Installation

{scrollbar}

This article provide details about the Apache Geronimo installation. Here you will find information about what are the prerequisite software, where to download Geronimo from and how to customize the installation to use custom ports other than the defaults.

In this document you will also find details about the installation and configuration of different topologies such as 2-tier with remote Web server.

This article is organized in the following sections:

20pxdisc

Prerequiste software

Apache Geronimo v2.0 builds on J2SE 1.5 using Maven 2, get the appropriate JVM versions from the following sites.

- J2SE SDK (http://java.sun.com/)
- Apache Maven (http://maven.apache.org/)

Downloading Geronimo

The source code and binaries for Geronimo v2.0 can be found at the following URL
http://geronimo.apache.org/downloads.html

Building from source

From a command line console uncompress the source code (.zip or tar.gz) and change directory **<geronimo_home>**.
Type the following command to build Apache Geronimo:

```
mvn install
```

Detailed steps and requirements to build Geronimo from the source are covered in deep in the Building Apache Geronimo with Maven 2 section.

Installing Geronimo from binaries

Depending on the platform you plan to install and run Apache Geronimo download the appropriate insallation image. Select the appropriate file compression format for your operating system (.zip, .tar.gz) by clicking directly on the link, download it and expand the binary to your hard drive in a new directory.

The installation of Apache Geronimo is as simple as uncompressing the .zip or .tar files. From a command line window change directory to <geronimo_home>/bin and start the server using the following command:

```
geronimo run
```

For your convenience, Apache Geronimo provides a serie of scripts to manage the server and applications, visit the Tools and commands section for further details.

Changing default ports

The HTTP port is the standard network that the Web container uses. This port number has to be used on any HTTP URL that calls Web applications running on Geronimo. The server and startup configuration modules of Geronimo are controlled by the **`config.xml`** file with the default HTTP listener configured on port 8080.

There are several reasons for changing the network ports, namely to run multiple instances of Geronimo. To configure the default port edit the `<geronimo_home>/var/config/config.xml` file. The out-of-the-box `config.xml` for Apache Geronimo v2.0 with Tomcat distribution is shown in the following example.

xmlsolidconfig.xml <?xml version="1.0" encoding="UTF-8"?> <!-- == --> <!-- Warning - This XML file is re-generated by Geronimo when --> <!-- changes are made to Geronimo's configuration, therefore --> <!-- any comments added to this file will be lost. --> <!-- Do not edit this file while Geronimo is running. --> <!-- == --> <attributes xmlns="http://geronimo.apache.org/xml/ns/attributes-1.1"> <module name="org.apache.geronimo.configs/rmi-naming/2.0-M2/car"> <gbean name="RMIRegistry"> <attribute name="port">1099</attribute> </gbean> <gbean name="NamingProperties"> <attribute name="namingProviderUrl">rmi://0.0.0.0:1099</attribute> </gbean> <gbean name="DownloadedPluginRepos"> <attribute name="repositoryList">http://geronimo.apache.org/plugins/plugin-repository-list-2.0.txt</attribute> <attribute name="userRepositories">[]</attribute> </gbean> </module> <module name="org.apache.geronimo.configs/j2ee-server/2.0-M2/car"/> <module name="org.apache.geronimo.configs/transaction-jta11/2.0-M2/car"/> <module name="org.apache.geronimo.configs/j2ee-security/2.0-M2/car"> <gbean name="JaasLoginServiceRemotingServer"> <attribute name="host">0.0.0.0</attribute> <attribute name="port">4242</attribute> </gbean> <gbean name="JMXService"> <attribute name="protocol">rmi</attribute> <attribute name="host">0.0.0.0</attribute> <attribute name="port">9999</attribute> <attribute name="urlPath">/jndi/rmi://0.0.0.0:1099/JMXConnector</attribute> </gbean> </module> <module name="org.apache.geronimo.configs/axis/2.0-M2/car"/>

