

MAVEN TASK

Chandan Gowda H S

- Maven is a powerful **project management tool** that is based on POM (project object model).
- It is used for projects build, dependency and documentation.
- Maven makes life of developer easy while creating reports, checks, build and testing automation setups.
- Maven is a popular open-source build tool developed by **the Apache Group**.
- Programming languages used: **Java**

Installation of maven in linux(ubuntu)

- Command to install maven
-- sudo apt install maven
- Then install an jdk
-- (sudo apt update)
-- sudo apt install default-jdk
- To check maven install
-- maven -version
- To check jdk install
-- java -version

Project Object Model (POM)

- It is fundamental unit of work in Maven.
- It is an XML file that resides in the base directory of the project as pom.xml.
- It is an XML file that resides in the base directory of the project as pom.xml.

- POM also contains the goals and plugins. While executing a task or goal, Maven looks for the POM in the current directory. It reads the POM, gets the needed configuration information, and then executes the goal. Some of the configuration that can be specified in the POM are following
 - * project dependencies, plugins, goals, build profiles, project version.

The minimum requirement for a POM are the following:

project root

modelVersion - should be set to 4.0.0

groupId - the id of the project's group.

ArtifactId - the id of the artifact (project)

version - the version of the artifact under the specified group

Example:

```
<project>
<modelVersion>4.0.0</modelVersion>
  <groupId>com.mycompany.app</groupId>
<artifactId>my-app</artifactId>
<version>1</version>
</project>
```

What is Build Lifecycle?

A Build Lifecycle is a well-defined sequence of phases, which define the order in which the goals are to be executed. Here phase represents a stage in life cycle. As an example, a typical Maven Build Lifecycle consists of the following sequence of phases.

Phases of Lifecycle

1. Validate
2. Compile
3. Test
4. Package
5. Install
6. Deploy

Maven Validate

- Validating the information
- Validates if the project is correct and if all necessary information is available.
- This step validates if the project structure is correct.
- For example – It checks if all the dependencies have been downloaded and are available in the local repository.

Syntax – mvn validate

Maven Compile

- ◆ Syntax- mvn compile
- ◆ It compiles the source code, converts the .java files to .class and stores the classes in target/classes folder.

Maven Test

- Syntax- mvn test
- It runs unit tests for the project and stores the test-classes in - target/test-classes folder.
- For example – pom.xml using Junit plugin to unit test the code

Maven Package

- Syntax- `mvn package`
- This step packages the compiled code in distributable format like JAR or WAR.
- A JAR (Java ARchive) is a package file format typically used to aggregate many Java class files and associated metadata and resources into one file for distribution
- in this step it stores the package into target directory

Maven Install

- Syntax- `mvn install`
- This step installs the packaged code to the local Maven repository.

Maven deploy

- Syntax – `mvn deploy`
- done in an integration or release environment,
- copies the final package to the remote repository for sharing with other developers and projects

Maven site

- **Syntax** – `mvn site`
- generates site documentation for this project

Maven clean

- Syntax – `mvn site`
- cleans up artifacts created by prior builds

Maven Example problem

- ◆ Create Ec2 of ubuntu and install git and maven
 - sudo apt update
 - sudo apt install git
 - sudo apt install maven (it will install java 11 also)
- ◆ Then clone repository into ubuntu ec2
 - sudo git clone <url>

