

This series is intended to provide documentation of a simple server cluster based on two physical servers and four virtual servers per machine, and configured to host an instance of the Moodle Learning Management System.



### Topics

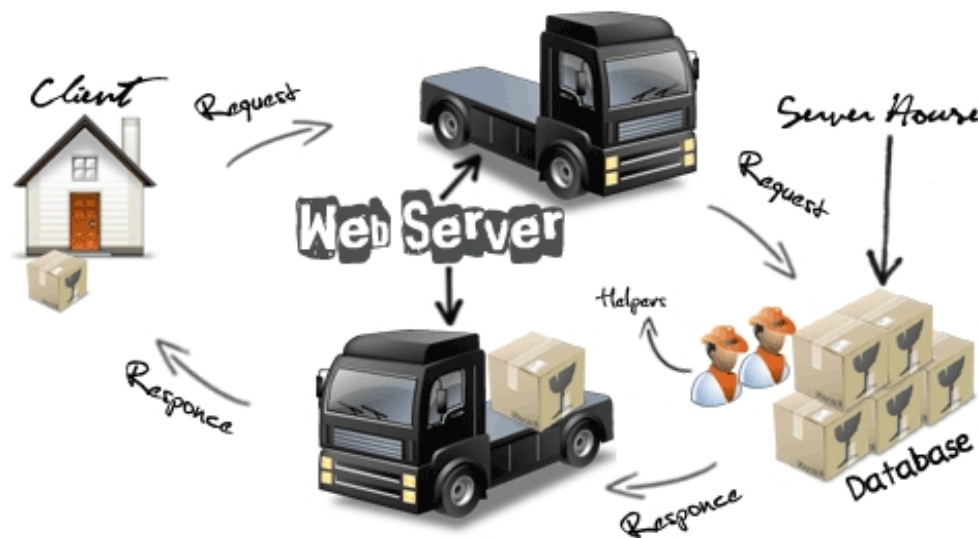
- [Introduction](#)
- [System hardware and Virtual Server environments](#)
- [Load Balancing and failover](#)
- [Web Server, sessions, and updates](#)
- [File server, repositories, synchronisation and network issues](#)
- Database implementation, tuning, and replication
- System monitoring and operations
- System trouble shooting

### Web Server, Sessions & Updates

What is the role of the Web Server? While sounding a slightly pithy questions, it is important to understand what the webserver is required to do.

The Web Server provides the interface between the HTTPD protocol used to communicate with the client browser, and the application code, in Moodle's case, PHP, which accesses the

backend database and filesystem to generate the HTTPD response sent back to the client browser.



So the Web Server itself is performing a Router and Resource Manager role. The server running the Web Server also usually runs the Application code. But not the database queries which generally take up the larger CPU tasks.

So the webserver requires average CPU resources ( maybe 2-4 cores ) and a reasonable amount of RAM (4-8 Gbyte). And in high demand uses, fast network access.

In large clusters you will need more than one web server, running in parallel, to increase the scalability of the system. This is easy to understand, but requires some fundamental issues to be resolved.

### Sessions

To understand how sessions affect your system you need to understand how web applications work. When you use any interactive website, you are communicating with the server in an asynchronous manner. This means that you are making individual connections to the webserver, each time the application needs to update the view in the browser. This may be affected by Javascript running in the browser environment, but generally each time you refresh the page, there is a REQUEST sent to the webserver, and in turn a RESPONSE is sent back from which your page is updated.

The web server is said to run in a "promiscuous mode" - ie it serves many clients at the same time. In between the requests you make, the web server may well have served 1,000 other clients.

To allow the server recognise you each time you make a request, the application keeps a record of you each time, issuing a token which generally is sent back and forth using a cookie. The period where this token is valid is called a session. Sessions have a timeout, and you may have well experienced a session timeout if you leave your computer for 30 minutes or so, and have to log back in. Your session expired, and a new token will be generated for a new session.

Sessions are by default managed by a Web Server as a set of files in a directory.

Now the issue that arises when you have more than one web server, is that the session handling has to be managed in a common location. With Moodle this is done by moving the session data back into the Database, so that each of the Web Servers has a common resource to generate and manage session handling.

This can be also handled by the `ldirectord` configuration, and keeping any connection from a particular IP dedicated to one particular web server.

### Updating

Often web servers can be configured to run in Virtual Machines (VM's), accessing their VM files from a common NAS Server. This simplifies updating. Otherwise it is required to have a scripted environment to update all web servers from a common master unit. In such situations it is important to make use of the Moodle "Maintenance Mode" to ensure a consistent update.