部署k8s-1.28-containerd

前提条件

确认机器已清空，之前reset的要彻底把相关数据清理。

确认网络没问题。

每个节点的操作

1.主机配置

1.1.主机名配置

hostnamectl set-hostname master

hostnamectl set-hostname worker

1.2.主机IP地址配置

确认为静态IP地址，不可为动态DHCP，若变化则集群出现异常。

1.3.主机名与IP地址解析配置

vim /etc/hosts

x.x.x.x master

x.x.x.x worker

1.4.防火墙配置

systemctl stop firewalld && systemctl disable firewalld

firewall-cmd --state

确认系统返回 **not running**。

1.5.SELINUX配置

setenforce 0 （临时生效）

sed -ri 's/SELINUX=enforcing/SELINUX=disabled/' /etc/selinux/config (重启后才生效)

sestatus

1.6.时间同步配置

ntpdate、chronyd随意，搭配crond定时任务保证时间同步。

ubuntu系统：

sudo apt update

sudo apt install ntp

sudo vim /etc/ntp.conf

```
server 0.ubuntu.pool.ntp.org iburst server 1.ubuntu.pool.ntp.org iburst
server 2.ubuntu.pool.ntp.org iburst server 3.ubuntu.pool.ntp.org iburst
```

sudo systemctl restart ntp.service

timedatectl status

*1.7.升级操作系统内核

1.8.内核转发配置及网桥过滤

cat <<EOF > /etc/sysctl.d/k8s.conf

net.bridge.bridge-nf-call-ip6tables=1

net.bridge.bridge-nf-call-iptables=1

net.ipv4.ip_forward=1

vm.swappiness=0

EOF

sysctl -p /etc/sysctl.d/k8s.conf

*如果报错为缺少br_netfilter模块，则需要加载该模块。

1.9.安装ipset及ipvsadm

说明：所有主机均需要操作。

安装ipset及ipvsadm：

yum -y install ipset ipvsadm

ipvsadm -ln 查看ipvs状态、版本

配置ipvsadm模块加载方式

添加需要加载的模块

cat > /etc/sysconfig/modules/ipvs.modules <

#!/bin/bash

modprobe -- ip_vs

modprobe -- ip_vs_rr

modprobe -- ip_vs_wrr

modprobe -- ip_vs_sh

modprobe -- nf_conntrack

EOF

chmod 755 /etc/sysconfig/modules/ipvs.modules && bash /etc/sysconfig/modules/ipvs.modules && lsmod | grep -e ip_vs -e nf_conntrack

1.10.关闭SWAP分区

swapoff -a

sed -i "/swap/s/UUID/#UUID/g" /etc/fstab

2.容器运行时containerd准备

2.1.containerd准备

以containerd 1.7.3为例

每个节点执行：

wget https://github.com/containerd/containerd/releases/download/v1.7.3/cri-containerd-1.7.3-linux-amd64.tar.gz

tar xf cri-containerd-1.7.3-linux-amd64.tar.gz -C /

containerd --version确认版本，现在1.7包出来也是bundle到1.6版本

mkdir /etc/containerd

cd /etc/containerd

containerd config default > config.toml

vim config.toml：

1. sandbox image pause版本改为3.9
2. SystemdCgroup = true

```
[plugins."io.containerd.grpc.v1.cri".containerd.runtimes.runc]
  base_runtime_spec = ""
  cni_conf_dir = ""
  cni_max_conf_num = 0
  container_annotations = []
  pod_annotations = []
  privileged_without_host_devices = false
  runtime_engine = ""
  runtime_path = ""
  runtime_root = ""
  runtime_type = "io.containerd.runc.v2"

  [plugins."io.containerd.grpc.v1.cri".containerd.runtimes.runc.options]
    BinaryName = ""
    CriuImagePath = ""
    CriuPath = ""
    CriuWorkPath = ""
    IoGid = 0
    IoUid = 0
    NoNewKeyring = false
    NoPivotRoot = false
    Root = ""
    ShimCgroup = ""
    SystemdCgroup = true
```

systemctl enable containerd --now && systemctl restart containerd

systemctl status containerd 确认运行状态

2.2.runc准备

runc -v 命令查看版本，没有则安装（一般都会有），基于limseccomp

3.K8s集群部署

3.1.yum源准备

用阿里云源

3.2.kube老三样安装

3.2.1.yum安装

yum install -y kubeadm-1.28.2-0 kubelet-1.28.2-0 kubectl-1.28.2-0

3.2.2.kubelet配置

为了实现docker使用的cgroupdriver与kubelet使用的cgroup一致，修改kubelet配置。

配置成如下

vim /etc/sysconfig/kubelet

KUBELET_EXTRA_ARGS="--cgroup-driver=systemd"

systemctl enable kubelet --now

master节点操作

3.3.K8s集群初始化

ctr -n k8s.io i tag registry.aliyuncs.com/google_containers/pause:3.6 registry.k8s.io/pause:3.9

kubeadm init --image-repository registry.aliyuncs.com/google_containers --kubernetes-version=v1.28.2 --pod-network-cidr=10.244.0.0/16 --apiserver-advertise-address=10.180.224.46 --cri-socket unix:///var/run/containerd/containerd.sock

> **说明**：--apiserver-advertise-address为你的机器IP地址。
> --cri-socket string：Path to the CRI socket to connect. If empty kubeadm will try to auto-detect this value; use this option only if you have more than one CRI installed or if you have non-standard CRI socket.
> 根据提示操作

4.网络插件calico部署

1. kubectl create -f https://raw.githubusercontent.com/projectcalico/calico/v3.26.1/manifests/tigera-operator.yaml

2. wget https://raw.githubusercontent.com/projectcalico/calico/v3.26.1/manifests/custom-resources.yaml

3. custom-resources.yaml的IP地址段改成10.244.0.0/16。

4. kubectl create -f custom-resources.yaml

5.初始化

kubeadm init --image-repository registry.aliyuncs.com/google_containers --kubernetes-version=v1.28.2 --pod-network-cidr=10.244.0.0/16 --apiserver-advertise-address=10.180.224.46 --cri-socket unix:///var/run/containerd/containerd.sock

根据初始化提示操作，mkdir那些。

Worker节点操作

执行join命令

master节点操作

查看集群节点状态，确认为ready，要等5分钟

kubectl get po -n calico-system

kubectl get po -n kube-system

kubectl get nodes -owide