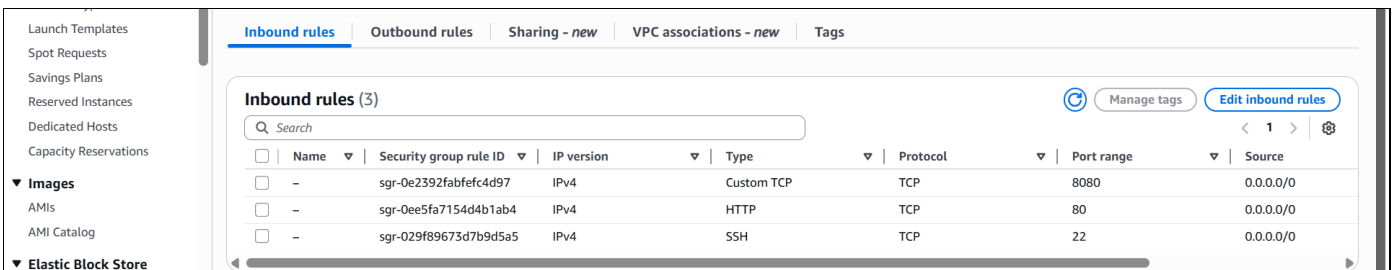
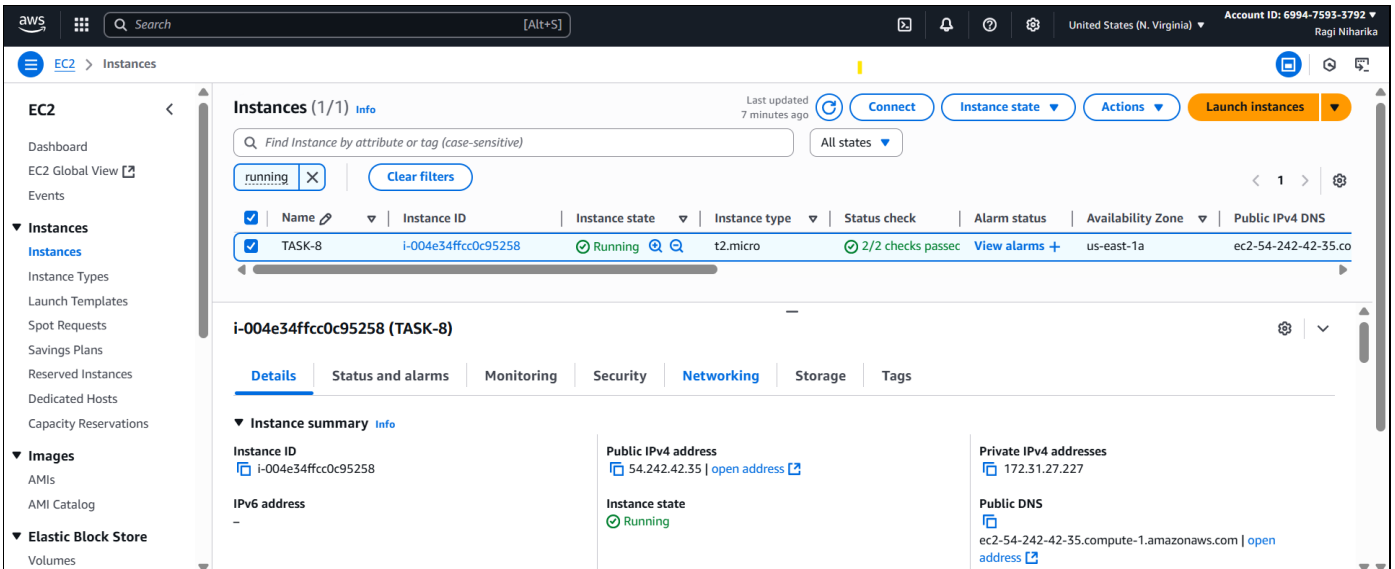


TASK 8: Run a Simple Java Maven Build Job in Jenkins

Step 1: Launch an EC2 Instance

1. Log in to your **AWS Console** → EC2 → Launch Instance.
2. Choose **Ubuntu Server 22.04 LTS (Free tier eligible)**.
3. Select instance type: **t2.micro** (Free tier).
4. Configure security group:
 - Allow **SSH (22)** → My IP.
 - Allow **HTTP (80)** → Anywhere (optional).
 - Allow **Custom TCP 8080** → Anywhere (needed for Jenkins UI).
5. Launch instance with a key pair.