<module load="false" name="org.apache.geronimo.configs/axis2/2.0-M2/car"/> <module load="false" name="org.apache.geronimo.configs/cxf/2.0-M2/car" /> <module name="org.apache.geronimo.configs/openejb/2.0-M2/car"/> <module load="false" name="org.apache.geronimo.configs/j2ee-corba-yoko/2.0-M2/car"> <gbean name="NameServer"> <attribute name="port">1050</attribute> <attribute name="host">localhost</attribute> </gbean> </module> <module load="false" name="org.apache.geronimo.configs/j2ee-corba-sun/2.0-M2/car"> <gbean name="NameServer"> <attribute name="port">1050< /attribute> <attribute name="host">localhost</attribute> </gbean> </module> <module name="org.apache.geronimo.configs/system-database/2.0-M2/car" > <gbean name="DerbyNetwork"> <attribute name="host">0.0.0.0</attribute> <attribute name="port">1527</attribute> </gbean> </module> <module name="org.apache.geronimo.configs/activemq-broker/2.0-M2/car"> <gbean name="ActiveMQ.tcp.default"> <attribute name="host">0.0.0.0</attribute> <attribute name="port">61616</attribute> </gbean> </module> <module name="org.apache.geronimo.configs/activemq/2.0-M2/car"/> <module name=" org.apache.geronimo.configs/tomcat6/2.0-M2/car"> <gbean name="TomcatResources"/> <gbean name="TomcatWebConnector"> <attribute name="host" >0.0.0.0</attribute> <attribute name="port">8080</attribute> <attribute name="redirectPort">8443</attribute> </gbean> <gbean name=" TomcatAJPConnector"> <attribute name="host">0.0.0.0</attribute> <attribute name="port">8009</attribute> <attribute name="redirectPort">8443< /attribute> </gbean> <gbean name="TomcatWebSSLConnector"> <attribute name="host">0.0.0.0</attribute> <attribute name="port">8443</attribute> < /gbean> <gbean name="org.apache.geronimo.configs/tomcat6/2.0-M2/car?ServiceModule=org.apache.geronimo.configs/tomcat6/2.0-M2/car, j2eeType=GBean,name=TomcatWebContainer"> <attribute name="catalinaHome">var/catalina</attribute> </gbean> </module> <module name="org. apache.geronimo.configs/geronimo-gbean-deployer/2.0-M2/car"/> <module name="org.apache.geronimo.configs/j2ee-deployer/2.0-M2/car"> <gbean name="WebBuilder"> <attribute name="defaultNamespace">http://geronimo.apache.org/xml/ns/j2ee/web/tomcat-1.2</attribute> </gbean> <gbean name=" EnvironmentEntryBuilder"> <attribute name="eeNamespaces">http://java.sun.com/xml/ns/j2ee,http://java.sun.com/xml/ns/javaee</attribute> </gbean> < /module> <module name="org.apache.geronimo.configs/connector-deployer/2.0-M2/car"> <gbean name="ResourceRefBuilder"> <attribute name=" eeNamespaces">http://java.sun.com/xml/ns/j2ee,http://java.sun.com/xml/ns/javaee</attribute> </gbean> <gbean name="AdminObjectRefBuilder"> <attribute name="eeNamespaces">http://java.sun.com/xml/ns/j2ee,http://java.sun.com/xml/ns/javaee</attribute> </gbean> </module> <module name="org. apache.geronimo.configs/persistence-jpa10-deployer/2.0-M2/car"/> <module name="org.apache.geronimo.configs/openejb-deployer/2.0-M2/car"> <gbean name="EJBBuilder"> <reference name="ServiceBuilders"> <pattern> <name>GBeanBuilder</name> </pattern> <pattern> <name>PersistenceUnitBuilder< /name> </pattern> </reference> <reference name="WebServiceBuilder"> <pattern> <name>CXFBuilder</name> </pattern> <pattern> <name>Axis2Builder</name> </pattern> <pattern> <name>WebServiceBuilder</name> </pattern> </reference> </gbean> <gbean name="EjbRefBuilder" > <attribute name="eeNamespaces">http://java.sun.com/xml/ns/j2ee,http://java.sun.com/xml/ns/javaee</attribute> </gbean> <gbean name=" ClientEjbRefBuilder"> <attribute name="eeNamespaces">http://java.sun.com/xml/ns/j2ee,http://java.sun.com/xml/ns/javaee</attribute> </gbean> < /module> <module name="org.apache.geronimo.configs/client-deployer/2.0-M2/car"/> <module name="org.apache.geronimo.configs/cxf-deployer/2.0-M2 /car"/> <module name="org.apache.geronimo.configs/axis2-deployer/2.0-M2/car"/> <module name="org.apache.geronimo.configs/axis-deployer/2.0-M2 /car"> <gbean name="AxisServiceRefBuilder"> <attribute name="eeNamespaces">http://java.sun.com/xml/ns/j2ee,http://java.sun.com/xml/ns/javaee< /attribute> </gbean> </module> <module name="org.apache.geronimo.configs/javamail/2.0-M2/car"> <gbean name="SMTPTransport"> <attribute name=" host">localhost</attribute> <attribute name="port">25</attribute> </gbean> </module> <module name="org.apache.geronimo.configs/sharedlib/2.0-M2 /car"> <gbean name="SharedLib"> <attribute name="classesDirs">var/shared/classes</attribute> <attribute name="libDirs">var/shared/lib</attribute> < /gbean> </module> <module name="org.apache.geronimo.configs/tomcat6-deployer/2.0-M2/car"/> <module name="org.apache.geronimo.configs /welcome-tomcat/2.0-M2/car"/> <module name="org.apache.geronimo.configs/webconsole-tomcat/2.0-M2/car"/> <module load="false" name="org.apache. geronimo.configs/uddi-tomcat/2.0-M2/car"/> <module name="org.apache.geronimo.configs/remote-deploy-tomcat/2.0-M2/car"/> <module name="org. apache.geronimo.configs/hot-deployer/2.0-M2/car"/> <module load="false" name="org.apache.geronimo.configs/ca-helper-tomcat/2.0-M2/car"/> <module name="console.dbpool/LocalDB/1.0/rar"/> <module name="console.dbpool/jdbc%2FTradeDataSource/1.0/rar"/> </attributes>

