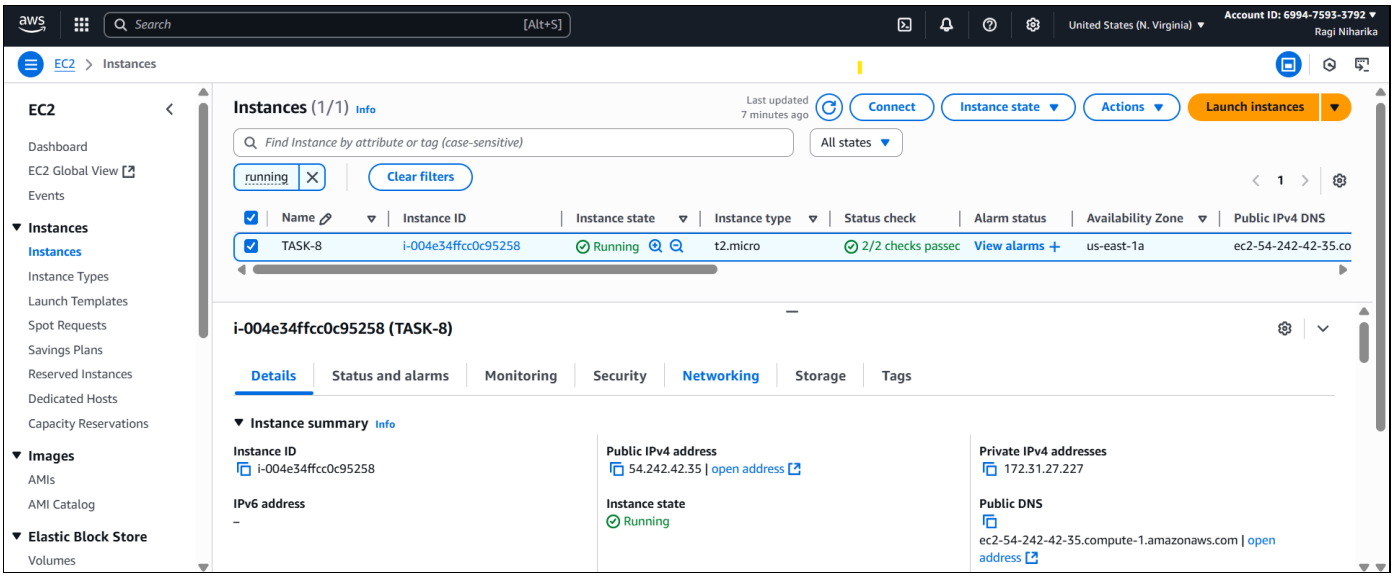
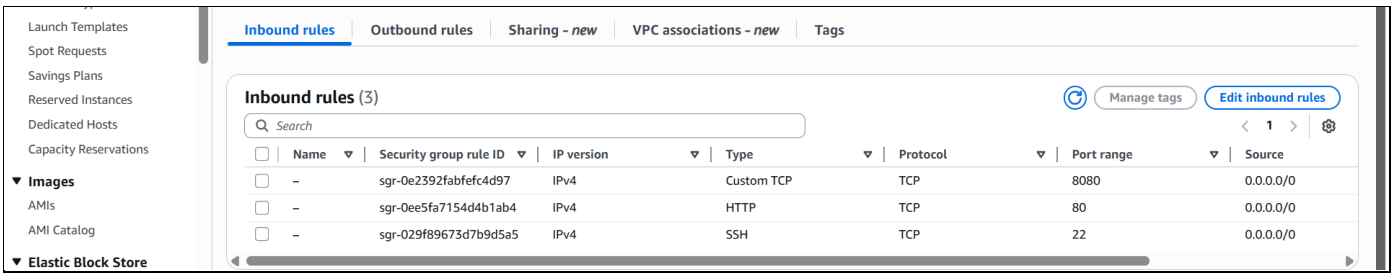
TASK 8: Run a Simple Java Maven Build Job in Jenkins

