**目录**

[**一、DB和Nginx镜像使用 2**](#_Toc531362770)

[**1. mysql镜像 2**](#_Toc531362771)

[**2. redis镜像 4**](#_Toc531362772)

[**3. nginx镜像 4**](#_Toc531362773)

[**二、swarm集群和DB、nginx镜像结合使用 7**](#_Toc531362774)

[**1. 创建集群Leader 7**](#_Toc531362775)

[**2. 添加其他结点 7**](#_Toc531362776)

[**3. 创建跨主机overlay网络 7**](#_Toc531362777)

[**4. 修改Dockerfile和添加配置中心配置 7**](#_Toc531362778)

[**4.1 cms、system等模块的Dockerfile修改为 7**](#_Toc531362779)

[**4.2 配置中心添加cms-swarm.properties、system-swarm.properties等 7**](#_Toc531362780)

[**5. 编写yml 8**](#_Toc531362781)

[**6. 所有结点环境同步 11**](#_Toc531362782)

[**7 运行镜像 12**](#_Toc531362783)

# 一、DB和Nginx镜像使用

1. **mysql镜像**

目录结构



1. **创建my.cnf，准备sql文件**

# Copyright (c) 2017, Oracle and/or its affiliates. All rights reserved.  
#  
# This program is free software; you can redistribute it and/or modify  
# it under the terms of the GNU General Public License as published by  
# the Free Software Foundation; version 2 of the License.  
#  
# This program is distributed in the hope that it will be useful,  
# but WITHOUT ANY WARRANTY; without even the implied warranty of  
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
# GNU General Public License for more details.  
#  
# You should have received a copy of the GNU General Public License  
# along with this program; if not, write to the Free Software  
# Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA  
   
#  
# The MySQL Server configuration file.  
#  
# For explanations see  
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html  
   
[mysqld]  
pid-file = /var/run/mysqld/mysqld.pid  
socket = /var/run/mysqld/mysqld.sock  
datadir = /var/lib/mysql  
secure-file-priv= NULL  
# Disabling symbolic-links is recommended to prevent assorted security risks  
symbolic-links=0  
   
# Custom config should go here  
!includedir /etc/mysql/conf.d/

**my.cnf和sql文件放在/u06/root/pkg/docker-compose/mysql/目录下；**

**my.cnf配置和存放目录可以自行更改，但一些配置需与启动命令相对应。**

* 1. **启动容器**

**命令：**

sudo docker run --name bhmall-mysql -v /u06/root/pkg/docker-compose/mysql/my.cnf:/etc/mysql/my.cnf -v /data/mysql/logs:/logs -v /data/mysql/data:/var/lib/mysql -p 44417:3306 -e MYSQL\_ROOT\_PASSWORD=zhiyeDEV654321 -d mysql

**注：为了数据持久化，需要使用-v 选项将docker目录文件挂载到本地，因此每次删除容器后，都使用同一命令重新启动**

**“：”号前面的路径为本地路径，后面的路径为docker容器内部路径（固定不变）**

* 1. **导入SQL文件**

1. **将宿主机上的数据sql复制到容器的文件下(/opt/):**

**docker cp 目录/文件名.sql bhmall-mysql:/opt/文件名.sql**

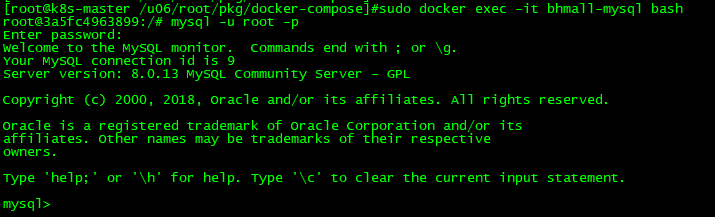
1. **进入容器：**

**docker exec -it bhmall-mysql bash**

1. **进入mysql：（注意复制粘贴可能会乱码）**

**mysql –u root –p**

**输入密码**



1. **创建并使用数据库**

**create database 数据库名;**

**use 数据库名**

1. **导入SQL**

**source /opt/文件名.sql**

1. **退出**

**quit**

**exit**

1. **redis镜像**

**目录结构**



#### 2.1 修改redis.conf配置文件

**redis.conf配置文件可从官网下载后修改**

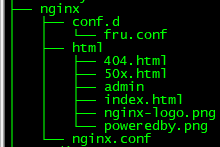
**注释文件中的bind 127.0.0.1**

#### 2.2 启动redis

sudo docker run -p 6379:6379 --name bhmall-redis -v /u06/root/pkg/docker-compose/redis/redis.conf:/etc/redis/redis.conf -v /data/redis/data:/data -d redis redis-server /etc/redis/redis.conf --appendonly yes

