**GMALL**

系统部署文档

# 一、linux虚拟机准备

## 1、购买阿里云-线上环境

## 2、自己搭建-测试环境

### 1、下载linux虚拟机镜像

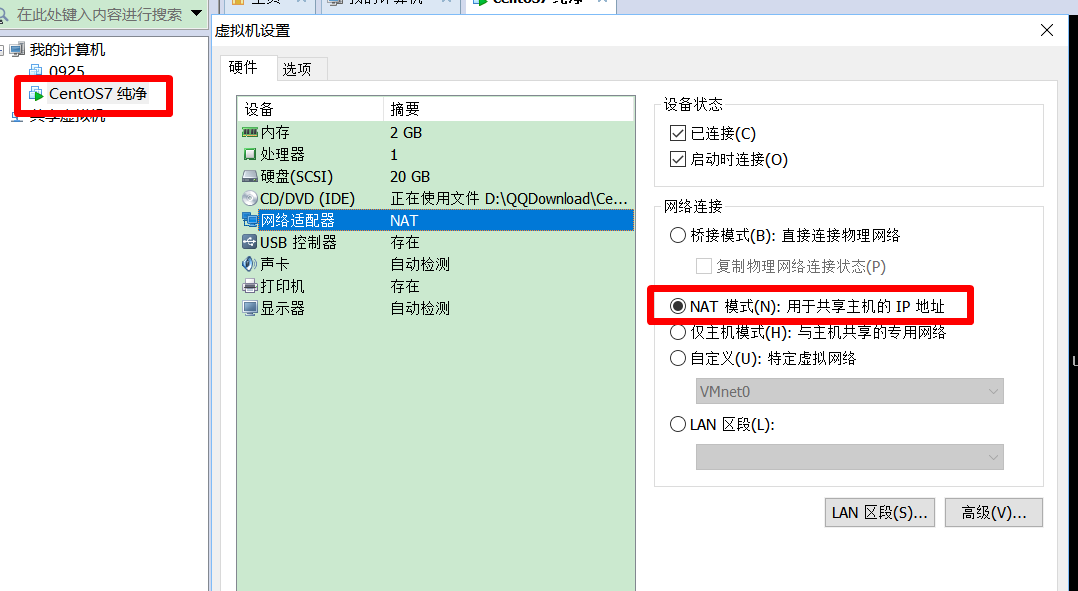
<http://mirrors.aliyun.com/centos/7/isos/x86_64/>

