

EXPERIMENT NO : 04

AIM : To understand continuous integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build job

THEORY :

MAVEN

Maven is a build automation tool used primarily for Java projects. Maven can also be used to build and manage projects written in C#, Ruby, Scala and other languages. The Maven project is hosted by the Apache Software Foundation, where it was formerly part of the Jakarta project.

ANT

Apache Ant is a Java library and command-line tool whose mission is to drive processes described in build files as targets and extensions points dependent upon each other. The main known usage of Ant is the build of Java applications. Ant supplies a number of built-in tasks allowing to compile, assemble, test and run Java applications. Ant can also be used effectively to build non-Java applications, for instance C or C++ applications.

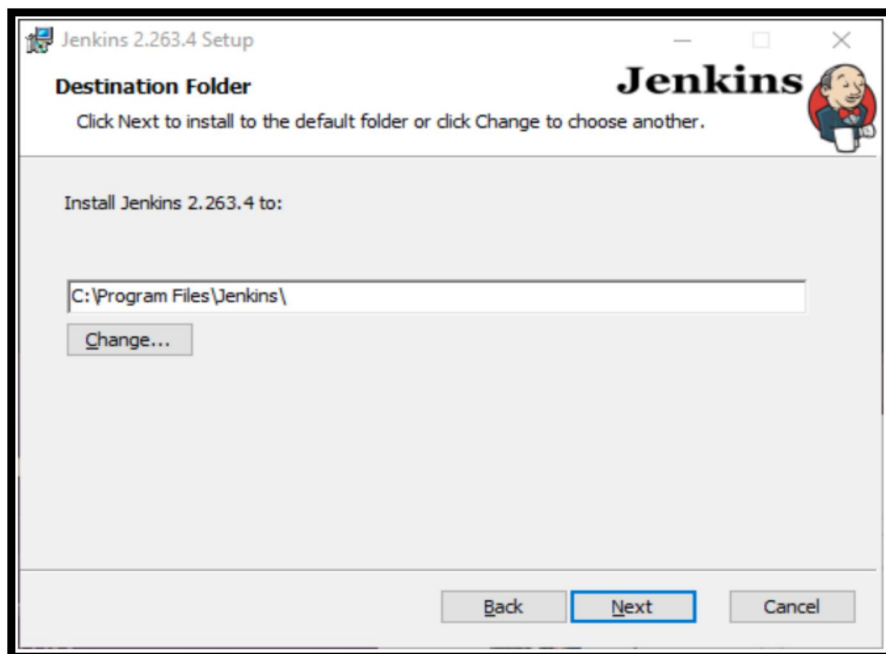
STEPS FOR INSTALLING AND SETUP OF JENKINS:

Installing Steps:

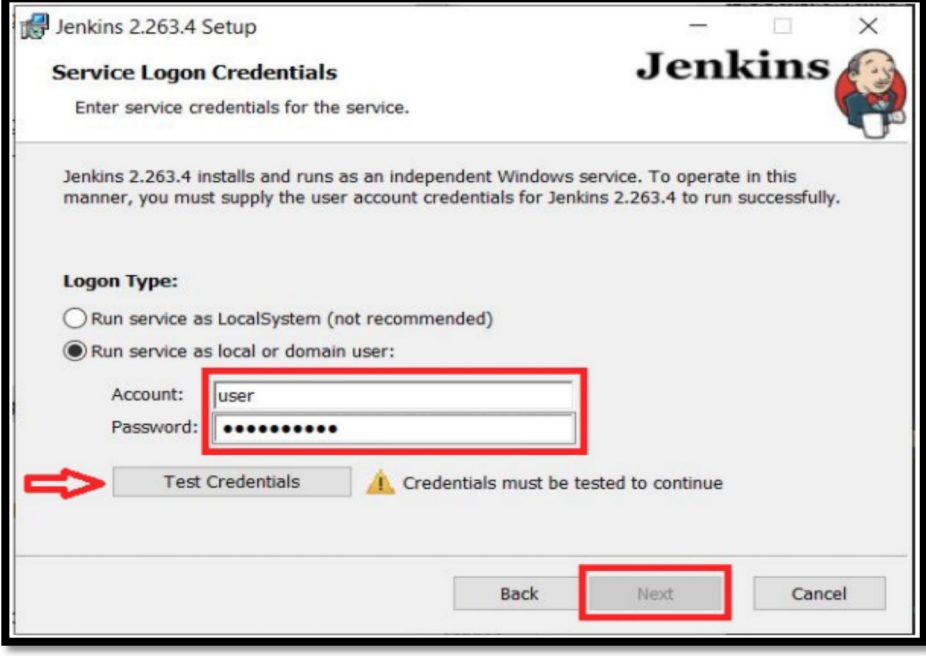
Step 1:



Step 2:



Step 3:



Jenkins 2.263.4 Setup

Service Logon Credentials

Enter service credentials for the service.

Jenkins 2.263.4 installs and runs as an independent Windows service. To operate in this manner, you must supply the user account credentials for Jenkins 2.263.4 to run successfully.



Logon Type:

☐ Run service as LocalSystem (not recommended)

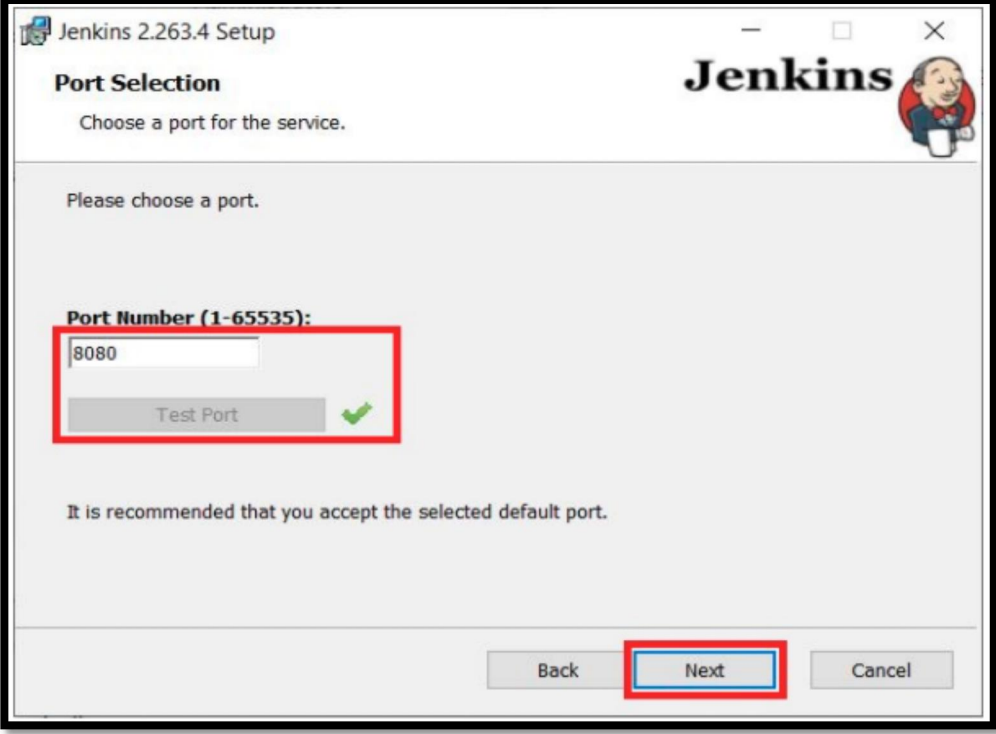
☒ Run service as local or domain user:

Account:

Password:

  Credentials must be tested to continue

Step 4:




Jenkins 2.263.4 Setup

Port Selection

Choose a port for the service.

