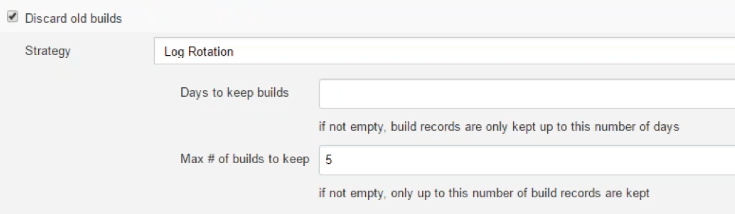
**Job creation:**

* Free style job in Jenkins can be used to create any type of project
* All the type of projects has templates, but free style project contains general package. We can create it as any kind of job

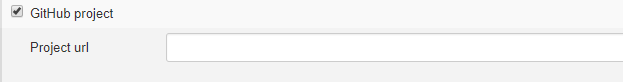
**Discard old builds**

* **Discard old builds** is the feature used to manage the old builds, we can use to keep last no.of build or the builds of last no.of days
* This is just to save the space



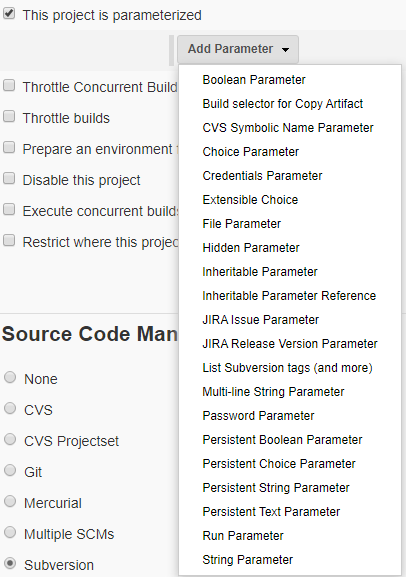
**GutHub project:**

* If we have **GitHub project**, we can add it. This is optional, anyways we can add it even at the source code management age also

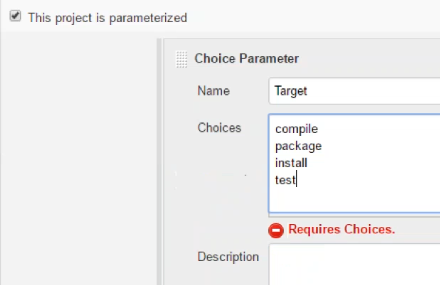


**Parameterized:**

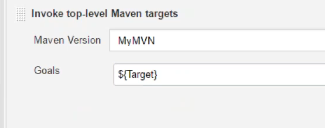
* This project is **parameterized** is the one more important one, we can set parameters with these



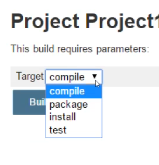
* we can make sure of the parameters based on the requirement. Below is the one example for **choice parameter**



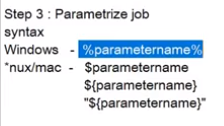
* After setting it, we need to set build action as below



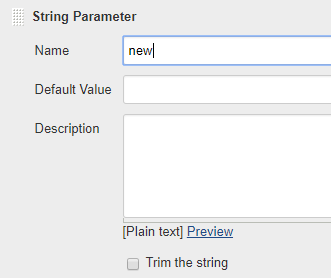
* Then while building the project, we could see like this
* If we configure the choice for automation build, it will pick the first command



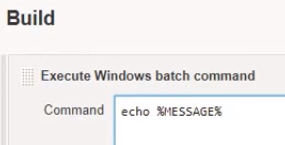
* We can use the parameters in different formats as below



