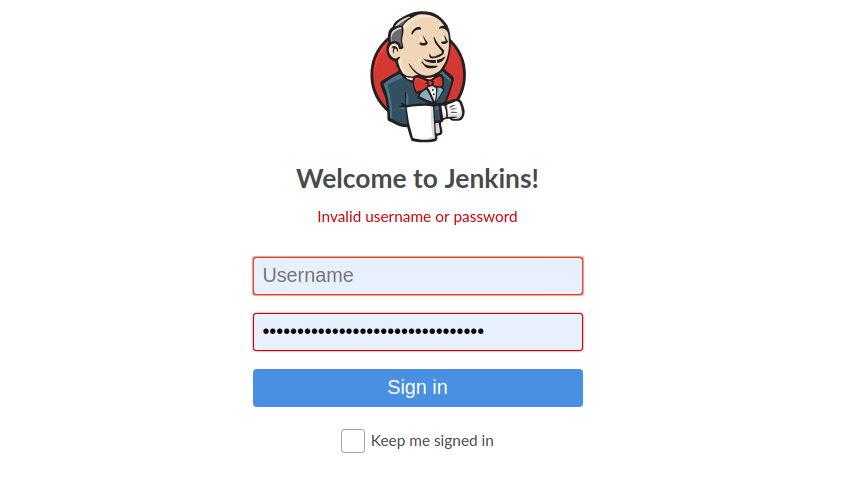
**Jenkins**

1. Installing Jenkins

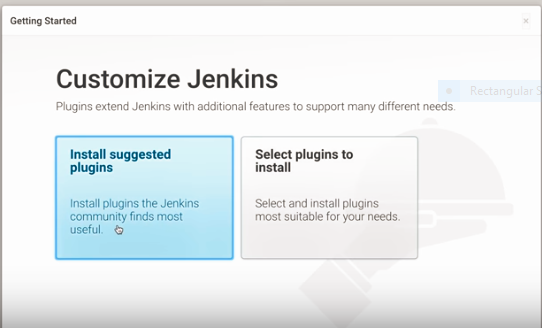
**Link:-** [**https://wiki.jenkins.io/display/JENKINS/Installing+Jenkins+on+Ubuntu**](https://wiki.jenkins.io/display/JENKINS/Installing+Jenkins+on+Ubuntu)

1. Unlocking Jenkins

* When you first access a new Jenkins instance, you are asked to unlock it using an automatically generated password.
* Browse to localhost:8080 (or whichever port you configured for Jenkins when installing it) and wait until the Unlock Jenkins page appears.
* From the Jenkins console log output, copy the automatically-generated alphanumeric password (between the 2 sets of asterisks).
* On the Unlock Jenkins page, paste this password into the Administrator password field and click Continue.

