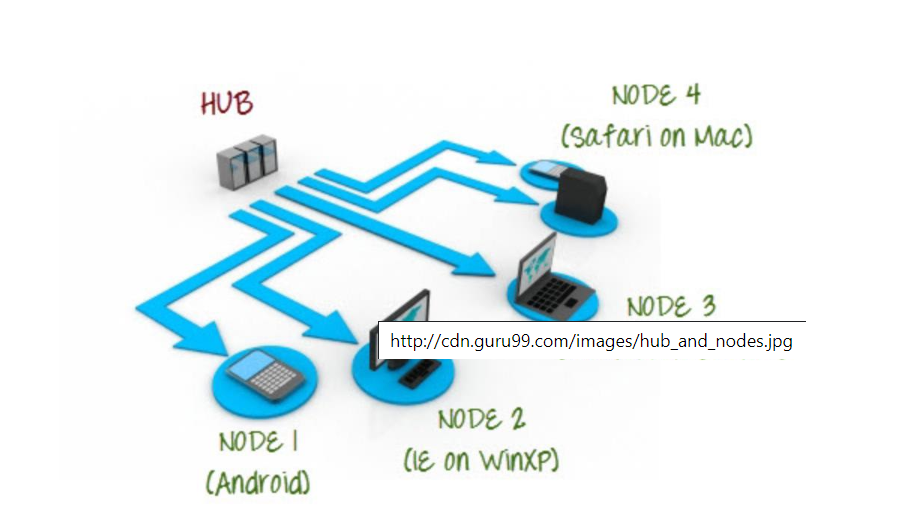
**DAY 4 - Introduction to Selenium Grid**

What is Selenium Grid?

Selenium Grid is a part of the Selenium Suite that specializes on running multiple tests across different browsers, operating systems, and machines in parallel.



Selenium Grid uses a hub-node concept where you only run the test on a single machine called a hub, but the execution will be done by different machines called nodes.

**When to Use Selenium Grid?**

You should use Selenium Grid when you want to do either one or both of following:

· Run your tests against different browsers, operating systems, and machines all 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, say, 4 tests at a time, then you would be able to finish the whole suite around 4 times faster.

**What is a Hub and Node?**

**The Hub**

· The hub is the central point where you load your tests into.

· There should only be one hub in a grid.

· The hub is launched only on a single machine, say, 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.

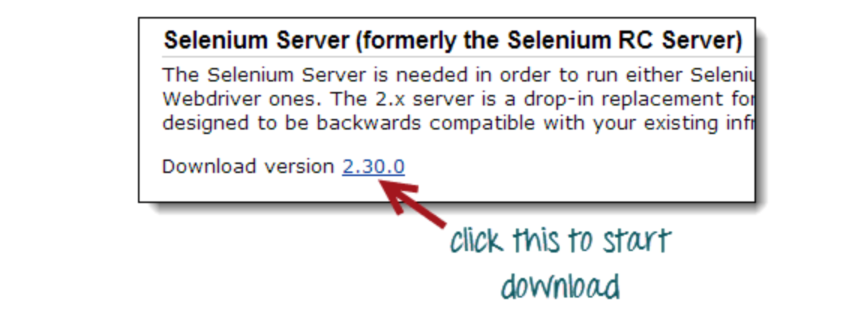
· Nodes can be launched on multiple machines with different platforms and browsers.

· The machines running the nodes need not be the same platform as that of the hub.

**How to Install and Use Grid 2.0?**

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. For simplicity, let us call the machine where the hub runs as "Machine A" while the machine where the node runs will be "Machine B". It is also important to note their IP addresses. Let us say that Machine A has an IP address of 192.168.1.3 while Machine B has an IP of 192.168.1.4.

**Step 1 Download the Selenium Server by here.**



**Step 2**

You can place the Selenium Server .jar file anywhere in your HardDrive. But for the purpose of this tutorial, place it on the C drive of both Machine A and Machine B. After doing this, you are now done installing Selenium Grid. The following steps will launch the hub and the node.

**Step 3**

· We are now going to launch a hub. Go to Machine A. Using the command prompt, navigate to the root of Machine A's - C drive, because that is the directory where we placed the Selenium Server.

