Mecoin服务系统

安装部署说明书

[1. 硬件配置 2](#_Toc515876829)

[2. 系统运行环境搭建 2](#_Toc515876830)

[3. 系统部署与运行 6](#_Toc515876831)

[5. 系统部署架构图(供参考) 8](#_Toc515876832)

# 硬件配置

三台Linux服务器+一台mysql数据库

Linux1：部署api服务

Linux2：部署task服务

Linux3：部署后台管理php服务

Mysql: 数据库服务

# 系统运行环境搭建

###### 安装jdk1.8

**第一步：下载安装包**

下载Linux环境下的jdk1.8，请去（[官网](http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html" \t "_blank)）中下载jdk的安装文件；

64位的Linux环境，请下载[jdk-8u131-linux-x64.tar.gz](http://download.oracle.com/otn-pub/java/jdk/8u131-b11/d54c1d3a095b4ff2b6607d096fa80163/jdk-8u131-linux-x64.tar.gz)。

如果Linux本身连接到互联网，我们可以直接通过wget命令直接把JDK安装包下载下来，如图所示：



要是没有外网的环境，还是安装上面的方法下载安装包，然后上传到服务器当中

**第二步、解压安装包**

将我们下载好的JDK安装包上传到服务器，进行解压

解压命令进行解压

1 $ cd /home/cmfchina

2 $ tar -zxvf jdk-8u131-linux-x64.tar.gz

https://images2015.cnblogs.com/blog/506829/201707/506829-20170712154949462-1567604342.png

解压完成之后，可以在当前目录下看到一个名字为【jdk1.8.0\_131】的目录，里面存放的是相关文件

我们要将解压后的【jdk1.8.0\_131】里面的所有数据移动到我们需要安装的文件夹当中，我们打算将jdk安装在usr/java当中，我们在usr目录下新建一个java文件夹

mkdir /usr/java

https://images2015.cnblogs.com/blog/506829/201707/506829-20170712155622509-501402302.png

将【jdk1.8.0\_131】里的数据拷贝至java目录下

mv /home/cmfchina/jdk1.8.0\_131 /usr/java

https://images2015.cnblogs.com/blog/506829/201707/506829-20170712160243556-345559176.png

**第三步、修改环境变量**

至此，我们最后需要修改环境变量，通过命令

vim /etc/profile

https://images2015.cnblogs.com/blog/506829/201707/506829-20170712161307134-1500571117.png

用vim编辑器来编辑profile文件，在文件末尾添加一下内容（按“i”进入编辑）：

1 export JAVA\_HOME=/usr/java/jdk1.8.0\_131

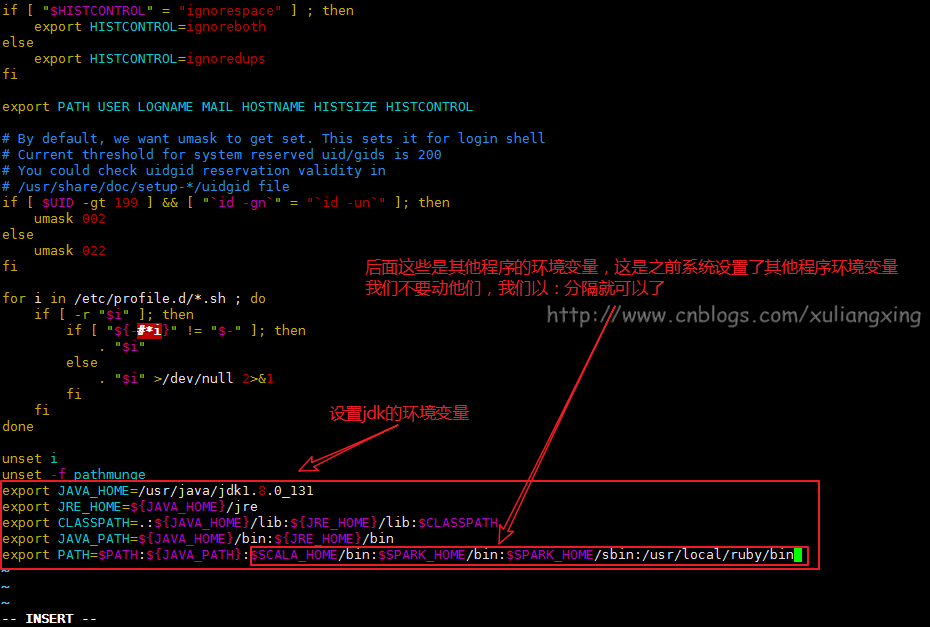
2 export JRE\_HOME=${JAVA\_HOME}/jre

3 export CLASSPATH=.:${JAVA\_HOME}/lib:${JRE\_HOME}/lib:$CLASSPATH

4 export JAVA\_PATH=${JAVA\_HOME}/bin:${JRE\_HOME}/bin

5 export PATH=$PATH:${JAVA\_PATH}

如图所示：



然后，保存并退出(按：wq!)

