# Kubernetes 1.6安装部署文档(CentOS x64)

参考文献:

<https://kubernetes.io/docs/getting-started-guides/kubeadm/>

https://segmentfault.com/a/1190000007074726

**(1/4) 安装kubelet 和 kubeadm**

所有的主机需要安装以下包:

**docker** 容器运行环境，推荐v1.12版本

**kubelet** kubernetes核心组件，可以运行在集群内的所有主机上，允许操作pod和容器

**kubectl** kubernetes集群命令行工具，一般在master节点使用较多，但其他nodes节点上可能也会使用到

**kubeadm** 引导安装kubernetes集群的命令行工具

**注意：**如果主机上已经安装过kubeadm 请及时更新到最新版本

取得root权限进行安装

(不能联网需要自行搭建本地yum源)

新增yum源:

cat <<EOF > /etc/yum.repos.d/kubernetes.repo

[kubernetes]

name=Kubernetes

baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-${ARCH}

enabled=1

gpgcheck=1

repo\_gpgcheck=1

gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg

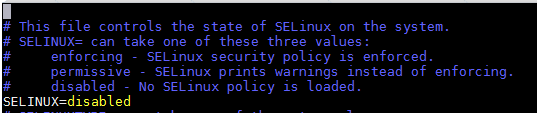
https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg

EOF

yum makecache

#关闭SELinux防火墙 docker容器网络需要访问宿主机文件系统 且容器间相互访问

setenforce 0 (永久关闭 修改/etc/sysconfig/selinux 配置文件 SELINUX=disabled或者permissive)

