

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
 Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
 HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
 Gestor de contenidos (Alfresco)
 Aplicaciones híbridas

Tareas programadas (Quartz)
 Gestor documental (Alfresco)
 Inversión de control (Spring)

Control de autenticación y
 acceso (Spring Security)
 UDDI
 Web Services
 Rest Services
 Social SSO
 SSO (Cas)

JPA-Hibernate, MyBatis
 Motor de búsqueda empresarial (Solr)
 ETL (Talend)

Dirección de Proyectos Informáticos.
 Metodologías ágiles
 Patrones de diseño
 TDD

BPM (jBPM o Bonita)
 Generación de informes (JasperReport)
 ESB (Open ESB)



[Home](#) | [Quienes Somos](#) | [Empleo](#) | [Foros](#) | [Tutoriales](#) | [Servicios Gratuitos](#) | [Contacte](#)

Tutorial desarrollado por: [Carlos León Villamayor](#)

Puedes encontrarme en [Autentia](#)



Descargar este documento en formato PDF [mod_jk_linux_all.pdf](#)

CCS

Centro de Cálculo de Sabadell 1ª
empresa española de software

Jboss Jmx Console

Monitor JBoss JVM, JMX, EJBs, DB
Easy configuration. Download
now!

XML to PDF in Java & J2EE

Output PDF, PCL5, HTML in Java
J2EE Websphere, Weblogic,
Tomcat, Jetty.

Web Load Testing Tool

Easy to use and Powerful.
Monitors, Reporting, SSL, IP
Spoofing

Anuncios Goooooogle

Anúnciese en este sitio

Instalación de mod_jk en la plataforma linux Apache2-JBoss

Tanto en mi proyecto fin de carrera como en [Autentia](#) siempre estamos buscando soluciones a problemas comunes en varias plataformas, en este caso, la instalación del **mod_jk** sobre cualquier distribución **linux**.

Introducción:

El conector "**mod_jk**" se encarga de enviar las peticiones que nosotros configuremos desde un servidor web a un servidor de aplicaciones. Esta estrategia se suele utilizar para liberar de carga al servidor de aplicaciones; para ello dejaremos que la parte estática de nuestra aplicación web sea procesada por el servidor [apache](#), mientras que la parte dinámica sea gestionada por el servidor de aplicaciones [Jboss](#). También se podría aplicar para dejar nuestro servidor de aplicaciones oculto.

En este caso de ejemplo tan solo tendremos acceso a las consolas de administración de **JBoss**.

Como aclaración inicial, partiremos de una instalación de los programas por separado de cada distribución de linux. Para ello necesitamos la descarga del siguiente software (ver los enlaces a pie de página):

- Compilador **gcc** instalado para compilar los fuentes de apache2 y de mod_jk.
- Distribución de **J2SE 1.4.X** (no es recomendable usar todavía la 1.5 con JBoss).
- Servidor Web **Apache2** (en este tutorial utilizaremos la versión 2.0.55).
- Conector **mod_jk** (en este tutorial usaremos la versión 1.2.15).
- Servidor de aplicaciones **JBoss** (en este tutorial utilizaremos la versión 4.0.3 SP1).

NOTA: Todos los pasos los haremos como **root** y las distribuciones de cada componente estarán descargadas en el directorio **/root**. Seguiremos el siguiente orden de instalación:

1. Instalación del servidor Web Apache2:

Para la instalación y partiendo del fuente:

1. `[root@localhost root]# tar -xvzf httpd-2.0.55.tar.gz`
2. `[root@localhost root]# cd httpd-2.0.55`
3. `[root@localhost root]# ./configure`
4. `[root@localhost root]# make`
5. `[root@localhost root]# make install`
6. `[root@localhost root]# groupadd jboss`
7. `[root@localhost root]# useradd jboss -g jboss`
8. Ahora editamos el fichero `"/usr/local/apache2/conf/httpd.conf"` y modificamos los parámetros "user" y "group" por "jboss".

