所有使用到的系统环境

liunx、Redis、Mysql（主从）、activeMq、Dubbo+zookeper、java、docker、jenkins

所有环境都运行到docker容器中，服务系统使用linux

最终实现自动化部署及自动集群分布

docker容器地址

zookeeper

<https://c.163.com/dashboard#/m/mirrorRepo/detail/info/?repoId=76193&repoType=1>

docker pull hub.c.163.com/liu2160619/zookeeper:latest

启动 docker run -p 2181:2181 --name zookeeper-2181 --restart always -d 5291027d4199

redis

<https://c.163.com/dashboard#/m/mirrorRepo/detail/info/?repoId=76465&repoType=1>

docker pull hub.c.163.com/liu2160619/redis:latest

docker run -p 6379:6379 -v /opt/redis/redis.conf:/usr/local/etc/redis/redis.conf --name redis -d b6dddb991dfa redis-server /usr/local/etc/redis/redis.conf

docker run -p 6378:6379 -v /opt/redis/dev/redis.conf:/usr/local/etc/redis/redis.conf --name dev-redis -d b6dddb991dfa redis-server /usr/local/etc/redis/redis.conf

mysql主库

<https://c.163.com/dashboard#/m/mirrorRepo/detail/info/?repoId=76189&repoType=1>

docker pull hub.c.163.com/liu2160619/ms:latest

docker run -v /opt/mysql/data:/var/lib/mysql -p 3307:3306 --name zhumysql -e MYSQL\_ROOT\_PASSWORD=12345678 -d mysql:5.7

mysql从库

<https://c.163.com/dashboard#/m/mirrorRepo/detail/info/?repoId=76190&repoType=1>

docker pull hub.c.163.com/liu2160619/mysqlcong:latest

docker run -v /opt/mysql/data:/var/lib/mysql -p 3308:3306 --name cmysql -e MYSQL\_ROOT\_PASSWORD=12345678 -d mysql:5.7

安装rabbitmq  
docker pull hub.c.163.com/liu2160619/rabbitmq-3.6.11-man:latest

启动 rabbitmq  
docker run -d --hostname dev-rabbit --name prod-rabbit-mana -e RABBITMQ\_ERLANG\_COOKIE='devcookie' -e RABBITMQ\_DEFAULT\_USER=prodmq -e RABBITMQ\_DEFAULT\_PASS=prodmq -p 7001:15672 -p 5672:5672  67ab2eaad9fd

nginx

<https://c.163.com/dashboard#/m/mirrorRepo/detail/info/?repoId=76466&repoType=1>

docker pull hub.c.163.com/liu2160619/nginx:latest

运行命令:

docker run -d -p 80:80 -i -t -v /opt/nginx/nginx.conf:/etc/nginx/nginx.conf -v /opt/nginx/www:/usr/share/nginx/www --name nginx  dbdec5227d88

文件地址:

/usr/share/nginx/www

配置文件地址

/etc/nginx/nginx.conf

配置https需要映射KEY文件，及443端口

docker run -d -p 80:80 -p 443:443 -i -t -v /opt/nginx/nginx.conf:/etc/nginx/nginx.conf -v /opt/nginx/www:/usr/share/nginx/www -v /opt/nginx/cert:/etc/nginx/cert --name nginx  dbdec5227d88

Tomcat7

<https://c.163.com/dashboard#/m/mirrorRepo/detail/info/?repoId=79629&repoType=1>

docker pull hub.c.163.com/liu2160619/tomcat7:latest

docker run -d -p 8010:80 -i -t -v /opt/sanhu-api:/opt/tomcat/webapps/ -v /etc/localtime:/etc/localtime --name sanhu-api  --restart=always dbdec5227d88 sh  -c 'service tomcat start && tail -f /opt/tomcat/logs/catalina.out'

linux系统dubbo出现消费者无法链接到服务解决办法

查出 linux主机名

命令：hostname

修改hosts文件

命令： vim /etc/hosts

增加主机IP和主机名

39.106.148.123 iZ2zebukn5j9it9yx2dnscZ

docker基本命令

yum install docker  //安装docker  
yum install docker-io //安装docker  
service docker start//启动

