

# Setting up a Load Balancer - mod\_jk2

- download, install and start Apache httpd (beyond the scope of this page! - try Google)
- if it does not include mod\_jk2, do the same for that (beyond the scope of this page! - try Google)
- at the bottom of your e.g. /etc/httpd/conf/http.conf (file location may vary), place e.g. the following:

```
<VirtualHost *:80>
<Location "/wadi">
JkUriSet group cluster
</Location>
</VirtualHost>
```

```
# Load the Tomcat/Apache connector module, if it isn't already
<IfModule !mod_jk2.c>
LoadModule jk2_module modules/mod_jk2.so
</IfModule>
```

- copy \$WADI\_HOME/conf/workers2.properties to e.g. /etc/httpd/conf/ (location may vary)
- apachectl restart
- point your browser at e.g. : <http://localhost:80/jkstatus>

You should see the mod\_jk2 status page, listing all the potential nodes in your cluster along with their AJP ports.

Congratulations - you now have a two-tier deployment - Apache/mod\_jk2 over a Jetty/Tomcat cluster with a distributable webapp 😊

- restart your browser or throw out any cookies returned from your cluster
- hit <http://localhost:80/wadi/jsp/create.jsp> - the page colour informs you as to which node the session was created upon
- hit <http://localhost/wadi/jsp/index.jsp?limit=25&refresh=3>

You will see a page of nine frames

Each frame contains a rendering of your current session

The background colour of the frame informs you of the node rendering the frame

The cells within the frame represent the session itself

Each time you access the session you add a new cell (up to a maximum)

The cell's colour records the node on which the cell was added to the session

You should see the session moving from node to node as requests, landing [pseudo-]randomly within the cluster are rendered in the correct context.

Congratulations - you now have a working cluster 😊 - but no session affinity - an important optimisation which will prevent a lot of session relocation (migration) and request relocation (proxying/redirecting), by trying to consistently deliver requests for the same session to the same node.

- look in WEB-INF/wadi-web.xml to see how to enable the WADI/mod\_jk[2] integration
- restart your cluster and browser - try again

You should now see your session remaining in one place (unless you then take down that node, in which case WADI should find it on another node and restick your client to its new location).

Congratulations - you are now a web clustering expert 😊