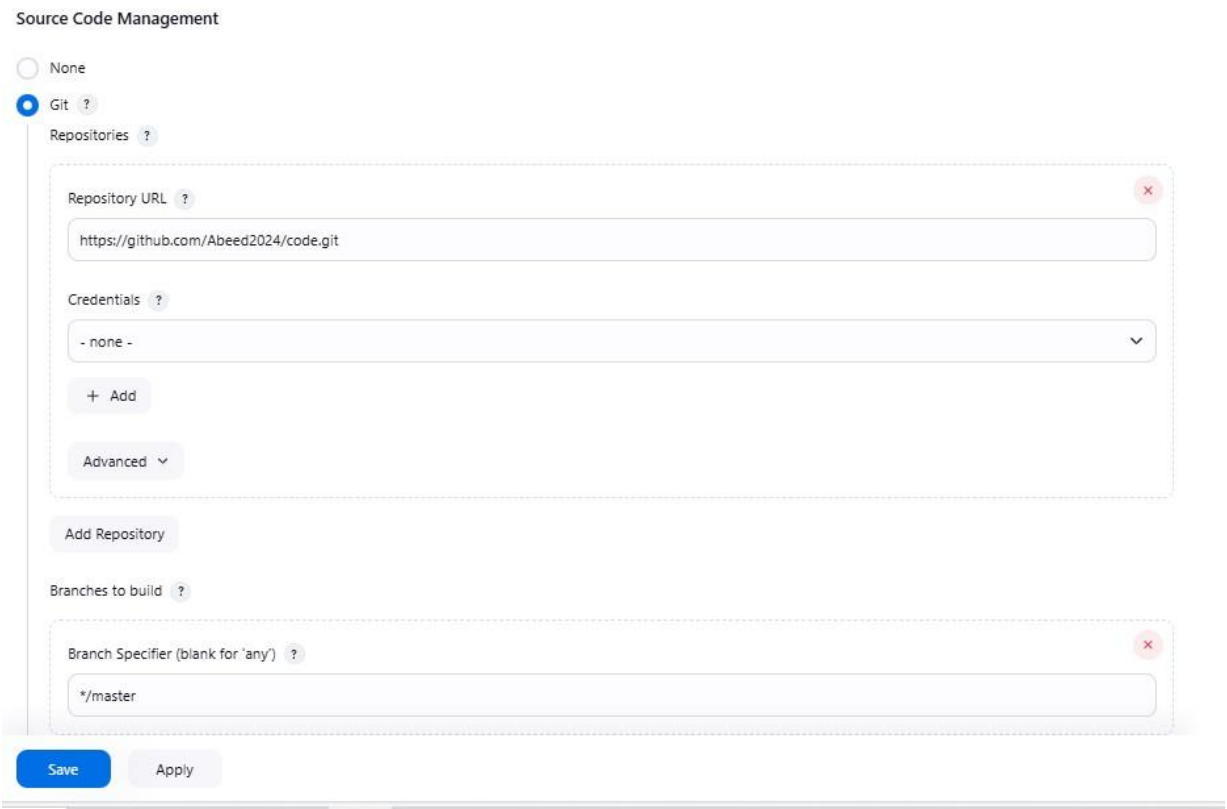
# DEPLOYMENT WITH ALL THE AUTOMATION TOOLS

## Step 1:

1. Launch the server for Jenkins
    - Update the server
    - Install the java 17 version
    - Install Jenkins
    - Start the Jenkins and see the status of Jenkins
2. Login in to Jenkins server
    - Install the suggested plugins
    - Create the job with free style or pipeline
    - If you created the job with free style then initialize the git as shown in below figure



3. If the configure of git is done then perform the build, then the build get success it means work is done
4. Create the job with pipeline
    - By selecting the stage is checkout and the step is created by generating the pipeline syntax
    - After the copying of pipeline syntax and paste in script to perform the build as shown in below figure

```
pipeline {
    agent any

    stages {
        stage('checkout') {
            steps {
                git 'https://github.com/Abeed2024/code.git'
            }
        }
```

