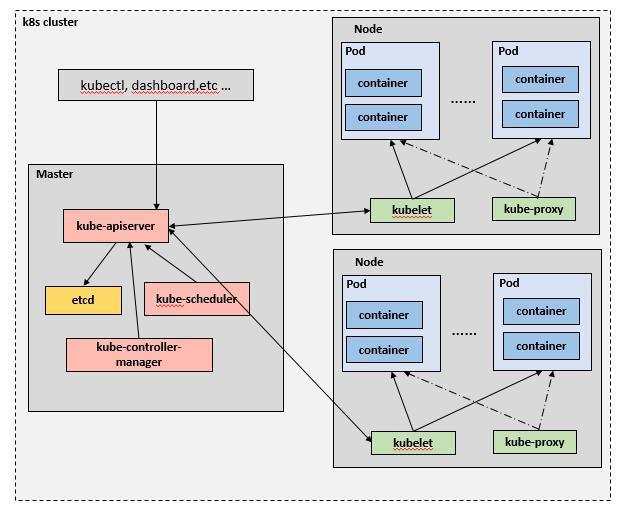
Kubernetes部署操作手册

中国移动（成都）产业研究院

2020年8月13日

#### Kubernetes介绍

1、工作模型



2、适用范围

3、工作原理

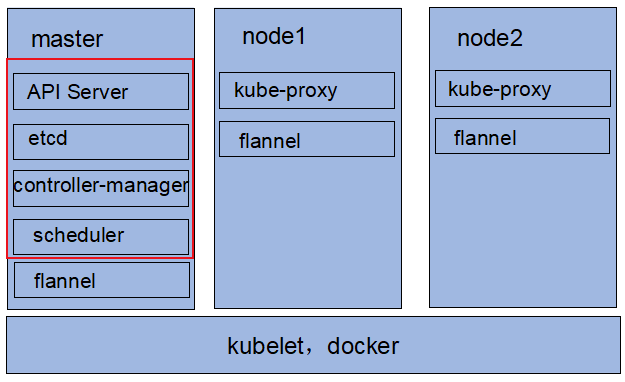
4、组件介绍

**master：etcd、API Server、Controller Manager、Scheduler**

**node：Kubelet、Proxy、[Docker](http://lib.csdn.net/base/docker) daemon**

**常见创建yaml中kind有：**

**service pod deployment token 或其他namespace secrets endpoint**



#### 部署步骤说明

1、本方案先完成部署单master结构，然后再次基础上做kubernetes集群，当master集群部署完成后无需任何操作，只需在node节点先准备系统环境，然后装kubernetes相关服务，最后node加入master初始化时创建的集群即可；

2、集群部署完成后，我们可以根据实际需要编拟相关服务yaml(本方案只以nginx为例，其他服务类似)，这时我们就可以看到kubernetes会根据预先设定在yaml里的参数对pod进行创建、启动、重启、销毁，监控及负载，保证高可用和负载均衡。

#### 系统环境准备(all\_node部署)

1、阿里云YUM源

源文件下载

cd /etc/yum.repos.d/

wget http://mirrors.aliyun.com/repo/Centos-7.repo

阿里云YUM源缓存

yum clean all && yum repolist && yum makecache

2、网卡DNS地址修改

vim /etc/sysconfig/network-scripts/ifcfg-ens33

DNS1=223.5.5.5 //新增一行

3、关闭并永久注释swap分区

swapoff -a

vim /etc/fstab

# UUID=f07d05ef-1672-402d-8a1a-6913c8f5aa4e swap swap defaults 0 0

4、配置路由转发

vim /proc/sys/net/ipv4/ip\_forward

1

vim /etc/sysctl.d/k8s.conf

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

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

sysctl --system 让修改配置生效

5、主机与ip对应关系

vim /etc/hosts

192.168.12.129 jumpserver

192.168.12.130 node1

192.168.12.131 node2

6、关闭防火墙、selinux

#### 安装docker-ce(all\_node部署)

mkdir /usr/local/kubernetes/

cd /usr/local/kubernetes/

1、安装Docker社区版（CE）

yum-config-manager --add-repo <https://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo>

