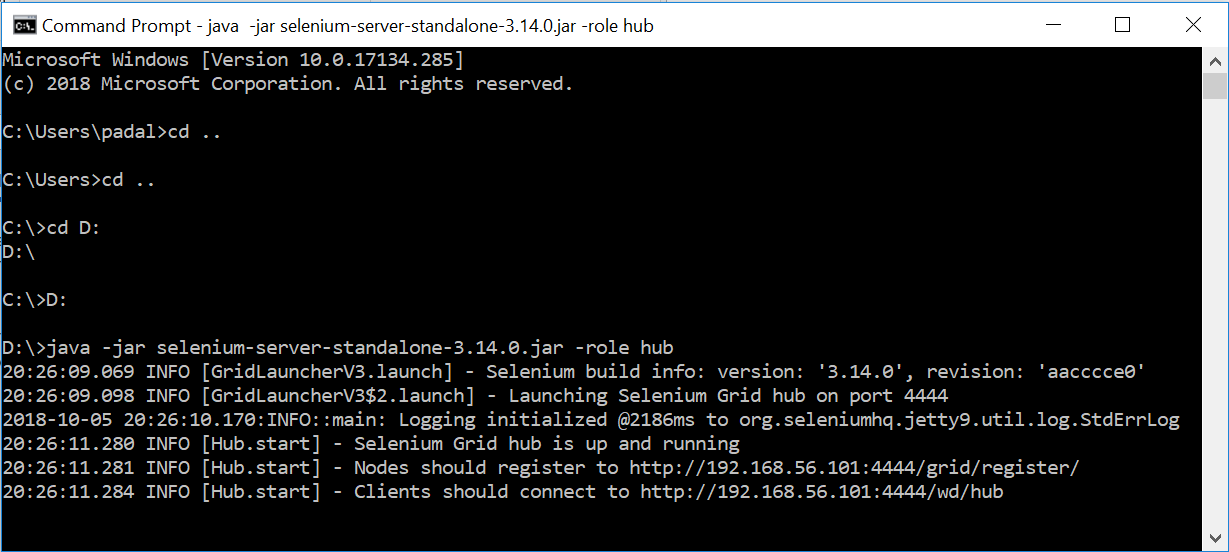
Now on the hub open command prompt and go to the folder where selenium stand alone jar file is there and run the command

java -jar selenium-server-standalone-3.13.0.jar -role hub

Once the command is run, it shows that “Selenium Grid Hub is up and running” and it should show where Nodes and clients should be connected.



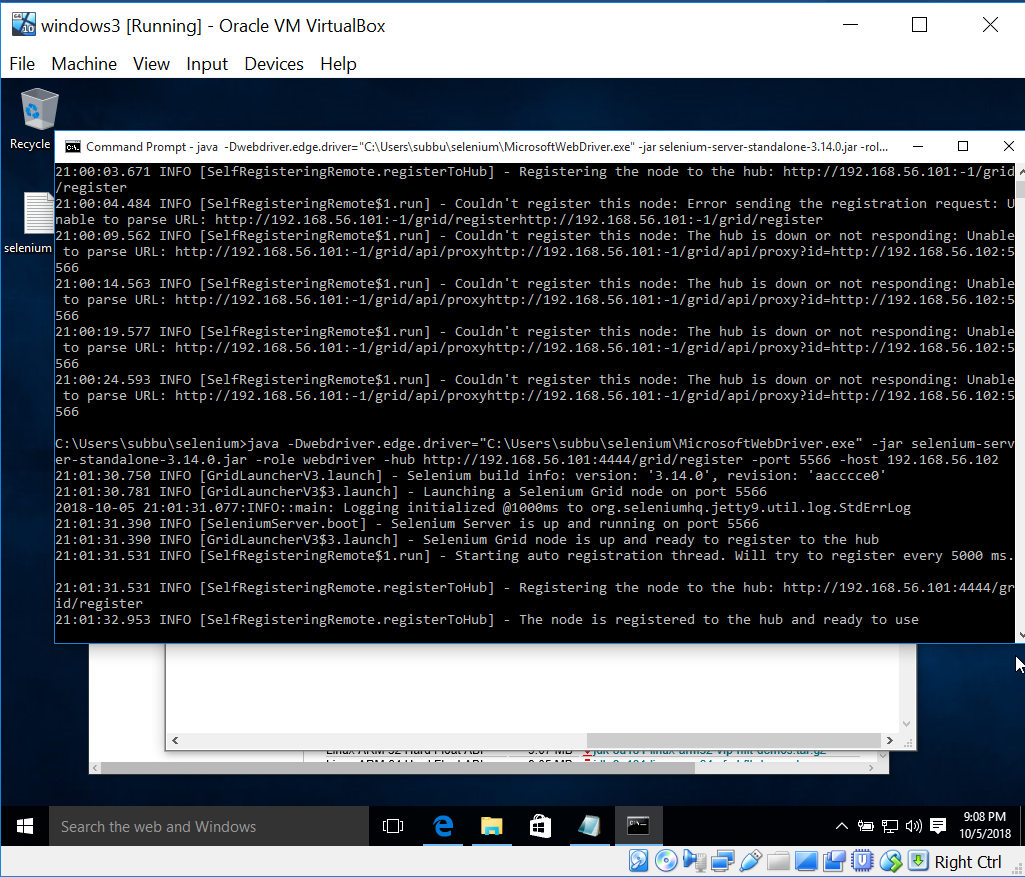
When we start the nodes we use the url mentioned in the statement “Nodes should register to ….” and in the actual program when we set tup the WebDriver we use the url mentioned in the statement “Clients should connect to ….”.

