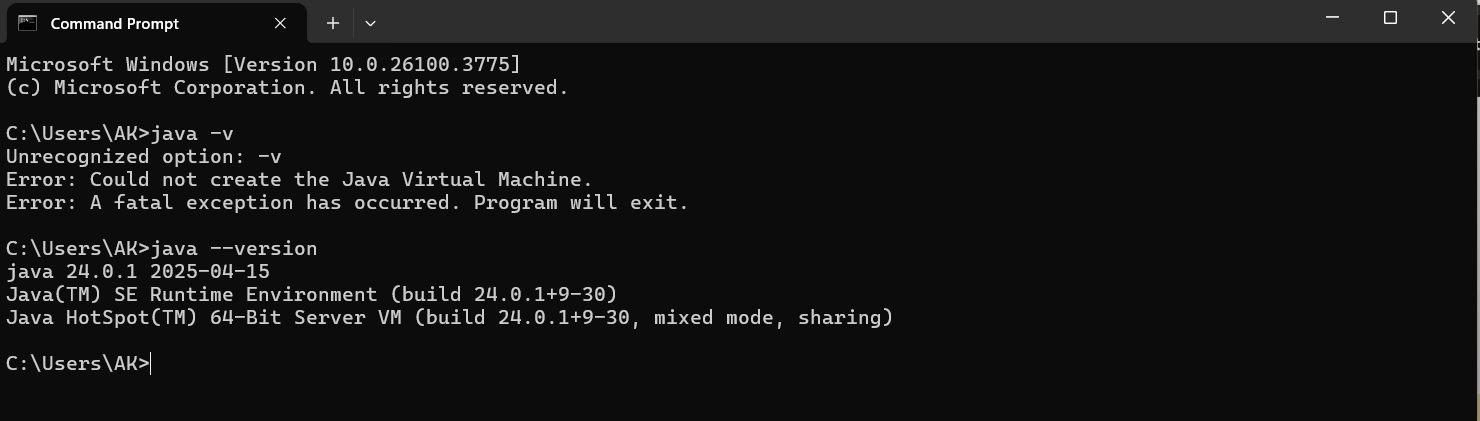
# **Jenkins Setup Guide with JDK 21**

## **Step 1: Check JDK Version**

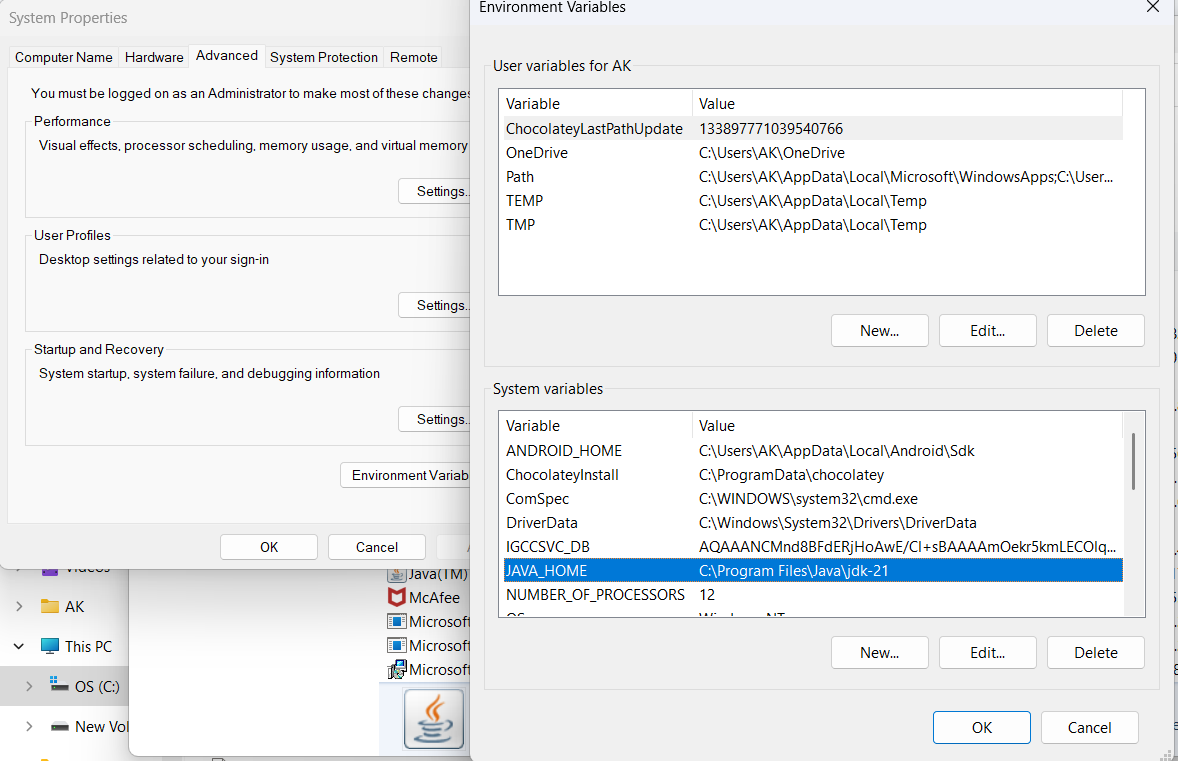
1. Open **Command Prompt**.
2. Run the following command to check the installed JDK version:  
   **java —-version**
3. If the output shows **JDK 21**, proceed to the next step.  
   [](https://download.oracle.com/java/21/archive/jdk-21.0.6_windows-x64_bin.exe)
4. If a **lower or higher version** is installed, uninstall it and download **JDK 21**, as it has Long-Term Support (LTS).

