

[一、下载 RockyLinux9 镜像](#header-n4)  
 [二、环境初始化](#header-n8)  
 [三、部署网络插件](#header-n25)  
 [固定网卡(可选)](#header-n28)  
 [四、修改kube-proxy 模式为 ipvs](#header-n34)

## 一、下载 RockyLinux9 镜像

# 官方下载地址  
https://rockylinux.org/download  
  
# 阿里云镜像下载地址  
https://mirrors.aliyun.com/rockylinux/9/isos/x86\_64/?spm=a2c6h.25603864.0.0.29696621VzJej5

## 二、环境初始化

# 网卡配置  
# cat /etc/NetworkManager/system-connections/ens160.nmconnection  
[ipv4]  
method=manual  
address1=192.168.66.12/24,192.168.66.200  
dns=114.114.114.114;8.8.8.8  
# cat /etc/NetworkManager/system-connections/ens192.nmconnection  
[connection]  
autoconnect=false  
  
# 调用 nmcli 重启设备和连接配置  
nmcli d d ens192  
nmcli d r ens160   
nmcli c r ens160

# Rocky 系统软件源更换  
sed -e 's|^mirrorlist=|#mirrorlist=|g' \  
 -e 's|^#baseurl=http://dl.rockylinux.org/$contentdir|baseurl=https://mirrors.aliyun.com/rockylinux|g' \  
 -i.bak \  
 /etc/yum.repos.d/[Rr]ocky\*.repo  
   
dnf makecache

# 防火墙修改 firewalld 为 iptables  
systemctl stop firewalld  
systemctl disable firewalld  
  
yum -y install iptables-services  
systemctl start iptables  
iptables -F  
systemctl enable iptables  
service iptables save

# 禁用 Selinux  
setenforce 0  
sed -i "s/SELINUX=enforcing/SELINUX=disabled/g" /etc/selinux/config  
grubby --update-kernel ALL --args selinux=0  
# 查看是否禁用，grubby --info DEFAULT  
# 回滚内核层禁用操作，grubby --update-kernel ALL --remove-args selinux

# 设置时区  
timedatectl set-timezone Asia/Shanghai

# 关闭 swap 分区  
swapoff -a  
sed -i 's:/dev/mapper/rl-swap:#/dev/mapper/rl-swap:g' /etc/fstab  
  
# 修改主机名  
hostnamectl set-hostname k8s-node01

# 安装 ipvs  
yum install -y ipvsadm

# 开启路由转发  
echo 'net.ipv4.ip\_forward=1' >> /etc/sysctl.conf  
sysctl -p

# 加载 bridge  
yum install -y epel-release  
yum install -y bridge-utils  
  
modprobe br\_netfilter  
echo 'br\_netfilter' >> /etc/modules-load.d/bridge.conf  
echo 'net.bridge.bridge-nf-call-iptables=1' >> /etc/sysctl.conf  
echo 'net.bridge.bridge-nf-call-ip6tables=1' >> /etc/sysctl.conf  
sysctl -p

# 添加 docker-ce yum 源  
# 中科大(ustc)  
sudo dnf config-manager --add-repo https://mirrors.ustc.edu.cn/docker-ce/linux/centos/docker-ce.repo  
cd /etc/yum.repos.d  
# 切换中科大源  
sed -e 's|download.docker.com|mirrors.ustc.edu.cn/docker-ce|g' docker-ce.repo  
  
# 安装 docker-ce  
yum -y install docker-ce  
  
# 配置 daemon.  
cat > /etc/docker/daemon.json <<EOF  
{  
 "data-root": "/data/docker",  
 "exec-opts": ["native.cgroupdriver=systemd"],  
 "log-driver": "json-file",  
 "log-opts": {  
 "max-size": "100m",  
 "max-file": "100"  
 },  
 "insecure-registries": ["harbor.xinxainghf.com"],  
 "registry-mirrors": ["https://kfp63jaj.mirror.aliyuncs.com"]  
}  
EOF  
mkdir -p /etc/systemd/system/docker.service.d  
  
# 重启docker服务  
systemctl daemon-reload && systemctl restart docker && systemctl enable docker  
  
reboot

# 安装 cri-docker  
wget https://github.com/Mirantis/cri-dockerd/releases/download/v0.3.9/cri-dockerd-0.3.9.amd64.tgz  
tar -xf cri-dockerd-0.3.9.amd64.tgz  
cp cri-dockerd/cri-dockerd /usr/bin/  
chmod +x /usr/bin/cri-dockerd  
  