1. **nginx镜像**

**目录结构**



#### 3.1 在nginx创建conf.d目录、html目录

**mkdir conf.d**

**mkdir html**

**3.2 创建nginx.conf配置文件**

user nginx;  
worker\_processes 1;  
  
error\_log /var/log/nginx/error.log warn;  
pid /var/run/nginx.pid;  
  
events {  
 worker\_connections 1024;  
}  
  
http {  
 include /etc/nginx/mime.types;  
 default\_type application/octet-stream;  
  
 log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '  
 '$status $body\_bytes\_sent "$http\_referer" '  
 '"$http\_user\_agent" "$http\_x\_forwarded\_for"';  
  
 access\_log /var/log/nginx/access.log main;  
  
 sendfile on;  
 *#tcp\_nopush on;* keepalive\_timeout 65;  
  
 *#gzip on;* root /usr/share/nginx/html;  
 include /etc/nginx/conf.d/\*.conf;  
}

* 1. **在conf.d创建fru.conf配置文件**

server {  
*# listen 443;  
# server\_name frutest.zhiyesoft.cn;  
   
# ssl on;  
  
# ssl\_certificate conf.d/certFile/1\_frutest.zhiyesoft.cn\_bundle.crt;  
# ssl\_certificate\_key conf.d/certFile/2\_frutest.zhiyesoft.cn.key;  
  
# ssl\_session\_cache shared:SSL:1m;  
  
# ssl\_session\_timeout 5m;  
  
# ssl\_ciphers HIGH:!aNULL:!MD5;  
# ssl\_prefer\_server\_ciphers on;* location / {  
  
 }  
  
   
 location ~ ^(/m)(\S)+(-service) {  
 proxy\_pass http://localhost:10760;  
 proxy\_connect\_timeout 60000s;  
 proxy\_send\_timeout 60000s;  
 proxy\_read\_timeout 60000s;  
 send\_timeout 60000s;  
 }  
  
 location ~ ^(\S)+(-service) {  
 proxy\_pass http://localhost:8760;  
 proxy\_connect\_timeout 60000s;  
 proxy\_send\_timeout 60000s;  
 proxy\_read\_timeout 60000s;  
 send\_timeout 60000s;  
 }  
  
 error\_page 404 /404.html;  
 location = /40x.html {   
 }  
  
 error\_page 500 502 503 504 /50x.html;  
 location = /50x.html {  
 }  
  
}

* 1. **将静态资源存放在html目录下**
  2. **启动命令**

sudodocker run --name bhmall-nginx -d -p 80:80 -v /u06/root/pkg/mytest/nginx/html:/usr/share/nginx/html -v /u06/root/pkg/mytest/nginx/nginx.conf:/etc/nginx/nginx.conf -v /data/nginx/logs:/var/log/nginx –v /u06/root/pkg/mytest/nginx/conf.d:/etc/nginx/conf.d -d nginx

# 二、swarm集群和DB、nginx镜像结合使用

1. **创建集群Leader**

**docker swarm init**

1. **添加其他结点**

**在Leader机器上使用命令docker swarm join-token manager查看token**

**在其他结点的机器上运行命令获得的token**

**docker swarm join \**

**--token SWMTKN-1-3v652jtcfnca7ae8yl1tj7e06687spmpxdge5mj2hkha3xlcik-51v7xbp7f4enhmqim5jgxniku \**

**192.168.220.133:2377**

1. **创建跨主机overlay网络**

**sudo docker network create -d overlay --subnet=10.10.0.0/16 --gateway=10.10.0.254 --attachable=true bhmall-overlay**

1. **修改Dockerfile和添加配置中心配置**

**4.1 cms、system等模块的Dockerfile修改为**

FROM frolvlad/alpine-oraclejdk8:CST  
VOLUME /u06/ceta/pkg/bhmall-admin-root/share  
ADD \*.jar app.jar  
#RUN sh -c 'touch /app.jar'  
ENV JAVA\_OPTS="-Xms256m -Xmx512m"  
ENV PROFILE="dev"  
ENV HOST="localhost"  
ENV MYSQL=""  
ENV REDIS=""  
ENTRYPOINT [ "sh", "-c", "java $JAVA\_OPTS -Djava.security.egd=file:/dev/./urandom -jar /app.jar --spring.cloud.config.profile=$PROFILE $MYSQL $REDIS --eureka.client.serviceUrl.defaultZone=http://$HOST:8761/eureka/" ]

**主要改变为添加了ENV MYSQL=“” 和ENV REDIS=“”**

**4.2 配置中心添加cms-swarm.properties、system-swarm.properties等**

**配置与dev相同，修改下面两句**

spring.datasource.url=jdbc:mysql://${MYSQL}:44417/fru\_bhm\_shop?useUnicode=true&characterEncoding=UTF-8

spring.redis.host=${REDIS}

1. **编写yml**

#### 5.1 高可用eureka集群admin\_eureka.yml

**version:** "3"  
**services:  
 admin\_server:  
 image:** bhmall-admin-server:0.0.1  
 **deploy:  
 replicas:** 2  
 **networks:  
 bhmall-overlay:  
 aliases:** - Aserver  
 **ports:** - "8761:8761"  
 **environment:** - HOST=Aserver  
 - PROFILE=swarm  
  
