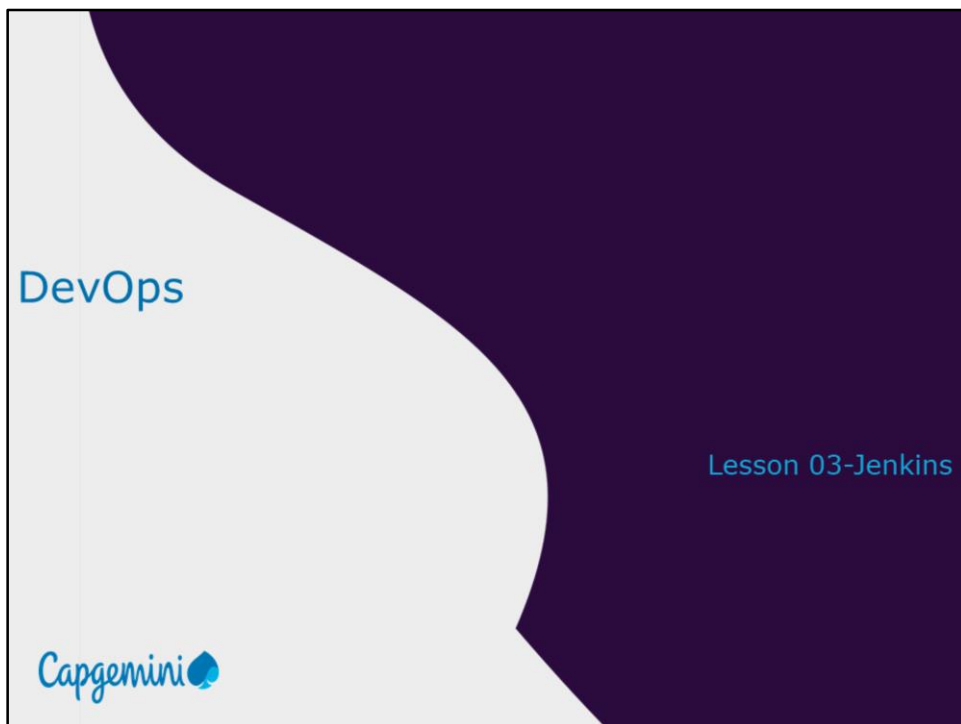


Instructor Notes:

Add instructor notes here.



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.

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.

Capgemini 

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.

The slide is titled "Introduction to CI" and "Why CI?". It features a blue header with a small blue icon in the top right corner. The main content is divided into three sections. The first section, titled "Software Development Before CI", contains two paragraphs: "Code changes made by individual team members are merged together into working software, which was known as Integration phase." and "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". The second section, titled "Multiple changes!", shows a list of activities: "writing code", "merging code", and "changing code", each followed by a blue box containing the word "code". To the right of this list is a small icon of a person with a lightbulb above their head. The third section, at the bottom, contains a paragraph: "Today businesses need new features to be incorporated into application into days/weeks not months. This requires a change in how softwares are built." The Capgemini logo is in the bottom left corner.

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!

