DevOps

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

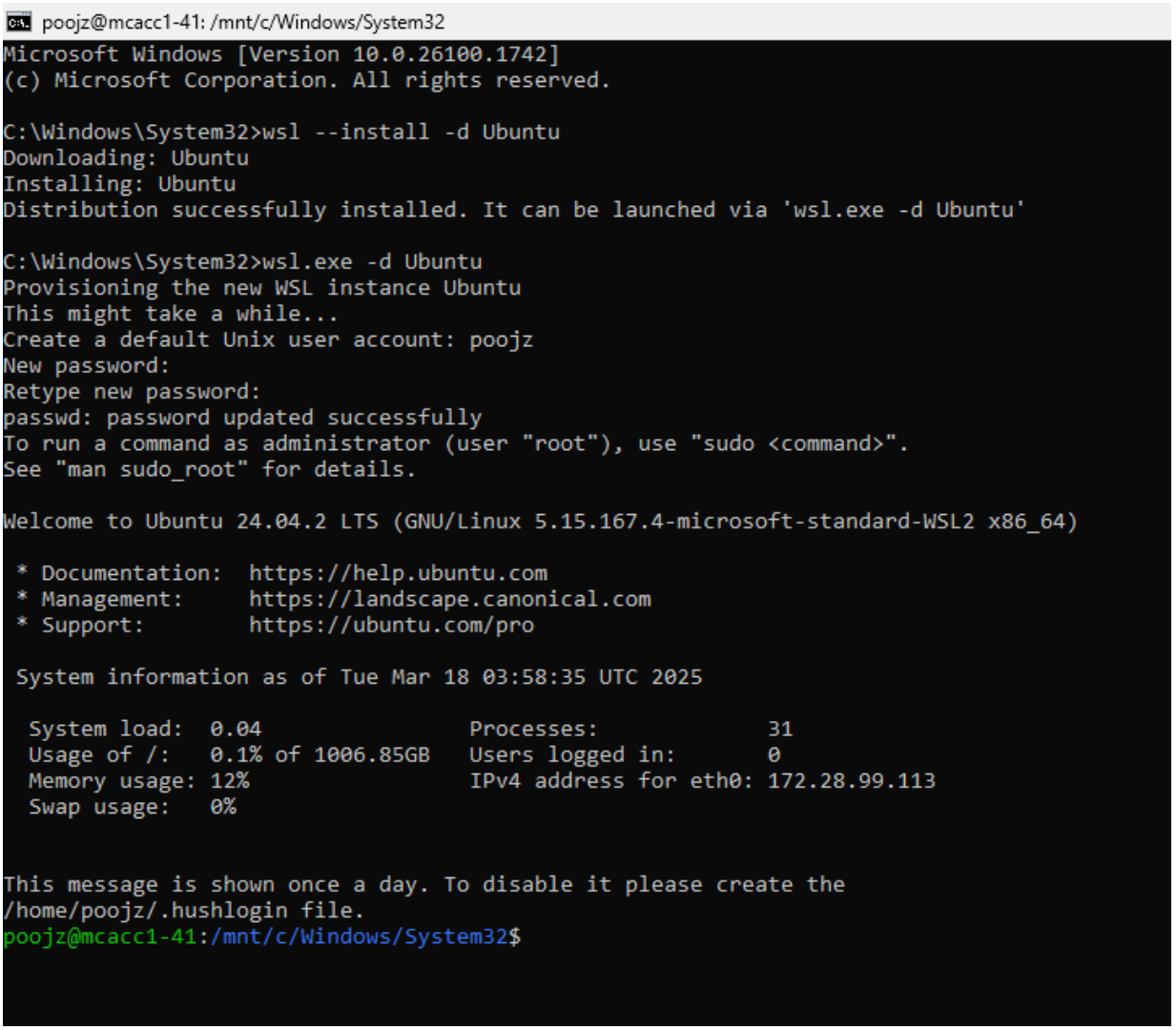
--unregister Ubuntu **Access**

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.