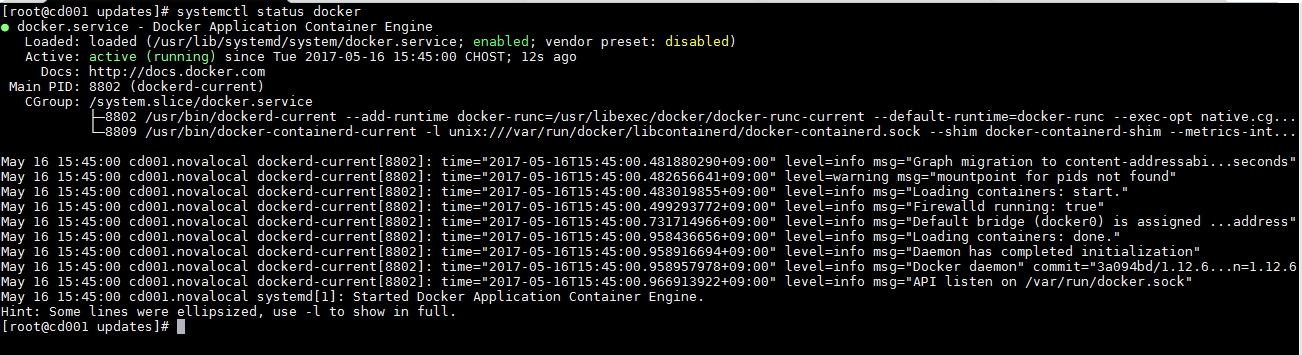


#安装docker kubernetes相关组件

yum install -y docker kubelet kubeadm kubectl kubernetes-cni

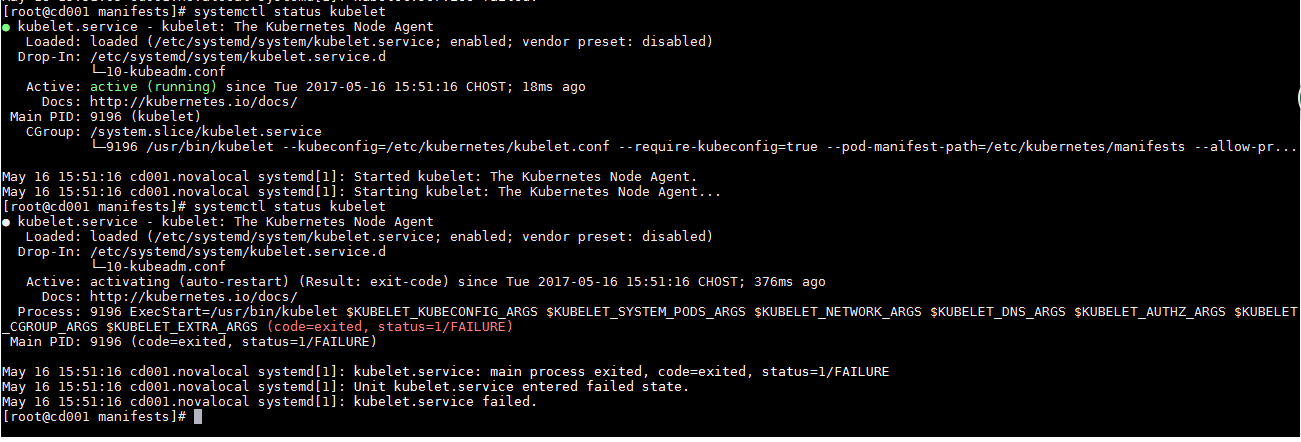
#启动docker服务并使其开机自启动

systemctl enable docker && systemctl start docker



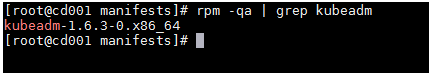
#启动kubelet服务并使其开机自启动

systemctl enable kubelet && systemctl start kubelet



注意：kubelet进程此时 每隔数秒就自动重启，因为它在循环等待kubeadm的调度

再次确认kubeadm安装成功 且是最新版本



**(2/4) 初始化master节点(单节点工作)**

master节点是控制层组件运行的主机，包含etcd和API Server等组件

选择任一安装了kubeadm的主机运行

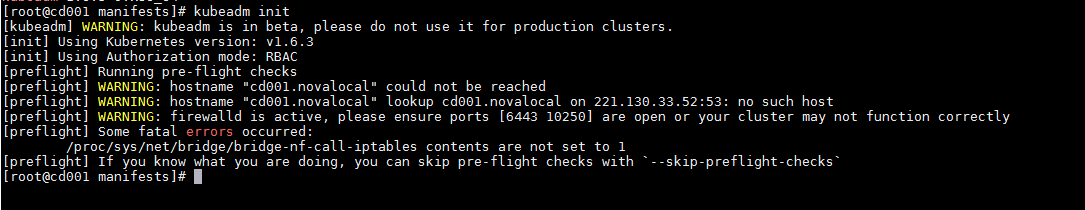
kubeadm init

注意：执行完上述命令后，将自动发现并使用默认网关的一张网卡，如果需要使用不同的网卡，可以加入如下参数 --apiserver-advertise-address=<ip-address>提供给kubeadm init

Master为其他node节点分配网络空间，订制pod的网络，如果使用flannel订制pod的网络，则需要为kubeadm init

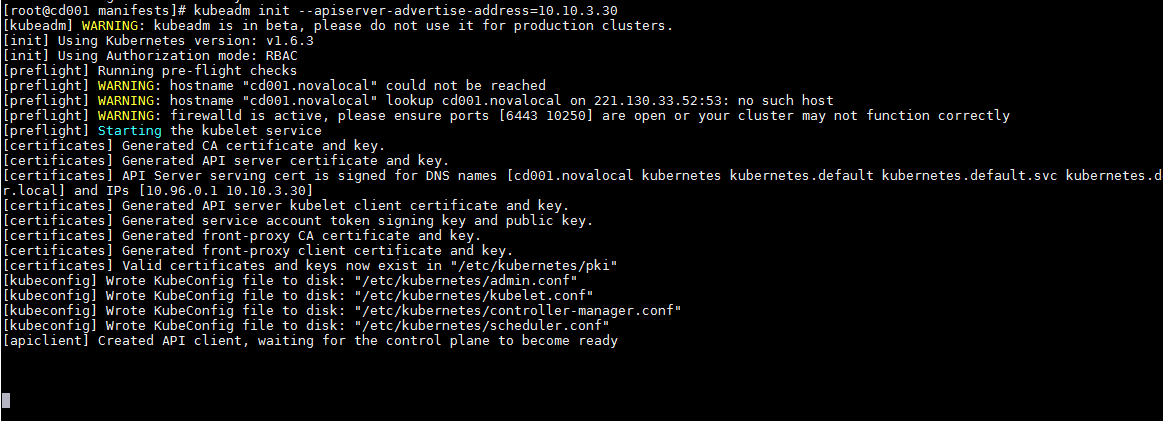
添加参数--pod-network-cidr=10.244.0.0/16 其他网络插件则无需配置此参数