* **String parameter** can be used for declaring any value as below

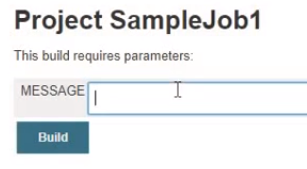


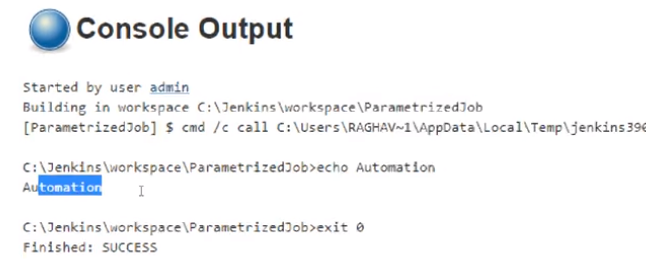
* Give the command as below



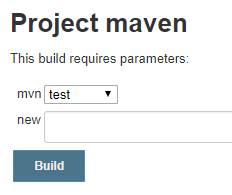


* When we ask to build with parameter, we can get the string which we have declared as below. And here we can type whatever we want

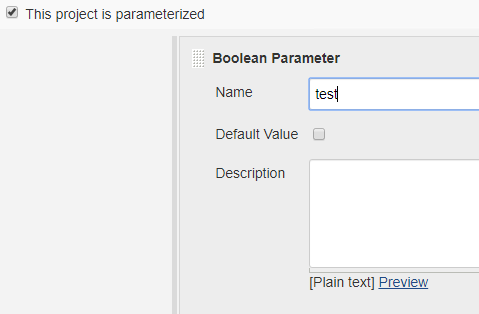




If we set two parameters, we can get as below



* **Boolean parameter** looks like below. If we check it then the “test” value will be true as per below example.

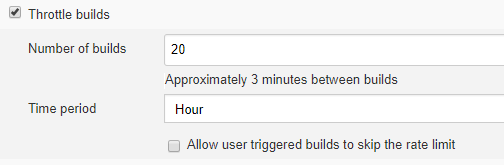


* We can also set the default value as true always



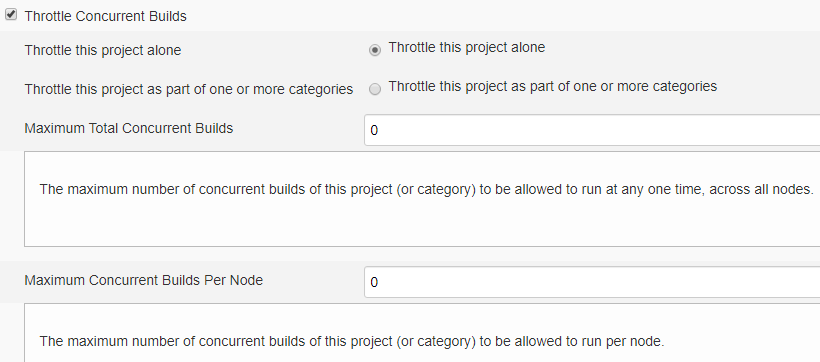
**Throttle build:**

* **Throttle build** is one more feature to restrict the builds, with the below example, we can perform only 20 builds in an hour, it won’t allow 21st build
* It is not much useful feature



**Throttle concurrent builds:**

* **Throttle concurrent builds** allows to restrict the number of concurrent builds to be allowed to run per node and across all the nodes as below



