CICD PIPELINE BY USING DECLARATIVE SCRIPT IN JENKINS

- By using pipeline script, we can integrate the different tools in devops.
- The time will be saving in developing the project.
- Continuously integrate and continuously deploy the project by using pipeline script.

JENKINS SHOULD HAVE A NODE:

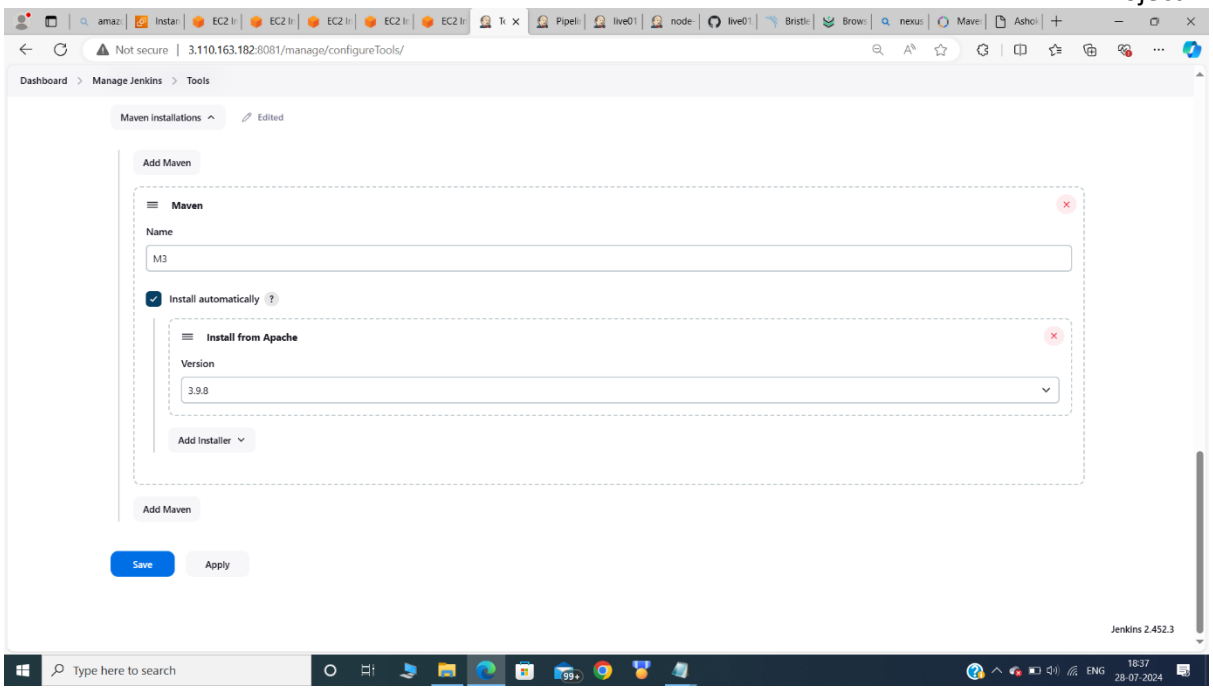
- For the load and work division purpose we can use the node or agent.
- **Dashboard>manage Jenkins>nodes,** here we can add the node and node credentials also.
- In script we can take the agent following script.

```
pipeline {
    agent any
        (or)
    pipeline {
        agent {label 'dev'}
```

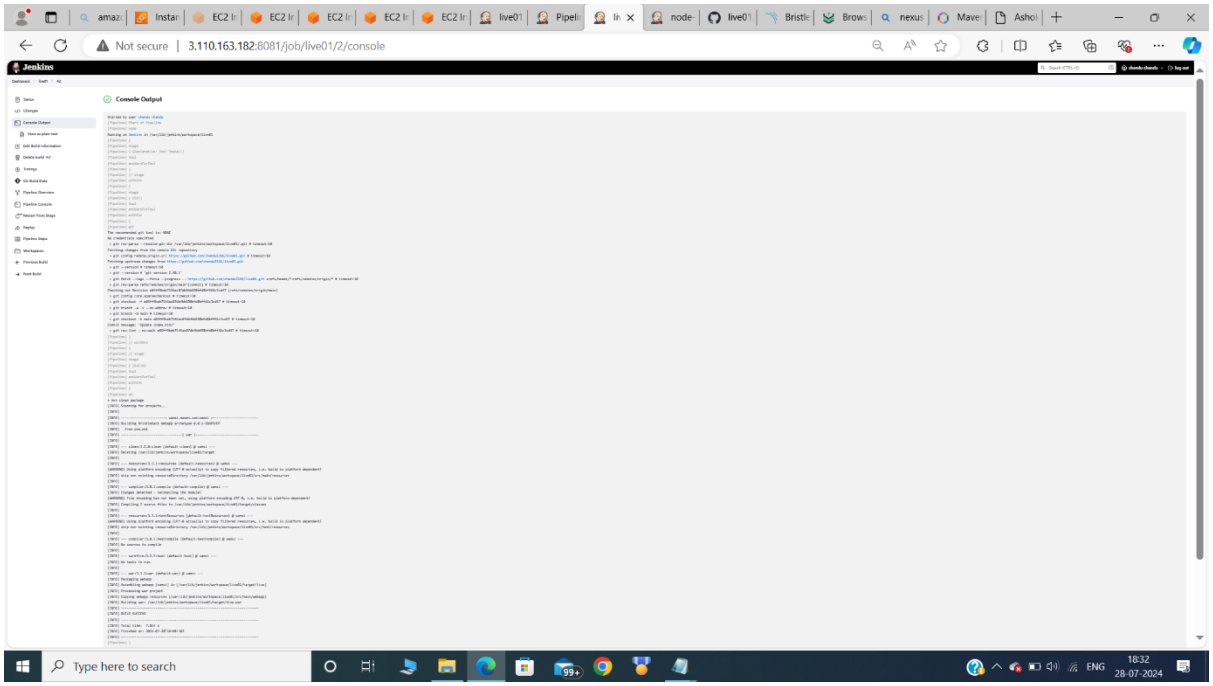
- If we give agent any the active node will take automatically.

GET THE REPOSITORY CODE FROM GIT HUB:

- First, we can start the Jenkins server and select the new item.
- Give name and select the pipeline option and click on ok.
- The declarative pipeline starts with (**pipeline {**).
- By using git plugin, we can add the GIT repository details as requirements shown.
- Second stage we can do the build steps here we can add the maven like following the steps.
- For deployment we can add the tools below the pipeline as shown in the fig.
- Now go to the dashboard>manage Jenkins>tools here, we can add the maven details here name and version and same name add the tools.



