

部署（超详细）

于 2022-04-13 11:40:13 修改

专栏收录该内容

0 订阅 4 篇文章

是Google 2014年创建管理的，是Google 10多年大规模容器管理技术Borg的开源版本。它是容器集群管理系统，是一个开源的平台，可以实现容器集群的自动化部署、自动扩缩容、维护等功能。

你可以：

应用

应用

新的应用功能

优化硬件资源的使用

点：

持公有云，私有云，混合云，多重云（multi-cloud）

模块化, 插件化, 可挂载, 可组合

自动部署，自动重启，自动复制，自动伸缩/扩展

；

前，部署Kubernetes集群机器需要满足以下几个条件：

机器，操作系统 CentOS7.7以上64位系统

2GB或更多RAM，2个CPU或更多CPU，硬盘30GB或更多

与机器之间网络互通

外网，需要拉取镜像

分区

！

s1.16.2版本，docker19.03版本

点上安装Docker和kubeadm，kubenet

metes Master

内容来源：csdn.net
作者昵称：Anime777
原文链接：https://blog.csdn.net/qq_40954652/article/details/123208590
作者主页：https://blog.csdn.net/qq_40954652qq_40954652

网络插件

ernetes Node，将节点加入Kubernetes集群中

三台机器

要太低否者可能会出问题（最好7.7以上）

```
.0.1.176 主机名: test-docker-176 系统: centos7.7 配置: 8核8G
.0.1.177 主机名: test-docker-177 系统: centos7.7 配置: 8核8G
.0.1.178 主机名: test-docker-178 系统: centos7.7 配置: 8核8G
```

要执行如下命令

selinux

```
ctl stop firewalld

ctl disable firewalld

's/enforcing/disabled/' /etc/selinux/config

orce 0
```

```
f -a # 临时关闭

tc/fstab # 注释到swap那一行 永久关闭
```

P对应关系(三台主机都执行)

```
.176 k8s-master-176
.177 k8s-node-177
.178 k8s-node-178
/etc/hosts
```

名 (从176-178各自顺序执行)

```
mectl set-hostname k8s-master-176

mectl set-hostname k8s-node-177

mectl set-hostname k8s-node-178
```

流量传递到iptables的链(三台主机都执行):

内容来源: [csdn.net](https://www.csdn.net)

作者昵称: [Anime777](#)

原文链接: https://blog.csdn.net/qq_40954652/article/details/123208590

作者主页: https://blog.csdn.net/qq_40954652qq_40954652

ip_forward如存在=0, 修改为1即可

```
/etc/sysctl.d/k8s.conf << EOF
net.ipv4.ip_forward=1
net.ipv4.tcp_tw_recycle = 0
bridge.bridge-nf-call-ip6tables = 1
bridge.bridge-nf-call-iptables = 1

--system
```

内核到最新

步骤: [linux \(centos7\) 内核升级](#) (低内核会导致 [docker](#) 和k8s运行不稳定)

安装Docker/kubeadm/kubelet

使用文档进行安装: [docker部署搭建](#) (注意: 这里docker最好安装19.03.9版本, 防止集成到k8s出现bug, 这里我出现过k8s监控服务无法获取资源信息, yum install -y docker-ce-19.03.9 docker-ce-cli-19.03.9

修改, 设置cgroup驱动, 这里用systemd

下

daemon.json

```
{
  "graphdriver": "overlay2",
  "storage": "/data/docker",
  "registry-mirrors": ["https://01xxgaft.mirror.aliyuncs.com"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  }
}
```

```
systemctl restart docker
```

阿里云YUM软件源

```
/etc/yum.repos.d/kubernetes.repo << EOF[kubernetes]
name=kubernetes
baseurl=https://mirrors.aliyun.com/kubernetes/yum/repos/kubernetes-el7-x86_64
enabled=1
```

内容来源: [csdn.net](#)
作者昵称: [Anime777](#)
原文链接: https://blog.csdn.net/qq_40954652/article/details/123208590
作者主页: https://blog.csdn.net/qq_40954652qq_40954652

```
ck=0  
pgcheck=0  
=https://mirrors.aliyun.com/kubernetes/yum/doc/yum-key.gpg https://mirrors.aliyun.com/kubernetes/yum/doc/rpm-package-key.gpg  
bernetes]
```

1, kubelet和kubectl

```
install -y kubelet-1.16.2 kubeadm-1.16.2 kubectl-1.16.2  
  
ctl start kubelet  
  
ctl enable kubelet
```

