GIT&Jenkin

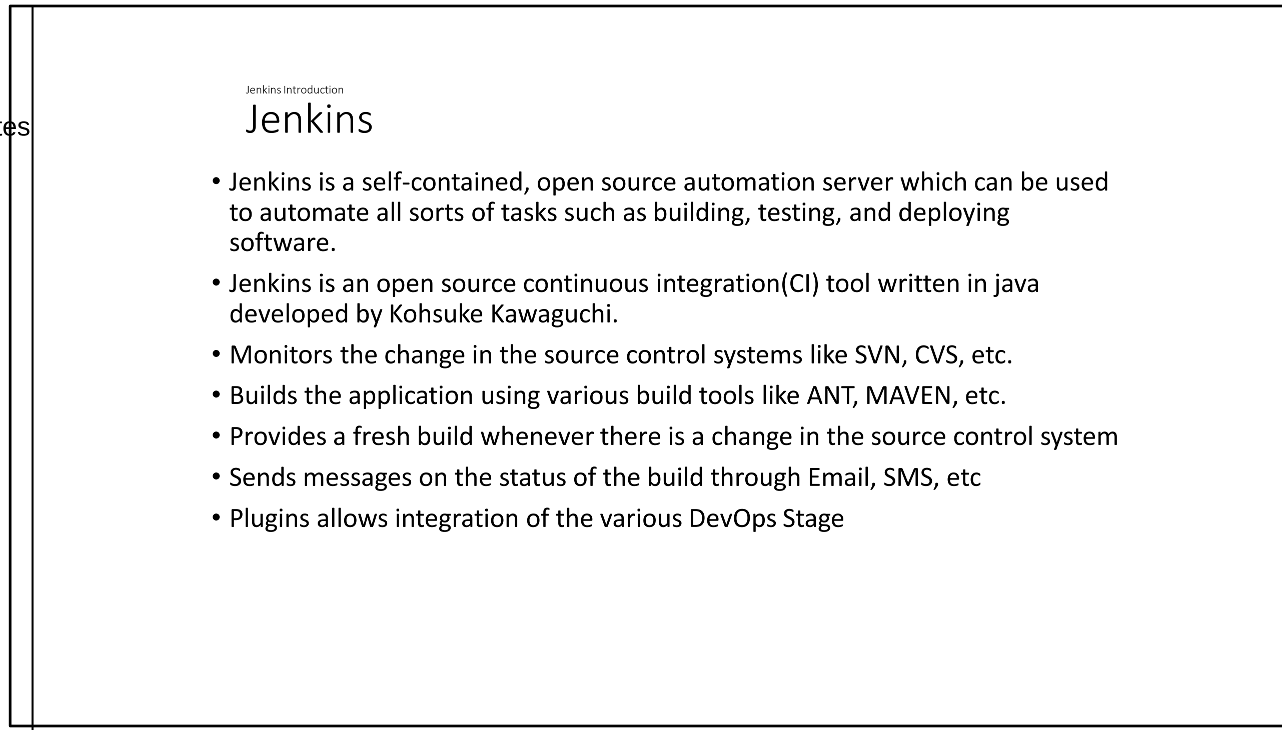
Lesson 06: Introduction to Jenkin

Capgemini

Add instructor notes here.

Lesson Objectives

- Jenkins Introduction
- Creating Configuring and Running Jenkins Jobs
- Adding plugin in Jenkins
- Creating Job with Maven & Git



Jenkins Introduction

Jenkins

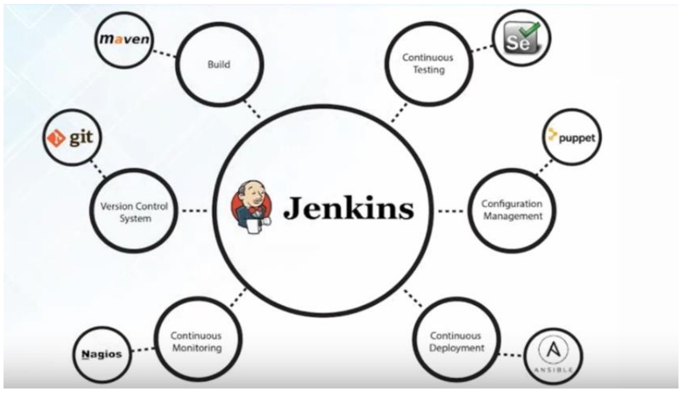
- Jenkins is a self-contained, open source automation server which can be used to automate all sorts of tasks such as building, testing, and deploying software.
- Jenkins is an open source continuous integration(CI) tool written in java developed by Kohsuke Kawaguchi.
- Monitors the change in the source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

Add instructor notes here.