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>

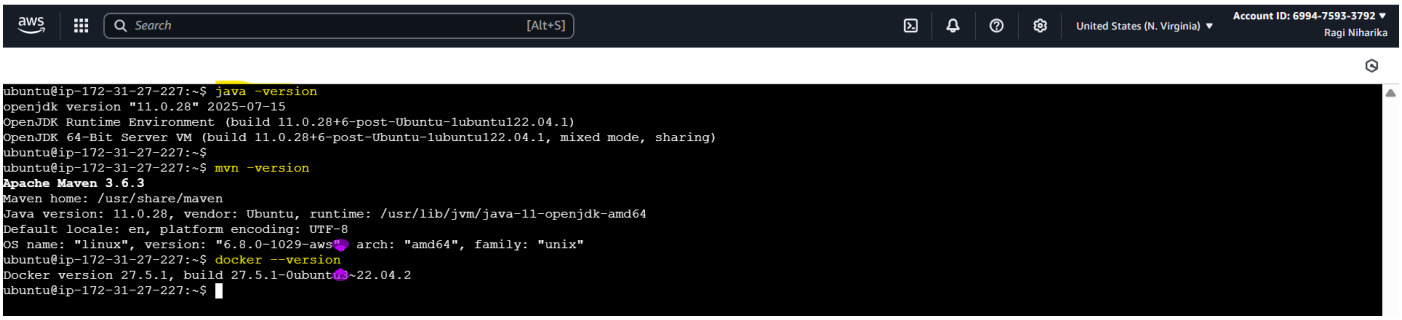
Step 3: Install Dependencies

- 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

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



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:lts
```

```
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:lts
Unable to find image 'jenkins/jenkins:lts' locally
lts: Pulling from jenkins/jenkins
ebed137c7c18: Pull complete
a16eed992861: Pull complete
d0340747bfc5: Pull complete
b3198e29cbc4: Pull complete
9fa62d58a0db: Pull complete
31a9a2be77c3: Pull complete
3cd347526f2b: Pull complete
654460caae81: Pull complete
6ce3758f4dbb: Pull complete
2b1923fe83ed: Pull complete
a3391befb457: Pull complete
48e332b529b0: Pull complete
Digest: sha256:0e66af38c9272490ba18757d5d4d41e4ac2160278ae40b69d6da9b5adbe98794
Status: Downloaded newer image for jenkins/jenkins:lts
a26fc6b72eb8087075a29f18b58c2135ddb5024[illegible]2f03c4d5150be9421b51800
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:~$
```

Get Jenkins admin password:

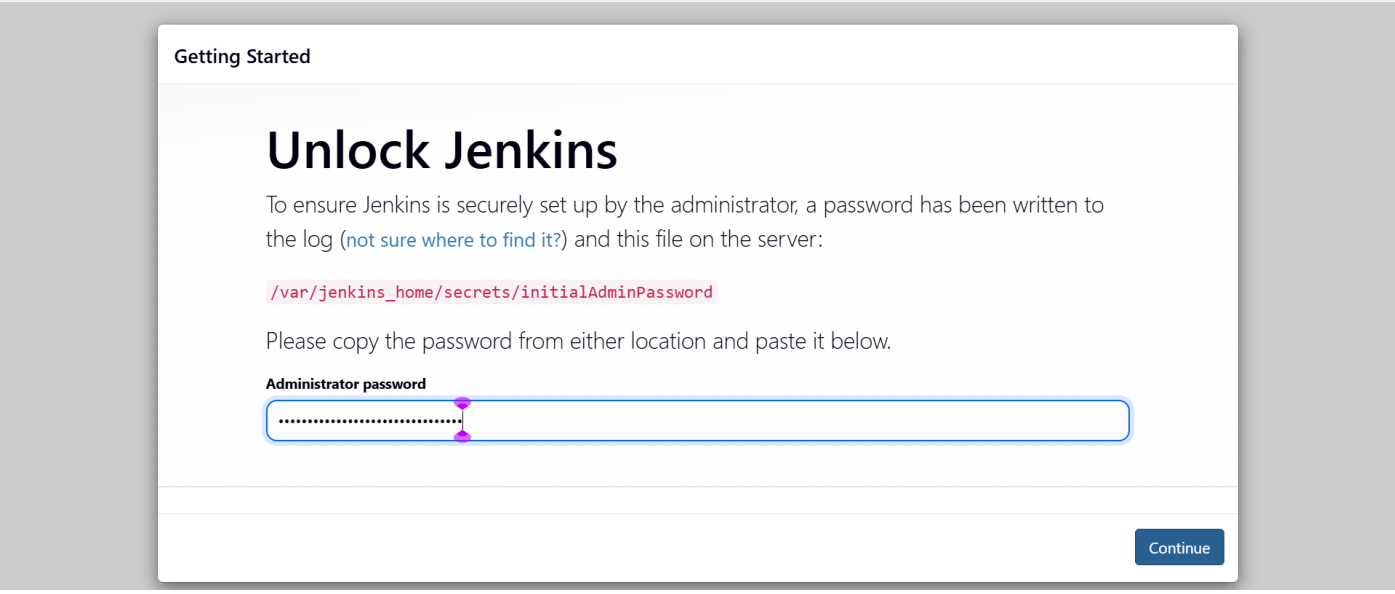
TASK 8: Run a Simple Java Maven Build Job in Jenkins

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:~$
```