Una vez finalizada la instalación probamos el servidor web como se muestra a continuación:

```
[root@localhost root]# /usr/local/apache2/bin/apachectl start
[root@localhost root]# ps -u jboss
  PID TTY          TIME CMD
 1506 tty3        00:00:00 bash
 1810 ?            00:00:00 httpd
 1811 ?            00:00:00 httpd
 1812 ?            00:00:00 httpd
 1813 ?            00:00:00 httpd
 1814 ?            00:00:00 httpd
[root@localhost root]#
```

2. Instalación del servidor de aplicaciones JBoss:

Partiendo del fichero descargado:

1. `[root@localhost root]# tar -xvzf jboss-4.0.3SP1.tar.gz`
2. `[root@localhost root]# mv jboss-4.0.3SP1 /usr/local`
3. `[root@localhost root]# cd /usr/local`

- Ahora editamos el archivo `"/usr/local/jboss-4.0.3SP1/bin/run.conf"` y descomentamos la línea que contiene la variable `JAVA_HOME` y la establecemos en nuestro directorio de instalación de java. En nuestro caso será:
`JAVA_HOME="/usr/java/j2sdk1.4.2_10"`
- Editamos el fichero `"/usr/local/jboss-4.0.3SP1/server/default/deploy/jbossweb-tomcat55.sar/server.xml"` y añadimos el nombre de nuestro worker como se muestra en la imagen:

```

protocol="AJP/1.3"/>

<!-- SSL/TLS Connector configuration using the admin dev1 guide keystore
<Connector port="8443" address="{jboss.bind.address}"
  maxThreads="100" strategy="ms" maxHttpHeaderSize="8192"
  emptySessionPath="true"
  scheme="https" secure="true" clientAuth="false"
  keystoreFile="{jboss.server.home.dir}/conf/chap8.keystore"
  keystorePass="rmi+ssl" sslProtocol = "TLS" />
-->

<Engine name="jboss.web" defaultHost="localhost" jvmRoute="node1">

  <!-- The JAAS based authentication and authorization realm implementati
on
  that is compatible with the jboss 3.2.x realm implementation.
  - certificatePrincipal : the class name of the
  org.jboss.security.auth.certs.CertificatePrincipal impl
  used for mapping X509[] cert chains to a Principal.
  -->
  <Realm className="org.jboss.web.tomcat.security.JBossSecurityMgrRealm"
    certificatePrincipal="org.jboss.security.auth.certs.SubjectDNMapping
  "
  />
-- SELECCIONAR --

```

16	37,72	22%
----	-------	-----

- Editamos el archivo `"/usr/local/jboss-4.0.3SP1/server/default/deploy/jbossweb-tomcat55.sar/META-INF/jboss-service.xml"` y activamos el atributo `UseJK` como se muestra en la imagen:

```

<!--
Whether to use MOD_JK(2) for load balancing with sticky session
combined with JvmRoute. If set to true, it will insert a JvmRouteFilter
to intercept every request and replace the JvmRoute if it detects a
failover. In addition, you will need to set the JvmRoute inside
Tomcat, e.g.,
  Engine name="jboss.web" jmvRoute="Node1" defaultHost="localhost"
in server.xml.

For clustering purpose only.
-->
<attribute name="UseJK">true</attribute>

<attribute name="Domain">jboss.web</attribute>

<!-- A mapping to the server security manager service which must be
operation compatible with type
org.jboss.security.plugins.JaasSecurityManagerServiceMBean. This is only
needed if web applications are allowed to flush the security manager
authentication cache when the web sessions invalidate.
-->
<depends optional-attribute-name="SecurityManagerService"
  proxy-type="attribute">jboss.security:service=JaasSecurityManager
</depends>
-- SELECCIONAR --