poojz@mcacc1-41: /mnt/c/Windows/System32
Microsoft Windows [Version 10.0.26100.1742]
(c) Microsoft Corporation. All rights reserved.
C:\Windows\System32>wsl --install -d Ubuntu
Downloading: Ubuntu
Installing: Ubuntu
Distribution successfully installed. It can be launched via 'wsl.exe -d Ubuntu'
C:\Windows\System32>wsl.exe -d Ubuntu
Provisioning the new WSL instance Ubuntu
This might take a while...
Create a default Unix user account: poojz
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 Tue Mar 18 03:58:35 UTC 2025
System load: 0.04
Usage of /: 0.1% of 1006.85GB
Memory usage: 12%
Swap usage: 0%
Processes: 31
Users logged in: 0
IPv4 address for eth0: 172.28.99.113
This message is shown once a day. To disable it please create the
/home/poojz/.hushlogin file.
poojz@mcacc1-41:/mnt/c/Windows/System32$

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)"*.

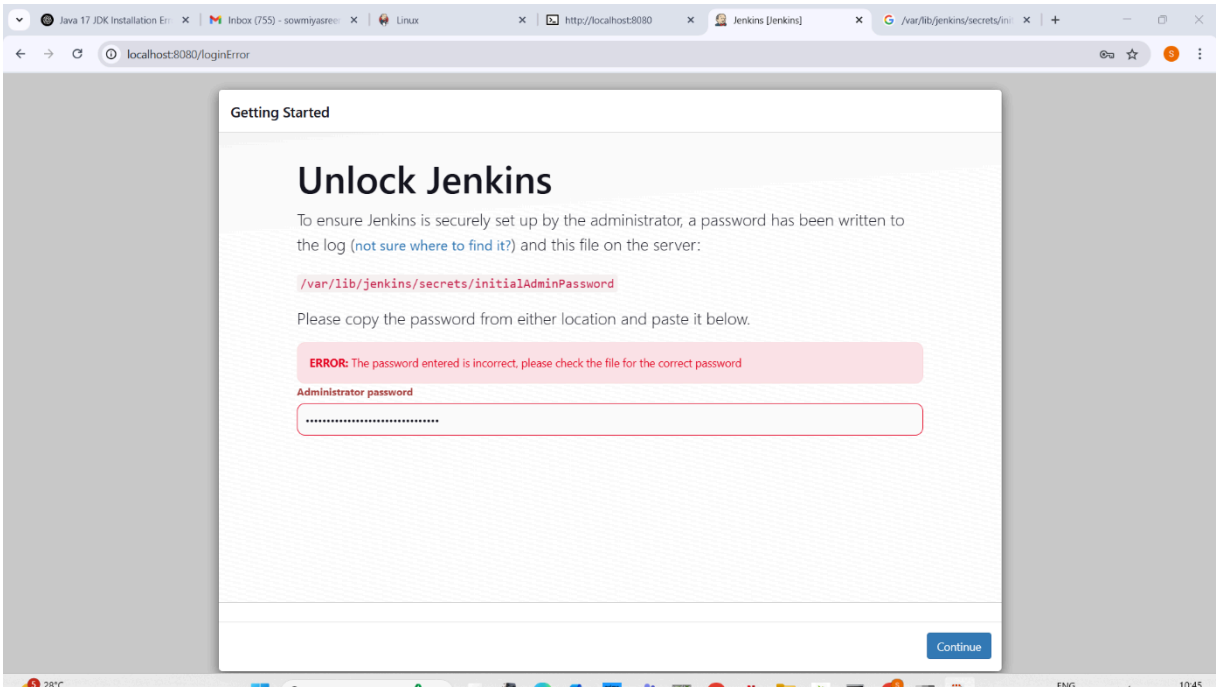
2. Open Nginx in Browser http://<VM_IP>

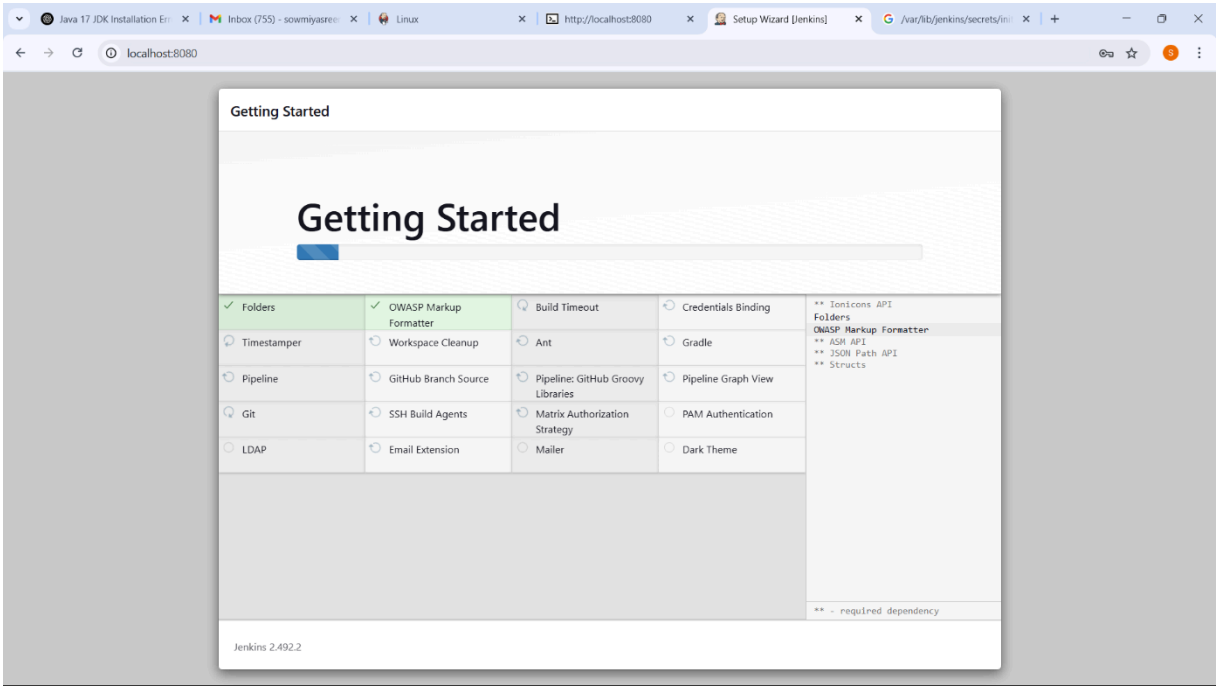
You should see the default Nginx welcome page.

