

# D. Configurando Hosts Virtuales en Geronimo-Tomcat

{scrollbar}

top

This article shows how to configure Virtual Hosts in Apache Geronimo with Tomcat. By default, when you deploy and start an application in Geronimo, that application will be listening on every available host name. By configuring a virtual host you can make an application to listen on a specific host name or IP. The configuration steps described in this article are also valid then you are sharing a single IP among several host names.

To configure a virtual host in Geronimo you basically need to:

- [#Configure local host or DNS](#)
- [#Define Virtual host](#)
- [#Declare the virtual host in the deployment plan](#)
- [#Deploy the application](#)

This article used the simple HelloWorld application as a reference, this application is covered in the [Quick start - Apache Geronimo for the impatient](#) section.

## Configure local host or DNS

To make this configuration work you need to ensure that Geronimo can resolve the virtual host name you are about to define. Depending on your network configuration you can add an entry to your DNS, an alias to the Geronimo server IP. Alternatively you can add an entry to the local host table, each operating system has its own way to define a local host table. For example Windows will have %SystemRoot%\system32\drivers\etc\hosts, UNIX based operating systems would normally have an /etc/hosts.

For this example we will be using the following host names defined in the local host table:

```
127.0.0.1 localhost virtualhost1.com virtualhost2.com virtualhost3.com virtualhost4.com
```

Make sure your system can resolve these names.

## Define Virtual host

Now you need to define those virtual hosts in Geronimo's `config.xml` so it can recognize them. This section provides two different virtual hosts definitions, that is creating two new **HostGBean** (TomcatVirtualHost1 and TomcatVirtualHost2) in the Geronimo configuration, one of those will have multiple host names aliases. The goal of this example is to have an application listening on a single virtual host (this will be virtualhost1.com) and another application listening on a different virtual host (this will be virtualhost2.com) with two additional aliases (this will be virtualhost3.com and virtualhost4.com).

Open the `config.xml` file located in the `<geronimo_home>/var` directory and look for the following line `<module name="geronimo/tomcat/1.1/car">`. This is the beginning of the Tomcat configuration module, all the additional virtual host configuration will be done immediately after this line.

To define the first HostGBean `TomcatVirtualHost_1` add the following lines right after `<module name="geronimo/tomcat/1.1/car">`.

```
xmlsolidExcerpt from config.xml ... <gbean gbeanInfo="org.apache.geronimo.tomcat.HostGBean" name="geronimo/tomcat/1.1/car?
ServiceModule=geronimo/tomcat/1.1/car,j2eeType=Host,name=TomcatVirtualHost_1"> <attribute name="className">org.apache.catalina.core.
StandardHost</attribute> <attribute name="initParams">name=virtualhost1.com appBase= workDir=work</attribute> </gbean> ...
```

To add the second HostGBean `TomcatVirtualHost_2` add the following lines right after the first HostGBean. These two HostGBEans have been split so it is easier to identify them. The main difference between these two is the `<attribute name="aliases">...</attribute>` line to define the aliases.

```
xmlsolidExcerpt from config.xml ... <gbean gbeanInfo="org.apache.geronimo.tomcat.HostGBean" name="geronimo/tomcat/1.1/car?
ServiceModule=geronimo/tomcat/1.1/car,j2eeType=Host,name=TomcatVirtualHost_2"> <attribute name="className">org.apache.catalina.core.
StandardHost</attribute> <attribute name="initParams">name=virtualhost2.com appBase= workDir=work</attribute> <attribute name="aliases"
>virtualhost3.com,virtualhost4.com</attribute> </gbean> ...
```

At this point you have successfully configured two different virtual hosts in Geronimo. Make sure you save the changes to the `config.xml` file and **start** Geronimo.

[Back to Top](#)