一、docker内容  
1.查看镜像  
docker images //拥有镜像  
docker ps    //运行中容器  
docker ps -a  //停止中的容器  
docker restart XXXX号  //重启容器，停止的也可重启  
docker stop XXXX号  //停止容器

docker rename old容器名  new容器名   //容器改名

2.运行镜像  -p:端口映射主机端口映射到容器端口  -v文件挂载 将主机文件挂在容器文件中 sh -c运行命令  
docker run -d -p 8013:80 -p 8813:8886 -i -t -v /opt/wk/:/opt/tomcat/webapps/ 4f93bc400c72 sh  -c 'service tomcat start && tail -f /opt/tomcat/logs/catalina.out'

docker commit  bfe8dd9053ca centos7.2java:v0.2  //提交镜像/生成镜像  
docker attach <ContainerID>  //进入镜像  
docker search centos  //搜索网络镜像库  
//下载镜像  
docker pull centos

//删除镜像  
docker rmi  XXXXX  
//删除容器，容器是保存到硬盘中，不要留太多的容器  
docker rm XXXXX  
//保存镜像  
docker save 9610cfc68e8d > /home/myubuntu-save-1204.tar    
//装载镜像  
docker load < /home/myubuntu-save-1204.tar  
//将主机文件夹共享到镜像  
//假设我们需要将本机的/data 目录分享到Docker的/mnt 目录下, 我们可以通过这样的命令:  
$ docker run -v /data:/mnt -i -t image-id bash

启动镜像 启动时参数很多，以下都是启动语句，不代sh -c 执行语句的启动容器，可以使用attach进行到内部，使用sh -c就不能进去。  
docker run -d -i -t 98d35105a391  /bin/bash  
docker run -d -p 8010:80 -i -t -v /root/webapps/:/opt/tomcat/webapps/  676de7d9aaad /bin/bash

docker run -d -p 8010:80 -i -t -v /opt/wk/:/opt/tomcat/webapps/ 4f93bc400c72 sh  -c 'service tomcat start && tail -f /opt/tomcat/logs/catalina.out'

//系统自动启动、使用主机时间、共享目录  
docker run -d -p 8010:80 -p 8810:8886 -i -t -v /opt/wk/:/opt/tomcat/webapps/ -v /etc/localtime:/etc/localtime  --restart=always 4f93bc400c72 sh  -c 'service tomcat start && tail -f /opt/tomcat/logs/catalina.out'

docker run -d -p 8011:80 -p 8811:8886 -i -t -v /opt/wk/:/opt/tomcat/webapps/ -v /etc/localtime:/etc/localtime 4f93bc400c72 sh  -c 'service tomcat start && tail -f /opt/tomcat/logs/catalina.out'

docker run -d -p 8012:80 -p 8812:8886 -i -t -v /opt/wk/:/opt/tomcat/webapps/ 4f93bc400c72 sh  -c 'service tomcat start && tail -f /opt/tomcat/logs/catalina.out'

docker run -d -p 8013:80 -p 8813:8886 -i -t -v /opt/wk/:/opt/tomcat/webapps/ 4f93bc400c72 sh  -c 'service tomcat start && tail -f /opt/tomcat/logs/catalina.out'

执行容器命令  
docker exec -it 13178d504a1b /bin/sh  
想看日志  
docker logs -f 容器id

Docker            Requires: docker-ce-selinux >= 17.03.2.ce-1.el7.centos

yum install https://download.docker.com/linux/centos/7/x86\_64/stable/Packages/docker-ce-selinux-17.03.2.ce-1.el7.centos.noarch.rpm

#然后再安装 docker-ce-17.03.2.ce，就能正常安装

sudo yum install docker-ce-17.03.2.ce-1.el7.centos