

# DevOps Training

## Day-1

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

#### 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`

This installs the default Linux distribution and enables necessary components.

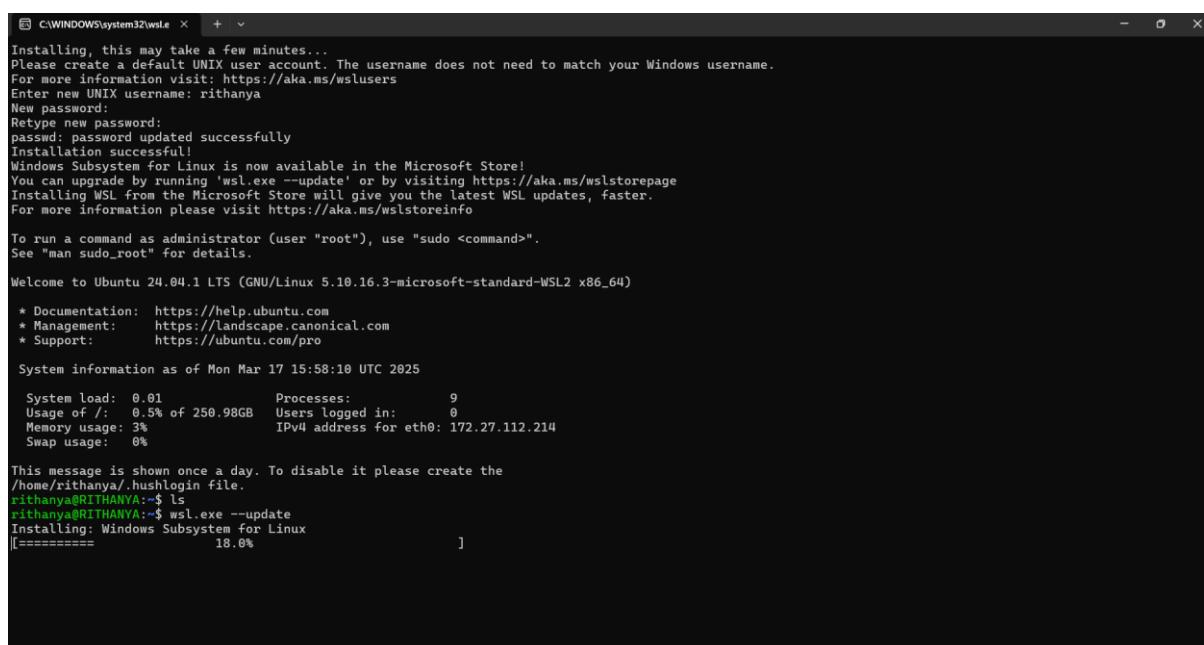
3. If WSL is already installed but not enabled, use:

4. `dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart`

5. Enable the Virtual Machine Platform feature (required for WSL 2):

6. `dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart`

7. Restart your computer to apply changes.



The screenshot shows a Windows terminal window titled 'C:\WINDOWS\system32\wsl.exe'. The window displays the following text:

```
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: rithanya
New password:
Retype new password:
passwd: password updated successfully
Installation successful!
Windows Subsystem for Linux is now available in the Microsoft Store!
You can upgrade by running 'wsl.exe --update' or by visiting https://aka.ms/wslstorepage
Installing WSL from the Microsoft Store will give you the latest WSL updates, faster.
For more information please visit https://aka.ms/wslstoreinfo

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 5.10.16.3-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Mon Mar 17 15:58:10 UTC 2025

 System load:  0.01      Processes:          9
 Usage of /:   0.5% of 250.98GB  Users logged in:     0
 Memory usage: 3%
 Swap usage:  0%
 IPv4 address for eth0: 172.27.112.214

This message is shown once a day. To disable it please create the
/home/rithanya/.hushlogin file.
rithanya@RITHANYA:~$ ls
rithanya@RITHANYA:~$ wsl.exe --update
Installing: Windows Subsystem for Linux
|[=====          18.0%]
```

