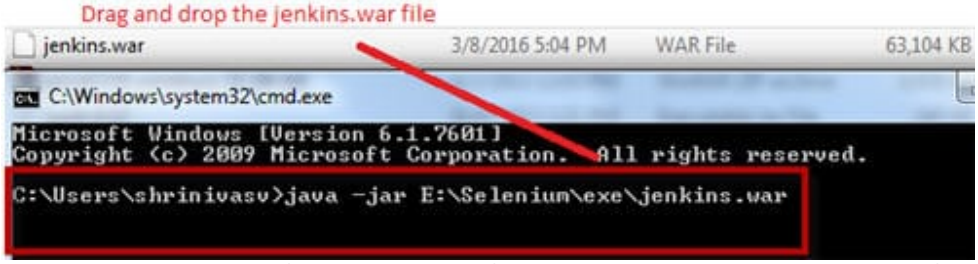
Jenkins usage and integration with selenium

Follow the below step-by-step procedure to use Jenkins with Selenium

Step #1:

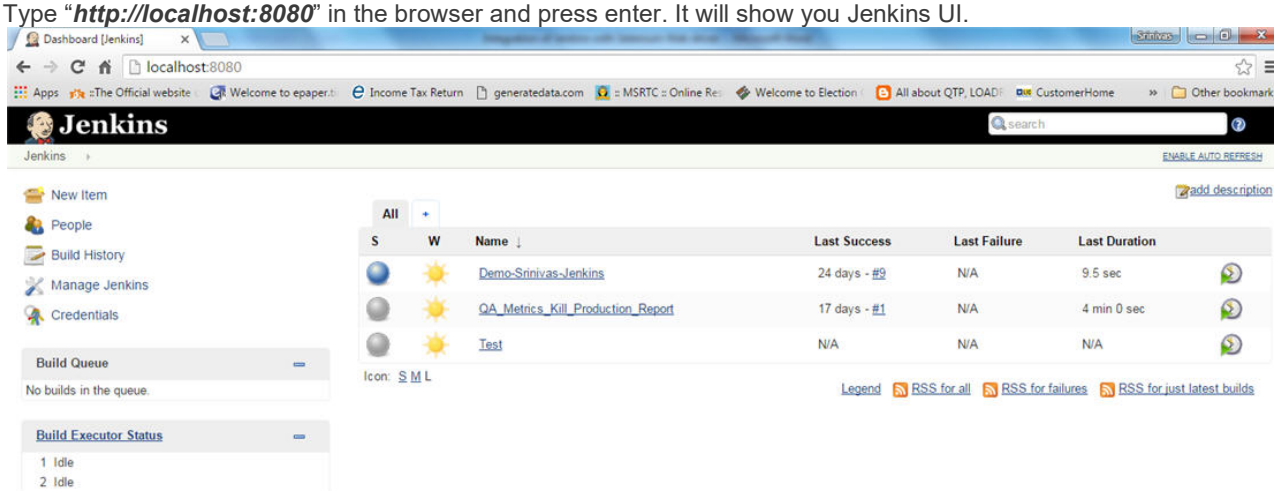
Download Jenkins from the official website of Jenkins – **Jenkins**. Download the latest .war file. Jenkins can be started via the command line or can run in a web application server.
Refer to the below steps for the execution through the command line:

1) Open the command prompt and type ***java –jar*** and enter the path of a .war file
(Note: Click on any image for enlarged view)



2) Press enter and check if your Jenkins.war file started to run and check the status information on the command prompt console.
It should show – **Jenkins is fully up and running**

3) Now check whether your Jenkins is ready to use; by default, it uses port 8080.
Type "***http://localhost:8080***" in the browser and press enter. It will show you Jenkins UI.



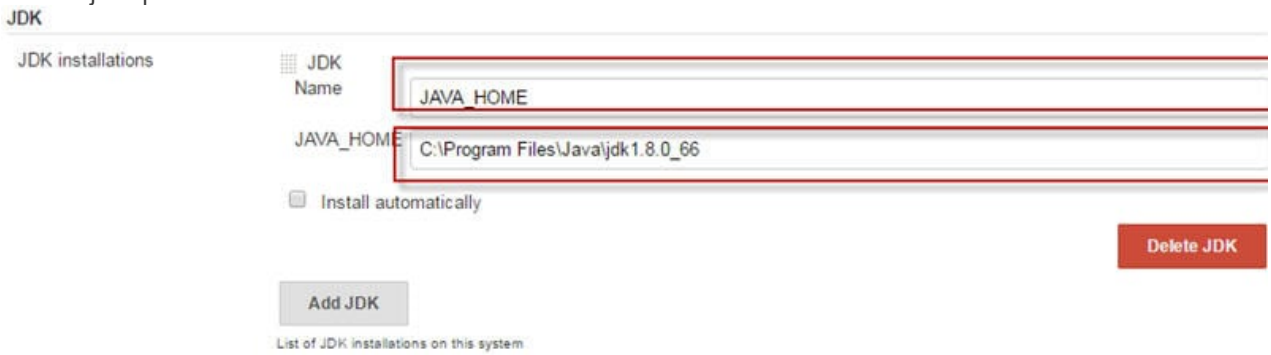
It will load the Jenkins dashboard empty by default. I created some Jenkins job in the above screenshot as an example and hence, it did not load empty.

