**If you have forgotten Jenkins username and password follow below video**

**https://www.youtube.com/watch?v=baAY-agw2fI&index=4&list=LLZlWnpFfBwsMAuKz4iu2fBA&t=0s**

**Jenkins Security**

1. Go to the Jenkins dashboard, usually http://\_server\_:8080 or http://\_server\_/jenkins:8080, where *server* is the host on which Jenkins is running
2. Select *Manage Jenkins*, then *Configure Global Security*
3. Click *Enable Security*. The page will expand to offer a choice of access control.
4. Select *Jenkins’ own user database*
5. Place a check mark next to *Allow users to sign up*
6. Continue with *Authorization*, below. In particular, do not forget to press the *Save* button at the bottom of the page.

**Git Signing in**

1. Launch [www.github.com](http://www.github.com)
2. Sign up by giving username and Password.
3. Enter correct Email ID.
4. On clicking Submit, an Email will be triggered to your specified E mail ID.
5. Verify your email and thus you are signed in to GitHub
6. Click new repository
7. Give name to repository as ExecutionsRep
8. Public
9. Create repository
10. Under …or push an existing repository from the command line save the command. Example is below

git remote add origin https://github.com/KeerthanaBusyQA/Executions.git

1. Install git in your pc from the below url

<https://git-scm.com/download/win>

1. Go to folder in workspace where your pom file is present -> right click ->Git Bash here

Git bash opens that looks like command prompt.

|  |  |
| --- | --- |
| **Commands** | **Description** |
| git config --global user.name “[firstname lastname]” | set a name that is identifiable for credit when review version history |
| git config --global user.email “[valid-email]” | set an email address that will be associated with each history marker |
| git init | initialize an existing directory as a Git repository. .git file is added in the repository |
| git clone [url] | retrieve an entire repository from a hosted location via URL |
| git status | show modified files in working directory, staged for your next commit |
| git add [file] | add a file as it looks now to your next commit (stage) |
| git reset [file] | unstage a file while retaining the changes in working directory |
| git commit -m “message” | Commit code in your local Git repository. Give a message related to what changes you have made |
| git push [URL] [branch] | Transmit local branch commits to the remote repository branch |

1. Git Commands sequence

git config --global user.name “[Keerthana D]”

git init

git remote add origin <url>

git status

git add .

git status

git commit -m “message”

git push origin master

1. Verify new project is added to the repository created in git hub.
2. Click on popular repository in the dashboard and choose the repository to which the code is uploaded.
3. Verify if all the files are uploaded.

1. To download from git to your pc –
   1. Click and open the repository you have to download
   2. Click clone or download button
   3. Copy the link that appears in the small window that opens. Ex

<https://github.com/KeerthanaBusyQA/Executions.git>

Notice that it should end with .git

* 1. Open git bash where you want the repository to be downloaded and enter the below command

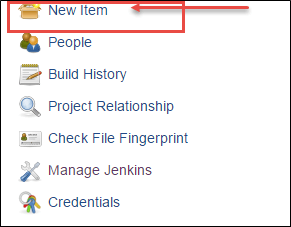
Git clone https://github.com/KeerthanaBusyQA/Executions.git

1. Integrating Git to Jenkin

Launch browser and open Jenkins.

Click Manage Jenkins -> Configure global security -> Give path to git Executable on your system

Click on new Item.