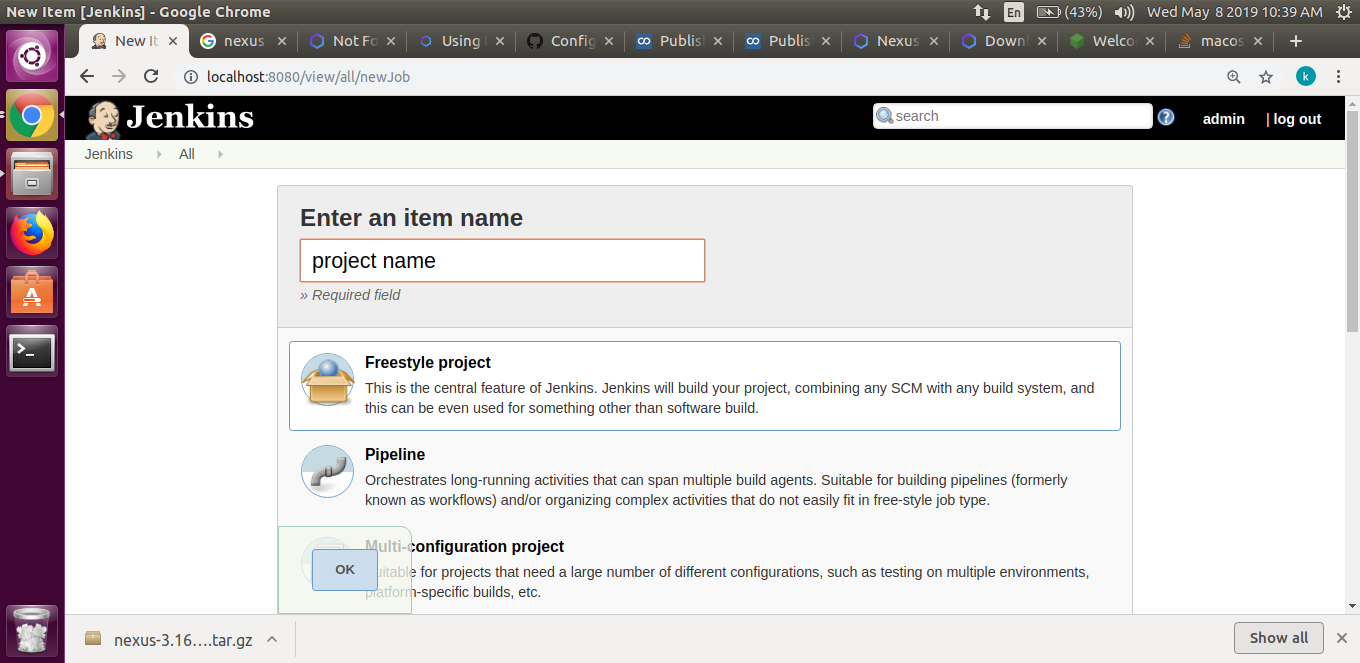


**NOTE:-**After entering the password it might take some time as this is the first time we are accessing our Jenkins



**NOTE:-** we select the **install suggested plugin** if we want to download all the plugin suggested by Jenkins(default) OR we can also install selected plugin by selecting the **select plugins to install** based on our need.

1. Creating Freestyle project



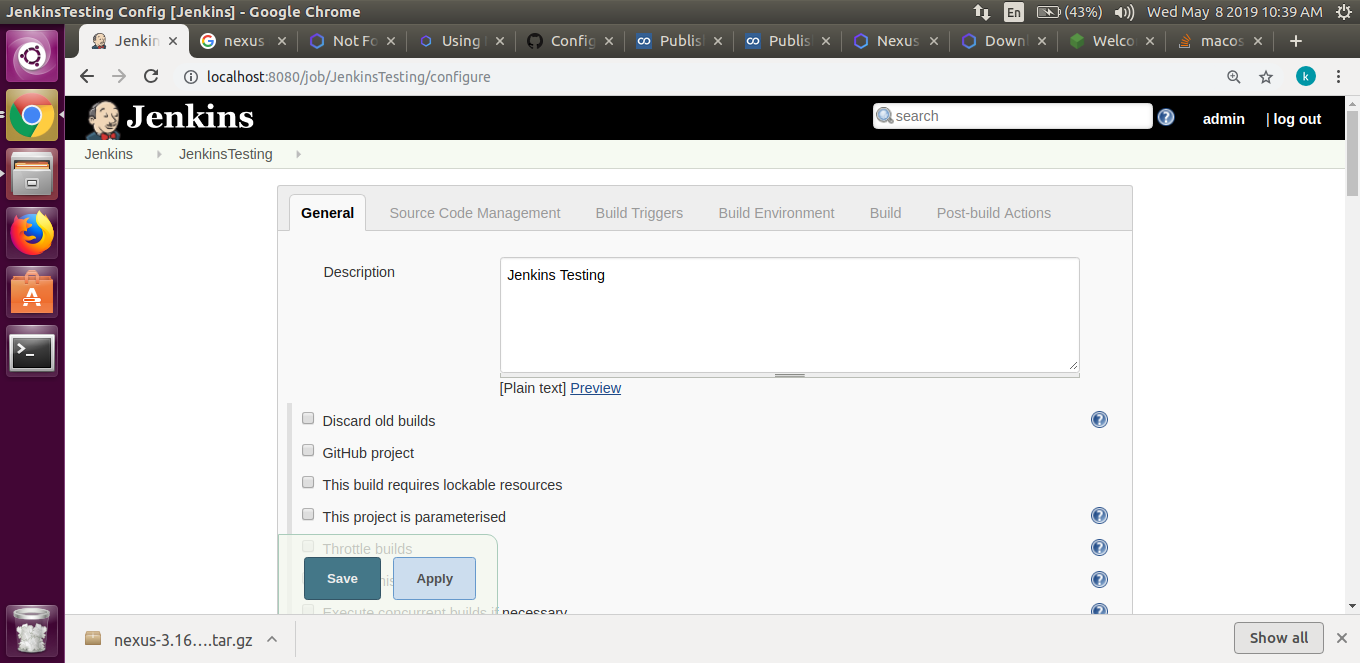
Click one of the two options shown:

• Enter project name(name of your own choice for your project)the name used here will be used by jenkins to differentiate between all jobs on Jenkins dashboard

