

kubernetes单节点快速安装文档

环境:

内存: 4G

CPU: 单核

系统: CentOS Linux release 7.4.1708 (Core)

前提:

配置hosts, 如果不添加host, 在部署node节点的时候, 会出现node端添加成功, master端无法显示的问题

host禁止使用 '_' (下划线), 不然在后面安装的时候会报错, 可以使用数字, 字母, - (中横线)

ip	hosthosthost
192.168.253.100	k8s-master
192.168.253.101	k8s-node01

Master端部署

1、安装环境依赖

```
$ sudo yum install ebtables ethtool iproute iptables socat util-linux wget  
-y  
$ sudo yum install -y nfs-utils rpcbind ## NFSNFS
```

2、安装docker 17.03

```
$ sudo wget -O -  
https://raw.githubusercontent.com/cherryleo/scripts/master/centos7-install  
-docker.sh | sudo sh  
  
#!/bin/bash  
  
# step 0: aliyun yum  
# mv /etc/yum.repos.d/CentOS-Base.repo  
/etc/yum.repos.d/CentOS-Base.repo.backup  
# wget -O /etc/yum.repos.d/CentOS-Base.repo  
http://mirrors.aliyun.com/repo/Centos-7.repo  
# yum makecache  
  
# step 1:  
yum install -y yum-utils device-mapper-persistent-data lvm2  
  
# Step 2:  
yum-config-manager --add-repo
```

```
http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo
```

```
# Step 3: Docker-CE
```

```
yum makecache fast
```

```
# yum -y install docker-ce
```

```
# docker-ce
```

```
# Docker-CE:
```

```
# Step 1: Docker-CE:
```

```
# yum list docker-ce.x86_64 --showduplicates | sort -r
```

```
# Loading mirror speeds from cached hostfile
```

```
# Loaded plugins: branch, fastestmirror, langpacks
```

```
# docker-ce.x86_64 17.03.1.ce-1.el7.centos
```

```
docker-ce-stable
```

```
# docker-ce.x86_64 17.03.1.ce-1.el7.centos
```

```
@docker-ce-stable
```

```
# docker-ce.x86_64 17.03.0.ce-1.el7.centos
```

```
docker-ce-stable
```

```
# Available Packages
```

```
# Step2 : Docker-CE: (VERSION 17.03.0.ce-1-1.el7.centos)
```

```
# sudo yum -y install docker-ce-[VERSION]
```

```
# docker
```

```
yum install -y --setopt=obsoletes=0 docker-ce-17.03.0.ce-1.el7.centos
```

```
docker-ce-selinux-17.03.0.ce-1.el7.centos
```

```
# cat << EOF > /etc/docker/daemon.json
```

```
# {
```

```
# "storage-driver": "overlay2",
```

```
# "storage-opts": [
```

```
# "overlay2.override_kernel_check=true"
```

```
# ]
```

```
# }
```

```
# EOF
```

```
# Step 4: Docker
systemctl start docker.service
systemctl enable docker.service
```

3、配置环境

```
$ sudo swapoff -a
$ sudo systemctl disable firewalld
$ sudo systemctl stop firewalld
$ sudo sysctl net.bridge.bridge-nf-call-iptables=1
$ export KUBERNETES_VERSION="1.10.0"
```

