**Kubernetes基础集群-01**

Wrote by JunXiTang

**一、角色划分以及地址分配**

|  |
| --- |
| Bash 192.168.135.151 k8-master-01 192.168.135.152 k8-slave-01 |

**二、配置网络环境（运行kubernetes的主机）**

|  |
| --- |
| Bash sudo tee /etc/modules-load.d/k8s.conf <<EOF overlay br\_netfilter EOF |

|  |
| --- |
| Bash sudo modprobe overlay sudo modprobe br\_netfilter |

|  |
| --- |
| Bash sudo tee /etc/sysctl.d/k8s.conf <<EOF net.bridge.bridge-nf-call-ip6tables = 1 net.bridge.bridge-nf-call-iptables = 1 net.ipv4.ip\_forward = 1 EOF |

|  |
| --- |
| Bash sudo sysctl --system |

|  |
| --- |
| Bash sysctl net.bridge.bridge-nf-call-iptables net.bridge.bridge-nf-call-ip6tables net.ipv4.ip\_forward |

**三、关闭分区**

|  |
| --- |
| Bash sudo nano /etc/fstab #nano #Ctrl+O 保存 #Ctrl+X 退出 #一般系统有两种挂载磁盘方式 注释掉  #UUID=xxxx-xxxx-xxxx-xxxx swap swap defaults 0 0  或者   #/swapfile none swap sw 0 0 #非LVM安装ubuntu  #linux-desktop  sed -i 's@/swap@#/swap@g' /etc/fstab  #linux-server  sed -i 's@/swap.img@#/swap.img@g' /etc/fstab #切勿注释掉系统分区 #否则无法正常开机 sudo swapoff -a #重启系统以确保更改持久生效。 #请注意，在编辑/etc/fstab文件时要小心，因为不正确的更改可能会导致系统无法启动。如果你不确定，请先在非生产系统上尝试这些步骤。 |

**四、安装containerd（运行kubernetes的主机）**

**1.下载containerd**

|  |
| --- |
| Bash #使用Debain原生 sudo apt update  sudo apt install -y containerd |

**2.修改containerd默认配置**

|  |
| --- |
| Bash sudo mkdir /etc/containerd sudo touch /etc/containerd/config.toml  #把containerd默认配置导入config.toml文件 containerd config default > /etc/containerd/config.toml |

|  |
| --- |
| Bash #修改配置 nano /etc/containerd/config.toml  #修改镜像源 把sandbox\_image = "registry.k8s.io/pause:3.8" 修改为sandbox\_image = "registry.aliyuncs.com/google\_containers/pause:3.9"   把SystemdCgroup设置为true SystemdCgroup = true |

|  |
| --- |
| Bash #直接使用指令 sed -i 's@registry.k8s.io/pause:3.8@registry.aliyuncs.com/google\_containers/pause:3.9@g' /etc/containerd/config.toml sed -i 's@SystemdCgroup = false@SystemdCgroup = true@g' /etc/containerd/config.toml |

|  |
| --- |
| Bash #对于containerd修改/etc/containerd/config.toml文件 #找到 [plugins."io.containerd.grpc.v1.cri".registry.configs] #和 [plugins."io.containerd.grpc.v1.cri".registry.mirrors] #在下面添加配置 nano /etc/containerd/config.toml #忽略证书问题 [plugins."io.containerd.grpc.v1.cri".registry.configs]  [plugins."io.containerd.grpc.v1.cri".registry.configs."registry.docker.io:5000".tls]  insecure\_skip\_verify = true  #**insecure\_skip\_verify**是一个配置选项，用于在TLS连接中忽略证书验证。  #当设置为true时，可以忽略SSL证书的验证，这通常用于测试环境或与自签名证书的服务器通信。     #添加镜像地址 [plugins."io.containerd.grpc.v1.cri".registry.mirrors]  [plugins."io.containerd.grpc.v1.cri".registry.mirrors."docker.io"]  endpoint = ["https://docker.ketches.cn"]  [plugins."io.containerd.grpc.v1.cri".registry.mirrors."registry.docker.io:5000"]  endpoint = ["http://registry.docker.io:5000"] |

**3.保存重启containerd**

|  |
| --- |
| Bash #重载daemon sudo systemctl daemon-reload sudo systemctl restart containerd  #验证版本信息 sudo containerd --veriosn |

**五、安装kubernetes（运行kubernetes的主机）**

**1.更新 apt 包索引，并安装使用 Kubernetes apt 仓库所需要的包：**

|  |
| --- |
| Bash sudo apt-get update *# apt-transport-https 可以是一个虚拟包；如果是这样，你可以跳过这个包* sudo apt-get install -y apt-transport-https ca-certificates curl gnupg |