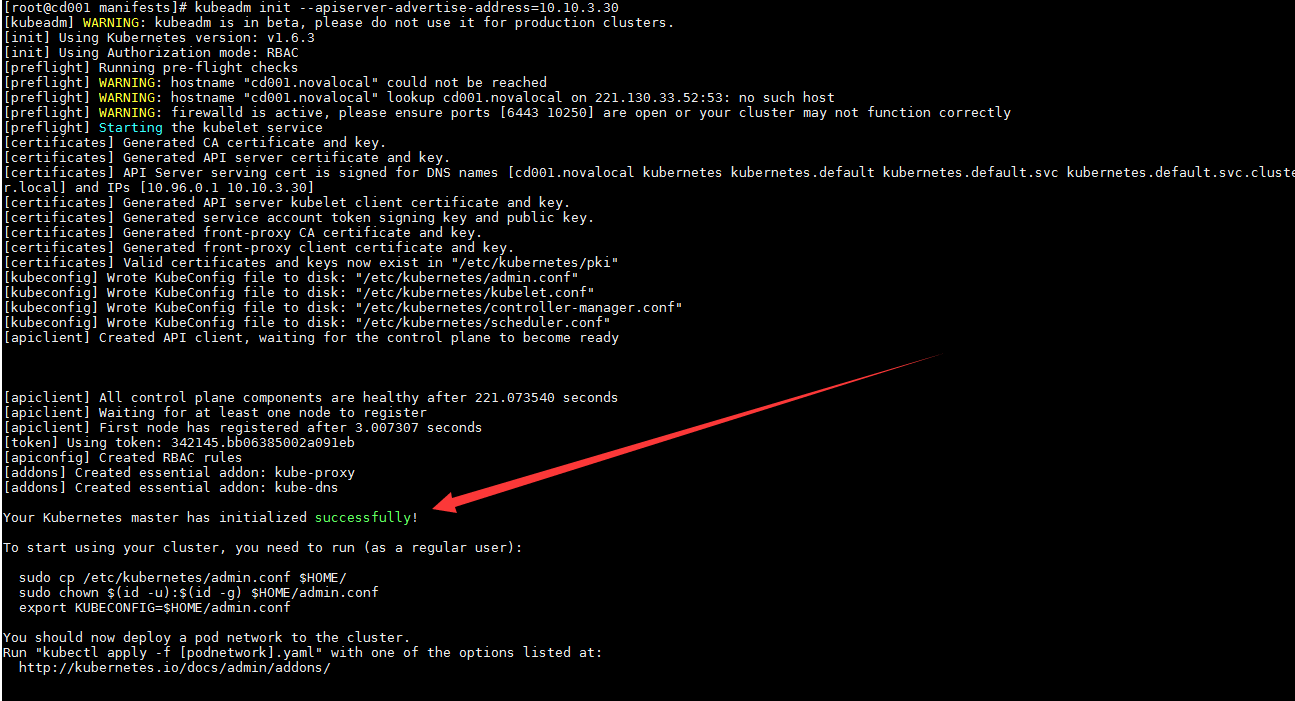
kubeadm init将会首先执行一系列的与检查，来确保该主机能够运行kubernetes，一旦检查出错，则会显示告警并退出。后续会下载和安装集群数据库等控制组件。此环节会持续数分钟，当第一个kubeadm未结束时 不要开启第二个。

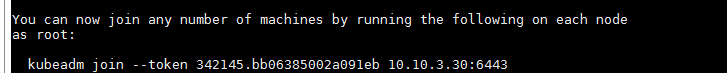


解决相关报错

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







记录下kubeadm join命令 和认证文件，稍后会使用

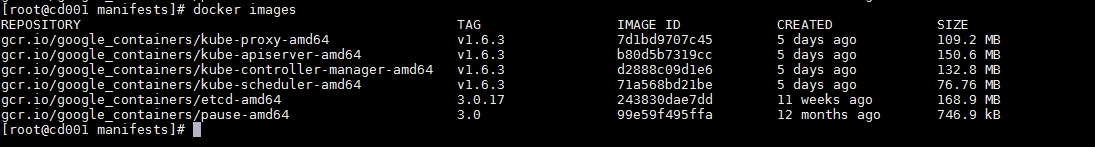
kubeadm join --token 342145.bb06385002a091eb 10.10.3.30:6443

