

JENKINS AND CONTINUOUS INTEGRATION

A report by,
Aayush Pathak
Danish K. Chaus
Interns at TCS, Powai

Submitted on: 4/7/2016
Submitted to: Mrs. Vrushali

TABLE OF CONTENTS:

1. INTRODUCTION

- 1.1 Jenkins
- 1.2 Continuous Integration
- 1.3 Why Continuous Integration

2. INSTALLING JENKINS

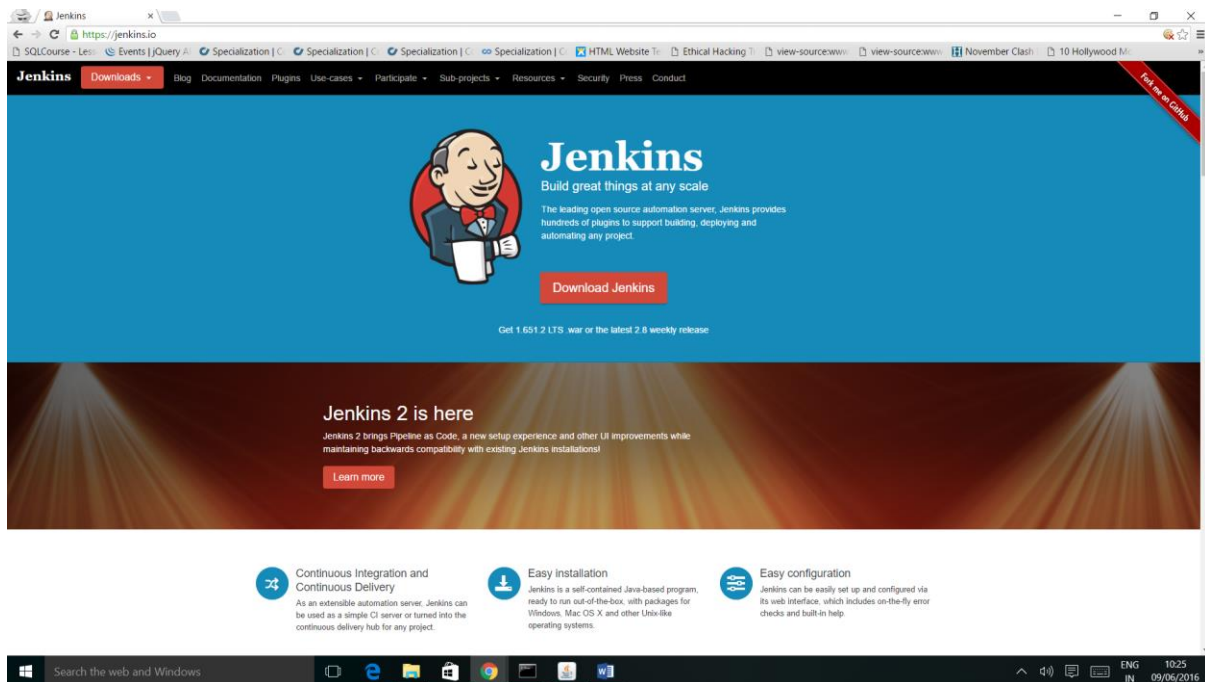
- 2.1 Requirements
- 2.2 Windows Installation
- 2.3 Linux Installation

3. JENKINS MODULES

- 3.1 New Item
- 3.2 People
- 3.3 Build History
- 3.4 Manage Jenkins
 - 3.4.1 Configure Jenkins
 - 3.4.2 Manage Plugins
 - 3.4.3 Creating New User

4. JENKINS PIPELINE

JENKINS AND CONTINUOUS INTEGRATION



1. INTRODUCTION:

1.1 JENKINS:

Jenkins is an open-source continuous integration software tool written in the Java programming language for testing and reporting on isolated changes in a larger code base in real time. The software enables developers to find and solve defects in a code base rapidly and to automate testing of their builds.

1.2 CONTINUOUS INTEGRATION:

Continuous Integration is a development practice that requires developers to integrate code into a shared repository at regular intervals. Continuous integration requires the developers to have frequent builds. The common practice is that whenever a code commit occurs, a build should be triggered.

