**Task 5. CI-CD Pipeline**

**Building a CI/CD pipeline with Jenkins**

**Six steps to build a pipeline using Jenkins**

We need Java development kit and knowledge in Linux commands.

**STEP 1:- DOWNLOAD JENKINS**

Download Jenkins from the Jenkins downloads page <https://www.jenkins.io/download/>

**STEP 2:- EXECUTE JENKINS AS A JAVA BINARY**

Open the terminal window and enter cd <your path>

Use the command java -jar ./Jenkins. war to run the WAR file

**STEP 3:-CREATE JENKINS JOB**

Open the web browser and open localhost:8080

The Jenkins dashboard that opens, create new jobs there

**STEP 4:-CREATE A PIPELINE JOB**

Select and define Jenkins job that should be created.

Select a pipeline and give it a name.

Write pipeline script or retrieve the Jenkins file from source code management.

**STEP 5:- CONFIGURE AND EXECUTE A PIPELINE JOB WITH A DIRECT SCRIPT.**

Choose a pipeline script as the destination and paste the jenkinsfile content in the script from GitHub then save. Then build it using build now option. To check the output, click on stage and then log.

**STEP 6:- CONFIGURE AND EXECUTE A PIPELINE JOB WITH SCM**

Copy the GitHub repository URL by clicking on download and click on Configure to modify the existing job.

Scroll to the Advanced Project Options setting, and select Pipeline script from SCM option.

Paste the GitHub repository URL here .

Type Jenkinsfile in the Script, and then click on Save button.

Next, click on Build Now to execute the job again.

There will be an additional stage in this case, i.e. Declaration: Checkout SCM

Click on any stage and click on Log

**CODE:-**

node(){

def sonarHome = tool name: 'SonarScanner', type: 'hudson.plugins.sonar.SonarRunnerInstallation'

stage('Code Checkout'){

checkout changelog: false, poll: false, scm: scmGit(branches: [[name: '\*/master']], extensions: [], userRemoteConfigs: [[credentialsId: 'GitHubCreds', url: 'https://github.com/anujdevopslearn/MavenBuild']])

}

stage('Build Automation'){

sh """

ls -lart

mvn clean install

ls -lart target

"""

}

stage('Code Scan'){

withSonarQubeEnv(credentialsId: 'SonarQubeCreds') {

sh "${sonarHome}/bin/sonar-scanner"

