Jenkins Build 2

Objectives:

- Configure Unit tests with Jenkins
- Configure a second build node
  - This requires a second vm (we are going to use linux)
- Configure email-based notifications
- Integrate Jenkins and SonarQube

Requirements:

Continue where you left off on your last project with Jenkins. You will start in chapter 2 at the ‘Configuring JUnit’ section, and work to the end of the chapter.

Plugins

When configuring the dashboard view plugin, after configuring it, you must attach it to a build job. Attach it to your PetClinic build job you created in the last assignment. I don’t care what portlets you add.

Nodes

Configure a second VM as a build node. Though the book uses a windows machine, we will use a Linux vm as our second node. Make sure to install default-jre, default-jdk, git.

On that machine:

- Add a new user called jenkins
- As that new user, in their home directory, create a directory called jenkins_slave.
- From the jenkins web interface, as you are adding your new node, use the following configuration:
  - remote root directory /home/jenkins/jenkins_slave
  - Launch slave agent via SSH
  - Configure the host ip of second machine
  - Add new credentials (will ask you for name and password of jenkins user)
  - Use non-verifying host strategy
- Verify the log status of the agent that it is online
  - I ran my build project on this node at this point. You only need to edit your project settings and select where the project can run.
- Make sure that you can build your project on BOTH the master node and your new slave node.

Very easy email settings

Make sure you are in Email notification section NOT Extended email notification.

- SMTP server sol.cs.dixie.edu
- port 25
- put reply-to address as whatever you want
- test send an email
- make sure you get it

SonarQube

- I installed on same node as jenkins.
- Default username/password = admin/admin
- The real command to run it is ./sonar.sh start
  - This has to be re-run if you reboot the node
- One thing that the text doesn’t make clear is that you MUST add a build step to your project to have the SonarQube scanner run.
I also found that my build failed with SonarQube, unless I added the following line to sonar-project.properties:

```
sonar.java.binaries=src
```

**Check off procedure:**

For this project, you should submit the following screenshots:

- JUnit tests are showing up
- That you have added some dashboard plugins
- That you can successfully execute a build job on a second node
- That you can receive an email notification (the test email notification is fine)
- That sonarqube has received some information from your build job.