链滴

使用 Kubeadm 安装 Kubernetes 集群 v1.15.3（完整步骤）

1. 各相关组件及机器环境

系统及应用版本：

- OS：CentOS 7.5 x86_64
- Container runtime：Docker 18.06.ce
- Kubernetes：1.15.3

机器配置参数：

IP地址	主机名	角色	CPU	emory
10.211.55.3	master.ilinux.io	master	> 2C	>=2G
10.211.55.4	node01.ilinux.io	node	>= C	>=2G
10.211.55.5	node02.ilinux.io	node	>= C	>=2G

2. 前置步骤（以下步骤每台master和node都需要操作）

2.1. 编辑Master和各node的/etc/hosts,解析如下

```
10.211.55.3 master.ilinux.io master
10.211.55.4 node01.ilinux.io node01
```

```
10.211.55.5 node02.ilinux.io node02
```

2.2. 主机时间同步（这里同步互联网时间）

```
[root@master ~]# systemctl enable chronyd.service
[root@master ~]# systemctl status chronyd.service
● chronyd.service - NTP client/server
   Loaded: loaded (/usr/lib/systemd/system/chronyd.service; enabled; vendor preset: enabled)
   Active: inactive (dead)
     Docs: man:chronyd(8)
           man:chrony.conf(5)
```

2.3. 关闭防火墙和Selinux服务

```
[root@master ~]# systemctl stop firewalld && systemctl disable firewalld
Removed symlink /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
[root@master ~]# setenforce 0
[root@master ~]# vim /etc/selinux/config
......
SELINUX=disabled
......
```

2.4. 禁用Swap设备（可选操作）

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

3. 部署Kubernetes集群（每台master和node都需要操作）

3.1 在Master及各Node安装Docker、kubelet及kubeadm,并以守护进程的式启动Docker和kubelet

Docker的安装参照https://docs.docker.com/install/linux/docker-ce/centos/

3.2 配置内核参数，将桥接的IPv4流量传递到iptables的链

```
[root@master ~]# cat > /etc/sysctl.d/k8s.conf <<EOF
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
EOF
[root@master ~]# sysctl --system
```

3.3 配置国内Kubenetes的yum源，由于网络原因，中国无法直接连接到Googe的网络，需要配置阿里云的yum源

```
[root@master ~]# cat <<EOF > /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://mirrors.aliyun.com/kubernetes/yum/repos/kubernetes-el7-x86_64/
```

```
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://mirrors.aliyun.com/kubernetes/yum/doc/yum-key.gpg https://mirrors.aliyun.
om/kubernetes/yum/doc/rpm-package-key.gpg
EOF
```

3.4 yum安装、启动并设置开机启动 kubelet kubeadm kubectl

```
[root@master ~]# yum install -y kubelet kubeadm kubectl
[root@master ~]# systemctl daemon-reload
[root@master ~]# systemctl start kubelet && systemctl enable kubelet
```

Kubelet负责与其他节点集群通信，并进行本节点Pod和容器生命周期的管理。Kubeadm是Kubernetes的自动化部署工具，降低了部署难度，提高效率。Kubectl是Kubernetes集群管理工具

温馨提示：如果yum安装提示找不到镜像之类的，请yum makecache更新下yum源

3.5 在master上使用kubeadm初始化集群

```
[root@master ~]# kubeadm init --kubernetes-version=1.15.3 \
--apiserver-advertise-address=0.0.0.0 \
--image-repository registry.aliyuncs.com/google_containers \
--service-cidr=10.96.0.0/12 \
--pod-network-cidr=10.244.0.0/16
```

以下是执行完毕后输出的部分信息

```
Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown (id -u):(id -g) $HOME/.kube/config

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

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 10.211.55.3:6443 --token ecnjgm.wam5flgy9n9q9uyq

--discovery-token-ca-cert-hash sha256:18a5857ef4c06a2f35822d3bf56d9fb7c6ba2c531160f9
da6c6344a071c296d
```

init参数解释说明

- --kubernetes-version 正在使用的Kubernetes程序组件的版本号，需要与kubelet 的版本号相同 。

● --pod-network-cidr : Pod网络的地址范围，其值为CIDR格式的网络地址；使用flannel网络插件，其默认地址为10.244.0.0/16 。

● --service-cidr: Service 的网络地址范围，其值为CIDR格式的网络地址，默认地址为10.96.0.0/12 。

● --apiserver-advertise-address : API server通告给其他组件的IP地址，一般应该为Master节点的IP 地址，0.0.0.0 表示节点上所有可用的地址 。

