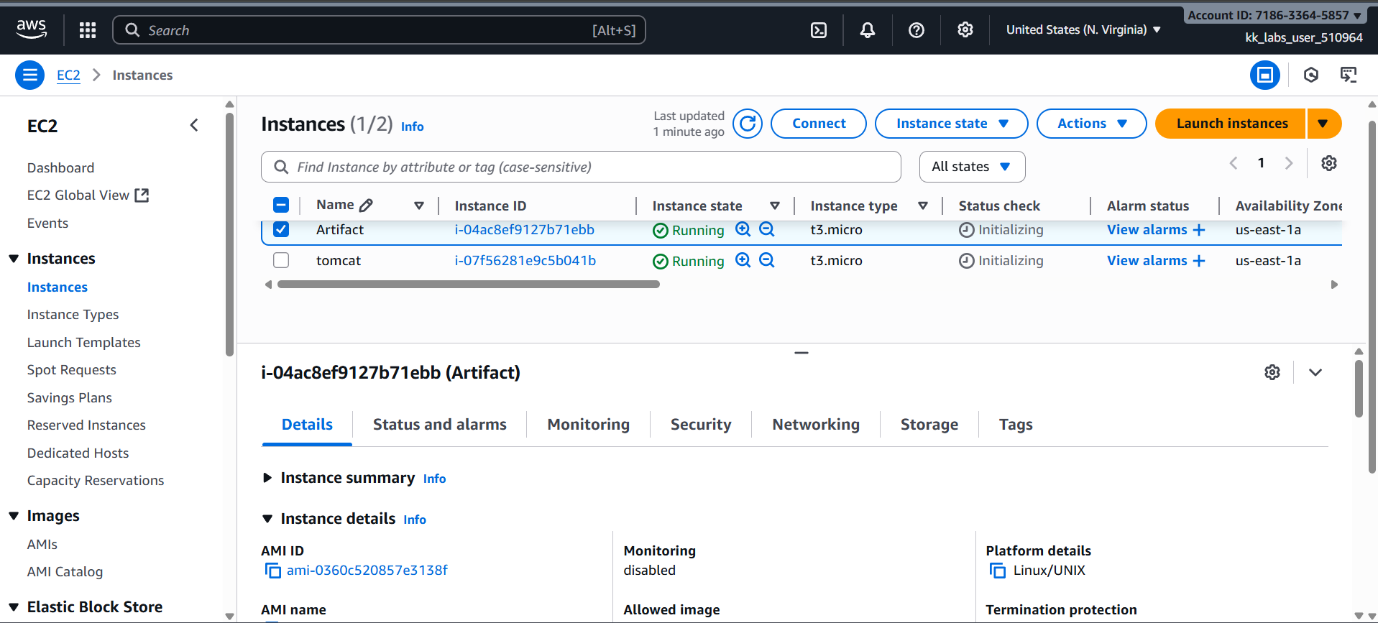
**🌱 Application Deployment with Database (Easy Step-by-Step Guide)**

**Step 1: Build Server Setup**

**🖥️ Launch Two Servers**

* Create **two EC2 instances** (or any Linux servers).  
  Name them:
  + **Build Server**
  + **Deploy Server**



**⚙️ On the Build Server**

We’ll build the Java application here.

