

PROGRAM-1

Introduction to Maven and Gradle: Overview of Build Automation Tools, Key Differences Between Maven and Gradle, Installation and Setup

STEP 1 : Install JDK and Eclipse

- a) Update the local package repository
- Sudo apt update

```
student@student1: $ sudo apt update
[sudo] password for student:
Hit:1 https://packages.microsoft.com/repos/code stable InRelease
Hit:2 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu bionic InRelease
Hit:5 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:6 http://archive.ubuntu.com/ubuntu jammy-security InRelease
Hit:7 https://repositories.intel.com/graphcis/ubuntu focal InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 packages can be upgraded. Run 'apt list --upgradable' to see them.
W: http://in.archive.ubuntu.com/ubuntu/dists/bionic/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details
M: Target Packages (main/binary-amd64/Packages) is configured multiple times in /etc/apt/sources.list:3 and /etc/apt/sources.list:8
M: Target Packages (main/binary-i386/Packages) is configured multiple times in /etc/apt/sources.list:3 and /etc/apt/sources.list:8
M: Target Packages (main/binary-all/Packages) is configured multiple times in /etc/apt/sources.list:3 and /etc/apt/sources.list:8
M: Target Translations (main/i18n/Translation-en_IN) is configured multiple times in /etc/apt/sources.list:3 and /etc/apt/sources.list:8
M: Target Translations (main/i18n/Translation-en) is configured multiple times in /etc/apt/sources.list:3 and /etc/apt/sources.list:8
M: Target DEP-11 (main/dep11/Components-amd64.yml) is configured multiple times in /etc/apt/sources.list:3 and /etc/apt/sources.list:8
M: Target DEP-11 (main/dep11/Components-all.yml) is configured multiple times in /etc/apt/sources.list:3 and /etc/apt/sources.list:8
M: Target DEP-11 Icons (main/dep11/icons-48x48.tar) is configured multiple times in /etc/apt/sources.list:3 and /etc/apt/sources.list:8
M: Target DEP-11 Icons (main/dep11/icons-64x64.tar) is configured multiple times in /etc/apt/sources.list:3 and /etc/apt/sources.list:8
M: Target DEP-11 Icons-HDPI (main/dep11/icons-64x64@2.tar) is configured multiple times in /etc/apt/sources.list:3 and /etc/apt/sources.list:8
M: Target CNF (main/cnf/Commands-and64) is configured multiple times in /etc/apt/sources.list:3 and /etc/apt/sources.list:8
M: Target CNF (main/cnf/Commands-all) is configured multiple times in /etc/apt/sources.list:3 and /etc/apt/sources.list:8
M: Target Packages (universe/binary-amd64/Packages) is configured multiple times in /etc/apt/sources.list:3 and /etc/apt/sources.list:8
M: Target Packages (universe/binary-i386/Packages) is configured multiple times in /etc/apt/sources.list:3 and /etc/apt/sources.list:8
M: Target Packages (universe/binary-all/Packages) is configured multiple times in /etc/apt/sources.list:3 and /etc/apt/sources.list:8
M: Target Translations (universe/i18n/Translation-en_IN) is configured multiple times in /etc/apt/sources.list:3 and /etc/apt/sources.list:8
M: Target Translations (universe/i18n/Translation-en) is configured multiple times in /etc/apt/sources.list:3 and /etc/apt/sources.list:8
M: Target DEP-11 (universe/dep11/Components-amd64.yml) is configured multiple times in /etc/apt/sources.list:3 and /etc/apt/sources.list:8
M: Target DEP-11 (universe/dep11/Components-all.yml) is configured multiple times in /etc/apt/sources.list:3 and /etc/apt/sources.list:8
M: Target DEP-11 Icons (universe/dep11/icons-48x48.tar) is configured multiple times in /etc/apt/sources.list:3 and /etc/apt/sources.list:8
M: Target DEP-11 Icons (universe/dep11/icons-64x64.tar) is configured multiple times in /etc/apt/sources.list:3 and /etc/apt/sources.list:8
```

- b) Install JDK 17 un ubuntu

Sudo apt install openjdk_17_jdk

```
student@student1: $ sudo apt install openjdk-17-jdk
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  gyp libcc-ares2 libljs-events libljs-highlight.js libljs-inherits libljs-ts-typedarray libljs-psl libljs-source-map libljs-sprintf-js
  libljs-typedarray-to-buffer libljs-dev libnode-dev libnode-abbrev node-ansi-regex node-ansi-styles node-ansi-types
  node-are-we-there-yet node-arrify node-asap node-asyncnlkt node-balanced-match node-brace-expansion node-chownr node-clean-yaml-object
  node-color-convert node-color-name node-commander node-core-util-is node-decompress-response node-delayed-stream node-delegates node-dep
  node-diff node-encoding node-end-of-stream node-err-code node-escape-string-regexp node-fancy-log node-foreground-child node-fs.realpath
  node-function-blnd node-get-stream node-glob node-growl node-has-flav node-has-unicode node-hosted-git-lt INFO node-iconv-lite node-iferr
  node-imurmurhash node-indent-string node-inflight node-inherits node-int node-tp node-ip-reger node-is-buffer node-is-plain-obj node-ts-typedarray
  node-isarray node-isexe node-json-parse-better-errors node-jsonparse node-kind-of node-lodash-packages node-lowercase-keys node-lru-cache
  node-micro-response node-minimatch node-minimatch node-nlpass node-npm-stream node-negotiateode-pnp-bundled node-once node-osenv
  node-pkg-absolute-path node-pkg-absolute-path node-pkg-absolute-path node-pkg-absolute-path node-pkg-absolute-path node-pkg-absolute-path node-pump
  node-quic-lru node-read node-readable-stream node-resolve node-retry node-safe-buffer node-set-blocking node-signal-exit node-slash
  node-slice-ansi node-source-map node-spdx-correct node-spdx-exceptions node-spdx-expression-parse node-spdx-license-ids node-sprintf-js
  node-stealthy-require node-string-decoder node-supports-color node-text-table node-time-stamp node-tmatch node-typedarray-to-buffer
  node-universallyf node-util-deprecate node-validate-npm-package-license node-webidl-conversions node-whatwg-fetch node-wrappy node-yallist
  nodejs-dot
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  openjdk-17-jdk-headless openjdk-17-jre openjdk-17-jre-headless
Suggested packages:
  openjdk-17-demo openjdk-17-source visualvm fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei | fonts-wqy-zenhei
The following NEW packages will be installed:
  openjdk-17-jdk openjdk-17-jdk-headless openjdk-17-jre openjdk-17-jre-headless
0 upgraded, 4 newly installed, 0 to remove and 31 not upgraded.
Need to get 72.8 MB/121 MB of archives.
After this operation, 274 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 openjdk-17-jdk-headless amd64 17.0.14+7-1-22.04.1 [71.3 MB]
Get:2 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 openjdk-17-jdk amd64 17.0.14+7-1-22.04.1 [1,521 kB]
Fetched 72.8 MB in 14s (5,296 kB/s)
Selecting previously unselected package openjdk-17-jre-headless:amd64.
(Reading database ... 283345 files and directories currently installed.)
```

c) Install eclipse in ubuntu

Sudo apt install eclipse

```
student@student1: $ sudo apt install eclipse
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  gyp libc-ares2 libjs-highlight.js libjs-inherits libjs-is-typedarray libjs-psl libjs-source-map libjs-sprintf-js
  libjs-typedarray-to-buffer libnode-dev libnode72 libssl-dev libuv1-dev node-abbrev node-ansi-regex node-ansi-styles node-ansi-styles
  node-are-we-there-yet node-arrifit node-asean node-aworkit node-balanced-match node-brace-expansion node-chownr node-clean-yaml-object
```

STEP 2 : Maven installation

a) Install maven from offcial ubuntu repository

Sudo apt install maven -y

b) To check current maven version and to verify the installation

mvn -version

```
student@student1: $ mvn -version
Apache Maven 3.6.3
Maven home: /usr/share/maven
Java version: 17.0.14, vendor: Ubuntu, runtime: /usr/lib/jvm/java-17-openjdk-amd64
Default locale: en_IN, platform encoding: UTF-8
OS name: "linux", version: "6.8.0-52-generic", arch: "amd64", family: "unix"
```

STEP 3 : Gradle installation

a) Go to google and type “green cloud how to install gradle in ubuntu”

Then copy the following commands

```
wget -c https://services.gradle.org/distributions/gradle-7.4.2-bin.zip -P /tmp
```

```
student@student1: $ # wget -c https://services.gradle.org/distributions/gradle-7.4.2-bin.zip -P /tmp
student@student1: $ wget -c https://services.gradle.org/distributions/gradle-7.4.2-bin.zip -P /tmp
--2025-03-05 10:16:15-- https://services.gradle.org/distributions/gradle-7.4.2-bin.zip
Resolving services.gradle.org (services.gradle.org)... 104.16.73.101, 104.16.72.101, 2606:4700:9a91:6599:2ddd:d2d:6810:4865
Connecting to services.gradle.org (services.gradle.org)|104.16.73.101|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://github.com/gradle/gradle-distributions/releases/download/v7.4.2/gradle-7.4.2-bin.zip [following]
--2025-03-05 10:16:15-- https://github.com/gradle/gradle-distributions/releases/download/v7.4.2/gradle-7.4.2-bin.zip
Resolving github.com (github.com)... 20.207.73.82
Connecting to github.com (github.com)|20.207.73.82|:443... connected.
HTTP request sent, awaiting response... 302 Found
```

b) Once download completed extract zip file in the opt gradle directory

Sudo apt install unzip

```
unzip -d /opt/gradle /tmp/gradle-*.zip
```

```
student@student: $ sudo apt install unzip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
unzip is already the newest version (6.0-26ubuntu3.2).
unzip set to manually installed.
The following packages were automatically installed and are no longer required:
  gyp libc-ares2 libjs-events libjs-highlight.js libjs-inherits libjs-is-typedarray libjs-psl libjs-source-map libjs-sprintf.js
  libjs-typedarray-to-buffer libnode-dev libnode72 libssl-dev libuv1-dev node-abbrev node-ansi-regex node-ansi-styles node-clean-yaml-object
  node-are-we-there-yet node-arrify node-asap node-asynckit node-balanced-match node-brace-expansion node-chownr node-clean-yaml-object
  node-color-convert node-color-name node-commander node-core-util-is node-decompress-response node-delayed-stream node-delegates node-depd
  node-diff node-encoding node-end-of-stream node-err-code node-escape-string-regexp node-fancy-log node-foreground-child node-fs.realpath
  node-function-bind node-get-stream node-glob node-growl node-has-flag node-has-unicode node-hosted-git-info node-iconv-lite node-iferr
  node-imurmurhash node-indent-string node-inflight node-inherits node-int node-ip node-ip-regex node-is-buffer node-isplain-obj node-is-typedarray
  node-isarray node-isexe node-parse-better-errors node-jsonparse node-kind-of node-lodash-packages node-lowercase-keys node-tru-cache
  node-mimic-response node-minimatch node-minipass node-nute-stream node-negotiator node-npm-bundled node-once node-osenv
  node-p-cancelable node-p-map node-path-is-absolute node-process-nextick-args node-promise-inflight node-promise-retry node-promzard node-pump
  node-quick-lru node-read node-readable-stream node-resolve node-retry node-safe-buffer node-set-blocking node-signal-exit node-slash
  node-slice-ansi node-source-map node-spdx-correct node-spdx-exceptions node-spdx-expression-parse node-spdx-license-ids node-sprintf.js
  node-stealthy-require node-string-decoder node-supports-color node-text-table node-time-stamp node-tmatch node-typedarray-to-buffer
  node-universalify node-util-deprecate node-validate-npm-package-license node-webidl-conversions node-whatwg-fetch node-wrappy node-yallist
  nodejs-doc
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 31 not upgraded.
student@student: $ sudo unzip -d /opt/gradle /tmp/gradle-*.zip
Archive: /tmp/gradle-7.4.2-bin.zip
  creating: /opt/gradle/gradle-7.4.2/
  inflating: /opt/gradle/gradle-7.4.2/LICENSE
  inflating: /opt/gradle/gradle-7.4.2/NOTICE
  inflating: /opt/gradle/gradle-7.4.2/README
  creating: /opt/gradle/gradle-7.4.2/init.d/
  inflating: /opt/gradle/gradle-7.4.2/init.d/readme.txt
  creating: /opt/gradle/gradle-7.4.2/bin/
  inflating: /opt/gradle/gradle-7.4.2/bin/gradle
  inflating: /opt/gradle/gradle-7.4.2/bin/gradle.bat
```

- c) Now again list down the unzipped file by executing the ls command

ls/opt/gradle

- d) Set up environmental variable

sudo nano/etc/profile.d/gradle.sh

a new window will open then type

export PATH=/opt/gradle/gradle-7.4.2/bin:\$PATH

then save and exit the file. This script will be sourced at shell startup

- e) Make the script executable from the following command

chmod +x /etc/profile.d/gradle.sh

- f) Load the environmental variable using source code

source /etc/profile.d/gradle.sh

- g) To verify the gradle installation

Gradle -v

```
student@student: $ ls /opt/gradle
gradle-7.4.2
student@student: $ sudo nano /etc/profile.d/gradle.sh
student@student: $ chmod +x /etc/profile.d/gradle.sh
student@student: $ source /etc/profile.d/gradle.sh
student@student: $ gradle -v
Welcome to Gradle 7.4.2!
Here are the highlights of this release:
- Aggregated test and Jacoco reports
- Marking additional test source directories as tests in IntelliJ
- Support for Adoptium JDKs in Java toolchains
For more details see https://docs.gradle.org/7.4.2/release-notes.html

-----
Gradle 7.4.2
-----
Build time: 2022-03-31 15:25:29 UTC
Revision: 540473b8118064efcc264694cbcaa4b677f61041
Kotlin: 1.5.31
Groovy: 3.0.9
Ant: Apache Ant(TM) version 1.10.11 compiled on July 10 2021
JVM: 17.0.14 (Ubuntu 17.0.14+7-Ubuntu-122.04.1)
OS: Linux 6.8.0-52-generic amd64
```

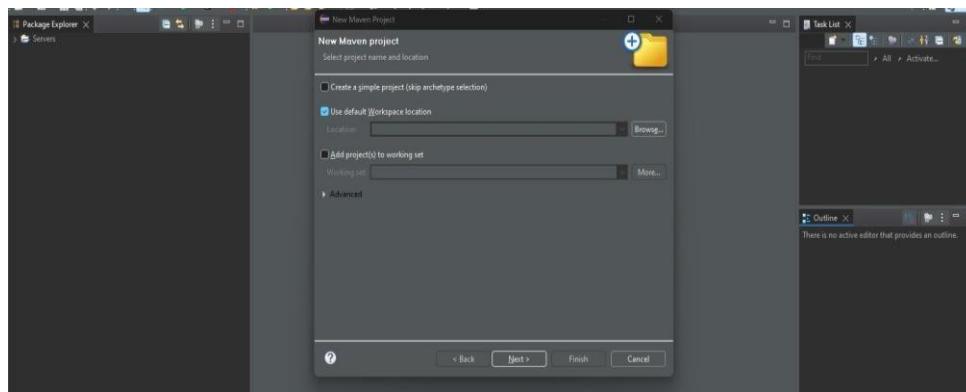
PROGRAM-2

Working with Maven: Creating a Maven Project, Understanding the POM File, Dependency Management and Plugins

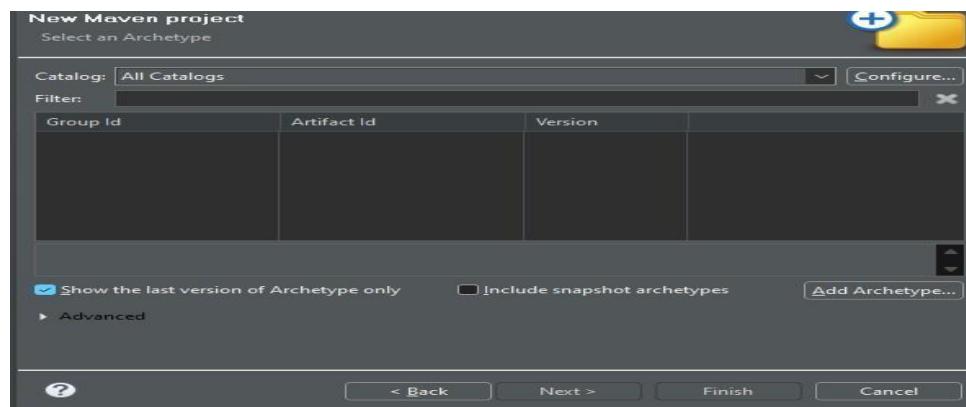
STEP 1: Open Eclipse then follow this navigation

