
《数据挖掘实践任务教学平台》 部署说明书

版本历史

版本/状态	作者	参与者	起止日期	备注
1.0	刘祥德	刘祥德		开始编写

修改记录

日期	修订版本	修改章节	修改描述	作者

目录

1. 引言.....	3
1.1 编写目的.....	3
1.2 适用范围.....	3
1.3 部署说明.....	3
2. Linux 环境部署.....	3
2.1 安装 Dokcer CE.....	3
2.2 打开 Docker daemon 远程连接.....	4
2.2.1 修改 Docker 配置.....	4
2.3 安装 My SQL 镜像.....	5
2.3.1 下载 5.6 版本的 MySQL 镜像.....	5
\$ docker pull mysql:5.6.....	5
2.3.2 启动 MySQL.....	5
2.4 安装 Docker 的 Redis 镜像.....	5
2.4.1 拉取 Redis3.2 镜像.....	5
2.4.2 启动 Redis.....	5
2.5 安装 phpmyadmin 镜像.....	6
2.5.1 拉取 phpmyadmin 镜像.....	6
2.5.2 启动 phpmyadmin.....	6
2.6 安装 Tomcat.....	6
2.7 构建后端服务器镜像.....	6
2.7.1 配置 Docker 服务器地址.....	6
2.7.2 构建镜像.....	7
2.7.3 确认镜像.....	7
2.7.4 启动服务器.....	7
2.8 部署前端.....	7
2.8.1 打包.....	7
2.8.2 复制文件.....	8
2.8.3 部署到 Tomcat.....	8
3. 访问测试.....	8

1. 引言

1.1 编写目的

本文档针对《基于 Rest 风格的数据挖掘实践教学平台》在 linux 服务器基础平台的规划、安装、以及常规配置等方面进行技术细节描述和说明，主要是为指导用户更快的搭建安装、部署环境。

1.2 适用范围

本文档读者为系统应用环境的工程人员以及系统用户环境的管理人员。默认本手册默认读者为已具备 Linux 服务器下管理、维护、部署技能基础，也即言阅读该文档的人员应当是软件运维的专业技术人员，其已经能够安装 Linux/CenOs7.0 操作系统、部署线上云环境、书写测试部署脚本代码、完成一键式自动化部署的能力，可以理解本文档的部署步骤和技术术语。

1.3 部署说明

本文档主要提供腾讯云主机 CentOS 7.0(其他云服务器可以参照)环境下的软件安装与部署，软件建议使用 Docker 容器装箱链接，但不限于其他基于 Tomcat、Ningx 等具有普适性的部署。

2. Linux 环境部署

2.1 安装 Dokcer CE

(1) 清除历史 Docker 遗留文件

```
$ sudo yum remove docker \
    docker-client \
    docker-client-latest \
    docker-common \
    docker-latest \
    docker-latest-logrotate \
    docker-logrotate \
    docker-selinux \
```

```
docker-engine-selinux \
```

```
docker-engine
```

(2) 设置 Docker 仓库

1、配置管理器

```
$ sudo yum install -y yum-utils \
```

```
device-mapper-persistent-data \
```

```
lvm2
```

2、设置镜像地址

```
sudo yum-config-manager \
```

```
--add-repo \
```

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

(3) 安装 Docker CE

```
sudo yum install docker-ce
```

(4) 启动 Docker

```
sudo systemctl start docker
```

(5) 测试 Docker

```
sudo docker run hello-world
```

Shell 中如果出现以下命令，则说明 Docker 安装成功

```
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://cloud.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/engine/userguide/
```

2.2 打开 Docker daemon 远程连接

打开此连接的目的是为了更方便接下来服务器镜像的一键部署，但该操作可能让非法攻击者获取到服务器的根目录权限，在部署完成后，请关闭该端口，或使用安全规则屏蔽该端口，防止服务器被攻击

2.2.1 修改 Docker 配置

```
$sudo vi /etc/docker/daemon.json
```

在文件中加入

```
{
  "hosts": [
    "tcp://0.0.0.0:2376",
    "unix:///var/run/docker.sock"
  ],
  "registry-mirrors": ["https://mirror.ccs.tencentyun.com"] #DockerHub 腾讯镜像
  下载加速
}
```

Esc,键入:wq, 保存退出

此时, Docker 的守护进程运行任何主机连接到其端口 2376, 客户端可以通过调用 Docker Remote API 操作 Docker。

若要使此端口和访问常开, 建议开启证书守护, 具体文档说明如下:

<https://docs.docker.com/engine/security/https/>

2.3 安装 My SQL 镜像

2.3.1 下载 5.6 版本的 MySQL 镜像

```
$ docker pull mysql:5.6
```

```
docker run --name mysql
```

2.3.2 启动 MySQL

```
\\-p 3306:3306 #将容器 3306 端口映射到云主机 3306 端口
```

```
\\-v /opt/docker_v/mysql/conf.d:/etc/mysql/conf.d 挂载主机文件夹到容器
```

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

```
-e \\MYSQL_ROOT_PASSWORD=15298382837LIUxia 设置数据库密码
```

```
\\-d mysql 后台进程运行
```

2.4 安装 Docker 的 Redis 镜像

2.4.1 拉取 Redis3.2 镜像

```
$ docker pull redis:3.2
```

2.4.2 启动 Redis

```
docker run --name redis-server #容器名字
```

```
-p 6379:6379 #暴露端口
-v $PWD/data:/data #挂载路径
-d redis:3.2 #后台运行镜像
--appendonly yes
```

2.5 安装 phpmyadmin 镜像

安装 phpmyadmin 的目的是为了快速恢复小数据量的数据库文件，方便测试和部署。

2.5.1 拉取 phpmyadmin 镜像

```
$ docker pull phpmyadmin
```

2.5.2 启动 phpmyadmin

```
$sudo docker run --name phpadmin
--link mysql:db -p 9998:80 #链接之前启动的数据库容器，并将其
                           主机名映射为 db(phpmyadmin 通过 db 访问数据库)
-d phpmyadmin/phpmyadmin
```

2.6 安装 Tomcat

```
$ docker pull tomcat
```

2.7 构建后端服务器镜像

2.7.1 配置 Docker 服务器地址

这里默认开发人员已经将工程源码导入到 IntelliJ IDEA Ultimate，在 IntelliJ IDEA Ultimate 打开\data-minning-platform\dm-web\pom.xml

```
<configuration>
  <dockerHost>${HOST_ADDESS}</dockerHost>
<imageName>${docker.image.prefix}/${project.build.finalName}</imageName>
<dockerDirectory>src/main/docker</dockerDirectory>
  <resources>
    <resource>
      <targetPath></targetPath>
      <directory>${project.build.directory}</directory>
      <include>${project.build.finalName}.jar</include>
```



```
</resource>
</resources>
</configuration>
```

将\$HOST_ADDESS 替换成主机名:2376(Docker 远程访问端口),如若主机名为 118.24.62.94。则\$HOST_ADDESS 替换为 118.24.62.94:2376。

2.7.2 构建镜像

使用 Maven 在\data-minning-platform 根目录下运行 clean install，完成所有依赖构建

进入到\data-minning-platform\dm-web 运行 docker:build -DskipTests。

此时 maven 会自动连接远程已经安装好 Docker 的服务器，并在服务器端打包部署好对应的镜像。

2.7.3 确认镜像

```
$sudo docker images
```

若出现以下镜像信息，说明服务器后端程序已经部署到云主机上。

```
89d1a5a0f2de    dm/dm-platform    "java -Djava.securit-" 6 hours ago        Up 6 hours          0.0.0.0:8080->8080/tcp    dm-web
```

2.7.4 启动服务器

```
sudo docker run --name dm-web
```

```
\\--link mysql:mysql #链接到数据库容器
```

```
\\--link redis-server:redis-server #链接到 Redis 容器
```

```
\\-p 8080:8080 #暴露服务端口
```

```
\\-d dm/dm-platform
```

2.8 部署前端

2.8.1 打包

这里要求部署人员已经配置好 Node.js>=6.6 的前端构建环境，cd 进入到的前端源代码\dm-admin-vue 目录。

```
npm run build
```

出现 Build complete 字样无错，说明构建完成,disk 文件夹会出现如下文件

名称	修改日期	类型	大小
static	2018/4/13 16:57	文件夹	
favicon.ico	2018/4/13 16:57	图标	67 KB
index.html	2018/4/13 16:57	Chrome HTML D...	1 KB

2.8.2 复制文件

`$mkdir /usr/local/tomcat/webapps` 创建 tomcat 镜像要挂载的目录
将 disk 文件夹下的所有文件复制到云主机的对应目录下。

2.8.3 部署到 Tomcat

`$docker run -d`

`-p 9527:8080`

`-v /usr/local/tomcat/webapps:/usr/local/tomcat/webapps #挂载`

`--name tomcat_dm tomcat`

查看 docker 日志，当提示

`org.apache.catalina.startup.Catalina.start Server startup in 101655 ms`

服务器启动完成。

3. 访问测试

访问{主机名或者域名}:9527/data_mining_platform