### 2、使用vmware安装

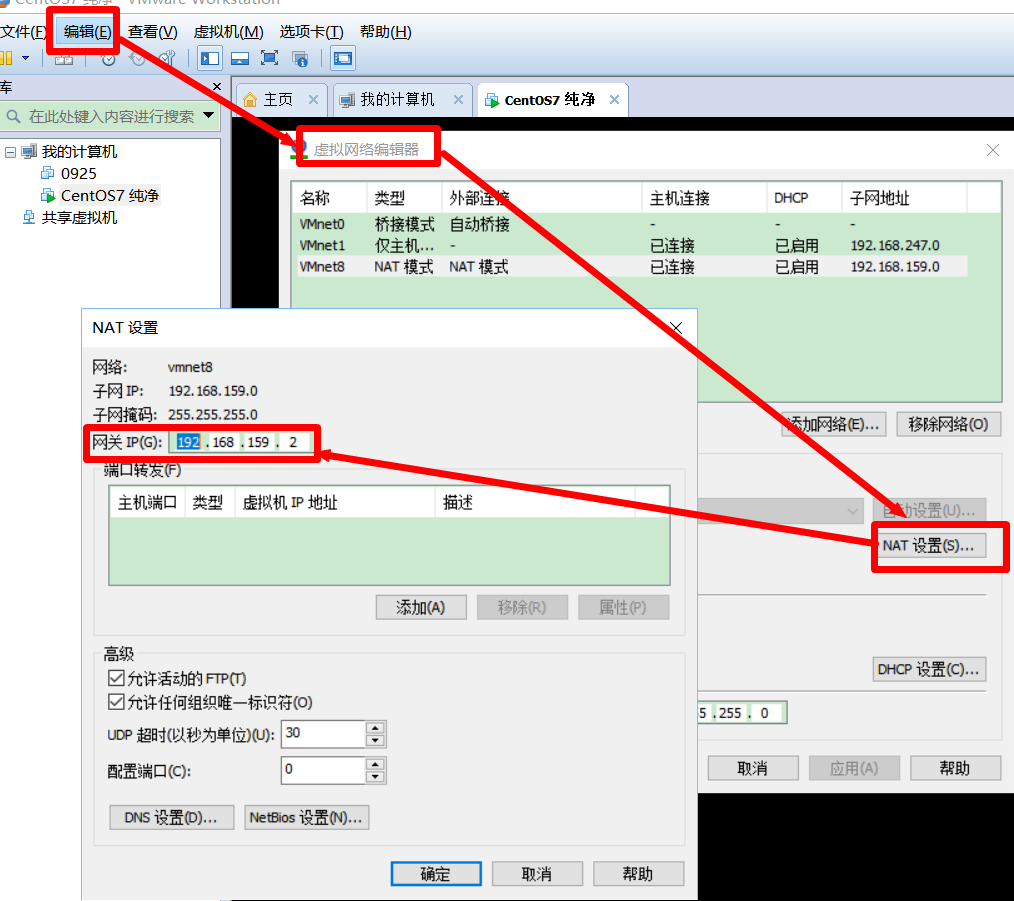
### 3、linux虚拟机初始化设置

#### 1、网络-固定ip设置

1）、使用NAT模式



2）、查看网关地址并记住



3）、修改网卡信息：

|  |
| --- |
| cd /etc/sysconfig/network-scripts  vi ifcfg-eno16777736 |
| 修改红色部分。并追加内容 |
| 重启网卡 |
| service network restart |
| 测试是否连通 |
| ping baidu.com（是否可以连通外网）  ip addr（查看本机ip地址是否正确） |

#### 2、基础-工具安装

##### 1、wget、vim

yum –y install wget

yum –y install vim\*

##### 2、更新yum源为阿里云

1、备份

mv /etc/yum.repos.d/CentOS-Base.repo /etc/yum.repos.d/CentOS-Base.repo.backup

2、下载新的CentOS-Base.repo 到/etc/yum.repos.d/

CentOS 5

wget -O /etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-5.repo

CentOS 6

wget -O /etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-6.repo

CentOS 7

wget -O /etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-7.repo

## 3、关闭防火墙

systemctl status firewalld （查看防火墙状态）

systemctl stop firewalld （停止防火墙）

systemctl disable firewalld（禁止开机启动）

# 二、Docker安装

<https://www.docker.com/products/docker-engine>

## 1、安装docker

检查内核版本，返回的值大于3.10即可。

uname -r

如果内核版本低，需要更新系统

yum update

或者安装官网进行完整安装：

操作系统 CentOS7.6 x64

Docker版本 18.06.1-ce

<https://docs.docker.com/install/linux/docker-ce/centos/>

1、Install required packages. yum-utils provides the yum-config-manager utility, and device-mapper-persistent-data and lvm2 are required by the devicemapper storage driver.

yum install -y yum-utils \

device-mapper-persistent-data \

lvm2

2、Use the following command to set up the **stable** repository.

yum-config-manager \

--add-repo \

https://download.docker.com/linux/centos/docker-ce.repo

3、查看版本

yum list docker-ce --showduplicates | sort -r

#### INSTALL DOCKER CE

yum -y install docker-ce-18.03.1.ce-1.el7.centos

systemctl start docker(启动docker)

systemctl enable docker(设为开机启动)