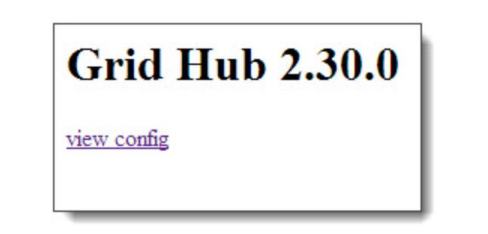
· On the command prompt, type java -jar selenium-server-standalone-2.30.0.jar -role hub

· The hub should successfully be launched. Your command prompt should look similar to the image below



**Step 4**

Another way to verify whether the hub is running is by using a browser. Selenium Grid, by default, uses Machine A's port 4444 for its web interface. Simply open up a browser and go tohttp://localhost:4444/grid/console

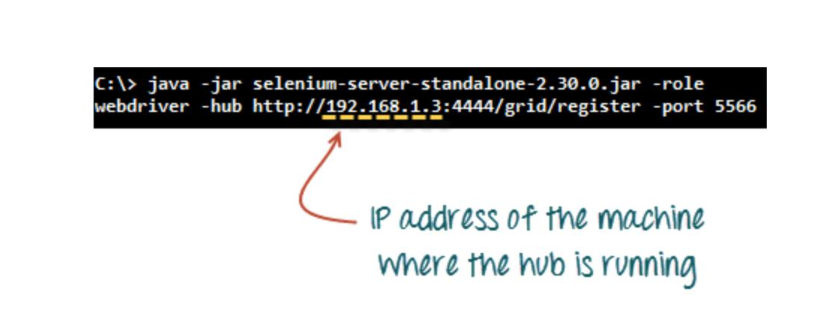


Also, you can check if Machine B can access the hub's web interface by launching a browser there and going to where "iporhostnameofmachineA" should be the IP address or the hostname of the machine where the hub is running. Since Machine A's IP address is 192.168.1.3, then on the browser on Machine B you should typehttp://192.168.1.3:4444/grid/console

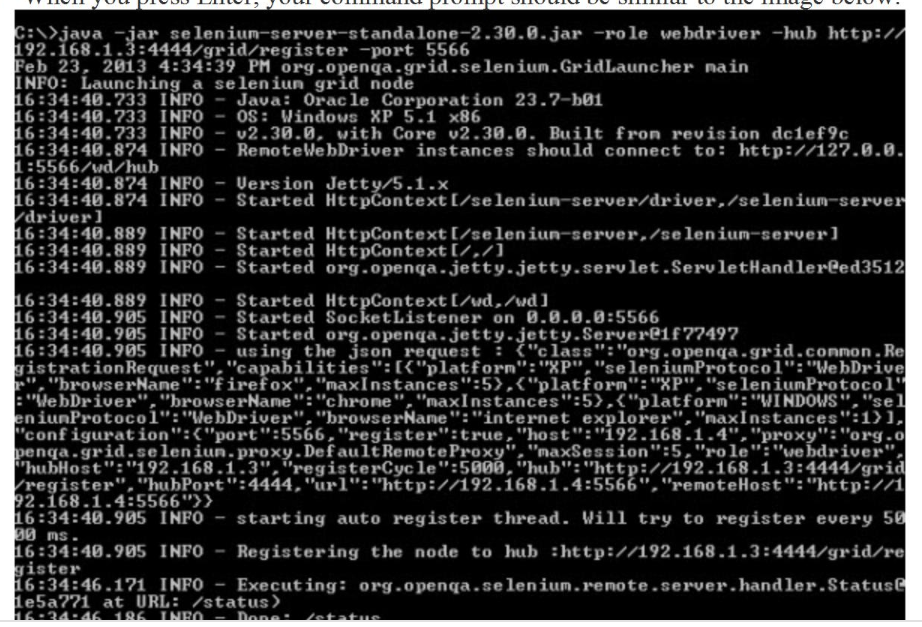
**Step 5**

· Now that the hub is already set up, we are going to launch a node. Go to Machine B and launch a command prompt there.

· Navigate to the root of Drive C and type the code below. We used the IP address 192.168.1.3 because that is where the hub is running. We also used port 5566 though you may choose any free port number you desire.

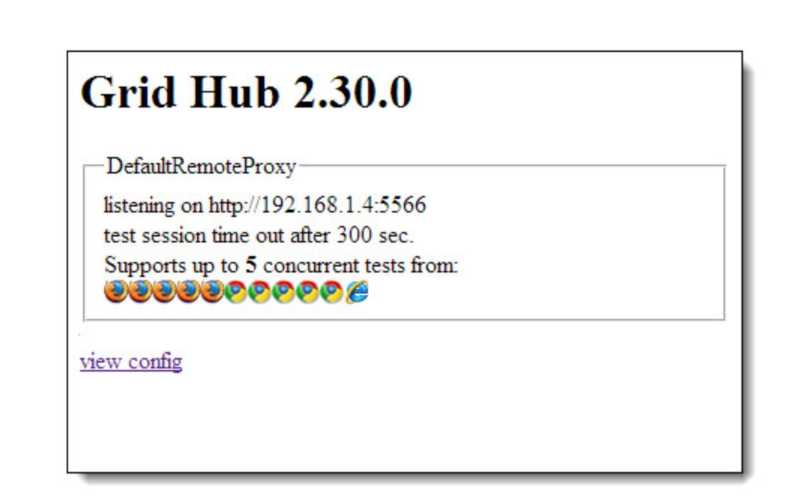


· When you press Enter, your command prompt should be similar to the image below.