Step #2:

To use Selenium with Jenkins you need to configure Jenkins with Selenium.

Follow the below steps:

- Go to Jenkins dashboard
- Click on manage Jenkins
- Click on configure Jenkins
- Click on JDK installation – In JDK name section enter the name, under Java Home section – give your java path



The radio button, **Install automatically** is checked by default. You need to uncheck it because it will automatically update with the new Java version and there might be a possibility that Selenium doesn't support the new Java version. It is better to uncheck it. Now click on apply and save.
Your Jenkins is configured with Selenium and is now ready to be used with Selenium. Both Jenkins and Selenium code is written in Java. Hence, if you give the Java path then internally it will communicate and process your job.

Step #3:

Now, create a Selenium script and a TestNG XML file. This TestNG XML file will be required for creating a batch file and we will use that batch file in Jenkins. Refer below TestNG code:

Refer below TestNG code:

```
package session_2;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.testng.Assert;
import org.testng.annotations.Test;

public class jenkins_demo
{

    @Test
    public void testgooglrsearch(){

        WebDriver driver = new FirefoxDriver();

        driver.get("http://google.in");
        String Expectedtitle = "Google";
        String Actualtitle = driver.getTitle();
        System.out.println("Before Assetion " + Expectedtitle + Actualtitle);
        Assert.assertEquals(Actualtitle, Expectedtitle);
        System.out.println("After Assertion  " + Expectedtitle + Actualtitle + " Title matched ");

    }
}
```

```
package session_2;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.testng.Assert;
```

```
import org.testng.annotations.Test;

public class jenkins_demo
{

@Test
publicvoidtestgooglrsearch(){

WebDriver driver = newFirefoxDriver();
//it will open the goggle page
driver.get("http://google.in");
//we expect the title "Google " should be present
String Expectedtitle = "Google";
//it will fetch the actual title
String Actualtitle = driver.getTitle();
System.out.println("Before Assetion " + Expectedtitle + Actualtitle);
//it will compare actual title and expected title
Assert.assertEquals(Actualtitle, Expectedtitle);
//print out the result
System.out.println("After Assertion " + Expectedtitle + Actualtitle + " Title
matched ");
 }
}
```

Output: Before Assertion GoogleGoogle
After Assertion, GoogleGoogle Title matched

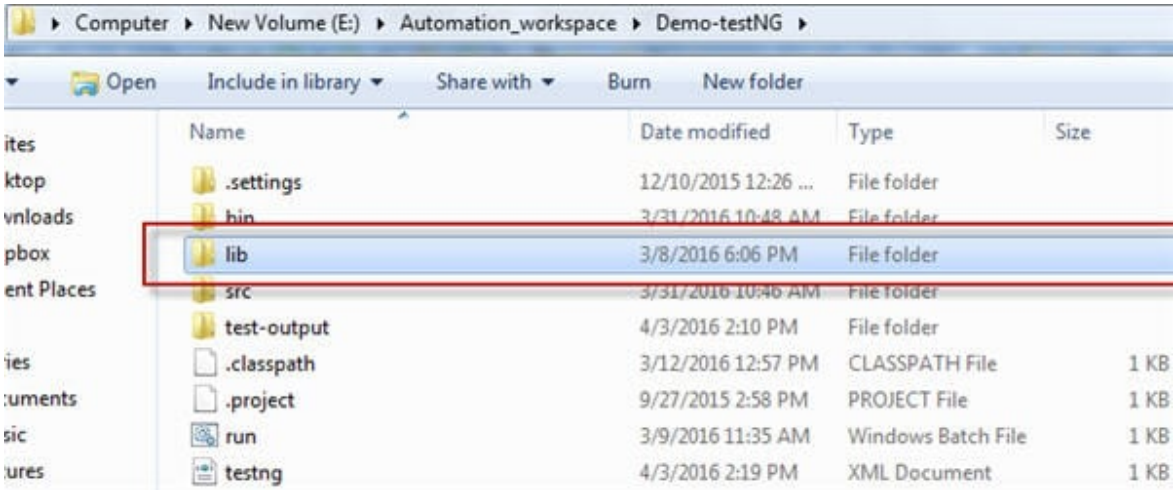
PASSED: testgooglrsearch

Create a TestNG xml file, refer below code:

