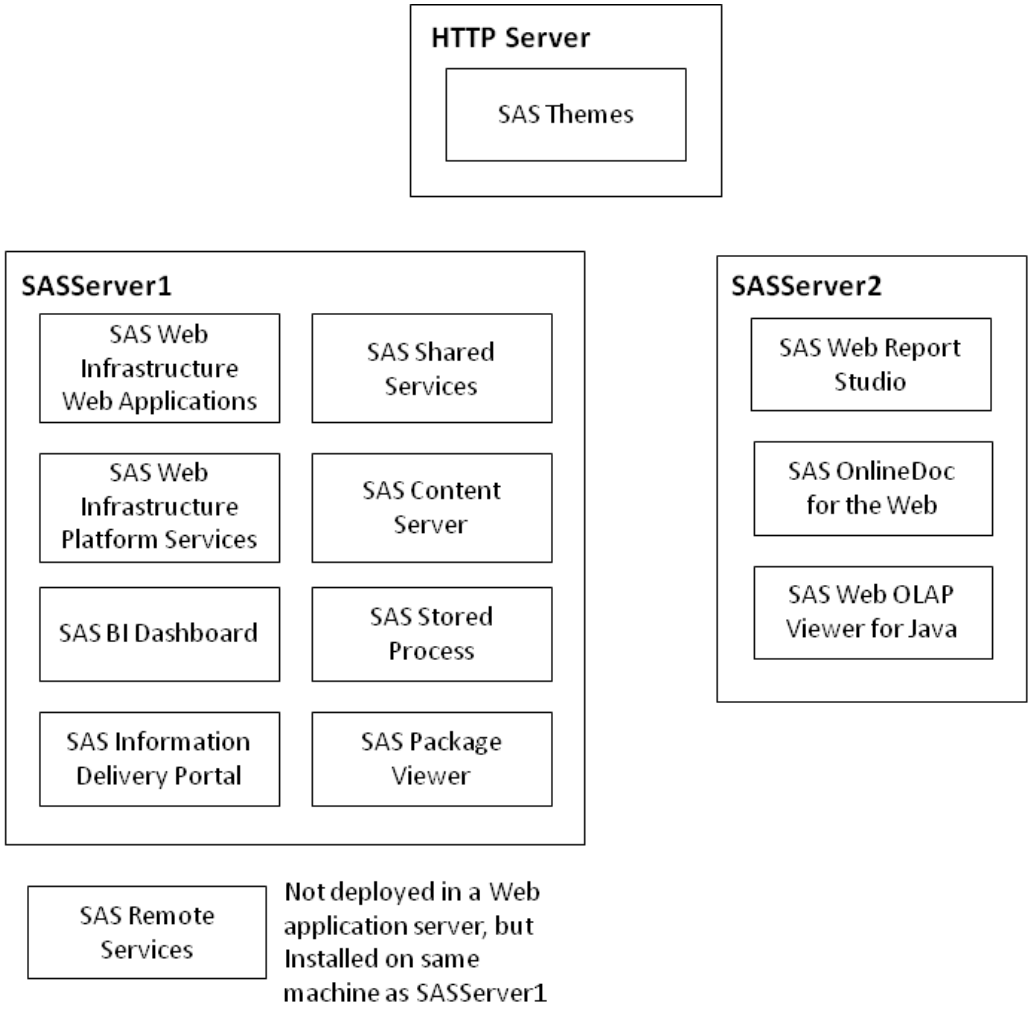
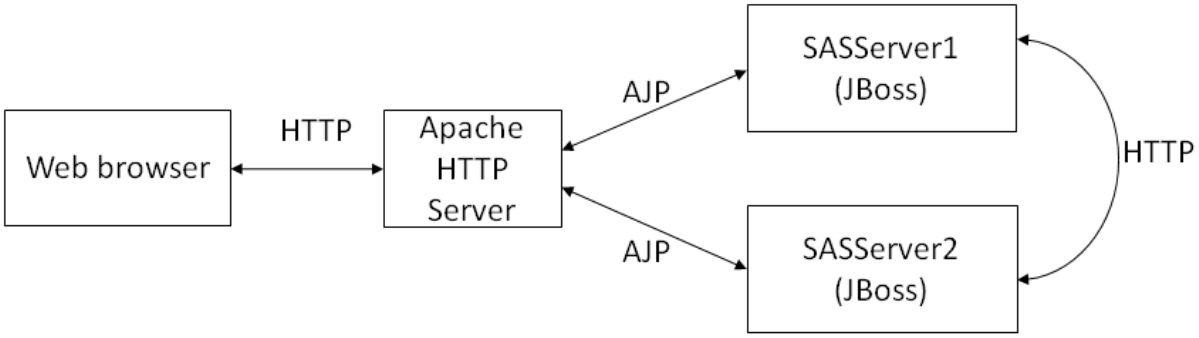
Configuring Apache HTTP Server as a Reverse Proxy Server for SAS Web Applications Deployed on JBoss