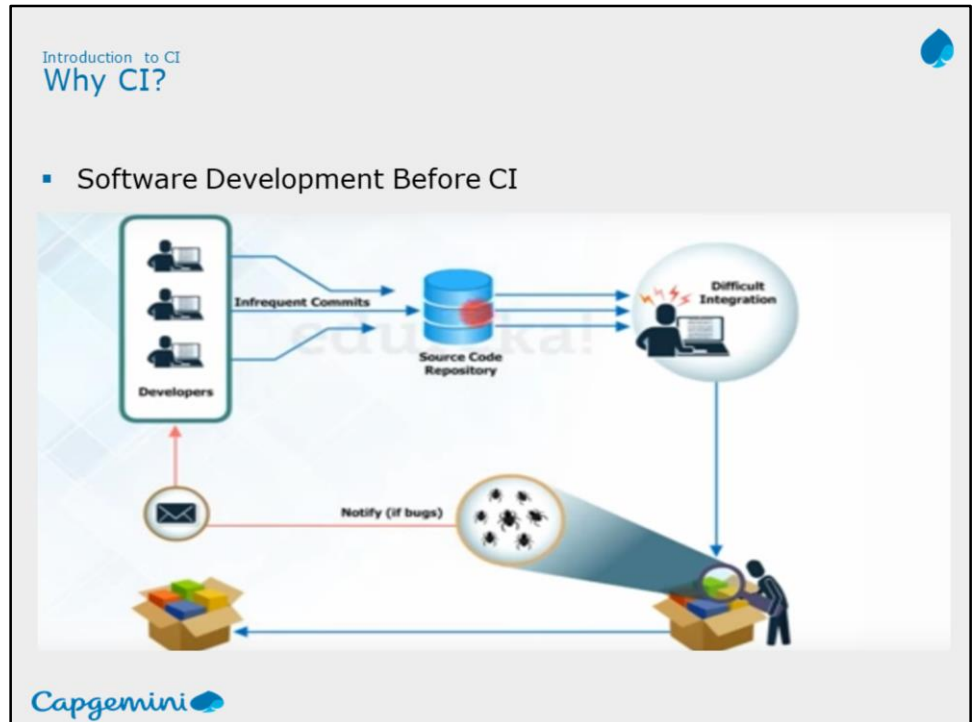
writing code
merging code
changing code

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

Instructor Notes:

Add instructor notes here.



Instructor Notes:

Add instructor notes here.

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

Instructor Notes:


Add instructor notes here.


Introduction to CI

Software Development With CI



- After every commit of the source code an auto build is triggered & then it is automatically deployed on the test server
- If the test results shows that there is bug in the code then developer only checks the last commit made to the source code
- This also increases the frequency of the new software release
- The concerned teams always provided with the relevant feedback





- 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.

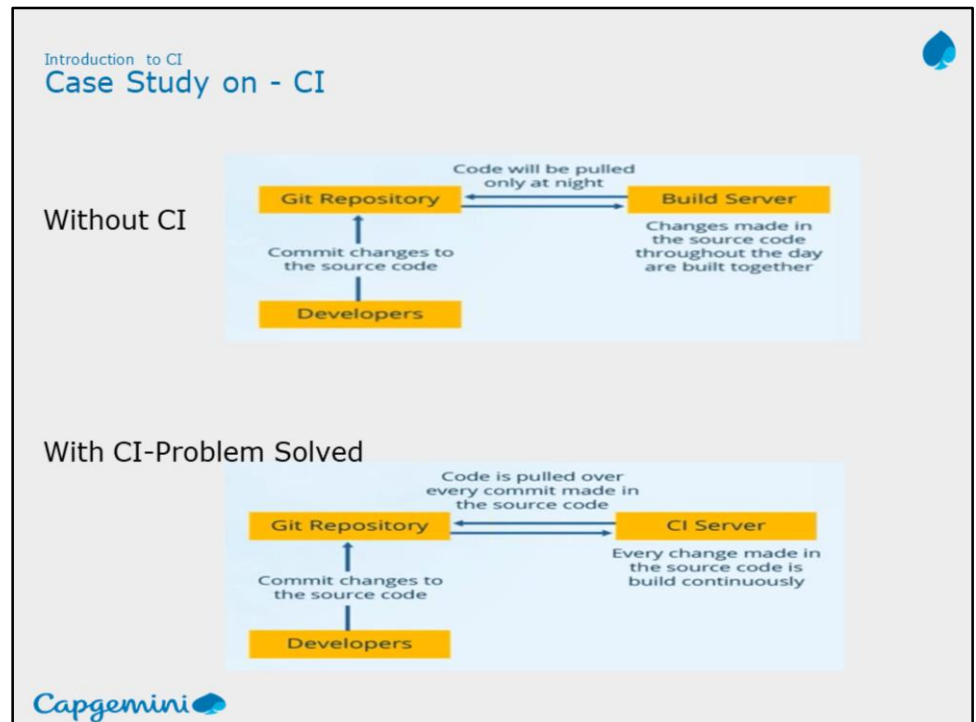
Tools can be used to automate deployment.

