Grouper book -Installing an application server and the Grouper UI

Installing and running an application server

If you wish to use the Grouper user interface (UI) you will need to install an application server in which to run it. The application server must support the Java Servlet specification version 2.4 or greater, of which there is a wide choice, including:

- Apache Tomcat
- Jetty
- IBM Websphere
- Oracle app server
- Oracle Glassfish

Grouper does not require any of the services provided by a JEE server beyond those required by the Servlet specification, so it would not be worth procuring an expensive commercial server for this reason. The majority of Grouper UI installation are running on Tomcat, so we will concentrate on this server.

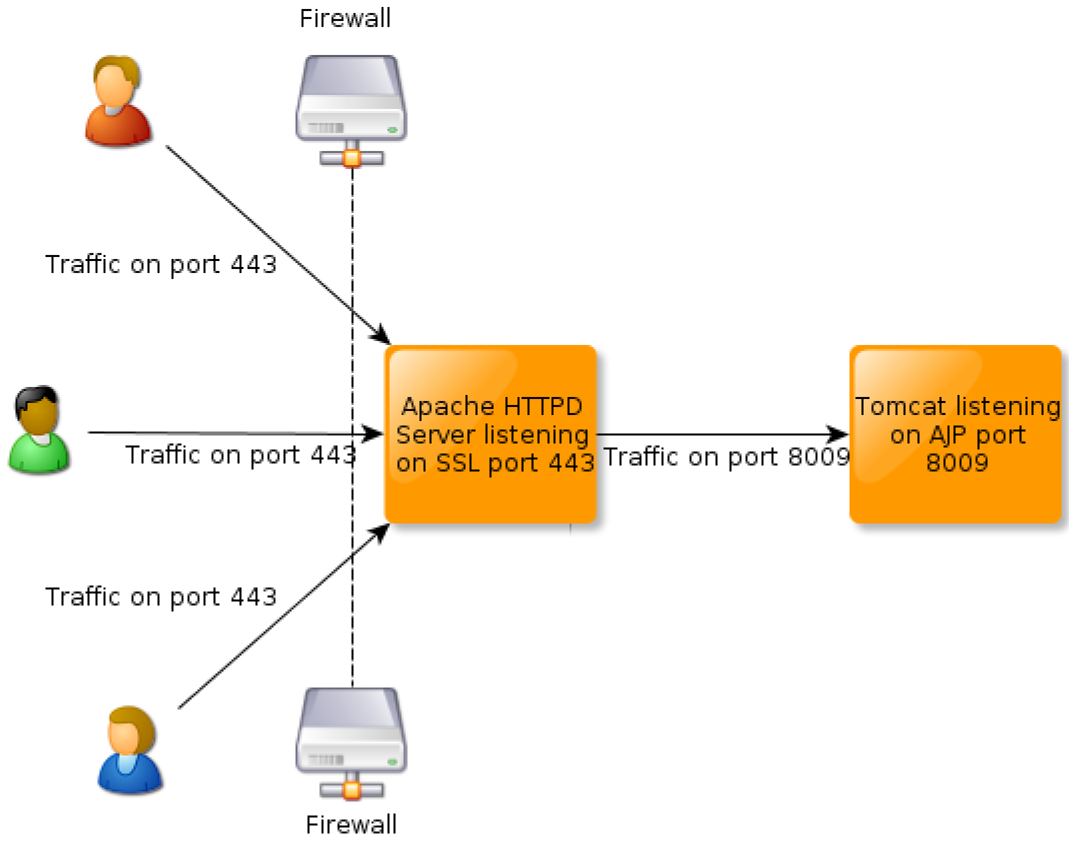
Obtaining, installing and setting up Tomcat

We covered these procedures in the quick start section. Stop before setting up the web application.

Installing Apache HTTP server to sit in front of Tomcat.

It is common (and good practice) to install Apache HTTP server to site between the client browser and the Tomcat application server.

- Client browsers connect to a webserver on port 80 by default (for non-secure connections) and 443 (for default connections). Connecting to an alternative port requires that port to be included in the address in the form http://<server>:<port>/<location>. Typically users prefer not to bother with the port
- Ports below 1024 are typically reserved for privileged users (such as root). Tomcat could listen on Port 80 instead of 8080, but would need to run as root in order to do so. This would mean that all code running inside Tomcat would also be running as root - not a good idea from a security point of view!
- Apache HTTPD server is engineered to initialise as a privileged user, then run all processes as a non-privileged user. It is therefore an ideal candidate for acting as an intermediary between the client browser and tomcat
- Typically, port 80 is allowed through firewalls (perhaps via a proxy server), but port 8080 is not
- It is a good idea to access the Grouper UI through an encrypted connection. Setting Apache up to service an SSL connection is easier than doing so for Tomcat



Obtaining and installing Apache HTTPD server is beyond the scope of this document, and procedures depend on your operating system and distribution. Apache HTTPD 2.2 with the mod_proxy and mod_proxy_ajp modules is recommended, so please consult your system documentation to go about getting it. Once it's up and running, the following configuration snippet will proxy incoming connections to the tomcat ajp connector running on port 8009:

```
ProxyPass /grouper ajp://localhost:8009/grouper/
ProxyPassReverse /grouper ajp://localhost:8009/grouper/
```

For documentation on these apache modules see http://httpd.apache.org/docs/2.2/mod/mod_proxy.html and http://httpd.apache.org/docs/2.2/mod/mod_proxy_ajp.html

It is also possible to proxy to a range of backend servers, balancing connections between members of a cluster. We will use this as one option for setting up a basic cluster for the Grouper UI in a later section. See http://httpd.apache.org/docs/2.2/mod/mod_proxy_balancer.html for more details.