Now go to the node and run the following command.

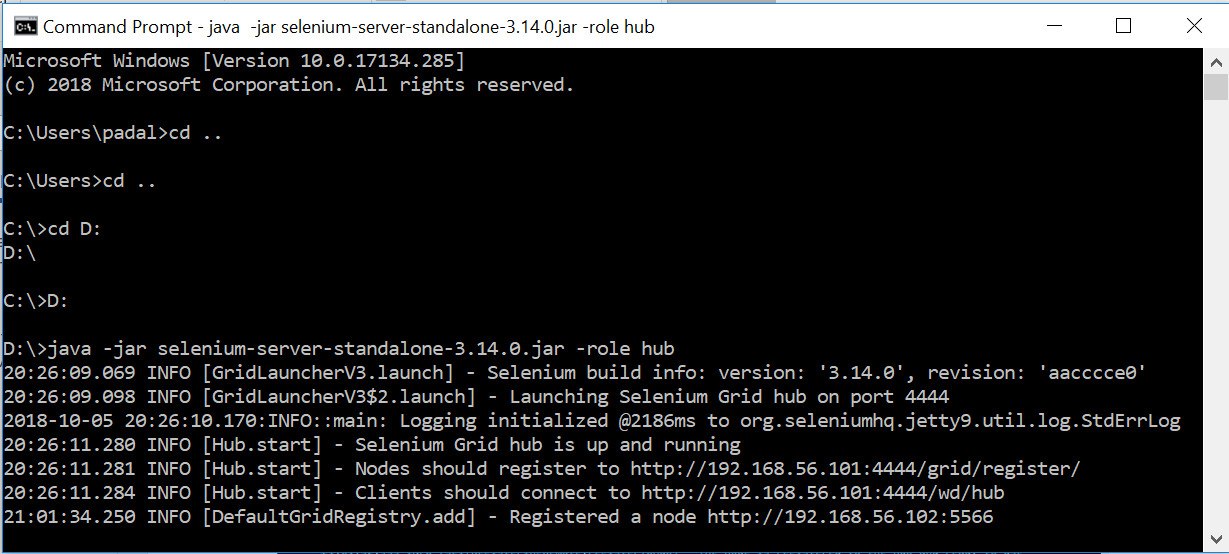
java -Dwebdriver.gecko.driver=”C:\Users\subbu\selenium\geckodriver.exe” -jar selenium-server-standalone-3.14.0.jar -role webdriver -hub [http://192.168.56.101:4444/grid/register -port 5566](http://192.168.56.101:4444/grid/register%20-port%205566) -host Host-Only adapter ip address

Replace the paths like the driver files paths, server jar files extensions etc. with your paths and versions.