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



Instructor Notes:

Add instructor notes here.

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

Capgemini **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:

Introduction of CI

Continues Integration Tools

- Jenkins
- Buildbot
- Travis CI
- Bamboo

Capgemini

Instructor Notes:

Add instructor notes here.

Jenkins Introduction
Jenkins

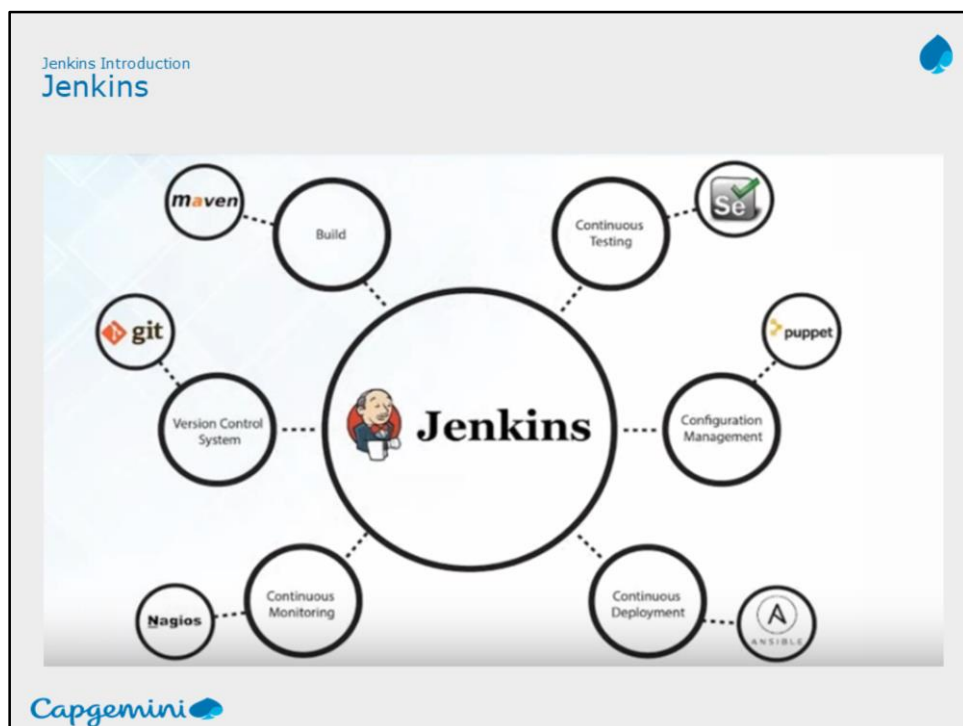


- Jenkins is a self-contained, open source automation server which can be used to automate all sorts of tasks such as building, testing, and deploying software.
- Jenkins is an open source continuous integration(CI) tool written in java developed by Kohsuke Kawaguchi.
- Monitors the change in the source 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

