

**Instructor Notes:**

Add instructor notes  
here.

**DevOps**

Lesson 04-Jenkins

## Instructor Notes:

Add instructor notes here.

## Lesson Objectives

- Introduction to CI
- Jenkins Introduction
- Creating Job in Jenkins
- Adding plugin in Jenkins
- Creating Job with Maven & Git



**Instructor Notes:**

Add instructor notes here.

4.1: Introduction to CI

## Continuous Integration(CI)

- Continuous Integration involves a tool that monitors version control system for any changes and automates application building.
- CI system must be executed under configuration management.
- Developers are notified automatically if any build action fails.
- CI brings a practice to integrate work frequently in software development.
- Monitoring of Code Quality and Code coverage metrics is automated.



Copyright © Capgemini 2016. All Rights Reserved 5

In a Continuous Integration Environment source code is maintained in a central location where an application monitors the repository and springs into action when it notices changes (commits) to the code.

CI System must be able to be built and tested automatically.

A coding standard is the set of guidelines that developers must adhere to on a project. On many projects, ensuring adherence is largely a manual process that is performed by a code review. CI can run a build script to report on adherence to the coding standards by running a suite of automated static analysis tools that inspect the source code against the established standard whenever a change is applied

## Instructor Notes:

Add instructor notes here.

4.1: Introduction to CI

# Why CI?


- Software Development Before CI

Code changes made by individual team members are merged together into working software, which was known as Integration phase.


Integration phase was a hard work which often results in code conflicts, hard to find bugs and even harder to fix them which lead to significant delivery delays

### Multiple changes!

writing code  
merging code  
changing code



Today businesses need new features to be incorporated into application into days/weeks not months. This requires a change in how softwares are built.

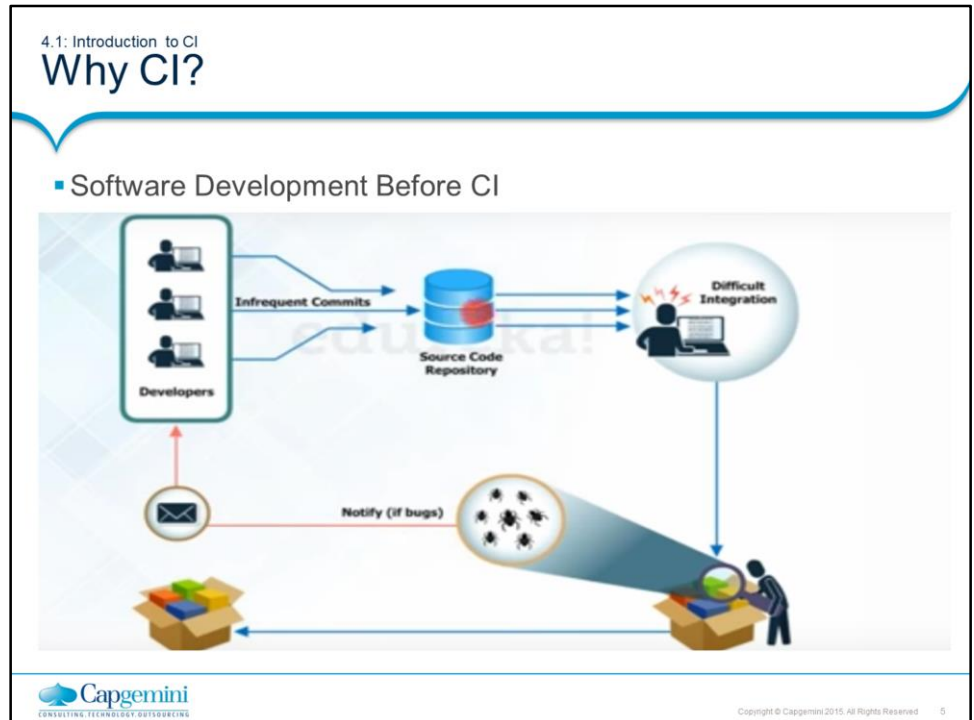


Capgemini  
CONSULTING TECHNOLOGY OUTSOURCING

Copyright © Capgemini 2016. All Rights Reserved 4

## Instructor Notes:

Add instructor notes here.



**Instructor Notes:**

Add instructor notes here.


4.1: Introduction to CI

## Problem-Before CI

Developers have to wait till the complete software is developed for the test results.

