

Day -1 DevOps Training

Installing and Setting Up WSL with Ubuntu on Windows 11

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.

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 distributions and their versions:

```
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:

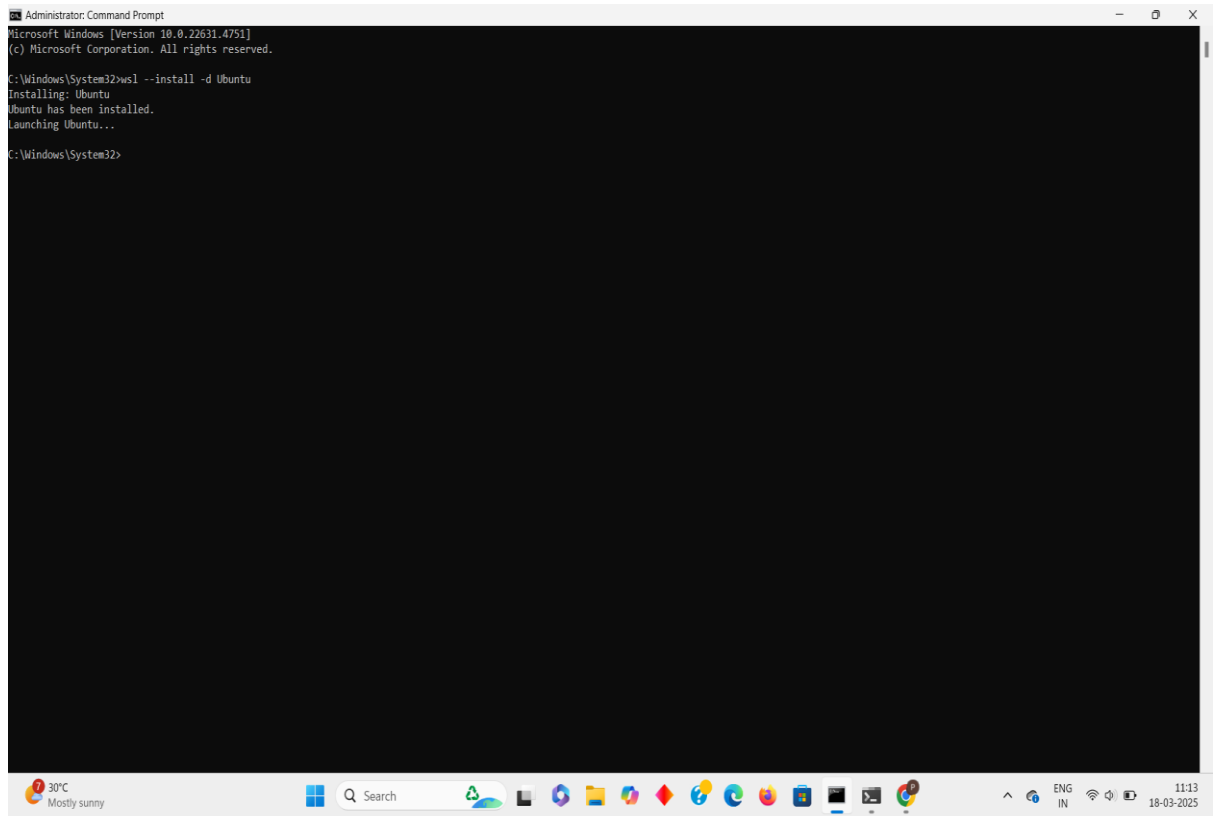
```
wsl --unregister Ubuntu
```

Access Windows Files in WSL:

```
cd /mnt/c
```

Conclusion

You have successfully installed and set up WSL with Ubuntu on Windows 11. You can now use the Ubuntu terminal to run Linux commands and manage your system efficiently.



```
prasath@prasath: ~  
prasath@prasath:~$ java -version  
openjdk version "17.0.14" 2025-01-21  
OpenJDK Runtime Environment (build 17.0.14+7-Ubuntu-124.04)  
OpenJDK 64-Bit Server VM (build 17.0.14+7-Ubuntu-124.04, mixed mode, sharing)  
prasath@prasath:~$ sudo systemctl enable jenkins  
[sudo] password for prasath:  
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  
prasath@prasath:~$ sudo systemctl start jenkins  
prasath@prasath:~$ 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 06:03:30 UTC; 1min 45s ago  
     Main PID: 160 (java)  
       Tasks: 60 (limit: 4550)  
      Memory: 885.0M  
    CGroup: /system.slice/jenkins.service  
            └─160 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080  
  
Mar 18 06:03:19 prasath jenkins[160]: c6d9c42137114984864a6f68f0664650  
Mar 18 06:03:19 prasath jenkins[160]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword  
Mar 18 06:03:19 prasath jenkins[160]: *****  
Mar 18 06:03:19 prasath jenkins[160]: *****  
Mar 18 06:03:19 prasath jenkins[160]: *****  
Mar 18 06:03:19 prasath jenkins[160]: *****  
Mar 18 06:03:30 prasath jenkins[160]: 2025-03-18 06:03:30.618+0000 [id=36] INFO Jenkins.InitReactorRunner$1#onAttained: Completed initializat  
Mar 18 06:03:30 prasath jenkins[160]: 2025-03-18 06:03:30.631+0000 [id=25] INFO hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and r  
Mar 18 06:03:30 prasath systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.  
Mar 18 06:03:32 prasath jenkins[160]: 2025-03-18 06:03:32.864+0000 [id=70] INFO h.m.DownloadService$Downloadable#load: Obtained the updated d  
Mar 18 06:03:32 prasath jenkins[160]: 2025-03-18 06:03:32.865+0000 [id=70] INFO hudson.util.Retrier#start: Performed the action check updates  
lines 1-19/19 (END)
```

Check Java Version

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/debianstable 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

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:

```
#!/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!"
```

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://<VM_IP>
```

You should see the default Nginx welcome page.

Conclusion

You have successfully set up a Jenkins Freestyle Job to install Nginx on a local Ubuntu VM. This guide covers everything from Jenkins installation, configuration, and running the job to verify that Nginx is installed and running correctly.

Now, your Jenkins automation is ready to deploy Nginx effortlessly!

Search

1

1

prasath s

log out

onsole Output

✓ Console Output

DownloadCopyView as plain text

Started by user [prasath s](#)
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/creating job project
[creating job project] \$ /bin/bash /tmp/jenkins4204161475276412329.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

REST APIJenkins 2.492.2