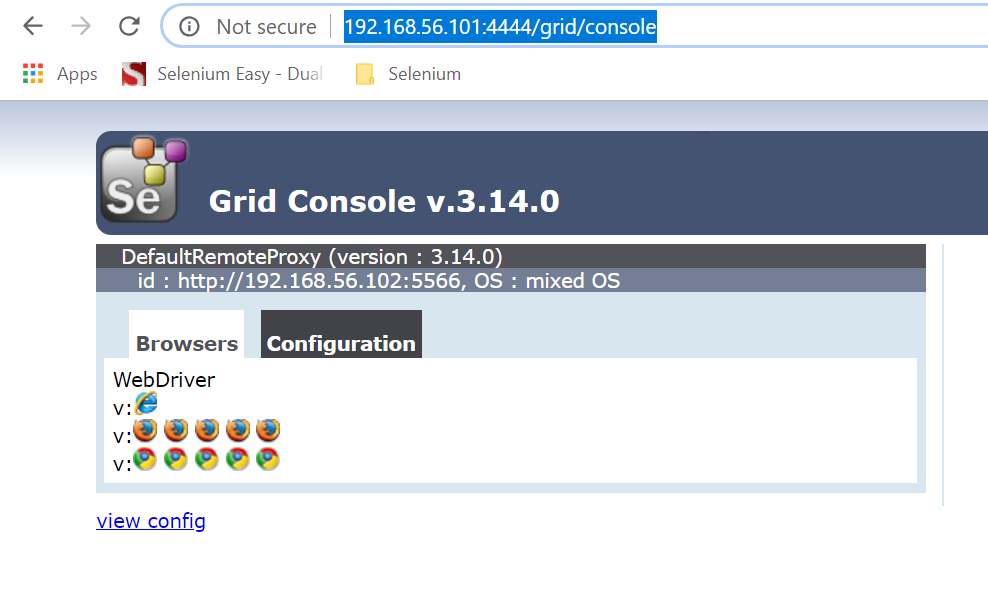
If you run the above command on the VMBox machine instance, it should show that it is connected to the hub and ready to use.



The following url should show all the systems that are connected to the hub. On the hub machine also it should show that a node is registered.



Now if you go to the console, it should show all the browsers registered to the hub. Go to the url “<http://192.168.56.101:4444/grid/console>” and you will see the nodes registered.



It should show the text “DefaultRemoteProxy (Version of the selenium server) and the id of the node (ip address). If it does not show then it won’t work.

Now write two scripts as follows.

Script1:

**package** SeleniumGridExamples;

**import** java.net.MalformedURLException;

**import** java.net.URL;

**import** org.openqa.selenium.Platform;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.remote.DesiredCapabilities;

**import** org.openqa.selenium.remote.RemoteWebDriver;

**import** org.testng.annotations.Test;

**public** **class** SLCase1 {

@Test

**public** **static** **void** test1() **throws** MalformedURLException, InterruptedException {

// **TODO** Auto-generated method stub

DesiredCapabilities dc = **new** DesiredCapabilities();

dc.setBrowserName("firefox");

dc.setPlatform(Platform.***WINDOWS***);

System.***out***.println("desired cap. done case1");

WebDriver driver = **new** RemoteWebDriver(**new** URL("http://192.168.56.101:4444/wd/hub"), dc);

driver.get("http://www.eenadu.net");

Thread.*sleep*(5000);

driver.quit();

}

}

Script2:

**package** SeleniumGridExamples;

**import** java.net.MalformedURLException;

**import** java.net.URL;

**import** org.openqa.selenium.Platform;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.remote.DesiredCapabilities;

**import** org.openqa.selenium.remote.RemoteWebDriver;

**import** org.testng.annotations.Test;

**public** **class** SLCase2 {

@Test

**public** **static** **void** test2() **throws** MalformedURLException, InterruptedException {

// **TODO** Auto-generated method stub

DesiredCapabilities dc = **new** DesiredCapabilities();

dc.setBrowserName("firefox");

dc.setPlatform(Platform.***WINDOWS***);

System.***out***.println("desired cap. done case2");

WebDriver driver = **new** RemoteWebDriver(**new** URL("http://192.168.56.101:4444/wd/hub"), dc);

driver.get("http://www.sakshi.com");

Thread.*sleep*(5000);

driver.quit();

}

}

**Desired Capabilities:**

When we run the scripts on selenium grid we need to set the desired capabilities. Desired capabilities set the browser we want, platform we want.

Here the WebDriver line we use is

WebDriver driver = **new** RemoteWebDriver(**new** URL("http://192.168.56.101:4444/wd/hub"), dc);

We set the url that we see in the hub’s command prompt “Clients should connect to <http://192.168.56.101:4444/wd/hub>”. We also pass the DesiredCapabilities object to this method.

Now conver the above two test cases into TestNG in eclipse. (Right on the package and convert them).

TestNG XML:

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">

<suite name=*"Suite"*>

<test thread-count=*"5"* name=*"Test"*>

<classes>

<class name=*"SeleniumGridExamples.SLCase2"*/>

<class name=*"SeleniumGridExamples.SLCase1"*/>

</classes>

</test> <!-- Test -->

</suite> <!-- Suite -->

Now the TestNG xml and you should be able to see that the tests are running on the node.