What different models we can integrate with jenkin

Jenkins Introduction

Jenkins

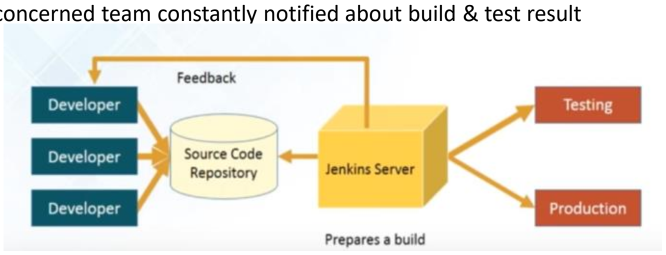


Add instructor notes here.

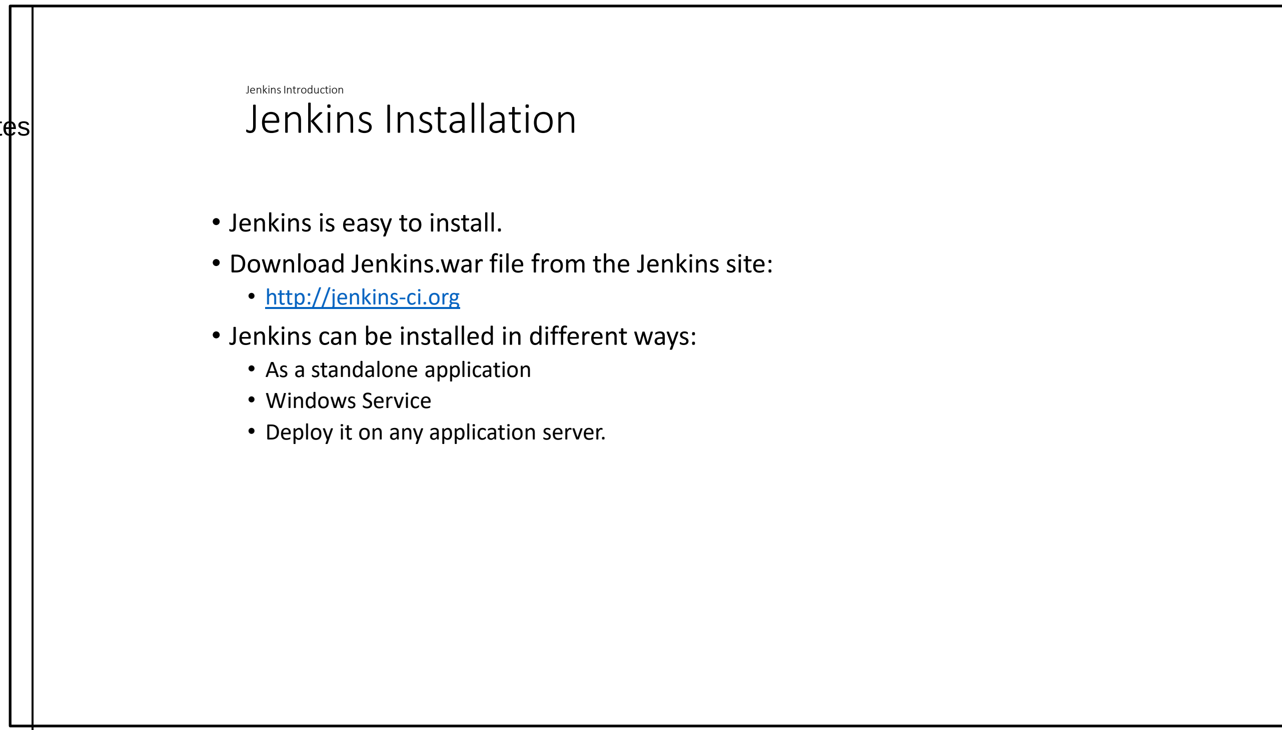
Jenkins Introduction

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



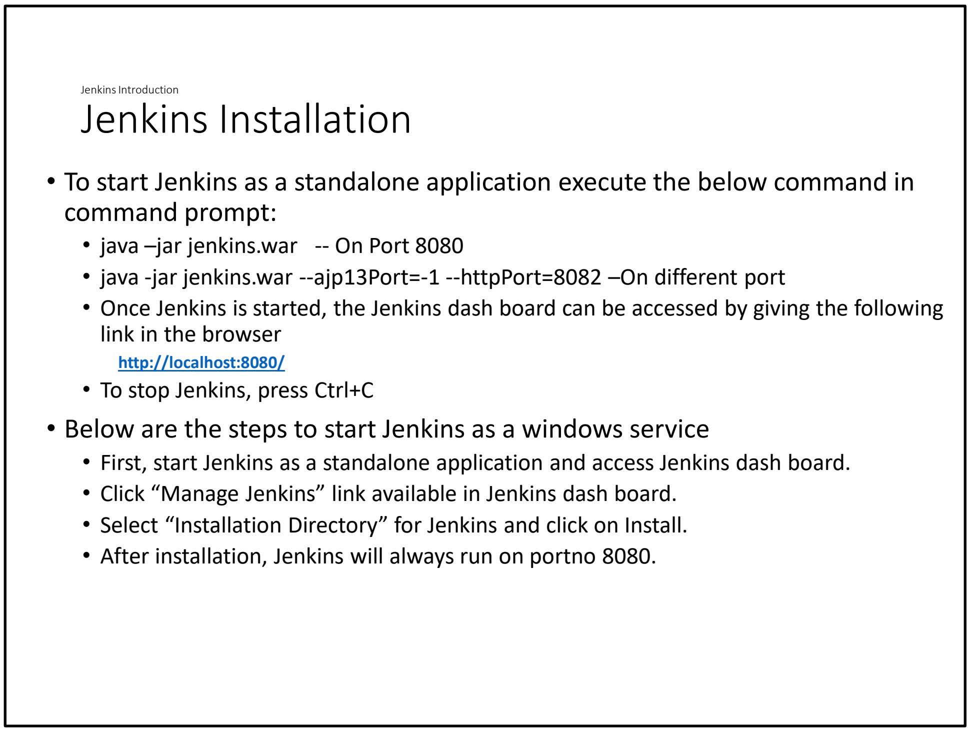
Add instructor notes here.



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.


Jenkins Introduction

Jenkins Installation

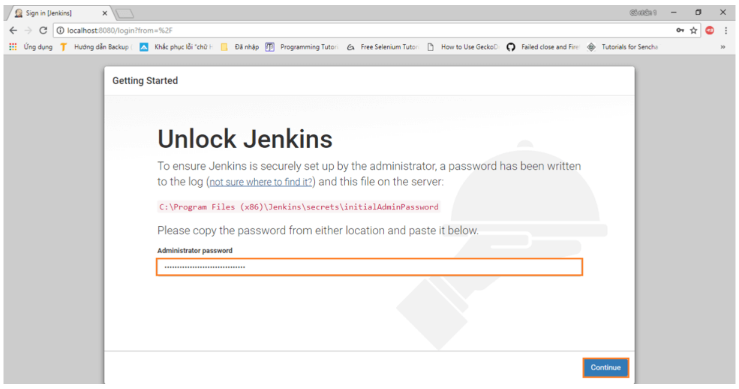
- To start Jenkins as a standalone application execute the below command in command prompt:
  - java –jar jenkins.war -- On Port 8080