Here are two excerpts from the `config.xml` file, one for Tomcat and one for Jetty. These excerpts show the entire section where the listeners are defined, we have just replaced the standard **8080** with an arbitrary port, **9000** in this case.

xmlsolidExcerpt from config.xml - Tomcat <module name="org.apache.geronimo.configs/tomcat6/2.0-M2/car"> <gbean name="TomcatResources"/> <gbean name="TomcatWebConnector"> <attribute name="host">0.0.0.0</attribute> <attribute name="port">9000</attribute> <attribute name=" redirectPort">8443</attribute> </gbean> <gbean name="TomcatAJPConnector"> <attribute name="host">0.0.0.0</attribute> <attribute name="port" >8009</attribute> <attribute name="redirectPort">8443</attribute> </gbean> <gbean name="TomcatWebSSLConnector"> <attribute name="host">0.0.0.0 </attribute> <attribute name="port">8443</attribute> </gbean> <gbean name="org.apache.geronimo.configs/tomcat6/2.0-M2/car?ServiceModule=org. apache.geronimo.configs/tomcat6/2.0-M2/car,j2eeType=GBean,name=TomcatWebContainer"> <attribute name="catalinaHome">var/catalina</attribute> < /gbean> </module> xmlsolidExcerpt from config.xml - Jetty <module name="org.apache.geronimo.configs/jetty6/2.0-M2/car"> <gbean name=" JettyWebConnector"> <attribute name="host">0.0.0.0</attribute> <attribute name="port">9000</attribute> <attribute name="redirectPort">8443< /attribute> </gbean> <gbean name="JettySSLConnector"> <attribute name="host">0.0.0.0</attribute> <attribute name="port">8443</attribute> </gbean> < /module> <module name="org.apache.geronimo.configs/geronimo-gbean-deployer/2.0-M2/car"/> <module name="org.apache.geronimo.configs/j2ee-deployer/2.0-M2/car"> <gbean name="WebBuilder"> <attribute name="defaultNamespace">http://geronimo.apache.org/xml/ns/j2ee/web/jetty-1.2< /attribute> </gbean> <gbean name="EnvironmentEntryBuilder"> <attribute name="eeNamespaces">http://java.sun.com/xml/ns/j2ee,http://java.sun.com/xml /ns/javaee</attribute> </gbean> </module>

Save the file and restart the server, with the new configuration loaded point your web browser to the new port.

http://localhost:9000/console

Changing the username and password

Apache Geronimo installs with a the default username - **system** and password - **manager**. There are three options to change the default username and password.