Capgemini 

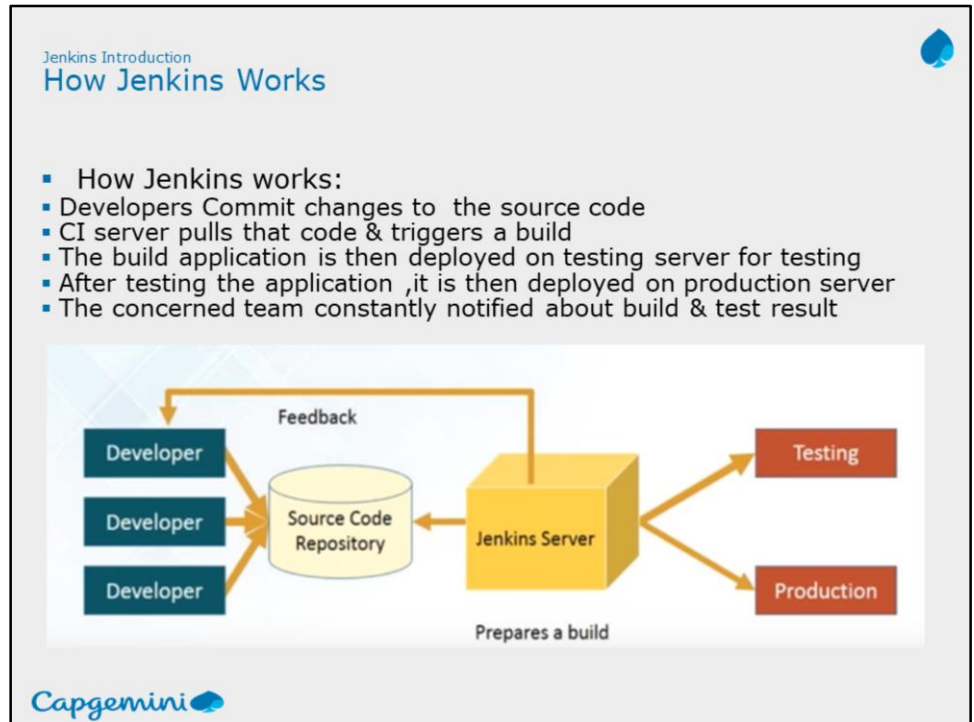
Instructor Notes:

What different models we can integrate with jenkins



Instructor Notes:

Add instructor notes here.



Instructor Notes:

Add instructor notes here.

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.

Capgemini 


Instructor Notes:

Jenkins Introduction

Jenkins Installation

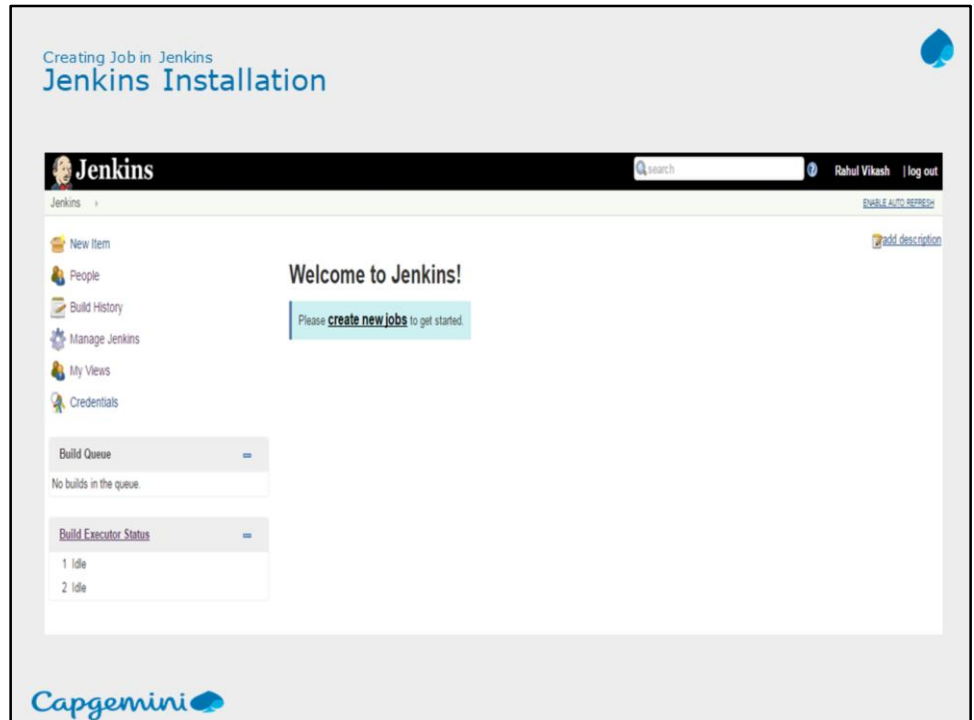


- To start Jenkins as a standalone application execute the below command in command prompt:
- `java -jar jenkins.war -- On Port 8080`
- `java -jar jenkins.war --ajp13Port=-1 --httpPort=8082 -On different port`
- Once Jenkins is started, the Jenkins dash board can be accessed by giving the following link 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.