For additional reference, this is the entire `config.xml` with the two HostGBean already defined.

```
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="geronimo/rmi-naming/1.1/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://people.apache.org/~ammulder/plugin-repository-list-1.1.txt</attribute> <attribute name="userRepositories">[]</attribute> <
/gbean> </module> <module name="geronimo/j2ee-server/1.1/car"> <module name="geronimo/j2ee-security/1.1/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="geronimo/axis/1.1/car"> <module name="
geronimo/openejb/1.1/car"> <gbean name="EJBNetworkService"> <attribute name="host">0.0.0.0</attribute> <attribute name="port">4201</attribute> <
/gbean> </module> <module load="false" name="geronimo/j2ee-corba/1.1/car"> <gbean name="NameServer"> <attribute name="dbDir">var/cosnaming.
db</attribute> <attribute name="port">1050</attribute> </gbean> <gbean name="Server"> <attribute name="args">-ORBInitRef, NameService=corbaloc::
localhost:1050/NameService</attribute> </gbean> <gbean name="UnprotectedServer"> <attribute name="args">-ORBInitRef, NameService=corbaloc::
localhost:1050/NameService</attribute> </gbean> </module> <module name="geronimo/system-database/1.1/car"> <gbean name="DerbyNetwork">
<attribute name="host">0.0.0.0</attribute> <attribute name="port">1527</attribute> </gbean> </module> <module name="geronimo/activemq-broker/1.1
/car"> <gbean name="ActiveMQ.tcp.default"> <attribute name="host">0.0.0.0</attribute> <attribute name="port">61616</attribute> </gbean> </module>
<module name="geronimo/activemq/1.1/car"> <module name="geronimo/tomcat/1.1/car"> <!-- =====
> <!-- First HostGBean TomcatVirtualHost_1 defined --> <!-- ===== --> <gbean gbeanInfo="org.apache.
geronimo.tomcat.HostGBean" name="geronimo/tomcat/1.1/car?ServiceModule=geronimo/tomcat/1.1/car,j2eeType=Host,name=TomcatVirtualHost_1">
<attribute name="className">org.apache.catalina.core.StandardHost</attribute> <attribute name="initParams">name=virtualhost1.com appBase=
workDir=work</attribute> </gbean> <!-- ===== --> <!-- Second HostGBean TomcatVirtualHost_1
defined --> <!-- ===== --> <gbean gbeanInfo="org.apache.geronimo.tomcat.HostGBean" name="
geronimo/tomcat/1.1/car?ServiceModule=geronimo/tomcat/1.1/car,j2eeType=Host,name=TomcatVirtualHost_2"> <attribute name="className">org.
apache.catalina.core.StandardHost</attribute> <attribute name="initParams">name=virtualhost2.com appBase= workDir=work</attribute> <attribute
name="aliases">virtualhost3.com,virtualhost4.com</attribute> </gbean> <!-- ===== --> <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="geronimo/tomcat/1.1/car?ServiceModule=geronimo/tomcat/1.1
/car,j2eeType=GBean,name=TomcatWebContainer"> <attribute name="catalinaHome">var/catalina</attribute> </gbean> </module> <module name="
geronimo/geronimo-gbean-deployer/1.1/car"> <module name="geronimo/j2ee-deployer/1.1/car"> <gbean name="WebBuilder"> <attribute name="
defaultNamespace">http://geronimo.apache.org/xml/ns/j2ee/web/tomcat-1.1</attribute> </gbean> </module> <module name="geronimo/openejb-deployer
/1.1/car"> <gbean name="EJBBuilder"> <attribute name="listener">?name=TomcatWebContainer</attribute> </gbean> </module> <module name="
geronimo/client-deployer/1.1/car"> <module name="geronimo/axis-deployer/1.1/car"> <module load="false" name="geronimo/javamail/1.1/car"> <gbean
name="SMTPTransport"> <attribute name="host">localhost</attribute> <attribute name="port">25</attribute> </gbean> </module> <module name="
geronimo/sharedlib/1.1/car"> <gbean name="SharedLib"> <attribute name="classesDirs">var/shared/classes</attribute> <attribute name="libDirs">var
/shared/lib</attribute> </gbean> </module> <module name="geronimo/tomcat-deployer/1.1/car"> <module name="geronimo/welcome-tomcat/1.1/car">
<module name="geronimo/webconsole-tomcat/1.1/car"> <module load="false" name="geronimo/uddi-tomcat/1.1/car"> <module name="geronimo/remote-
deploy-tomcat/1.1/car"> <module name="geronimo/hot-deployer/1.1/car"> <module load="false" name="sample.applications/HelloWorldApp/1.1/war"> <
/attributes>

```

[Back to Top](#)

## Declare the virtual host in the deployment plan

As mentioned before, for this example we are using the HelloWorld sample application covered in the [Quick start - Apache Geronimo for the impatient](#) section. Since this is a Web application, the deployment plan that we are modifying is the **geronimo-web.xml**. If you use a different type of application you may need to modify a different deployment plan, for example **geronimo-application.xml**. Refer to [Deployment plans](#) for further details.

In the previous section we defined two Virtual Hosts, now we will be configuring two applications so they can be deployed to those Virtual Hosts exclusively.

We will use the same application in both cases but will differentiate one deployment from the other by giving it a different **artifactId**, this way we avoid modifying the code but still can identify each deployment.