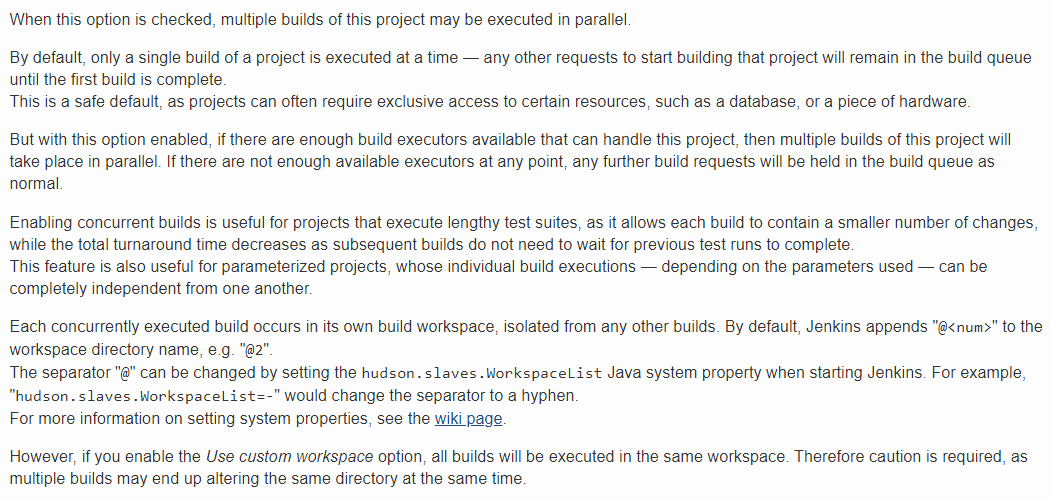
**Concurrent build:**

* **Concurrent build** is useful to build more than one build at a time with by using the executors in job



**Execute concurrent builds if necessary:**

* **Execute concurrent builds if necessary** is an option to build the project for multiple times if necessary

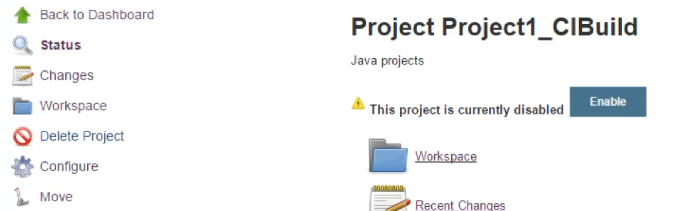


**Disable the project:**

* **Disable this project** is a simple option, with this we can disable the project



* After this, we can’t see the build option



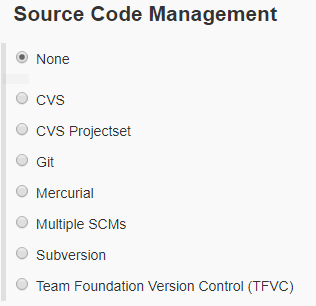
**Custom workspace:**

* In job configuration, click on advanced and select use **custom workspace** to change the workspace directory

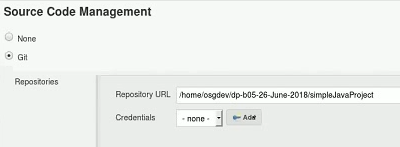


**Source code management:**

* We can use any of these SCM to pull the code from



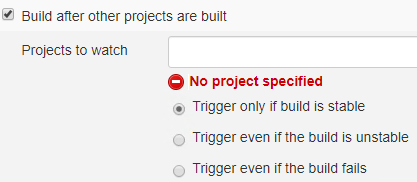
* We can use it as below



* Instead of giving local repo, we can give remote repo url
* We also need to add the credentials. We mush have installed git in Jenkins server or in global tools to use this

**Build after the other projects are built:**

* We can set the build after other projects built as below. This is also called as upstream and downstream projects



**Poll SCM:**

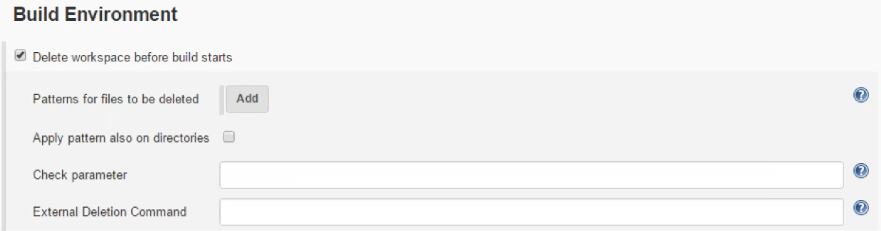
* To build automatically, we need to use poll SCM, it checks the SCM
* So, if we made any changed in source code, it will trigger the build
* We can schedule it just like the cron job
* For example, if we have scheduled it as \* \* \* \* \* which means 1 min. so, it checks every 1 min once and if any change happened then it triggers automatically
* If we configured remote repo, then it triggers when we push the code to remote
* If we configure local repo, then it will trigger when we commit the code

**Build periodically:**

* Instead of poll SOM, we can use build periodically option, we use same method to configure this also
* Here it won’t look for the changes, it triggers as scheduled
* Under build step, we can use multiple commands, but Jenkins will take is as one