This document describes how to configure Apache HTTP Server as a reverse proxy server to two JBoss servers that are hosting the SAS 9.2 Web applications. The SAS Web applications are distributed between the two JBoss servers. The Apache HTTP Server is also configured to serve the static content from the SAS Themes Web application. This paper does not describe using a JBoss cluster. The following figure shows a simplified topology with the two servers and the distribution of SAS Web applications.



The first stage of this document configures the topology to use HTTP and AJP (Apache JServ Protocol) protocols as shown in the following figure.



Overview

The following list identifies the high-level tasks that you must perform to configure Apache HTTP Server as a reverse proxy server and reconfigure a SAS 9.2 middle tier to redistribute the SAS Web applications to two JBoss servers:

- Configure Apache HTTP Server to use mod_jk. The mod_jk software module is used to perform the proxying function and the configuration files for mod_jk identify the URL locations for the SAS Web applications.
- Configure SASServer2 with the SAS Web applications. A second managed JBoss server is needed and must be configured with the resources that are needed by the SAS Web applications such as a MailSession and a JDBC Data Source. Afterward, the EAR files for the Web applications are moved from the `deploy_sas` directory on SASServer1 to the `deploy_sas` directory on SASServer2.

Required Software

In addition to SAS 9.2 software that includes the SAS Web applications, you must have the following software available:

- Apache HTTP Server — http://httpd.apache.org
- mod_jk — http://tomcat.apache.org/download-connectors.cgi

Configure Apache HTTP Server and mod_jk

This section describes how to configure Apache HTTP Server as a reverse proxy server by using the mod_jk module. The files `workers.properties` and `uriworkermap.properties` control how Apache HTTP Server routes HTTP requests to the two JBoss servers. If you have distributed the SAS Web applications differently at your site, then you need to modify the examples shown in this document.

You must have installed mod_jk in *`APACHE_HOME`*`/modules` before beginning these steps.

Configure Apache HTTP Server to use mod_jk

The following steps show how to configure Apache HTTP Server with the mod_jk software module so that Apache HTTP Server acts as a reverse proxy HTTP server.

1. Add the following information to the *APACHE_HOME*/conf/httpd.conf file:

```
LoadModule jk_module modules/mod_jk.so

JkWorkersFile conf/workers.properties
JkLogFile logs/mod_jk.log
JkLogStampFormat "[%a %b %d %H:%M:%S %Y] "
JkMountFile conf/uriworkermap.properties
JkShmFile run/jk.shm # You might need to create this run directory
```

2. Create and add the following information to the *APACHE_HOME*/conf/workers.properties file:

```
worker.list=sasserver1,sasserver2

worker.sasserver1.host=hostname1.example.com
worker.sasserver1.port=8009
worker.sasserver1.type=ajp13

worker.sasserver2.host=hostname2.example.com
worker.sasserver2.port=8009
worker.sasserver2.type=ajp13
```