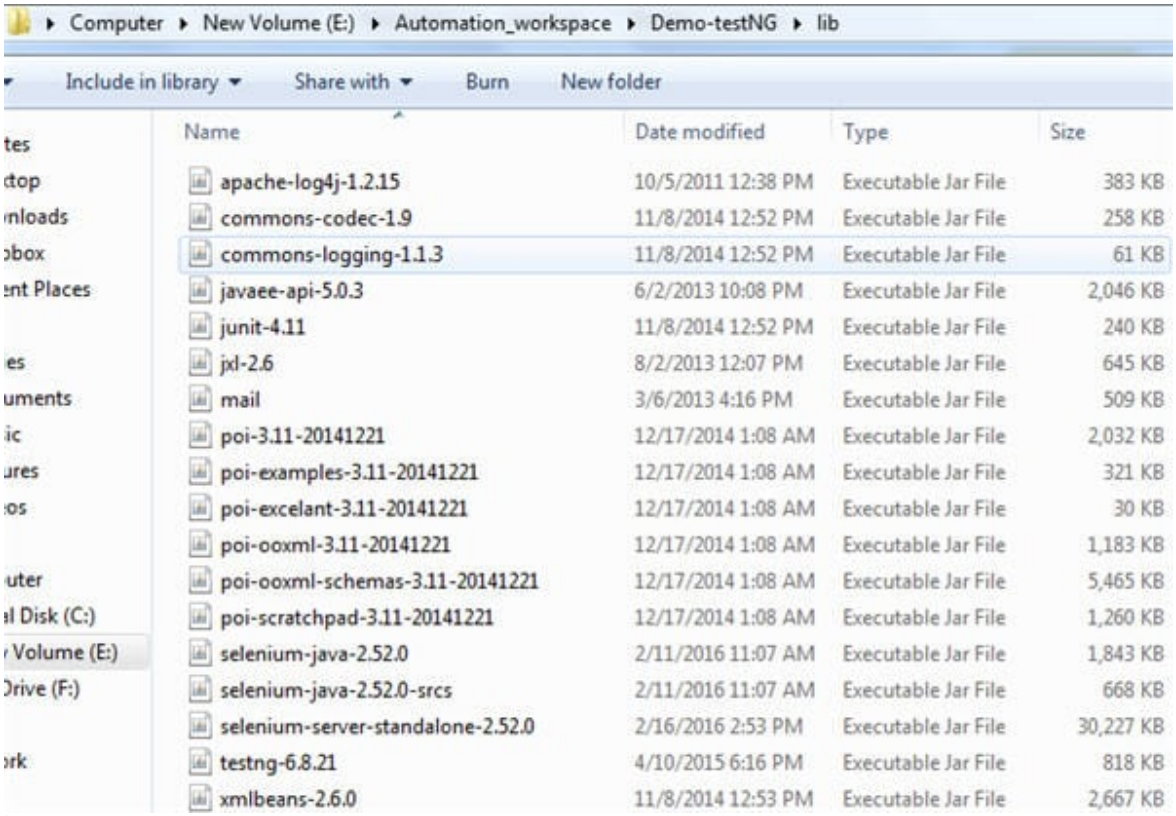
```
<suite name="Sample Suite">
 <test name="Learning Jenkins" >
<classes>
  <class name="session_2.jenkins_demo"> </class>
 </classes>
 </test>
</suite>
```

Step #4:
Go to your project root directory and create a library folder.

Refer the screenshot below:



Now, add all your jar files which are required for running your Selenium script:



Step #5:

Creating a batch file and using it in Jenkins

Create a batch file by following the below steps:

1) Open the notepad and type-: *Java –cp* bin;lib/* *org.testng.TestNG testng.xml*

By doing this, **Java –cp** will compile and execute a .class file which is located at **bin** directory and all our executable jar file is located at **lib** directory and we are using a TestNG framework so specify **org.testng.TestNG.** Also, specify the **name of xml** file which will trigger the expected TestNG script.

2) Save the file with .bat extension and check the type of file. It should be “windows batch file”. To cross-check whether the batch file is created properly, double-click the batch file and it will execute the code. Refer the below code of batch file:

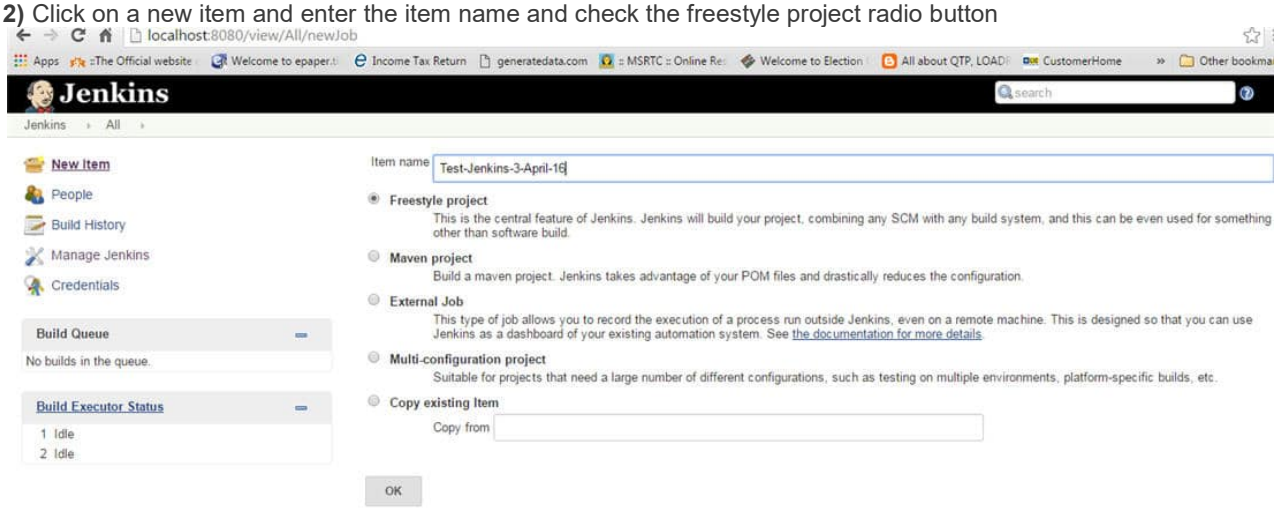
Step #6:

Next, we need to add a batch file in Jenkins.

