Introduction to GIT, Jenkins

Welcome to the Git series. This is the first tutorial in this series. In this tutorial we are going to study about **Git** which is a revision control system & about **Jenkins** which is a CI tool. Please read this tutorial thoroughly before you proceed to the next one.

What you will Learn:

Introduction about Git & its challenges
Jenkins as saviour
Download Jenkins
Start Jenkins
Create dummy project in Jenkins
Build Project
Configure Project to run periodically
Setup Email Notification
Conclusion
Recommended Reading

Introduction about Git & its challenges:

Suppose a team has 4 team members A, B, C & D. Every team member is working on the same project & each team member has to build 5 test cases. After building test cases, the team has to merge the code & store it at some common place, let us call this place as SCM (source code management system). GIT is the most popular SCM system today. The team members can store their code on the central repository GIT. Any team member can download the code from GIT on his local machine.

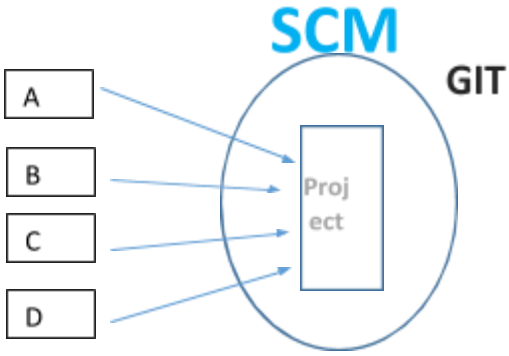

Figure 1

Suppose team members B & D download the project code on their local machines. Let suppose that the project code contains 2 files: Sel1.java & Sel2.java. Now 'B' does not know that 'D' has downloaded the same project & 'D' does not know that 'B' has downloaded the same project. Thus, both of them do not know that they have downloaded the same selenium files, Sel1.java & Sel2.java (see Figure 2)

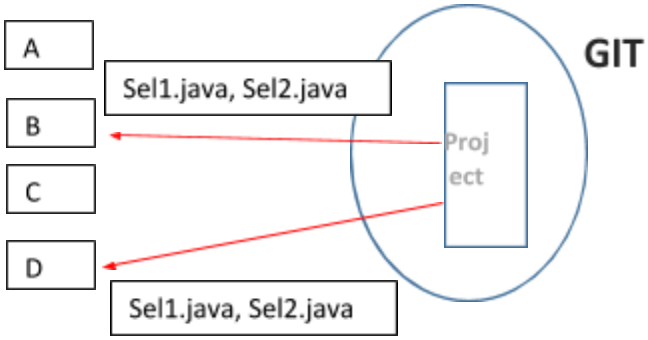

Figure 2

Now 'B' makes changes to both the selenium files. He tests them in his local machine & finds that they are working as expected. 'B' uploads (see Figure 3) both the code files to the GIT & everything works fine on GIT as well. But 'B' does not know that 'D' is also making the changes to the same files.

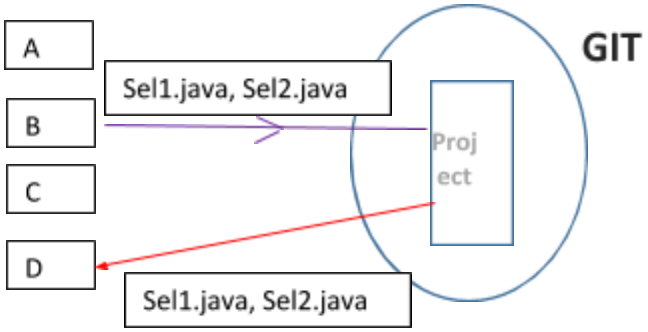

Figure 3

Now suppose 'D' deletes the file Sel2.java in his local machine & everything works fine in his local machine. He now puts his code in the GIT. Suppose there is another file Sel3.java in GIT that is dependent on Sel2.java. Now guess what! ***GIT breaks!!*** There will be compilation error on GIT. The worst part is that no one comes to know about the error (see Figure 4).

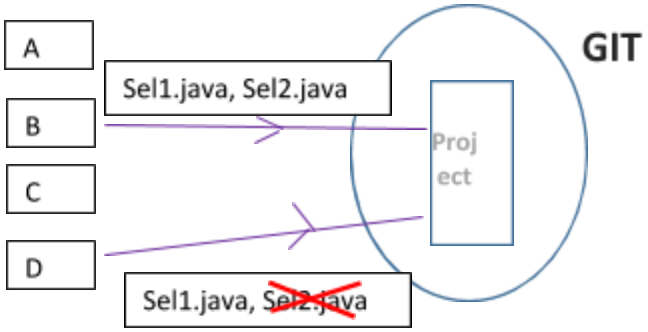

Figure 4

Now when 'C' downloads the code (when he comes on Monday lets suppose) & tries to compile, the compilation will fail in his local machine. So this is the problem with any SCM system, not just GIT (see Figure 5).

