

黑马23期Linux+Nginx

1. 安装JDK

卸载残留的jdk (Linux预装)

先查询

```
[atao@centos001 bin]$ java -version
java version "1.7.0_45"
OpenJDK Runtime Environment (rhel-2.4.3.3.el6-i386 u45-b15)
OpenJDK Client VM (build 24.45-b08, mixed mode, sharing)
[atao@centos001 bin]$ cd
[atao@centos001 ~]$ rpm -qa |grep java
tzdata-java-2013g-1.el6.noarch
java-1.7.0-openjdk-1.7.0.45-2.4.3.3.el6.i686
java-1.6.0-openjdk-1.6.0.0-1.66.1.13.0.el6.i686
```

发现有个jdk1.6的版本重复了

然后卸载 (需要root)

```
[atao@centos001 ~]$ sudo rpm -e --nodeps java-1.6.0-openjdk-1.6.0.0-1.66.1.13.0.el6.i686
[sudo] password for atao:
[atao@centos001 ~]$ rpm -qa |grep java
tzdata-java-2013g-1.el6.noarch
java-1.7.0-openjdk-1.7.0.45-2.4.3.3.el6.i686
[atao@centos001 ~]$
```

重装一次JDK, 换个目录

/bin: 存放的是一些二进制文件, 但是在 Linux 中二进制文件是可以被执行的。这个目录中的命令文件是给普通用户使用 (非超级管理员用户)。

/etc: Linux 下所有的配置文件都会存放到 etc 目录。

/home: 是所有非 root 用户家目录的一个集合。

/root: root 用户的家目录。

/sbin: sbin 表示 super bin, 里面同样也类似于 bin 目录, 也存放了一些二进制文件, 只不过这些命令只能是由超级管理员去运行。

/usr: 存放着一些用户自己安装的软件, 其实类似于 windows 下的 Program Files 目录。

/var: 存放着 Linux 下的一些日志文件, 在实际开发的时候有一些公司也习惯把 Apache 或者 nginx 的站点目录也会放到这个目录中。

首先需要安装依赖:

```
yum install glibc.i686
```

```
Updated:
glibc.i686 0:2.12-1.209.el6_9.2

Dependency Updated:
glibc-common.i686 0:2.12-1.209.el6_9.2 glibc-devel.i686 0:2.12-1.209.el6_9.2 glibc-headers.i686 0:2.12-1.209.el6_9.2
tzdata.noarch 0:2017b-1.el6

Complete!
[root@centos001 local]#
```

然后解压安装包

非root用户要sudo

```
sudo tar -xvf jdk-7u65-linux-i586.tar.gz -C /usr/local/java
```

```

总用量 19760
drwxr-xr-x. 8 uucp 143 4096 6月 17 2014 jdk1.7.0_65
[atao@centos001 java]$ cd jdk1.7.0_65/
[atao@centos001 jdk1.7.0_65]$ ll
总用量 19760
drwxr-xr-x. 2 uucp 143 4096 6月 17 2014 bin
-r--r--r--. 1 uucp 143 3339 6月 17 2014 COPYRIGHT
drwxr-xr-x. 4 uucp 143 4096 6月 17 2014 db
drwxr-xr-x. 3 uucp 143 4096 6月 17 2014 include
drwxr-xr-x. 5 uucp 143 4096 6月 17 2014 jre
drwxr-xr-x. 5 uucp 143 4096 6月 17 2014 lib
-r--r--r--. 1 uucp 143 40 6月 17 2014 LICENSE
drwxr-xr-x. 4 uucp 143 4096 6月 17 2014 man
-r--r--r--. 1 uucp 143 114 6月 17 2014 README.html
-rw-r--r--. 1 uucp 143 498 6月 17 2014 release
-rw-r--r--. 1 uucp 143 19902896 6月 17 2014 src.zip
-rw-r--r--. 1 uucp 143 110114 6月 17 2014 THIRDPARTYLICENSEREADME-JAVAFX.txt
-r--r--r--. 1 uucp 143 173559 6月 17 2014 THIRDPARTYLICENSEREADME.txt
[atao@centos001 jdk1.7.0_65]$

```

配置环境变量

方法一：

首先克隆session，找出jdk安装目录

```

drwxr-xr-x. 8 uucp 143 4096 6月 17 2014 jdk1.7.0_65
[atao@centos001 java]$ pwd
/usr/local/java
[atao@centos001 java]$ cd jdk1.7.0_65/
[atao@centos001 jdk1.7.0_65]$ pwd
/usr/local/java/jdk1.7.0_65
[atao@centos001 jdk1.7.0_65]$

```

然后 sudo vim /etc/profile

末尾插入

```

70 . "$1"
71 else
72 . "$i" >/dev/null 2>&1
73 fi
74 fi
75 done
76
77 unset i
78 unset -f pathmunge
79 export JAVA_HOME=/usr/local/java/jdk1.7.0_65
80 export PATH=$PATH:$JAVA_HOME/bin
-- 插入 --

```

然后完成

```

[atao@centos001 jdk1.7.0_65]$ cd
[atao@centos001 ~]$ java -version
java version "1.7.0_65"
Java(TM) SE Runtime Environment (build 1.7.0_65-b17)
Java HotSpot(TM) Client VM (build 24.65-b04, mixed mode)
[atao@centos001 ~]$

```

也可以第二种方法

vim /etc/profile

```

#set java environment
JAVA_HOME=/usr/local/src/java/jdk1.7.0_71
CLASSPATH=.:$JAVA_HOME/lib/tools.jar
PATH=$JAVA_HOME/bin:$PATH
export JAVA_HOME CLASSPATH PATH

```

重新加载配置文件：

```
source /etc/profile
```

2. 安装mysql

先卸载预装的MySQL

```
Java HotSpot(TM) Client VM (build 24.65-b04, mix
[atao@centos001 ~]$ rpm -qa |grep mysql
mysql-libs-5.1.73-8.el6_8.i686
[atao@centos001 ~]$
```

Ready

```
sudo rpm -e --nodeps mysql-libs-5.1.73-8.el6_8.i686
```

```
[atao@centos001 ~]$ rpm -e --nodeps mysql-libs-5.1.73-8.el6_8.i686
error: can't create transaction lock on /var/lib/rpm/.rpm.lock (权限不够)
[atao@centos001 ~]$ sudo rpm -e --nodeps mysql-libs-5.1.73-8.el6_8.i686
[sudo] password for atao:
[atao@centos001 ~]$ rpm -qa |grep mysql
[atao@centos001 ~]$
```

切换目录，新建mysql目录

```
[atao@centos001 local]$ sudo mkdir mysql
[atao@centos001 local]$ ll
总用量 44
drwxr-xr-x. 2 root root 4096 9月 23 2011 bin
drwxr-xr-x. 2 root root 4096 9月 23 2011 etc
drwxr-xr-x. 2 root root 4096 9月 23 2011 games
drwxr-xr-x. 2 root root 4096 9月 23 2011 include
drwxr-xr-x. 4 root root 4096 8月 21 10:31 java
drwxr-xr-x. 2 root root 4096 9月 23 2011 lib
drwxr-xr-x. 2 root root 4096 9月 23 2011 libexec
drwxr-xr-x. 2 root root 4096 8月 21 10:49 mysql
drwxr-xr-x. 2 root root 4096 9月 23 2011 sbin
drwxr-xr-x. 5 root root 4096 4月 10 04:53 share
drwxr-xr-x. 2 root root 4096 9月 23 2011 src
[atao@centos001 local]$
```

从本地磁盘复制下载好的tar文件到Linux中

然后解压

```
sudo tar -xvf MySQL-5.5.56-1.el6.i686.rpm-bundle.tar -C /usr/local/mysql/
```

```
[atao@centos001 ~]$ sudo tar -xvf MySQL-5.5.56-1.el6.i686.rpm-bundle.tar -C /usr/local/mysql/
MySQL-server-5.5.56-1.el6.i686.rpm
MySQL-devel-5.5.56-1.el6.i686.rpm
MySQL-embedded-5.5.56-1.el6.i686.rpm
MySQL-test-5.5.56-1.el6.i686.rpm
MySQL-client-5.5.56-1.el6.i686.rpm
MySQL-shared-compat-5.5.56-1.el6.i686.rpm
MySQL-shared-5.5.56-1.el6.i686.rpm
[atao@centos001 ~]$
```

安装server和client

先安装server

切换到MySQL目录

然后

```
sudo rpm -ivh MySQL-server-5.5.56-1.el6.i686.rpm
```

```

MySQL-shared-5.5.56-1.el6.i686.rpm
[atao@centos001 ~]$ sudo rpm -ivh MySQL-server-5.5.56-1.el6.i686.rpm
error: open of MySQL-server-5.5.56-1.el6.i686.rpm failed: 没有那个文件或目录
[atao@centos001 ~]$ cd /usr/local/mysql/
[atao@centos001 mysql]$ sudo rpm -ivh MySQL-server-5.5.56-1.el6.i686.rpm
warning: MySQL-server-5.5.56-1.el6.i686.rpm: Header V3 DSA/SHA1 Signature, key ID 5072e1f5: NOKEY
Preparing... ##### [100%]
1:MySQL-server ##### [100%]
warning: user mysql does not exist - using root
warning: group mysql does not exist - using root
170821 10:59:42 [Note] Ignoring --secure-file-priv value as server is running with --bootstrap.
170821 10:59:42 [Note] /usr/sbin/mysqld (mysqld 5.5.56) starting as process 8746 ...
170821 10:59:42 [Note] Ignoring --secure-file-priv value as server is running with --bootstrap.
170821 10:59:42 [Note] /usr/sbin/mysqld (mysqld 5.5.56) starting as process 8753 ...

```

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
To do so, start the server, then issue the following commands:

```

/usr/bin/mysqladmin -u root password 'new-password'
/usr/bin/mysqladmin -u root -h centos001 password 'new-password'

```

Alternatively you can run:
/usr/bin/mysql_secure_installation

which will also give you the option of removing the test
databases and anonymous user created by default. This is
strongly recommended for production servers.

See the manual for more instructions.

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
To do so, start the server, then issue the following commands:

```

/usr/bin/mysqladmin -u root password 'new-password'
/usr/bin/mysqladmin -u root -h centos001 password 'new-password'

```

Alternatively you can run:
/usr/bin/mysql_secure_installation

which will also give you the option of removing the test
databases and anonymous user created by default. This is
strongly recommended for production servers.

See the manual for more instructions.

Please report any problems at <http://bugs.mysql.com/>

```

[atao@centos001 mysql]$ █

```

再装client

```

sudo rpm -ivh MySQL-client-5.5.56-1.el6.i686.rpm

```

```

[atao@centos001 mysql]$ sudo rpm -ivh MySQL-client-5.5.56-1.el6.i686.rpm
warning: MySQL-client-5.5.56-1.el6.i686.rpm: Header V3 DSA/SHA1 Signature, key ID 5072e1f5: NOKEY
Preparing... ##### [100%]
1:MySQL-client ##### [100%]
[atao@centos001 mysql]$

```

dy

ssh2: AES-256-CTR 31, 25 31 Rows, 12

安装好后, 对mysql进行设置

先开服务

启动mysql服务

```

service mysql start

```

```

Cleaning up...
[atao@centos001 mysql]$ sudo service mysql start
Starting MySQL.Logging to '/var/lib/mysql/centos001.err'.
..[确定]
[atao@centos001 mysql]$ █

```

```

sudo /usr/bin/mysql_secure_installation

```

```
1. MySQL Client ##### [10]
[atao@centos001 mysql]$ sudo /usr/bin/mysql_secure_installation
```

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MySQL
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MySQL to secure it, we'll need the current
password for the root user. If you've just installed MySQL, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):

第一次是没有密码的。直接回车就好

```
Enter current password for root (enter for none):
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
Enter current password for root (enter for none):
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
Enter current password for root (enter for none):
OK, successfully used password, moving on...
```

Setting the root password ensures that nobody can log into the MySQL
root user without the proper authorisation.

Set root password? [Y/n] █

设置password 123

```
Set root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!
```

By default, a MySQL installation has an anonymous user, allowing anyone
to log into MySQL without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] █

Normally, root should only be allowed to connect from 'localhost'. This ensures that someone cannot guess at the root password from the network.

```
Disallow root login remotely? [Y/n] n
... skipping.
```

By default, MySQL comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.

```
Remove test database and access to it? [Y/n] n
... skipping.
```

Reloading the privilege tables will ensure that all changes made so far will take effect immediately.

```
Reload privilege tables now? [Y/n] y
... Success!
```

立刻加载个人设置

Cleaning up...

All done! If you've completed all of the above steps, your MySQL installation should now be secure.

Thanks for using MySQL!

设置完成

登录mysql

```
[atao@centos001 mysql]$ mysql -uroot -p123
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 5.5.56 MySQL Community Server (GPL)

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql      |
| performance_schema |
| test      |
+-----+
4 rows in set (0.00 sec)

mysql>
```

退出

```
mysql> exit
Bye
[atao@centos001 mysql]$
```

设置 自启动 可以在开机的时候用service mysql status来检测服务是否已经开启，如果开启就不用设置了

2.3. 系统启动时自动启动 mysql 服务

加入到系统服务:

```
chkconfig --add mysql
```

自动启动:

```
chkconfig mysql on
```

查询列表:

```
chkconfig
```

说明: 都没关闭 (off) 时是没有自动启动。

```
/etc/mysql/mysql.conf.d/mysqld.cn: line 54: /var/lib/mysql/localhost.localdomain.pid: Permission denied
MySQL is not running, but PID file exists[失败]
[atao@localhost ~]$ sudo chkconfig --add mysql
[atao@localhost ~]$ sudo chkconfig mysql on
[atao@localhost ~]$ chkconfig
NetworkManager 0:关闭 1:关闭 2:启用 3:启用 4:启用 5:启用 6:关闭
abrt-ccpp       0:关闭 1:关闭 2:关闭 3:启用 4:关闭 5:启用 6:关闭
abrt-d         0:关闭 1:关闭 2:关闭 3:启用 4:关闭 5:启用 6:关闭
acpid          0:关闭 1:关闭 2:启用 3:启用 4:启用 5:启用 6:关闭
atd            0:关闭 1:关闭 2:关闭 3:启用 4:启用 5:启用 6:关闭
auditd         0:关闭 1:关闭 2:启用 3:启用 4:启用 5:启用 6:关闭
autofs         0:关闭 1:关闭 2:关闭 3:启用 4:启用 5:启用 6:关闭
blk-availability 0:关闭 1:关闭 2:关闭 3:启用 4:启用 5:启用 6:关闭
blk-availability 0:关闭 1:启用 2:启用 3:启用 4:启用 5:启用 6:关闭
```

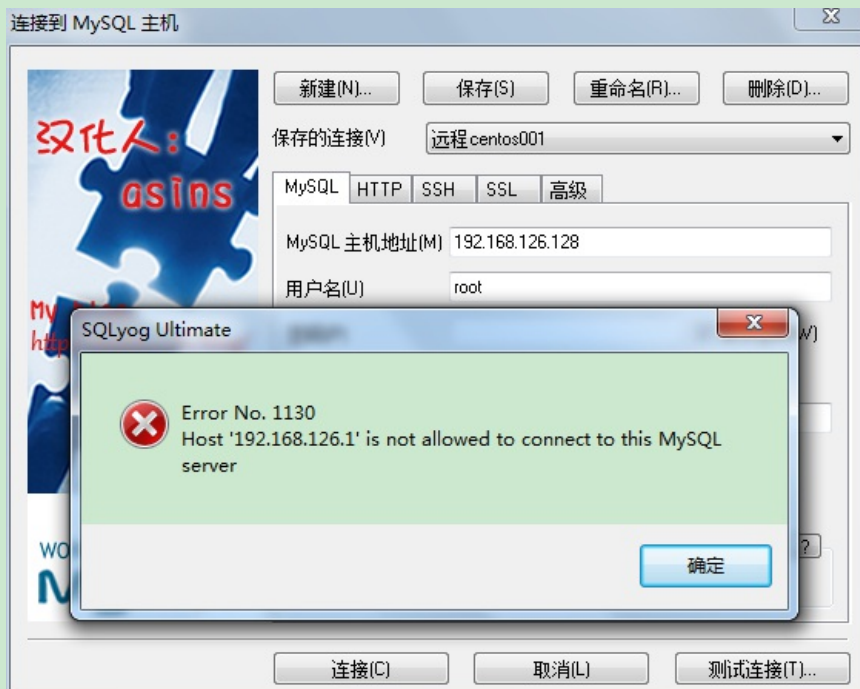
Ready

```
yppbind 0:关闭 1:关闭 2:关闭 3:关闭 4:关闭 5:关闭 6:关闭
[atao@localhost ~]$ chkconfig |grep mysql
mysql 0:关闭 1:关闭 2:启用 3:启用 4:启用 5:启用 6:关闭
[atao@localhost ~]$
```

Ready

之前上一步安装的时候有设置的话其实也不用设置远程?