Capgemini

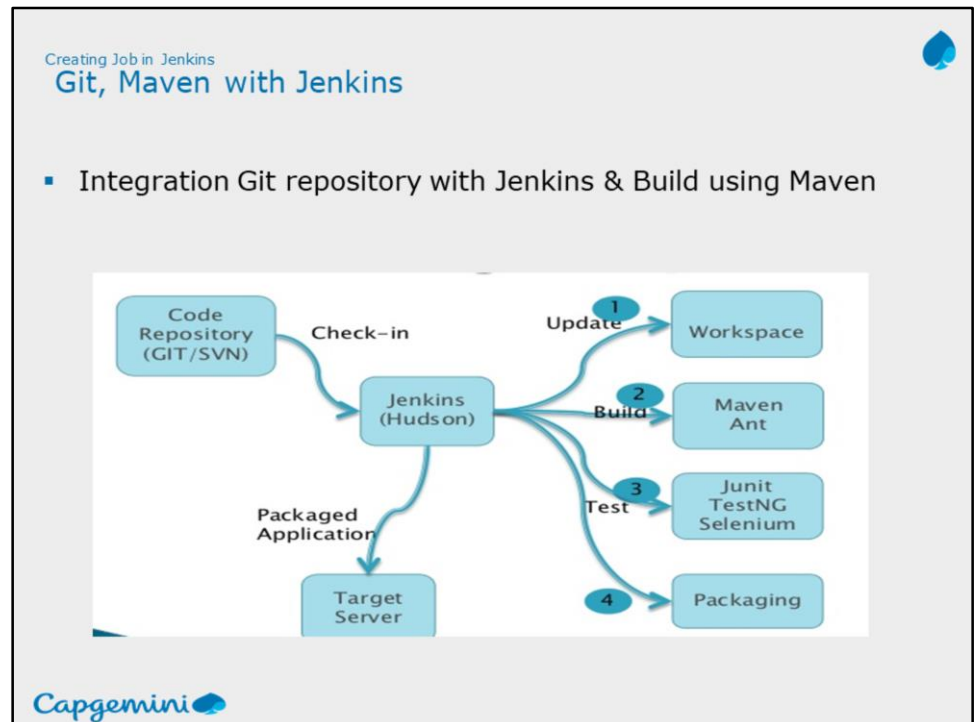
By default, Jenkins will run on the 8080 port. To specify the port manually, use the --httpPort option:

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

Instructor Notes:

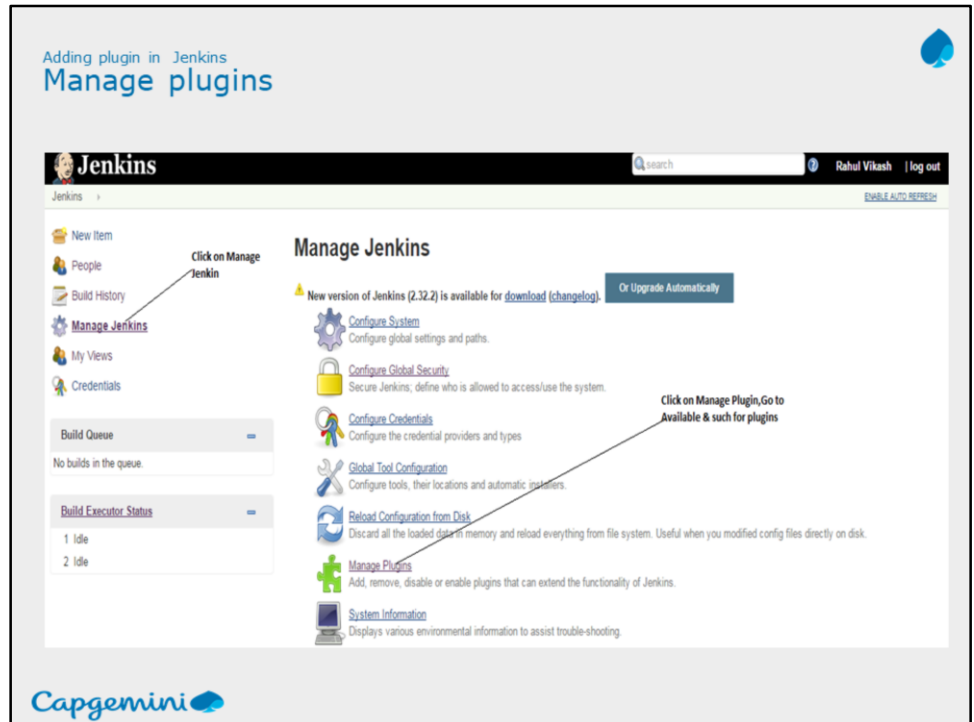
By default, Jenkins will run on the 8080 port. To specify the port manually, use the --httpPort option:

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


Instructor Notes:

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:

Adding plugin in Jenkins

Manage plugins

- Download Maven ,Git plugin

The screenshot shows the Jenkins 'Manage Plugins' interface. A list of plugins is displayed with their names, descriptions, and versions. A box highlights the 'Git plugin' and 'Maven Integration plugin'. The 'Git plugin' is currently at version 3.0.5, and the 'Maven Integration plugin' is at version 2.15.1. There are 'Downgrade' buttons for several plugins, including 'Git plugin' (downgrade to 3.0.4), 'GitHub Integration Plugin' (downgrade to 0.1.0-rc19), and 'GitHub plugin' (downgrade to 1.25.1).

Plugin Name	Description	Version	Action
Git client plugin	Shared library plugin for other Git related Jenkins plugins	2.2.1	
Git plugin	This plugin integrates Git with Jenkins.	3.0.5	Downgrade to 3.0.4
GitHub API Plugin	This plugin provides GitHub API for other plugins.	1.84	
GitHub Integration Plugin	Advanced trigger for GitHub Pull Requests and Branches.	0.1.0-rc20	Downgrade to 0.1.0-rc19
GitHub plugin	This plugin integrates GitHub to Jenkins.	1.25.0	Downgrade to 1.25.1
Javadoc Plugin	This plugin adds Javadoc support to Jenkins.	1.4	
JUnit Plugin	Allows JUnit-format test results to be published.	1.19	
Mailer Plugin	This plugin allows you to configure email notifications for build results. This is a break-out of the original core based email component.	1.19	
Matrix Project Plugin	Multi-configuration (matrix) project type.	1.8	
Maven Integration plugin	This plugin provides an advanced integration for Maven 2/3 projects.	2.15.1	

Capgemini

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:

Adding plugin in Jenkins

Manage plugins

■ Setting Configuration

■ Go to Manage Jenkin->Global Tools Configuration

JDK installations

JDK

Name

JDK1.8

JAVA_HOME

C:\Program Files\Java\jdk1.8.0_31

Install automatically

Put JDK Path

Add JDK

List of JDK installations on this system

Delete JDK

Git

Git installations

Git

Name

Default

Path to Git executable

C:\Program Files\Git\bin\git.exe

Install automatically

Use GIT.Exe Path

Add Git

Delete Git

Maven

Maven installations

Maven

Name

Maven3.2.5

MAVEN_HOME

D:\maven\apache-maven-3.2.5

Install automatically

Use Maven Path

Capgemini

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-20


Instructor Notes:


Creating Job with Maven & Git
Creating Maven Project


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

Enter an item name

* 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.

Capgemini

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:

Creating Job with Maven & Git
Creating Maven Project

- Integrating Git with Jenkins by giving repository url(GitHub URL)

Source Code Management

☐ None
☒ Git

Repositories

Repository URL:

Credentials:

Advanced...