**networks:  
 bhmall-overlay:  
 external:  
 name:** bhmall-overlay

**replicas为副本数量，aliases为overlay网络中的服务别名，作用与admin\_server一样，跨主机的容器间可以通过这个别名访问服务(或ip)**

**使用 HOST=Aserver 修改 server 中 defaultZone 为 http://Aserver:8761/eureka/ 这样启动的多个 server 会相互注册，成为 eureka 集群**

**5.2 config注册中心 admin\_config.yml**

**version:** "3"  
**services:  
 admin\_config:  
 image:** bhmall-admin-config:0.0.1  
 **deploy:  
 replicas:** 2  
 **ports:** - "8762:8762"  
 **networks:** - bhmall-overlay  
 **environment:** - HOST=Aserver  
 - PROFILE=swarm  
  
**networks:  
 bhmall-overlay:  
 external:  
 name:** bhmall-overlay

* 1. **admin\_zuul.yml**

**version:** "3"  
**services:  
 admin\_zuul:  
 image:** bhmall-admin-zuul:0.0.1  
 **deploy:  
 replicas:** 1  
 **resources:  
 limits:  
 memory:** 256M  
 **reservations:  
 memory:** 128M  
 **networks:** - bhmall-overlay  
 **ports:** - "8760:8760"  
 **environment:** - HOST=admin\_erver  
 - PROFILE=swarm

**networks:  
 bhmall-overlay:  
 external:  
 name:** bhmall-overlay

* 1. **cms和system模块admin\_cms.yml、admin\_system.yml**

**此处的MYSQL=bhmall-mysql对应的是mysql.yml 中的aliases的参数**

**version:** "3"  
**services:  
 admin\_cms:  
 image:** bhmall-admin-cms:0.0.1  
 **deploy:  
 replicas:** 2  
 **resources:  
 limits:  
 memory:** 512M  
 **reservations:  
 memory:** 256M  
 **networks:** - bhmall-overlay  
 **ports:** - "8765:8765"  
 **environment:** - HOST=Aserver  
 - PROFILE=swarm  
 - MYSQL=bhmall-mysql  
 - REDIS=bhmall-redis  
  
**networks:  
 bhmall-overlay:  
 external:  
 name:** bhmall-overlay

**version:** "3"  
**services:  
 admin\_system:  
 image:** bhmall-admin-system:0.0.1  
 **deploy:  
 replicas:** 2  
 **resources:  
 limits:  
 memory:** 512M  
 **reservations:  
 memory:** 256M  
 **networks:** - bhmall-admin-overlay  
 **ports:** - "8775:8775"  
 **environment:** - HOST=Aserver  
 - PROFILE=swarm  
 - MYSQL=bhmall-mysql  
 - REDIS=bhmall-redis  
  
**networks:  
 bhmall-admin-overlay:  
 external:  
 name:** bhmall-admin-overlay

1. **所有结点环境同步**
   1. **使用scp复制mytest目录下所有文件其他结点机器**

**scp -r -P 41133 /u06/root/pkg/mytest/ root@192.168.220.134:/u06/root/pkg/mytest/**

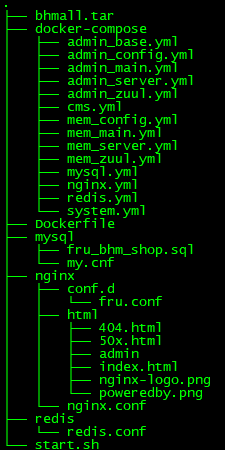
**scp -r -P 41133 /u06/root/pkg/mytest/ root@192.168.220.135:/u06/root/pkg/mytest/**

**ssh -p 41133 root@192.168.220.134  "docker load -i /u06/root/pkg/mytest/bhmall.tar"**

**ssh -p 41133 root@192.168.220.135  "docker load -i /u06/root/pkg/mytest/bhmall.tar"**

**注意：所有机器文件路径必须一致**

**这里的mytest目录结构和内容：**



**①bhmall.tar存放的是项目所有镜像**

**②docker-compose存放的是docker compose的yml文件，其中包含第5部中的yml和mysql、nginx、redis的镜像启动文件**

**③二级目录mysql、nginx、redis存放的是启动时需要的配置文件**

* 1. **所有机器解压bhmall.tar**

**docker load -i bhmall.tar**

**机器镜像同步完毕**

1. **运行镜像**

**在mytest目录下**

**运行mysql:**

**docker stack deploy -c docker-compose/mysql.yml bhmall**

**第一次运行mysql需要导入sql数据**

**运行redis：**

**docker stack deploy -c docker-compose/redis.yml bhmall**

**运行nginx**

**docker stack deploy -c docker-compose/nginx.yml bhmall**

**运行server**

**docker stack deploy -c docker-compose/admin\_server.yml bhmall**

**运行config**

**docker stack deploy -c docker-compose/admin\_config.yml bhmall**

**运行cms**

**docker stack deploy -c docker-compose/admin\_cms.yml bhmall**

**运行system**

**docker stack deploy -c docker-compose/admin\_system.yml bhmall**