

**Node can be a master or agent**

**Jenkins Installations:** To run Jenkins on machine (Windows or Linux) first Java should be installed

We need to run below commands on Linux to install Java

### Install JAVA

sudo add-apt-repository ppa:openjdk-r/ppa

sudo apt-get update

sudo apt-get install -y openjdk-8-jdk

**Note : when we install java in linux it will be installed in /usr/lib/jvm**

### Install Jenkins

sudo wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -

sudo echo deb https://pkg.jenkins.io/debian-stable binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list

sudo apt-get update

sudo apt-get install jenkins

**Note : Jenkins will be installed in cd /var/lib**

### Install Maven ( on Jenkins machine )

cd /tmp ; sudo wget https://dlcdn.apache.org/maven/maven-3/3.8.5/binaries/apache-maven-3.8.5-bin.tar.gz

cd /tmp ; sudo tar -xzf apache-maven-3.8.5-bin.tar.gz -C /opt/

##### **Optional Set JAVA\_HOME & MAVEN\_HOME as environment variables on Jenkins machine**

sudo echo "MAVEN\_HOME=\"/opt/apache-maven-3.8.5\"" >> /etc/profile

sudo echo "JAVA\_HOME=\"/usr/lib/jvm/java-8-openjdk-amd64\"" >> /etc/profile

sudo echo "PATH=\$JAVA\_HOME/bin:\$MAVEN\_HOME/bin:\$PATH" >> /etc/profile

source /etc/profile

##### how to restart Jenkins

sudo systemctl restart jenkins # to restart

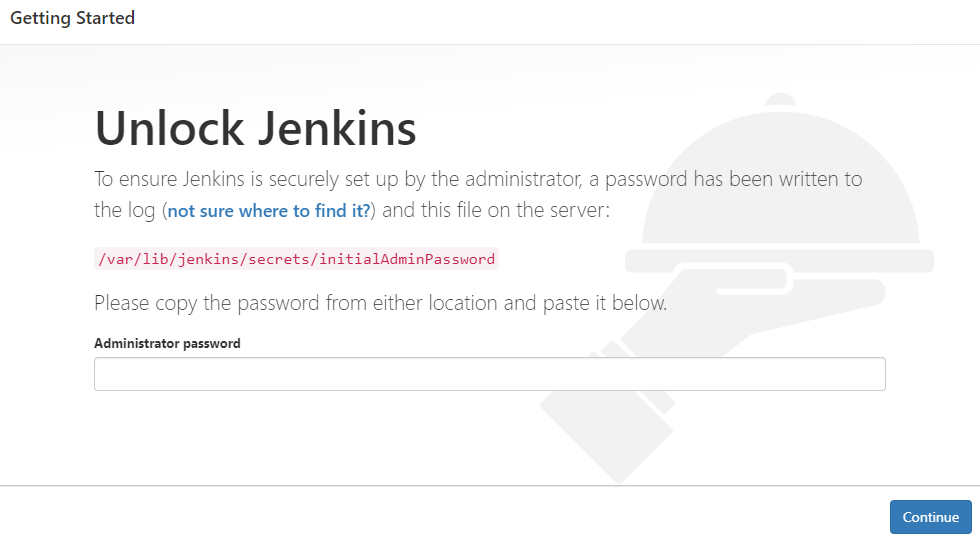
sudo systemctl stop jenkins # to stop

sudo systemctl start jenkins # to start

sudo systemctl status jenkins # to check the status

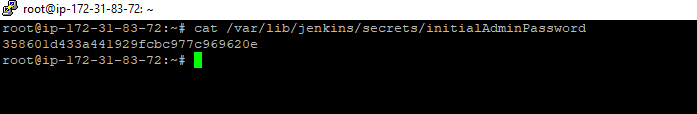
After running Java and Jenkins commands on Linux, we can launch GUI of Jenkins with ipaddress and portnumber 8080

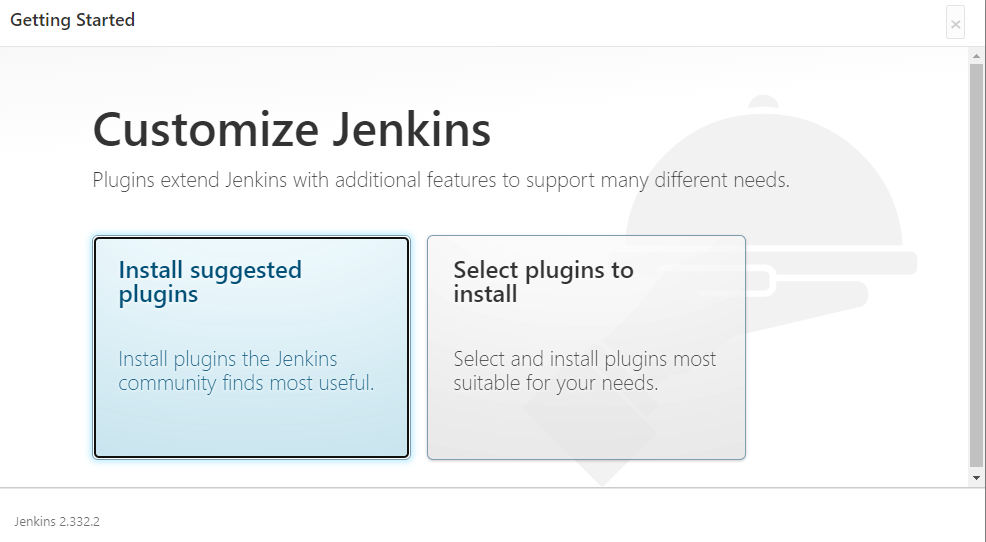
For example : Ip address in AWS EC2 for my current linux is 3.86.202.47 so on the address bar give <http://3.86.202.47:8080/>

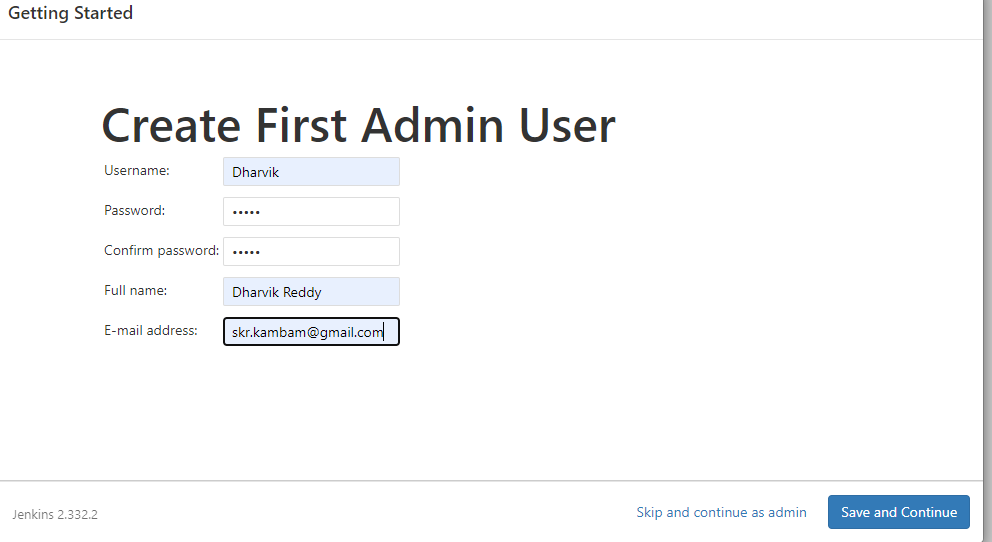


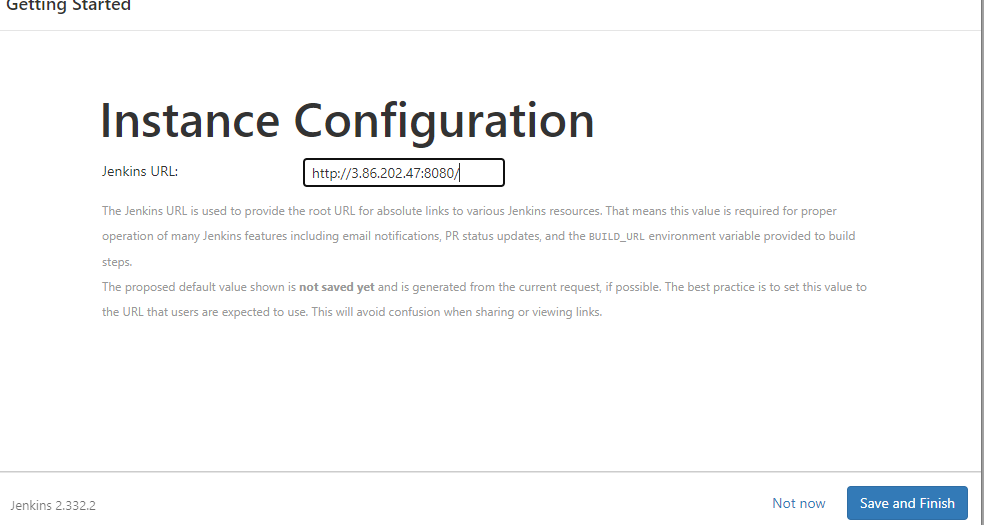
For first time login we will get above page and from the path mentioned in above screen we can get admin password

