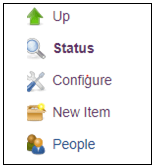
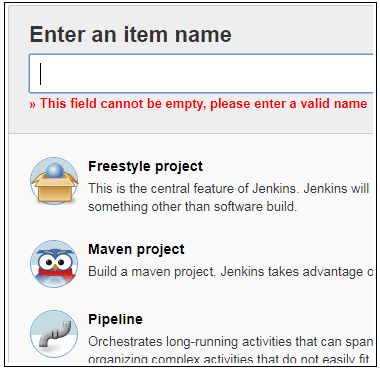
we will explore the process of creating pipeline scripts for SonarQube integration. Here are the steps.

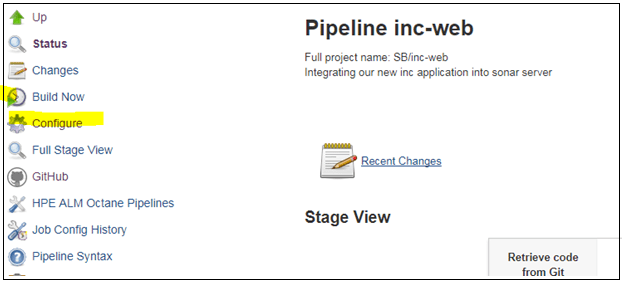
* Log in to your configuration domain (e.g. https://jenkins.domain.com).
* Go to the specific profile in Jenkins (Here, in this case, it is the root project).
* To configure a Sonar job, select *‘New Item’* available on the left side panel in Jenkins.



* In the subsequent screen provide a job name. Click on *‘Pipeline’* option, if you intend to run a Pipeline, else select the *‘Maven’* option.



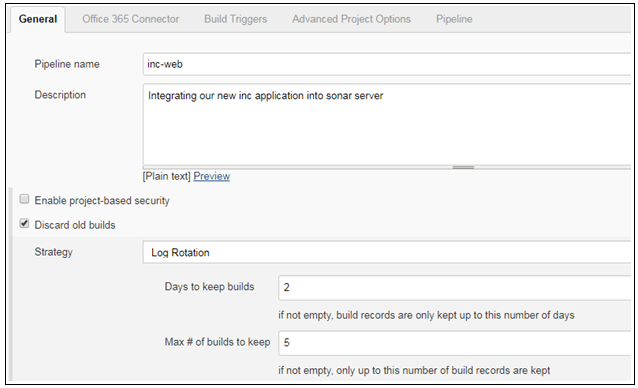
* Click on *‘OK’* button to provide configuration details. Now, let us take a look at the various links available on the left side of the screen.
  + Changes – The *‘Changes’* option enables developers to change the name of a job.
  + Build Now – This feature allows developers to run a job in Jenkins. It starts to read the code from the repository and builds the code.
  + Configure – This option enables developers to read the code from the Git/SVN repository.



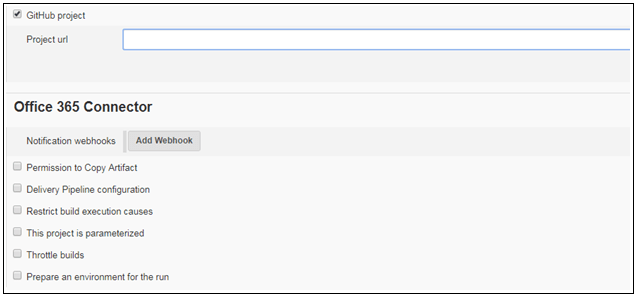
**SonarQube Integration with Jenkins**

Here is the complete process of SonarQube integration with Jenkins.

* Click on *‘Configure’* option, which will redirect developers to the following screen, enabling them to read the code from the Git/SVN repository.
* In the General tab, developers can provide a Pipeline name and log build details, such as how many days the logs should be kept etc. In the *‘Days to keep builds’* field, enter the number of days.

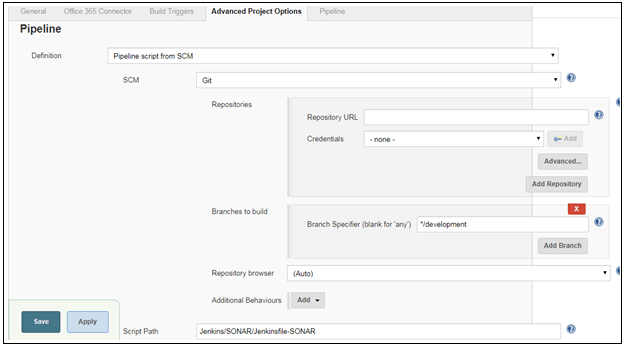


* If using a Git repository, select Git project, else proceed to the next tab.
* The next tab is ‘*Office 365 Connector*’, this screen is used to run a build based on parameters.
* The option *‘This project is parameterized’* is one of the parameters that developers should have a brief idea. It is used to build a job based on branches. In situations, where developers want to run a job based on development/master branches, they can define it using this option else, the job will run as default.



**Build Trigger Section** – This option can be utilized, if developers are keen to run the jobs based on specific time intervals e.g. once a day or twice a day. Additionally, jobs can be scheduled to run automatically by using the ‘Build periodically’ option.

* **Pipeline Section** – The pipeline section is the core feature of a job; it reads data/code from a specific repository in GIT/SVN. However, developers need to specify the script file, which is available in the GIT/SVN application.



Here is the pipeline script that needs to be added to the Jenkins file.

pwd

ls

cd svc-po

docker stop $(docker ps -a -q --filter ancestor=svc-po:latest)

docker rm $(docker ps -a -q --filter ancestor=svc-po:latest)

docker image rm svc-po:latest

docker build -f DockerFile -t svc-po:latest .

docker run -d -p 8084:8084 svc-po:latest