operation 05

=============================================

**一，使用tomcat搭建网站服务**

cd ~/lnmp\_soft/

tar -xf apache-tomcat-8.0.30.tar.gz

cp -r apache-tomcat-8.0.30 /usr/local/tomcat

cd /usr/local/tomcat/

yum -y install java-1.8.0-openjdk

tomcat目录：

bin 存放主程序

logs 存放日志

conf 配置文件

webapps 存放网站页面

lib 存放库文件

work 存放编译后页面文件

[root@proxy tomcat]# bin/startup.sh //开启服务

http://192.168.2.5:8080/ //使用火狐访问

netstat -ntulp | grep java //检查如果有8005、8009、8080端口

则正常

**由于tomcat服务开启时需要大量随机字符，但系统中随机字符**

**不够多的话会导致tomcat服务启动失败**

**解决方案1：**

mv /dev/random /dev/random.bak //将random改名

ln -s /dev/urandom /dev/random //用urandom替换random

**解决方案2：**

yum -y install rng-tools

systemctl start rngd

killall java //杀掉所有java程序

[root@proxy tomcat]# bin/startup.sh //重新开启服务

netstat -ntulp | grep java

**测试静态页面：**

cd /usr/local/tomcat

echo "tomcat-test~~~~" > webapps/ROOT/test01.html

http://192.168.2.5:8080/test01.html 火狐访问测试

**测试动态页面**

vim webapps/ROOT/test02.jsp

<html>

<body>

<center>

Now time is: <%=new java.util.Date()%>

</center>

</body>

</html>

-----------------------------------------------------------

使用tomcat创建虚拟主机

回顾，

httpd搭建虚拟主机

<virtualhost \*:80>

servername [www.a.com](http://www.a.com)

documentroot /var/www/html

</virtualhost>

nginx搭建虚拟主机

http {

server {

listen 80;

server\_name [www.a.com](http://www.a.com);

root html;

index index.html;

}

}

tomcat搭建虚拟主机

<Host name=域名 appBase=网站页面位置 >

</Host>

vim conf/server.xml //打开配置文件在122行添加虚拟主机配置

<Host name="www.b.com" appBase="web\_b">

</Host>

mkdir -p web\_b/ROOT //之后创建b网站的目录

echo "tomcat-A~~~" > webapps/ROOT/index.html

echo "tomcat-B~~~" > web\_b/ROOT/index.html

tail -1 /etc/hosts //查看域名解析

192.168.2.5 [www.a.com](http://www.a.com) [www.b.com](http://www.b.com) www.c.com

bin/shutdown.sh //关闭tomcat

bin/startup.sh //开启

[root@proxy tomcat]# curl [www.a.com:8080](http://www.a.com:8080) //测试

tomcat-A~~~

[root@proxy tomcat]# curl www.b.com:8080

tomcat-B~~~

再次修改虚拟主机配置

<Host name="www.b.com" appBase="web\_b"

unpackWARs="true" autoDeploy="true"> //添加unpackWARs自动

解war包，和autoDeploy可以自动更新开发工程师编写的网站功能代码

</Host>

bin/shutdown.sh //然后重启tomcat

bin/startup.sh //开启服务

netstat -ntulp | grep java //检查

yum -y install java-1.8.0-openjdk-devel //安装可以制作war包的工具

jar -cf b.war /var/log //之后可以使用jar命令打war包

cp b.war web\_b/ //将war包放入虚拟主机的网页目录

ls web\_b/ //检查，可以看到已经自动解开

-----------------------------------------------------------------

**tomcat访问路径与页面位置**

**Context path 匹配用户访问路径**

**docBase 定义页面位置**

**测试 1：**

<Host name="www.b.com" appBase="web\_b"

unpackWARs="true" autoDeploy="true">

<Context path="" docBase="" />

</Host>

echo "web\_b/ROOT/index.html" > web\_b/ROOT/index.html //默认页面路径

echo "web\_b/index.html" > web\_b/index.html //创建素材，定义新路径

curl [www.b.com:8080](http://www.b.com:8080) //测试可以看到的是默认的页面路径

web\_b/ROOT/index.html

之后重启tomcat

curl [www.b.com:8080](http://www.b.com:8080) //重启后测试可以看到新路径

web\_b/index.html

**测试 2：**

<Host name="www.b.com" appBase="web\_b"