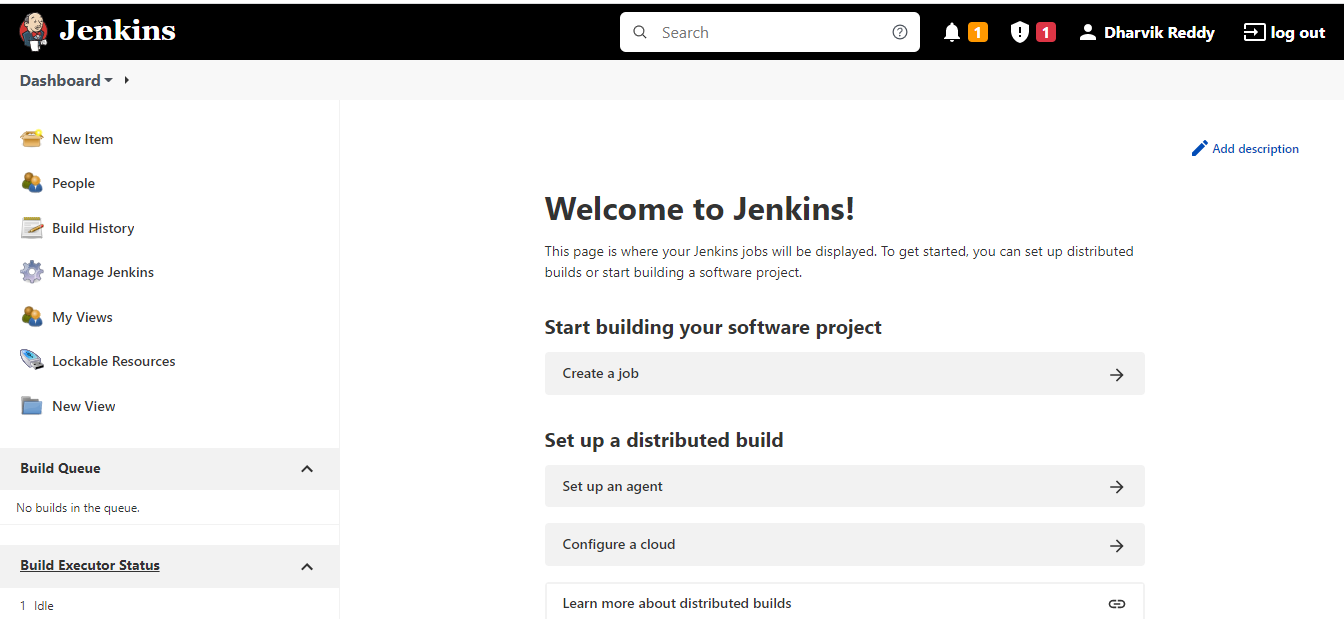
Use Cat command to open the file







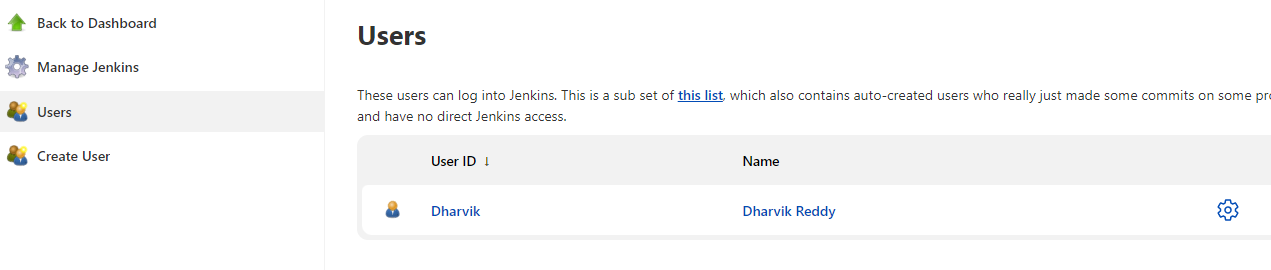


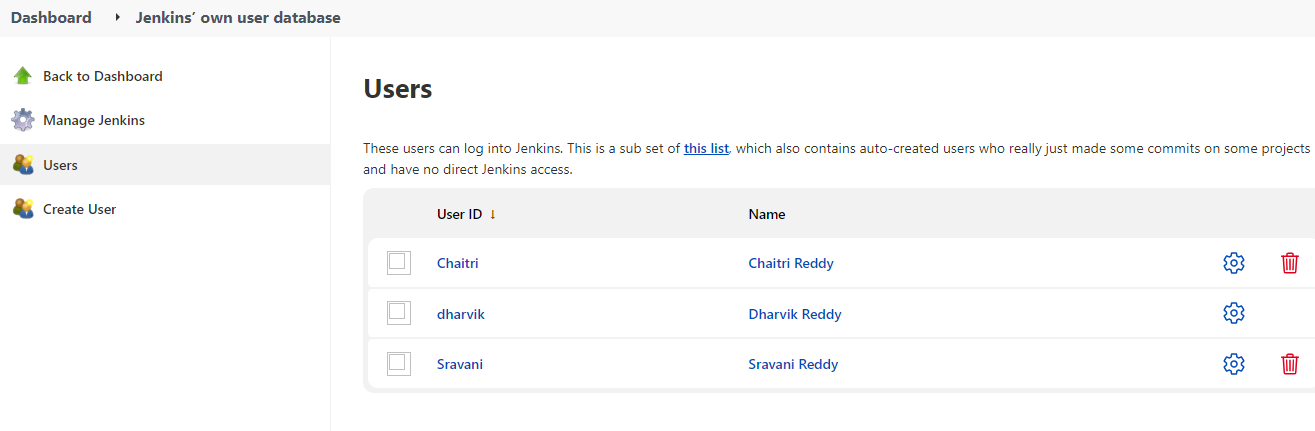


Jenkins is ready and ready to work

**Creating Users:**

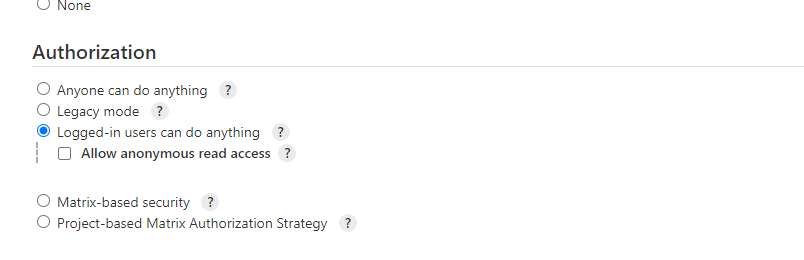
Go to Manage Jenkins > Manage users > Create users





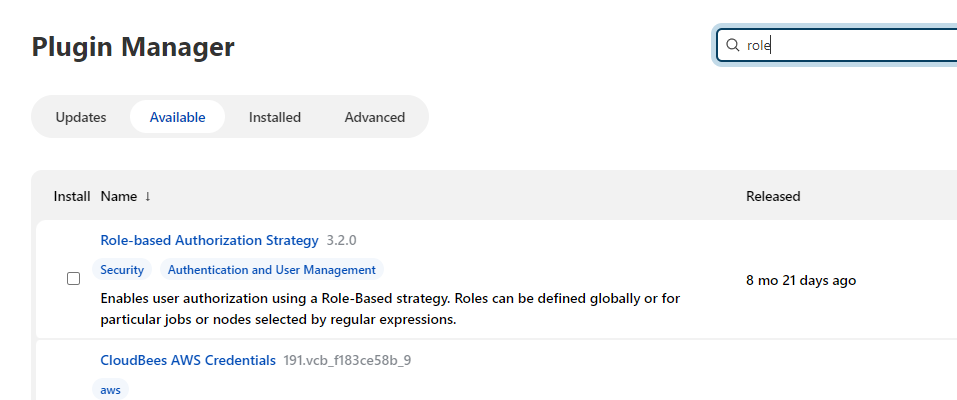
New users can be created from Create user tab.

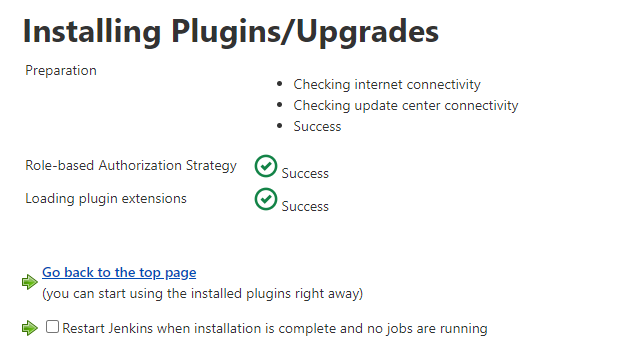
Now to assign the roles, go to configure global security

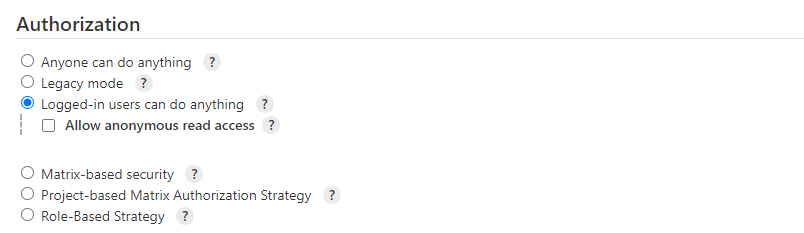
Matri

Matrix based security can be used only if there are few users but for wide number of users we need to use role based strategy.

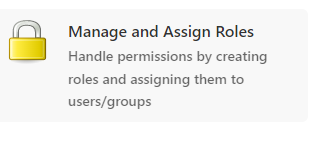
We do not have role based authorisation so install Role based authorisation plugin



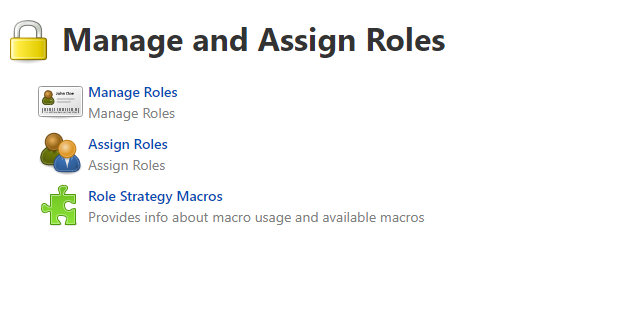




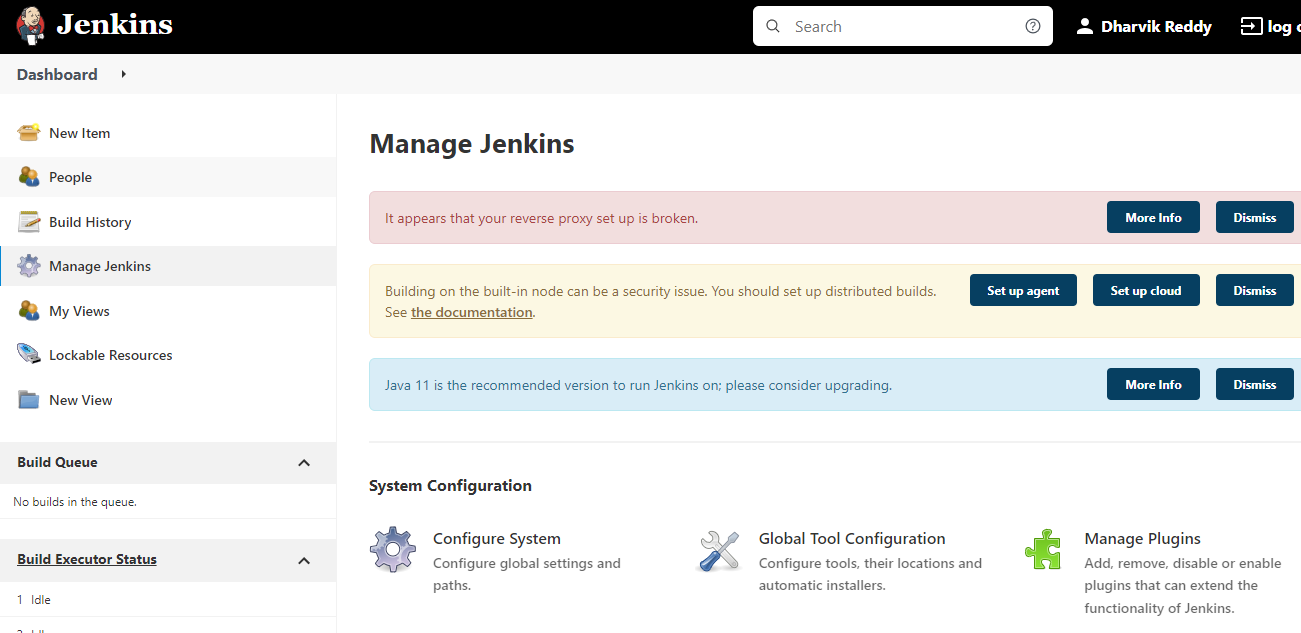
Now we can see Role based strategy, select and save



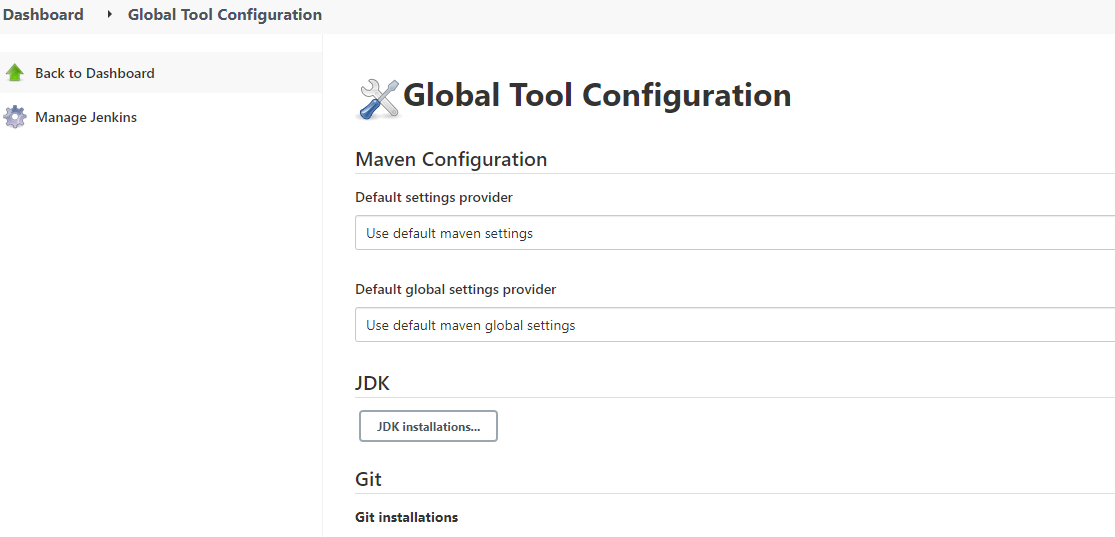
We can see new option (Manage and Assign roles)