  - java -jar jenkins.war --ajp13Port=-1 --httpPort=8082 –On different port
  - Once Jenkins is started, the Jenkins dash board can be accessed by giving the following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

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

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

Unlock Jenkins


Getting Started

Unlock Jenkins

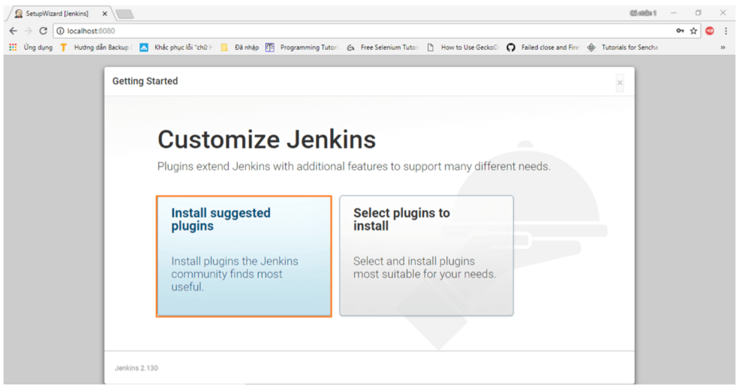
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

```
C:\Program Files (x86)\Jenkins\secrets\initialAdminPassword
```

Please copy the password from either location and paste it below.

