**operation 运维 day 05**

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

**使用tomcat搭建网站服务**

**由于tomcat的运行需要依赖java，要先安装jdk工具包**

**1，配置java环境并拷贝tomcat程序到指定目录(该目录位置可以自定义)**

yum -y install java-1.8.0-openjdk //首先安装java环境软件包

cd ~/lnmp\_soft

tar -xf apache-tomcat-8.0.30.tar.gz //在lnmp\_soft目录下释放tomcat软件包

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

cd /usr/local/tomcat/

**tomcat目录介绍**

bin 存放tomcat主要程序

logs 日志

conf 存放配置文件

work 存放动态网站(页面)被解析时的临时文件

webapps 存放网站页面

bin/startup.sh //开启tomcat服务

bin/shutdown.sh //关闭tomcat服务

**2，运行前准备**

**由于tomcat运行需要random中有海量随机数，有时随机数不足会导致8005端口无法启动**

**解决方案一：**

mv /dev/random /dev/random.bak //备份原有random文件

ln -s /dev/urandom /dev/random //使用urandom创建软连接替代random即可

[root@web1 tomcat]# bin/shutdown.sh //关闭tomcat

yum -y install psmisc

killall java //如果tomcat异常时就杀死java程序

ss -ntulp | grep java

[root@web1 tomcat]# bin/startup.sh //重新开启tomcat服务

ss -ntulp | grep java //查询8005端口是否开启，该端口在服务关闭时需要，如果没有启动则服务不能使用，另外还有8080端口是提供网站服务的端口，8009端口可以做内部测试使用，也可以连接其他网站服务，目前的实验不使用，但必须看到这些端口都开启才说明tomcat服务正常

**解决方案二：**

yum -y install rng-tools //如果8005端口依然无法开启就安装该软件包

[root@web1 ~]# systemctl start rngd //开启服务

**3，测试tomcat服务**

<http://192.168.2.100:8080/> //使用浏览器访问tomcat默认页，由于tomcat没有使用默认的80端口而是8080，所以地址后面不要忘记写

**测试tomcat自定义静态页面与动态页面**

vim webapps/ROOT/test.jsp //编写基于java语言的动态页面

<html>

<body>

<center>

Now time is: <%=new java.util.Date()%> //显示服务器当前时间

</center>

</body>

</html>

http://192.168.2.100:8080/test.jsp //使用浏览器访问可以看到当前时间

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

**利用tomcat创建虚拟主机**

**回顾：**

**httpd的创建方法是写多个virtualhost**

<VirtualHost \*:80>

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

Documentroot /var/www/html

</virtualhost>

**nginx的创建方法是写多个server**

http {

server {

listen :80;

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

root html;

index index.html;

}

}

**Tomcat的创建方法是写多个Host**

<Host name=www.a.com appbase=webapps>

</Host>

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

**在tomcat中创建虚拟主机实际操作**

[root@web1 tomcat]# vim conf/server.xml //打开tomcat主配置文件，在122~124行添加以下内容

122 <Host name="www.b.com" appBase="web\_b" //name后面写域名，appbase后面写网站的页面存放目录，另外在后面第125行也就是默认的虚拟主机中的name由localhost修改为www.a.com

123 unpackWARs="true" autoDeploy="true">

124 </Host>

[root@web1 tomcat]# mkdir -p web\_b/ROOT //创建b网站的目录

echo "web\_b/ROOT/index.html" > web\_b/ROOT/index.html //编写b网站默认页面

echo "web\_a/ROOT/index.html" > webapps/ROOT/index.html //编写a网站的默认页面

vim /etc/hosts /修改本机hosts文件

192.168.2.100 www.a.com www.b.com www.c.com

curl www.b.com:8080 //测试

curl www.a.com:8080

如果用windows测试：

C:\Windows\System32\drivers\etc\hosts

192.168.2.100 www.a.com www.b.com www.c.com

使用浏览器访问 [www.a.com:8080](http://www.a.com:8080) 或者 www.b.com:8080

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

**另外，在虚拟主机中还有autoDeploy和unpackWARs其功能解释如下：**

**autoDeploy //自动更新网站功能，有助于开发工程师**

**unpackWARs //自动解war包，通常开发工程师习惯将网站文件打成war包(类似tar包)上传到服务器，但是还需要解包很麻烦，但有了这个功能就可以自动解开**

**测试自动解war包功能：**

yum -y install java-1.8.0-openjdk-devel //安装创建war包的工具

jar -cf xyz.war /var/log //到/下使用jar命令创建war包，名字是xyz.war，内容是/var/log

mv xyz.war /usr/local/tomcat/web\_b //把war包扔到虚拟主机的网页目录下

ls /usr/local/tomcat/web\_b //查看b网站目录里面会自动解开xyz.war包，并产生xyz目录，前提是tomcat服务正常开启中

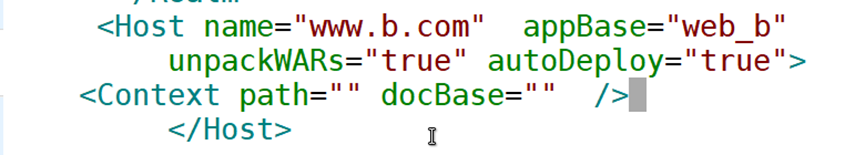
------------------------------------------------------------------