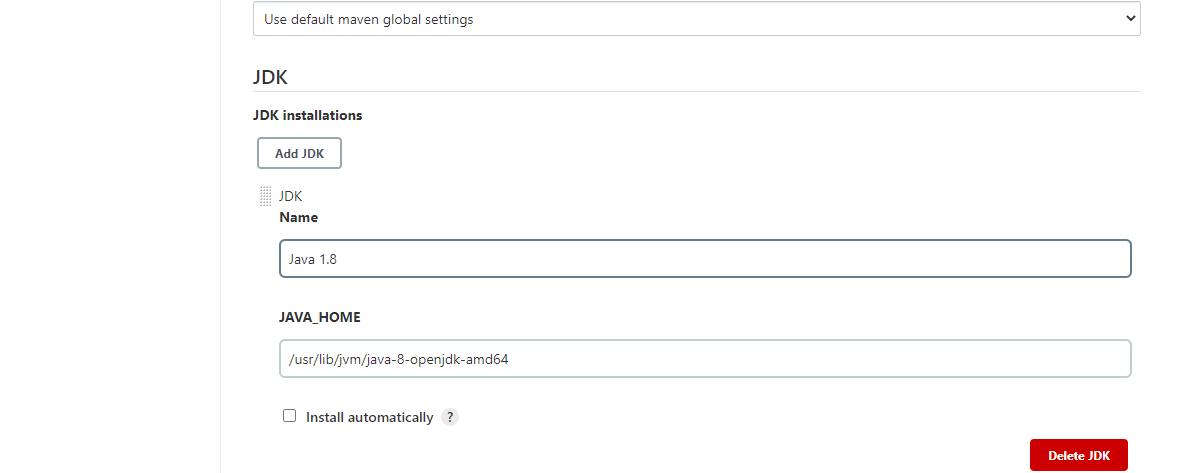


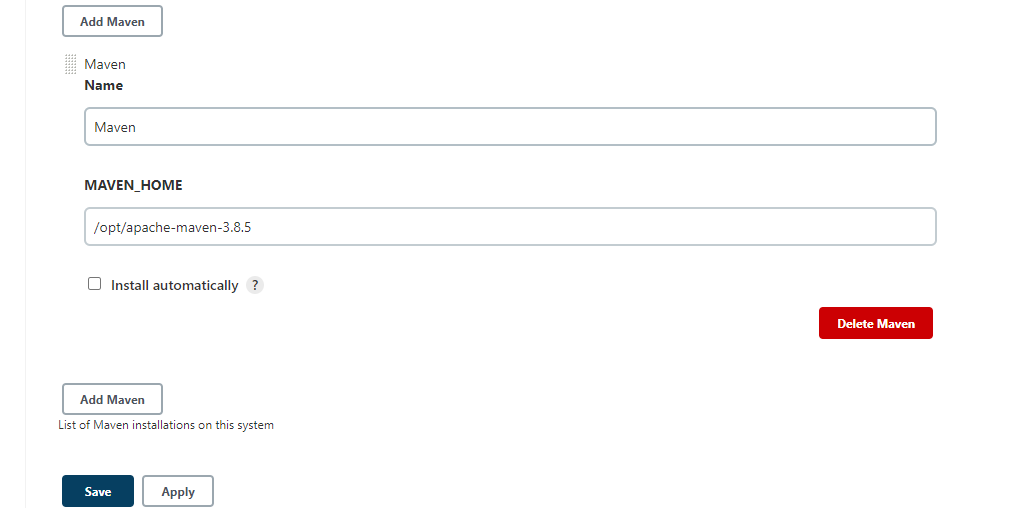
Go to Manage Jenkins and Configure Java



Go to global tool configuration



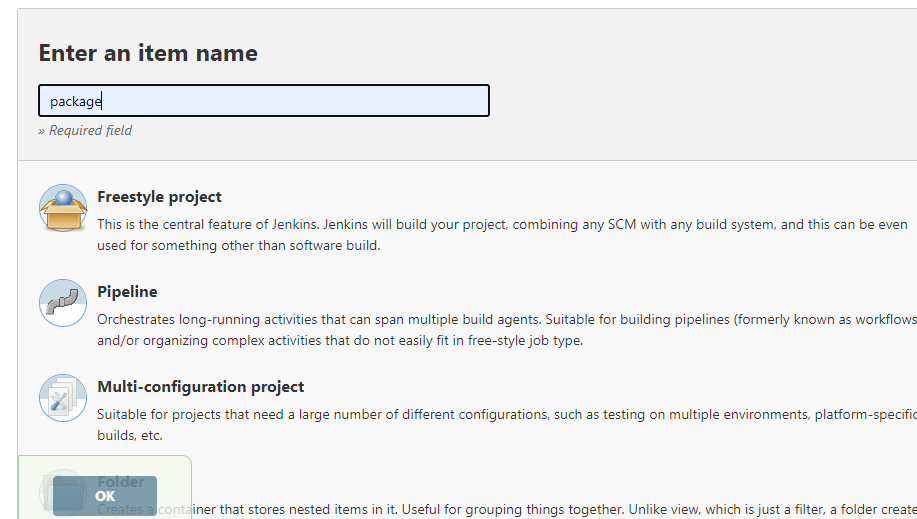




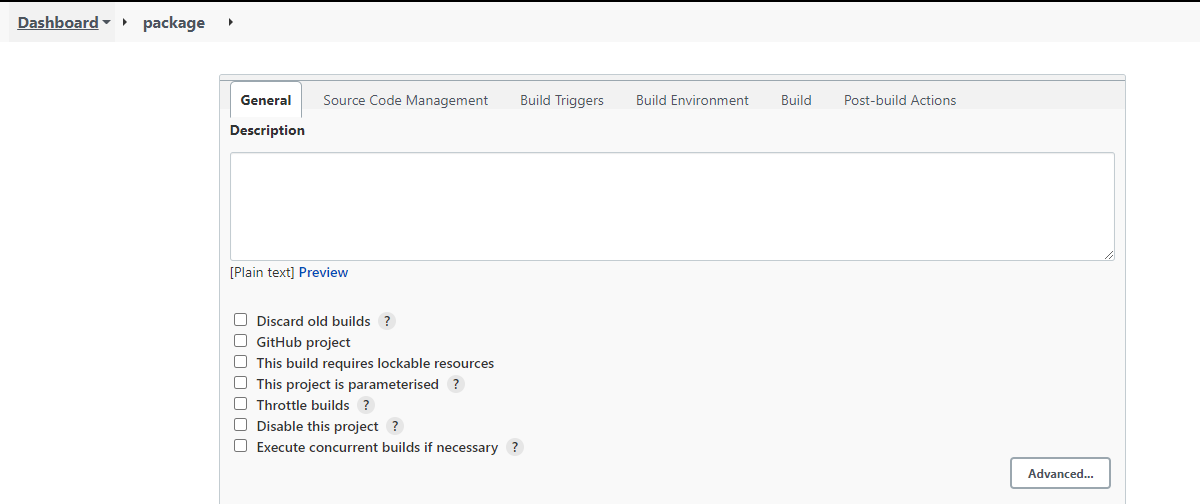
Add java and maven paths from Linux files

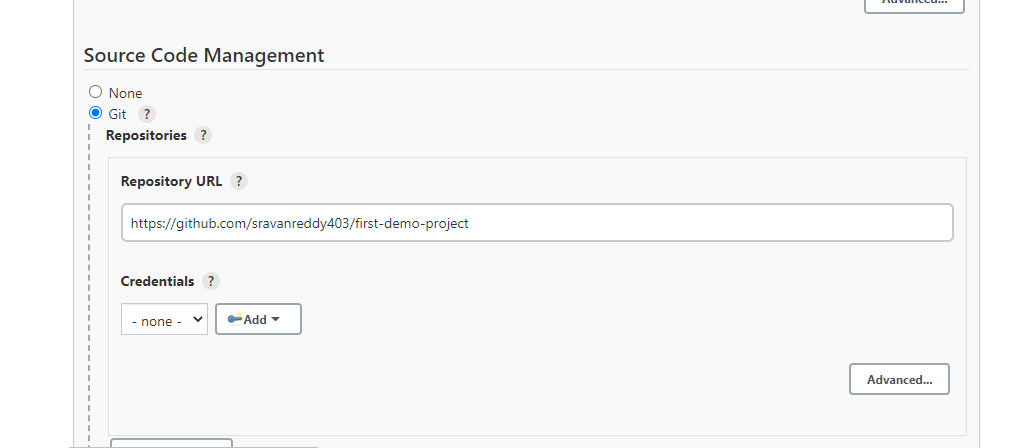
Now we can create New item

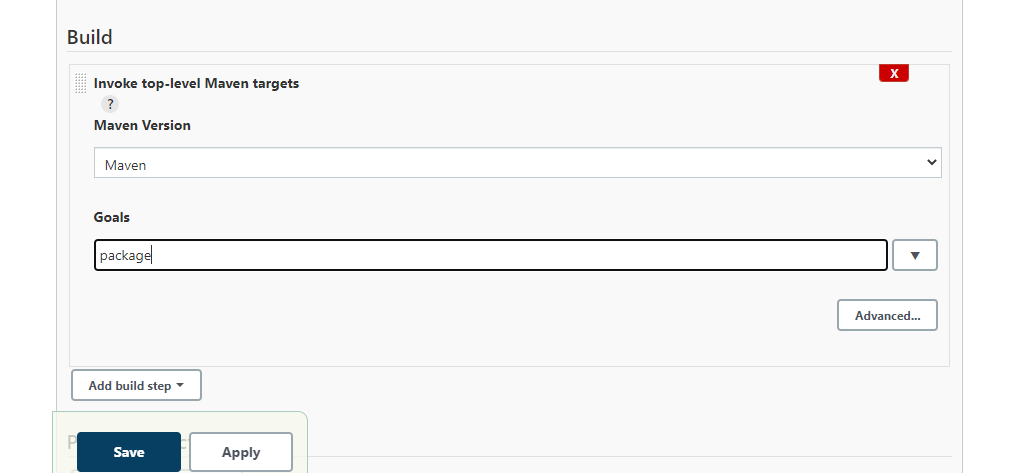
Here I am creating new item for Package of free style project and adding github url