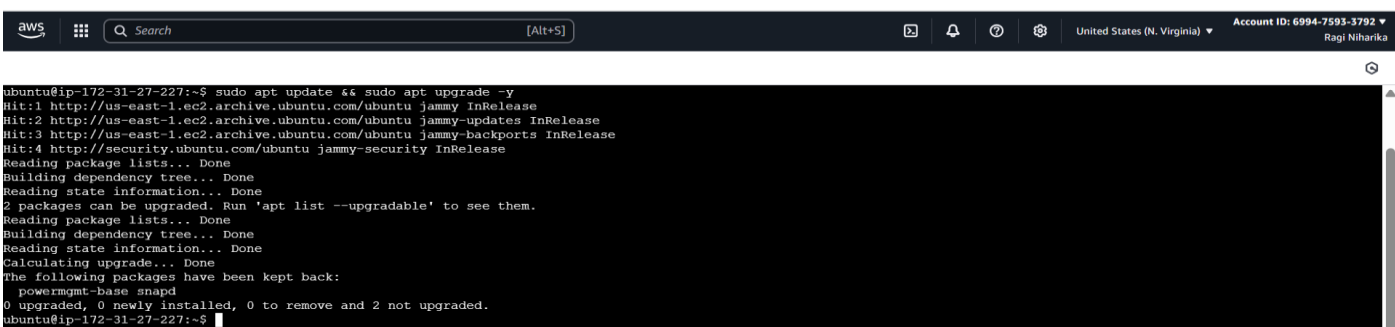


Step 2: Connect to EC2

`ssh -i your-key.pem ubuntu@<EC2-Public-IP>`

Step 3: Install Dependencies

- Update packages:
- `sudo apt update && sudo apt upgrade -y`



TASK 8: Run a Simple Java Maven Build Job in Jenkins

- Install Java, Maven, Git, Docker:
- `sudo apt install -y openjdk-11-jdk maven git docker.io`

```
aws [Search] [Alt+S] United States (N. Virginia) Account ID: 6994-7593-3792 Ragi Niharika

Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1029-aws x86_64)

ubuntu@ip-172-31-27-227:~$ sudo apt install -y openjdk-11-jdk maven git docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.34.1-lubuntu1.15).
```

- Check versions:
- `java -version`
- `mvn -version`
- `docker --version`

```
aws [Search] [Alt+S] United States (N. Virginia) Account ID: 6994-7593-3792 Ragi Niharika

ubuntu@ip-172-31-27-227:~$ java -version
openjdk version "11.0.28" 2025-07-15
OpenJDK Runtime Environment (build 11.0.28+6-post-Ubuntu-lubuntu122.04.1)
OpenJDK 64-Bit Server VM (build 11.0.28+6-post-Ubuntu-lubuntu122.04.1, mixed mode, sharing)
ubuntu@ip-172-31-27-227:~$ mvn -version
Apache Maven 3.6.3
Maven home: /usr/share/maven
Java version: 11.0.28, vendor: Ubuntu, runtime: /usr/lib/jvm/java-11-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "6.8.0-1029-aws" arch: "amd64", family: "unix"
ubuntu@ip-172-31-27-227:~$ docker --version
Docker version 27.5.1, build 27.5.1-0ubuntu~22.04.2
ubuntu@ip-172-31-27-227:~$
```

Step 4: Run Jenkins in Docker

- `sudo systemctl start docker`
- `sudo systemctl enable docker`
- `sudo usermod -aG docker ubuntu`
- `newgrp docker`

```
9 sudo systemctl start docker
10 sudo systemctl enable docker
11 sudo systemctl status docker
12 sudo usermod -aG docker ubuntu
13 newgrp docker
```