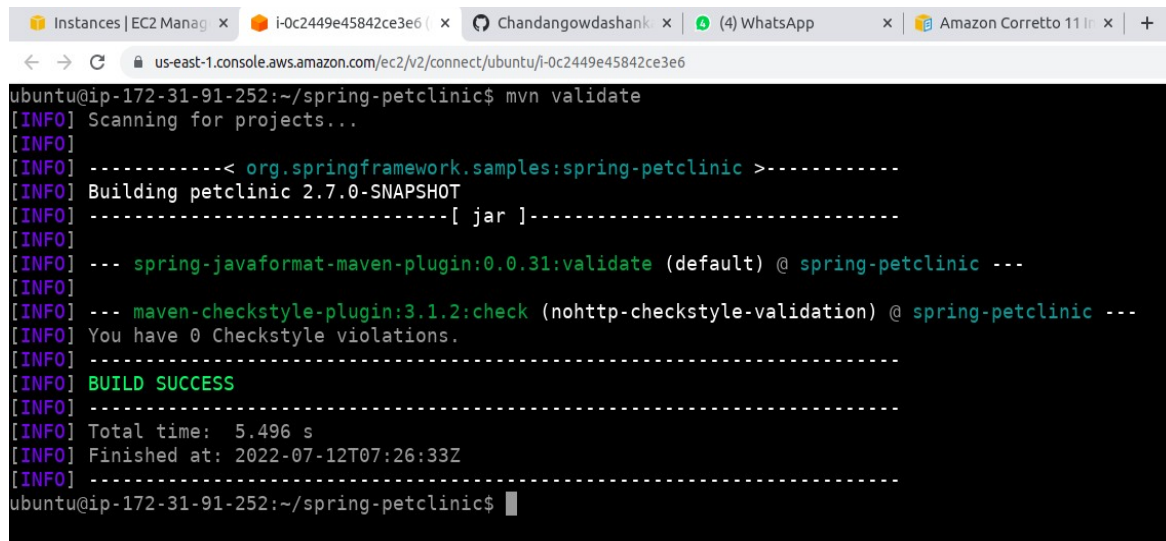
example sudo git clone <https://github.com/spring-projects/spring-petclinic.git>

it will clone the spring-petclinic named folder
- ◆ change the directory to that folder
 - cd spring-petclinic

```
Connect to instance | EC2 x i-003b4cc2a8193af4b (Doc x Chandangowdashankar/ x +
us-east-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-003b4cc2a8193af4b
[ec2-user@ip-172-31-85-122 maventask]$ ls
[ec2-user@ip-172-31-85-122 maventask]$ git clone https://github.com/Chandangowdashankar/petclinictask.git
fatal: could not create work tree dir 'petclinictask': Permission denied
[ec2-user@ip-172-31-85-122 maventask]$ sudo git clone https://github.com/Chandangowdashankar/petclinictask.git
Cloning into 'petclinictask'...
remote: Enumerating objects: 328, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 328 (delta 0), reused 0 (delta 0), pack-reused 325
Receiving objects: 100% (328/328), 46.97 MiB | 33.66 MiB/s, done.
Resolving deltas: 100% (102/102), done.
[ec2-user@ip-172-31-85-122 maventask]$ ls
petclinictask
[ec2-user@ip-172-31-85-122 maventask]$ cd petclinictask/
[ec2-user@ip-172-31-85-122 petclinictask]$ ls
build.gradle  gradle  gradlew.bat  mvnw  pom.xml  settings.gradle  target
docker-compose.yml  gradlew  LICENSE.txt  mvnw.cmd  readme.md  src
[ec2-user@ip-172-31-85-122 petclinictask]$ mvn
```