## Step 2: Install Ubuntu

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

If the installation fails due to timeout issues, retry the command after shutting down WSL:

```
wsl --shutdown
```

```
wsl --install -d Ubuntu
```

3. Once installed, start Ubuntu:
4. `wsl.exe -d Ubuntu`

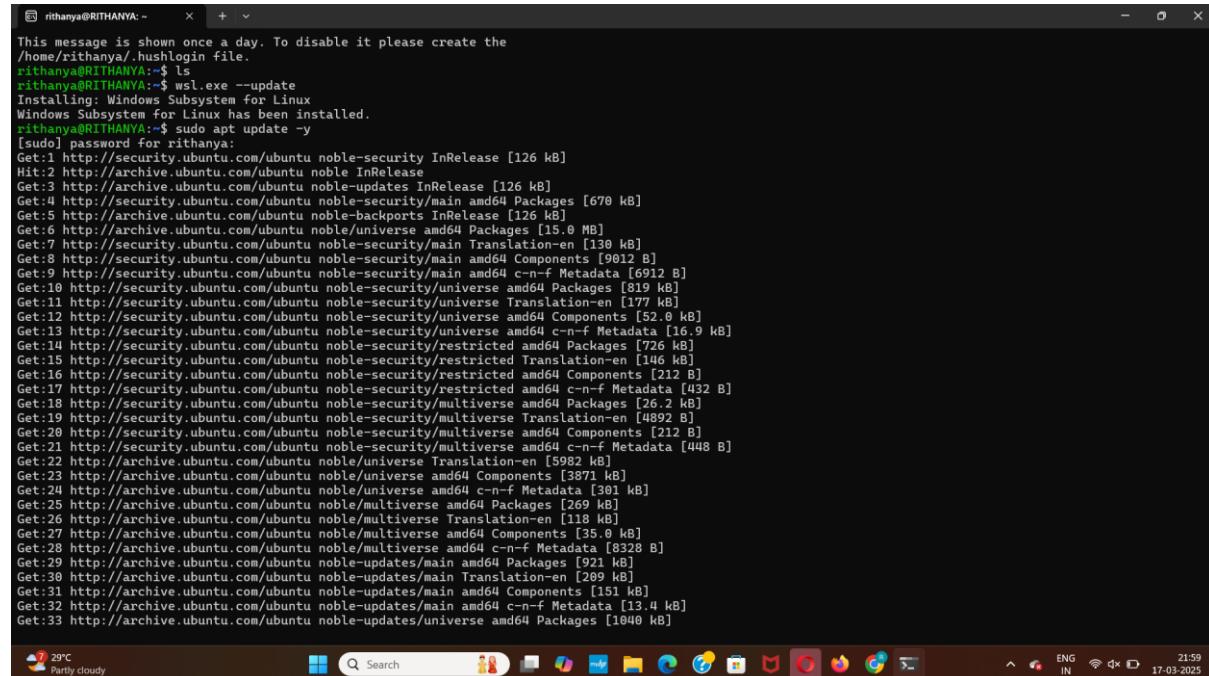
## Step 3: Set Up Ubuntu

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.

## Step 4: Verify Installation

To check the installed distribution



```
This message is shown once a day. To disable it please create the
/home/rithanya/.hushlogin file.
rithanya@RITHANYA:~$ ls
rithanya@RITHANYA:~$ wsl.exe --update
Installing: Windows Subsystem for Linux
Windows Subsystem for Linux has been installed.
rithanya@RITHANYA:~$ sudo apt update
[sudo] password for rithanya:
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:2 http://archive.ubuntu.com/ubuntu noble InRelease
Get:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [670 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.8 MB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [138 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [9012 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [6912 B]
Get:10 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [819 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [177 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.0 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [16.9 kB]
Get:14 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [726 kB]
Get:15 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [146 kB]
Get:16 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:17 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [432 B]
Get:18 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [26.2 kB]
Get:19 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [4892 B]
Get:20 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]
Get:21 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [448 B]
Get:22 http://archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:23 http://archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:24 http://archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [381 kB]
Get:25 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:26 http://archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:27 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.8 kB]
Get:28 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:29 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [921 kB]
Get:30 http://archive.ubuntu.com/ubuntu noble-updates/main Translation-en [209 kB]
Get:31 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [151 kB]
Get:32 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [13.4 kB]
Get:33 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1040 kB]
```

```
wsl -l -v
```

