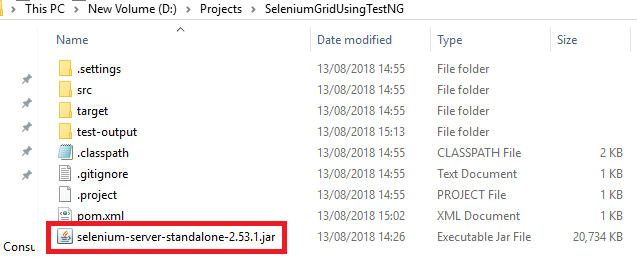
Selenium Grid using TestNG

1. Start Selenium Grid:

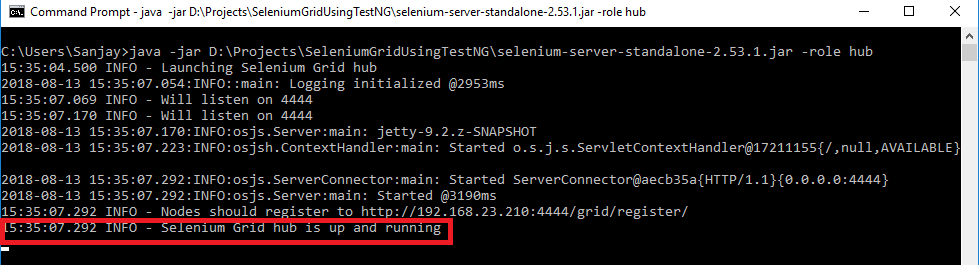
Down Selenium Standalone Server from seleniumhe.org.

For this project I have downloaded “selenium-server-standalone-2.53.1.jar” and saved in my project folder



Open CMD and execute the below command (Correctly give the selenium stand along server path

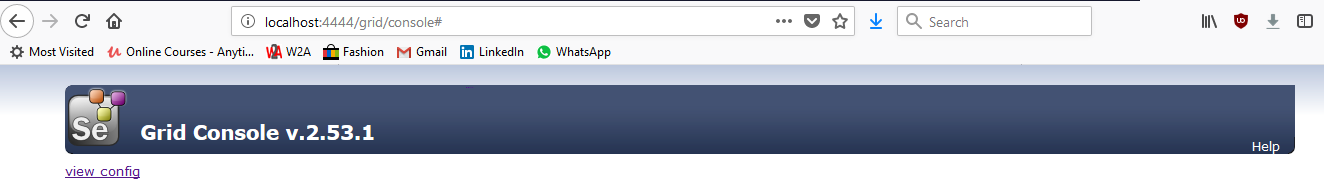
**java -jar selenium-server-standalone-2.53.1.jar -role hub**



Once Selenium hub  is ready then you can check the status on the browser as well.

Open http://localhost:4444/grid/consoles on local browser and hit enter

You will get below screen with no node connected

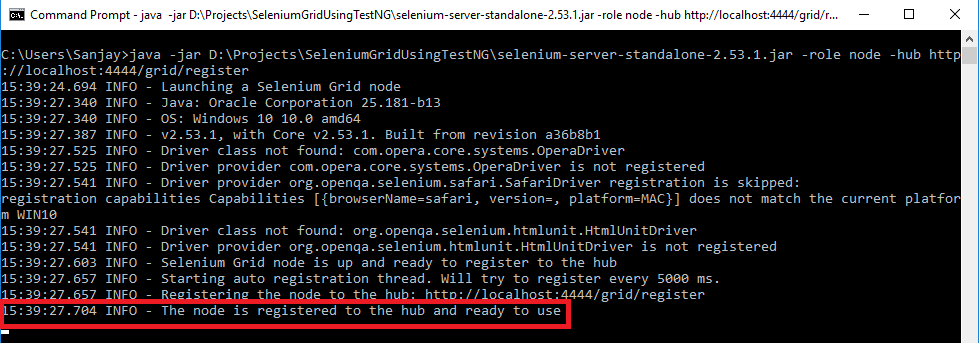


Now we need to create node and hub will manage the nodes.Let’s create a node and then we will trigger the test on the node.