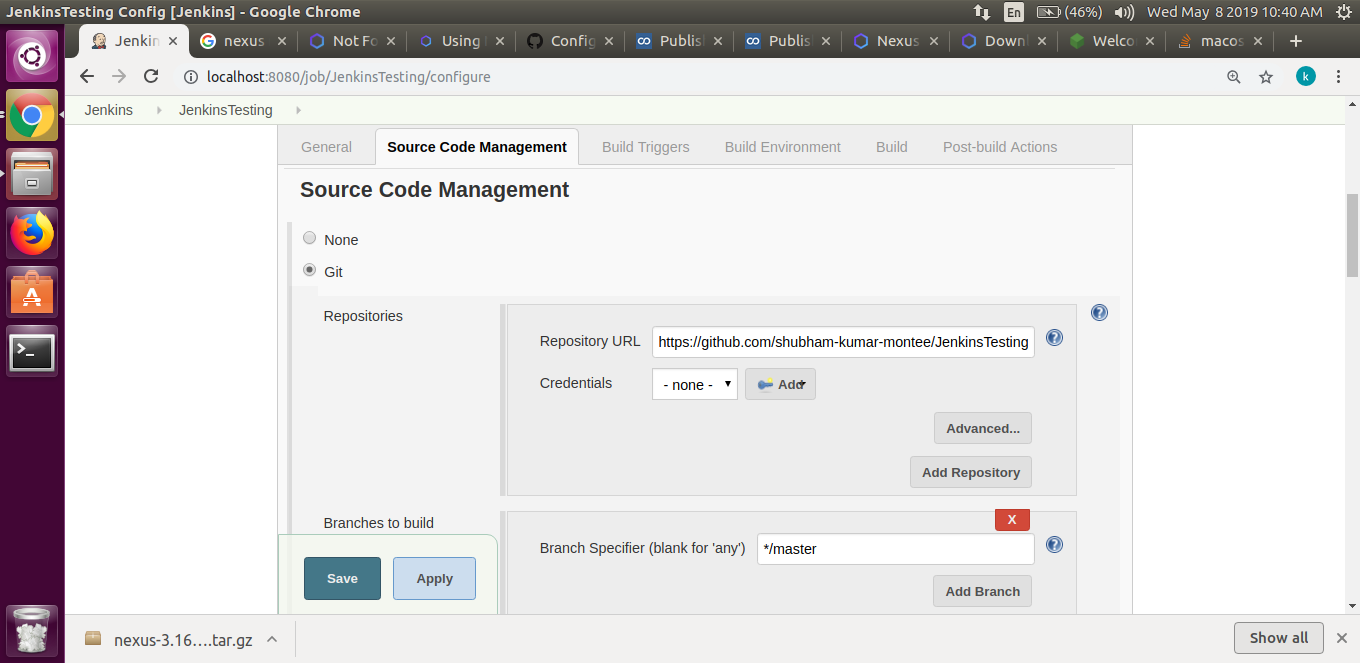
• Select the type of project you want to create (we for now have selected the free style project)

• After selecting the type of project we can hit the ok button as seen in the image above.

