**Webproxy 服务器搭建（LINUX）**

1.首先导入webproxy资源到服务器，其中webproxy资源目录下，doc是windows环境的配置文档，schema下是mysql需要执行的文件，source下是对应的服务jar包，webapps下是服务所需的war包。

2.安装jdk，建议使用jdk1.7

配置jdk环境变量：

/etc下的profile文件里配置：

export JAVA\_HOME=/home/lihebin/lihebin/jdk1.7.0\_79/

export PATH=$JAVA\_HOME/bin:$PATH

export CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

重启：source profile

java -version

3.安装tomcat，并配置：

打开在Tomcat的安装目录的bin文件的catalina.sh文件,进入编辑状态.

在注释后面加上如下脚本:

JAVA\_OPTS='-Xms512m -Xmx1024m'

JAVA\_OPTS="$JAVA\_OPTS -server -XX:PermSize=64M -XX:MaxPermSize=256m -Xms64m

-Xmx128m

-Dlogdir=...\...\log

-Dlog.upay.dir=...\...\log\upay\

-Dshouqianba.flavor=prod"

bin目录下

./startup.sh 启动

./shutdown.sh 关闭

4.安装mysql: apt-get install mysql-server

打开"终端窗口"，输入"mysql -u root -p"-->回车-->输入mysql中"root"用户的密码-->回车-->登录成功

并将schema下的.sql写入到mysql中:

mysql>source /home/test.sql

或直接导入命令为： mysql -h localhost -u root -p temp

会生成两个数据库:mapstore和keystore

并配置这两个库的用户名和密码都为：wosai,wosai

设置mysql用户名密码：

首先以root用户名登录，

然后进入没有密码的mysql，然后use mysql----------> insert into user(host,user,password) values('%','root',"\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

然后mysqladmin -u root password "your password"则OK了

其中需要先在mapstore里写入：

insert into proxy\_merchant(id,client\_merchant\_sn,merchant\_sn,ctime,mtime,deleted,version)

values('cb0dd313-b300-11e5-9987-6c92bf21cf9b','1','1680000496197',UNIX\_TIMESTAMP()\*1000,UNIX\_TIMESTAMP()\*1000,0,1);

5.安装并配置redis：

**第一部分：安装redis**   
希望将redis安装到此目录

|  |  |
| --- | --- |
| 1 | /usr/local/redis |

希望将安装包下载到此目录

|  |  |
| --- | --- |
| 1 | /usr/local/src |

那么安装过程指令如下：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | $ mkdir /usr/local/redis  $ cd /usr/local/src  $ wget http://redis.googlecode.com/files/redis-2.6.14.tar.gz  $ tar xzf redis-2.6.14.tar.gz  $ ln -s redis-2.6.14 redis #建立一个链接  $ cd redis  $ make PREFIX=/usr/local/redis install #安装到指定目录中 |

注意上面的最后一行，我们通过PREFIX指定了安装的目录。如果make失败，一般是你们系统中还未安装gcc,那么可以通过yum安装：

|  |  |
| --- | --- |
| 1 | yum install gcc |

安装完成后，继续执行make.

在安装redis成功后，你将可以在/usr/local/redis看到一个bin的目录，里面包括了以下文件：

|  |  |
| --- | --- |
| 1 | redis-benchmark  redis-check-aof  redis-check-dump  redis-cli  redis-server |

**第二部分：将redis做成一个服务**

1.复制脚本到/etc/rc.d/init.d目录   
ps: /etc/rc.d/init.d/目录下的脚本就类似与windows中的注册表，在系统启动的时候某些指定脚本将被执行   
按以上步骤安装Redis时，其服务脚本位于：

|  |  |
| --- | --- |
| 1 | /usr/local/src/redis/utils/redis\_init\_script |

必须将其复制到/etc/rc.d/init.d的目录下：

|  |  |
| --- | --- |
| 1 | cp /usr/local/src/redis/utils/redis\_init\_script /etc/rc.d/init.d/redis |