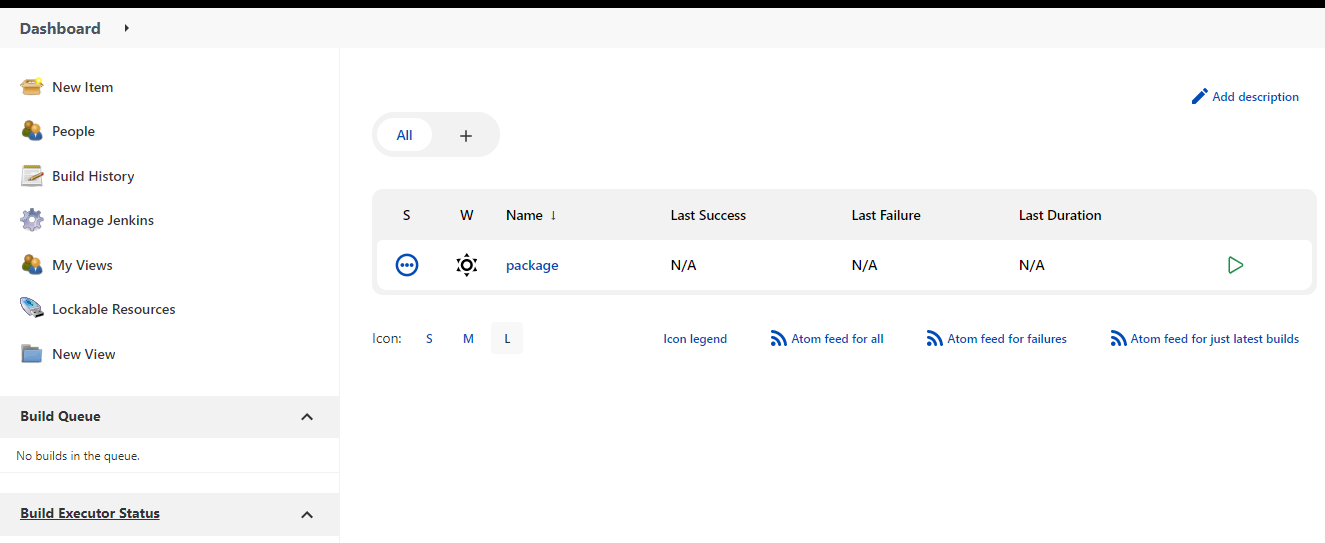




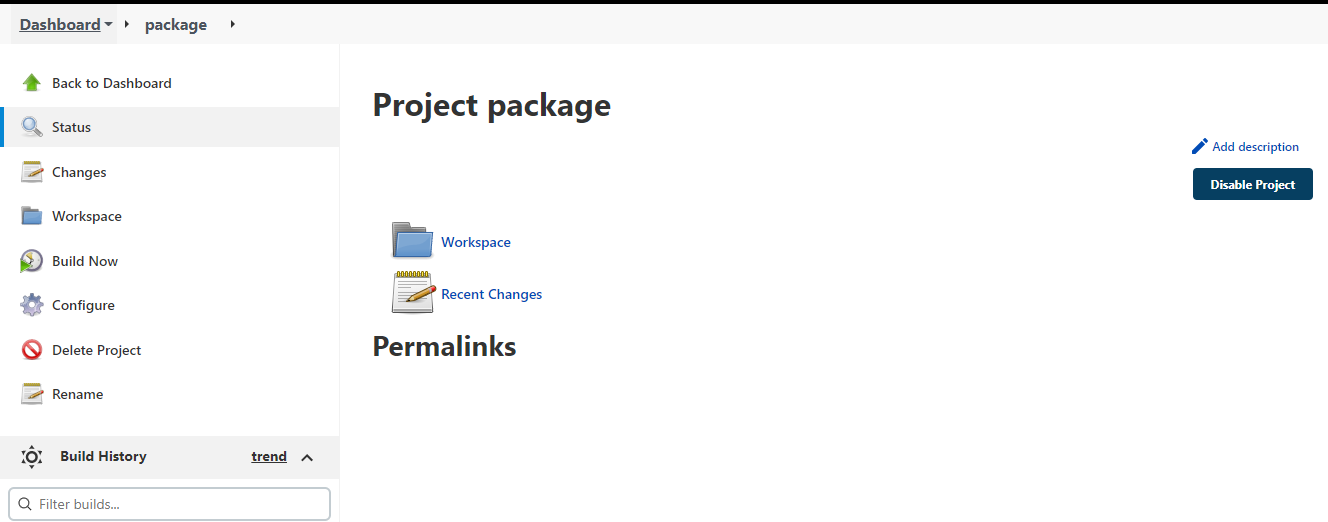


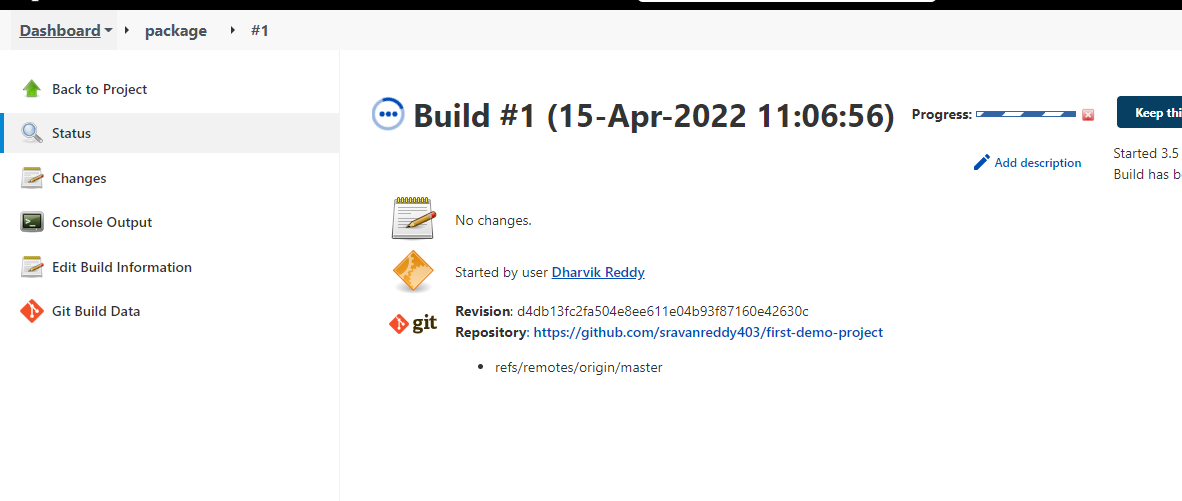


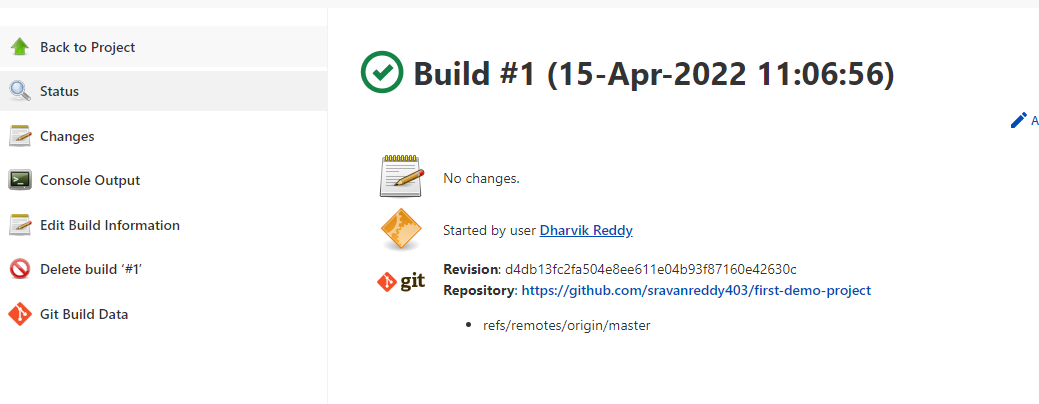




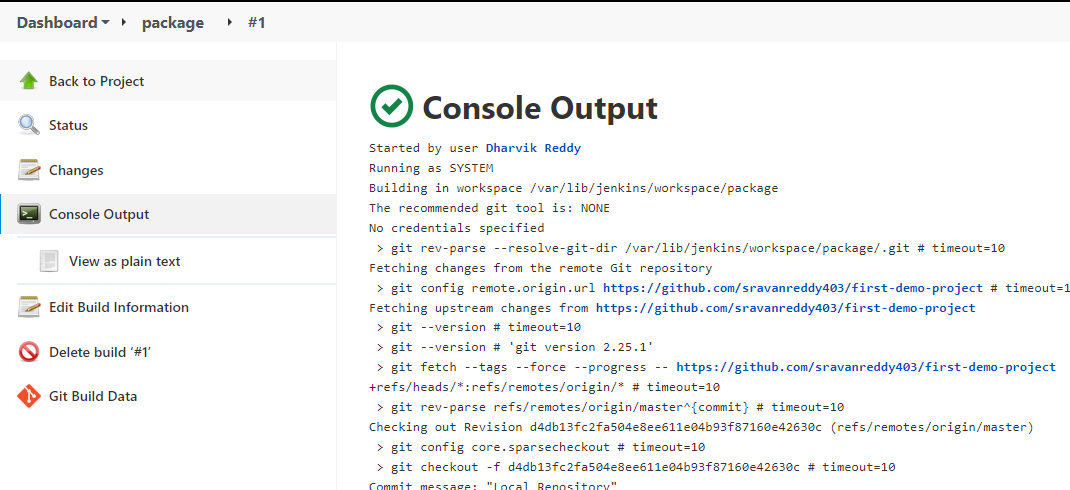
Package project is created and we can build now this package

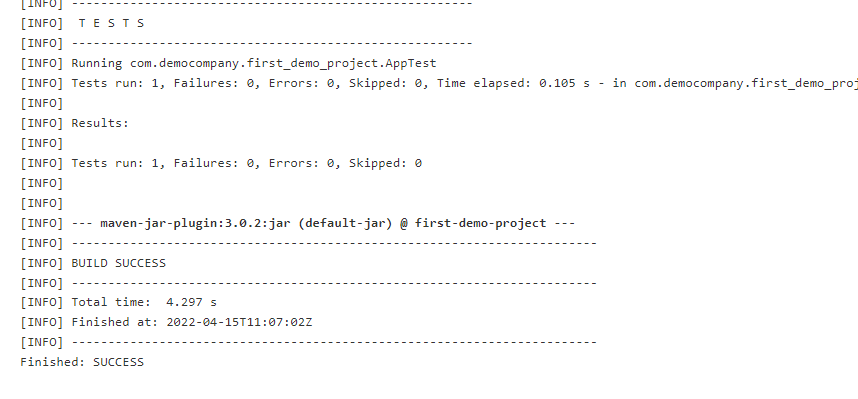






We can see Console output for logs

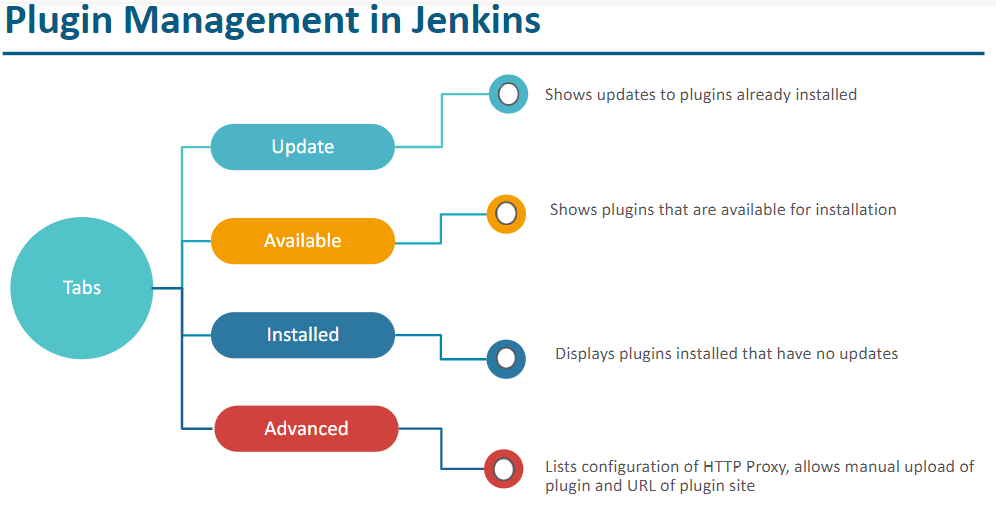


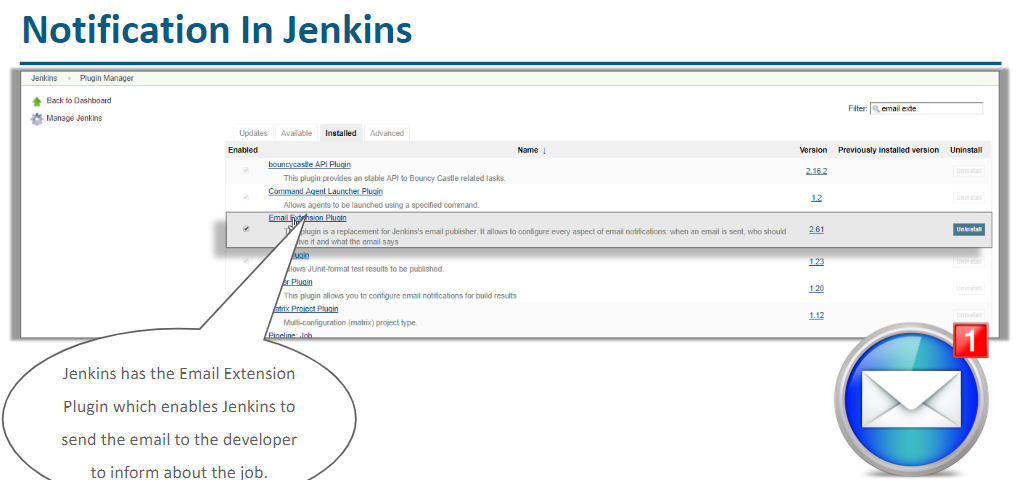


**CI Steps :**

1. **Compile**
2. **Codereview**
3. **Unit test**
4. **Code coverage**
5. **package**

**Jenkins plugins:**





Plugin is an add-on that extends the basic functionality.