I hope the code works fine in testing

Software delivery process was slow




If the test fails then locating and fixing bugs is very difficult. Developers have to check the entire source code of the software.

I have to check the entire source code

Entire source code of the software

Continuous feedback pertaining to things like coding or architectural issues, build failures, test status etc. was not present

The Feedback loop  
Build, Measure and  
Learn

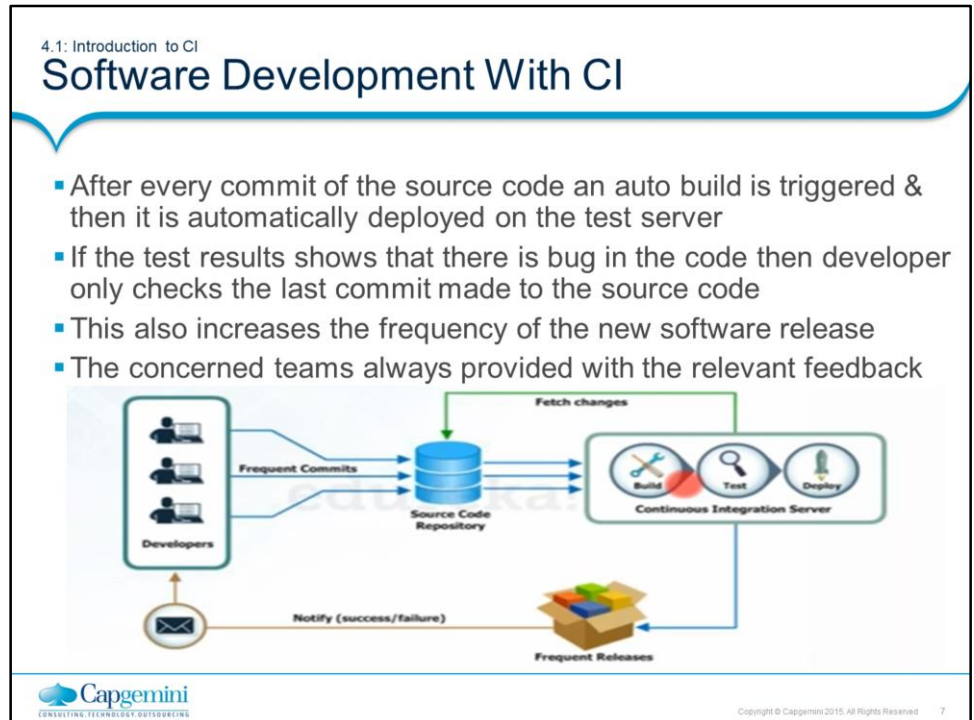


Capgemini  
CONSULTING TECHNOLOGY OUTSOURCING

Copyright © Capgemini 2016. All Rights Reserved 6

**Instructor Notes:**

Add instructor notes here.



- 1: Application must run under source control management
- 2: Daily code commits to SVN will be baseline.
- 3: CI polls for any code changes in SVN and triggers build actions if any.
- 4: Automated build, testing and deployment of an application will be performed by CI.
- 5: After build action, developer will be accessible with latest code and build
- 6: Developers will be notified with any build errors and automated test results.

### Need of CI in software Development

Helps to locate code based defects in a centralized location.

Tools can be used to automate deployment.

Minimizes integration errors in SVN during build process (Errors are uncovered during Manual Build) by invoking automation.

Increase amount of quality code and improve development standards.

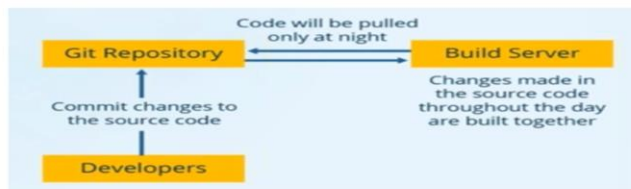
## Instructor Notes:

Tell about scenario if user is doing with or without CI .Give example of Nokia uploading

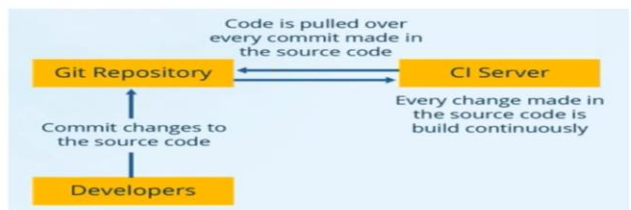
4.1: Introduction to CI

### Case Study on - CI

#### Without CI



#### With CI-Problem Solved





**Instructor Notes:**

Add instructor notes here.

