster. You should have one instance/server and follow 1–4 steps on the new server which is called node.

7. After finishing 1–4 steps on the nod one last step will be run the notes which you made note during master setup. This step is to run on node to get registered with Master

```
agv$ kubeadm join 10.0.2.203:6443 --token
49ub6n.b97ie9hxthvfyjtx --discovery-token-ca-cert-hash
sha256:09e35eb11e535c64171d50059a584ea209a8d2479d00de30c566f
47dbc7128cf
```

```
## Run below command on Master Node #####
agv$ kubectl get nodes
NAME                          STATUS    ROLES     AGE
VERSION
ip-10-0-0-139.vpc.internal    Ready     <none>    25s
v1.11.1
ip-10-0-2-203.vpc.internal    Ready     master    17h
v1.11.1
```

Congratulations!! now you are all set. You have Master and a node. Now you can start playing with creating pods,deployments,namespaces etc.. Also I am planning to cover these in my upcoming blog posts.