

1、部署规划

在开始之前，部署Kubernetes集群机器需要满足以下几个条件：

- 一台或多台机器，操作系统 CentOS7.x-86_x64
- 硬件配置：2GB或更多RAM，2个CPU或更多CPU，硬盘30GB或更多
- 集群中所有机器之间网络互通
- 可以访问外网，需要拉取镜像
- 禁止swap分区

role	地址
k8s-master	192.168.10.128
k8s-node1	192.168.10.129
k8s-node2	192.168.10.130

2、部署环境准备

```
#根据规划设置主机名:
[root@k8a-master ~]# hostnamectl set-hostname <hostname>
# 在所有节点上关闭防火墙
[root@k8s-master ~]# systemctl stop firewalld
[root@k8s-master ~]# systemctl disable firewalld
Removed symlink /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.

# 关闭selinux
[root@k8s-master ~]# setenforce 0
[root@k8s-master ~]# sed -i 's/enforcing/disabled/' /etc/selinux/config

# 关闭swap
[root@k8s-master ~]# swapoff -a
[root@k8s-master ~]# sed -ri 's/.*swap.*/#&/' /etc/fstab

# 修改hosts
cat >> /etc/hosts << EOF
192.168.10.128 k8s-master
192.168.10.129 k8s-node1
192.168.10.130 k8s-node2
EOF

# 将桥接的IPv4流量传递到iptables的链:
[root@k8s-master ~]# cat > /etc/sysctl.d/k8s.conf << EOF
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
EOF
[root@k8s-master ~]# sysctl --system
* Applying /usr/lib/sysctl.d/00-system.conf ...
```

```

* Applying /usr/lib/sysctl.d/10-default-yama-scope.conf ...
kernel.yama.ptrace_scope = 0
* Applying /usr/lib/sysctl.d/50-default.conf ...
kernel.sysrq = 16
kernel.core_uses_pid = 1
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.default.promote_secondaries = 1
net.ipv4.conf.all.promote_secondaries = 1
fs.protected_hardlinks = 1
fs.protected_symlinks = 1
* Applying /etc/sysctl.d/99-sysctl.conf ...
* Applying /etc/sysctl.d/k8s.conf ...
* Applying /etc/sysctl.conf ...

# 时间同步
[root@k8s-master ~]# yum install ntpdate -y
[root@k8s-master ~]# ntpdate time.windows.com
3 Jul 11:31:51 ntpdate[7617]: adjust time server 40.81.94.65 offset 0.008700 sec

```

3、所有节点安装Docker/kubeadm/kubelet

- 安装Docker

```

# master
[root@k8s-master ~]# wget https://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo -O /etc/yum.repos.d/docker-ce.repo
[root@k8s-master ~]# yum -y install docker-ce-18.06.1.ce-3.el7
[root@k8s-master ~]# systemctl enable docker && systemctl start docker
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.
[root@k8s-master ~]# docker --version
Docker version 18.06.1-ce, build e68fc7a

# 节点一
[root@k8s-node1 ~]# wget https://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo -O /etc/yum.repos.d/docker-ce.repo
[root@k8s-node1 ~]# yum -y install docker-ce-18.06.1.ce-3.el7
[root@k8s-node1 ~]# systemctl enable docker && systemctl start docker
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.
[root@k8s-node1 ~]# docker --version
Docker version 18.06.1-ce, build e68fc7a

# 节点二
[root@k8s-node2 ~]# wget https://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo -O /etc/yum.repos.d/docker-ce.repo
[root@k8s-node2 ~]# yum -y install docker-ce-18.06.1.ce-3.el7
[root@k8s-node2 ~]# systemctl enable docker && systemctl start docker
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.
[root@k8s-node2 ~]# docker --version
Docker version 18.06.1-ce, build e68fc7a

```

```
#在所有节点配置docker 加速源
[root@k8s-node2 ~]# cat > /etc/docker/daemon.json << EOF
{
  "registry-mirrors": ["https://b9pmyelo.mirror.aliyuncs.com"]
}
EOF
```