4.1: Introduction of CI

## Continuous Integration -CI

Benefit to CI:

- Aims to eliminate code integration issues
- Minimizes project risk with notification of defects and code quality issues
- Reduces cost of quality
- Early warning of conflicting changes code
- Automation of build and testing of an application



Copyright © Capgemini 2016. All Rights Reserved 9

**Reduce risks**

By integrating many times a day, you can reduce risks on your project. Doing so facilitates the detection of defects, the measurement of software health and a reduction of assumptions.

**Defects are detected and fixed sooner:** Because CI integrates and runs tests and inspections several times a day, there is a greater chance that defects are discovered *when they are introduced* (i.e., when the code is checked into the version-control repository) instead of during late-cycle testing.

**Health of software is measurable:** By incorporating continuous testing and inspection into the automated integration process, the software product's health attributes, such as complexity, can be tracked over time.

**Reduce assumptions:** By rebuilding and testing software in a clean environment using the same process and scripts on a continual basis, you can reduce assumptions (e.g., whether you are accounting for third-party libraries or environment variables).

CI provides a safety net to reduce the risk that defects will be introduced into the code base. The following are some of the risks that CI helps to mitigate. We discuss these and other risks in the next chapter.

Lack of cohesive, deployable software

Late defect discovery

Low-quality software

Lack of project visibility

**Instructor Notes:**

4.1: Introduction of CI

## Continues Integration Tools

- Jenkins
- Buildbot
- Travis CI
- Bamboo

**Instructor Notes:**

Add instructor notes here.

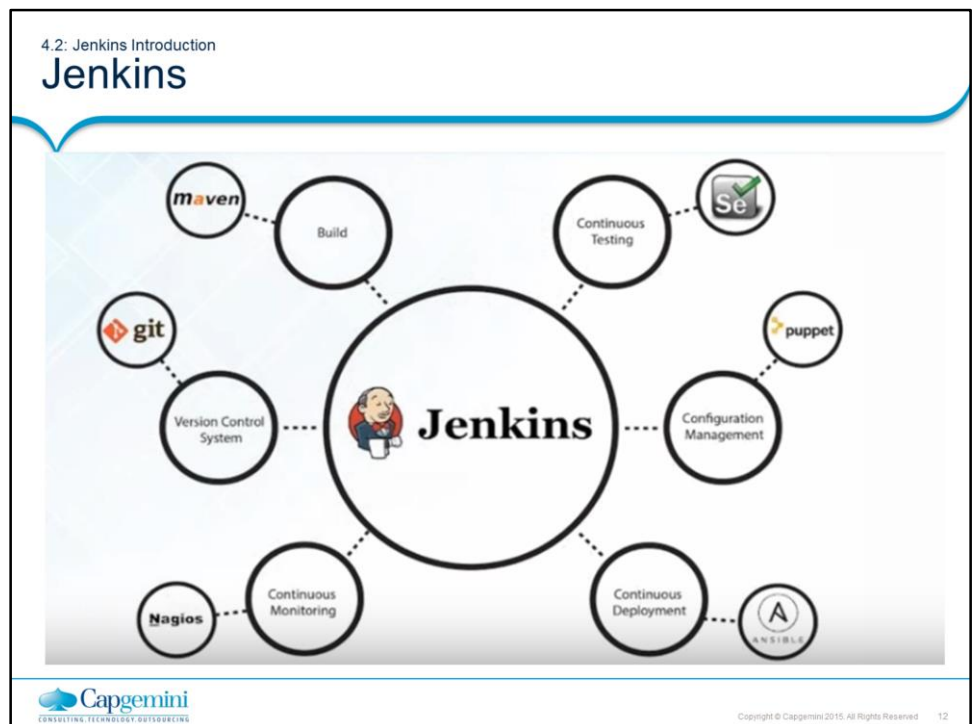
4.2: Jenkins Introduction

## Jenkins

- Jenkins is a self-contained, open source automation server which can be used to automate all sorts of tasks such as building, testing, and deploying software.
- Jenkins is an open source continuous integration(CI) tool written in java developed by Kohsuke Kawaguchi.
- Monitors the change in the source control systems like SVN, CVS, etc.
- Builds the application using various build tools like ANT, MAVEN, etc.
- Provides a fresh build whenever there is a change in the source control system
- Sends messages on the status of the build through Email, SMS, etc
- Plugins allows integration of the various DevOps Stage

## Instructor Notes:

What different models we can integrate with jenkins



**Instructor Notes:**

Add instructor notes here.

