

# Appendix G Deploying SOBA on Linux

If you have not done so, I strongly suggest that you build, deploy and test SOBA on Windows first and make sure that SOBA works on your Windows PC. Then, you can follow the procedure given in this appendix to deploy SOBA on Linux. This procedure includes the following tasks to be performed on a Linux system:

- Installing a JDK
- Installing Tomcat 7.0.29
- Installing MySQL 5.5.22
- Deploying SOBA

The Linux box I tested is a Red Hat Enterprise Linux Server release 6.0 (Santiago) system (this information was obtained with the command `cat /etc/redhat-release`). If you decided to deploy SOBA on Linux, I assume that you know how to get around on a Linux system and you know what necessary changes to make to suit your needs when you follow the procedure below. In addition, my directory on my Linux box for holding JDK, Tomcat and MySQL is `/data1/dev`. I used `root` to perform all tasks.

You need to make sure you have enough disk space for the entire stack of MySQL, Tomcat, JDK and SOBA. If you do not have enough disk space, the install will fail midway. To check your disk space, use the following command:

- `$df`

Then, if the disk space allocated to the user `root` is near 100% used, cd to the `“/”` directory and execute the next command to find out all files larger than 100 MB. This is actually what happened to my system, and I had to delete many files of `*.sqlite` in `root`'s `cache` directory to make room for installing the entire SOBA stack.

- `$find / -type f -size +100000k -exec ls -lh {} \; | awk '{ print $NF ": " $5 }'`

## G.1 INSTALLING A JDK

The JDK I installed is jdk 1.6.0\_33 named *jdk-6u33-linux-x64.bin*. You can use any 1.6.0 build, for example, between u18 and u33.

I first ftp-ed the above downloaded jdk file to the */data1/dev* directory on my Linux system. Then I executed the following commands:

- `$chmod +x jdk-6u33-linux-x64.bin`
- `$/ jdk-6u33-linux-x64.bin`
- `$vi ~/.bash_profile`
- Added the `JAVA_HOME` environment variable to the `PATH` environment variable as follows:

```
JAVA_HOME=/data1/dev/jdk1.6.0_33
PATH$PATH:$HOME/bin:${JAVA_HOME}/bin
```

- `$source ~/.bash_profile`

You can verify with the commands “`echo $JAVA_HOME`” and ““`$JAVA_HOME/bin/java -version`” to make sure JDK is installed properly.

Install Tomcat next.

## G.2 INSTALLING TOMCAT

I first downloaded *apache-tomcat-7.0.29.tar.gz* from Apache’s website, ftp-ed it to the */data1/dev* on my Linux box, and issued the command

- `$tar xvfz apache-tomcat-7.0.29.tar.gz`

to have Tomcat unfolded at */data1/dev/apache-tomcat-7.0.29*. Follow the procedure below to complete Tomcat install:

- Run “`keytool -genkey -alias tomcat -keyalg RSA`” to create a keystore. This step described in Section 2.4.2 for installing Tomcat on Windows applies to Linux as well.
- Carry out those two steps for Tomcat 7 regarding *server.xml* and *tomcat-users.xml* given in Section 2.4.2 here on Linux – they all apply to Linux as well.

To start/stop Tomcat, cd to its *bin* directory and execute

- `$/startup.sh`

or

- `$/shutdown.sh`

accordingly.

You can check out if Tomcat works by trying out <https://<yourLinuxHost>:8443/>. If the page does not return successfully, fix the issue before moving forward.

The next step is to install MySQL.

## G.3 INSTALLING MYSQL

First, I downloaded the MySQL server and client packages *MySQL-server-5.5.22-1.linux2.6.x86\_64.rpm* and *MySQL-client-5.5.22-1.el6.x86\_64.rpm* from

<http://downloads.mysql.com/archives.php?p=mysql-6.0&o=linux->, and ftp-ed them to my */data1/dev* directory on my Linux box. Download the proper packages of MySQL5.5.22 on to your Linux box and get ready for installing both MySQL server and MySQL client.