**Step 6**

Go to the Selenium Grid web interface and refresh the page. You should see something like this.



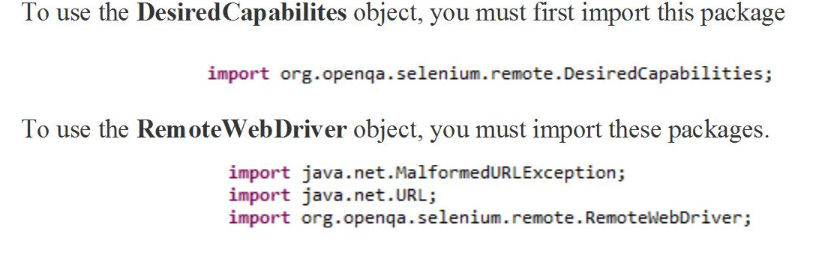
At this point, you have already configured a simple grid. You are now ready to run a test remotely on Machine B.

Designing Test Scripts That Can Run on the Grid

To design test scripts that will run on the grid, we need to use DesiredCapabilites and the RemoteWebDriverobjects.

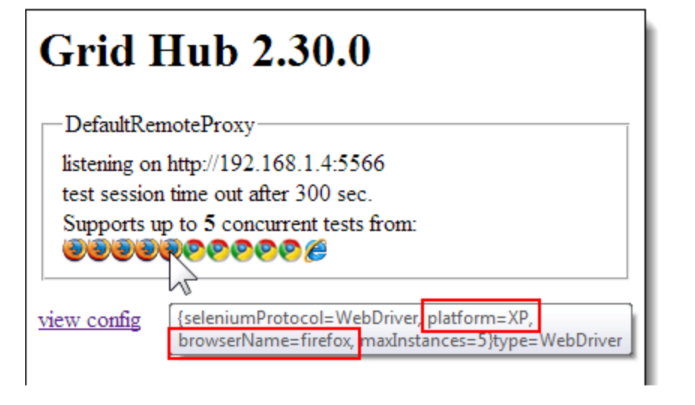
· DesiredCapabilites is used to set the type of browser and OS that we will automate

· RemoteWebDriver is used to set which node (or machine) that our test will run against.



Using the DesiredCapabilites Object

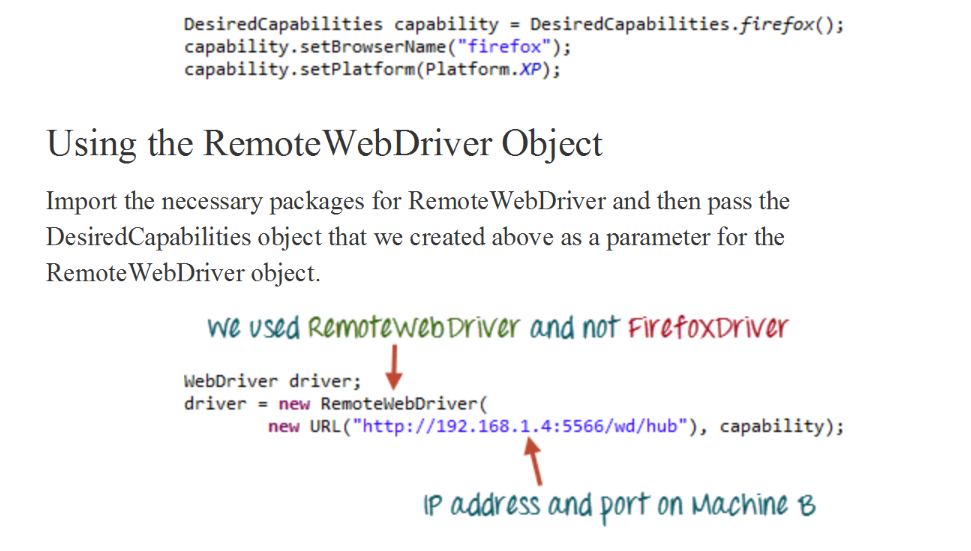
Go to the Grid's web interface and hover on an image of the browser that you want to automate. Take note of the platform and the browserName shown by the tooltip.



In this case, the platform is "XP" and the browserName is "firefox".

We will use the platform and the browserName in our WebDriver as shown below (of course you need to import the necessary packages first).

Using the RemoteWebDriver Object



Import the necessary packages for RemoteWebDriver and then pass the DesiredCapabilities object that we created above as a parameter for the RemoteWebDriver object.

Running a Sample Test Case on the Grid





Below is a simple WebDriver TestNG code that you can create **in Eclipse on Machine A**. Once you run it, automation will be **performed on Machine B.**

