

# BearSoft



## Guía Práctica JDK, TOMCAT, JBOSS6



Av. Abdón Saavedra entre calles Guachalla y Chaco, Pasaje Wigger #2108, Sopocachi  
La Paz, Bolivia  
Teléfono: 591 – 77781631  
Correo electrónico: [rafael.rios@bearsoft.com.bo](mailto:rafael.rios@bearsoft.com.bo)  
Web: <http://bearsoft.com.bo>



INSTALACIÓN DE JBOSS6, TOMCAT, JSDK EN LINUX..... 3  
PASO1: INSTALACIÓN DEL JSDK:..... 3  
PASO2: INSTALACIÓN DEL JBOSS (OPCIONAL):..... 9

## INSTALACIÓN DE JBOSS6, TOMCAT, JSDK EN LINUX

A continuación se detalla una guía paso a paso para la instalación del **JSDK**, el **TOMCAT** y el **JBOSS**, cabe destacar que el único requisito previo necesario para la instalación de estos servidores de aplicaciones, es el **JSDK**.

Ambos servidores pueden coexistir en un mismo ambiente, el **TOMCAT** y el **JBOSS**, no son excluyentes, son independientes y pueden funcionar simultáneamente.

### PASO1: INSTALACIÓN DEL JSDK:

El **JSDK** tiene dos versiones, la **STANDARD EDITION SE** y la **ENTERPRISE EDITION EE**, la **EE** tiene toda una plataforma **JVM**, librerías e incluso un servidor de aplicaciones incorporado, la **SE** tiene la **JVM** y las librerías necesarias para la compilación y ejecución de aplicaciones desarrolladas en **JAVA**.

Para la ejecución de los servidores de aplicaciones que queremos instalar y configurar y para la ejecución de nuestras aplicaciones utilizaremos la versión **SE**. Primero debemos descargar el **JSDK** dependiendo de la necesidad la **SE** o **EE** en cualquiera de las siguientes direcciones:

EE: <http://www.oracle.com/technetwork/java/javase/downloads/java-ee-sdk-6u3-downloads-439814.html>

SE: <http://www.oracle.com/technetwork/java/javase/downloads/jdk-7u4-downloads-1591156.html>

Una vez seleccionemos la versión **SE** que corresponde a la distribución de Linux que tenemos instalada realizamos las siguientes operaciones:

Creamos el directorio **java** dentro de **/usr**

```
[root@servidor ~]# mkdir /usr/java
```

Luego copiamos el archivo descargado que puede tener las siguientes extensiones **.tar.gz** o **rpm** para ambos casos se realiza la copia del directorio donde se haya hecho la descarga al directorio que acabamos de crear.

```
[root@servidor ~]# mv jdk-7u2-linux-x64.tar.gz /usr/java/jdk-7u2-linux-x64.tar.gz
```

```
[root@servidor ~]# mv jdk-7u2-linux-x64.rpm /usr/java/jdk-7u2-linux-x64.rpm
```

Luego de mover los archivos cualquiera que sea la extensión que hayamos seleccionado se debe realizar lo siguiente:

```
[root@servidor ~]# cd /usr/java
```

```
[root@servidor java]# tar -xzf jdk-7u2-linux-x64.tar.gz
```

```
[root@servidor java]# rpm -ivh jdk-7u2-linux-x64.rpm
```

Luego comprobamos que se haya creado el directorio que contiene el **JDK**. Hacemos:

```
[root@servidor java]# ls
```

Nos mostrará lo siguiente:

```
jdk1.7.0_02
```

Ahora se debe crear las variables de entorno, para que pueda ser reconocido en todo el sistema operativo, se puede realizar de dos formas, la primera es modificando el archivo **bashrc** (para algunas distribuciones de Linux) o **bash.bashrc** (en otras distribuciones de Linux) ubicado en **/etc**. Cada que inicia el sistema ejecuta este archivo y asigna estas variables para que sean parte del entorno.

La segunda forma es creando un script del tipo **shell (.sh)** que se ejecute cada vez que inicia el sistema operativo y que debe ser ubicado en **/etc/profile.d**. Este Shell se ejecutará de manera automática y asignará y publicará las variables.

Si realizamos la edición del archivo **bashrc** o **bash.bashrc** se debe introducir las siguientes líneas de código al final del archivo:

```
JAVA_HOME=/usr/java/jdk1.6.0_23
export JAVA_HOME
PATH=$JAVA_HOME/bin:$PATH
export PATH
```

Una vez introducidas las líneas anteriores grabamos el archivo, debemos comprobar que la variable **JAVA\_HOME** esté disponible en el entorno, para esto hacemos:

```
[root@servidor etc]# echo $JAVA_HOME
```

Debería de mostrar nos una respuesta como la siguiente:

```
/usr/java/jdk1.7.0_02
```

Con esto ya se han establecido las variables de entorno.