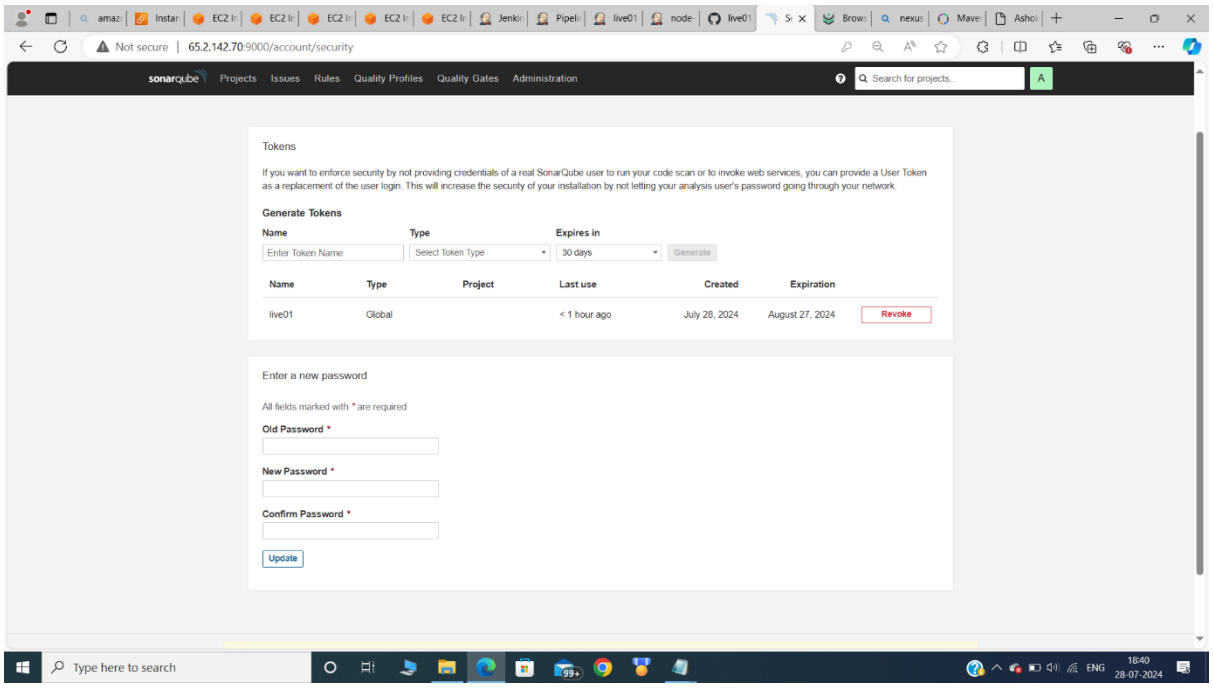
- Now we click on save and click on build now the out put will be shown in the console output.



INTEGRATE WITH THE SONARQUBE TOOL:

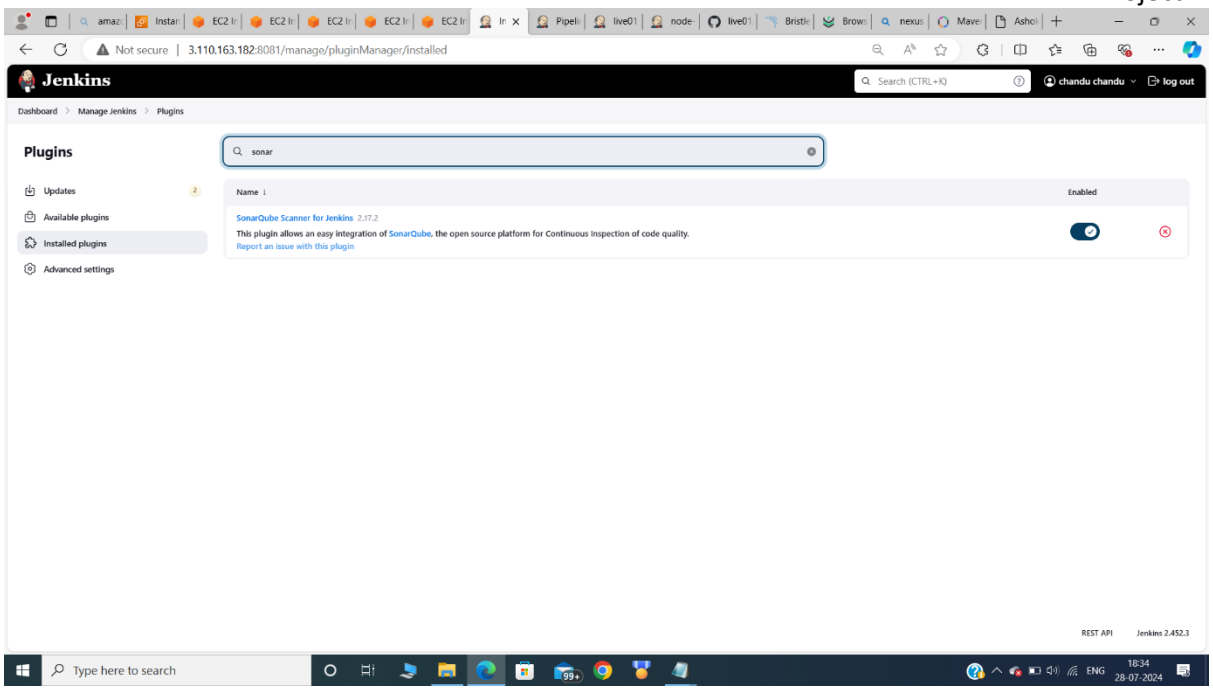
- For this integration we can take the different server and download the SonarQube.
- Start the SonarQube server with allocated port number and login the credentials.

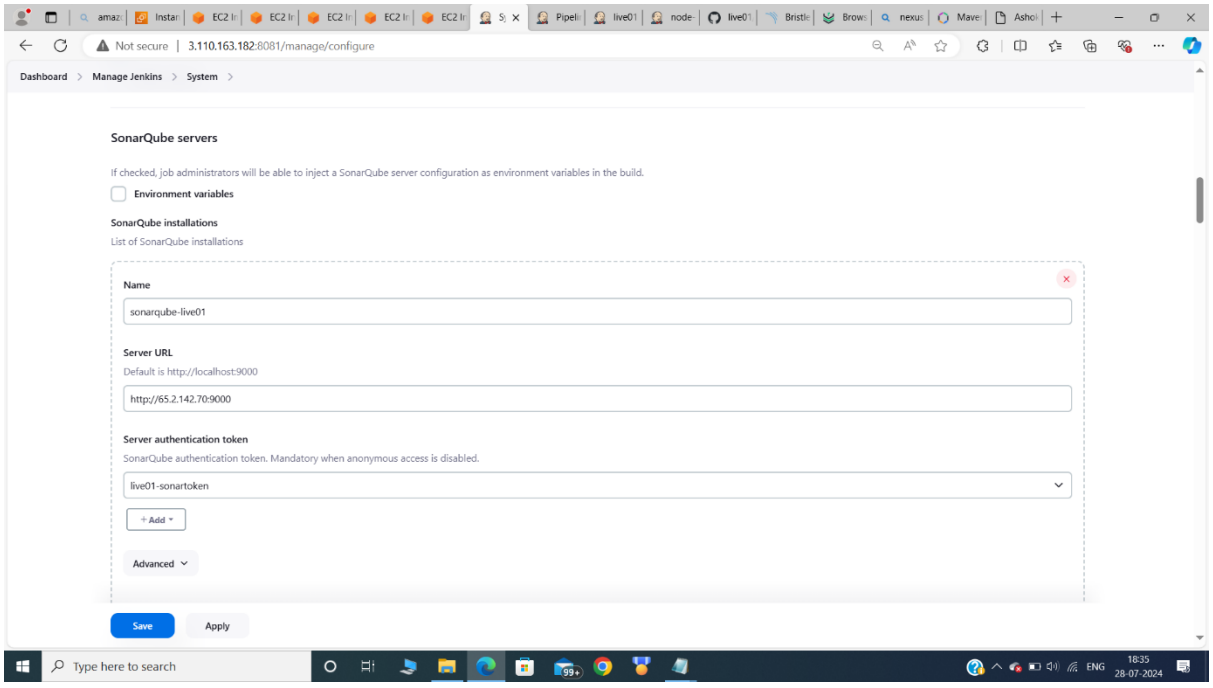
- `Here we can create the token in the SonarQube.
- Click on **administration>security>users.**