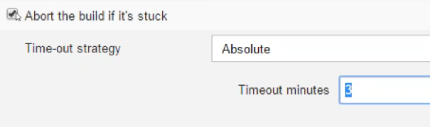
**Delete workspace before build starts:**

* With the below, we can clean the workspace before the build



**Abort the build if it’s stuck:**

* With the below, we can abort the build if it is stuck for 3 minutes



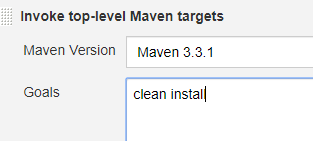
**Timestamps to the console:**

* If we check the **timestamp** option, we can get the timestamp for each line in console output

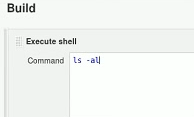


**Build:**

* Under the build actions, we can select the “invoke top level maven” for maven builds as below
* We need to select the maven version which we have set in global tools as below



* We can also use “execute shell” option to execute any command or script in the Jenkins server



* After building the job, it will create a workspace folder and creates sub folders.



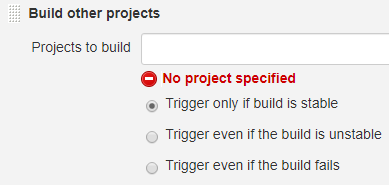
* We can build now
* We might do the mistake here without committing the files, so that Jenkins will give an error
* If we use the same java file to build multiple times, it will overwrite the class file
* The other way of doing it is, we can have two build steps



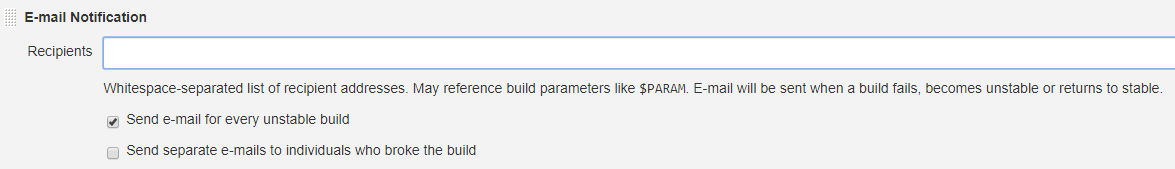
* Here, if the first build fails, it won’t go for second one. But in earlier one, if first command fails, then also it will go for second one
* If we use two commands in same build, if the first one fails and second one succeeded, then the build output will be as success
* We can also once one maven build step and another as execute shell or whatever