La otra manera de realizar esta operación es crear un archivo denominado **java.sh**, ese archivo debería de contener lo siguiente:

```
#java
JAVA_HOME=/usr/java/jdk1.7.0_02
export JAVA_HOME
PATH=$JAVA_HOME/bin:$PATH
export PATH
```

Se debe guardar el archivo y copiarlo a **/etc/profile.d** de la siguiente manera:

```
[root@servidor java]# cp java.sh /etc/profile.d
```

Una vez copiado el archivo en el directorio **profile.d** debemos ejecutar el siguiente comando para que el archivo sea ejecutable cuando inicia el sistema.

```
[root@servidor java]# chmod 755 /etc/profile.d/java.sh #sh /etc/profile.d/java.sh
```

Luego verificamos que las variables hayan sido asumidas por el entorno

```
[root@servidor java]# echo $JAVA_HOME
```

Debería de mostrar nos una respuesta como la siguiente:

```
/usr/java/jdk1.7.0_02
```

## PASO2: INSTALACIÓN DEL TOMCAT (OPCIONAL):

Este paso describe la instalación y configuración del **TOMCAT**, primero descargamos el paquete correspondiente a la distribución Linux que estemos usando.

El **TOMCAT**, viene en paquetes del tipo **tar.gz**. La dirección de descarga es:

<http://tomcat.apache.org/download-70.cgi>

Una vez descargado el archivo, hacemos lo siguiente:

```
[root@servidor ~]# mv apache-tomcat-7.0.23.tar.gz /usr/apache-tomcat-7.0.23.tar.gz
```

Luego debemos descomprimir el archivo de la siguiente manera

```
[root@servidor ~]# cd /usr
```

```
[root@servidor usr]# tar -xzf apache-tomcat-7.0.23.tar.gz
```

```
[root@servidor usr]# mv apache-tomcat-7.0.23 tomcat7
```

Con eso hemos cambiado el nombre del directorio que tiene por defecto a un nombre mucho más corto que nos permita operar de manera más cómoda.

Ahora es momento de crear el script que permita ejecutar al **TOMCAT** como servicio de Linux, esto permitirá iniciar el servidor cada que el sistema operativo inicie e instanciará el servicio para poder subir y bajar a demanda, sin necesidad de ejecutar los scripts del directorio **bin** del **TOMCAT**.

Hacemos lo siguiente para crear el servicio:

```
[root@servidor usr]# cd /etc/init.d  
[root@servidor init.d]# vi tomcat
```

Este último comando creará un archivo vacío denominado **tomcat**, dentro de este archivo debemos de colocar las siguientes instrucciones:

```
#!/bin/bash  
# descripción: Tomcat Start Stop Restart  
# nombre del proceso: tomcat  
# chkconfig: 234 20 80  
JAVA_HOME=/usr/java/jdk1.7.0_02  
export JAVA_HOME  
PATH=$JAVA_HOME/bin:$PATH  
export PATH  
CATALINA_HOME=/usr/tomcat7
```

```
case $1 in
start)
sh $CATALINA_HOME/bin/startup.sh
;;
stop)
sh $CATALINA_HOME/bin/shutdown.sh
;;
restart)
sh $CATALINA_HOME/bin/shutdown.sh
sh $CATALINA_HOME/bin/startup.sh
;;
esac
exit 0
```

Luego de editar el archivo con las líneas de código anteriores, se graba el archivo y se procede a darle permisos de ejecución

```
[root@servidor init.d]# chmod 755 tomcat
```

Luego de esto creamos el servicio para que inicie junto con el sistema y esté a demanda

```
[root@servidor init.d]# chkconfig --add tomcat
[root@servidor init.d]# chkconfig --level 234 tomcat on
```

Podemos verificar la instanciación del servicio de la siguiente manera

```
[root@servidor init.d]# chkconfig --list tomcat
tomcat    0:off 1:off 2:on 3:on 4:on 5:off 6:off
```

Ahora podemos iniciar el servicio haciendo lo siguiente:

```
[root@servidor init.d]# cd /root
```

```
[root@servidor ~]# service tomcat start
```

```
[root@morris ~]# service tomcat start
Using CATALINA_BASE:   /usr/local/tomcat
Using CATALINA_HOME:   /usr/local/tomcat
Using CATALINA_TMPDIR: /usr/local/tomcat/temp
Using JRE_HOME:        /usr/java/jdk1.7.0_02
Using CLASSPATH:       /usr/local/tomcat/bin/bootstrap.jar:/usr/local/tomcat/bin/tomcat-juli.jar
[root@morris ~]# █
```