Assuming you followed the steps covered in the [Quick start - Apache Geronimo for the impatient](#) section you should have the following structure:

```
solid <app_home>\ + HelloWorld.jsp + WEB-INF\ + geronimo-web.xml + web.xml
```

Open the **geronimo-web.xml** file and edit the **artifactId** and **context-root** to make this deployment unique. Within the **web-app** section add the **host** attribute and specify the Virtual Host you want this application to listen, in this case **virtualhost1.com**.

```

xmldsolidgeronimo-web.xml for HelloWorld_1 <?xml version="1.0" encoding="UTF-8"?> <web-app xmlns="http://geronimo.apache.org/xml/ns/j2ee/web-1.1"
> <environment> <moduleid> <groupid>sample.applications</groupid> <artifactId>HelloWorldApp_1</artifactId> <version>1.1</version> </moduleid> <
/environment> <context-root>/hello_1</context-root> <host>virtualhost1.com</host> </web-app>

```

Save the changed to the **geronimo-web.xml** file and generate a WAR file by typing the following command from the **<app\_home>** directory:

```
jar -cvf HelloWorld_1.war *
```

Once deployed this application should only listed in the **virtualhost1.com** host name.

We will now repeat this steps to create a second WAR. Edit once again the **geronimo-web.xml** file and copy the content form the following example. Note that we are only changing the **artifactId**, **context-root** and **host**.

```

xmldsolidgeronimo-web.xml for HelloWorld_2 <?xml version="1.0" encoding="UTF-8"?> <web-app xmlns="http://geronimo.apache.org/xml/ns/j2ee/web-1.1"
> <environment> <moduleid> <groupid>sample.applications</groupid> <artifactId>HelloWorldApp_2</artifactId> <version>1.1</version> </moduleid> <
/environment> <context-root>/hello_2</context-root> <host>virtualhost2.com</host> </web-app>

```

Save the changed to the **geronimo-web.xml** file and generate a second WAR file by typing the following command from the **<app\_home>** directory:

```
jar -cvf HelloWorld_2.war *
```

You now have two applications ready to be deployed to two different virtual hosts.

[Back to Top](#)

## Deploy the application

At this point you have configured Geronimo to use two different Virtual Hosts, one of them is also configured to listen under additional aliases. All you need to do now is to deploy the applications and test them. To deploy the applications type the following commands from the <geronimo\_home>\bin directory:

```
java -jar deployer.jar --user system --password manager deploy <app_home>\HelloWorld_1.war
```

You should get a successful confirmation message similar to this one:

Note that the deployer tool will not tell the Virtual Host where the application was deployed to, it will only tell the machine's defined host name. If you try to access that URL you should get a **HTTP Status 404 - /hello\_1**, resource not available. That error message is a good sign since we deployed the application to a particular Virtual Host.

Repeat the deployment for the second application.

```
java -jar deployer.jar --user system --password manager deploy <app_home>\HelloWorld_2.war
```

You should get a successful confirmation message similar to this one:

With the applications deployed the only thing left is to test them. Test **hello\_1** first, try to access the hosts names defined on the Geronimo server machine:

Host name / Virtual Host	Access
<a href="http://hcunico:8080/hello_1">http://hcunico:8080/hello_1</a>	Fail
<a href="http://localhost:8080/hello_1">http://localhost:8080/hello_1</a>	Fail
<a href="http://virtualhost1.com:8080/hello_1">http://virtualhost1.com:8080/hello_1</a>	<b>SUCCESS!!!</b>
<a href="http://virtualhost2.com:8080/hello_1">http://virtualhost2.com:8080/hello_1</a>	Fail
<a href="http://virtualhost3.com:8080/hello_1">http://virtualhost3.com:8080/hello_1</a>	Fail
<a href="http://virtualhost4.com:8080/hello_1">http://virtualhost4.com:8080/hello_1</a>	Fail

Now repeat the tests for **hello\_2**.

Host name / Virtual Host	Access
<a href="http://hcunico:8080/hello_2">http://hcunico:8080/hello_2</a>	Fail
<a href="http://localhost:8080/hello_2">http://localhost:8080/hello_2</a>	Fail
<a href="http://virtualhost1.com:8080/hello_2">http://virtualhost1.com:8080/hello_2</a>	Fail
<a href="http://virtualhost2.com:8080/hello_2">http://virtualhost2.com:8080/hello_2</a>	<b>SUCCESS!!!</b>
<a href="http://virtualhost3.com:8080/hello_2">http://virtualhost3.com:8080/hello_2</a>	<b>SUCCESS!!!</b>
<a href="http://virtualhost4.com:8080/hello_2">http://virtualhost4.com:8080/hello_2</a>	<b>SUCCESS!!!</b>

Congratulations!!! you have successfully configured and deployed two applications to two different virtual hosts and aliases.

[Back to Top](#)