连接到 MySQL 主机



连不上, 需要设置

开启远程访问

登录:

```
mysql -uroot -p123
```

```
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
[atao@localhost ~]$ sudo mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.5.56 MySQL Community Server (GPL)

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Ready

Feb 2, A

设置远程访问（使用root密码）：

```
grant all privileges on *.* to 'root' @'%' identified by '123';
flush privileges;
```

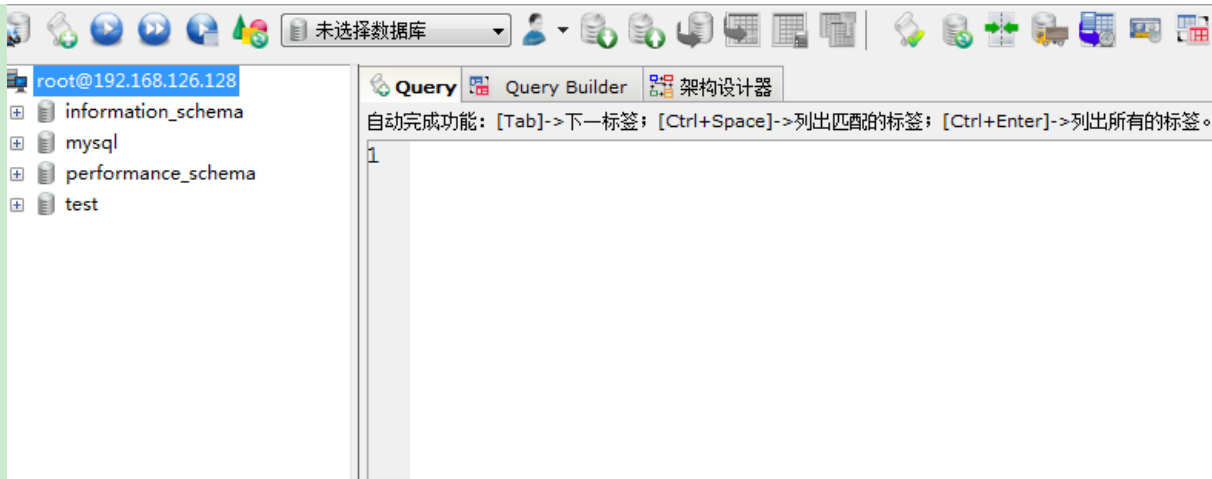
防火墙打开3306端口（我的做法是直接卸载防火墙，常见之前的做法）

```
/sbin/iptables -I INPUT -p tcp --dport 3306 -j ACCEPT
/etc/rc.d/init.d/iptables save
/etc/init.d/iptables status
```

不建议卸载防火墙，服务器会有安全问题

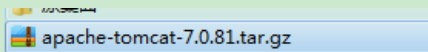
设置好后，



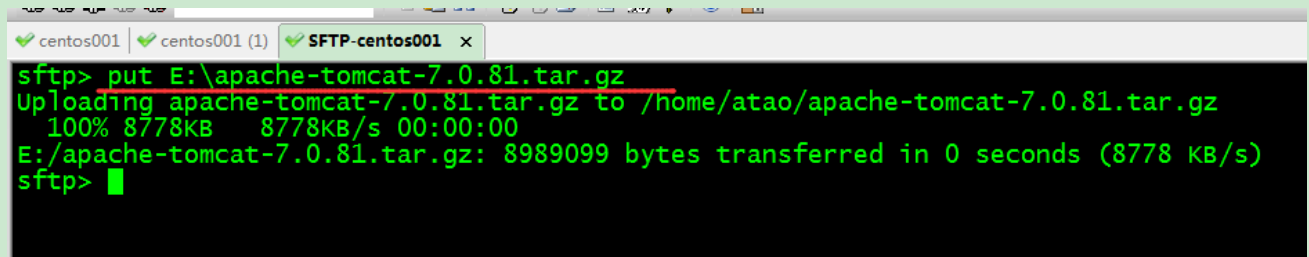


3. 安装tomcat

本地下载

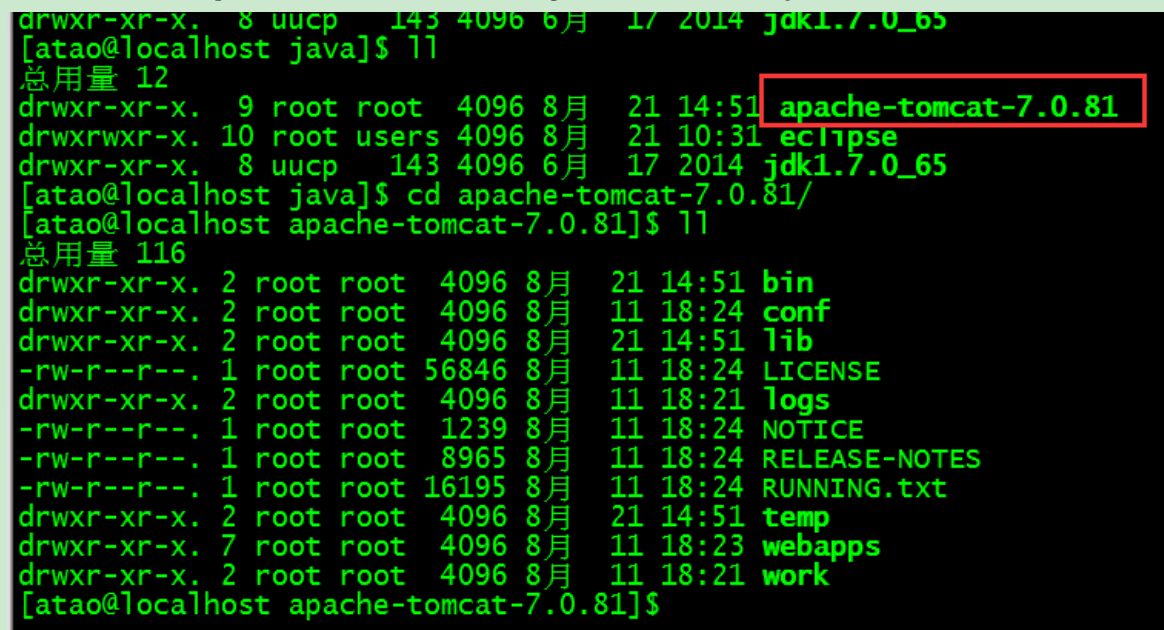


上传到linux



解压

`sudo tar -xvf apache-tomcat-7.0.81.tar.gz -C /usr/local/java`



启动tomcat（必须切换到root用户）

命令

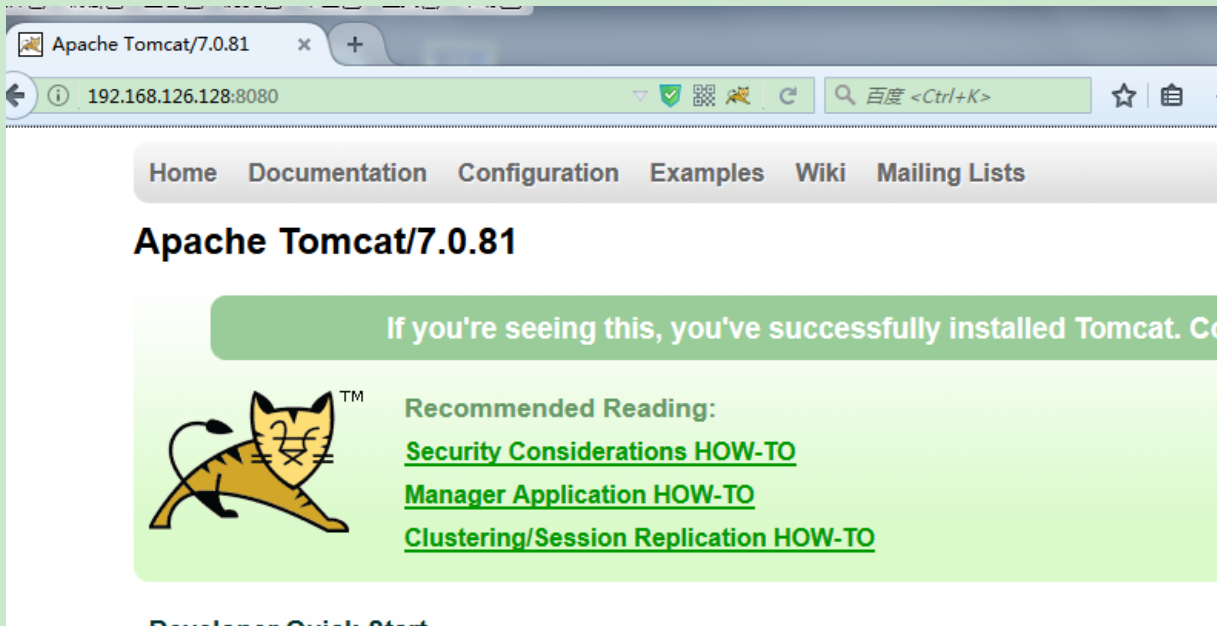
`./startup.sh`

```

密码:
[root@localhost bin]# ./startup.sh
Using CATALINA_BASE:   /usr/local/java/apache-tomcat-7.0.81
Using CATALINA_HOME:   /usr/local/java/apache-tomcat-7.0.81
Using CATALINA_TMPDIR: /usr/local/java/apache-tomcat-7.0.81/temp
Using JRE_HOME:        /usr/local/java/jdk1.7.0_65
Using CLASSPATH:       /usr/local/java/apache-tomcat-7.0.81/bin/bootstrap.jar:/usr/local/java/apache-tomcat-7.0.81/bin/
mcatal-juli.jar
Tomcat started.
[root@localhost bin]#

```

浏览器访问<http://192.168.126.128:8080/>



如果配置了host文件（windows的host文件以及Linux的/etc/hosts）

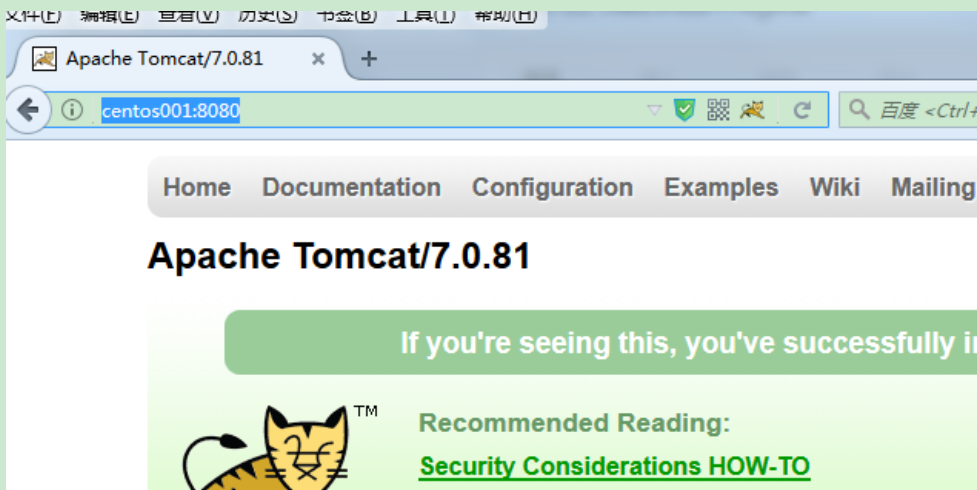
```

#          38.25.63.10      x.acme.com          # x client host

# localhost name resolution is handled within DNS itself.
#   127.0.0.1      localhost
#       ::1        localhost
192.168.126.128    centos001

```

<http://centos001:8080/>



关闭服务器

`./shutdown.sh`

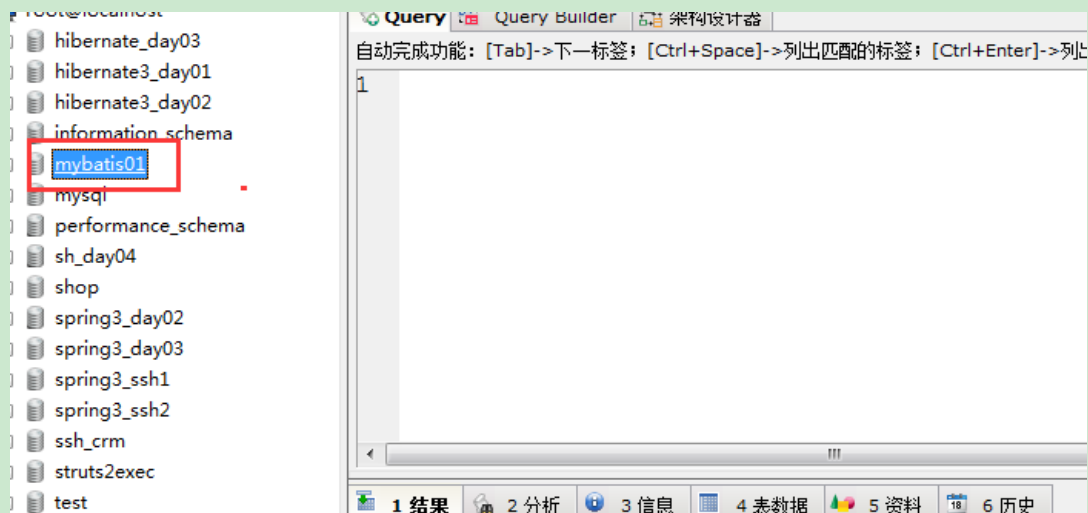
```

tomcat started.
[root@localhost bin]# ./shutdown.sh
Using CATALINA_BASE:   /usr/local/java/apache-tomcat-7.0.81
Using CATALINA_HOME:   /usr/local/java/apache-tomcat-7.0.81
Using CATALINA_TMPDIR: /usr/local/java/apache-tomcat-7.0.81/temp
Using JRE_HOME:        /usr/local/java/jdk1.7.0_65
Using CLASSPATH:       /usr/local/java/apache-tomcat-7.0.81/bin/bootstrap.jar:/usr/local/java/apache-tomcat-7.0.81/bin/t
mcatal-juli.jar
[root@localhost bin]#

```

4. 把项目发布到Linux上面

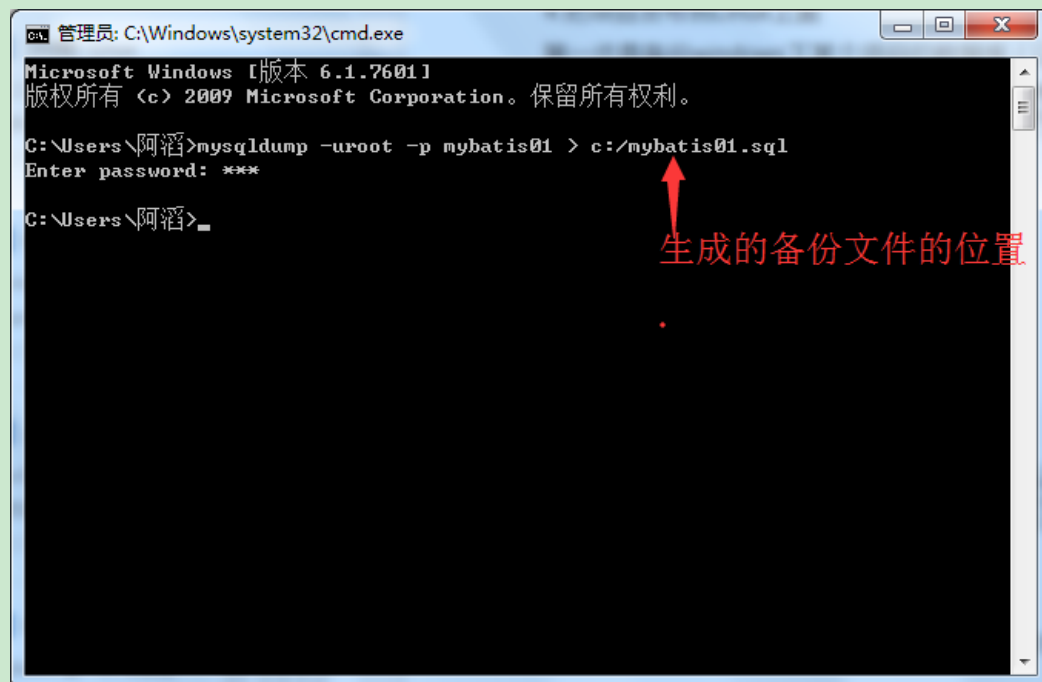
第一步是备份windows下某个项目的数据库（以便于在linux还原数据库）



怎么备份？

第一种方式——命令行