Luego debemos probar que el servicio se pueda detener

```
[root@servidor ~]# service tomcat stop
```

```
[root@morris ~]# service tomcat stop
Using CATALINA_BASE:   /usr/local/tomcat
Using CATALINA_HOME:   /usr/local/tomcat
Using CATALINA_TMPDIR: /usr/local/tomcat/temp
Using JRE_HOME:        /usr/java/jdk1.7.0_02
Using CLASSPATH:       /usr/local/tomcat/bin/bootstrap.jar:/usr/local/tomcat/bin/tomcat-juli.jar
[root@morris ~]#
```

También podemos probar que el servicio haga un reinicio

```
[root@servidor ~]# service tomcat restart
```

```
[root@morris ~]# service tomcat restart
Using CATALINA_BASE:   /usr/local/tomcat
Using CATALINA_HOME:   /usr/local/tomcat
Using CATALINA_TMPDIR: /usr/local/tomcat/temp
Using JRE_HOME:        /usr/java/jdk1.7.0_02
Using CLASSPATH:       /usr/local/tomcat/bin/bootstrap.jar:/usr/local/tomcat/bin/tomcat-juli.jar
Using CATALINA_BASE:   /usr/local/tomcat
Using CATALINA_HOME:   /usr/local/tomcat
Using CATALINA_TMPDIR: /usr/local/tomcat/temp
Using JRE_HOME:        /usr/java/jdk1.7.0_02
Using CLASSPATH:       /usr/local/tomcat/bin/bootstrap.jar:/usr/local/tomcat/bin/tomcat-juli.jar
[root@morris ~]#
```

Una última prueba la podemos hacer a través del navegador y hacer lo siguiente:

<http://localhost:8080>

Esto debería desplegar la siguiente pantalla, mostrando la instalación correcta del **TOMCAT**.



Apache Tomcat



The Apache Software Foundation  
<http://www.apache.org/>

<b>Administration</b>
<a href="#">Status</a>
<a href="#">Tomcat Manager</a>
<b>Documentation</b>
<a href="#">Release Notes</a>
<a href="#">Change Log</a>
<a href="#">Tomcat Documentation</a>
<b>Tomcat Online</b>
<a href="#">Home Page</a>
<a href="#">FAQ</a>
<a href="#">Bug Database</a>
<a href="#">Open Bugs</a>
<a href="#">Users Mailing List</a>
<a href="#">Developers Mailing List</a>
<a href="#">IRC</a>
<b>Miscellaneous</b>
<a href="#">Servlets Examples</a>
<a href="#">JSP Examples</a>
<a href="#">Sun's Java Server Pages Site</a>
<a href="#">Sun's Servlet Site</a>

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

As you may have guessed by now, this is the default Tomcat home page. It can be found on the local filesystem at:

`$CATALINA_HOME/webapps/ROOT/index.html`

where "\$CATALINA\_HOME" is the root of the Tomcat installation directory. If you're seeing this page, and you don't think you should be, then you're either a user who has arrived at new installation of Tomcat, or you're an administrator who hasn't got his/her setup quite right. Providing the latter is the case, please refer to the [Tomcat Documentation](#) for more detailed setup and administration information than is found in the INSTALL file.

**NOTE: For security reasons, using the manager webapp is restricted to users with role "manager".** Users are defined in `$CATALINA_HOME/conf/tomcat-users.xml`.

Included with this release are a host of sample Servlets and JSPs (with associated source code), extensive documentation, and an introductory guide to developing web applications.

Tomcat mailing lists are available at the Tomcat project web site:

- [users@tomcat.apache.org](mailto:users@tomcat.apache.org) for general questions related to configuring and using Tomcat
- [dev@tomcat.apache.org](mailto:dev@tomcat.apache.org) for developers working on Tomcat

Thanks for using Tomcat!

Powered by



TOMCAT  
Copyright © 1999-2011 Apache Software Foundation  
All Rights Reserved

La estructura de directorios del **TOMCAT** debería de quedar de la siguiente manera:

```
[root@morris ~]# ls -ltr /usr/local/tomcat/
total 132
-rw-r--r--  1 root  root  10597 nov  20  15:38 RUNNING.txt
-rw-r--r--  1 root  root   8824 nov  20  15:38 RELEASE-NOTES
-rw-r--r--  1 root  root   1192 nov  20  15:38 NOTICE
-rw-r--r--  1 root  root  56797 nov  20  15:38 LICENSE
drwxr-xr-x  2 root  root   4096 ene   8  12:55 temp
drwxr-xr-x  2 root  root   4096 ene   8  12:55 bin
drwxr-xr-x  2 root  root   4096 ene   8  13:50 lib
drwxr-xr-x  3 root  root   4096 ene   8  13:57 work
drwxr-xr-x  3 root  root   4096 ene   8  13:57 conf
drwxr-xr-x 11 root  root   4096 feb   7  17:52 webapps
drwxr-xr-x  2 root  root  20480 may   3  00:04 logs
[root@morris ~]# █
```

Toda librería del tipo **.jar**, ya sean propias de las aplicaciones o librerías del tipo **DATASOURCE** (conectividad a todo tipo de bases de datos) deben ser incluidas dentro del directorio **lib** del **TOMCAT**, luego debe ser reiniciado para que los cambios tengan efecto.