1. Editing the *.properties files manually.
2. Modifying the security configuration from the Administration Console.
3. Creating a new security realm.

This last option is covered more in detail in the Configuring security section.

Editing the *.properties files manually

Open the **`<geronimo_home>/var/security/groups.properties`** file and edit the contents of this file. Add any username that you require and save the file.

solidgroups.properties admin=user1,user2

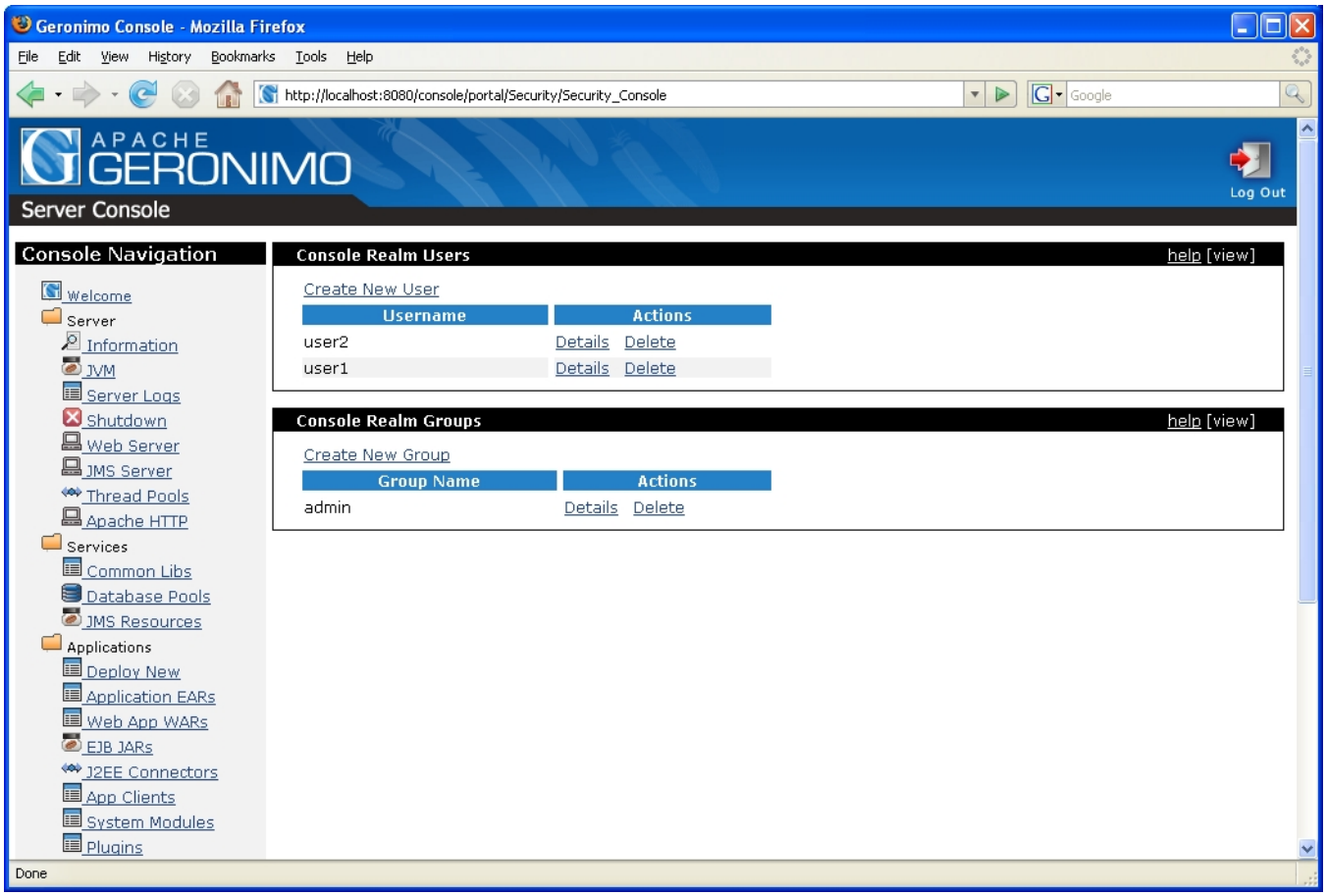
Next, open the **<geronimo_home>/var/security/users.properties** file. You can change the password for the existing system account and/or add new users. When adding new users, the username has to be the same as the ones added to the groups.properties file.