1.3 WHY CONTINUOUS INTEGRATION:

- **Run your tests in the real world.**
- **Increase your code coverage.**
- **Deploy your code to production.**
- **Build stuff now.**
- **Build stuff faster.**
- **Don't break stuff.**
- **Decrease code review time.**

2. INSTALLING JENKINS:

2.1 REQUIREMENTS:

1. JDK must be installed on the system.
2. Memory: 2GB RAM

2.2 WINDOWS INSTALLATION:

- **Download link:** <https://jenkins.io/>

On the Jenkins Homepage (<https://jenkins.io/>), click **“Download Jenkins”** and select the long-term support release.

- **Starting Jenkins:**

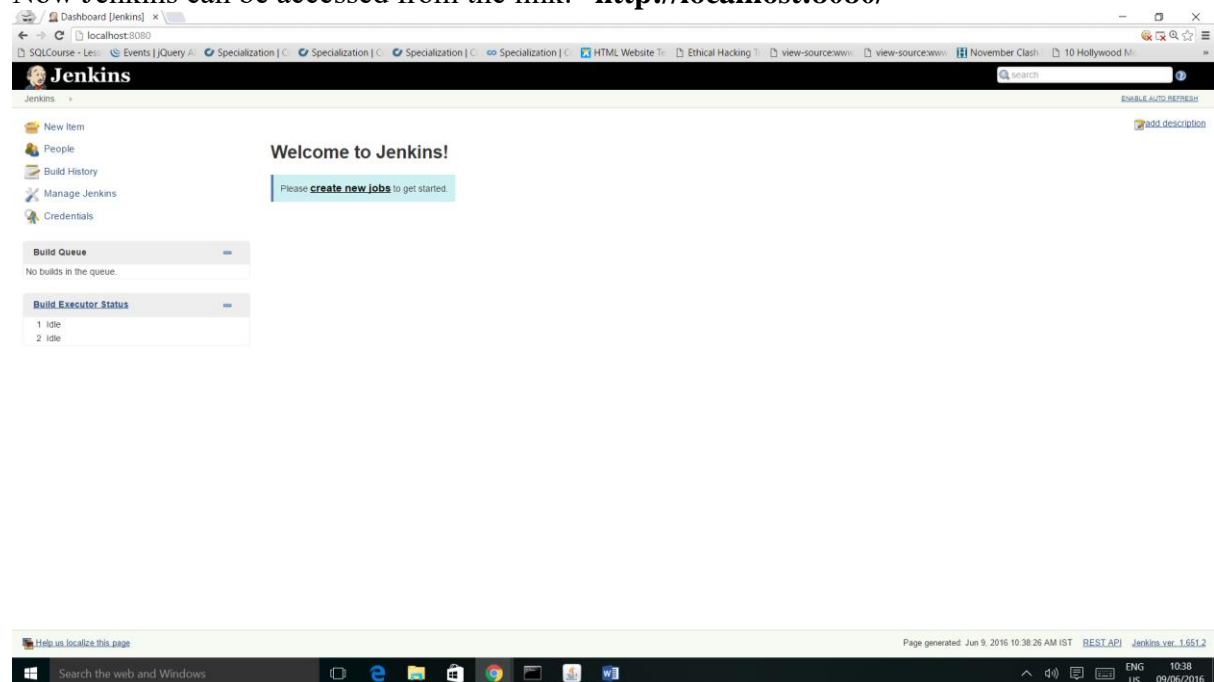
In command prompt, browse to the directory where the jenkins.war file is present. Run the following command:

“Java -jar jenkins.war”

Once processing is complete, the following line will be output:

“Jenkins is fully up and running”

Now Jenkins can be accessed from the link: **“http://localhost:8080/”**



2.3 LINUX INSTALLATION:

Run the following commands in terminal to install Jenkins:

1. `wget -q -O - https://jenkins-ci.org/debian/jenkins-ci.org.key | sudo apt-key add -`
2. `sudo sh -c 'echo deb http://pkg.jenkins-ci.org/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'`
3. `sudo apt-get update`
4. `sudo apt-get install jenkins`

3. JENKINS MODULES:

3.1 NEW ITEM:

This module is used to create a new project/item in Jenkins. To create a new item:

1. Enter Item Name
2. Select type of Project
3. Click OK.

The configuration window for the Item opens up. Enter the configuration details and click Save.

