Kubernetes - Build /Configuration

Build Preparation

1. CentOS7 machines
   - master
   - minion1
   - minion2
2. Disable selinux
   ```
   cat /etc/sysconfig/selinux
   SELINUX=disabled
   ```

3. Update /etc/hosts
   ```
   192.168.146.200 kubemaster
   192.168.146.201 minion1
   192.168.146.202 minion2
   ```

4. Disable firewall
   ```
   systemctl disable firewalld
   ```

5. NTP installation (Time Synch across hosts)
   ```
   yum remove crony -y
   yum install ntp -y
   systemctl enable ntpd.service
   systemctl start ntpd.service
   ```

6. update yum repo for kubernetes binaries check on 3 machines
   ```
   cat /etc/yum.repos.d/virt7-docker-common-release.repo
   [virt7-docker-common-release]
   name=virt7-docker-common-release
   baseurl=http://cbs.centos.org/repos/virt7-docker-common-release/x86_64/os/
   gpgcheck=0
   ```
7. Install etcd, kubernetes & flannel on all 3 machines
   ```
   yum install --enablerepo=virt7-docker-common-release etcd kubernetes flannel
   ```

8. Install docker, kubernetes & flannel on all minions machines
   ```
   yum install --enablerepo=virt7-docker-common-release kubernetes flannel
   ```

Kubernetes - Build /Configuration

Build Configuration (Master, Nodes)

1. How the controller-manager, scheduler, and proxy find the apiserver

/etc/kubernetes/config

```
KUBE_LOGTOSTDERR=“—logtostderr=true"
KUBE_LOG_LEVEL=“-v=0”
KUBE_ALLOW_PRIV=“—allow-privileged=false”
KUBE_MASTER=“--master=http://kubemaster:8080"
```

Kubernetes - Build /Configuration

Build Configuration (Master)

/etc/etcd/etcd.conf

```
#[Member]
#ETCD_CORS=""
ETCD_NAME="default"
ETCD_DATA_DIR="/var/lib/etcd/default.etcd"
#ETCD_WAL_DIR=""
#ETCD_LISTEN_PEER_URLS="http://localhost:2380"
ETCD_LISTEN_CLIENT_URLS="http://0.0.0.0:2379"

#[Clustering]
#ETCD_INITIAL_ADVERTISE_PEER_URLS="http://localhost:2380"
ETCD_ADVERTISE_CLIENT_URLS="http://0.0.0.0:2379"
```

/etc/kubernetes/apiserver

```
# The address on the local server to listen to.
KUBE_API_ADDRESS=“—address=0.0.0.0"
# The port on the local server to listen on.
KUBE_API_PORT=“--port=8080"
# Port minions listen on
KUBELET_PORT=“--kubelet-port=10250"
# Comma separated list of nodes in the etcd cluster
KUBE_ETCD_SERVERS=“--etcd-servers=http://kubemaster:2379"
# Address range to use for services
KUBE_SERVICE_ADDRESSES=“--service-cluster-ip-range=10.254.0.0/16"
# default admission control policies

#KUBE_ADMISSION_CONTROL=“--admission-control=NamespaceLifecycle,NamespaceExists,LimitRanger,SecurityContextDeny,ServiceAccount,ResourceQuota"
KUBE_API_ARGS=“”
```

Kubernetes - Build /Configuration

Build Configuration (Master)

Check the network 172.30.0.0/16 . It should be unused in your hosts

```
systemctl start etcd
etcdctl mkdir /kube-centos/network
etcdctl mk /kube-centos/network/config "{ \"Network\": \"172.30.0.0/16\", \"SubnetLen\": 24, \"Backend\": { \"Type\": \"vxlan\" } }”
```

Kubernetes - Build /Configuration

Build Configuration (Master, Nodes)

1. /etc/sysconfig/flanneld

```
FLANNEL_ETCD_ENDPOINTS="http://kubemaster:2379"
FLANNEL_ETCD_PREFIX=“/kube-centos/network"
FLANNEL_OPTIONS=“--iface=ens33"
```

Build Configuration (Nodes)

1. /etc/kubernetes/**kubelet**

```
KUBELET_ADDRESS="--address=0.0.0.0"
KUBELET_PORT="--port=10250"
KUBELET_HOSTNAME="--hostname-override=minion1“   # replace n with node number
KUBELET_API_SERVER="--api-servers=http://kubemaster:8080"
KUBELET_ARGS=““
```

Kubernetes - Build /Configuration

Start services (Master)

```
for SERVICES in etcd kube-apiserver kube-controller-manager kube-scheduler flanneld; do
systemctl restart $SERVICES
systemctl enable $SERVICES
systemctl status $SERVICES
done
```

Start Services (Nodes)

```
for SERVICES in kube-proxy kubelet flanneld docker; do
systemctl restart $SERVICES
systemctl enable $SERVICES
systemctl status $SERVICES
done
```

Kubernetes - Build /Configuration

Configure kubectl on the nodes

```
kubectl config set-cluster default-cluster —server=http://kubemaster:8080
kubectl config set-context default-context --cluster=default-cluster —user=default-admin
kubectl config use-context default-context
kubectl get nodes
```

Start Services (Nodes)

```
for SERVICES in kube-proxy kubelet flanneld docker; do
systemctl restart $SERVICES
systemctl enable $SERVICES
systemctl status $SERVICES
done
```

Kubernetes - Build /Configuration

Configure kubectl on the *nodes*

```
ip a l grep flannel l grep inet
```