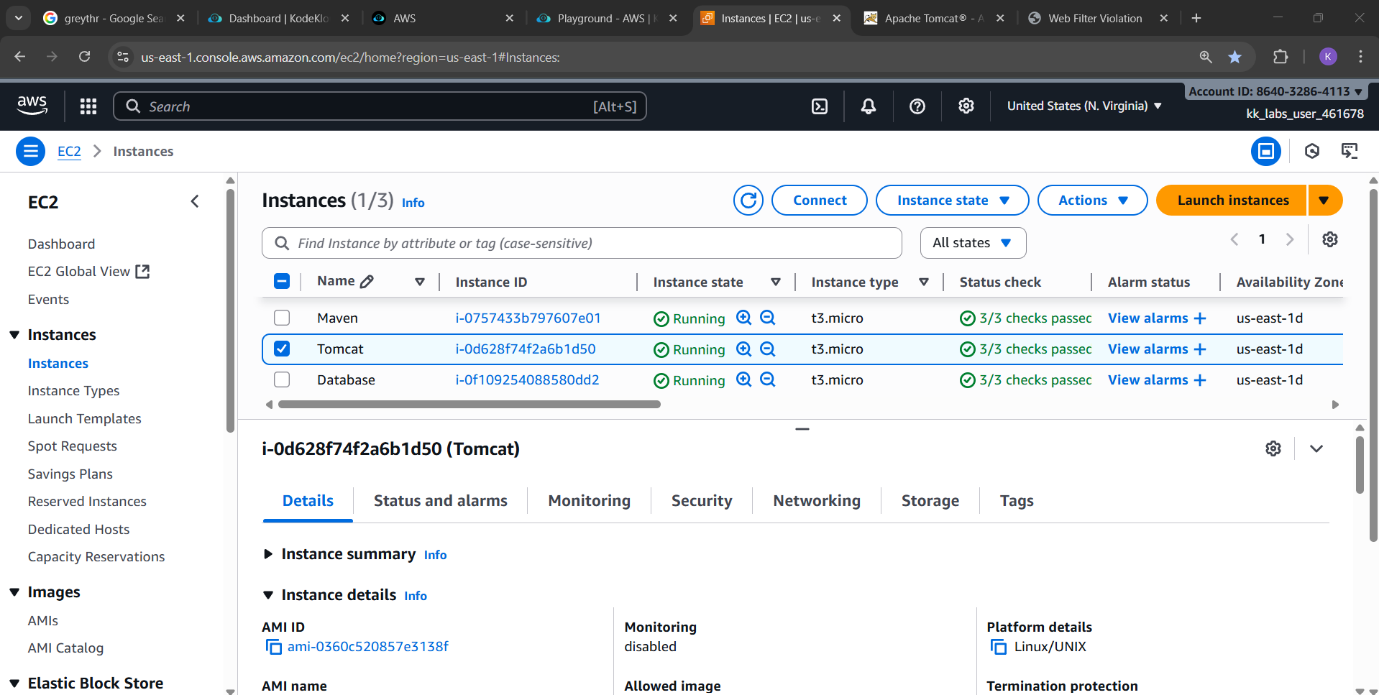
**Java Application Deployment Using 3 Servers – Step by Step**

We will use **three servers**:

1. **Build Server** – Compile and package Java code.
2. **Deployment Server** – Run the application using Tomcat.
3. **Database Server** – Host MySQL database.



**Step 1: Setup Build Server (Java + Maven)**

**1. Launch a Linux server**

* Use Ubuntu or any Linux distribution.