根据提示 添加环境变量 使本机的kubectl等具有访问apiserver的权限

export KUBECONFIG=/etc/kubernetes/admin.conf (可以加在profile文件里)

以下控制组件都将在pod中运行，他们的镜像如果缺少则会在kubeadm init的时候被kubelet自动拉下来





docker pull dipperroy/kube-controller-manager:1.6.3 //使用本地的镜像 官方的镜像有问题

删除正在使用gcr.io/google\_containers/kube-controller-manager-amd64:v1.6.3官方镜像的容器

删除镜像 并为本地镜像tag与官方镜像一样名字

docker tag docker.io/dipperroy/kube-controller-manager:1.6.3 gcr.io/google\_containers/kube-controller-manager-amd64:v1.6.3

出于安全角度的考虑，默认状态下，kubernetes不会将pod调度到master节点。

若有特殊需求，需要将master也加入调度的集合，可以使用以下命令:

#不建议这么使用

kubectl taint nodes --all node-role.kubernetes.io/master-

**(3/4) 安装pod网络环境**

没有配置网络环境的pod间是无法通讯的，因此部署服务之前必须先解决网络环境问题。同时kube-dns也需要在网络环境安装之后才能运行。Kubeadm仅支持基于CNI模型的插件(不支持Kubenet)

使用如下命令可以选择一种CNI模型的插件提供网络服务

kubectl apply -f <add-on.yaml>

如使用calico网络，可以在官方下载calico.yaml 并完成配置

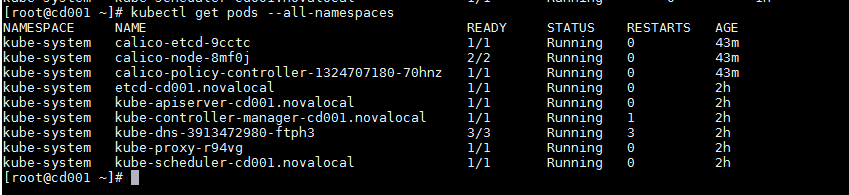
(获取calico.yaml) wget http://docs.projectcalico.org/v2.2/getting-started/kubernetes/installation/hosted/kubeadm/1.6/calico.yaml

#创建calico网络

kubectl create -f calico.yaml

注意:单个集群只能安装一种pod网络环境

使用kubectl get pods --all-namespaces命令查看kube-dns、calico等容器是否正确运行，可以作为判断pod网络环境是否安装正确的条件。



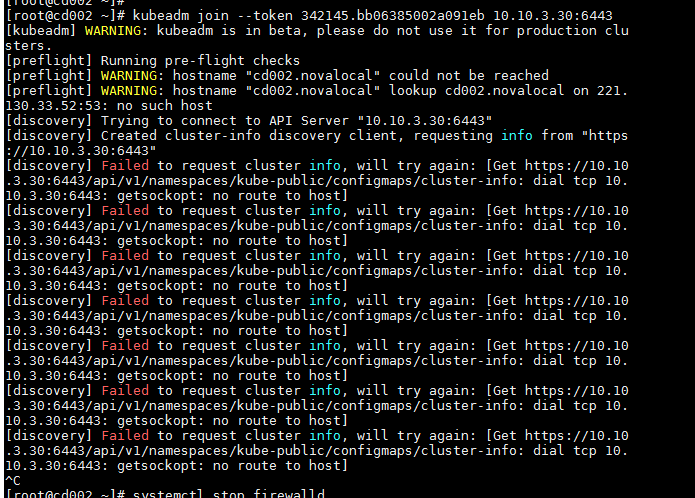
**(4/4) 添加Node进集群**

Node节点上运行着各个具体的工作单元(如pod、容器、etcd等)，为了将新的nodes加入到集群，需要在每一台node上运行作以下准备工作:

ssh 连接到node主机。

取得root权限。

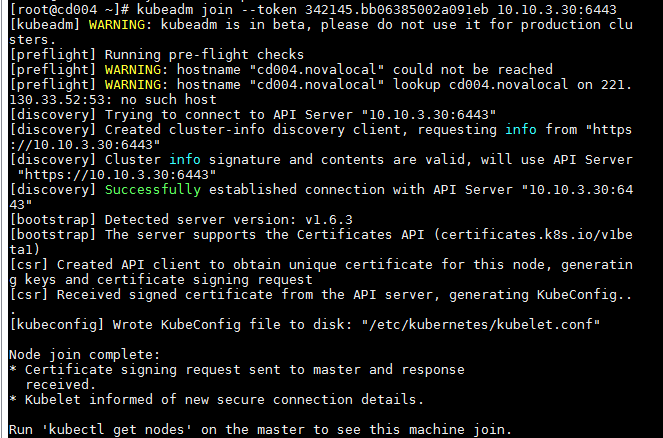
kubeadm join --token 342145.bb06385002a091eb 10.10.3.30:6443



解决相关告警 报错信息

systemctl stop firewalld

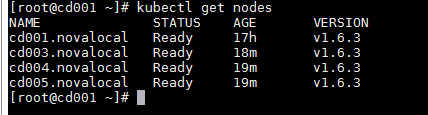
成功加入后去 master节点上查看node状态



首先等待各个node的网络初始化完成

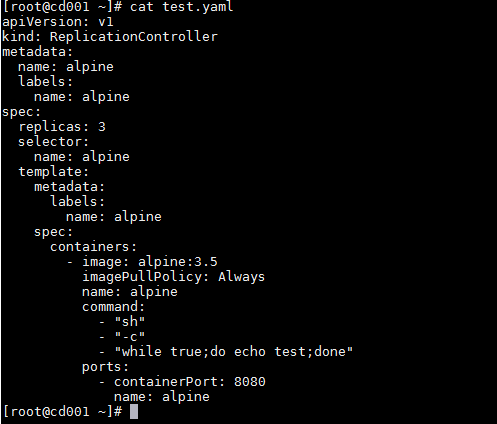


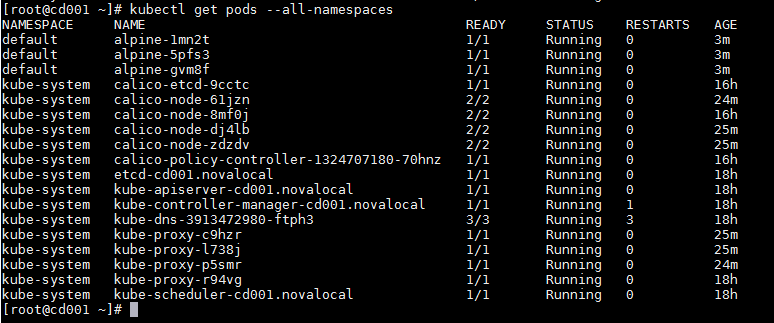
查看集群状态



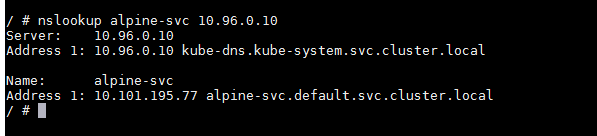
尝试创建几个pod相互访问



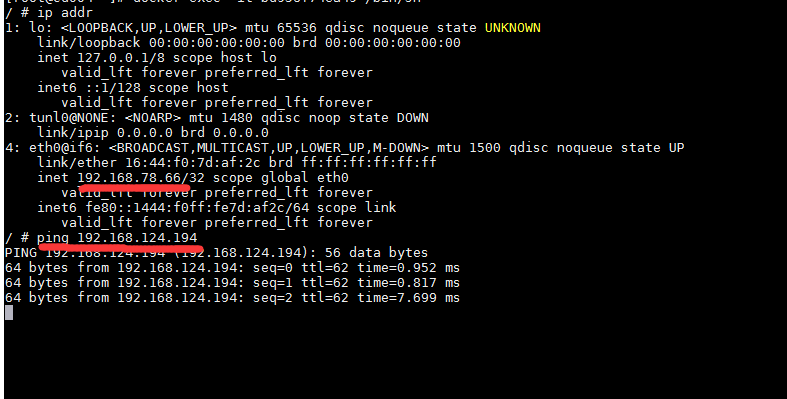




在容器内尝试解析自身服务的域名



再容器内尝试ping其他节点上的容器IP



集群安装成功！