```

4	112,35	86%
---	--------	-----

- Asignamos este directorio al usuario creado anteriormente (`jboss`):
`[root@localhost root]# chown -R jboss:jboss /usr/local/jboss-4.0.3SP1/` Una vez finalizada la instalación del servidor de aplicaciones realizamos la prueba de ejecución. Debemos obtener una pantalla como esta o parecida:

```

19:56:39,146 INFO [ConnectionFactoryBindingService] Bound ConnectionManager 'jboss.jca:name=DefaultDS,service=DataSourceBinding' to JNDI name 'java:DefaultDS'
19:56:39,937 INFO [A] Bound to JNDI name: queue/A
19:56:39,941 INFO [B] Bound to JNDI name: queue/B
19:56:39,944 INFO [C] Bound to JNDI name: queue/C
19:56:39,948 INFO [D] Bound to JNDI name: queue/D
19:56:39,951 INFO [ex] Bound to JNDI name: queue/ex
19:56:39,997 INFO [testTopic] Bound to JNDI name: topic/testTopic
19:56:40,001 INFO [securedTopic] Bound to JNDI name: topic/securedTopic
19:56:40,005 INFO [testDurableTopic] Bound to JNDI name: topic/testDurableTopic
19:56:40,305 INFO [testQueue] Bound to JNDI name: queue/testQueue
19:56:40,406 INFO [UILServerILService] JBossMQ UIL service available at : /0.0.0.0:8093
19:56:40,492 INFO [DLQ] Bound to JNDI name: queue/DLQ
19:56:40,829 INFO [ConnectionFactoryBindingService] Bound ConnectionManager 'jboss.jca:name=JmsXA,service=ConnectionFactoryBinding' to JNDI name 'java:JmsXA'
19:56:40,885 INFO [TomcatDeployer] deploy, ctxPath=/jmx-console, warUrl=.../deploy/jmx-console.war/
19:56:42,540 INFO [Http11Protocol] Starting Coyote HTTP/1.1 on http-0.0.0.0-8080
19:56:42,927 INFO [ChannelSocket] JK: ajp13 listening on /0.0.0.0:8009
19:56:43,088 INFO [JkMain] Jk running ID=0 time=0/404 config=null
19:56:43,102 INFO [Server] JBoss (MX MicroKernel) [4.0.3SP1 (build: CUSTag=JBoss_4_0_3_SP1 date=200510231054)] Started in 1m:16s:313ms

```

NOTA: para parar el servidor nos basta con pulsar CTRL+C

3. Instalación del conector mod_jk:

Una vez tenemos instalados ambos servidores, nos disponemos a instalar el conector jk_mod. Seguiremos los siguientes pasos:

1. `[root@localhost root]# tar -xvzf jakarta-tomcat-connectors-1.2.15-src.tar.gz`
2. `[root@localhost root]# cd jakarta-tomcat-connectors-1.2.15-src/jk/native`
3. `[root@localhost native]# ./configure --with-apxs=/usr/local/apache2/bin/apxs`
4. `[root@localhost native]# make`
5. `[root@localhost native]# mv apache-2.0/mod_jk.so /usr/local/apache2/modules/`
6. Editamos el archivo `"/usr/local/apache2/conf/httpd.conf"` y añadimos al final del todo la siguiente línea:

```
Include conf/mod-jk.conf
```

7. Creamos el archivo `"/usr/local/apache2/conf/mod-jk.conf"` con el siguiente contenido:

```

# Cargamos el modulo
LoadModule jk_module modules/mod_jk.so

# workers.properties
JkWorkersFile conf/workers.properties

# logs
JkLogFile logs/mod_jk.log

# jk nivel log [debug/error/info]
JkLogLevel info

# formato de log
JkLogStampFormat "[%a %b %d %H:%M:%S %Y]"

# JkOptions
JkOptions +ForwardKeySize +ForwardURICompat -ForwardDirectories

# JkRequestLogFormat
JkRequestLogFormat "%w %V %T"

# Archivo de URI de aplicaciones
JkMountFile conf/uriworkermap.properties

# Para shared memory.
JkShmFile logs/jk.shm

# jkstatus
<Location /jkstatus/>
  JkMount status
  Order deny,allow
  Deny from all
  Allow from 127.0.0.1
</Location>

```

8. Creamos el archivo `"/usr/local/apache2/conf/workers.properties"` con el siguiente contenido:

```

# Define una lista de workers que se usaran
# para mapear las peticiones
worker.list=loadbalancer,status

# Definimos el Nodo1
worker.node1.port=8009

```

```

worker.node1.host=localhost
worker.node1.type=ajp13
worker.node1.lbfactor=1
worker.node1.cachesize=10

# Definimos el Nodo2
worker.node2.port=8009
worker.node2.host=localhost
worker.node2.type=ajp13
worker.node2.lbfactor=1
worker.node2.cachesize=10

worker.loadbalancer.type=lb
worker.loadbalancer.balance_workers=node1,node2
worker.loadbalancer.sticky_session=1
worker.status.type=status