```
mysqldump -uroot -p mybatis01 > c:/mybatis01.sql
```



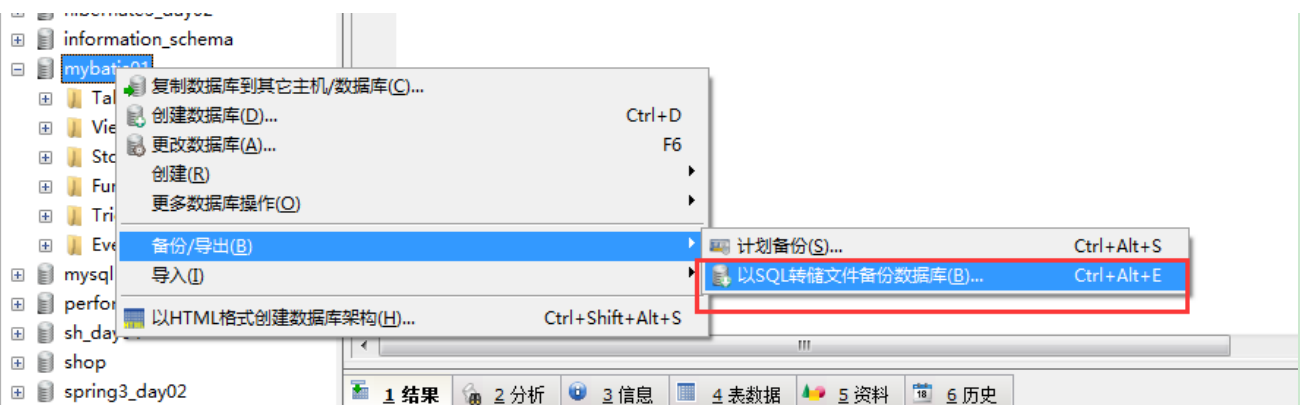
mybatis01.sql

2017/8/21 15:15

SQL 文件

6 KB

第二种 图形化界面



然后在Linux下，先登录，创建一个新的数据库

```
mysql> create database mybatis01
-> ;
Query OK, 1 row affected (0.05 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mybatis01 |
| mysql |
| performance_schema |
| test |
+-----+
5 rows in set (0.01 sec)

mysql>
```

然后把windows下的备份sql文件传到Linux

```
sftp> put C:\mybatis01.sql
Uploading mybatis01.sql to /home/atao/mybatis01.sql
100% 5KB 5KB/s 00:00:00
c:/mybatis01.sql: 5954 bytes transferred in 0 seconds (5954 bytes/s)
sftp>
```

查看

```
[root@localhost ~]# cd /home/atao
[root@localhost atao]# ll
总用量 290740
-rw-rw-r--. 1 atao atao 8989099 8月 21 14:44 apache-tomcat-7.0.81.tar.gz
-rw-rw-r--. 1 atao atao 35 8月 20 17:25 hello.txt
-rw-rw-r--. 1 atao atao 143588167 7月 18 2014 jdk-7u65-linux-i586.tar.gz
-rw-rw-r--. 1 atao atao 5954 8月 21 15:15 mybatis01.sql
-rwxrwxrwx. 1 root root 145090560 8月 21 10:45 MySQL-5.5.56-1.el6.i686.rpm-bundle.tar
drwxr-xr-x. 2 atao atao 4096 8月 21 14:03 公共的
drwxr-xr-x. 2 atao atao 4096 8月 21 14:03 模板
drwxr-xr-x. 2 atao atao 4096 8月 21 14:03 视频
drwxr-xr-x. 2 atao atao 4096 8月 21 14:03 图片
drwxr-xr-x. 2 atao atao 4096 8月 21 14:03 文档
drwxr-xr-x. 2 atao atao 4096 8月 21 14:03 下载
drwxr-xr-x. 2 atao atao 4096 8月 21 14:03 音乐
drwxr-xr-x. 2 atao atao 4096 8月 21 14:03 桌面
[root@localhost atao]#
```

再登录mysql

执行

```
source /home/atao/mybatis01.sql
```

```
+-----+
5 rows in set (0.00 sec)

mysql> use mybatis01;
Database changed
mysql> source /home/atao/mybatis01.sql
```

回车执行

```
centos001 x SFTP-centos001

Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
Query OK, 7 rows affected (0.00 sec)
Records: 7  Duplicates: 0  Warnings: 0
Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
mysql> █
```

检查一下是否有东西

```
mysql> show tables;
+-----+
| Tables_in_mybatis01 |
+-----+
| items                |
| orderdetail          |
| orders               |
| user                  |
+-----+
4 rows in set (0.00 sec)

mysql> █
```

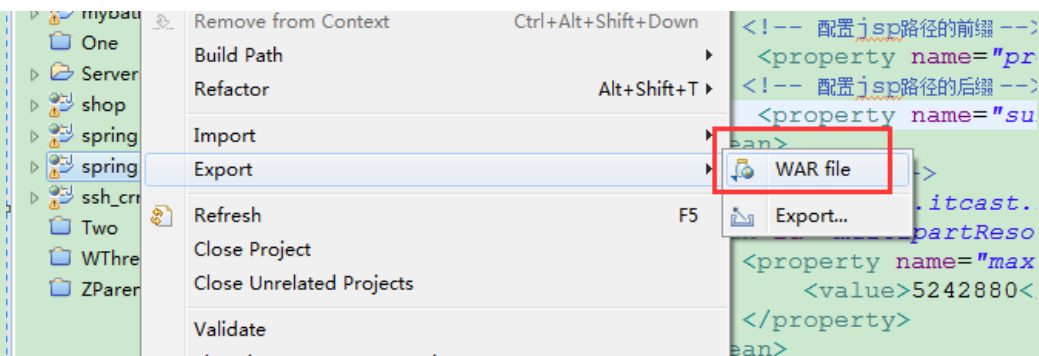
当然也可以用sqlyog进行操作

下面是第二步

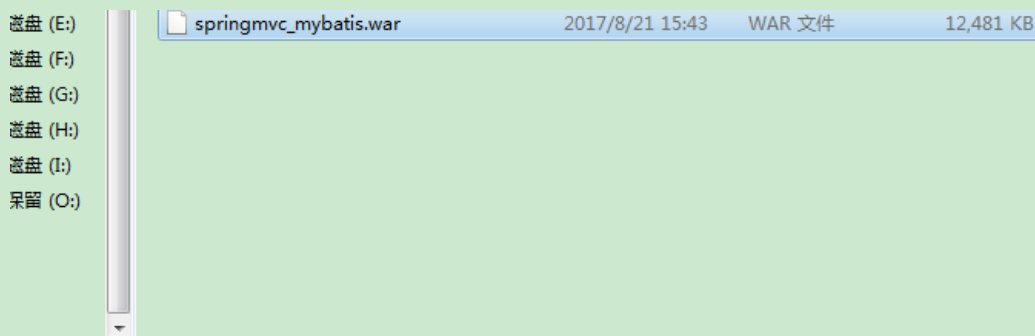
发布项目到Linux

首先，把项目导出war包

```
> springmvc_first
> springmvc_mybatis
> ssh_crm
```



设置导出到c盘



springmvc_mybatis.war 修改日期: 2017/8/21 15:43 创建日期: 2017/8/21 15:42
WAR 文件 大小: 12.1 MB

然后上传到linux

使用filezilla上传到Linux 的 root目录下



```

[root@localhost webapps]# cd
[root@localhost ~]# ll
总用量 12696
-rw-r--r--. 1 root root      1641 4月 10 05:12 anaconda-ks.cfg
-rw-r--r--. 1 root root    47446 8月 20 14:58 dabaowenjian.tar.gz
-rw-r--r--. 1 root root    45503 4月 10 05:12 install.log
-rw-r--r--. 1 root root    10033 4月 10 05:10 install.log.syslog
-rwxr-xr-x. 1 root root      1625 8月  6 13:32 javapwd.txt
-rw-r--r--. 1 root root      2310 8月  7 13:25 passwd
drwxr-xr-x. 2 root root      4096 8月 20 15:11 pck001
-rw-r--r--. 1 root root    47430 8月 20 15:12 pzk001.tar.gz
-rw-r--r--. 1 root root 12780025 8月 21 15:45 springmvc_mybatis.war
drwxr-xr-x. 3 root root      4096 8月 21 10:32 workspace
drwxr-xr-x. 2 root root      4096 4月  9 21:21 公共的
drwxr-xr-x. 2 root root      4096 4月  9 21:21 模板
drwxr-xr-x. 2 root root      4096 4月  9 21:21 视频
drwxr-xr-x. 2 root root      4096 4月  9 21:21 图片
drwxr-xr-x. 2 root root      4096 4月  9 21:21 文档

```

复制一份到Tomcat目录中

```
cp springmvc_mybatis.war /usr/local/java/apache-tomcat-7.0.81/webapps
```

```

[root@localhost ~]# cp springmvc_mybatis.war /usr/local/java/apache-tomcat-7.0.81/webapps
[root@localhost ~]# cd /usr/local/java/apache-tomcat-7.0.81/webapps
[root@localhost webapps]# ll
总用量 12504
drwxr-xr-x. 14 root root      4096 8月 21 14:51 docs
drwxr-xr-x.  7 root root      4096 8月 21 14:51 examples
drwxr-xr-x.  5 root root      4096 8月 21 14:51 host-manager
drwxr-xr-x.  5 root root      4096 8月 21 14:51 manager
drwxr-xr-x.  3 root root      4096 8月 21 14:51 ROOT
-rw-r--r--.  1 root root 12780025 8月 21 15:47 springmvc_mybatis.war
[root@localhost webapps]#

```

一启动Tomcat，war包就自动解压（webapps目录下）

```

[root@localhost apache-tomcat-7.0.81]# cd bin/
[root@localhost bin]# ./startup.sh
Using CATALINA_BASE:   /usr/local/java/apache-tomcat-7.0.81
Using CATALINA_HOME:   /usr/local/java/apache-tomcat-7.0.81
Using CATALINA_TMPDIR: /usr/local/java/apache-tomcat-7.0.81/temp
Using JRE_HOME:        /usr/local/java/jdk1.7.0_65
Using CLASSPATH:       /usr/local/java/apache-tomcat-7.0.81/bin/bootstrap.jar:/usr/
mcatal-juli.jar
Tomcat started.
[root@localhost bin]#

```

自动解压后

```

[root@localhost bin]# cd ../webapps
[root@localhost webapps]# ll
总用量 12508
drwxr-xr-x. 14 root root      4096 8月 21 14:51 docs
drwxr-xr-x.  7 root root      4096 8月 21 14:51 examples
drwxr-xr-x.  5 root root      4096 8月 21 14:51 host-manager
drwxr-xr-x.  5 root root      4096 8月 21 14:51 manager
drwxr-xr-x.  3 root root      4096 8月 21 14:51 ROOT
drwxr-xr-x.  5 root root      4096 8月 21 15:51 springmvc_mybatis
-rw-r--r--.  1 root root 12780025 8月 21 15:47 springmvc_mybatis.war
[root@localhost webapps]#

```

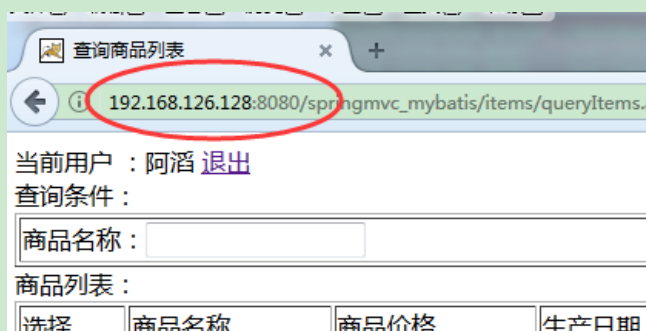
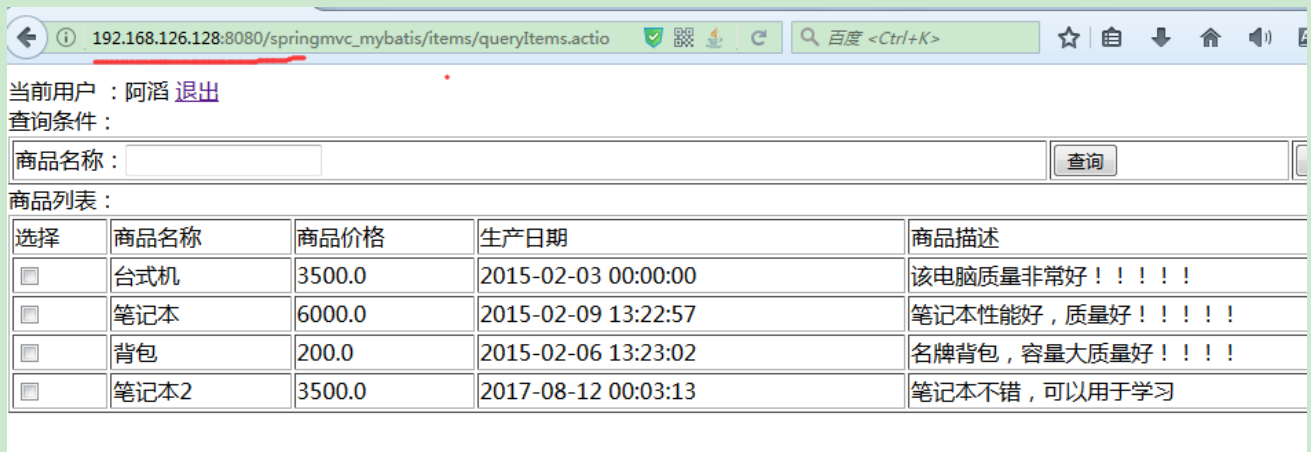
然后修改项目的配置文件——数据库连接信息


```
~
~
~
~
"db.properties" [dos] 4L, 128C 已写入
[root@localhost classes]#
```

然后重启tomcat