Go to manage plugins from configure Jenkins and available install cobertura and buid pipeline

**Cobertura and sonar are few code coverage tools**

Install Plugins required for CI Pipeline

Manage Jenkins --> manage plugins -- Available -- search & install the below

Warning NG -- install without restart

cobertura -- install without restart

Build Pipeline Plugin -- install without restart

Java configuration in Jenkins console

Manage Jenkins --> Global Tool Configuration --> JDK --> Add JDK

Name: myjava ( can be any string )

JAVA\_HOME: /path/to/javahome ( ex: /usr/lib/jvm/java )

Maven Configuration in Jenkins console

Manage Jenkins --> Global Tool Configuration --> Maven --> Add Maven

Name: maven3.6 ( can be any string )

MAVEN\_HOME: /path/to/mavenhome ( /opt/apache-maven-3.6.0 )

Jenkins ( CI )

CI ( continuous Integration )

Jenkins Job : maven goal : post build action

job1-codereview : -P metrics pmd:pmd : \*\*/pmd.xml ( publish pmd analysis results )

job2-unittest : test : target/surefire-reports/\*.xml

job3-codecoverage : cobertura:cobertura -Dcobertura.report.format=xml : target/site/cobertura/coverage.xml ( publish cobertura coverage report )

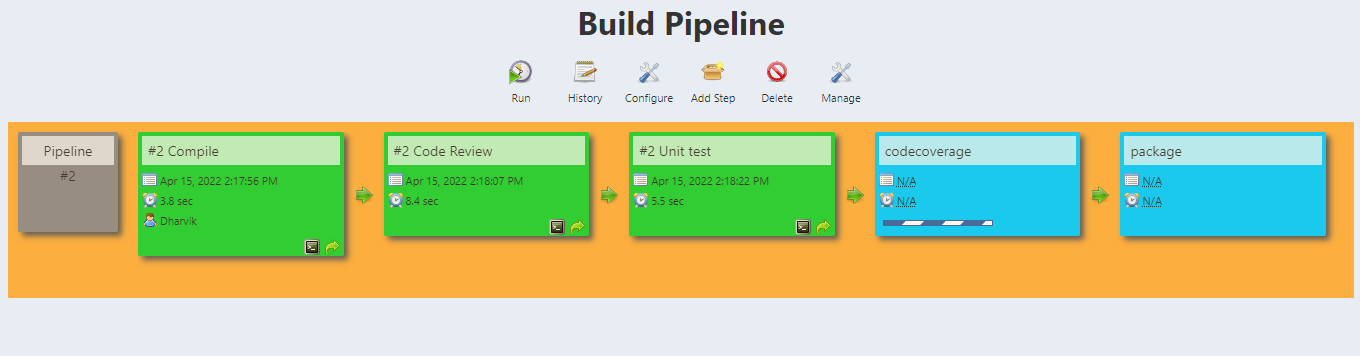
job4-package : package : NA

Pipeline : anything that we can run in sequential

**Compile, code review, unit test, code coverage, package – Running these steps in sequential is CI pipeline or CI process**

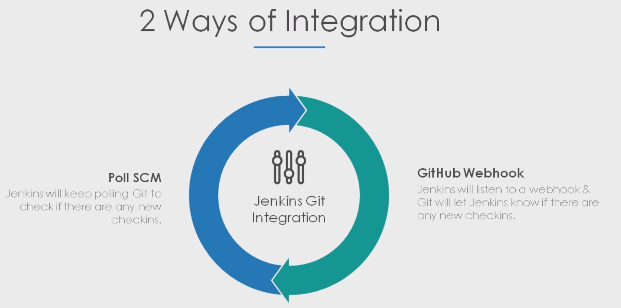
***Create the process , add build pipeline process?***

***Build pipeline plugin need to be installed to visualise which job is dependant on other job***



**Jenkins \_ GIT Integration :**

* **Poll SCM** ( Jenkins can check for every specific time, like for specified time interval jenkins will check if any channges in GIT and run the pipeline)
* **Webhook** ( This webhook will let jenkins whenevr there is commit in GIT)



## **Now if we want to run this pipeline as when there is change in github we need to follow the below process.**

First add webhook in git hub repository

## **trigger-build-with-github-webhook.txt**

1. Navigate to the “Settings” tab in GitHub Project
2. ) Select the “Webhooks” option on the left menu

Click “Add Webhook”

For “Payload URL”:

Use the address for the Jenkins server instance (e.g. <http://ip:8080>)

Add /github-webhook/ to the end of it.

Note: Make sure to include the last /!

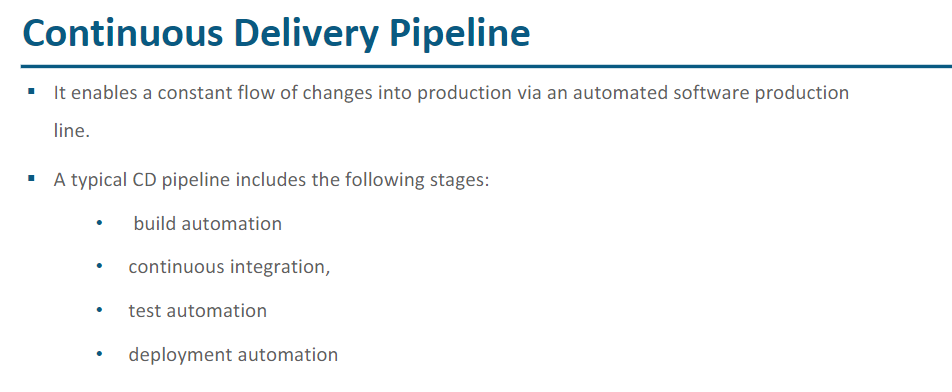
example: http://ip:8080/github-webhook/

We can connect Jenkins to LDAP or Active Directory and add users from that.

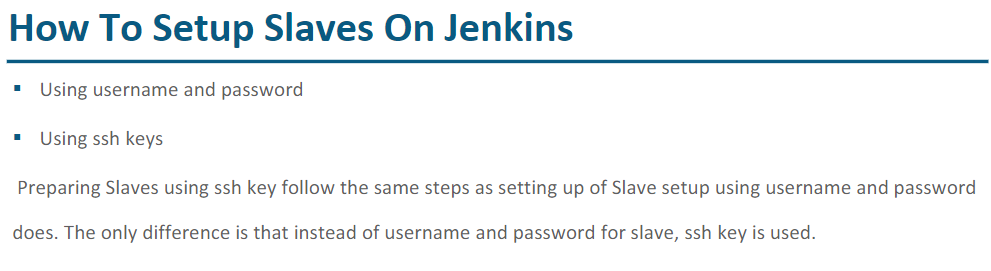
Jenkins provides default user option stored on own Jenkins database

For LDAP user access, go to manage Jenkins > configure global security (Select LDAP radio button and add LDAP server details).

If we enable auditing plugin then we can see user based activities.



**Jenkins Master – Agent configuration:**



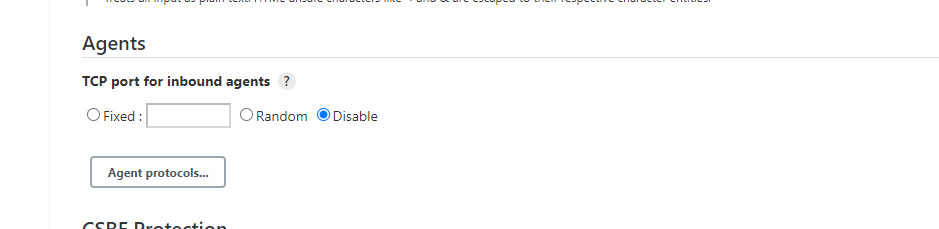
Running Jenkins is just a process running on system which will use system resources. If same Jenkins is shared with multiple teams and multiple pipelines are constructed then system will be slow or hung state and all the pipelines will fail.

So having Jenkins a single instance is not right option and we need to use Master –agent configuration

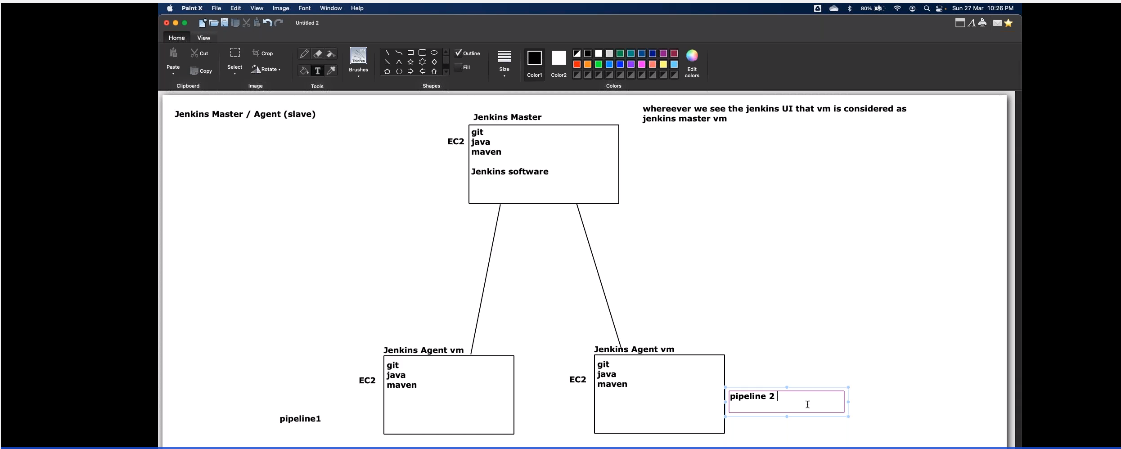
We will install Jenkins on master but not on agent

Steps:

1. Create new instance from AWS EC2 for Agent1.
2. Launch Ubuntu from putty with newly configured instance details
3. Install git, Java and Maven (Do not install Jenkins).
4. Now go to master Jenkins, Manage Jenkins > Configure global security