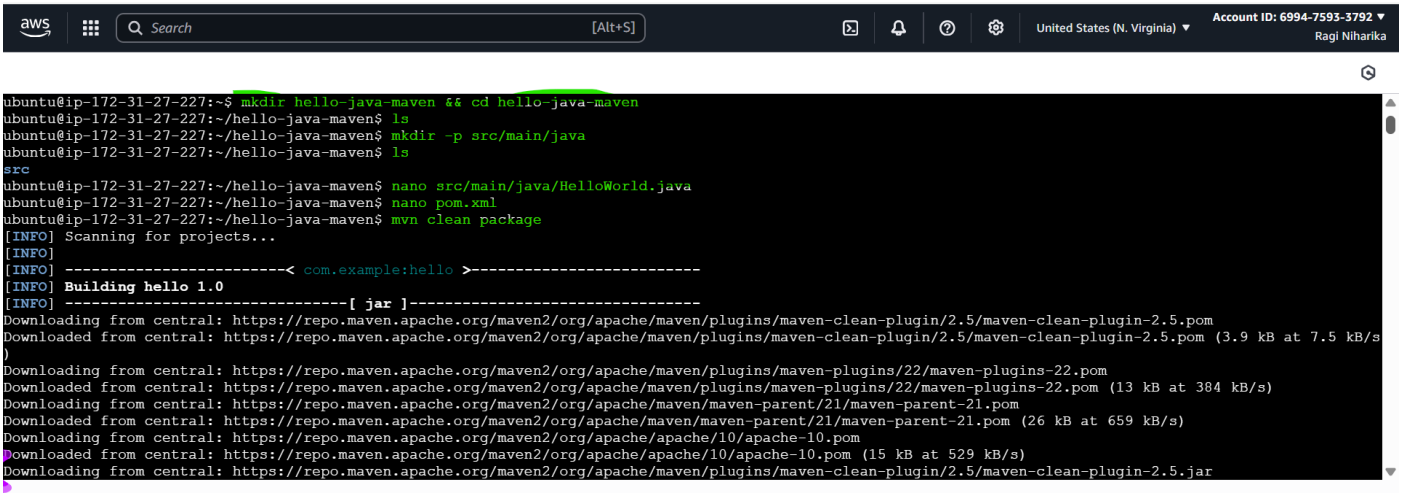
Open Jenkins UI in browser:

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



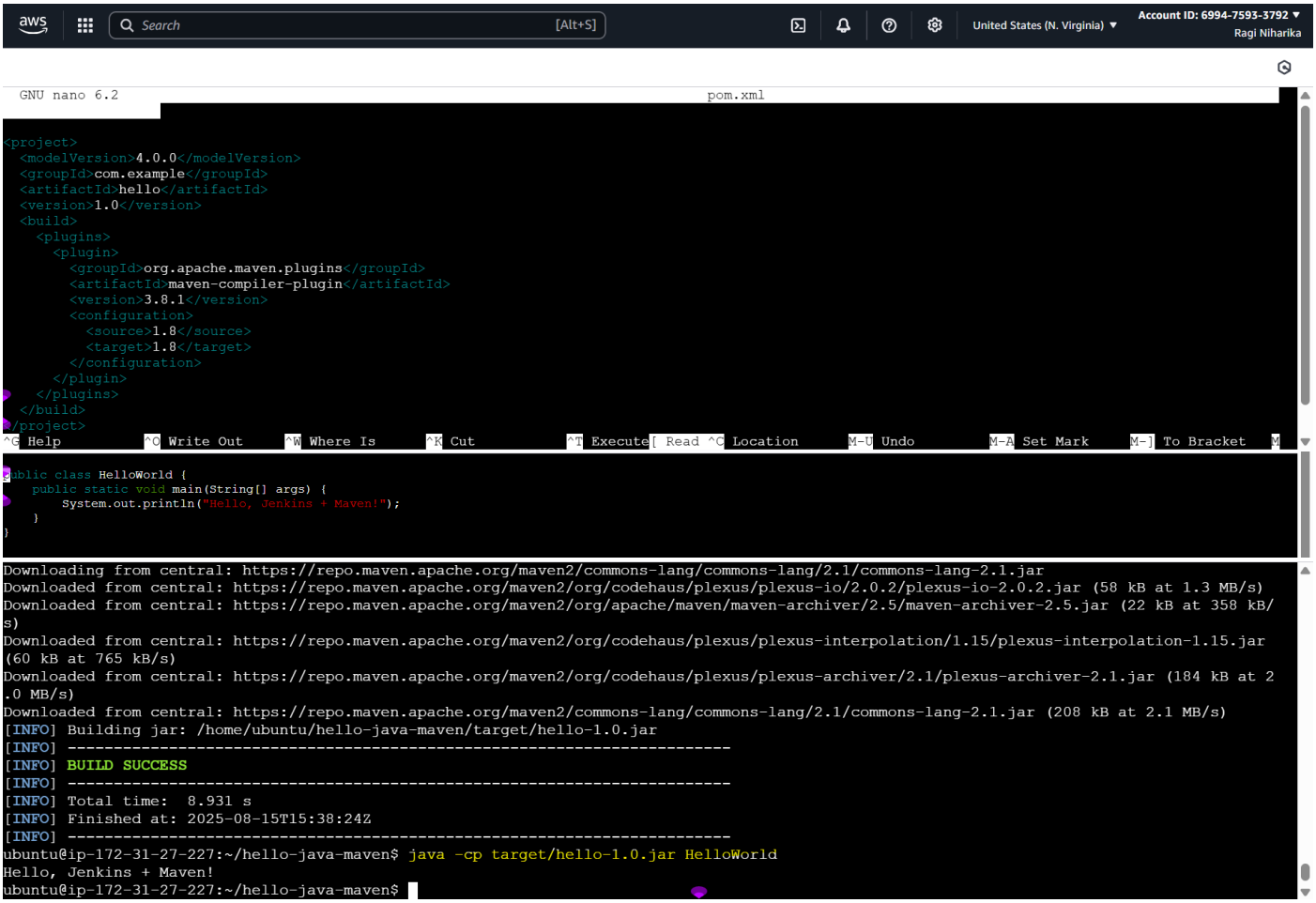
Step 5: Prepare Java App (hello-java-maven)