# 配置 cri-docker 服务  
cat <<"EOF" > /usr/lib/systemd/system/cri-docker.service  
[Unit]  
Description=CRI Interface for Docker Application Container Engine  
Documentation=https://docs.mirantis.com  
After=network-online.target firewalld.service docker.service  
Wants=network-online.target  
Requires=cri-docker.socket  
[Service]  
Type=notify  
ExecStart=/usr/bin/cri-dockerd --network-plugin=cni --pod-infra-container-image=registry.aliyuncs.com/google\_containers/pause:3.8  
ExecReload=/bin/kill -s HUP $MAINPID  
TimeoutSec=0  
RestartSec=2  
Restart=always  
StartLimitBurst=3  
StartLimitInterval=60s  
LimitNOFILE=infinity  
LimitNPROC=infinity  
LimitCORE=infinity  
TasksMax=infinity  
Delegate=yes  
KillMode=process  
[Install]  
WantedBy=multi-user.target  
EOF  
  
# 添加 cri-docker 套接字  
cat <<"EOF" > /usr/lib/systemd/system/cri-docker.socket  
[Unit]  
Description=CRI Docker Socket for the API  
PartOf=cri-docker.service  
[Socket]  
ListenStream=%t/cri-dockerd.sock  
SocketMode=0660  
SocketUser=root  
SocketGroup=docker  
[Install]  
WantedBy=sockets.target  
EOF  
  
# 启动 cri-docker 对应服务  
systemctl daemon-reload  
systemctl enable cri-docker  
systemctl start cri-docker  
systemctl is-active cri-docker

# 添加 kubeadm yum 源  
cat <<EOF > /etc/yum.repos.d/kubernetes.repo  
[kubernetes]  
name=Kubernetes  
baseurl=https://pkgs.k8s.io/core:/stable:/v1.29/rpm/  
enabled=1  
gpgcheck=1  
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.29/rpm/repodata/repomd.xml.key  
exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni  
EOF  
  
# 安装 kubeadm 1.29 版本  
yum install -y kubelet-1.29.0 kubectl-1.29.0 kubeadm-1.29.0  
systemctl enable kubelet.service

# 初始化主节点  
kubeadm init --apiserver-advertise-address=192.168.66.11 --image-repository registry.aliyuncs.com/google\_containers --kubernetes-version 1.29.2 --service-cidr=10.10.0.0/12 --pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=all --cri-socket unix:///var/run/cri-dockerd.sock

# work token 过期后，重新申请  
kubeadm token create --print-join-command

# worker 加入  
kubeadm join 192.168.10.11:6443 --token a6xh07.yg9wh2vru2grluwb --discovery-token-ca-cert-hash sha256:7cd8499abae48c8403800152cc0f655ac704ea00ae30a549acd9bbac7b26dca4 --cri-socket unix:///var/run/cri-dockerd.sock

## 三、部署网络插件

https://docs.tigera.io/calico/latest/getting-started/kubernetes/self-managed-onprem/onpremises#install-calico-with-kubernetes-api-datastore-more-than-50-nodes  
  
curl https://raw.githubusercontent.com/projectcalico/calico/v3.26.3/manifests/calico-typha.yaml -o calico.yaml  
 CALICO\_IPV4POOL\_CIDR 指定为 pod 地址  
   
# 修改为 BGP 模式  
# Enable IPIP  
- name: CALICO\_IPV4POOL\_IPIP  
 value: "Always" #改成Off

#### 固定网卡(可选)

# 目标 IP 或域名可达  
 - name: calico-node  
 image: registry.geoway.com/calico/node:v3.19.1  
 env:  
 # Auto-detect the BGP IP address.  
 - name: IP  
 value: "autodetect"  
 - name: IP\_AUTODETECTION\_METHOD  
 value: "can-reach=www.google.com"  
kubectl set env daemonset/calico-node -n kube-system IP\_AUTODETECTION\_METHOD=can-reach=www.google.com

# 匹配目标网卡  
- name: calico-node  
 image: registry.geoway.com/calico/node:v3.19.1  
 env:  
 # Auto-detect the BGP IP address.  
 - name: IP  
 value: "autodetect"  
 - name: IP\_AUTODETECTION\_METHOD  
 value: "interface=eth.\*"

# 排除匹配网卡  
- name: calico-node  
 image: registry.geoway.com/calico/node:v3.19.1  
 env:  
 # Auto-detect the BGP IP address.  
 - name: IP  
 value: "autodetect"  
 - name: IP\_AUTODETECTION\_METHOD  
 value: "skip-interface=eth.\*"

# CIDR  
- name: calico-node  
 image: registry.geoway.com/calico/node:v3.19.1  
 env:  
 # Auto-detect the BGP IP address.  
 - name: IP  
 value: "autodetect"  
 - name: IP\_AUTODETECTION\_METHOD  
 value: "cidr=192.168.200.0/24,172.15.0.0/24"

## 四、修改kube-proxy 模式为 ipvs

# kubectl edit configmap kube-proxy -n kube-system  
mode: ipvs  
  
kubectl delete pod -n kube-system -l k8s-app=kube-proxy