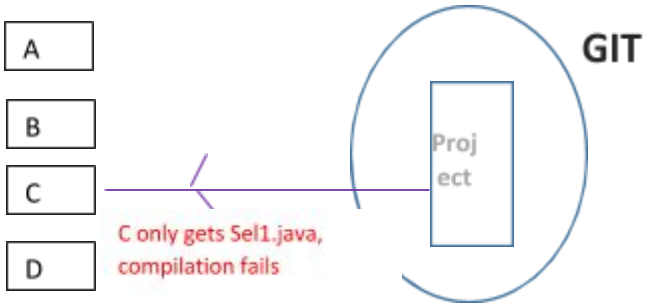

Figure 5

Jenkins comes as saviour:

Now here comes another tool called 'Jenkins'. Jenkins is a continuous integration tool. You can connect Jenkins with GIT. Jenkins will keep on calling GIT in every 5 minutes (example) & it keeps on checking if the code has been changed on GIT. If the code is changed, jenkins immediately builds the code & mails all the team members if build fails, so that all team members come to know that something wrong has been uploaded to GIT (see Figure 6).

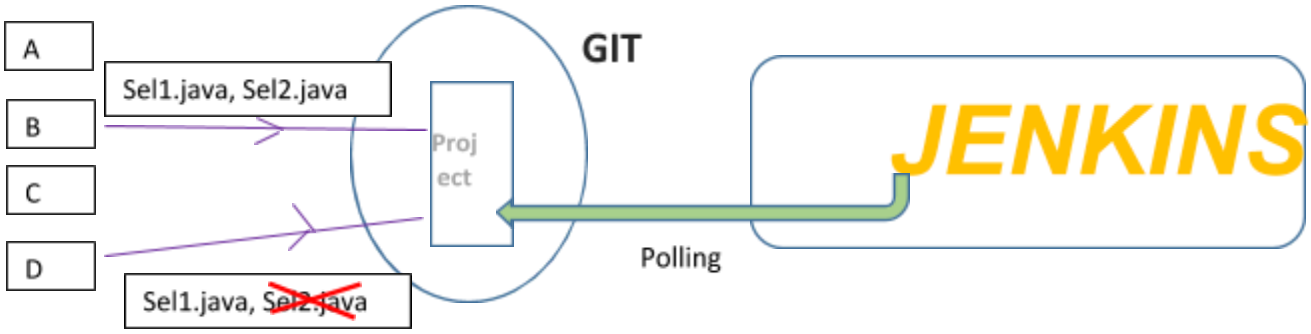

Figure 6

Another advantage of jenkins is that it can be used to schedule builds. Jenkins can help you run your project everyday at specific time, say for example at 3pm

So next we will look at 3 things: 1) How to setup Jenkins 2) How to setup GIT 3) How to integrate both

So let us see how to setup jenkins!

Download Jenkins:

Go to https://jenkins.io/download/ and click 'Download' to download the generic java package *jenkins.war* file in your local machine. Please note, we do not install jenkins, we configure it.

Figure 7

Start Jenkins:

Open the command prompt & go to the location where you have downloaded the war file. Type the command *java –jar jenkins.war* & run

```
>java -jar jenkins.war
```

Figure 8

After a while, jenkins should be up and you should see the below message

```
INFO: Jenkins is fully up and running
```

Figure 9

Jenkins runs on port number 8080 of your local machine

Open a browser, type- localhost:8080 and hit enter, the below screen comes up

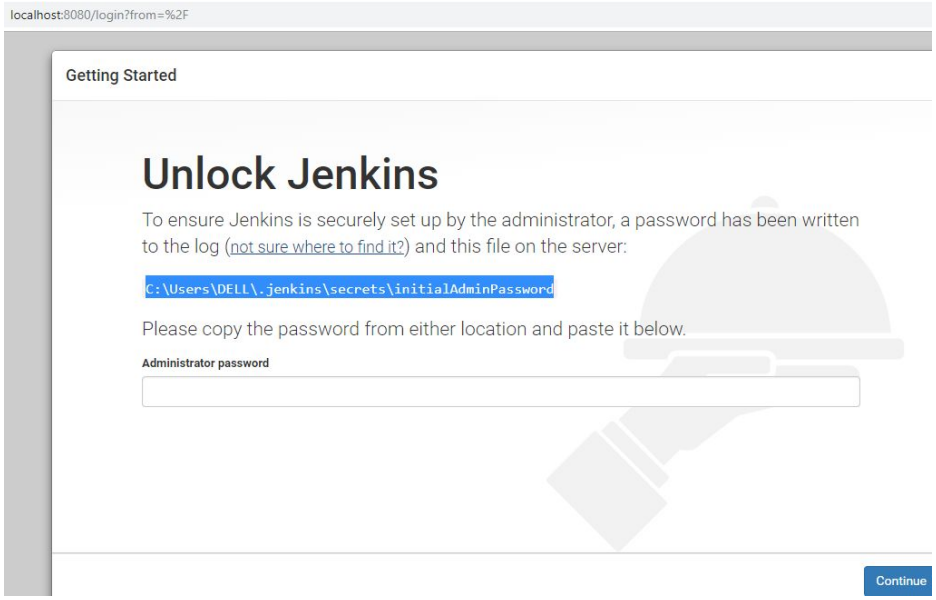

Figure 10

Go to the highlighted location to know the password, enter the password & hit continue. The below screen comes up.