如果安装报错

wget https://download.docker.com/linux/centos/7/x86\_64/edge/Packages/containerd.io-1.2.6-3.3.el7.x86\_64.rpm

rpm -ivh containerd.io-1.2.6-3.3.el7.x86\_64.rpm

yum -y install docker-ce

2、创建aliyundocker仓库加速包

vim /etc/docker/daemon.json

{

"registry-mirrors": ["https://fl791z1h.mirror.aliyuncs.com"]

}

#### 安装&初始化

1、增加kubernetes相关yum源(all\_node部署)

vim /etc/yum.repos.d/CentOS-Base.repo

[kubernetes]

name=Kubernetes

baseurl=https://mirrors.aliyun.com/kubernetes/yum/repos/kubernetes-el7-x86\_64/

enabled=1

gpgcheck=1

repo\_gpgcheck=1

gpgkey=https://mirrors.aliyun.com/kubernetes/yum/doc/yum-key.gpg https://mirrors.aliyun.com/kubernetes/yum/doc/rpm-package-key.gpg

2、安装(all\_node部署)

yum -y install kubectl kubelet kubeadm

systemctl start kubelet.service

systemctl enable kubelet.service

说明：本方案安装时阿里云最新为v1.18.8，如想指定安装版本可以用一下命令安装：

yum install -y kubelet-1.18.8 kubeadm-1.18.8 kubectl-1.18.8

**3、初始化(这一步非常重要)** (master\_node部署)

kubeadm init --apiserver-advertise-address=192.168.12.129 \

--image-repository registry.aliyuncs.com/google\_containers \

--kubernetes-version=1.18.0 \

--service-cidr=10.10.0.0/16 \

--pod-network-cidr=10.244.0.0/16

//如初始化过程中有error需要处理，warning无需处理，所有ip地址和网段需根据实际情况设定，由于默认谷歌镜像位于国外，这里需指定阿里云镜像做拉取镜像地址的转换。

**初始化成功最后一行信息需记录，方便后期其他节点加入该集群**

**kubeadm join 192.168.12.129:6443 --token ltk4d6.zbgkk45mdrj6m18w --discovery-token-ca-cert-hash sha256:42aeb453952d4b85b378338c28304812f1d155f123bf29b5cfa76f385c75b544**

初始化完成后可通过命令tree /etc/kubernetes查看生成的所有配置文件

初始化完成后查看正在运行的程序

netstat -tnlp

根据提示创建kubectl

echo $HOME

/root

mkdir -p $HOME/.kube

cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

chown $(id -u):$(id -g) $HOME/.kube/config

**执行下面命令，使kubectl可以自动补充**

**source <(kubectl completion bash)**

查看节点node

kubectl get node

kubectl get pods --all-namespaces

**安装calico网络**(master\_node部署)

**kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml**

查看节点node

kubectl get node

kubectl get pod --all-namespaces

#### dashboard安装(master\_node部署)

1、下载kubernetes-dashboard.yaml

wget https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.0-rc7/aio/deploy/recommended.yaml

注：若失败，在/etc/hosts添加项

151.101.76.133 raw.githubusercontent.com

**注意：下载完成后查看有无images地址是国外地址**

2、vim recommended.yaml

kind: Service

……

spec:

**type: NodePort**

ports:

- port: 443

targetPort: 8443

**nodePort: 30000**

……

注：红色为新增项

启动

**kubectl apply -f recommended.yaml**

查看节点pod

**kubectl get pod --all-namespaces**

查看pod所在节点

**kubectl get pods -n kubernetes-dashboard -o wide**

查看pod所暴露的端口

**kubectl get svc -n kubernetes-dashboard**

查看pod日志

**kubectl logs -f kubernetes-dashboard-5d4dc8b976-7vhlk -n kubernetes-dashboard**

#### dashboard-admin安装(master\_node部署)

1、vim admin.yaml



2、起服务

kubectl apply -f admin-sa.yaml

3、取token

**kubectl get secret -n kube-system | grep admin**

**kubectl get secret admin-token-s7krx -o jsonpath={.data.token} -n kube-system |base64 -d**

4.网页登录

#### node节点安装部署(节点node部署)

1、检查系统环境及kubernetes相关服务

检查防火墙、selinux、swap、/etc/hosts、sysctl、docker、yum源、kubeadm、kubectl、kubelet

2、加入集群(节点必须能上外网)