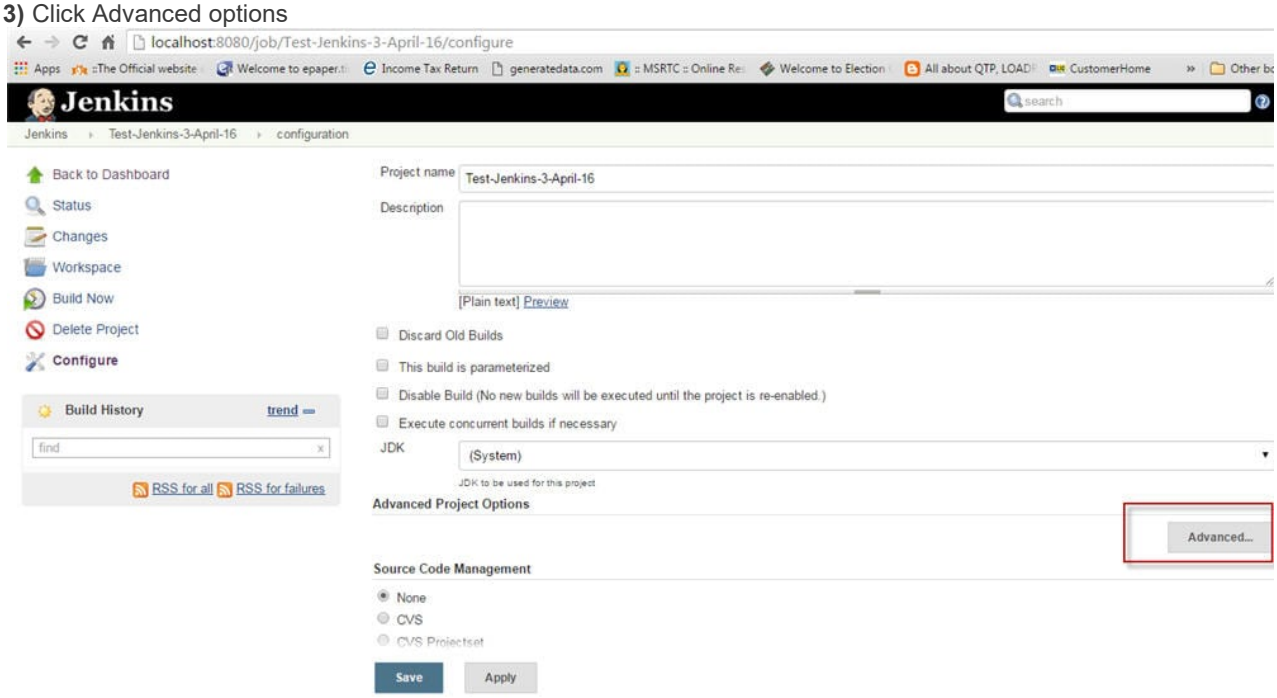
For adding the batch file follow the below steps:

1) Go to the Jenkins dashboard, create a new job in Jenkins

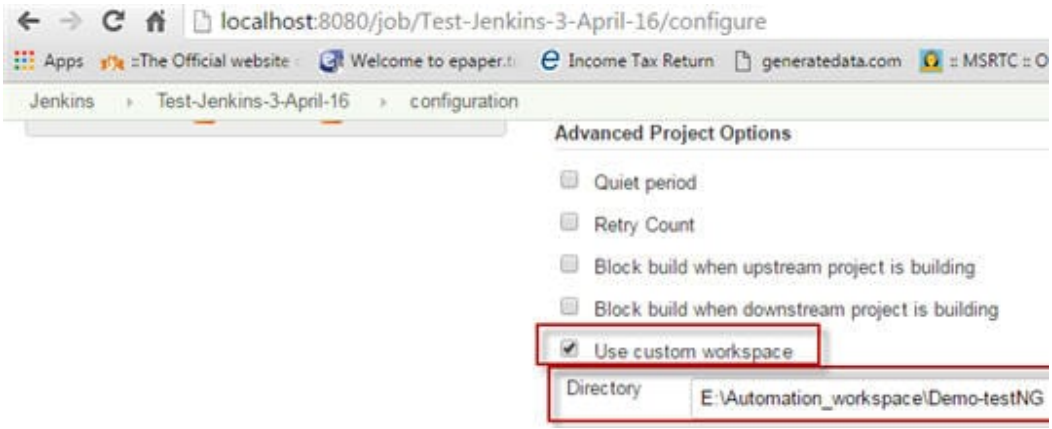
2) Click on a new item and enter the item name and check the freestyle project radio button