Click 'Install suggested plugins'

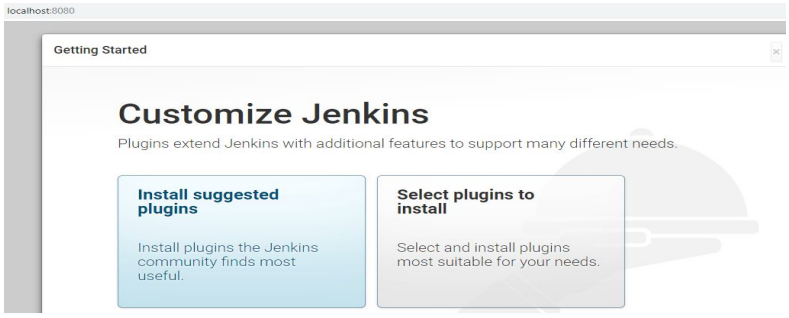

Figure 11

The below screen comes up

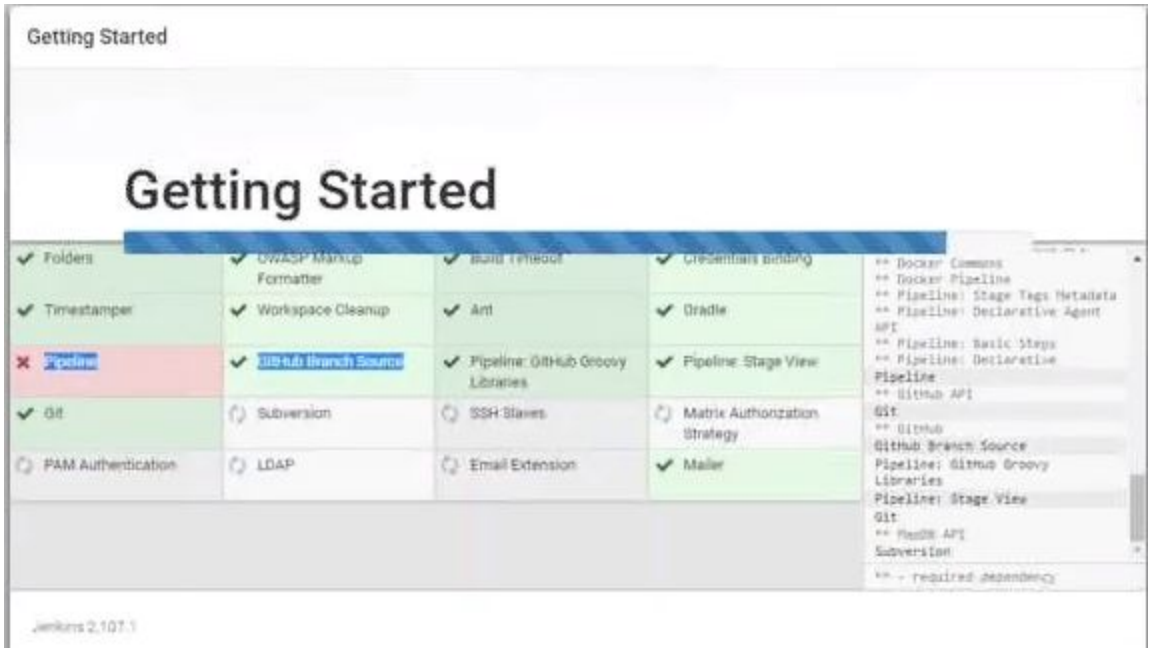

Figure 12

After a while you will see below message

Figure 13

Click 'Start using Jenkins'. If, for some reason the jenkins does not start, than shutdown Jenkins by closing the command line window, re-open the command line window, re-execute the same command, wait for jenkins to be up and running, type localhost:8080, the below screen should come up, enter username as admin & same password that you typed above, hit Sign in'