**Post build actions:**

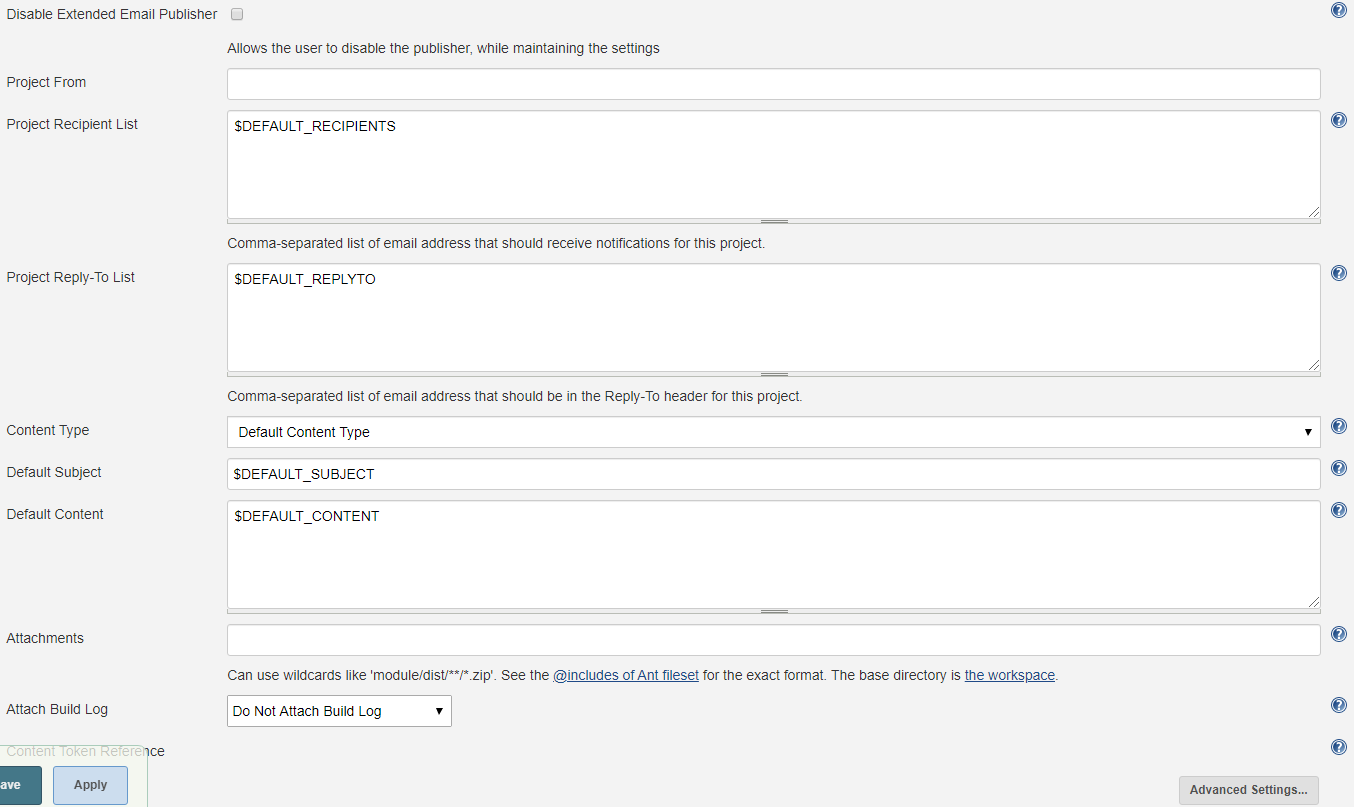
* Once the build is done, we can use post build actions to trigger the downstream jobs, delete the workspace, sending email notification, sending sonar reports etc.
* We can trigger the other projects using “**build the other projects**” option as below



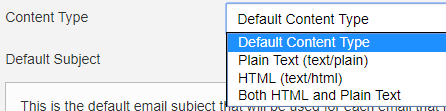
* We can also send email notifications as below using “**email notification**” option



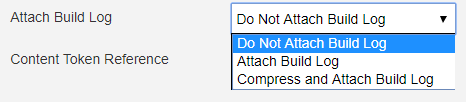
* “**editable email notification**” comes with some extra configurations and we can use that as below



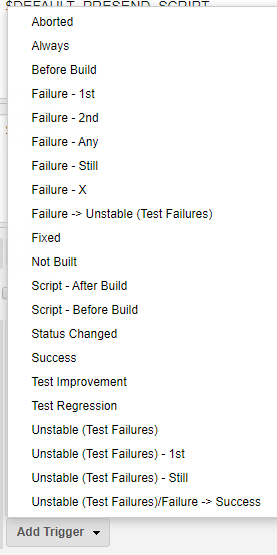
* We can give the recipient list, select the mail context type as HTML etc. we can also give the subject line other than default one and also, we can change the default mail content as well