File -----> New -----> Maven Project

After that Screen be as in below

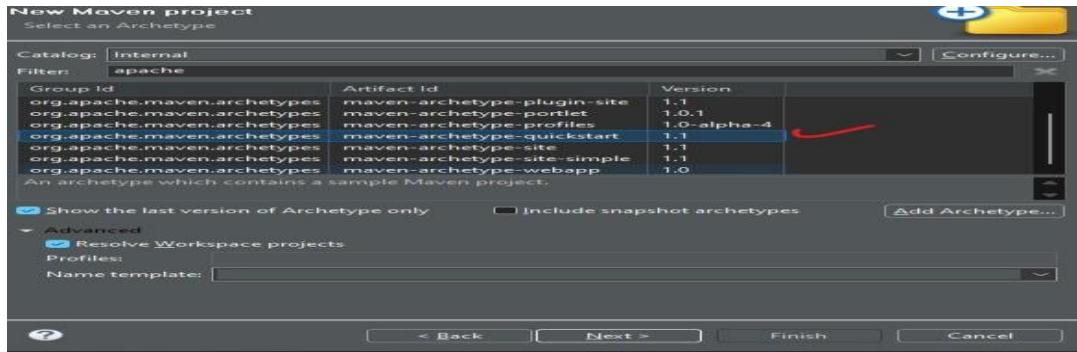


STEP 2: Make sure Use default Workspace Location is selected, then click Next The screen be as in below



STEP 3: In Screen shown above, click near the entry place of Filter and type "Apache" or select catalog as Internal

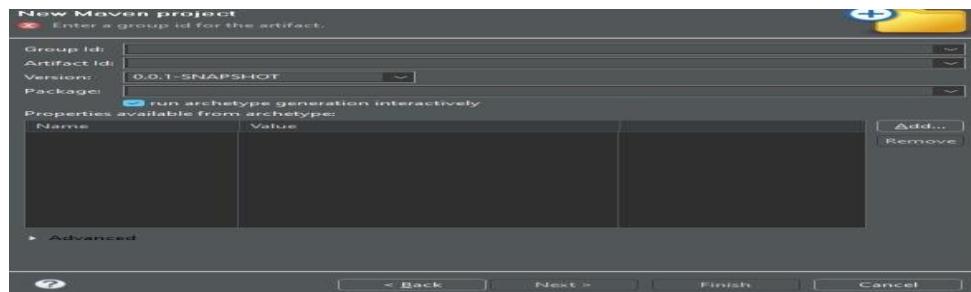
We want a simple maven JAR based application. So, we will choose the “mavenarchetype-quickstart” artifact to create the project.



STEP4: Enter Group Id:com.program2.maven

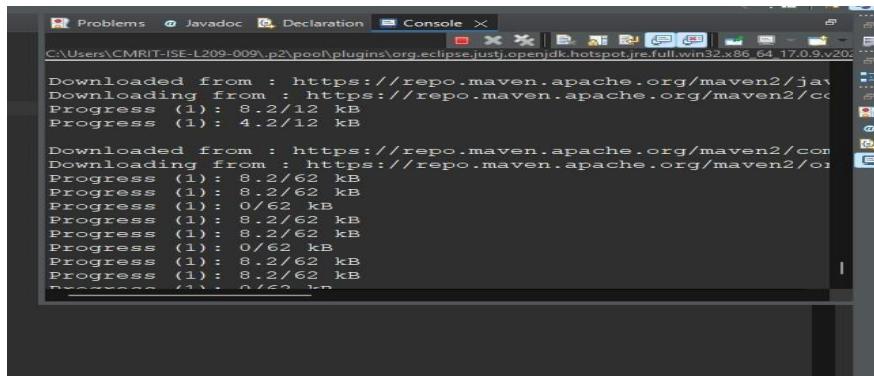
Artifact Id:program2-example-jar Keep snapshot as it is

Package:com.program2.maven.program2



After entering above mentioned details click on Finish

You be able to see the automation build happening for Maven Jar Project



It asks for Configuration confirmation just click Y

The Result be as in below

```

package: com.program2.maven.program2
Y: Y
[INFO] -----
[INFO] Using following parameters for creating project from Old (1.x) Archetype: maven-archetype-quickstart:1.1
[INFO] -----
[INFO] Parameter: basedir, Value: C:\Users\CMRIT-ISE-L209-009\Desktop\IS147
[INFO] Parameter: package, Value: com.program2.maven.program2
[INFO] Parameter: groupId, Value: com.program2.maven
[INFO] Parameter: artifactId, Value: program2-example-jar
[INFO] Parameter: packageName, Value: com.program2.maven.program2
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: C:\Users\CMRIT-ISE-L209-009\Desktop\IS147\program2-example-jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 03:37 min
[INFO] Finished at: 2025-01-29T10:31:45+05:30
[INFO] -----

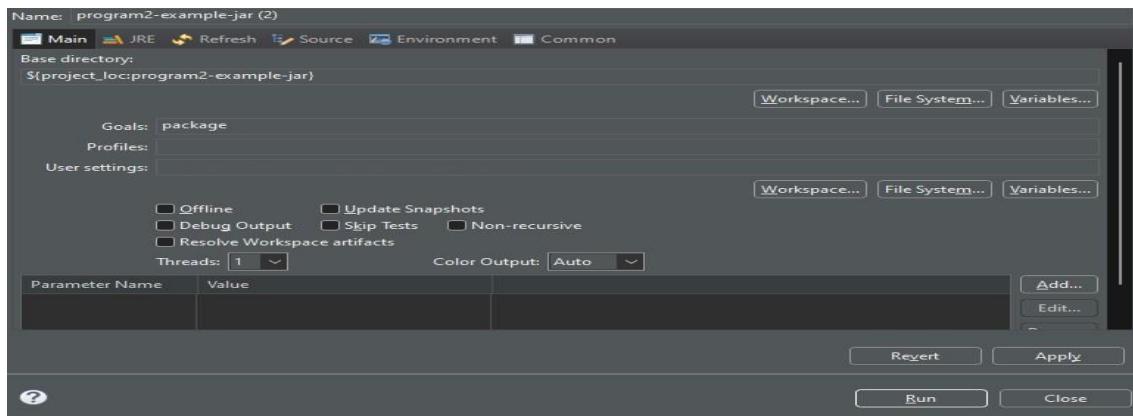
```

STEP 5: Now its time to build Maven Project Go to Maven Project -----

>Right Click on the Project and select Maven Build

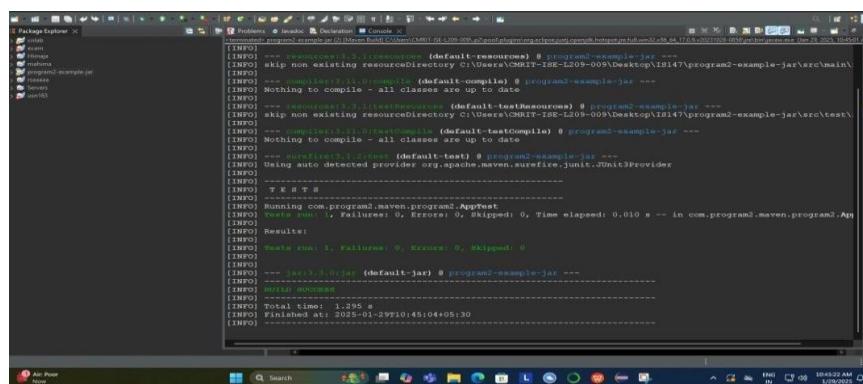
After the above procedure is done

Select Goal as package

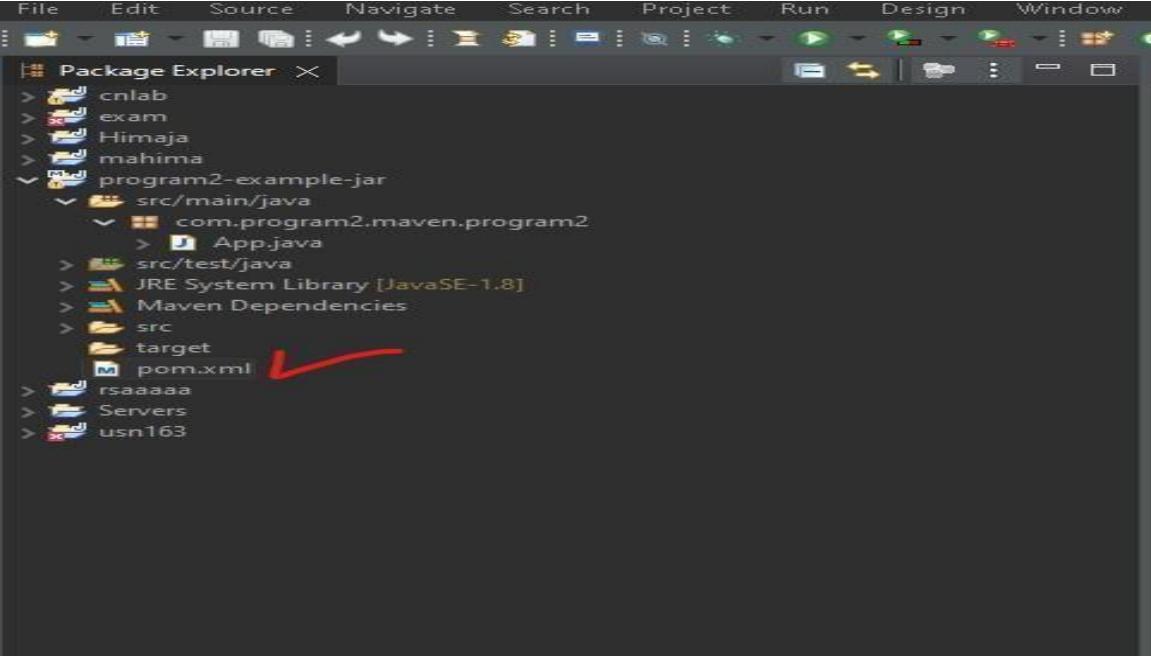


And Click Run

The result be as in below



Now goto App.java finally run java application

HOW POM.XML LOOKS IS AS IN SCREEN BELOW

```
http://maven.apache.org/xsd/maven-4.0.0.xsd (xsi:schemaLocation with catalog)
1<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
3  <modelVersion>4.0.0</modelVersion>
4
5  <groupId>com.program2.maven</groupId>
6  <artifactId>program2-example-jar</artifactId>
7  <version>0.0.1-SNAPSHOT</version>
8  <packaging>jar</packaging>
9
10 <name>program2-example-jar</name>
11 <url>http://maven.apache.org</url>
12
13● <properties>
14   <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
15 </properties>
16
17● <dependencies>
18  <dependency>
19    <groupId>junit</groupId>
20    <artifactId>junit</artifactId>
21    <version>3.8.1</version>
22    <scope>test</scope>
23  </dependency>
24 </dependencies>
25 </project>
26
```

PROGRAM 3:

Working with Gradle: Setting Up a Gradle Project, Understanding Build Scripts (Groovy and Kotlin DSL), Dependency Management and Task Automation

STEP 1: First Goto Command Prompt

Then first make a new directory the command is mkdir pgm3

For changing to a current directory the command is cd pgm3

Now run

gradle init

After execution of command the screen shows as in below where we opt for build type select as 1

```
C:\Users\CMRIT-ISE-L209-009\gradletest>gradle init
Starting a Gradle Daemon (subsequent builds will be faster)

Select type of build to generate:
 1: Application
 2: Library
 3: Gradle plugin
 4: Basic (build structure only)
Enter selection (default: Application) [1..4] 1
```

After selecting application type next it asks for Implementation language select as groovy

```
Select implementation language:
 1: Java
 2: Kotlin
 3: Groovy
 4: Scala
 5: C++
 6: Swift
Enter selection (default: Java) [1..6] 3
```

After selecting Implementation language it will ask for Java version and project name

```
6. SWIFT  
Enter selection (default: Java) [1..6] 3  
  
Enter target Java version (min: 7, default: 21): 21  
  
Project name (default: gradletest): gradletest
```

After providing version and project name

Select application structure as Single application structure and Domain Specific Language as Kotlin

```
Select application structure:  
1: Single application project  
2: Application and library project  
Enter selection (default: Single application project) [1..2] 1  
  
Select build script DSL:  
1: Kotlin  
2: Groovy  
Enter selection (default: Groovy) [1..2] 1
```

After every procedure is over it shows Build successful

```
Generate build using new APIs and behavior (some features may change in the next minor release)? (default: no) [yes, no] yes  
  
> Task :init  
Learn more about Gradle by exploring our Samples at https://docs.gradle.org/8.12.1/samples/sample\_building\_groovy\_applications.html  
  
BUILD SUCCESSFUL in 2m 2s  
1 actionable task: 1 executed
```

STEP 2: Now its time to Build the script Just type the command as:

gradlew run

It will take atleast 3-5 minutes to run the configuration script we have set through steps finally the output be as in below If You want to see the structure of an application run the command as tree

```
C:\Users\CMRIT-ISE-L209-009\gradletest>gradlew run
Calculating task graph as no cached configuration is available for tasks: run

> Task :app:run
Hello World!
```

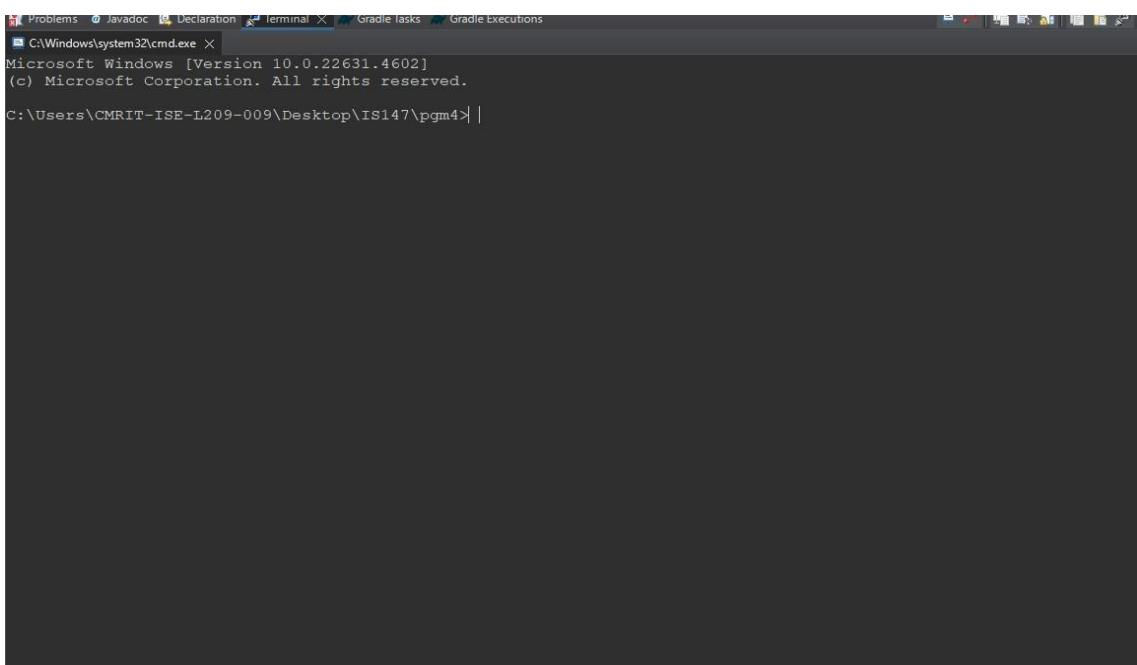
```
C:\Users\CMRIT-ISE-L209-009\gradletest>tree
Folder PATH listing for volume Windows
Volume serial number is CA13-EB08
C:
    +-- .gradle
        +-- 8.12.1
            +-- checksums
            +-- executionHistory
            +-- expanded
            +-- fileChanges
            +-- fileHashes
            +-- vcsMetadata
            +-- buildOutputCleanup
            +-- configuration-cache
                +-- 87330068-729b-48ed-8a38-57771bbaae67
                    +-- 8qybe7c3ykh3sf9t2sllkie4w
            +-- vcs-l
    +-- .settings
    +-- app
        +-- build
            +-- classes
                +-- groovy
                    +-- main
                        +-- org
                            +-- example
            +-- generated
                +-- sources
                    +-- annotationProcessor
                        +-- groovy
                            +-- main
            +-- tmp
                +-- compileGroovy
                    +-- groovy-java-stubs
        +-- src
            +-- main
                +-- groovy
                    +-- org
                        +-- example
                +-- resources
            +-- test
                +-- groovy
                    +-- org
```