**2.下载 Kubernetes 软件包仓库的公共签名密钥。 同一个签名密钥适用于所有仓库，因此你可以忽略 URL 中的版本信息：**

|  |
| --- |
| Bash *# 如果 `/etc/apt/keyrings` 目录不存在，则应在 curl 命令之前创建它，请阅读下面的注释。* *sudo mkdir -p -m 755 /etc/apt/keyrings* curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28.2/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg sudo chmod 644 /etc/apt/keyrings/kubernetes-apt-keyring.gpg *# allow unprivileged APT programs to read this keyring* |

**3.添加合适的 Kubernetes apt 仓库。如果你想用 v1.32 之外的 Kubernetes 版本， 请将下面命令中的 v1.32 替换为所需的次要版本：**

|  |
| --- |
| Bash *# 这会覆盖 /etc/apt/sources.list.d/kubernetes.list 中的所有现存配置* # 建议使用阿里云镜像 sudo curl -fsSL https://mirrors.aliyun.com/kubernetes/apt/doc/apt-key.gpg |  sudo apt-key add sudo echo "deb https://mirrors.aliyun.com/kubernetes/apt/ kubernetes-xenial main" |  sudo tee /etc/apt/sources.list.d/kubernetes.list |

**4.要将 kubectl 升级到别的次要版本，你需要先升级 /etc/apt/sources.list.d/kubernetes.list 中的版本， 再运行 apt-get update 和 apt-get upgrade 命令。 更详细的步骤可以在**[**更改 Kubernetes 软件包存储库**](https://kubernetes.io/zh-cn/docs/tasks/administer-cluster/kubeadm/change-package-repository/)**中找到。**

更新 apt 包索引，然后安装 kubectl、kubeadm、kubelet：

|  |
| --- |
| Bash sudo apt-get update sudo apt-get install -y kubectl kubeadm kubelet sudo apt-mark hold kubelet kubeadm kubectl #固定版本防止自动升级 |

**5.配置Kubernetes初始化文件（从节点无需操作）**

|  |
| --- |
| Bash #导出默认配置 sudo mkdir /etc/kubernetes sudo touch /etc/kubernetes/kubeadm-init.yaml kubeadm config print init-defaults > /etc/kubernetes/kubeadm-init.yaml |

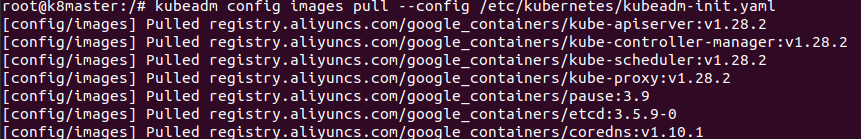
|  |
| --- |
| Bash #目前建议手动 #修改节点名字 nano /etc/kubernetes/kubeadm-init.yaml  #slave节点就改为对应的节点名 name: k8-master-01 #修改版本 kubernetesVersion: 1.28.2 #修改IP advertiseAddress： 192.168.135.136 #your host #修改镜像源 #把镜像修改为： imageRepository: registry.aliyuncs.com/google\_containers #添加podSubnet: podSubnet: 10.244.0.0/16 |

|  |
| --- |
| Bash #使用指令 #k8-master-01 sed -i 's@name: node@name: k8-master-01@g' /etc/kubernetes/kubeadm-init.yaml sed -i 's@advertiseAddress: 1.2.3.4@advertiseAddress: 192.168.135.151@g' /etc/kubernetes/kubeadm-init.yaml sed -i 's@kubernetesVersion: 1.28.0@kubernetesVersion: 1.28.2@g' /etc/kubernetes/kubeadm-init.yaml sed -i 's@imageRepository: registry.k8s.io@imageRepository: registry.aliyuncs.com/google\_containers@g' /etc/kubernetes/kubeadm-init.yaml sed -i 's@serviceSubnet: 10.96.0.0/12@serviceSubnet: 10.96.0.0/12\n podSubnet: 10.244.0.0/16@g' /etc/kubernetes/kubeadm-init.yaml |

**六、初始化kubernetes集群：**

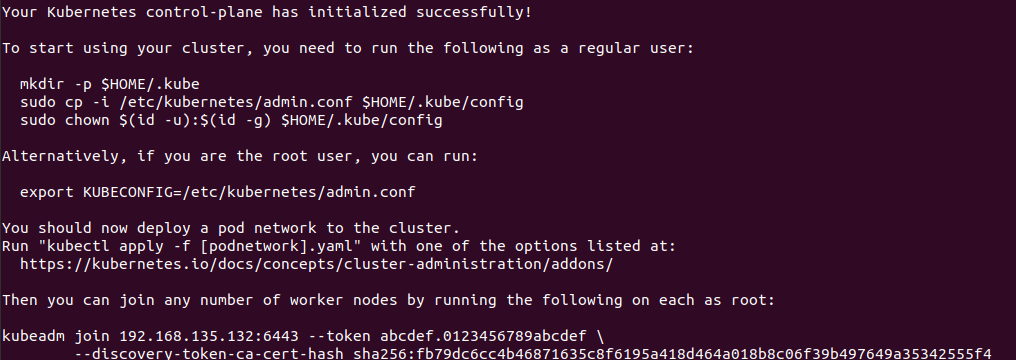
**1.拉取镜像（用阿里云的镜像，也可以按照第八步提前下载镜像然后上传到自己的私有仓库）**