**kubeadm join 192.168.12.129:6443 --token ymday9.kic13b5tuldh5lia --discovery-token-ca-cert-hash sha256:42aeb453952d4b85b378338c28304812f1d155f123bf29b5cfa76f385c75b544**

**加入时有error需处理**

3、检查集群

可在网页kubernetes-dashboard上查看

查看node状态、namespace状态、service状态、deployment状态、pod状态、secret状态等等，有error时需处理

#### 部署nginx(master\_node部署)

1、部署pv

安装nfs服务端(all\_node部署)

yum -y install nfs-utils rpcbind

建立共享目录

mkdir /root/share

chmod 777 /root/share

配置nfs服务端

vim /etc/exports

/root/share \*(rw,sync,insecure,no\_subtree\_check,no\_root\_squash)

起服务

systemctl restart rpcbind.service

systemctl restart nfs.service

客户端验证

showmount -e 192.168.12.129

挂载(客户端)

mkdir /data

mount -t nfs 192.168.12.129:/root/share /data

验证

echo “This is a test page”> /data/test.txt (客户端)

ls /root/share/test.txt (服务端)

编写pv\_nfs.yaml



起服务

kubectl apply -f volume/pv\_nfs.yaml(起服务后STATUS为Available)

2、部署pvc



起服务

kubectl apply -f volume/pv\_nfs.yaml(起服务后STATUS为Bound)

3、编拟yaml(服务启动，端口映射，注意有两种方式

方式1(nginx.yaml)：本方式通过各个pod所落在的节点映射其宿主机的内容，两宿主机可

通过rsync定期同步保持数据一致，该方式可以保证高可用，即使一台宿主机宕机，服务仍不受影响，缺点是数据存在时延

方式2(nginx1.yaml)：本方式通过某一台宿主机提供一共享目录，其他宿主机均从该共享目录获取数据，在启动pod时指定工作目录为该共享目录，缺点不保证高可用，但数据一致得到保证



4、起服务

cd /etc/kubernetes

kubectl apply -f nginx.yaml

5、查看状态

kubectl get namespaces //查看新创建的namespace

get pods -n nginx-service -o wide //查看新创建的pod状态(ip地址及其物理宿主机)

kubectl get svc -n nginx-service //查看service对外暴露的端口

6、访问测试

http://192.168.12.130:32000 或

<http://192.168.12.131:32000>

#### prometheus监控 (master\_node，启动顺序按如下步骤)

第一步------》第十五步

1、namespace创建

vim monitor\_namespace.yaml



注意事项：无

2、serviceaccounts创建

vim prometheus-rbac.yaml



注意事项：无

3、configmap创建



注意事项：如果配置完后prometheus\_graph无数据，很有可能是这里的配置问题，配置文件job在这里无任何展示，说明该配置文件也有问题，该配置文件相当于prometheus的主配置文件。

4、ingress创建



5、pv&pvc创建



注意：pv规格和pvc需对应才能匹配上

6、service创建



7、prometheus主服务创建



注意：如果要使用pvc，需要与pvc名字对应，storagename与pvc要一一对应

K8S资源对象 kube-state-metrics(Pod/Deployment/Service) 端口 30001

K8SNode性能 node-exporter(节点CPU，内存利用率) 端口 31900

K8SPod性能 cAdvisor(容器CPU，内存利用率) 端口

K8S主服务 prometheus-statefulset(prometheus主配置文件)端口 30900

K8sGrafana grafana-deployment(页面展示) 端口 31999

8、node-exporter创建

vim node-exporter.yaml



注意：相当于在容器里面跑这个node\_exporter，需要映射出来查看是否已经得到数据，prometheus监控原理是需要一个exporter先获取数据，prometheus主服务只是做一个汇聚的过程。

