Jenkins

Lab Book

Document Revision History

Date	Revision No.	Author	Summary of Changes
24/05/2013	0.1	Rathnajothi Perumalsamy	First version

Table of Contents

Getting Started

Overview

This lab book is a guided tour for learning Jenkins. It comprises examples and 'To Do' assignments. Follow the steps provided in the examples and work out the 'To Do' assignments given.

Setup Checklist for Jenkins

Here is what is expected on your machine in order for the lab to work.

Minimum System Requirements

- Intel Pentium 90 or higher (P166 recommended)
- Microsoft Windows 95, 98, or NT 4.0, 2k, XP.
- Memory: 32MB of RAM (64MB or more recommended)
- Internet Explorer 6.0 or higher
- MS-Access/Connectivity to Oracle database
- Apache Tomcat Version 5.0.

Please ensure that the following is done:

- Eclipse 3.5 or above with SVN or CVS plugin is installed.
- JDK 1.5 or above is installed. (This path is henceforth referred as <java_install_dir>)
- ANT OR MAVEN is installed
- Connected to SVN or CVS

Instructions

- Create a directory by your name in drive <drive>. In this directory, create a subdirectory Jenkins_assgn. For each lab exercise create a directory as lab <lab number>.

Learning More (Bibliography if applicable)

- Jenkins Continuous Integration Cookbook by Alan Mark Berg
- Jenkins – The Definitive Guide by John Ferguson Smart

Lab 1. Installing and running Jenkins

Goals	• Learn to install and start Jenkins
Time	30 minutes

1.1: Installing Jenkins

1.1.1: As a standalone application

Step 1: Download jenkins.war file from http://jenkins-ci.org and save it into your local disk.

Step 2: Open Command Prompt and change working directory where jenkins.war file exists.

Step 3: Install Jenkins by executing the command as highlighted in the below screen.

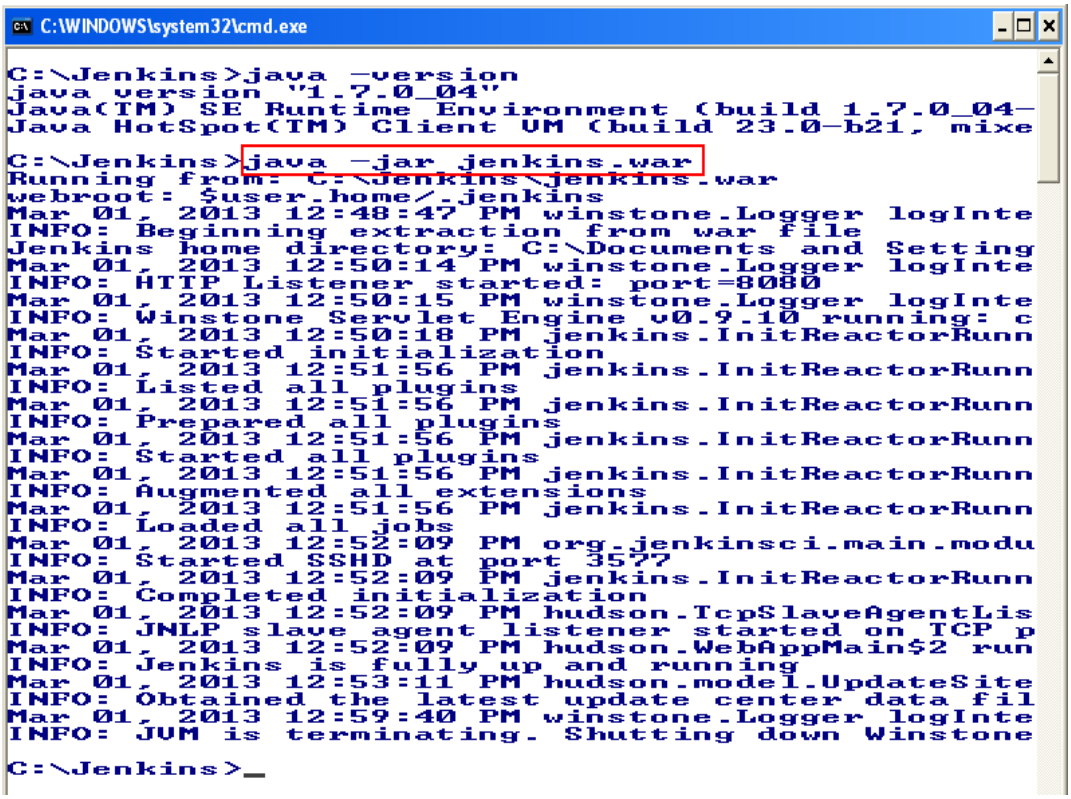

Figure 1: Jenkins Installation

Step 4: Once Jenkins is started, the Jenkins dash board can be accessed by giving the following link in the browser

http://localhost:8080/

1.1.2: As a windows service

Step 1: Start Jenkins as a standalone application.

Step 2: Open Jenkins Dashboard by requesting the URL http://localhost:8080

Step 3: Click “Manage Jenkins” link.

Step 4: Click “Install as Windows Service” button.

Step 5: Select the installation directory of Jenkins and click “Install”.

Step 6: Once Jenkins is successfully installed as a windows service, Jenkins dashboard will always be accessible by requesting the URL http://localhost:8080

1.2: <<TODO>

Install and run Jenkins as a windows Service

Lab 2. Jenkins Configuration

Goals	At the end of this lab session, you will be able to understand: • How to configure Jenkins
Time	30 minutes

2.1: Configuring Jenkins

Step 1: Open Jenkins Dashboard by requesting the URL http://localhost:8080

Step 2: Click "Manage Jenkins" link.

Step 3: Click on "Configure System" Link and fill the field values as shown below:

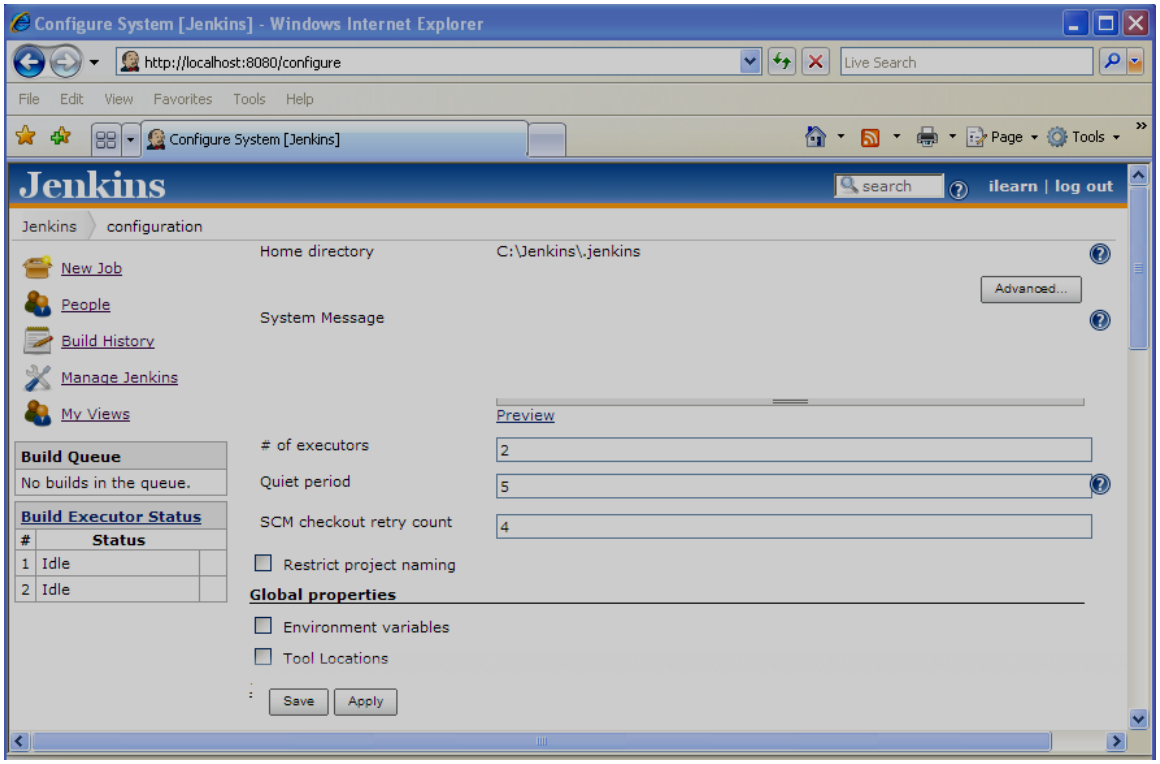

Figure 2: Configure System

2.1.1: Configuring JDK

Step 1: Type name of the JDK. For example, JENKINS_JDK

Step 2: Specify the JDK installation directory

JDK installation can be automated by following the below steps

1. Check “Install Automatically” option in JDK section
2. Select the option to download required JDK version and install.

2.1.2: Configuring ANT

Step 1: Type name of the ANT installation in Name field under ANT section. For example, JENKINS_ANT

Step 2: Specify the ANT installation directory.

ANT installation can be automated by following the below steps

1. Check “Install Automatically” option in ANT section
2. Select the option to download required ANT version and install.

More than one instance of JDK and ANT can be specified.

2.1.3: Email Configuration in Jenkins

To send a feedback to the developers during build failure, E-mail notification configuration is mandatory.

Step 1: Specify details such as SMTP server name, sender credentials, SMTP port in E-mail notification section as shown below:

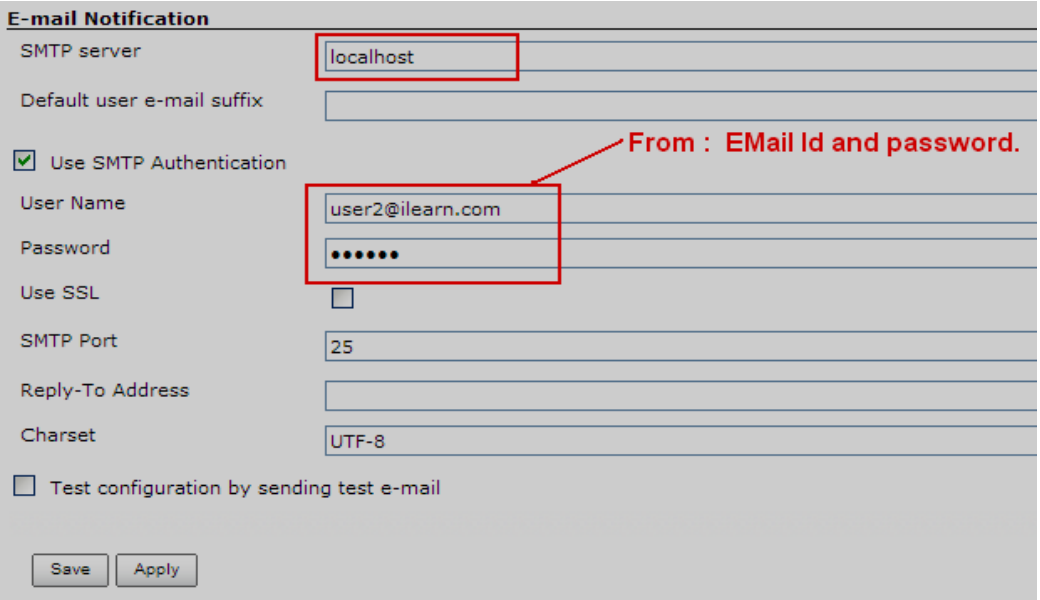

Figure 3: E-mail Configuration

Step 2: Test Email configuration by checking “Test Configuration by sending test e-mail” field and type “Recipient Address” as shown below:

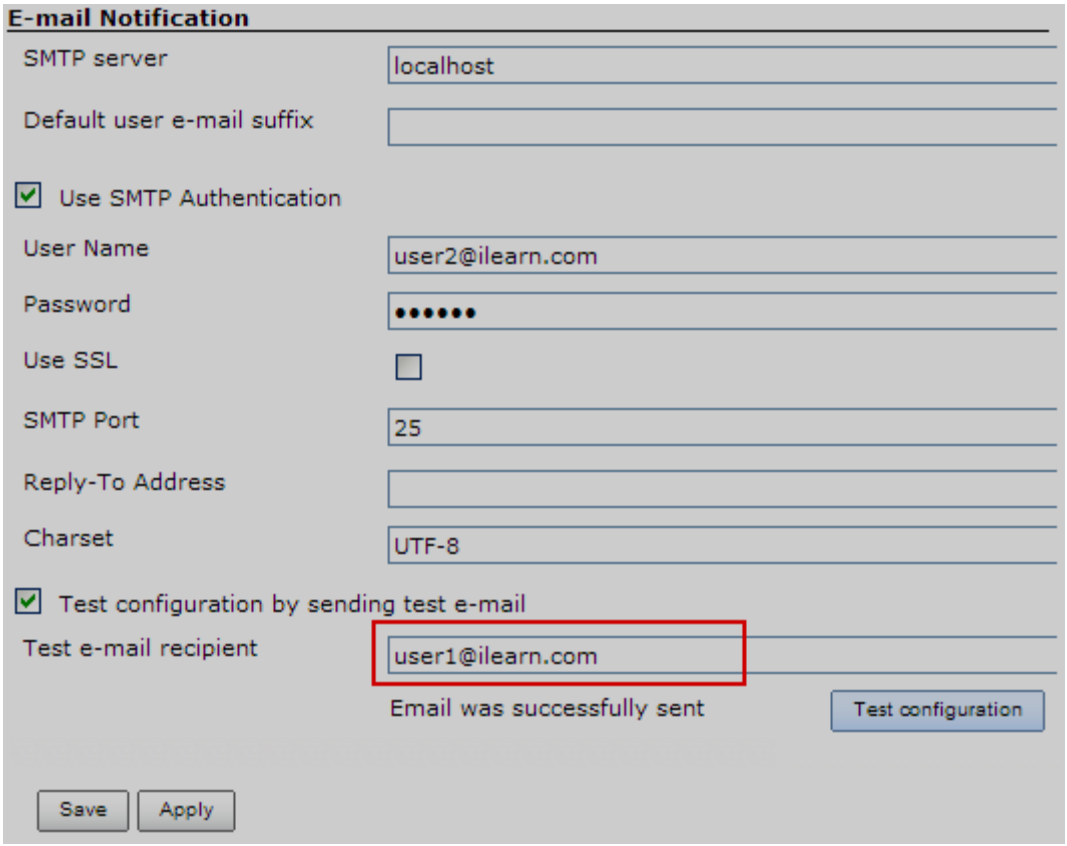

Figure 4: Testing E-mail configuration

Step 3: Test Email will be sent successfully, if the configuration is done correctly.

Lab 3. Build Job creation in Jenkins

Goals	At the end of this lab session, you will be able to understand: • How to create build job and configure build job • Scheduling build job • Execution of project in jenkins
Time	120 minutes

3.1: Creating build job in Jenkins

Step 1: Open Jenkins Dashboard by requesting the URL http://localhost:8080

Step 2: Click "New Job" link.

Step 3: Type job name and select type of project as shown below:

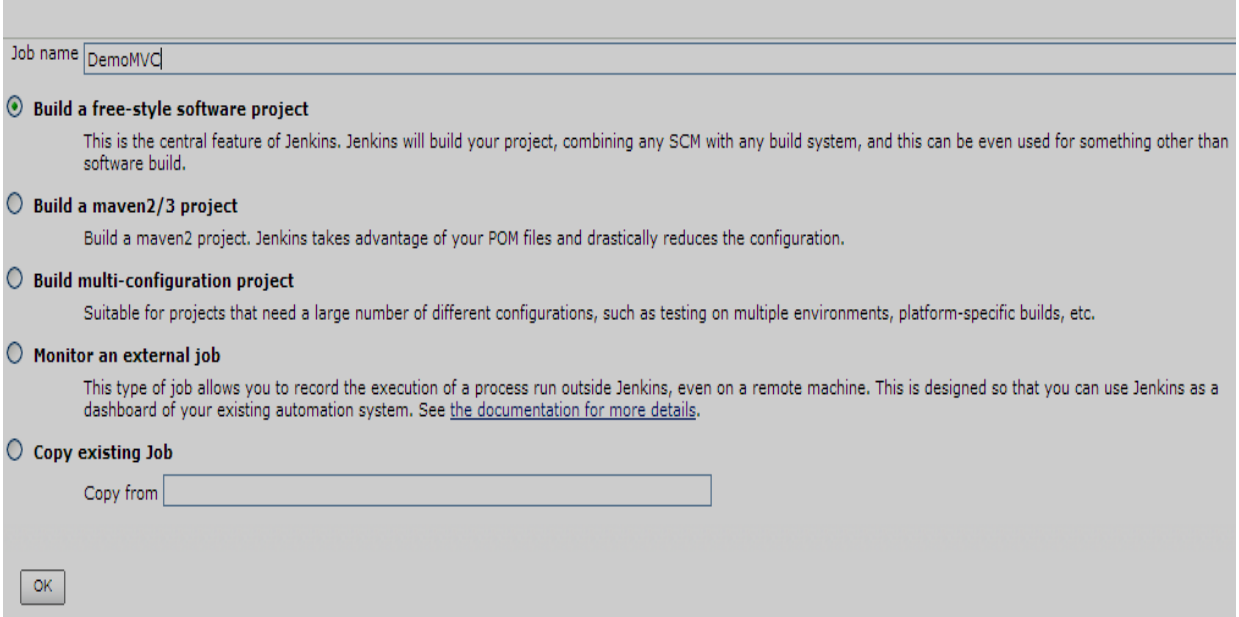

Figure 5: Build job Creation

Step 4: Click ok

3.2: Configure build job in Jenkins

Step 1: Type the description of the project in the "Project Configuration" page (Which will be displayed once build job is created at first time otherwise manually this page should be opened) as shown below:

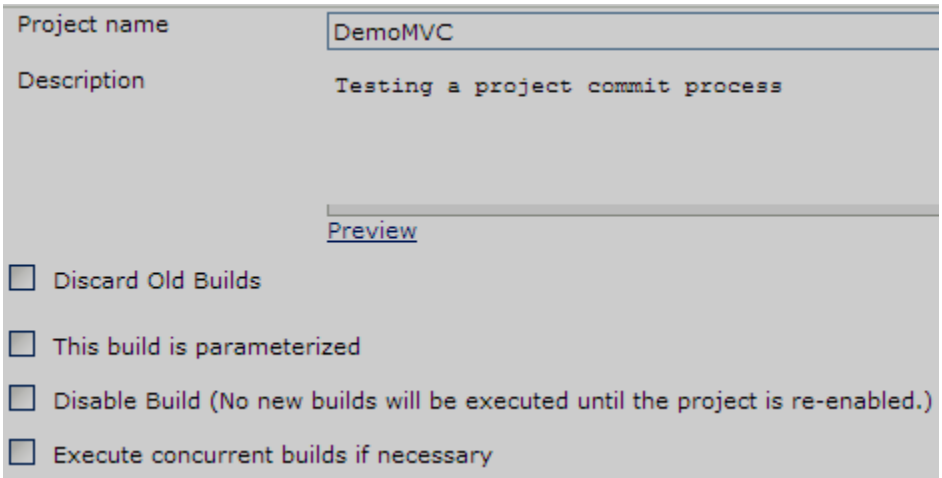

Figure 6: Project Configuration

3.3: Configure SCM

Step 1: Open "Project Configuration" page

Step 2: Select "Subversion" in SCM section

Step 3: Specify Repository URL

Step 4: Type "." in Local module directory field to avoid creation of new directory for source in Local workspace as shown below:

Step 5: Select Check-out strategy as "Use 'svn update' as much as possible, with 'svn revert' before update" (which will systematically run svn revert before running svn update)

