## Kubernetes使用手册

（含zookeeper、kafka服务部署）

环境为从hicloud申请的虚机，安装有centos7.2操作系统。

1. 安装Kubernetes

参考<http://rnd-mirrors.huawei.com/helps/centosmirror在/etc/yum.repos.d/CentOS-Base.repo>文件中设置yum源到rnd-mirrors.huawei.com

在.bashrc中设置代理：

export http\_proxy=http://h00406972:\_hifi2046@proxy.huawei.com:8080/

export https\_proxy=$http\_proxy

export no\_proxy=10.71.156.200,10.71.156.242,.huawei.com

安装kubernetes包：

yum makecache

sudo yum install etcd

sudo yum install kubernetes

修改/etc/kubernetes/apiserver文件admission行，删掉ServiceAccount：

KUBE\_ADMISSION\_CONTROL="--admission-control=NamespaceLifecycle,NamespaceExists,LimitRanger,SecurityContextDeny,ResourceQuota"

参考《Docker使用手册》配置http-proxy.conf和daemon.json，以访问私有仓库

1. 配置集群主节点（同时也是工作节点）

关闭防火墙

sudo systemctl disable firewalld

sudo systemctl stop firewalld

在/etc/kubernetes/apiserver文件中配置server信息：

KUBE\_API\_ADDRESS="--insecure-bind-address=100.101.88.200"

在/etc/kubernetes/config文件中配置master信息：

KUBE\_MASTER="--master=http://100.101.88.200:8080"

在/etc/kubernetes/kubelet文件中配置master和本地ip信息：

KUBELET\_API\_SERVER="--api-servers=http://100.101.88.200:8080"

KUBELET\_ADDRESS="--address=100.101.88.200"

KUBELET\_HOSTNAME="--hostname-override=100.101.88.200"

启动kubernetes服务：

sudo systemctl start etcd

sudo systemctl start docker

sudo systemctl start kube-apiserver

sudo systemctl start kube-controller-manager

sudo systemctl start kube-scheduler

sudo systemctl start kubelet

sudo systemctl start kube-proxy

检查安装是否成功：

kubectl --version

kubectl cluster-info

kubectl get ns

kubectl get nodes

浏览器可访问：http://100.101.88.200:8080

1. 配置集群工作节点

关闭防火墙

sudo systemctl disable firewalld

sudo systemctl stop firewalld

在/etc/kubernetes/config文件中配置master信息：

KUBE\_MASTER="--master=http://100.101.88.200:8080"

在/etc/kubernetes/kubelet文件中配置master和本地ip信息：

KUBELET\_API\_SERVER="--api-servers=http://100.101.88.200:8080"

KUBELET\_ADDRESS="--address=100.101.56.162"

KUBELET\_HOSTNAME="--hostname-override=100.101.56.162"

启动kubernetes服务：

sudo systemctl start docker

sudo systemctl start kubelet

sudo systemctl start kube-proxy

工具配置

kubectl config set-cluster default-cluster --server=http://100.101.88.200:8080/

kubectl config set-context default-context --cluster=default-cluster --user=default-admin

kubectl config use-context default-context

检查安装是否成功：

kubectl --version

kubectl cluster-info

kubectl get ns

kubectl get nodes

看到S1、S2两个节点都已加入集群

1. 配置Flannel网络（集群主节点）

在/etc/etcd/etcd.conf文件中设置etcd配置参数：

ETCD\_LISTEN\_CLIENT\_URLS="http://100.101.88.200:2379,http://localhost:2379"

ETCD\_ADVERTISE\_CLIENT\_URLS="http://100.101.88.200:2379"

在/etc/kubernetes/apiserver文件中配置etcd信息：

KUBE\_ETCD\_SERVERS="--etcd-servers=http://100.101.88.200:2379"

重启服务：

sudo systemctl daemon-reload

sudo systemctl restart etcd

确认etcd可用：

etcdctl --version

etcdctl ls /

ps –ef | grep etcd

设置flannel网络键值：

etcdctl mk /coreos.com/network/config '{"Network":"172.17.0.0/16", "SubnetMin": "172.17.1.0", "SubnetMax": "172.17.254.0"}'

Flannel软件包已安装

在集群所有节点的/etc/sysconfig/flanneld文件中修改配置：

FLANNEL\_ETCD\_ENDPOINTS="http://100.101.88.200:2379"

~~FLANNEL\_ETCD\_KEY="/coreos.com/network"~~

FLANNEL\_ETCD\_PREFIX="/coreos.com/network"

在集群所有节点启动服务：

sudo systemctl daemon-reload

sudo systemctl restart flanneld

sudo systemctl restart docker

sudo systemctl restart kubelet

sudo systemctl restart kube-proxy

在集群主节点启动服务：

sudo systemctl restart kube-apiserver

sudo systemctl restart kube-controller-manager

sudo systemctl restart kube-scheduler

查看子网配置：

etcdctl ls /coreos.com/network/subnets

etcdctl get /coreos.com/network/subnets/172.17.83.0-24

etcdctl get /coreos.com/network/subnets/172.17.76.0-24

ip addr

kubectl get po -o wide

1. 部署hello服务

制作定义文件hello.yaml，2副本

kubectl delete rc hello ；如果执行多次，需要先做清理

kubectl create -f hello.yaml

kubectl get po -o wide

kubectl describe po hello

分别在两台主机登录容器，检测网络连通性：

docker exec -it xxxx bash

ping 172.17.76.2

ping 172.17.83.4

1. 部署mysql / tomcat服务（不成功）

清除过往容器：

docker get pods -o wide

docker delete rc mysql

docker delete rc webapp

docker get service

docker delete service mysql

docker delete service mysqlo

docker delete service myweb

重建容器：

kubectl create -f mysql-rc.yaml

检查网络设置：

ps -ef | grep mysql

docker exec -it xxxx bash

ip a

docker cp /sbin/route xxxx:/sbin/

route -n

env

继续创建容器：

kubectl create -f mysql-svc.yaml

kubectl create -f myweb-rc.yaml

kubectl create -f myweb-svc.yaml

无法显示页面，报错：Error:com.mysql.jdbc.exceptions.jdbc4.CommunicationsException: Communications link failure The last packet sent successfully to the server was 0 milliseconds ago. The driver has not received any packets from the server.

1. 部署zk服务

Zookeeper为RC，开放内部服务；一个RC，包含3个POD

zk.yaml直接设置RC，结果是三个standalone节点：

kubectl create -f zk.yaml

echo stat | nc 172.17.76.4 2181

zk-ss.yaml设置StatefulSet，注意必须要serviceName；POD后缀自动为-0, -1, -2；依然是三个standalone节点：

kubectl create -f zk-ss.yaml

echo stat | nc 172.17.76.4 2181

kubectl exec zookeeper-0 -- hostname

kubectl exec zookeeper-0 – hostname -f

kubectl exec zookeeper-0 -- cat /var/lib/zookeeper/data/myid

修改yaml文件，不工作；采用官网教程yaml文件，不工作；可能是kubernetes版本过低

1. 部署kafka服务

Kafka为RC，开放外部服务