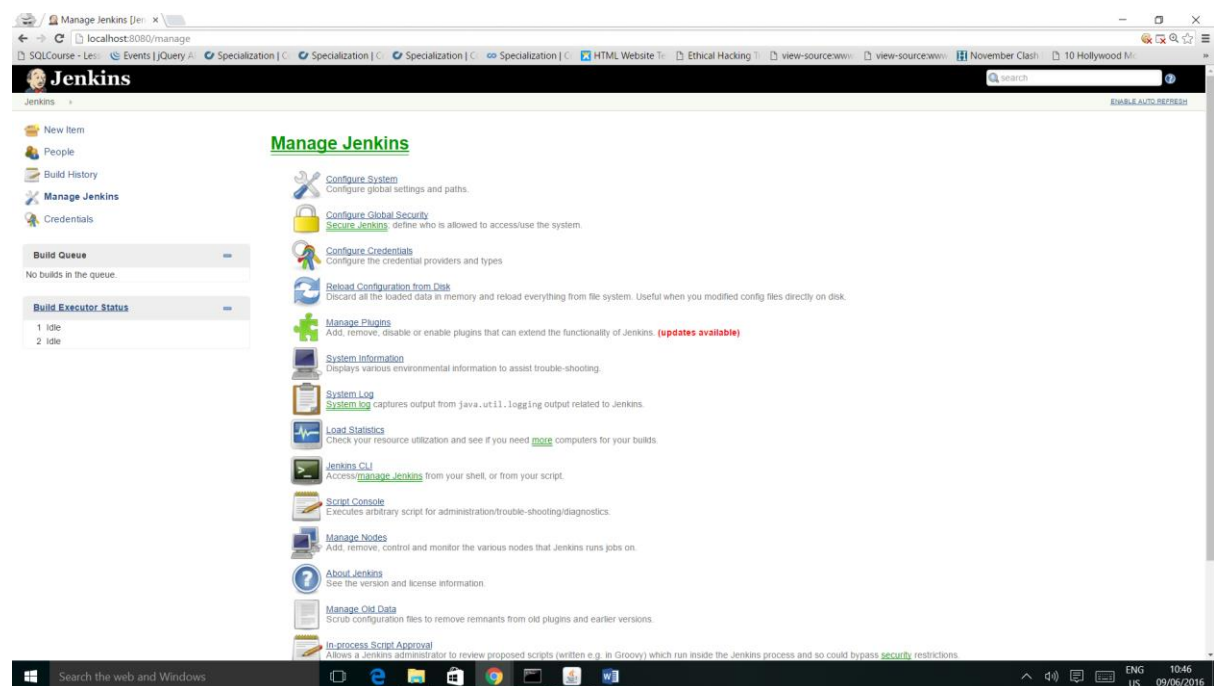
3.2 PEOPLE:

It includes all known “users”, including login identities which the current security realm can enumerate, as well as people mentioned in commit messages in recorded changelog. When a new user is added, his details get added to this section.

3.3 BUILD HISTORY:

It gives the details of all the builds of the items created by the user. A list of all the builds can be found here.

