

## **DAY-1**

### **DEVOPS**

#### **Installing and Setting Up WSL with Ubuntu on Windows 10**

##### **Step 1: Enable WSL**

Before installing Ubuntu, ensure that WSL is enabled on your Windows system. **Enable WSL Feature**

1. Open **PowerShell** as Administrator and run:
2. `wsl --install`

##### **Step 2: Install Ubuntu**

1. Open **Command Prompt** or **PowerShell** and run:
2. `wsl --install -d Ubuntu`
3. `wsl.exe -d Ubuntu`

##### **Step 3: Set Up Ubuntu**

When Ubuntu runs for the first time, it will ask you to create a new user account.

1. **Enter a username** (must start with a lowercase letter or underscore, and contain only lowercase letters, digits, underscores, and dashes).
2. **Set a password** (enter and confirm the password). If passwords do not match, you will need to retry.
3. Once successful, Ubuntu will be set up and ready to use.

#### **Install Jenkins on Ubuntu**

# Update package lists

`sudo apt update -y`

# Install Java (Required for Jenkins) `sudo apt install -y openjdk-17-jdk`

```
#verify java
version java -
version
```

### **Add Jenkins GPG Key**

```
wget -q -O- https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee
/usr/share/keyrings/jenkins-keyring.asc > /dev/null
```

### **Add the Jenkins Repository**

```
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]
https://pkg.jenkins.io/debian-stable binary/" | sudo tee /etc/apt/sources.list.d/jenkins.list
> /dev/null
```

### **# Install Jenkins**

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

### **# Start and enable Jenkins**

```
service sudo systemctl start
jenkins
```

## **Step 4: Add Jenkins Repository Key**

### **Step 4.1: Add Jenkins GPG Key**

```
wget -q -O- https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo
tee /usr/share/keyrings/jenkinskeyring.asc > /dev/null
```

### **Step 4.2: Add Jenkins Repository**

```
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian
stable binary/" |
sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null
```

## **Step 5: Install Jenkins**

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

## Step 6: Start and Enable Jenkins Service

```
sudo systemctl start jenkins
```

```
sudo systemctl enable jenkins
```

## Step 7: Check Jenkins Status

```
sudo systemctl status jenkins
```

cat copy the localhost:8080 admin path

display the password

=>By default, Jenkins runs as a system user (jenkins). If your script requires sudo, you must allow the Jenkins user to run commands without a password.

```
sudo visudo
```

Add the following line at the end:

```
jenkins ALL=(ALL) NOPASSWD: ALL
```

Save and exit.

```
jeeva@jeeva:~$ sudo apt update -y
Ign:1 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:2 https://pkg.jenkins.io/debian-stable binary/ Release
Hit:4 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:6 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:7 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
13 packages can be upgraded. Run 'apt list --upgradable' to see them.
jeeva@jeeva:~$ sudo apt install -y jenkins
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  jenkins
0 upgraded, 1 newly installed, 0 to remove and 13 not upgraded.
Need to get 94.8 MB of archives.
After this operation, 96.8 MB of additional disk space will be used.
Get:1 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.492.2 [94.8 MB]
Fetched 94.8 MB in 3min 7s (588 kB/s)
Selecting previously unselected package jenkins.
(Reading database ... 42910 files and directories currently installed.)
Preparing to unpack .../jenkins_2.492.2_all.deb ...
Unpacking jenkins (2.492.2) ...
Setting up jenkins (2.492.2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /usr/lib/systemd/system/jenkins.service.
jeeva@jeeva:~$ sudo systemctl start jenkins
jeeva@jeeva:~$ sudo systemctl enable jenkins
Synchronizing state of jenkins.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable jenkins
jeeva@jeeva:~$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: enabled)
   Active: active (running) since Tue 2025-03-18 05:41:05 UTC; 37s ago
     Main PID: 9986 (java)
       Tasks: 56 (limit: 4585)
      Memory: 722.6M ()
     CGroup: /system.slice/jenkins.service
             └─9986 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Mar 18 05:40:34 Jeeva jenkins[9986]: b0e507d6b0f14897ba848a5e1dd67f6d
Mar 18 05:40:34 Jeeva jenkins[9986]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Mar 18 05:40:34 Jeeva jenkins[9986]: *****
Mar 18 05:40:34 Jeeva jenkins[9986]: *****
Mar 18 05:40:54 Jeeva jenkins[9986]: 2025-03-18 05:40:54.806+0000 [id=70] INFO h.m.DownloadService$Downloadable#load: Obtained the updated data file for hudson.tasks.Maven.M
Mar 18 05:40:54 Jeeva jenkins[9986]: 2025-03-18 05:40:54.806+0000 [id=70] INFO hudson.util.Retrier#start: Performed the action check updates server successfully at the attempt
Mar 18 05:41:05 Jeeva jenkins[9986]: 2025-03-18 05:41:05.107+0000 [id=47] INFO jenkins.InitReactorRunner$1#onAttained: Completed initialization
```