PROGRAM 4

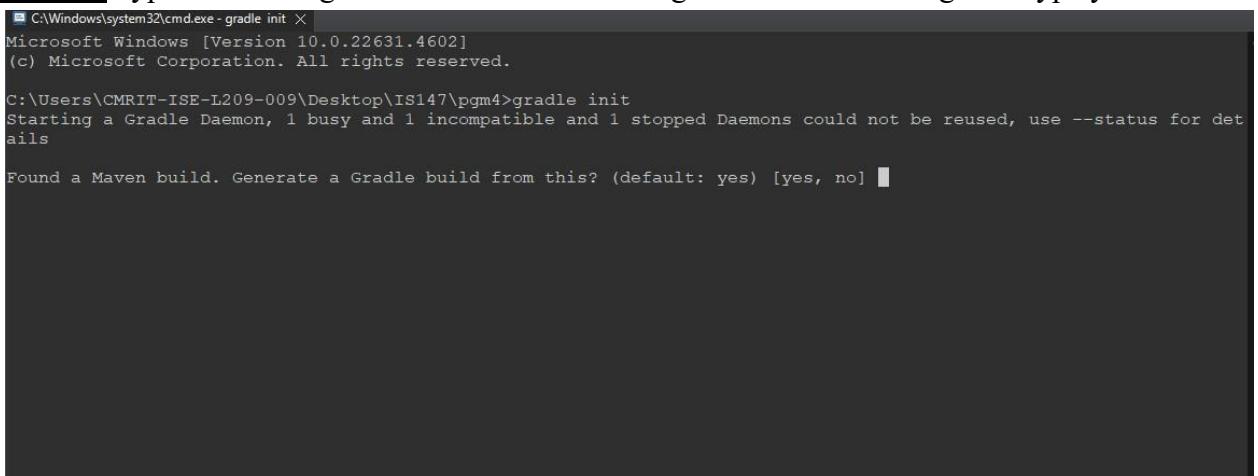
Practical Exercise: Build and Run a Java Application with Maven, Migrate the Same Application to Gradle

STEP 1: First create a Maven Project as in PROGRAM2 then build the project and run java application you will get Hello World Message

STEP 2: Then to migrate to gradle use shortcut Key Ctrl+Alt+Shift+T To get Terminal screen as in below:

A screenshot of a terminal window titled "Terminal". The window shows a command prompt at the bottom with the path "C:\Users\CMRIT-ISE-L209-009\Desktop\IS147\pgm4> |". Above the prompt, there is some system information and a copyright notice from Microsoft.

STEP 3: Type command gradle init it will ask for migrate from maven to gradle type yes

A screenshot of a terminal window titled "Terminal". The user has typed "gradle init" and is prompted with "Found a Maven build. Generate a Gradle build from this? (default: yes) [yes, no] |". The terminal also displays the system path and some initial setup messages.

STEP4 : After the above command is validated to yes it prompts to select Domain Specific Language as in screen below select 2 (as we have done for Kotlin)

```
Select build script DSL:  
 1: Kotlin  
 2: Groovy  
Enter selection (default: Kotlin) [1..2] ■
```

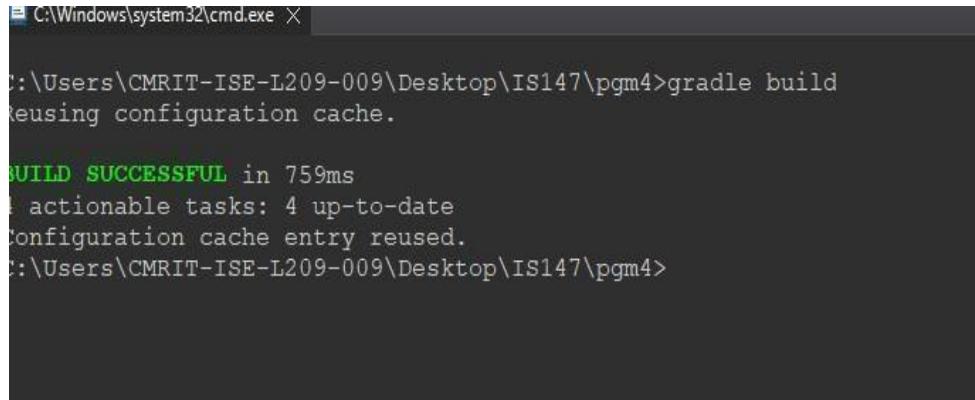
STEP 5: After selecting Groovy it asks for validating prompt for API Generator just validate as yes

```
C:\Windows\system32\cmd.exe - gradle init X  
  
C:\Users\CMRIT-ISE-L209-009\Desktop\IS147\pgm4>gradle init  
  
Found a Maven build. Generate a Gradle build from this? (default: yes) [yes, no] yes  
  
Select build script DSL:  
 1: Kotlin  
 2: Groovy  
Enter selection (default: Kotlin) [1..2] 2  
  
Generate build using new APIs and behavior (some features may change in the next minor release)? (default: no) [yes, no] ||
```

Finally it runs the init phase as been selected

```
> Task :init  
Maven to Gradle conversion is an incubating feature.  
For more information, please refer to https://docs.gradle.org/8.12.1/userguide/migrating_from_maven.html in the Gra  
  
BUILD SUCCESSFUL in 6m 44s  
1 actionable task: 1 executed  
C:\Users\CMRIT-ISE-L209-009\Desktop\IS147\pgm4> |
```

STEP 6: Type the command gradle build



```
C:\Windows\system32\cmd.exe X
C:\Users\CMRIT-ISE-L209-009\Desktop\IS147\pgm4>gradle build
reusing configuration cache.

BUILD SUCCESSFUL in 759ms
1 actionable tasks: 4 up-to-date
configuration cache entry reused.
C:\Users\CMRIT-ISE-L209-009\Desktop\IS147\pgm4>
```

Now to get exact program output of our java file Locate to build gradle File from ur local repository and Copy paste the code as in below shown in red color

```
plugins { id("java-library") id("mavenpublish") id("application") }
```

```
application { mainClass.set("com.pgm4.test.App") // Use .set() for properties }
```

```
repositories { mavenCentral()
    // Uncomment if you need to publish locally
    // mavenLocal()
}
```

```
dependencies { testImplementation("junit:junit:4.13.2") // Use Kotlin syntax
    for dependencies }
```

```
group      =
"com.pgm4.test" version
= "0.0.1-
SNAPSHOT"
description = "pgm4"
java.sourceCompatibility = JavaVersion.VERSION_11 // Consider upgrading
```

```
publishing
{
    publications {
        create< MavenPublication>("maven") { from(components["java"])
    }}
```

```
}

}

tasks.withType<JavaCompile>().configureEach {
    options.encoding = "UTF-8"
}

tasks.withType<Javadoc>().configureEach {
    options.encoding = "UTF-8"
}
```

AFTER DOING ALL CHANGES FINAL STEP

To run commands gradle clean build

gradle run You will get

Output as

Hello World! Welcome to pgm4

```
BUILD SUCCESSFUL in 1s
8 actionable tasks: 6 executed, 2 from cache
Configuration cache entry stored.
C:\Users\CMRIT-ISE-L209-009\Desktop\IS147\pgm4>gradle run
Calculating task graph as configuration cache cannot be reused because file 'build.gradle' has changed.

> Task :run
Hello World! Welcome to pgm4

BUILD SUCCESSFUL in 883ms
2 actionable tasks: 1 executed, 1 up-to-date
Configuration cache entry stored.
C:\Users\CMRIT-ISE-L209-009\Desktop\IS147\pgm4>gradlew run
Reusing configuration cache.

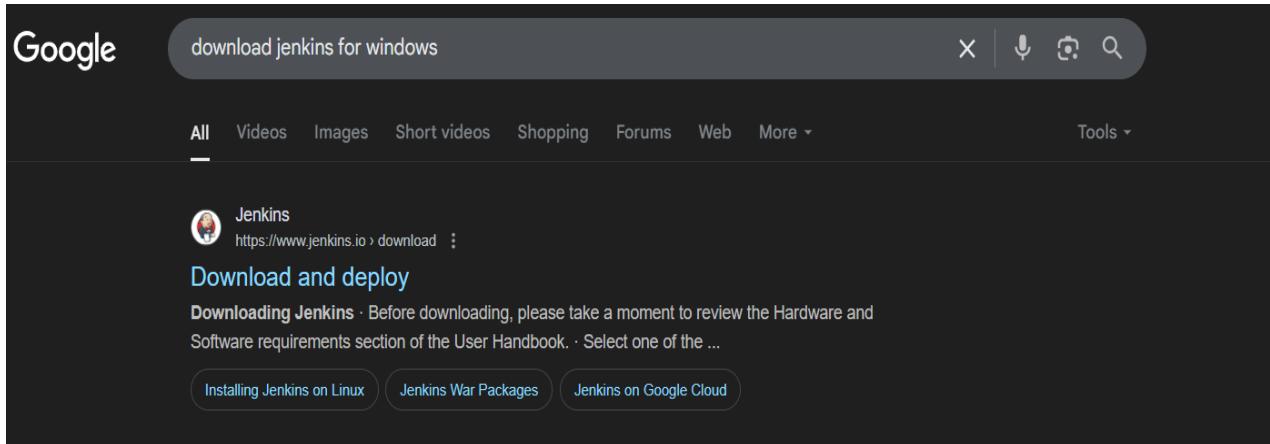
> Task :run
Hello World! Welcome to pgm4

BUILD SUCCESSFUL in 810ms
2 actionable tasks: 1 executed, 1 up-to-date
Configuration cache entry reused.
C:\Users\CMRIT-ISE-L209-009\Desktop\IS147\pgm4> |
```

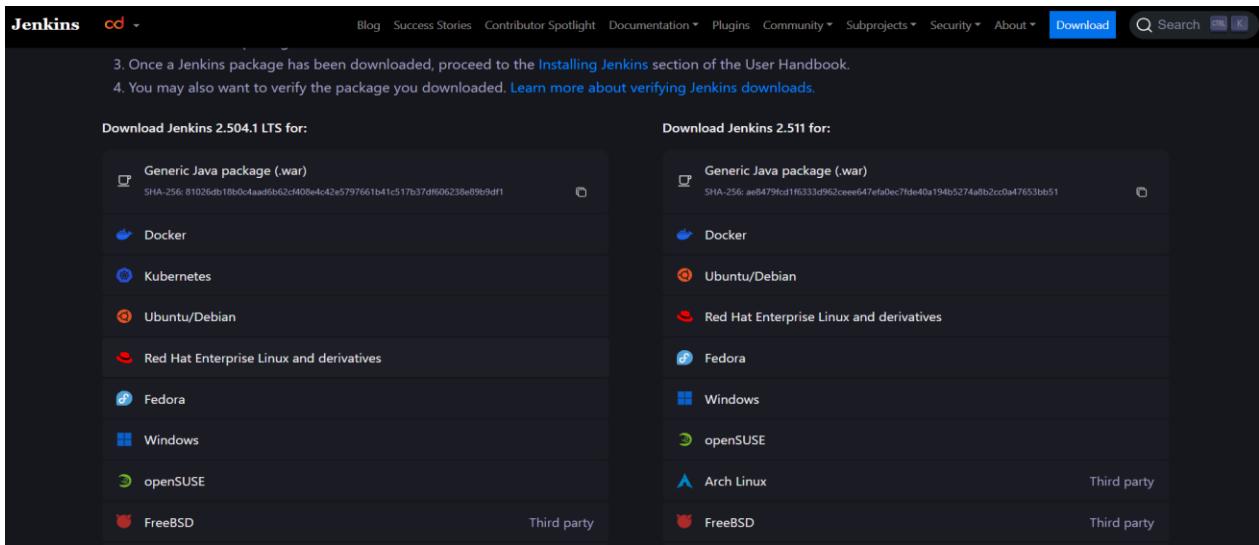
PROGRAM5:

Introduction to Jenkins: What is Jenkins?, Installing Jenkins on Local or Cloud Environment, Configuring Jenkins for First Use .

STEP 1: Type Jenkins download for windows in Browser and Click on Download Deploy



STEP 2: Go to first Column -> Download Jenkins 2.504.1 LTS for -> Ubuntu/Debian



STEP 3: Copy the commands from long term support release and paste in terminal then execute one by one

```

student@student1: $ java --version
openjdk 17.0.14 2025-01-21
OpenJDK Runtime Environment (build 17.0.14+7-Ubuntu-122.04.1)
OpenJDK 64-Bit Server VM (build 17.0.14+7-Ubuntu-122.04.1, mixed mode, sharing)
student@student1: $ sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian-stable binary/" | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins
[sudo] password for student:
--2025-04-08 15:25:12-- https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
Resolving pkg.jenkins.io (pkg.jenkins.io)... 151.101.158.133, 2a04:4e42:25::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|151.101.158.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3175 (3.1K) [application/pgp-keys]
Saving to: '/usr/share/keyrings/jenkins-keyring.asc'

/usr/share/keyrings/jenkins-keyring 100%[=====] 3.10K ---KB/s in 0s
2025-04-08 15:25:12 (16.0 MB/s) - '/usr/share/keyrings/jenkins-keyring.asc' saved [3175/3175]

Ign:2 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:3 https://pkg.jenkins.io/debian-stable binary/ Release [2,044 B]
Get: https://packages.microsoft.com/repos/code_stable InRelease [3,590 B]
Get:4 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Hit:5 https://in.archive.ubuntu.com/ubuntu bionic InRelease
Get: https://packages.microsoft.com/repos/code_stable/main amd64 Packages [18.2 kB]
Hit:7 https://archive.ubuntu.com/ubuntu jammy InRelease
Get:8 https://packages.microsoft.com/repos/code_stable/main armhf Packages [18.5 kB]
Get:9 https://packages.microsoft.com/repos/code_stable/main arm64 Packages [18.4 kB]
Get:10 https://pkg.jenkins.io/debian-stable binary/ Packages [28.8 kB]
Hit:11 https://repositories.intel.com/graphics/ubuntu focal InRelease
Get:12 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:15 http://archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [784 kB]

