

EXPERIMENT NO. 4

Date of Performance	
Date of Submission	

Aim : To understand Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job.

THEORY

DevOps practices reflect the idea of continuous improvement and automation. Many practices focus on one or more development cycle phases. These practices include:

- **Continuous development:**
This practice spans the planning and coding phases of the DevOps lifecycle. Version-control mechanisms might be involved.
- **Continuous testing:**
This practice incorporates automated, prescheduled, continued code tests as application code is being written or updated. Such tests can speed the delivery of code to production.
- **Continuous integration (CI):**
This practice brings configuration management (CM) tools together with other test and development tools to track how much of the code being developed is ready for production. It involves rapid feedback between testing and development to quickly identify and resolve code issues.
- **Continuous delivery:**
This practice automates the delivery of code changes, after testing, to a preproduction or staging environment. A staff member might then decide to promote such code changes into production.
- **Continuous deployment (CD):**
Similar to continuous delivery, this practice automates the release of new or changed code into production. A company doing continuous deployment might release code or feature changes several times per day. The use of container technologies, such as Docker and Kubernetes, can enable continuous deployment by helping to maintain consistency of the code across different deployment platforms and environments.
- **Continuous monitoring:**
This practice involves ongoing monitoring of both the code in operation and the underlying infrastructure that supports it. A feedback loop that reports on bugs or issues then makes its way back to development.
- **Infrastructure as code:**
This practice can be used during various DevOps phases to automate the provisioning of infrastructure required for a software release. Developers add infrastructure “code” from within their existing development tools. For example, developers might create a storage volume on demand from Docker, Kubernetes, or OpenShift. This practice also allows operations teams to monitor environment configurations, track changes, and simplify the rollback of configurations.

Jenkins is an open source automation server. With Jenkins, organizations can accelerate the software development process by automating it. Jenkins manages and controls software delivery processes throughout the entire lifecycle, including build, document, test, package, stage, deployment, static code analysis and much more.

You can set up Jenkins to watch for any code changes in places like GitHub, Bitbucket or GitLab and automatically do a build with tools like Maven and Gradle. You can utilize container technology such as Docker and Kubernetes, initiate tests and then take actions like rolling back or rolling forward in production.

OUTPUT

JENKINS INSTALLATION (Windows)

The simplest way to install Jenkins on Windows is to use the Jenkins Windows installer. That program will install Jenkins as a service using a 64-bit JVM chosen by the user. Keep in mind that to run Jenkins as a service, the account that runs Jenkins must have permission to log in as a service.

Steps:

1. Setup wizard
2. Select destination folder
3. Service logon credentials
4. Port selection
5. Select Java home directory
6. Custom setup
7. Install Jenkins
8. Finish Jenkins installation

Post-installation Setup Wizard

After downloading, installing and running Jenkins, the post-installation setup wizard begins. This setup wizard takes you through a few quick "one-off" steps to unlock Jenkins, customize it with plugins and create the first administrator user through which you can continue accessing Jenkins.

Unlocking Jenkins:

When you first access a new Jenkins instance, you are asked to unlock it using an automatically-generated password.

- Browse to <http://localhost:8080> (or whichever port you configured for Jenkins when installing it) and wait until the **Unlock Jenkins** page appears.
- The initial Administrator password should be found under the Jenkins installation path (set at Step 2 in Jenkins Installation).