3) Click Advanced options



4) Click on use custom workspace and give your Selenium script project workspace path:
"*E:\Automation_workspace\Demo-testNG*"

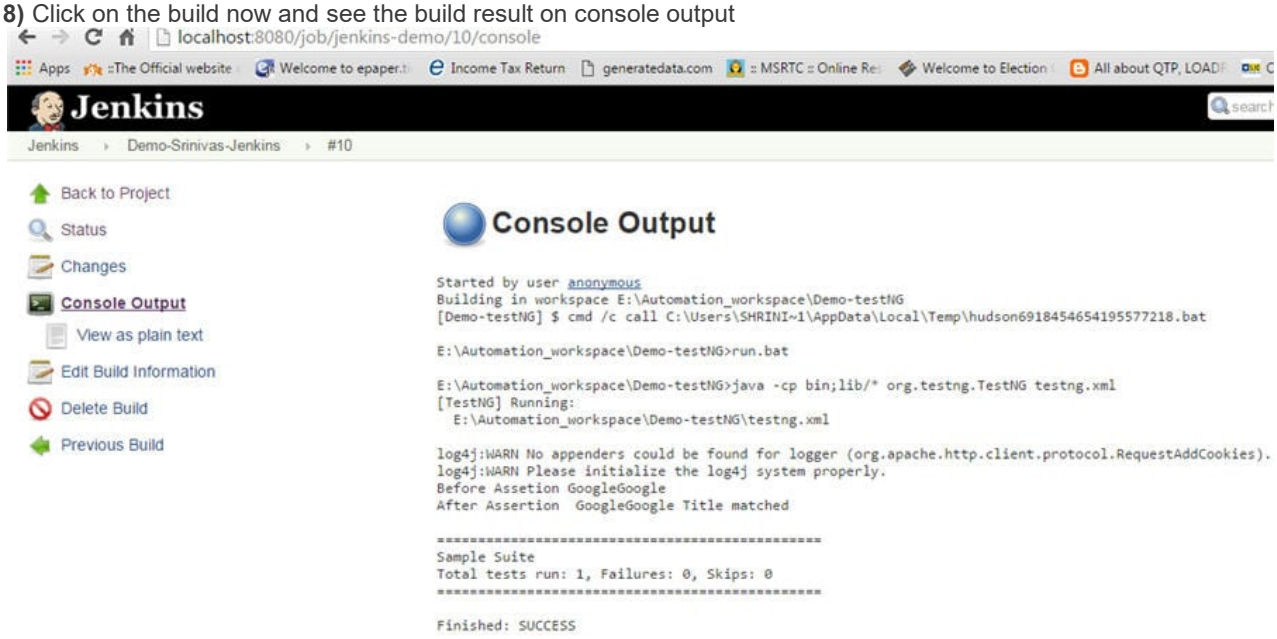


5) Then go to Build and Select an option from the drop-down box, execute your build through Windows batch command

6) And give your batch file name here – "*run.bat*"

7) Click on apply and save

8) Click on the build now and see the build result on console output



So far we have learned:

- How to start Jenkins
- How to configure Jenkins with Selenium

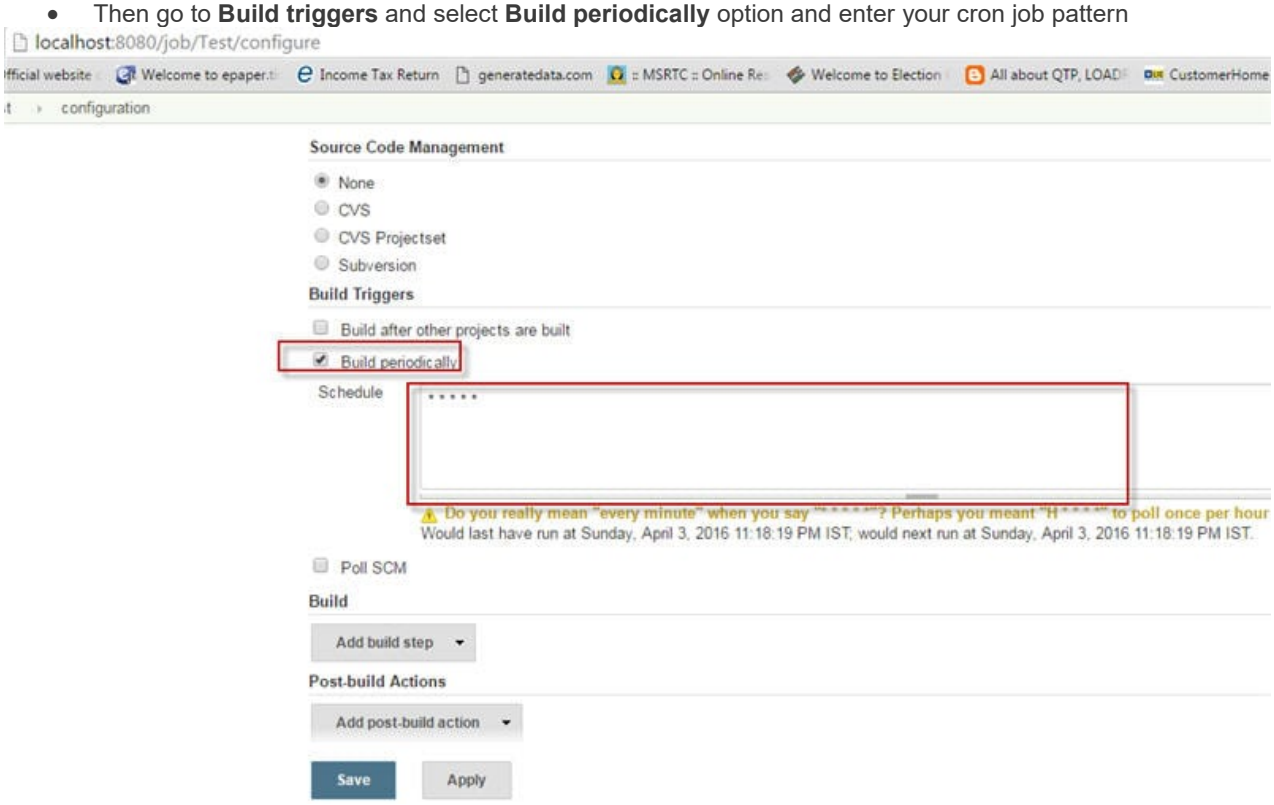
- Creating your batch file and executing it through Jenkins.