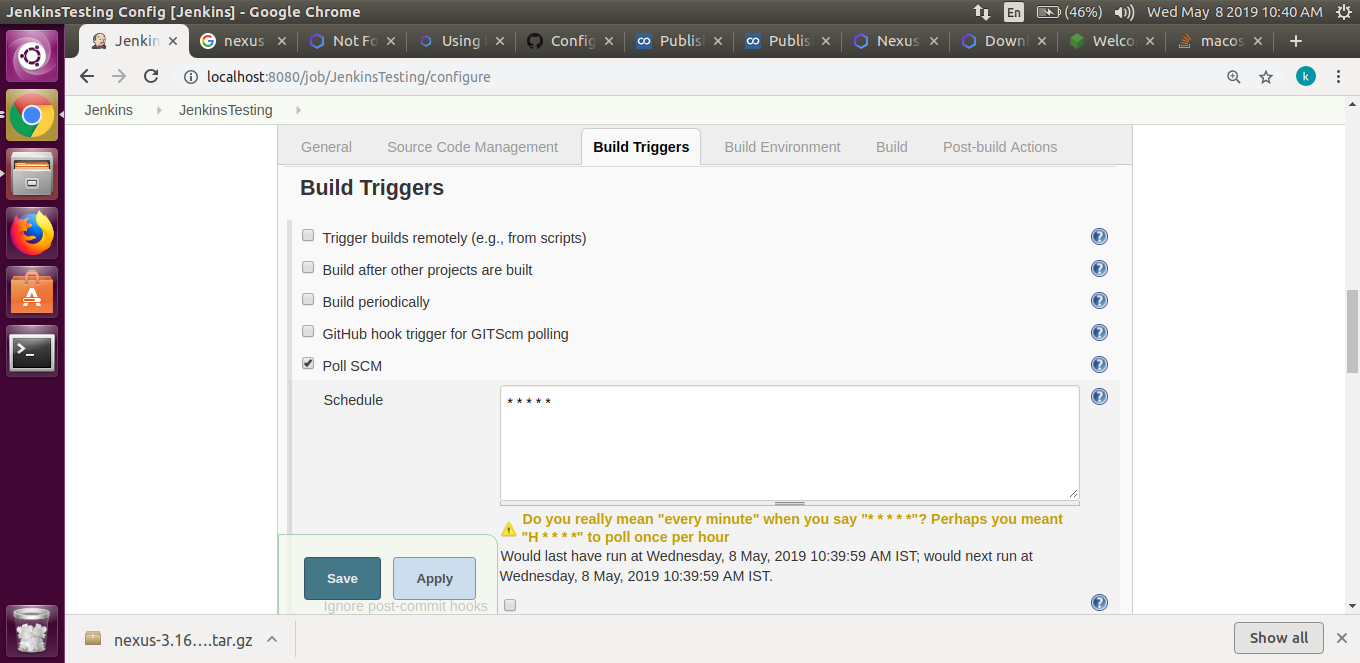
1. Configuring Freestyle project



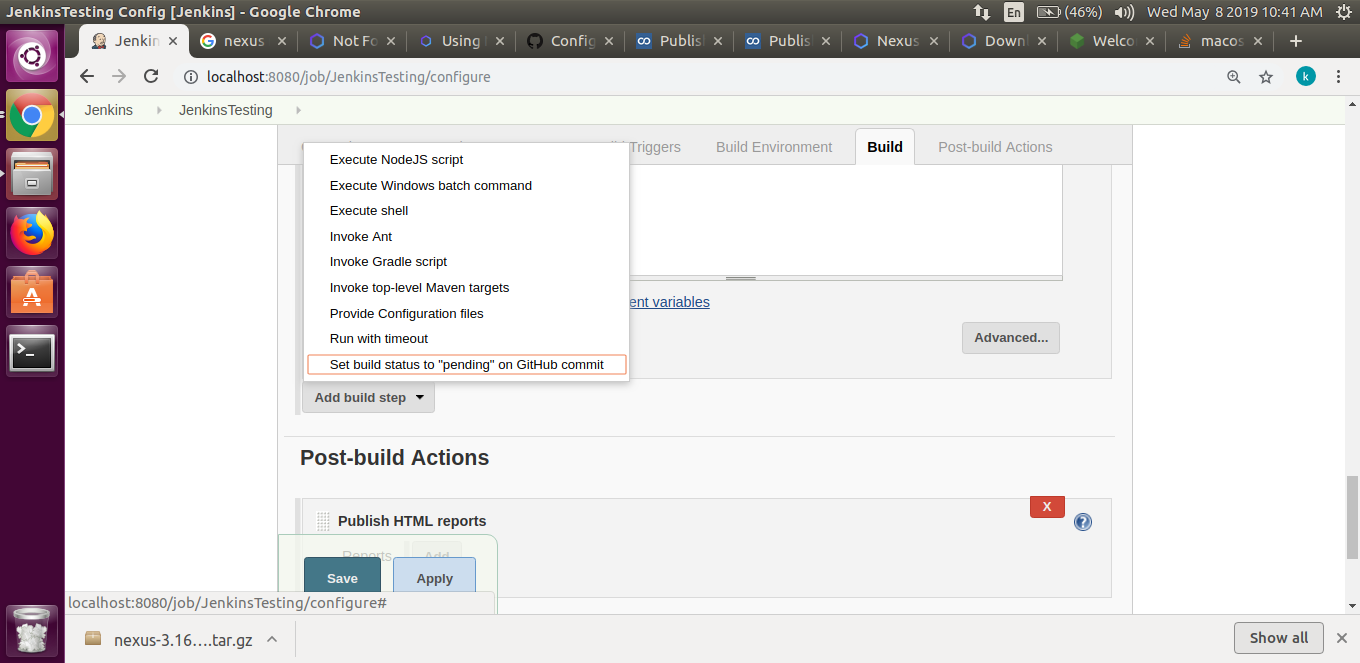
•Give the description of the project (any meaningful description related to the project



•In **source code management** we opt the option **GIT** and then give our repository url from where we want Jenkins to pull our code for build.