```

STEP 4: Go to Start Jenkins execute commands one by one

```

student@student1: ~
student@student1: $ node-fs.realpath node-function-bind node-get-stream node-glob node-growl node-has-flag node-has-unicode node-hosted-git-info
node-iconv-lite node-iferr node-imurmurhash node-indent-string node-inflight node-inherits node-init node-ip node-ip-regex node-is-buffer
node-ts-plain-obj node-is-typedarray node-isarray node-lsexec node-json-parse-better-errors node-jsonparse node-kind-of
node-lodash-packages node-lowercase-keys node-lru-cache node-mimic-response node-minimatch node-minimist node-minipass node-mute-stream
node-negotiator node-npm-bundled-node-once node-osenv node-p-cancelable node-p-map node-path-is-absolute node-process-nextick-args
node-promise-inflight node-promise-retry node-promzard node-pump node-quick-lru node-read node-readable-stream node-resolve node-retry
node-safe-buffer node-set-blocking node-signal-exit node-slash node-slice-ansi node-source-map node-spdx-correct node-spdx-exceptions
node-spdx-expression-parse node-spdx-license-ids node-sprintf-js node-stealthy-require node-string-decoder node-supports-color
node-text-table node-time-stamp node-tmatch node-typedarray-to-buffer node-universalfy node-util-deprecate
node-validate-npm-package-license node-webidl-conversions node-whatwg-fetch node-wrappy node-yallist nodejs-doc
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 127 not upgraded.
student@student1: $ sudo systemctl enable jenkins
Synchronizing state of jenkins.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable jenkins
student@student1: $ sudo systemctl start jenkins
student@student1: $ sudo systemctl status jenkins
● Jenkins.service - Jenkins Continuous Integration Server
  Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
  Active: active (running) since Tue 2025-04-08 15:26:33 IST; 3min 34s ago
    Main PID: 8765 (java)
      Tasks: 45 (limit: 4503)
        Memory: 637.3M
          CPU: 19.902s
        CGroup: /system.slice/jenkins.service
           └─8765 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Apr 08 15:26:27 student1 jenkins[8765]: 3ae6bb5143734fd1b412b624e28a46b6
Apr 08 15:26:27 student1 jenkins[8765]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Apr 08 15:26:27 student1 jenkins[8765]: ****
Apr 08 15:26:27 student1 jenkins[8765]: ****
Apr 08 15:26:33 student1 jenkins[8765]: 2025-04-08 09:56:33.525+0000 [id=30]      INFO      jenkins.InitReactorRunner$1#onAttained: Com>
Apr 08 15:26:33 student1 jenkins[8765]: 2025-04-08 09:56:33.541+0000 [id=23]      INFO      hudson.lifecycle.Lifecycle#onReady: Jenkins>
Apr 08 15:26:33 student1 systemd[1]: Started Jenkins Continuous Integration Server.
Apr 08 15:26:35 student1 jenkins[8765]: 2025-04-08 09:56:35.239+0000 [id=48]      INFO      h.m.DownloadService$Downloadable#load: Obta>
Apr 08 15:26:35 student1 jenkins[8765]: 2025-04-08 09:56:35.239+0000 [id=48]      INFO      hudson.util.Retriger#start: Performed the ac>
lines 1-20/20 (END)

```

STEP5: Then Go to browser type localhost :8080

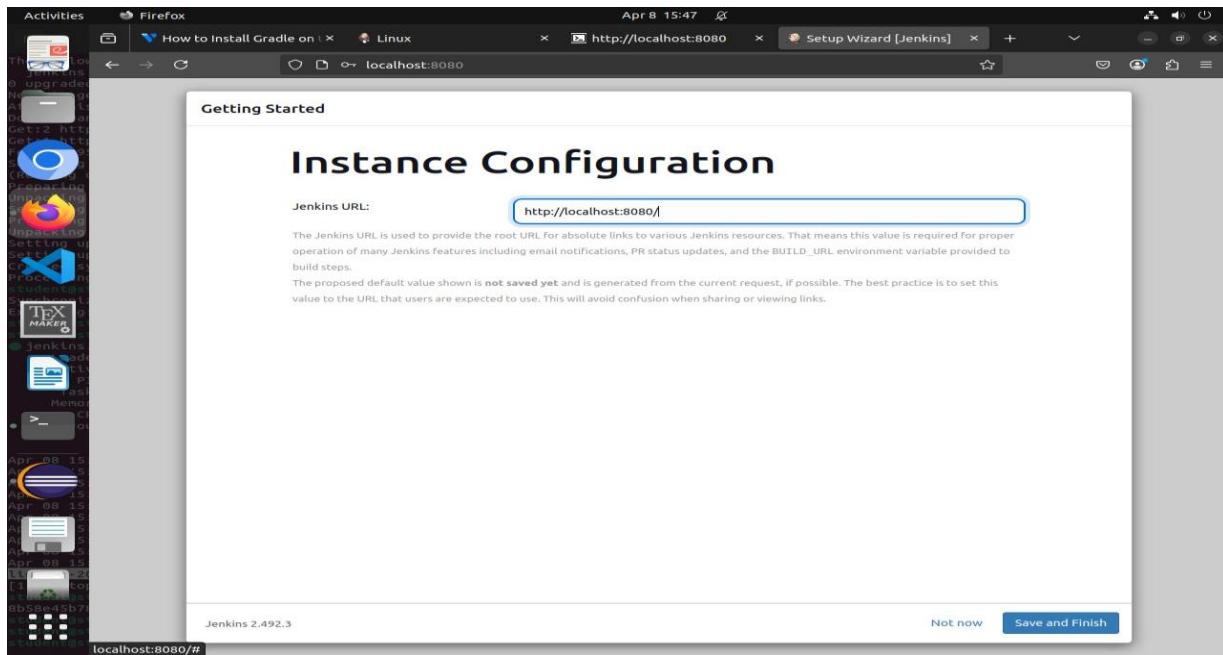
- Click on first link, click on 1st window then you will get path copy that path and paste it on terminal then you get the password



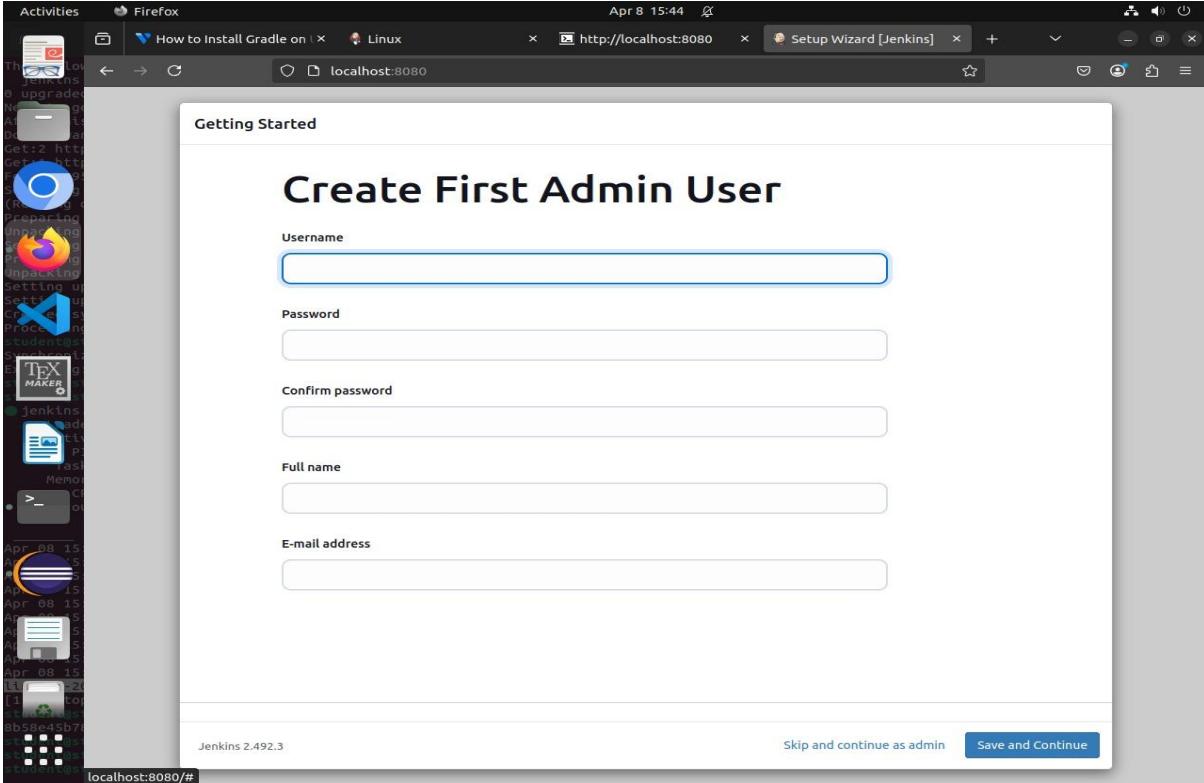
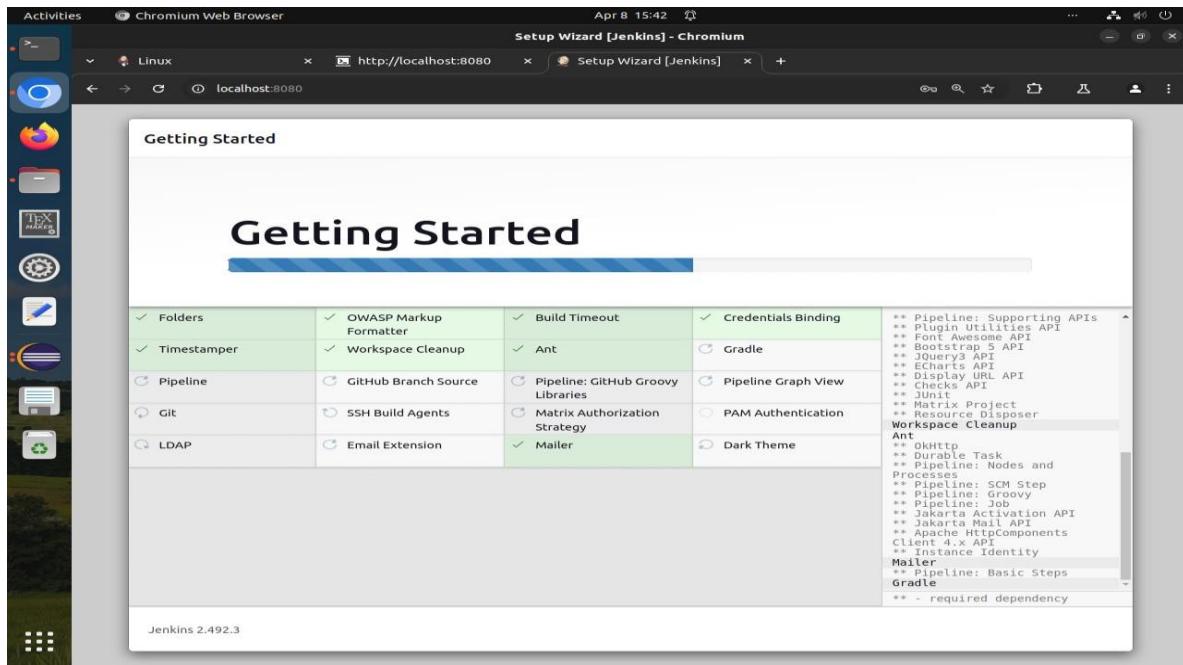
```
student@student1:~$ sudo more /var/lib/jenkins/secrets/initialAdminPassword
3aeobb5143734fd1b412b624e28a46b6
student@student1:~$
```

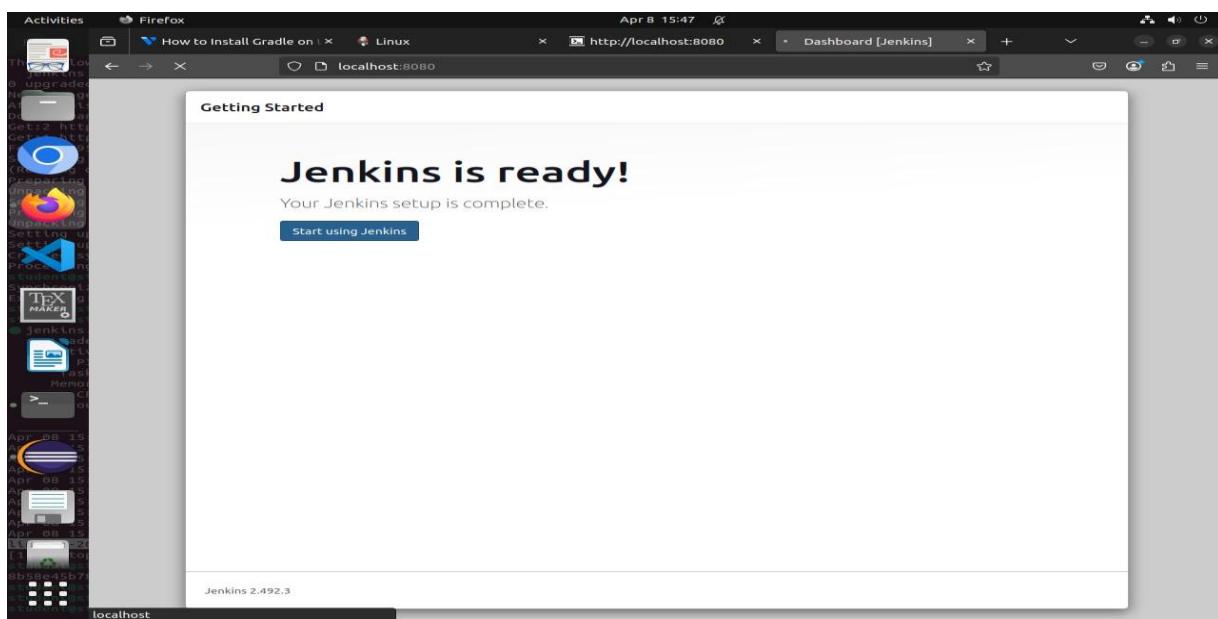
STEP 6: Paste the password to Jenkins window and click on continue

- Click on install suggested plugins

**STEP 7: Next Sign-Up page will open after all plugin's installation**

- **Username:**
- **Password:**
- **Confirm password:**
- **Full name:**
- **Email**
- Press save and continue then URL of Jenkins
- Save and finish





A screenshot of a Chromium Web Browser window. The title bar shows 'Activities Chromium Web Browser Apr 8 15:45'. The main content area displays the Jenkins Dashboard with the message 'Welcome to Jenkins!', build history, and various management options like 'Create a job', 'Set up an agent', and 'Configure a cloud'. The status bar at the bottom indicates 'Jenkins 2.492.3'.

PROGRAM 6:

Continuous Integration with Jenkins: Setting Up a CI Pipeline, Integrating Jenkins with Maven/Gradle, Running Automated Builds and Tests How Is Jenkins Used for Continuous Integration?