9、应用state-service创建

vim kube-state-metrics-service.yaml



10、应用state-servicesaccout创建

vim kube-state-metrics-rbac.yaml



11、应用state-deployment创建

vim kube-state-metrics-deployment.yaml



12、grafana-pvc创建

vim grafana-pvc.yaml



13、grafana\_job.yaml创建

vim grafana\_job.yaml



14、grafana-ingress创建

vim grafana-ingress.yaml



15、grafana-deployment创建

vim grafana-deployment.yaml



创建后无grafana的endpoints，此时需要注意检查 service selector 和 pod lable 是否对应

16、grafana-service创建

vim grafana-service.yaml



17、网页导入对应dashboard模板

注意，前提是prometheus grafa有数据，prometheus实现原理：先在各节点收集并整理数据，数据要求为prometheus主服务能够识别的数据，然后各节点的exporter有的数据和监控项和导入的dashboard配置监控项一一对应，如果有dashboard为no data的，要么导入dashboard表达式有误，要么根本无相关的exporter服务启动并收集整理数据。

#### 常见命令及问题

**常见命令**

1、状态查看相关

kubectl get namespaces -o wide //查看命名空间

kubectl get pods --all-namespace -o wide //查看pod

kubectl get nodes -o wide //查看节点

kubectl get service --all-namespaces -o wide //查看service，跟svc一样

kubectl get svc --all-namespaces -o wide //查看svc，跟service一样

kubectl get secrets --all-namespaces -o wide //查看认证

kubectl get endpoints --all-namespaces -o wide //容器的ip及端口(所有服务映射源)

kubectl exec -n kube-system -it kube-proxy-p82j8 sh //进入容器

kubectl exec -n nginx-service nginx-5d69464f48-qbbjh cat /etc/resolv.conf //查看容器里面配置

…… //其他组件查看

2、排错相关

docker images //起服务时如果需要重新拉取镜像，镜像拉取失败需查看

kubectl describe pod -n kubernetes-dashboard kubernetes-dashboard-5d4dc8b976-swv8t //若pod状态非正常，需查看pod具体报错

kubectl logs -f -n kubernetes-dashboard kubernetes-dashboard-5d4dc8b976-swv8t

//若pod运行出错，需查看pod具体报错

journalctl -f -u kubelet //若节点状态不正常，需查看节点日志

dashboard网页查看 //查看所有组件报错原因

3、启动相关

kubectl create -f nginx.yml

kubectl apply -f nginx.yml

kubectl 命令创建：kubectl run 名称 --image= 镜像

4、删除相关

kubectl delete -f nginx.yml //删除yaml指定组件

kubectl delete nodes node1 //删除节点

kubectl delete namespaces nginx-service //删除命名空间

kubectl delete pods -n nginx-service nginx-684bd4f4c-lqmxr //删除pod

kubectl delete service -n nginx-service nginx-svc //删除service

kubectl delete secrets -n kubernetes-dashboard kubernetes-dashboard-token-4nt9f

//删除secrets

kubectl delete endpoints -n kubernetes-dashboard kubernetes-dashboard //删除endpoint

…… //删除其他组件

**常见问题及解决办法**

**1、注意事项**

**后续创建所有的service pod deployment token 或其他，先创建namespace做空间隔离，启动时需加上创建的namespace**

2、master初始化失败

swapoff -a

kubeadm reset

systemctl daemon-reload

systemctl restart kubelet

iptables -F && iptables -t nat -F && iptables -t mangle -F && iptables -X

再重新jion集群

3、dashboard相关问题及办法

a、直接登录显示：A http request to http server 解决办法：自签证书，dashboard版本

b、dashboard自签证书 解决办法：修改

c、dashboard版本问题 解决办法：要bate2版本 需要的版本

d、ERR\_cert\_invalid 解决办法：输入thisisunsafe

1. crashloopbackoff 解决办法：扩大内存

f、登录后所有业务为空 解决办法：创建admin用户

4、node加入集群及集群相关问题及办法

a、token过期无法加入集群（先查看集群token过期时间，一般有效期为24小时）

解决办法：kubeadm token list 命令： kubeadm token create

b、node1无法执行kubectl相关命令 解决办法：master上admin.conf拷贝到/etc/Kubernetes 然后在/etc/profile增加一行