保存完之后，我们还需要让这个环境变量配置信息里面生效，要不然只能重启电脑生效了。

通过命令source /etc/profile让profile文件立即生效，如图所示

https://images2015.cnblogs.com/blog/506829/201707/506829-20170712162556978-2141450378.png

###### 安装redis

**下载地址：**[http://redis.io/download](http://redis.io/download" \t "_blank)，下载最新稳定版本。

本教程使用的最新文档版本为 2.8.17，下载并安装：

$ wget http://download.redis.io/releases/redis-2.8.17.tar.gz

$ tar xzf redis-2.8.17.tar.gz

$ cd redis-2.8.17

$ make

make完后 redis-2.8.17目录下会出现编译后的redis服务程序redis-server,还有用于测试的客户端程序redis-cli,两个程序位于安装目录 src 目录下：

下面启动redis服务.

$ cd src

$ ./redis-server

注意这种方式启动redis 使用的是默认配置。也可以通过启动参数告诉redis使用指定配置文件使用下面命令启动。

$ cd src

$ ./redis-server redis.conf

redis.conf是一个默认的配置文件。我们可以根据需要使用自己的配置文件。

启动redis服务进程后，就可以使用测试客户端程序redis-cli和redis服务交互了。 比如：

$ cd src

$ ./redis-cli

redis> set foo bar

OK

redis> get foo

"bar"

### 查看 redis 是否启动？

$ redis-cli

以上命令将打开以下终端：

redis 127.0.0.1:6379>

127.0.0.1 是本机 IP ，6379 是 redis 服务端口。现在我们输入 PING 命令。

redis 127.0.0.1:6379> ping

PONG

以上说明我们已经成功安装了redis。

具体信息的配置可参考https://blog.csdn.net/yjqyyjw/article/details/73293455

# 系统部署与运行

.供参考

项目打包上传至目标目录

1.在根目录下使用

Grade build –x test

打包后生产.Jar文件

2.上传后使用

Java –jar xxx.jar命令启动项目

3. 把应用配置成可以通过service启动和停止，通过下面的命令在init.d目录建立一个软链接即可。这样的好处是完全不用自己写应用的启动和停止脚本了。  
ln -s 应用jar文件路径 /etc/init.d/应用名称

配置好后就可以通过标准的命令去启停应用。

service 应用名称 start/stop

4.应用的配置就是jar目录下和jar文件同名的conf后缀的文件， 可以在里面配置JVM参数，还有Spring Boot的环境变量，下面的例子中，server.port会覆盖application-test.yml中同名的变量。

JAVA\_OPTS="-D tlwe.home=/opt/coral/tlwe -Dspring.profiles.active=test -XX:+PrintGCDateStamps -XX:+PrintGCDetails -verbose:gc -Xloggc:/opt/coral/tlwe/logs/gc.log"

RUN\_ARGS="--server.port=9000"

5．应用升级

为方便起见可以把生产服务器的公钥复制到测试服务器，这样每次从测试服务器scp文件就不需要输入密码。具体的步骤是在生产服务器生成公钥 （通过ssh-keygen -t rsa），然后把公钥内容添加到测试服务器的vim ~/.ssh/authorized\_keys文件的后面。

升级脚本的步骤如下：

(1). 从测试服务器（注意此过程中不要提交代码， 否则可能下载到损坏的jar）下载最新的jar到tmp目录

(2).下载完成后，停止当前版本的应用

(3). 把当前服务的jar备份到backup目录， 复制tmp中的最新文件到jar目录。

(4). 启动服务器(service 应用名称 start)

交易大盘的实际例子如下：

echo "从测试服务器下载最新jar文件到临时目录"

scp coral@119.23.217.175://opt/coral/tlwe/jar/api-0.0.1-SNAPSHOT.jar /var/tlwe/tmp

echo "停止服务"

service tlwe-api stop

sleep 2

echo "备份当前版本"

mv /var/tlwe/jar/api-0.0.1-SNAPSHOT.jar /var/tlwe/backup/api-0.0.1-SNAPSHOT.jar.`date +"%Y-%m-%d-%H%M%S"`

cp /var/tlwe/tmp/api-0.0.1-SNAPSHOT.jar /var/tlwe/jar

echo "启动服务"

service tlwe-api start

# 系统部署架构图(供参考)

下图为应对高访问量高可用性系统的应用部署架构，客户可根据实际情况自行进行系统扩展延伸。

