

## 2 第二章 集群环境搭建(kubeadm 方式)

### 2.1 前置知识点

目前生产部署Kubernetes 集群主要有两种方式：

#### kubeadm

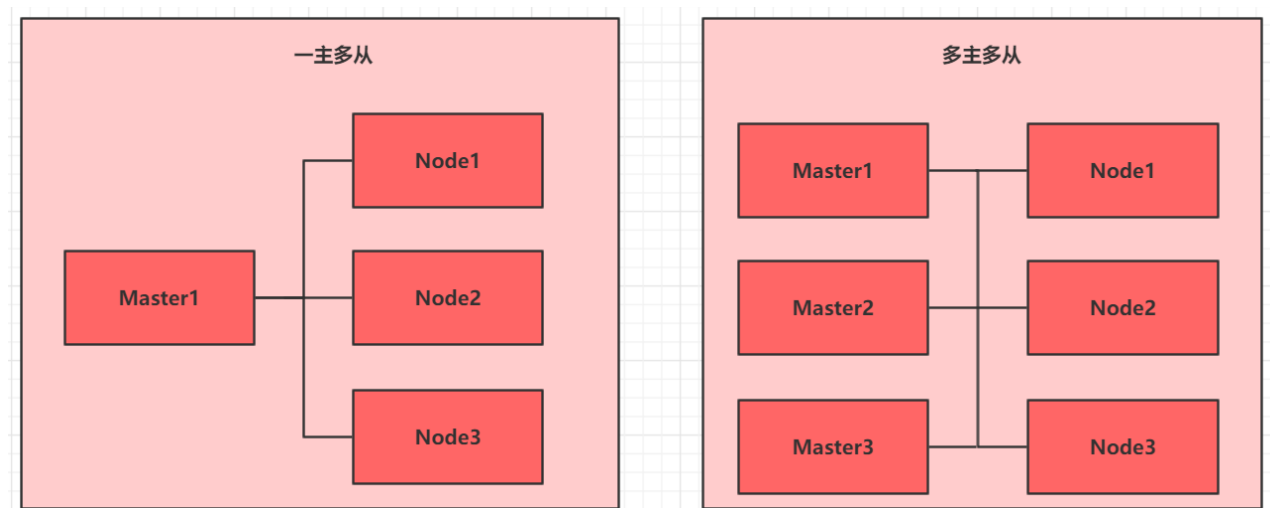
Kubeadm 是一个K8s 部署工具，提供kubeadm init 和kubeadm join，用于快速部署Kubernetes 集群。

官方地址：<https://kubernetes.io/docs/reference/setup-tools/kubeadm/kubeadm/>

#### 二进制包

从github 下载发行版的二进制包，手动部署每个组件，组成Kubernetes 集群。

Kubeadm 降低部署门槛，但屏蔽了很多细节，遇到问题很难排查。如果想更容易可控，推荐使用二进制包部署Kubernetes 集群，虽然手动部署麻烦点，期间可以学习很多工作原理，也利于后期维护。



### 2.2 kubeadm 部署方式介绍

kubeadm 是官方社区推出的一个用于快速部署kubernetes 集群的工具，这个工具能通过两条指令完成一个kubernetes 集群的部署：

- 创建一个Master 节点kubeadm init
- 将Node 节点加入到当前集群中\$ kubeadm join <Master 节点的IP 和端口>

### 2.3 安装要求

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

## 2.4 最终目标

- 在所有节点上安装Docker 和kubeadm
- 部署Kubernetes Master
- 部署容器网络插件
- 部署Kubernetes Node，将节点加入Kubernetes 集群中
- 部署Dashboard Web 页面，可视化查看Kubernetes 资源

## 2.5 准备环境

角色	IP地址	组件
k8s-master01	192.168.5.3	docker, kubectl, kubeadm, kubelet
k8s-node01	192.168.5.4	docker, kubectl, kubeadm, kubelet
k8s-node02	192.168.5.5	docker, kubectl, kubeadm, kubelet

## 2.6 系统初始化

### 2.6.1 设置系统主机名以及 Host 文件的相互解析

```
1 hostnamectl set-hostname k8s-master01 && bash
2 hostnamectl set-hostname k8s-node01 && bash
3 hostnamectl set-hostname k8s-node02 && bash
```

```
1 cat <<EOF>> /etc/hosts
2 192.168.5.3      k8s-master01
3 192.168.5.4      k8s-node01
4 192.168.5.5      k8s-node02
5 EOF
```

```
1 scp /etc/hosts root@192.168.5.4:/etc/hosts
2 scp /etc/hosts root@192.168.5.5:/etc/hosts
```

## 2.6.2 安装依赖文件（所有节点都要操作）

```
1 yum install -y conntrack ntpdate ntp ipvsadm ipset jq iptables curl sysstat  
libseccomp wget vim net-tools git
```

## 2.6.3 设置防火墙为 Iptables 并设置空规则（所有节点都要操作）

```
1 systemctl stop firewalld && systemctl disable firewalld  
2 yum -y install iptables-services && systemctl start iptables && systemctl  
enable iptables && iptables -F && service iptables save
```

## 2.6.4 关闭 SELINUX（所有节点都要操作）

```
1 swapoff -a && sed -i '/ swap / s/^\(.*\)$/#\1/g' /etc/fstab  
2 setenforce 0 && sed -i 's/^SELINUX=.*SELINUX=disabled/' /etc/selinux/config
```