```
Tomcat started.
[root@localhost bin]# ./shutdown.sh
Using CATALINA_BASE:   /usr/local/java/apache-tomcat-7.0.81
Using CATALINA_HOME:   /usr/local/java/apache-tomcat-7.0.81
Using CATALINA_TMPDIR: /usr/local/java/apache-tomcat-7.0.81/temp
Using JRE_HOME:        /usr/local/java/jdk1.7.0_65
Using CLASSPATH:        /usr/local/java/apache-tomcat-7.0.81/bin/bootstrap.jar:/usr/local/java/ap
mcatal-juli.jar
[root@localhost bin]# ./startup.sh
Using CATALINA_BASE:   /usr/local/java/apache-tomcat-7.0.81
Using CATALINA_HOME:   /usr/local/java/apache-tomcat-7.0.81
Using CATALINA_TMPDIR: /usr/local/java/apache-tomcat-7.0.81/temp
Using JRE_HOME:        /usr/local/java/jdk1.7.0_65
Using CLASSPATH:        /usr/local/java/apache-tomcat-7.0.81/bin/bootstrap.jar:/usr/local/java/ap
mcatal-juli.jar
Tomcat started.
[root@localhost bin]#
```

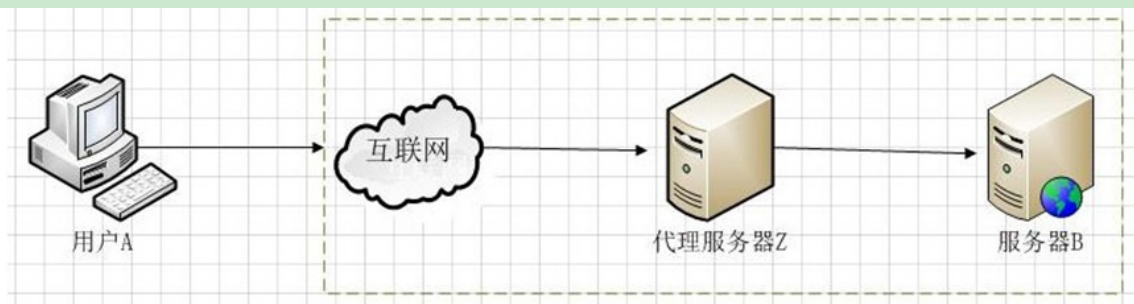
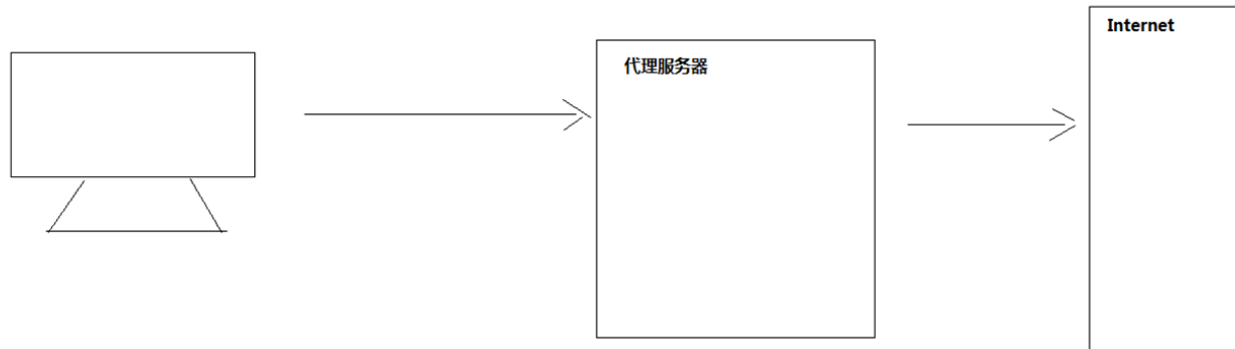
在windows本地访问



5. Nginx的概述

正向代理

正向代理

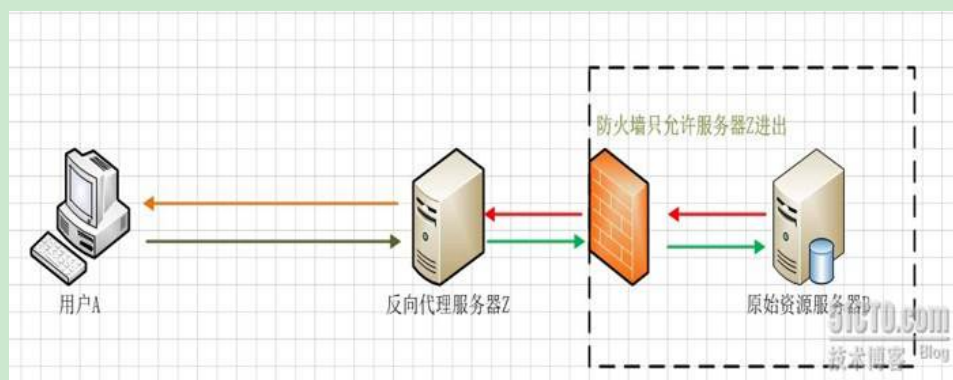
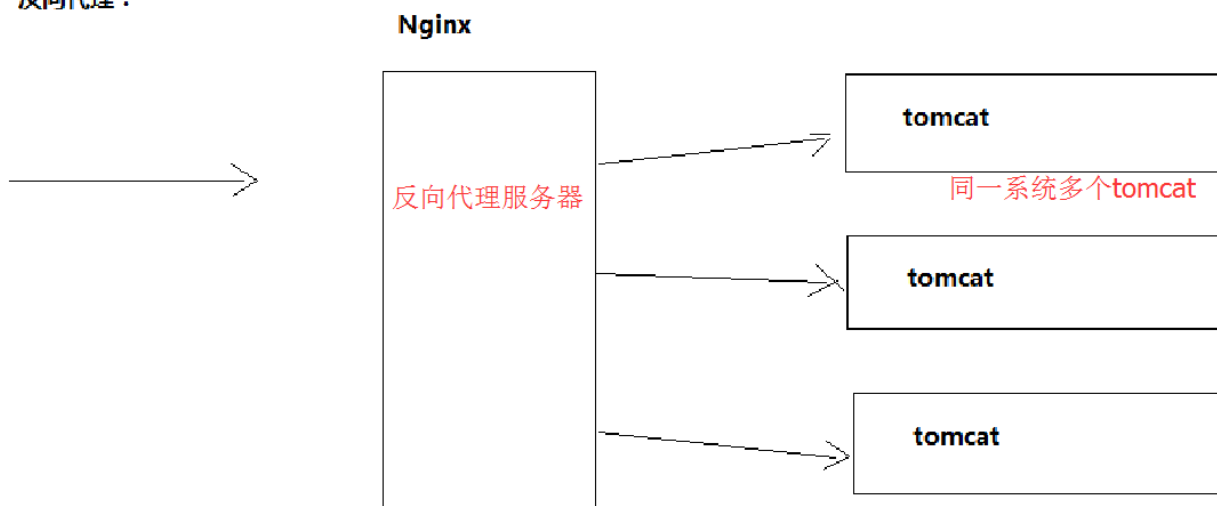


(图 1.1)

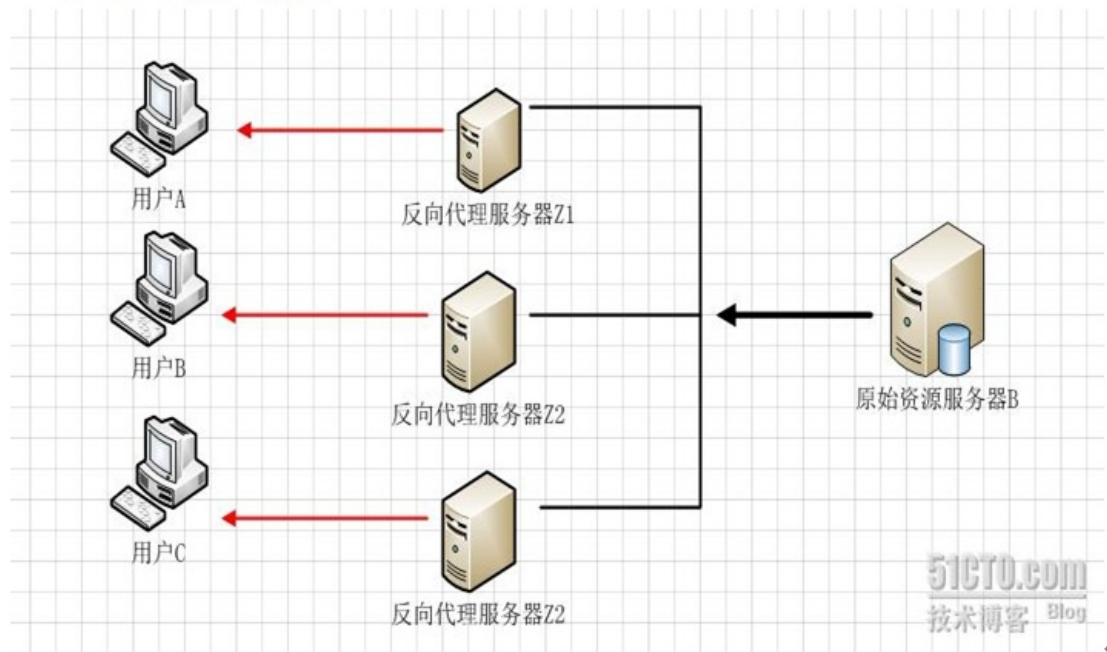
从上面的概念中，我们看出，文中所谓的**正向代理**就是**代理服务器**替代访问方【用户A】去访问目标服务器【服务器B】。

反向代理

反向代理：



2、负载均衡如下图 2.2



(图 2.2)

正向代理与反向代理

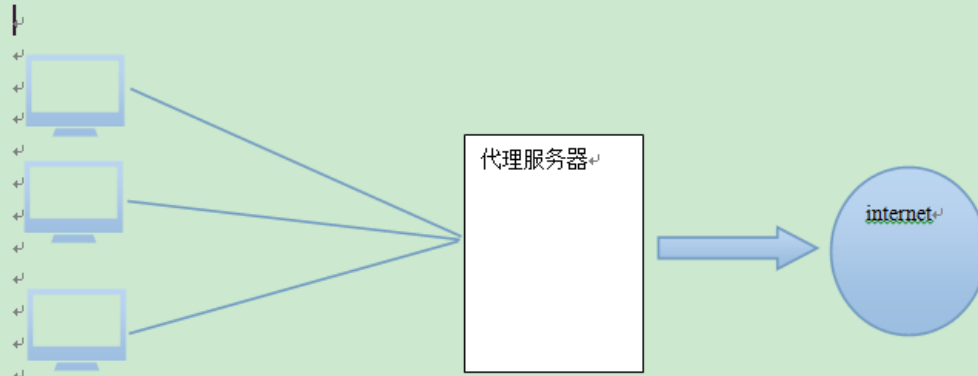
正向代理是一个位于客户端和原始服务器(origin server)之间的服务器，为了从原始服务器取得内容，客户端向代理发送一个请求并指定目标(原始服务器)，然后代理向原始服务器转交请求并将获得的内容返回给客户端。客户端必须设置正向代理服务器，当然前提是要知道正向代理服务器的IP地址，还有代理程序的端口。

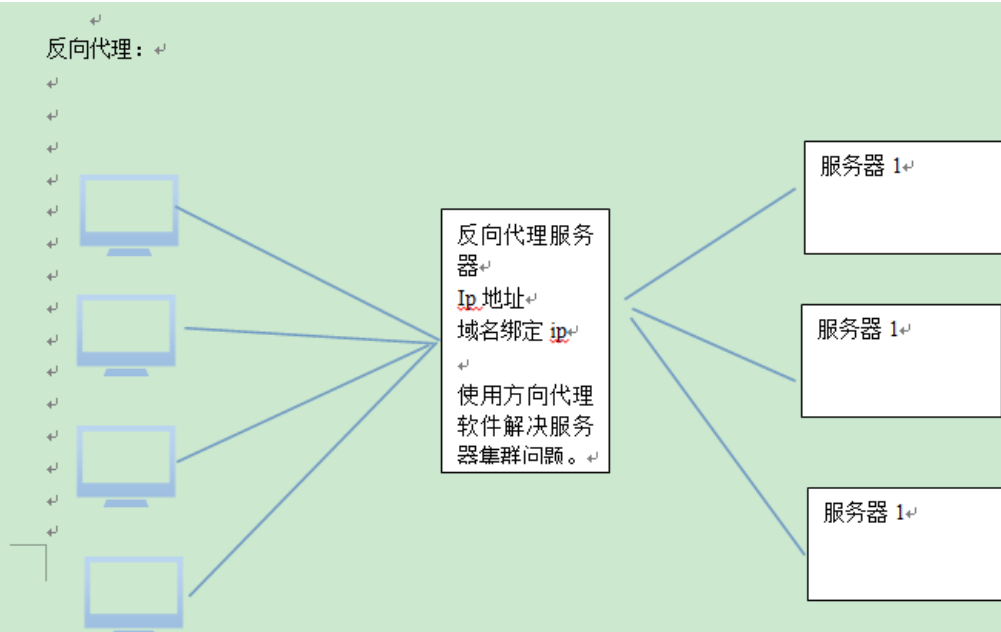
反向代理正好与正向代理相反，对于客户端而言代理服务器就像是原始服务器，并且客户端不需要进行任何特别的设置。客户端向反向代理的命名空间(name-space)中的内容发送普通请求，接着反向代理将判断向何处(原始服务器)转交请求，并将获得的内容返回给客户端。

·1 什么是 Nginx

Nginx ("engine x") 是一个高性能的 HTTP 和反向代理服务器，也是一个 IMAP/POP3/SMTP 服务器。

正向代理：





很多大网站都是使用nginx做反向代理，应用非常广泛。

Nginx是一款高性能的http 服务器/反向代理服务器及电子邮件（IMAP/POP3）代理服务器。由俄罗斯的程序员设计师Igor Sysoev所开发，官方测试nginx能够支撑**5万并发链接**，并且cpu、内存等资源消耗却非常低，运行非常稳定。

Nginx

1、http服务器，可以做静态网页的http服务器。

2、配置虚拟机。

一个域名可以被多个ip绑定。可以根据域名的不同吧请求转发给运行在不同端口的服务器。

3、反向代理，**负载均衡**。把请求转发给不同的服务器。

6. Nginx+Tomcat的集群搭建

首先解压两个相同的tomcat到不同的目录（用来模拟多个服务器的效果）

修改tomcat2的server.xml，防止与tomcat1端口冲突

修改端口号防止冲突

```

17  <!--
18  <!-- Note: A "Server" is not itself a "Container", so you may not
19  define subcomponents such as "Valves" at this level.
20  Documentation at /docs/config/server.html
21  -->
22  <Server port="8006" shutdown="SHUTDOWN">
23  <!-- Security listener. Documentation at /docs/config/listeners.html
24  <Listener className="org.apache.catalina.security.SecurityListener" />
25  -->
26  <!--APR library loader. Documentation at /docs/apr.html -->
27  <Listener className="org.apache.catalina.core.AprLifecycleListener" SS
28  <!--Initialize Jasper prior to webapps are loaded. Documentation at /c
29  <Listener className="org.apache.catalina.core.JasperListener" />
30  <!-- Prevent memory leaks due to use of particular java/javafx APIs-->
31  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionI
32  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecvc

```

8005改为8006

```

66 java AJP Connector: /docs/config/ajp.html
67 APR (HTTP/AJP) Connector: /docs/apr.html
68 Define a non-SSL HTTP/1.1 Connector on port 8080
69 -->
70 <Connector port="8081" protocol="HTTP/1.1"
71 connectionTimeout="20000"
72 redirectPort="8443" />
73 <!-- A "Connector" using the shared thread pool-->
74 <!--
75 <Connector executor="tomcatThreadPool"
76 port="8080" protocol="HTTP/1.1"

```