To verify Ubuntu is running:

```
wsl -d Ubuntu
```

## **Step 5: Configure Ubuntu**

### **Update System Packages**

After logging in, update the package list and upgrade installed packages:

```
sudo apt update && sudo apt upgrade -y
```

### **Set Default WSL Version**

To use WSL 2 as the default version for future installations:

```
wsl --set-default-version 2
```

To check the current WSL version:

```
wsl -l -v
```

To convert an existing installation to WSL 2:

```
wsl --set-version Ubuntu 2
```

## **Step 6: Enable .hushlogin to Suppress Login Message**

To disable the daily login message, create a .hushlogin file in your home directory:

```
touch ~/.hushlogin
```

### **Additional Commands**

#### **Restart WSL:**

```
wsl --shutdown
```

#### **Uninstall a Distribution:**

```
wsl --unregister Ubuntu
```

#### **Access Windows Files in WSL:**

```
cd /mnt/c
```

```
ri@rithanya@RITHANYA:~ x + x
Get:54 http://archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
Fetched 32.8 MB in 2min 9s (255 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
130 packages can be upgraded. Run 'apt list --upgradable' to see them.
ri@rithanya@RITHANYA:~$ sudo apt install -y openjdk-17-jdk
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
alsa-topology-conf alsu-ucm-conf ca-certificates-java fonts-dejavu-extra java-common libasound2-data libasound2t64 libatk-wrapper-java
libatk-wrapper-java-jni libgiff7 libice6 libica6 libnss3 libpcsc-lite libpthread-stubs0-dev libsm-dev libsm6 libx11-dev libxau-dev libxaw7
libxcb-shape libxcb1-dev libxdmp-dev libxf2t libxkbfile1 libxml2 libxpm4 libxt-dev libxt6t64 libxv1 libxxf86dgal openjdk-17-jdk-headless
openjdk-17-jre openjdk-17-jre-headless x11-utils x11proto-dev xorg-sgml-doctools xtrans-dev
Suggested packages:
default-jre alsu-utils libasound2-plugins libice-doc pscsd libsm-doc libx11-doc libxcb-doc libxt-doc openjdk-17-demo openjdk-17-source visualvm
libnss-mdns fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei | fonts-wqy-zenhei fonts-indic mesa-utils
Recommended packages:
luit
The following NEW packages will be installed:
alsa-topology-conf alsu-ucm-conf ca-certificates-java fonts-dejavu-extra java-common libasound2-data libasound2t64 libatk-wrapper-java
libatk-wrapper-java-jni libgiff7 libice6 libica6 libnss3 libpcsc-lite libpthread-stubs0-dev libsm-dev libsm6 libx11-dev libxau-dev libxaw7
libxcb-shape libxcb1-dev libxdmp-dev libxf2t libxkbfile1 libxml2 libxpm4 libxt-dev libxt6t64 libxv1 libxxf86dgal openjdk-17-jdk
openjdk-17-jdk-headless openjdk-17-jre openjdk-17-jre-headless x11-utils x11proto-dev xorg-sgml-doctools xtrans-dev
0 upgraded, 40 newly installed, 0 to remove and 130 not upgraded.
Need to get 128 MB of archives.
After this operation, 298 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble/main amd64 alsu-topology-conf all 1.2.5.1-2 [15.5 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble/main amd64 libasound2-data all 1.2.11-1build2 [21.0 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble/main amd64 libasound2t64 amd64 1.2.11-1build2 [399 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 alsu-ucm-conf all 1.2.10-1ubuntu5.4 [64.8 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble/main amd64 ca-certificates-java all 20240118 [11.6 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble/main amd64 fonts-dejavu-extra all 2.37-8 [1947 kB]