docker run hello-world

## 2、卸载docker

### 1.查询安装过的包

yum list installed | grep docker

docker.x86\_64 2:1.13.1-91.git07f3374.el7.centos @extras

docker-client.x86\_64 2:1.13.1-91.git07f3374.el7.centos @extras

docker-common.x86\_64 2:1.13.1-91.git07f3374.el7.centos @extras

### 2.删除安装的软件包

yum -y remove docker.x86\_64

yum -y remove docker-client.x86\_64

yum -y remove docker-common.x86\_64

### 3.删除镜像/容器等

rm -rf /var/lib/docker

## 3、docker镜像加速

阿里云，容器镜像服务

针对Docker客户端版本大于 1.10.0 的用户

您可以通过修改daemon配置文件/etc/docker/daemon.json来使用加速器

sudo mkdir -p /etc/docker

sudo tee /etc/docker/daemon.json <<-'EOF'

{

"registry-mirrors": ["https://82m9ar63.mirror.aliyuncs.com"]

}

EOF

sudo systemctl daemon-reload

sudo systemctl restart docker

## 4、docker-compose安装

<https://github.com/docker/compose/releases>

curl -L <https://github.com/docker/compose/releases/download/1.24.0-rc1/docker-compose> -`uname -s`-`uname -m` -o /usr/local/bin/docker-compose

chmod +x /usr/local/bin/docker-compose

# 三、MySQL安装

## 1、下载镜像文件

docker pull mysql:5.7

## 2、创建实例并启动

docker run -p 3306:3306 --name mysql \

-v /mydata/mysql/log:/var/log/mysql \

-v /mydata/mysql/data:/var/lib/mysql \

-v /mydata/mysql/conf:/etc/mysql \

-e MYSQL\_ROOT\_PASSWORD=root \

-d mysql:5.7

参数说明

* -p 3306:3306：将容器的3306端口映射到主机的3306端口
* -v /mydata/mysql/conf:/etc/mysql：将配置文件夹挂在到主机
* -v /mydata/mysql/log:/var/log/mysql：将日志文件夹挂载到主机
* -v /mydata/mysql/data:/var/lib/mysql/：将配置文件夹挂载到主机
* -e MYSQL\_ROOT\_PASSWORD=root：初始化root用户的密码

MySQL配置

vim /mydata/mysql/conf/my.cnf

[client]

default-character-set=utf8

[mysql]

default-character-set=utf8

[mysqld]

init\_connect='SET collation\_connection = utf8\_unicode\_ci'

init\_connect='SET NAMES utf8'

character-set-server=utf8

collation-server=utf8\_unicode\_ci

skip-character-set-client-handshake

## 3、通过容器的mysql命令行工具连接

docker exec -it mysql mysql -uroot -proot

## 4、设置root远程访问

grant all privileges on \*.\* to 'root'@'%' identified by 'root' with grant option;

flush privileges;

## 5、进入容器文件系统

docker exec -it mysql /bin/bash

## 6、解决MySQL连接慢的问题

在配置文件中加入如下，并重启mysql

[mysqld]

skip-name-resolve

解释：

skip-name-resolve：跳过域名解析

# 四、Redis安装

## 1、下载镜像文件

docker pull redis:3.2

## 2、创建实例并启动

docker run -p 6379:6379 --name redis -v /mydata/redis/data:/data -d redis:3.2 redis-server --appendonly yes

## 3、使用redis镜像执行redis-cli命令连接

docker exec -it redis redis-cli

# 五、Nginx安装

## 1、下载镜像文件

docker pull nginx:1.10

## 2、创建实例并启动

docker run -p 80:80 --name nginx \

-v /mydata/nginx/html:/usr/share/nginx/html \

-v /mydata/nginx/logs:/var/log/nginx \

-d nginx:1.10

docker run -p 80:80 --name nginx -d nginx:1.10



## 3、修改nginx配置

1. 将容器内的配置文件拷贝到当前目录：docker container cp nginx:/etc/nginx .
2. 修改文件名称：mv nginx conf 把这个conf移动到/mydata/nginx下
3. 终止容器：docker stop nginx
4. 执行命令删除原容器：docker rm $ContainerId
5. 执行以下命令：

