

安装k8s操作流程记录

- 1 docker环境准备工作
 - 1.1 安装依赖包
 - 1.2 设置docker源
 - 1.3 docker安装版本查看
 - 1.4 安装docker
 - 1.5 镜像加速
- 2 安装K8S具体操作
 - 2.1 修改hostname文件并使立即生效
 - 2.2 修改hosts文件
 - 2.3 验证mac地址和UUID是否唯一
 - 2.4 内核参数修改
 - 2.5 设置kubernetes源
 - 2.6 Master节点安装
 - 2.7 下载镜像
 - 2.8 初始化Master
 - 2.9 加载环境变量
 - 2.10 安装pod网络
 - 2.11 删除Master污点标记
 - 2.12 Node节点安装
 - 2.13 防火墙
 - 2.14 安装Dashboard
 - 2.15 测试集群
 - 2.16 查看状态命令
 - 2.17 命令补全

安装k8s操作流程记录

参考文件地址

- [1集群](#)
- [15主备](#)
- [16高可用](#)

环境说明：

主机名	操作系统版本	IP	备注
master01	CentOS7.9	192.168.205.133	Master主机
work01	CentOS7.9	192.168.205.134	Node主机
work02	CentOS7.9	192.168.205.135	Node主机

安装软件说明：

软件名称	版本号	安装主机	备注
docker	19.03.9	Master+Node	容器
kubelet	1.19.4	Master+Node	主要服务
kubeadm	1.19.4	Master	启动集群及加入集群
kubectl	1.19.4	Master	命令行
kubernetes-dashboard	2.0.4	Master	UI操作面板

1 docker环境准备工作

1.1 安装依赖包

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

1.2 设置docker源

```
1 | yum-config-manager --add-repo  
https://download.docker.com/linux/centos/docker-ce.repo
```

1.3 docker安装版本查看

```
1 | yum list docker-ce --showduplicates | sort -r
```

1.4 安装docker

```
1 | yum install docker-ce
```

1.5 镜像加速

登录[地址](#): 找到镜像中心->镜像加速器, 查看相关配置

```
1 | 1. mkdir -p /etc/docker  
2 | 2. tee /etc/docker/daemon.json <<- 'EOF'  
3 | {  
4 |     "registry-mirrors": ["https://maxlibsu.mirror.aliyuncs.com"]  
5 | }  
6 | EOF  
7 | 3. systemctl daemon-reload  
8 | 4. systemctl restart docker
```

2 安装K8S具体操作

2.1 修改hostname文件并使立即生效

```
1 hostnamectl set-hostname master01
2 sysctl kernel.hostname=$(cat /etc/hostname)
```

2.2 修改hosts文件

```
1 cat >> /etc/hosts << EOF
2 192.168.205.133 master01
3 192.168.205.134 work01
4 192.168.205.135 work02
5 EOF
```

2.3 验证mac地址和UUID是否唯一

```
1 cat /sys/class/net/ens33/address
2 cat /sys/class/dmi/id/product_uuid
```

2.4 内核参数修改

```
1 cat << EOF > /etc/sysctl.d/k8s.conf
2 net.bridge.bridge-nf-call-ip6tables = 1
3 net.bridge.bridge-nf-call-iptables = 1
4 EOF
5 sysctl -p /etc/sysctl.d/k8s.conf
```

2.5 设置kubernetes源

```
1 cat << EOF > /etc/yum.repos.d/kubernetes.repo
2 [kubernetes]
3 name=Kubernetes
4 baseurl=https://mirrors.aliyun.com/kubernetes/yum/repos/kubernetes-el7-
  x86_64/
5 enabled=1
6 gpgcheck=1
7 repo_gpgcheck=1
8 gpgkey=https://mirrors.aliyun.com/kubernetes/yum/doc/yum-key.gpg
  https://mirrors.aliyun.com/kubernetes/yum/doc/rpm-package-key.gpg
9 EOF
10
11 yum clean all
12 yum -y makecache
```

[] 中括号中的是repository id，唯一，用来标识不同仓库

name 仓库名称，自定义

baseurl 仓库地址

enabled 是否启用该仓库，默认为1表示启用

gpgcheck 是否验证从该仓库获得程序包的合法性，1为验证

repo_gpgcheck 是否验证元数据的合法性，元数据就是程序包列表，1为验证

gpgkey 数字签名的公钥文件所在位置，如果gpgcheck值为1，此处就需要指定gpgkey文件的位置，如果gpgcheck值为0就不需要此项了

2.6 Master节点安装

```
1 yum install -y kubelet kubeadm kubectl
2
3 systemctl enable kubelet
4 systemctl start kubelet
```

kubelet 运行在集群所有节点上，用于启动Pod和容器等对象的工具

kubeadm 用于初始化集群，启动集群的命令工具

kubectl 用于和集群通信的命令行，通过kubectl可以部署和管理应用，查看各种资源，创建、删除和更新各种组件

2.7 下载镜像

```
1 镜像下载脚本image.sh
2
3 #!/bin/bash
4 url=registry.cn-hangzhou.aliyuncs.com/google_containers
5 version=v1.19.4
6 images=(`kubeadm config images list --kubernetes-version=$version|awk -F '/'
7 '{print $2}'`)
8 for imagename in ${images[@]}; do
9     docker pull $url/$imagename
10    docker tag $url/$imagename k8s.gcr.io/$imagename
11    docker rmi -f $url/$imagename
12 done
```

下载镜像

```
1 chmod u+x image.sh
2 ./image.sh
3 docker images
```

2.8 初始化Master

```
1 kubeadm init --apiserver-advertise-address 192.168.205.133 --pod-network-
2 cidr=10.244.0.0/16
3 # 得到类似如下信息，复制出来
4 kubeadm join 192.168.205.133:6443 --token wqczze.m4hqc5z001jq9v4e \
5     --discovery-token-ca-cert-hash
6     sha256:7dc10399821a534200c8a898818f7b960adc9ba8b228907abea37546d4542078
```