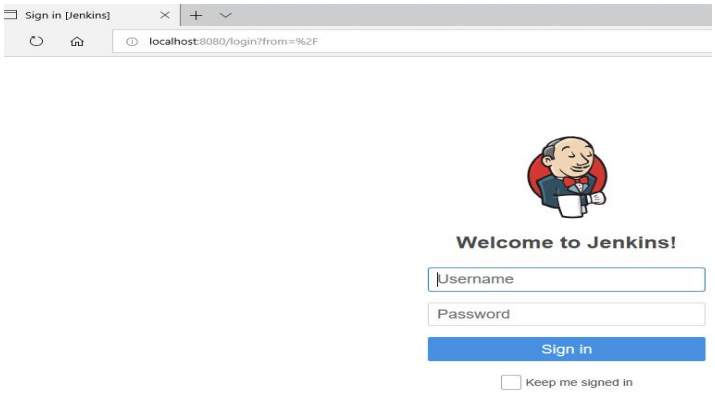

Figure 14

The below window should come up

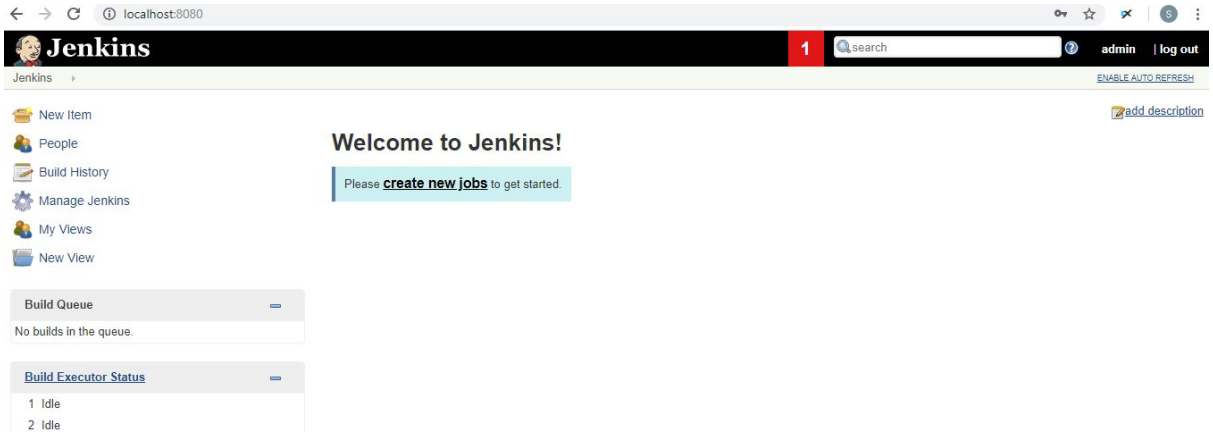

Figure 15

Create dummy project in Jenkins:

Click 'New Item' 

(see Figure 15) to create a dummy project. Enter any name, click 'Freestyle project' >click OK

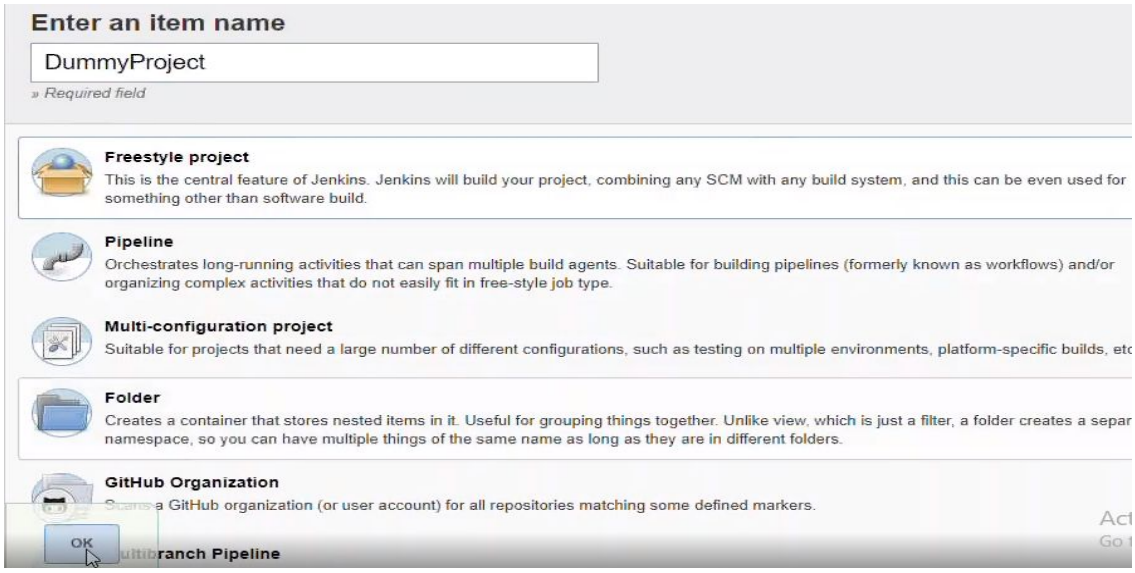

Figure 16

Click 'Source Code Management' tab. You can see 3 options, so basically we have to tell jenkins where our source code is lying. If you select 'None', it would mean that the code is lying on our local machine.