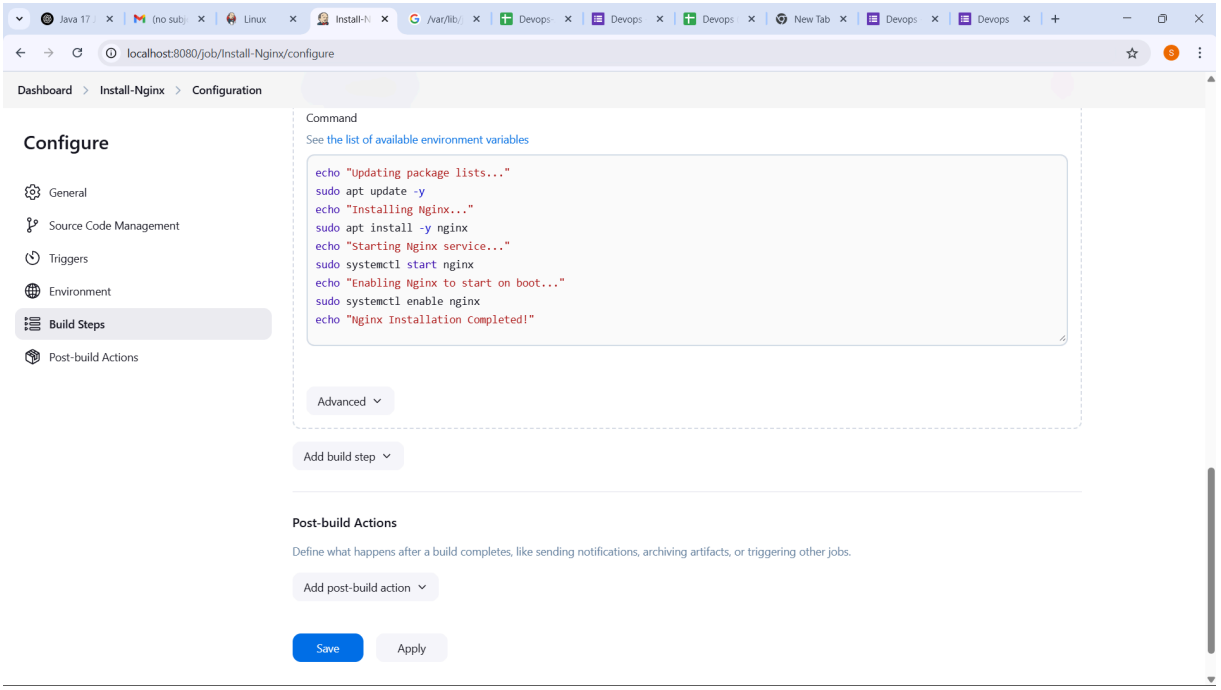
Conclusion

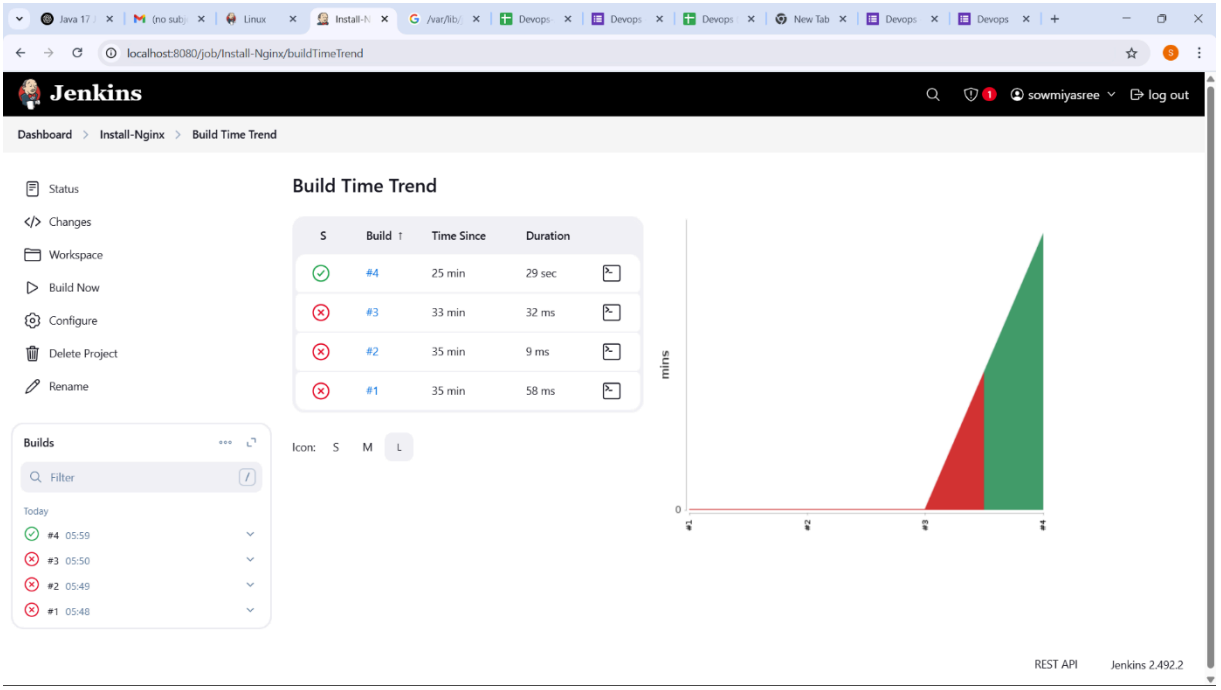
You have successfully set up a Jenkins Freestyle Job to install Nginx on a local Ubuntu VM. This guide covers everything from Jenkins installation, configuration, and running the job to verify that Nginx is installed and running correctly.