Run Jenkins container:

```
docker run -d -p 8080:8080 -p 50000:50000 --name jenkins \
-v jenkins_home:/var/jenkins_home jenkins/jenkins:its
```

```
ubuntu@ip-172-31-27-227:~$ docker run -d -p 8080:8080 -p 50000:50000 --name jenkins \
-v jenkins_home:/var/jenkins_home jenkins/jenkins:its
Unable to find image 'jenkins/jenkins:its' locally
its: Pulling from jenkins/jenkins
ebed137c7c18: Pull complete
a16eed992861: Pull complete
d0340747bfc5: Pull complete
b3190e29bcb4: Pull complete
9fa62d58a0db: Pull complete
31a9a2be77c3: Pull complete
9cd347526f2b: Pull complete
654460caae81: Pull complete
6ce3758f4dbb: Pull complete
2b1923fe83ed: Pull complete
a3391befb457: Pull complete
48e332b529b0: Pull complete
Digest: sha256:0e66af30c9272490ba18757d5d4d41e4ac2160278ae40b69d6da9b5adbe98794
Status: Downloaded newer image for jenkins/jenkins:its
a26fc6b72eb8087075a29f18b58c2135ddb502402f03c4d5150be9421b51800
ubuntu@ip-172-31-27-227:~$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
jenkins/jenkins its 627182afbe2b 3 weeks ago 472MB
ubuntu@ip-172-31-27-227:~$
```

Get Jenkins admin password:

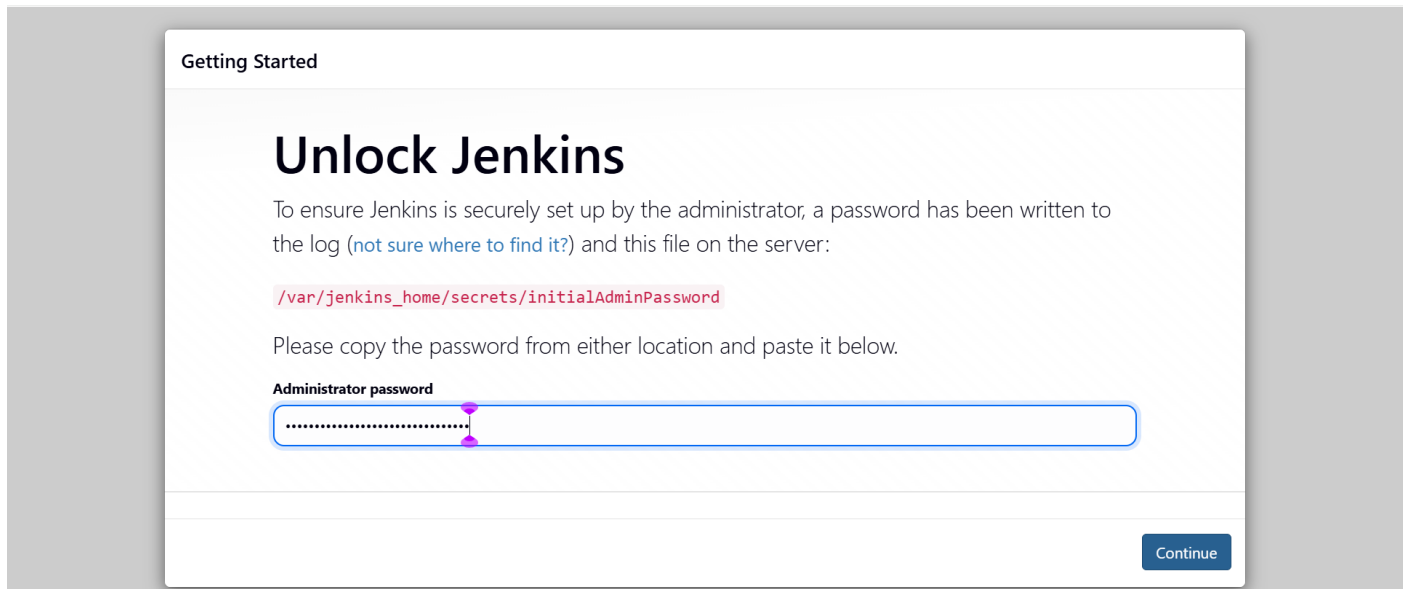
TASK 8: Run a Simple Java Maven Build Job in Jenkins