**tomcat页面路径测试**

**默认情况下，tomcat网站页面的文件存放在webapps目录的ROOT下，如果需要自定义路径可以按以下几种方式进行配置**

**种类1:**

在虚拟主机配置中添加Context path等内容当docBase为空时，访问页面会直接在web\_b目录中



echo "web\_b/test.html" > web\_b/test.html //创建测试页

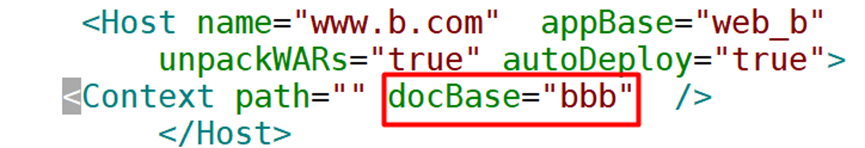
重启tomcat服务

<http://www.b.com:8080/test.html> //测试，看到的是web\_b

中的页面，而不是web\_b/ROOT下

**种类2:**

当docBase=”bbb”时，访问页面会在web\_b/bbb目录中



[root@web1 tomcat]# mkdir web\_b/bbb 创建测试目录

echo "web\_b/bbb/index.html" > web\_b/bbb/index.html 创建测试

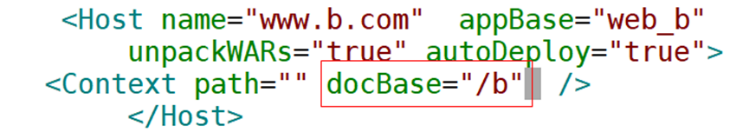
页面

重启tomcat服务

www.b.com:8080 访问测试，看到的是web\_b/bbb中的页面

**种类3:**

当docBase=”/b”时，访问页面会在/b目录中



[root@web1 tomcat]# mkdir /b 创建测试目录

[root@web1 tomcat]# echo " /b/index.html" > /b/index.html 创建

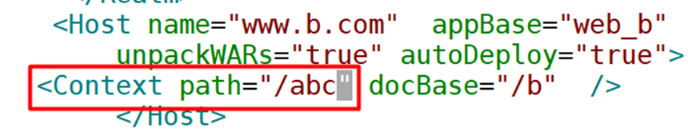
测试页面

重启tomcat服务

www.b.com:8080 访问测试，看到的是/b中的页面

**种类4:**

当context path=”/abc”时，docBase=”/b”时，访问页面www.b.com:8080/abc/ 会显示/b目录中的页面



重启tomcat服务

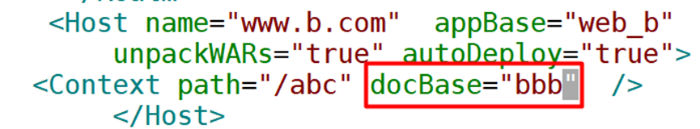
www.b.com:8080/abc/ 访问测试，看到的是/b中的页面

www.b.com:8080 访问测试，看到的是web\_b/ROOT中的页面

ss -ntulp | grep java

**种类5:**

当context path=”/abc”时，docBase=”bbb”时，访问页面www.b.com:8080/abc/ 会显示web\_b/bbb/目录中的页面



重启tomcat服务

curl www.b.com:8080/abc/ 访问测试，看到的是web\_b/bbb/中的

页面

curl www.b.com:8080 访问测试，看到的web\_b/ROOT/中的页面

[root@web1 tomcat]# ss -ntulp | grep java 如果实验失败要先查询

服务状态

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

**练习：**

下列需求如何实现？

访问www.b.com:8080 看到的是/usr/local/tomcat/abc/a中的内容

访问www.b.com:8080/abc/ 看到是/var/www/html中的内容

**答案：**

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

unpackWARs="true" autoDeploy="true">

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

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

</Host>

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

**使用tomcat开启安全加密效果**

首先修改配置文件，添加下图内容



keytool -genkeypair -alias tomcat -keyalg RSA -keystore /usr/local/to

mcat/keystore //在创建秘钥对文件，命令输入完毕后，先输入两

次123456的密码，然后回答问题，最后y确认，然后密码不用输

入直接回车。-genkeypair是创建密钥对，-alias是别名，-keyalg是用

什么算法，RSA是一种非对称加密算法，-keystore是密钥对文件的存储

路径，如果忘记该敲什么内容可以用—help查询

重启tomcat 服务

ss -ntulp | grep java 可以看到8443端口

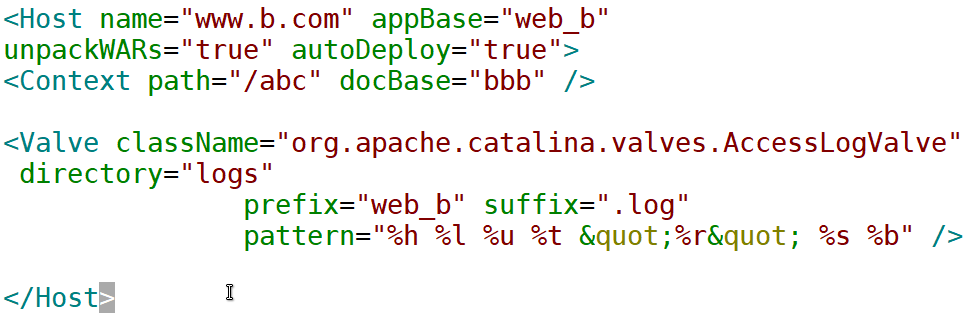
curl -k https://www.b.com:8443/ 查看安全加密页面

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

**为b网站虚拟主机添加日志**

**tomcat只为默认的虚拟主机添加了日志功能，如果新建的虚拟主机需要该功能就按下列方式配置：**

打开配置文件，添加日志功能<Valve ..... />(该配置从默认的虚拟主机中复制即可，最后一页)，修改prefix日志名字，suffix日志后缀



之后保存退出，重启tomcat服务，然后访问几次b网站

tail -1 logs/web\_b加tab键可以找到新日志并查看最后一行内容