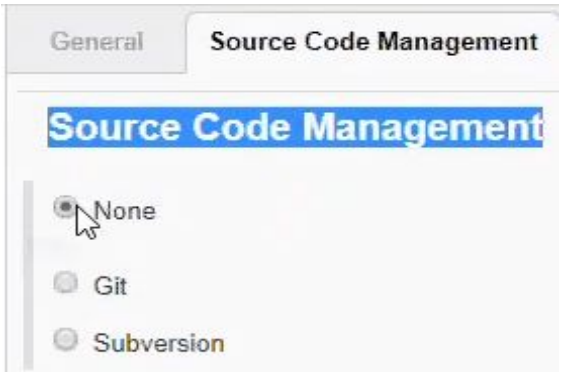

Figure 17

Now come to the bottom of above screen, you will see 'Build' section. Select 'Execute Windows batch command'

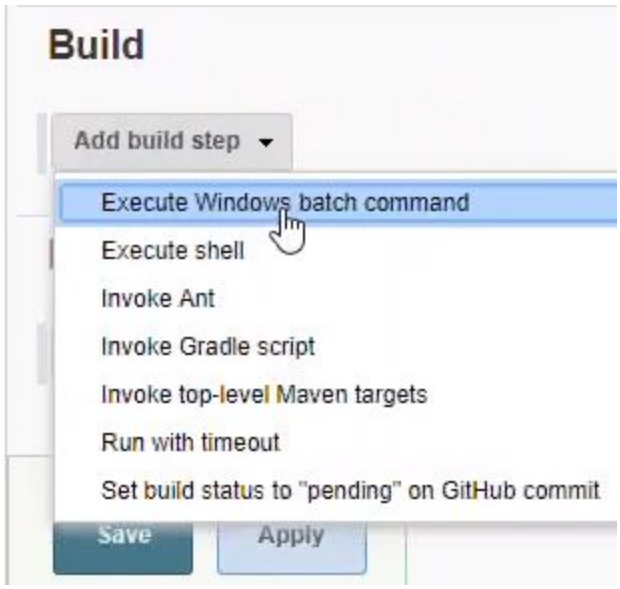

Figure 18

Now write a simple echo command

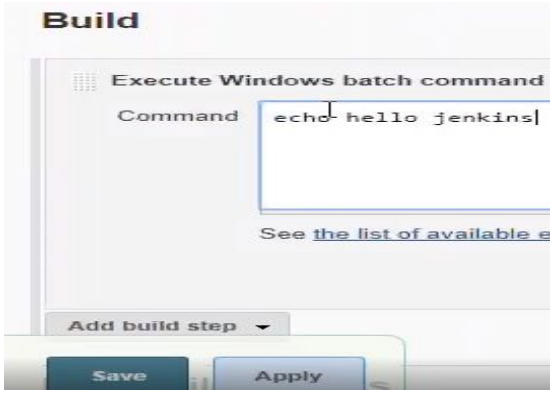

Figure 19

Click Save

So, the run/build command for this project is, 'echo hello jenkins'

Build Project:

Click 'Build Now' to run the project

Figure 20

So under 'Build History' we can see the project getting build (see Figure 21)

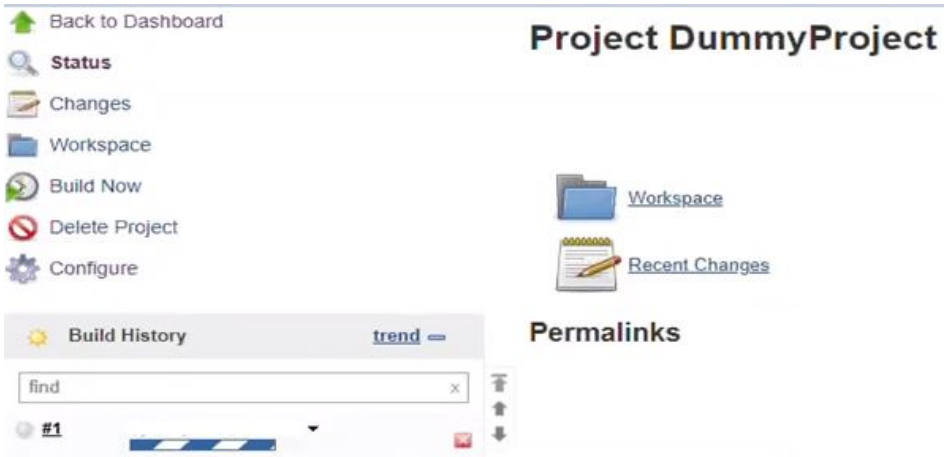

Figure 21

Click on this build project serial number link #1

Figure 22

Click 'Console Output'

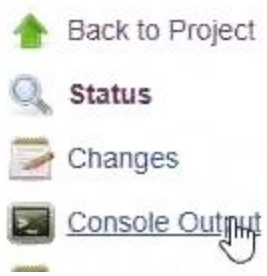

Figure 23

We see 'hello jenkins' getting printed in the console o/p

Figure 24

You can cancel the build by clicking the cross icon 

Figure 25

Click Ok on the popup message to abort the build

Figure 26

Click 'Back to Dashboard' to go back to the project

Figure 27

Configure Project to run periodically:

Click small arrow dropdown (see Figure 27) and click 'Configure'.