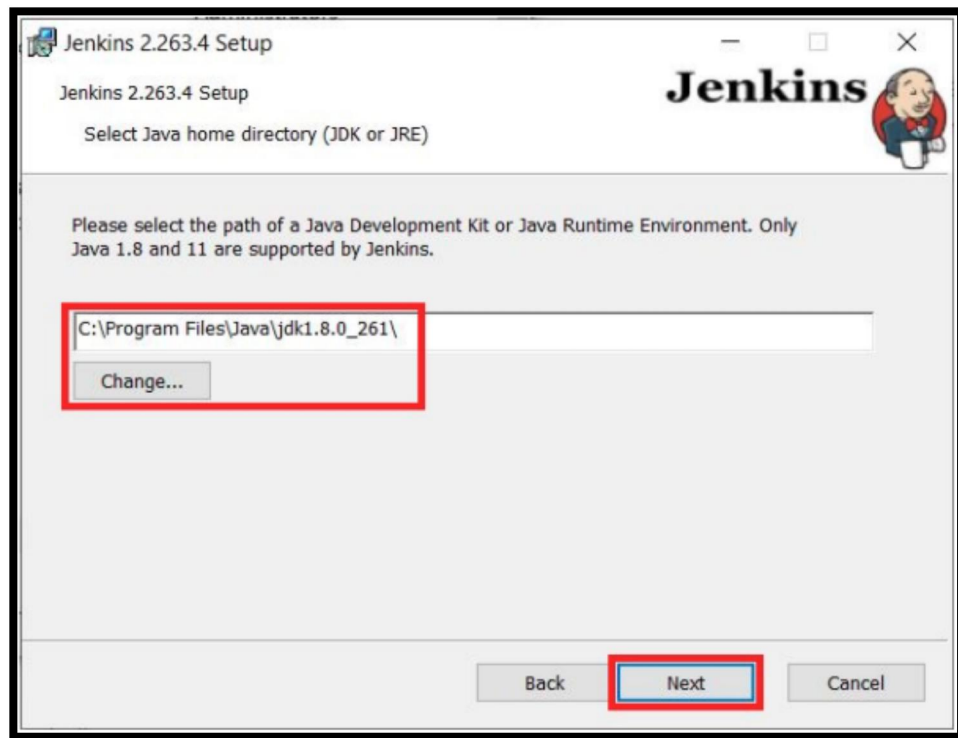
Please choose a port.

Port Number (1-65535):

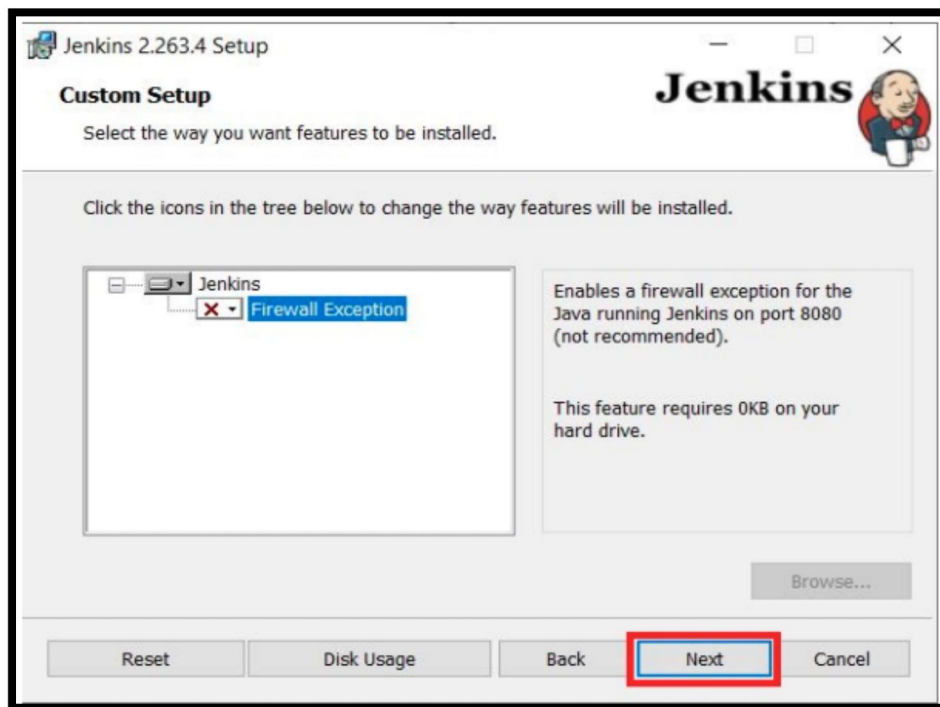


It is recommended that you accept the selected default port.

