

## DevOps Training-Day-1

### 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`

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 10. You can now use the Ubuntu terminal to run Linux commands and manage your system efficiently.

```
Microsoft Windows [Version 10.0.22631.4890]
(c) Microsoft Corporation. All rights reserved.

C:\Users\NISANTH>wsl --install -d Ubuntu
Failed to fetch the list distribution from 'https://raw.githubusercontent.com/microsoft/WSL-Distro-List/main/distributionInfo.json'. The operation timed out
Error code: Wsl/InstallDistro/0x80072ee2

C:\Users\NISANTH>wsl --shutdown

C:\Users\NISANTH>wsl --install -d Ubuntu
Downloading: Ubuntu
Installing: Ubuntu
Distribution successfully installed. It can be launched via 'wsl.exe'

C:\Users\NISANTH>wsl.exe -d Ubuntu
Provisioning the new WSL instance Ubuntu
This might take a while...
Create a default Unix user account: Nisanth
Invalid username. A valid username must start with a lowercase letter, digits, underscores, and dashes.
Create a default Unix user account: nisanth
New password:
Retype new password:
Sorry, passwords do not match.
passwd: Authentication token manipulation error
passwd: password unchanged
Try again? [y/N] y
New password:
Retype new password:
```

```
create a default Unix user account: nisanth
New password:
Retype new password:
Sorry, passwords do not match.
passwd: Authentication token manipulation error
passwd: password unchanged
Try again? [y/N] y
New password:
Retype new password:
passwd: password updated successfully
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 5.15.167.4-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 14:03:20 UTC 2025

System load:  0.22               Processes:           31
Usage of /:   0.1% of 1006.85GB   Users logged in:    0
Memory usage: 12%               IPv4 address for eth0: 172.30.33.214
Swap usage:   0%

This message is shown once a day. To disable it please create the
/home/nisanth/.hushlogin file.
nisanth@LAPTOP-G3U57BJM:/mnt/c/Users/NISANTH$ |
```

## 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>](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!