}

}

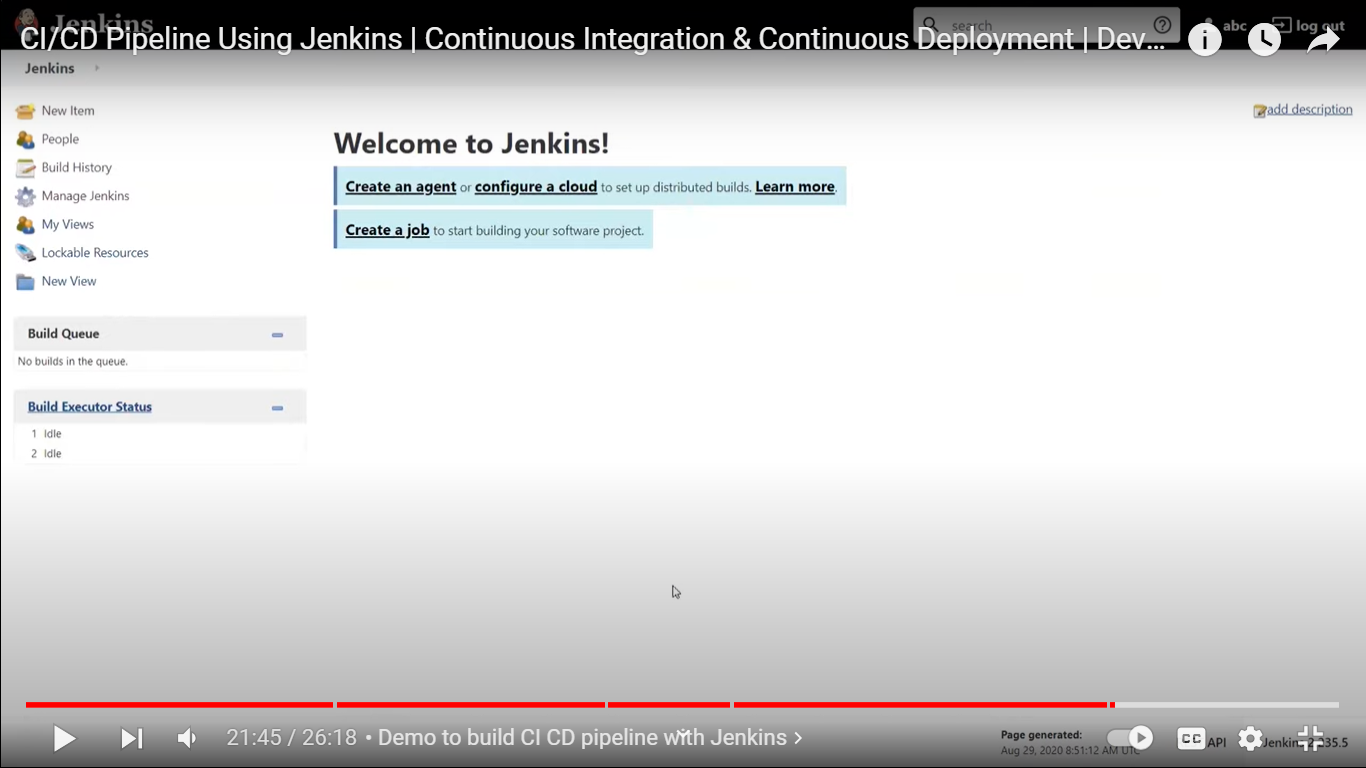
stage('Code Deployment'){

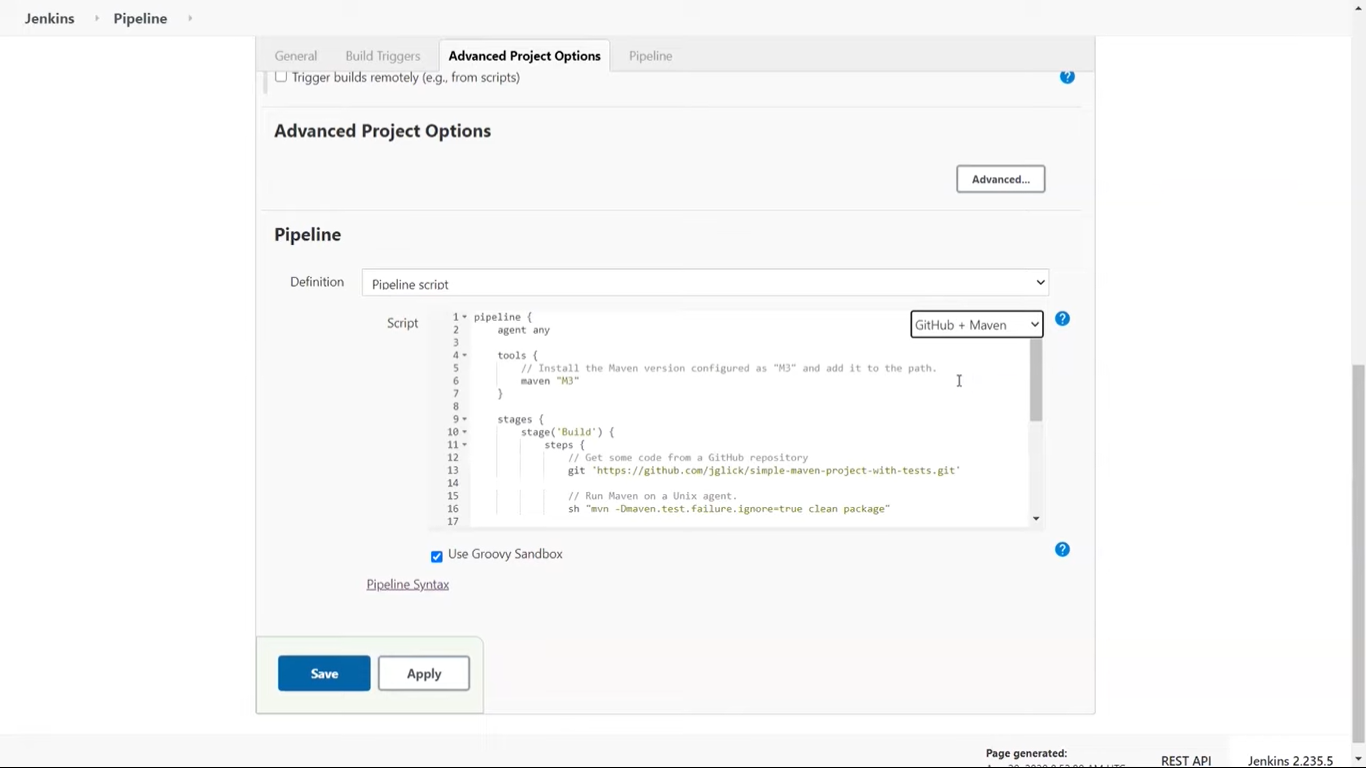