`docker exec -it jenkins cat /var/jenkins_home/secrets/initialAdminPassword`

```
ubuntu@ip-172-31-27-227:~$ docker exec -it jenkins cat /var/jenkins_home/secrets/initialAdminPassword
9eefd5302d394d3fb04197e56a9387f1
ubuntu@ip-172-31-27-227:~$
```

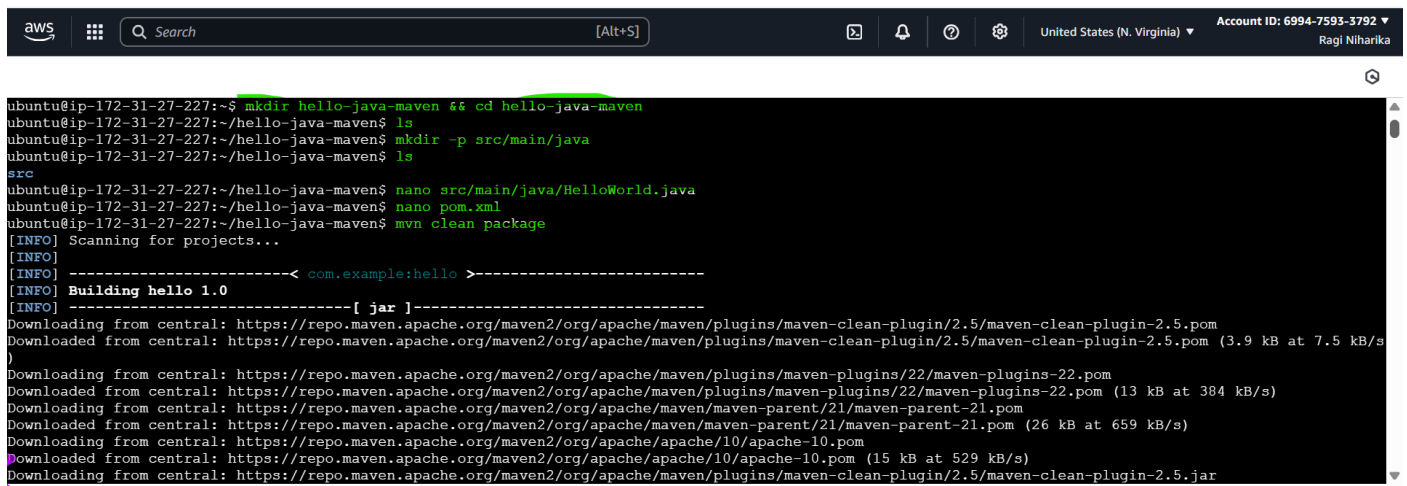
Open Jenkins UI in browser:

`http://<EC2-Public-IP>:8080`



Step 5: Prepare Java App (hello-java-maven)

- On EC2, create a project folder:
- `mkdir hello-java-maven && cd hello-java-maven`
- Create **HelloWorld.java**:
- `mkdir -p src/main/java`
- `nano src/main/java/HelloWorld.java`
- Create **pom.xml**:
- `nano pom.xml`
- Test locally:
- `mvn clean package`
- `java -cp target/hello-1.0.jar HelloWorld`



TASK 8: Run a Simple Java Maven Build Job in Jenkins

```
aws [Search] [Alt+S] United States (N. Virginia) Account ID: 6994-7593-3792 Ragi Niharika

GNU nano 6.2 pom.xml

<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.example</groupId>
  <artifactId>hello</artifactId>
  <version>1.0</version>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.8.1</version>
        <configuration>
          <source>1.8</source>
          <target>1.8</target>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^R Read ^C Location ^M-U Undo ^M-A Set Mark ^M-I To Bracket ^M

public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, Jenkins + Maven!");
    }
}

Downloading from central: https://repo.maven.apache.org/maven2/commons-lang/commons-lang/2.1/commons-lang-2.1.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-io/2.0.2/plexus-io-2.0.2.jar (58 kB at 1.3 MB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-archiver/2.5/maven-archiver-2.5.jar (22 kB at 358 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-interpolation/1.15/plexus-interpolation-1.15.jar (60 kB at 765 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-archiver/2.1/plexus-archiver-2.1.jar (184 kB at 2.0 MB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/commons-lang/commons-lang/2.1/commons-lang-2.1.jar (208 kB at 2.1 MB/s)
[INFO] Building jar: /home/ubuntu/hello-java-maven/target/hello-1.0.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 8.931 s
[INFO] Finished at: 2025-08-15T15:38:24Z
[INFO] -----
ubuntu@ip-172-31-27-227:~/hello-java-maven$ java -cp target/hello-1.0.jar HelloWorld
Hello, Jenkins + Maven!
ubuntu@ip-172-31-27-227:~/hello-java-maven$
```