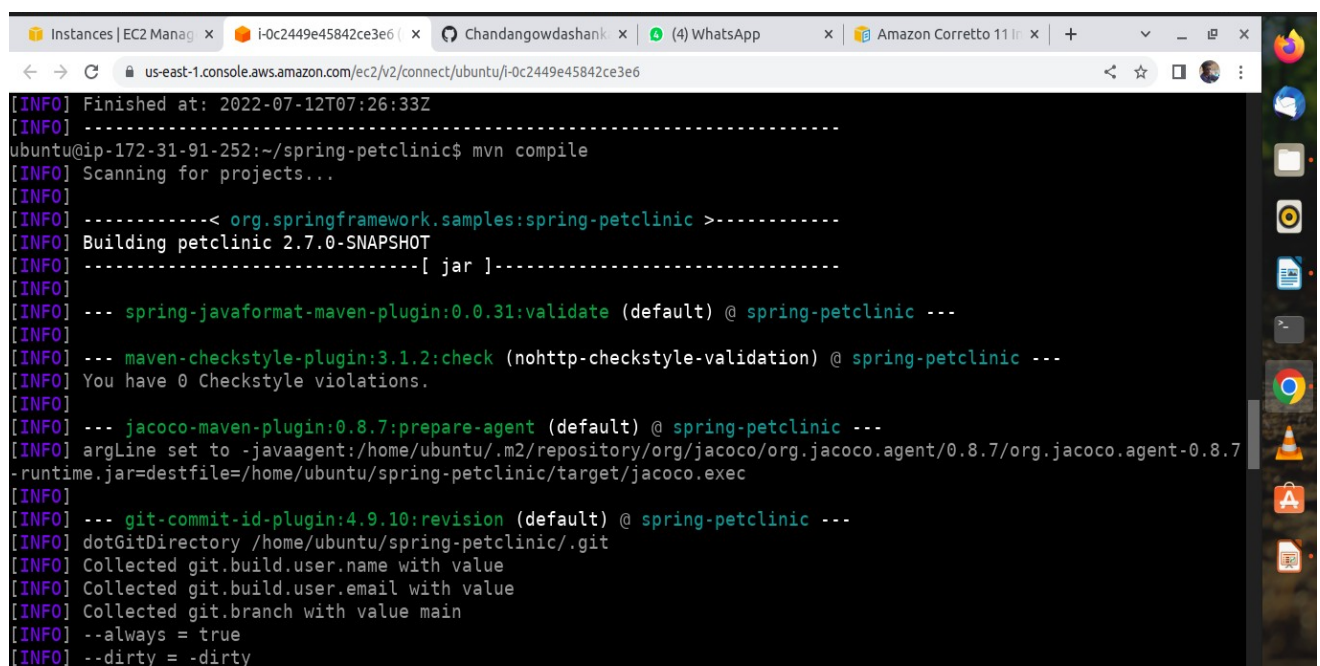
◆ Then we can achieve the maven job

➤ mvn validate – Validating the information



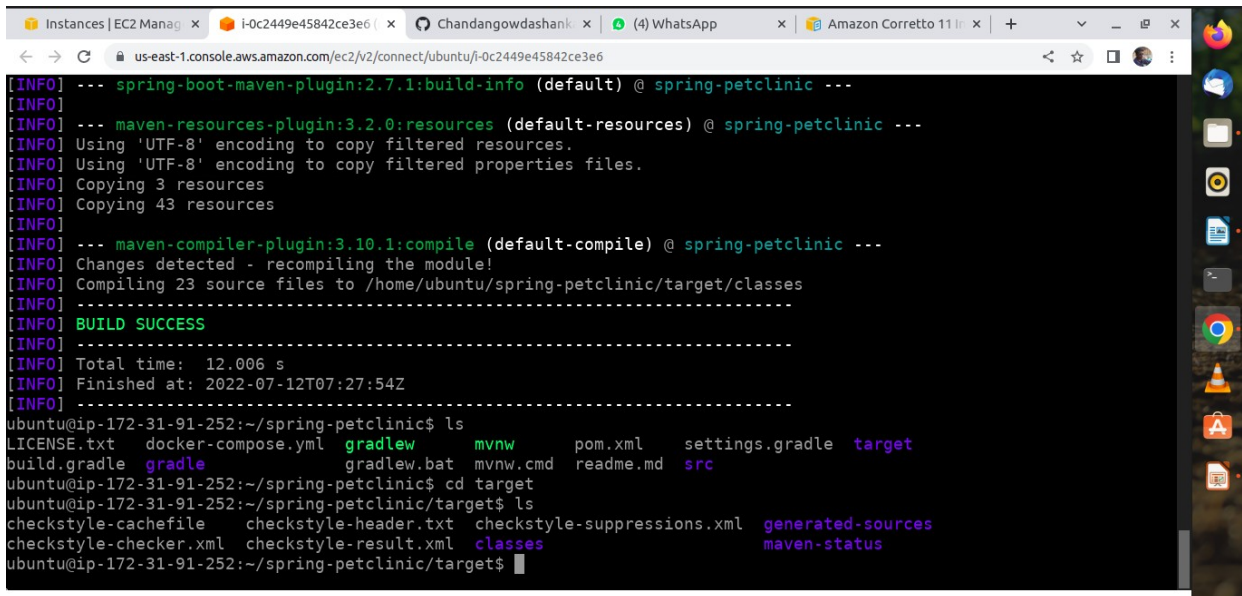
```
ubuntu@ip-172-31-91-252:~/spring-petclinic$ mvn validate
[INFO] Scanning for projects...
[INFO]
[INFO] -----< org.springframework.samples:spring-petclinic >-----
[INFO] Building petclinic 2.7.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- spring-javaformat-maven-plugin:0.0.31:validate (default) @ spring-petclinic ---
[INFO]
[INFO] --- maven-checkstyle-plugin:3.1.2:check (nohttp-checkstyle-validation) @ spring-petclinic ---
[INFO] You have 0 Checkstyle violations.
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 5.496 s
[INFO] Finished at: 2022-07-12T07:26:33Z
[INFO]
ubuntu@ip-172-31-91-252:~/spring-petclinic$
```