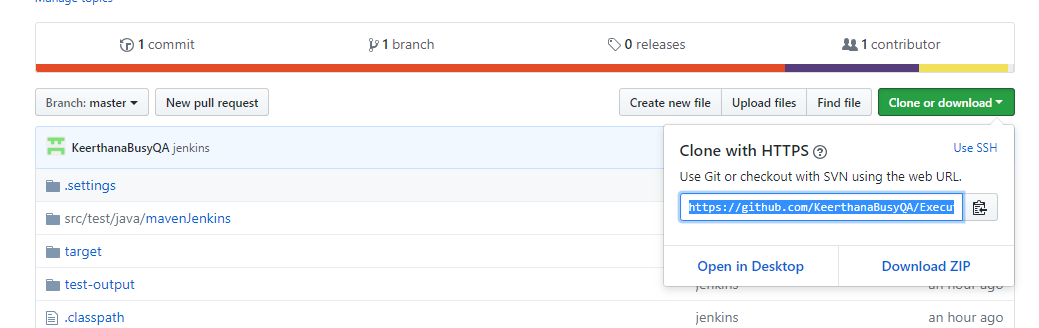
In this step,

1. Enter Item name
2. Select Maven Project
3. Click on ok button

In this step, we will configure Git Hub in Jenkins

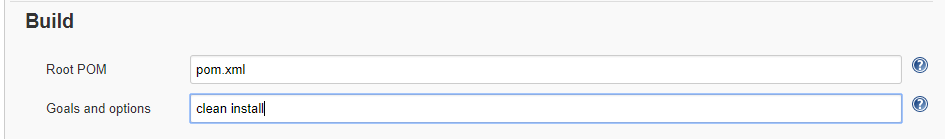
1. In configure page scroll down to real source code management
2. Click on Git and
3. Enter the Repository URI
4. Click on Add repository

We can get the URL in Git Hub



In this step,

1. Add the pom.xml file location in the textbox and
2. Specify the goals and options for Maven as clean install
3. Select option on how to run the test
4. Click on save button.



once we click on save below screen will appear,

Now we can build our project click on build.

From Build Triggers, If you want Jenkins to monitor the repository and start a build whenever any changes have been committed, We can choose to pick the Poll SCM option and enter syntax of cron.

The other options include Build periodically (for example, once a day, ), Build whenever a SNAPSHOT dependency is built etc.

Here "SNAPSHOT" means build is is still under active development. Ex;- If it is easy-1.0-SNAPSHOT.jar library, Maven will know that this version is not stable and is subject to changes. Before 1.0 release (or any other release) is done, there exists a 1.0-SNAPSHOT version.

The difference between a "real" easy-1.0 version and a easy-1.0-snapshot version is that snapshots might get updates. That means that downloading 1.0-SNAPSHOT might give you a different code than downloading it yesterday or tomorrow.

*Builds Periodically* will trigger builds as per the schedule (If we specify H/5 \* \* \* \*, every 5 minutes) even if you haven't changed anything. *Poll SCM* will check for changes before triggering any build, if there are changes to the previous version then only the build will be triggered.

Both these fields follows below syntax of cron which consists of 5 fields separated by TAB or white space: -

MINUTE (0-59), HOUR (0-23), DAY (1-31), MONTH (1-12), DAY OF THE WEEK (0-7)

DAY OF THE WEEK - The day of the week (0–7) where 0 and 7 are Sunday.

**Example:-**

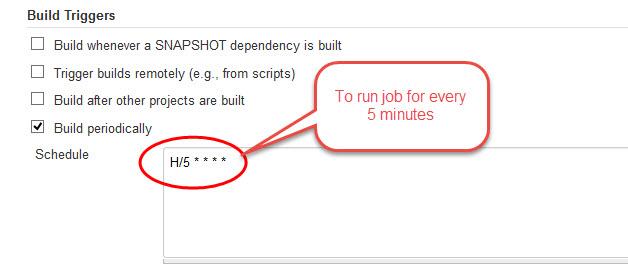
# If you want to run job for every fifteen minutes, we should specify cron syntax as below :

H/5 \* \* \* \*

#If you want to trigger for, every 2 hours

H \*/2 \* \* \* to distribute load evenly throughout the hour

In addition to the above, Jenkins also support convenient aliases as @yearly, @annually, @monthly, @weekly, @daily, @midnight, and @hourly.



Click save

Go to *Manage jenkins -> Configure System -> Git installations* add there the git exe path (for example: C:\Program Files\Git\bin\git.exe), or you can use environment variable.