Administrator password

Customize Jenkins


Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

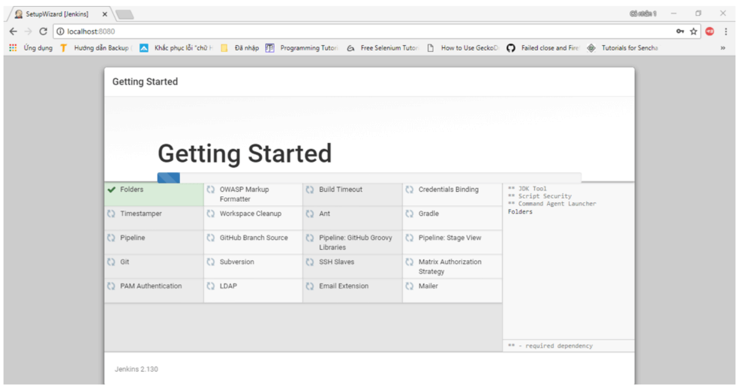
Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Jenkins 2.130

Install Plugins



SetupWizard [Jenkins]

localhost:8080

Getting Started

Getting Started

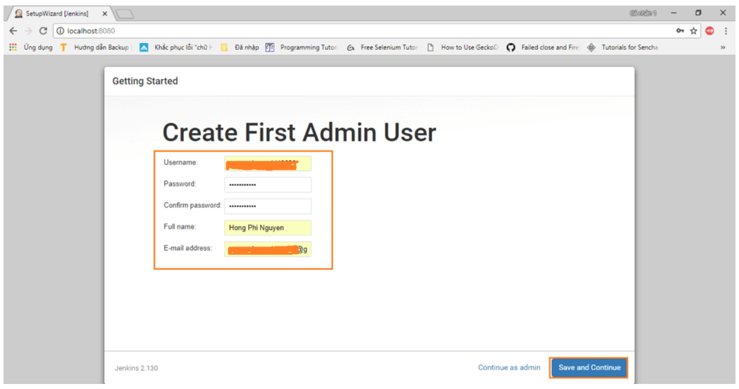
✓ Folders	OWASP Markup Formatter	Build Timeout	Credentials Binding
Timestamper	Workspace Cleanup	Ant	Gradle
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View
Git	Subversion	SSH Slaves	Matrix Authorization Strategy
PAM Authentication	LDAP	Email Extension	Mailer

```
** JDK Tool
** Script Security
** Command Agent Launcher
Folders
```

```
** - required dependency
```

Jenkins 2.130

Create Admin User



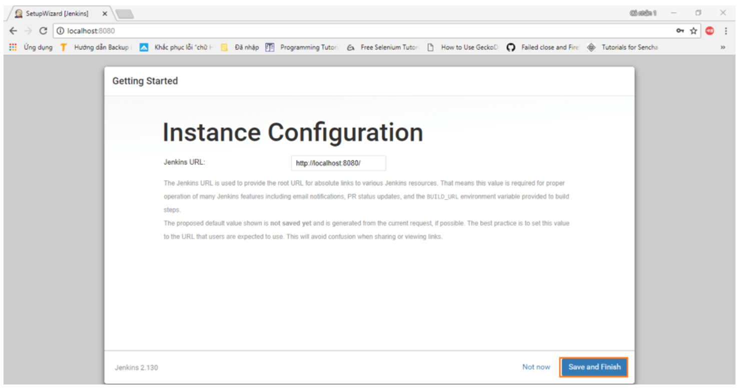
Getting Started

Create First Admin User

Username:	[redacted]
Password:	••••••••••••
Confirm password:	••••••••••••
Full name:	Hong Phi Nguyen
E-mail address:	[redacted]

Jenkins 2.130

Instance Configuration


Getting Started

Instance Configuration

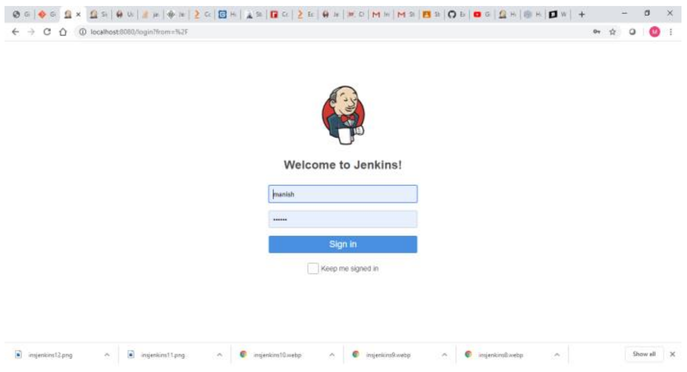
Jenkins URL: http://localhost:8080/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