➤ mvn compile – compile java source code and store it into



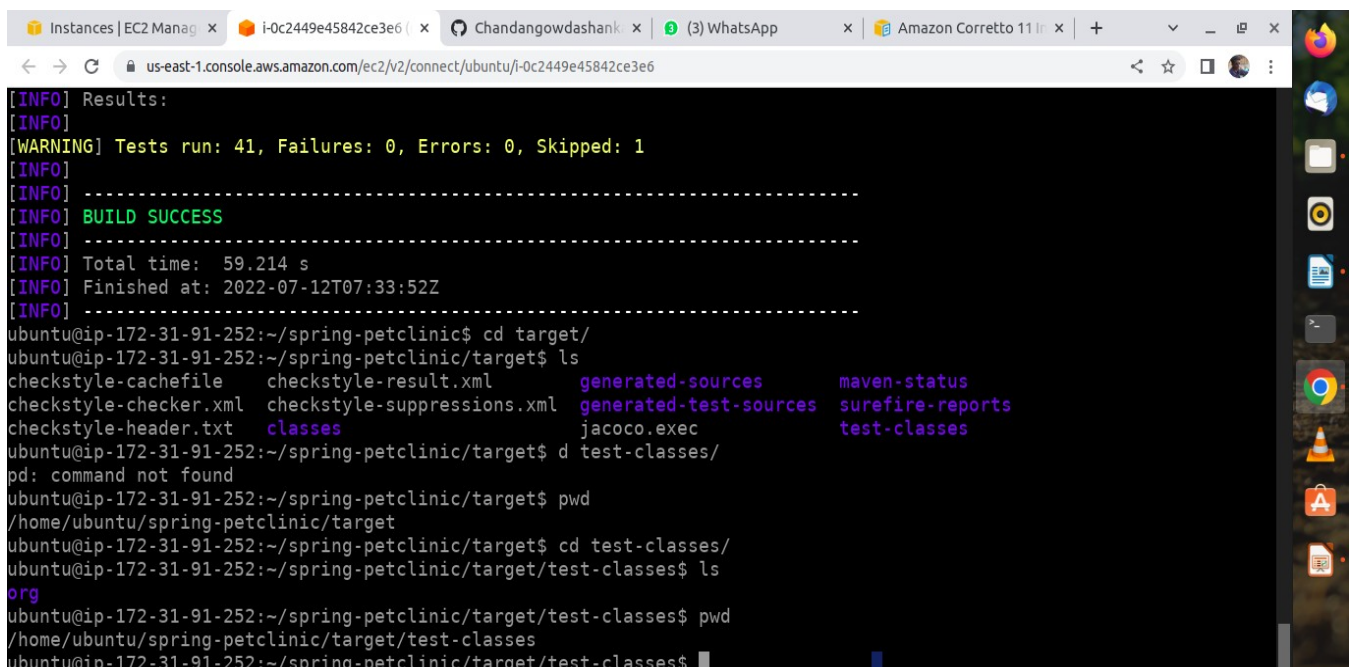
```
[INFO] Finished at: 2022-07-12T07:26:33Z
[INFO]
ubuntu@ip-172-31-91-252:~/spring-petclinic$ mvn compile
[INFO] Scanning for projects...
[INFO]
[INFO] -----< org.springframework.samples:spring-petclinic >-----
[INFO] Building petclinic 2.7.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- spring-javaformat-maven-plugin:0.0.31:validate (default) @ spring-petclinic ---
[INFO]
[INFO] --- maven-checkstyle-plugin:3.1.2:check (nohttp-checkstyle-validation) @ spring-petclinic ---
[INFO] You have 0 Checkstyle violations.
[INFO]
[INFO] --- jacoco-maven-plugin:0.8.7:prepare-agent (default) @ spring-petclinic ---
[INFO] argLine set to -javaagent:/home/ubuntu/.m2/repository/org/jacoco/org.jacoco.agent/0.8.7/org.jacoco.agent-0.8.7
-runtime.jar=destfile=/home/ubuntu/spring-petclinic/target/jacoco.exec
[INFO]
[INFO] --- git-commit-id-plugin:4.9.10:revision (default) @ spring-petclinic ---
[INFO] dotGitDirectory /home/ubuntu/spring-petclinic/.git
[INFO] Collected git.build.user.name with value
[INFO] Collected git.build.user.email with value
[INFO] Collected git.branch with value main
[INFO] --always = true
[INFO] --dirty = -dirty
```