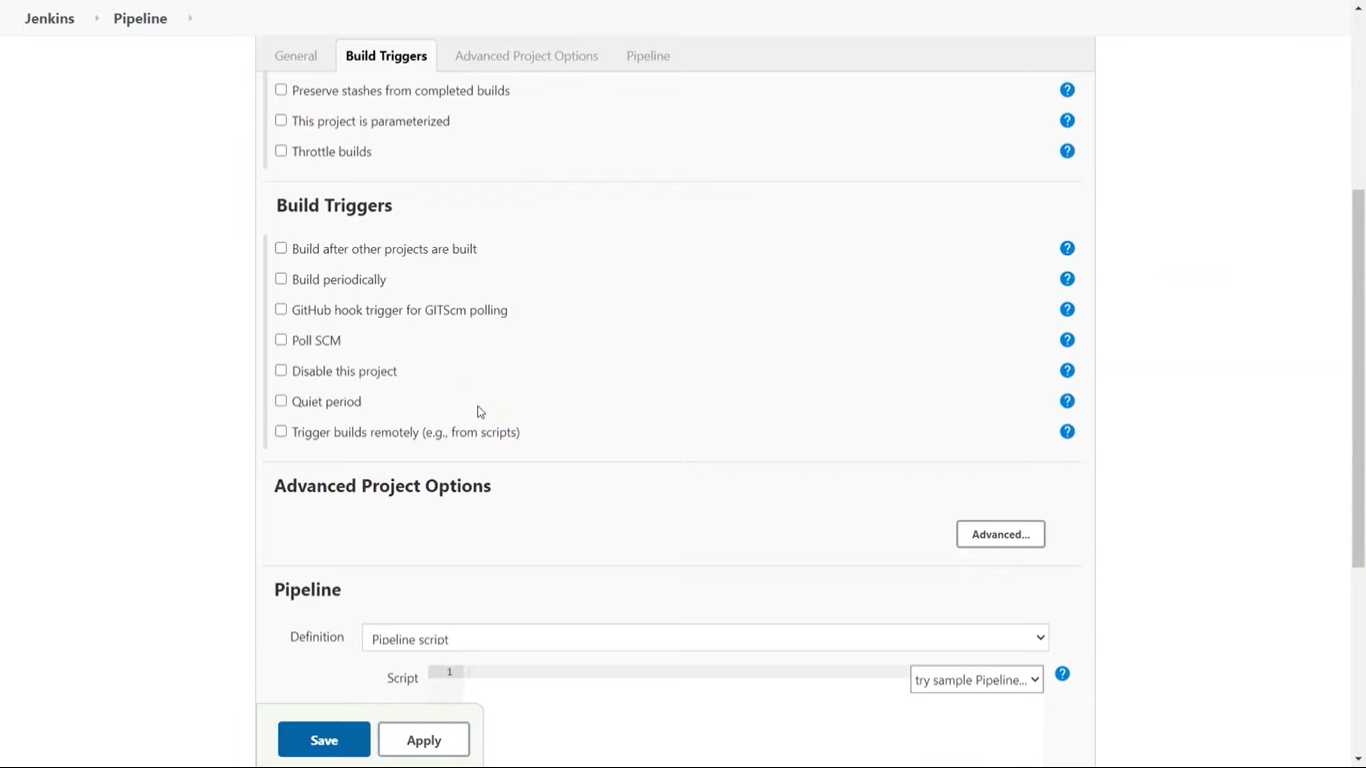
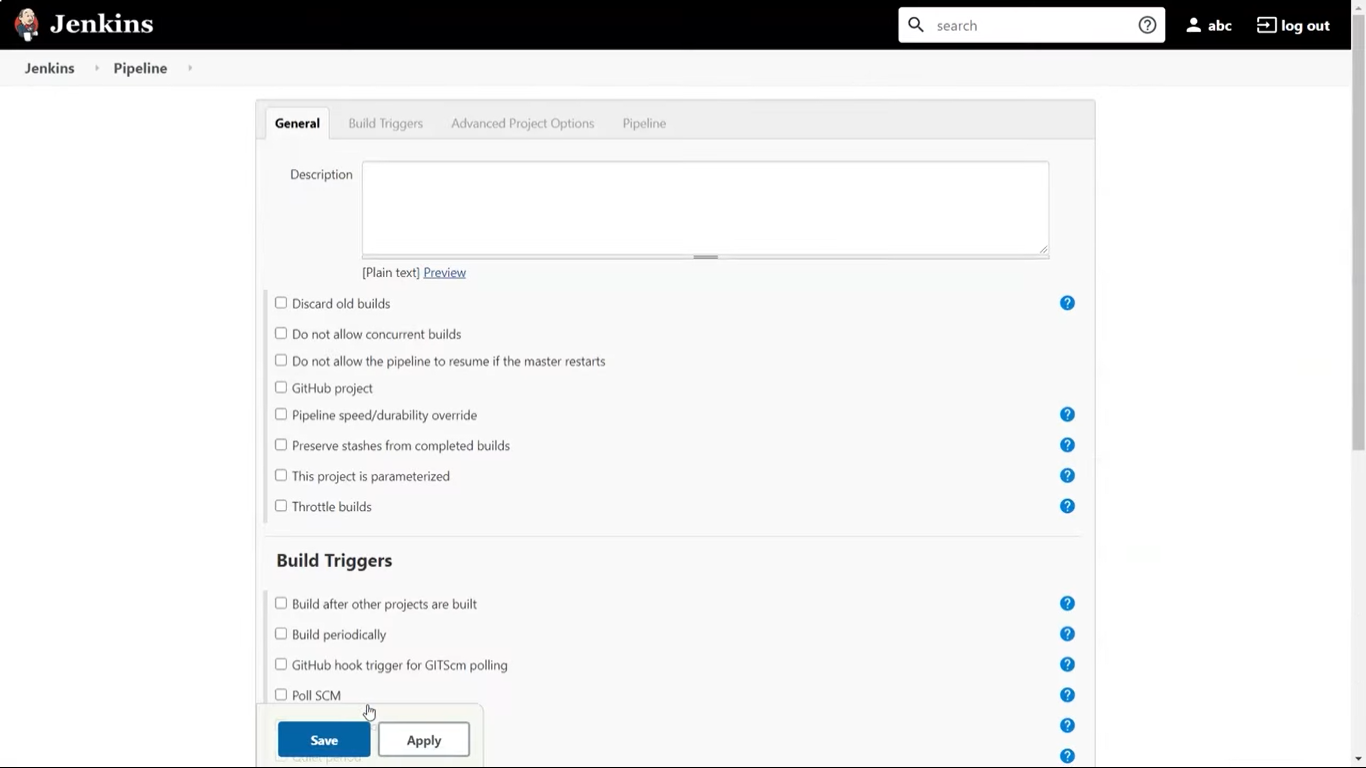
deploy adapters: [tomcat9(credentialsId: 'TomcatCreds', path: '', url: 'http://54.197.62.94:8080/')], contextPath: 'Planview', onFailure: false, war: 'target/\*.war'