Para poder cambiar el puerto por defecto que el **TOMCAT** utiliza que es el 8080 se debe editar el archivo de configuración denominado **server.xml** que se encuentra dentro del directorio **conf** del **TOMCAT**, la imagen a continuación muestra el contenido de este archivo:

```
-->↓
<Connector port="8080" protocol="HTTP/1.1" ↓
    connectionTimeout="20000" ↓
    redirectPort="8443" />↓
```

En el sector donde indica **port=8080** se debe poner el puerto al cual queremos que se ejecute el servicio del **TOMCAT**. Luego de esto se reinicia el servicio.

Dentro del directorio **webapps** del **TOMCAT** se deben colocar las aplicaciones que queremos ejecutar con este servidor de aplicaciones, por cada aplicación nueva que incluyamos debemos reiniciar el servicio.

Para poder administrar desde la consola web, debemos editar el archivo **tomcat-user.xml** ubicado en el directorio **conf** del **TOMCAT**. Se debe editar este archivo e incluir lo siguiente:

```
<tomcat-users>
<role rolename="manager-gui"/>
<user username="tomcat" password="secret" roles="manager-gui"/>
</tomcat-users>
```





```

<tomcat-users>↓
  <!--↓
  <role rolename="tomcat"/>↓
  <role rolename="role1"/>↓
  <user username="tomcat" password="tomcat" roles="tomcat"/>↓
  <user username="both" password="tomcat" roles="tomcat,role1"/>↓
  <user username="role1" password="tomcat" roles="role1"/>↓
-->↓
<user name="admin" password="adminadmin" roles="admin,manager"/>↓
</tomcat-users>↓

```

Ya sea la forma descrita en las líneas más arriba o de la manera que está en el recuadro en rojo de la imagen anterior, se puede tener acceso a la administración web del **TOMCAT** tal como se ve en la imagen a continuación.

### Gestor de Aplicaciones Web de Tomcat

Mensaje: OK

**Gestor**

[Listar Aplicaciones](#)
[Ayuda HTML de Gestor](#)
[Ayuda de Gestor](#)
[Estado de Servidor](#)

Aplicaciones				
Trayectoria	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/	Welcome to Tomcat	true	0	Arrancar Parar Recargar Replegar <input type="button" value="Expirar sesiones"/> sin trabajar z <input type="text" value="30"/> minutos
/bi		true	0	Arrancar Parar Recargar Replegar <input type="button" value="Expirar sesiones"/> sin trabajar z <input type="text" value="30"/> minutos
/conf		true	0	Arrancar Parar Recargar Replegar <input type="button" value="Expirar sesiones"/> sin trabajar z <input type="text" value="30"/> minutos
/docs	Tomcat Documentation	true	0	Arrancar Parar Recargar Replegar <input type="button" value="Expirar sesiones"/> sin trabajar z <input type="text" value="30"/> minutos
tomcat.apache.org	Servlet and JSP Examples	true	0	Arrancar Parar Recargar Replegar <input type="button" value="Expirar sesiones"/> sin trabajar z <input type="text" value="30"/> minutos

## PASO2: INSTALACIÓN DEL JBOSS (OPCIONAL):

Este paso describe la instalación y configuración del **JBOSS**, primero descargamos el paquete correspondiente a la distribución Linux que estemos usando.

El **JBOSS**, viene en paquetes del tipo **zip**. La dirección de descarga es:

<http://www.jboss.org/jbossas/downloads/>

Una vez descargado el archivo, hacemos lo siguiente:

```
[root@servidor ~]# mv jboss-as-distribution-6.1.0.Final.zip /usr/jboss-as-distribution-6.1.0.Final.zip
```

Luego debemos descomprimir el archivo de la siguiente manera

```
[root@servidor ~]# cd /usr
```

```
[root@servidor usr]# unzip -uq jboss-as-distribution-6.1.0.Final.zip
```

```
[root@servidor usr]# mv jboss-as-distribution-6.1.0.Final jboss6
```

Con eso hemos cambiado el nombre del directorio que tiene por defecto a un nombre mucho más corto que nos permita operar de manera más cómoda.

Luego debemos añadir el usuario y el grupo denominado **jboss** para que el servidor pueda iniciar correctamente. La manera de añadir en Linux es:

```
[root@servidor ~]# groupadd jboss
```

```
[root@servidor ~]# useradd -s /bin/bash -g jboss jboss
```

Luego debemos de cambiar de usuario y de grupo al directorio **jboss6** que hemos creado en **/usr**, este cambio se realiza de la siguiente manera:

```
[root@servidor ~]# chown -Rf jboos.jboss /usr/jboss6
```

Ahora al igual que en el **JDK** se debe crear las variables de entorno, para que pueda ser reconocido en todo el sistema operativo, se puede realizar de dos formas, la primera es modificando el archivo **bashrc** (para algunas distribuciones de Linux) o **bash.bashrc** (en otras distribuciones de Linux) ubicado en **/etc**. Cada que inicia el sistema ejecuta este archivo y asigna estas variables para que sean parte del entorno.