Click 'Build Triggers' tab (see Figure 28) & select 'Build periodically' checkbox. Click the help icon shown against the 'Schedule' section. So basically there are 5 parameters that need to be set:
MINUTE, HOUR, DOM, MONTH, and DOW

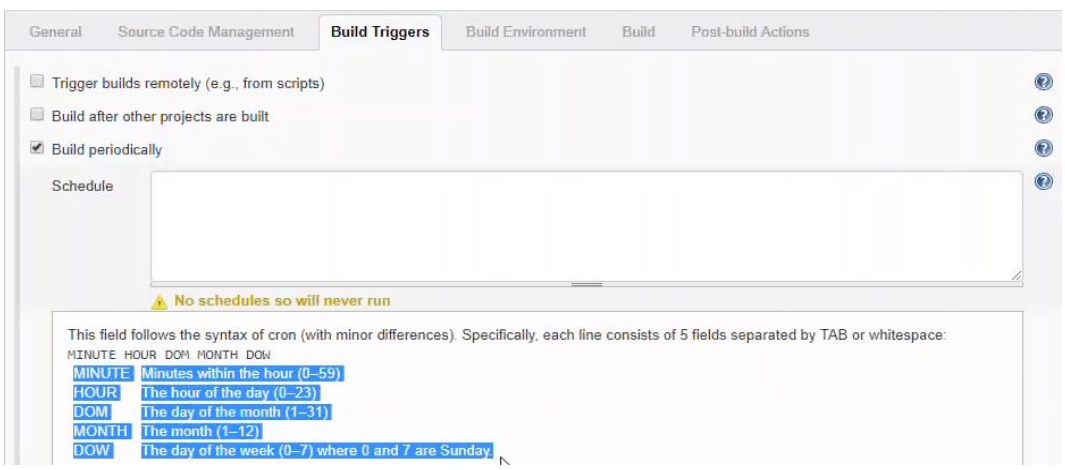

Figure 28

Each parameter can be represented by a star * (see Figure 29)
If we put 5 stars * * * * * (separated by whitespace), this would mean, run the build every minute

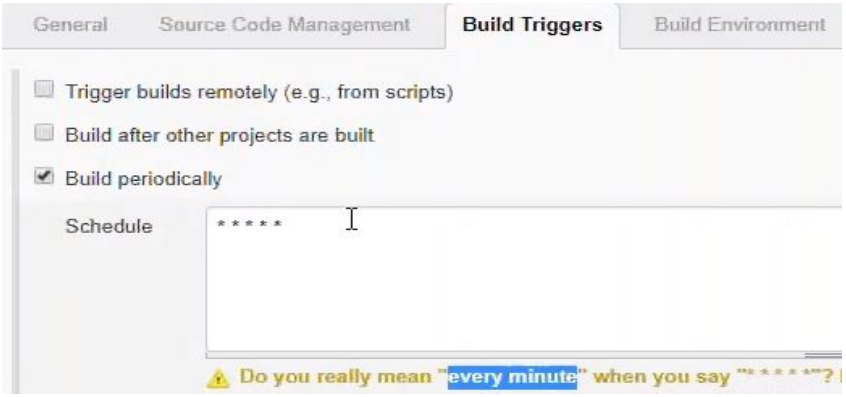

Figure 29

H/30 in the MINUTE parameter would mean: run the build every 30 minutes (note that there is 30 mins time difference between 5:06:35 PM & 5:36:35 PM mentioned below the Schedule section, see Figure 30)