## **Step 2: Download and Install JDK 21**

* Download the JDK 21 installer from:  
  <https://download.oracle.com/java/21/archive/jdk-21.0.6_windows-x64_bin.exe>
* Run the installer and complete the installation process.
* Set the **Environment Variable**:

## 

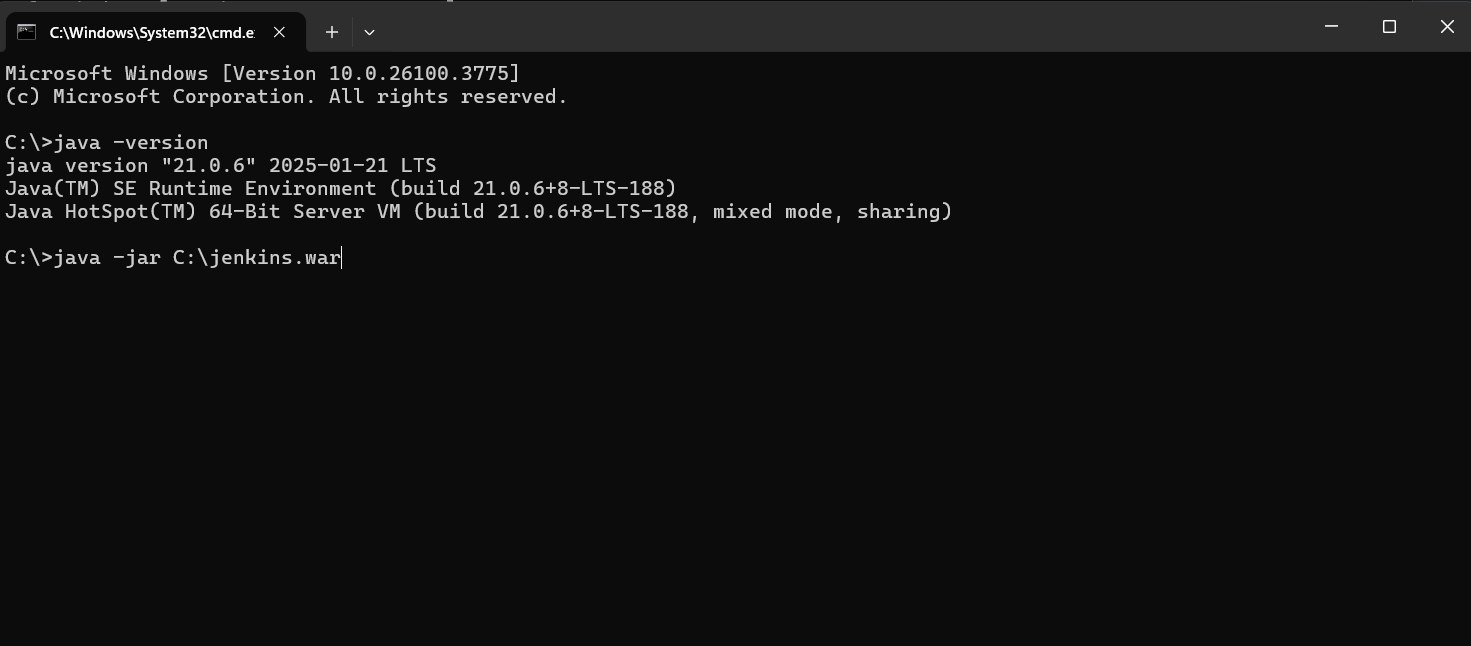
* 1. Open **System Properties** > **Environment Variables**.
  2. Under **System Variables**, find Path and click **Edit**.
  3. Add the JDK 21 bin directory path (e.g., C:\Program Files\Java\jdk-21)