Step 6: Perform Git Operations

```
ubuntu@ip-172-31-27-227:~$ ls
hello-java-maven
ubuntu@ip-172-31-27-227:~$ cd hello-java-maven
ubuntu@ip-172-31-27-227:~/hello-java-maven$ ls
pom.xml src target
ubuntu@ip-172-31-27-227:~/hello-java-maven$ git add .
fatal: not a git repository (or any of the parent directories): .git
ubuntu@ip-172-31-27-227:~/hello-java-maven$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/ubuntu/hello-java-maven/.git/
ubuntu@ip-172-31-27-227:~/hello-java-maven$ git branch
ubuntu@ip-172-31-27-227:~/hello-java-maven$ git add .
ubuntu@ip-172-31-27-227:~/hello-java-maven$ git commit -m "maven job commits"
ubuntu@ip-172-31-27-227:~/hello-java-maven$ git commit -m "maven job commits"
[master (root-commit) 2b1a61] maven job commits
Committer: Ubuntu <ubuntu@ip-172-31-27-227.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

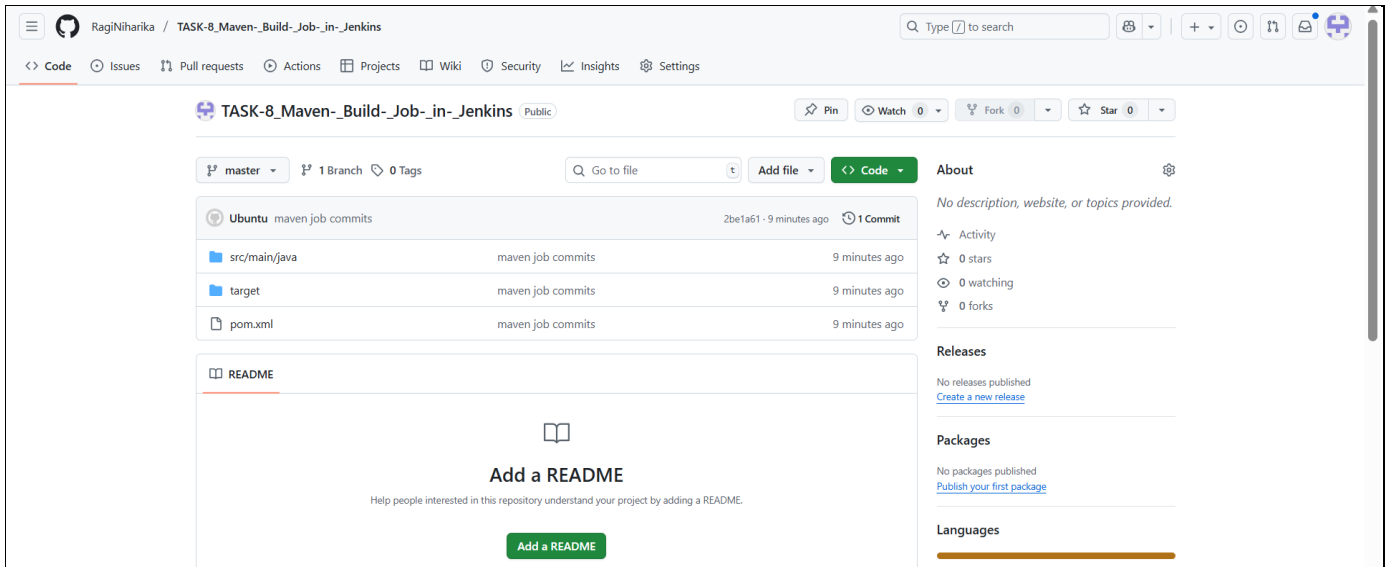
After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

7 files changed, 31 insertions(+)
create mode 100644 pom.xml
create mode 100644 src/main/java/HelloWorld.java
create mode 100644 target/classes/HelloWorld.class
create mode 100644 target/hello-1.0.jar
create mode 100644 target/maven-archiver/pom.properties
create mode 100644 target/maven-status/maven-compiler-plugin/compile/default-compile/createdFiles.lst
create mode 100644 target/maven-status/maven-compiler-plugin/compile/default-compile/inputFiles.lst
ubuntu@ip-172-31-27-227:~/hello-java-maven$ git branch
* master
```