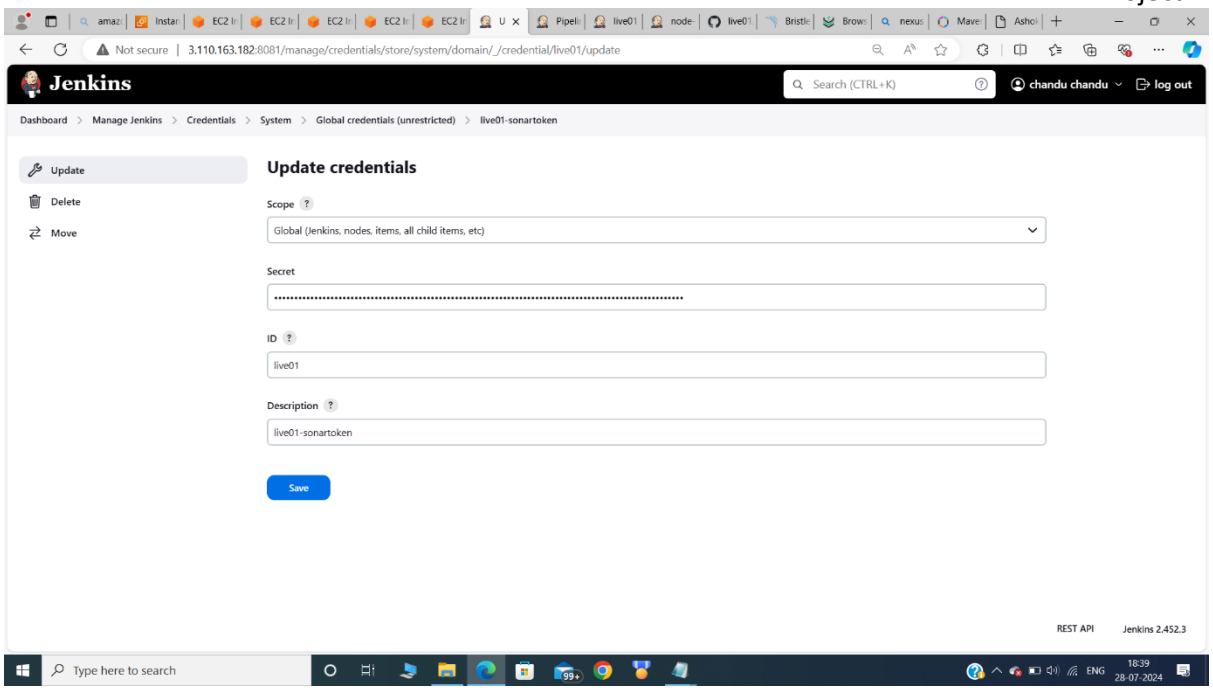


- Here we can create the token copy the token, paste in the node pad for the further use.
- Now go to Jenkins we can download the **SonarQube Scanner plugin. Dashboard>manage Jenkins>plugins.**
- Now we can go to the Jenkins add the SonarQube details in the **dashboard>manage Jenkins>system.**
- Under the SonarQube servers we can add the details about SonarQube as show in the fig.
- We can add the token of the SonarQube in the **dashboard>manage Jenkins>credentials** and click on global credentials and next click on the add credentials as show in the image.

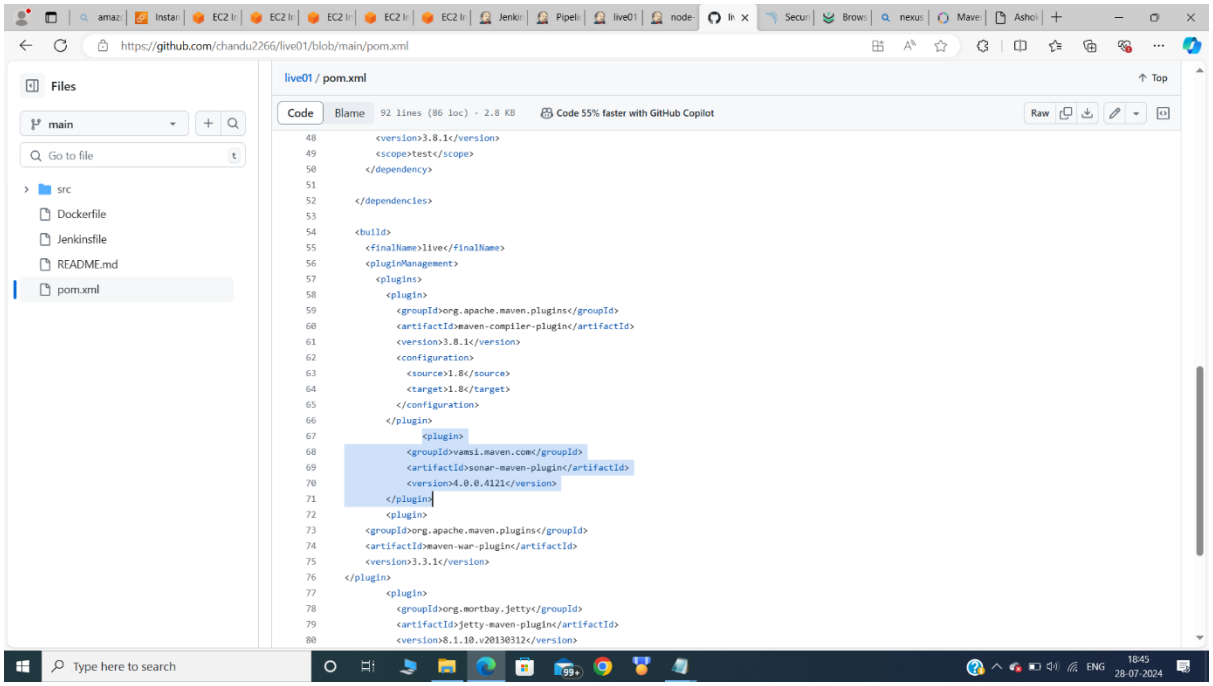
  Note: here we can add the SonarQube token by using secret text in kind.


Not secure | 3.110.163.182:8081/manage/pluginManager/installed
Jenkins
Search (CTRL+K)
chandu chandu
log out
Dashboard > Manage Jenkins > Plugins
Plugins
sonar
Updates
Available plugins
Installed plugins
Advanced settings
Name
Enabled
SonarQube Scanner for Jenkins 2.17.2
This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.
Report an issue with this plugin
REST API
Jenkins 2.452.3
Type here to search
ENG
18:34
28-07-2024

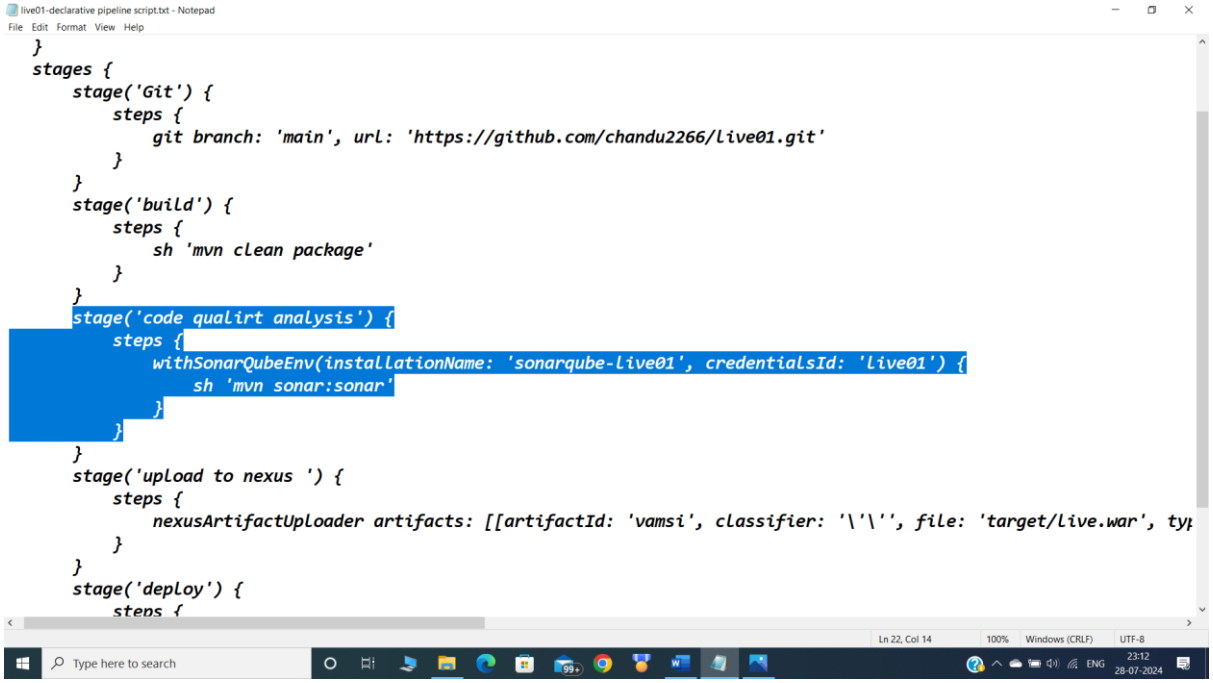

Not secure | 3.110.163.182:8081/manage/configure
Dashboard > Manage Jenkins > System
SonarQube servers
If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.
Environment variables
SonarQube installations
List of SonarQube installations
Name
sonarqube-live01
Server URL
Default is http://localhost:9000
http://65.2.142.70:9000
Server authentication token
SonarQube authentication token. Mandatory when anonymous access is disabled.
live01-sonartoken
+ Add
Advanced
Save
Apply
Type here to search
ENG
18:35
28-07-2024