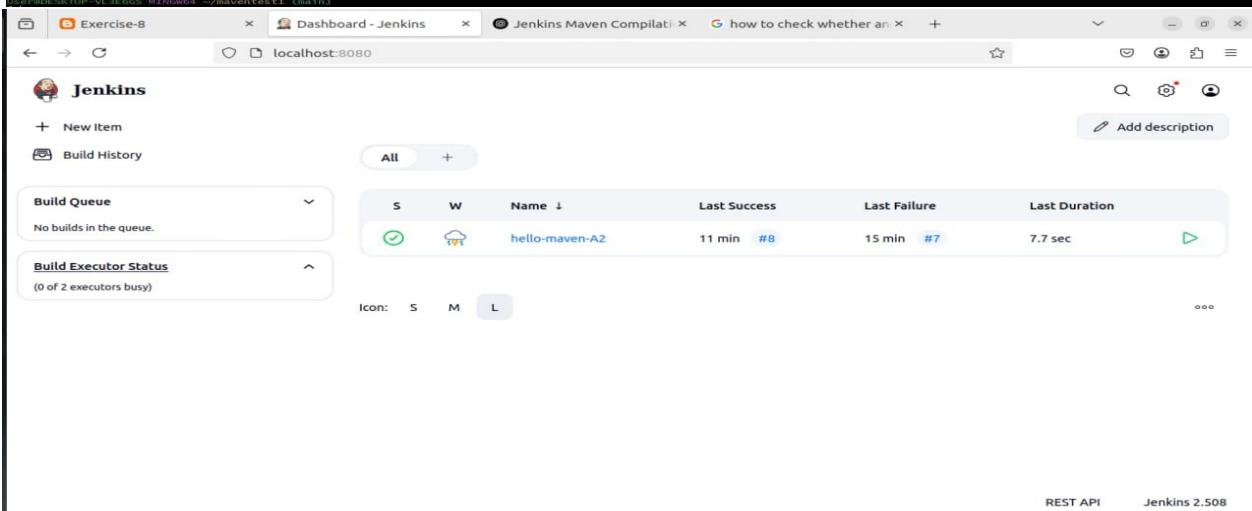
STEP 1: Create pom.xml in the terminal -> create a directory maven project

Now coming to our program to set CI pipeline for maven go to Jenkins dashboard keep on manage Jenkins -> Plugins

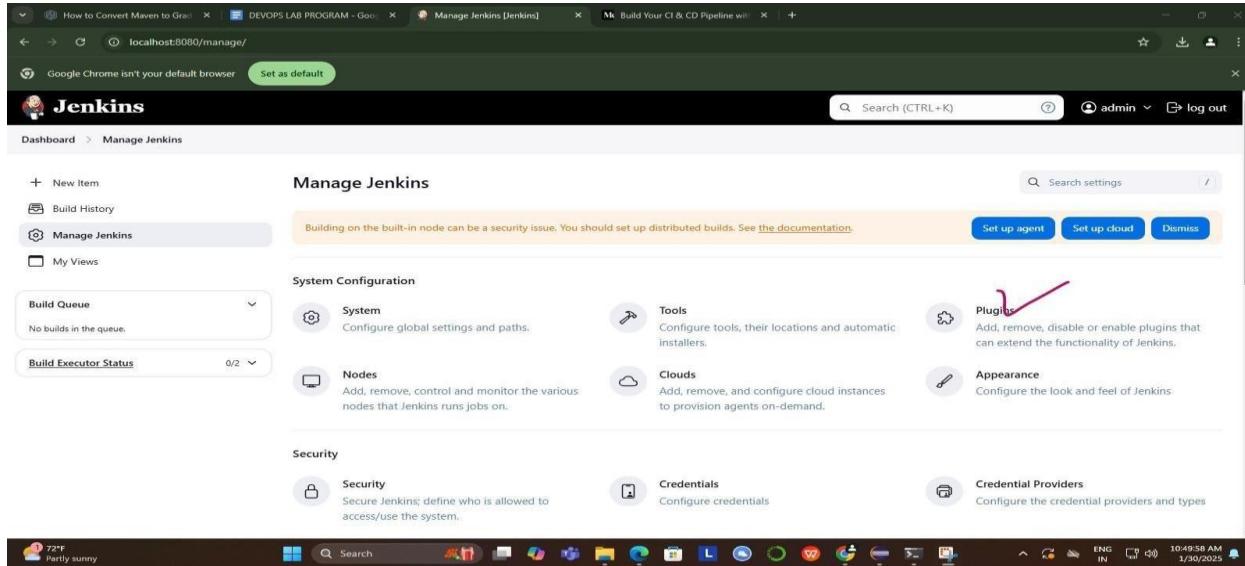
```

$ mvn archetype:generate -DgroupId=com.dineshonjava -DartifactId=javaHelloWorld -DarchetypeArtifactDescriptorId=maven-archetype-quickstart -DinteractiveMode=false
[INFO] Scanning for projects...
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/3.1.2/maven-install-plugin-3.1.2.pom (8.5 kB at 9.8 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/3.1.2/maven-install-plugin-3.1.2.jar (32 kB at 360 kB/s)
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/3.1.2/maven-install-plugin-3.1.2.pom (9.6 kB at 171 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-deploy-plugin/3.1.2/maven-deploy-plugin-3.1.2.pom (40 kB at 964 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-deploy-plugin/3.1.2/maven-deploy-plugin-3.1.2.jar (40 kB at 964 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-deploy-plugin/3.1.2/maven-deploy-plugin-3.1.2.pom (15 kB at 524 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/4.1/maven-plugins-41.pom (4 kB at 100 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-assembly-plugin/3.2.1/maven-assembly-plugin-3.2.1.jar (240 kB at 1.6 MB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-dependency-plugin/3.7.0/maven-dependency-plugin-3.7.0.pom (19 kB at 301 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-dependency-plugin/3.7.0/maven-dependency-plugin-3.7.0.jar (19 kB at 301 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-dependency-plugin/3.7.0/maven-dependency-plugin-3.7.0.pom (207 kB at 4.2 MB/s)
[INFO] Generating Maven Stub Project (No POM) ...
[INFO] Building Maven Stub Project (No POM) [ pom ]
[INFO] 
[INFO] >>> archetype:3.3:generate  (default-cli) > generate-sources @ standalone-pom >>>
[INFO] <<< archetype:3.3:generate  (default-cli) < generate-sources @ standalone-pom <<<
[INFO] 
[INFO] --- archetype:3.3:generate  (default-cli) @ standalone-pom ---
[INFO] Generating project in batch mode
[INFO] Using following parameters for creating project from Old CLI < Archetype: maven-archetype-quickstart>1.0>
[INFO] Parameter: basedir, Value: C:\Users\User\maventest1
[INFO] Parameter: groupId, Value: com.dineshonjava
[INFO] Parameter: artifactId, Value: javaHelloWorld
[INFO] Parameter: packageName, Value: com.dineshonjava
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] project created from Old CLI < Archetype in dir: C:\Users\User\maventest1\javaHelloWorld
[INFO] BUILD SUCCESS
[INFO] Total time: 5.691 s
[INFO] Finished at: 2025-02-04T17:58:22+05:30
[INFO] 

```



STEP2: Select Plugins

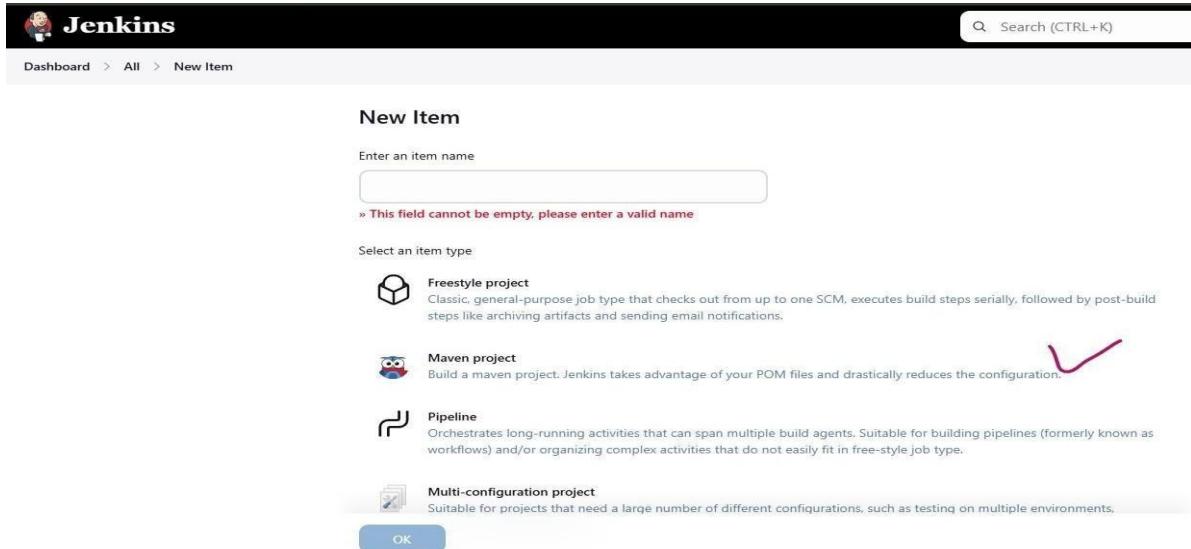


STEP 3: Search for Maven Integration Plugin in Available Plugins and Install

The screenshot shows the Jenkins 'Test Result' page for a Maven project. On the left sidebar, there are links: Status, Changes, Console Output, Edit Build Information, History, Timings, Git Build Data, and Test Result (which is selected). The main area shows the 'Test Result' section with the message '0 failures, 2 skipped'. Below it, it says '54 tests Took 23 sec.' and has a 'Add description' button. The 'All Tests' section displays a table of test results:

Package	Duration	Fail	(diff)	Skip	(diff)	Pass	(diff)	Total
org.springframework.samples.petclinic	0.6 sec	0		2	+2	2	+2	4
org.springframework.samples.petclinic.model	1.4 sec	0		0		1	+1	1
org.springframework.samples.petclinic.owner	2.3 sec	0		0		33	+33	33
org.springframework.samples.petclinic.service	0.25 sec	0		0		10	+10	10

STEP 4: After Maven Integration Plugin is Installed We able to see Maven Project as New Item



STEP 5: YET not completed we have to configure the Location to properly Build and Run Maven Project So again click on Manage Jenkins and select Tools

Google Chrome isn't your default browser Set as default

Jenkins

Dashboard > Manage Jenkins

Manage Jenkins

Building on the built-in node can be a security issue. You should set up distributed builds. See [the documentation](#).

Set up agent Set up cloud Dismiss

System Configuration

- Build Queue**
- Build Executor Status** (0 of 2 executors busy)
- Nodes**
- Tools** **✓** Configure tools, their locations and automatic installers.
- Clouds** Add, remove, and configure cloud instances to provision agents on-demand.
- Plugins** Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
- Appearance** Configure the look and feel of Jenkins

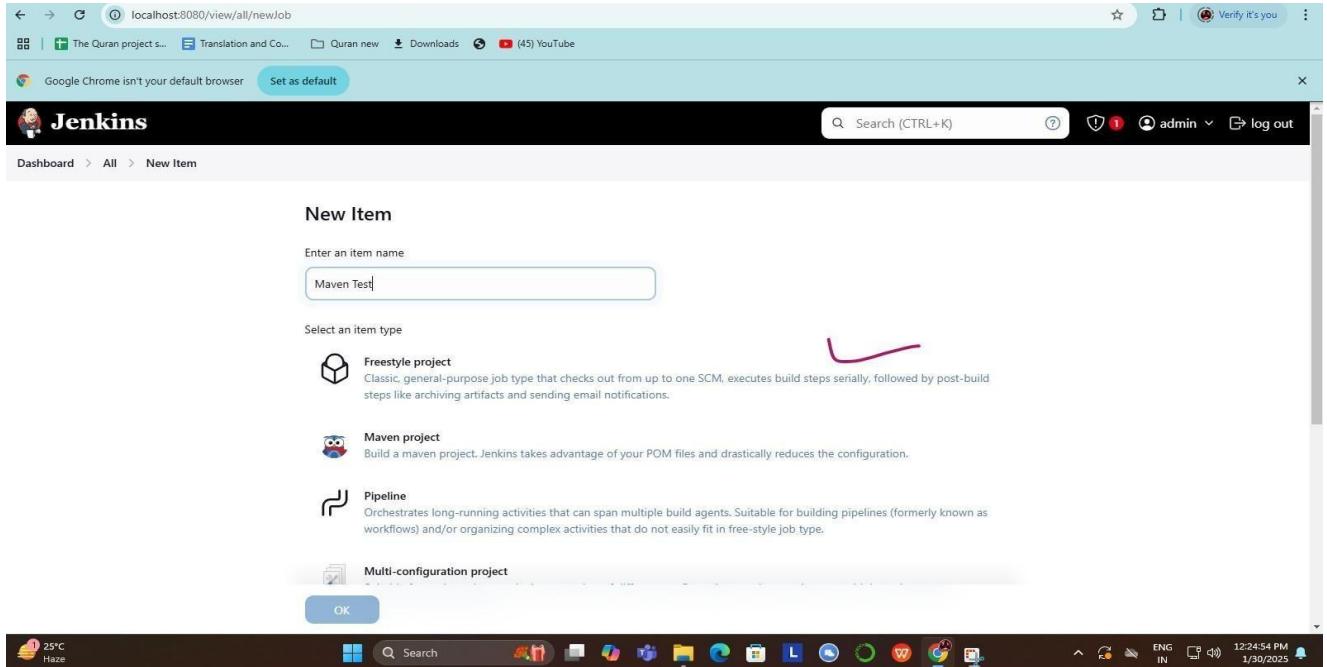
Security

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

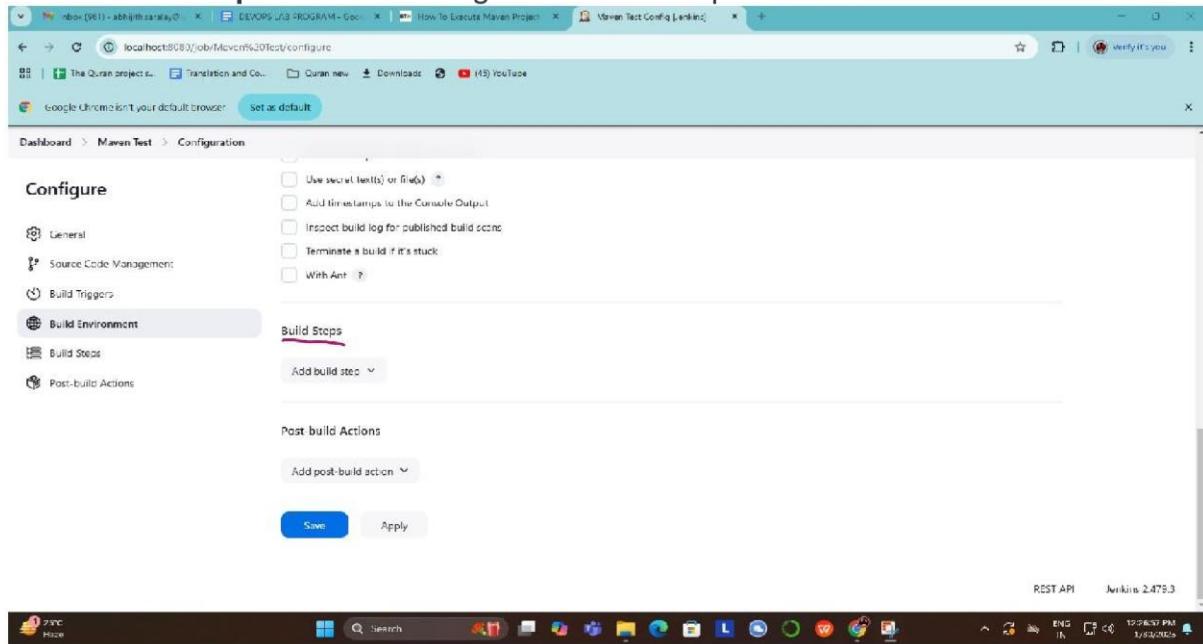
73°F Partly sunny Search ENG IN 11:17:44 AM 1/30/2025

STEP 6: Now lets not select Maven Project as new Item as we already have Maven project in local systems lets see how we can run the Maven Project with POM.XML

- Click on New Item
- Provide Item Name and select Freestyle Project



- c) Scroll down to 'Build' option. Click on 'Add Build Step' and choose the value 'Invoke top-level Maven targets' from the drop down list.



- c) After selecting Invoke top-level Maven targets opt for proper environment version as in set in previous steps in my case its MAVEN_HOME

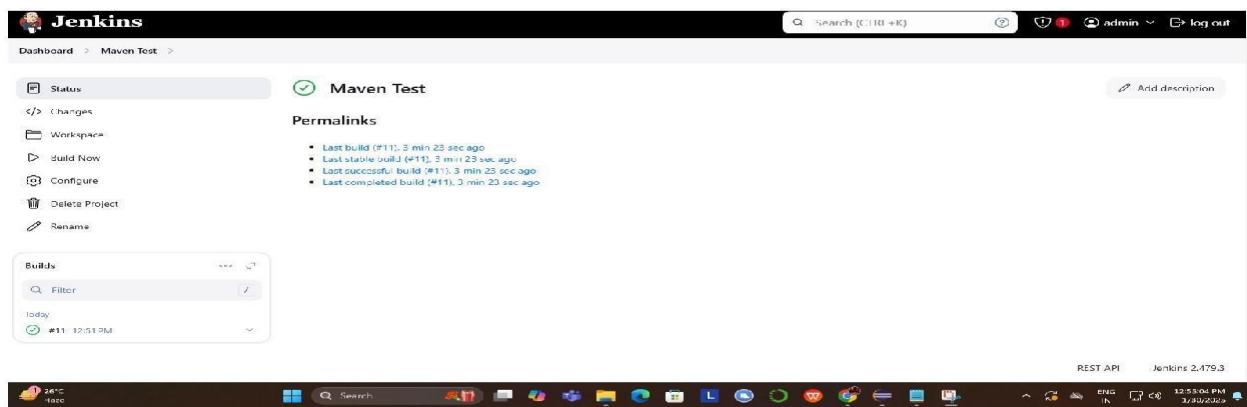
Build Steps



- d) Enter Goal as clean install
e) Before you save and apply just below Goal there is Advance option add pom.xml path



Goto pom.xml of your particular pgm and take path in my case its
C:\Users\CMRIT-ISE-L209-009\Desktop\IS147\pgm4\pom.xml
After all Steps is over click on Build button the output be as in below



The screenshot shows the Jenkins interface with two separate console output sections for different builds.