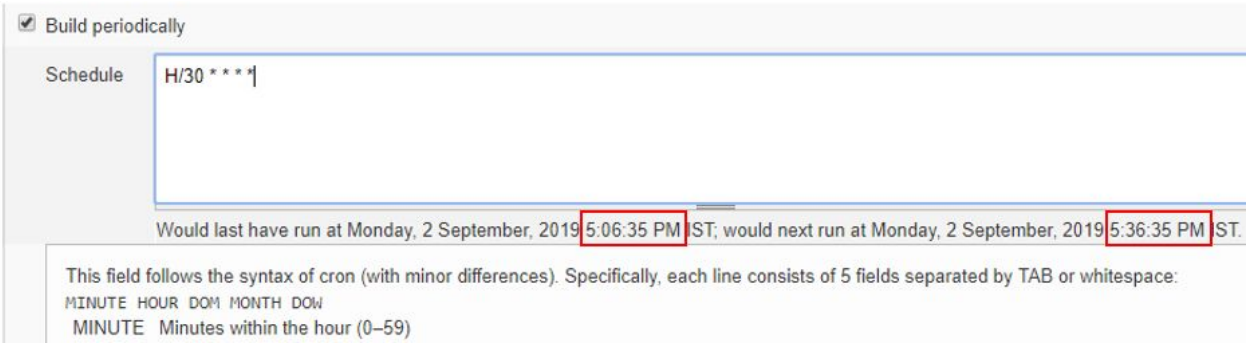

Figure 30

Similarly H/10 * * * * would mean build every 10 minutes

H H/3 * * * would mean every 3 hours since we have mentioned H/3 in the HOURS parameter (note that there is 3 hour time difference between 2:36:11 PM & 5:36:11 PM mentioned below the Schedule section, see Figure 31)

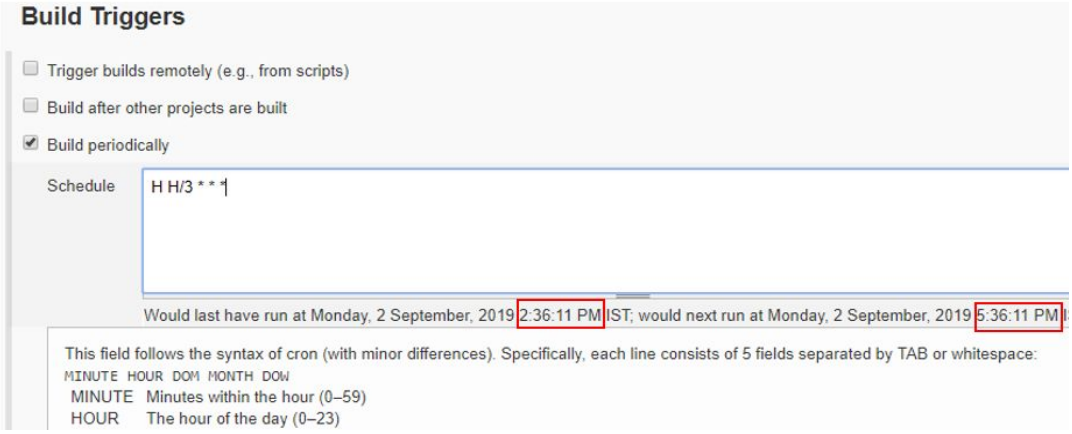

Figure 31

If you save this configuration now, the project will build automatically after every 3 hours!

Figure 32

Note: Jenkins automatically create a workspace for this project & can be seen under the Users/.../.jenkins/workspace directory

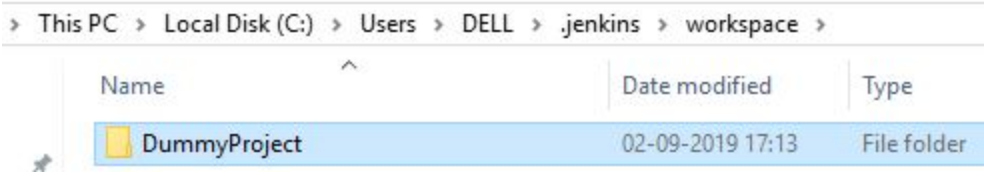

Figure 33

Setup Email Notification through Jenkins:

If you want to send an email, click 'Manage Jenkins' & then 'Configure System'

Figure 34

You will see ‘E-mail Notification’ section

Figure 35

Click Advanced, so you will see additional fields

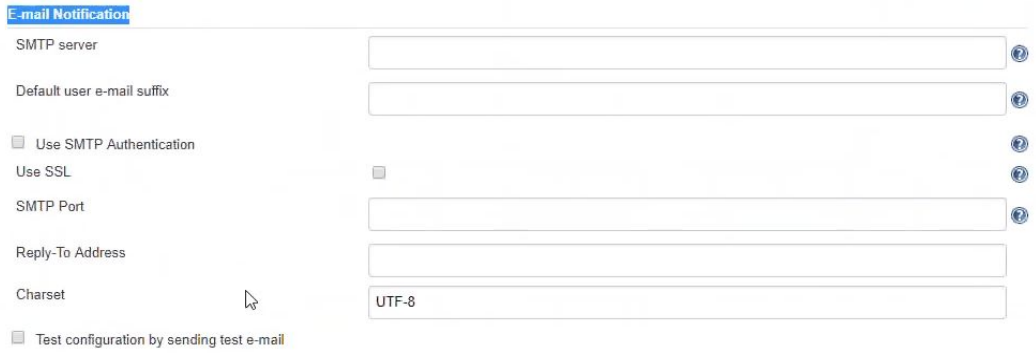

Figure 36

Select the checkbox ‘Use SMTP Authentication’. Populate all these fields (you can check IT admin in your office for the details)