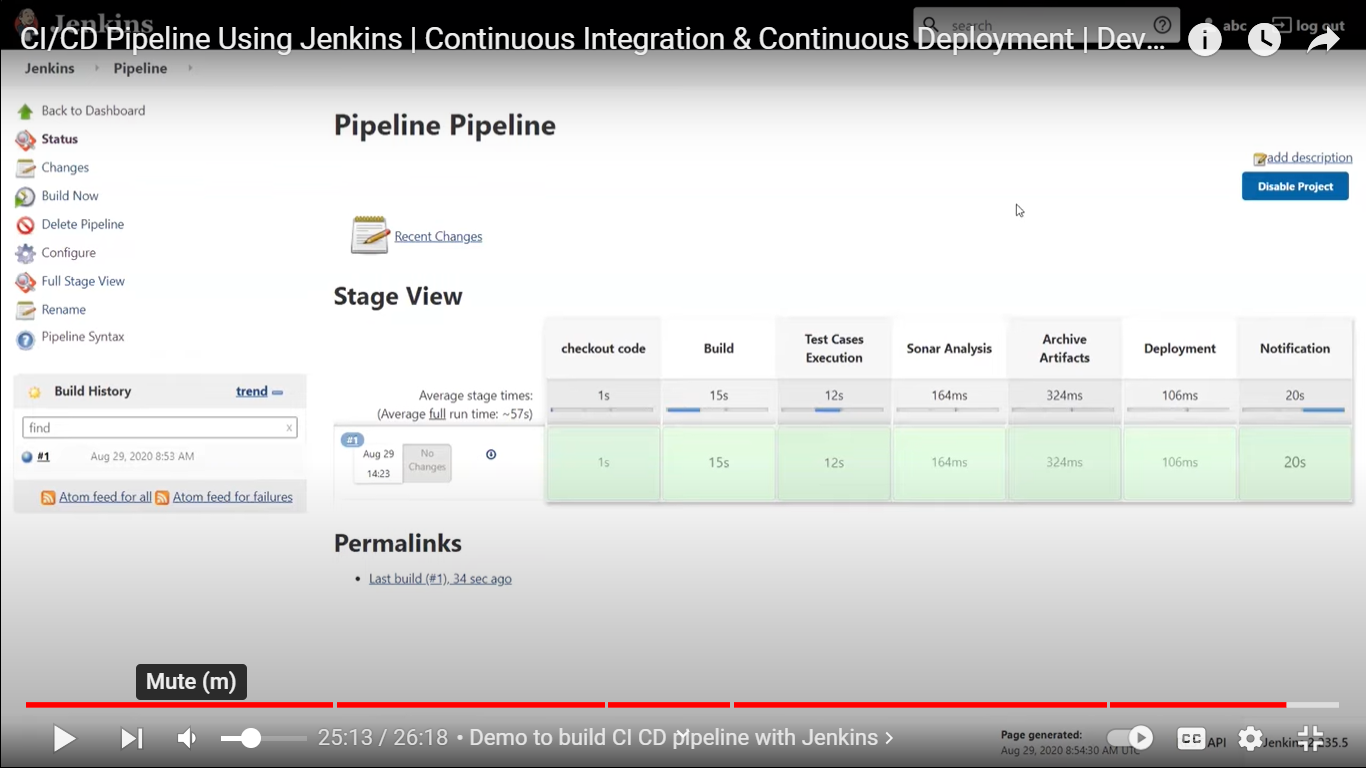
}

}

**SCREENSHOTS:-**





****