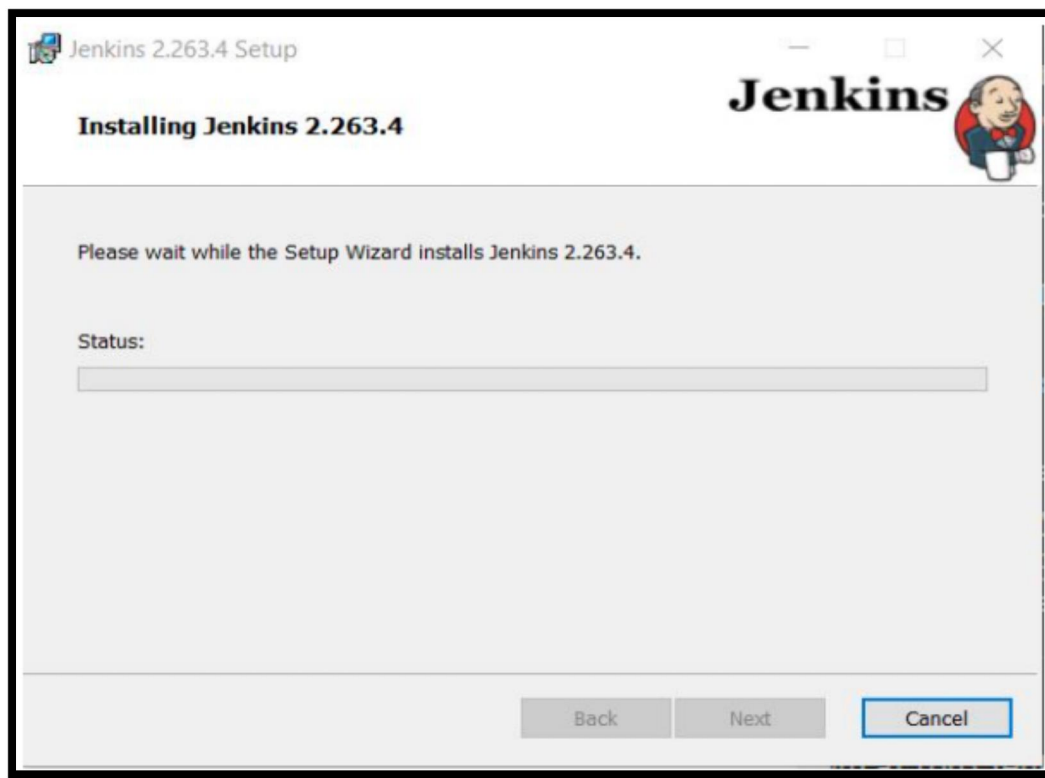
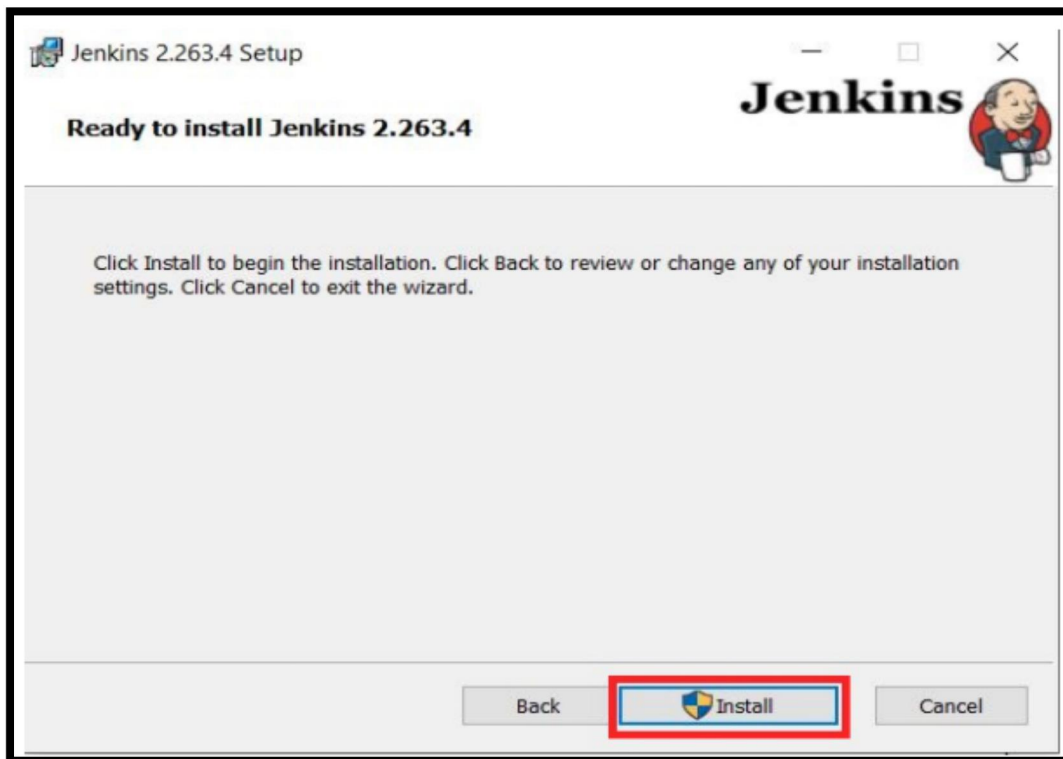
Step 5:



Step 6:



Step 7:



UNLOCKING JENKINS

Getting Started

Window Snip

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:\WINDOWS\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins\secrets\initialAdminPassword
```

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

Administrator password

Continue

CUSTOMIZE JENKINS WITH PLUGINS

Getting Started

Window Snip

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 2.303.1


Getting Started

Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding	** Pipeline: Multibranch
✓ Timestampers	✓ Workspace Cleanup	✓ Ant	✓ Gradle	** Authentication Tokens API
✓ Pipeline	✓ GitHub Branch Source	✓ Pipeline: GitHub Groovy Libraries	✓ Pipeline: Stage View	** Docker Commons
✓ Git	Subversion	SSH Slaves	Matrix Authorization Strategy	** Docker Pipeline
PAM Authentication	LDAP	Email Extension	✓ Mailer	** Pipeline: Stage Tags Metadata
				** Pipeline: Declarative Agent API
				** Pipeline: Declarative Lockable Resources
				Pipeline
				** GitHub API
				Git
				** GitHub
				GitHub Branch Source
				Pipeline: GitHub Groovy Libraries
				Pipeline: Stage View
				Git
				** Hap08 API
				Subversion
				** - required dependency

Jenkins 2.215

USER LOGIN:



Welcome to Jenkins!

Sign in

☐ Keep me signed in

MANAGE JENKINS:



The screenshot shows the Jenkins 'Manage Jenkins' dashboard. On the left is a sidebar with navigation links: New Item, People, Build History, Manage Jenkins (selected), My Views, Lockable Resources, and New View. Below these are sections for 'Build Queue' (showing 'No builds in the queue') and 'Build Executor Status' (showing two idle executors). The main content area is titled 'Manage Jenkins' and includes a notification for Jenkins 2.303.1. Below this is a warning about building on the controller node. The dashboard is divided into three main sections: 'System Configuration' with links for 'Configure System', 'Global Tool Configuration', 'Manage Plugins' (with an update alert), and 'Manage Nodes and Clouds'; and 'Security' with links for 'Configure Global Security', 'Manage Credentials', and 'Configure Credential Providers'.