docker run -p 80:80 --name nginx \

-v /mydata/nginx/html:/usr/share/nginx/html \

-v /mydata/nginx/logs:/var/log/nginx \

-v /mydata/nginx/conf:/etc/nginx \

-d nginx:1.10

# 六、Rabbitmq安装

## 1、下载镜像文件

docker pull rabbitmq:management

## 2、创建实例并启动

docker run -d --name rabbitmq --publish 5671:5671 \

--publish 5672:5672 --publish 4369:4369 --publish 25672:25672 --publish 15671:15671 --publish 15672:15672 \

rabbitmq:management

注：

4369 -- erlang发现口

5672 --client端通信口

15672 -- 管理界面ui端口

25672 -- server间内部通信口

## 3、测试

在web浏览器中输入地址：http://虚拟机ip:15672/

输入默认账号: guest 密码: guest

# 七、Mongodb安装

## 1、下载镜像文件

docker pull mongo:3.2

## 2、创建实例并运行

docker run -p 27017:27017 --name mongo -v /mydata/mongo/db:/data/db -d mongo:3.2

## 3、使用mongo命令进入容器

docker exec -it mongo mongo

## 4、安装mongo客户端

<https://robomongo.org/>

Robo 3T

# 八、ElasticSearch+Logstash+Kibana安装

## 1、下载镜像文件

docker pull elasticsearch:5.6.11

docker pull kibana:5.6.11

docker pull logstash:5.6.15

## 2、创建实例

### 1、ElasticSearch

mkdir -p /mydata/elasticsearch/config

mkdir -p /mydata/elasticsearch/data

echo "http.host: 0.0.0.0" >> /mydata/elasticsearch/config/elasticsearch.yml

docker run --name elasticsearch -p 9200:9200 -p 9300:9300 \

-e "discovery.type=single-node" \

-e ES\_JAVA\_OPTS="-Xms256m -Xmx256m" \

-v /mydata/elasticsearch/config/elasticsearch.yml:/usr/share/elasticsearch/config/elasticsearch.yml \

-v /mydata/elasticsearch/data:/usr/share/elasticsearch/data -d elasticsearch:5.6.11

特别注意：

-e ES\_JAVA\_OPTS="-Xms256m -Xmx256m" \ 测试环境下，设置ES的初始内存和最大内存，否则导致过大启动不了ES

### 2、Kibana

docker run --name kibana -e ELASTICSEARCH\_URL=http://192.168.159.130:9200 -p 5601:5601 \

-d kibana:5.6.11

### 3、Logstash

1）、在mydata/logstash中创建logstash.conf文件：文件内容如下

input {  
 tcp {  
 port => 4560  
 codec => json\_lines  
 }  
}  
output{  
 elasticsearch {

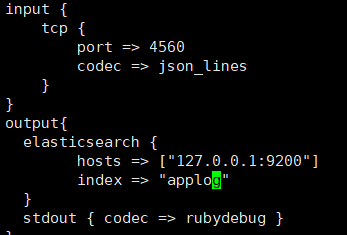
hosts => ["192.168.159.130:9200"]

index => "applog"

}  
 stdout { codec => rubydebug }  
}

注意：

hosts一定不要写127或者localhost；这样docker容器内部127没有es实例，连不上



docker run -d -p 4560:4560 \

-v /mydata/logstash/logstash.conf:/etc/logstash.conf \

--link elasticsearch:elasticsearch \

--name logstash logstash:5.6.15 \

logstash -f /etc/logstash.conf

安装插件：

<https://github.com/logstash-plugins>

docker exec –it logstash /bin/bash （进入容器内容）

cd /usr/share/logstash/bin （可以whereis logstash找到这个位置）

logstash-plugin install logstash-codec-json\_lines

我们对容器做了改变；为了以后方便，可以将这个容器再打包成新的镜像；

可以将修改好的容器制作为镜像，方便下次使用（也可直接推送到镜像仓库）；

docker commit logstash logstash\_gmall:0.0.1

上传Docker镜像到阿里云（阿里云仓库管理里面有步骤）：

* 命令行登录到阿里云的doker仓库，--username为阿里云的用户名

docker login --username=forsumlove registry.cn-hangzhou.aliyuncs.com

* 为本地镜像添加tag

docker tag [ImageId] registry.cn-hangzhou.aliyuncs.com/lfy/gmall:[镜像版本号]

docker tag fb11fcb28ea0 registry.cn-hangzhou.aliyuncs.com/lfy/gmall:logstatsh\_0.0.1

