

Configurar Apache HTTPd con Jakarta Tomcat Connector (mod_jk)

{scrollbar}

top

La comunicación entre el servidor HTTP y Geronimo, también puede realizarse mediante conectores AJP. Por defecto, ambas distribuciones Apache Geronimo (Jetty y Tomcat), ya cuentan con un oyente AJP v13 predefinido; los puertos donde esos oyentes están definidos, varían de una distribución a la otra.

- Para Jetty el puerto del oyente AJP13 predefinido es 8019
- Para Tomcat el puerto del oyente AJP13 predefinido es 8009

Esta sección cubrirá la configuración para ambas distribuciones Geronimo, ya que los pasos a seguir son muy similares (más no iguales).

El módulo Jakarta Tomcat Connector **mod_jk** se proporciona como un conector en el código fuente de Apache Tomcat. Jetty (y obviamente Tomcat) es completamente compatible con este conector. Este módulo está disponible cuando descargas al código fuente de Tomcat, pero también está disponible como descarga separada; consulta la siguiente URL para obtener la versión apta para tu sistema.

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

Además de **mod_jk**, requerirás al archivo **workers.properties**, también disponible en la distribución de código fuente de Apache Tomcat. Este archivo de propiedades indica al plugin **mod_jk** el cómo conectarse al servidor Geronimo. Para una explicación detallada en toda opción disponible dentro de la configuración de Jakarta Tomcat Connector visita la siguiente URL:

<http://tomcat.apache.org/connectors-doc/config/workers.html>

Por fines prácticos, el módulo **mod_jk_1.2.6_2.0.50.dll** para Windows y el archivo **workers.properties** se incluyen en la sección de **Attachments** (*Adjuntos*) de este artículo.

[Regresar a la sección superior](#)

Configurar Apache HTTPd

Descarga la versión de **mod_jk** apropiada para tu plataforma desde el sitio web [Tomcat](#), renómbrolo a **mod_jk.dll** (ó .so dependiendo de tu plataforma) y copialo al directorio **<httpd_home>\modules**. Descarga y expande **workers.properties** del código fuente de Apache Tomcat (como alternativa, aquí lo puedes descargar desde la sección **Attachments**) al directorio **<httpd_home>\conf**.

Edita al archivo **httpd.conf** ubicado en el directorio **<httpd_home>\conf** para cargar al módulo **mod_jk** Jakarta Tomcat Connector. Agrega las siguientes líneas al final del archivo **httpd.conf**.

```
LoadModule jk_module modules/mod_jk.dll # Carga al módulo Jakarta Tomcat Connector JkWorkersFile <httpd_home>\conf\workers.properties #
Indica al módulo la ubicación del archivo workers.properties JkLogFile <httpd_home>\logs\mod_jk.log # Especifica la ubicación del archivo bitácora
específico de este módulo JkLogLevel info # Establece el nivel de bitácora del módulo a info JkLogStampFormat "[%a %b %d %H:%M:%S %Y]" #
Establece el formato de la etiqueta de tiempo en la bitácora del módulo JkAutoAlias <httpd_home>\config-store # Automáticamente Alias a directorios del
contexto webapp hacia el espacio de documento Apache. JkMount /console/* ajp13 # Establece un punto de montaje desde un contexto a un trabajador
Tomcat. En este caso, permitirá el acceso (redirigirá la petición) a la consola.
```

JkMount mapeará cualquier cosa detrás de **/console/** al trabajador **ajp13**. El nombre **ajp13** se define en el archivo **workers.properties**, el cual se describe a continuación. Necesitarás agregar más directivas **JkMount** dependiendo de las aplicaciones que desees que sean accedidas mediante el HTTPd remoto.

En este ejemplo, la **console** ha sido habilitada sólo por fines demostrativos. En un ambiente de producción no convendrá tener a la **console** accesible desde otra red (normalmente Internet). El tener a la **console** accesible representa un gran riesgo de seguridad.

La regla es que todo debería tener acceso restringido, donde una firewall normalmente se colocaría entre los servidores HTTP y de aplicaciones (dependiendo de la topología), con lo cual sólo deberías mapear los recursos mínimos necesarios para poder tener a tu aplicación funcionando y disponible para el otro sector.

[Regresar a la sección superior](#)

Configurar workers.properties

El archivo **workers.properties** le indica a HTTPd, entre otras cosas, donde se encuentra el servidor Geronimo, que versión de AJP debería usar y el puerto en el cual Geronimo está escuchando.

Edita al archivo **workers.properties** ubicado en el directorio **<httpd_home>\conf** de tu ambiente. El siguiente ejemplo es una porción del archivo **workers.properties** empleando sólo las variables a las cuales te deberías enfocar.

```
solid workers.java_home=<java_home> # Establece a JAVA_HOME ps=\ # Para sistemas Windows emplea la diagonal inversa "\", para sistemas Unix
emplea la diagonal normal "/" worker.ajp13.port=8009 # Emplea al puerto predeterminado 8009 para Apache Tomcat # Cambia dicho valor a 8019 para
```

Jetty worker.ajp13.host=localhost # Especifica la ubicación del servidor Geronimo. Emplea localhost para escenarios de una-capa (single-tier). Especifica el nombre del host del servidor Geronimo para ambientes multi-capas (multi-tier). worker.ajp13.type=ajp13 # Establece la versión usada de AJP. Los oyentes AJP definidos en Geronimo son AJP v13.

De este ejemplo nota como el nombre del trabajador es definido, observa la definición de variables worker.ajp13.* , **ajp13** es el nombre del trabajador que especificaste previamente en el archivo httpd.conf.

Como último paso, deten y reinicia a Apache HTTPd para asegurar que los cambios previos sean cargados.

[Regresar a la sección superior](#)

Pruebas

Para probar la configuración, asegúrate de que tanto Geronimo y HTTPd se encuentren activos y en ejecución.

1. Verifica la conectividad de Geronimo mediante el acceso a <http://localhost:8080/console> , deberías obtener la Consola de Administración de Geronimo.
2. Verifica la conectividad de HTTPd mediante el acceso a <http://localhost> , deberías obtener la página de bienvenida de Apache HTTPd.
3. Verifica la redirección de peticiones HTTPd - Geronimo mediante el acceso a <http://localhost/console/> , deberías ser redirigido a la Consola de Administración de Geronimo. Nota que al final de la URL existe un "/", ya que el no incluirle causa un error **Not Found** (*No Encontrado*) por parte del módulo Jakarta Tomcat Connector.

[Regresar a la sección superior](#)