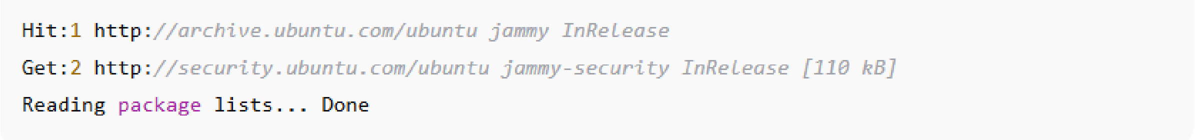
All commands below assume you are on **Ubuntu 24.04** and you run them with a user that has sudo privileges

Step 1 — Update the system

sudo apt update

Sample Output:



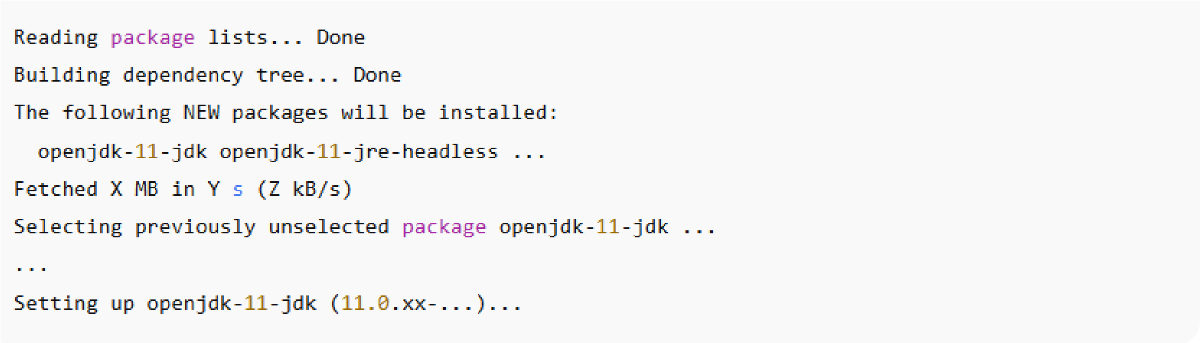
```
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Reading package lists... Done
```

Step 2: Install Java

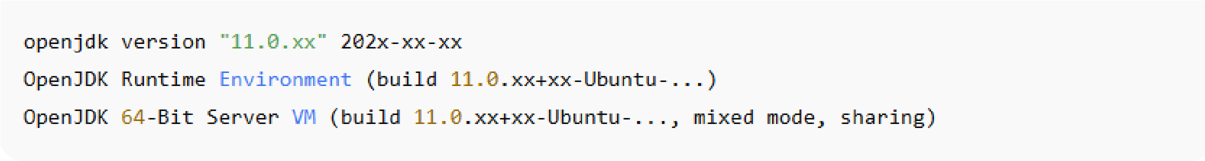
sudo apt install openjdk-11-jdk -y

java -version

Sample Output:



```
Reading package lists... Done
Building dependency tree... Done
The following NEW packages will be installed:
  openjdk-11-jdk openjdk-11-jre-headless ...
Fetched X MB in Y s (Z kB/s)
Selecting previously unselected package openjdk-11-jdk ...
...
Setting up openjdk-11-jdk (11.0.xx-...)...
```



```
openjdk version "11.0.xx" 202x-xx-xx
OpenJDK Runtime Environment (build 11.0.xx+xx-Ubuntu-...)
OpenJDK 64-Bit Server VM (build 11.0.xx+xx-Ubuntu-..., mixed mode, sharing)
```

Java runtime and compiler will be available. Jenkins uses the JRE/JDK to run. If Java fails to install, Jenkins install will fail or warn.

