

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 **VirtualMachinePlatform** feature (required for WSL2):
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

Step5: Configure Ubuntu

Update System Packages

After logging in, update the package list and upgrade installed packages: sudo apt update && sudo apt upgrade -y

SetDefaultWSLVersion

To use WSL2 as the default version for future installations:

wsl--set-default-version2

To check the current WSL version:

wsl-l-v

To convert an existing installation to WSL2:

wsl--set-versionUbuntu2

Step6: 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--unregisterUbuntu

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.

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
```

```
jenkins sudo systemctl enable jenkins
```

Step 7: Check Jenkins Status

```
sudo systemctl status jenkins
```

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 net`

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**.

Step2:ConfiguretheJob Add

Build Step

1. Scrolldownto**Build**→Click**Add build step**→Select**Execute shell**.
2. Pastethefollowingscriptinthecommandbox:

```
#!/bin/bash  
echo "Updatingpackagelists..." sudo  
apt update -y
```

```
echo "InstallingNginx..." sudo  
apt install -y nginx
```

```
echo "StartingNginxservice..."  
sudo systemctl start nginx
```

```
echo "EnablingNginxtostartonboot..."  
sudo systemctl enable nginx
```

```
echo "NginxInstallationCompleted!"
```

Step3:SaveandRuntheJob

1. Click**Save**.
2. Click**BuildNow**.
3. Checkthe**ConsoleOutput**toverifythe installation.

Step4:VerifytheInstallation

1. CheckNginx Status

```
systemctlstatusnginx  
Ifrunning, you should seeoutput like "active(running)".
```

2. OpenNginxin 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!

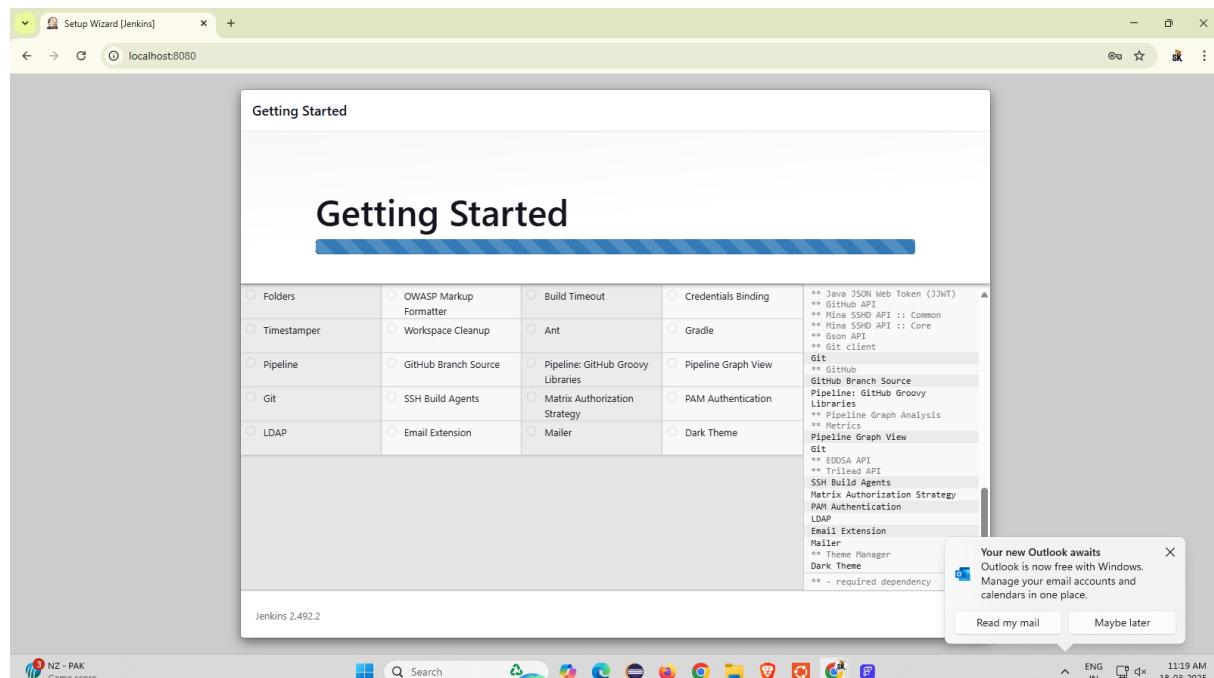
Screenshot

```
student@CTS-6: ~
z * Forward one window (and set window to N).
w * Backward one window (and set window to N).
ESC-SPACE * Forward one window, but don't stop at end-of-file.
d "D * Forward one half-window (and set half-window to N).
u "U * Backward one half-window (and set half-window to N).
ESC-) RightArrow * Right one half screen width (or N positions).
ESC-( LeftArrow * Left one half screen width (or N positions).
ESC-{ ~RightArrow Right to the first column.
ESC-{ ^LeftArrow Forward forever; like "tail -f".
ESC-F Like F but stop when search pattern is found.
r "R ^L Repaint screen.
R Repaint screen, discarding buffered input.

Default "window" is the screen height.

HELP - Press [F1] for help, [F2] for search, [F3] for skipping...
nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
   Active: active (running) since Tue 2025-03-18 06:08:54 UTC; 8s ago
     Docs: man:nginx(8)
  Process: 1728 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
 Process: 1721 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
Main PID: 1723 (nginx)
   Tasks: 17 (limit: 4559)
      Memory: 11.5M
      CGroup: /system.slice/nginx.service
             └─1723 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"

Mar 18 06:08:54 CTS-6 systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
student@CTS-6:~$ hostname -i
172.23.210.62
student@CTS-6:~$
```



The screenshot shows the Jenkins dashboard at localhost:8080. The top navigation bar includes links for 'Dashboard', 'Build History', 'Manage Jenkins', and 'My Views'. A search bar and user information ('Senthilkumar') are also present. The main content area features a 'Welcome to Jenkins!' message and a section titled 'Start building your software project' with links for 'Create a job', 'Set up a distributed build', 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'. On the left, there are sections for 'Build Queue' (No builds in the queue) and 'Build Executor Status' (0/2). The bottom right corner shows 'REST API' and 'Jenkins 2.492.2'.

The screenshot shows the Jenkins console output for a build job named 'test #1'. The URL is localhost:8080/job/test/1/console. The left sidebar lists 'Status', 'Changes', 'Console Output' (which is selected), 'Edit Build Information', 'Delete build #1', and 'Timings'. The main content area displays the console output with a green checkmark icon and the text: 'Started by user Senthilkumar', 'Running as SYSTEM', 'Building in workspace /var/lib/jenkins/workspace/test', and 'Finished: SUCCESS'. There are also 'Download', 'Copy', and 'View as plain text' buttons. The bottom right corner shows 'REST API' and 'Jenkins 2.492.2'.