* push到docker仓库

docker push registry.cn-hangzhou.aliyuncs.com/lfy/gmall:[镜像版本号]

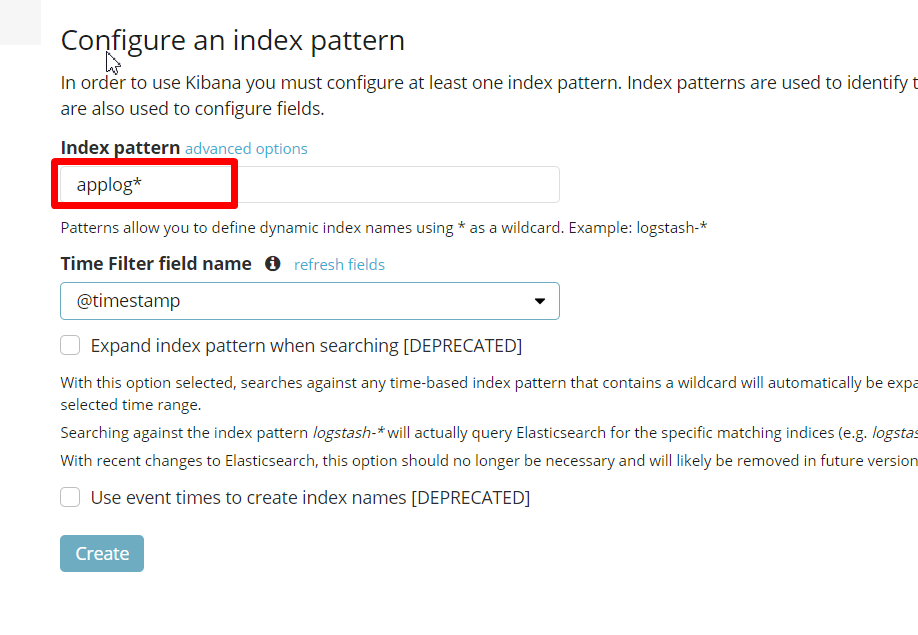
docker push registry.cn-hangzhou.aliyuncs.com/lfy/gmall:logstatsh\_0.0.1