5. After that perform the build step with pipeline as shown in below figure

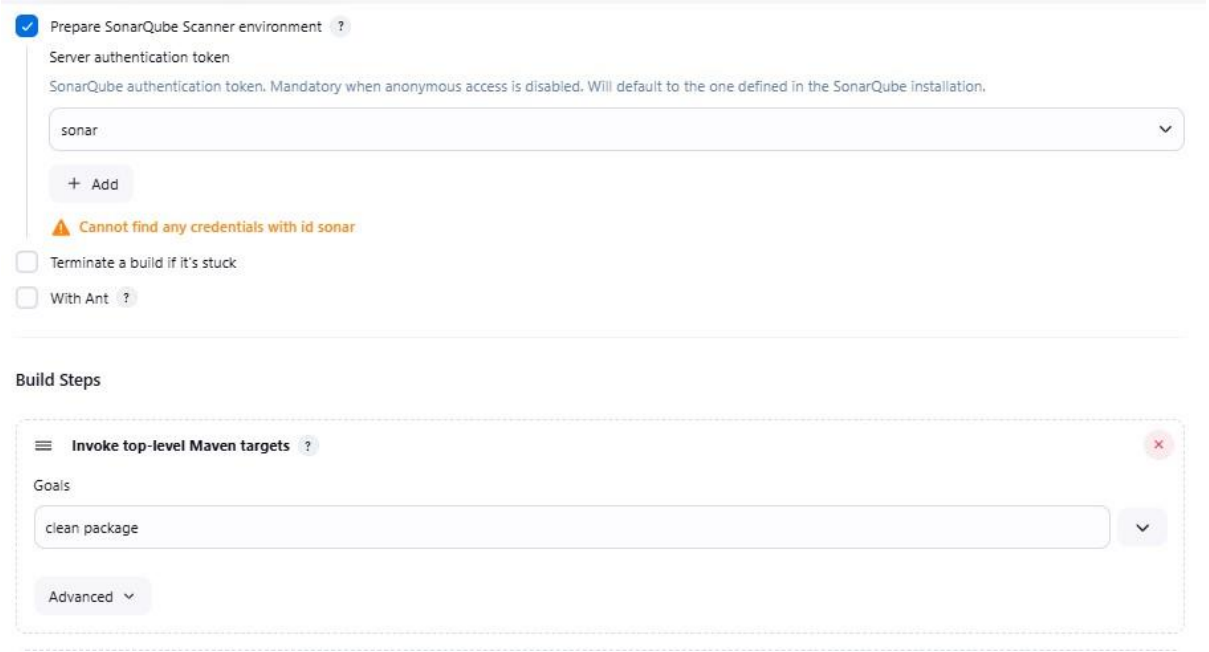
```
pipeline {
    agent any

    stages {
        stage('checkout') {
            steps {
                git 'https://github.com/Abeed2024/code.git'
            }
        }
         stage('build') {
            steps {
                sh 'mvn clean package'
            }
        }
```

## Step 2:

1. Create the server for SonarQube
    - Update the server
    - Install the java 17 version

- Install unzip packages
- Create the sonar user and switch to that created user
- Copy the zip file and paste in terminal to install zip file
- Unzip the zip file by using the file name
- Give the owner permissions and file permissions to the unzip file
- Start the SonarQube scanner to create the server of SonarQube

2. Switch to the SonarQube server and generate the token and copy the generated token
3. Install the plugins of SonarQube in the Jenkins server
4. Provide the credentials for SonarQube
5. Add the SonarQube credentials in created free style job as shown in below figure




6. Create the SonarQube with the pipeline syntax
7. Provide the credentials for SonarQube in pipeline and generate the pipe line syntax and copy the syntax and paste in the script as shown in below figure

```
            }
        }
        stage('sonarqube scanner') {
            steps {
                withSonarQubeEnv('sonar') {
    sh 'mvn sonar:sonar'
}
            }
        }
```

## Step 3:

1. Create the server for nexus
    - Update the server
    - Install the java 17 version
    - Create the nexus user and switch to created user
    - Copy the file from nexus page and paste in created user terminal
    - Un tar the tar file and provide the owner permissions to the file and give the executable file permission to that file
    - Start the nexus for the nexus server
2. Switch to nexus server
    - Create the repository in the nexus server for the created free style job
    - Switch to Jenkins server and install the nexus plugins in the Jenkins server
    - After installing the plugins and restart the Jenkins server
    - Go to build steps in the Jenkins server and select the nexus artifacts uploader
    - Provide the credentials in the nexus artifacts uploader in build steps
    - Provide all the pom file related information in the nexus artifacts uploader as shown in below figure


3. Create the nexus artifacts uploader by using the pipeline
4. Generate the pipeline script by using the nexus artifacts uploader
5. Copy the generated script and paste the script in script as shown in below figure

```
        stage('sonarqube scanner') {
            steps {
                withSonarQubeEnv('sonar') {
sh 'mvn sonar:sonar'
}
                }
            }
            stage('nexus uploader') {
                steps {
                    nexusArtifactUploader artifacts: [[artifactId: 'vprofile', classifier: '', file: 'target/vprofile-v2.war', type: 'war']], credentials
                }
            }
```