的master和node节点

点, 在10.0.1.176执行

节点

```
m init --apiserver-advertise-address=10.0.1.176 --image-repository registry.aliyuncs.com/google_containers --kubernetes-version v1.16.0 --service-cidr=10.140.0.0/16 --pod-network-cidr=1
```

个网段不要重, 后面是/16, 不要与当前机器网段一样

生成一串命令用于node节点的加入, 记录下来, 接着执行以下命令

```
-p $HOME/.kube  
  
/etc/kubernetes/admin.conf $HOME/.kube/config  
  
$(id -u):$(id -g) $HOME/.kube/config
```

l, 在10.0.1.177和178执行

生成的一串命令, 我这里只是示例, 命令根据你实际生成的复制去node节点执行

```
m join 10.0.1.176:6443 --token 1146ci.3elqkwg3b2sdqp6l --discovery-token-ca-cert-hash sha256:3562d07fbe47d39ddc04c9170464e2699f1328c08ded2bb0198bde9a55ce8e5e
```

插件 (CNI)

二选一, master上执行

件 (轻量级用于快速搭建使用, 初学推荐)

```
http://download.stisd.cn/k8s/kube-flannel.yml
```

son下面的网段为上面init pod-network-cidr的网段地址 (必须正确否则会导致集群网络问题)

内容来源: csdn.net

作者昵称: Anime777

原文链接: https://blog.csdn.net/qq_40954652/article/details/123208590

作者主页: https://blog.csdn.net/qq_40954652qq_40954652

```
's/10.244.0.0/10.240.0.0/' kube-flannel.yml
```

牛, 执行

```
l apply -f kube-flannel.yml

l get pods -n kube-system
```

不下来, 就将199.232.68.133 raw.githubusercontent.com 放到hosts内

‡ (用于复杂网络环境)

```
http://download.stisd.cn/k8s/calico-3.9.2.yaml
```

内网段为上面init pod-network-cidr的网段地址 (必须正确否则会导致集群网络问题)

```
's/192.168.0.0/10.240.0.0/g' calico-3.9.2.yaml
```

牛, 执行

```
l apply -f calico.yaml

l get pod --all-namespaces -o wide
```

ernetes集群

s集群中创建一个pod, 验证是否正常运行

开放对外端口访问, 这里会随机映射一个端口

```
l create deployment nginx --image=nginx

l expose deployment nginx --port=80 --type=NodePort
```

必须是running状态而且ready是1, 并查看nginx svc的80端口映射到了哪个端口

```
l get pod,svc
```

内刚刚查看的映射端口, 看看是否nginx已经运行

命令

rice, endpoints, secret等等的状态

```
l get 组件名 # 例如kubectl get pod 查看详细信息可以加上-o wide 其他namespace的指定 -n namespace名
```

内容来源: csdn.net

作者昵称: 阿狸m777

原文链接: https://blog.csdn.net/qq_40954652/article/details/123208590

作者主页: https://blog.csdn.net/qq_40954652qq_40954652

个yaml文件内资源，也可以是目录，目录内包含一组yaml文件（实际使用中都是以yaml文件为主，直接使用命令创建pod的很少，推荐多使用yaml文件）

```
l apply -f xxx.yaml # 例如kubectl apply -f nginx.yaml 这里是如果没有则创建，如果有则变更，比create好用
```

文件内资源，也可以是目录，目录内包含一组yaml文件

```
l delete -f xxx.yaml # 例如kubectl delete -f nginx.yaml
```

比如有一组deployment内的pod没起来，一般用于pod调度过程出现的问题排查

```
l describe pod pod名 # 先用kubectl get pod查看 有异常的复制pod名使用这个命令
```

用于pod状态未就绪的故障排查

```
l logs pod名 # 先用kubectl get pod查看 有异常的复制pod名使用这个命令
```

或者是pod资源（cpu，内存资源）使用情况

```
l top 组件名 # 例如kubectl top node kubectl top pod
```

```
l exec -ti pod名 /bin/bash # 先用kubectl get pod查看 有需要的复制pod名使用这个命令
```

丁用

8s单master节点部署想学习高可用部署，参考 [k8s高可用搭建部署](#)

文章对你有帮助的话请给我点个👍吧，感谢

点与官方知识档案匹配，可进一步学习相关知识

网 > 容器编排(生产环境 k8s) > kubelet,kubectl,kubeadm三件套 393 人正在系统学习中