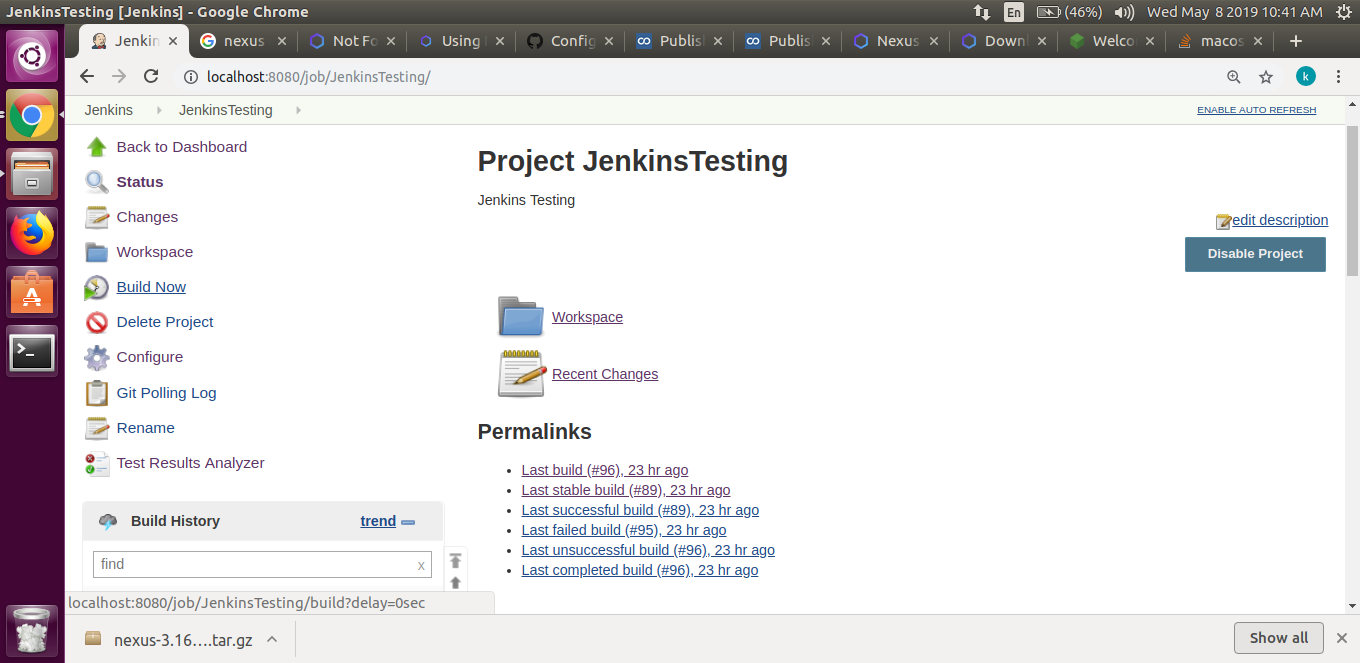


•In build trigger we give the trigger option (**i.e** when do we want to trigger the build here we have opted (**\*\*\*\*\***) which means we want to trigger the build every minute).

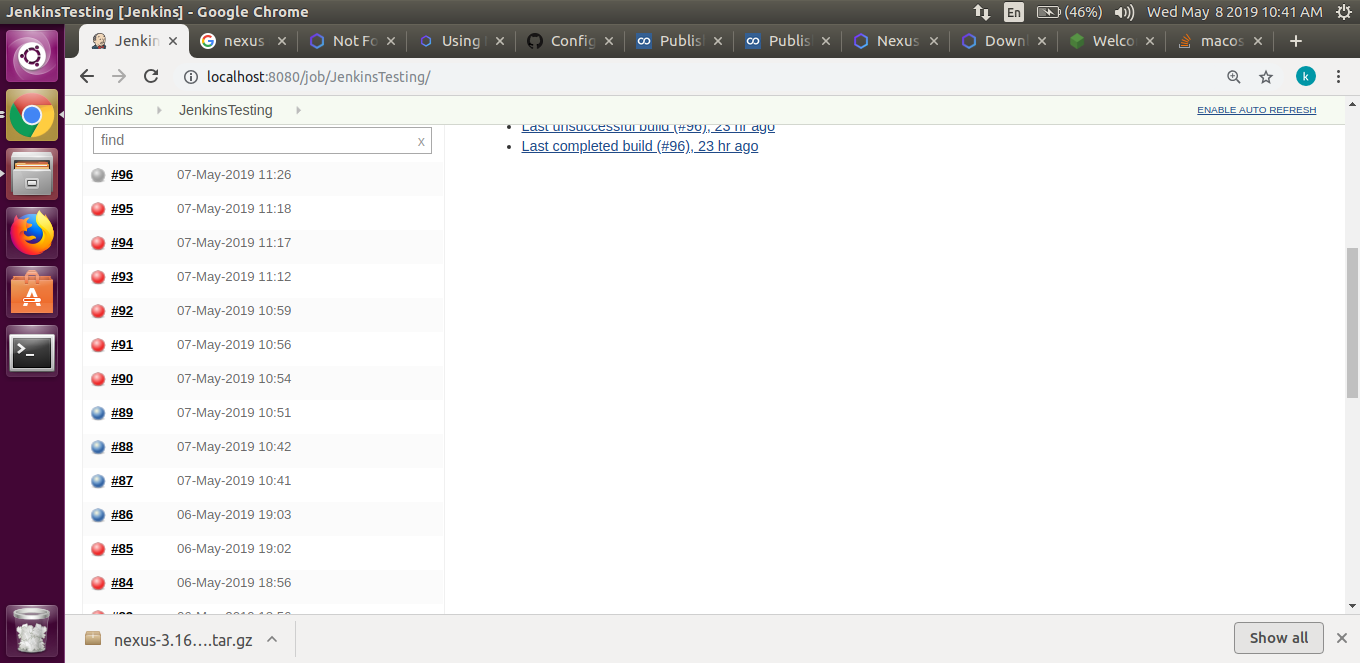


•In build we select the build option like **on windows** we select the option Execute windows batch command **on Ubuntu** we opt for Execute shell .