|  |
| --- |
| Bash  sudo kubeadm config images list --config /etc/kubernetes/kubeadm-init.yaml  sudo kubeadm config images pull --config /etc/kubernetes/kubeadm-init.yaml |

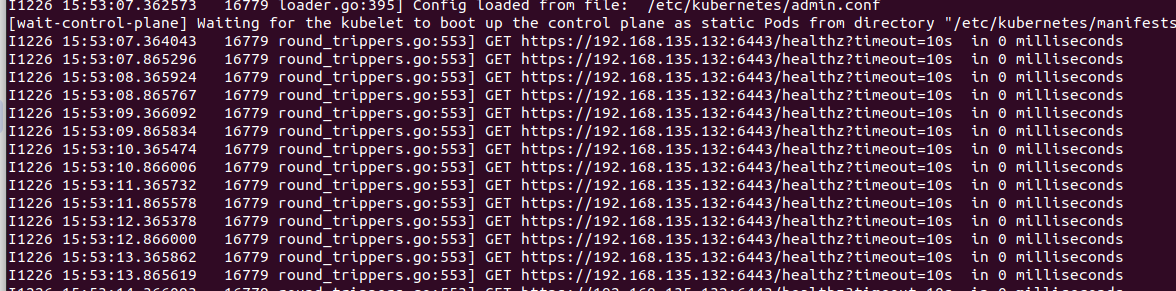


**2.初始化master节点:**

|  |
| --- |
| Bash  kubeadm init --config=/etc/kubernetes/kubeadm-init.yaml \  #--ignore-preflight-errors=all #警告不重要，眼不见心不烦 |



**3.出现下面情况正常**



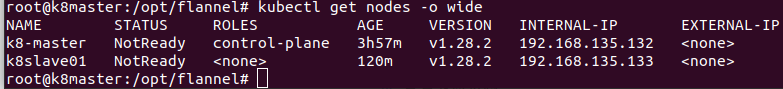
**4.按提示创建文件夹和加文件权限（slave节点加入时也要操作一遍，需要把admin.conf文件copy一份）**

|  |
| --- |
| Bash #To start using your cluster, you need to run the following as a regular user:  #主从节点都需要执行一下操作 mkdir -p $HOME/.kube  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config  sudo chown $(id -u):$(id -g) $HOME/.kube/config  #Alternatively, if you are the root user, you can run: export KUBECONFIG=/etc/kubernetes/admin.conf  #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/ |

**5.初始化slave节点（所有slave节点）**

|  |
| --- |
| Bash #Then you can join any number of worker nodes by running the following on each as root: #修改/etc/kubernetes所有者 chown k8-slave-01:k8-slave-02 /etc/kubernetes #加入之前需要把master节点的admin.conf文件复制一份到slave节点 scp ubuntu@10.10.6.204:/etc/kubernetes/admin.conf /etc/kubernetes#大概率报错操作不被允许  #子节点加入cluster集群,用你自己master节点的token加入 kubeadm join 192.168.135.151:6443 --token abcdef.0123456789abcdef \  --discovery-token-ca-cert-hash sha256:ff919978afc0cbf7f089b255f50508c879966b864106132cc07d2a31e2be72dc |

**6.查看节点状态**



需要下载cni插件,初始化集群网络

**七、下载cni容器网络接口插件（运行kubernetes的主机）**

**（国内docker镜像网站目前基本上被封杀了）**

在Kubernetes（K8S）集群中安装CNI（Container Network Interface，容器网络接口）是一个重要的步骤，它允许管理员选择不同的网络插件以满足其集群的不同需求。下面是一个详细的安装步骤指南，主要以calico插件为例：

**1.配置DNS**

|  |
| --- |
| Bash #修改hosts文件 sudo nano /etc/hosts  #添加DNS映射,解决部分网站跨域问题 10.10.6.204 registry.docker.io 185.199.111.133 raw.githubusercontent.com 147.75.40.148 kubernetes.io 20.205.243.166 github.com 185.199.108.133 raw.githubusercontent.com |

|  |
| --- |
| Bash #使用指令 sed -i '$ a 185.199.108.133 raw.githubusercontent.com' /etc/hosts sed -i '$ a 185.199.111.133 raw.githubusercontent.com' /etc/hosts sed -i '$ a 147.75.40.148 kubernetes.io' /etc/hosts sed -i '$ a 20.205.243.166 github.com' /etc/hosts sed -i '$ a 147.75.40.148 kubernetes.io' /etc/hosts #私服 sed -i '$ a 10.10.6.204 registry.docker.io' /etc/hosts |