Create an *APACHE_HOME*/conf/uriworkermap.properties file. In conjunction with the workers.properties file, this file maps the location of the SAS Web applications to the JBoss servers that host each of the applications. Add the following information to the file:

```
/SASBIDashboard=sasserver1
/SASBIDashboard/*=sasserver1
/SASBIPortlets=sasserver1
/SASBIPortlets/*=sasserver1
/SASBIWS=sasserver1
/SASBIWS/*=sasserver1
/SASJSR168RemotePortlet=sasserver1
/SASJSR168RemotePortlet/*=sasserver1
/SASPortal/*=sasserver1
/SASPortal=sasserver1
/SASLogon=sasserver1
/SASLogon/*=sasserver1
/SASPackageViewer=sasserver1
/SASPackageViewer/*=sasserver1
/SASPreferences=sasserver1
/SASPreferences/*=sasserver1
/SASSharedApps=sasserver1
/SASSharedApps/*=sasserver1
/SASStoredProcess=sasserver1
/SASStoredProcess/*=sasserver1
/SASSharedPortlets=sasserver1
/SASSharedPortlets/*=sasserver1

# uncomment the next four entries if you want SAS Web Administration
# console to be available from Apache HTTP Server
```

```
#/SASAdmin=sasserver1
#/SASAdmin/*=sasserver1
#/SASContentServer=sasserver1
#/SASContentServer/*=sasserver1

# uncomment the next two entries if you want SAS Web services
# available from Apache HTTP Server
#/SASWIPSoapServices=sasserver1
#/SASWIPSoapServices/*=sasserver1

# uncomment the next two entries if the site is hosting a SAS solution
# that has desktop applications that access the SAS Web applications
#/SASWIPClientAccess=sasserver1
# /SASWIPClientAccess/*=sasserver1

/SASWebOLAPViewer=sasserver2
/SASWebOLAPViewer/*=sasserver2
/SASWebReportStudio=sasserver2
/SASWebReportStudio/*=sasserver2
/SASWebDoc=sasserver2
/SASWebDoc/*=sasserver2
```

Configure Apache HTTP Server to Serve SASThemes Static Content

The SAS Themes Web application is actually a collection of static HTML resources such as images and cascading style sheets. This step removes the sas.themes9.2.ear file from the Web application server and configures Apache HTTP Server to serve that content. By serving the SAS Themes content through the HTTP server, the processing load is shifted from the Web application server to the HTTP server.

1. In a temporary directory, extract the contents of `SAS-config-dir/Levn/Web/Staging/sas.themes9.2.ear`:

   ```
   jar xf /opt/SAS/Config/Lev1/Web/Staging/sas.themes9.2.ear
   ```

   The WAR file, `sas.theme.default.war` is extracted.

2. Change directory to *APACHE_HOME*`/htdocs` and make a new directory named `SASTheme_default`.

3. Extract the `sas.theme.default.war` file into *APACHE_HOME*`/htdocs/SASTheme_default`.

4. Move the `sas.themes9.2.ear` file out of the `deploy_sas` directory on SASServer1 so that it is not loaded unnecessarily.

Configure SASServer2

The starting configuration for this document is a SAS 9.2 middle tier that has just one JBoss server that was configured by the SAS Deployment Wizard. Adding a second JBoss server requires a high level of skill with JBoss administration. Use the configuration information at `SAS-config-dir/Levn/Documents/Instructions.html` for information about how to perform these high-level configuration steps:

1. Create and configure a Java Mail Session.

2. Create and configure a JMS Server and resources.
3. Configure JDBC Driver - SAS Table Server.
4. Configure Data Source – Shared Service.
5. Deploy the respective SAS 9.2 middle tier applications (EAR files) to the server by moving them from the `deploy_sas` directory on SASServer1 to the `deploy_sas` directory on SASServer2.
6. Configure the JAAS Login Module for Application Logins – PFS and SCS.

Configure SASServer1 and SASServer2 for Proxying

At this stage of the reconfiguration, Apache HTTP Server is prepared as the reverse proxy server and the second managed JBoss server is configured with the SAS Web applications that it hosts. This stage describes the steps that you must perform to configure each server to interface with the reverse proxy server. Follow these steps for SASServer1 and SASServer2:

1. Edit *JBOSS_HOME*`/server/SASServer`*n*`/deploy/jboss-web.deployer/server.xml` and make the following changes:

   a. Confirm that the AJP Connector on port 8009 is enabled:

   ```
   <Connector address="${jboss.bind.address}" emptySessionPath="true"
     enableLookups="false" port="8009" protocol="AJP/1.3"
     redirectPort="8443"/>
   ```

   b. Add a jvmRoute attribute to the Engine element:

   ```
   <Engine defaultHost="localhost" name="jboss.web"
     jvmRoute="sasserver1">
   ```

   Use “sasserver2” when you edit the file for SASServer2. These jvmRoute values match the names used in the uriworkermap.properties file.

   c. (Optional) Enable logging of access requests. You can disable this after confirming that your configuration is stable:

   ```
   <!-- Access logger -->
   <Valve className="org.apache.catalina.valves.AccessLogValve"
     prefix="localhost_access_log." suffix=".log"
     pattern="common" directory="${jboss.server.home.dir}/log"
     resolveHosts="false" />
   ```