4、安装kubernetes相关服务

```
wget -O -
https://raw.githubusercontent.com/cherryleo/cherryleo/master/install-k8s-p
ackages.sh | sudo -E bash

"http://file.odc.sunline.cn/download/ServerSoftware/kubernetes/"

#!/bin/bash
set -xue
set -o pipefail

KUBERNETES_VERSION=${KUBERNETES_VERSION:-"1.10.0"}
Ubuntu=$(cat /etc/*release | grep VERSION_CODENAME) || true
CentOS=$(cat /etc/*release | grep CENTOS_MANTISBT_PROJECT_VERSION) || true

# CentOS7+, Ubuntu16.04
function install() {
    if [[ ${Ubuntu##*=} =~ "xenial" ]]; then
        apt-get install -y wget
        wget
        https://cherryleo-1253732882.cos.ap-chengdu.myqcloud.com/kubernetes-cni_0.
        6.0-00_amd64.deb
        wget
        https://cherryleo-1253732882.cos.ap-chengdu.myqcloud.com/kubelet_${KUBERNE
        TES_VERSION}-00_amd64.deb
        wget
        https://cherryleo-1253732882.cos.ap-chengdu.myqcloud.com/kubeadm_${KUBERNE
        TES_VERSION}-00_amd64.deb
        wget
        https://cherryleo-1253732882.cos.ap-chengdu.myqcloud.com/kubect1_${KUBERNE
        TES_VERSION}-00_amd64.deb
        dpkg -i kubernetes-cni_0.6.0-00_amd64.deb
        kubelet_${KUBERNETES_VERSION}-00_amd64.deb
        kubeadm_${KUBERNETES_VERSION}-00_amd64.deb
        kubect1_${KUBERNETES_VERSION}-00_amd64.deb
        systemctl enable kubelet.service
```

```
    elif [[ ${CentOS##*=} =~ "7" ]]; then
        yum install -y wget
        wget
        https://cherryleo-1253732882.cos.ap-chengdu.myqcloud.com/kubernetes-cni-0.
        6.0-0.x86_64.rpm
        wget
        https://cherryleo-1253732882.cos.ap-chengdu.myqcloud.com/kubelet-${KUBERNE
        TES_VERSION}-0.x86_64.rpm
        wget
        https://cherryleo-1253732882.cos.ap-chengdu.myqcloud.com/kubeadm-${KUBERNE
        TES_VERSION}-0.x86_64.rpm
        wget
        https://cherryleo-1253732882.cos.ap-chengdu.myqcloud.com/kubect1-${KUBERNE
        TES_VERSION}-0.x86_64.rpm
        yum install -y kubernetes-cni-0.6.0-0.x86_64.rpm
        kubelet-${KUBERNETES_VERSION}-0.x86_64.rpm
        kubeadm-${KUBERNETES_VERSION}-0.x86_64.rpm
        kubect1-${KUBERNETES_VERSION}-0.x86_64.rpm
        systemctl enable kubelet.service
    else
        echo "The current operating system version is not supported."
    fi
}
```

```
install
```

5、编辑kubeadm配置

```
$ sudo docker info | grep -i cgroup          # docker cgroup driver

# /etc/systemd/system/kubelet.service.d/10-kubeadm.conf
# 10-kubeadm.confcgroupdocker

$ vim /etc/systemd/system/kubelet.service.d/10-kubeadm.conf
[Service]
Environment="KUBELET_KUBECONFIG_ARGS=--bootstrap-kubeconfig=/etc/kubernetes/
bootstrap-kubelet.conf --kubeconfig=/etc/kubernetes/kubelet.conf"
Environment="KUBELET_SYSTEM_PODS_ARGS=--pod-manifest-path=/etc/kubernetes/
manifests --allow-privileged=true"
Environment="KUBELET_NETWORK_ARGS=--network-plugin=cni
--cni-conf-dir=/etc/cni/net.d --cni-bin-dir=/opt/cni/bin"
Environment="KUBELET_DNS_ARGS=--cluster-dns=10.96.0.10
--cluster-domain=cluster.local"
Environment="KUBELET_AUTHZ_ARGS=--authorization-mode=Webhook
--client-ca-file=/etc/kubernetes/pki/ca.crt"
# Value should match Docker daemon settings.
# Defaults are "cgroupfs" for Debian/Ubuntu/OpenSUSE and "systemd" for
Fedora/CentOS/RHEL
Environment="KUBELET_CGROUP_ARGS=--cgroup-driver=cgroupfs"
Environment="KUBELET_CADVISOR_ARGS=--cadvisor-port=0"
Environment="KUBELET_CERTIFICATE_ARGS=--rotate-certificates=true"
Environment="KUBE_PAUSE=--pod-infra-container-image=ccr.ccs.tencentyun.com
/cherryleo/pause-amd64:3.0"
ExecStart=
ExecStart=/usr/bin/kubelet $KUBELET_KUBECONFIG_ARGS
$KUBELET_SYSTEM_PODS_ARGS $KUBELET_NETWORK_ARGS $KUBELET_DNS_ARGS
$KUBELET_AUTHZ_ARGS $KUBELET_CGROUP_ARGS $KUBELET_CADVISOR_ARGS
$KUBELET_CERTIFICATE_ARGS $KUBE_PAUSE $KUBELET_EXTRA_ARGS
```