```

88 -->
89
90 <!-- Define an AJP 1.3 Connector on port 8009 -->
91 <Connector port="8010" protocol="AJP/1.3" redirectPort="8443"
92
93 8009-->8010
94 <!-- An Engine represents the entry point (within Catalina)
95 every request. The Engine implementation for Tomcat starts
96 analyzes the HTTP headers included with the request, and

```

修改tomcat2的端口后

1 会到库中 共享 刻录 新建文件夹

名称	修改日期	类型	大小
tomcat1	2017/8/21 20:28	文件夹	
tomcat2	2017/8/21 20:29	文件夹	

Tomcat

八月 21, 2017 8:30:44 下午 org.apache.catalina.util.SessionIdGenerator createSecureRandom

信息: Creation of SecureRandom instance for session ID generation using [SHA1PRNG] took [396] milliseconds.

八月 21, 2017 8:30:45 下午 org.apache.catalina.startup.HostConfig deployDirectory

信息: Deploying web application directory E:\tomcat集群\tomcat2\webapps\examples

八月 21, 2017 8:30:45 下午 org.apache.catalina.startup.HostConfig deployDirectory

信息: Deploying web application directory E:\tomcat集群\tomcat2\webapps\host-manager

八月 21, 2017 8:30:45 下午 org.apache.catalina.startup.HostConfig deployDirectory

信息: Deploying web application directory E:\tomcat集群\tomcat2\webapps\manager

八月 21, 2017 8:30:45 下午 org.apache.catalina.startup.HostConfig deployDirectory

信息: Deploying web application directory E:\tomcat集群\tomcat2\webapps\ROOT

八月 21, 2017 8:30:45 下午 org.apache.coyote.AbstractProtocol start

信息: Starting ProtocolHandler ["http-bio-8081"]

八月 21, 2017 8:30:45 下午 org.apache.coyote.AbstractProtocol start

信息: Starting ProtocolHandler ["ajp-bio-8010"]

八月 21, 2017 8:30:45 下午 org.apache.catalina.startup.Catalina start

信息: Server startup in 1041 ms

Tomcat

八月 21, 2017 8:30:34 下午 org.apache.catalina.util.SessionIdGenerator createSecureRandom

信息: Creation of SecureRandom instance for session ID generation using [SHA1PRNG] took [368] milliseconds.

八月 21, 2017 8:30:34 下午 org.apache.catalina.startup.HostConfig deployDirectory

信息: Deploying web application directory E:\tomcat集群\tomcat2\webapps\examples

八月 21, 2017 8:30:34 下午 org.apache.catalina.startup.HostConfig deployDirectory

信息: Deploying web application directory E:\tomcat集群\tomcat2\webapps\host-manager

八月 21, 2017 8:30:34 下午 org.apache.catalina.startup.HostConfig deployDirectory

信息: Deploying web application directory E:\tomcat集群\tomcat2\webapps\manager

八月 21, 2017 8:30:34 下午 org.apache.catalina.startup.HostConfig deployDirectory

信息: Deploying web application directory E:\tomcat集群\tomcat2\webapps\ROOT

八月 21, 2017 8:30:34 下午 org.apache.coyote.AbstractProtocol start

信息: Starting ProtocolHandler ["http-bio-8081"]

八月 21, 2017 8:30:34 下午 org.apache.coyote.AbstractProtocol start

信息: Starting ProtocolHandler ["ajp-bio-8010"]

八月 21, 2017 8:30:34 下午 org.apache.catalina.startup.Catalina start

信息: Server startup in 920 ms

同时启动两个tomcat就没有端口冲突问题了

写一个简单的web项目用于测试

这个项目只有一个jsp页面

test821

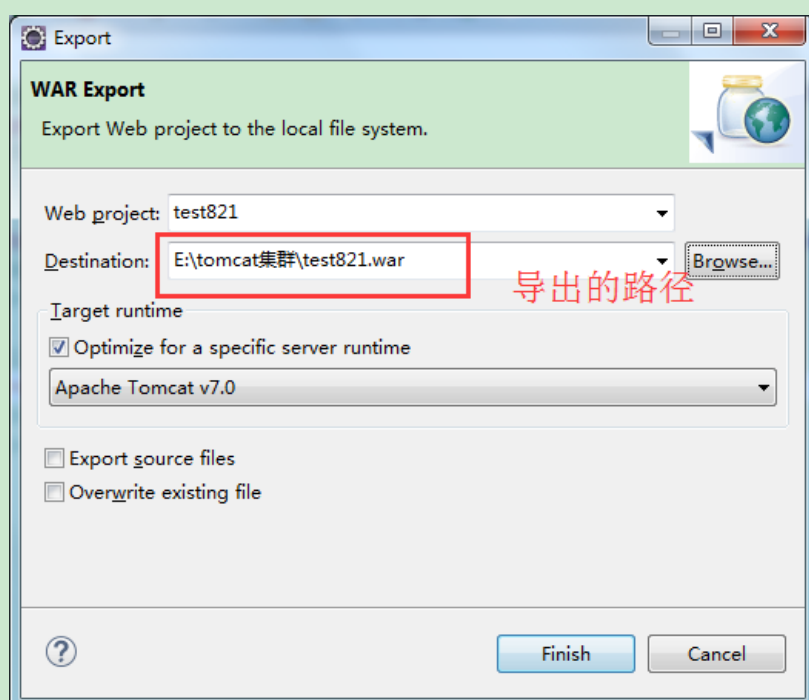
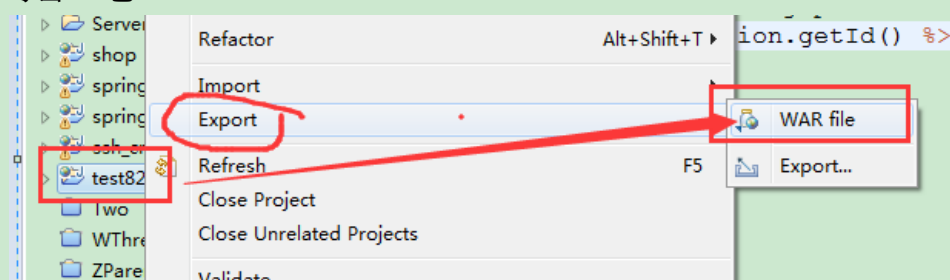
- JAX-WS Web Services
- Deployment Descriptor: test821
- Java Resources
- JavaScript Resources
- build
- WebContent
 - META-INF
 - WEB-INF
 - index.jsp
- Two

```

index.jsp
1 <%@ page language="java" contentType="text/html; cha
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transit
4 <html>
5 <head>
6 <meta http-equiv="Content-Type" content="text/html;
7 <title>Insert title here</title>
8 </head>
9 <body>
10   <h1>index.jsp</h1>
11   <%=session.getId() %>
12 </body>
13 </html>

```

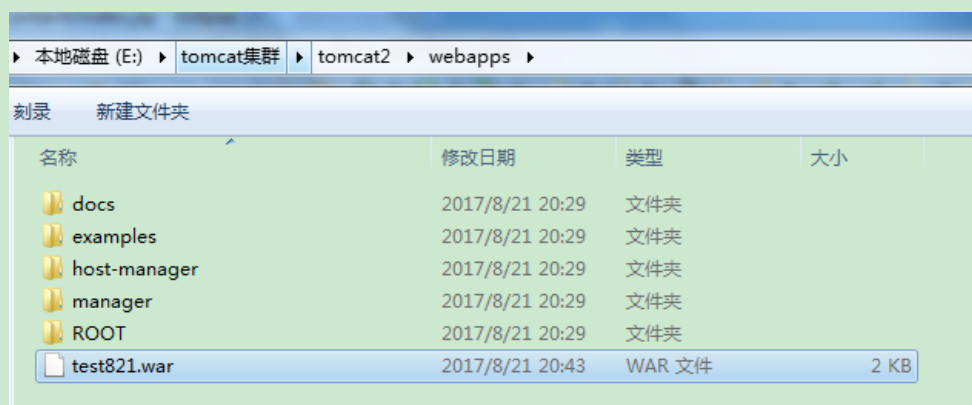
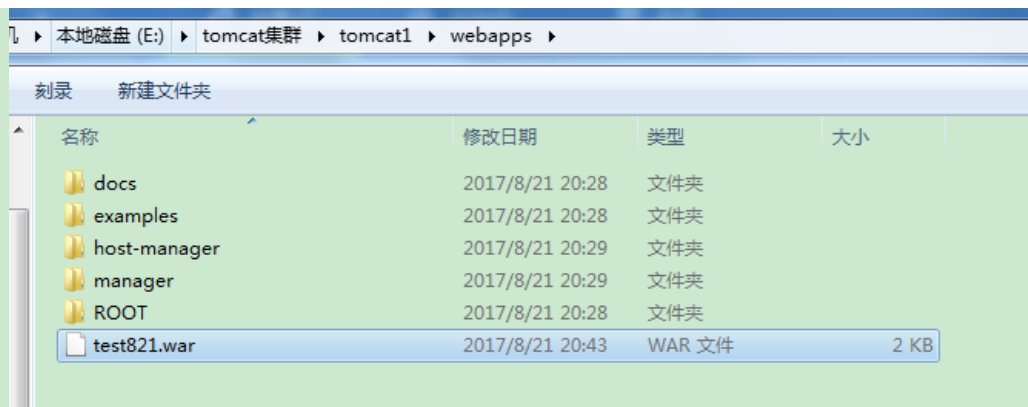
导出war包



计算机 > 本地磁盘 (E:) > tomcat集群 >

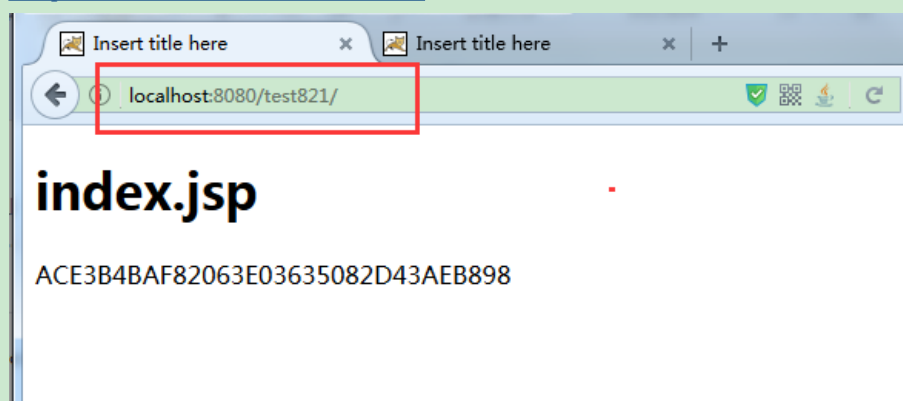
名称	修改日期	类型	大小
tomcat1	2017/8/21 20:28	文件夹	
tomcat2	2017/8/21 20:29	文件夹	
test821.war	2017/8/21 20:43	WAR 文件	2 KB

把这个war包放到两个tomcat的webapps目录下

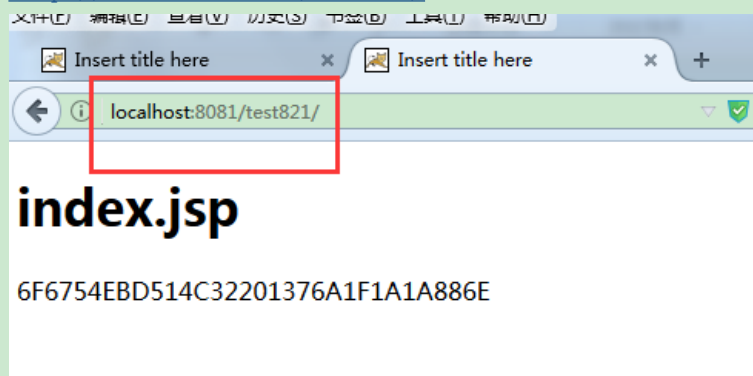


打开浏览器，访问两个tomcat

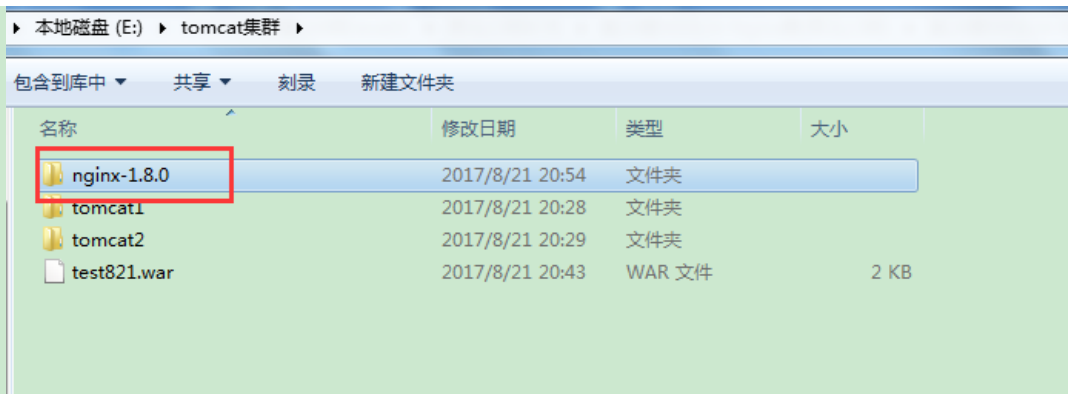
<http://localhost:8080/test821/>



<http://localhost:8081/test821/>

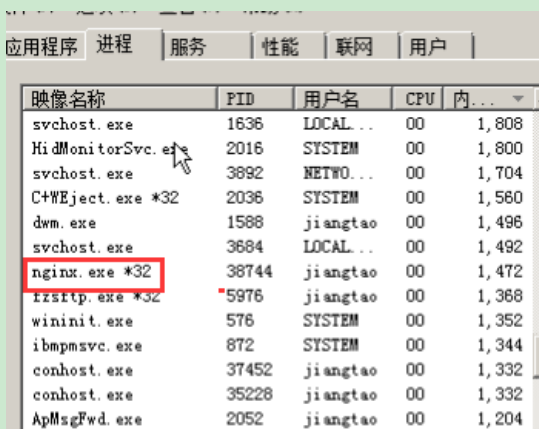


可以实现两个地址（两个tomcat）访问同一个项目，但是用户访问的时候是不会知道修改端口号去访问的，那么必须要有统一的入口让用户访问，那么这个入口就是Nginx
nginx--解压到任意目录就行



那么怎么发现它启动了？

任务管理器



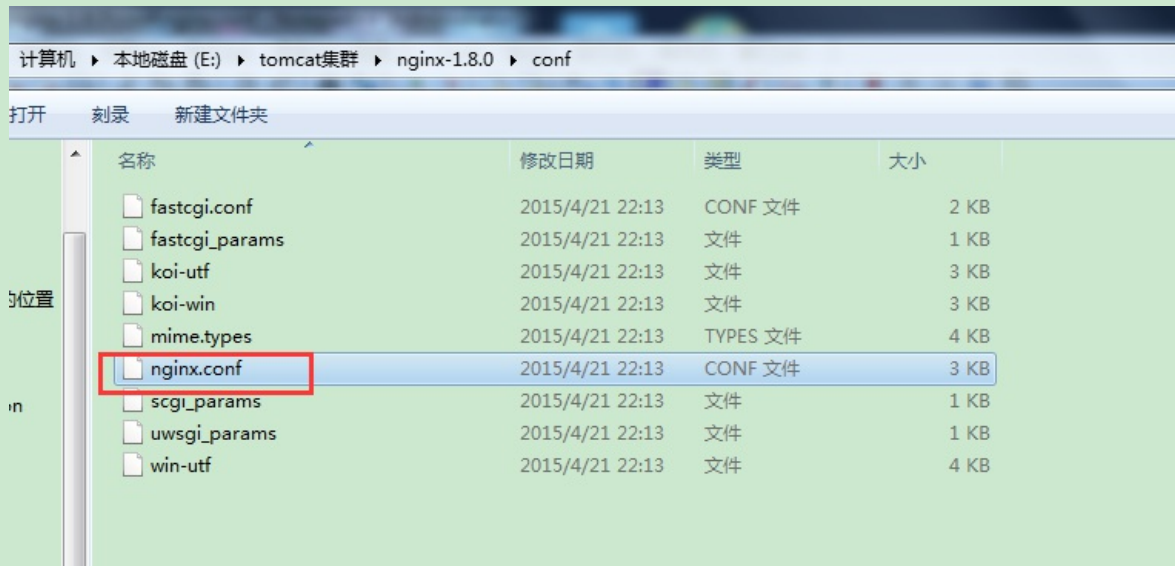
如果启动不成功，有可能是80端口被占用

在windows下执行如下命令

netstat -aon|findstr ":80"

```
C:\Users\阿滔>netstat -aon|findstr ":80"
TCP 0.0.0.0:8009 0.0.0.0:0 LISTENING 20648
TCP 0.0.0.0:8010 0.0.0.0:0 LISTENING 19668
TCP 0.0.0.0:8080 0.0.0.0:0 LISTENING 20648
TCP 0.0.0.0:8081 0.0.0.0:0 LISTENING 19668
TCP 127.0.0.1:8005 0.0.0.0:0 LISTENING 20648
TCP 127.0.0.1:8006 0.0.0.0:0 LISTENING 19668
TCP 172.29.111.34:1317 218.30.116.11:80 ESTABLISHED 5372
TCP 172.29.111.34:1330 101.199.97.112:80 ESTABLISHED 5372
TCP 172.29.111.34:1445 1.192.193.42:80 ESTABLISHED 5372
TCP 172.29.111.34:9508 123.58.182.253:80 ESTABLISHED 7736
TCP 172.29.111.34:10416 59.111.179.137:80 CLOSE_WAIT 7736
TCP 172.29.111.34:10534 43.226.6.79:80 CLOSE_WAIT 11548
TCP 172.29.111.34:11045 101.201.170.241:80 ESTABLISHED 21500
TCP 172.29.111.34:11140 219.133.60.213:80 CLOSE_WAIT 7532
TCP 172.29.111.34:11145 221.228.204.185:80 TIME_WAIT 0
TCP 172.29.111.34:11146 220.181.150.242:80 TIME_WAIT 0
```


发现不少进程在占用80端口，这个时候只能修改默认端口了（但是一般不会有事）



修改配置文件

在nginx.conf

```
32
33     #gzip on;
34
35     server {                                默认是80端口
36         listen      80;
37         server_name localhost;
38
39         #charset koi8-r;
40
41         #access_log logs/host.access.log main;
42
43         location / {
```

关闭80端口后，再运行我们的nginx服务。

另外，我们需要知道的nginx三个主要的命令：

启动服务：start nginx

停止服务：nginx -s stop

重新加载：nginx -s reload (配置文件被修改后需要执行它)

执行命令 `nginx -s reload`

还有另外一个报错

```
管理员: C:\Windows\system32\cmd.exe

E:\tomcat集群\nginx-1.8.0>nginx -s stop
nginx: [emerg] CreateFile() "E:\tomcat集群\nginx-1.8.0/conf/nginx.conf" failed (1113: No mapping for the Unicode character exists in the target multi-byte code page)

E:\tomcat集群\nginx-1.8.0>nginx -s reload
nginx: [emerg] CreateFile() "E:\tomcat集群\nginx-1.8.0/conf/nginx.conf" failed (1113: No mapping for the Unicode character exists in the target multi-byte code page)