La segunda forma es creando un script del tipo **shell (.sh)** que se ejecute cada vez que inicia el sistema operativo y que debe ser ubicado en **/etc/profile.d** Este Shell se ejecutará de manera automática y asignará y publicará las variables.

Si realizamos la edición del archivo **bashrc** o **bash.bashrc** se debe introducir las siguientes líneas de código al final del archivo:

```
JBOSS_HOME=/usr/jboss6  
export JBOSS_HOME
```

Una vez introducidas las líneas anteriores grabamos el archivo, debemos comprobar que la variable **JBOSS\_HOME** esté disponible en el entorno, para esto hacemos:

```
[root@servidor etc]# echo $JBOSS_HOME
```

Debería de mostrar nos una respuesta como la siguiente:

```
/usr/jboss6
```

Con esto ya se han establecido las variables de entorno.

La otra manera de realizar esta operación es crear un archivo denominado **jboss.sh**, ese archivo debería de contener lo siguiente:

```
#jboss AS
JBOSS_HOME=/usr/jboss7
export JBOSS_HOME
```

Se debe guardar el archivo y copiarlo a **/etc/profile.d** de la siguiente manera:

```
[root@servidor jboss6]# cp jboss.sh /etc/profile.d
```

Una vez copiado el archivo en el directorio **profile.d** debemos ejecutar el siguiente comando para que el archivo sea ejecutable cuando inicia el sistema.

```
[root@servidor jboss6]# chmod 755 /etc/profile.d/jboss.sh #sh /etc/profile.d/jboss.sh
```

Luego verificamos que las variables hayan sido asumidas por el entorno

```
[root@servidor jboss6]# echo $JBOSSHOME
```

Debería de mostrar nos una respuesta como la siguiente:

```
/usr/jboss6
```

Ahora es momento de editar el script que permite ejecutar al **JBOSS** como servicio de Linux, esto permitirá iniciar el servidor cada que el sistema operativo inicie e instanciará el servicio para poder subir y bajar a demanda, sin necesidad de ejecutar los scripts del directorio **bin** del **JBOSS**.

Hacemos lo siguiente para crear el servicio:

```
[root@servidor ~]# cd /usr/jboss6/bin
[root@servidor ~]# cp jboss_init_redhat.sh /etc/init.d/jboss
```

En este caso estamos hablando de una distribución de RED HAT Linux, por eso copiamos el archivo mencionado, existen otros archivos con el mismo formato para otras distribuciones de Linux.

Debemos editar las líneas que se ven en negrillas a continuación:

```
#!/bin/sh
#
# description: JBoss Start Stop Restart
# processname: jboss6
# chkconfig: 234 20 80
#
# $Id: jboss_init_redhat.sh 81068 2008-11-14 15:14:35Z dimitris@jboss.org $
#
# JBoss Control Script
#
# To use this script run it as root - it will switch to the specified user
#
# Here is a little (and extremely primitive) startup/shutdown script
# for RedHat systems. It assumes that JBoss lives in /usr/local/jboss,
# it's run by user 'jboss' and JDK binaries are in /usr/local/jdk/bin.
```

```

# All this can be changed in the script itself.
#
# Either modify this script for your requirements or just ensure that
# the following variables are set correctly before calling the script.

#define where jboss is - this is the directory containing directories log, bin, conf etc
JBOSS_HOME=${JBOSS_HOME:-"/usr/jboss6"}

#define the user under which jboss will run, or use 'RUNASIS' to run as the current user
JBOSS_USER=${JBOSS_USER:-"jboss"}

#make sure java is in your path
JAVAPATH=${JAVAPATH:-"/usr/java/jdk1.7.0_02"}

#configuration to use, usually one of 'minimal', 'default', 'all'
JBOSS_CONF=${JBOSS_CONF:-"default"}

#if JBOSS_HOST specified, use -b to bind jboss services to that address
JBOSS_HOST="0.0.0.0"
JBOSS_BIND_ADDR=${JBOSS_HOST:+"-b $JBOSS_HOST"}

#define the classpath for the shutdown class
JBOSSCP=${JBOSSCP:-"$JBOSS_HOME/bin/shutdown.jar:$JBOSS_HOME/client/jnet.jar"}

#define the script to use to start jboss
JBOSSSH=${JBOSSSH:-"$JBOSS_HOME/bin/run.sh -c $JBOSS_CONF $JBOSS_BIND_ADDR"}

if [ "$JBOSS_USER" = "RUNASIS" ]; then
    SUBIT=""
else
    SUBIT="su - $JBOSS_USER -c "
fi

if [ -n "$JBOSS_CONSOLE" -a ! -d "$JBOSS_CONSOLE" ]; then
    # ensure the file exists
    touch $JBOSS_CONSOLE
    if [ ! -z "$SUBIT" ]; then
        chown $JBOSS_USER $JBOSS_CONSOLE
    fi
fi

if [ -n "$JBOSS_CONSOLE" -a ! -f "$JBOSS_CONSOLE" ]; then
    echo "WARNING: location for saving console log invalid: $JBOSS_CONSOLE"
    echo "WARNING: ignoring it and using /dev/null"
    JBOSS_CONSOLE="/dev/null"
fi

#define what will be done with the console log
JBOSS_CONSOLE=${JBOSS_CONSOLE:-"/dev/null"}