- 安装kubeadm, kubelet和kubectl

```
# 添加yum源
[root@k8s-master ~]# cat > /etc/yum.repos.d/kubernetes.repo << EOF
[kubernetes]
name=Kubernetes
baseurl=https://mirrors.aliyun.com/kubernetes/yum/repos/kubernetes-el7-x86_64
enabled=1
gpgcheck=0
repo_gpgcheck=0
gpgkey=https://mirrors.aliyun.com/kubernetes/yum/doc/yum-key.gpg
https://mirrors.aliyun.com/kubernetes/yum/doc/rpm-package-key.gpg
EOF
[root@k8s-master ~]# yum install -y kubelet-1.18.0 kubeadm-1.18.0 kubectl-1.18.0
[root@k8s-master ~]# systemctl enable kubelet
Created symlink from /etc/systemd/system/multi-user.target.wants/kubelet.service to
/usr/lib/systemd/system/kubelet.service.

[root@k8s-node1 ~]# cat > /etc/yum.repos.d/kubernetes.repo << EOF
> [kubernetes]
> name=Kubernetes
> baseurl=https://mirrors.aliyun.com/kubernetes/yum/repos/kubernetes-el7-x86_64
> enabled=1
> gpgcheck=0
> repo_gpgcheck=0
> gpgkey=https://mirrors.aliyun.com/kubernetes/yum/doc/yum-key.gpg
https://mirrors.aliyun.com/kubernetes/yum/doc/rpm-package-key.gpg
> EOF
[root@k8s-node1 ~]# yum install -y kubelet-1.18.0 kubeadm-1.18.0 kubectl-1.18.0
[root@k8s-node1 ~]# systemctl enable kubelet
Created symlink from /etc/systemd/system/multi-user.target.wants/kubelet.service to
/usr/lib/systemd/system/kubelet.service.

[root@k8s-node2 ~]# cat > /etc/yum.repos.d/kubernetes.repo << EOF
> [kubernetes]
> name=Kubernetes
> baseurl=https://mirrors.aliyun.com/kubernetes/yum/repos/kubernetes-el7-x86_64
> enabled=1
> gpgcheck=0
> repo_gpgcheck=0
> gpgkey=https://mirrors.aliyun.com/kubernetes/yum/doc/yum-key.gpg
https://mirrors.aliyun.com/kubernetes/yum/doc/rpm-package-key.gpg
> EOF
[root@k8s-node2 ~]# yum install -y kubelet-1.18.0 kubeadm-1.18.0 kubectl-1.18.0
[root@k8s-node2 ~]# systemctl enable kubelet
Created symlink from /etc/systemd/system/multi-user.target.wants/kubelet.service to
```

```
/usr/lib/systemd/system/kubelet.service.
```

- 部署Kubernetes Master

由于默认拉取镜像地址k8s.gcr.io国内无法访问，这里指定阿里云镜像仓库地址。

```
kubeadm init \  
  --apiserver-advertise-address=192.168.10.128 \  
  --image-repository registry.aliyuncs.com/google_containers \  
  --kubernetes-version v1.18.0 \  
  --service-cidr=10.96.0.0/12 \  
  --pod-network-cidr=10.244.0.0/16
```

创建配置文件

```
[root@k8a-master ~]# mkdir -p $HOME/.kube  
[root@k8a-master ~]# sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config  
[root@k8a-master ~]# sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

加入节点

```
[root@k8s-node1 ~]# kubeadm join 192.168.10.128:6443 --token 41mnke.nxfxj456arcav384 \  
> --discovery-token-ca-cert-hash  
sha256:a72655d4dba17c8df7a5512414c3ccff8ac176f7539aaf165dfb27d8e81a48b1  
[root@k8s-node2 ~]# kubeadm join 192.168.10.128:6443 --token 41mnke.nxfxj456arcav384 \  
> --discovery-token-ca-cert-hash  
sha256:a72655d4dba17c8df7a5512414c3ccff8ac176f7539aaf165dfb27d8e81a48b1
```

查看节点

```
[root@k8a-master ~]# kubectl get nodes  
NAME           STATUS    ROLES    AGE   VERSION  
k8a-master     NotReady  master   81m   v1.18.0  
k8s-node1      NotReady  <none>    2m33s v1.18.0  
k8s-node2      NotReady  <none>    2m36s v1.18.0
```

#默认token有效期为24小时，当过期之后，该token就不可用了。这时就需要重新创建token，操作如下：

```
# kubeadm token create  
# kubeadm token list  
# openssl x509 -pubkey -in /etc/kubernetes/pki/ca.crt \  
| openssl rsa -pubin -outform der 2>/dev/null | openssl dgst -sha256 -hex | sed 's/^.*  
//'  
63bca849e0e01691ae14eab449570284f0c3ddeea590f8da988c07fe2729e924
```

```
# kubeadm join 192.168.31.61:6443 --token nuja6n.o3jrhsffiqs9swnu \  
--discovery-token-ca-cert-hash  
sha256:63bca849e0e01691ae14eab449570284f0c3ddeea590f8da988c07fe2729e924
```

【*】 参考文件