E:\tomcat集群\nginx-1.8.0>start nginx

E:\tomcat集群\nginx-1.8.0>nginx -s stop
nginx: [emerg] CreateFile() "E:\tomcat集群\nginx-1.8.0/conf/nginx.conf" failed (1113: No mapping for the Unicode character exists in the target multi-byte code page)

E:\tomcat集群\nginx-1.8.0>
```

nginx路径包含中文的话也是会报错的

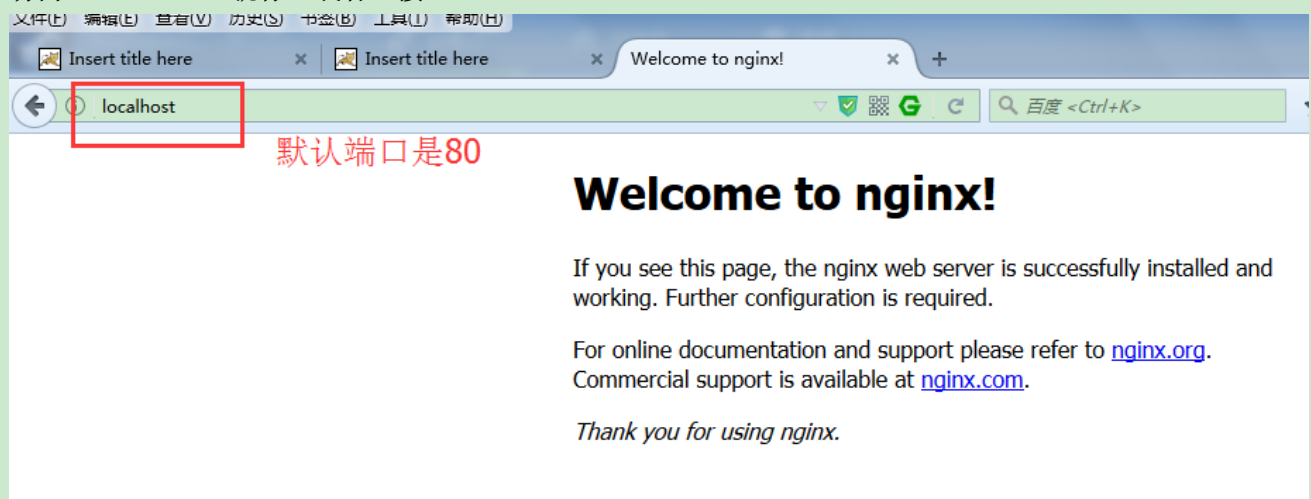
分类： nginx (1)

版权声明：本文为博主原创文章，未经博主允许不得转载。

使用windows版本的nginx启动时遇到(1113: No mapping for the Unicode character exists in the target multi-byte code page)这个错误

后来查阅资料发现是因为解压的路径里面包含有中文的缘故，只要把解压后的文件剪切到没有包含中文的目录即可解决问题

访问localhost:80就行（或者直接localhost）



设置负载均衡

一个地址访问两台服务器

Nginx 的负载均衡的配置:

打开 C:\nginx-1.8.0\conf\nginx.conf 这个文件:

```
upstream server_lb{
    server 127.0.0.1:8080;
    server 127.0.0.1:8081;
}
server {
    listen      80;
    server_name www.taoge.com;

    location / {
        root    html;
        proxy_pass http://server_lb;
        index   index.jsp index.html index.htm;
    }
}
```

服务器列表

hosts文件中配置

反向代理

圈中的是nginx.conf配置文件中需要添加的内容

```
32
33     #gzip on;
34
35     upstream server_lb{
36         server localhost:8080;
37         server localhost:8081;
38     }
39
40     server {
41         listen      80;
42         server_name localhost;
43
44         #charset koi8-r;
45
46         #access_log logs/host.access.log main;
47
48         location / {
49             root    html;
50             proxy_pass http://server_lb;
51             index   index.html index.htm;
52         }
53
54         #error_page 404              /404.html;
55
```

```
upstream server_lb{
    server localhost:8080;
    server localhost:8081;
}

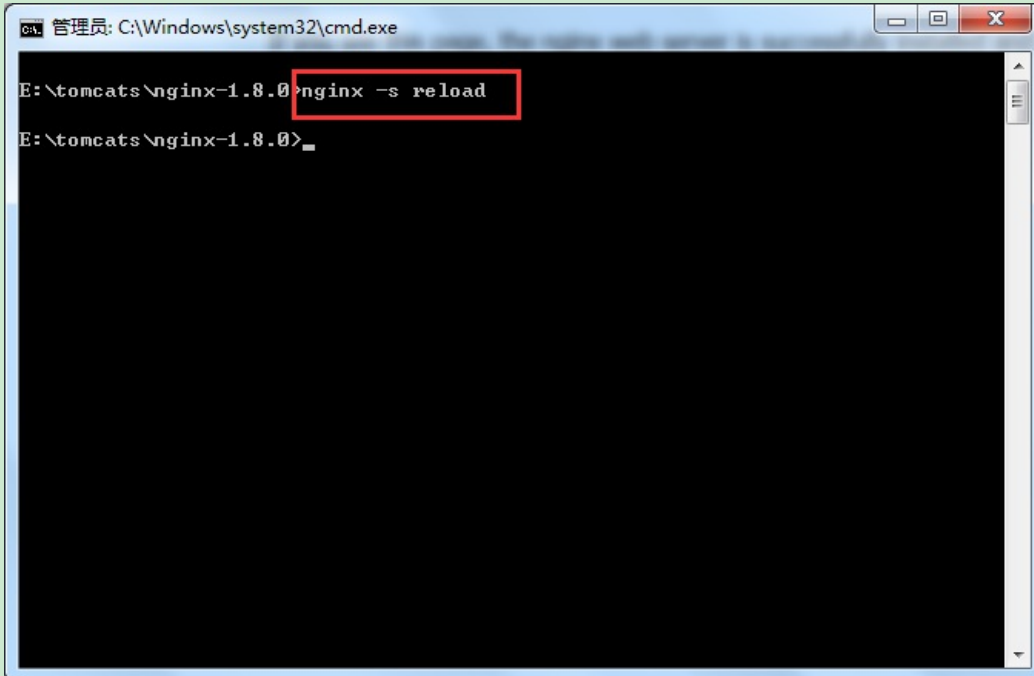
server {
    listen      80;
    server_name localhost;

    #charset koi8-r;

    #access_log logs/host.access.log main;
```

```
location / {  
    root    html;  
    proxy_pass http://server_lb;  
    index  index.html index.htm;  
}
```

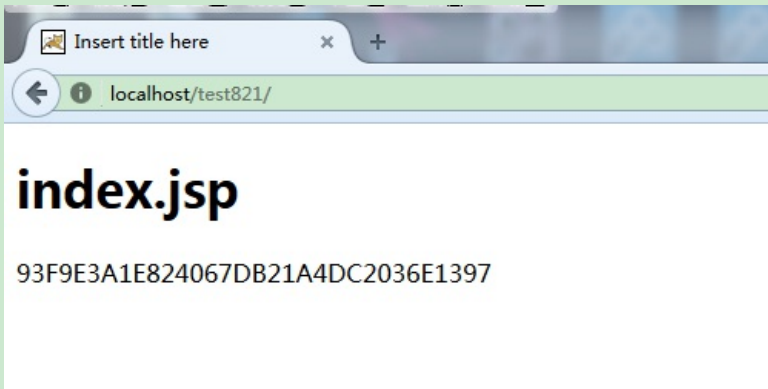
然后用命令加载一下刚修改的配置



然后访问同一个网址

<http://localhost/test821/>

可以看到 一个现象就是，不断刷新这个网址，会交替访问两个服务器的页面



刷新页面两次后



可以发现两次的sessionid是不一样的，是不同页面

然后再刷新，又变不一样

这样其实还不算太明显，可以修改页面让其更清晰

tomcat1

```

5 <head>
6 <meta http-equiv="Content-Type" content="text,
7 <title>Insert title here</title>
8 </head>
9 <body>
10 <h1>tomcat1</h1>
11 <%=session.getId() %>
12 </body>
13 </html>

```

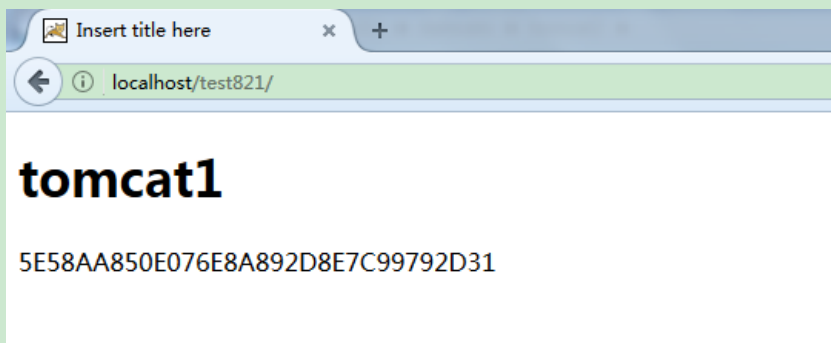
另一个tomcat

```

1 <title>Insert title here</title>
2 </head>
3 <body>
4 <h1>tomcat2</h1>
5 <%=session.getId() %>
6 </body>
7 </html>

```

再次访问



刷新两次



可以看到tomcat有切换，通过同一个端口访问了两个tomcat服务器

session共享问题

nginx.conf

```

32
33 #gzip on;
34
35 upstream server_lb{
36     server localhost:8080;
37     server localhost:8081;
38     ip_hash;
39 }
40
41 server {
42     listen      80;
43     server_name localhost;
44
45     location / {
46         root    html;
47         proxy_pass http://server_lb;
48         index  index.html index.htm;
49     }

```

加上这个后：一个session（一个用户）只会访问一个服务器(tomcat)，不会跳到其它服务器

```
#gzip on;

upstream server_lb{
    server localhost:8080 weight=4;
    server localhost:8081 weight=10;
    ip_hash;
}

server {
    listen 80;
    server_name localhost;
```

也可以设置不同服务器的权重，就是访问哪个服务器的概率更大

7. Tomcat集群的session共享

1. 一种解决办法：一个用户进来以后只在tomcat1上进行操作，另一个用户进行只在tomcat2上进行操作。

2. session的共享

一种使用tomcat广播机制完成session的共享（不推荐的方式）

一种使用redis服务器的方式完成session的共享（推荐的方式）

**** 通过以上的配置我们已经可以通过访问 www.taoge.com 访问到不同的 tomcat 来分担服务器端的压力。请求负载均衡过程中会话信息不能丢失，那么需要在多个 tomcat 中 session 需要共享。

* 配置 Tomcat 的 session 共享可以有三种解决方案：

第一种是以负载均衡服务器本身提供的 session 共享策略，每种服务期的配置是不一样的并且 nginx 本身是没有的。

第二种是利用 web 容器本身的 session 共享策略来配置共享。针对于 weblogic 这种方式还是靠谱的。但是针对于 tomcat 这种方式存在很大的缺陷，主要因为它是依靠广播方式来实现的 session 复制，会浪费很多带宽导致整个网络反映缓慢。官网也建议这种方式最好不要超过 4 台 tomcat，具体的内容可参考 [/webapps/docs/cluster-howto.html](http://webapps/docs/cluster-howto.html) 里面有详细的说明。下面是具体的配置过程。

第三种是 Tomcat 集群+redis 的 Session 共享配置方法。

在这里我们以第二种方式为例：

实际开发使用第三种方式但是现在演示第二种方式

```
104 -->
105 <Engine name="Catalina" defaultHost="localhost">
106
107 <!--For clustering, please take a look at documentation at:
108      /docs/cluster-howto.html (simple how to)
109      /docs/config/cluster.html (reference documentation) -->
110
111 <Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>
112
113
114 <!-- Use the LockOutRealm to prevent attempts to guess user passwords
115      via a brute-force attack -->
116 <Realm className="org.apache.catalina.realm.LockOutRealm">
117 <!-- This Realm uses the UserDatabase configured in the global JNDI
118      resources under the key "UserDatabase". Any edits
119      that are performed against this UserDatabase are immediately
120      available for use by the Realm -->
121
122 </Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>
```

```

<!-- You should set jvmRoute to support load balancing via AJP ie :
<Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1">
-->
<Engine name="Catalina" defaultHost="localhost" jvmRoute="tomcat1">

<!--For clustering, please take a look at documentation at:
/docs/cluster-howto.html (simple how to)
/docs/config/cluster.html (reference documentation) -->

<Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>

<!-- Use the LockOutRealm to prevent attempts to guess user passwords
via a brute-force attack -->

```

可选:加上后
session打印的信
息会添加这内容

```

101
102 <!-- You should set jvmRoute to support load-balancing via AJP ie :
103 <Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1">
104 -->
105 <Engine name="Catalina" defaultHost="localhost" jvmRoute="tomcat2">
106
107 <!--For clustering, please take a look at documentation at:
108 /docs/cluster-howto.html (simple how to)
109 /docs/config/cluster.html (reference documentation) -->
110
111 <Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>
112
113
114 <!-- Use the LockOutRealm to prevent attempts to guess user passwords
115 via a brute-force attack -->
116 <Realm className="org.apache.catalina.realm.LockOutRealm">
117 <!-- This Realm uses the UserDatabase configured in the global JNDI
118 resources under the key "UserDatabase". Any edits
119 that are performed against this UserDatabase are immediately

```

两个tomcat测试项目中的web.xml也需要添加内容

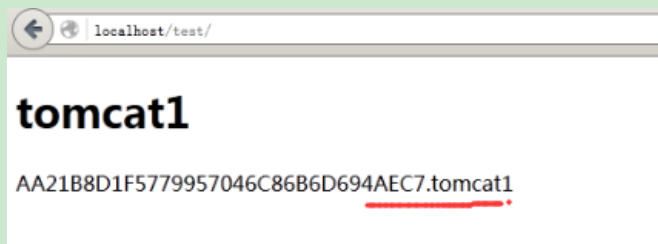
<distributable/>

```

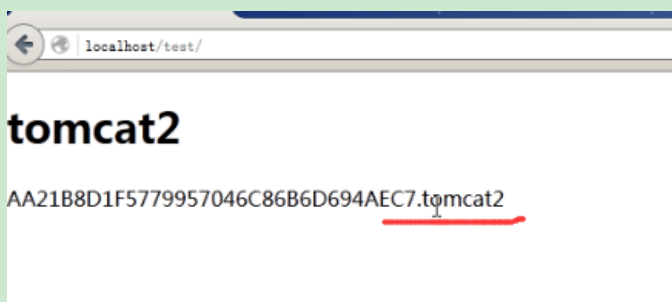
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
<display-name>test821</display-name>
<distributable/>
<welcome-file-list>
<welcome-file>index.html</welcome-file>
<welcome-file>index.htm</welcome-file>
<welcome-file>index.jsp</welcome-file>
<welcome-file>default.html</welcome-file>
<welcome-file>default.htm</welcome-file>
<welcome-file>default.jsp</welcome-file>
</welcome-file-list>
</web-app>

```

访问



刷新两次



session相同, 不会因为tomcat改变而变

=====

8. Linux上安装多个tomcat

在usr/local下新建tomcats文件夹

```
密码:
[root@centos001 local]# ll
总用量 48
drwxr-xr-x. 2 root root 4096 9月 23 2011 bin
drwxr-xr-x. 2 root root 4096 9月 23 2011 etc
drwxr-xr-x. 2 root root 4096 9月 23 2011 games
drwxr-xr-x. 2 root root 4096 9月 23 2011 include
drwxr-xr-x. 5 root root 4096 8月 21 14:51 java
drwxr-xr-x. 2 root root 4096 9月 23 2011 lib
drwxr-xr-x. 2 root root 4096 9月 23 2011 libexec
drwxr-xr-x. 2 root root 4096 8月 21 11:10 mysql
drwxr-xr-x. 2 root root 4096 9月 23 2011 sbin
drwxr-xr-x. 5 root root 4096 4月 10 04:53 share
drwxr-xr-x. 2 root root 4096 9月 23 2011 src
drwxr-xr-x. 2 root root 4096 8月 22 09:34 tomcats
[root@centos001 local]# cd tomcats/
```

两个tomcat目录

```
总用量 0
[root@centos001 tomcats]# mkdir tomcat1
[root@centos001 tomcats]# mkdir tomcat2
[root@centos001 tomcats]# ll
总用量 8
drwxr-xr-x. 2 root root 4096 8月 22 09:38 tomcat1
drwxr-xr-x. 2 root root 4096 8月 22 09:38 tomcat2
[root@centos001 tomcats]# pwd
/usr/local/tomcats
[root@centos001 tomcats]#
```

解压tomcat到两个目录