## 2.6.5 调整内核参数，对于 K8S（所有节点都要操作）

```
1 modprobe br_netfilter  
2 cat <<EOF> kubernetes.conf  
3 net.bridge.bridge-nf-call-iptables=1  
4 net.bridge.bridge-nf-call-ip6tables=1  
5 net.ipv4.ip_forward=1  
6 net.ipv4.tcp_tw_recycle=0  
7 vm.swappiness=0 # 禁止使用 swap 空间，只有当系统 OOM 时才允许使用它  
8 vm.overcommit_memory=1 # 不检查物理内存是否够用  
9 vm.panic_on_oom=0 # 开启 OOM  
10 fs.inotify.max_user_instances=8192  
11 fs.inotify.max_user_watches=1048576  
12 fs.file-max=52706963  
13 fs.nr_open=52706963  
14 net.ipv6.conf.all.disable_ipv6=1  
15 net.netfilter.nf_conntrack_max=2310720  
16
```

```
17 EOF
18 cp kubernetes.conf /etc/sysctl.d/kubernetes.conf
19 sysctl -p /etc/sysctl.d/kubernetes.conf
```

## 2.6.6 调整系统时区（所有节点都要操作）

```
1 # 设置系统时区为 中国/上海
2 timedatectl set-timezone Asia/Shanghai
3 # 将当前的 UTC 时间写入硬件时钟
4 timedatectl set-local-rtc 0
5 # 重启依赖于系统时间的服务
6 systemctl restart rsyslog
7 systemctl restart crond
```

## 2.6.7 设置 rsyslogd 和 systemd journald（所有节点都要操作）

```
1 # 持久化保存日志的目录
2 mkdir /var/log/journal
3 mkdir /etc/systemd/journal.conf.d
4 cat > /etc/systemd/journal.conf.d/99-prophet.conf <<EOF
5 [Journal]
6 # 持久化保存到磁盘
7 Storage=persistent
8 # 压缩历史日志
9 Compress=yes
10 SyncIntervalSec=5m
11 RateLimitInterval=30s
12 RateLimitBurst=1000
13 # 最大占用空间 10G
14 SystemMaxUse=10G
15 # 单日志文件最大 200M
16 SystemMaxFileSize=200M
17 # 日志保存时间 2 周
18 MaxRetentionSec=2week
19 # 不将日志转发到 syslog
20 ForwardToSyslog=no
21 EOF
```

## 2.6.8 kube-proxy开启ipvs的前置条件（所有节点都要操作）

```
1 cat <<EOF> /etc/sysconfig/modules/ipvs.modules
2 #!/bin/bash
3 modprobe -- ip_vs
4 modprobe -- ip_vs_rr
5 modprobe -- ip_vs_wrr
6 modprobe -- ip_vs_sh
7 modprobe -- nf_conntrack_ipv4
8
9 EOF
10
11 chmod 755 /etc/sysconfig/modules/ipvs.modules && bash
   /etc/sysconfig/modules/ipvs.modules && lsmod | grep -e ip_vs -e
   nf_conntrack_ipv4
```

## 2.6.9 安装 Docker 软件（所有节点都要操作）

```
1 yum install -y yum-utils device-mapper-persistent-data lvm2
2
3
4 yum-config-manager --add-repo http://mirrors.aliyun.com/docker-
   ce/linux/centos/docker-ce.repo
5
6 yum install -y docker-ce
7 ## 创建 /etc/docker 目录
8 mkdir /etc/docker
9
10 cat > /etc/docker/daemon.json <<EOF
11 {
12     "exec-opts": ["native.cgroupdriver=systemd"],
13     "log-driver": "json-file",
14     "log-opts": {"max-size": "100m"}
15 }
16 EOF
17 mkdir -p /etc/systemd/system/docker.service.d
```

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

## 2.6.10 安装 Kubeadm （所有节点都要操作）

```
1 cat <<EOF > /etc/yum.repos.d/kubernetes.repo
2 [kubernetes]
3 name=Kubernetes
4 baseurl=http://mirrors.aliyun.com/kubernetes/yum/repos/kubernetes-el7-x86_64
5 enabled=1
6 gpgcheck=0
7 repo_gpgcheck=0
8 gpgkey=http://mirrors.aliyun.com/kubernetes/yum/doc/yum-key.gpg
9 http://mirrors.aliyun.com/kubernetes/yum/doc/rpm-package-key.gpg
10 EOF
11 yum install -y kubelet kubeadm kubectl && systemctl enable kubelet
```

## 2.7 部署Kubernetes Master

### 2.7.1 初始化主节点（主节点操作）

```
1 kubeadm init --apiserver-advertise-address=192.168.5.3 --image-repository
  registry.aliyuncs.com/google_containers --kubernetes-version v1.21.1 --service-
  cidr=10.96.0.0/12 --pod-network-cidr=10.244.0.0/16
2 mkdir -p $HOME/.kube
3 sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
4 sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

### 2.7.2 加入主节点以及其余工作节点

```
1 kubeadm join 192.168.5.3:6443 --token h0uelc.l46qp29nxscke7f7 \
2 --discovery-token-ca-cert-hash
  sha256:abc807778e24bff73362ceeb783cc7f6feec96f20b4fd707c3f8e8312294e28f
```

### 2.7.3 部署网络

```
1 kubectl apply -f
  https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-
  flannel.yml
```

## 2.8 测试kubernetes 集群

### 2.8.1 部署nginx 测试

```
1 kubectl create deployment nginx --image=nginx
2 kubectl expose deployment nginx --port=80 --type=NodePort
3 kubectl get pod,svc
```