Step 3 — Add Jenkins repository and key

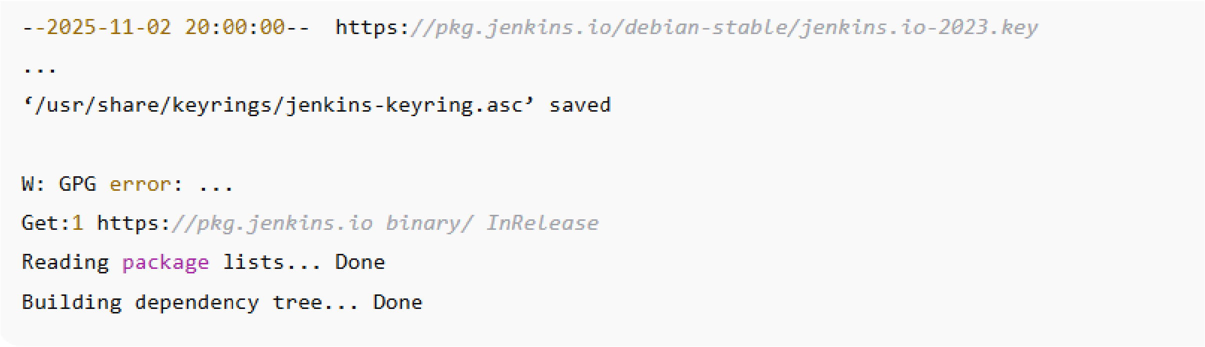
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \

  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key

```
sudo sh -c 'echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian binary/ > /etc/apt/sources.list.d/jenkins.list'
```

```
sudo apt update
```

Sample Output:



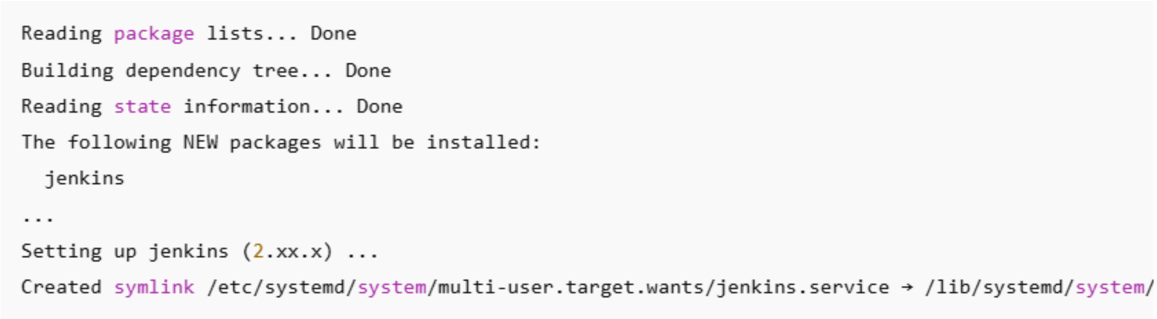
```
--2025-11-02 20:00:00--  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
...
'/usr/share/keyrings/jenkins-keyring.asc' saved

W: GPG error: ...
Get:1 https://pkg.jenkins.io binary/ InRelease
Reading package lists... Done
Building dependency tree... Done
```