```
tar -xvf apache-tomcat-7.0.81.tar.gz -C /usr/local/tomcats/tomcat1/
```

```
tar -xvf apache-tomcat-7.0.81.tar.gz -C /usr/local/tomcats/tomcat2/
```

然后修改tomcat2的配置文件

修改和windows下差不多，就是修改3个端口号防止冲突

然后把windows下的测试war包传到Linux

```
[root@centos001 ~]# ll
总用量 12700
-rw-----. 1 root root 1641 4月 10 05:12 anaconda-ks.cfg
-rw-r--r--. 1 root root 47446 8月 20 14:58 dabaowenjian.tar.gz
-rw-r--r--. 1 root root 45503 4月 10 05:12 install.log
-rw-r--r--. 1 root root 10033 4月 10 05:10 install.log.syslog
-rwxr-xr-x. 1 root root 1625 8月 6 13:32 javapwd.txt
-rw-r--r--. 1 root root 2310 8月 7 13:25 passwd
drwxr-xr-x. 2 root root 4096 8月 20 15:11 pck001
-rw-r--r--. 1 root root 47430 8月 20 15:12 pzk001.tar.gz
-rw-r--r--. 1 root root 12780025 8月 21 15:45 springmvc mybatis.war
-rw-r--r--. 1 root root 1422 8月 22 09:54 test821.war
drwxr-xr-x. 3 root root 4096 8月 21 10:32 workspace
drwxr-xr-x. 2 root root 4096 4月 9 21:21 公共的
drwxr-xr-x. 2 root root 4096 4月 9 21:21 模板
drwxr-xr-x. 2 root root 4096 4月 9 21:21 视频
drwxr-xr-x. 2 root root 4096 4月 9 21:21 图片
drwxr-xr-x. 2 root root 4096 4月 9 21:21 文档
drwxr-xr-x. 2 root root 4096 4月 9 21:21 下载
drwxr-xr-x. 2 root root 4096 4月 9 21:21 音乐
drwxr-xr-x. 2 root root 4096 8月 21 10:22 桌面
[root@centos001 ~]#
```

复制项目到

```
cp test821.war /usr/local/tomcats/tomcat2/apache-tomcat-7.0.81/webapps
```

```
cp test821.war /usr/local/tomcats/tomcat1/apache-tomcat-7.0.81/webapps
```



```
drwxr-xr-x. 2 root root 4096 8月 21 10:22 桌面
[root@centos001 ~]# cp test821.war /usr/local/tomcats/tomcat2/apache-tomcat-7.0.81/webapps
[root@centos001 ~]# cp test821.war /usr/local/tomcats/tomcat1/apache-tomcat-7.0.81/webapps
[root@centos001 ~]#
```

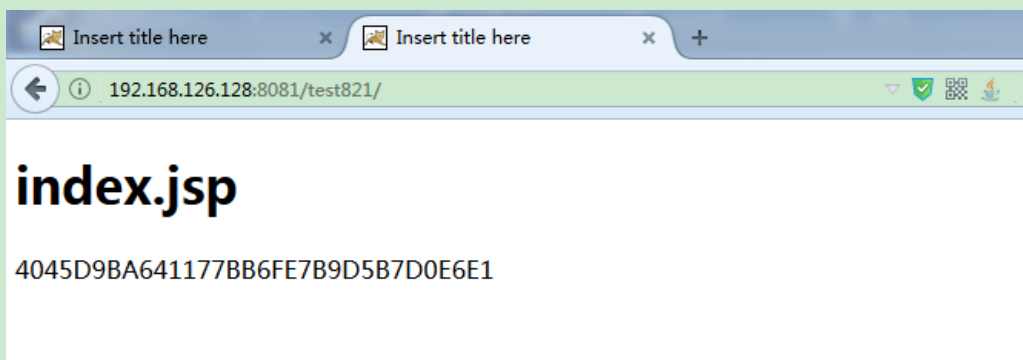
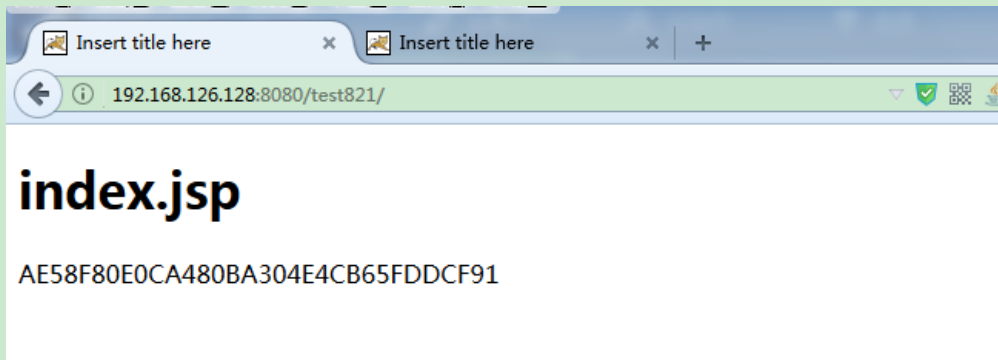
Ready

ssh2: AES-256-CTR

31, 21 31 Ro

```
[root@centos001 apache-tomcat-7.0.81]# cd webapps/
[root@centos001 webapps]# ll
总用量 24
drwxr-xr-x. 14 root root 4096 8月 22 09:43 docs
drwxr-xr-x. 7 root root 4096 8月 22 09:43 examples
drwxr-xr-x. 5 root root 4096 8月 22 09:43 host-manager
drwxr-xr-x. 5 root root 4096 8月 22 09:43 manager
drwxr-xr-x. 3 root root 4096 8月 22 09:43 ROOT
-rw-r--r--. 1 root root 1422 8月 22 09:58 test821.war
[root@centos001 webapps]#
```

启动两个 tomcat



那如果启动不成功，怎么看报错信息呢？

查看日志

Tomcat目录下有一个logs文件夹

```
[root@centos001 bin]# cd ../
[root@centos001 apache-tomcat-7.0.81]# ll
总用量 116
drwxr-xr-x. 2 root root 4096 8月 22 09:43 bin
drwxr-xr-x. 3 root root 4096 8月 22 10:03 conf
drwxr-xr-x. 2 root root 4096 8月 22 09:43 lib
-rw-r--r--. 1 root root 56846 8月 11 18:24 LICENSE
drwxr-xr-x. 2 root root 4096 8月 22 10:03 logs
-rw-r--r--. 1 root root 1239 8月 11 18:24 NOTICE
-rw-r--r--. 1 root root 8965 8月 11 18:24 RELEASE-NOTES
-rw-r--r--. 1 root root 16195 8月 11 18:24 RUNNING.txt
drwxr-xr-x. 2 root root 4096 8月 22 09:43 temp
drwxr-xr-x. 8 root root 4096 8月 22 10:03 webapps
drwxr-xr-x. 3 root root 4096 8月 22 10:03 work
[root@centos001 apache-tomcat-7.0.81]#
```

```
drwxr-xr-x. 5 root root 4096 8月 22 10:03 work
[root@centos001 apache-tomcat-7.0.81]# cd logs/
[root@centos001 logs]# ll
总用量 24
-rw-r--r--. 1 root root 6595 8月 22 10:03 catalina.2017-08-22.log
-rw-r--r--. 1 root root 6595 8月 22 10:03 catalina.out
-rw-r--r--. 1 root root 0 8月 22 10:03 host-manager.2017-08-22.log
-rw-r--r--. 1 root root 476 8月 22 10:03 localhost.2017-08-22.log
-rw-r--r--. 1 root root 994 8月 22 10:05 localhost_access_log.2017-08-22.txt
-rw-r--r--. 1 root root 0 8月 22 10:03 manager.2017-08-22.log
[root@centos001 logs]#
```

选最新的log查看

Ready

ssh2: AES-25

cat catalina.2017-08-22.log

```
[root@centos001 logs]# cat catalina.2017-08-22.log
八月 22, 2017 10:03:45 上午 org.apache.catalina.startup.VersionLoggerListener log
信息: Server version: Apache Tomcat/7.0.81
八月 22, 2017 10:03:45 上午 org.apache.catalina.startup.VersionLoggerListener log
信息: Server built: Aug 11 2017 10:21:27 UTC
八月 22, 2017 10:03:45 上午 org.apache.catalina.startup.VersionLoggerListener log
信息: Server number: 7.0.81.0
八月 22, 2017 10:03:45 上午 org.apache.catalina.startup.VersionLoggerListener log
信息: OS Name: Linux
八月 22, 2017 10:03:45 上午 org.apache.catalina.startup.VersionLoggerListener log
信息: OS Version: 2.6.32-431.el6.i686
八月 22, 2017 10:03:45 上午 org.apache.catalina.startup.VersionLoggerListener log
信息: Architecture: i386
八月 22, 2017 10:03:45 上午 org.apache.catalina.startup.VersionLoggerListener log
信息: Java Home: /usr/local/java/jdk1.7.0_65/jre
八月 22, 2017 10:03:45 上午 org.apache.catalina.startup.VersionLoggerListener log
信息: JVM Version: 1.7.0_65-b17
八月 22, 2017 10:03:45 上午 org.apache.catalina.startup.VersionLoggerListener log
```

```
信息: Deployment of web application directory /usr/local/tomcats/tomcat2/apache-tomcat-
ished in 81 ms
八月 22, 2017 10:03:46 上午 org.apache.catalina.startup.HostConfig deployDirectory
信息: Deploying web application directory /usr/local/tomcats/tomcat2/apache-tomcat-7.0.
八月 22, 2017 10:03:46 上午 org.apache.catalina.startup.HostConfig deployDirectory
信息: Deployment of web application directory /usr/local/tomcats/tomcat2/apache-tomcat-
d in 205 ms
八月 22, 2017 10:03:46 上午 org.apache.coyote.AbstractProtocol start
信息: Starting ProtocolHandler ["http-bio-8081"]
八月 22, 2017 10:03:46 上午 org.apache.coyote.AbstractProtocol start
信息: Starting ProtocolHandler ["ajp-bio-8010"]
八月 22, 2017 10:03:46 上午 org.apache.catalina.startup.Catalina start
信息: Server startup in 933 ms
root@centos001 logs]#
```

如果访问不了，有可能是防火墙拦截

1.3.1.4 设置防火墙:

```
/sbin/iptables -I INPUT -p tcp --dport 8080 -j ACCEPT
/etc/rc.d/init.d/iptables save
```

9. Linux下安装nginx

Linux下安装nginx比较麻烦一点

(1)环境要求

nginx是C语言开发，建议在linux上运行，本教程使用Centos6.4作为安装环境。

gcc

安装nginx需要先将官网下载的源码进行编译，编译依赖gcc环境，如果没有gcc环境，需要安装gcc: yum

install gcc-c++

```

Verifying : libstdc++-devel-4.4.7-18.el6.i686
Verifying : gcc-c++-4.4.7-18.el6.i686
Verifying : libstdc++-4.4.7-4.el6.i686
Verifying : gcc-c++-4.4.7-4.el6.i686
Verifying : gcc-4.4.7-4.el6.i686
Verifying : libgcc-4.4.7-4.el6.i686
Verifying : libstdc++-devel-4.4.7-4.el6.i686
Verifying : libgomp-4.4.7-4.el6.i686
Verifying : libgfortran-4.4.7-4.el6.i686
Verifying : cpp-4.4.7-4.el6.i686
Verifying : gcc-gfortran-4.4.7-4.el6.i686
Updated:
gcc-c++.i686 0:4.4.7-18.el6
Dependency Updated:
  cpp.i686 0:4.4.7-18.el6          gcc.i686 0:4.4.7-18.el6          gcc-gfortran.i686 0:4.4.7-18.el6
  libgcc.i686 0:4.4.7-18.el6      libgfortran.i686 0:4.4.7-18.el6      libgomp.i686 0:4.4.7-18.el6
  libstdc++.i686 0:4.4.7-18.el6    libstdc++-devel.i686 0:4.4.7-18.el6
Complete!
[root@centos001 ~]#

```

PCRE

PCRE(Perl Compatible Regular Expressions)是一个Perl库，包括 perl 兼容的正则表达式库。nginx的http模块使用pcre来解析正则表达式，所以需要在linux上安装pcre库。

yum install -y pcre pcre-devel

注：pcre-devel是使用pcre开发的一个二次开发库。nginx也需要此库。

```

Transaction Test Succeeded
Running Transaction
  Updating      : pcre-7.8-7.el6.i686
  Installing    : pcre-devel-7.8-7.el6.i686
  Cleanup       : pcre-7.8-6.el6.i686
  Verifying     : pcre-devel-7.8-7.el6.i686
  Verifying     : pcre-7.8-7.el6.i686
  Verifying     : pcre-7.8-6.el6.i686
Installed:
pcre-devel.i686 0:7.8-7.el6
Updated:
pcre.i686 0:7.8-7.el6
Complete!
[root@centos001 ~]#

```

Ready

ssh2: AES-2

zlib

zlib库提供了很多种压缩和解压缩的方式，nginx使用zlib对http包的内容进行gzip，所以需要在linux上安装zlib库。

yum install -y zlib zlib-devel

```

Total download size: 44 k
Installed size: 115 k
Downloading Packages:
zlib-devel-1.2.3-29.el6.i686.rpm
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing    : zlib-devel-1.2.3-29.el6.i686
  Verifying     : zlib-devel-1.2.3-29.el6.i686
Installed:
zlib-devel.i686 0:1.2.3-29.el6
Complete!
[root@centos001 ~]#

```

Ready

openssl

OpenSSL 是一个强大的安全套接字层密码库，囊括主要的密码算法、常用的密钥和证书封装管理功能

及SSL协议，并提供丰富的应用程序供测试或其它目的使用。

nginx不仅支持http协议，还支持https（即在ssl协议上传输http），所以需要在linux安装openssl库。

```
yum install -y openssl openssl-devel
```

```
Installed:
  openssl-devel.i686 0:1.0.1e-57.el6

Dependency Installed:
  keyutils-libs-devel.i686 0:1.4-5.el6      krb5-devel.i686 0:1.10.3-1.el6
  libkadm5.i686 0:1.10.3-65.el6            libselinux-devel.i686 0:2.0.96-8.el6

Updated:
  openssl.i686 0:1.0.1e-57.el6

Dependency Updated:
  e2fsprogs.i686 0:1.41.12-23.el6      e2fsprogs-libs.i686 0:1.41.12-23.el6
  keyutils-libs.i686 0:1.4-5.el6      krb5-libs.i686 0:1.10.3-1.el6
  libcom_err.i686 0:1.41.12-23.el6    libselinux.i686 0:2.0.96-8.el6
  libselinux-utils.i686 0:2.0.94-7.el6  libss.i686 0:1.41.12-23.el6

Complete!
[root@centos001 ~]#
```

从本地上传nginx到Linux

```
Complete!
[root@centos001 ~]# ll
总用量 13516
-rw-r--r--. 1 root root      1641 4月 10 05:12 anaconda-ks.cfg
-rw-r--r--. 1 root root    47446 8月 20 14:58 dabaowenjian.tar.gz
-rw-r--r--. 1 root root    45503 4月 10 05:12 install.log
-rw-r--r--. 1 root root    10033 4月 10 05:10 install.log.syslog
-rwxr-xr-x. 1 root root     1625 8月  6 13:32 javapwd.txt
-rw-r--r--. 1 root root   832104 8月 22 10:39 nginx-1.8.0.tar.gz
-rw-r--r--. 1 root root     2310 8月  7 13:25 passwd
drwxr-xr-x. 2 root root     4096 8月 20 15:11 pck001
-rw-r--r--. 1 root root    47430 8月 20 15:12 pzk001.tar.gz
-rw-r--r--. 1 root root  12780025 8月 21 15:45 springmvc_mybatis.war
-rw-r--r--. 1 root root     1422 8月 22 09:54 test821.war
drwxr-xr-x. 3 root root     4096 8月 21 10:32 workspace
drwxr-xr-x. 2 root root     4096 4月  9 21:21 公共的
drwxr-xr-x. 2 root root     4096 4月  9 21:21 模板
drwxr-xr-x. 2 root root     4096 4月  9 21:21 视频
drwxr-xr-x. 2 root root     4096 4月  9 21:21 图片
drwxr-xr-x. 2 root root     4096 4月  9 21:21 文档
drwxr-xr-x. 2 root root     4096 4月  9 21:21 下载
drwxr-xr-x. 2 root root     4096 4月  9 21:21 音乐
drwxr-xr-x. 2 root root     4096 8月 21 10:22 桌面
[root@centos001 ~]#
```

解压