As you all are aware Jenkins is a very powerful tool which is mainly used for running nightly builds. Hence, we shall now learn how to schedule your build and send email notifications to the concerned team.

Scheduling Jenkins job

For scheduling you batch file, perform the below steps:

- Go to dashboard and click on the Jenkins job
- Click on configure and then on the advanced option
- Then go to **Build triggers** and select **Build periodically** option and enter your cron job pattern



- To understand cron job pattern follow this wiki link

I entered * * * * * which means it will run my job every minute

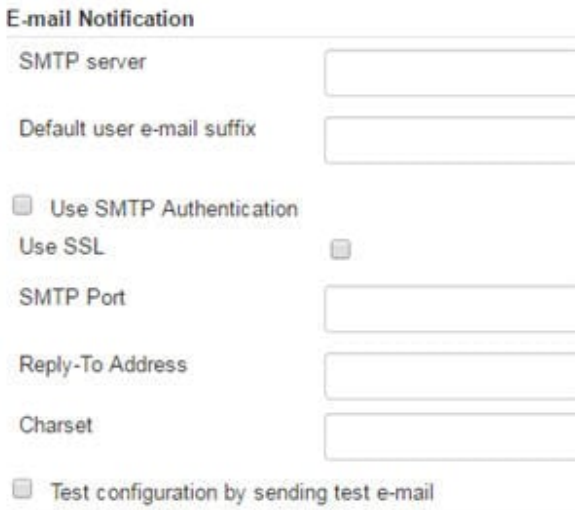
- Click on apply and save

There is no manual intervention. After scheduling the script, it will run at the scheduled time.

