**《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](#_Toc511404129)

[1.1 编写目的 3](#_Toc511404130)

[1.2 适用范围 3](#_Toc511404131)

[1.3 部署说明 3](#_Toc511404132)

[2. Linux环境部署 3](#_Toc511404133)

[2.1 安装Dokcer CE 3](#_Toc511404134)

[2.2 打开Docker daemon远程连接 4](#_Toc511404135)

[2.2.1 修改Docker配置 4](#_Toc511404136)

[2.3 安装My SQL镜像 5](#_Toc511404137)

[2.3.1 下载5.6版本的MySQL镜像 5](#_Toc511404138)

[$ docker pull mysql:5.6 5](#_Toc511404139)

[2.3.2 启动MySQL 5](#_Toc511404140)

[2.4 安装Docker的Redis镜像 5](#_Toc511404141)

[2.4.1 拉取Redis3.2镜像 5](#_Toc511404142)

[2.4.2 启动Redis 5](#_Toc511404143)

[2.5 安装phpmyadmin镜像 6](#_Toc511404144)

[2.5.1 拉取phpmyadmin镜像 6](#_Toc511404145)

[2.5.2 启动phpmyadmin 6](#_Toc511404146)

[2.6 安装Tomcat 6](#_Toc511404147)

[2.7 构建后端服务器镜像 6](#_Toc511404148)

[2.7.1 配置Docker服务器地址 6](#_Toc511404149)

[2.7.2 构建镜像 7](#_Toc511404150)

[2.7.3 确认镜像 7](#_Toc511404151)

[2.7.4 启动服务器 7](#_Toc511404152)

[2.8 部署前端 7](#_Toc511404153)

[2.8.1 打包 7](#_Toc511404154)

[2.8.2 复制文件 8](#_Toc511404155)

[2.8.3 部署到Tomcat 8](#_Toc511404156)

[3. 访问测试 8](#_Toc511404157)

# 引言

## 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。

## 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。

## 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。

# Linux环境部署

## 安装Dokcer CE

#### 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

#### 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

#### 安装Docker CE

sudo yum install docker-ce

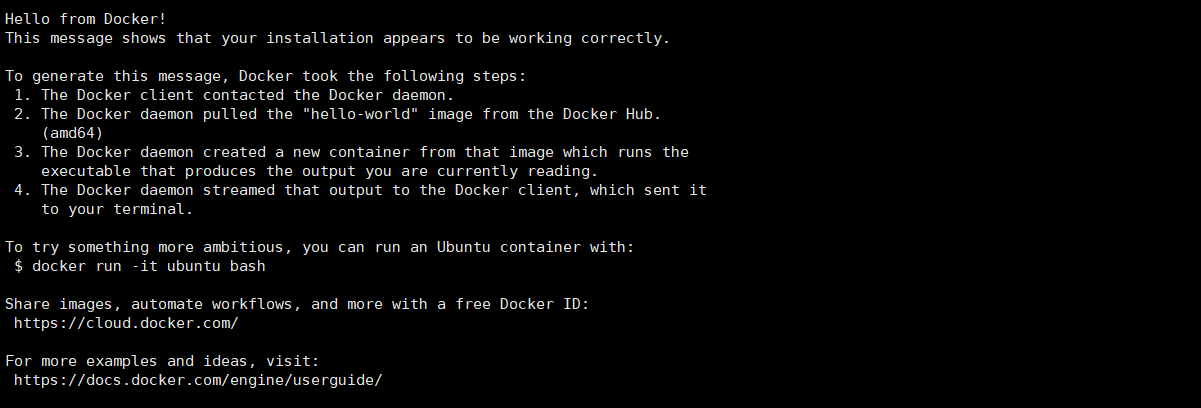
#### 启动Docker

sudo systemctl start docker

#### 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



## 打开Docker daemon远程连接

打开此连接的目的是为了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

### 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/

## 安装My SQL镜像

### 下载5.6版本的MySQL镜像

### $ docker pull mysql:5.6

docker run --name mysql

### 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\_](file:///\\MYSQL_ROOT_PASSWORD=15298382837LIUxia_) 设置数据库密码

\\-d mysql 后台进程运行

## 安装Docker的Redis镜像

### 拉取Redis3.2镜像

$ docker pull redis:3.2

### 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

## 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。

### 拉取phpmyadmin镜像

$ docker pull phpmyadmin

### 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

## 安装Tomcat

$ docker pull tomcat

## 构建后端服务器镜像

### 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.

### 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。

### 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。



### 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

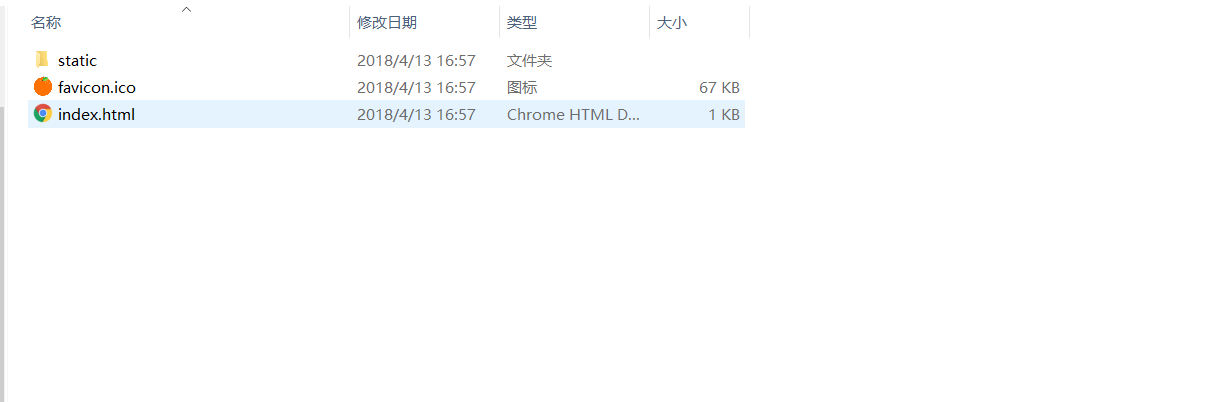
## 部署前端

### 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



### 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。

### 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。

# 访问测试

访问{主机名或者域名}:9527/data\_mining\_platform