### How to start Selenium node

To start the node execute below command and it will start the node

***java -jar selenium-server-standalone-2.53.1.jar -role node  -hub http://localhost:4444/grid/register***

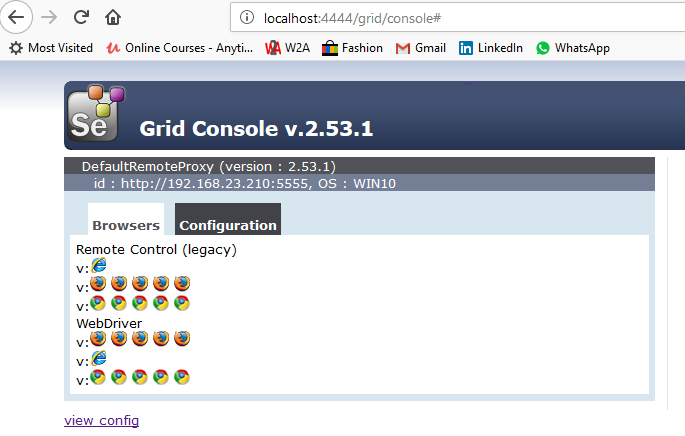


Once node is connected just open the hub Dashboard and you can see node is connected and you can see the webdriver and RC instances

By default, once node is created you will get 11 instances for WebDriver and 11 Instance of  RC

5 Firefox, 5 Chrome and 1 IE browser

It means you can run 5 FF,5 Chrome, and 1 IE browser if you want to run more test on this browser then you can change from the command line while creating the node.



## Selenium Grid for remote execution

Now we can start implementing part, we have to use RemoteWebDriver to execute our test on grid environment. We also have to use DesiredCapability class to specify platform name and browsers as well.

**package** com.sscl.seleniumgrid;

**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** SeleniumGridTest {

@Test

**public** **void** runOnChrome() **throws** MalformedURLException {

// We have to mention browser which we want to use for test execution

DesiredCapabilities cap = DesiredCapabilities.*chrome*();

// Set the platform where we want to run our test- we can use MAC and Linux and

// other platforms as well

// As I am running on Windows where are my both hub and node

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

// Here you can use hub address, hub will take the responsibility to execute the

// test on respective node

URL url = **new** URL("http://localhost:4444/wd/hub");

// Create driver with hub address and capability

WebDriver driver = **new** RemoteWebDriver(url, cap);

// start application

driver.get("https://github.com/SanjayAdav1972/"); // I'm opening my GitHub page

driver.manage().window().maximize();

// get the title and print the same

System.***out***.println("Page title is " + driver.getTitle());

// close opened instance of my WebDriver to clean up resources

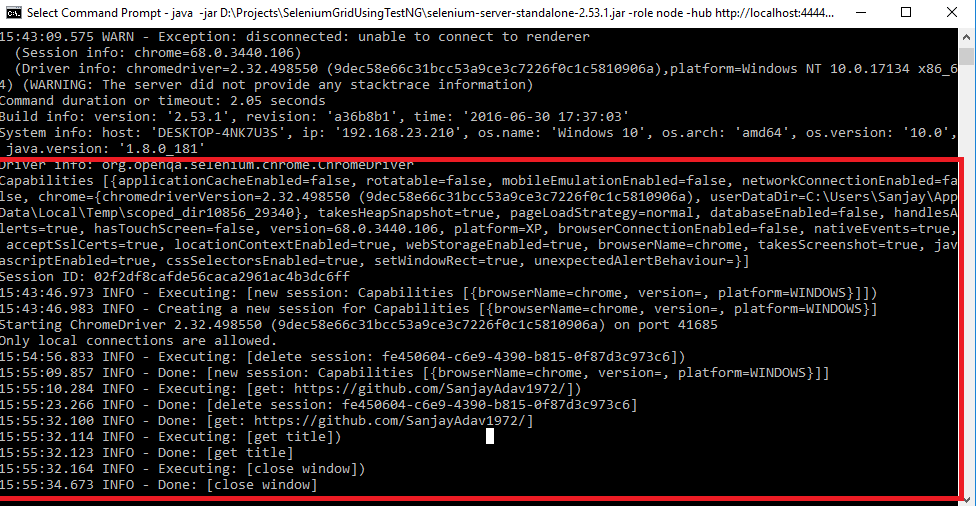
driver.close();

}

}

### Console output

Once test will execute you can see all the commands on node cmd.



### Some point to remember while using Selenium Grid for remote execution

* In above example, I used windows as the platform but you can use any platform.
* I have used chrome as browser but you can use any other browser like Firefox and IE etc
* You can use Selenium grid for parallel execution as well.
* Here I have used TectNG framework but you can use Junit or BDD Cucumber any test frameworks.
* If you do not have the environment to run the test then you can use Cloud-based environment like [***Browser Stack***](https://www.browserstack.com/start) and [***Sauce lab***](https://saucelabs.com/) which will allow you to run the test on their cloud.

I will provide how to run the selenium script on BrowserStack and SuaceLab soon!