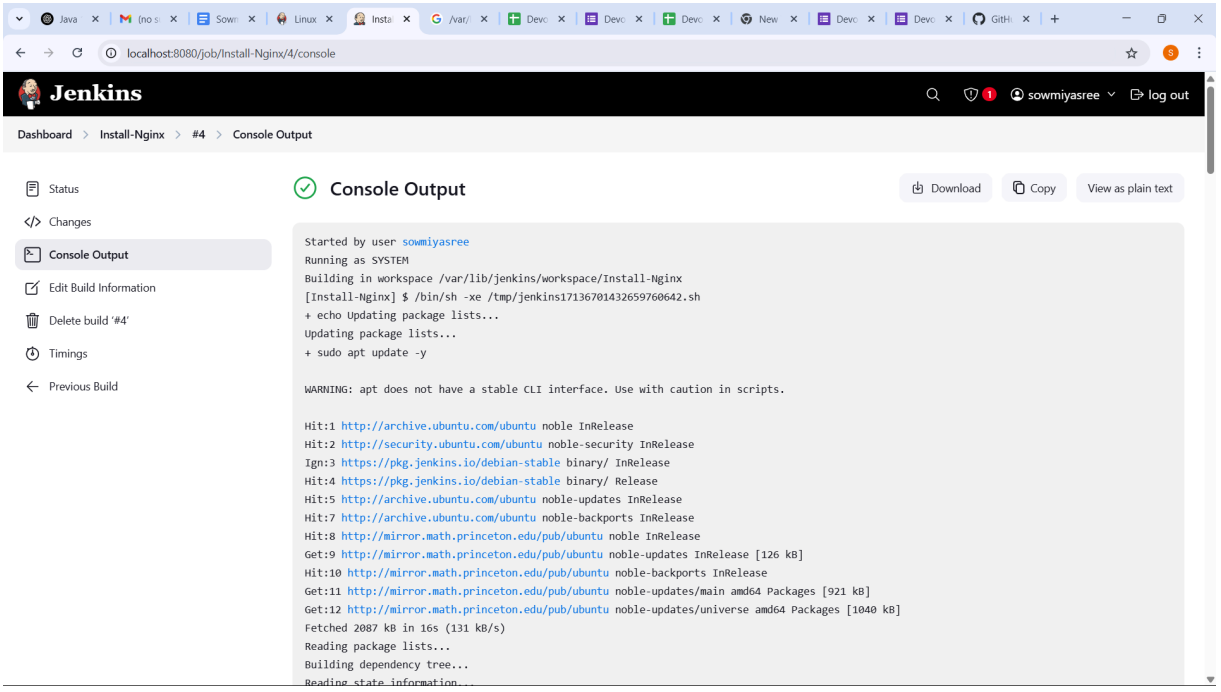
Now, your Jenkins automation is ready to deploy Nginx effortlessly!