```
[root@centos001 ~]# cd /usr/local
[root@centos001 local]# mkdir nginx
[root@centos001 local]# tar -xvf nginx-1.8.0.tar.gz -C /usr/local/nginx
```

Ready

ssh2: i

```
drwxr-xr-x. 1 root root 4096 8月 22 09:58 local
[root@centos001 local]# cd
[root@centos001 ~]# tar -xvf nginx-1.8.0.tar.gz -C /usr/local/nginx
```

Ready

```
tar -xvf nginx-1.8.0.tar.gz -C /usr/local/nginx
```

```
nginx-1.8.0/auto/cc/sunc
[root@centos001 ~]# cd /usr/local/nginx/
[root@centos001 nginx]# ll
总用量 4
drwxr-xr-x. 8 1001 1001 4096 4月 21 2015 nginx-1.8.0
[root@centos001 nginx]#
```

ready

需要编译

```
[root@centos001 nginx]# cd nginx-1.8.0/
[root@centos001 nginx-1.8.0]# ll
总用量 652
drwxr-xr-x. 6 1001 1001 4096 8月 22 10:54 auto
-rw-r--r--. 1 1001 1001 249124 4月 21 2015 CHANGES
-rw-r--r--. 1 1001 1001 379021 4月 21 2015 CHANGES.ru
drwxr-xr-x. 2 1001 1001 4096 8月 22 10:54 conf
-rwxr-xr-x. 1 1001 1001 2478 4月 21 2015 configure
drwxr-xr-x. 4 1001 1001 4096 8月 22 10:54 contrib
drwxr-xr-x. 2 1001 1001 4096 8月 22 10:54 html
-rw-r--r--. 1 1001 1001 1397 4月 21 2015 LICENSE
drwxr-xr-x. 2 1001 1001 4096 8月 22 10:54 man
-rw-r--r--. 1 1001 1001 49 4月 21 2015 README
drwxr-xr-x. 8 1001 1001 4096 8月 22 10:54 src
[root@centos001 nginx-1.8.0]#
```

执行configure

```
drwxr-xr-x. 2 1001 1001 4096 8月 22 10:54 conf
-rwxr-xr-x. 1 1001 1001 2478 4月 21 2015 configure
drwxr-xr-x. 4 1001 1001 4096 8月 22 10:54 contrib
drwxr-xr-x. 2 1001 1001 4096 8月 22 10:54 html
-rw-r--r--. 1 1001 1001 1397 4月 21 2015 LICENSE
drwxr-xr-x. 2 1001 1001 4096 8月 22 10:54 man
-rw-r--r--. 1 1001 1001 49 4月 21 2015 README
drwxr-xr-x. 8 1001 1001 4096 8月 22 10:54 src
[root@centos001 nginx-1.8.0]# ./configure
```

Ready

完成

```
Configuration summary
+ using system PCRE library
+ OpenSSL library is not used
+ md5: using system crypto library
+ sha1: using system crypto library
+ using system zlib library

nginx path prefix: "/usr/local/nginx"
nginx binary file: "/usr/local/nginx/sbin/nginx"
nginx configuration prefix: "/usr/local/nginx/conf"
nginx configuration file: "/usr/local/nginx/conf/nginx.conf"
nginx pid file: "/usr/local/nginx/logs/nginx.pid"
nginx error log file: "/usr/local/nginx/logs/error.log"
nginx http access log file: "/usr/local/nginx/logs/access.log"
nginx http client request body temporary files: "client_body_temp"
nginx http proxy temporary files: "proxy_temp"
nginx http fastcgi temporary files: "fastcgi_temp"
nginx http uwsgi temporary files: "uwsgi_temp"
nginx http scgi temporary files: "scgi_temp"

[root@centos001 nginx-1.8.0]#
```

然后执行make

完成

```

objs/src/http/modules/nginx_http_geo_module.o \
objs/src/http/modules/nginx_http_map_module.o \
objs/src/http/modules/nginx_http_split_clients_module.o \
objs/src/http/modules/nginx_http_referer_module.o \
objs/src/http/modules/nginx_http_rewrite_module.o \
objs/src/http/modules/nginx_http_proxy_module.o \
objs/src/http/modules/nginx_http_fastcgi_module.o \
objs/src/http/modules/nginx_http_uwsgi_module.o \
objs/src/http/modules/nginx_http_scgi_module.o \
objs/src/http/modules/nginx_http_memcached_module.o \
objs/src/http/modules/nginx_http_empty_gif_module.o \
objs/src/http/modules/nginx_http_browser_module.o \
objs/src/http/modules/nginx_http_upstream_hash_module.o \
objs/src/http/modules/nginx_http_upstream_ip_hash_module.o \
objs/src/http/modules/nginx_http_upstream_least_conn_module.o \
objs/src/http/modules/nginx_http_upstream_keepalive_module.o \
objs/nginx_modules.o \
-lpthread -lcrypt -lpcre -lcrypto -lcrypto -lz
make[1]: Leaving directory `/usr/local/nginx/nginx-1.8.0'
make -f objs/Makefile manpage
make[1]: Entering directory `/usr/local/nginx/nginx-1.8.0'
sed -e "s|%%PREFIX%%|/usr/local/nginx|" \
    -e "s|%%PID_PATH%%|/usr/local/nginx/logs/nginx.pid|" \
    -e "s|%%CONF_PATH%%|/usr/local/nginx/conf/nginx.conf|" \
    -e "s|%%ERROR_LOG_PATH%%|/usr/local/nginx/logs/error.log|" \
    < man/nginx.8 > objs/nginx.8
make[1]: Leaving directory `/usr/local/nginx/nginx-1.8.0'
[root@centos001 nginx-1.8.0]#

```

完成后 makeinstall

```

test -f /usr/local/nginx/conf/mime.types || cp conf/mime.types /usr/local/nginx/conf
cp conf/mime.types /usr/local/nginx/conf/mime.types.default
test -f /usr/local/nginx/conf/fastcgi_params || cp conf/fastcgi_params /usr/local/nginx/conf
cp conf/fastcgi_params /usr/local/nginx/conf/fastcgi_params.default
test -f /usr/local/nginx/conf/fastcgi.conf || cp conf/fastcgi.conf /usr/local/nginx/conf
cp conf/fastcgi.conf /usr/local/nginx/conf/fastcgi.conf.default
test -f /usr/local/nginx/conf/uwsgi_params || cp conf/uwsgi_params /usr/local/nginx/conf
cp conf/uwsgi_params /usr/local/nginx/conf/uwsgi_params.default
test -f /usr/local/nginx/conf/scgi_params || cp conf/scgi_params /usr/local/nginx/conf
cp conf/scgi_params /usr/local/nginx/conf/scgi_params.default
test -f /usr/local/nginx/conf/nginx.conf || cp conf/nginx.conf /usr/local/nginx/conf/nginx.conf.default
cp conf/nginx.conf /usr/local/nginx/conf/nginx.conf.default
test -d /usr/local/nginx/logs || mkdir -p /usr/local/nginx/logs
test -d /usr/local/nginx/html || mkdir -p /usr/local/nginx/html
test -d /usr/local/nginx/html || cp -R html /usr/local/nginx
test -d /usr/local/nginx/logs || mkdir -p /usr/local/nginx/logs
make[1]: Leaving directory `/usr/local/nginx/nginx-1.8.0'
[root@centos001 nginx-1.8.0]#

```

Ready

ech2: AFS-256-CTR 31 31 31 Rows: 121 Cols: VT10

默认安装路径是/usr/local/nginx

```

drwxr-xr-x. 2 root root 4096 9月 23 2011 games
drwxr-xr-x. 2 root root 4096 9月 23 2011 include
drwxr-xr-x. 5 root root 4096 8月 21 14:51 java
drwxr-xr-x. 2 root root 4096 9月 23 2011 lib
drwxr-xr-x. 2 root root 4096 9月 23 2011 libexec
drwxr-xr-x. 2 root root 4096 8月 21 11:10 mysql
drwxr-xr-x. 7 root root 4096 8月 22 10:58 nginx
drwxr-xr-x. 2 root root 4096 9月 23 2011 sbin
drwxr-xr-x. 5 root root 4096 4月 10 04:53 share
drwxr-xr-x. 2 root root 4096 9月 23 2011 src
drwxr-xr-x. 4 root root 4096 8月 22 09:38 tomcats
[root@centos001 local]# cd nginx/
[root@centos001 nginx]# ll
总用量 20
drwxr-xr-x. 2 root root 4096 8月 22 10:58 conf
drwxr-xr-x. 2 root root 4096 8月 22 10:58 html
drwxr-xr-x. 2 root root 4096 8月 22 10:58 logs
drwxr-xr-x. 9 1001 1001 4096 8月 22 10:56 nginx-1.8.0
drwxr-xr-x. 2 root root 4096 8月 22 10:58 sbin
[root@centos001 nginx]#

```

新编译的东西

在sbin目录下可以启动nginx

```

bash: ./nginx: 没有那个文件或目录
[root@centos001 nginx]# cd sbin/
[root@centos001 sbin]# ./nginx -> 启动nginx
[root@centos001 sbin]# ps -ef |grep nginx 查看nginx想进程
root      7894      1    0 11:06 ?        00:00:00 nginx: master process ./nginx
nobody    7895    7894    0 11:06 ?        00:00:00 nginx: worker process
root      7897    5579    0 11:07 pts/0    00:00:00 grep nginx
[root@centos001 sbin]#

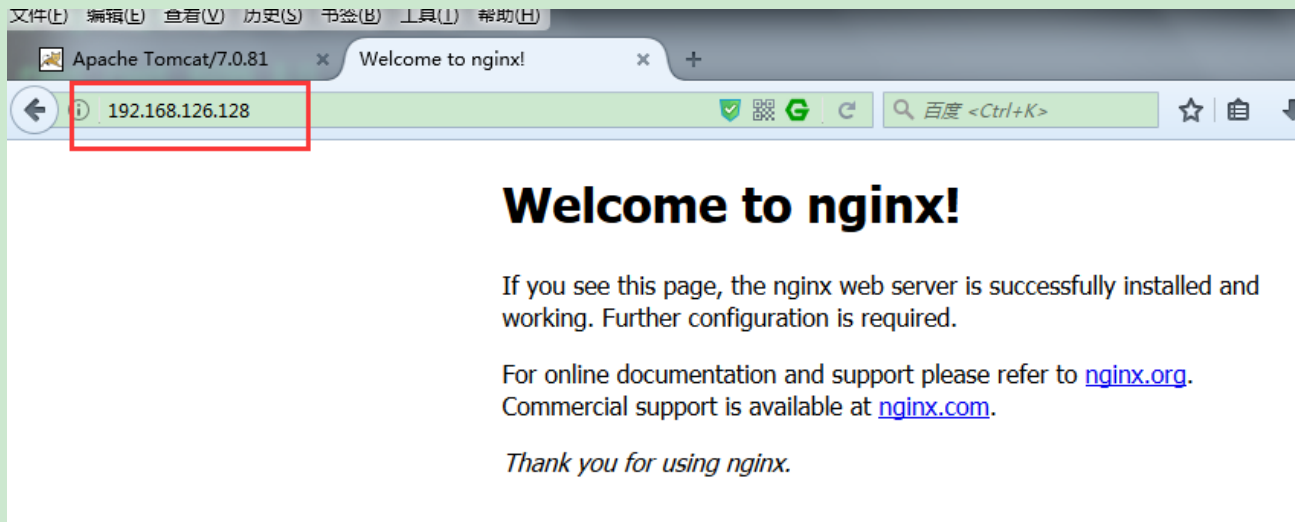
```

ready

ssh2: AES-256-CTR

浏览器访问

<http://192.168.126.128/>



下面进行一些个性化配置

配置负载均衡 nginx.conf

和windows下是一样的

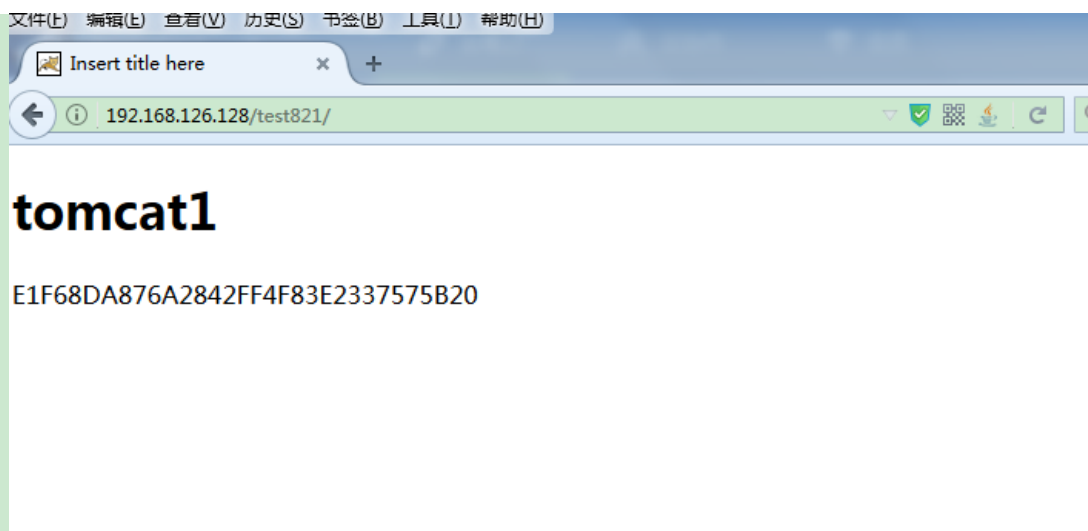
```

rw-r--r--. 1 root root 623 8月 22 10:58 uwsgi_params.default
rw-r--r--. 1 root root 3610 8月 22 10:58 win-utf
root@centos001 conf]# vim nginx.conf
35 upstream server_lb {
36     server 192.168.126.128:8080;
37     server 192.168.126.128:8081;
38 }
39
40 server {
41     listen 80;
42     server_name 192.168.126.128;
43
44     #charset koi8-r;
45
46     #access_log logs/host.access.log main;
47
48     location / {
49         root html;
50         proxy_pass http://server_lb;
51         index index.html index.htm;
52     }
53
54     #error_page 404 /404.html;
55

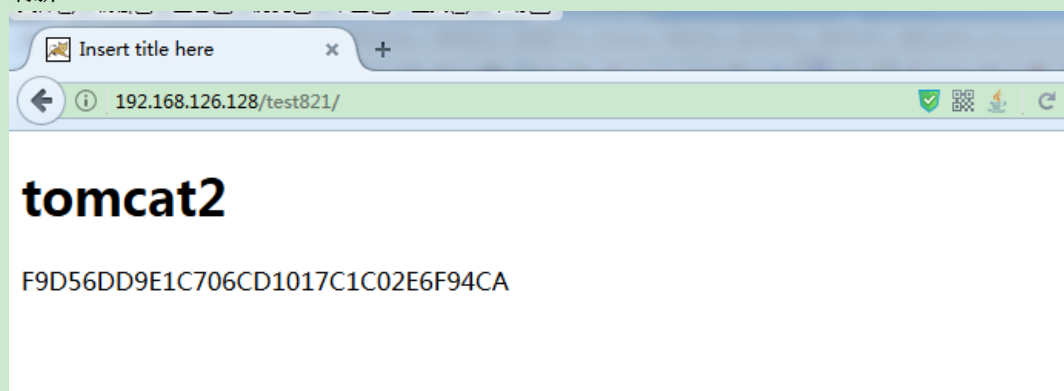
```

注意把localhost改为相应的ip地址

<http://192.168.126.128/test821/>

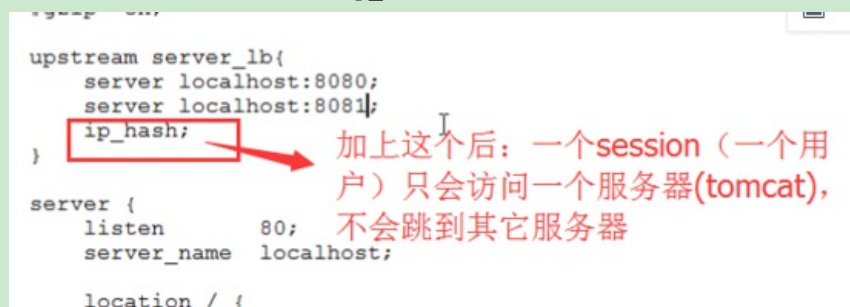


刷新



一个网址，访问两台服务器

session共享问题可以设置 ip_hash 解决



加上这个后：一个session（一个用户）只会访问一个服务器(tomcat)，不会跳到其它服务器