If you're on the same machine as Jenkins, use:

`http://localhost:8080`

## **Step-by-Step Guide to Creating a Freestyle Job in Jenkins to Install Nginx Step 1: Create a New Freestyle Job**

1. Click on **New Item** from the Jenkins Dashboard.
2. Enter a name for the job, e.g., *Install-Nginx*.
3. Select **Freestyle project**.
4. Click **OK**.

## **Step 2: Configure the Job**

### **Add Build Step**

1. Scroll down to **Build** → Click *Add build step* → Select **Execute shell**.
2. Paste the following script in the command box:

```
echo "Updating package lists..."
sudo apt update -y
echo "Installing Nginx..."
sudo apt install -y nginx
echo "Starting Nginx service..."
sudo systemctl start nginx
echo "Enabling Nginx to start on boot..."
sudo systemctl enable nginx
echo "Nginx Installation Completed!"
```

## **Step 3: Save and Run the Job**

1. Click **Save**.
2. Click **Build Now**.
3. Check the **Console Output** to verify the installation.

## **Step 4: Verify the Installation**

### **1. Check Nginx Status**

```
systemctl status nginx
```

If running, you should see output like *"active (running)"*.

### **2. Open Nginx in Browser**

http://localhost:80

You should see the default Nginx welcome page

Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Getting Started

Getting Started

<input checked="" type="checkbox"/> Folders	<input checked="" type="checkbox"/> OWASP Markup Formatter	<input type="checkbox"/> Build Timeout	<input type="checkbox"/> Credentials Binding	<div><div>** Ionicons API</div><div>Folders</div><div>OWASP Markup Formatter</div><div>** ASM API</div><div>** JSON Path API</div><div>** Struts</div><div>** Pipeline: Step API</div><div>** Token Macro</div></div>
<input type="checkbox"/> Timestampers	<input type="checkbox"/> Workspace Cleanup	<input type="checkbox"/> Ant	<input type="checkbox"/> Gradle	
<input type="checkbox"/> Pipeline	<input type="checkbox"/> GitHub Branch Source	<input type="checkbox"/> Pipeline: GitHub Groovy Libraries	<input type="checkbox"/> Pipeline Graph View	
<input type="checkbox"/> Git	<input type="checkbox"/> SSH Build Agents	<input type="checkbox"/> Matrix Authorization Strategy	<input type="checkbox"/> PAM Authentication	
<input type="checkbox"/> LDAP	<input type="checkbox"/> Email Extension	<input type="checkbox"/> Mailer	<input type="checkbox"/> Dark Theme	

\*\* - required dependency

Jenkins 2.501

# Jenkins is ready!

You have skipped the **setup of an admin user**.

To log in, use the username: "admin" and the administrator password you used to access the setup wizard.


You have skipped the configuration of the Jenkins URL.

To configure the Jenkins URL, go to "Manage Jenkins" page.

Your Jenkins setup is complete.


Start using Jenkins


Jenkins 2.501


 **Jenkins**

Dashboard >

+ New Item

 Build History

 Manage Jenkins

 My Views

Build Queue

No builds in the queue.

Build Executor Status

0/2

+

Add description

## Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

### Start building your software project

Create a job

### Set up a distributed build

Set up an agent

Configure a cloud

Learn more about distributed builds

REST API Jenkins 2.501

Dashboard > Install-Nginx > Configuration

Automate your build process with ordered tasks like code compilation, testing, and deployment.

### Configure

- General
- Source Code Management
- Triggers
- Environment
- Build Steps**
- Post-build Actions

**Execute shell** ?

Command

See the list of available environment variables

```
#!/bin/  
echo "Updating package lists..."  
sudo apt update -y  
echo "Installing Nginx..."  
sudo apt install -y nginx
```

Advanced ▾

Add build step ▾

**Post-build Actions**

Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.

Add post-build action ▾

Save Apply

REST API Jenkins 2.492.2

Dashboard > Install-Nginx

## Install-Nginx

Permalinks

Builds

Today

#1 5:55 AM

REST API Jenkins 2.492.2