vim /etc/profile

export KUBECONFIG=/etc/kubernetes/admin.conf

source /etc/profile

c、node1没法起calico解决办法：是否能连外网docker 查看无镜像，手动拉取镜像到本地

d、nginx pods启动 解决办法：是否能连外网docker 查看无镜像，手动拉取镜像到本地

e、master无8080端口 解决办法：无需启动只需执行kubectl相关命令即可

f、node1一直not ready 解决办法：/opt/cni /etc/cni /var/lib/cni拷贝到node

g、刷新节点 解决办法：删除节点后 重新加入

h、查看集群当前状态 解决办法：journalctl -f -u kubelet

5、部署服务相关问题及办法(以nginx为例)

a、如何确定容器查看服务返回值 解决办法：

curl -I -m 10 -o /dev/null -s -w %{http\_code} 10.244.166.129:80 //查看网页状态返回码

b、端口映射不成功，endpoint没有ip

c、不能跨主机访问容器之间的访问

d、本机pod映射宿主机本地目录不成功且无endpoint-----app和name不对应

e、app和kind的name要一一对应

f、一个服务落在几个node上，在所有node上端口都会被映射

g、某高可用pod集群所在的所有节点，提供服务时都会轮循到

h、cluster ip端口只能在有该集群pod时才能被访问

i、node ip + 被映射端口 在所有节点和浏览器均能被访问

j、容器 ip+ 端口只能在有该集群pod时才能被访问

k、想要进入容器，可在web UI找到pod 点执行选项

l、在资源受限的情况下，可以选择pod落在一个节点上

m、volume可以直接映射pod所在宿主机目录，也可以使所有pod映射nfs提供的共享目录内容

n、生产环境中所有node提供的网页是一样的，测试环境中为了看轮循效果，将不同node的index.html改为不同内容

o、生成：server共享目录---all\_client检查---pv(指定ip和目录)---pvc(根据给定条件自动匹配)---yaml引用

退出：先停掉yaml引用，再删pvc，再删pv，再删共享目录

6、部署prometheus监控相关问题及办法

1、Nfs -重启服务-Pv—pvc—调用(删除走反向)

2、configmap相关与prometheus的主配置文件，configmap里面文件路径需与本地符合

3、pv和pvc之间通过条件对应自动匹配且pv状态必须为available

4、deployment里面调用configmap时名字，两者名字要对应，

所有mountpath均为container里面的路径，

5、加入集群时注意token过期时间

6、今日grafana-deployment起不来

#### 批量部署

1. 批量示范(以ansible部署prometheus为例)



1. 服务配置相关操作
2. 配置文件修改(Hostname根据agent实际情况更改)

vim /usr/local/zabbix/etc/zabbix\_agentd.conf

Server=127.0.0.1

ServerActive=192.168.12.129

Hostname=Agent1

StartAgents=0 #禁止使用被动模式

RefreshActiveChecks=120

1. 服务启动

cd /etc/ansbile

ansible all -a 'systemctl start mysqld;/usr/local/nginx/sbin/nginx;/usr/local/zabbix/sbin/zabbix\_agentd;systemctl start php-fpm;systemctl start httpd'

1. 服务状态查看

netstat -antup | grep nginx

netstat -antup | grep zabbix\_agent

1. 监控脚本及配置文件验证

如果脚本和配置文件无误，可以得到访问结果

/usr/local/zabbix/bin/zabbix\_get -s 127.0.0.1 -p 10050 -k nginx.status[active]

/usr/local/zabbix/bin/zabbix\_get -s 127.0.0.1 -p 10050 -k nginx.status[reading]

1. 网页配置

参考zabbix\_server网页配置

1. 图像查看

查看zabbix\_agent监控数据