## Docker使用手册

（含zookeeper、kafka集群搭建）

环境为从hicloud申请的虚机，安装有ubuntu16.04操作系统。

1. 安装Docker

在.bashrc中设置proxy代理，并令其生效（用户需要具备外网访问权限）：

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

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

检查：

curl www.baidu.com

如果能显示html标签源码，则配置成功

在/etc/apt/sources.list中使用rnd-mirrors.huawei.com和rnd-mirrors-langfang.huawei.com的deb源：

deb http://rnd-mirrors.huawei.com/ubuntu/ xenial main multiverse restricted universe

deb http://rnd-mirrors.huawei.com/ubuntu/ xenial-backports main multiverse restricted universe

deb http://rnd-mirrors.huawei.com/ubuntu/ xenial-proposed main multiverse restricted universe

deb http://rnd-mirrors.huawei.com/ubuntu/ xenial-security main multiverse restricted universe

deb http://rnd-mirrors.huawei.com/ubuntu/ xenial-updates main multiverse restricted universe

deb-src http://rnd-mirrors.huawei.com/ubuntu/ xenial main multiverse restricted universe

deb-src http://rnd-mirrors.huawei.com/ubuntu/ xenial-backports main multiverse restricted universe

deb-src http://rnd-mirrors.huawei.com/ubuntu/ xenial-proposed main multiverse restricted universe

deb-src http://rnd-mirrors.huawei.com/ubuntu/ xenial-security main multiverse restricted universe

deb-src http://rnd-mirrors.huawei.com/ubuntu/ xenial-updates main multiverse restricted universe

deb http://rnd-mirrors-langfang.huawei.com/docker/apt/repo ubuntu-xenial main

为apt-get安装https访问包（非必须，如果deb源中没有https）：

sudo apt-get install apt-transport-https

安装docker：

sudo apt-get update

sudo apt-get install docker-engine

检查：

sudo docker version

可看到安装版本为docker 17.05.0-ce

2. 运行docker镜像

设置docker代理：

sudo mkdir -p /etc/systemd/system/docker.service.d

sudo vi /etc/systemd/system/docker.service.d/http-proxy.conf

文件内容：

[Service]

Environment="http\_proxy=http://h00406972:\_hifi2046@proxy.huawei.com:8080" "https\_proxy=http://h00406972:\_hifi2046@proxy.huawei.com:8080" "NO\_PROXY=registry.dfr.huawei.com,localhost,127.0.0.1,10.71.156.200,10.71.156.242,S0"

重启docker服务，令配置生效：

sudo systemctl daemon-reload  
sudo systemctl restart docker

检查远程仓库是否可用，如果网络不畅可能需要多试几次：

sudo docker pull hello-world

sudo docker run hello-world

3. 免sudo使用

将当前用户添加到docker组中：

sudo gpasswd -a $USER docker

检查，确认用户加入组中：

groups $USER

切换到新组？：

newgrp - docker

或添加权限：

sudo chmod a+rw /var/run/docker.sock

检查：

docker version

4. 访问私有仓库

仓库地址：10.71.156.242，用户：huangyinfei，密码：Huawei123

~~将密钥文件ca.crt放到/etc/docker/certs.d/目录下：~~

~~sudo mkdir -p /etc/docker/certs.d/10.71.156.242/~~

~~sudo cp ca.crt /etc/docker/certs.d/10.71.156.242/~~

登录用户（注意docker要设置NO\_PROXY）：

docker login 10.71.156.242 -u huangyinfei

docker pull 10.71.156.242/cloudha/zookeeper-install:0.1

4b. 自建私有仓库（非必须）

在S0主机上运行仓库：

docker pull registry

docker run -d -p 5000:5000 -v /home/hifi/data:/tmp/registry registry

确认私有仓库生效：

docker tag hello-world localhost:5000/hello-world

docker images

可以看到添加了本地镜像

docker push localhost:5000/hello-world

可以在<http://10.71.156.200:5000/v2/_catalog>页面看到镜像清单

在客户端主机上设置/etc/docker/daemon.json内容（仓库服务主机也要设置）：

{ "insecure-registries": ["10.71.156.200:5000","10.71.156.242","S0:5000","s0:5000"] }