2. Edit *JBOSS_HOME*`/server/SASServer`*n*`/deploy/jboss-web.deployer/META-INF/jboss-service.xml` and enable the `UseJK` attribute:

   ```
   <attribute name="UseJK" replace="true" trim="true">true</attribute>
   ```

Change the Connections for the SAS Web Applications

After SAS Themes is deployed to the Apache HTTP server and the SAS Web applications are distributed to the different servers, information about access to the applications such as host and port number must be updated in SAS metadata. You must change the connection information for the external (or customer-facing) SAS Web applications. Change the connection information to a URL

that includes the host name and port number for the Apache HTTP Server. The following list identifies the SAS Web applications that must be reconfigured:

- BI Dashboard 4.2
- BI Portlets 4.2
- BI Web Services for Java 9.2
- Help Viewer Meta Config 9.2
- Information Delivery Portal 4.2
- JSR 168 Remote Portlet 4.2
- Logon Manager 9.2
- Package Viewer 4.2
- Preferences Manager 9.2
- SASTheme_default
- Shared Applications 9.2
- Shared Portlets 9.2
- Stored Process Web App 9.2
- Web Administration Console 9.2 (optional)
- Web Infra Platfrm ClntAccss 9.2 (if the site has a SAS solution with desktop clients that access SAS Web applications)
- Web Infra Platfrm Soap Svcs 9.2 (if the site wants SAS Web services available from Apache HTTP Server)
- Web OLAP Viewer 4.2
- Web Report Studio 4.2

To change the connection access point, follow these steps in SAS Management Console:

1. Select **Application Management** > **Configuration Manager**.
2. Right-click on the SAS Web application you want to reconfigure, and select **Properties**.
3. Click the **Connection** tab, set **Host Name** and **Port Number** to the host name and port number of the Apache HTTP Server, and then click **OK**.

If you choose to provide access to the SAS Web Administration Console from the Apache HTTP Server, then you must also reconfigure the connection information for SAS Content Server. The connection information is for the WebDAV service that the SAS Content Server provides. Refer to these sections:

- "Modify SAS Metadata for SAS Web Application and WebDAV Connections" on page 10 (start at step 2)
- "Change the WebDAV Repository URL" on page 10

SAS Web Report Studio 4.2 Specific Update

By default, SAS Web Report Studio 4.2 uses a special redirection filter. When used with a proxy server, this filter must be disabled. To disable the filter, perform the following steps with SAS Management Console:

1. Select **Application Management** > **Configuration Manager**.
2. Right-click **Web Report Studio 4.2**, and select **Properties**.
3. Click **Advanced**, and then click **Add**.

4. Enter a Property Name of `App.RedirectionFilterDisabled` and a Value of `true`.

Verify the Configuration

Start, or restart, the software applications in the following order:

1. SAS Remote Services
2. Apache HTTP Server
3. SASServer1
4. SASServer2

Once the servers have started, log on to a sample Web application to confirm the configuration is valid:

1. Open a Web browser to http://httpserver.example.com/SASBIDashboard (or any SAS Web application that is deployed on SASServer1).

   You are challenged for log on credentials. Provide credentials and then click **Log On**. Confirm that the *JBOSS_HOME*`/server/SASServer1/log/localhost_access_log.`*YYYY-MM-DD*`.log` shows activity like the following example:

   ```
   10.0.0.10 - - [11/Mar/2009:16:40:22 -0400] "GET /SASBIDashboard/
   scripts/sas_FastClickProtection_CustomFunctions.js HTTP/1.1" 304 -
   10.0.0.10 - - [11/Mar/2009:16:40:22 -0400] "GET/SASBIDashboard/
   scripts/sas_FastClickProtection.js HTTP/1.1" 304 -
   ```