Get:7 http://archive.ubuntu.com/ubuntu noble/main amd64 java-common all 0.75+exp1 [6798 kB]
Get:8 http://archive.ubuntu.com/ubuntu noble/main amd64 libice6 amd64 2:1.0.19-1build3 [41.4 kB]
Get:9 http://archive.ubuntu.com/ubuntu noble/main amd64 libsm6 amd64 2:1.2.3-1build3 [15.7 kB]
Get:10 http://archive.ubuntu.com/ubuntu noble/main amd64 libxt6t64 amd64 1:1.2.1-1.2build1 [171 kB]
Get:11 http://archive.ubuntu.com/ubuntu noble/main amd64 libxml2 amd64 2:1.1.3-3build2 [47.6 kB]
Get:12 http://archive.ubuntu.com/ubuntu noble/main amd64 libxpm4 amd64 1:3.5.17-1build2 [36.5 kB]
7% [12 libxpm4 2716 B/36.5 kB 7%] 371 kB/s 5min 37s|
```

```
ri@rithanya@RITHANYA:~ x + x
executions in the same directory, define RTFILEPREFIX in the environment.
Doing so may significantly increase files generated.

Requires: curl, xmllint, GNU core utilities.
Optional: xsltproc for output formatting, gpg for encrypted credential.
Optional: wget as alternate for curl when not available.
ri@rithanya@RITHANYA:~$ sudo apt-get install fontconfig openjdk-17-jre
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
fontconfig is already the newest version (2.15.0-1.lubuntu2).
fontconfig set to manually installed
openjdk-17-jre is already the newest version (17.0.14+7-1~24.04).
openjdk-17-jre set to manually installed
The following package was automatically installed and is no longer required:
 liblvm17t64
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ri@rithanya@RITHANYA:~$ sudo apt-get install jenkins
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package jenkins is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source

E: Package 'jenkins' has no installation candidate
ri@rithanya@RITHANYA:~$ wget -O- https://pkg.jenkins.io/debian/jenkins.io-2023.key | sudo tee /usr/share/keyrings/jenkins-keyring.asc > /dev/null
--2025-03-17 17:02:29-- https://pkg.jenkins.io/debian/jenkins.io-2023.key
Resolving pkg.jenkins.io (pkg.jenkins.io)... 151.101.158.133, 2a04:4e42:25::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|151.101.158.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3175 (3.1K) [application/pgp-keys]
Saving to: 'STDOUT'

-          100%[=====] 3.10K --.-kB/s   in 0s
2025-03-17 17:02:31 (28.9 MB/s) - written to stdout [3175/3175]

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

## **Step-by-Step Guide to Creating a Freestyle Job in Jenkins to Install Nginx on a Local Ubuntu VM**

### **Prerequisites for Setting Up a Freestyle Job to Install Nginx in Jenkins**

Before creating the Freestyle Job, ensure that the following prerequisites are met:

#### **1. Install Jenkins on Ubuntu (If Not Installed)**

If Jenkins is not installed on your Ubuntu VM, follow these steps:

##### **Step 1: Update Package Lists**

```
sudo apt update -y
```

##### **Step 2: Install Java (Required for Jenkins)**

```
sudo apt install -y openjdk-17-jdk
```

##### **Step 3: Verify Java Version**

```
java -version
```

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

(Note: The apt-key add command is deprecated in newer Ubuntu versions. Use the correct method below.)

##### **Correct Way to Add Jenkins Repository (Without apt-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
```