**2. Install Java**

sudo apt update

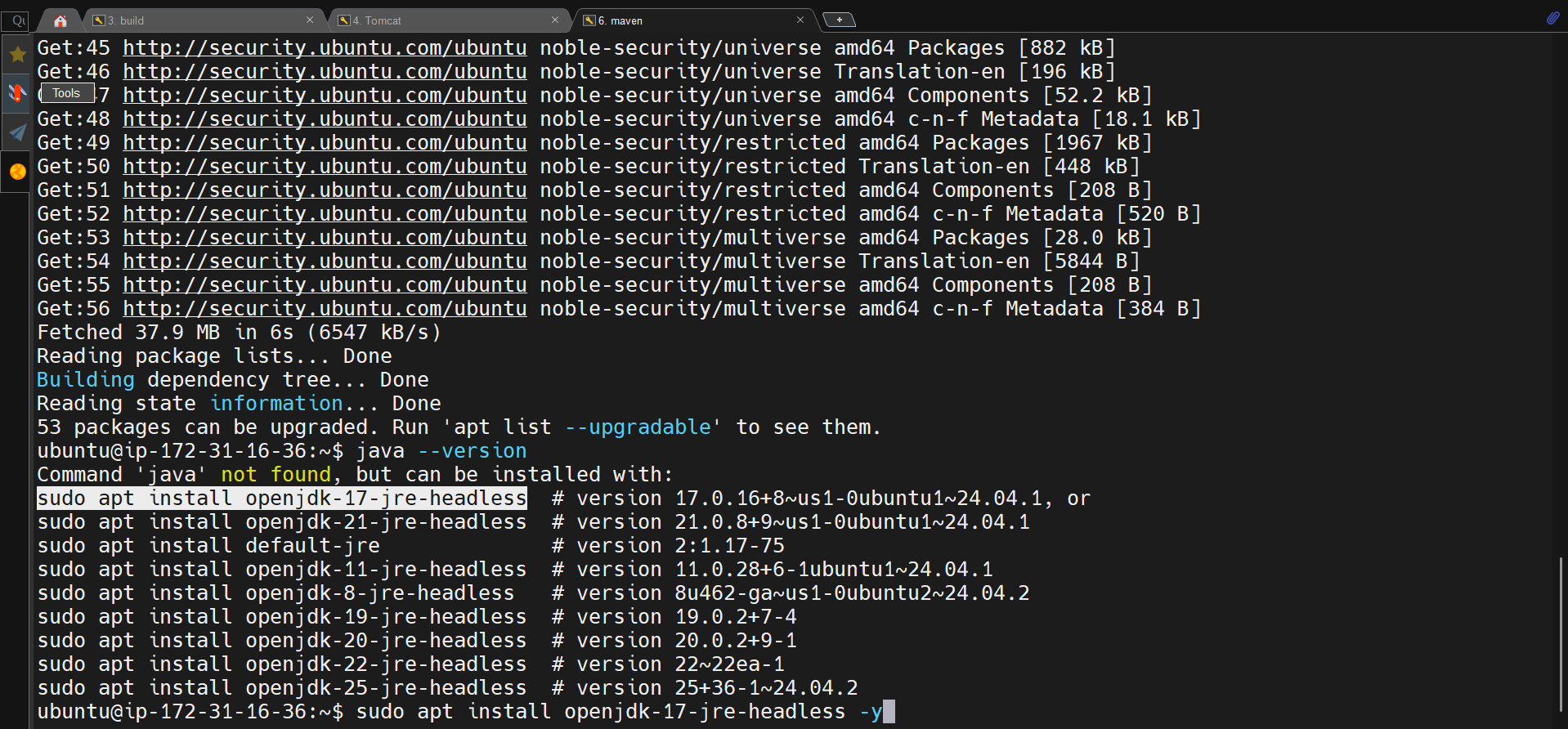
sudo apt install openjdk-17-jdk -y

java -version

**3. Install Maven**

sudo apt install maven -y

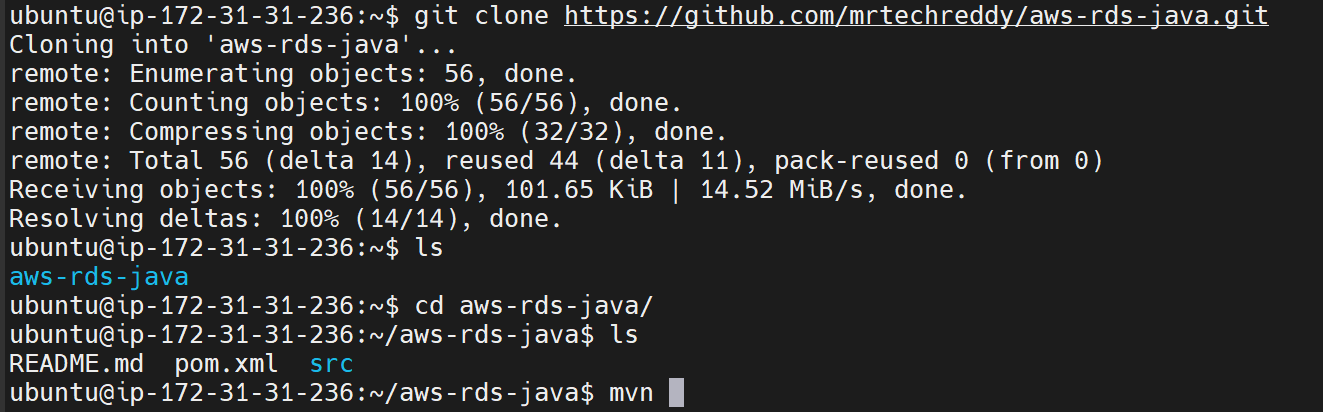
mvn -version



**4. Clone your Java project repository**

git clone <your\_repo\_url>