Build #1 Console Output:

```

10:24:01 Started by user admin
10:24:01 Running as SYSTEM
10:24:03 Building in workspace /var/lib/jenkins/workspace/ci-pipeline
10:24:03 [WS-CLEANUP] Deleting project workspace...
10:24:03 [WS-CLEANUP] Deferred wipeout is used...
10:24:07 [WS-CLEANUP] Done
10:24:08 The recommended git tool is: NONE
10:24:08 No credentials specified
10:24:08 Cloning the remote Git repository
10:24:09 Cloning repository https://github.com/namrathank/jenkins-ci-pipeline.git
10:24:09 > git init /var/lib/jenkins/workspace/ci-pipeline # timeout=10
10:24:09 Fetching upstream changes from https://github.com/namrathank/jenkins-ci-pipeline.git
10:24:09 > git --version # timeout=10
10:24:09 > git --version # 'git version 2.34.1'
10:24:09 > git fetch --tags --force --progress -- https://github.com/namrathank/jenkins-ci-pipeline.git +refs/heads/*:refs/remotes/origin/* # timeout=10
10:27:38 > git config remote.origin.url https://github.com/namrathank/jenkins-ci-pipeline.git # timeout=10
10:27:42 > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10

```

Build #2 Console Output:

```

10:29:58 [INFO] Installing /var/lib/jenkins/workspace/ci-pipeline/target/my-app-1.0-SNAPSHOT.jar
10:29:58 [INFO] Archiving /var/lib/jenkins/workspace/ci-pipeline/pom.xml to /var/lib/jenkins/.m2/repository/com/example/my-app/1.0-SNAPSHOT/my-app-1.0-SNAPSHOT.pom
10:29:58 [INFO] Installing /var/lib/jenkins/workspace/ci-pipeline/target/my-app-1.0-SNAPSHOT-jar-with-dependencies.jar to /var/lib/jenkins/.m2/repository/com/example/my-app/1.0-SNAPSHOT/my-app-1.0-SNAPSHOT-jar-with-dependencies.jar
10:29:53 [INFO] ...
10:29:53 [INFO] BUILD SUCCESS
10:29:53 [INFO] ...
10:29:53 [INFO] Total time: 11.932 s
10:29:53 [INFO] Finished at: 2025-04-12T10:29:53+05:30
10:29:53 [INFO] ...
10:29:53 Waiting for Jenkins to finish collecting data
10:29:55 [JENKINS] Archiving /var/lib/jenkins/workspace/ci-pipeline/pom.xml to com.example/my-app/1.0-SNAPSHOT/my-app-1.0-SNAPSHOT.pom
10:29:55 [JENKINS] Archiving /var/lib/jenkins/workspace/ci-pipeline/target/my-app-1.0-SNAPSHOT.jar to com.example/my-app/1.0-SNAPSHOT/my-app-1.0-SNAPSHOT.jar
10:29:55 [JENKINS] Archiving /var/lib/jenkins/workspace/ci-pipeline/target/my-app-1.0-SNAPSHOT-jar-with-dependencies.jar to com.example/my-app/1.0-SNAPSHOT-jar-with-dependencies.jar
10:29:55 channel stopped
10:29:57 Finished: SUCCESS

```

Both sections include timestamp configuration dropdowns and standard Jenkins navigation links.

REST API Jenkins 2.505

PROGRAM 7:

Configuration Management with Ansible: Basics of Ansible: Inventory, Playbooks, and Modules, Automating Server Configurations with Playbooks, Hands-On: Writing and Running a Basic Playbook. How Do I Install Ansible on Ubuntu?

STEP 1: Configure Ansible Control Node

The Ansible control node is a system used to connect to and manage Ansible host servers.

Proceed with the steps below to set up the control node on the main server:

1. Create an administrator-level user for the control node. Use the adduser command: optionally, provide more details about the user by answering questions. Press enter to skip a question

3. Use the following usermod command to assign superuser privileges to the account:

sudo usermod -aG sudo [username]

sudo adduser [username]

2. When prompted, define a strong password.

```
marko@phoenixnap:~$ sudo adduser ansible
[sudo] password for marko:
Adding user `ansible' ...
Adding new group `ansible' (1001) ...
Adding new user `ansible' (1001) with group `ansible' ...
Creating home directory `/home/ansible' ...
Copying files from `/etc/skel' ...
New password: ←
Retype new password:
passwd: password updated successfully
Changing the user information for ansible
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
marko@phoenixnap:~$
```

A membership in the sudo group allows the user to utilize the sudo command to perform administrative tasks.

4. Switch to the newly created user on the control node:

```
sudo su [username]
```

STEP 2: Set up an SSH Key pair

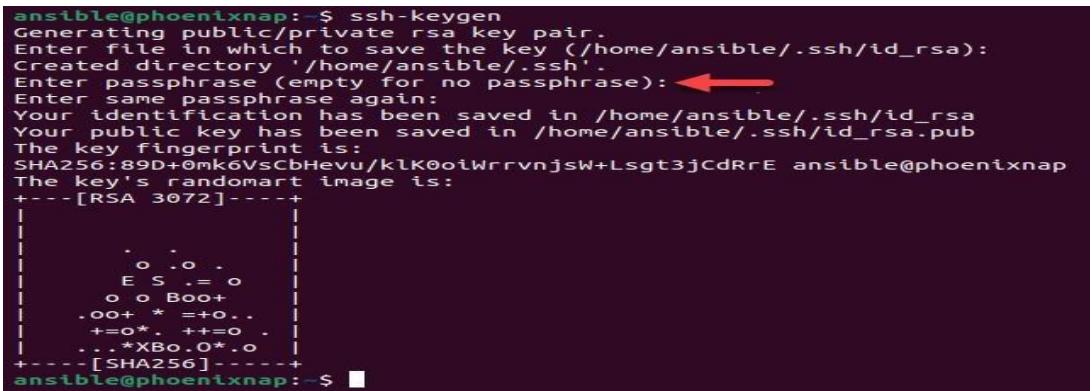
The Ansible control node uses SSH to connect to hosts. Generate an SSH key pair for the Ansible user by executing the following steps:

1. Enter the command below using the Ansible control node command line:

```
ssh-keygen
```

2. When prompted, provide a passphrase. While adding a strong passphrase is recommended, pressing Enter allows the user to skip the passphrase creation.

The system generates the public/private key pair and prints the randomart image.



```
ansible@phoenixnap:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ansible/.ssh/id_rsa):
Created directory '/home/ansible/.ssh'.
Enter passphrase (empty for no passphrase): ←
Enter same passphrase again:
Your identification has been saved in /home/ansible/.ssh/id_rsa
Your public key has been saved in /home/ansible/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:89D+0mk6VscbHevu/klK0oiWrrvnjsW+Lsgt3jCdRrE ansible@phoenixnap
The key's randomart image is:
+---[RSA 3072]---+
|          . o . .
|          E S . = o
|          o o Boo+
|          .oo+ * =+o..
|          +=o*. ++=o .
|          ...*XBo.O*.o
+---[SHA256]---+
ansible@phoenixnap:~$
```

STEP 3: Configure an Ansible Host

1. Use the following ssh-copy-id command on the control node to copy the public key to a host:

```
key to a host: ssh-copy-id [username]@[remote-host] ssh ansible@192.168.0.81
```

2. Type yes and hit Enter when asked whether to continue connecting to an authenticated host.
3. Enter the remote host account password.

```
ansible@phoenixnap:~$ ssh-copy-id ansible@192.168.0.81
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ansible/.ssh/id_rsa.pub"
The authenticity of host '192.168.0.81 (192.168.0.81)' can't be established.
ED25519 key fingerprint is SHA256:fg67eTA0FjkEJlRACQyxna/MDC7zX4f0dABzt+aktGM.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
ansible@192.168.0.81's password: ←
Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'ansible@192.168.0.81'"
and check to make sure that only the key(s) you wanted were added.

ansible@phoenixnap:~$
```

The utility uploads the public key to the remote host account.

STEP 4 : Install Ansible

Use the APT package to install the Ansible package on the control node system:

1. Ensure the package index is up to date

```
sudo apt update
```

2. Install Ansible on Ubuntu with the following command: sudo apt install

```
ansible -y
```

STEP 5: Verify the Installation

Check that Ansible was successfully installed on your Ubuntu system using the ansible command: ansible –version

The output displays the Ansible version number, the location of the configuration file, the path to the executable, and other information.

```
ansible@phoenixnap: $ ansible --version
ansible 2.10.8 ←
  config file = /home/ansible/ansible.cfg
  configured module search path = ['~/home/ansible/.ansible/plugins/modules', '/usr/share
/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0]
ansible@phoenixnap: $
```

STEP 6: Set up the inventory file

Once Ansible installed on the control node, set up an inventory file to allow Ansible to communicate with remote hosts. The inventory file contains all the information about the remote hosts managed through the Ansible control node.

Follow the steps below to create an inventory file on the control node:

1. Create the ansible subdirectory in the etc directory:

```
sudo mkdir -p /etc/ansible
```

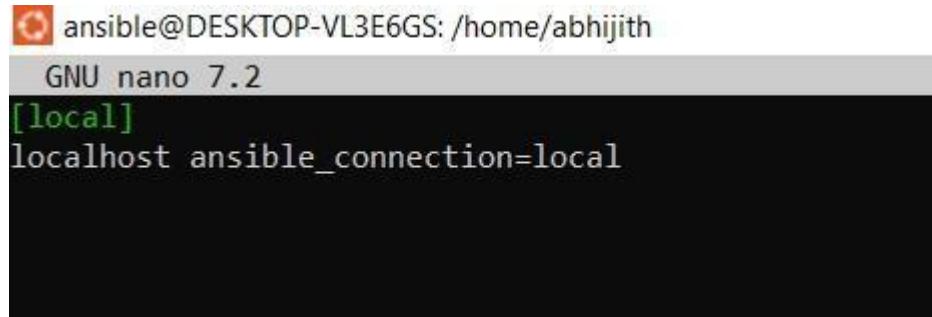
2. Use a text editor such as Nano to create a file named hosts:

```
sudo nano /etc/ansible/hosts
```

3. Add localhost that the control node will manage. Use the following format:

```
[local] localhost ansible_connection=local
```

The [local] line allows for the creation of categories to organize local hosts. The following example adds a local host using its local IP address 192.168.0.81 and sorts it into the servers category:



The screenshot shows a terminal window titled 'GNU nano 7.2' with the following content:

```
[local]
localhost ansible_connection=local
```

4. Save the file and exit.

5. Enter the command below to check the items in the inventory: ansible-inventory --list -y

The output lists the hosts:

```
ansible@DESKTOP-VL3E6GS:/home/abhijith$ ansible-inventory --list -y
all:
  children:
    local:
      hosts:
        localhost:
          ansible_connection: local
ansible@DESKTOP-VL3E6GS:/home/abhijith$
```

STEP 7: Test the Connection

To ensure the Ansible control node can connect to the local hosts and run commands, use the following ansible command to ping the hosts from the control node: sudo ansible all -m ping

The output confirms the successful connection.

```
ansible@DESKTOP-VL3E6GS:/home/abhijith$ sudo ansible all -m ping
localhost | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
ansible@DESKTOP-VL3E6GS:/home/abhijith$
```

PROGRAM 8:

Set up a Jenkins CI Pipeline for a Maven Project, Use Ansible to Deploy Artifacts Generated by Jenkins

STEP 1: Create a simple Maven ‘HelloWorld’ project using either Eclipse IDE or the terminal and Run a test build.

STEP 2: Create Jenkins Job

- Go to Jenkins dashboard → click New Item → Freestyle Project
- Enter job name: HelloWorld-Maven-Build
- Select Git as a source code Management
- Repository URL: Use a local git repo or push your Maven project to GitHub and use that link.

STEP 3: Configuration build

- Under Build, add:
mvn clean package
- Under Post-build Actions, select:
Archive the artifact: target/helloworld-1.0-SNAPSHOT.jar

Save and Build project

STEP 4: Create an Ansible Playbook for Deployment

Create the directory for the playbook:

```
mkdir -p ~/ansible-deploy  
cd ~/ansible-deploy  
nano deploy.yml
```