```

9. Creamos el archivo `"/usr/local/apache2/conf/uriworkermap.properties"` con el siguiente contenido:

```

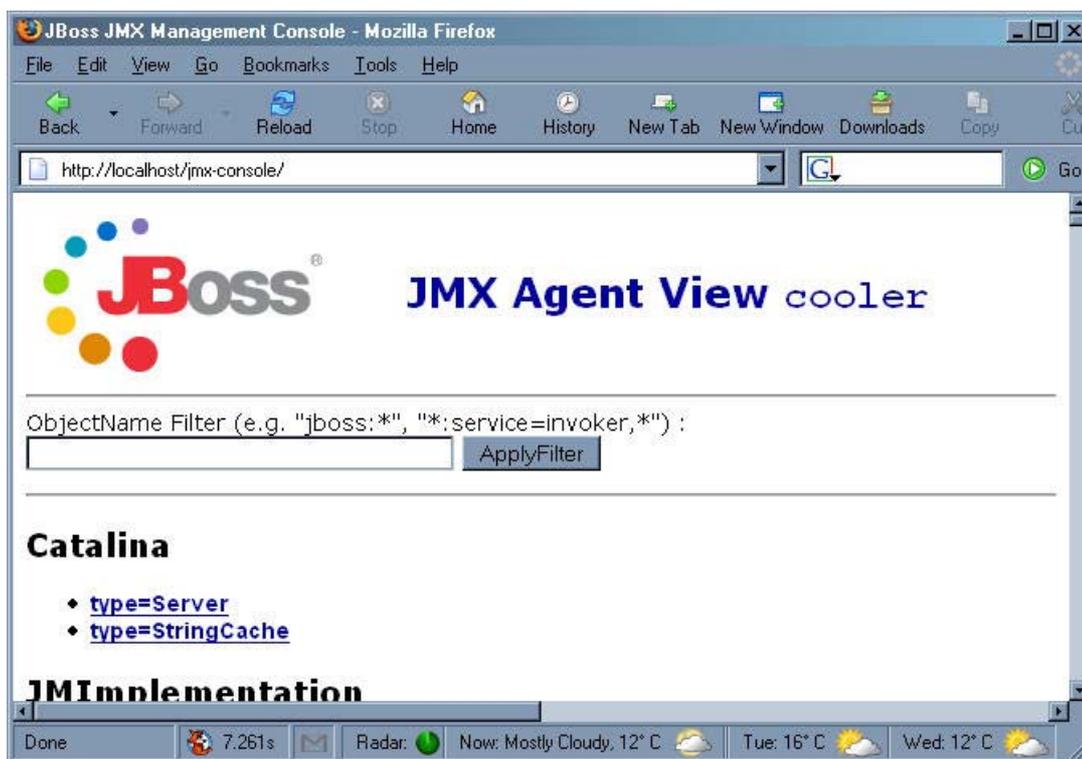
# Contextos de las aplicaciones que queremos que tengan acceso

/jmx-console=loadbalancer
/jmx-console/*=loadbalancer
/web-console=loadbalancer
/web-console/*=loadbalancer

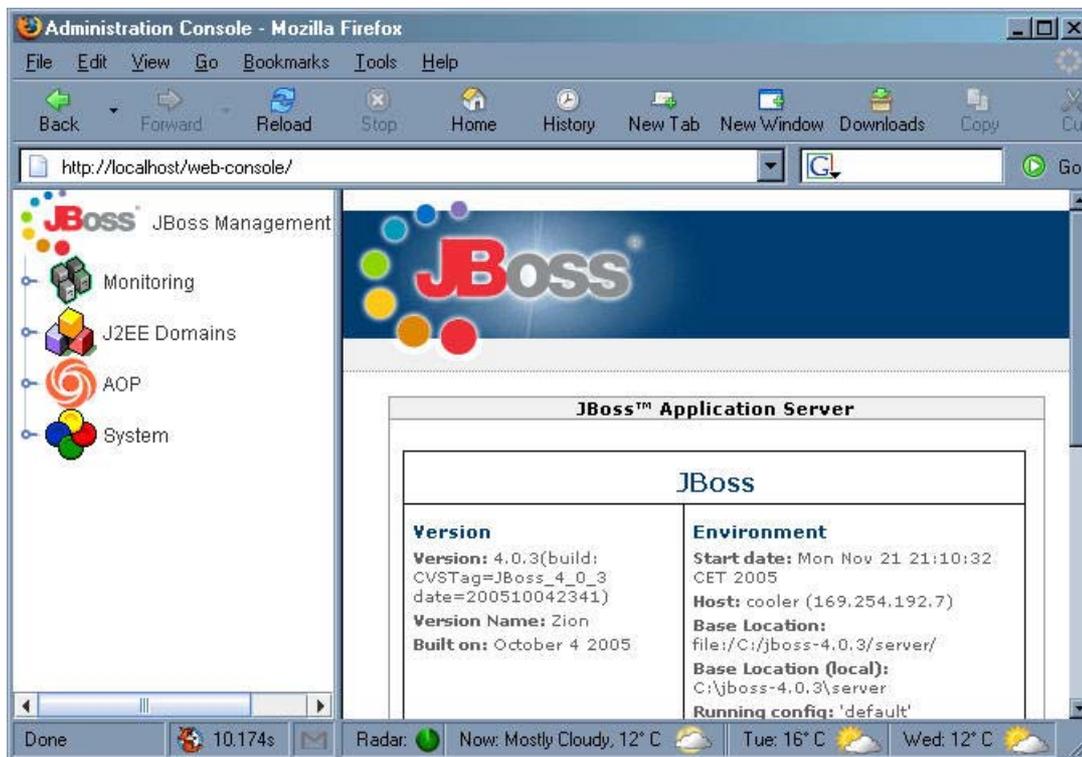
```