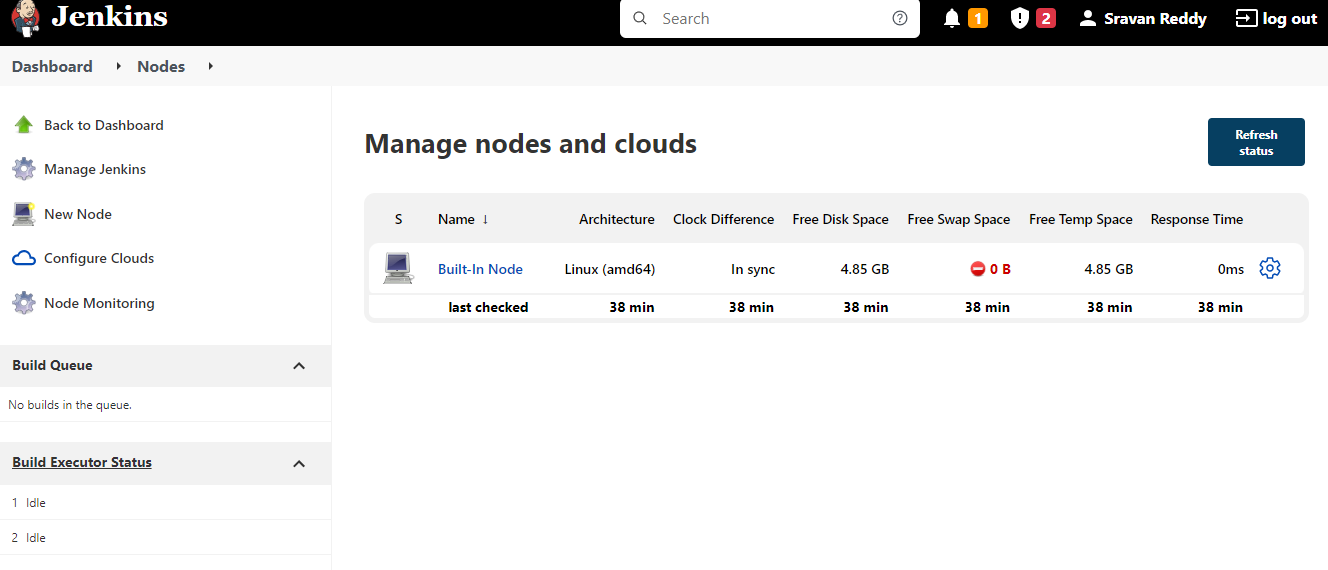
**By default agents will be disabled, Select Random and save**



Master node is used for administration purpose and agents are used for pipeline process and send back results to master

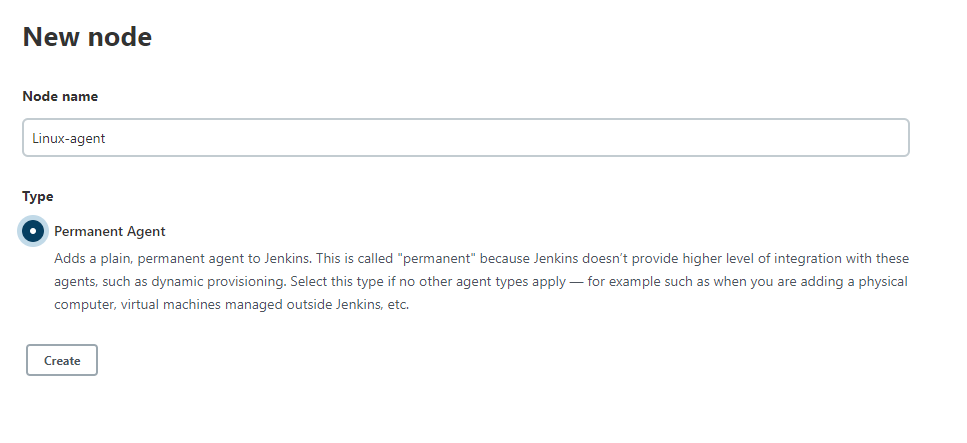
Master is always single instance and agents are multiple.

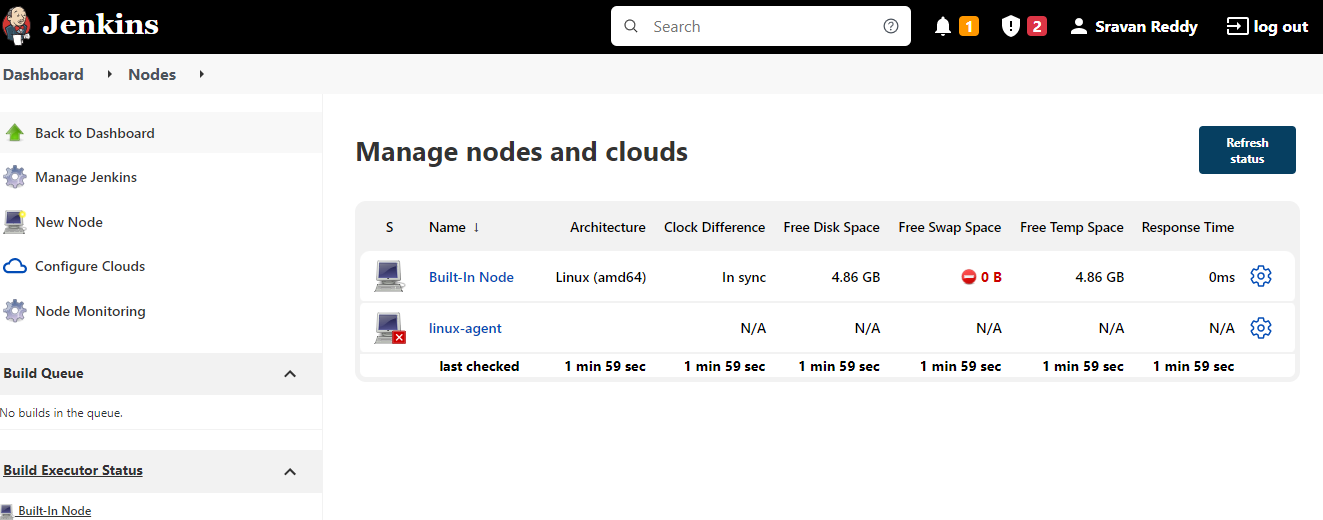
1. **Now go to Manage Jenkins > Manages nodes and clouds**

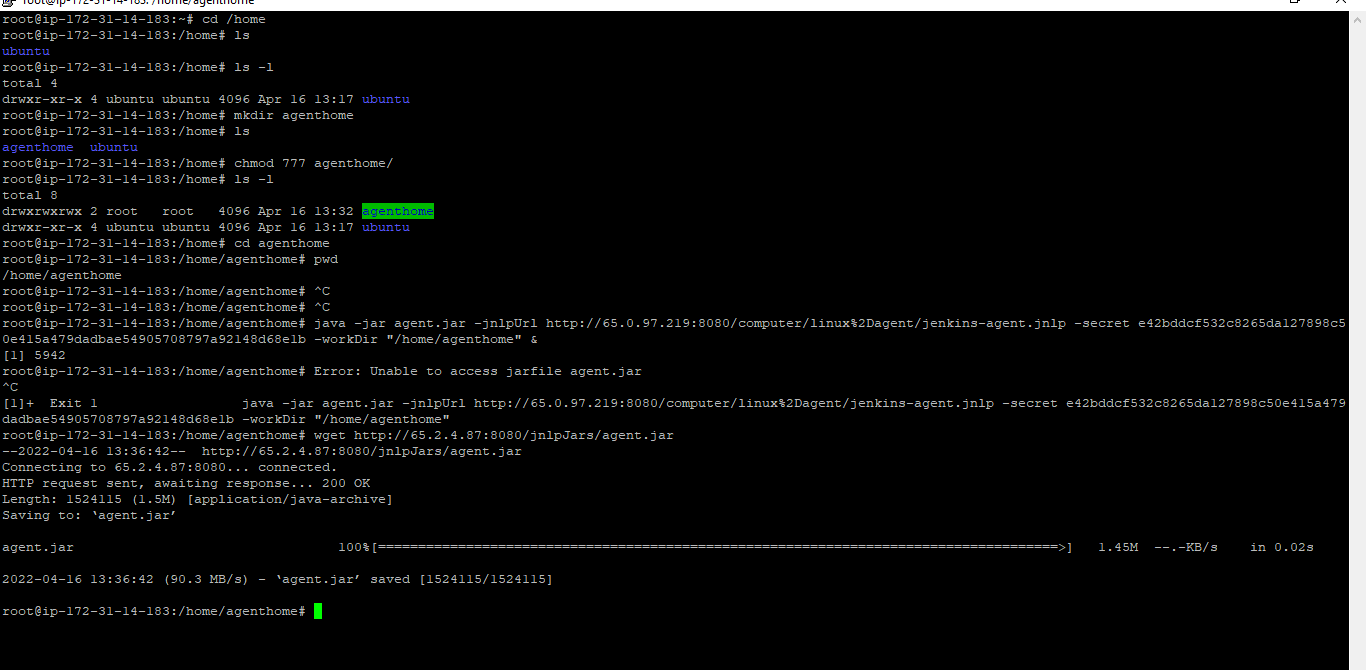


**Built in node is master itself**

**Now to create new agent, go to New Node**

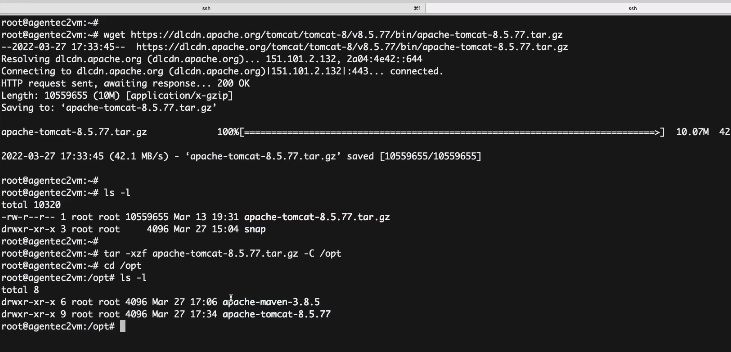






**After wget agent.jar run below command**

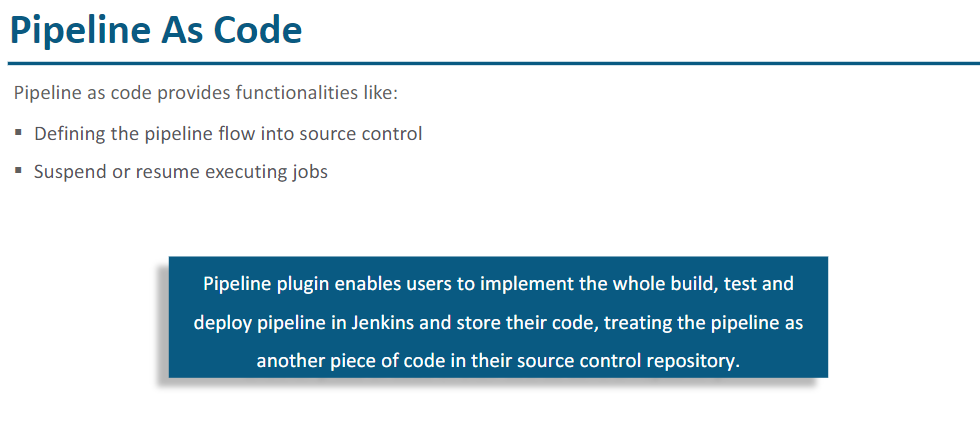
java -jar [agent.jar](http://65.2.4.87:8080/jnlpJars/agent.jar) -jnlpUrl http://65.0.97.219:8080/computer/linux%2Dagent/jenkins-agent.jnlp -secret e42bddcf532c8265da127898c50e415a479dadbae54905708797a92148d68e1b -workDir "/home/agenthome"