- 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

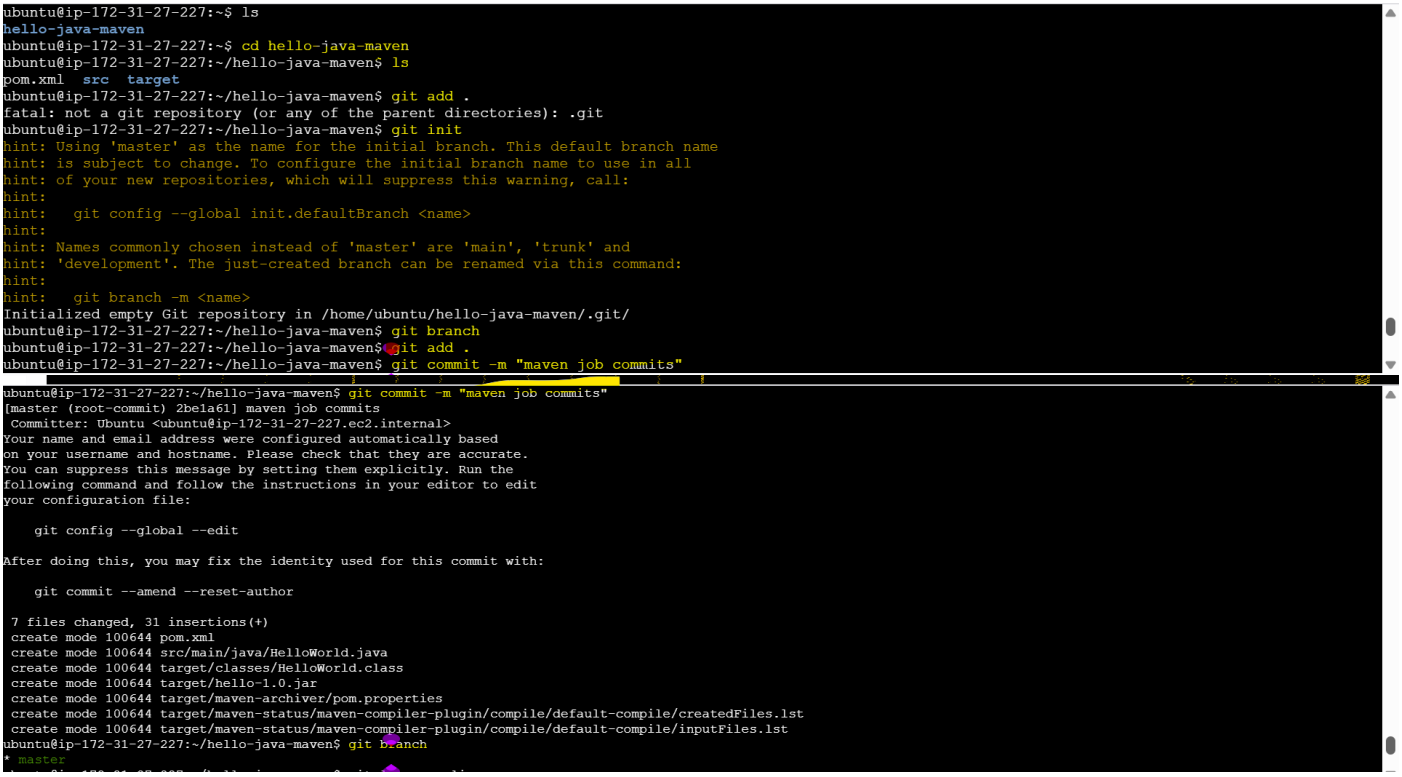



```
ubuntu@ip-172-31-27-227:~$ mkdir hello-java-maven && cd hello-java-maven
ubuntu@ip-172-31-27-227:~/hello-java-maven$ ls
ubuntu@ip-172-31-27-227:~/hello-java-maven$ mkdir -p src/main/java
ubuntu@ip-172-31-27-227:~/hello-java-maven$ ls
src
ubuntu@ip-172-31-27-227:~/hello-java-maven$ nano src/main/java/HelloWorld.java
ubuntu@ip-172-31-27-227:~/hello-java-maven$ nano pom.xml
ubuntu@ip-172-31-27-227:~/hello-java-maven$ mvn clean package
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------< com.example:hello >-------------------------
[INFO] Building hello 1.0
[INFO] --------------------------------[ jar ]---------------------------------
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.5/maven-clean-plugin-2.5.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.5/maven-clean-plugin-2.5.pom (3.9 kB at 7.5 kB/s
)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/22/maven-plugins-22.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/22/maven-plugins-22.pom (13 kB at 384 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/21/maven-parent-21.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/21/maven-parent-21.pom (26 kB at 659 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/apache/10/apache-10.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/apache/10/apache-10.pom (15 kB at 529 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.5/maven-clean-plugin-2.5.jar
```