6、重启服务

```
$ sudo systemctl daemon-reload
$ sudo systemctl stop kubelet
```

7、安装k8smaster节点

7.1 创建配置

//修改pod默认网段 可根据需求是否修改POD网段

// 修改nodeport端口范围 默认范围为3000-32767，可根据需求自行修改

```
$ cat >config.yaml <<EOF
apiVersion: kubeadm.k8s.io/v1alpha1
kind: MasterConfiguration
api:
  advertiseAddress: 192.168.253.148    #IP
networking:
  podSubnet: 10.244.0.0/16    # podflannel.yaml
apiServerCertSANS:
- 192.168.253.148 # IP
imageRepository: ccr.ccs.tencentyun.com/cherryleo
kubernetesVersion: v${KUBERNETES_VERSION}
EOF
```

7.2 执行安装

```
$ sudo -E kubeadm init --config=config.yaml
...
...
Your Kubernetes master has initialized successfully!
(123)
1.sudo sysctl net.bridge.bridge-nf-call-iptables=1
2.kubeadm reset
3.sudo -E kubeadm init --config=config.yaml

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/
You can now join any number of machines by running the following on each
node
as root:
  kubeadm join 192.168.253.100:6443 --token hwrvlq.zliewykh54whepp
--discovery-token-ca-cert-hash
sha256:03df785551b9c7474874f828bf265e24865f6e16f00e0ad4e0436c8e1cf472b1
# node
```

7.3启动服务

```
$ sudo systemctl start kubelet
```

7.4 创建kubectl配置文件

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

7.5 安装插件

```
# flannelpod
kubectl apply -f
https://raw.githubusercontent.com/cherryleo/k8s-apps/master/k8s-flannel/flannel.yaml

# dashboard
$ kubectl apply -f
https://raw.githubusercontent.com/cherryleo/k8s-apps/master/k8s-dashboard/kubernetes-dashboard.yaml

# admin
$ kubectl apply -f
https://raw.githubusercontent.com/cherryleo/k8s-apps/master/k8s-dashboard/admin-user.yaml
```

7.6 查看集群状态

```
$ kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
localhost.localdomain	Ready	master	2h	v1.10.0


```
$ kubectl get pods --all-namespaces
```

NAMESPACE	NAME	STATUS	RESTARTS	AGE	READY
kube-system	etcd-localhost.localdomain	Running	0	2h	1/1
kube-system	kube-apiserver-localhost.localdomain	Running	0	2h	1/1
kube-system	kube-controller-manager-localhost.localdomain	Running	0	2h	1/1
kube-system	kube-dns-7dd59b9bdb-jwgwx	Running	0	2h	3/3
kube-system	kube-flannel-ds-h2tz2	Running	0	2h	1/1
kube-system	kube-proxy-dpm9z	Running	0	2h	1/1
kube-system	kube-scheduler-localhost.localdomain	Running	0	2h	1/1
kube-system	kubernetes-dashboard-6888bf8db6-mvvqc	Running	0	2h	1/1

7.8 获取token

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

7.8 访问https://ip:30080进入登陆页面，使用获取的token登陆

使用火狐浏览器，dahsboard是谷歌的angular.js写的，和部分浏览器不兼容

Kubernetes 仪表板

☐ Kubeconfig

请选择您已配置用来访问集群的 kubeconfig 文件，请浏览[配置对多个集群的访问](#)一节，了解更多关于如何配置和使用 kubeconfig 文件的信息

☒ 令牌

每个服务帐号都有一条保密字典保存持有者令牌，用来在仪表板登录，请浏览[验证](#)一节，了解更多关于如何配置和使用持有者令牌的信息

输入令牌

登录

跳过

kubernetes

Search

+ 创建 |

集群

集群

命名空间

节点

持久化存储卷

角色

存储类

命名空间

default

概况

工作负载

定时任务

守护进程集

部署

任务

容器组

副本集

副本控制器

命名空间

名称	标签	状态	已创建
default	-	Active	4 小时
kube-public	-	Active	4 小时
kube-system	-	Active	4 小时

节点

名称	标签	已就绪	CPU 请求值 (核)	CPU 限制值 (核)	内存请求值 (字节)	内存限制值 (字节)	已创建
localhost.localdomain	beta.kubernetes.io/arch... beta.kubernetes.io/os: li... kubernetes.io/hostnam... node-role.kubernetes.io...	True	0.81 (81.00%)	0 (0.00%)	110 Mi (6.03%)	170 Mi (9.32%)	4 小时

角色

名称	角色类型	命名空间	已创建
flannel	Cluster Role	All Namespaces	2 小时

node节点添加

1、安装环境依赖 (node端)