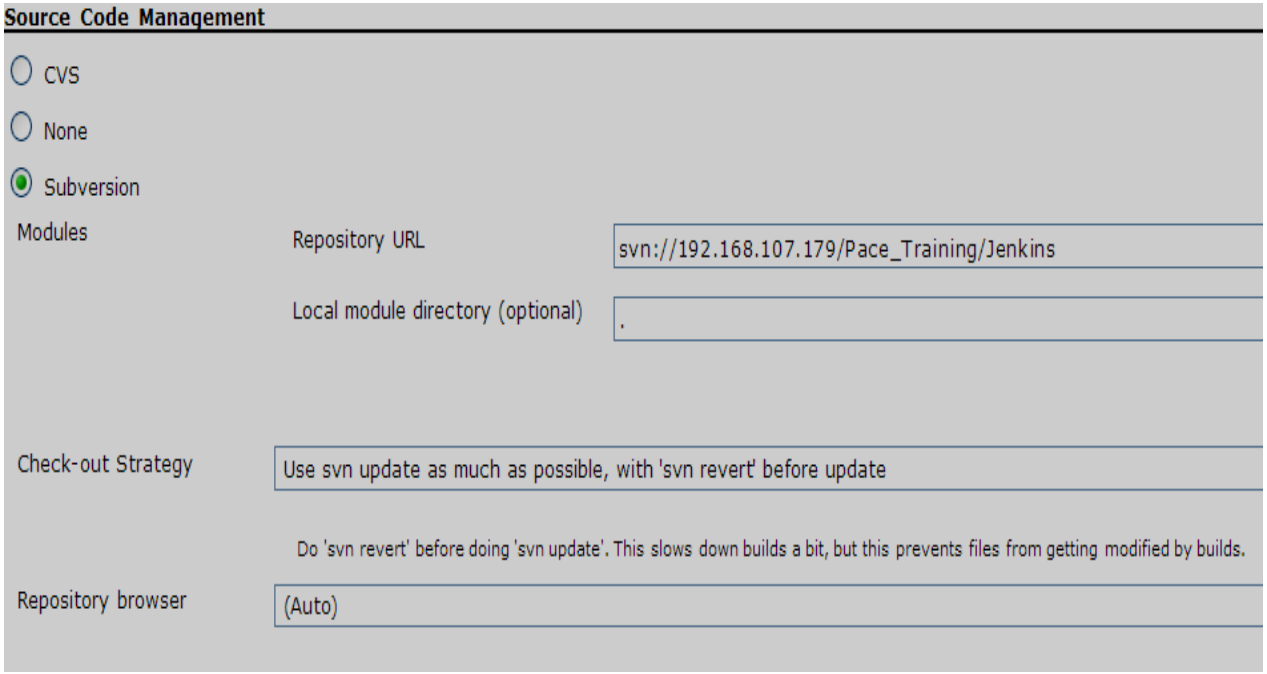

Figure 7: SCM Configuration

3.3: Scheduling job execution

Step 1: Open “Project Configuration” page

Step 2: Select “Poll SCM” in Build Triggers section and type * * * * * in the schedule as shown below: (which polls SCM for every change commits and triggers build execution)

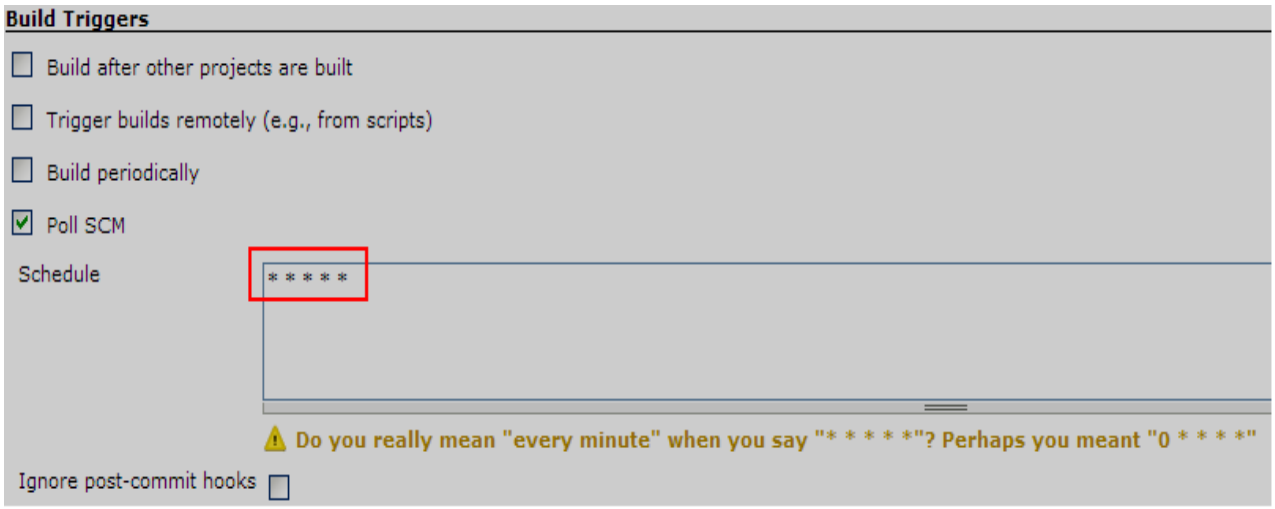

Figure 8: Build scheduling

To build the job periodically, check Build periodically field and type relevant expression in schedule field. For an Example, to build job every day at 8 hours, the expression should be “0 **8 * * *”**

Cron expression should be created to schedule the job trigger.

* * * * *

1st Field - MINUTES Minutes in one hour (0-59)
2nd Field - HOURS Hours in one day (0-23)
3rd Field - DAYMONTH Day in a month (1-31)
4th Field - MONTH Month in a year (1-12)
5th Field - DAYWEEK Day of the week (0-7) where 0 and 7 are Sunday

To schedule job every minute, use cron expression * * * * *

3.4: Invoke Ant

Step 1: Once SCM is configured, choose “Invoke Ant option” from “Add build step” drop down list in build section as shown below:

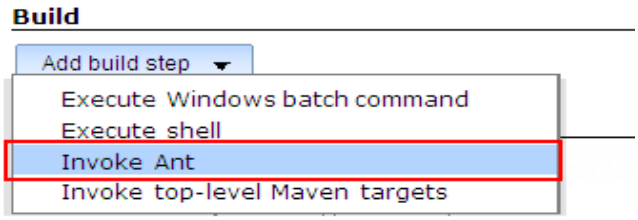

Figure 9: Invoking Ant

Step 2: Specify the Ant version and type target name (which need to be executed from your build targets) in Targets field as shown below:

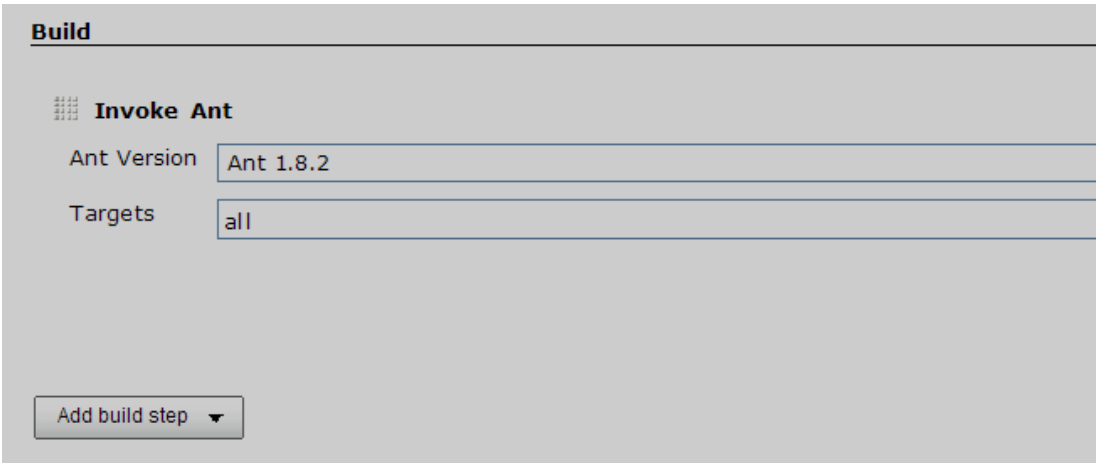

Figure 10: Selecting target in build

Step 3: Click on save

Step 4: Schedule the build to be executed immediately and view the console output as shown below:

```
Started by user anonymous
Building in workspace C:\Jenkins\.jenkins\jobs\DemoMVC\workspace
Reverting C:\Jenkins\.jenkins\jobs\DemoMVC\workspace\.
Updating svn://192.168.107.179/Pace_Training/Jenkins
At revision 1556
[workspace] $ cmd.exe /C ""C:\Material-iGATE Format\Developer Workbench\Demos-ANT\apache-ant-1.8.2-bin\apa
Buildfile: C:\Jenkins\.jenkins\jobs\DemoMVC\workspace\build.xml

init:

compile:
    [javac] C:\Jenkins\.jenkins\jobs\DemoMVC\workspace\build.xml:16: warning: 'includeantruntime' was not s

war:

BUILD SUCCESSFUL
Total time: 0 seconds
Finished: SUCCESS
```

Figure 11: Build execution output

<<TODO>>

Create a new project in Jenkins for an application, which should build the project automatically after every 24 hours.

Lab 4. Automated deployment and continuous delivery

Goals	At the end of this lab session, you will be able to: • Automate the deployment of an application to a tomcat server
Time	60 minutes

4.1: Installation of Deploy plug-in

Step 1: Open Jenkins Dashboard by requesting the URL http://localhost:8080

Step 2: Click "Manage Jenkins" link.

Step 3: Click on "Manage Plugins" Link

Step 4: Select "Available" tabs, check "Deploy to container Plugin" and click on Download and install button.

Alternate Approach:

Step 1: Download deploy.hpi file

Step 2: Follow steps 1 to 3 given in Installation of deploy plug-in

Step 2: Install plug-in manually by selecting "Advanced tab" and upload plug-in as shown below:

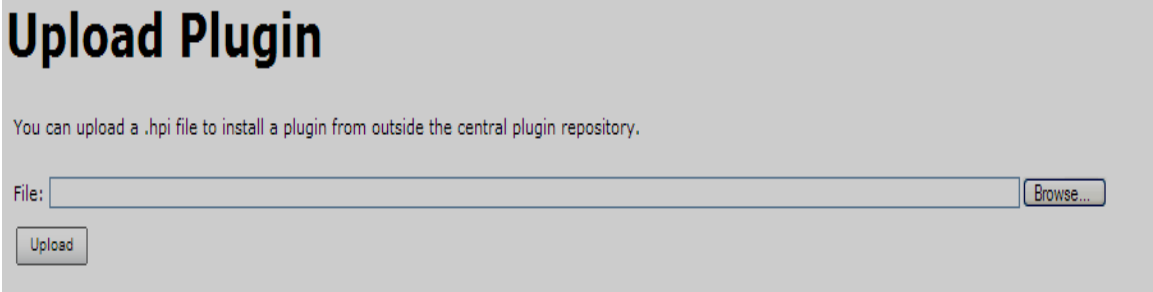

Figure 12: Install plug-in manually

4.2: Automate deployment of an application

Step 1: Open Project (build job) configuration.

Step 2: In Post-build Actions, choose "Deploy war/ear to a container" from "Add post-build action" drop down list

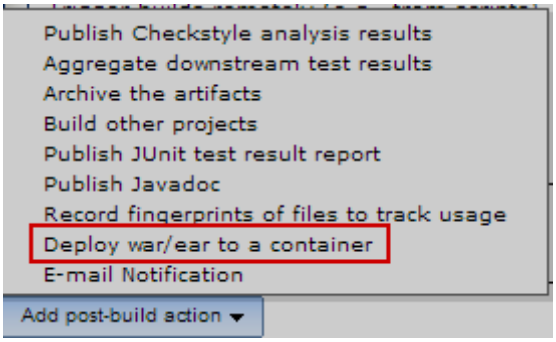

Figure 13: Deploying war file to tomcat server

Step 3: Configure Tomcat server details as shown below:

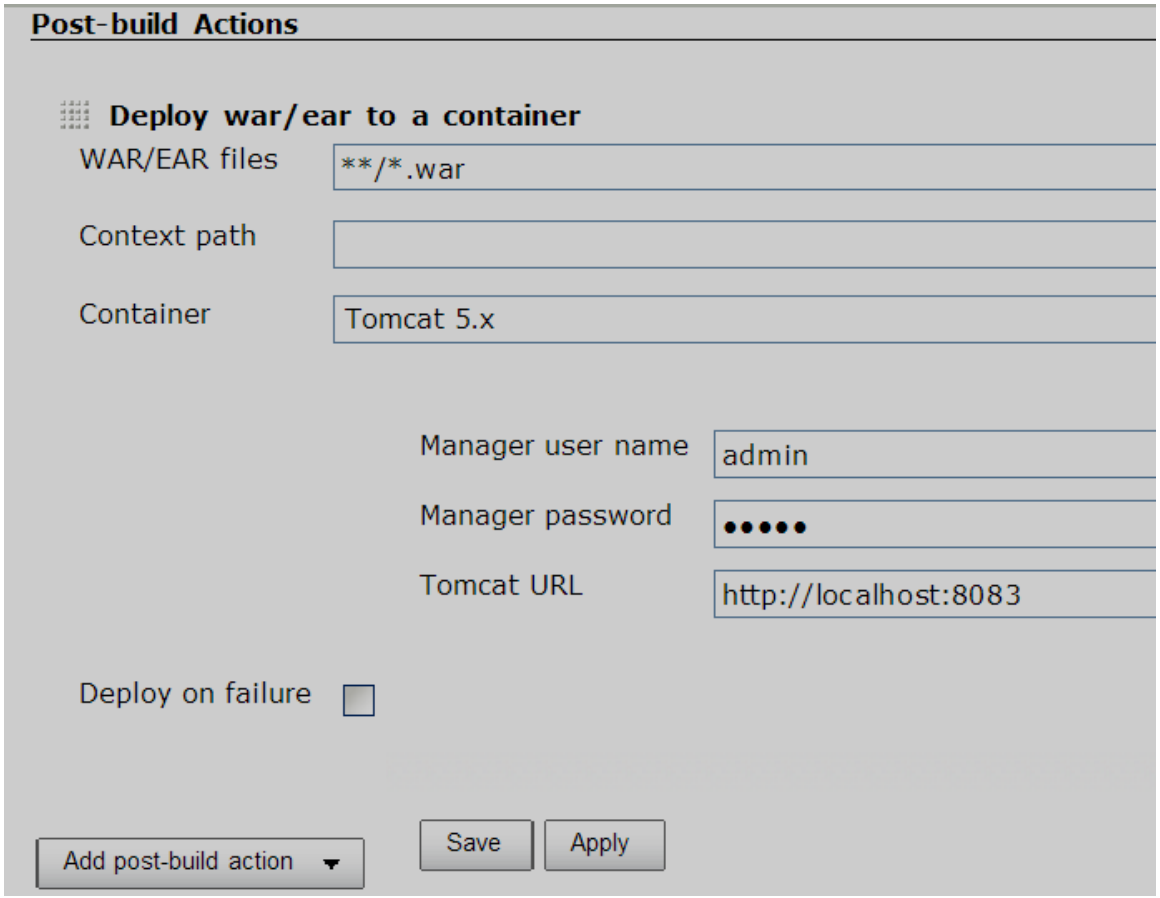

Figure 14: Deployment server configuration details

Step 4: Check Deploy on failure field, in order to send feedback to the developers once build/deployment is failed.

Step 5: choose “E-mail Notification” from “Add post-build action” drop down list as shown below:

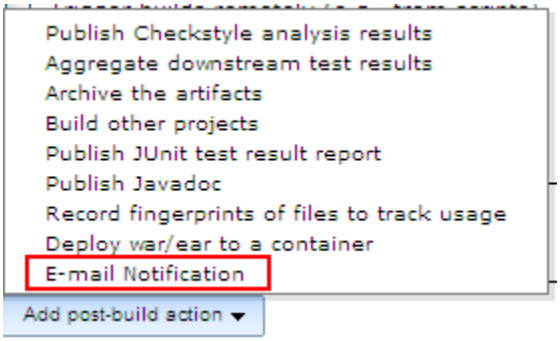

Figure 15: E-mail notification

Step 6: Type recipient mail id and check “Send e-mail for every unstable build.

Step 7: Click on save and schedule the job execution immediately.

<<TODO>>

Create a new project in Jenkins for an application, which should automate the deployment of project into tomcat server.

Lab 5. Automating Testing using Jenkins

Goals	At the end of this lab session, you will be able to: • Automate the unit testing of an application
Time	60 minutes

5.1: Automate unit testing of an application

Step 1: In application build.xml file, ensure target is included to invoke unit test class execution.

Step 2: When build file is executed, automatically test class execution is also invoked.

Step 2: Open Project (build job) configuration.

Step 3: In Post-build Actions, choose "Publish JUnit test result report" from "Add post-build action" drop down list to display test results.

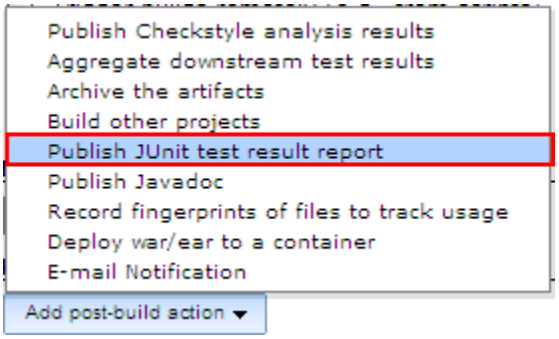

Figure 16: JUnit test result report

Step 4: To generate and publish JUnit test result report, configure report generation as shown below:

Figure 17: Publishing JUnit test report

Step 5: Once test is build, "Test result Trend" will be displayed in the Project home page as shown below:

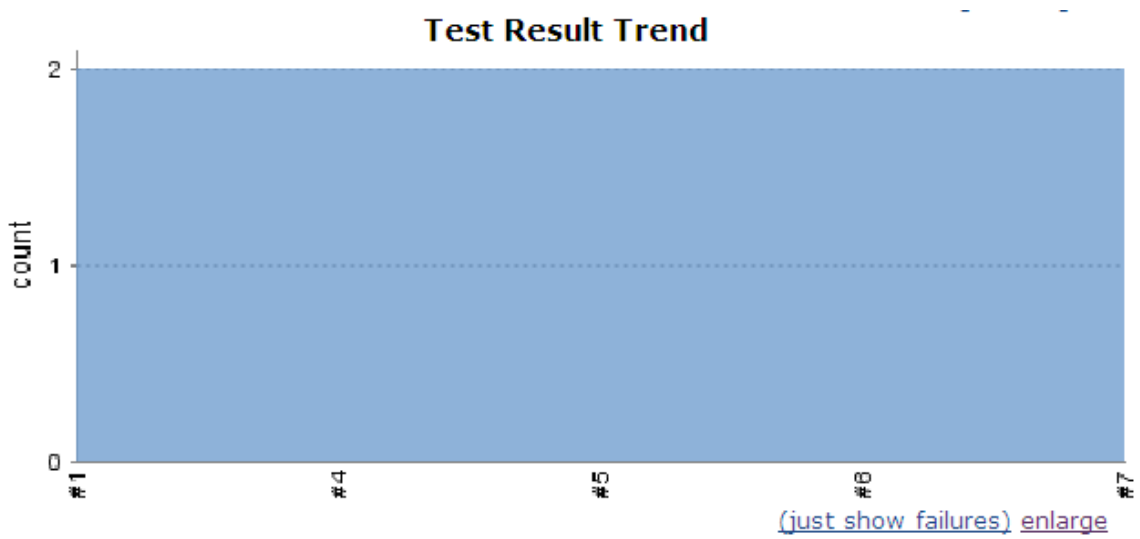

Figure 18: Test Result Trend

Alternate Approach:

Step 5: Once test is build, Click on "Latest Test Result" link in the Project home page as shown below. Detailed test result will be displayed with duration, number of tests failed, etc...

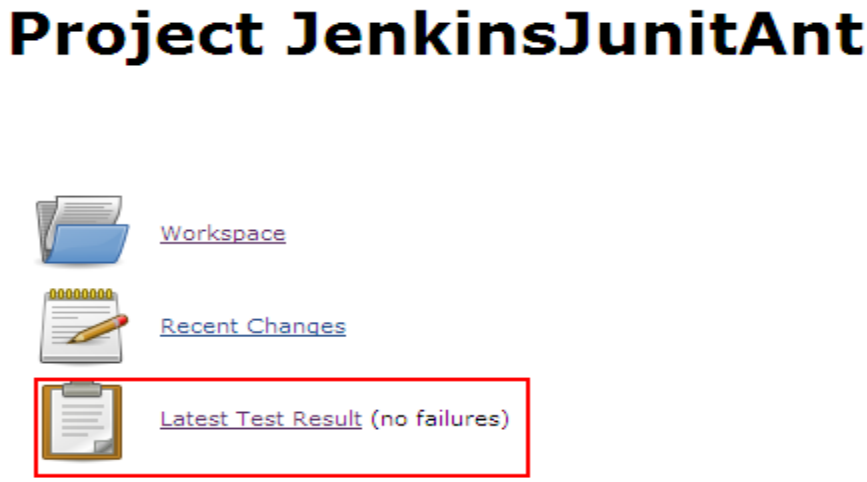

Figure 19: Test Result

<<TODO>>

Create a new project in Jenkins for an application, which should automate unit testing of an application.