3.6 配置kubectl工具

```
[root@master ~]# mkdir -p /root/.kube
[root@master ~]# sudo cp /etc/kubernetes/admin.conf /root/.kube/config
[root@master ~]# sudo chown $(id -u):$(id -g) $HOME/.kube/config
[root@master ~]# kubectl get cs
NAME STATUS MESSAGE ERROR
etcd-0 Healthy {"health":"true"}
controller-manager Healthy ok
scheduler Healthy ok
```

上面的STATUS结果为"Healthy"，表示组件处于健康状态，否则需要检查错误，如果排除不了问题可以使用"kubeadm reset" 命令重置集群后重新初始化

```
[root@master ~]# kubectl get nodes
NAME STATUS ROLES AGE VERSION
master.ilinux.io NotReady master 10m v1.15.3
```

此时的Master处于"NotReady"（未就绪），因为集群中尚未安装网络插件，部署网络后会变成Ready

3.7 部署flannel网络（只在master上部署）

部署安装

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

查看集群状态

```
[root@master ~]# kubectl get nodes
NAME STATUS ROLES AGE VERSION
master.ilinux.io Ready master 17m v1.15.3
```

集群处于Ready状态，此时node节点可以加入集群中

注意：master节点在部署flannel过程中以及后续node节点join过程中，系统会在后台拉取flannel所网络组件的docker镜像，由于网络原因可能会拉取较慢，若未拉取完成则node状态不会变为Ready此时请先耐心等待。博主较慢的一个节点大概等待了10多分钟。可通过sudo journalctl -f -u kubelet看实时日志：

```
Aug 23 16:11:24 master.ilinux.io kubelet[20090]: W0823 16:11:24.864501   20090 cni.go:213]
```

```
nable to update cni config: No networks found in /etc/cni/net.d
Aug 23 16:11:25 master.ilinux.io kubelet[20090]: E0823 16:11:25.506314   20090 kubelet.go:21
9] Container runtime network not ready: NetworkReady=false reason:NetworkPluginNotRead
 message:docker: network plugin is not ready: cni config uninitialized
```

通过docker images|grep flannel查看镜像拉取结果

3.8 node节点加入集群

```
[root@node01 ~]# kubeadm join 10.211.55.3:6443 --token ecnjgm.wam5flgy9n9q9uyq \
--discovery-token-ca-cert-hash sha256:18a5857ef4c06a2f35822d3bf56d9fb7c6ba2c531160f9
da6c6344a071c296d
```

以下是部分输出信息

This node has joined the cluster:

* Certificate signing request was sent to apiserver and a response was received.

* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

执行完毕后稍等一会，在主节点上查看集群的状态，到这里我们一个最简单的包含最心组件的集群搭建完毕！

```
[root@master ~]# kubectl get nodes
NAME STATUS ROLES AGE VERSION
master.ilinux.io Ready master 34m v1.15.3
node01.ilinux.io Ready <none> 6m14s v1.15.3
node02.ilinux.io Ready <none> 6m8s v1.15.3
```

4. 安装其他附件组件

4.1 查看集群信息

查看集群的API通告地址

```
[root@master ~]# kubectl cluster-info
Kubernetes master is running at https://10.211.55.3:6443
KubeDNS is running at https://10.211.55.3:6443/api/v1/namespaces/kube-system/services/k
be-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
```

查看集群的版本

```
[root@master ~]# kubectl version --short
Client Version: v1.15.3
Server Version: v1.15.3
```

4.2 安装Dashboard，使用UI界面管理集群

创建dashboard的yaml文件

```
[root@master ~]# wget https://raw.githubusercontent.com/kubernetes/dashboard/v1.10.1/sr
/deploy/recommended/kubernetes-dashboard.yaml
```

修改部分配置文件内容

```
[root@master ~]# sed -i 's/k8s.gcr.io/loveone/g' kubernetes-dashboard.yaml
[root@master ~]# sed -i '/targetPort:/a\ \ \ \ \ \ nodePort: 30001\n\ \ type: NodePort' kubern
tes-dashboard.yaml
```

部署Dashboard

```
[root@master ~]# kubectl create -f kubernetes-dashboard.yaml
secret/kubernetes-dashboard-certs created
serviceaccount/kubernetes-dashboard created
role.rbac.authorization.k8s.io/kubernetes-dashboard-minimal created
rolebinding.rbac.authorization.k8s.io/kubernetes-dashboard-minimal created
deployment.apps/kubernetes-dashboard created
service/kubernetes-dashboard created
```

4.3 创建完成后，检查各服务运行状态

```
[root@master ~]# kubectl get deployment kubernetes-dashboard -n kube-system
NAME                   READY   UP-TO-DATE   AVAILABLE   AGE
kubernetes-dashboard   1/1     1            1           6d18h
[root@master ~]# kubectl get services -n kube-system
NAME                   TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)                  AGE
kube-dns               ClusterIP   10.96.0.10       <none>        53/UDP,53/TCP,9153/TCP   6d19h
kubernetes-dashboard   NodePort    10.107.246.132   <none>        443:30001/TCP            6d1
h
[root@master ~]# netstat -ntlp|grep 30001
tcp6       0      0 :::30001                :::*                    LISTEN      20739/kube-proxy
```