- it will create target file and add the class file in Classes folder



```
[INFO] --- spring-boot-maven-plugin:2.7.1:build-info (default) @ spring-petclinic ---
[INFO] --- maven-resources-plugin:3.2.0:resources (default-resources) @ spring-petclinic ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Using 'UTF-8' encoding to copy filtered properties files.
[INFO] Copying 3 resources
[INFO] Copying 43 resources
[INFO] --- maven-compiler-plugin:3.10.1:compile (default-compile) @ spring-petclinic ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 23 source files to /home/ubuntu/spring-petclinic/target/classes
[INFO] BUILD SUCCESS
[INFO] Total time: 12.006 s
[INFO] Finished at: 2022-07-12T07:27:54Z
ubuntu@ip-172-31-91-252:~/spring-petclinic$ ls
LICENSE.txt  docker-compose.yml  gradlew  mvnw  pom.xml  settings.gradle  target
build.gradle  gradle  gradlew.bat  mvnw.cmd  readme.md  src
ubuntu@ip-172-31-91-252:~/spring-petclinic$ cd target
ubuntu@ip-172-31-91-252:~/spring-petclinic/target$ ls
checkstyle-cachefile  checkstyle-header.txt  checkstyle-suppressions.xml  generated-sources
checkstyle-checker.xml  checkstyle-result.xml  classes  maven-status
ubuntu@ip-172-31-91-252:~/spring-petclinic/target$
```

- Then mvn test will done unit test on our project and store test classes file in /home/ubuntu/spring-petclinic/target/test-classes

