Week 4: CI/CD

|  |  |
| --- | --- |
| **Devops tools** | Git, Maven, Jenkins, SonarQube |

|  |  |
| --- | --- |
| Task | Day |
| Task 1 | Day 1(week4) |
| Task 2 | Day 2(week4) |
| Task 3 | Day 3(week4) |
| Task 4 | Day 4(week4) |
| Troubleshooting and find fix for issues if any | Day 5(week4) |

**Prerequisites**:

1. **Before you begin with the task, kindly ensure that you can use your existing GitHub account with GitHub organization managed by Capgemini. Creation of public GitHub repositories on GitHub is not allowed. You should not use them in Capgemini network. In case if you don’t have access kindly raise the request in service central portal. Use the attached doc to get access.**
2. **In the upcoming Task you would be integrating SonarQube with Jenkins. Hence ensure that your project contains the files that SonarQube needs before you push the code to GitHub.**



Task 1.

1. Create a repo with internal access in the GitHub organization for which you have got access.
2. Push the code created in week 3 to the repo created in step 1. You can use IDE or git bash to push the code.
3. Write appropriate commit message when you push the code

Note: GitHub no longer authenticates using password. You need to general personal access token and use it instead of password for authentication.

Task 2:

1. Setup Jenkins. Ensure to check Java version prior installation
2. Check if you are able to see the Jenkins home page.
3. Ensure all recommended plugins are installed.
4. Ensure the GitHub, Maven, SonarQube plugin is install
5. Configure GitHub Enterprise Servers in Jenkins🡪Manage Jenkins🡪Configure system.

Task 3:

1. Set up SonarQube on your machine
2. Ensure to get the default dashboard of sonarqube
3. Ensure to do SonarQube server setup in Jenkins🡪Manage Jenkins🡪Configure system
4. In the Global Tool configuration configure Maven, JDK, SonarScanner

Task 4:

Ensure that you have completed step 1 to step 3

1. Create a freestyle job or pipeline in Jenkins.
2. Configure source code management session.
3. Configure the build step so that it is able to execute maven command clean install
4. Configure another build step to do static code analysis using SonarQube Scanner
5. On successful build the surefire test report should be viewable.
6. The static code analysis report should be viewable on sonar dashboard.