

springboot 项目 docker容器实现 jar 与 mysql redis容器互联发布应用

springboot 项目打包后，生成JAR包后，要么通过服务的形式发布，要么通过打WAR包放到TOMCAT中的发布，这里我们介绍通过使用容器化的技术，实现JAR包的应用容器与mysql容器和redis容器来进行应用的发布

[原创，转载就注明出处]

一 mysql 容器

```
docker run -d -p 0.0.0.0:3307:3306 --network my-net --restart=always -e MYSQL_ROOT_PASSWORD="whyI@20200810" --name=mysql57 -v /etc/localtime:/etc/localtime -v $PWD/my.cnf:/etc/my.cnf -v $PWD/data:/var/lib/mysql mysql:5.7 --default-authentication-plugin=mysql_native_password
```

一般使用mysql 5.7这个版本，这里需要注意的是：一定要将本地的localtime映射到容器中，使容器的时间与宿主机的时间同步，【-v /etc/localtime:/etc/localtime 】

\$PWD/my.cnf:/etc/my.cnf 这是将当前目录下的mysql的配置文件映射到容器的mysql中去

创建好mysql 容器后，可以通过navicate尝试远程连接，使用设置的用户名和密码，这里我们后面是通过内部容器互联来连接mysql 的，在我们的应用联接的yaml配置中，后面你会发现，仍然使用的是3306端口，而不是这里映射的宿主机的3307，在这里，我特意，设置一个不一样的端口好区分后面的联接配置

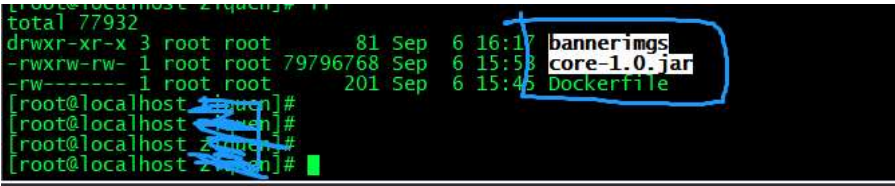
二 redis容器

```
docker run --name redis -p 6380:6379 redis --requirepass 123456
```

这里也故意使宿主机的映射端口与redis的默认端口不一样，以便后续方便理解区分

三 发布应用

1 首先要打jar包的镜像



在当前目录下 创建Dockerfile，Dockerfile的内容如下

```
FROM java:8
MAINTAINER wy
ADD core-1.0.jar myjar.jar
ADD ./bannerimgs ./bannerimgs
EXPOSE 9080
ENTRYPOINT ["java","-jar","myjar.jar","--spring.profiles.active=prod","--server.port=9080"]

ADD core-1.0.jar myjar.jar 意思是把当前目录下的core-1.0.jar 拷贝到容器中去 并命名为myjar.jar

ENTRYPOINT ["java","-jar","myjar.jar","--spring.profiles.active=prod","--server.port=9080"] 容器镜像启动脚本 这里可以按具体的要求自己添加springboot javar -jar 相关参数
```

2. 应用启用的配置文件，如何配置

公告

昵称: freewsf  
园龄: 9年2个月  
粉丝: 1  
关注: 5  
+加关注

2023年5月				
<	日	一	二	三
	30	1	2	3
	7	8	9	10
	14	15	16	17
	21	22	23	24
	28	29	30	31
	4	5	6	7

搜索

常用链接

- 我的随笔
- 我的评论
- 我的参与
- 最新评论
- 我的标签

随笔分类

- Android(1)
- AngularJs(4)
- CentOS(8)
- django 踩坑系列(11)
- Docker(3)
- Python(4)
- SpringBoot系列(4)
- springcloud Alibaba(3)
- 运维相关(7)

阅读排行榜

- 1. JAVA 基础--开发环境 vsco
- 2. Centos7 查看MySQL配置文
- 3. Docker容器 - 容器时间跟宿
- 4. Android Studio 配置Gradl
- 5. Centos7 下安装以及使用m