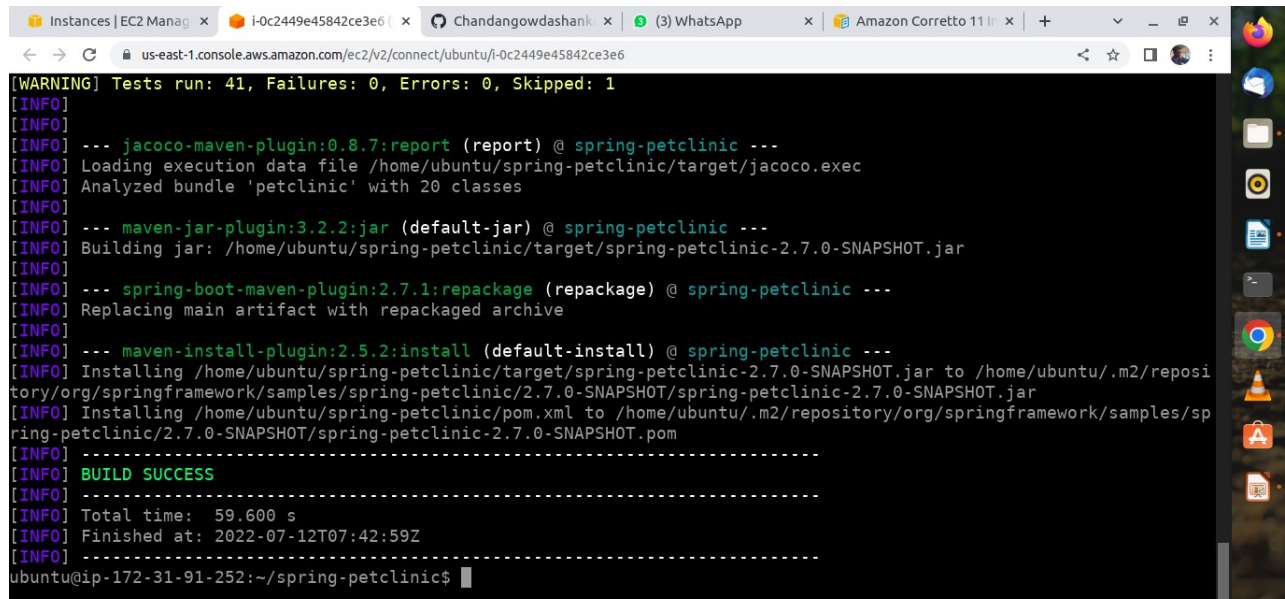


```
[INFO] Results:
[INFO]
[WARNING] Tests run: 41, Failures: 0, Errors: 0, Skipped: 1
[INFO]
[INFO] BUILD SUCCESS
[INFO] Total time: 59.214 s
[INFO] Finished at: 2022-07-12T07:33:52Z
ubuntu@ip-172-31-91-252:~/spring-petclinic$ cd target/
ubuntu@ip-172-31-91-252:~/spring-petclinic/target$ ls
checkstyle-cachefile  checkstyle-result.xml  generated-sources  maven-status
checkstyle-checker.xml  checkstyle-suppressions.xml  generated-test-sources  surefire-reports
checkstyle-header.txt  classes  jacoco.exec  test-classes
ubuntu@ip-172-31-91-252:~/spring-petclinic/target$ d test-classes/
pd: command not found
ubuntu@ip-172-31-91-252:~/spring-petclinic/target$ pwd
/home/ubuntu/spring-petclinic/target
ubuntu@ip-172-31-91-252:~/spring-petclinic/target$ cd test-classes/
ubuntu@ip-172-31-91-252:~/spring-petclinic/target/test-classes$ ls
org
ubuntu@ip-172-31-91-252:~/spring-petclinic/target/test-classes$ pwd
/home/ubuntu/spring-petclinic/target/test-classes
ubuntu@ip-172-31-91-252:~/spring-petclinic/target/test-classes$
```