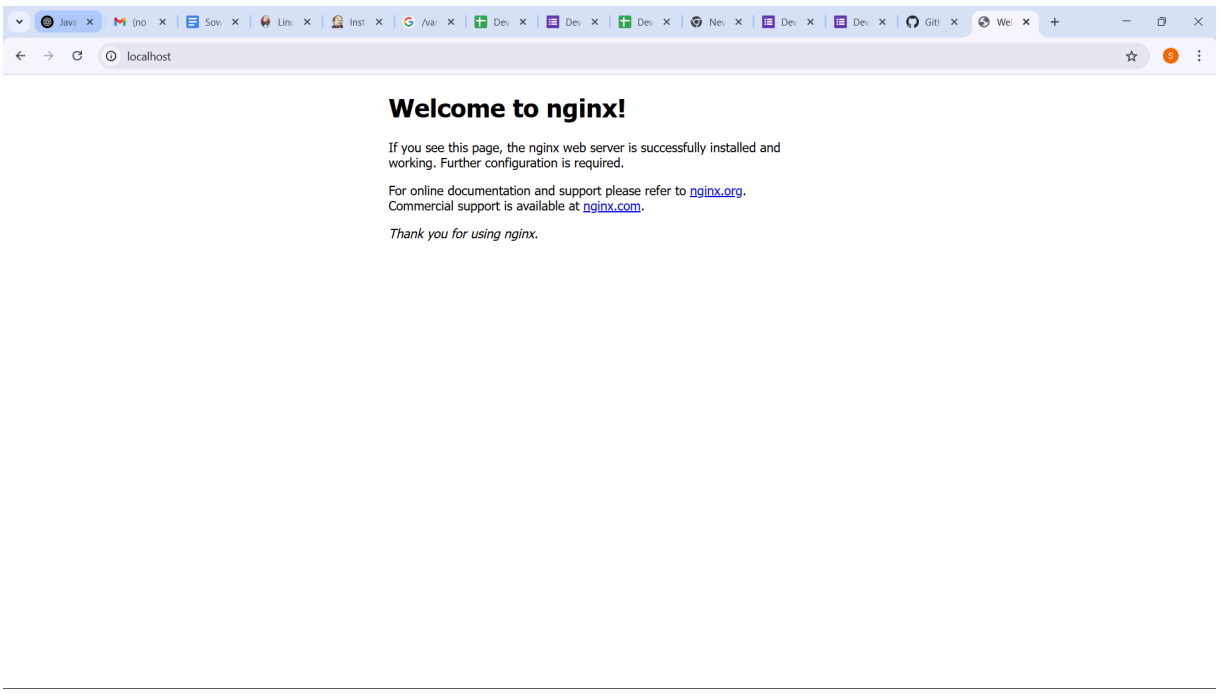



Getting Started
Getting Started
Folders
OWASP Markup Formatter
Build Timeout
Credentials Binding
Timestamper
Workspace Cleanup
Ant
Gradle
Pipeline
GitHub Branch Source
Pipeline: GitHub Groovy Libraries
Pipeline Graph View
Git
SSH Build Agents
Matrix Authorization Strategy
PAM Authentication
LDAP
Email Extension
Mailer
Dark Theme
** Ionicons API
Folders
OWASP Markup Formatter
** ASM API
** JSON Path API
** Structs
** - required dependency
Jenkins 2.492.2


Dashboard
Install-Nginx
Configuration
Configure
General
Source Code Management
Triggers
Environment
Build Steps
Post-build Actions
Command
See the list of available environment variables
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!"
Advanced
Add build step
Post-build Actions
Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.
Add post-build action
Save
Apply


localhost:8080/job/Install-Nginx/buildTimeTrend
Jenkins
sowmiyasree
log out
Dashboard > Install-Nginx > Build Time Trend
Status
Changes
Workspace
Build Now
Configure
Delete Project
Rename
Build Time Trend
S
Build
Time Since
Duration
#4 25 min 29 sec
#3 33 min 32 ms
#2 35 min 9 ms
#1 35 min 58 ms
Icon: S M L
mins
0
Builds
Filter
Today
#4 05:59
#3 05:50
#2 05:49
#1 05:48
REST API
Jenkins 2.492.2


localhost:8080/job/Install-Nginx/4/console
Jenkins
sowmiyasree
log out
Dashboard > Install-Nginx > #4 > Console Output
Status
Changes
Console Output
Edit Build Information
Delete build '#4'
Timings
Previous Build
Console Output
Download
Copy
View as plain text
Started by user sowmiyasree
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/Install-Nginx
[Install-Nginx] $ /bin/sh -xe /tmp/jenkins17136701432659760642.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.
Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://security.ubuntu.com/ubuntu noble-security InRelease
Ign:3 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:4 https://pkg.jenkins.io/debian-stable binary/ Release
Hit:5 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:7 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:8 http://mirror.math.princeton.edu/pub/ubuntu noble InRelease
Get:9 http://mirror.math.princeton.edu/pub/ubuntu noble-updates InRelease [126 kB]
Hit:10 http://mirror.math.princeton.edu/pub/ubuntu noble-backports InRelease
Get:11 http://mirror.math.princeton.edu/pub/ubuntu noble-updates/main amd64 Packages [921 kB]
Get:12 http://mirror.math.princeton.edu/pub/ubuntu noble-updates/universe amd64 Packages [1040 kB]
Fetched 2087 kB in 16s (131 kB/s)
Reading package lists...
Building dependency tree...


localhost
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.
Commercial support is available at nginx.com.
Thank you for using nginx.