Lab 6. Securing Jenkins

Goals	At the end of this lab session, you will be able to: • Enable more security in Jenkins
Time	60 minutes

6.1: Activating Securing in Jenkins

Step 1: Click “Configure Global Security” link in “Manage Jenkins” screen.

Step 2: Check “Enable Security” Field

Step 3: Click on “Jenkins own user database” and select “Allow users to sign up” field under Access Control (To authenticate users based on existing credentials in Jenkins own database and to support new users sign up) as shown below:

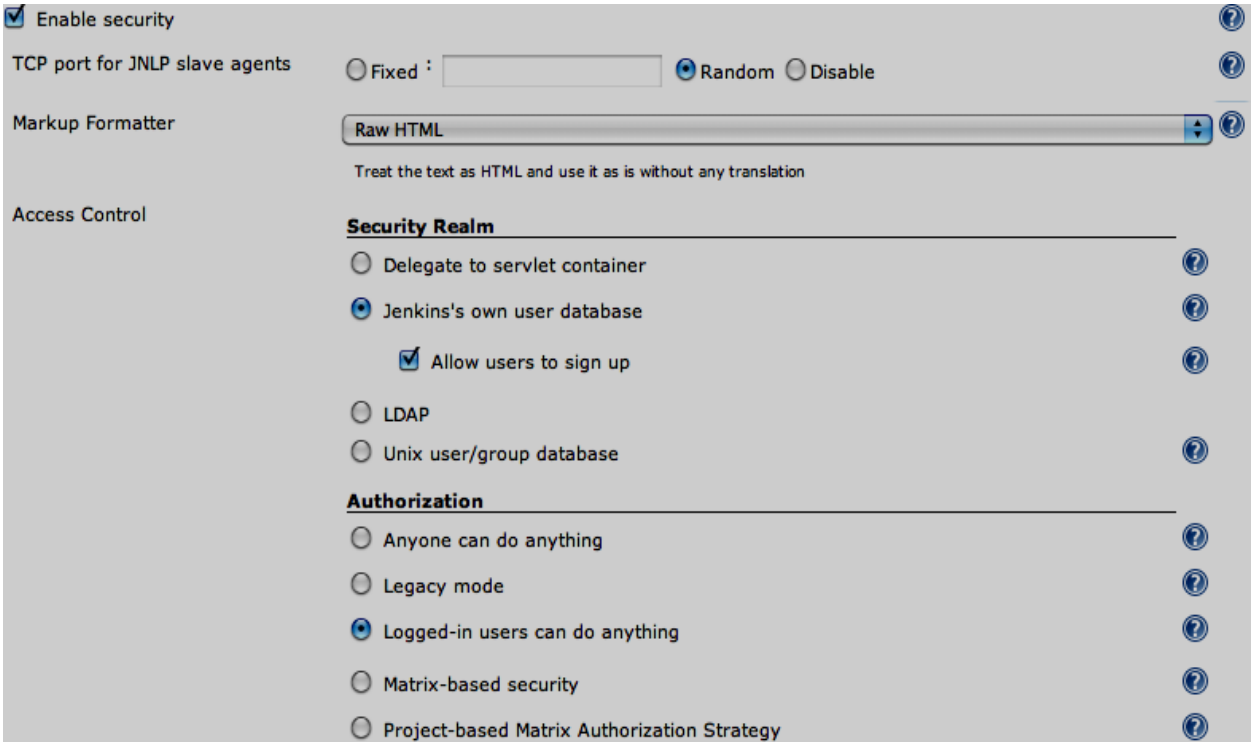

Figure 20: Activating security in Jenkins

6.2: Creating new users in Jenkins

Step 1: Open Jenkins Dashboard

Step 2: Click “sign up” link to create new user.

Step 3: Fill the user’s details and click on Sign up.

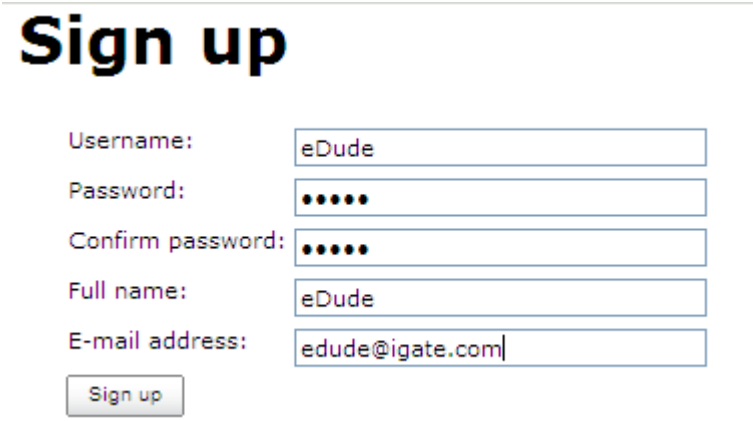

Figure 21: Creating new User

Step 4: View list of created users details by clicking on “Manage Users” link in “Manage Jenkins” screen.

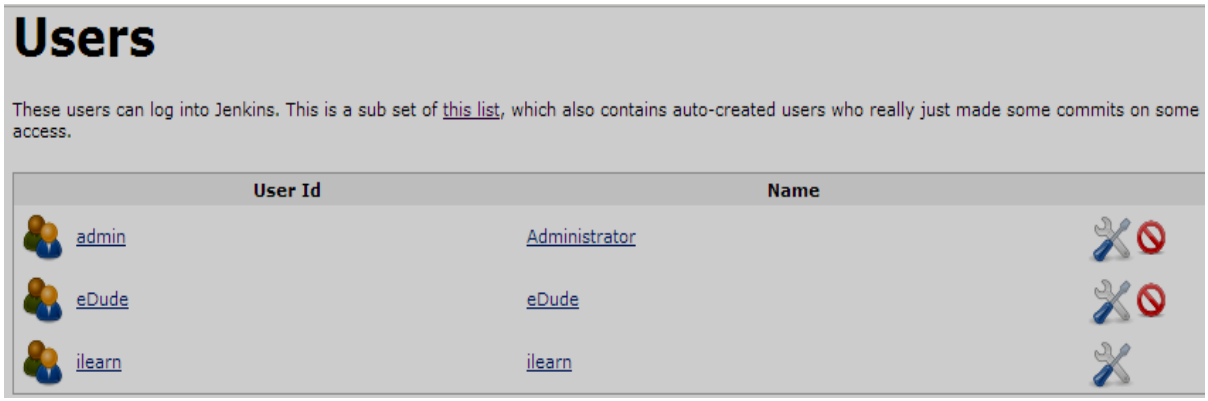

Figure 22: List of existing users

6.3: Authorizing users in Jenkins

Step 1: Click “Configure Global Security” link in “Manage Jenkins” screen.