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

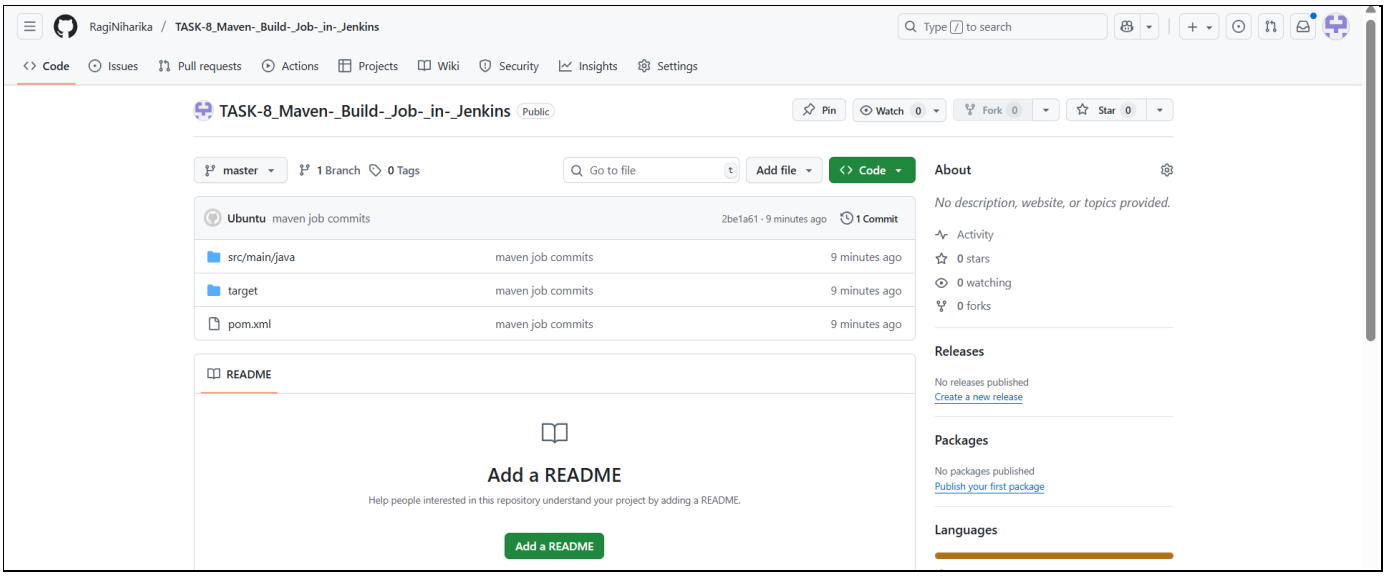


Step 6: Perform Git Operations



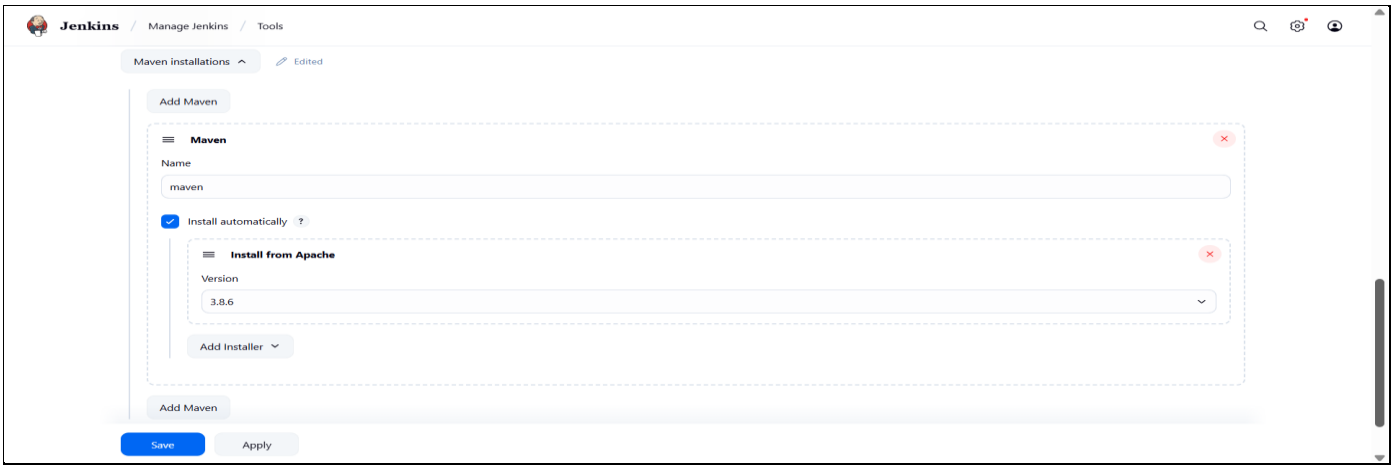
TASK 8: Run a Simple Java Maven Build Job in Jenkins

```
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
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
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://RagiNiharika@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$
```