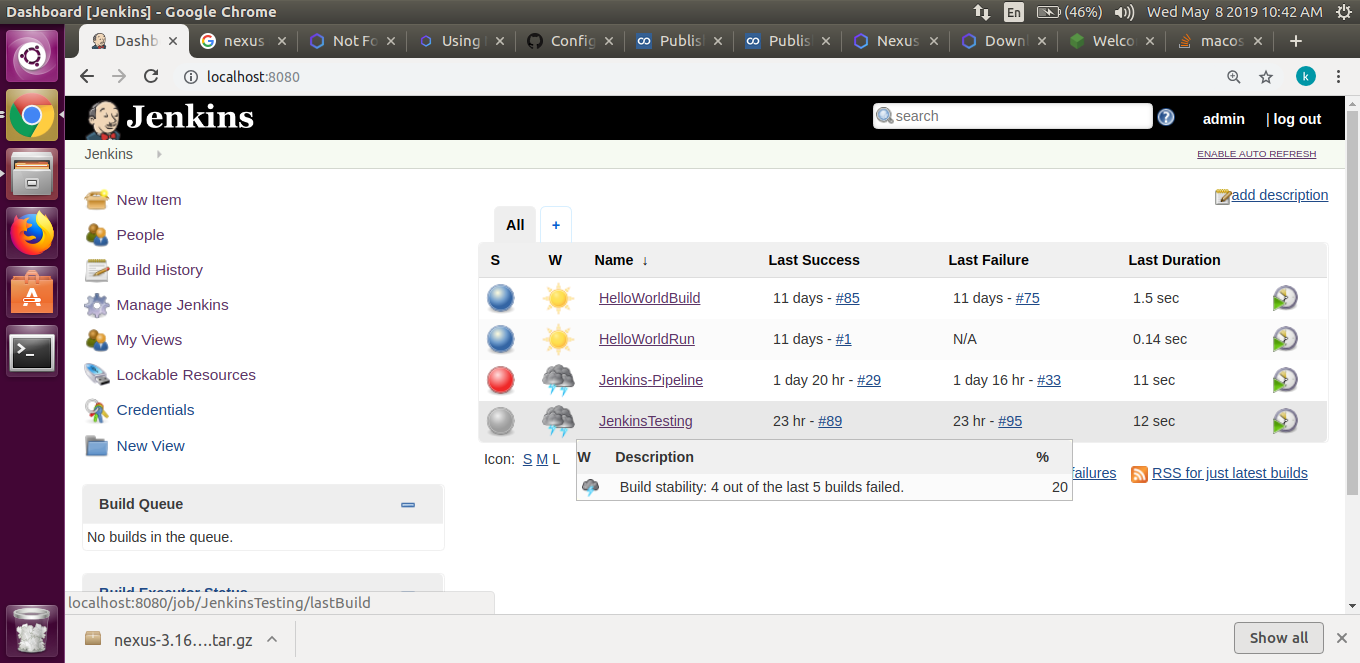
•Then we write the command we want to run in the field and click on save/apply.

****

•The build happens based on the condition we provide in the **poll** SCM but if we want to forcibly trigger the build we can do it by the **Build** option shown above in the diagram.



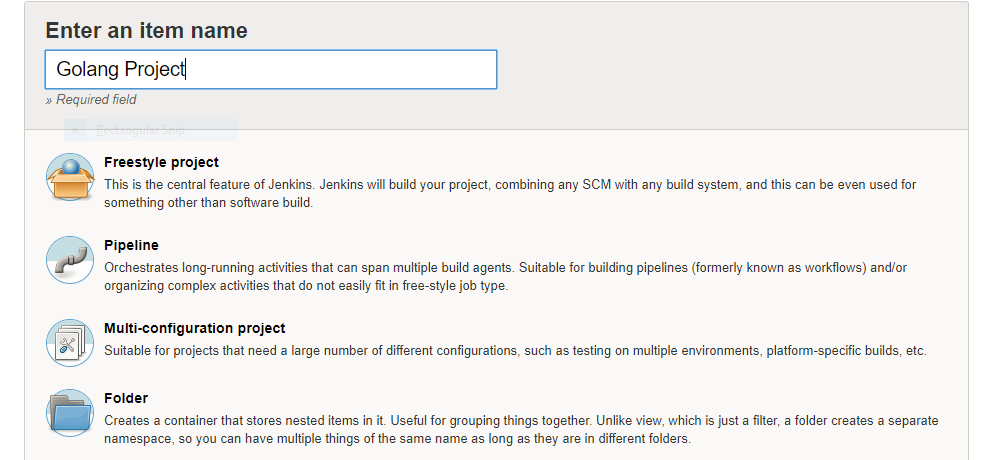
•List of all the build can be seen here **Red** represent build failure and **Blue** represents successful builds



