9. Demonstrate Running Tests on Selenium Grid on Multiple Browsers .

Step 1: Running the Tests on Grid

- Open Eclipse.
- Click on **File>New>Other> Class.**
- Give a valid Class name (example: GridTest).
- Check the **public static void main** checkbox and click on **finish** , which will then create a blank Java class.
- Write the desired capabilities in the class, which will look like:

```
package testing.sidTesting;

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

public class GridTest {

  public static void main(String[] args) {
    DesiredCapabilities cap = new DesiredCapabilities();
    cap.setBrowserName("chrome");
    cap.setPlatform(Platform.WIN10);
  }

}
```

- Start the selenium grid hub in the command prompt using **java -jar selenium-server-standalone-3.141.59.jar -role hub** command.
- Start the Selenium grid node in the command prompt using **java -Dwebdriver.chrome.driver="chromedriver.exe -jar selenium-server-standalone-3.141.59.jar -role node -hub http://localhost:4444/grid/register** command.
- Go to Eclipse and add a statement for remoteWebdriver, which has an implementation of WebDriver, to pass the hub port (`http://192.168.1.248:4444/wd/hub`), and DesiredCapabilities object as parameters.
- Write Selenium code to open the browser and navigate to any web page (example: Google page).

```java
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;

public class GridTest {

  public static void main(String[] args) throws MalformedURLException {
    DesiredCapabilities cap = new DesiredCapabilities();
    cap.setBrowserName("chrome");
    cap.setPlatform(Platform.WIN10);


    URL url = new URL("http://192.168.1.248:4444/wd/hub");
    WebDriver driver = new RemoteWebDriver(url, cap);

    driver.get("https://www.google.com");
    System.out.println("Google Title: " + driver.getTitle());

    driver.close();
  }

}
```

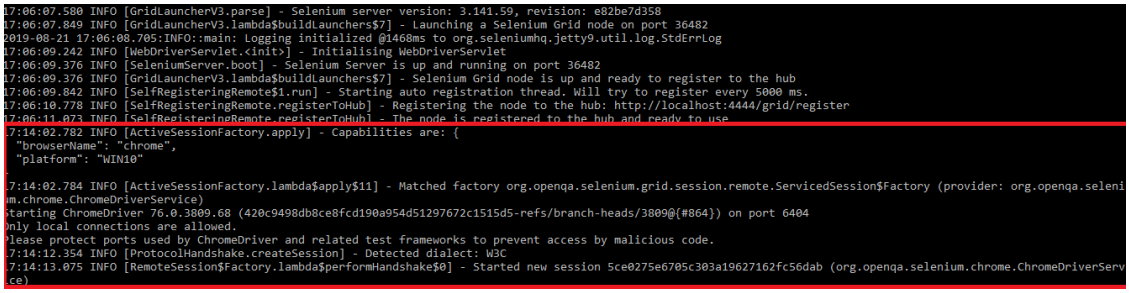
- Execute the Java program by right-clicking on the program and navigating to **Run As**--> **1 Java Application.**
- This is how it looks like in the Eclipse console.

```
Aug 21, 2019 5:14:13 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Google Title: Google
```

- We can see that the capabilities passed through are displayed in both command prompts in the server (hub) as well as in clients (node).
- Selenium grid hub in command prompt with desired capabilities will look like:

```
17:01:00.989 INFO [GridLauncherV3.parse] - Selenium server version: 3.141.59, revision: e82be7d358
17:01:01.317 INFO [GridLauncherV3.lambda$buildLaunchers$5] - Launching Selenium Grid hub on port 4444
2019-08-21 17:01:02.084:INFO::main: Logging initialized @1821ms to org.seleniumhq.jetty9.util.log.StdErrLog
17:01:03.315 INFO [Hub.start] - Selenium Grid hub is up and running
17:01:03.317 INFO [Hub.start] - Nodes should register to http://192.168.1.248:4444/grid/register/
17:01:03.317 INFO [Hub.start] - Clients should connect to http://192.168.1.248:4444/wd/hub
17:06:11.072 INFO [DefaultGridRegistry.add] - Registered a node http://192.168.1.248:36482
17:14:02.672 INFO [RequestHandler.process] - Got a request to create a new session: Capabilities {browserName: chrome, platform: WIN10}
17:14:02.678 INFO [TestSlot.getNewSession] - Trying to create a new session on test slot {server:CONFIG_UUID=64178522-5111-44a4-b432-a610c8b45434, seleniumProtocol=WebDriver, browserName=chrome, maxInstances=5, platformName=WIN10, platform=WIN10}
```

- Selenium grid node in the command prompt with desired capabilities will look like:



Step 2: Pushing the code to your GitHub repositories
Open your command prompt and navigate to the folder where you have created your files.

cd <folder path>

Initialize your repository using the following command:

git init

Add all the files to your git repository using the following command:

git add .

Commit the changes using the following command:

git commit . -m “Changes have been committed.”

Push the files to the folder you initially created using the following command:

git push -u origin master