If you are running on a Microsoft server platform then Internet Information Server can perform the same task in a slightly different way. One way of doing this can be found at http://www.iisadmin.co.uk/?p=8

Installing Grouper UI on the application server

The Grouper UI now needs to be built with your grouper configuration files. Download the Grouper UI from http://www.internet2.edu/grouper/software.html at the time of writing the most up-to-date version is found at http://www.internet2.edu/grouper/release/1.6.2/grouper.ui-1.6.2.tar.gz

Unpack the downloaded archive, we will refer to the location on the file system where it is unpacked as GROUPER_UI_SRC from now on. You will also need both Java and Apache ant set up. Please refer to the quick start section for details of how to do this.

Rename the GROUPER_UI_SRC/build.properties.template file to GROUPER_UI_SRC/build.properties and edit it. Change the value of the grouper.folder property so that it points to your GROUPER_HOME directory (the location of the grouper.apiBinary-<VERSION> in which you edited the configuration files and ran gsh during setup). This ensures that the build process has access to the grouper libraries and your configuration files. This will mean that it will connect to the same grouper database and subject source(s) as the grouper Daemon, which is required.
If you wish, copy the file GROUPER_UI_SRC/log4j.properties.template to GROUPER_UI_SRC/log4j.properties and edit it to change the log levels and locations

Open a command line, and from within GROUPER_UI_SRC run this command:

```
ant war
```

This will build an installable web application archive called grouper.war which you will find in GROUPER_UI_SRC/build. Copy this file to TOMCAT_HOME /webapps. You will also want to edit the password for GrouperSystem contained in GROUPER_UI_SRC/tomcat-users.xml and copy the file to TOMCAT_HOME/conf. Restart tomcat. If you have setup Apache HTTPD in from of Tomcat, point your browser to:

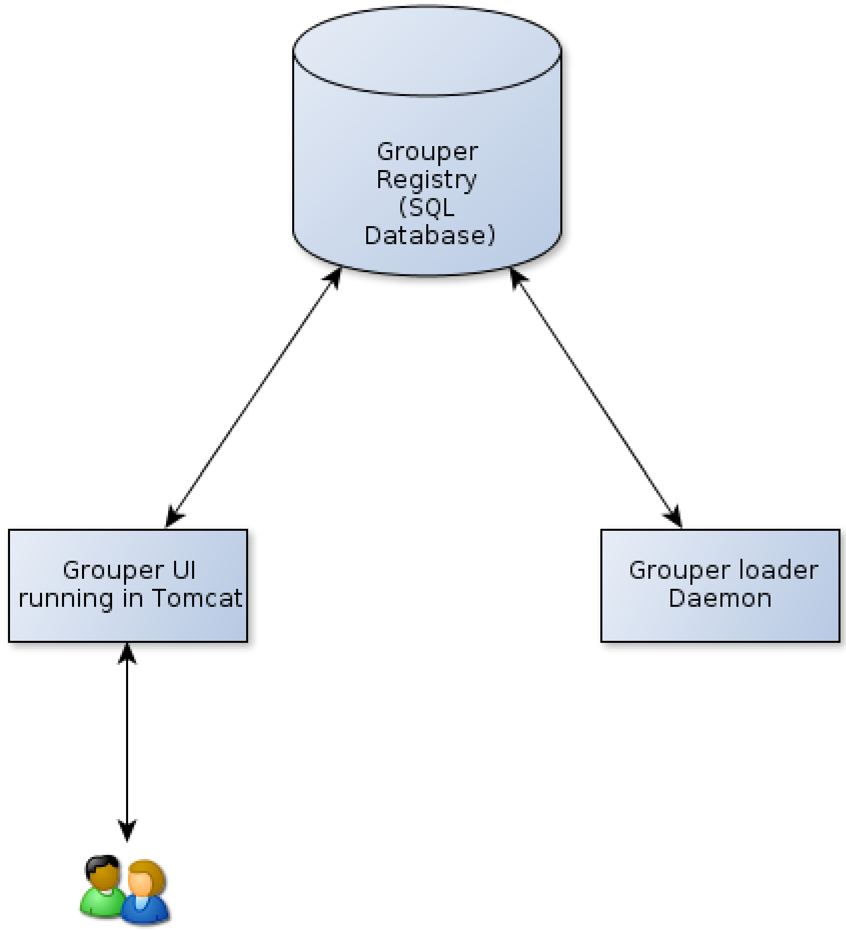
http://<tomcat host>/grouper/

If you are running tomcat without Apache HTTPD point your browser to:

http://<tomcat host>:8080/grouper/

You should show be able to log in as GrouperSystem using the password you set in tomcat-users.xml.

Your Grouper architecture will now look like this, with the Grouper Loader Daemon and Grouper UI existing as standalone applications, both communicating with the same database. For simplicity the subject source has been omitted from the diagram.


Grouper Registry (SQL Database)
Grouper UI running in Tomcat
Grouper loader Daemon