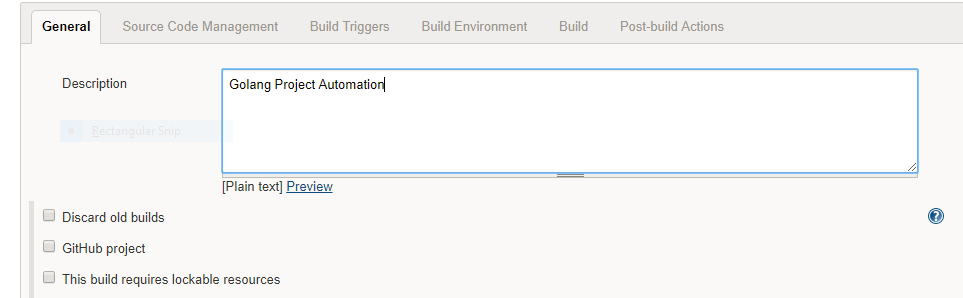
•List of all the project we have created on Jenkins can be viewed on Jenkins dashboard

1. Objective 1. Automate **Golang** Project (**jenkins**)

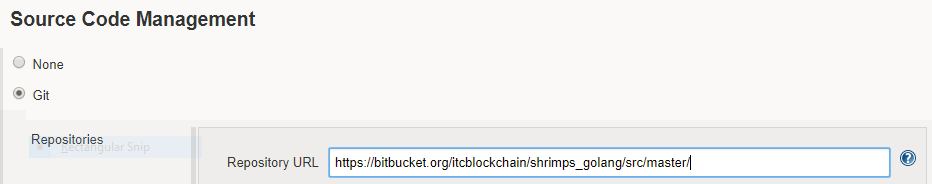
* Install the **golang** plugin
* And then follow these steps



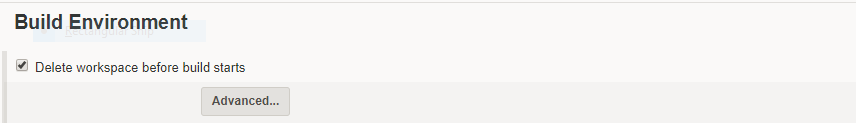
•First we give name to our **golang** project



• Give some meaningful description for the respective project.



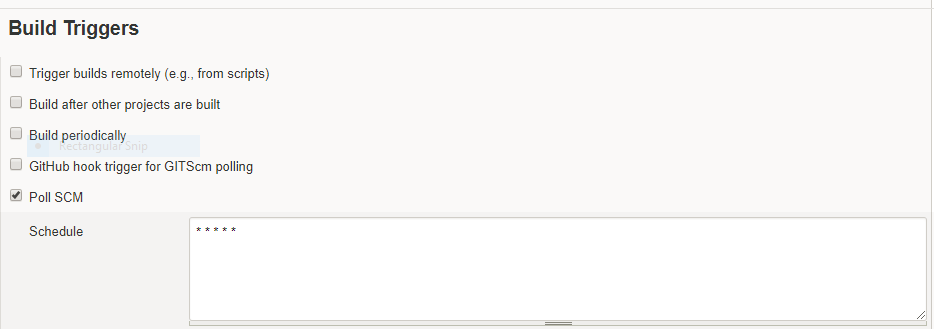
• In **source code management** select the **GIT** radio button and enter the **URL** of your project in Repository url



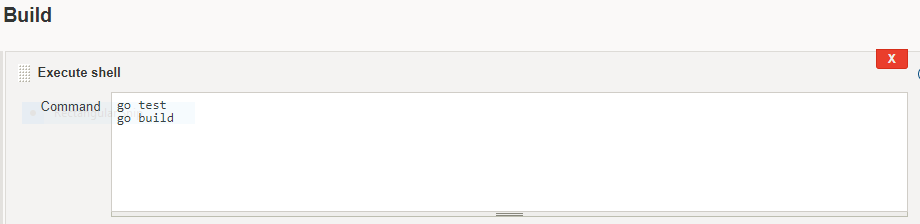
•We setup the build environment.

•Here we have opted to **Delete workspace before build starts** .

•By checking this option we delete the older workspace every time a new build starts.