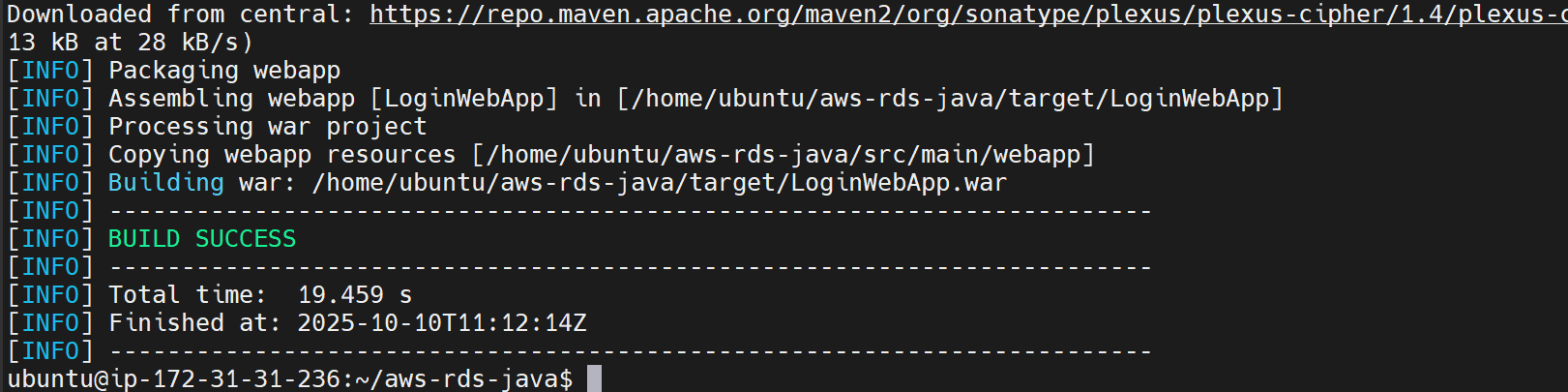
cd <project\_folder>



**5. Build the project**

mvn clean package

* This will generate a **WAR** (or JAR) file in the target/ directory.

**6. Verify the artifact**

ls target/

* Ensure you see something like app.war.