First, execute the command of

```
■ $ rpm -qa | grep -i mysql
```

to see if you already have MySQL installed. If you find you already have MySQL installed on your Linux box, you can use it or use the command of

```
■ $ yum remove <mysql-prefix>
```

to remove it all, where *<mysql-prefix>* is the part that does not include the numerical version number.

Now, execute the following command to install MySQL server:

```
■ $ rpm -i MySQL-server-5.5.22-1.linux2.6.x86_64.rpm
```

And similarly for the MySQL client

```
■ $ rpm -i MySQL-client-5.5.22-1.el6.x86_64.rpm
```

To start/stop MySQL, execute

```
■ $ /etc/init.d/mysql start
```

or

```
■ $ /etc/init.d/mysql stop
```

accordingly. After you start up your MySQL, execute the following command to change the password for the *root* user:

```
■ $ /usr/bin/mysqladmin -u admin root password 'MySql5522'
```

Note that I entered *'MySql5522'* for my install and you can certainly change it to anything you like. Also, note that *mysqladmin* tool comes only with the MySQL client install.

Then I downloaded *soba\_db\_scripts.zip* from my website <http://www.perfmath.com> and unzipped it to my */data1/dev* directory on my Linux box.

To create the SOBA database, execute the following scripts (you need to enter the *root* password for the first command and *soba31admin's* password of "*soba31admin*" for the second command);

```
■ $ mysql -h 127.0.0.1 -u root -p < create_soba31_mysql.sql
```

```
■ $ mysql soba31 -h 127.0.0.1 -u soba31admin -p < create_all.sql
```

At this point, we need to set the variable *lower\_case\_table\_names = 1* in the *my.cnf* file. The default values for this parameter are 0 on Linux, 1 on Windows, and 2 on Mac OS. "0" means table names case sensitive, "1" means converting table names to lower case to make it case insensitive, and "2" means keeping the original case while making it case insensitive. Follow the steps below:

- Run the command “`find /-name *.cnf`” or “`locate *.cnf`” to find out where your MySQL configuration file `my.cnf` file resides. Mostly you probably don’t see it in your `/etc` directory and you may see many of them in `/usr/share/mysql/` directory. If this is the case, run the command “`cp /usr/share/mysql/my-large.cnf /etc/my.cnf`” to copy it to your `/etc` directory. Note that you may choose a different file than `my-large.cnf` based on the capacity of your system.
- Edit your `my.cnf` file copied to the `/etc` directory by adding a line of `lower_case_table_names = 1` in the section as shown in Figure G.1 (if you add it in other places, it may not work).
- Start up your MySQL server.
- Verify that `lower_case_table_names` is set to 1 by running “`mysqladmin -uroot -p variables.`” This step is necessary, so make sure it’s done properly.

```
# The MySQL server
[mysqld]
port                = 3306
socket              = /var/lib/mysql/mysql.sock
skip-external-locking
key_buffer_size    = 256M
max_allowed_packet = 1M
table_open_cache   = 256
sort_buffer_size   = 1M
read_buffer_size   = 1M
read_rnd_buffer_size = 4M
myisam_sort_buffer_size = 64M
thread_cache_size  = 8
query_cache_size   = 16M
# Try number of CPU's*2 for thread_concurrency
thread_concurrency = 8
lower_case_table_names = 1
```

**Figure G.1** Add `lower_case_table_names` parameter in `my.cnf` file.

Next, deploy SOBA on your Linux box.

## G.4 DEPLOYING SOBA

I used my Maven 3 build of `soba3.2-3.2-mvn` and moved it to my `/data1/dev` directory on my Linux box. Then I changed to the directory of `/data1/dev/soba3.2-3.2-mvn` on my Linux box and issued the command

- `cp -a * /data1/dev/apache-tomcat-7.0.29/webapps/soba`

to copy all files and directories over to Tomcat’s `webapps` directory. In addition, copy `mysql-connector-java-5.1.10.jar` from SOBA’s `WEB-INF/lib` directory to Tomcat’s `lib` directory. Then, start or restart Tomcat and try it out at <http://<yourLinuxHost>:8443/soba>. The above entire procedure works on my Linux system. Now you can test your SOBA deployment using the procedures given in Section 2.7 of this book.