It specializes on running multiple tests across different browsers and operating systems in parallel.

It has 2 versions – Grid 1 and Grid 2.

Grid 1 is being deprecated by the Selenium.

It uses a hub-node (Master Slave) concept where tests are only run on a single machine (hub), but execution will be done by different machines called nodes in parallel.

**When to use?**

* Need to run tests against multiple browsers and operating systems at the same time. **This will ensure that the application you are testing is fully compatible with a wide range of browser-O.S combinations.**
* Save time in the execution of your test suites. **If you set up Selenium Grid to run, 10 test cases at a time, then you would be able to finish the whole suite around 10 times faster.**
* **The Hub**
* Central point where you load your tests.
* There should only be one hub in a grid.
* The hub is launched only on a single machine, a computer whose O.S is Windows 7 and whose browser is IE.
* The machine containing the hub is where the tests will be run, but you will see the browser being automated on the node.
* **The Nodes**
* Nodes are the Selenium instances that will execute the tests that you loaded on the hub.
* There can be one or more nodes in a grid.
* Nodes can be launched on multiple machines with different platforms and browsers.
* And it does not have to be the same platform as that of the hub.
* In this section, you will use 2 machines. The first machine will be the system that will run the hub while the other machine will run a node.
* It is also important to note their IP addresses.
* **Step 1:**
* Download the selenium server from <http://docs.seleniumhq.org/download/>
* **Step 2:**
* We are now going to launch a hub. Go to hub. Using the command prompt, navigate to the directory Selenium Server jar is placed.
* Command to run selenium server on hub:
* [**java**](http://www.guru99.com/java-tutorial.html) **-jar selenium-server-standalone-2.47.1.jar -role hub**
* **Name of the jar files varies on version 2.47.1 is not always true**
* **Step 3:**
* **Verify server is running**
* Another way to verify whether the hub is running is by using a browser. Selenium Grid, by default, uses port 4444 for its web interface.
* Simply open up a browser and go to <http://localhost:4444/grid/console>
* Click on view config and see the configuration of the hub
* By default hub provides 5 sessions on the server
* Using the command prompt, navigate to the directory where Selenium Server jar is placed.
* Command to run selenium server on node:
* [**java**](http://www.guru99.com/java-tutorial.html) **-jar selenium-server-standalone-2.47.1.jar –role node –hub http://<hostname/ip of hub>:4444/grid/register –port 5555**
* **http://<hostname/ip of hub>:4444/grid/console**
* Go to the hub and open <http://localhost:4444/grid/console>
* The port defaults to 5555 if not specified whenever the "-role" option is provided and is not hub.
* By default, starting the node allows for concurrent use of 11 browsers
* 5 Firefox, 5 Chrome, 1 Internet Explorer. The maximum number of concurrent tests is set to 5 by default.
* To change this and other browser settings, you can pass in parameters to
* -browser switch (each switch represents a node based on your parameters). If you use the -browser parameter, the default browsers will be ignored and only what you specify command line will be used.
* **-browser**
* java -jar selenium-server-standalone-2.39.0.jar -role node -hub http://<hostname/ip>:4444/grid/register -port 5555 -browser browserName=firefox,version=39.0,maxInstances=1,platform=MAC
* **Tip: If you need to provide a space somewhere in your browser parameters, then surround the parameters with quotation marks**
* "browserName=internet explorer,version=11,maxinstance=1,platform=WINDOWS" –port 5656

**-host <IP | hostname>** specify the host name or IP. usually not needed and determined automatically.

* **-timeout 30 (300 is default)** The timeout in **seconds** before the hub automatically releases a node that hasn't received any requests for more than the specified number of seconds. After this time, the node will be released for another test in the queue. This helps to clear client crashes without manual intervention. To remove the timeout completely, specify -timeout 0 and the hub will never release the node.
* **-maxSession 5 (5 is default)** The maximum number of browsers that can run in parallel on the node. This is different from the maxInstance of supported browsers (Eg: For a node that supports FF 3.6, FF 4 and IE 8, maxSession=1 will ensure that you never have more than 1 browser running).
* **-registerCycle N** = how often in ms the node will try to register itself again.Allow to restart the hub without having to restart the nodes.
* **Step 1:**
* Download the selenium server from <http://docs.seleniumhq.org/download/>
* Put it on Hub and all Nodes
* **Step 2:**
* Go to hub. Using the command prompt, navigate to the directory Selenium Server jar is placed.
* Run selenium server using this command
* [**java**](http://www.guru99.com/java-tutorial.html) **-jar selenium-server-standalone-2.47.1.jar -role hub**
* **Name of the jar files varies on version 2.47.1 is not always true**
* Step 3:
* Go to all nodes. Using the command prompt, navigate to the directory Selenium Server jar is placed.
* Run selenium server using this command
* [**java**](http://www.guru99.com/java-tutorial.html) **-jar selenium-server-standalone-2.47.1.jar –role webdriver –hub http://<hostname/ip of hub>:4444/grid/register –port 5555**
* **Advance command for specific instance**
* java -jar selenium-server-standalone-2.47.1.jar -role node –hub <hub hostname>/grid/register -browser "browserName=internet explorer,version=11,maxinstance=1, platform=WINDOWS" –port 5656