**1️⃣ Install Java**

sudo apt update

sudo apt install openjdk-17-jdk -y

**2️⃣ Install Maven**

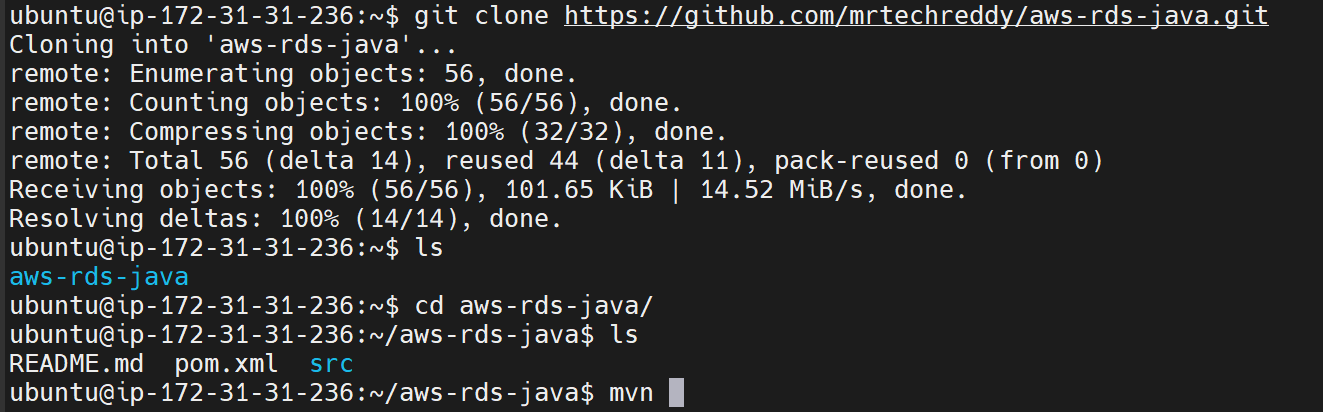
sudo apt install maven -y

**3️⃣ Clone the project from GitHub**

Example:

git clone https://github.com/your-repo-name.git

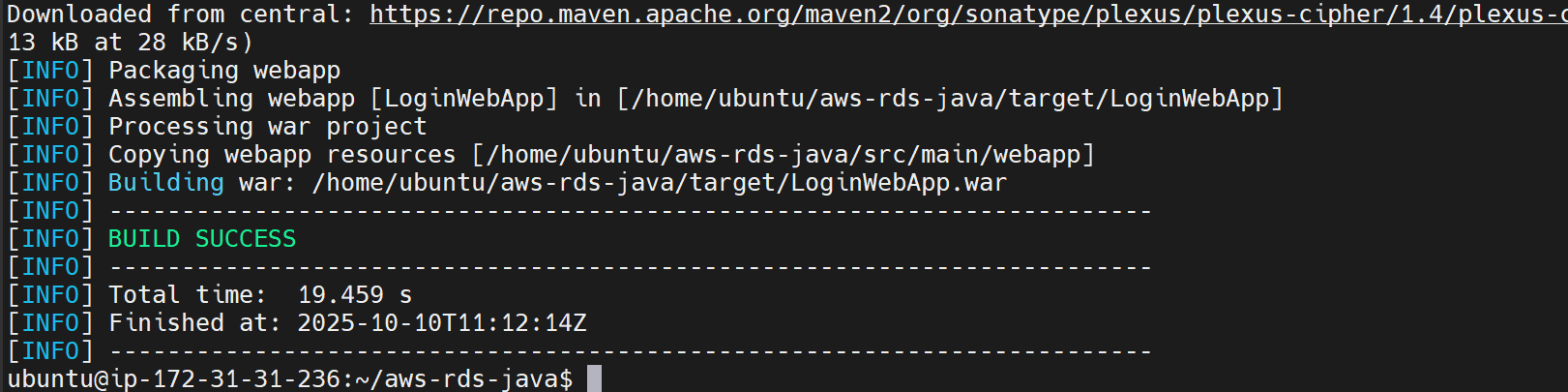
cd your-repo-name



**4️⃣ Build the project**

mvn package

✅ If everything is correct, you’ll see **BUILD SUCCESS** and a .war file will be created inside the target/ folder.



**Step 2: Deploy Server Setup**

We’ll deploy the .war file on this server using Tomcat.

**1️⃣ Install Java**

sudo apt install openjdk-17-jdk -y

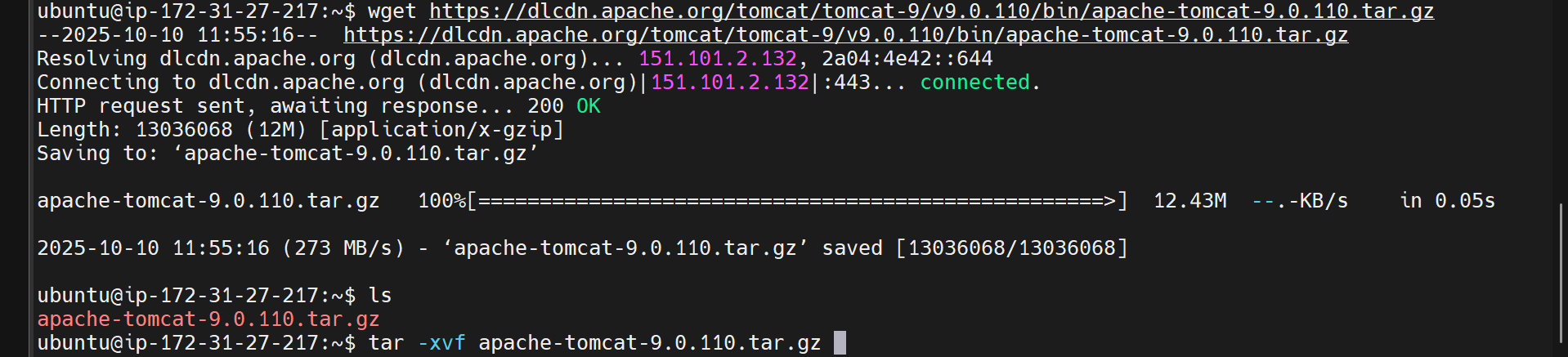
**2️⃣ Install Apache Tomcat**

Download Tomcat 9:

wget https://downloads.apache.org/tomcat/tomcat-9/v9.0.85/bin/apache-tomcat-9.0.85.tar.gz

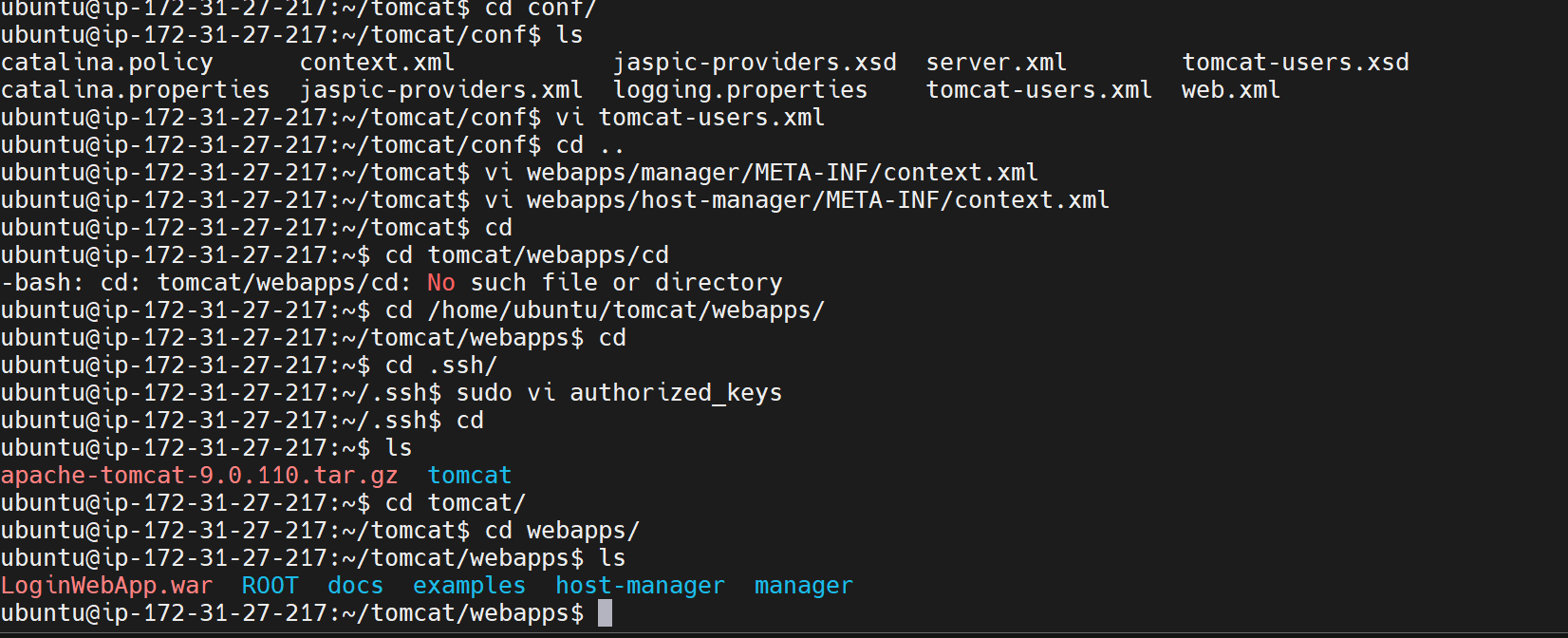
tar -xvzf apache-tomcat-9.0.85.tar.gz

mv apache-tomcat-9.0.85 tomcat



**3️⃣ Edit configuration files**

* **a.** In tomcat/conf/tomcat-users.xml, add:
* <user username="admin" password="admin" roles="manager-gui,admin-gui"/>
* **b.** In webapps/manager/META-INF/context.xml and webapps/host-manager/META-INF/context.xml,  
  comment out the lines that restrict access:
* <!-- <Valve className="org.apache.catalina.valves.RemoteAddrValve"
* allow="127\.\d+\.\d+\.\d+|::1" /> -->

**4️⃣ Open port 8080 in your AWS security group**

**5️⃣ Start Tomcat**

cd ~/tomcat/bin

./startup.sh

**6️⃣ Access Tomcat**

Open your browser and go to:

http://<Deploy-Server-Public-IP>:8080

Login with:

Username: admin

Password: admin

**Step 3: Enable Secure File Transfer**

Now we’ll securely transfer the .war file from the **Build Server** to the **Deploy Server**.