Dashboard

- New Item
- People
- Build History
- Manage Jenkins**
- My Views
- Lockable Resources
- New View

Build Queue

No builds in the queue.

Build Executor Status

- 1 Idle
- 2 Idle

Manage Jenkins

New version of Jenkins (2.303.1) is available for [download](#) ([changelog](#)). [Or Upgrade Automatically](#)

Building on the controller node can be a security issue. You should set up distributed builds. See [the documentation](#). [Set up agent](#) [Set up cloud](#) [Dismiss](#)

System Configuration

- Configure System**
Configure global settings and paths.
- Global Tool Configuration**
Configure tools, their locations and automatic installers.
- Manage Plugins**
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
⚠ There are updates available
- Manage Nodes and Clouds**
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Security

- Configure Global Security**
Secure Jenkins: define who is allowed to access/use the system.
- Manage Credentials**
Configure credentials
- Configure Credential Providers**
Configure the credential providers and types

ADD JDK:



The screenshot shows the 'Add JDK' form in Jenkins. It has a title 'JDK' and a section 'JDK installations' with an 'Add JDK' button. Below is a table with one entry for 'JDK'. The 'Name' field contains 'JAVA_HOME'. The 'JAVA_HOME' field contains the path 'C:\Program Files\Java\jdk1.8.0_191'. There is an unchecked checkbox for 'Install automatically' and a 'Delete JDK' button. At the bottom, there is another 'Add JDK' button and a note 'List of JDK installations on this system'.

JDK

JDK installations

[Add JDK](#)

JDK
Name JAVA_HOME
JAVA_HOME C:\Program Files\Java\jdk1.8.0_191
<input type="checkbox"/> Install automatically
Delete JDK

[Add JDK](#)

List of JDK installations on this system

ADD GIT:

Git


Git installations




Git

Name

Path to Git executable 

 There's no such file: C:\Program Files\Git\bin\git.exe

☐ Install automatically 

Add Git ▾


Delete Git

ADD GRADLE:


Gradle

Gradle installations

Add Gradle



Gradle

name 

☒ Install automatically 



Install from Gradle.org