```
$ sudo yum install ebtables ethtool iproute iptables socat util-linux wget -y
```

2、安装docker 17.03 (node端)

```
$ sudo wget -O - https://raw.githubusercontent.com/cherryleo/scripts/master/centos7-install-docker.sh | sudo sh
```

3、配置环境 (node端)

```
$ sudo swapoff -a  
$ sudo systemctl disable firewalld  
$ sudo systemctl stop firewalld  
$ sudo sysctl net.bridge.bridge-nf-call-iptables=1
```

4、安装kubernetes相关服务 (node端)

```
$ wget -O - https://raw.githubusercontent.com/cherryleo/cherryleo/master/install-k8s-packages.sh | sudo -E bash
```

5、生成token信息 (master)

此信息非必要执行，如果忘记master初始化生成的token信息可以重新生成 (master安装第 7.2 执行安装)

```
$ kubeadm token create --print-join-command  
kubeadm join 192.168.253.100:6443 --token hwrvlq.zliewykhh54whepp  
--discovery-token-ca-cert-hash  
sha256:03df785551b9c7474874f828bf265e24865f6e16f00e0ad4e0436c8e1cf472b1
```

6、添加node节点 (node端: 执行步骤5生成的语句)

```
$ kubeadm join 192.168.253.100:6443 --token hwrvlq.zliewykhh54whepp  
--discovery-token-ca-cert-hash  
sha256:03df785551b9c7474874f828bf265e24865f6e16f00e0ad4e0436c8e1cf472b1
```

注：如果添加失败

```
$ kubeadm join 192.168.253.100:6443 --token hwrvtq.z1ewykh54whepp --discovery-token-ca-cert-hash sha256:03df785551b9c7474874f828bf265e24865f6e16f00e0ad4e0436c8e1cf472b1
```

```
$ kubectl get node
```

NAME	STATUS	ROLES	AGE	VERSION
k8s-master	Ready	master	1h	v1.10.0
k8s-node01	Ready	<none>	1h	v1.10.0

集群

命名空间

节点

持久化存储卷

角色

存储类

命名空间

default

概况

工作负载

定时任务

守护进程集

部署

任务

容器组

副本集

命名空间

名称	标签	状态	已创建
kube-public	-	Active	1 小时
default	-	Active	1 小时
kube-system	-	Active	1 小时

节点

名称	标签	已就绪	CPU 请求值 (核)	CPU 限制值 (核)	内存请求值 (字节)	内存限制值 (字节)	已创建
k8s-node01	beta.kubernetes.io/arch... beta.kubernetes.io/os: li... kubernetes.io/hostnam...	True	0 (0.00%)	0 (0.00%)	0 (0.00%)	0 (0.00%)	1 小时
k8s-master	beta.kubernetes.io/arch... beta.kubernetes.io/os: li... kubernetes.io/hostnam... node-role.kubernetes.io...	True	0.81 (81.00%)	0 (0.00%)	110 Mi (6.03%)	170 Mi (9.32%)	1 小时