solidusers.properties user1=password1 user2=password2

In this example, two users have been added, **user1** and **user2** and the default system account has been deleted. Both user1 and user2 can access the console and the command line deployment tools.

Modifying the security configuration from the Administration Console

Log into the console and click on **Console Realm** on the console navigation panel. This will display the **Console Realm Users** and **Console Realm Groups** portlets.



- Click on the **Create New User** link to add a new user and on **Details** to edit existing users.
- Click on **Create New Group** link to add new user to the group.
- Once the new user name and password is added, log out of the console and try the new user name and password.

Configuring security in Apache Geronimo v2.0 is covered with more details in the Configuring security section.

Topologies

In today's globalized world, modern organizations face a lot of opportunities and challenges every day. Many of these challenges can put an onus on the organization's IT infrastructure therefore its configuration is crucial. The Apache Geronimo application server supports small-to-medium-sized enterprise applications and provides robust, secure support for the latest J2EE specification. This section will highlight the different configuration options that can be used in your production environment.

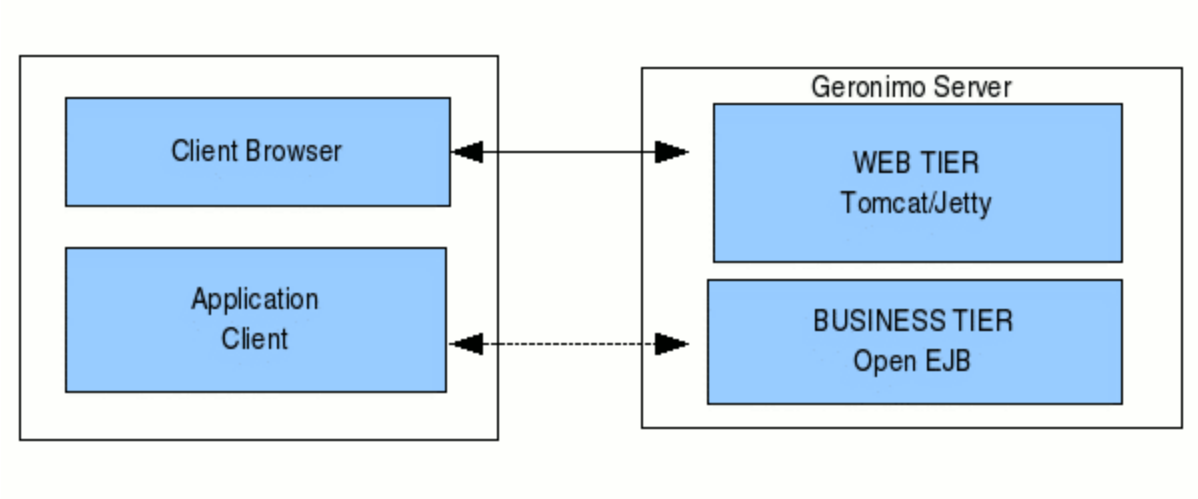
Two-tier system

The figure below illustrates a client machine accessing an application hosted in the Geronimo server. Althougth the figure depicts only one client machine, several machines can be connected to the server and users on these machines can access the hosted Web applications using a standard Web browser.

The client-side application may vary ranging from being a simple command line user interface to a full-fledged user interface such as those created using popular client-side GUI technology. These applications can access the Web tier by connecting to the server using their own HTTP connections, or they can access business and EIS tier objects through the help of the Geronimo client application container.