登录阿里云查看已上传的镜像列表：https://cr.console.aliyun.com/#/imageList

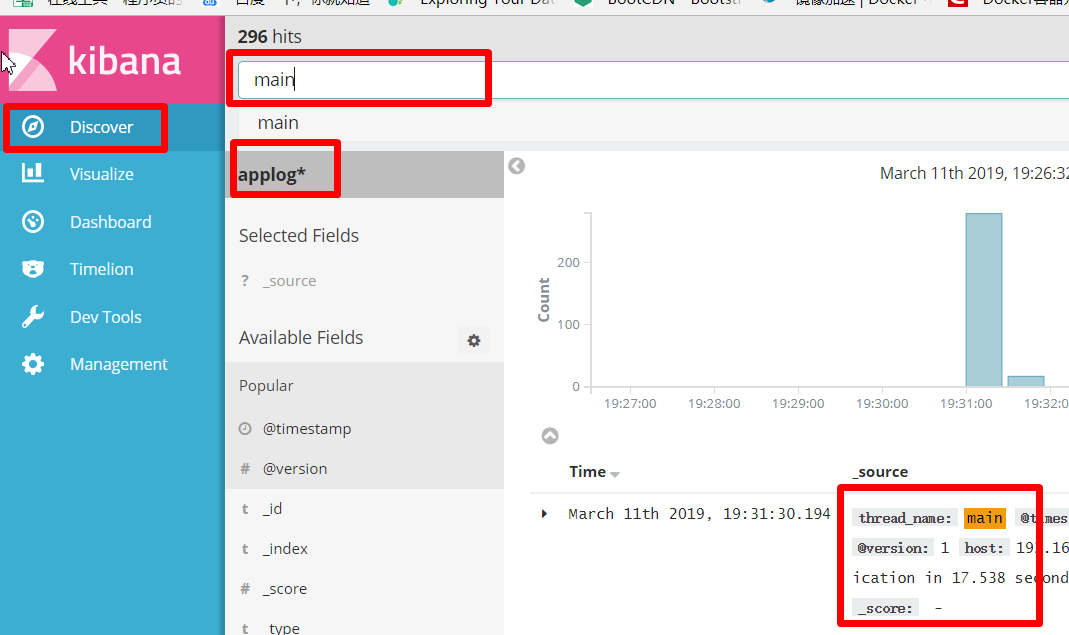
* 从阿里云自己的仓库拉取

docker pull registry.cn-hangzhou.aliyuncs.com/lfy/gmall:[镜像版本号]

### 4、建立索引



### 5、日志检索



# 九、Zookeeper安装

## 1、下载Zookeeper镜像

docker pull zookeeper

## 2、启动容器并添加映射

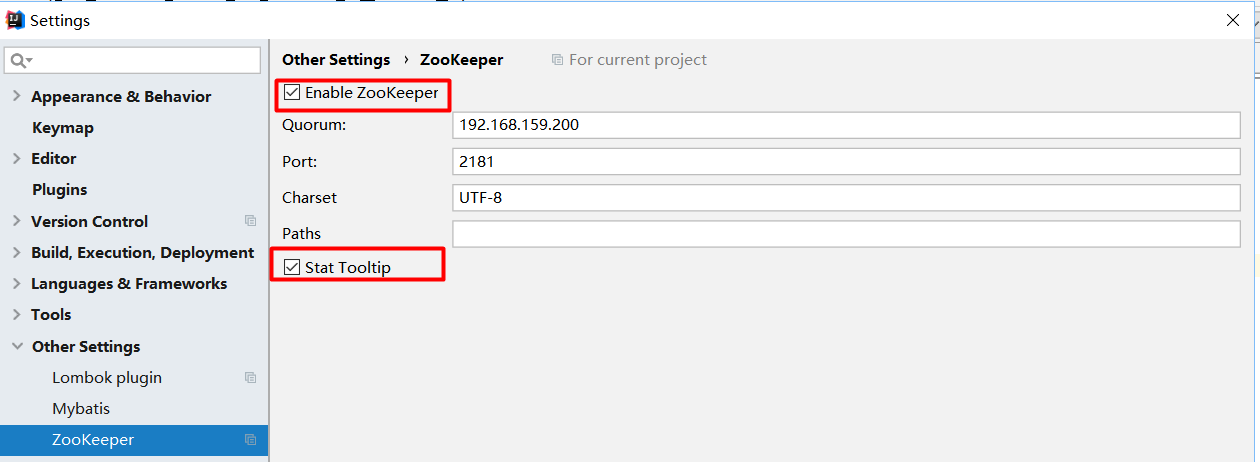
docker run -p 2181:2181 --privileged=true --name zookeeper -d zookeeper

## 3、idea安装zk插件

idea提供了一个Zookeeper插件，以供连接Zookeeper服务中心和查看内容

打开idea –》 Settings -》Plugins,搜索Zoo进行下载安装

测试zookeeper连接



# 十、Apollo安装(携程)

<https://github.com/ctripcorp/apollo>

## 1、下载镜像（非官方）

<https://hub.docker.com/r/idoop/docker-apollo>

docker pull idoop/docker-apollo

## 2、配置数据库

<https://github.com/ctripcorp/apollo/wiki/%E5%88%86%E5%B8%83%E5%BC%8F%E9%83%A8%E7%BD%B2%E6%8C%87%E5%8D%97#21-%E5%88%9B%E5%BB%BA%E6%95%B0%E6%8D%AE%E5%BA%93>