将redis\_init\_script复制到/etc/rc.d/init.d/，同时易名为redis。

如果这时添加注册服务：

|  |  |
| --- | --- |
| 1 | chkconfig --add redis |

将报以下错误：

|  |  |
| --- | --- |
| 1 | redis服务不支持chkconfig |

为此，我们需要更改redis脚本。

2.更改redis脚本   
打开使用vi打开脚本，查看脚本信息：

|  |  |
| --- | --- |
| 1 | vim /etc/rc.d/init.d/redis |

看到的内容如下(下内容是更改好的信息)：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42 | #!/bin/sh  #chkconfig: 2345 80 90  # Simple Redis init.d script conceived to work on Linux systems  # as it does use of the /proc filesystem.    REDISPORT=6379  EXEC=/usr/local/redis/bin/redis-server  CLIEXEC=/usr/local/redis/bin/redis-cli    PIDFILE=/var/run/redis\_${REDISPORT}.pid  CONF="/etc/redis/${REDISPORT}.conf"    case "$1" in      start)          if [ -f $PIDFILE ]          then                  echo "$PIDFILE exists, process is already running or crashed"          else                  echo "Starting Redis server..."                  $EXEC $CONF &          fi          ;;      stop)          if [ ! -f $PIDFILE ]          then                  echo "$PIDFILE does not exist, process is not running"          else                  PID=$(cat $PIDFILE)                  echo "Stopping ..."                  $CLIEXEC -p $REDISPORT shutdown                  while [ -x /proc/${PID} ]                  do                      echo "Waiting for Redis to shutdown ..."                      sleep 1                  done                  echo "Redis stopped"          fi          ;;      \*)          echo "Please use start or stop as first argument"          ;;  esac |

和原配置文件相比：

1.原文件是没有以下第2行的内容的，

|  |  |
| --- | --- |
| 1 | #chkconfig: 2345 80 90 |

2.原文件EXEC、CLIEXEC参数，也是有所更改。

|  |  |
| --- | --- |
| 1  2 | EXEC=/usr/local/redis/bin/redis-server  CLIEXEC=/usr/local/redis/bin/redis-cli |

3.redis开启的命令，以后台运行的方式执行。

|  |  |
| --- | --- |
| 1 | $EXEC $CONF & |

ps:注意后面的那个“&”，即是将服务转到后面运行的意思，否则启动服务时，Redis服务将

占据在前台，占用了主用户界面，造成其它的命令执行不了。   
4.将redis配置文件拷贝到/etc/redis/${REDISPORT}.conf

|  |  |
| --- | --- |
| 1  2 | mkdir /etc/redis  cp /usr/local/src/redis/redis.conf /etc/redis/6379.conf |

这样，redis服务脚本指定的CONF就存在了。默认情况下，Redis未启用认证，可以通过开启6379.conf的requirepass 指定一个验证密码。

以上操作完成后，即可注册yedis服务：

|  |  |
| --- | --- |
| 1 | chkconfig --add redis |

3.启动redis服务

|  |  |
| --- | --- |
| 1 | service redis start |

**第三，将Redis的命令所在目录添加到系统参数PATH中**

修改profile文件：

|  |  |
| --- | --- |
| 1 | vi /etc/profile |

在最后行追加:

|  |  |
| --- | --- |
| 1 | export PATH="$PATH:/usr/local/redis/bin" |

然后马上应用这个文件：

|  |  |
| --- | --- |
| 1 | . /etc/profile |

这样就可以直接调用redis-cli的命令了，如下所示：

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | $ redis-cli  redis 127.0.0.1:6379> auth superman  OK  redis 127.0.0.1:6379> ping  PONG  redis 127.0.0.1:6379> |

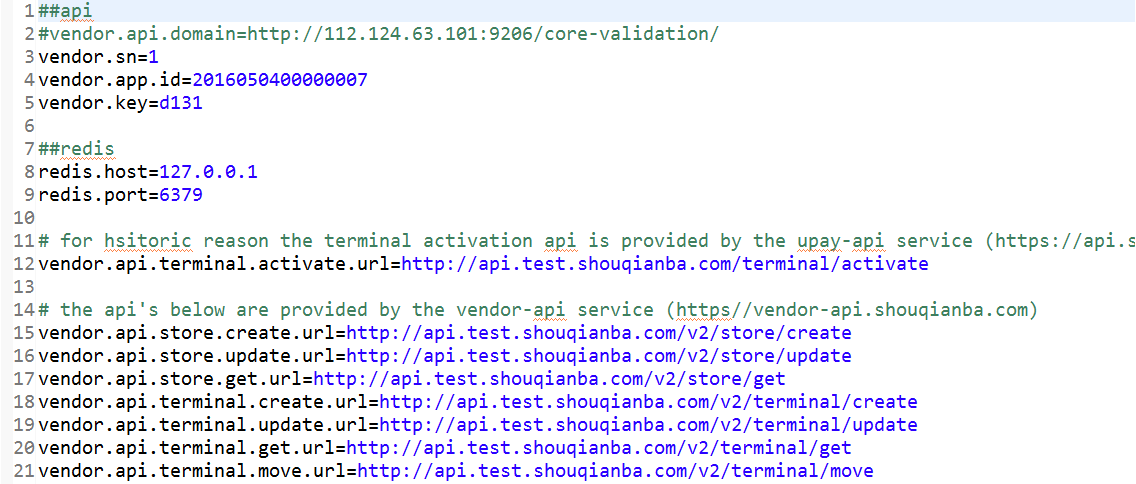
至此，redis 就成功安装了。

#打开服务  
service redisd start  
#关闭服务  
service redisd stop

6.重命名webapps目录下的三个war文件，去除版本号，如：proxy-core.war，proxy-upay.war,proxy-auto.war,然后拷贝webapps目录中的三个war文件至tomcat安装目录/webapps/下,并启动tomcat，启动成功后会在tomcat安装目录/webapps/下生成三个文件夹:proxy-auto,proxy-core,proxy-upay；然后分别在这三个路径下的进行相应的配置：