4.2: Jenkins Introduction

How Jenkins Works

How Jenkins works:

▪ Developers Commit changes to the source code

▪ CI server pulls that code & triggers a build

▪ The build application is then deployed on testing server for testing

▪ After testing the application ,it is then deployed on production server

▪ The concerned team constantly notified about build & test result

```
graph LR; subgraph Developers; D1[Developer]; D2[Developer]; D3[Developer]; end; SCR[(Source Code Repository)]; JS[Jenkins Server]; T[Testing]; P[Production]; JS -- Feedback --> D1; JS -- Feedback --> D2; JS -- Feedback --> D3; D1 --> SCR; D2 --> SCR; D3 --> SCR; JS --> SCR; JS -- "Prepares a build" --> T; T --> P;
```

Capgemini  
CONSULTING TECHNOLOGY OUTSOURCING

Copyright © Capgemini 2016. All Rights Reserved 13

**Instructor Notes:**

Add instructor notes here.

4.2: Jenkins Introduction

## Jenkins Installation

- Jenkins is easy to install.
- Download Jenkins.war file from the Jenkins site:
  - <http://jenkins-ci.org>
- Jenkins can be installed in different ways:
  - As a standalone application
  - Windows Service
  - Deploy it on any application server.

**Instructor Notes:**

4.2: Jenkins Introduction

## Jenkins Installation

- To start Jenkins as a standalone application execute the below command in command prompt:
  - `java -jar jenkins.war -- On Port 8080`
  - `java -jar jenkins.war --ajp13Port=-1 --httpPort=8082 --On different port`
  - Once Jenkins is started, the Jenkins dash board can be accessed by giving the following link in the browser  
<http://localhost:8080/>
  - To stop Jenkins, press Ctrl+C
- Below are the steps to start Jenkins as a windows service
  - First, start Jenkins as a standalone application and access Jenkins dash board.
  - Click "Manage Jenkins" link available in Jenkins dash board.
  - Select "Installation Directory" for Jenkins and click on Install.
  - After installation, Jenkins will always run on portno 8080.



Copyright © Capgemini 2016. All Rights Reserved 15

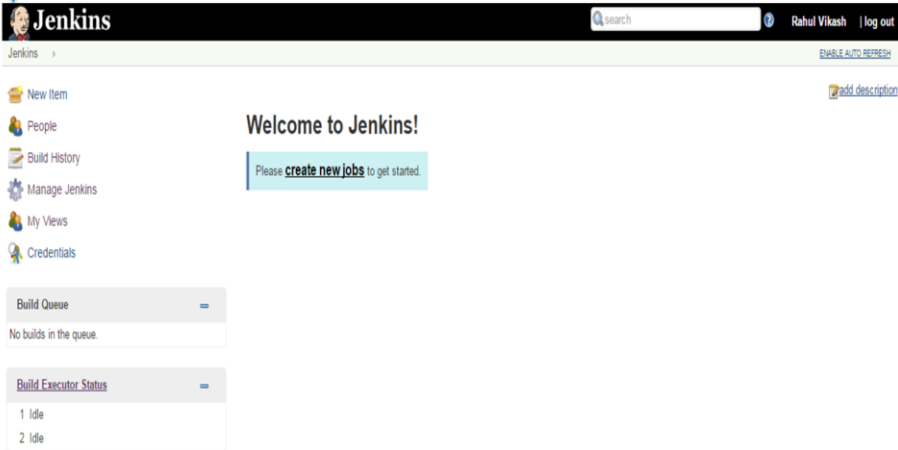
By default, Jenkins will run on the 8080 port. To specify the port manually, use the --httpPort option:

```
java -jar jenkins.war --httpPort=8081
```

**Instructor Notes:**

4.3: Creating Job in Jenkins

## Jenkins Installation



The screenshot shows the Jenkins web interface. At the top, there's a header with the Jenkins logo, a search bar, and user information (Rahul Vikash, log out). Below the header, there's a sidebar with navigation links: New Item, People, Build History, Manage Jenkins, My Views, and Credentials. The main content area displays a 'Welcome to Jenkins!' message with a button to 'create new jobs'. Below this, there are two sections: 'Build Queue' showing 'No builds in the queue' and 'Build Executor Status' showing two idle executors.