Save and exit.

```
user1@user1-A520M-K:/ansible-deploy$ ansible-playbook deploy.yml
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'

PLAY [Deploy Maven Artifact Locally] ****
TASK [Gathering Facts] ****
ok: [localhost]

TASK [Copy JAR to /opt/app] ****
changed: [localhost]

PLAY RECAP ****
localhost : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

STEP 5: Integrate Ansible with Jenkins

- Open Jenkins Dashboard → select HelloWorld-Maven_Build job
- Go to configure → Post-build Actions
- Click “Invoke Ansible Playbook”
- Playbook path: /home/user/ansible-deploy/deploy.yml
- Inventory file: /etc/ansible/host

Save and Trigger the Build.

S	W	Name ↓	Last Success	Last Failure	Last Duration
Green checkmark	Cloud icon	hello-maven-A2	11 min #8	15 min #7	7.7 sec

PROGRAM 9:

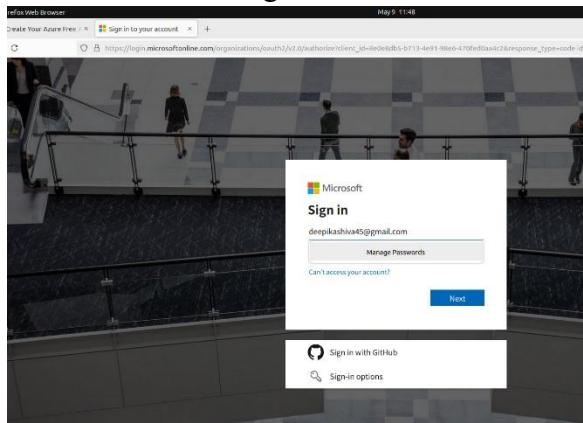
Introduction to Azure DevOps: Overview of Azure DevOps Services, Setting Up an Azure DevOps Account and Project.

STEP 1: Go to Google chrome and type azure for students <https://azure.microsoft.com/en-us/free/students>

Click on Create one, If u have Github account u can Sign in using github account better way is to create one account

The screenshot shows a search result for "azure for students". Below the search bar, there are navigation links: All, Videos, Images, Shopping, News, Web, Books, More. The first result is from Microsoft Azure, titled "Azure for Students – Free Account Credit". It states: "With Microsoft Azure for Students, get a \$100 credit when you create your free account. There is no credit card needed and 12 months of free Azure services." Below it is another result for "Azure for College Students—Offer Details", which says: "Students, get Azure for free courtesy of Microsoft Azure. College students enrolled full time are eligible for Azure free account with \$100 in credits."

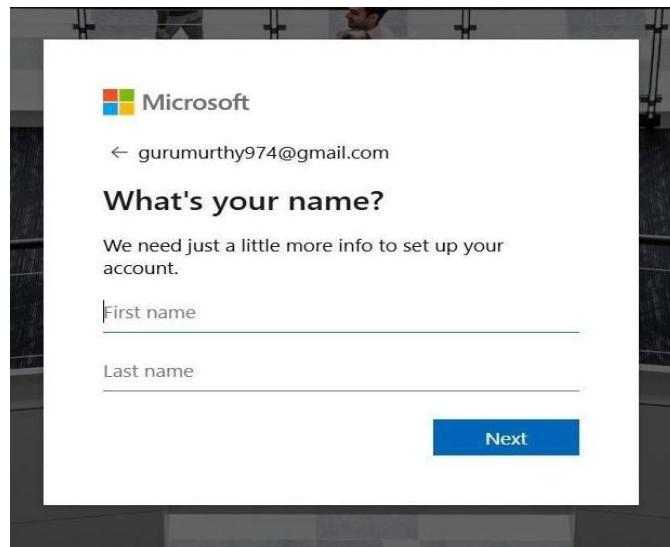
STEP 2: click on start free after that u get screen as below



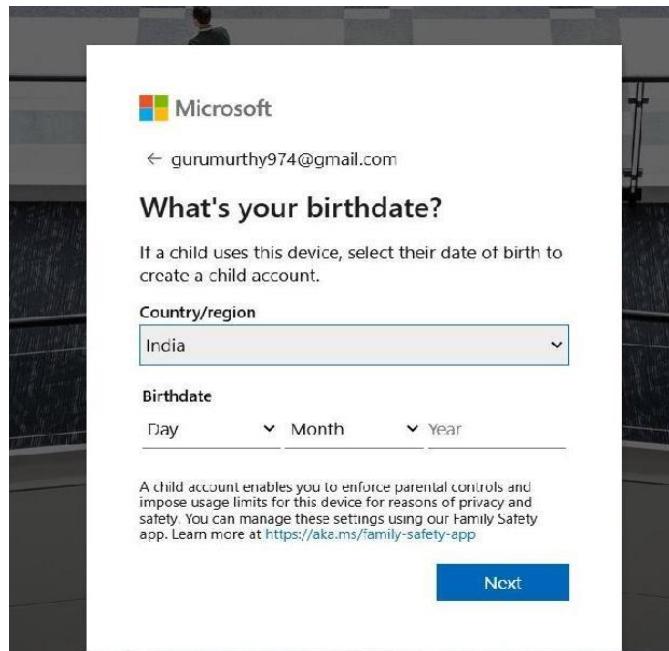
STEP 3: Provide your email id at place of create account



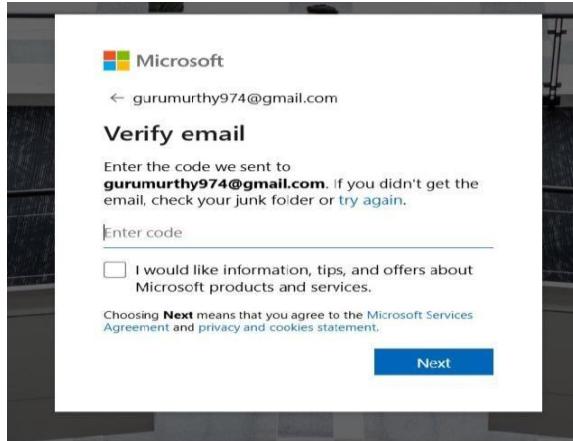
STEP 4: After password is set provide your first name last name



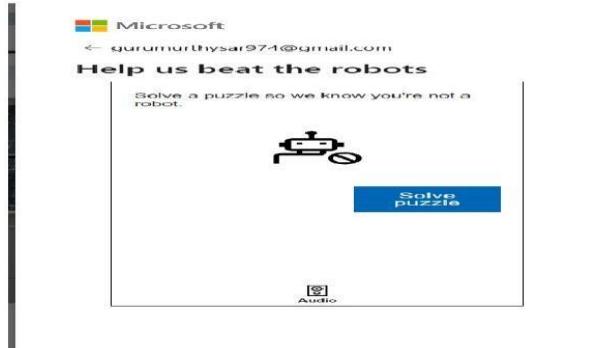
Then provide Country, Date of Birth



STEP 5 : Verification code be mailed to the mentioned once kindly type it



STEP 6: After code is verified as u got in the mail referred U be given an option to solve puzzle game



STEP 7: This step is the important once where u fill your Academic Details properly where u provide College email id as in provided by your Individual colleges later the verification code again comes

After proper college email is given u get verification mail with link to mail u have provided

Hello,

You have received this email because you recently requested verification via **Microsoft's Academic Verification** service. If you did not submit your email for this program, please disregard this email.

To complete your academic status verification, please click the link below. The link will automatically expire if not used within 5 days. After clicking the link, your verification status will be confirmed and you will return to the site.

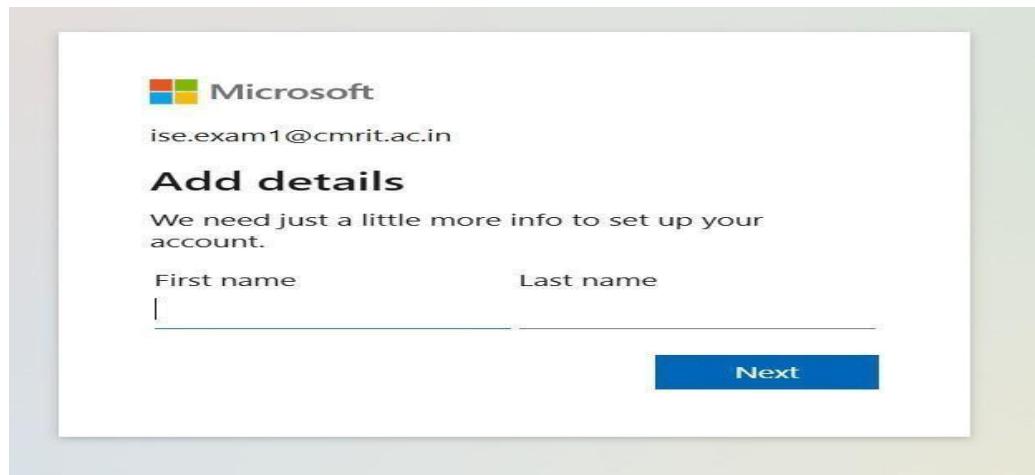
Navigate to: <https://verifyemail.microsoft.com/v1.0/tokenverification/verify?signature=YOQldNtLoPJ8nJy8QXzhbyR5Y5wAfwKVaNghX5hgX6Yb0hOygw5Aa5CsLqDGTWbHGIcSKjx37ROJFB7B0BogKgDTcsJ3ADCIXL9McGPZs97Ms2qbP54fxOA>

GGTedu7Y2SGVdpLBk98PiP33rVZf%2F2ES3z4lRBzM%2BjuRvGxNuTK%2B4lM2ifKm2Ksbp%2B1754BLzSJW4216NcWohhks8j%2Fd2qk7fgLurMrUhi8MaiPwQjLzyBmWbd9bjRY4PtX%2FfhwpOZTPGdROYFUpjyUf6vwYojsylbtBATxAI1se5jlzSAL2qdHWV03x8Yhm3gjN4TGYdxzjzLEOCGCn6h8GVjys%2BQEIO96N0Rvq1%62FnIccNbsdcmtAN0u9ivHcdpoHRnwX22Q58JgggCzyefDnDbFgxulteaHQnx%2BPkgKnOs2GOHSdeVVFc%2B0MBsBAQjBTfIHeI7XFkydahHXReTh8a77XxbRAQAdXGXifFW0caH5pUHafaNKxiOBM28dR%2BPgZB0n3kKFMIr20ltYQtxpSkOGa6DOssh38bumS2WrSD%3D

Thank You,
The Microsoft Academic Verification Team

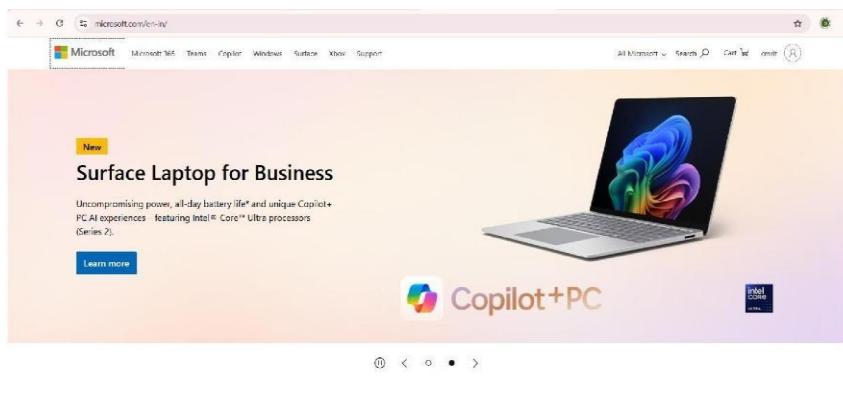
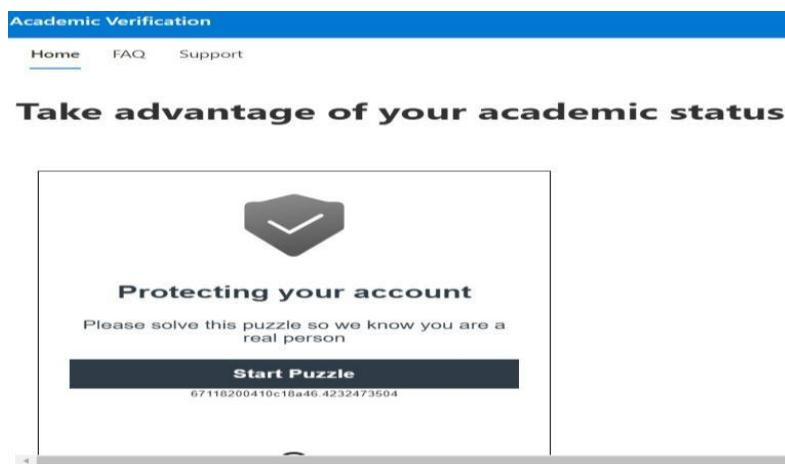
Click on link sent

After u Click The screen be as in Below

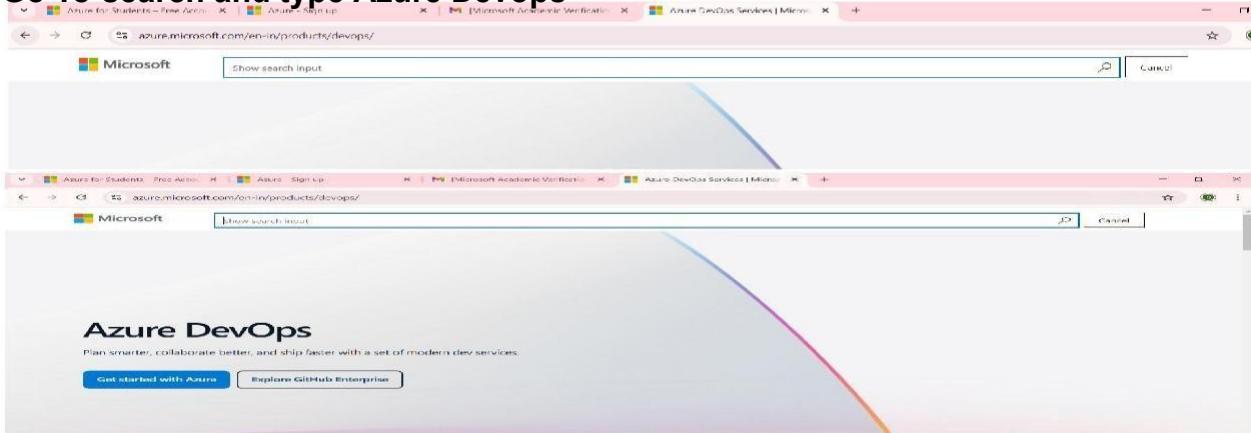


to

make ur account secure it again have puzzle

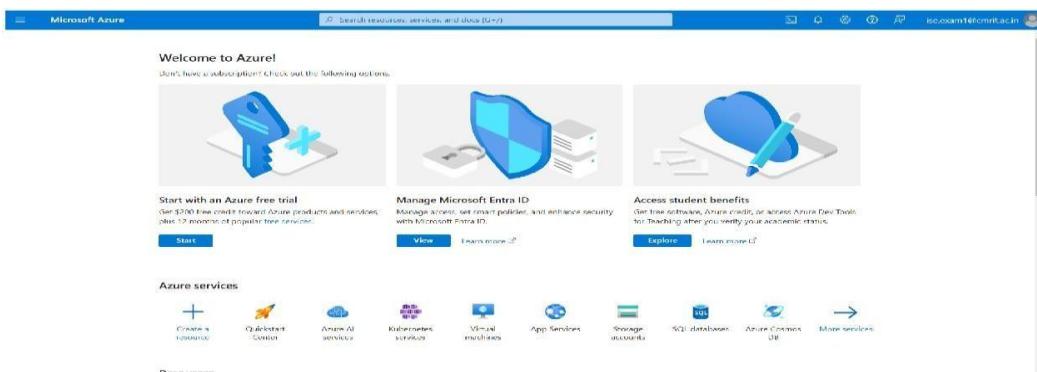


Go To search and type Azure Devops



Click on Get started with Azure

After the click u get the screen of Get free need not to do anything just click on Sign in You will get screen as in below



This is the home page of the Microsoft azure where you can see n number of services now our target is Azure Devops

In top where have search for services

Type Azure Devops

STEP 8: Select Azure Devops Organization



You will be able to see Organization is Created
Azure DevOps Organizations

Create new organization

✓ dev.azure.com/iseexam10557 (Owner)

Create a Team Project and start collaborating with your team now!

New project



Actions

Open in Visual Studio

Finally After Creating a New Organization

U can create Project of ur choice as per requirement

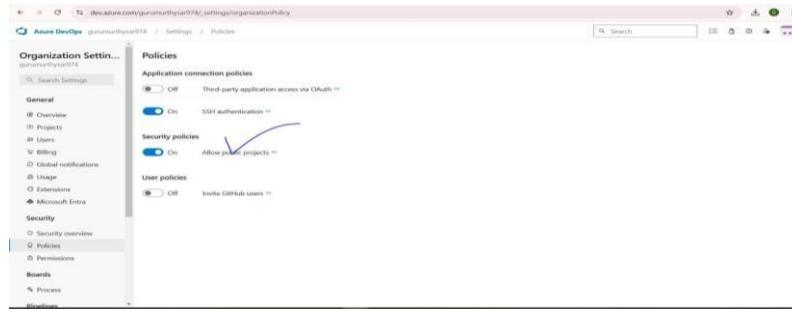
Every time u need not to sign in u can bookmark or add the below link as shortcut

<https://aex.dev.azure.com/> <https://portal.azure.com/#home>

PROGRAM 10:

Creating Build Pipelines: Building a Maven/Gradle Project with Azure Pipelines, Integrating Code Repositories (e.g., GitHub, Azure Repos), Running Unit Tests and Generating Reports.

STEP 1: On creating organization go to Organization settings go to Policy And Allow Public Projects active



STEP 2: Goto gitbash type commands as in below
mkdir maventest1
cd maventest1

STEP 3: to create simple hellow world maven project type command as in below
mvn archetype:generate -DgroupId=com.dineshonjava -DartifactId=Javateam -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false

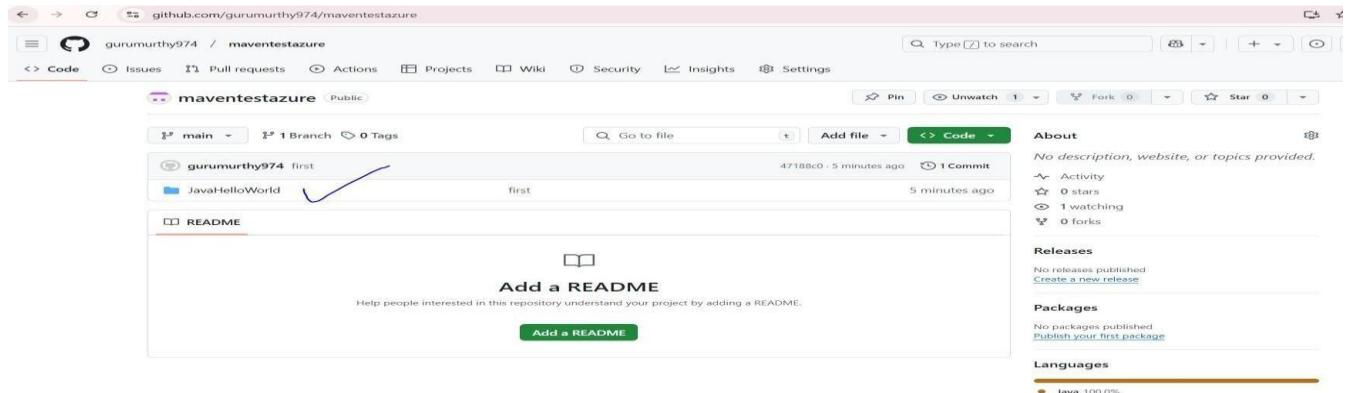
```