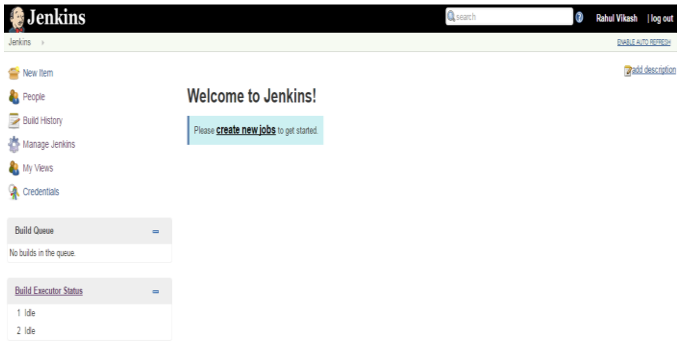
Jenkins 2.130

Login Screen



Creating Job in Jenkins

Jenkins Installation



Jenkins

search Rahul Vikash | log out

Jenkins ›

ENABLE AUTO REFRESH

add description

New Item
People
Build History
Manage Jenkins
My Views
Credentials

Welcome to Jenkins!

Please **create new jobs** to get started.

Build Queue

No builds in the queue.

Build Executor Status

1 Idle
2 Idle

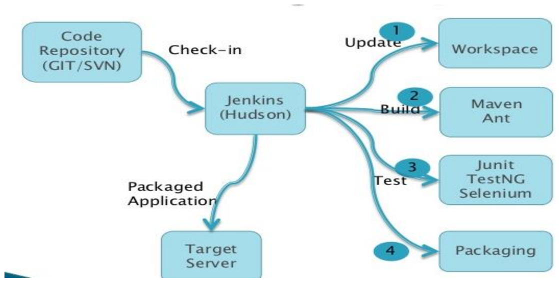
By default, Jenkins will run on the 8080 port. To specify the port manually, use the --httpPort option:

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

Creating Job in Jenkins

Git, Maven with Jenkins

- Integration Git repository with Jenkins & Build using Maven

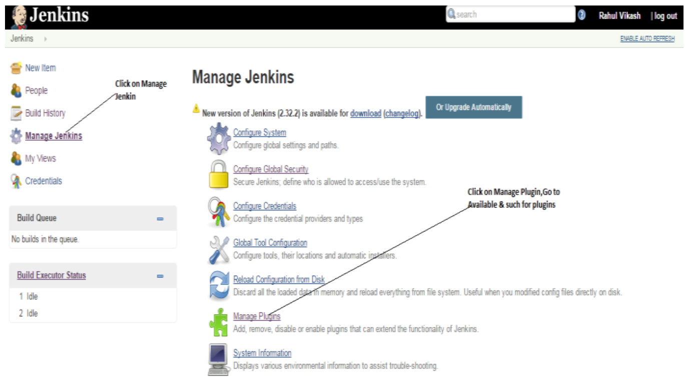


By default, Jenkins will run on the 8080 port. To specify the port manually, use the --httpPort option:

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

Adding plugin in Jenkins

Manage plugins



Manage Jenkins

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

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.

Build Queue

No builds in the queue.

Build Executor Status

1 Idle
2 Idle

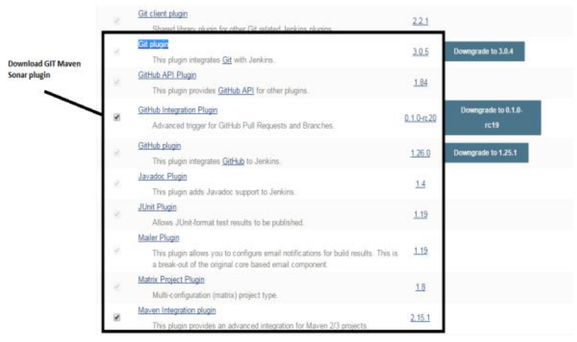
By default, Jenkins will run on the 8080 port. To specify the port manually, use the --httpPort option:

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

Adding plugin in Jenkins

Manage plugins

- Download Maven ,Git plugin



✓	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.	1.26.0	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	

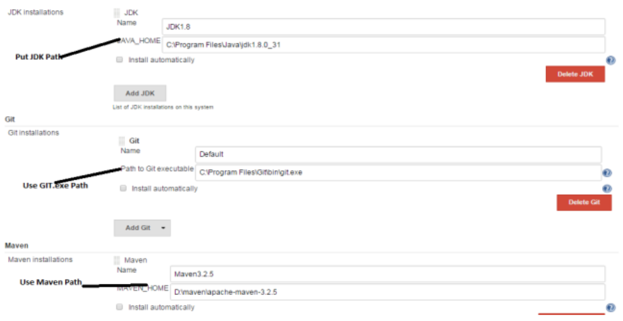
By default, Jenkins will run on the 8080 port. To specify the port manually, use the --httpPort option:

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

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
List of JDK installations on this system

Git
Git installations
Git
Name: Default
Path to Git executable: C:\Program Files\Git\bin\git.exe
Install automatically

Maven
Maven installations
Maven
Name: Maven3.2.5
MAVEN_HOME: D:\maven\apache-maven-3.2.5
Install automatically

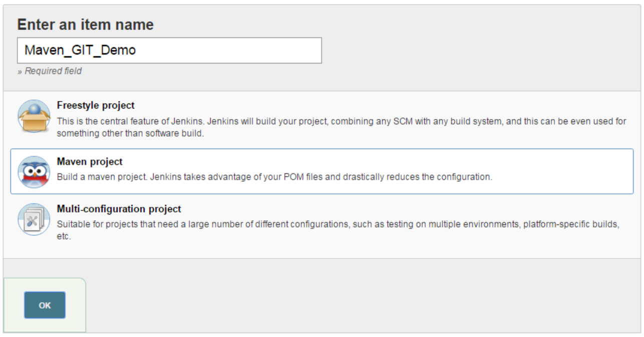
By default, Jenkins will run on the 8080 port. To specify the port manually, use the --httpPort option:

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

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.

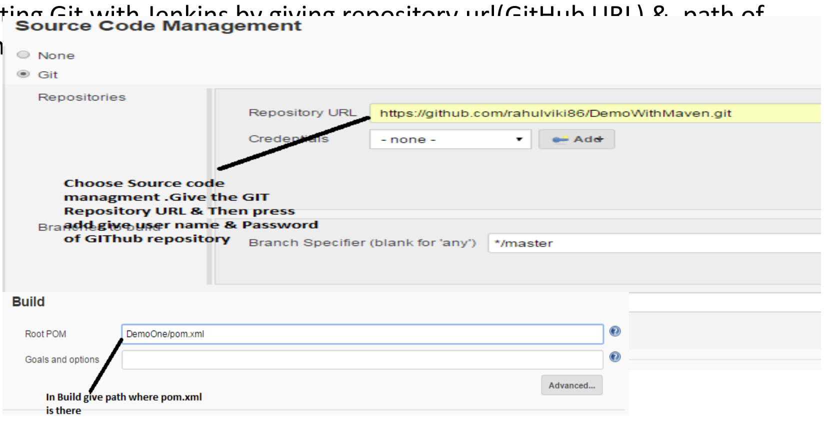
By default, Jenkins will run on the 8080 port. To specify the port manually, use the --httpPort option:

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

Creating Job with Maven & Git

Creating Maven Project

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



Source Code Management

None

Git

Repositories

Repository URL: https://github.com/rahulviki86/DemoWithMaven.git

Credentials: - none -

Add

Choose Source code managment .Give the GIT Repository URL & Then press add give user name & Password of GIThub repository

Branches to build

Branch Specifier (blank for 'any'): */master

Build

Root POM: DemoOne/pom.xml

Goals and options

Advanced...

In Build give path where pom.xml is there

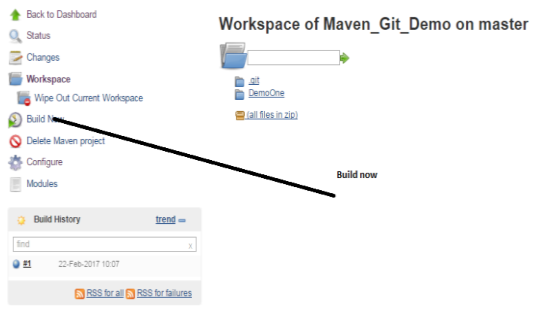
By default, Jenkins will run on the 8080 port. To specify the port manually, use the --httpPort option:

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

Creating Job with Maven & Git

Creating Maven Project

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

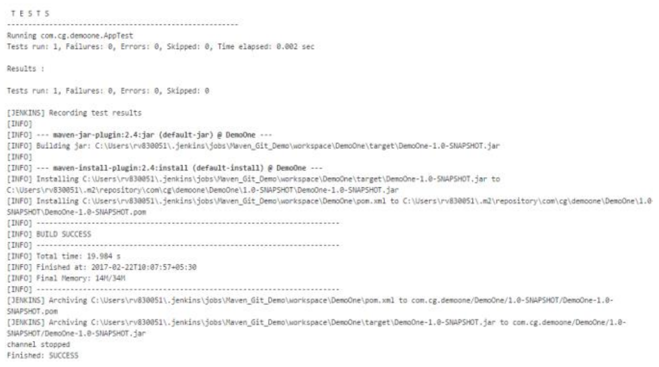


By default, Jenkins will run on the 8080 port. To specify the port manually, use the --httpPort option:

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

Creating Job with Maven & Git

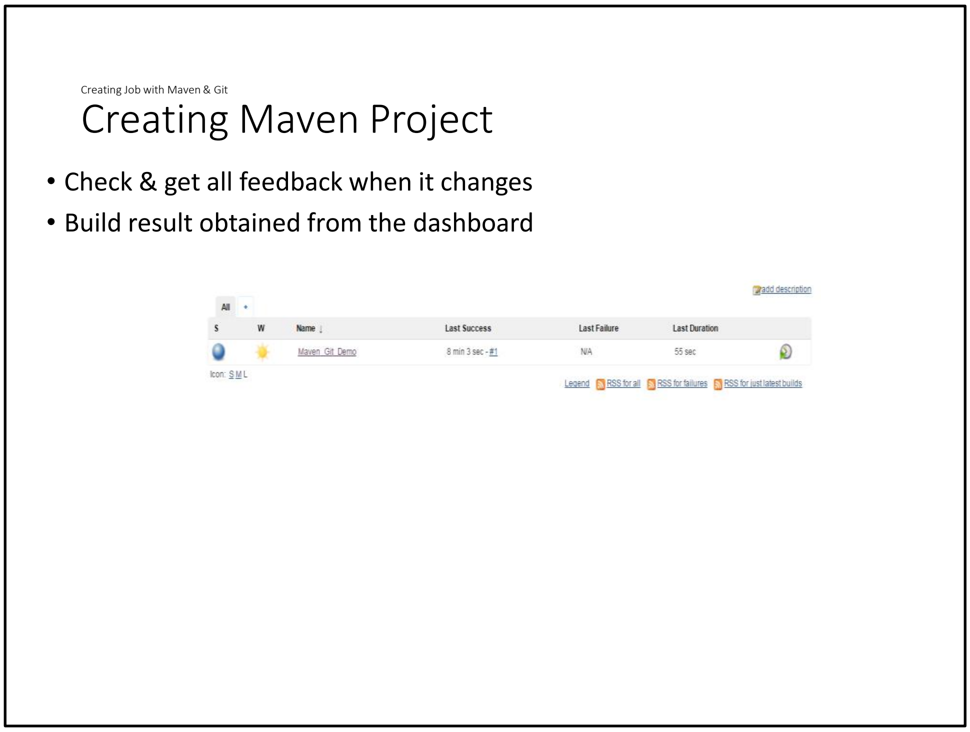
Creating Maven Project



```
 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\rv830051\.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\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] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 19.984 s
[INFO] Finished at: 2017-02-22T10:07:57+05:30
[INFO] Final Memory: 14M/34M
[INFO] ------------------------------------------------------------------------
[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
```



Creating Job with Maven & Git

Creating Maven Project

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

add description

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

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

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

Add instructor notes here.