TASK 8: Run a Simple Java Maven Build Job in Jenkins

```
ubuntu@ip-172-31-27-227:~/hello-java-maven$ git log --oneline
2be1a61 (HEAD -> master) maven job commits
ubuntu@ip-172-31-27-227:~/hello-java-maven$ git remote add origin https://github.com/RagiNiharika/TASK-8_Maven_Build_Job_in_Jenkins.git
https://github.com/RagiNiharika/TASK-8_Maven_Build_Job_in_Jenkins.git
error: remote origin already exists.
ubuntu@ip-172-31-27-227:~/hello-java-maven$ git push origin master
Username for 'https://github.com': RagiNiharika
Password for 'https://github.com':
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Compressing objects: 100% (10/10), done.
Writing objects: 100% (19/19), 2.83 KiB | 1.41 MiB/s, done.
Total 19 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/RagiNiharika/TASK-8_Maven_Build_Job_in_Jenkins.git
 * [new branch] master -> master
ubuntu@ip-172-31-27-227:~/hello-java-maven$
```




Step 7: Configure Jenkins Job

1. In Jenkins User Interface → **Manage Jenkins** → **Global Tool Configuration**
 - Add Maven (name: Maven-3.8.6)
 - Jenkins will auto-install it.
2. **Create New Freestyle Job** → Name: hello-java-maven.
3. Under **Source Code Management**:
 - We are using **GitHub**, enter repo URL.
 - local EC2 folder, skip
4. Under **Build**:
 - Add **Invoke top-level Maven targets**.
 - Goals: **clean package**
5. Save & Build
6. Verify Console output



TASK 8: Run a Simple Java Maven Build Job in Jenkins

 **Jenkins** / All / New Item

Q

⚙️


👤

New Item


Enter an item name

hello-java-maven


Select an item type



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




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



Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments,

OK

 **Jenkins** / hello-java-maven / Configuration

Q

⚙️

👤

Configure

⚙️ General

🔑 Source Code Management

🕒 Triggers

🌐 Environment

📋 Build Steps

🔧 Post-build Actions

Source Code Management

Connect and manage your code repository to automatically pull the latest code for your builds.

☐ None

☒ **Git** ?

Repositories ?

Repository URL ?

https://github.com/RagNihaika/TASK-8_Maven_Build_Job_in_Jenkins.git

Credentials ?

- none -

+ Add

Advanced ▾

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/master

Add Branch

Repository browser ?


(Auto)

Additional Behaviours

Add ▾

Save

Apply

 **Jenkins** / hello-java-maven / Configuration

Q

⚙️

👤

Configure

Automate your build process with ordered tasks like code compilation, testing, and deployment.

⚙️ General

🔑 Source Code Management

🕒 Triggers

🌐 Environment

📋 Build Steps

🔧 Post-build Actions

≡ **Invoke top-level Maven targets** ?