Add Repository

Choose Source code management . Give the GIT Repository URL & Then press add, give user name & Password of Github repository

Branch Specifier (blank for 'any'):

Add Branch

Repository browser:

Additional Behaviours:

Build

Root POM:

Goals and options:

Advanced...

In Build give path where pom.xml is there

Capgemini

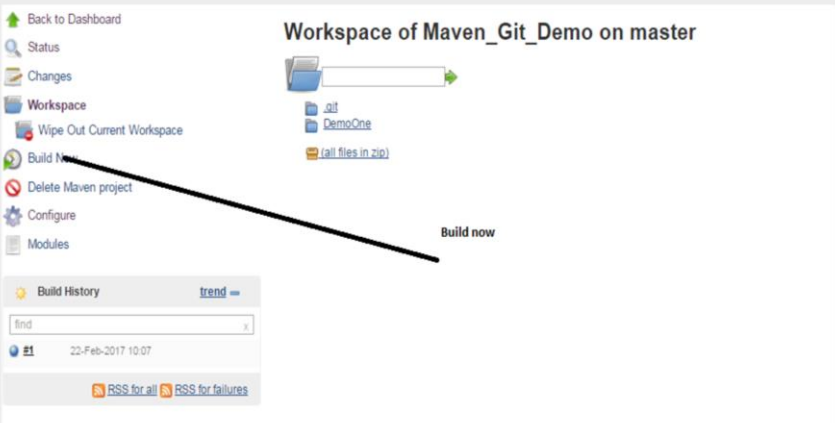
By default, Jenkins will run on the 8080 port. To specify the port manually, use the --httpPort option:

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

Instructor Notes:

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

Capgemini

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:

Creating Job with Maven & Git

Creating Maven Project

TESTS

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] --- maven-jar-plugin:2.4:jar (default-jar) @ DemoOne ---

[INFO] Building jar: C:\Users\rv830051\.jenkins\jobs\Maven_Git_Demo\workspace\DemoOne\target\DemoOne-1.0-SNAPSHOT.jar

[INFO] --- maven-install-plugin:2.4:install (default-install) @ DemoOne ---

[INFO] Installing C:\Users\rv830051\.jenkins\jobs\Maven_Git_Demo\workspace\DemoOne\target\DemoOne-1.0-SNAPSHOT.jar to C:\Users\rv830051\.m2\repository\com\cg\demoone\DemoOne\1.0-SNAPSHOT\DemoOne-1.0-SNAPSHOT.jar

[INFO] Installing C:\Users\rv830051\.jenkins\jobs\Maven_Git_Demo\workspace\DemoOne\pom.xml to C:\Users\rv830051\.m2\repository\com\cg\demoone\DemoOne\1.0-SNAPSHOT\DemoOne-1.0-SNAPSHOT.pom

[INFO] BUILD SUCCESS

[INFO] Total time: 19.984 s

[INFO] Finished at: 2017-02-22T10:07:57+05:30


[INFO] Final Memory: 14M/34M

[JENKINS] Archiving C:\Users\rv830051\.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\rv830051\.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




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:


Creating Job with Maven & Git



Creating Maven Project



- Check & get all feedback when it changes
- Build result obtained from the dashboard


All



S	W	Name ↓	Last Success	Last Failure	Last Duration
		Maven_Git_Demo	8 min 3 sec - #1	N/A	55 sec

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

[Legend](#) [RSS for all](#) [RSS for failures](#) [RSS for just latest builds](#)

Capgemini 

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:

Add instructor notes here.

Demo

- Demo on Maven-Git-Jenkins integration

Capgemini 

Add the notes here.

Instructor Notes:

Add instructor notes here.

Lab

- Lab 03

Capgemini

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

Capgemini 

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

Capgemini 

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



- _____ is the process of deploying the latest code into production.
 - Build job
 - Continuous Deployment
 - Continuous Testing
 - None of the above



Add the notes here.