Demo

- Demo on Maven-Git-Jenkins integration

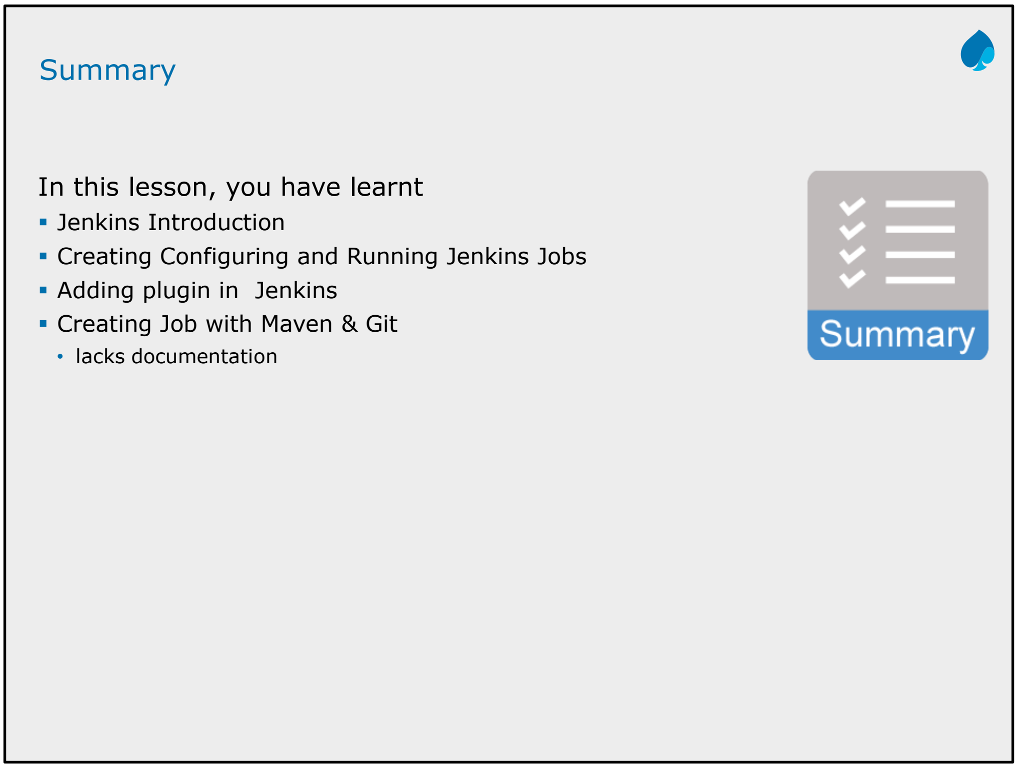
Add the notes here.

Add instructor notes here.

Lab

- Lab 03

Add the notes here.



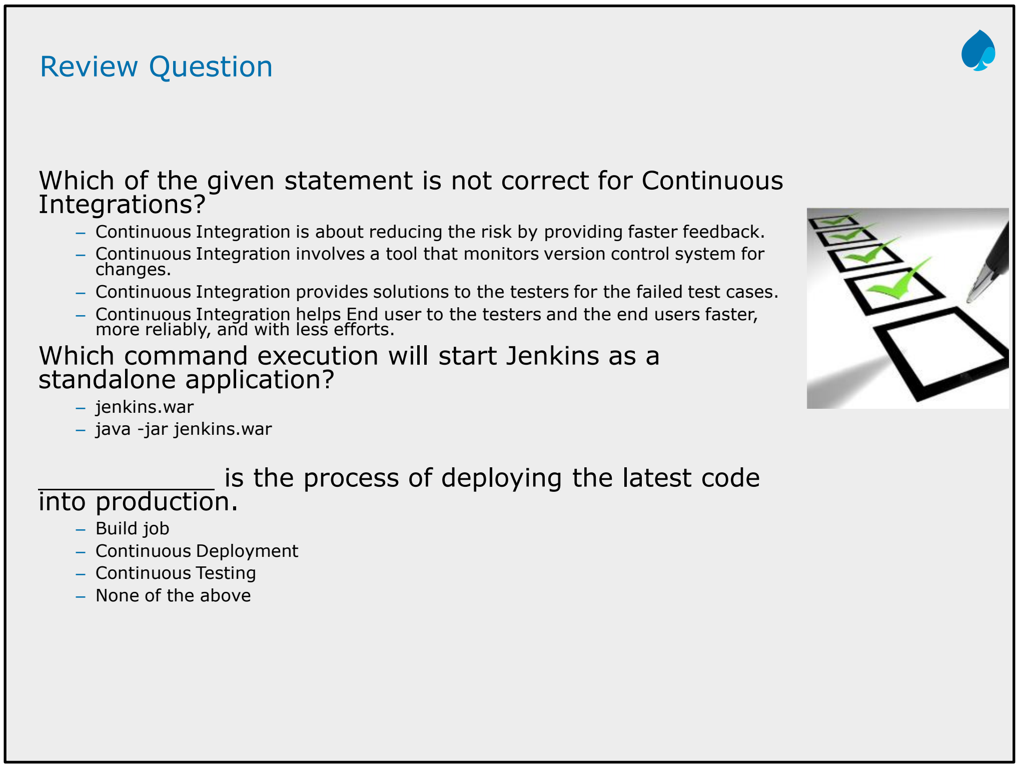
Summary

In this lesson, you have learnt

- Jenkins Introduction
- Creating Configuring and Running Jenkins Jobs
- Adding plugin in Jenkins
- Creating Job with Maven & Git
  - lacks documentation

Summary

Add the notes here.



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

____________ is the process of deploying the latest code into production.

- Build job
- Continuous Deployment
- Continuous Testing
- None of the above