## **Step 3: Download Jenkins WAR File**

* Download the **Jenkins.war** file from:

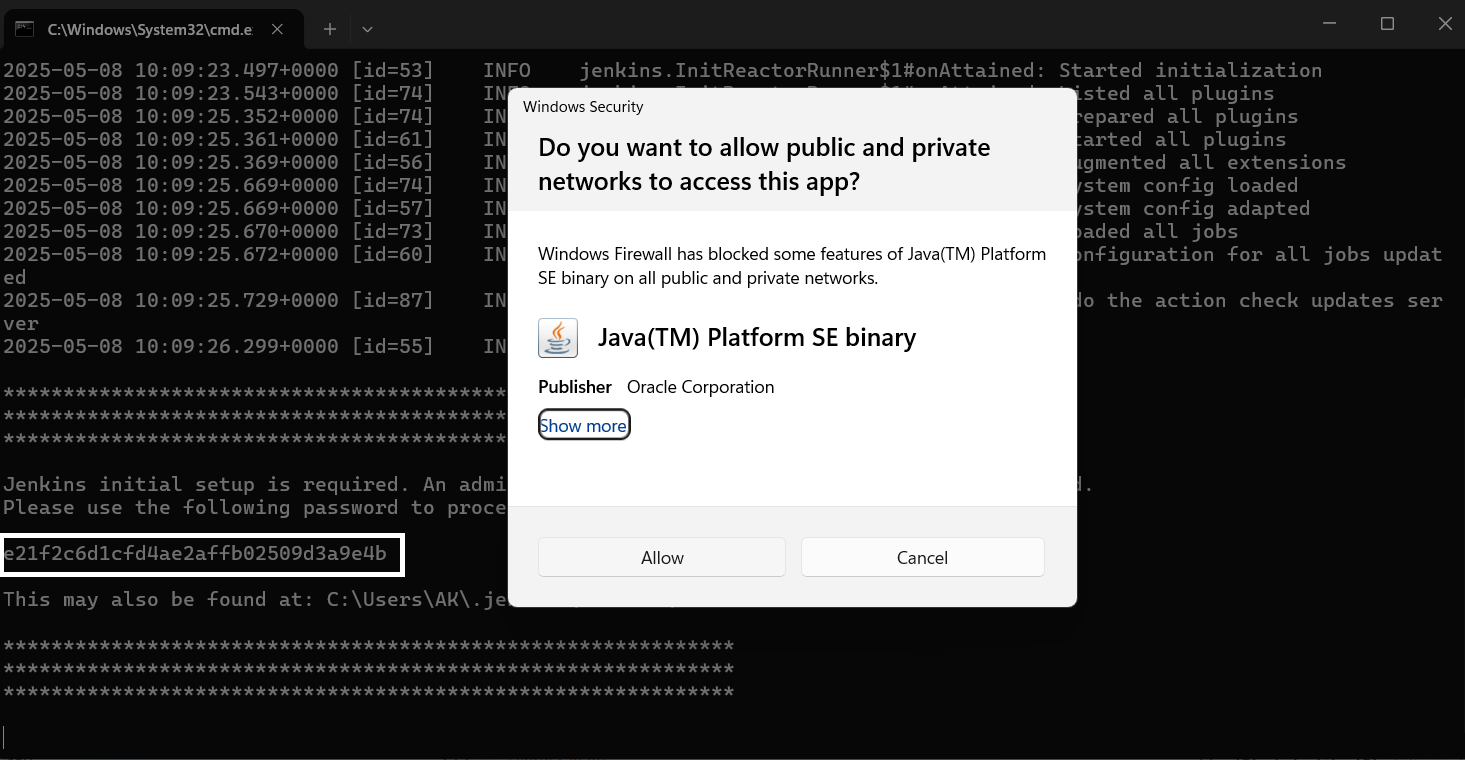
<https://updates.jenkins.io/download/war/2.462.1/jenkins.war>