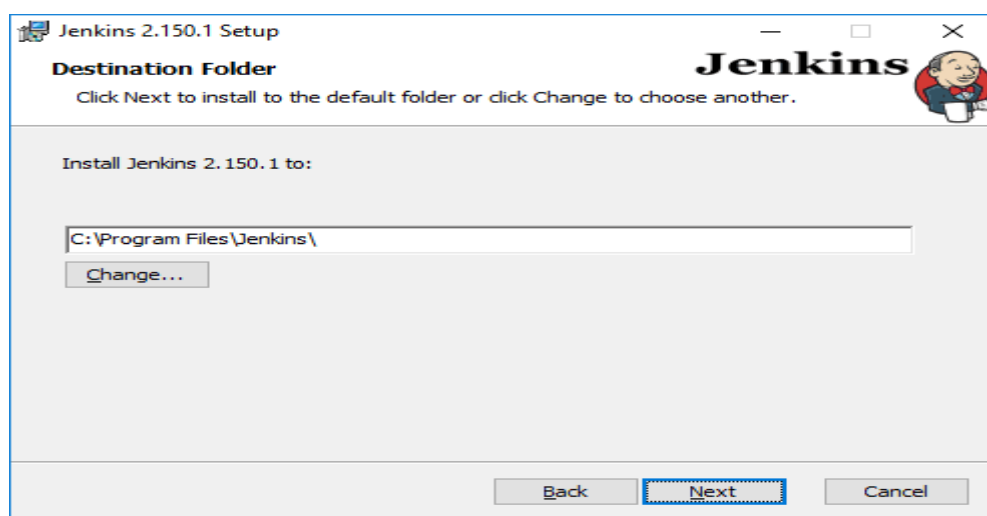
MAVEN INSTALLATION

Maven is a powerful project management and comprehension tool that provides a complete build life cycle framework to assist developers.

It is based on the concept of a **POM (Project Object Model)** that includes project information and configuration information for Maven such as:

- Construction directory
- Source directory
- Test source directory
- Dependency
- Goals
- Plugins

EXECUTION: JENKINS INSTALLATION AND SETUP



Jenkins 2.401.3 Setup

Service Logon Credentials

Enter service credentials for the service.

Jenkins 2.401.3 installs and runs as an independent Windows service. To operate in this manner, you must supply the user account credentials for Jenkins 2.401.3 to run successfully.


Logon Type:

☐ Run service as LocalSystem (not recommended)

☒ Run service as local or domain user:

Account:

Password:

 Credentials must be tested to continue


Jenkins 2.401.3 Setup

Port Selection

Choose a port for the service.

Please choose a port.

Port Number (1-65535):

 Click 'Test Port' button to proceed

It is recommended that you accept the selected default port.

Jenkins 2.401.3 Setup


Custom Setup

Select the way you want features to be installed.

Click the icons in the tree below to change the way features will be installed.

Jenkins

Start Service

 Firewall Exception

Enables a firewall exception for the Java running Jenkins on port 8080 (not recommended).

This feature requires 0KB on your hard drive.

Jenkins 2.401.3 Setup

Completed the Jenkins 2.401.3 Setup Wizard

Click the Finish button to exit the Setup Wizard.



EXECUTION- MAVEN INSTALLATION AND SEETINT SET VARIABLES AND PATH

maven.apache.org/download.cgi

Apache Maven Project
http://maven.apache.org/

Maven™

Download | Get Sources | Last Published: 2025-08-31

Downloading Apache Maven 3.9.11

Apache Maven 3.9.11 is the latest release. It is the recommended version for all users.

System Requirements

Java Development Kit (JDK)	Maven 3.9+ requires JDK 8 or above to execute. It still allows you to build against 1.3 and other JDK versions by using toolchains .
Memory	No minimum requirement
Disk	Approximately 10MB is required for the Maven installation itself. In addition to that, disk space will be used for your local Maven repository. The size of your local repository will vary depending on usage but expect at least 500MB.
Operating System	No minimum requirement. Start up scripts are included as shell scripts (tested on many Unix flavors) and Windows batch files.

Files

Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the [installation instructions](#). Use a source archive if you intend to build Maven yourself. In order to guard against corrupted downloads/installations, it is highly recommended to [verify the signature](#) of the release bundles against the public [KEYS](#) used by the Apache Maven developers.

	Link	Checksums	Signature
Binary tar.gz archive	apache-maven-3.9.11-bin.tar.gz	apache-maven-3.9.11-bin.tar.gz.sha512	apache-maven-3.9.11-bin.tar.gz.asc
Binary zip archive	apache-maven-3.9.11-bin.zip	apache-maven-3.9.11-bin.zip.sha512	apache-maven-3.9.11-bin.zip.asc
Source tar.gz archive	apache-maven-3.9.11-src.tar.gz	apache-maven-3.9.11-src.tar.gz.sha512	apache-maven-3.9.11-src.tar.gz.asc
Source zip archive	apache-maven-3.9.11-src.zip	apache-maven-3.9.11-src.zip.sha512	apache-maven-3.9.11-src.zip.asc

- 3.9.11 Release Notes and Release Reference Documentation
- latest source code from source repository
- Distributed under the Apache License, version 2.0
- other
 - All current release sources (plugins, shared libraries, ...) available at <https://downloads.apache.org/maven/>

Maven Daemon

Apache Maven Daemon (mmd) is available as a separate download. In order to guard against corrupted downloads/installations, it is highly recommended to [verify the signature](#) of the release bundles against the public [KEYS](#) used by the Apache Maven developers.