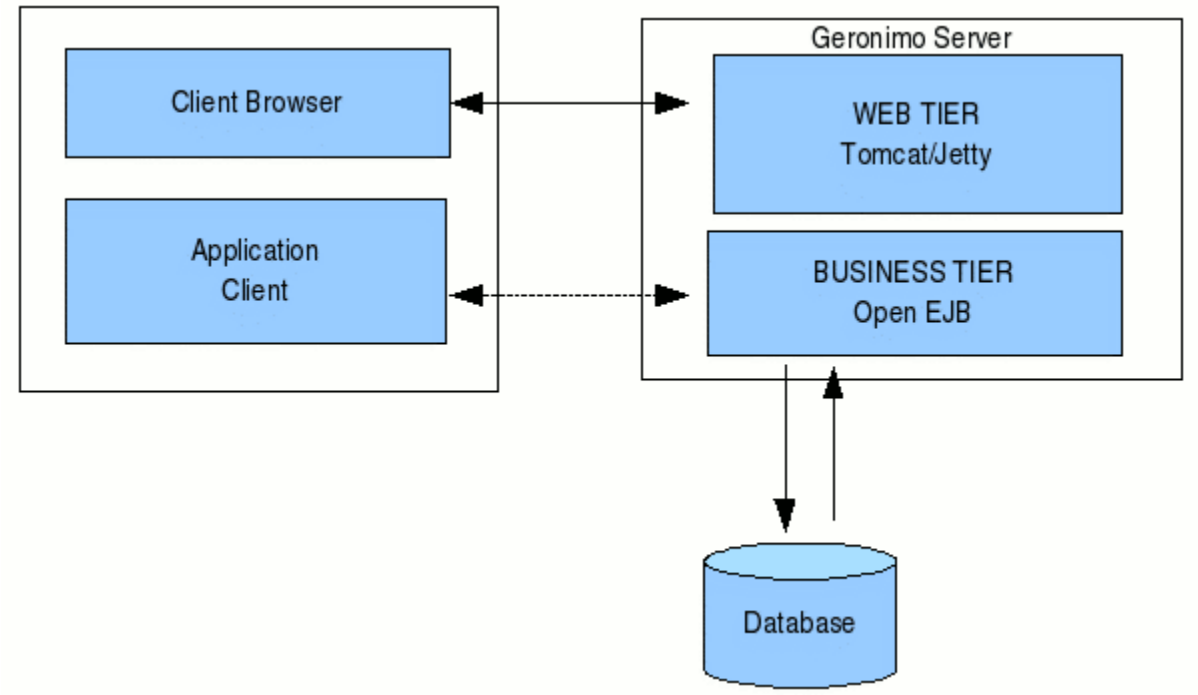
Features of the Geronimo application client:

- Seperate from the Geronimo server.
- Communication with the Geronimo server is over the network.
- Provides mapping dependency management for the client application and reference resolution.



Three-tier system

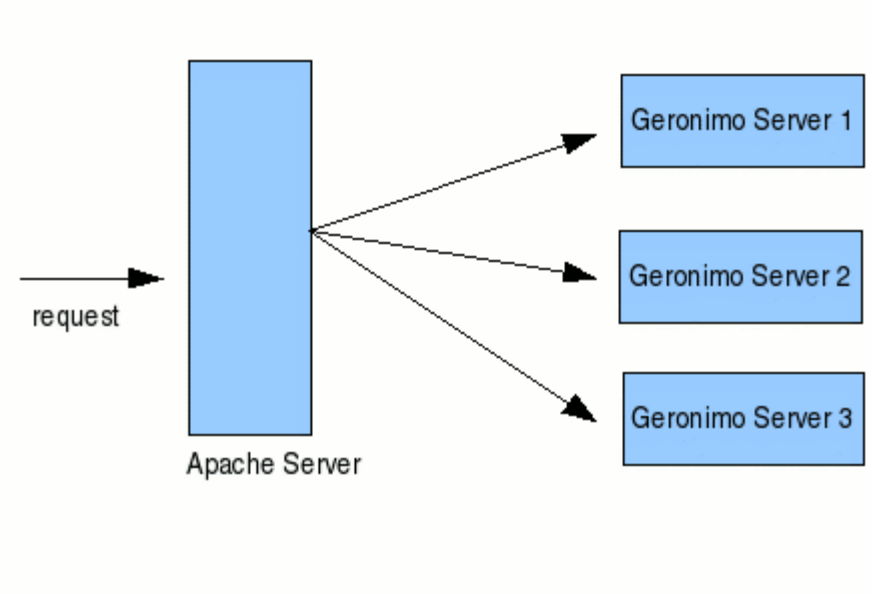
Three tier architecture system is more scalable than two tier as it supports hundreds of users and organizations. It also increases flexibility and freedom.