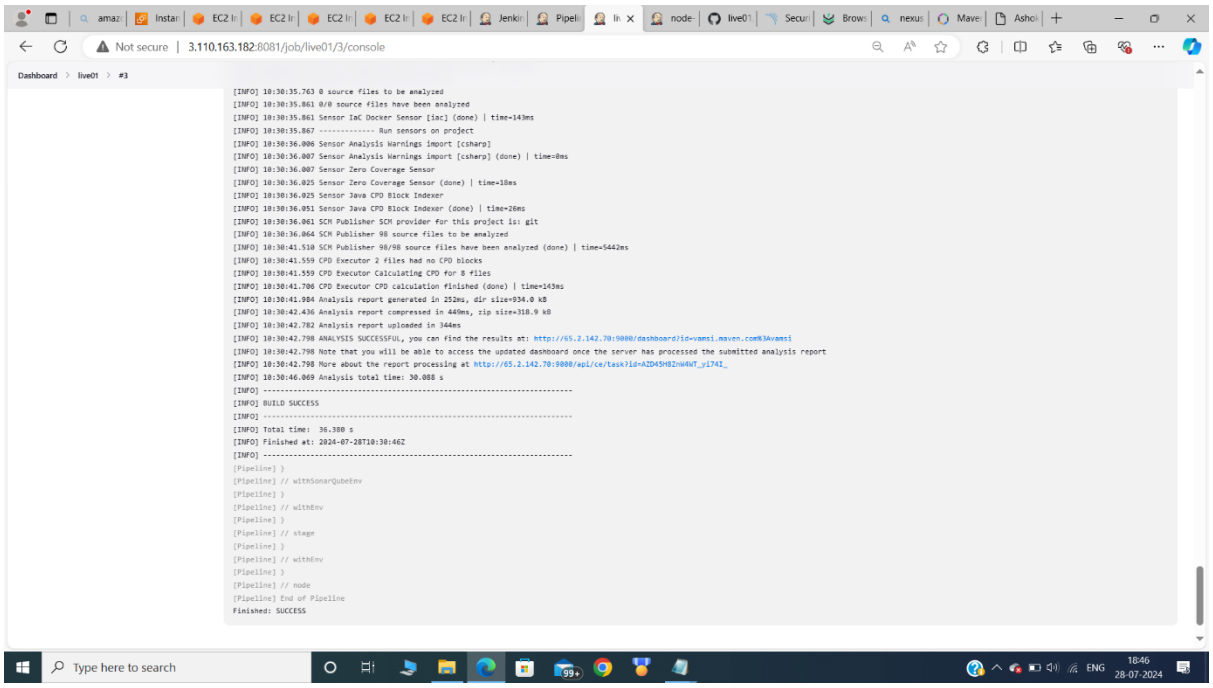
- Now we can go to the git hub -we selected git repository(eg:live01) add the plugin in the plugin management.as shown in the fig.



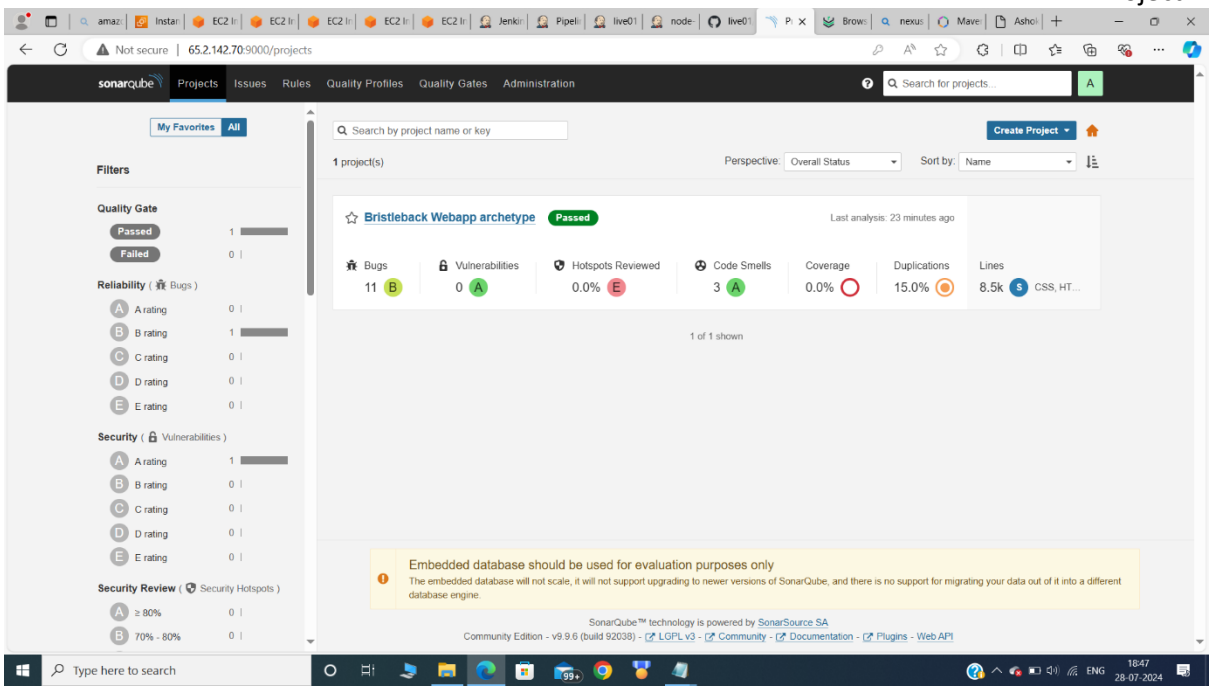
- Now come to the job click on configure, here we can add the next stage for SonarQube.
- Click on pipeline syntax select the with SonarQube scanner plugin add the requirement details. click on generate pipeline script. copy the script and paste in script.
- Here we can add some line as shown in the image.