```
kubeadm token create --print-join-command
```

<https://kubernetes.io/docs/reference/setup-tools/kubeadm/kubeadm-join/>

- 部署CNI网络

#介绍地址

<https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/create-cluster->

kubeadm/#pod-network

Flannel搭建一

```
[root@k8a-master ~]# kubectl apply -f
https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
```

Flannel搭建二

```
[root@k8a-master ~]# wget
https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
[root@k8a-master ~]# sed -i -r "s#quay.io/coreos/flannel:.*-amd64#lizhenliang/flannel:v0.12.0-amd64#g" kube-flannel.yml
[root@k8a-master ~]# kubectl apply -f kube-flannel.yml
```

Calico 搭建

```
https://docs.projectcalico.org/getting-started/kubernetes/quickstart
kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml
```

下载完后还需要修改里面配置项:

- 根据实际网络规划修改Pod CIDR (CALICO_IPV4POOL_CIDR)
- 选择工作模式 (CALICO_IPV4POOL_IPIP), 支持**BGP (Never) **、**IPIP (Always) **、**CrossSubnet** (开启BGP并支持跨子网)

```
# kubectl apply -f calico.yaml
```

```
# kubectl get pods -n kube-system
```

• 测试kubernetes集群

- 验证Pod工作
- 验证Pod网络通信
- 验证DNS解析

在Kubernetes集群中创建一个pod, 验证是否正常运行:

```
[root@k8s-master ~]# kubectl create deployment nginx --image=nginx
deployment.apps/nginx created
[root@k8s-master ~]# kubectl expose deployment nginx --port=80 --type=NodePort
service/nginx exposed
```

```
[root@k8s-master ~]# kubectl get pod,svc
```

NAME	READY	STATUS	RESTARTS	AGE
pod/nginx-f89759699-kqbtX	0/1	ContainerCreating	0	13s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	133m
service/nginx	NodePort	10.101.216.252	<none>	80:30880/TCP	5s

```
# 访问地址: http://NodeIP:Port
```

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

4、部署 Dashboard

```
[root@k8s-master mainfeat]# kubectl apply -f \
https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.0-
beta8/aio/deploy/recommended.yaml
```

```
[root@k8s-master mainfeat]# cat /htdocs/mainfeat/dashboard_service.yaml
```

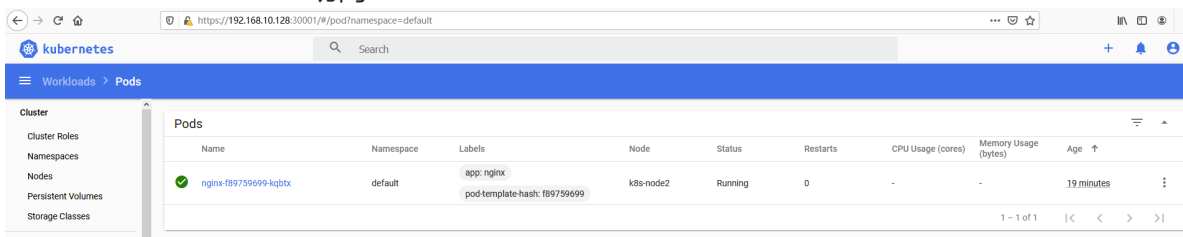
```
kind: Service
apiVersion: v1
metadata:
  labels:
    k8s-app: kubernetes-dashboard
  name: kubernetes-dashboard
  namespace: kubernetes-dashboard
spec:
  ports:
    - port: 443
      targetPort: 8443
      nodePort: 30001
  type: NodePort
selector:
  k8s-app: kubernetes-dashboard
```

```
kubectl create serviceaccount dashboard-admin -n kube-system
```

```
kubectl create clusterrolebinding dashboard-admin --clusterrole=cluster-admin --
serviceaccount=kube-system:dashboard-admin
```

```
kubectl describe secrets -n kube-system $(kubectl -n kube-system get secret | awk
'/dashboard-admin/{print $1}')
```

• kubernetes-dashboard访问



The screenshot shows the Kubernetes Dashboard web interface. The browser address bar displays the URL `https://192.168.10.128:30001/#/pod?namespace=default`. The dashboard header includes the 'kubernetes' logo and a search bar. The left sidebar contains navigation links for 'Cluster', 'Cluster Roles', 'Namespaces', 'Nodes', 'Persistent Volumes', and 'Storage Classes'. The main content area is titled 'Pods' and displays a table of pod information.

Name	Namespace	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Age
nginx-f89759699-kqbtz	default	app: nginx pod-template-hash: f89759699	k8s-node2	Running	0	-	-	19 minutes

At the bottom right of the table, it indicates '1 - 1 of 1' pods and provides navigation controls.