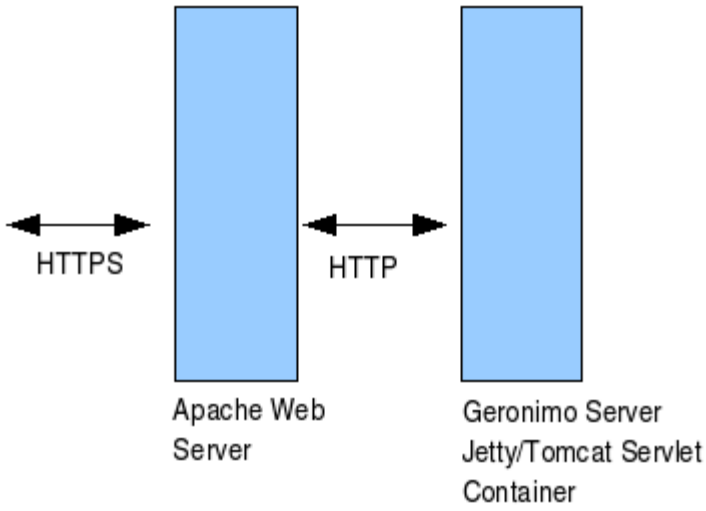
Configuring with Apache HTTPd

The Apache Web server is the best, and most popular, HTTP server software in use on the Internet today. In your production enviroment using Geronimo with the Apache Web server would give you some rigorous advantages as given below.

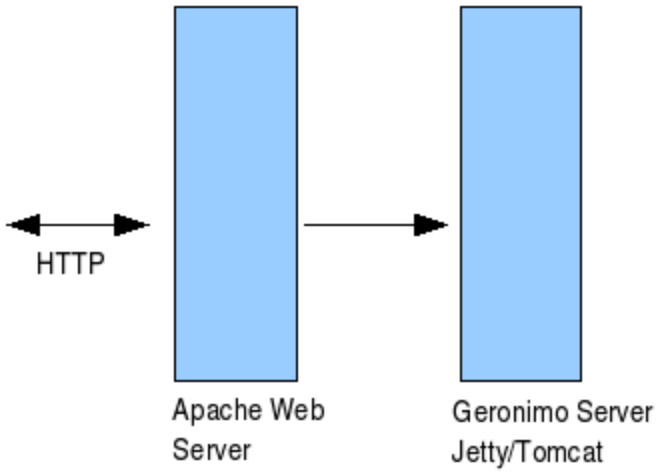
1. Clustering & Load Balancing
   Clustering allows an application server to support multiple nodes with failover, session data sharing, and load balancing across many network nodes. Load balancing of the application can be done with the help of inbuilt support of Apache server.



2. SSL Encoder
   The advantages of Web server proxies are that they offer a way to get server affinity for SSL-encrypted messages, without any extra hardware. But extensive SSL processing puts an extra load on the proxy. It will be handled by Apache server while Geronimo can handle business logic of the application.



3. Application Load Sharing
   In many serving systems, web and application servers work together to handle all HTTP requests. Apache handles the request for static pages (including HTML, JPEG, and GIF files), while Geronimo handles requests for dynamic pages (JSPs or servlets) with the help of Tomcat/Jetty. Geronimo servers can also handle static pages, but in combined systems, they are usually configured to handle dynamic requests.



Summary

Using the Geronimo application server lets you build a secure enterprise infrastructure conforming to the latest standards and requirements. This article gave you an insight into custom configuration of the standard Geronimo installation, with several installation methods.