4. Prueba de la instalación:

<http://localhost/jmx-console/>



<http://localhost/web-console/>



Conclusión:

Mediante esta técnica, podemos añadir aplicaciones web con tan solo insertar una entrada en el archivo con el **context-root** de nuestras aplicaciones y los recursos a solicitar.

De todas formas ya sabéis donde [encontrarnos](#) si necesitáis cursos o soluciones personalizadas para vuestro problema.

Enlaces:

[mod_jk-1.2.15](#)

[Apache2-2.0.55](#)

[JBoss 4.0.3 RC1](#)

[j2se-1.4.2](#)

Si desea contratar formación, consultoría o desarrollo de piezas a medida puede contactar con

Gestión de contenidos

[Autentia S.L.](#) Somos expertos en:
J2EE, C++, OOP, UML, Vignette, Creatividad ..
 y muchas otras cosas

Nuevo servicio de notificaciones

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales, inserta tu dirección de correo en el siguiente formulario.

Subscribirse a Novedades	
e-mail	<input type="text"/>
	<input type="button" value="Enviar"/>

Otros Tutoriales Recomendados ([También ver todos](#))

Nombre Corto	Descripción
Navegador Mozilla FireFox	En esta ocasión probamos el estado de evolución del navegador gratuito Mozilla FireFox, una verdadera alternativa en el mercado.
Certificados en IIS para activación SSL	En este tutorial vamos a habilitar el soporte SSL (Secure Socket Layer, comunicación segura por https) en un servidor IIS (Internet Information Server de Microsoft).
Arquitectura SOA e Integración de aplicaciones	Os presentamos un breve pero intenso pdf sobre la tecnología Arquitectura SOA y la Integración de aplicaciones
Manejar dos bases de datos distintas con Hibernate	Alejandro Pérez nos enseña como manejar dos bases de datos distintas con Hibernate
Framework desarrollo eclipse	Aquí os mostramos algunas de las características de Eclipse
Activar el soporte SSL en Struts	Os mostramos las particularidades de uso y configuración de Struts para trabajar con SSL
Almacenamiento en Windows Pocket 2003	Cesar Crespo nos enseña como utilizar ObjectStore en nuestros programas Visual C++ . El "Object Store" (ObS) en Windows Pocket 2003 cumple en muchos casos la misma función que el disco duro en un equipo de escritorio.
Reingeniería JDO con Druid	Os mostramos como crear vuestras clases y descriptores JDO, de tablas existentes, con la herramienta gratuita Druid.
Uso de Tiles en Struts	Os mostramos como utilizar el sistema de plantillas proporcionado por Struts (tiles)
Modelado UML con Visual Paradigm	Os mostramos como instalar y utilizar la versión gratuita de Visual Paradigm for UML. Cabe destacar que permite extraer elementos de diseño desde textos de análisis.

Nota: Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores.

En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo.

Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.

[Patrocinados por enredados.com Hosting en Castellano con soporte Java/J2EE](#)

