

Install and Run Jenkins

simple jenkins step by step add and run a pipeline(windows) very detailed

Here's a **very detailed, step-by-step guide** to add and run a Jenkins pipeline on Windows. This covers installation, setup, and running a basic pipeline using **Jenkins Pipeline (Declarative syntax)**.

✓ 1. Install Jenkins on Windows

Step 1.1 – Install Java (Jenkins prerequisite)

1. Download Java JDK (e.g., JDK 17) from:
<https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html>
2. Install it.
3. Set Java environment variables:
 - Open **System Properties** → **Environment Variables**.
 - Under System Variables:
 - Add new variable:
Name: JAVA_HOME
Value: C:\Program Files\Java\jdk-17 (or your install path)
 - Edit **Path** → Add: %JAVA_HOME%\bin
4. Open a Command Prompt and run:

```
bash
```

```
java -version
```

You should see the Java version.

Step 1.2 – Install Jenkins

1. Go to <https://www.jenkins.io/download/>
 2. Click **Windows** → Download the `.msi` installer.
 3. Run the installer:
 - Choose port (default: 8080)
 - Install as a Windows service
 - Choose Java path (if needed)
-

Step 1.3 – Initial Jenkins Setup

1. After install, open `http://localhost:8080`
 2. Unlock Jenkins:
 - Go to: `C:\Program Files\Jenkins\secrets\initialAdminPassword`
 - Open that file and copy the password.
 - Paste into the web browser prompt.
 3. Install **Suggested Plugins** (recommended)
 4. Create an admin user when prompted
-

2. Create Your First Pipeline Job

Step 2.1 – Open Jenkins Dashboard

1. Go to `http://localhost:8080`
2. Click "New Item"

Step 2.2 – Create a Pipeline Project

1. Enter a name like `MyFirstPipeline`
 2. Select **Pipeline**
 3. Click **OK**
-

Step 2.3 – Define Your Pipeline Script

1. Scroll down to the **Pipeline** section
2. In **Definition**, select: **Pipeline script**
3. Paste this simple pipeline script:

```
groovy

pipeline {
    agent any

    stages {
        stage('Build') {
            steps {
                echo 'Building...'
            }
        }
        stage('Test') {
            steps {
                echo 'Testing...'
            }
        }
        stage('Deploy') {
            steps {
                echo 'Deploying...'
            }
        }
    }
}
```

4. Click **Save**

3. Run the Pipeline

Step 3.1 – Trigger the Job

1. On the left, click "**Build Now**"
2. A build will appear in the **Build History**

3. Click on the build number (e.g., #1)
4. Click **Console Output** to see the step-by-step logs

You'll see:

```
bash

[Pipeline] echo
Building...
[Pipeline] echo
Testing...
[Pipeline] echo
Deploying...
```

✅ 4. (Optional) – Use a Jenkinsfile from GitHub

You can also pull the pipeline from a Git repo.

Step 4.1 – Create a GitHub repo with a Jenkinsfile

Example Jenkinsfile :

```
groovy

pipeline {
    agent any
    stages {
        stage('Hello') {
            steps {
                echo 'Hello from Git!'
            }
        }
    }
}
```

Step 4.2 – Configure the Pipeline to Use SCM

1. In your Jenkins job:
 - Go to **Configure**

- Under **Pipeline** section:
 - Set **Definition** to: `Pipeline script from SCM`
 - SCM: Git
 - Repository URL: `https://github.com/yourusername/your-repo.git`
 - Script Path: `Jenkinsfile` (default)

2. Save and click **Build Now**

✓ 5. Tips & Troubleshooting

Problem	Solution
Jenkins not starting	Ensure port 8080 is free. Restart service: <code>services.msc</code> → <code>Jenkins</code>
Git not found	Install Git for Windows and add it to PATH
Environment variables not working	Restart machine after setting JAVA_HOME or PATH

Let me know if you want:

- A Jenkinsfile template for real projects (Node, Python, Java, etc.)
- How to trigger builds from Git pushes
- How to set up agents/nodes on Windows or Linux