• We Check in **poll SCM** and mention how frequently we want our build to happen

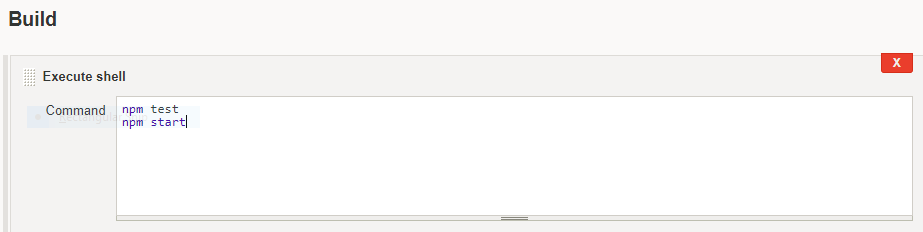


•In our execute shell option for our **golang** project we run two command go test and go build

* Go test :- to run the test script every time before **go build**
* Go build :- to build our **go source code** if test is successful(previous command go test)

1. Objective 2. Automate **Nodejs** Project (**jenkins**)

* Only one change in order to run the node test is to we have change the build step and change the command in the **execute shell**

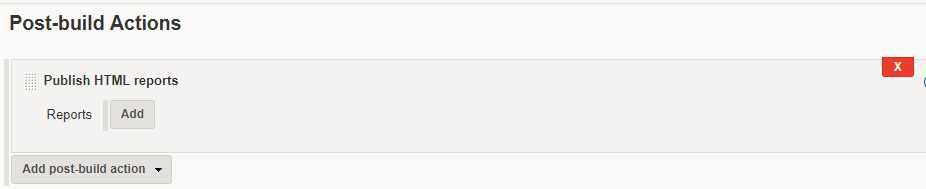


• In our execute shell option for our **Nodejs** project we run two command npm test and npm start

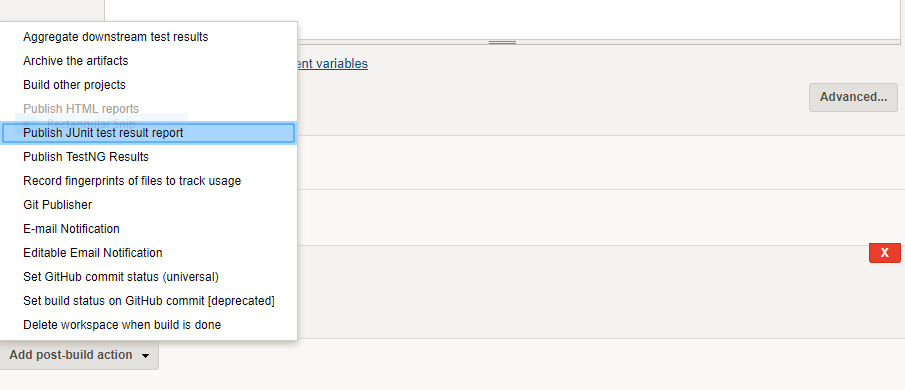
* Npm test :- to run the test script every time build is triggered takes place
* npm start :- to run our **node js app** if test is successful(previous command go test)
* **Npm start** is optional only if you want to run you project (not a good practice to use this command in jenkins).

1. Objective 3. Generate Report

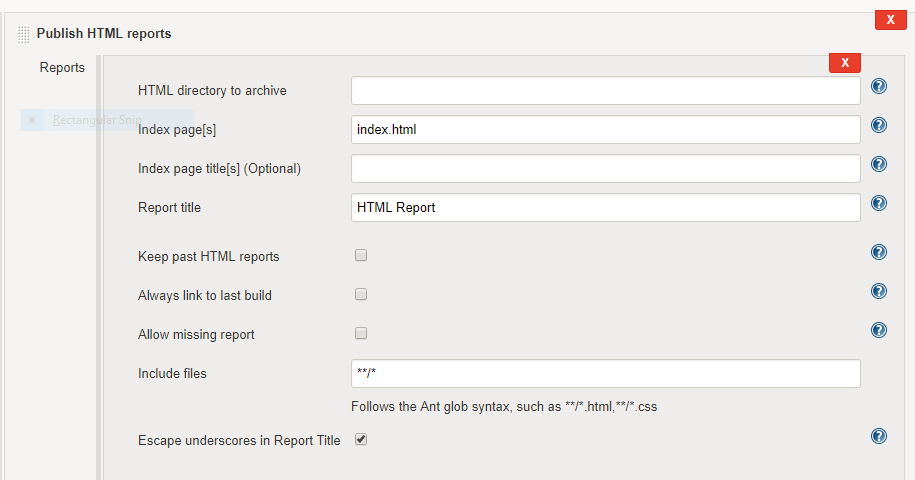
* Install the plugin according to your requirement such as :
* Install the **HTML publish** plugin to generate a html report
* Install **cobertura** for generating code coverage
* Similarly we can install plugins for **test analysis**
* After installing the plugin follow these step



* Click on the **Add Post-build action**
* After click we get a dropdown as folllows



* We can select any of the options we require
* For example
* **HTML publish** to generate html report
* **Publish Junit test result report** for test analysis report
* **E-mail Notification** too receive an e-mail on every build whether successful or failed

****

* In order to publish any report we have fill the following setting in the respective report setting after adding post build option
* like the path of the directory
* the name of the **html** page
* the title of the **html** page
* the title of the report page
* by checking the **keep past HTML reports** we specify that we need past report as well as the current report