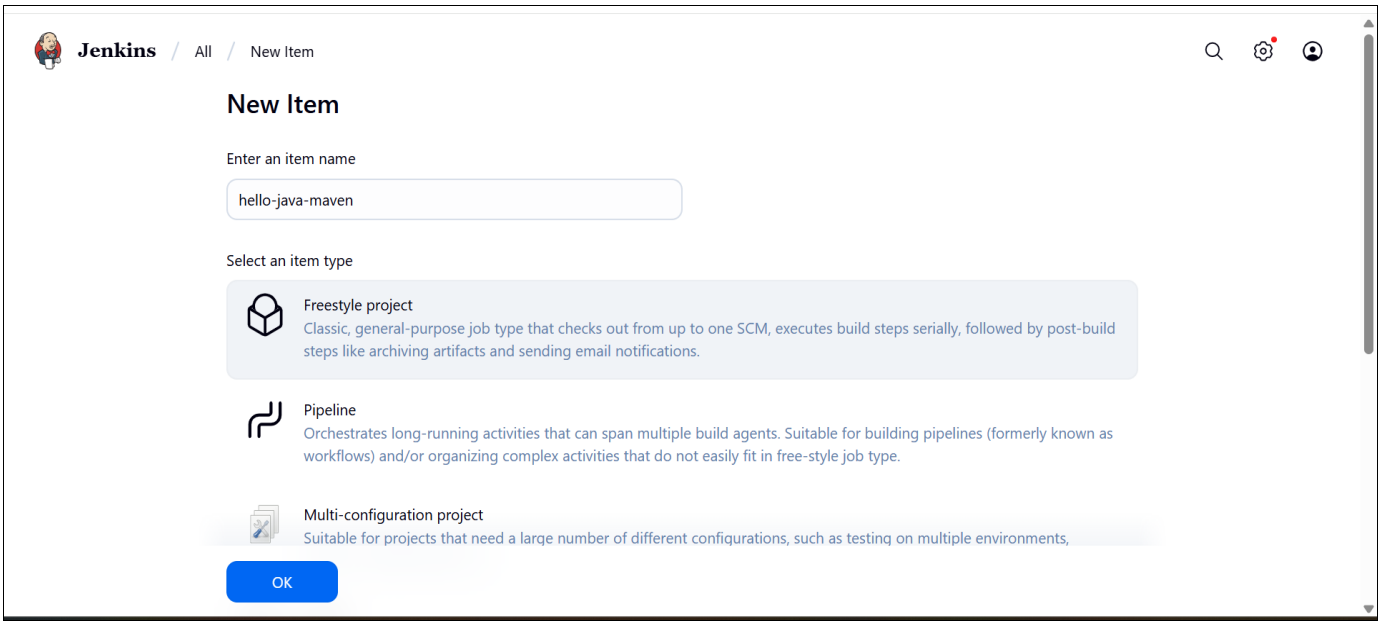


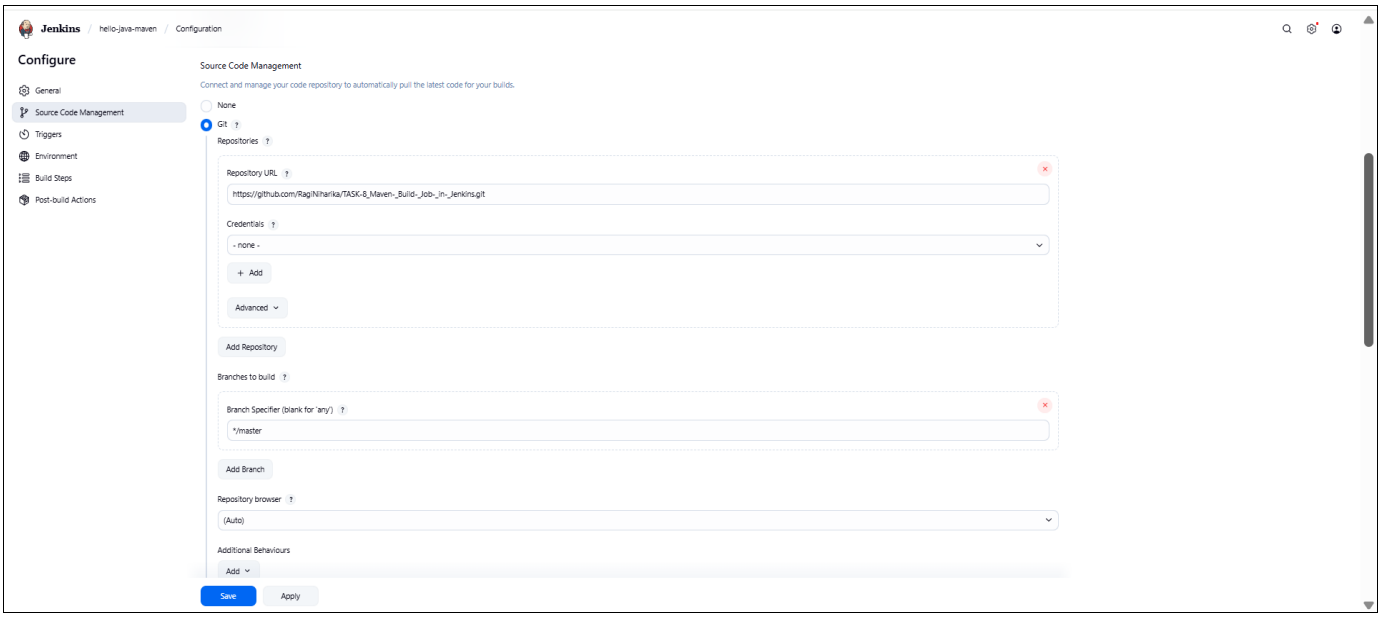
Step 7: Configure Jenkins Job

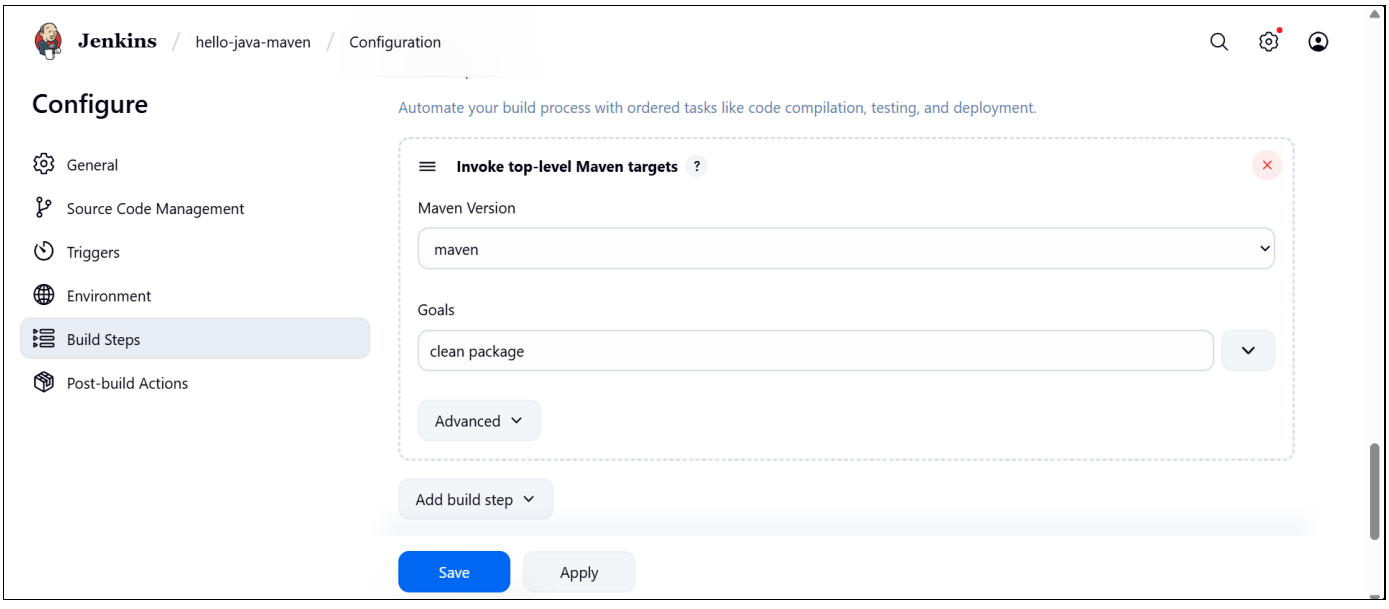
1. In Jenkins User Interface → **Manage Jenkins → Global Tool Configuration**
   - Add Maven (name: Maven-3.8.6)
   - Jenkins will auto-install it.
2. **Create New Freestyle Job** → Name: hello-java-maven.
3. Under **Source Code Management**:
   - We are using GitHub, enter repo URL.
   - local EC2 folder,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