Step 4: Install Jenkins

```
sudo apt install jenkins -y
```

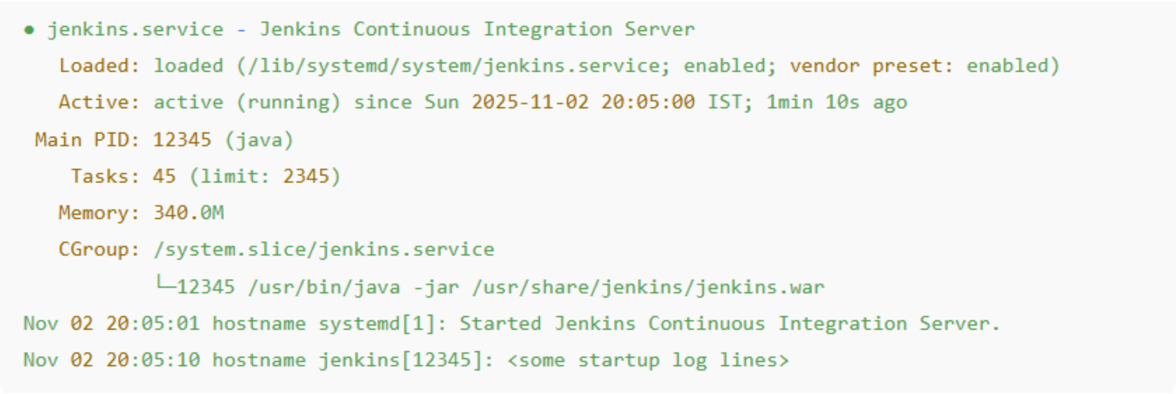
Sample Output:



```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  jenkins
...
Setting up jenkins (2.xx.x) ...
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /lib/systemd/system/
```

Step 5 — Start and enable Jenkins service

```
sudo systemctl start jenkins
sudo systemctl enable jenkins
sudo systemctl status Jenkins
```



```
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2025-11-02 20:05:00 IST; 1min 10s ago
 Main PID: 12345 (java)
    Tasks: 45 (limit: 2345)
   Memory: 340.0M
   CGroup: /system.slice/jenkins.service
           └─12345 /usr/bin/java -jar /usr/share/jenkins/jenkins.war
Nov 02 20:05:01 hostname systemd[1]: Started Jenkins Continuous Integration Server.
Nov 02 20:05:10 hostname jenkins[12345]: <some startup log lines>
```

Step 6 — Open port and access Jenkins UI

Access Jenkins in browser: http://localhost:8080.

What you will see the first time:

- A web page asking to **"Unlock Jenkins"** with a field for the initial admin password. The UI prompts to paste the password from the next step.

If you cannot access:

- If accessing from another machine, ensure firewall allows port 8080 (e.g., sudo ufw allow 8080), or use SSH port forwarding.
- If Jenkins uses another port (customized), use that port instead.

Step 7 — Unlock Jenkins & initial setup

Command to get the initial password

sudo cat /var/lib/jenkins/secrets/initialAdminPassword

What it does:

- Prints a randomly created admin password. This is shown only once at first boot.

```
a1b2c3d4e5f6g7h8i9j0k...
```

What you will do next (UI flow):

1. Paste that password into the browser unlock page.

2. Choose “Install suggested plugins” (recommended for beginners) — Jenkins downloads and installs common plugins.
3. Create the first admin user (username/password).
4. Click Save & Finish.

Post-installation configuration

- Configure tools like JDK and Git under Manage Jenkins → Global Tool Configuration.
- Create a job/pipeline.
- Install any additional plugins you need.
- Schedule backups of Jenkins config (e.g., copy /var/lib/jenkins or use a backup plugin).

Create & execute the example pipeline

Checkout → Build → Test from a GitHub repository (e.g., https://github.com/your-username/HelloWorldProject.git).

```
pipeline {
  agent any
  stages {
    stage('Checkout') {
      steps {
        echo 'Cloning source code from GitHub...'
        git branch: 'main', url: 'https://github.com/your-username/HelloWorldProject.git'
      }
    }
    stage('Build') {
      steps {
        echo 'Building Java project...'
        sh 'javac HelloWorld.java'
      }
    }
    stage('Test') {
            steps {
                echo 'Running tests...'
                sh 'java HelloWorld'
            }
        }
    }
    post {
        success { echo 'Pipeline executed successfully!' }
        failure { echo 'Pipeline failed. Check logs for details.' }
    }
}
```

Steps to create pipeline in UI

- Open Jenkins Dashboard → **New Item**
- Enter name (e.g., HelloWorldPipeline)
- Choose **Pipeline** → Click OK
- Paste the pipeline script into the **Pipeline Script** box → Save → **Build Now**.