Step 2: Click “Matrix-based security” under Authorization to authorize the users based on username and assign permissions as shown below.

(As default, Logged-in users can do anything is selected under Authorization).

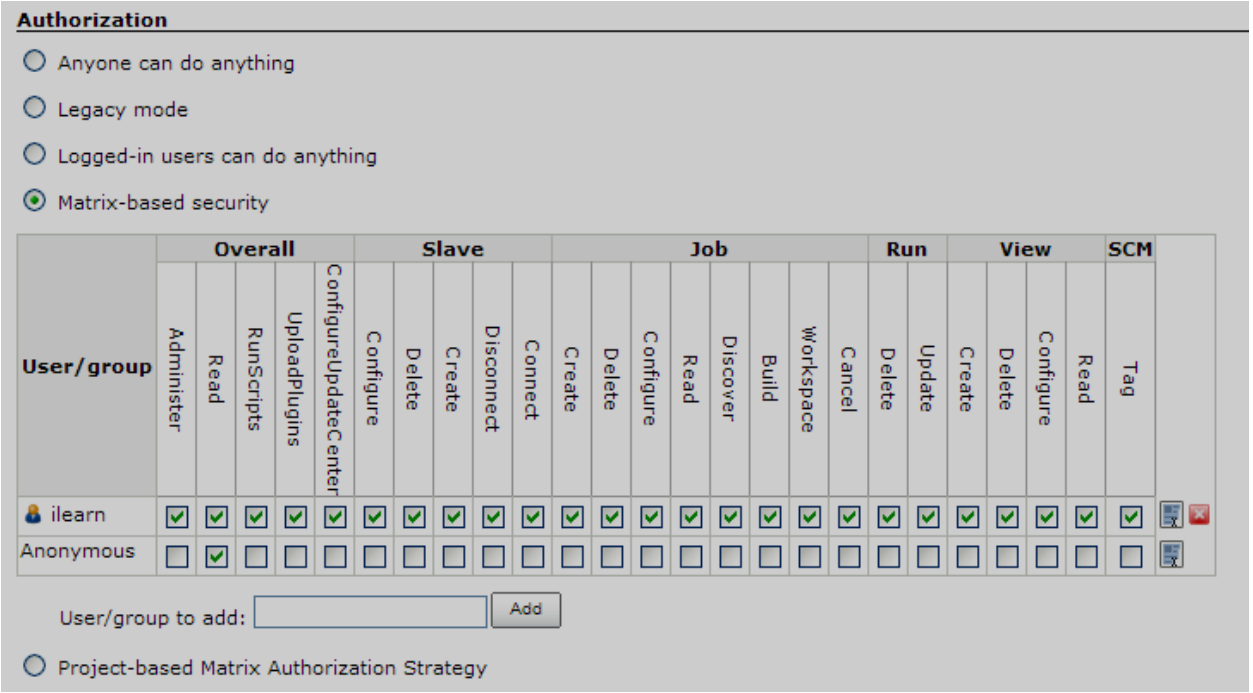

Figure 23: Authorizing Users

Step 3: Click “Project-based Matrix Authorization Strategy” under Authorization to authorize the users based on projects.

<<TODO>>

1. Create a new user in Jenkins. Authenticate and authorize new user to schedule a specific job execution. Other users should be allowed to view the job.
2. Implement authorization based on project-matrix.

Lab 7. Code Quality

Goals	At the end of this lab session, you will be able to: • Ensure quality of the code.
Time	60 minutes

7.1: Ensuring code quality in Jenkins

Step 1: Install Checkstyle, PMD, and FindBugs plugins.

Step 2: Open Project Configuration screen.

Step 3: In Post-build Actions, choose "Publish Checkstyle analysis report" from "Add post-build action" drop down list as shown below.

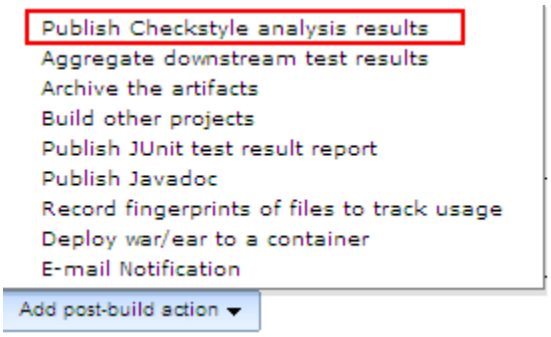

Figure 24: Checkstyle configuration

Step 3: To generate and publish Checkstyle analysis result, configure report generation as shown below:

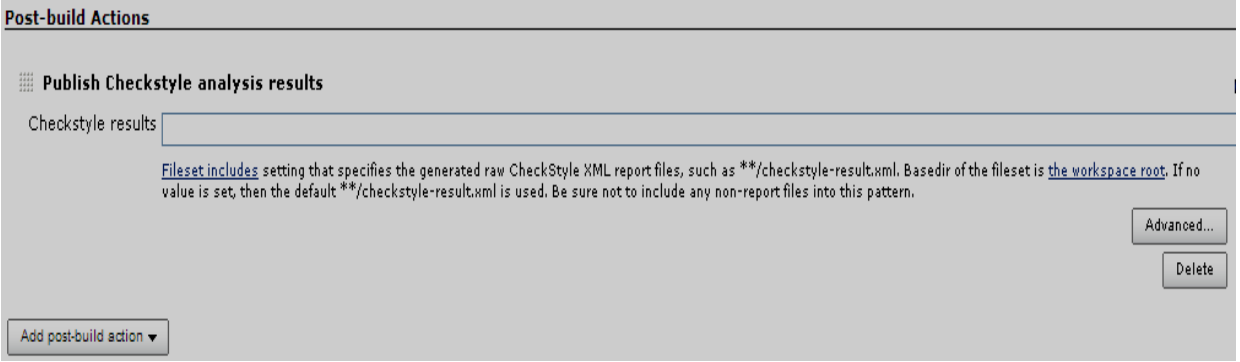

Figure 25: Checkstyle result configuration

(As default, **/Checkstyle-result.xml used as value in Checkstyle results field.)

Step 4: Once job is build, "Checkstyle Trend" will be displayed in the Project home page

Alternate Approach:

Step 4: Once job is executed, the following output displays in build home page.

Figure 26: Checkstyle result

<<TODO>>

1. Check the quality of the code by configuring checkstyle in existing job.

Appendices

Appendix A: Table of Figures