Platform	Format	Download Link (Checksum, Signature)
----------	--------	-------------------------------------

Environment Variables

User variables for alex

Variable	Value
ChocolateyLastPathUpdate	132786800071138935
GPU_FORCE_64BIT_PTR	0
GPU_MAX_ALLOC_PERCENT	100
GPU_MAX_HEAP_SIZE	100
GPU_SINGLE_ALLOC_PERCE...	100
GPU_USE_SYNC_OBJECTS	1
OneDrive	C:\Users\akova\OneDrive

New... Edit... Delete

System variables

Variable	Value
ChocolateyInstall	C:\ProgramData\chocolatey
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
JAVA_HOME	C:\Program Files (x86)\Java\jre1.8.0_321
MSMPL_BENCHMARKS	C:\Program Files\Microsoft MPI\Benchmarks\
MSMPL_BIN	C:\Program Files\Microsoft MPI\Bin\
NUMBER OF PROCESSORS	8

New... Edit... Delete

OK Cancel

Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	⚙ Credentials Binding	** Icons API
⚙ Timestamp	⚙ Workspace Cleanup	⚙ Ant	⚙ Gradle	** ASP API
⚙ Pipeline	⚙ GitHub Branch Source	⚙ Pipeline: GitHub Groovy Libraries	⚙ Pipeline Graph View	** Structs
⚙ Git	⚙ SSH Build Agents	⚙ Matrix Authorization Strategy	⚙ PAM Authentication	** Pipeline: Step API
⚙ LDAP	⚙ Email Extension	⚙ Mailer	⚙ Dark Theme	** Token Macro

Build Timeout

** bouncycastle API

** Credentials

** Plain Credentials

** Variant

** SSH Credentials

Jenkins

+ New Item

Build History

Build Queue

No builds in the queue.

Build Executor Status

0/2

All

+

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	Experiment 4	21 days #5	N/A	0.27 sec
✓	☀	P2	21 days #1	N/A	0.66 sec

Icon: S M L

localhost:8080/view/all/newjob

Verify that it's you

Jenkins / All / New Item

New Item

Enter an item name

Maven Project TE IT USER

Select an item type



Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.



Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.



Multi-configuration project

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



Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

OK

localhost:8080/manage/

Verify that it's you

Jenkins / Manage Jenkins

System Configuration



System

Configure global settings and paths.



Tools

Configure tools, their locations and automatic installers.



Plugins

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



Nodes

Add, remove, control and monitor the various nodes that Jenkins runs jobs on.



Clouds

Add, remove, and configure cloud instances to provision agents on demand.



Appearance

Configure the look and feel of Jenkins.

Security



Security

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



Credentials

Configure credentials



Credential Providers

Configure the credential providers and types



Users

Create/delete/modify users that can log in to this Jenkins.

Status Information



System Information

Displays various environmental information to assist trouble shooting.



System Log

System log captures output from java.util.logging output related to Jenkins.



Load Statistics

Check your resource utilization and see if you need more computers for your builds.



About Jenkins

See the version and license information.

Troubleshooting



Manage Old Data

Scrub configuration files to remove remnants from

Download progress

- 19

- Checking internet connectivity
- Checking update center connectivity
- Success

Downloaded Successfully. Will be activated during the next boot

Downloaded Successfully. Will be activated during the next boot

Downloaded Successfully. Will be activated during the next boot

Downloaded Successfully. Will be activated during the next boot

Downloaded Successfully. Will be activated during the next boot

Downloaded Successfully. Will be activated during the next boot

Downloaded Successfully. Will be activated during the next boot

Downloaded Successfully. Will be activated during the next boot

Downloaded Successfully. Will be activated during the next boot

Downloaded Successfully. Will be activated during the next boot

Downloaded Successfully. Will be activated during the next boot

Downloaded Successfully. Will be activated during the next boot

Downloaded Successfully. Will be activated during the next boot

Downloaded Successfully. Will be activated during the next boot


Downloaded Successfully. Will be activated during the next boot

Downloaded Successfully. Will be activated during the next boot

Downloaded Successfully. Will be activated during the next boot

Downloaded Successfully. Will be activated during the next boot

Downloaded Successfully. Will be activated during the next boot

 Success

→ [Go back to the top page](#)

CONCLUSION :- Successfully Installed and setup Jenkins 2.303.1 and integrated it with MAVEN 3.8.2 and created a Job.