```

```

JBOSS_CMD_START="cd $JBOSS_HOME/bin; $JBOSSSH"
JBOSS_CMD_STOP=${JBOSS_CMD_STOP:-"java -classpath $JBOSSCP org.jboss.Shutdown --shutdown"}

if [ -z ""echo $PATH | grep $JAVAPTH"" ]; then
    export PATH=$PATH:$JAVAPTH
fi

if [ ! -d "$JBOSS_HOME" ]; then
    echo JBOSS_HOME does not exist as a valid directory : $JBOSS_HOME
    exit 1
fi

echo JBOSS_CMD_START = $JBOSS_CMD_START

case "$1" in
start)
    cd $JBOSS_HOME/bin
    if [ -z "$SUBIT" ]; then
        eval $JBOSS_CMD_START >${JBOSS_CONSOLE} 2>&1 &
    else
        $SUBIT "$JBOSS_CMD_START >${JBOSS_CONSOLE} 2>&1 &"
    fi
    ;;
stop)
    if [ -z "$SUBIT" ]; then
        $JBOSS_CMD_STOP
    else
        $SUBIT "$JBOSS_CMD_STOP"
    fi
    ;;
restart)
    $0 stop
    $0 start
    ;;
*)
    echo "usage: $0 (start | stop | restart | help)"
esac

```

Luego de editar el archivo con las líneas de código anteriores, se graba el archivo y se procede a darle permisos de ejecución

```
[root@servidor init.d]# chmod 755 jboss
```

Luego de esto creamos el servicio para que inicie junto con el sistema y esté a demanda

```
[root@servidor init.d]# chkconfig --add jboss
[root@servidor init.d]# chkconfig --level 234 jboss on
```

Ahora podemos iniciar el servicio haciendo lo siguiente:

```
[root@servidor ~]# service jboss start
```

```
root@Wexford:/home/more# service jboss start
JBoss CMD START = cd /usr/jboss6/bin; /usr/jboss6/bin/run.sh -c default -b 0.0.0.0
Sin directorio, entrando con HOME=/
root@Wexford:/home/more#
```

Luego debemos probar que el servicio se pueda detener

```
[root@servidor ~]# service jboss stop
```

```
root@Wexford:/home/more# service jboss stop
JBoss CMD START = cd /usr/jboss6/bin; /usr/jboss6/bin/run.sh -c default -b 0.0.0.0
Sin directorio, entrando con HOME=/
-su: java: orden no encontrada
root@Wexford:/home/more#
```

También podemos probar que el servicio haga un reinicio

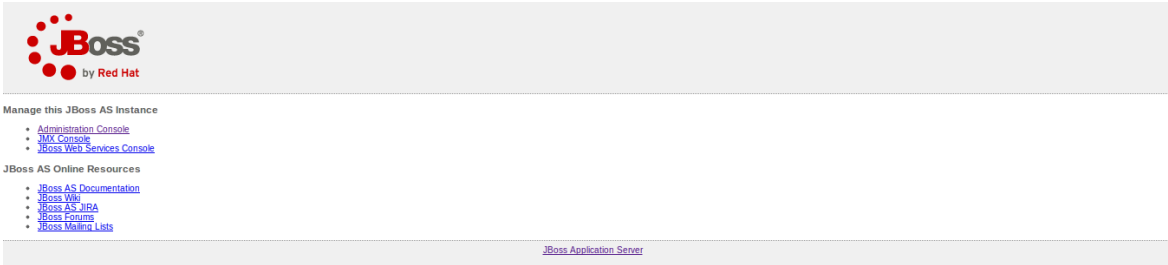
```
[root@servidor ~]# service jboss restart
```

```
root@Wexford:/home/more# service jboss restart
JBoss CMD START = cd /usr/jboss6/bin; /usr/jboss6/bin/run.sh -c default -b 0.0.0.0
JBoss CMD START = cd /usr/jboss6/bin; /usr/jboss6/bin/run.sh -c default -b 0.0.0.0
Sin directorio, entrando con HOME=/
-su: java: orden no encontrada
JBoss CMD START = cd /usr/jboss6/bin; /usr/jboss6/bin/run.sh -c default -b 0.0.0.0
Sin directorio, entrando con HOME=/
root@Wexford:/home/more#
```

Una última prueba la podemos hacer a través del navegador y hacer lo siguiente:

<http://localhost:8080>

Esto debería desplegar la siguiente pantalla, mostrando la instalación correcta del **JBoss**.