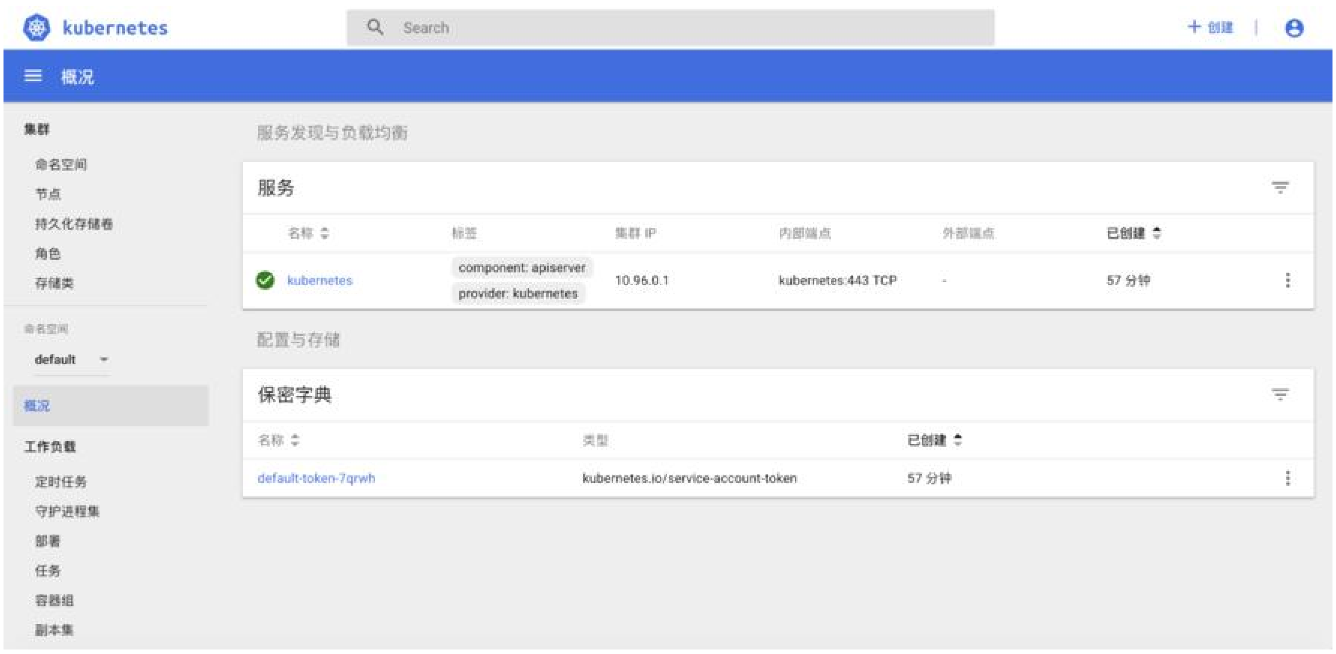
4.4 查看访问Dashboard的token

查看访问Dashboard的token

```
[root@master ~]# kubectl create serviceaccount dashboard-admin -n kube-system

serviceaccount/dashboard-admin created

[root@master ~]# kubectl create clusterrolebinding dashboard-admin --clusterrole=cluster-a
min --serviceaccount=kube-system:dashboard-admin

clusterrolebinding.rbac.authorization.k8s.io/dashboard-admin created

[root@master ~]# kubectl describe secrets -n kube-system $(kubectl -n kube-system get secr
t | awk '/dashboard-admin/{print $1}')
```

```
Name: dashboard-admin-token-9hglw

Namespace: kube-system

Labels: <none>

Annotations: kubernetes.io/service-account.name: dashboard-admin

kubernetes.io/service-account.uid: 30efdd50-92bd-11e9-91e3-000c296bd9bc

Type: kubernetes.io/service-account-token

Data

====

ca.crt: 1025 bytes

namespace: 11 bytes

token:
```

4.5 打开控制台体验~



服务发现与负载均衡

服务

名称	标签	集群 IP	内部端点	外部端点	已创建
kubernetes	component: apiserver provider: kubernetes	10.96.0.1	kubernetes:443 TCP	-	57 分钟

配置与存储

保密字典

名称	类型	已创建
default-token-7qrwh	kubernetes.io/service-account-token	57 分钟

原文链接：使用 Kubeadm 安装 Kubernetes 集群 v1.15.3（完整步骤）