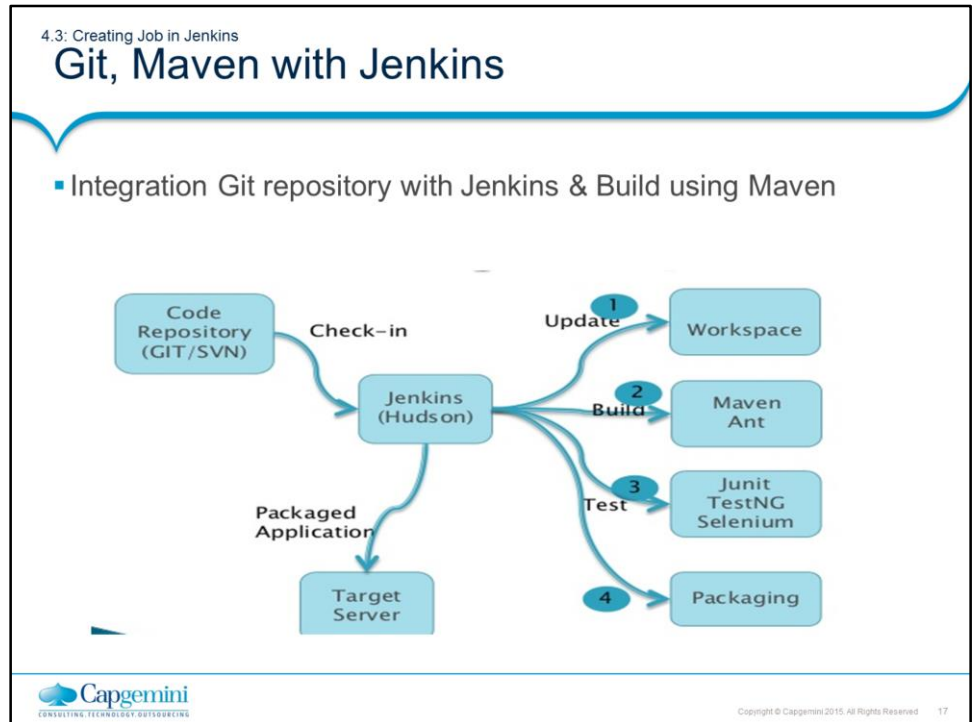
Capgemini  
CONSULTING TECHNOLOGY OUTSOURCING

Copyright © Capgemini 2016. All Rights Reserved 16

By default, Jenkins will run on the 8080 port. To specify the port manually, use the --httpPort option:

```
java -jar jenkins.war --httpPort=8081
```



**Instructor Notes:**

By default, Jenkins will run on the 8080 port. To specify the port manually, use the `--httpPort` option:

```
java -jar jenkins.war --httpPort=8081
```

Instructor Notes:

4.4: Adding plugin in Jenkins

Manage plugins

Jenkins

search

Rahul Vikash | log out

Jenkins

ENABLE AUTO REFRESH

New Item

People

Build History

Manage Jenkins

My Views

Credentials

Build Queue

No builds in the queue

Build Executor Status

1 Idle

2 Idle

Click on Manage Jenkins

Manage Jenkins

New version of Jenkins (2.32.2) is available for [download](#) ([changelog](#)). 

Or Upgrade Automatically

Configure System

Configure global settings and paths.

Configure Global Security

Secure Jenkins; define who is allowed to access/use the system.

Configure Credentials

Configure the credential providers and types

Global Tool Configuration

Configure tools, their locations and automatic installers.

Reload Configuration from Disk

Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.

Manage Plugins

Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

System Information

Displays various environmental information to assist trouble-shooting.

Click on Manage Plugin, Go to Available & such for plugins

Capgemini

CONSULTING TECHNOLOGY OUTSOURCING

Copyright © Capgemini 2016. All Rights Reserved

18

By default, Jenkins will run on the 8080 port. To specify the port manually, use the --httpPort option:

```
java -jar jenkins.war --httpPort=8081
```

Instructor Notes:


4.4: Adding plugin in Jenkins

Manage plugins

Download Maven ,Git plugin

Download GIT Maven Sonar plugin

<input checked="" type="checkbox"/>	Git client plugin	2.2.1	
Shared library: plugin for other Git-related Jenkins plugins			
<input checked="" type="checkbox"/>	Git plugin	3.0.5	Downgrade to 3.0.4
<input checked="" type="checkbox"/>	This plugin integrates Git with Jenkins.		
<input checked="" type="checkbox"/>	GitHub API Plugin	1.84	
<input checked="" type="checkbox"/>	This plugin provides GitHub API for other plugins.		
<input checked="" type="checkbox"/>	GitHub Integration Plugin	0.1.0-rc20	Downgrade to 0.1.0-rc19
<input checked="" type="checkbox"/>	Advanced trigger for GitHub Pull Requests and Branches.		
<input checked="" type="checkbox"/>	GitHub plugin	1.26.0	Downgrade to 1.25.1
<input checked="" type="checkbox"/>	This plugin integrates GitHub to Jenkins.		
<input checked="" type="checkbox"/>	Javadoc Plugin	1.4	
<input checked="" type="checkbox"/>	This plugin adds Javadoc support to Jenkins.		
<input checked="" type="checkbox"/>	JUnit Plugin	1.19	
<input checked="" type="checkbox"/>	Allows JUnit-format test results to be published.		
<input checked="" type="checkbox"/>	Mailer Plugin	1.19	
<input checked="" type="checkbox"/>	This plugin allows you to configure email notifications for build results. This is a break-out of the original core based email component.		
<input checked="" type="checkbox"/>	Matrix Project Plugin	1.8	
<input checked="" type="checkbox"/>	Multi-configuration (matrix) project type.		
<input checked="" type="checkbox"/>	Maven Integration plugin	2.15.1	
<input checked="" type="checkbox"/>	This plugin provides an advanced integration for Maven 2/3 projects.		



Copyright © Capgemini 2016. All Rights Reserved 19

By default, Jenkins will run on the 8080 port. To specify the port manually, use the --httpPort option:

```
java -jar jenkins.war --httpPort=8081
```

**Instructor Notes:**

4.4: Adding plugin in Jenkins

## Manage plugins

- Setting Configuration
  - Go to Manage Jenkin->Global Tools Configuration

The screenshot shows the Jenkins 'Manage plugins' page with the 'Global Tools Configuration' tab selected. It displays configuration for three tools: JDK, Git, and Maven. Each tool has a 'Name' field, a 'Path' field, and an 'Install automatically' checkbox. The 'JDK' section shows 'JDK1.8' with path 'C:\Program Files\Java\jdk1.8.0\_31'. The 'Git' section shows 'Default' with path 'C:\Program Files\Git\bin\git.exe'. The 'Maven' section shows 'Maven3.2.5' with path 'D:\maven\apache-maven-3.2.5'. Arrows point to the 'Path' fields with labels: 'Put JDK Path', 'Use Git.Exe Path', and 'Use Maven Path'.

Capgemini  
CONSULTING TECHNOLOGY OUTSOURCING

Copyright © Capgemini 2016. All Rights Reserved 20

By default, Jenkins will run on the 8080 port. To specify the port manually, use the --httpPort option:

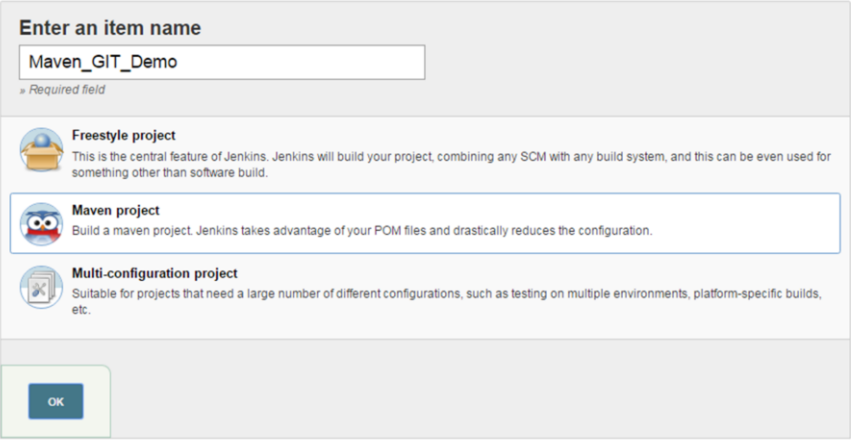
```
java -jar jenkins.war --httpPort=8081
```