- Then mvn package it will create jar/war file within target
- based on our pom.xml data it creates jar file

```
[INFO] --- jacoco-maven-plugin:0.8.7:report (report) @ spring-petclinic ---
[INFO] Loading execution data file /home/ubuntu/spring-petclinic/target/jacoco.exec
[INFO] Analyzed bundle 'petclinic' with 20 classes
[INFO]
[INFO] --- maven-jar-plugin:3.2.2:jar (default-jar) @ spring-petclinic ---
[INFO] Building jar: /home/ubuntu/spring-petclinic/target/spring-petclinic-2.7.0-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:2.7.1:repackage (repackage) @ spring-petclinic ---
[INFO] Replacing main artifact with repackaged archive
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 59.878 s
[INFO] Finished at: 2022-07-12T07:36:45Z
[INFO]
ubuntu@ip-172-31-91-252:~/spring-petclinic$ ls
LICENSE.txt  docker-compose.yml  gradlew  mvnw  pom.xml  settings.gradle  target
build.gradle  gradle  gradlew.bat  mvnw.cmd  readme.md  src
ubuntu@ip-172-31-91-252:~/spring-petclinic$ cd target/
ubuntu@ip-172-31-91-252:~/spring-petclinic/target$ ls
checkstyle-cachefile  checkstyle-suppressions.xml  jacoco.exec  spring-petclinic-2.7.0-SNAPSHOT.jar
checkstyle-checker.xml  classes  maven-archiver  spring-petclinic-2.7.0-SNAPSHOT.jar.original
checkstyle-header.txt  generated-sources  maven-status  surefire-reports
checkstyle-result.xml  generated-test-sources  site  test-classes
ubuntu@ip-172-31-91-252:~/spring-petclinic/target$
```

- in here it will create spring-petclinic-2.7.0-SNAPSHOT.jar
- we can run this using this command
java -jar <jar file name>
- mvn install
This step installs the packaged code to the local Maven repository.

A screenshot of an AWS EC2 console terminal window. The browser tabs at the top include 'Instances | EC2 Manag...', 'i-0c2449e45842ce3e6', 'Chandangowdashank', '(3) WhatsApp', and 'Amazon Corretto 11'. The terminal output shows the following: [WARNING] Tests run: 41, Failures: 0, Errors: 0, Skipped: 1; [INFO] --- jacoco-maven-plugin:0.8.7:report (report) @ spring-petclinic ---; [INFO] Loading execution data file /home/ubuntu/spring-petclinic/target/jacoco.exec; [INFO] Analyzed bundle 'petclinic' with 20 classes; [INFO] --- maven-jar-plugin:3.2.2:jar (default-jar) @ spring-petclinic ---; [INFO] Building jar: /home/ubuntu/spring-petclinic/target/spring-petclinic-2.7.0-SNAPSHOT.jar; [INFO] --- spring-boot-maven-plugin:2.7.1:repackage (repackage) @ spring-petclinic ---; [INFO] Replacing main artifact with repackaged archive; [INFO] --- maven-install-plugin:2.5.2:install (default-install) @ spring-petclinic ---; [INFO] Installing /home/ubuntu/spring-petclinic/target/spring-petclinic-2.7.0-SNAPSHOT.jar to /home/ubuntu/.m2/repository/org/springframework/samples/spring-petclinic/2.7.0-SNAPSHOT/spring-petclinic-2.7.0-SNAPSHOT.jar; [INFO] Installing /home/ubuntu/spring-petclinic/pom.xml to /home/ubuntu/.m2/repository/org/springframework/samples/spring-petclinic/2.7.0-SNAPSHOT/spring-petclinic-2.7.0-SNAPSHOT.pom; [INFO] BUILD SUCCESS; [INFO] Total time: 59.600 s; [INFO] Finished at: 2022-07-12T07:42:59Z; [INFO] The prompt is ubuntu@ip-172-31-91-252:~/spring-petclinic\$.

- It will install all the package into local that means to ubuntu ec2
- in the path of /home/ubuntu/spring-petclinic/target/spring-petclinic-2.7.0-SNAPSHOT.jar to /home/ubuntu/.m2
- it used in next build time
- mvn clean
- cleans up artifacts created by prior builds

Thank you