--apiserver-advertise-address: 指定Master的 interface

--pod-network-cidr: 指定pod网络的范围，这里使用 flannel 网络方案

2.9 加载环境变量

```
1 echo "export KUBECONFIG=/etc/kubernetes/admin.conf" >> ~/.bash_profile
2 source ~/.bash_profile
```

2.10 安装pod网络

[下载地址](#)

下载保存为kube-flannel.yml

使用 `kubectl apply -f kube-flannel.yml` 使生效

2.11 删除Master污点标记

```
1 # 查看污点
2 kubectl describe node master01 | grep -i taints
3 # 删除污点
4 kubectl taint nodes master01 node-role.kubernetes.io/master-
```

2.12 Node节点安装

前两步同Master节点前两部安装kubelet kubeadm kubectl并配置镜像

使用初始化Master中复制出来的信息，将Node节点加入集群，若上面token过期，Master节点上执行如下操作

```
1 # 查看是否过期
2 kubeadm token list
3 # 生成新的token令牌
4 kubeadm token create
5 # 生成新的加密串
6 openssl x509 -pubkey -in /etc/kubernetes/pki/ca.crt | openssl rsa -pubin -
  outform der 2>/dev/null | \
7 openssl dgst -sha256 -hex | sed 's/^.*://'
```

2.13 防火墙

开启端口有：2379(etcd客户端监听),2380(etcd节点间内部通讯),6443(kubernetes api端口),8443(LB VIP监听的服务),10250(kubelet服务监听),10251(controller manager),10252(scheduler),30000-32767/tcp (外部暴露),8472/udp (dashboard依赖端口, flannel通讯)

开启允许IP伪装：`firewall-cmd --add-masquerade --permanent`，重新加载使生效 `firewall-cmd --reload`

2.14 安装Dashboard

[下载地址](#)

命名为 `kubernetes-dashboard.yaml` 修改相关配置，并配置管理集群账号

```
1 kind: Service
2 apiVersion: v1
3 metadata:
4   labels:
5     k8s-app: kubernetes-dashboard
6   name: kubernetes-dashboard
7   namespace: kubernetes-dashboard
8 spec:
9   type: NodePort # 类型为NodePort，为下面暴露端口准备
10  ports:
11    - port: 443
12      targetPort: 8443
13      nodePort: 30001 # 此处为增加暴露端口
```

```

14     selector:
15         k8s-app: kubernetes-dashboard
16
17     # ----- 下面为新增内容 -----
18     ---
19
20     # ----- dashboard-admin ----- #
21     apiVersion: v1
22     kind: ServiceAccount
23     metadata:
24         name: dashboard-admin
25         namespace: kube-system
26
27     ---
28     apiVersion: rbac.authorization.k8s.io/v1
29     kind: ClusterRoleBinding
30     metadata:
31         name: dashboard-admin
32     subjects:
33     - kind: ServiceAccount
34       name: dashboard-admin
35       namespace: kube-system
36     roleRef:
37         apiGroup: rbac.authorization.k8s.io
38         kind: ClusterRole
39         name: cluster-admin

```

使生效 `kubectl apply -f kubernetes-dashboard.yaml`

查看相关状态

```

1 kubectl get deployment kubernetes-dashboard -n kubernetes-dashboard
2 kubectl get pods -n kubernetes-dashboard -o wide
3 kubectl get services -n kubernetes-dashboard

```

查看令牌

```

1 kubectl describe secrets -n kubernetes-dashboard dashboard-admin

```

使用浏览器[访问](#)，输入上一步获取的令牌进入查看

2.15 测试集群

命令方式

```

1 kubectl run httpd-app --image=httpd --replicas=3

```

配置方式

新建nginx.yaml并填入如下内容

```

1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4     name: nginx-demo
5 spec:

```

```

6   replicas: 3
7   selector:
8     matchLabels:
9       app: nginx
10  template:
11    metadata:
12      labels:
13        app: nginx
14    spec:
15      containers:
16      - name: nginx
17        image: nginx
18        ports:
19        - containerPort: 80
20
21  ---
22
23  apiVersion: v1
24  kind: Service
25  metadata:
26    name: nginx-demo
27  spec:
28    type: NodePort
29    ports:
30    - name: nginx
31      nodePort: 30002
32      port: 80
33      protocol: TCP
34      targetPort: 80
35  selector:
36    app: nginx

```

部署上述服务 `kubectl apply -f nginx.yaml`

2.16 查看状态命令

```

1  # 查看节点状态
2  kubectl get nodes
3  # 查看pod状态
4  kubectl get pod --all-namespaces
5  # 查看副本数
6  kubectl get deployments
7  kubectl get pod -o wide
8  # 查看deployment详细信息
9  kubectl describe deployments
10 # 查看集群状态
11 kubectl get cs

```

在查看集群状态时发现 `scheduler` 和 `controller-manager` 状态错误，修改 `/etc/kubernetes/manifests` 路径下的 `kube-scheduler.yaml` 和 `kube-controller-manager.yaml` 找到 `--port=0` 并注释该行 `# --port=0`，后重启kubelet即可 `systemctl restart kubelet`

2.17 命令补全

安装 `bash-completion`

```
1 # 安装
2 yum install -y bash-completion
3 # 加载
4 source /etc/profile.d/bash_completion.sh
5 # kubectl命令补全
6 echo "source <(kubectl completion bash)" >> ~/.bash_profile
7 source ~/.bash_profile
```