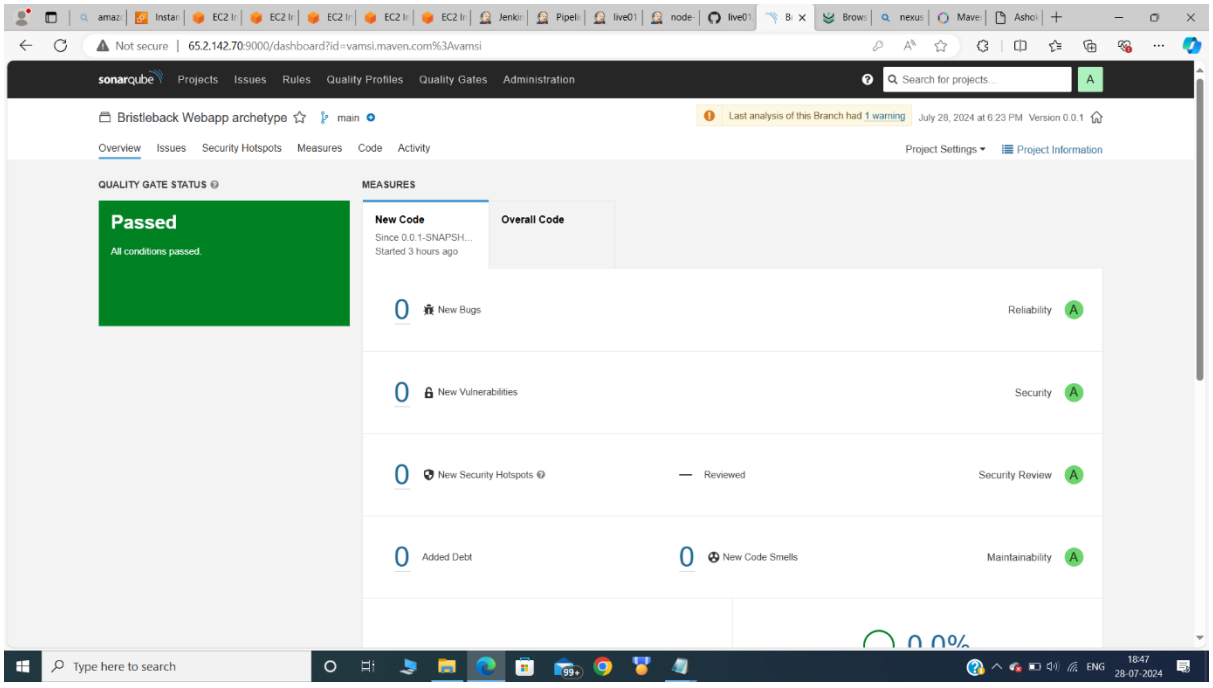


- Installation name is taken which given for the system settings for SonarQube server.
- After completed the script click apply and save. Click on build now. The build we success as shown in the fig.



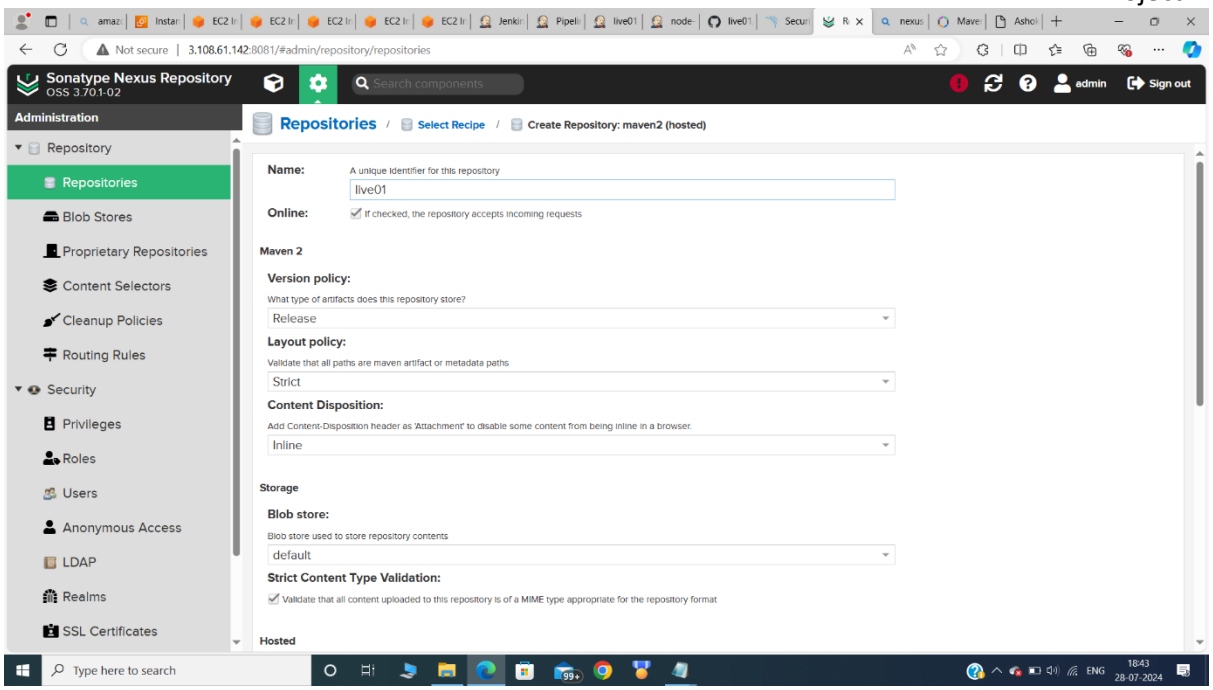
- Now we can go to SonarQube server refresh it the output we will see here. As shown in the fig.



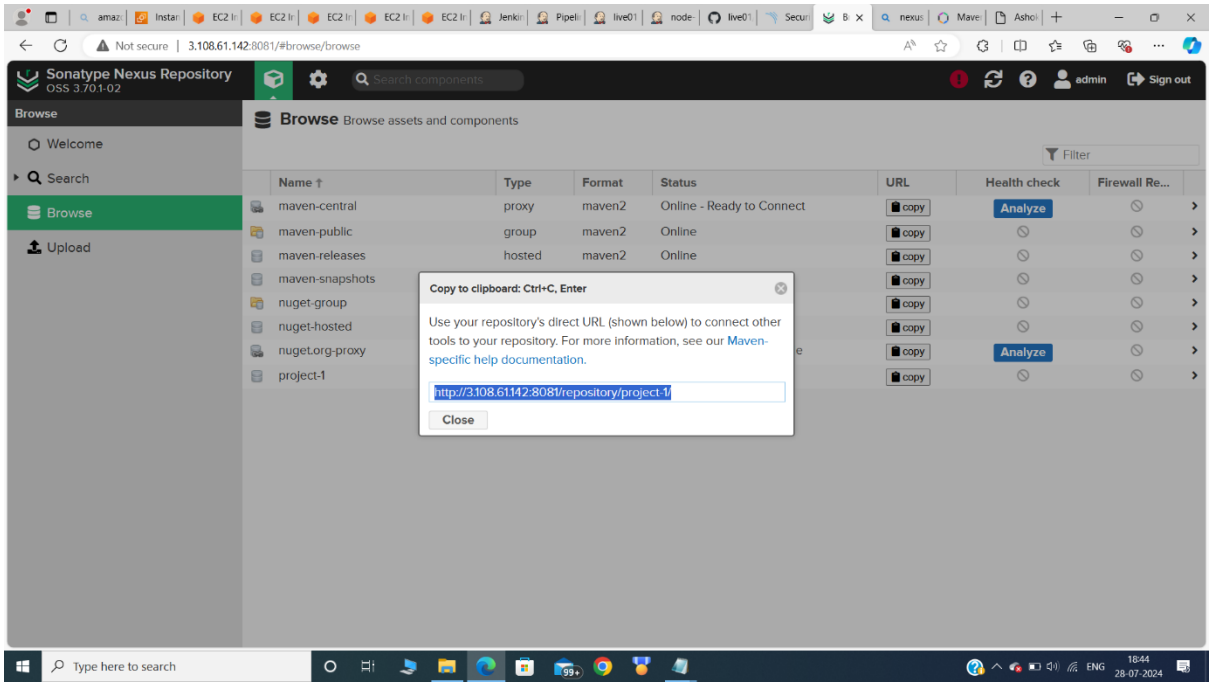


UPLOAD THE ARTIFACT TO NEXUS REPOSITORY:

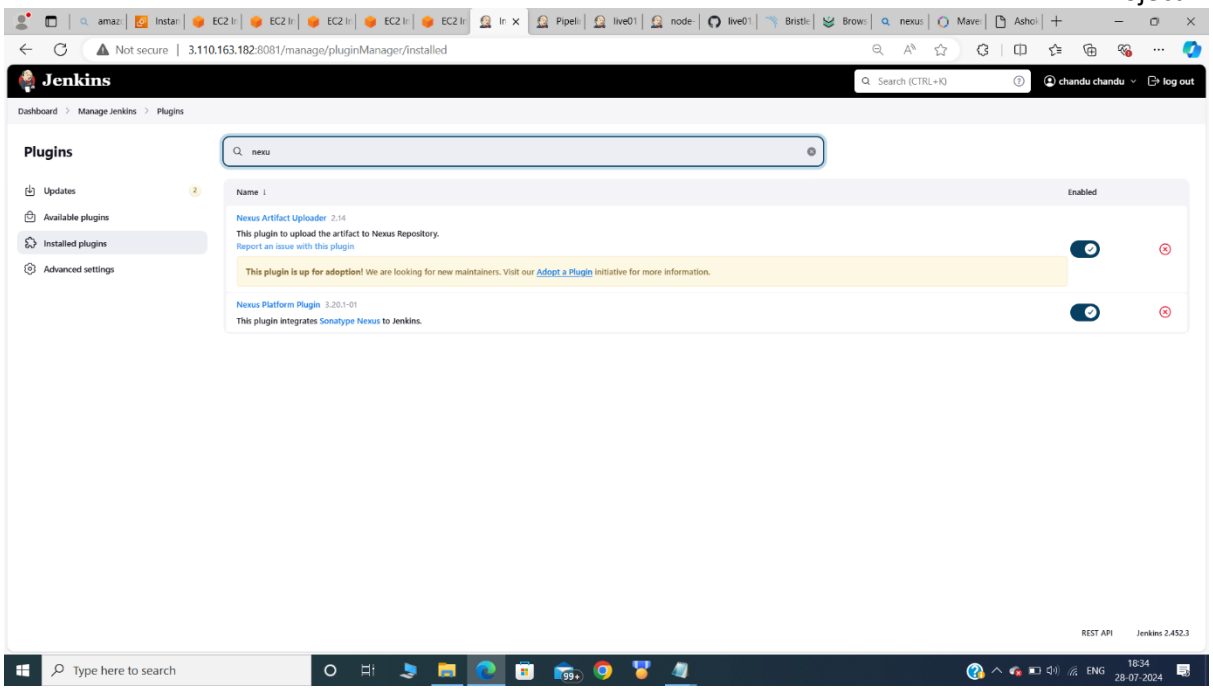
- For nexus tool we can install the nexus in the new server and start it, open the nexus and login.
- Go to the settings click on create the repository here, and add the details as show in the fig.



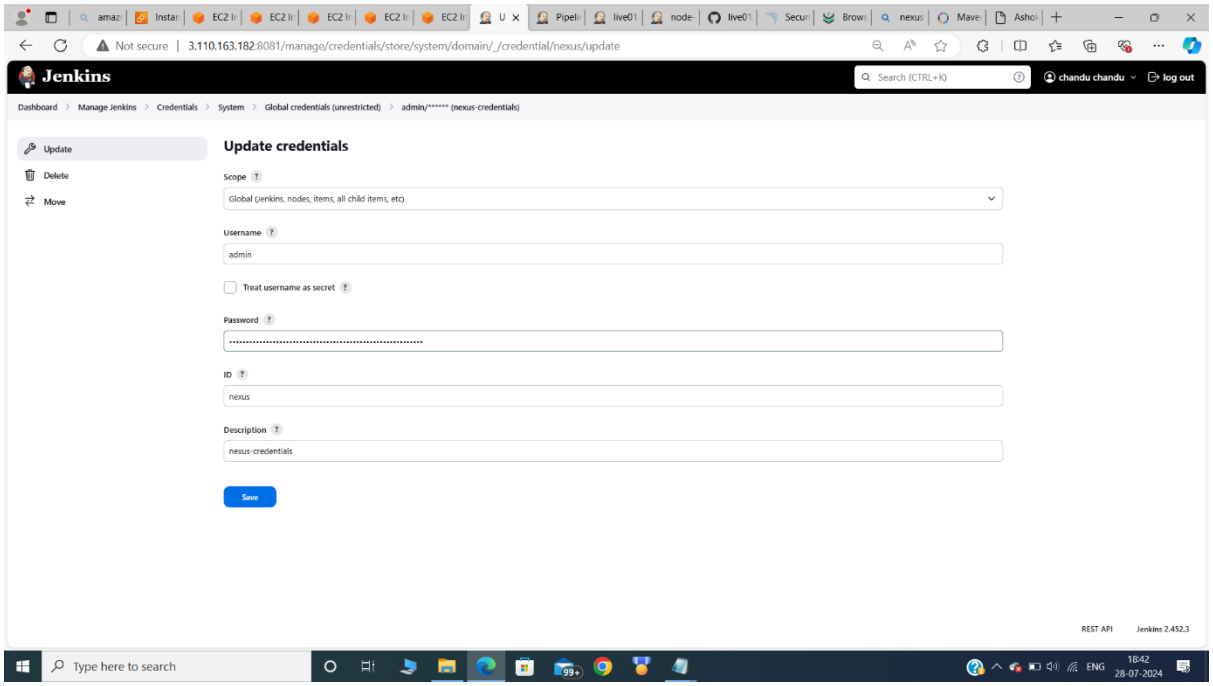
- After creating repository click on the copy and copy the URL and paste in notepad for the further use.



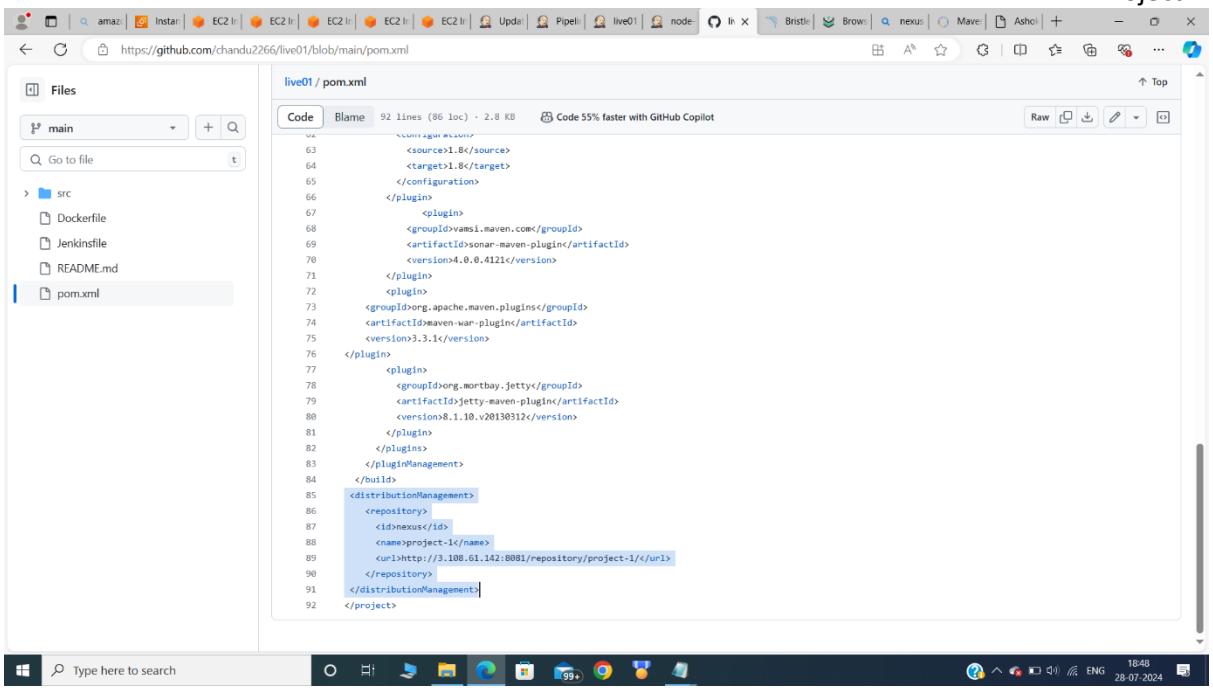
- Now come to the Jenkins we can install the plugin **nexus artifact uploader.** as shown in the fig.