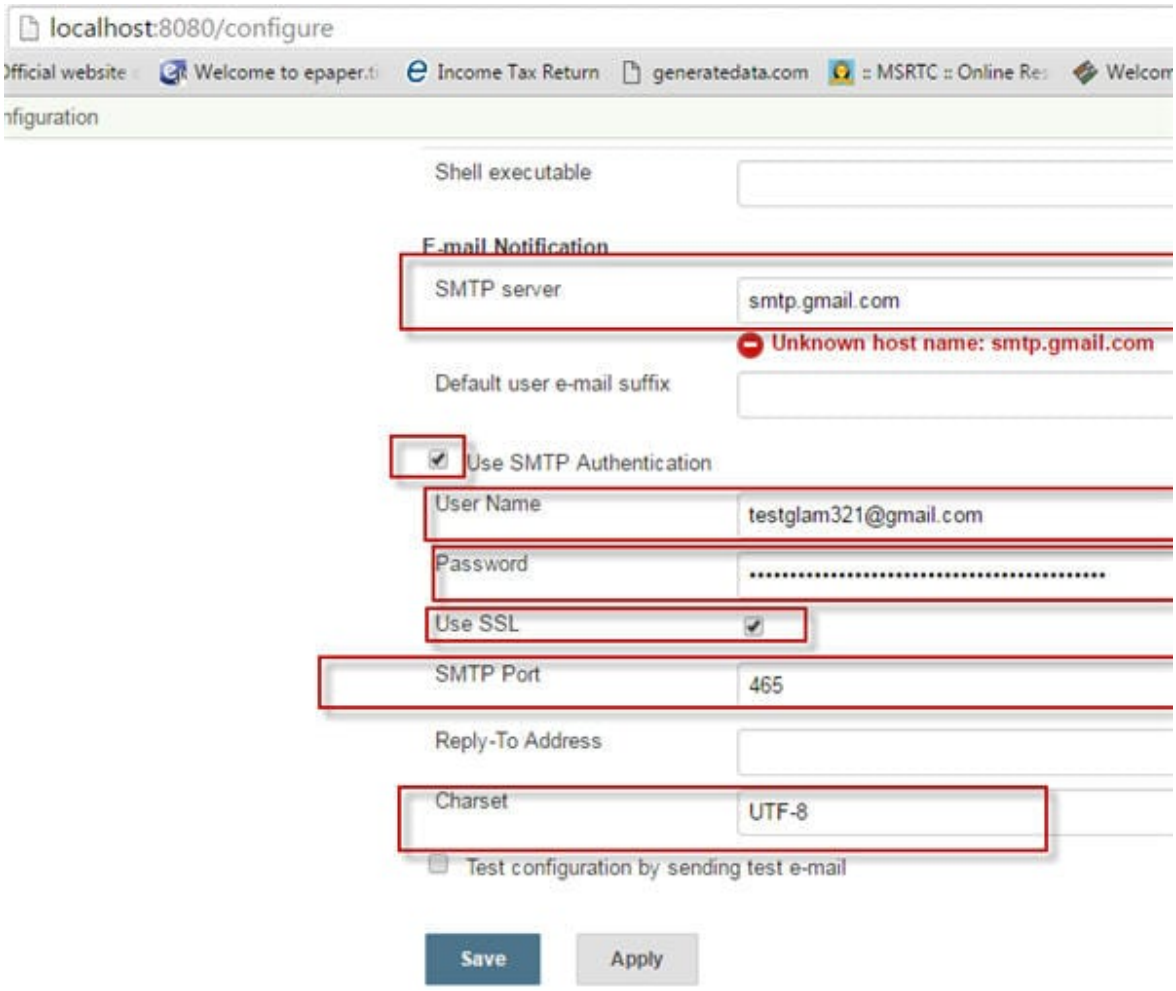
How to add email notifications

Next, we will cover how to add email notifications.
Refer the below steps:

- Go to the section ‘Manage Jenkins’
- Click on configure system
- Select Email notification



- Give your SMTP server address. I am using Gmail, as I can't mention my official server address. To know your official server address, contact your network support team
- I entered SMTP server name = **smtp.gmail.com**
- Click on the advance link and check Use SMTP Authentication check box
- Provide username, password and SMTP port number; it is 465 for Gmail. Check charset and make sure it is = UTF-8



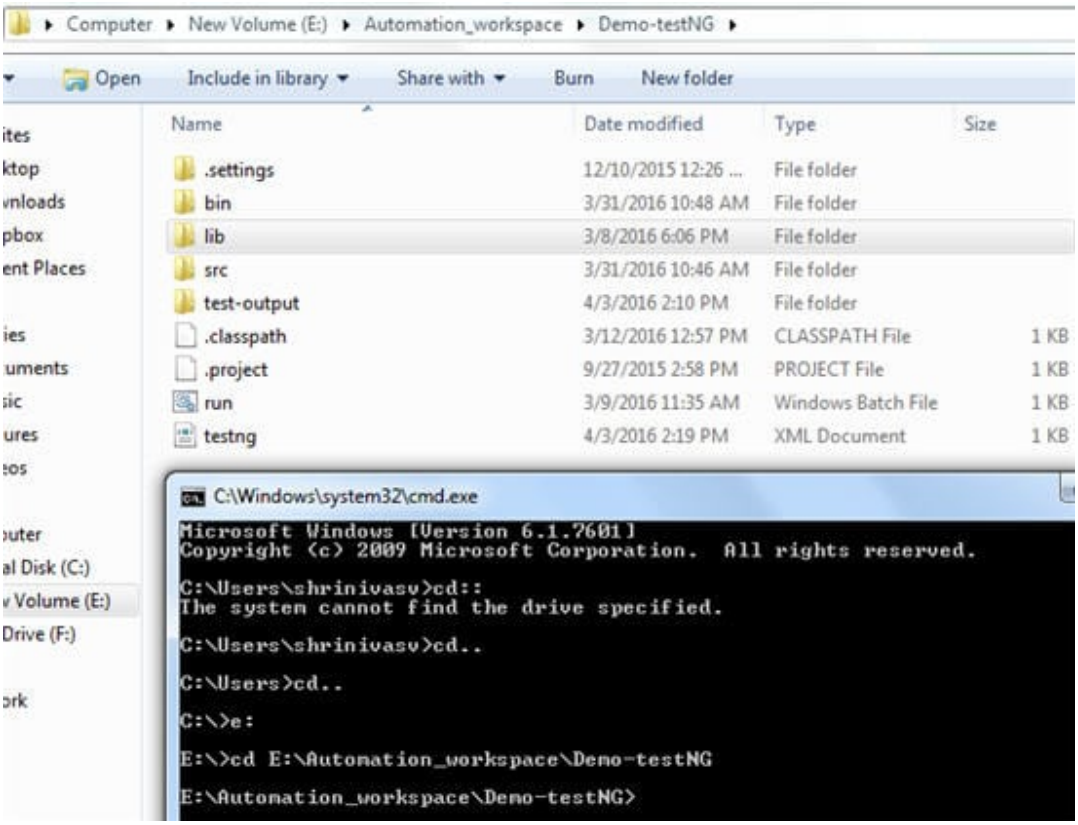
- Check your email configuration settings by clicking on Test configuration button.
- So, whenever the build passes or fails you will get the email notification.