评论排行榜

- 1. JAVA 基础--开发环境 vsco
- 2. gulp4以及和SASS的结合使
- 3. Centos7 下安装以及使用m

推荐排行榜

1. Spring Cloud Alibaba之na
2. Centos7 查看Mysql配置文
3. NetCore 2.0 应用程序在ce
- 发布(1)
4. 什么是反向代理，如何区别

最新评论

1. Re:gulp4以及与SASS的结
- 请问楼主gulp sassHandler的
- 种问题呢 `[16:40:21] Starti
- 'sassHandler'... Error in plu
2. Re:asdfasdfsdf
- [!]
3. Re:JAVA 基础--开发环境 vs
- vs code 择是想一统武林吗
4. Re:JAVA 基础--开发环境 vs
- 不好使

```
redis:
  database: 8
  host: redis
  port: 6379
  password: 123456      # 密码（默认为空）
  time out: 6000 # 连接超时时长（毫秒）
  pool:
    max-active: 1000 # 连接池最大连接数（使用负值表示没有限制）
    max-wait: -1      # 连接池最大阻塞等待时间（使用负值表示没有限制）
    max-idle: 10      # 连接池中的最大空闲连接
    min-idle: 5       # 连接池中的最小空闲连接
# 数据源配置
datasource:
  url: jdbc:mysql://mysql:3306/mydbname?useUnicode=true&characterEncoding=utf8&serverTimezone=Asia/Shang
  username: root
  password: why1@20200810
```

要特别注意，标红的几个地方，要与容器设置的一样，  
**redis**的**host**这里直接写的**redis**，就是我们的起的**redis**容器名，连接密码，端口，均写的是**redis**容器的，而不是映射主机  
**mysql**的**host**这里要直接写**mysql**的容器名**mysql157**，连接端口是**3306**，容器内部端口，而不是**3307**，用户名和密码是容器**mysql**启

### 3. 打应用的镜像

docker build -t myapplicationImage .

注意后面的小点，意思是使用当前的dockerfile

### 4. 启动应用，注意写法：

docker run -p 80:9080 --name myapp --link mysql57:mysql --link redis:redis -d -v \$PWD/bannerimgs:/bannerimgs myapplicationImage

注意：

--link mysql57:mysql 容器名：别名，这里别名，就是我们在应用程序中配置的别名，如 url: jdbc:mysql://mysql 这里的mysql 就是连接着容器mysql57，

--link redis:redis 容器名：别名,这里的别名，就是我们在应用中配置的别名：

```
redis:
  database: 8
  host: redis
```

## 四 启动后，我们可以看看应用是否正常

docker logs myapp 查看应用的容器是否正常

## 五 补充

当然，后面有docker-compose后，我们可以直拉通过一个docker-compose脚本来实现，容器的编排，但是原理上其实是一样的，只是把上面的步骤放到一个yaml文件中，然后进行启动

半斤八两开始写BLOG了

好文要顶 关注我 收藏该文





freewsf  
粉丝 - 1 关注 - 5

±加关注

0

0

编辑推荐:

- 记一次 .NET 某医院门诊软件 卡死分析
- 我试图通过这篇文章告诉你，这行源码有多牛逼
- [MAUI 程序设计] 界面多态与实现
- 记一次 某智能制造MES系统CPU 爆高分析
- 一个操作把 MySQL 主从复制整崩了

即构专区:

- 在映客的虚拟KTV里唱了一首“爱你”
- 即构携手智能对讲机品牌Runbo，打造可视化对讲通信系统
- 告别尬聊，解锁秀场+社交新玩法（内含源码+Demo）
- ZEGO即构自建MSDN有序网络，为实时音视频传输极致顺畅！
- 即构SDK7月迭代：新增支持按通道设置延迟模式，让卡顿大大减少

Copyright © 2023 freewsf  
Powered by .NET 7.0 on Kubernetes