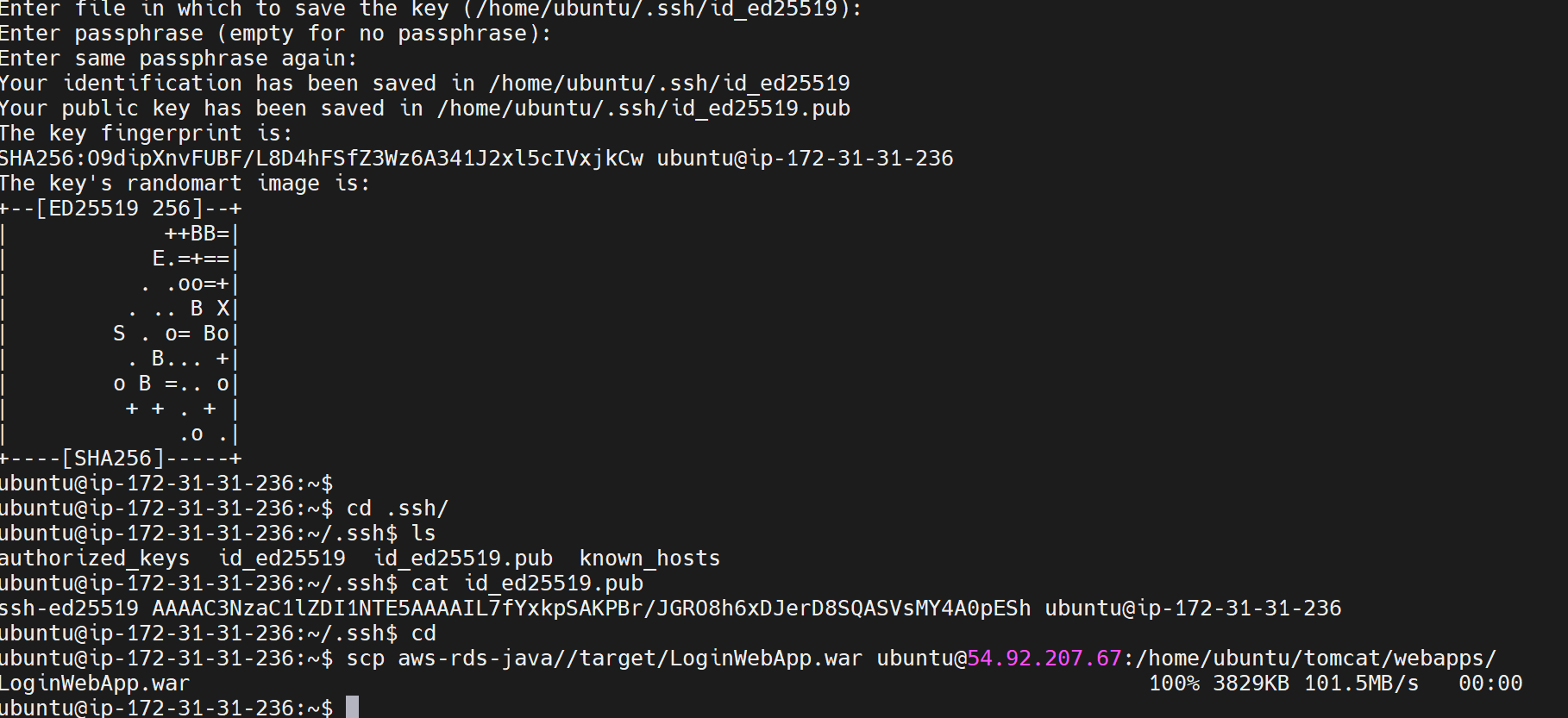
**1️⃣ On the Build Server:**

Generate SSH keys:

ssh-keygen

Then show the public key:

cat ~/.ssh/id\_ed25519.pub



**2️⃣ Copy that key.**

**3️⃣ On the Deploy Server:**

Paste the key inside the authorized\_keys file:

cd ~/.ssh

vi authorized\_keys

# paste the copied key here

This allows passwordless SSH/scp between the two servers.