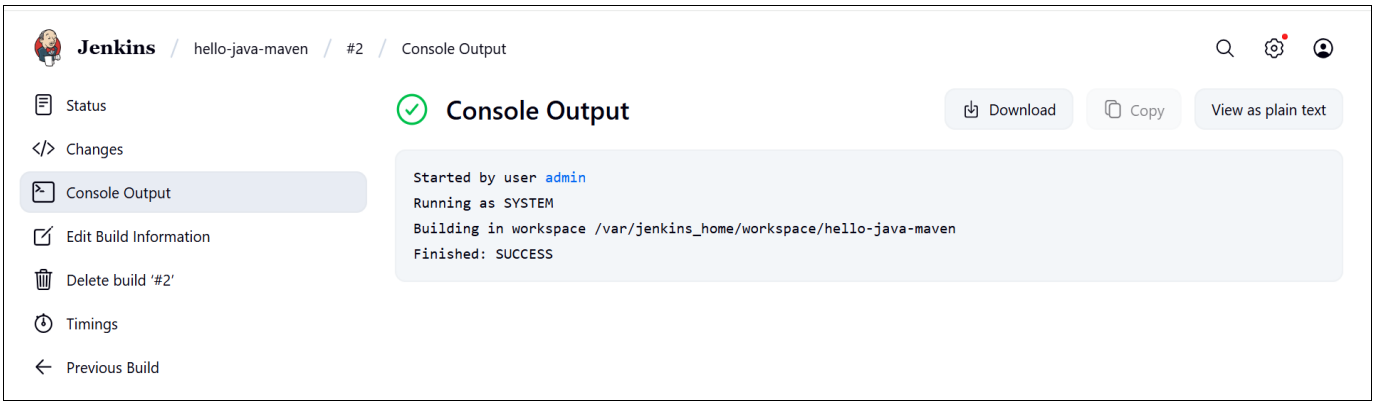
TASK 8: Run a Simple Java Maven Build Job in Jenkins

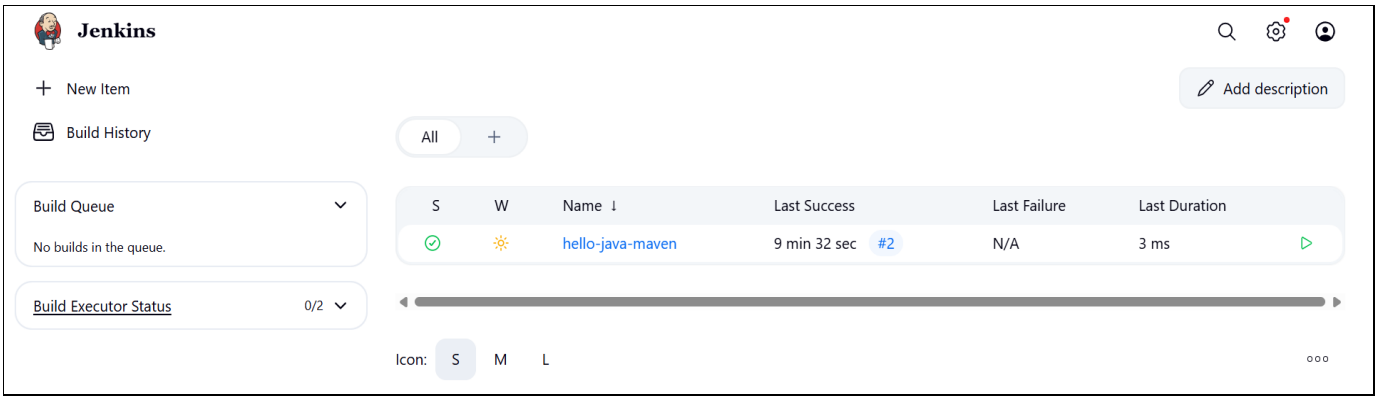






TASK 8: Run a Simple Java Maven Build Job in Jenkins





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