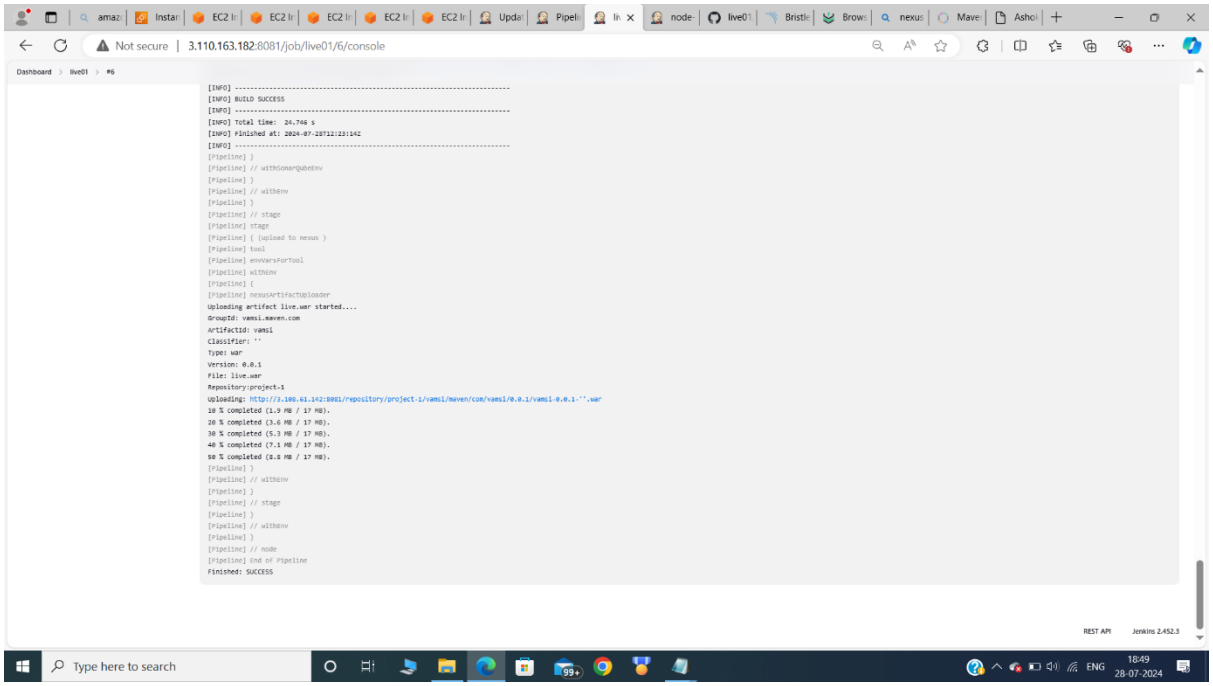
- Now we can add the credentials in Jenkins for the nexus. here we can add the nexus login details (username and password).



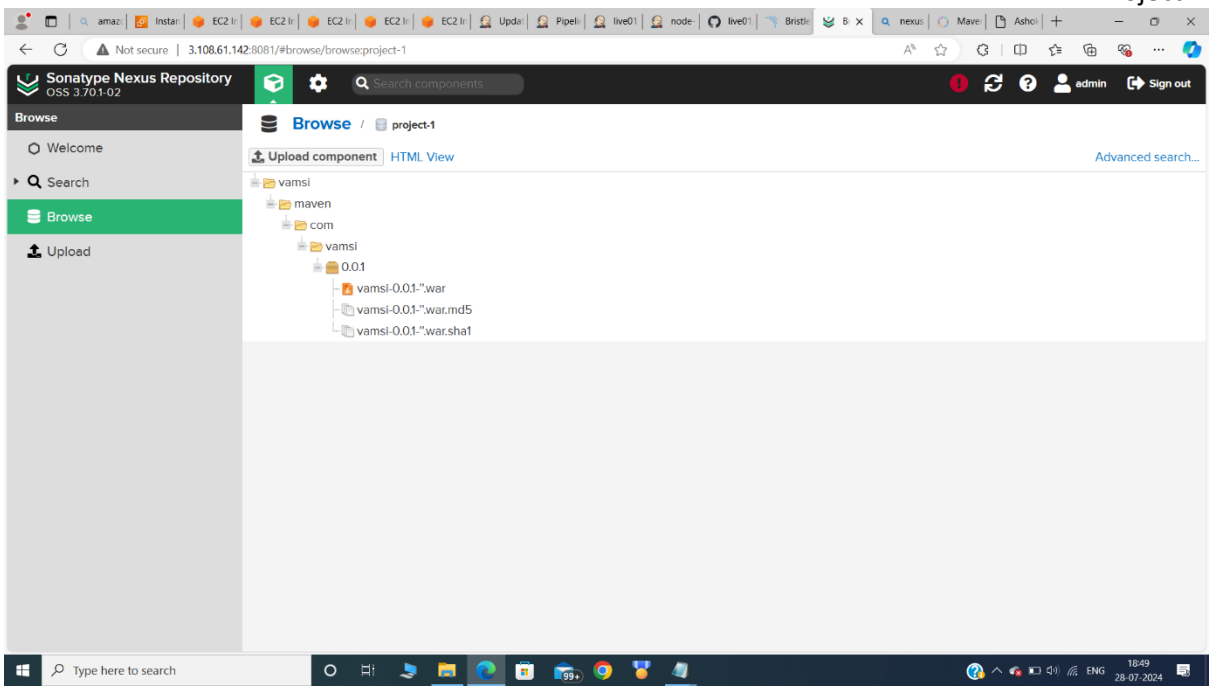
- Now we can edit the pom.xml file in the selected repository.as show in the fig.



- Now we can come to Jenkins configure the job (eg: live01) add the next stage for the nexus artifact.
- Click on pipeline syntax select the nexus artifact upload plugin add the requirements.
  Note: **File (**is taken from the first build) eg:target/live.war and Remaining options taken from pom.xml
- After completing the script save and apply. Click on build now the console output will come as shown in the fig.

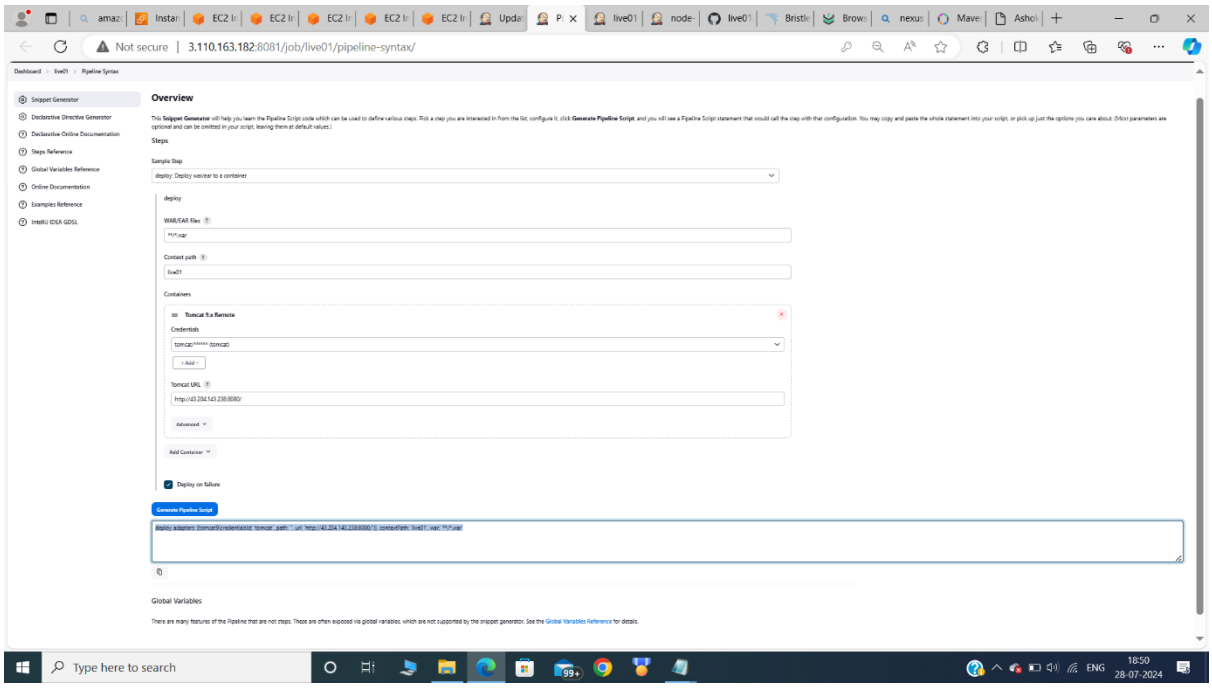


- Now we can go to nexus server and refresh it. click on our created repository.