### 1）、创建ApolloPortalDB 并导入如下sql

<https://github.com/ctripcorp/apollo/blob/master/scripts/db/migration/portaldb/V1.0.0__initialization.sql>

### 2）、创建ApolloConfigDBDev 并导入如下sql

<https://github.com/ctripcorp/apollo/blob/master/scripts/db/migration/configdb/V1.0.0__initialization.sql>

## 3、使用 Docker Compose 启动

1、创建 docker-compose.yaml 文件

|  |
| --- |
| version: '2'  services:  apollo:  image: idoop/docker-apollo:latest  # portal若出现504错误,则将网络模式改为host. host模式下如果想改端口,参考下方修改端口的环境变量  network\_mode: "host"  # ports:  # - 8070:8070  volumes:  # 如果需要查看日志,挂载容器中的/opt路径出来即可.  - ./logs:/opt  # 如果portal需要开启ldap或ad域验证,须挂载此ldap配置文件  # - ./application-ldap.yml:/apollo-portal/config/application-ldap.yml:ro  environment:  # 开启Portal,默认端口: 8070  PORTAL\_DB: jdbc:mysql://192.168.159.200:3306/ApolloPortalDB?characterEncoding=utf8  PORTAL\_DB\_USER: root  PORTAL\_DB\_PWD: root  # 如果portal需要开启ldap或ad域验证,须设置该环境变量为TRUE  #PORTAL\_LDAP: "TRUE"  # 开启dev环境, 默认端口: config 8080, admin 8090  DEV\_DB: jdbc:mysql:// 192.168.159.200:3306/ApolloConfigDB?characterEncoding=utf8  DEV\_DB\_USER: root  DEV\_DB\_PWD: root  # 开启fat环境, 默认端口: config 8081, admin 8091  # FAT\_DB: jdbc:mysql://192.168.1.28:3306/ApolloConfigDBFat?characterEncoding=utf8  # FAT\_DB\_USER: root  # FAT\_DB\_PWD: toor  # 可修改端口.  DEV\_CONFIG\_PORT: 8050  DEV\_ADMIN\_PORT: 8051  # 指定远程uat地址  #UAT\_URL: http://192.168.1.2:8080  # 指定远程pro地址  #PRO\_URL: <http://www.example.com:8080> |
| 整理可用的配置如下  version: '2'  services:  apollo:  image: idoop/docker-apollo:latest  network\_mode: "host"  volumes:  - ./logs:/opt  environment:  PORTAL\_DB: jdbc:mysql://192.168.159.200:3306/ApolloPortalDB?characterEncoding=utf8  PORTAL\_DB\_USER: root  PORTAL\_DB\_PWD: root      DEV\_DB: jdbc:mysql://192.168.159.200:3306/ApolloConfigDB?characterEncoding=utf8  DEV\_DB\_USER: root  DEV\_DB\_PWD: root |

2、启动服务

docker-compose up –d

3、如果启动出错，查看容器日志

docker logs 容器id

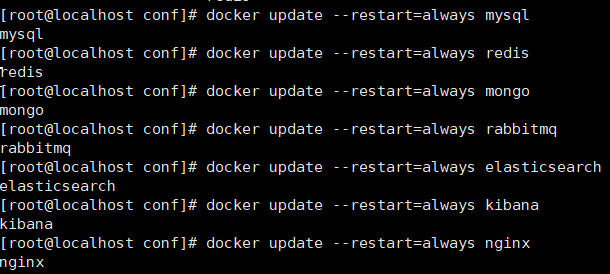
## 4、访问apollo

<http://192.168.159.200:8070>

账号 apollo 密码 admin

# 十一、Docker容器开机自启

docker update --restart=always xxx



docker update --restart=always mysql

docker update --restart=always mongo

docker update --restart=always rabbitmq

docker update --restart=always elasticsearch

docker update --restart=always kibana

docker update --restart=always logstash

docker update --restart=always nginx

docker update --restart=always redis

docker update --restart=always zookeeper

docker update --restart=always opt\_apollo\_1