**Setting up Nodes:**

Set up virtual machine as mentioned in the earlier videos. Set up two windows hosts. Install firefox and chrome on these machines. Install java on the remote machines.

Download selenium standalone server and save it.

Download geckodriver.exe and chromedriver.exe files and save them in the same folder that we saved standalone server.

**Settings on remote machine to make it faster:**

These are only required since we are doing it locally. In any company, you don’t need to do this as normally the systems are individual systems and they have high RAM memory and normally everything is set up on them

**Disable windows search:**

Press Windows logo key + R at the same time, and type in services.msc.

Search for “Windows Search”, double click on it and change the start up type to disabled.

**Disable windows update:**

Search for “Windows update”, double click on it and change the start up type to disabled.

**Disable windows module installer:**

Search for “Windows module installer”, double click on it and change the start up type to disabled.

**Disable SuperFetch Service:**

Open command prompt in admin mode. Type the command “net.exe stop superfetch” and it will stop superfetch.

**Sharing folders between local machine and remote machine:**

Now since we have to upload images as part of our tests and these images are in our local machine, when the tests are run on remote machine it can’t find the images and the tests will fail. So, we need to create a shared folder between the local machine and remote machine so that images can be uploaded in remote machine as well.

For this we will make our project folder itself shared. For this first we need to do some settings on our our remote machine to enable sharing.

We will be sharing the project folder which is “HybridKeywordDriven”. Right click on this folder, go to properties, click on sharing and enter “Everyone” in the text box and click on add. You can also change the permission level to “Read/Write”.

Once this is done, go to the virtual machine. Now go to “Devices” tab at the top and click on “Insert Guest Additions CD Image”.

You will see a pop up and click on that.

It will install everything for sharing and restart the system.

Once the system is restarted, go to devices, shared folders and shared folder settings.

Here you can share the folder you want. I am sharing the project folder “HybridKeywordDriven”.

**Update the paths in the data:**

Once the folder is shared change the data files needed to update it with the shared path. This is mainly required in postvehicle data sheet where we provided the images path.

We saved the images paths as below.

|  |
| --- |
| adminuploadimage1 |
| C:\Users\padal\eclipse-workspacenew\HybridKeywordDriven\data\images\audisedan\audi1.jpg |
| C:\Users\padal\eclipse-workspacenew\HybridKeywordDriven\data\images\nissanmaxima\nissan1.jpg |
| C:\Users\padal\eclipse-workspacenew\HybridKeywordDriven\data\images\toyotaavalon\toyota1.jpg |
| C:\Users\padal\eclipse-workspacenew\HybridKeywordDriven\data\images\toyotaavalon\toyota1.jpg |

We have to change the paths using shared path.

|  |
| --- |
| adminuploadimage1 |
| [\\VBOXSVR\HybridKeywordDriven\data\images\audisedan\audi1.jpg](file:///\\vboxsvr\HybridKeywordDriven\data\images\audisedan\audi1.jpg) |
| [\\VBOXSVR\HybridKeywordDriven\data\images\nissanmaxima\nissan1.jpg](file:///\\VBOXSVR\HybridKeywordDriven\data\images\nissanmaxima\nissan1.jpg) |
| \\VBOXSVR\HybridKeywordDriven\data\images\toyotaavalon\toyota1.jpg |
| \\VBOXSVR\HybridKeywordDriven\data\images\toyotaavalon\toyota1.jpg |

This is the only change required.

**Settings for Chrome Browser:**

When you run the tests on chrome browser on remote machine normally the tests will be very slow. Because it tries to load the proxy settings. (You can see the message “downloading proxy settings…” and this will make the tests slow and sometimes the tests fail.

So, to disable this, go to settings in chrome and go to advanced settings and go to “open proxy settings”.

It will open a pop up and click on “Lan Settings” and uncheck the box “Automatically detect settings”.

Once you do this, the tests will be faster on chrome browser on remote machines.

**Updating the DriverScript.java program:**

**startexecution() method:**

Since we want to run the tests parallelly need to mention the parameter “parallel = true” in the DataProvider annotation.

@DataProvider(name = "maintestcases",parallel = **true**)

**public** **static** String[][] startexecution() **throws** Exception{

excelUtilities eu = **new** excelUtilities();

Properties gldata = **new** Properties();

InputStream input = **new** FileInputStream("src/executionEngine/config.properties");

gldata.load(input);

List<List<String>> testcases = **new** ArrayList<List<String>>();

testcases = eu.getTestCases(gldata.getProperty("WB\_PATH\_TESTS"), gldata.getProperty("WB\_PATH\_TESTS\_SHEET"));

**int** no\_test\_cases = testcases.size();

String testcasesobject[][] = **new** String[no\_test\_cases][3];

**for**(**int** i=0; i<testcases.size(); i++) {

testcasesobject[i][0] = testcases.get(i).get(0);

testcasesobject[i][1] = testcases.get(i).get(1);

testcasesobject[i][2] = testcases.get(i).get(2);

}

System.***out***.println(Arrays.*toString*(testcasesobject));

**return** testcasesobject;

}

**TestNG xml modification:**

We need to modify the TestNG xml as well.

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

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

<suite name=*"Suite"* parallel=*"methods"* data-provider-thread-count=*"2"*>

<listeners>

<listener class-name=*"executionEngine.Listeners"*/>

</listeners>

<test thread-count=*"2"* name=*"Test"*>

<classes>

<class name=*"executionEngine.DriverScript"*/>

</classes>

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

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

We have added the parallel=”methods” attribute and data-provider-thread-count=”2” as we want to run the tests on two remote machines parallelly. We also made the thread-count as 2.

**Starting the hub and nodes:**

Disable windows firewall on the host machine and nodes as well. Now run the following command on hub.(your main machine)

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

Now run the following commands on node machines.

java -jar selenium-server-standalone-3.9.0.jar -role webdriver -hub [http://192.168.56.101:4444/grid/register -port 5567 -host 192.168.56.102](http://192.168.56.101:4444/grid/register%20-port%205567%20-host%20192.168.56.102)

make sure that you change the standalone server versions and also the node registration url and also the node ip address.

Run the above command change the ip address of the nodes on all the nodes.

Now the systems are ready for parallel execution.

Now right click on the TestNG xml and run as TestNG Suite and the test should execute parallelly.