* Save the file to a known location, e.g., C:\.

## **Step 4: Run Jenkins**

Open **Command Prompt** from the folder where the **jenkins.war** file is located:

1. **C:\java -jar C:\jenkins.war**
2. Wait for Jenkins to start. An **Administrator password** will be generated in the console output.enter the password.  
   When prompted, paste the **Administrator password** copied from the CMD output.
3. Follow the setup wizard:
   1. Install suggested plugins.
   2. Create the first admin user when prompted.

## **Step 5: Access Jenkins**

* Open a browser and go to:<http://localhost:8080/> to access the jenkins.
* Make sure that before accessing jenkins every time run command **C:\java -jar C:\jenkins.war in cmd**

**Step 6: Run a Program Using Jenkins (Local Setup)**

* Open Jenkins at<http://localhost:8080/>.
* Click on **“New Item”** in the Jenkins dashboard.
* Enter a name for your job.
* Select **“Freestyle project”** as the project type.
* Click **OK** to proceed.
* In the configuration page:
* Set up your **Source Code Management**, if applicable (e.g., Git).
* Configure your **Build Triggers** as needed.
* Under the **Build** section, add the necessary build steps (e.g., “Execute Windows batch command” or “Invoke Gradle Script”).
* Click **Save** to store the configuration.
* Click **Build Now** to run the job.

## **Step 7: Configure Build and Trigger Methods**

### **Add a Build Step**

* Under the **Build** section, add the Execute Windows batch command
* Ways to trigger the job is described below
* **schtasks /run /tn "run automation script"** this will run the job from the windows task scheduler
* “C:\Users\iray\source\repos\Automation\ReqnrollTestMP\ReqnrollTestMP>dotnet test” directly runs the code which is in the specified directory just like running the test from the CMD
* cd"C:\Users\iray\source\repos\Automation\ReqnrollTestMP\ReqnrollTestMP\Support\runautomation.bat”directly runs the code which is in the batch file.

**Triggering the Jenkins Job at particular instinct**

* Trigger from Windows Task Scheduler at a Specific Time
* Schedule Job at a Specific Time (Using Jenkins Cron)
* In the job configuration under **Build Triggers**, check **"Build periodically"** and enter a cron expression.  
   For example, to run the job at 3:30 PM every day:
* 0 9 \* \* 1 This is CORN syntax to run on Every Monday at 9:00 AM

| **30** | **15** | **\*** | **\*** | **\*** |
| --- | --- | --- | --- | --- |
| **MIN** | **HOUR** | **Date** | **Month** | **Day(ex:0 = sunday)** |
| **0-59** | **0-23** | **1-31** | **1-12** | **0-6** |

**Run a Program Using Jenkins (Via git repository)**

* Click on **“New Item”** in the Jenkins dashboard.
* Enter a name for your job.
* Select **“Freestyle project”** as the project type.
* Click **OK** to proceed.
* In the configuration page:
* Set up your **Source Code Management** enter your git repository link
* Specify the particular branch you need to work on
* Under the **Build** section, add the necessary build steps (e.g., “Execute Windows batch command” or “Invoke Gradle Script”).
* Here used some build triggers to Move to Project Directory

Used cd ReqnrollTestMP\ReqnrollTestMP

dotnet restore

dotnet build

dotnet test

* Click **Save** to store the configuration.
* Click **Build Now** to run the job.