**Instructor Notes:**

4.5: Creating Job with Maven & Git

## Creating Maven Project

- Create a Job, Give Job Name ,Select Maven Project & press Ok



**Enter an item name**

Maven\_GIT\_Demo


» Required field

**Freestyle project**  
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

**Maven project**  
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

**Multi-configuration project**  
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

OK

 Capgemini  
CONSULTING TECHNOLOGY OUTSOURCING

Copyright © Capgemini 2016. All Rights Reserved 21

By default, Jenkins will run on the 8080 port. To specify the port manually, use the --httpPort option:

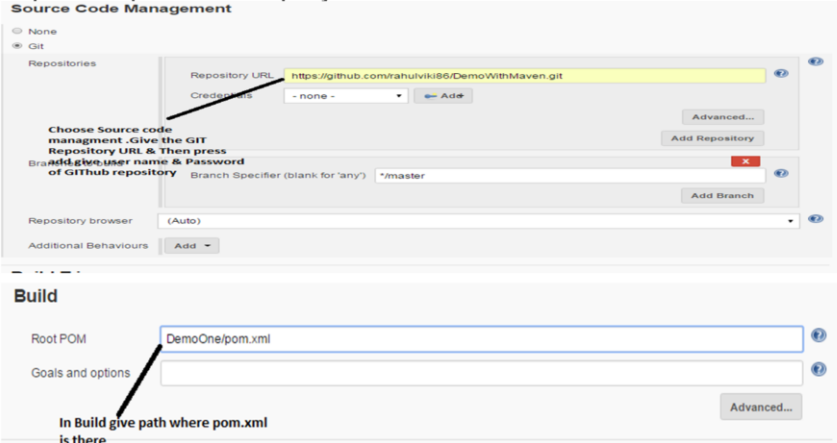
```
java -jar jenkins.war --httpPort=8081
```

**Instructor Notes:**

4.5: Creating Job with Maven & Git

## Creating Maven Project

- Integrating Git with Jenkins by giving repository url(GitHub URL) & path of pom.xml in project



The screenshot shows the Jenkins configuration interface. The 'Source Code Management' section is active, with 'Git' selected. The 'Repository URL' is set to 'https://github.com/rahuiviki85/DemoWithMaven.git'. The 'Branch Specifier' is set to '\*/master'. The 'Build' section shows the 'Root POM' set to 'DemoOne/pom.xml'. Annotations with arrows point to the 'Repository URL' and 'Root POM' fields, with text indicating to 'add give user name & Password of Github repository' and 'In Build give path where pom.xml is there' respectively.

Capgemini  
CONSULTING TECHNOLOGY OUTSOURCING

Copyright © Capgemini 2016. All Rights Reserved 22

By default, Jenkins will run on the 8080 port. To specify the port manually, use the --httpPort option:

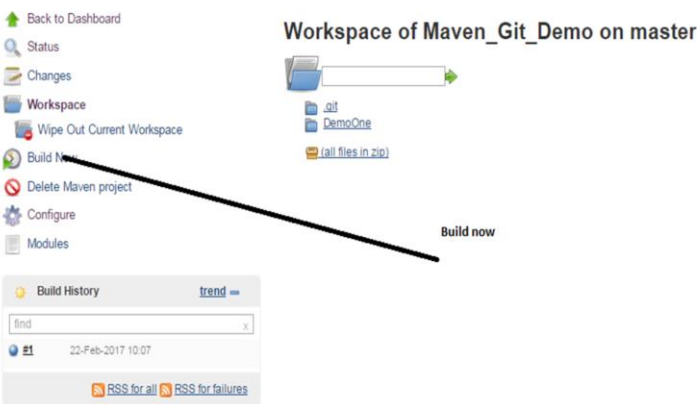
```
java -jar jenkins.war --httpPort=8081
```

**Instructor Notes:**

4.5: Creating Job with Maven & Git

## Creating Maven Project

- Save & check in workspace all data fetched from Git Repository & then build



Workspace of Maven\_Git\_Demo on master

Build now

Build History

find

22-Feb-2017 10:07

RSS for all RSS for failures

Capgemini

Copyright © Capgemini 2016. All Rights Reserved 23

By default, Jenkins will run on the 8080 port. To specify the port manually, use the --httpPort option:

```
java -jar jenkins.war --httpPort=8081
```