**Step 2: Transfer Artifact to Deployment Server**

Use scp to securely copy the artifact:

scp /home/ubuntu/project/target/app.war ubuntu@<deployment\_server\_ip>:/home/ubuntu/

**Notes:**

* Replace <deployment\_server\_ip> with your Deployment Server IP.
* Make sure SSH access works (keys or password authentication).

**Step 3: Setup Deployment Server (Java + Tomcat)**

**1. Launch Deployment Server**

* Use a fresh Linux instance.

**2. Install Java**

sudo apt update

sudo apt install openjdk-17-jdk -y

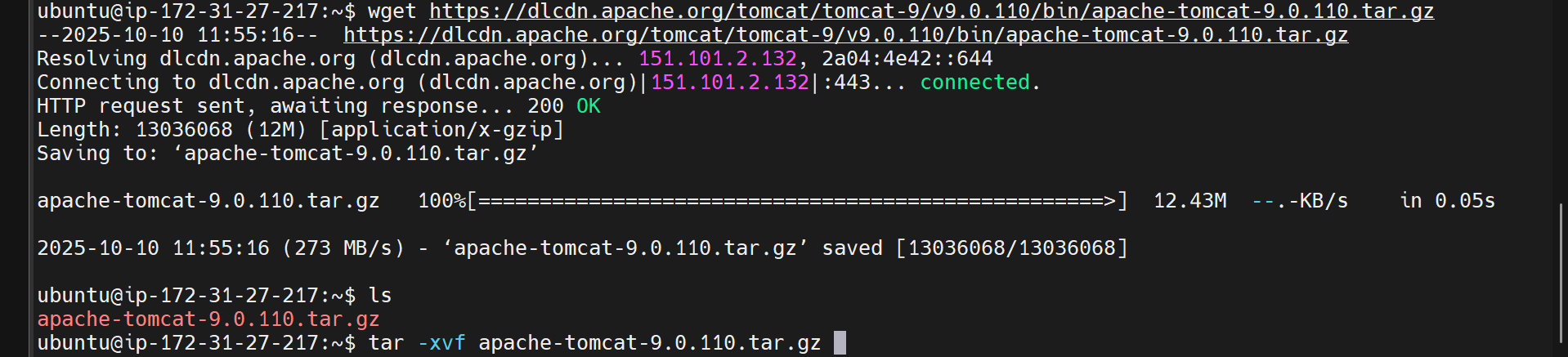
java -version

**3. Install Apache Tomcat**

wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.91/bin/apache-tomcat-9.0.91.tar.gz

tar -xvzf apache-tomcat-9.0.91.tar.gz

mv apache-tomcat-9.0.91 tomcat



**4. Deploy WAR file to Tomcat**

mv /home/ubuntu/app.war /home/ubuntu/tomcat/webapps/

**5. Start Tomcat**

cd /home/ubuntu/tomcat/bin

./startup.sh

**6. Verify application**

* Open browser:

http://<deployment\_server\_ip>:8080/app

**Step 4: Setup Database Server (MySQL)**

**1. Launch Database Server**

* Use a Linux instance.