The screenshot shows the JBoss AS Administration Console interface. At the top left is the JBoss logo with the text "by Red Hat". Below it, there is a section titled "Manage this JBoss AS Instance" with links for "Administration Console", "JMX console", and "JBoss Web Services Console". Another section titled "JBoss AS Online Resources" contains links for "JBoss AS Documentation", "JBoss Wiki", "JBoss AS JIRA", "JBoss Forums", and "JBoss Mailing Lists". At the bottom, there is a link for "JBoss Application Server".

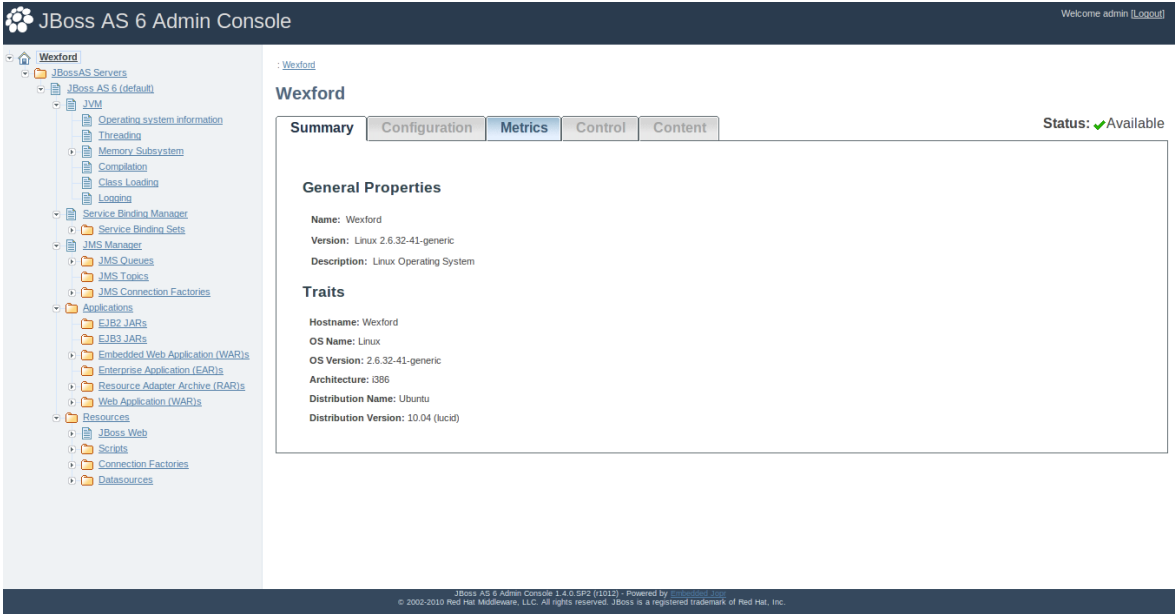
En la consola de comandos mientras se ejecuta el **JBoss** se tiene información en tiempo de ejecución, se ven datos como los mostrados en la figura a continuación:

```

19:00:35,035 INFO [org.jboss.seam.contexts.Contexts] starting up: org.jboss.seam.security.identity
19:00:35,038 INFO [org.jboss.seam.contexts.Contexts] starting up: org.jboss.seam.web.session
19:00:35,038 INFO [org.jboss.seam.contexts.Contexts] starting up: sessionInfoAction
19:00:46,444 INFO [org.jboss.on.embedded.ui.BootstrapAction] Initializing Administration Console v1.4.0.SP2...
19:00:46,483 INFO [org.rhq.core.pc.PluginContainer] Initializing Plugin Container v3.0.0...
19:00:47,937 INFO [org.rhq.core.clientapi.agent.metadata.PluginDependencyGraph] Optional plugin [JBossAS] was requested by plugins [[Hibernate]] but it does not exist
in the dependency
graph yet and will be ignored
19:00:47,938 INFO [org.rhq.core.clientapi.agent.metadata.PluginDependencyGraph] Optional plugin [JBossAS] was requested by plugins [[Infinispan]] but it does not exist
t in the dependency
graph yet and will be ignored
19:00:48,860 INFO [org.rhq.core.clientapi.agent.metadata.PluginMetadataParser] There is no resource type named [JBossAS Server] from a plugin named [JBossAS]. This is
probably because
that plugin is missing. Resource Type [Infinispan Cache Manager] will not have that missing type as a possible parent.
19:00:49,023 INFO [org.rhq.core.clientapi.agent.metadata.PluginMetadataParser] There is no resource type named [JBossAS Server] from a plugin named [JBossAS]. This is
probably because
that plugin is missing. Resource Type [Hibernate Statistics] will not have that missing type as a possible parent.
19:00:49,024 INFO [org.rhq.core.pc.inventory.InventoryManager] Initializing Inventory Manager...
19:00:49,065 INFO [org.rhq.core.pc.inventory.InventoryManager] Detected new Platform [Resource[id=-2, type=linux, key=Wexford, name=Wexford, parent=null, version=Li
nux 2.6.32-41-generic]] - adding to local inventory...
19:00:49,113 INFO [org.rhq.plugins.platform.LinuxPlatformComponent] Internal yum server is disabled.
19:00:49,120 INFO [org.rhq.core.pc.inventory.InventoryManager] Inventory Manager initialized.
19:00:49,133 INFO [org.rhq.core.pc.inventory.ResourceFactoryManager] Initializing
19:00:49,133 INFO [org.rhq.core.pc.content.ContentManager] Initializing Content Manager...
