

CI - Continuous integration (time: 1h00')

Next exercise is focused on know the bases of working over Jenkins and SonarQube, to create a system of continuous integration.

Before start

Please read carefully, go step by step. Verify if you have installed the required software for this exercise.

- GIT (www.git.org)
- SonarQube (<http://www.sonarqube.org/downloads/>)
- Jenkins (<https://jenkins-ci.org/>)
- JRE-JVM (Java Virtual Machine)

If not, download them. Install Java Runtime Environment, and GIT client (these two steps aren't covered at following step by step configuration)

Configuring CI system

1. Init SonarQube
 - Create a folder "sonar" and uncompress the package content on it
 - Open terminal, seach the folder sonar and navigate to bin
 - Execute `./sonar.sh start` or `startSonar.cmd` (if windows)
 - Check if it's all working: open browser and go to <http://localhost:9000>
2. Init Jenkins
 - Get the installer, and execute it.
 - Follow the steps to install, when the installation automatically starts <http://localhost:8080/>

Now you're ready to config your CI environment:

3. Add some nested plugins on jenkins
 - On Jenkins. Go to manage jenkins > Manage plugins
 - On tab "Available" search for "SonarQube"
 - Check "SonarQube"
 - Search for GIT client plugin, and check for install
 - And search for "repository" and check for install the plugin "Repository Connector"
 - Install and restart jenkins.
4. Configure SonarQube to run on jenkins
 - Go to Manage Jenkins > System Configuration
 - On "SonarQube Server" Name it, and leave rest as default
 - On "SonarQube Scanner", check install automatically, create new installation, select "Extract from *.zip/*.targz", and set the url: <https://sonarsource.bintray.com/Distribution/sonar-scanner-cli/sonar-scanner-2.6.1.zip> (to install sonar-runner (if you have an installation of sonar-runner locally, you can chose path).

5. Create a new GIT repository on Jenkins Home *(needs explanation from the trainer)*
 - Go to local jenkins installation and create a new folder.
 - Navigate to this folder, and init a git repo
 - Put some code and tests, and commit them

Notes:

- <http://www.cimgf.com/2015/05/26/setting-up-jenkins-ci-on-a-mac-2/>
- <http://blog.manula.org/2013/03/running-jenkins-under-different-user-in.html#!>

6. Create a new job
 - Go to jenkins home, and create new job.
 - Set a name for this job: "CI-Example"
 - Under "Source Code Management", select GIT and configure de the path to your repository url using file://<route>. Then select the branch to use for build
 - On Build Triggers, select Poll SCM *(needs explanation from trainer)*
 - at schedule text box set: H/5 * * * * *
 - At Build Actions select "Execute SonarQube Scanner"
 - At Post-build Actions, select publish junit tests.

*Note: besides using poll scm, use hooks to notify when changes has been done on your repo...
"curl http://yourserver/jenkins/git/notifyCommit?url=<URL of the Git repository>"*

HELPER:

Run Jenkins as different user:

- `sudo vim /Library/LaunchDaemons/org.jenkins-ci.plist>`
- `sudo launchctl unload -w /Library/LaunchDaemons/org.jenkins-ci.plist`
- `sudo launchctl load -w /Library/LaunchDaemons/org.jenkins-ci.plist`