

Tomcat配置:

查看Tomcat是否启动成功: 8080 8005 8009 这个三个端口

Tomcat默认端口是8080

```
#ss -nutlp | grep java
```

提示: 如果检查端口时, 8005端口启动非常慢, 默认tomcat启动需要从/dev/random读取大量的随机数据, 默认该设备生成随机数据的速度很慢, 可用使用下面的命令用urandom替换random (非必须操作)。

1. `[root@web1 ~]# mv /dev/random /dev/random.bak`
2. `[root@web1 ~]# ln -s /dev/urandom /dev/random` 做一个软连接

2 案例2: 使用Tomcat部署虚拟主机

2.1 问题

沿用练习二, 使用Tomcat部署加密虚拟主机, 实现以下要求:

- 实现两个基于域名的虚拟主机, 域名分别为: www.a.com和 www.b.com
- 使用www.a.com域名访问的页面根路径为/usr/local/tomcat/a/ROOT
- 使用www.b.com域名访问的页面根路径为/usr/local/tomcat/b/base
- 访问www.a.com/test时, 页面自动跳转到/var/www/html目录下的页面
- 访问页面时支持SSL加密通讯
- 私钥、证书存储路径为/usr/local/tomcat/conf/cert
- 每个虚拟主机都拥有独立的访问日志文件
- 配置tomcat集群环境

2.2 方案

修改server.xml配置文件, 创建两个域名的虚拟主机, 修改如下两个参数块:

1. `# cat /usr/local/tomcat/conf/server.xml`
2. `<Server>`
3. `<Service>`
4. `<Connector port=8080 />`
5. `<Connector port=8009 />`
6. `<Engine name="Catalina" defaultHost="localhost">`
7. `<Host name="www.a.com" appBase="a" unpackWARs="true" autoDeploy="true">`

8. `</Host>`
9. `<Host name="www.b.com" appBase="b" unpackWARS="true" autoDeploy="true">`
10. `</Host>`
11.

2.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：配置服务器虚拟主机

1) 修改server.xml配置文件，创建虚拟主机

1. `[root@web1 ~]# vim /usr/local/tomcat/conf/server.xml`
2.
3. `<Host name="www.a.com" appBase="a" unpackWARS="true" autoDeploy="true">`
4. `</Host>`
5. `<Host name="www.b.com" appBase="b" unpackWARS="true" autoDeploy="true">`
6. `</Host>`

2) 创建虚拟主机对应的页面根路径

1. `[root@web1 ~]# mkdir -p /usr/local/tomcat/{a,b}/ROOT`
2. `[root@web1 ~]# echo "AAA" > /usr/local/tomcat/a/ROOT/index.html`
3. `[root@web1 ~]# echo "BBB" > /usr/local/tomcat/b/ROOT/index.html`

3) 重启Tomcat服务器

1. `[root@web1 ~]# /usr/local/tomcat/bin/shutdown.sh`
2. `[root@web1 ~]# /usr/local/tomcat/bin/startup.sh`

4) 客户端设置host文件，并浏览测试页面进行测试(proxy充当客户端角色)

注意：ssh远程连接时使用使用-X参数才可以！！！！

1. `[root@proxy ~]# vim /etc/hosts`
2.
3. `192.168.2.100 www.a.com www.b.com`
4. `[root@proxy ~]# firefox http://www.a.com:8080/` //注意访问的端口为8080
5. `[root@proxy ~]# firefox http://www.b.com:8080/`

步骤二：修改www.b.com网站的首页目录为base

1) 使用docBase参数可以修改默认网站首页路径

1. `[root@web1 ~]# vim /usr/local/tomcat/conf/server.xml`
2. ...
3. `<Host name="www.a.com" appBase="a" unpackWARS="true" autoDeploy="true">`
4. `</Host>`
5. `<Host name="www.b.com" appBase="b" unpackWARS="true" autoDeploy="true">`
6. `<Context path="" docBase="base"/>` //决定于网页路径参数, 修改网页根路径参数
7. `</Host>`
8. ...
9. `[root@web1 ~]# mkdir /usr/local/tomcat/b/base`
10. `[root@web1 ~]# echo "BASE" > /usr/local/tomcat/b/base/index.html`
11. `[root@web1 ~]# /usr/local/tomcat/bin/shutdown.sh`
12. `[root@web1 ~]# /usr/local/tomcat/bin/startup.sh`

2) 测试查看页面是否正确(proxy充当客户端角色)