Running Selenium script through command line

We will now see **how we can run Selenium script through command prompt**. This part has nothing to do with Jenkins. I am sharing this to give extra insights on Selenium.

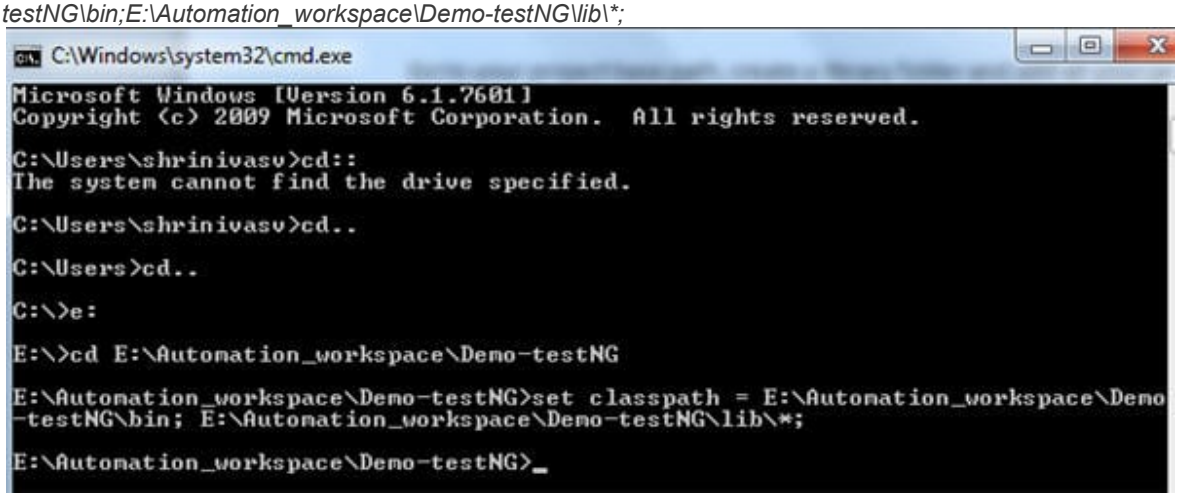
Follow the below steps:

- Open your command prompt and go to your project base path



- Set class path for your script file; which means we are specifying that our binary and library files are stored in this location

E:\Automation_workspace\Demo-testNG > set classpath = E:\Automation_workspace\Demo-testNG\bin;E:\Automation_workspace\Demo-testNG\lib;*



- Execute your testng.xml file by typing the command – ***java org.testng.TestNG testng.xml***

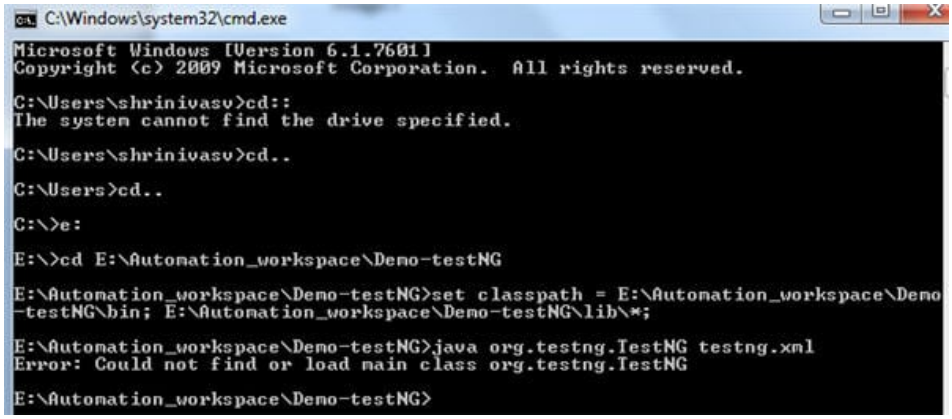
```
E:\Automation_workspace\Demo-testNG>java org.testng.TestNG testng.xml
[TestNG] Running:
  E:\Automation_workspace\Demo-testNG\testng.xml

log4j:WARN No appenders could be found for logger (org.apache.http.client.protoc
ol.RequestAddCookies).
log4j:WARN Please initialize the log4j system properly.
Before Assetion GoogleGoogle
After Assertion  GoogleGoogle Title matched

===============================================
Sample Suite
Total tests run: 1, Failures: 0, Skips: 0
===============================================
```

- When you press enter your script will start executing and you can see the test result in the UI

Sometimes while executing your script you may face error which says, “Could not find or load main class org.testng.TestNG”



Then you need to close your command prompt and again set your classpath as mentioned above and repeat the same steps. Your error will get resolved and the script will run.

Conclusion

Integration of Jenkins with selenium provides you to run your script each time there is any change in software code and deploy the code in a new environment. With Jenkins, you can save execution history and test reports.

In short, Jenkins is very useful when you have test cases ready and you want them to run using a single click. We can create or schedule a build to run the test cases using a batch file.