

## 一、tomcat与mysql连接

### JSP动态网站实例--JavaCenterHome

在前面tomcat提供正常服务基础上

1. 安装数据库mysql-5... mysql-server-...

2. 安装java连接mysql所需要的文件mysql-connector-java

上传mysql-connector-java-5.1.21.tar.gz

解包tar -xvf mysql-connector-java-5.1.21.tar.gz

mv mysql-connector-java-5.1.21/mysql-connector-java-5.1.21-bin.jar /opt/tomcat/lib

3. 开启mysql服务

service mysqld start

4. 上传JavaCenterHome网站文件:WebRoot

复制到网站根目录

5. 客户端访问安装 (ip:8080/install)

## 二、apache和tomcat整合的好处

如果客户端请求的是静态页面，则只需要Apache服务器响应请求。

如果客户端请求动态页面，则是Tomcat服务器响应请求。

1、安装Apache。

代码:

```
tar xzvf httpd-2.2.9.tar.gz
```

```
cd httpd-2.2.9
```

```
./configure --prefix=/usr/local/apache2 --enable-so
```

```
make
```

```
make install
```

2、安装jdk

```
rpm -ivh jdk-8u20-linux-x64.rpm
```

配置环境变量

```
#vi /etc/profile
```

在profile文件中 追加

```
export JAVA_HOME=/usr/java/jdk1.8.0_20
```

```
export JAVA_BIN=/usr/java/jdk1.8.0_20/bin
```

```
export PATH=$PATH:$JAVA_HOME/bin
```

```
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
```

```
export PATH=$JAVA_HOME/bin:$JRE_HOME/bin:$PATH
```

source /etc/profile 让/etc/profile 文件修改后立即生效

java 测试

```
java -version 显示版本
```

显示1.8.0\_20表示安装成功

3. 安装tomcat

```
tar -zxvf apache-tomcat-7.0.47.tar.gz
```

得到Tomcat的所有文件,然后把它拷贝到你要安装的位置,如/opt/tomcat

```
cp -r /home/zdw/software/apache-tomcat-7.0.47 /opt/tomcat
```

切换到/opt/tomcat/bin

```
cd /opt/tomcat/bin
```

运行命令

apxs是一个为Apache HTTP服务器编译和安装扩展模块的工具，用于编译一个或多个源程序或目标代码文件为动态共享对象，使之可以用由mod\_so提供的LoadModule指令在运行时加载到Apache服务器中。

因此，要使用这个扩展机制，你的平台必须支持DSO特性，而且Apache httpd必须内建了mod\_so模块。apxs工具能自动探测是否具备这样的条件，你也可以自己用这个命令手动探测：

./startup.sh

4、编译生成mod\_jk。（mod\_jk实质上是Apache与Tomcat的连接器，并藉此附带提供集群和负载均衡的功能。）

代码:

```
tar xzvf jakarta-tomcat-connectors-1.2.15-src.tar.gz
cd jakarta-tomcat-connectors-1.2.15-src/jk/native
./configure --with-apxs=/usr/local/apache2/bin/apxs
make
cp ./apache-2.0/mod_jk.so /usr/local/apache2/modules/
```

5、配置。

在/usr/local/apache2/conf/下面建立两个配置文件mod\_jk.conf和workers.properties。

**vim mod\_jk.conf**

添加以下内容:

```
# 指出mod_jk模块工作所需要的工作文件workers.properties的位置
JkWorkersFile /usr/local/apache2/conf/workers.properties
```

```
# 将所有servlet 和jsp请求通过ajp13的协议送给Tomcat，让Tomcat来处理
JkMount /*.jsp worker1
```

注意: # JkMount 表示使用Tomcat来解析，JkUnMount则相反

**vi workers.properties**

添加以下内容:

```
# Defining a worker named worker1 and of type ajp13
worker.list=worker1
```

```
# Set properties for worker1
worker.worker1.type=ajp13
worker.worker1.host=localhost
worker.worker1.port=8009
再配置httpd.conf，作以下修改:
将ServerName 修改为 ServerName LocalHost:80
在DirectoryIndex中添加
DirectoryIndex index.html index.jsp
```

增加关于加载mod\_jk的语句: 在httpd.conf文件后边添加如下两行代码:

```
LoadModule jk_module modules/mod_jk.so
Include /usr/local/apache2/conf/mod_jk.conf
```

最后将tomcat下的jsp测试页复制到apache的根目录下:

```
cp -r /opt/tomcat/webapps/ROOT/* /usr/local/apache2/htdocs
```

**启动apache**

```
/usr/local/apache2/bin/apachectl start
```

**启动tomcat、**

```
/opt/tomcat/bin/startup.sh
```

测试

<http://localhost/index.html>

如果出现页面，则说明apache可正确解析静态文件

<http://localhost/index.jsp>

如果出现页面，则说明apache可正确解析jsp动态文件