~~在S0主机（10.71.156.200）对/etc/init/docker.conf作配置（hifi：不起作用）：~~

~~script~~

~~# modify these in /etc/default/$UPSTART\_JOB (/etc/default/docker)~~

~~DOCKERD=/usr/bin/dockerd~~

~~DOCKER\_OPTS="--insecure-registry registry.dfr.huawei.com"~~

~~if [ -f /etc/default/$UPSTART\_JOB ]; then~~

~~. /etc/default/$UPSTART\_JOB~~

~~fi~~

~~exec "$DOCKERD" $DOCKER\_OPTS --raw-logs --insecure-registry 10.71.156.200:5000~~

~~end script~~

重启docker服务，令配置生效：

~~sudo systemctl daemon-reload~~sudo systemctl restart docker

在S1主机试验远端镜像操作（注意S1主机上docker要设置NO\_PROXY）：

docker pull 10.71.156.200:5000/hello-world

docker tag zookeeper 10.71.156.200:5000/zookeeper

docker push 10.71.156.200:5000/zookeeper

5. 建立zk集群

前置条件：S0主机建立私有镜像仓库，S1主机安装docker成功

对S2-S5主机分发.bashrc、sources.list、http-proxy.conf、daemon.json

for i in {2..5}; do scp .bashrc s$i:.; done

for i in {2..5}; do scp /etc/apt/sources.list s$i:.; done

for i in {2..5}; do scp /etc/systemd/system/docker.service.d/http-proxy.conf s$i:.; done

for i in {2..5}; do scp /etc/docker/daemon.json s$i:.; done

在S2-S5每台主机上执行：

. .bashrc

sudo cp sources.list /etc/apt/

sudo apt-get update

sudo apt-get install docker-engine

sudo mkdir -p /etc/systemd/system/docker.service.d/

sudo cp http-proxy.conf /etc/systemd/system/docker.service.d/http-proxy.conf

sudo mkdir -p /etc/docker/

sudo cp daemon.json /etc/docker/daemon.json

sudo systemctl daemon-reload  
sudo systemctl restart docker

对S2-S5主机设置免sudo

sudo groupadd docker

sudo gpasswd -a $USER docker

newgrp - docker

拉取zk镜像（之前S0主机已经拉取zookeeper镜像并本地上传）：

docker pull 10.71.156.200:5000/zookeeper

docker tag 10.71.156.200:5000/zookeeper zookeeper

在S1-S5主机上启动zk节点（注意主机名大小写要与hosts文件一致）：

docker run --name z${HOSTNAME:1} -d --net=host --env ZOO\_MY\_ID=${HOSTNAME:1} --env "ZOO\_SERVERS=server.1=S1:2888:3888 server.2=S2:2888:3888 server.3=S3:2888:3888 server.4=S4:2888:3888 server.5=S5:2888:3888" zookeeper

检查：

echo stat | nc s1 2181

echo stat | nc s2 2181

echo stat | nc s3 2181

echo stat | nc s4 2181

echo stat | nc s5 2181

6. 建立kafka集群

在S0主机下载kafka镜像：

docker pull wurstmeister/kafka

docker tag wurstmeister/kafka localhost:5000/kafka

docker push localhost:5000/kafka

在S1-S5主机拉取kafka镜像：

docker pull s0:5000/kafka

docker tag s0:5000/kafka kafka

在S1-S5主机启动kafka：

docker run -d --name k${HOSTNAME:1} --net=host --env "KAFKA\_ZOOKEEPER\_CONNECT=S1:2181,S2:2181,S3:2181,S4:2181,S5:2181" kafka

确认kafka工作正常：

docker exec -it k1 bash（也可以不执行此句，直接在外部试验）

kafka-topics.sh --create --zookeeper S1:2181 --replication-factor 1 --partitions 1 --topic test

kafka-topics.sh --list --zookeeper S1:2181

S1主机开两个终端，直接在外部试验：

kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 3 --partitions 1 --topic my-topic

for j in `ls /usr/local/kafka/libs/\*.jar`; do export CLASSPATH=$CLASSPATH:$j; done

cd jwork

java HiConsumer

java HiProducer

200万条消息，原先在虚机上22秒发送/26秒接收完成，叠加容器后花费32秒发送/36秒接收完成，性能损失较多