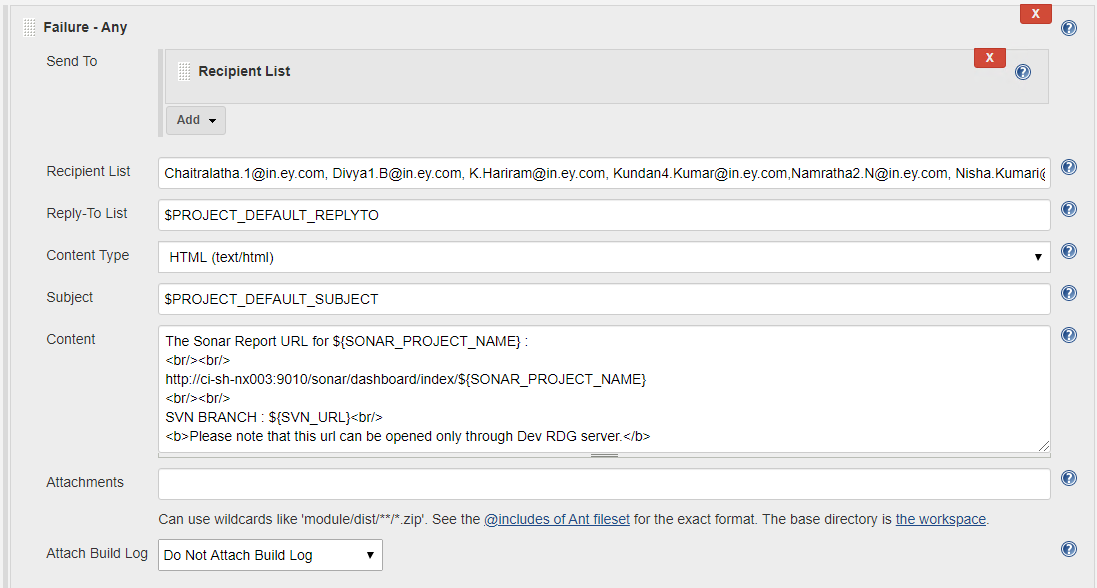
* We can also add the attachments and send the build log as well. Either by zipping or not zipping



* We can also go to the advance settings and add the triggers as below



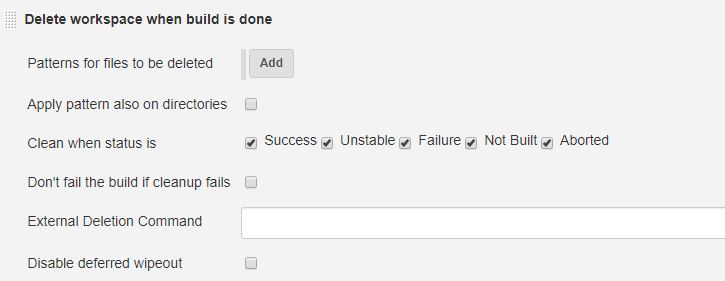
* In each trigger, we can go to advance settings and give the configurations as below



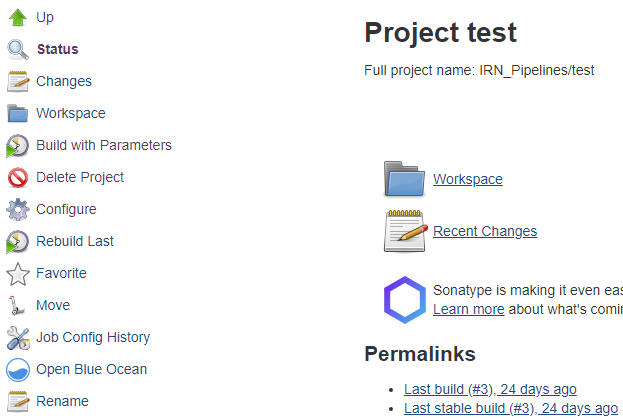
* We can give the recipient lists and content type as HTML as above and the email gets generated as below



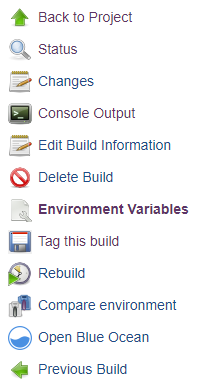
* “**delete the workspace when build is done**” is another option to delete the workspace after the build is done based on build status as below



**Other settings:**



* We can see the changes made in SCM from last build
* We can view the workspace from Jenkins dashboard itself
* We can delete and rename the project
* We can rebuild the job using the same parameters from last build
* We can check the job config history and revert to the old changes also



* We can go to the build, check for changes, check the console output, tag the build, check the environment variables, rebuild the same or delete the whole build

**Stable, unstable builds & publisher:**