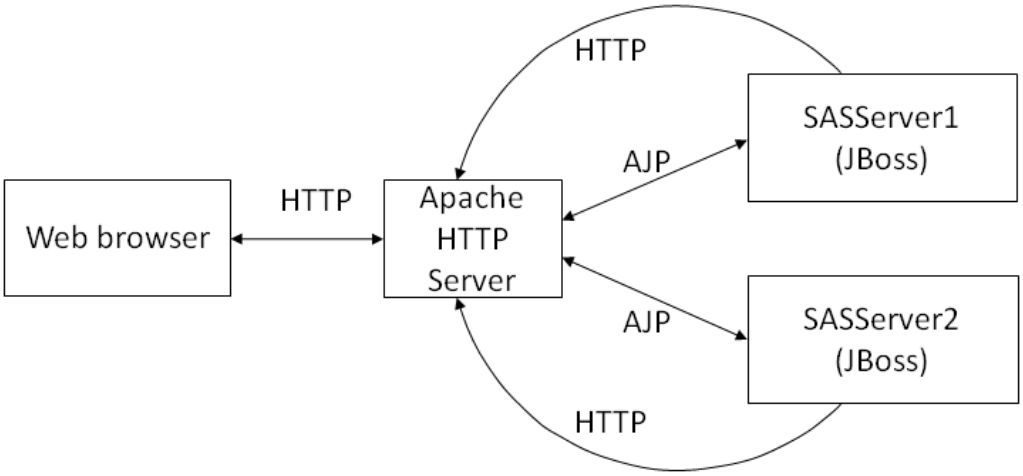
2. Open http://httpserver.example.com/SASWebReportStudio (or another SAS Web application that is deployed on SASServer2).

   Provide credentials at the log on challenge and click **Log On**. Confirm that *JBOSS_HOME*`/server/SASServer2/log/localhost_access_log.`*YYYY-MM-DD*`.log` shows activity like the following example:

   ```
   10.0.0.10 - - [11/Mar/2009:16:47:02 -0400] "GET /SASWebReportStudio/
   scripts/dojo/src/a11y.js HTTP/1.1" 304 -
   10.0.0.10 - - [11/Mar/2009:16:47:02 -0400] "GET /SASWebReportStudio/
   scripts/dojo/src/uri/__package__.js HTTP/1.1" 304 -
   ```

Configuring Additional Security

The information in this section and the remainder of this document builds upon the configuration steps that have already been described, and describes additional configuration tasks that you might choose to perform if your security policies require that Web application servers disable HTTP access. The steps in this section describe how to disable HTTP access to the JBoss servers. This action forces all communication with the SAS Web applications to be performed through the reverse proxy HTTP server. The protocol topology is shown in the following figure.



Reconfiguring the SAS Web applications and JBoss servers to reflect the protocol usage shown in the figure requires the following high-level steps:

- Add entries in the *APACHE_HOME*/conf/uriworkermap.properties file for the remaining SAS Web applications.
- Disable the HTTP connector for each JBoss server.
- Modify the JBoss server startup script to include the host and port access information for the SAS Web Infrastructure Platform.
- Modify SAS metadata to reference the Apache HTTP Server as the connection point for the remaining SAS Web applications, and as the WebDAV location for SAS Content Server.

Reconfigure the uriworkermap.properties File

Edit the *APACHE_HOME*/conf/uriworkermap.properties file with the access point information for the remaining SAS Web applications. Restart Apache HTTP Server after making the changes; see the following example (changes are shown in bold):

```
/SASBIDashboard=sasserver1
/SASBIDashboard/*=sasserver1
/SASBIWS=sasserver1
/SASBIWS/*=sasserver1
/SASPortal/*=sasserver1
/SASPortal=sasserver1
/SASLogon=sasserver1
/SASLogon/*=sasserver1
/SASPackageViewer=sasserver1
/SASPackageViewer/*=sasserver1
/SASPreferences=sasserver1
/SASPreferences/*=sasserver1
/SASSharedApps=sasserver1
/SASSharedApps/*=sasserver1
/SASStoredProcess=sasserver1
/SASStoredProcess/*=sasserver1
/SASAdmin=sasserver1
/SASAdmin/*=sasserver1
/SASContentServer=sasserver1
/SASContentServer/*=sasserver1
/sasweb=sasserver1
/sasweb/*=sasserver1
```

```
/SASSharedPortlets=sasserver1
/SASSharedPortlets/*=sasserver1
/SASSharedApps=sasserver1
/SASSharedApps/*=sasserver1
/SASWIPSoapServices=sasserver1
/SASWIPSoapServices/*=sasserver1
/SASWIPClientAccess=sasserver1
/SASWIPClientAccess/*=sasserver1
/SASWIPServices=sasserver1
/SASWIPServices/*=sasserver1
/SASBIDashboardEventGen=sasserver1
/SASBIDashboardEventGen/*=sasserver1

/SASWebOLAPViewer=sasserver2
/SASWebOLAPViewer/*=sasserver2
/SASWebReportStudio=sasserver2
/SASWebReportStudio/*=sasserver2
/SASWebDoc=sasserver2
/SASWebDoc/*=sasserver2
```