**2. Install MySQL**

sudo apt update

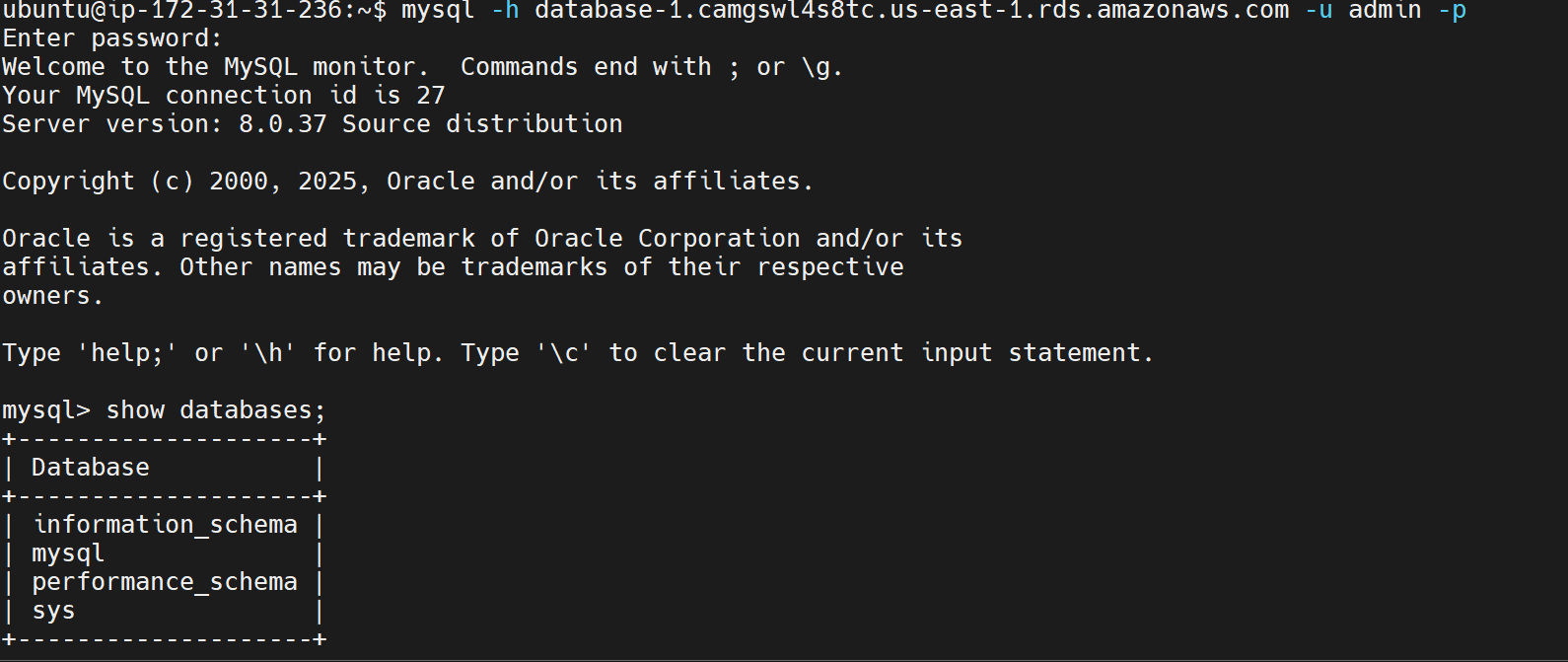
sudo apt install mysql-server -y

sudo systemctl start mysql

sudo systemctl enable mysql

**3. Secure MySQL**

sudo mysql\_secure\_installation



**4. Create database and user**

mysql -u root -p

CREATE DATABASE appdb;

CREATE USER 'appuser'@'%' IDENTIFIED BY 'password';

GRANT ALL PRIVILEGES ON appdb.\* TO 'appuser'@'%';

FLUSH PRIVILEGES;

EXIT;

**5. Enable remote access**

sudo vim /etc/mysql/mysql.conf.d/mysqld.cnf

* Change bind-address = 127.0.0.1 → bind-address = 0.0.0.0

**6. Restart MySQL**

sudo systemctl restart mysql

**Step 5: Connect Application to Database**

1. Update application database configuration (e.g., application.properties or context.xml):

spring.datasource.url=jdbc:mysql://<database\_server\_ip>:3306/appdb

spring.datasource.username=appuser

spring.datasource.password=password

1. Redeploy the application to Tomcat if necessary.
2. Test database connectivity from Deployment Server:

mysql -h <database\_server\_ip> -u appuser -p

**Step 6: Verification**

1. Check Tomcat service:

ps -ef | grep tomcat

1. Access the application in a browser:

http://<deployment\_server\_ip>:8080/app

1. Confirm the application connects to the database and all features work.