Version

Add Installer ▾

Delete Installer

Delete Gradle

ADD ANT:

Ant

Ant installations

Add Ant

Ant

Name

Ant

☒ Install automatically ?

Install from Apache

Version

1.10.11

Add Installer

Delete Installer

Delete Ant

ADD MAVEN:

Maven installations

Add Maven

Maven

Name

Maven

☒ Install automatically ?

Install from Apache

Version

3.8.2

Add Installer

Delete Installer

Delete Maven

Add Maven

Save


Apply

Activate Windows
Go to Settings to activate Windows.


A NEW BUILD JOB IN JENKINS:

Enter an item name


» This field cannot be empty, please enter a valid name




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.




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.



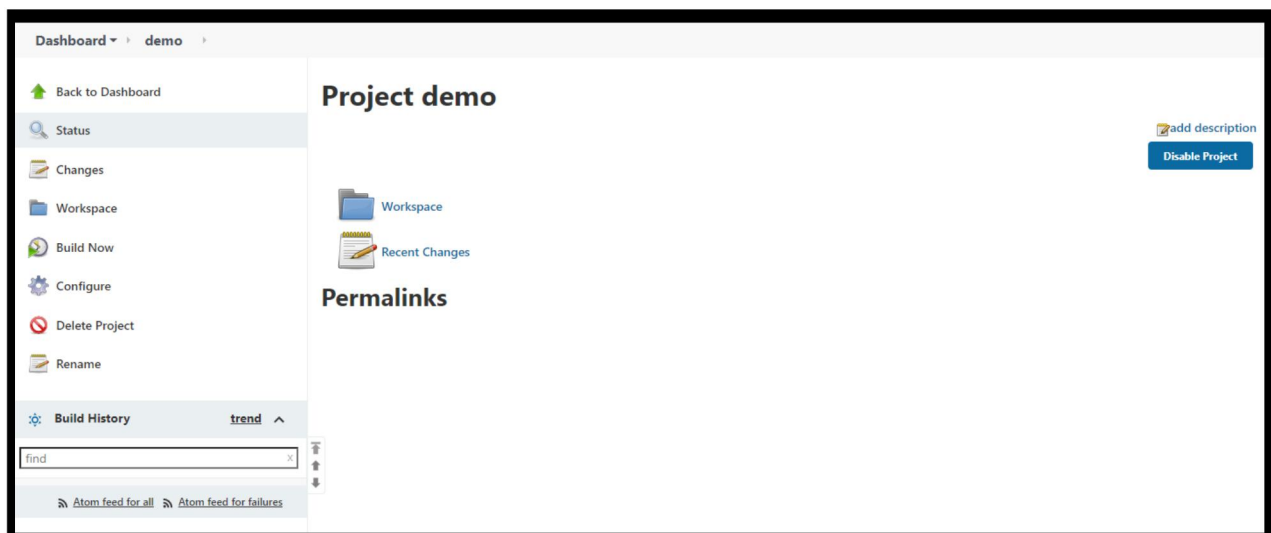
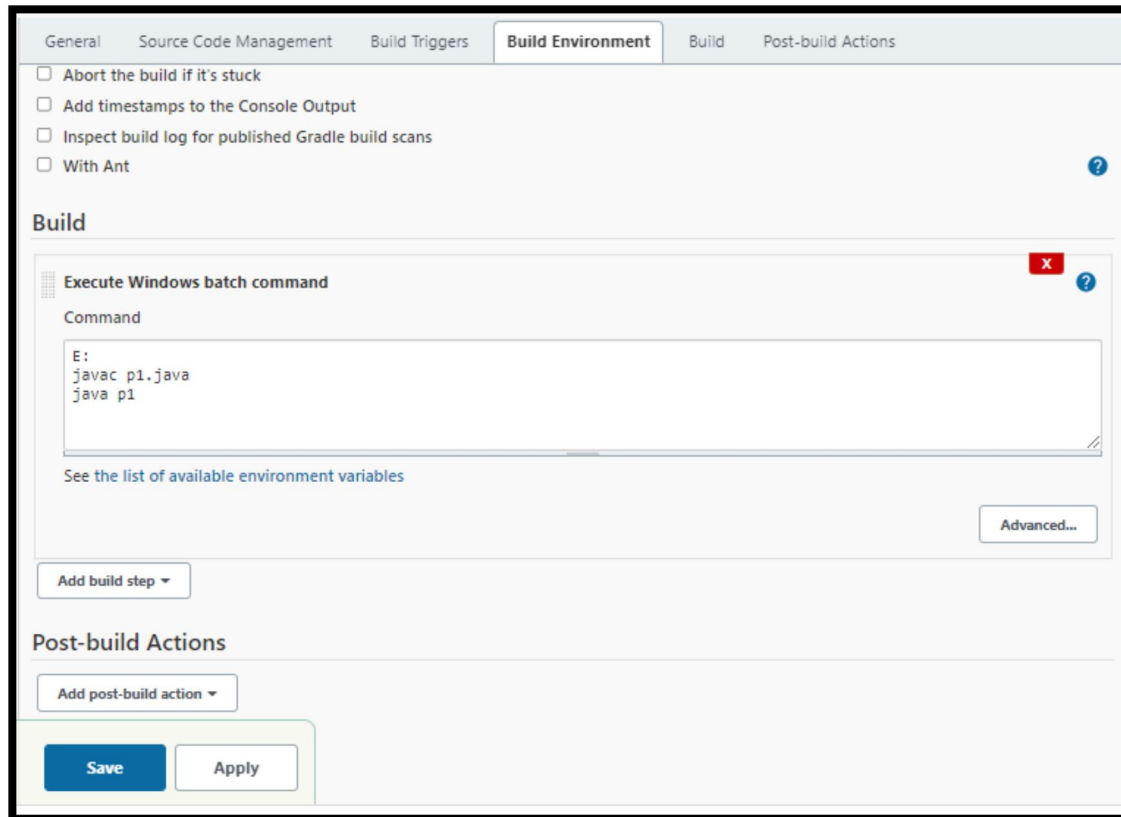
GitHub Organization
Creates a GitHub organization (or user account) for all repositories matching some defined markers.

GeneralSource Code ManagementBuild TriggersBuild EnvironmentBuildPost-build Actions

Build Triggers
☐ Trigger builds remotely (e.g., from scripts) ?
☐ Build after other projects are built ?
☐ Build periodically ?
☐ GitHub hook trigger for GITScm polling ?
☐ Poll SCM ?

Build Environment
☐ Delete workspace before build starts
☐ Use secret text(s) or file(s) ?
☐ Abort the build if it's stuck
☐ Add timestamps to the Console Output
☐ Inspect build log for published Gradle build scans
☐ With Ant ?

Build
Add build step ^
Execute Python script
Execute Windows batch command
Execute shell
Invoke Ant
Invoke Gradle script
Invoke top-level Maven targets
Run with timeout
Set build status to "pending" on GitHub commit



Console Output

```
Running as SYSTEM
Building in workspace C:\Windows\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins\workspace\demo5
[demo5] $ cmd /c call C:\Windows\TEMP\jenkins303012551348066408.bat

C:\Windows\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins\workspace\demo5>E:

E:\>javac p1.java

E:\>java p1
Hello World

E:\>exit 0
Finished: SUCCESS
```

CONCLUSION : Hence, we understood Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job.