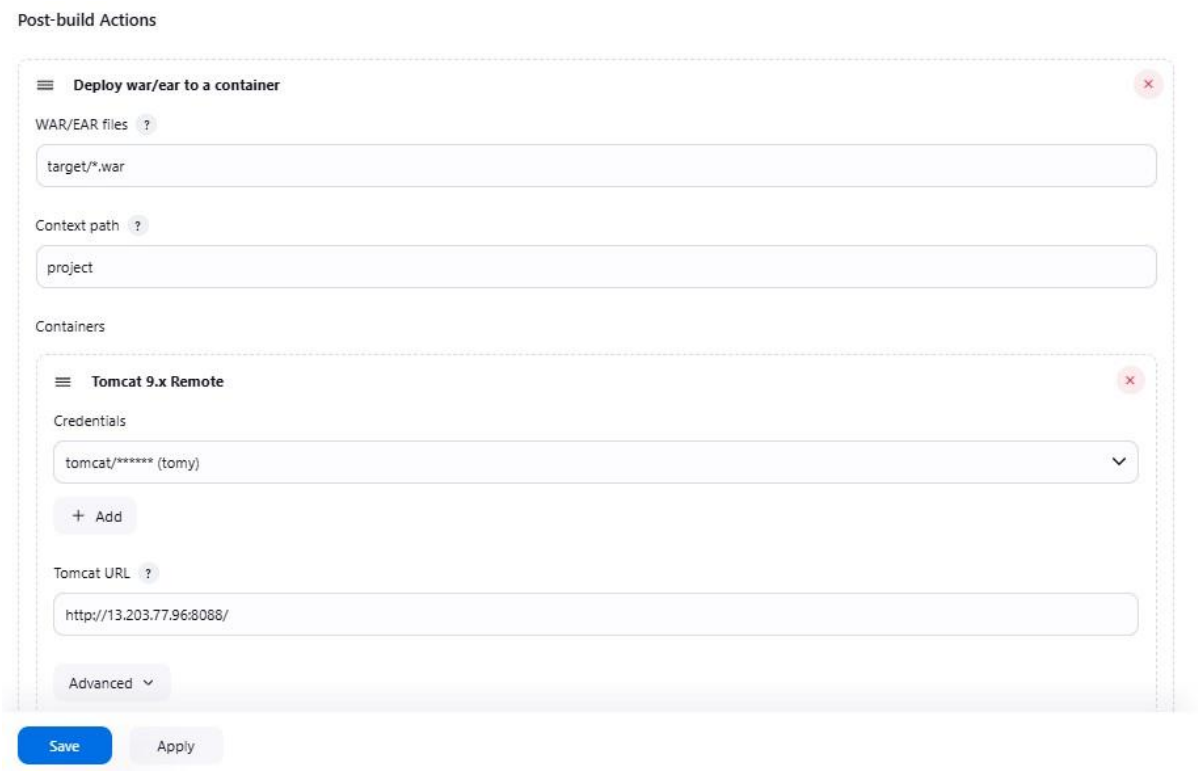
6. If the build is done then the work is done

## Step 4:

1. Create the server for tomcat
    - Update the server
    - Install the java 11 version

- Install the maven
- Copy the tomcat file in the server and paste in the tomcat terminal
- Un tar the tar file and modify the configure directory in the tomcat
- Modify the web apps directory in tomcat server
- Start the tomcat server

2. Give the credentials for tomcat in Jenkins server
3. By using post build option, we can select the deploy to war or ear option and provide the credentials in that as shown in below figure




4. Generate the pipeline syntax by providing the credentials of deploy to war or ear option and copy the syntax and paste the syntax in the script as shown in below figure

```
        }
        stage('nexus uploader') {
            steps {
                nexusArtifactUploader artifacts: [[artifactId: 'vprofile', classifier: '', file: 'target/vprofile-v2.war', type: 'war']], credentialsId: 'nexus', groupI
            }
        }
         stage('deploy') {
            steps {
                deploy adapters: [tomcat9(credentialsId: 'tomy', path: '', url: 'http://13.203.77.96:8088/')], contextPath: 'project-deployment', war: 'target/*.war'
            }
        }
    }
}
```

5. If the build gets success, then the code is deployed that is shown in below figure