kube-proxy开启ipvs的前置条件

```
modprobe br_netfilter

cat > /etc/sysconfig/modules/ipvs.modules <<EOF
#!/bin/bash
modprobe -- ip_vs
modprobe -- ip_vs_rr
modprobe -- ip_vs_wrr
modprobe -- ip_vs_sh
modprobe -- nf_conntrack_ipv4
EOF
chmod 755 /etc/sysconfig/modules/ipvs.modules && bash /etc/sysconfig/modules/ipvs.modules &&
lsmod | grep -e ip_vs -e nf_conntrack_ipv4
```

安装 Docker 软件

```
yum install -y yum-utils device-mapper-persistent-data lvm2

yum-config-manager \
  --add-repo \
  http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo

yum update -y && yum install -y docker-ce

## 创建 /etc/docker 目录
mkdir /etc/docker

# 配置 daemon.
cat > /etc/docker/daemon.json <<EOF
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  }
}
EOF
mkdir -p /etc/systemd/system/docker.service.d

# 重启docker服务
systemctl daemon-reload && systemctl restart docker && systemctl enable docker
```

在主节点启动 Haproxy 与 Keepalived 容器

```
导入脚本 > 运行 > 查看可用节点
```

安装 Kubeadm （主从配置）

```
cat <<EOF > /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=http://mirrors.aliyun.com/kubernetes/yum/repos/kubernetes-el7-x86_64
enabled=1
gpgcheck=0
repo_gpgcheck=0
gpgkey=http://mirrors.aliyun.com/kubernetes/yum/doc/yum-key.gpg
http://mirrors.aliyun.com/kubernetes/yum/doc/rpm-package-key.gpg
EOF

yum -y  install  kubeadm-1.15.1 kubectl-1.15.1 kubelet-1.15.1
systemctl enable kubelet.service
```

初始化主节点

```
kubeadm config print init-defaults > kubeadm-config.yaml
kubeadm init --config=kubeadm-config.yaml --experimental-upload-certs | tee kubeadm-init.log
```

加入主节点以及其余工作节点

```
执行安装日志中的加入命令即可
```

Etcd 集群状态查看

```
kubectl -n kube-system exec etcd-k8s-master01 -- etcdctl \
    --endpoints=https://192.168.92.10:2379 \
    --ca-file=/etc/kubernetes/pki/etcd/ca.crt \
    --cert-file=/etc/kubernetes/pki/etcd/server.crt \
    --key-file=/etc/kubernetes/pki/etcd/server.key cluster-health

kubectl get endpoints kube-controller-manager --namespace=kube-system  -o yaml
kubectl get endpoints kube-scheduler --namespace=kube-system  -o yaml
```

部署网络

```
kubectl apply -f kube-flannel.yml
```