1. `[root@proxy ~]# firefox http://www.b.com:8080/` //结果为base目录下的页面内容

步骤三: 跳转

1) 当用户访问`http://www.a.com/test`打开`/var/www/html`目录下的页面

1. `[root@web1 ~]# vim /usr/local/tomcat/conf/server.xml`
2. ...
3. `<Host name="www.a.com" appBase="a" unpackWARS="true" autoDeploy="true">`
4. `<Context path="/test" docBase="/var/www/html/" />` 访问后面加/test, 跳转页面是var/www/html目录下的首页
5. `</Host>`
6. `<Host name="www.b.com" appBase="b" unpackWARS="true" autoDeploy="true">`
7. `<Context path="" docBase="base" />`
8. `</Host>`
9. ...
10. `[root@web1 ~]# echo "Test" > /var/www/html/index.html`
11. `[root@web1 ~]# /usr/local/tomcat/bin/shutdown.sh`
12. `[root@web1 ~]# /usr/local/tomcat/bin/startup.sh`

2) 测试查看页面是否正确(proxy充当客户端角色)

1. [root@proxy ~]# firefox http://www.a.com:8080/test

2. //返回/var/www/html/index.html的内容

3. //注意, 访问的端口为8080

步骤四: 配置Tomcat支持SSL加密网站

1) 创建加密用的私钥和证书文件

1. [root@web1 ~]# keytool -genkeypair -alias tomcat -keyalg RSA -
keystore /usr/local/tomcat/keystore //提示输入密码为:123456

2. //-genkeypair 生成密钥对

3. //-alias tomcat 密钥别名

4. //-keyalg RSA 定义密钥算法为RSA算法

5. //-keystore 定义密钥文件存储在:/usr/local/tomcat/keystore

2) 再次修改server.xml配置文件, 创建支持加密连接的Connector

1. [root@web1 ~]# vim /usr/local/tomcat/conf/server.xml

2.

3. <Connector port="8443"

protocol="org.apache.coyote.http11.Http11NioProtocol"

4. maxThreads="150" SSLEnabled="true" scheme="https" secure="true"

5. keystoreFile="/usr/local/tomcat/keystore" keystorePass="123456"

clientAuth="false" sslProtocol="TLS" />

6. //备注, 默认这段Connector被注释掉了, 打开注释, 添加密钥信息即可

3) 重启Tomcat服务器

1. [root@web1 ~]# /usr/local/tomcat/bin/shutdown.sh

2. [root@web1 ~]# /usr/local/tomcat/bin/startup.sh

4) 客户端设置host文件, 并浏览测试页面进行测试(proxy充当客户端角色)

1. [root@proxy ~]# vim /etc/hosts

2.

3. 192.168.2.100 www.a.com www.b.com

4. [root@proxy ~]# firefox https://www.a.com:8443/

5. [root@proxy ~]# firefox https://www.b.com:8443/

6. [root@proxy ~]# firefox https://192.168.2.100:8443/

步骤五: 配置Tomcat日志

1) 为每个虚拟主机设置不同的日志文件

1. [root@web1 ~]# vim /usr/local/tomcat/conf/server.xml

2. ...

3. `<Host name="www.a.com" appBase="a" unpackWARS="true" autoDeploy="true">`

4. `<Context path="/test" docBase="/var/www/html/" />`

5. #从默认localhost虚拟主机中把Valve这段复制过来，适当修改下即可

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

7. `prefix="a_access" suffix=".txt"`

8. `pattern="%h %l %u %t "%r" %s %b" />`

9. `</Host>`

10. `<Host name="www.b.com" appBase="b" unpackWARS="true" autoDeploy="true">`

11. `<Context path="" docBase="base" />`

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

13. `prefix="b_access" suffix=".txt"`

14. `pattern="%h %l %u %t "%r" %s %b" />`

15. `</Host>`

16. ...

2) 重启Tomcat服务器

1. `[root@web1 ~]# /usr/local/tomcat/bin/shutdown.sh`

2. `[root@web1 ~]# /usr/local/tomcat/bin/startup.sh`

3) 查看服务器日志文件

1. `[root@web1 ~]# ls /usr/local/tomcat/logs/`

步骤六：扩展实验(配置Tomcat集群)

1) 在192.168.4.5主机上配置Nginx调度器（具体安装步骤参考前面的章节）

1. `[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf`

2. `http{`

3. `upstream toms {`

4. `server 192.168.2.100:8080;`

5. `server 192.168.2.200:8080;`

6. `}`

7. `server {`

8. `listen 80;`

9. `server_name localhost;`

```
10.     location / {  
11.         proxy_pass http://toms;  
12.     }  
13. }  
14. }
```

2) 在192.168.2.100和192.168.2.200主机上配置Tomcat调度器

以下以Web1为例：

```
1. [root@web1 ~]# yum -y install java-1.8.0-openjdk           //安装JDK  
2. [root@web1 ~]# yum -y install java-1.8.0-openjdk-headless //安装  
JDK  
3. [root@web1 ~]# tar -xzf apache-tomcat-8.0.30.tar.gz  
4. [root@web1 ~]# mv apache-tomcat-8.0.30 /usr/local/tomcat
```

3) 启动服务

```
1. [root@web1 ~]# /usr/local/tomcat/bin/startup.sh
```

4) 客户端验证

为了防止有数据缓存，可以使用真实主机的google-chrome访问代理服务器，输入Ctrl+F5刷新页面。