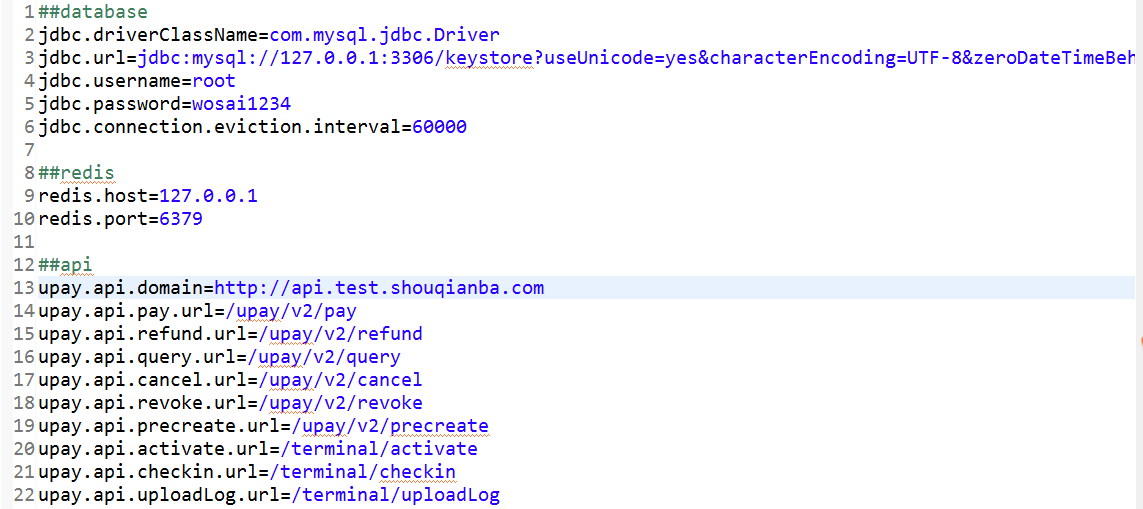
1）配置proxy-core的app参数、缓存连接信息、api domain，打开配置文件路径webapps\proxy-core\WEB-INF\classes\spring\flavor-prod.properties，

如图：



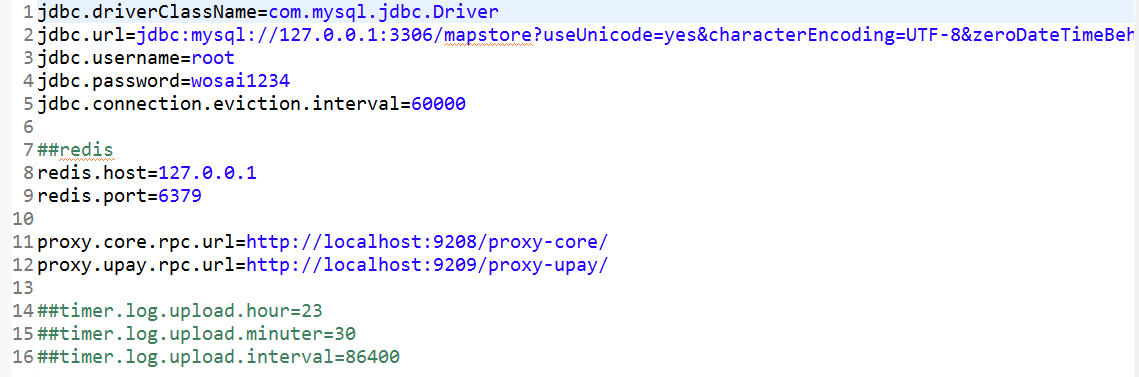
2）配置proxy-upay的数据库连接信息、缓存连接信息、api domain，打开配置文件路径webapps\proxy-upay\WEB-INF\classes\spring\flavor-prod.properties，

如图：



3）配置proxy-auto的数据库连接信息、缓存连接信息、rpc地址，打开配置文件路径webapps\proxy-auto\WEB-INF\classes\spring\flavor-prod.properties，

如图：



7.配置成功之后，重启tomcat和redis服务。并测试是否能访问成功：

测试，访问地址：<http://localhost:8080/proxy-auto/test/index.html>，进入下图页面代表服务启动成功。