**2.安装calico**

**1.第一种方式mainfest:**

|  |
| --- |
| Bash #download the yaml file sudo mkdir /etc/calico cd /etc/calico  wget https://raw.githubusercontent.com/projectcalico/calico/v3.29.1/manifests/calico.yaml |

**2.第二种方式operator:**

|  |
| --- |
| Bash #download the yaml file sudo mkdir /etc/calico cd /etc/calico wget https://raw.githubusercontent.com/projectcalico/calico/v3.29.1/manifests/tigera-operator.yaml wget https://raw.githubusercontent.com/projectcalico/calico/v3.29.1/manifests/custom-resources.yaml |

**3. 修改镜像：**

|  |
| --- |
| Bash #manifest: #search original images grep image: /etc/calico/calico.yaml   image: docker.io/calico/cni:v3.29.1 image: docker.io/calico/cni:v3.29.1 image: docker.io/calico/node:v3.29.1 image: docker.io/calico/node:v3.29.1 image: docker.io/calico/kube-controllers:v3.29.1 |

|  |
| --- |
| Bash #this image is avaliable: #modefy this yaml file #把镜像源修改为自己的私服镜像源 sed -i 's@docker.io/calico/@registry.docker.io:5000/@g' /etc/calico/calico.yaml |

|  |
| --- |
| Bash #operator grep image: /etc/calico/tigera-operator.yaml   image: image: quay.io/tigera/operator:v1.36.2 |

|  |
| --- |
| Bash sed -i 's@quay.io/tigera/@registry.docker.io:5000/@g' /etc/calico/tigera-operator.yaml |

**4.安装启用calico**

|  |
| --- |
| Bash #manifest kubectl apply -f /etc/calico/calico.yaml |

|  |
| --- |
| Bash #operator kubectl create -f /etc/calico/tigera-operator.yaml kubectl create -f /etc/calico/custom-resources.yaml |

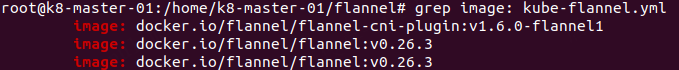
**3.安装flannel**

**1.下载flannel插件**

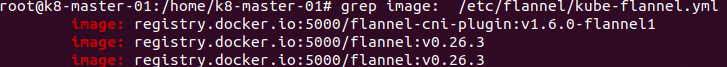
|  |
| --- |
| Bash sudo mkdir /etc/flannel cd /etc/flannel wget https://raw.githubusercontent.com/flannel-io/flannel/master/Documentation/kube-flannel.yml |

**2.查看所需镜像并修改镜像源**

|  |
| --- |
| Bash grep image: /etc/flannel/kube-flannel.yml |



|  |
| --- |
| Bash #把镜像源修改为自己的私服镜像源 sed -i 's@docker.io/flannel/@registry.docker.io:5000/@g' /etc/flannel/kube-flannel.yml #查看修改后的镜像 grep image: /etc/flannel/kube-flannel.yml |



**3.安装启用flannel**

|  |
| --- |
| Bash kubectl apply -f /etc/flannel/kube-flannel.yml |

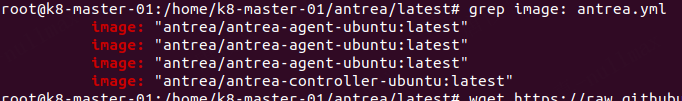
**4.安装antrea**

1.下载antrea

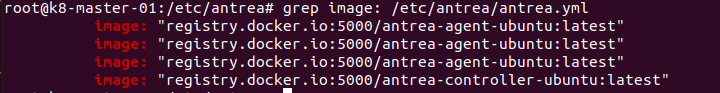
|  |
| --- |
| Bash sudo mkdir /etc/antrea cd /etc/antrea wget https://github.com/antrea-io/antrea/releases/download/v2.1.0/antrea.yml |

2.查看原镜像并修改镜像源

|  |
| --- |
| Bash grep image: /etc/antrea/antrea.yml |



|  |
| --- |
| Bash sed -i 's@antrea/@registry.docker.io:5000/@g' /etc/antrea/antrea.yml  grep image: /etc/antrea/antrea.yml |



**3.安装启用flannel**

|  |
| --- |
| Bash kubectl apply -f /etc/antrea/antrea.yml |

**5.查看节点是否正常运行**

|  |
| --- |
| Bash kubectl get node |