Maven Version

maven

Goals

clean package


Advanced ▾

Add build step ▾

Save

Apply

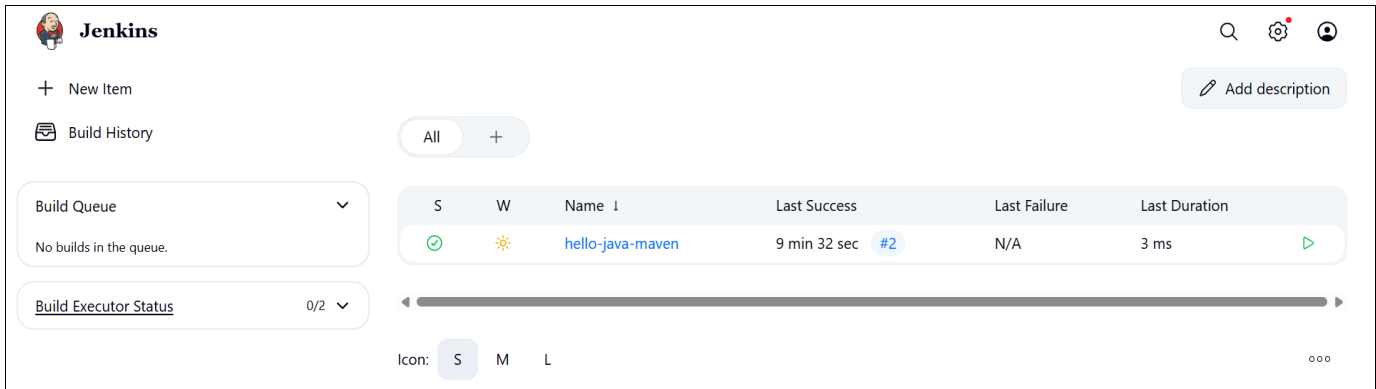
TASK 8: Run a Simple Java Maven Build Job in Jenkins



The screenshot shows the Jenkins web interface for a build job named 'hello-java-maven' (build #2). The 'Console Output' tab is selected, displaying a successful build log. The log indicates the build was started by user 'admin', running as 'SYSTEM', and completed successfully in the workspace '/var/jenkins_home/workspace/hello-java-maven'.

Console Output

```
Started by user admin
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/hello-java-maven
Finished: SUCCESS
```



The screenshot shows the Jenkins 'Build Queue' page. It displays a table of builds currently in the queue. The 'hello-java-maven' build is shown with a status of 'S' (Success), a duration of 9 min 32 sec, and a last success time of 3 ms. The 'Build Queue' section indicates 'No builds in the queue.'

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	hello-java-maven	9 min 32 sec	N/A	3 ms

```
ubuntu@ip-172-31-27-227:~$ ls
hello-java-maven
ubuntu@ip-172-31-27-227:~$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
jenkins/jenkins lts 627182afbe2b 3 weeks ago 472MB
ubuntu@ip-172-31-27-227:~$ docker ps
CONTAINER ID IMAGE NAMES COMMAND CREATED STATUS PORTS
a26fc6b72eb8 jenkins/jenkins:lts "/usr/bin/tini -- /u..." 11 hours ago Up 11 hours 0.0.0.0:8080->8080/tcp, :::8080->8080/tcp, 0.0.0.0:50000->50000/tcp, :::50000->50000/tcp jenkins
ubuntu@ip-172-31-27-227:~$ docker exec -it jenkins bash
jenkins@a26fc6b72eb8:/ $ ls -la /var/jenkins_home/workspace
total 12
drwxr-xr-x 3 jenkins jenkins 4096 Aug 16 01:24 .
drwxr-xr-x 14 jenkins jenkins 4096 Aug 16 01:25 ..
drwxr-xr-x 2 jenkins jenkins 4096 Aug 16 01:24 hello-java-maven
jenkins@a26fc6b72eb8:/ $ ls -la /var/jenkins_home/workspace/hello-java-maven
total 8
drwxr-xr-x 2 jenkins jenkins 4096 Aug 16 01:24 .
drwxr-xr-x 3 jenkins jenkins 4096 Aug 16 01:24 ..
jenkins@a26fc6b72eb8:/ $
```

How can we check Jenkins Workspace

- Since you're running Jenkins inside Docker, the workspace path:
- `/var/jenkins_home/workspace`
- We have two options
- **Option 1: Go inside the Jenkins container**
- `docker exec -it jenkins bash`
- Now you're inside the container. Run:
- `ls -la /var/jenkins_home/workspace`
- we should see our job folder, for example:
- `hello-java-maven`
- If you want to check inside the job:
- `ls -la /var/jenkins_home/workspace/hello-java-maven`
- **Option 2: Check from host without entering container**
- `docker exec -it jenkins ls -la /var/jenkins_home/workspace`
- Or a specific job:
- `docker exec -it jenkins ls -la /var/jenkins_home/workspace/hello-java-maven`

TASK 8: Run a Simple Java Maven Build Job in Jenkins