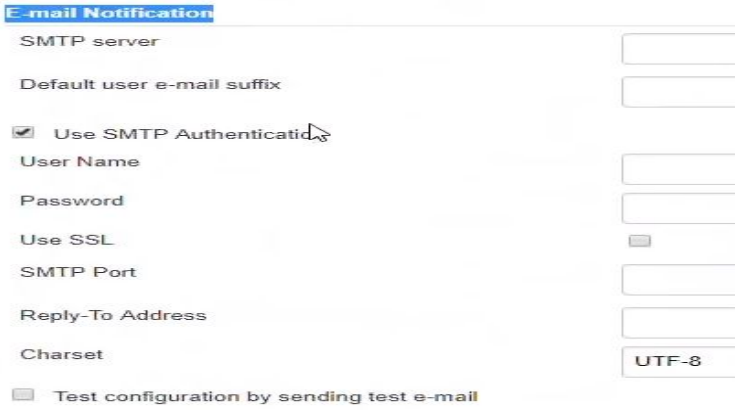

Figure 37

Create a new project (like we had done earlier)

In the 'Post-build Actions' section, you can see a dropdown 'Add post-build action'

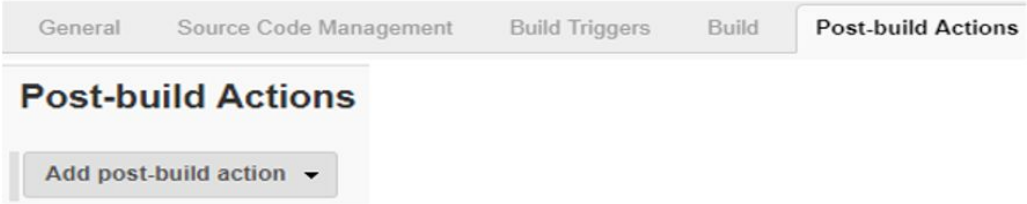

Figure 38

When you click this dropdown, one of the options that you see is 'E-mail Notification' (see Figure 39)

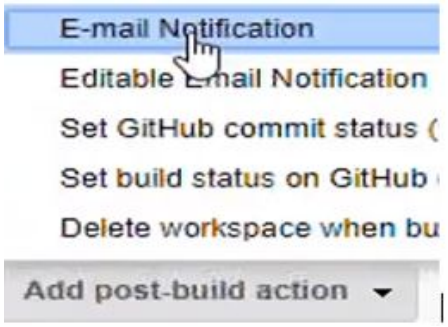

Figure 39

So, once the build is over, the notification will be sent to all the team members (that you mention in the Recipients field, see Figure 40)

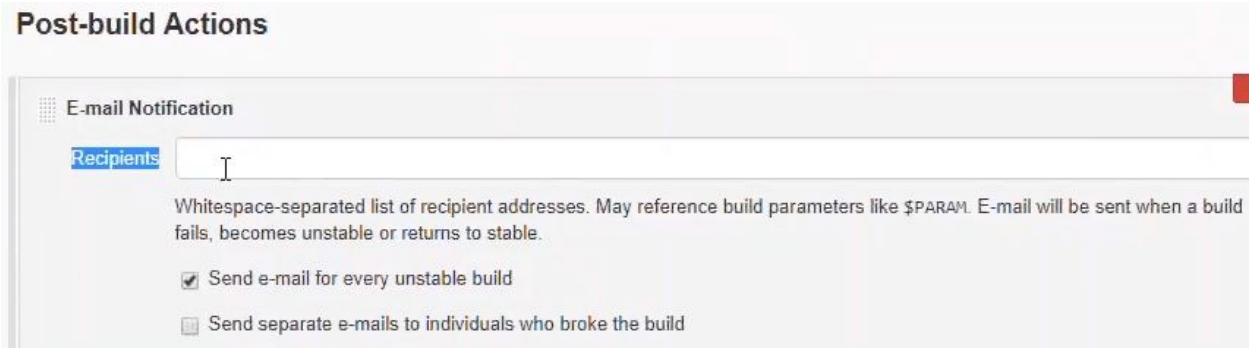

Figure 40

Conclusion

So in this tutorial we have seen the challenges in using GIT as a standalone tool. We have also seen how to setup Jenkins tool & how it is useful as a continuous integration tool which is primarily developed for automation. Here automation means automating daily routine things like building jars, pushing code to website, running automated test cases.
In our next tutorial, we will see how we can setup a GIT system & how to integrate jenkins with GIT.
Thank you for reading!

Recommended Reading

- Jenkins documentation

- GIT
- CVS