user@DESKTOP-VL3EFG5 MINGW64 ~% mvn archetype:generate -DgroupId=com.dineshonjava -DartifactId=JavaHelloWorld -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
[INFO] Scanning for projects...
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/3.1.2/maven-install-plugin-3.1.2.pom (8.5 kB at 9.8 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/3.1.2/maven-install-plugin-3.1.2.pom (8.5 kB at 9.8 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/42/maven-plugins-42.pom (7.7 kB at 60 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/42/maven-plugins-42.pom (7.7 kB at 60 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/3.1.2/maven-install-plugin-3.1.2.jar (32 kB at 360 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/3.1.2/maven-install-plugin-3.1.2.jar (32 kB at 360 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-deploy-plugin/3.1.2/maven-deploy-plugin-3.1.2.pom (9.6 kB at 271 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-deploy-plugin/3.1.2/maven-deploy-plugin-3.1.2.pom (9.6 kB at 271 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-deploy-plugin/3.1.2/maven-deploy-plugin-3.1.2.jar (40 kB at 964 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-assembly-plugin/3.7.1/maven-assembly-plugin-3.7.1.pom (15 kB at 524 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-assembly-plugin/3.7.1/maven-assembly-plugin-3.7.1.pom (15 kB at 524 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/41/maven-plugins-41.pom (7.4 kB at 104 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-assembly-plugin/3.7.1/maven-assembly-plugin-3.7.1.jar (240 kB at 1.6 MB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-dependency-plugin/3.7.0/maven-dependency-plugin-3.7.0.pom (19 kB at 301 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-dependency-plugin/3.7.0/maven-dependency-plugin-3.7.0.pom (19 kB at 301 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-dependency-plugin/3.7.0/maven-dependency-plugin-3.7.0.jar (207 kB at 4.2 MB/s)
[INFO]
[INFO] -----
[INFO] Building Maven Stub Project (No POM) 1
[INFO] [ pom ]
[INFO] -----
[INFO] >>> archetype:3.3.1:generate (default-cli) > generate-sources @ standalone-pom >>>
[INFO] <<< archetype:3.3.1:generate (default-cli) < generate-sources @ standalone-pom <<<
[INFO]
[INFO] -----
[INFO] -- archetype:3.3.1:generate (default-cli) @ standalone-pom --
[INFO] Generating project in Batch mode
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.0/maven-archetype-quickstart-1.0.jar (4.3 kB at 102 kB/s)
[INFO] -----
[INFO] using following parameters for creating project from Old (1.x) Archetype: maven-archetype-quickstart:1.0
[INFO] -----
[INFO] Parameter: basedir, Value: C:\Users\User\maventest1
[INFO] Parameter: package, Value: com.dineshonjava
[INFO] Parameter: groupId, Value: com.dineshonjava
[INFO] Parameter: artifactId, Value: JavaHelloWorld
[INFO] Parameter: packageName, Value: com.dineshonjava
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: C:\Users\User\maventest1\JavaHelloWorld
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 5.691 s
[INFO] Finished at: 2025-02-04T17:58:22+05:00
[INFO] -----

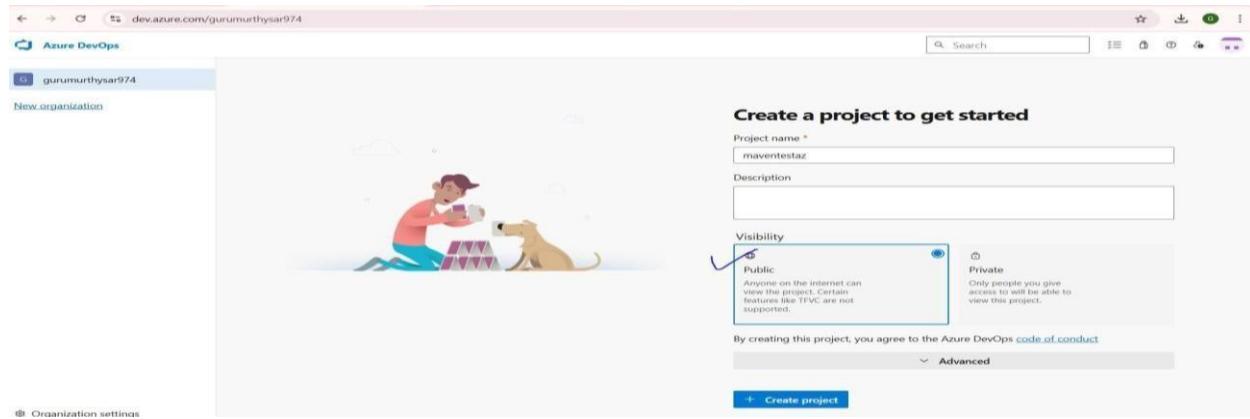
```

STEP 3: to add files from local to github Follow the procedure

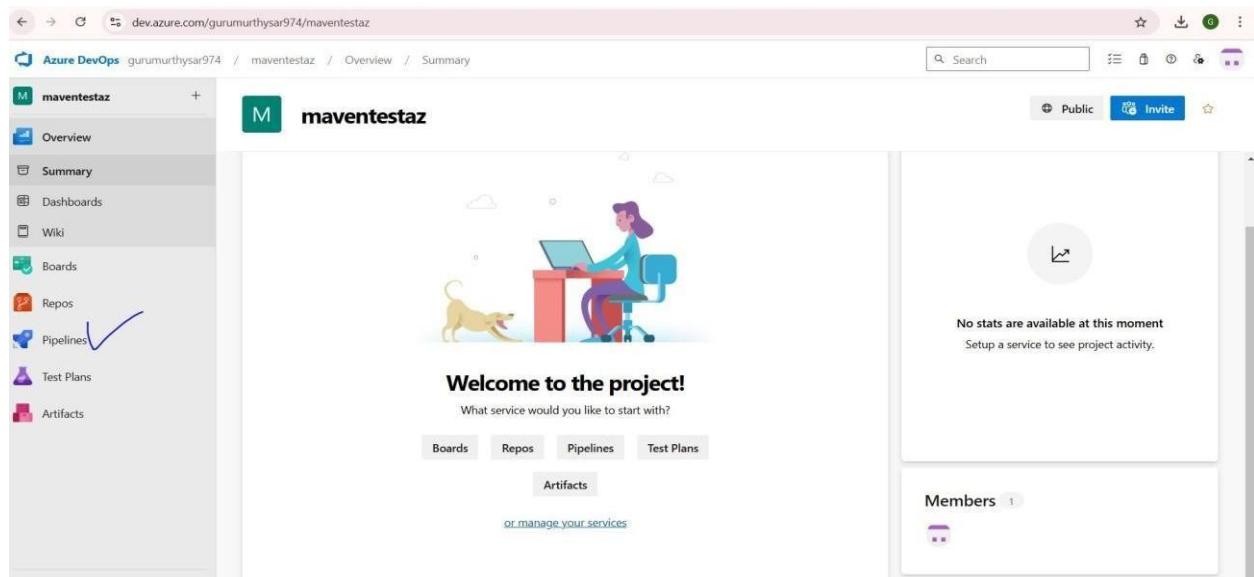
- First create a repository in github as maventestazure
- Then come to gitbash and type git init git add .



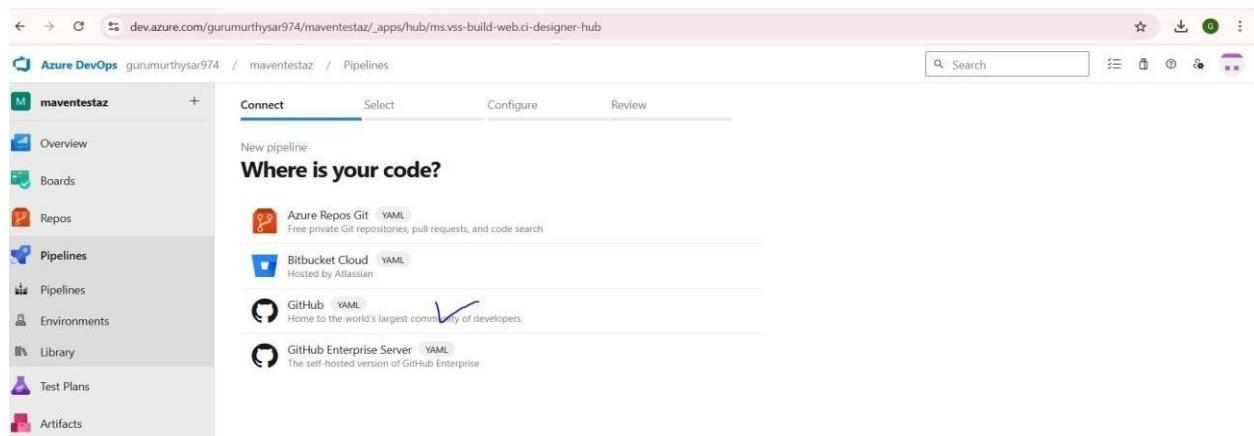
STEP 4: Now goto Azure Devops Organization create Public Project



STEP 5: Select pipeline and then click on create pipeline



STEP 6: After creating Pipeline select type of repo as Github



STEP 7: It asks for minimum signin verification after that ur screen be as in below select required repository there to run maven project in my case its maventest123

STEP 8: After required repo is selected the screen be as in below

Drag the screen down check once again the selected repository is correct or not then click on Approve and Install

STEP 9: It again verifies signin verification of microsoft account You be able to see starter pipeline select for Maven

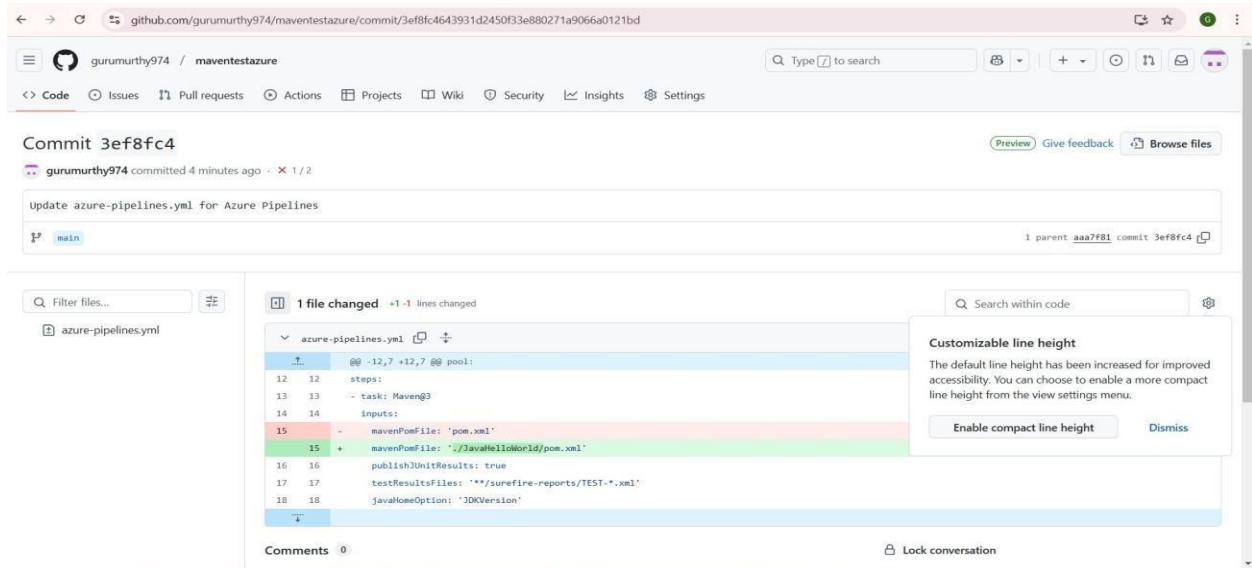
The screenshot shows the Azure DevOps Pipelines configuration interface. On the left, there's a sidebar with options like Overview, Boards, Repos, Pipelines (which is selected), Environments, Library, Test Plans, and Artifacts. The main area has tabs: Connect, Select, Configure (which is active), and Review. Below these tabs, it says 'New pipeline' and 'Configure your pipeline'. There are four options listed: 'Maven' (selected), 'Maven package Java project Web App to Linux on Azure', 'Starter pipeline', and 'Existing Azure Pipelines YAML file'. A blue checkmark is placed next to the 'Maven' option.

After selecting maven it asks for save and run just click on it

The screenshot shows the 'Review' tab of the Azure DevOps Pipelines configuration. The sidebar on the left is identical to the previous screenshot. The main area shows the 'Review your pipeline YAML' section. It displays the YAML code for a Maven pipeline. At the top right, there are buttons for 'Variables', 'Show assistant', and a prominent blue 'Save and run' button, which has a blue checkmark placed next to it.

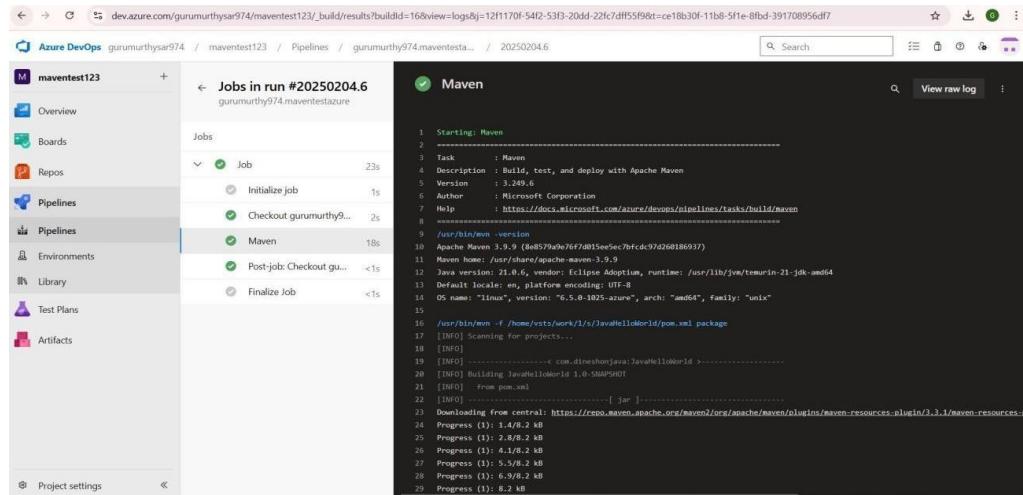
Finally You be able to see tasks running its failed bec we shld mention proper path For pom.xml

The screenshot shows the details of a pipeline run. At the top, it says 'Manually run by Gurumurthy Saralaya'. Below that, it shows 'Repository and version' as 'gurumurthy974/maventestazure' and 'main f6a6b586'. It also shows 'Time started and elapsed' as 'Just now' and '30s'. On the right, there are buttons for 'View change', 'Related', 'Tests and coverage', and 'Get started'. At the bottom, it shows 'Errors 2' and 'Warnings 2'.



The commits will also be visible in github

We can download and also see individual Raw Log Reports



Program 11:

Creating Release Pipelines: Deploying Applications to Azure App Services, Managing Secrets and Configuration with Azure Key Vault, Hands-On: Continuous Deployment with Azure Pipelines.

STEP1: Click on Organization setting and click on Pipeline Settings You get screen as in below

The screenshot shows the Azure DevOps Organization Settings page for the user 'gurumurthysar974'. The left sidebar lists General, Security, Boards, and Pipelines settings. The main area is titled 'Settings' under 'General' and contains several configuration options:

- Disable anonymous access to badges:** On
- Limit variables that can be set at queue time:** On
- Limit job authorization scope to current project for non-release pipelines:** On
- Limit job authorization scope to current project for release pipelines:** On
- Protect access to repositories in YAML pipelines:** On
- Disable stage chooser:** Off (highlighted with a blue oval)

STEP 2: Off the Disable creation of classic pipeline

The screenshot shows the same Azure DevOps Organization Settings page for the user 'gurumurthysar974'. The left sidebar lists General, Security, Boards, and Pipelines settings. The main area contains the following configuration options:

- Limit job authorization scope to current project for release pipelines:** On
- Protect access to repositories in YAML pipelines:** On
- Disable stage chooser:** Off
- Disable creation of classic build pipelines:** On (highlighted with a blue oval)
- Disable creation of classic release pipelines:** Off

STEP 3: Now you be able to see the visibility of Release for any pipeline creation as in screen below.

STEP 4: You can run simple test plans

We can build tasks and run them