Disable HTTP and Configure Startup Scripts for JBoss

To ensure that access to the SAS Web applications is only through the reverse proxy HTTP server, you disable the HTTP connector for each JBoss server so that they no longer listen on the HTTP port (8080, by default) for HTTP requests. Then, because HTTP is disabled, you must reconfigure the startup scripts so that the SAS Web Infrastructure Applications direct their communication requests to the Apache HTTP Server, which then get sent to the correct JBoss server. Follow these steps:

1. Edit *JBOSS_HOME*/server/SASServer*n*/deploy/jboss-web.deployer/server.xml and disable the HTTP connector by commenting it out:

   ```
   <!--
     <Connector acceptCount="100" address="${jboss.bind.address}"
       connectionTimeout="20000" disableUploadTimeout="true"
       emptySessionPath="true" enableLookups="false"
       maxHttpHeaderSize="8192" maxThreads="250" port="8080"
       protocol="HTTP/1.1" redirectPort="8443"
       useBodyEncodingForURI="true"/>
   -->
   ```

2. Change the startup scripts for each of the servers by editing either the SASServer*n*.bat file or the wrapper.conf file. Change the following options to reference the Apache HTTP Server host name and port:

   ```
   -Dsas.auto.publish.port=80
   -Dsas.auto.publish.host=httpserver.example.com
   ```

Modify SAS Metadata for SAS Web Application and WebDAV Connections

Now that all communication amongst the SAS Web applications must be directed through the Apache HTTP Server, the connection information for the remaining SAS Web applications must be reconfigured from the JBoss server to the Apache HTTP Server. In addition, access to the SAS Content Server is now through the Apache HTTP Server, so the metadata for the server and the WebDAV access information must be reconfigured as well. Follow these steps:

1. For the remaining SAS Web applications, reconfigure the connection information as was done in "Change the Connections for the SAS Web Applications."
2. In SAS Management Console, select **Environment Management** > **Server Manager** > **SAS Content Server**.
3. Right-click the **Connection** icon in the right pane and select **Properties**.
4. Select the **Options** tab and set the **Host name** and **Port number** values to the Apache HTTP Server host name and port number. Click **OK**.
5. In SAS Management Console, select the **Folders** tab.
6. Right-click the **SAS Folders** icon at the root of the folder tree in the left pane and select **Properties**.
7. Select the **Content Mapping** tab and use the Server menu to select **SAS Content Server**. The **URL** field then shows the Apache HTTP Server host name and port. Click **OK**.

Change the WebDAV Repository URL

There are five applications that use SAS metadata to identify the connection information for the SAS Content Server. These applications are identified in the following list:

- Remote Services
- SASBIPortlets4.2 Local Services
- SASJSR168RemotePortlet4.2 Local Services
- SASPackageViewer4.2 Local Services
- SASPortal4.2 Local Services
- SASStoredProcess9.2 Local Services
- SASWebReportStudio4.2 Local Services

To reconfigure the WebDAV URL for the applications, perform the following steps in SAS Management Console:

1. Select **Environment Management** > **Foundation Services Manager**.
2. Select the application and then select **Core** > **Information Service**.
3. Right-click **Information Service** and select **Properties**.
4. On the **Information Service Properties** dialog box, click the **Service Configuration** tab and then click **Configuration**.
5. On the **Information Service Configuration** dialog box, click the **Repositories** tab.
6. Select **WebDAV** and then click **Edit**.

7. Change the **Host** and **Port** values to the host name and port of the Apache HTTP Server.
8. Click **OK** to close the **Information Service Configuration** dialog box.
9. Click **OK** to close the **Information Service Properties** dialog box.

Verify the Configuration

Restart the applications in the same order as before:

1. SAS Remote Services
2. Apache HTTP Server
3. SASServer1
4. SASServer2

Recommended Reading

Community authored, 2009. "Using Mod_jk1.2 With JBoss." Available at http://www.jboss.org/community/docs/DOC-12525.

SAS and all other SAS Institute product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. Other brand and product names are registered trademarks or trademarks of their respective companies.
® indicates USA registration.
Copyright © 2011 SAS Institute Inc., Cary, NC, USA. All rights reserved.