unpackWARs="true" autoDeploy="true">

<Context path="" docBase="abc" /> //修改成abc

</Host>

mkdir web\_b/abc //创建测试目录

echo "web\_b/abc/index.html" > web\_b/abc/index.html //测试页面

重启tomcat

[root@proxy tomcat]# curl [www.b.com:8080](http://www.b.com:8080) //查看，见到新路径

web\_b/abc/index.html

**测试 3：**

<Host name="www.b.com" appBase="web\_b"

unpackWARs="true" autoDeploy="true">

<Context path="" docBase="/abc" /> //修改成/abc

</Host>

mkdir /abc //创建测试目录

echo "/abc/index.html" > /abc/index.html //创建测试文件

重启tomcat

[root@proxy tomcat]# curl [www.b.com:8080](http://www.b.com:8080) //得到新路径

/abc/index.html

**测试 4：**

<Host name="www.b.com" appBase="web\_b"

unpackWARs="true" autoDeploy="true">

<Context path="/test" docBase="/abc" /> //path后面添加内容可以匹配

用户访问路径，当用户访问www.b.com:8080/test/时，看到的是/abc的内容，如果

访问[www.b.com:8080](http://www.b.com:8080)就看到web\_b/ROOT的内容

</Host>

重启tomcat

[root@proxy tomcat]# curl www.b.com:8080/test/

/abc/index.html

[root@proxy tomcat]# curl www.b.com:8080

web\_b/ROOT/index.html

测试 5：

<Host name="www.b.com" appBase="web\_b"

unpackWARs="true" autoDeploy="true">

<Context path="/test" docBase="abc" /> //将/abc改成abc

当用户访问www.b.com:8080/test/时，可以看到web\_b/abc里的内容

当用户访问[www.b.com:8080](http://www.b.com:8080)时就看到web\_b/ROOT的内容

重启tomcat

[root@proxy tomcat]# curl www.b.com:8080/test/

web\_b/abc/index.html

[root@proxy tomcat]# curl www.b.com:8080

web\_b/ROOT/index.html

---------------------------------------------------------------------------------

给新创建的虚拟主机添加日志功能

vim conf/server.xml //修改配置文件，复制最后一页默认虚拟主机的日志配置

其中prefix是日志名称，suffix是日志后缀名

<Host name="www.b.com" appBase="test01"

unpackWARs="true" autoDeploy="true">

<Context path="" docBase="a" />

<Context path="/abc" docBase="/var/www/html" />

<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"

prefix="www\_b\_com" suffix=".log"

pattern="%h %l %u %t &quot;%r&quot; %s %b" />

</Host>

配置好之后开启或者重启tomcat服务

curl www.b.com:8080 //访问一次网站，可以产生日志记录

[root@proxy tomcat]# ls logs/ 查看日志目录可以看到b网站的日志名称

--------------------------------------------------------------------------

maven 是一个项目管理工具

1, 将三个文件传入虚拟机root的家目录,原地释放

apache-maven-3.6.3-bin.tar.gz //maven工具包

cms.tar.gz //网站项目，代码已经都编写完毕

m2.tar.gz //依赖程序，释放之后是隐藏目录.m2

如果有软件包无法释放可以尝试安装bzip2软件包

2，配置maven

mv apache-maven-3.6.3 /usr/local/maven //将maven移动到/usr/local下

yum -y install java-1.8.0-openjdk

yum -y install java-devel

vim /usr/local/maven/conf/settings.xml //打开配置文件在第158下面添加

公网服务器地址

<mirror>

<id>nexus-aliyun</id>

<mirrorOf>\*</mirrorOf>

<name>Nexus aliyun</name>

<url>http://maven.aliyun.com/nexus/content/groups/public</url>

</mirror>

3，配置数据库

yum -y install mariadb mariadb-server

systemctl start mariadb

[root@proxy ~]# cd ~/CMS/

cp src/main/resources/shishuocms.properties . //拷贝网站数据库

配置文件到当前目录，该文件定义了数据的用户名和密码等信息

mysql -uroot -p < sql/install.sql //导入网站相关数据库文件

Enter password: 直接回车

mysqladmin password //修改数据库密码

New password: //接下来输入12345678

Confirm new password: //在输入一次12345678

4，利用maven打包项目

/usr/local/maven/bin/mvn clean package //按照之前定义的

公网服务器地址、数据库配置等内容打包项目

rm -rf /usr/local/tomcat/webapps/ROOT //删除原有默认页面目录

cp target/shishuocms-2.0.1.war /usr/local/tomcat/webapps/ROOT.war //将新

项目移动到tomcat中，然后会自动解包

打开浏览器访问[www.a.com:8080](http://www.a.com:8080)看到“师说”页面则成功

如果tomcat恢复默认了，就打开浏览器访问192.168.2.5:8080