19:00:49,133 INFO [org.rhq.core.pc.content.ContentManager] Content Manager initialized...
19:00:49,138 INFO [org.rhq.core.pc.PluginContainer] Plugin Container initialized.
19:00:49,149 INFO [org.jboss.on.embedded.manager.pc.PluginContainerResourceManager] Discovering Resources...
19:00:49,150 INFO [org.rhq.core.pc.inventory.AutoDiscoveryExecutor] Executing server discovery scan...
19:00:52,355 INFO [org.rhq.core.pc.inventory.InventoryManager] Detected new Server [Resource[id=-3, type=JBossAS Server, key=/usr/jboss6/server/default, name=JBoss AS
6 (default), parent=null, version=6.1.0.Final]] - adding to local inventory...
19:00:52,500 INFO [org.rhq.core.pc.inventory.AutoDiscoveryExecutor] Found 0 servers.
19:00:52,501 INFO [org.rhq.core.pc.inventory.RuntimeDiscoveryExecutor] Running runtime discovery scan rooted at [platform]
19:00:52,539 INFO [org.rhq.core.pc.inventory.InventoryManager] Detected new Server [Resource[id=-4, type=JBoss AS JVM, key=JVM, name=JVM, parent=null, version=1.6.0
23]] - adding to local inventory...
19:00:52,543 INFO [org.rhq.plugins.jmx.JMXServerComponent] Starting connection to JMX Server JVM
19:00:53,258 INFO [org.rhq.core.pc.inventory.RuntimeDiscoveryExecutor] Scanned [0] servers and found [0] total descendant Resources.
19:00:53,327 INFO [org.rhq.core.pc.inventory.RuntimeDiscoveryExecutor] Running runtime discovery scan rooted at [platform]
19:00:53,456 INFO [org.rhq.core.pc.inventory.RuntimeDiscoveryExecutor] Scanned [0] servers and found [0] total descendant Resources.
19:01:01,840 INFO [org.jboss.web.tomcat.service.deployers.TomcatDeployment] deploy: ctxPath=/jmx-console
19:01:17,662 INFO [org.jboss.seam.contexts.Contexts] starting up: org.jboss.seam.security.identity
19:01:17,666 INFO [org.jboss.seam.contexts.Contexts] starting up: org.jboss.seam.web.session
19:01:17,666 INFO [org.jboss.seam.contexts.Contexts] starting up: sessionInfoAction

```

Si ingresamos al link denominado **ADMINISTRATION CONSOLE** tendríamos una vista como la siguiente:

A screenshot of the JBoss AS 6 Admin Console. The top header shows "JBoss AS 6 Admin Console" and "Welcome admin [Logout]". The left sidebar contains a tree view of the server configuration, including "JBossAS Servers", "JBoss AS 6 (default)", "JVM", "Service Binding Manager", "JMS Manager", "Applications", and "Resources". The main content area is titled "Wexford" and shows a "Summary" tab selected. The status is "Available". The "General Properties" section lists: Name: Wexford, Version: Linux 2.6.32-41-generic, and Description: Linux Operating System. The "Traits" section lists: Hostname: Wexford, OS Name: Linux, OS Version: 2.6.32-41-generic, Architecture: i386, Distribution Name: Ubuntu, and Distribution Version: 10.04 (lucid).

JBoss AS 6 Admin Console 1.4.0.SP2 (1012) - Powered by [Gizmo](#)  
© 2002-2010 Red Hat Middleware, LLC. All rights reserved. JBoss is a registered trademark of Red Hat, Inc.

En el siguiente manual tendremos la administración completa del **JBoss**.