## 2. Access Jenkins Web Interface

Jenkins will be available at `http://<VM_IP>:8080`

### To Get the Jenkins Server URL, Follow These Steps:

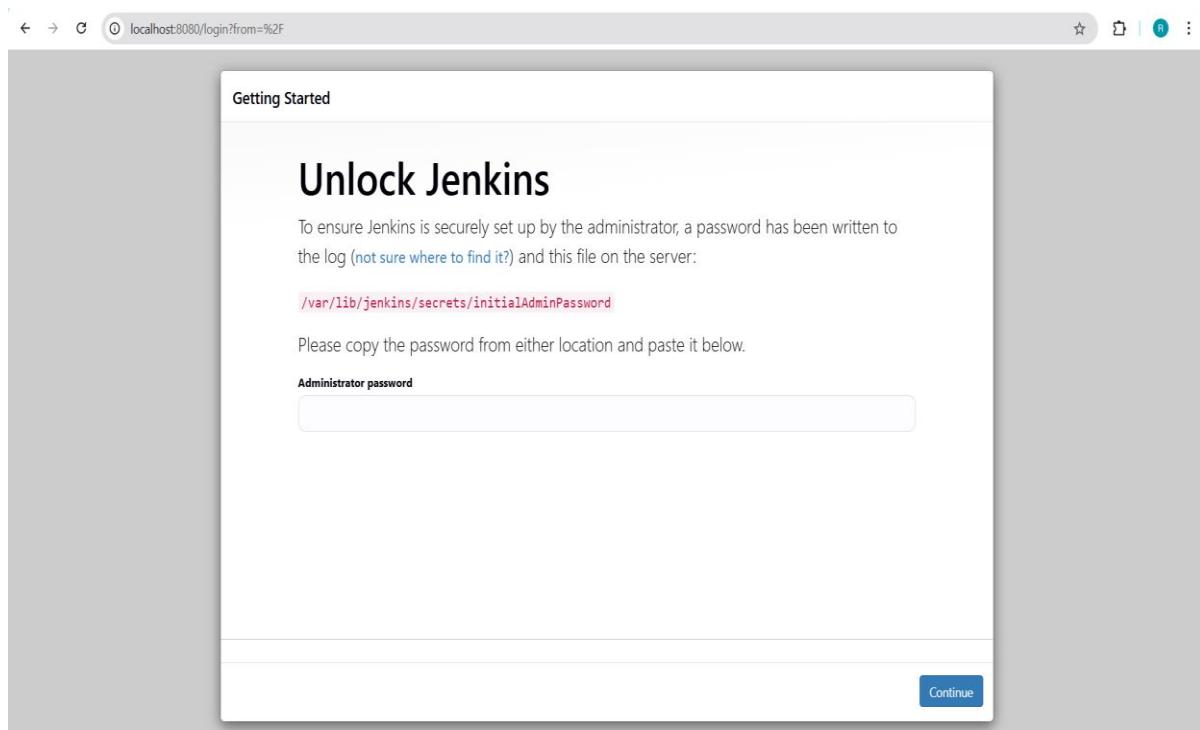
#### Method 1: Check the Default URL

By default, Jenkins runs on port 8080. Open in a browser:

`http://<your-server-ip>:8080`

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

`http://localhost:8080`



#### Method 2: Get Server IP Address

```
hostname -I
```

or

```
ip a | grep inet
```

#### Method 3: Check Jenkins Logs (If Unable to Access)

```
sudo journalctl -u jenkins --no-pager --lines=50
```

Look for lines mentioning "*Jenkins is fully up and running*" and the URL.

### 3. Access Jenkins Web Interface and Log In

1. Open a browser and go to `http://<JENKINS_SERVER_IP>:8080`
2. Enter the username (admin) and the admin password retrieved from the following command:

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

3. Choose *Install Suggested Plugins* (recommended) or manually select plugins.

### 4. Ensure Sudo Access for Jenkins User

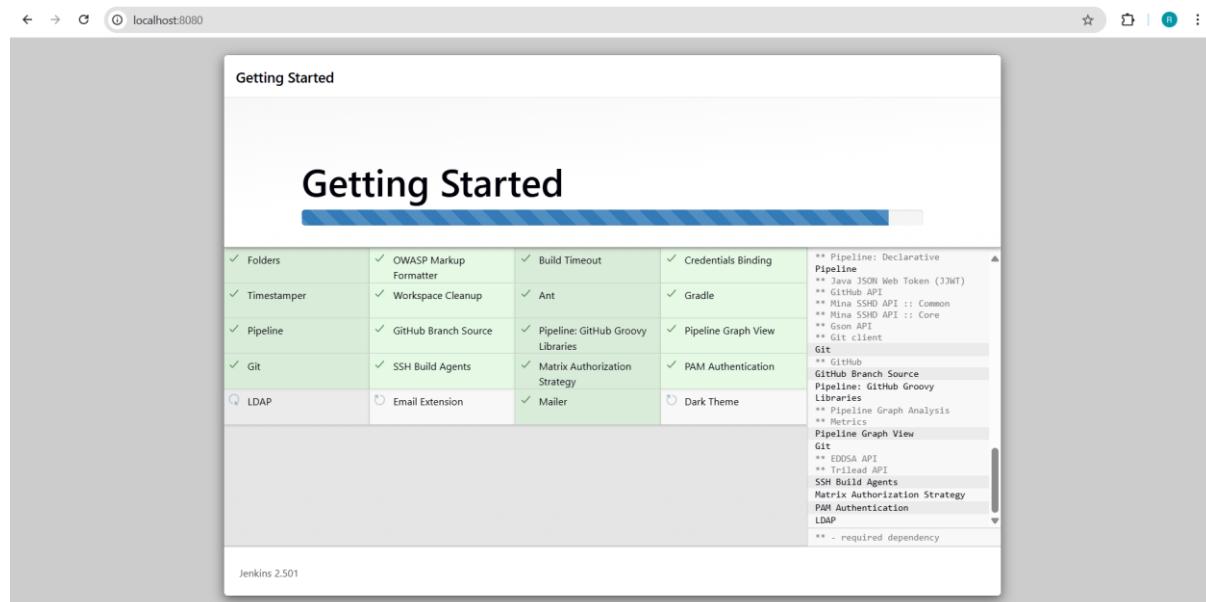
Jenkins runs as a system user (jenkins). If your script requires sudo, allow Jenkins to execute commands without a password:

```
sudo visudo
```

Add the following line at the end of the file:

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

Save and exit.



localhost:8080

Getting Started

Create FIRST ADMIN USER

Username: Rithanya

Password: .....

Confirm password: .....

Full name: Rithanya R R

E-mail address: rithanyarr.24mca@kongu.edu

Jenkins 2.501

Skip and continue as admin

Save and Continue

## 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**.

localhost:8080

Jenkins

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

New Item

Build History

Manage Jenkins

My Views

Build Queue

No builds in the queue.

Build Executor Status

0/2

Create a job

Set up a distributed build

Set up an agent

Configure a cloud

Learn more about distributed builds

Add description

REST API Jenkins 2.501

New Item

Enter an item name

Day1

Select an item type

 Freestyle project  
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

 Pipeline  
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

 Multi-configuration project  
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

 Folder  
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

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:

```
#!/bin/bash
```

```
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!"
```

The screenshot shows the Jenkins job configuration interface for a job named 'Day1'. The 'Build Steps' section is selected. A single step is defined: an 'Execute shell' command. The command is:

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

### Step 3: Save and Run the Job

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

The screenshot shows the Jenkins console output for build #1. The output shows the execution of the build steps, which included updating package lists and installing Nginx. The final message indicates a successful installation.

```
Started by user Rithanya R R
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/Day1
[Day1] $ /bin/bash /tmp/jenkins8786891287006566991.sh
Updating package lists...
sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure an askpass helper
sudo: a password is required
Installing Nginx...
sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure an askpass helper
sudo: a password is required
Starting Nginx service...
sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure an askpass helper
sudo: a password is required
Enabling Nginx to start on boot...
sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure an askpass helper
sudo: a password is required
Nginx Installation Completed!
Finished: SUCCESS
```

## Step 4: Verify the Installation

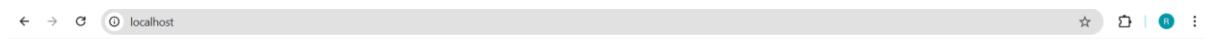
### 1. Check Nginx Status

```
systemctl status nginx
```

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

### 2. Open Nginx in Browser

```
http://<VM_IP>
```



#### Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