**Once installed go to webaps and copy the war file**

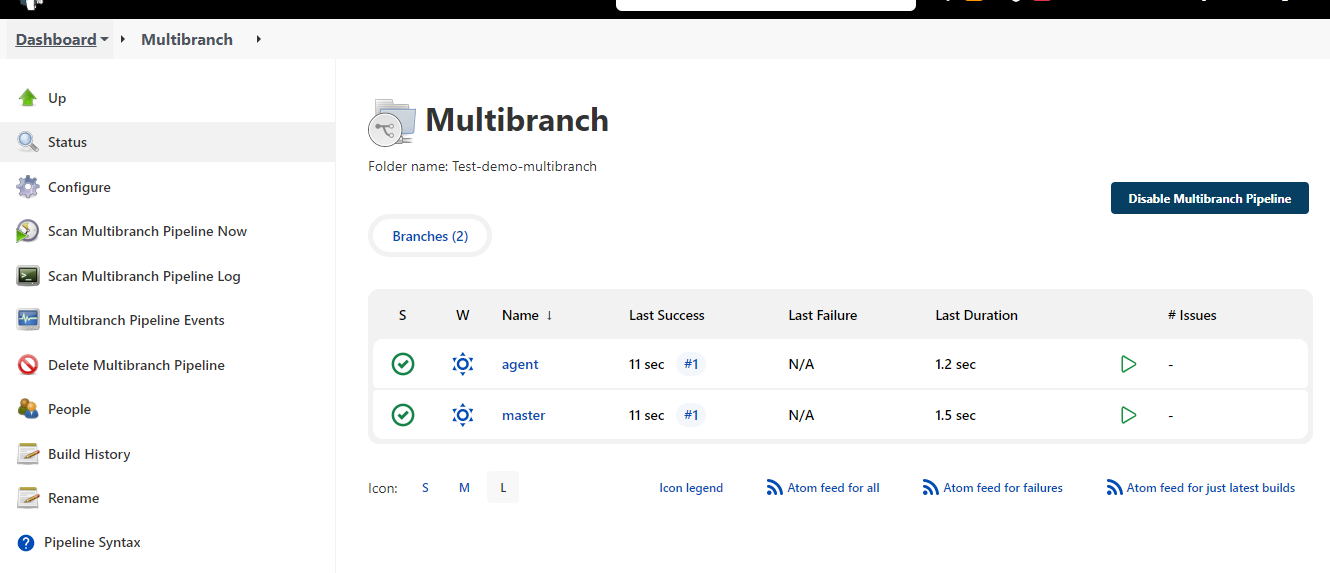
**And then go to bin folder and hit the bash statup.sh**

**Pipeline as Code**



Multibranch pipeline & pipeline: if we have multiple branches in our git and each branch has Jenkinsfile then we can create multibranch pipeline which will execute all the branches.

For examples if my github repository is having 3 branches and each one as Jenkins file then 3 pipelines will be created



Now if we add one more branch with Jenkins file we need to click on scan multibranch pipeline now.

To create Jenkins pipeline we have two ways to create

* Scripted pipeline ( requires knowledge on groovy script)
* Declarative pipeline

Pipeline {

Agent any

{

Stages{

Stage

{

**SONARCUBE: Static code analysis**

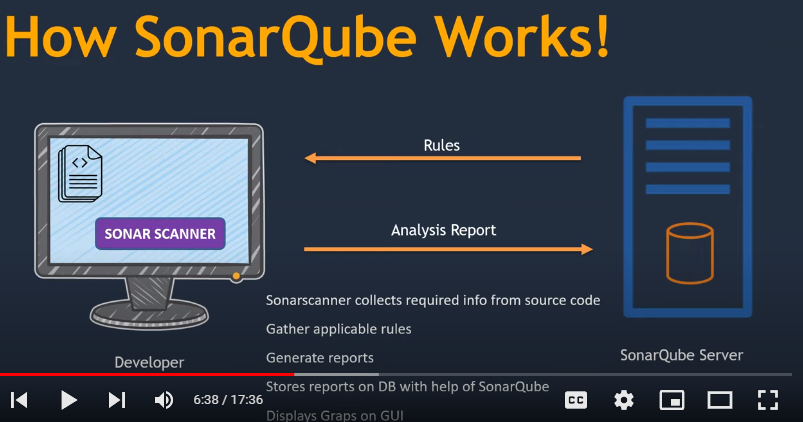
* + Sonorcube is used to automate the peer review of code coverage
  + It is quality management tool (Unit testing reports, Code coverage reports)



**Three major components in Sonarqube server**

* + **Rule :** Instructions to follow while writing code
  + **Database :** Analysis report must be stored in database after rules are run
  + **Web interface** : once rules are stored in database then with web interface we can analyse the reports.

**Sonar scanner:** It is service or agent which runs on the system where code exist, once sonar scanner runs on the source code then report will be published. (Wherever we are having sonar scanner there source code must be present)



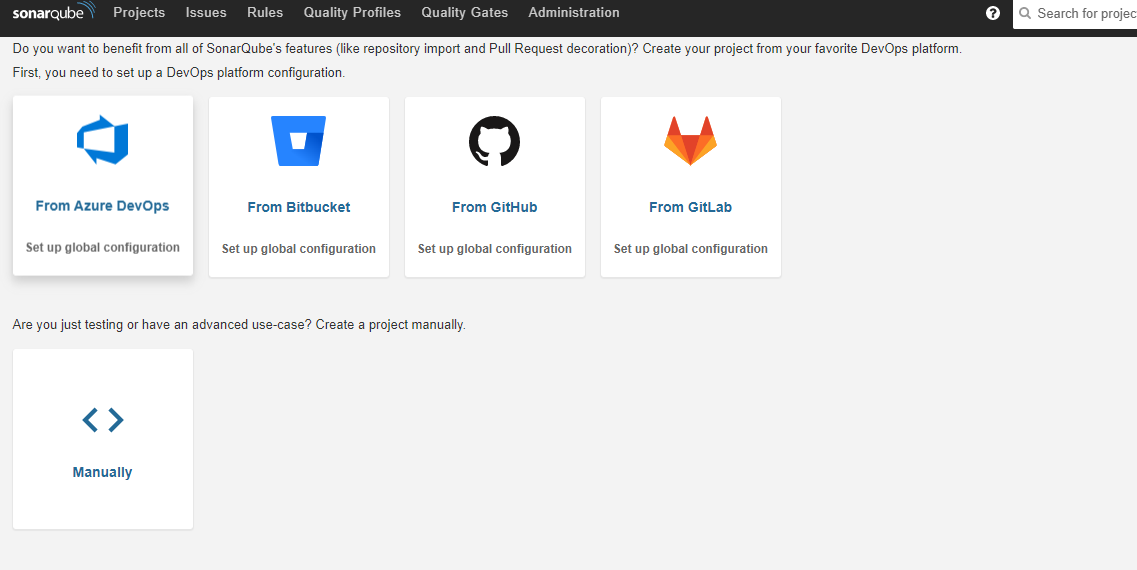
We must install sonar scanner in source code system,

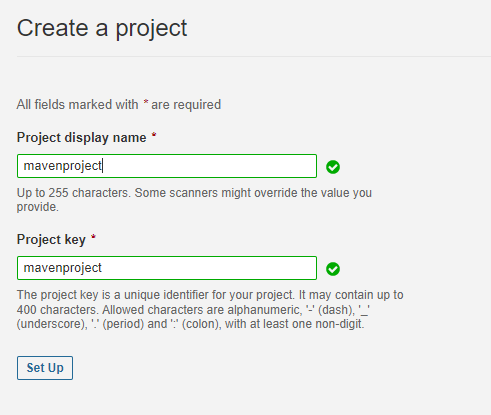
* + Sonar scanner collects required info from source code
  + Gather applicable rules
  + Generate reports
  + Stores reports on DB with the help of sonarQube
  + Display graphs on GUI

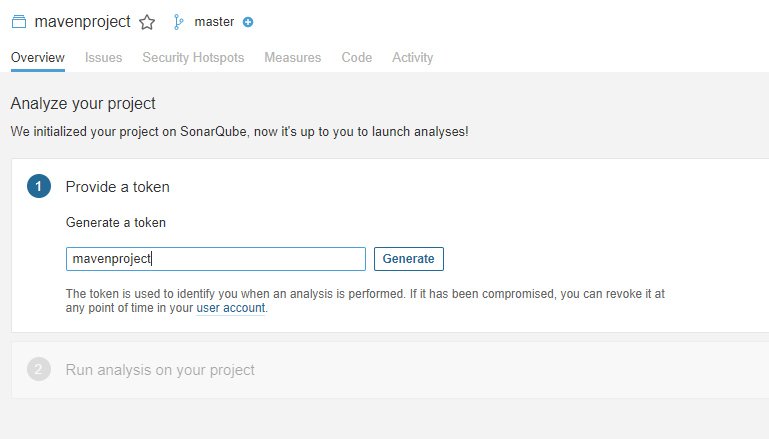
For installation steps

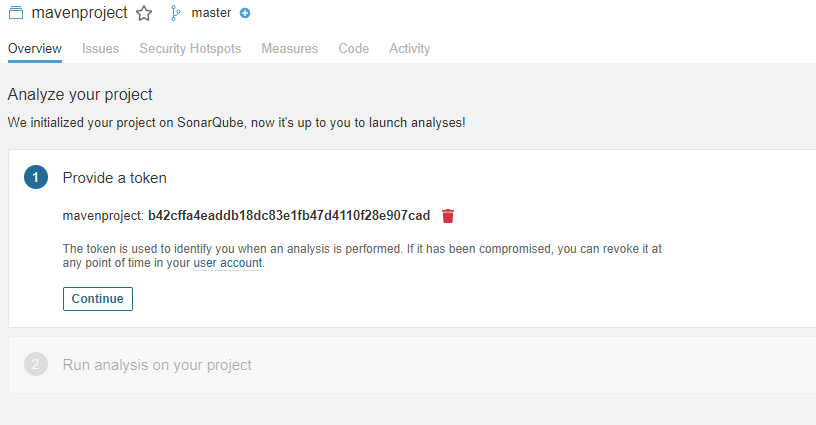
<https://github.com/ravdy/DevOps/blob/master/sonarqube/Setup_SonarQube.md>

In Linux to install and start sonarcube we need to create with user and not with root









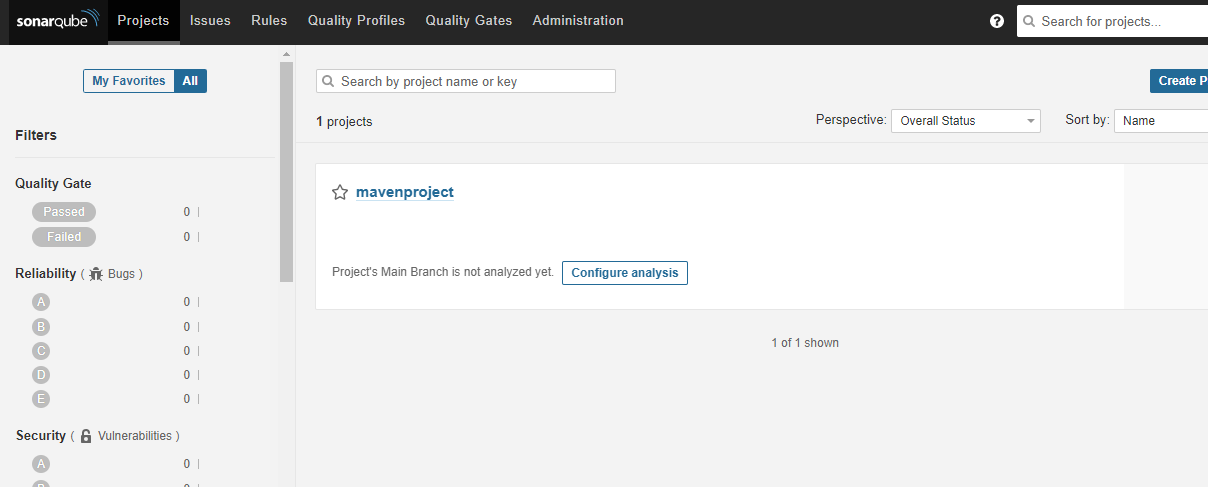
**b42cffa4eaddb18dc83e1fb47d4110f28e907cad**

