

## 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.

```
 samyuktha@LAPTOP-2QVFH: ~ + v - □ X □ X
- Filesize:759200 [weak]
- SHA256:eaf738d0e2571414d0519bf8d13f71ad5a10bf3f468403b19d95cd0e69bc61f6
- SHA1:ca2cc0fa16787cb123d4219c057067bab4c1a847 [weak]
- MD5Sum:23fdfff8ca6346a0fb59ea9fb6d4d5136 [weak]
Hashes of received file:
- SHA256:ed31282d021899729175a27ec1af9d9eb9eaad9bca611f7a5f133701e9da6968
- SHA1:dafdf1748907e00561a1eddebd59578172324589e [weak]
- MD5Sum:eeac75ad9c7f039b346e65a979135bfff [weak]
- Filesize:759200 [weak]
Last modification reported: Tue, 11 Mar 2025 12:29:25 +0000
Release file created at: Tue, 18 Mar 2025 02:29:19 +0000
E: Some index files failed to download. They have been ignored, or old ones used instead.
samyuktha@LAPTOP-2QVFH:~$ 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 alsamixer-conf ca-certificates-java fonts-dejavu-extra java-common libasound2-data libasound2t64
libatk-wrapper-java libatk-wrapper-java-jni libgif7 libice-dev libice6 libnspr4 libnss3 libpcslite1
libpthread-stubs0-dev libsm-dev libsm6 libx11-dev libxau-dev libxaw7 libxcb-shape0 libxcb1-dev libxdmcp-dev libxft2
libxkbfile1 libxmu6 libxpm4 libxt-dev libxt6t64 libxv1 libxf86dg1 openjdk-17-jdk-headless openjdk-17-jre
openjdk-17-jre-headless x11-utils x11proto-dev xorg-sgml-doctools xtrans-dev
Suggested packages:
default-jre alsamixer libasound2-plugins libice-doc pcscd 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 alsamixer-conf ca-certificates-java fonts-dejavu-extra java-common libasound2-data libasound2t64
```

```
samyuktha@LAPTOP-2QVVFHTD:~$ sudo apt update -y
[sudo] password for samyuktha:
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]
Ign:6 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages
Ign:7 http://archive.ubuntu.com/ubuntu noble/universe Translation-en
Ign:8 http://archive.ubuntu.com/ubuntu noble/universe amd64 Components
Get:9 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [130 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [96 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [130 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [130 kB]
```

## 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.

### **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! 🚀

The screenshot shows the Jenkins dashboard at [localhost:8080](http://localhost:8080). The top navigation bar includes tabs for "Introducing ChatGPT | OpenAI", "Jenkins service not found", "Linux", "(no subject) - samyukthak24m...", and "Dashboard [Jenkins]". The user "SamyukthaKonguraj" is logged in. The dashboard features a sidebar with links for "New Item", "Build History", "Manage Jenkins", and "My Views". A "Build Queue" section shows "No builds in the queue". A "Build Executor Status" section shows "0/2". The main content area has a heading "Welcome to Jenkins!" followed by a sub-heading: "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." Below this is a section titled "Start building your software project" with three buttons: "Create a job", "Set up a distributed build", and "Set up an agent". The "Set up a distributed build" section includes links for "Configure a cloud" and "Learn more about distributed builds". At the bottom right, there are links for "REST API" and "Jenkins 2.492.2". The taskbar at the bottom shows various icons and the date/time "18-03-2025".

The screenshot shows the Jenkins console output for a build named "sam". The browser tab is "localhost:8080/job/sam/1/console". The left sidebar shows "Status", "Changes", "Console Output" (which is selected), "Edit Build Information", "Delete build '#1'", and "Timings". The main content area is titled "Console Output" and shows the following log output:

```
Started by user SamyukthaKonguraj
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/sam
[sam] $ /bin/sh -xe /tmp/jenkins18209799421305693082.sh
+ echo Updating package lists...
Updating package lists...
+ sudo apt update -y

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Ign:1 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:2 https://pkg.jenkins.io/debian-stable binary/ Release
Hit:3 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:4 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:6 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:7 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists...
Building dependency tree...
Reading state information...
56 packages can be upgraded. Run 'apt list --upgradable' to see them.
+ echo Installing Nginx...
Installing Nginx...
+ sudo apt install -y nginx
```

The taskbar at the bottom shows various icons and the date/time "18-03-2025".



## 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.*