**Step 4: Transfer the WAR file**

From the **Build Server**, run:

scp /home/ubuntu/your-project/target/\*.war ubuntu@<Deploy-Server-Public-IP>:/home/ubuntu/tomcat/webapps/

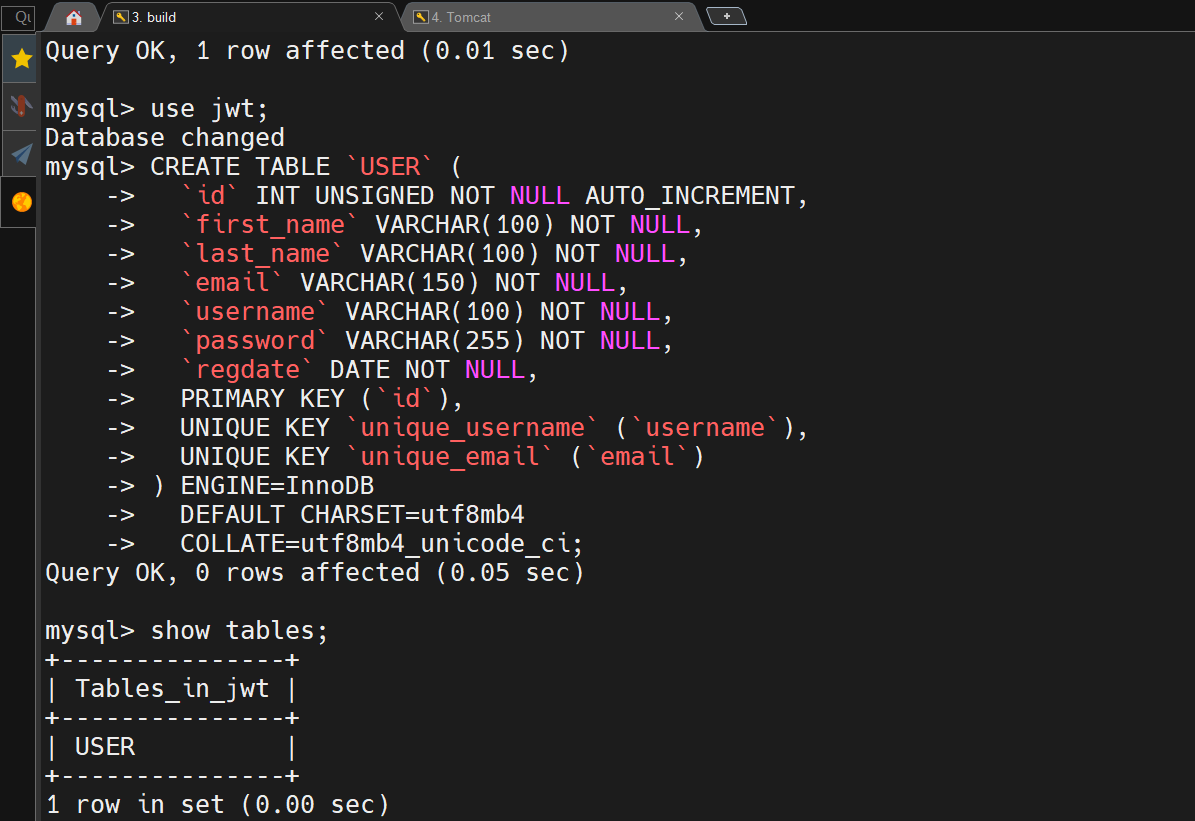
✅ This copies the .war file directly into Tomcat’s webapps folder.  
Tomcat automatically detects and deploys .war files placed there.

**Step 5: Create the Database**

If your app uses MySQL, connect to your MySQL server and create a database (for example: jwt):

CREATE DATABASE jwt;

Then configure your application to use this database in its application.properties or config file.



**Step 6: Access the Application**

Once the .war file is deployed and the database is ready, open your browser:

http://<Deploy-Server-Public-IP>:8080/<your-app-name>/

You should see your **application’s login page** connected to the database 🎉

**✅ Final Output**

* Application built successfully on Build Server
* .war file deployed automatically on Deploy Server (Tomcat)
* Database connected properly
* Web app accessible from browser