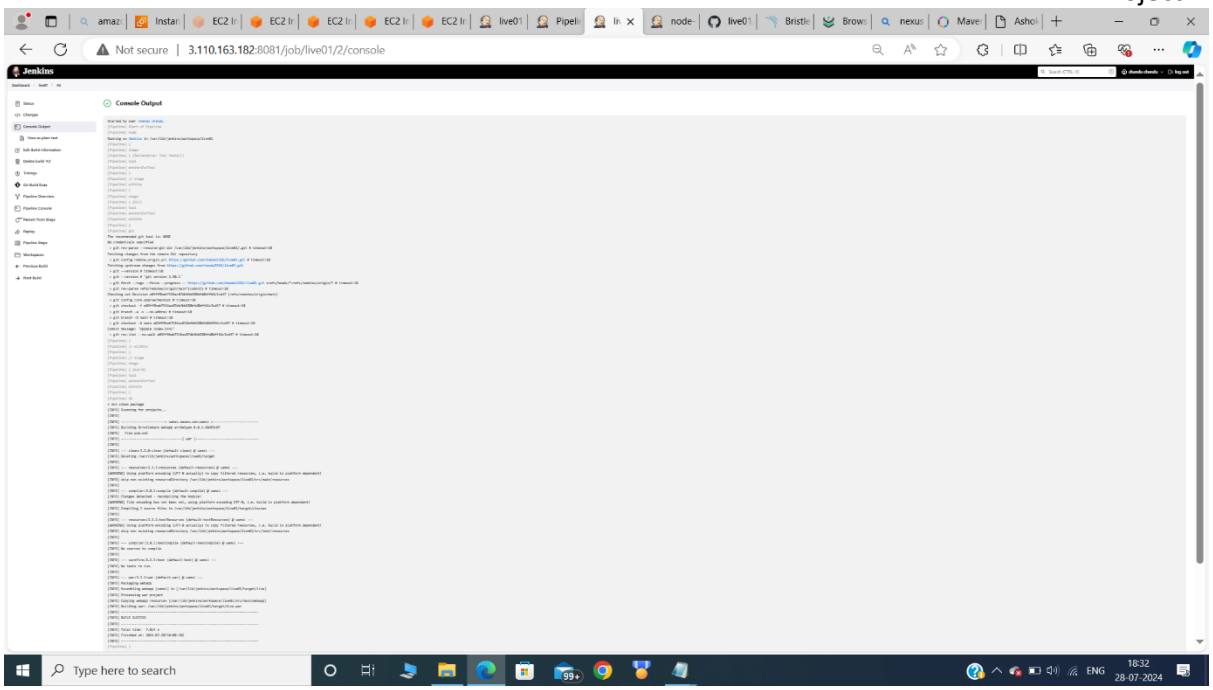


DEPLOY THE ARTIFACT TO THE TOMCAT SERVER:

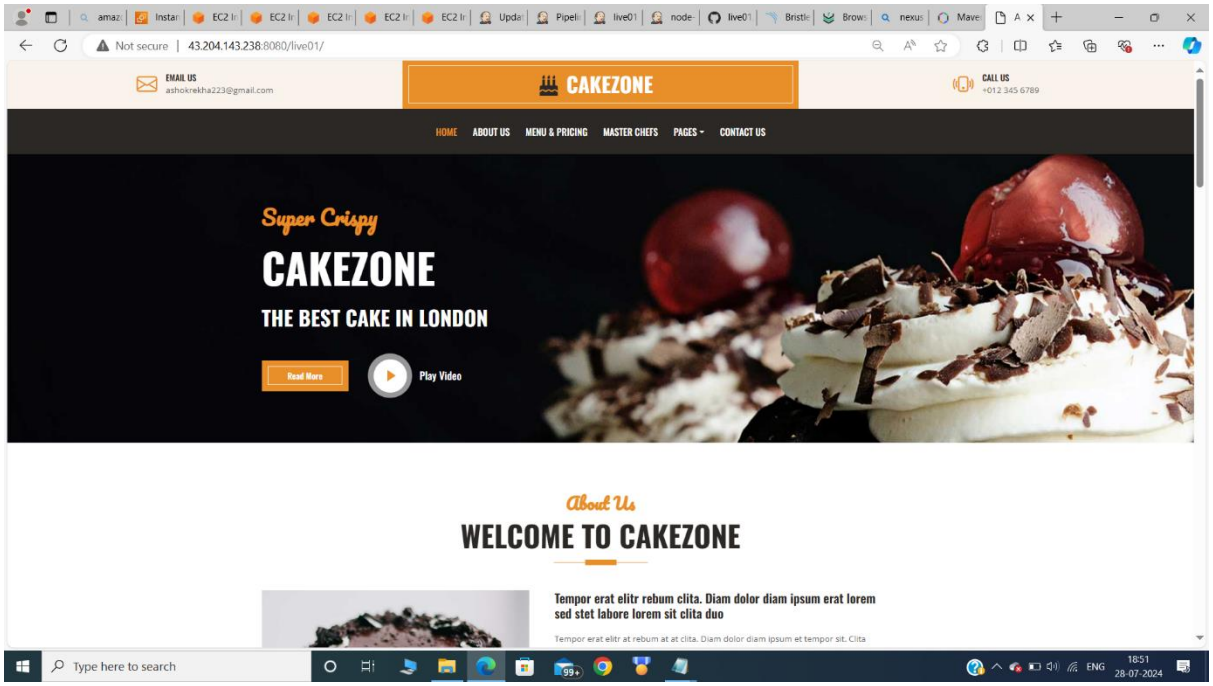
- Now we can take a new instance for the tomcat and install tomcat server.
- Add the permissions open the manager and GUI script options also.
- Now come to the Jenkins install the deploy plugin and add the tomcat credentials in credentials in the Jenkins.
- Now come to the job and configure it add the next stage for the deploy and click on the pipeline syntax select the deploy plugin and give the details for tomcat.
- Click on generate pipeline script. Copy it and paste in script.



- Now click on build now after click on save and apply.



- Now go to the tomcat server add the path which giving in the tomcat script(eg:live01).click on enter.



- We can see the total script like this.

```
pipeline {
   agent any
   tools {
      maven 'M3'
   }
   stages {
      stage('Git') {
         steps {
            git branch: 'main', url: 'https://github.com/chandu2266/live01.git'
         }
      }
      stage('build') {
         steps {
            sh 'mvn clean package'
         }
      }
      stage('code qualirt analysis') {
         steps {
            withSonarQubeEnv(installationName: 'sonarqube-live01', credentialsId: 'live01') {
               sh 'mvn sonar:sonar'
            }
         }
      }
      stage('upload to nexus ') {
         steps {
            nexusArtifactUploader artifacts: [[artifactId: 'vamsi', classifier: '\'\'', file: 'target/live.war',
type: 'war']], credentialsId: 'nexus', groupId: 'vamsi.maven.com', nexusUrl: '3.108.61.142:8081',
nexusVersion: 'nexus3', protocol: 'http', repository: 'project-1', version: '0.0.1'
         }
      }
      stage('deploy') {
         steps {
            deploy adapters: [tomcat9(credentialsId: 'tomcat', path: '', url:
'http://43.204.143.238:8080/')], contextPath: 'live01', war: '**/*.war'
         }
      }
   }
}
```