**Instructor Notes:**

## 4.5: Creating Job with Maven &amp; Git

## Creating Maven Project

## T E S T S

```
-----
Running com.cg.demoone.AppTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.002 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[JENKINS] Recording test results
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ DemoOne ---
[INFO] Building jar: C:\Users\rvb30051\.jenkins\jobs\Maven_Git_Demo\workspace\DemoOne\target\DemoOne-1.0-SNAPSHOT.jar
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ DemoOne ---
[INFO] Installing C:\Users\rvb30051\.jenkins\jobs\Maven_Git_Demo\workspace\DemoOne\target\DemoOne-1.0-SNAPSHOT.jar to
C:\Users\rvb30051\.m2\repository\com\cg\demoone\DemoOne\1.0-SNAPSHOT\DemoOne-1.0-SNAPSHOT.jar
[INFO] Installing C:\Users\rvb30051\.jenkins\jobs\Maven_Git_Demo\workspace\DemoOne\pom.xml to C:\Users\rvb30051\.m2\repository\com\cg\demoone\DemoOne\1.0-
SNAPSHOT\DemoOne-1.0-SNAPSHOT.pom
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 19.984 s
[INFO] Finished at: 2017-02-22T18:07:57+05:30
[INFO] Final Memory: 14M/34M
[INFO] -----
[JENKINS] Archiving C:\Users\rvb30051\.jenkins\jobs\Maven_Git_Demo\workspace\DemoOne\pom.xml to com.cg.demoone/DemoOne/1.0-SNAPSHOT/DemoOne-1.0-
SNAPSHOT.pom
[JENKINS] Archiving C:\Users\rvb30051\.jenkins\jobs\Maven_Git_Demo\workspace\DemoOne\target\DemoOne-1.0-SNAPSHOT.jar to com.cg.demoone/DemoOne/1.0-
SNAPSHOT/DemoOne-1.0-SNAPSHOT.jar
channel stopped
Finished: SUCCESS
```



Copyright © Capgemini 2016. All Rights Reserved 24

By default, Jenkins will run on the 8080 port. To specify the port manually, use the --httpPort option:

```
java -jar jenkins.war --httpPort=8081
```



**Instructor Notes:**

4.5: Creating Job with Maven & Git

Creating Maven Project

▪ Check & get all feedback when it changes

▪ Build result obtained from the dashboard

add description

S	W	Name ↓	Last Success	Last Failure	Last Duration
		<a href="#">Maven_Git_Demo</a>	8 min 3 sec - <a href="#">#1</a>	N/A	55 sec

Icon: [S](#) [M](#) [L](#)

Legend

RSS for all

RSS for failures

RSS for just latest builds

Capgemini

CONSULTING TECHNOLOGY OUTSOURCING

Copyright © Capgemini 2016. All Rights Reserved

25

By default, Jenkins will run on the 8080 port. To specify the port manually, use the --httpPort option:

```
java -jar jenkins.war --httpPort=8081
```


Page XX-25


## Instructor Notes:

Add instructor notes here.

# Demo

- Demo on Maven-Git-Jenkins integration



Copyright © Capgemini 2016. All Rights Reserved 26

Add the notes here.

## Instructor Notes:

Add instructor notes here.

### Lab

#### ■ Lab 02



Copyright © Capgemini 2016. All Rights Reserved 27

Add the notes here.

## Instructor Notes:

Add instructor notes here.

### Summary

- Continuous Integration involves a tool that monitors version control system for any changes and automates application building
- Jenkins is an open source continuous integration(CI) tool
- Integration Jenkins with Git & Maven



Add the notes here.

**Instructor Notes:**

Q1 Continuous Integration provides solutions to the testers for the failed test cases.

## Review Question

- Which of the given statement is not correct for Continuous Integrations?
  - Continuous Integration is about reducing the risk by providing faster feedback.
  - Continuous Integration involves a tool that monitors version control system for changes.
  - Continuous Integration provides solutions to the testers for the failed test cases.
  - Continuous Integration helps End user to the testers and the end users faster, more reliably, and with less efforts.
- Which command execution will start Jenkins as a standalone application?
  - `jenkins.war`
  - `java -jar jenkins.war`



Copyright © Capgemini 2015. All Rights Reserved 29

Add the notes here.

**Instructor Notes:**

Q1 Continuous Integration provides solutions to the testers for the failed test cases.

Q2. java -jar jenkins.war

Q3 Continuous Deployment

## Review Question

- java jenkins.war
- None of the above
- \_\_\_\_\_ is the process of deploying the latest code into production.
  - Build job
  - Continuous Deployment
  - Continuous Testing
  - None of the above



Add the notes here.