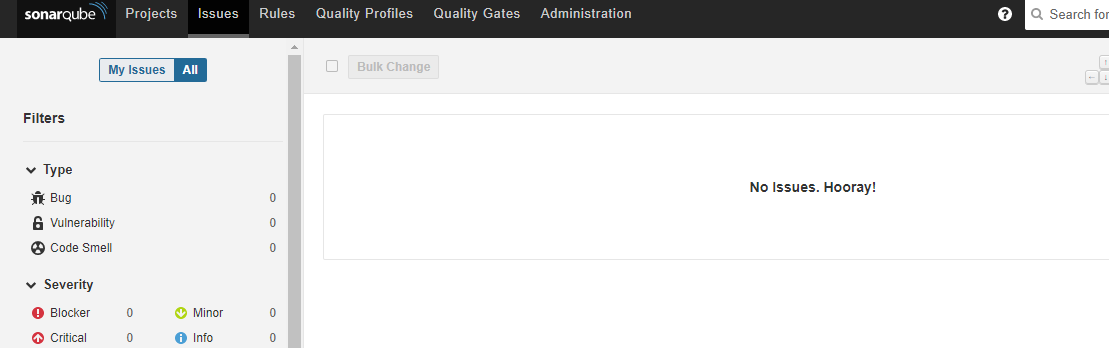
mvn clean verify sonar:sonar \

-Dsonar.projectKey=mavenproject \

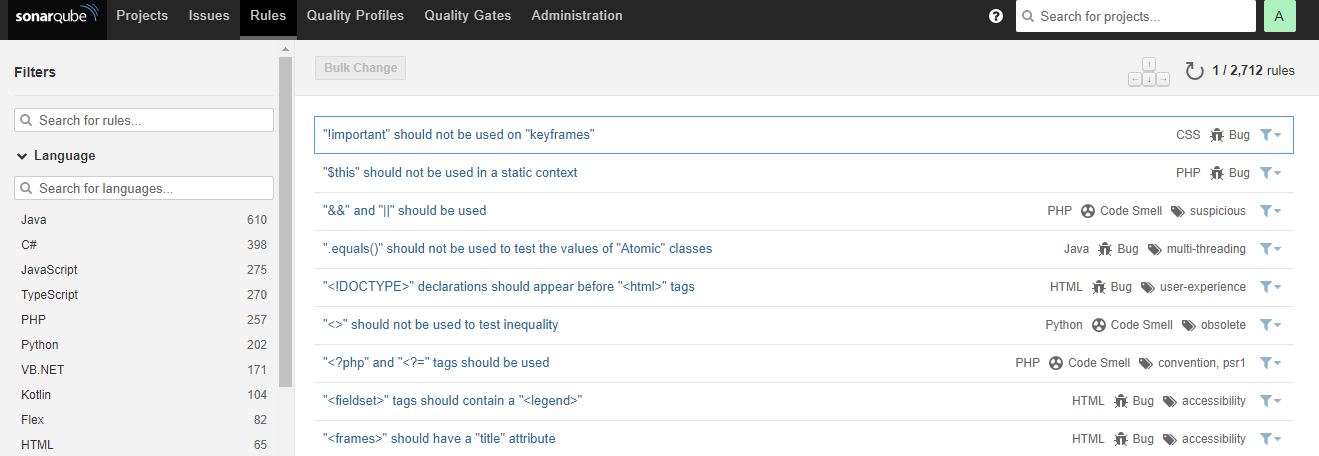
-Dsonar.host.url=http://localhost:9000 \

-Dsonar.login=b42cffa4eaddb18dc83e1fb47d4110f28e907cad



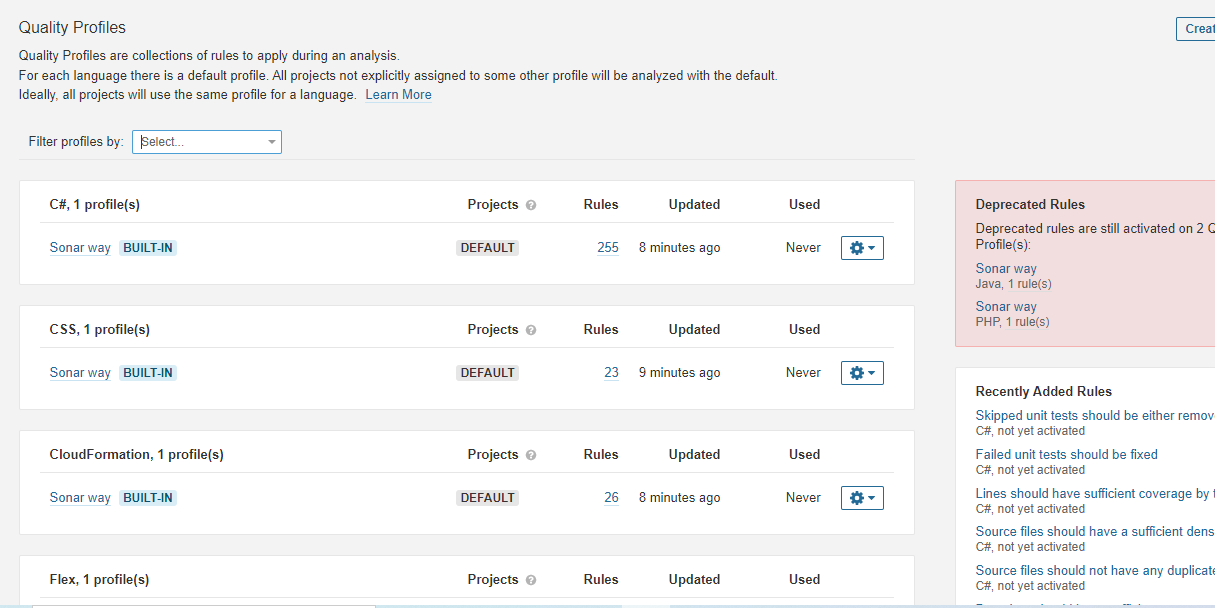


Developers can check the issues and fix the code identified by sonar



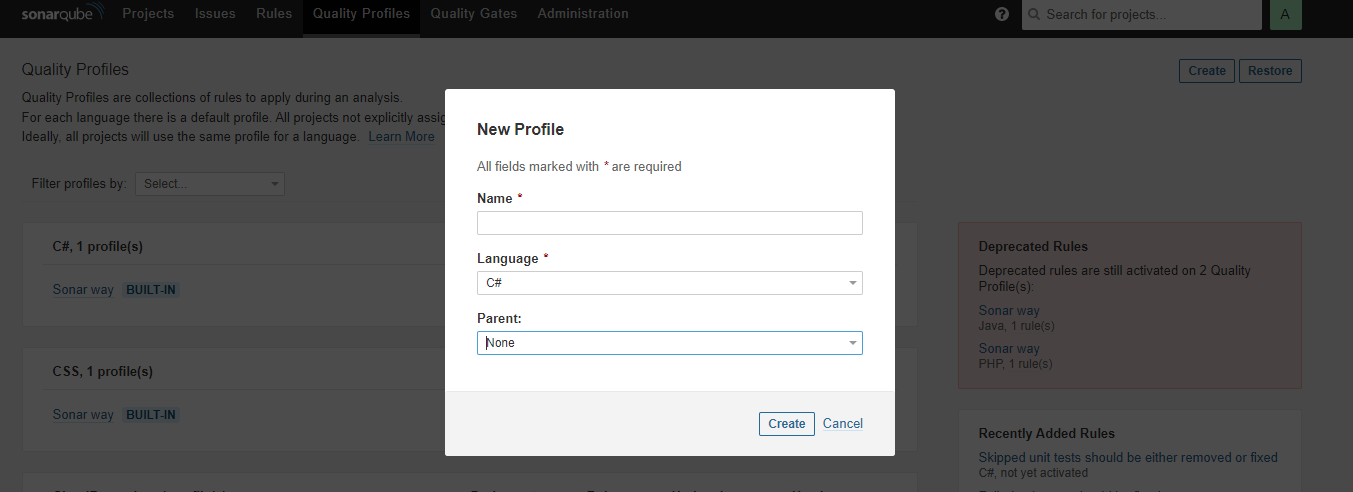
Predefined set of rules for different programming languages

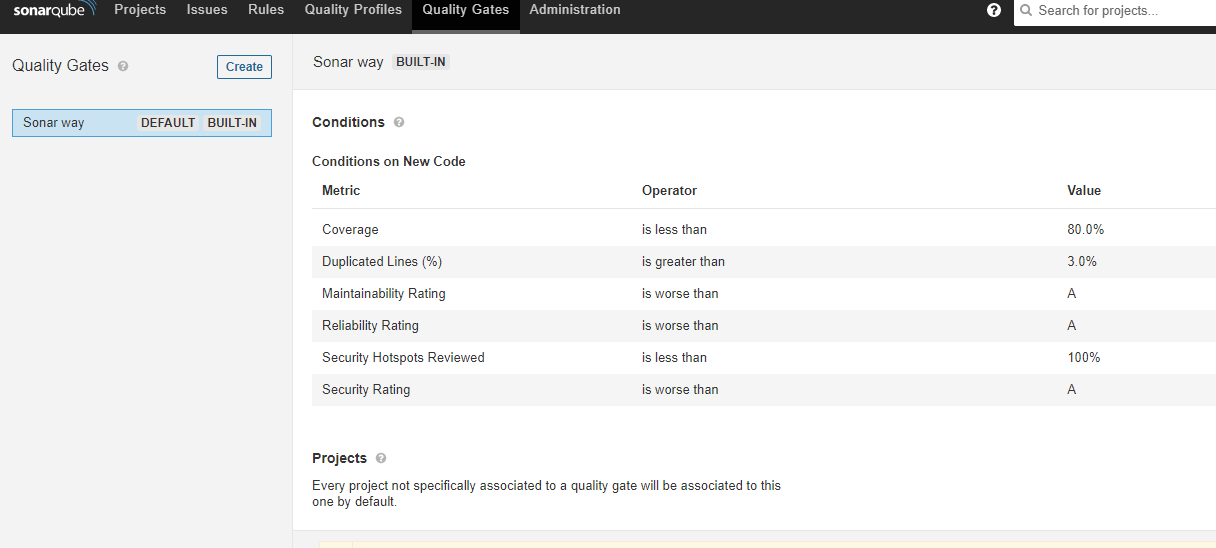
Quality Profiles: Collection of rules



Not all the rules are defined in default quality profile

We can create new profile by clicking Create profile





Quality gates tells that how many bugs are there after running sonar, we can provide the threshold value

If you are having maven then no need to execute sonar scanner as by default with maven sonar comes

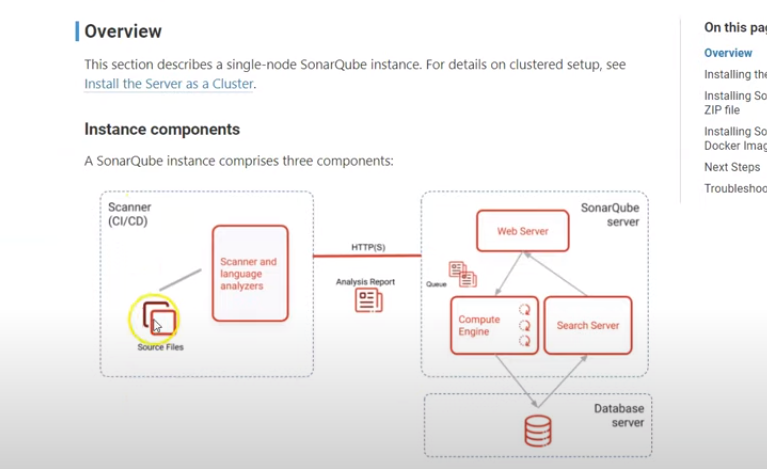
Run below command on source code system project

mvn clean verify sonar:sonar \

-Dsonar.projectKey=mavenproject \

-Dsonar.host.url=http://localhost:9000 \

-Dsonar.login=b42cffa4eaddb18dc83e1fb47d4110f28e907cad



Databases are postgre sql , Oracle, Microsoft sql server

In Jenkins install sonarcube scanner plugin

We can integrate AWS RDS database for posrtgre or Microsoft for sonarqube connection

/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.312.b07-1.amzn2.0.2.x86\_64