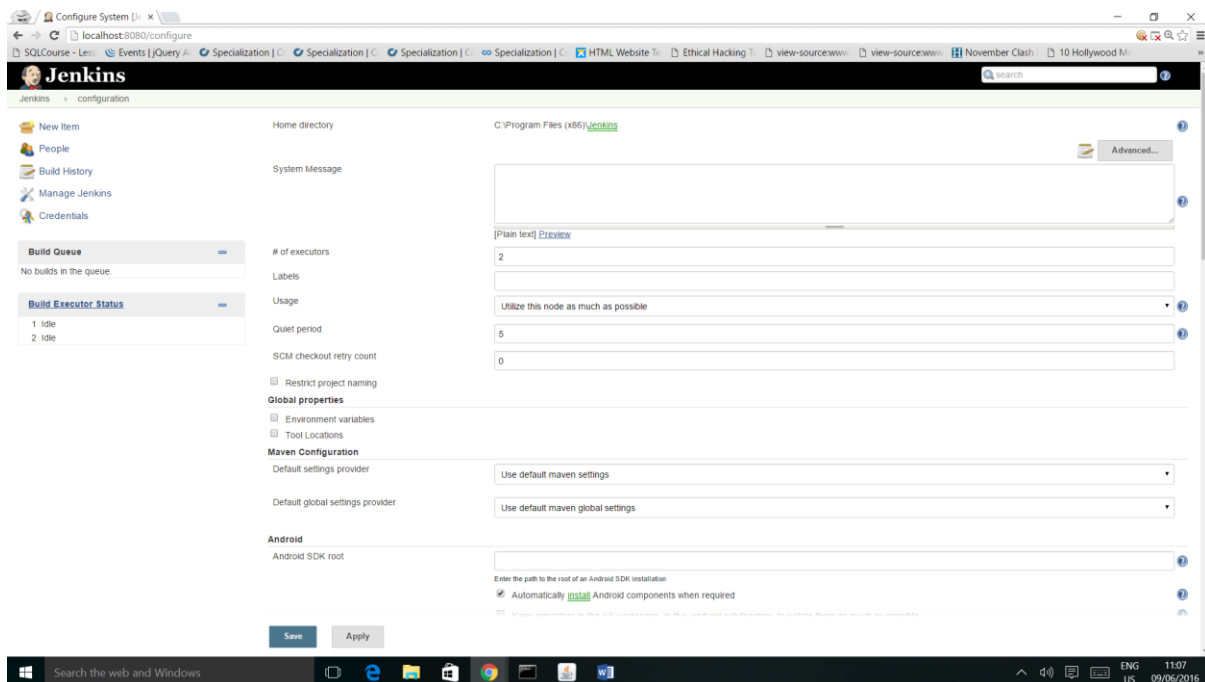
3.4 MANAGE JENKINS:



To manage Jenkins, click on the ‘Manage Jenkins’ option from the left hand menu side. So one can get the various configuration options for Jenkins by clicking the ‘Manage Jenkins’ option from the left hand menu side.

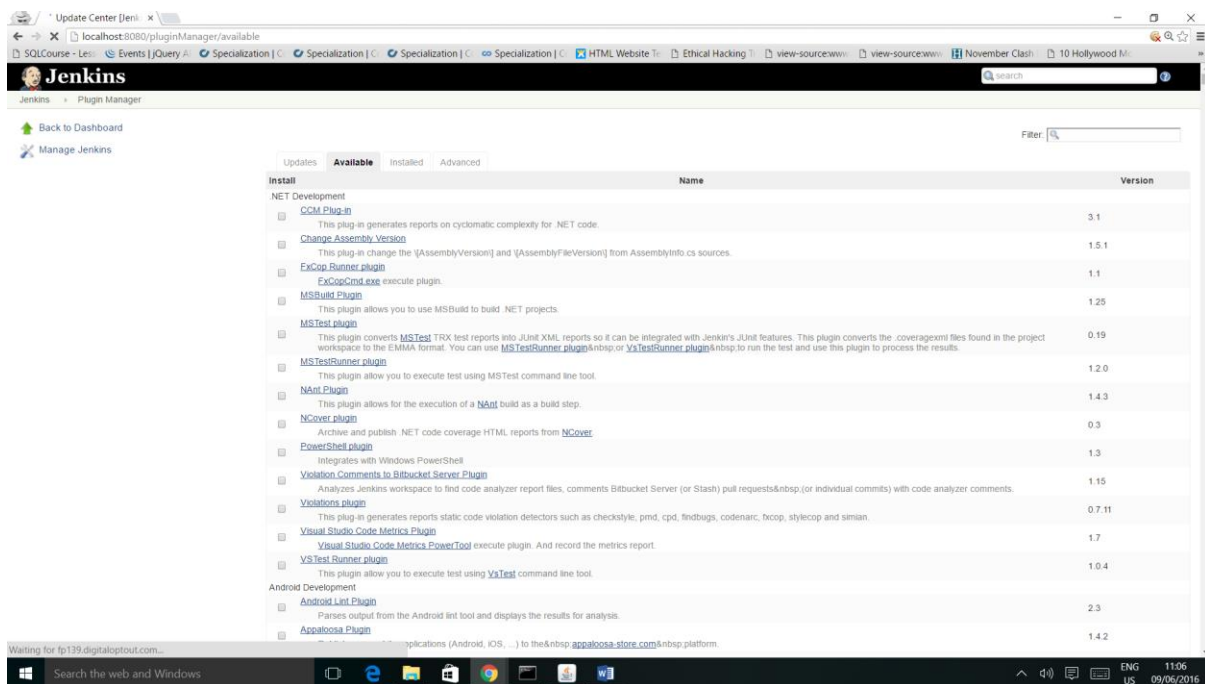
Some of the sub modules available under MANAGE JENKINS are:

3.4.1 CONFIGURE SYSTEM:



This is where one can manage paths to the various tools to use in builds, such as the JDKs, the versions of Ant and Maven, as well as security options, email servers, and other system-wide configuration details. When plugins are installed, Jenkins will add the required configuration fields dynamically after the plugins are installed.

3.4.2 MANAGE PLUGINS:



Here one can install a wide variety of third-party plugins right from different Source code management tools such as Git, Android Emulator Plugin, etc. Plugins can be installed, updated and removed through the Manage Plugins screen.

Some of the plugins are:

- Git Plugin: This plugin integrates Git with Jenkins.
- Android Emulator Plugin: Starts an Android emulator with given properties before a build, then shuts it down after.
- Pipeline Plugin: A suite of plugins that lets you orchestrate automation, simple or complex.
- Ant Plugin: This plugin adds Apache Ant support to Jenkins.
- JUnit Plugin: Allows JUnit-format test results to be published.

3.4.3 CREATING NEW USER:

To create a new user in Jenkins, follow the following steps:

Click Manage Jenkins -> Configure Global Security -> Enable Security -> Jenkins Own User Database -> Save.

This will turn ON the *create users* tab in *Manage Jenkins*.

After this new user can be created using following steps:

Click Manage Jenkins -> Manage User -> Create User -> Add Details -> Signup.

Sign up [Jenkins] - Mozilla Firefox

Sign up [Jenkins]

localhost:8080/securityRealm/addUser

Jenkins

Jenkins - own user database

Back to Dashboard

Manage Jenkins

Create User

Sign up

Username: DnChauz

Password:

Confirm password:

Full name: Danish K. Chauz

E-mail address:

Sign up

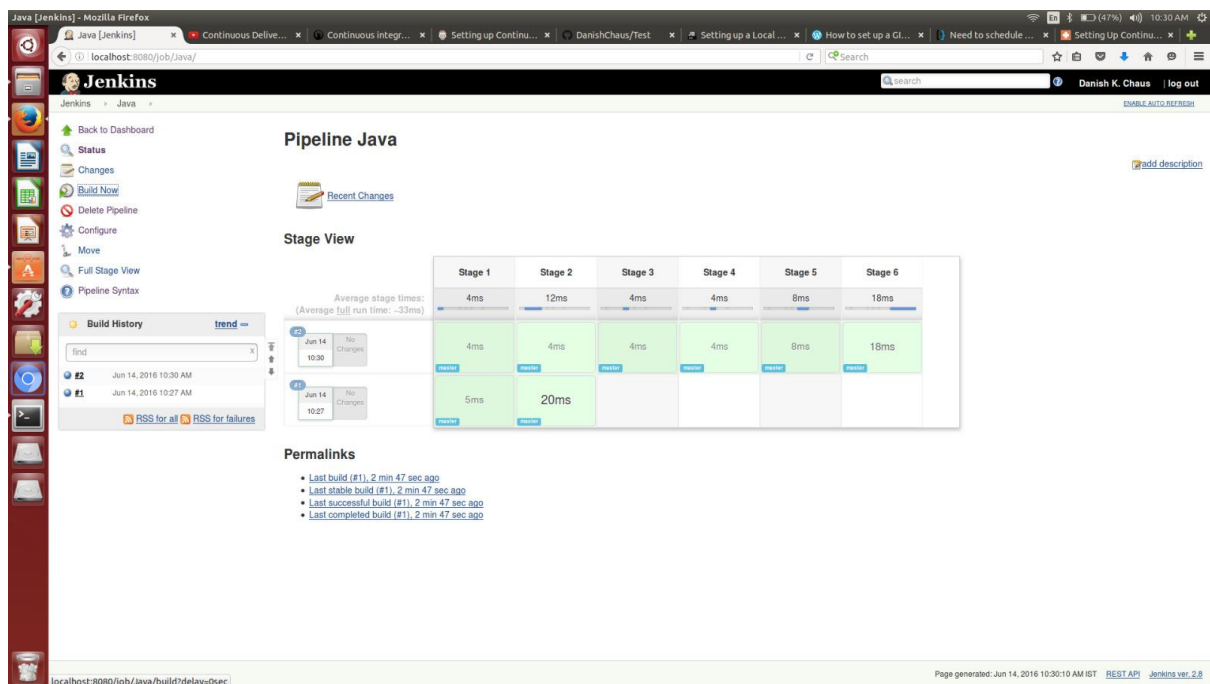
Help us localize this page

Page generated: Jun 28, 2016 9:29:06 AM IST Jenkins ver: 1.651.3

4. PIPELINE IN JENKINS:

Pipelines are Jenkins jobs enabled by the Pipeline (formerly called "workflow") plugin and built with simple text scripts that use a Pipeline DSL (domain-specific language) based on the **Groovy programming language**.

Pipeline functionality helps Jenkins to support continuous delivery (CD). The Pipeline plugin was built with requirements for a flexible, extensible, and script-based CD workflow capability in mind.



Step:

A "step" (often called a "build step") is a single task that is part of sequence. Steps tell Jenkins what to do.

Node:

In pipeline coding contexts, a "node" is a step that does two things, typically by enlisting help from available executors on agents:

- Schedules the steps contained within it to run by adding them to the Jenkins build queue (so that as soon as an executor slot is free on a node, the appropriate steps run).
- Creates a workspace, meaning a file directory specific to a particular job, where resource-intensive processing can occur without negatively impacting your pipeline performance. Workspaces created by node are automatically removed after all the steps contained inside the node declaration finish executing. It is a best practice to do all material work, such as building or running shell scripts, within nodes, because node blocks in a stage tell Jenkins that the steps within them are resource-intensive enough to be scheduled, request help from the agent pool, and lock a workspace only as long as they need it.

In Jenkins generally, "node" also means any computer that is part of your Jenkins installation, whether that computer is used as a master or as an agent.

Stage:

A "stage" is a logically distinct part of the execution of any task, with parameters for locking, ordering, and labeling its part of a process relative to other parts of the same process. Pipeline

syntax is often comprised of stages. Each stage step can have one or more build steps within it. It is a best practice to work within stages because they help with organization by lending logical divisions to a pipelines, and because the Jenkins Pipeline visualization feature displays stages as unique segments of the pipeline.

Familiarity with Jenkins terms such as "master," "agent," and "executor" also helps with understanding how pipelines work. These terms are not specific to pipelines:

Master:

A "master" is the computer where the Jenkins server is installed and running; it handles tasks for your build system. Pipeline scripts are parsed on masters, where Groovy code runs and node blocks allocate executors and workspaces for use by any nested steps (such as sh) that might request one or both.

Agent:

An "agent" (formerly "slave") is a computer set up to offload available projects from the master. Your configuration determines the number and scope of operations that an agent can perform. Operations are performed by executors.

Executor:

An "executor" is a computational resource for running builds or Pipeline steps. It can run on master or agent machines, either by itself or in parallel with other executors.