

DevOps Training - Day 1

Installing Ubuntu on VirtualBox

Step 1: Download Required Files

1. **Download VirtualBox** from the official website:
[VirtualBox Download](#)
2. **Download Ubuntu ISO** from the official Ubuntu website:
[Ubuntu Download](#)

Step 2: Install VirtualBox

1. Open the downloaded VirtualBox installer.
2. Follow the on-screen instructions to complete the installation.
3. Once installed, launch VirtualBox.

Step 3: Create a New Virtual Machine

1. Click **New** in VirtualBox.
2. Enter a name (e.g., "Ubuntu VM").
3. Set the **Type** to **Linux**.
4. Set the **Version** to **Ubuntu (64-bit)**.
5. Click **Next**.

Step 4: Allocate Memory (RAM)

- Assign at least **2GB (2048 MB)** of RAM (Recommended: **4GB or more**).
- Click **Next**.

Step 5: Create a Virtual Hard Disk

1. Choose **Create a virtual hard disk now** → Click **Create**.
2. Select **VDI (VirtualBox Disk Image)** → Click **Next**.
3. Select **Dynamically allocated** → Click **Next**.
4. Set at least **25GB storage** (Recommended: **50GB or more**).
5. Click **Create**.

Step 6: Attach Ubuntu ISO

1. Select the created VM from the list.
2. Click **Settings** → **Storage**.
3. Under **Controller: IDE**, click **Empty**.
4. Click the **CD icon** on the right → Choose a disk file.
5. Select the downloaded **Ubuntu ISO** file.
6. Click **OK**.

Step 7: Start the Virtual Machine

1. Select your Ubuntu VM → Click **Start**.
2. The Ubuntu installer will launch. Follow the on-screen instructions to install Ubuntu.

Setting Up Jenkins on Ubuntu VM

Step 1: Update Package Lists

sudo apt update -y

Step 2: Install Java (Required for Jenkins)

sudo apt install -y openjdk-17-jdk

```

suji@Suji: ~
sudo apt install udo
suji@Suji:~$ 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 alsa-ucm-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 libpcsc-lite1
  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 libxxf86dgal openjdk-17-jdk-headless openjdk-17-jre
  openjdk-17-jre-headless x11-utils x11proto-dev xorg-sgml-doctools xtrans-dev
Suggested packages:
  default-jre alsa-utils libasound2-plugins libice-doc pscd 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 alsa-ucm-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 libpcsc-lite1
  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 libxxf86dgal openjdk-17-jdk openjdk-17-jdk-headless
  openjdk-17-jre openjdk-17-jre-headless x11-utils x11proto-dev xorg-sgml-doctools xtrans-dev
0 upgraded, 40 newly installed, 0 to remove and 130 not upgraded.
Need to get 128 MB of archives.
After this operation, 298 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble/main amd64 alsa-topology-conf all 1.2.5.1-2 [15.5 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble/main amd64 libasound2-data all 1.2.11-1build2 [21.0 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble/main amd64 libasound2t64 amd64 1.2.11-1build2 [399 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 alsa-ucm-conf all 1.2.10-1ubuntu5.4 [64.8 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble/main amd64 ca-certificates-java all 20240118 [11.6 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble/main amd64 fonts-dejavu-extra all 2.37-8 [1947 kB]
Get:7 http://archive.ubuntu.com/ubuntu noble/main amd64 java-common all 0.75+exp1 [6798 B]
Get:8 http://archive.ubuntu.com/ubuntu noble/main amd64 libice6 amd64 2:1.0.10-1build3 [41.4 kB]
Get:9 http://archive.ubuntu.com/ubuntu noble/main amd64 libsm6 amd64 2:1.2.3-1build3 [15.7 kB]

```

Step 3: Verify Java Installation

java -version

- Expected output:
- openjdk version "17.0.10" 2024-01-16
- OpenJDK Runtime Environment (build 17.0.10+0)

Step 4: Add Jenkins Repository

```
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 13 not upgraded.
suji@suji:~$ sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee /etc/apt/keyrings/jenkins.asc > /dev/null
echo "deb [signed-by=/etc/apt/keyrings/jenkins.asc] https://pkg.jenkins.io/debian-stable binary/" | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null
suji@suji:~$ sudo apt update
suji@suji:~$ sudo apt install jenkins -y
Ign:1 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:2 https://pkg.jenkins.io/debian-stable binary/ Release [2044 B]
Get:3 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Get:4 https://pkg.jenkins.io/debian-stable binary/ Packages [28.7 kB]
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:6 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:7 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:8 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Fetched 31.6 kB in 1s (26.6 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
13 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm17t64
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  net-tools
The following NEW packages will be installed:
  jenkins net-tools
0 upgraded, 2 newly installed, 0 to remove and 13 not upgraded.
Need to get 95.0 MB of archives.
After this operation, 97.6 MB of additional disk space will be used.
Get:2 http://archive.ubuntu.com/ubuntu noble/main amd64 net-tools amd64 2.10-0.1ubuntu4 [204 kB]
Get:1 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.492.2 [94.8 MB]
Fetched 95.0 MB in 17s (5603 kB/s)
```

4.1 Add Jenkins GPG Key

```
wget -q -O- https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee
/usr/share/keyrings/jenkins-keyring.asc > /dev/null
```

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

```
Unpacking jenkins (2.492.2) ...
Setting up net-tools (2.10-0.1ubuntu4) ...
Setting up jenkins (2.492.2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /usr/lib/systemd/system/jenkins.service.
Processing triggers for man-db (2.12.0-4build2) ...
suji@Suji:~$ sudo systemctl enable --now jenkins
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
suji@Suji:~$ 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 Mon 2025-03-17 16:18:13 UTC; 1min 15s ago
     Main PID: 6238 (java)
       Tasks: 53 (limit: 4588)
      Memory: 869.9M (0)
     CGroup: /system.slice/jenkins.service
             └─6238 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Mar 17 16:18:07 Suji jenkins[6238]: 535dd9073c6041839ad1010bcb510b6f
Mar 17 16:18:07 Suji jenkins[6238]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Mar 17 16:18:07 Suji jenkins[6238]: *****
Mar 17 16:18:07 Suji jenkins[6238]: *****
Mar 17 16:18:07 Suji jenkins[6238]: *****
Mar 17 16:18:13 Suji jenkins[6238]: 2025-03-17 16:18:13.001+0000 [id=39] INFO jenkins.InitReactorRunner$1#onAttained: Completed in
initialization
Mar 17 16:18:13 Suji jenkins[6238]: 2025-03-17 16:18:13.017+0000 [id=25] INFO hudson.lifecycle.Lifecycle#onReady: Jenkins is fully
up and run>
Mar 17 16:18:13 Suji systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
Mar 17 16:18:14 Suji jenkins[6238]: 2025-03-17 16:18:14.173+0000 [id=62] INFO h.m.DownloadService$Downloadable#load: Obtained the
updated dat>
Mar 17 16:18:14 Suji jenkins[6238]: 2025-03-17 16:18:14.173+0000 [id=62] INFO hudson.util.Retrier#start: Performed the action chec
k updates s>
lines 1-19/19 (END)
```

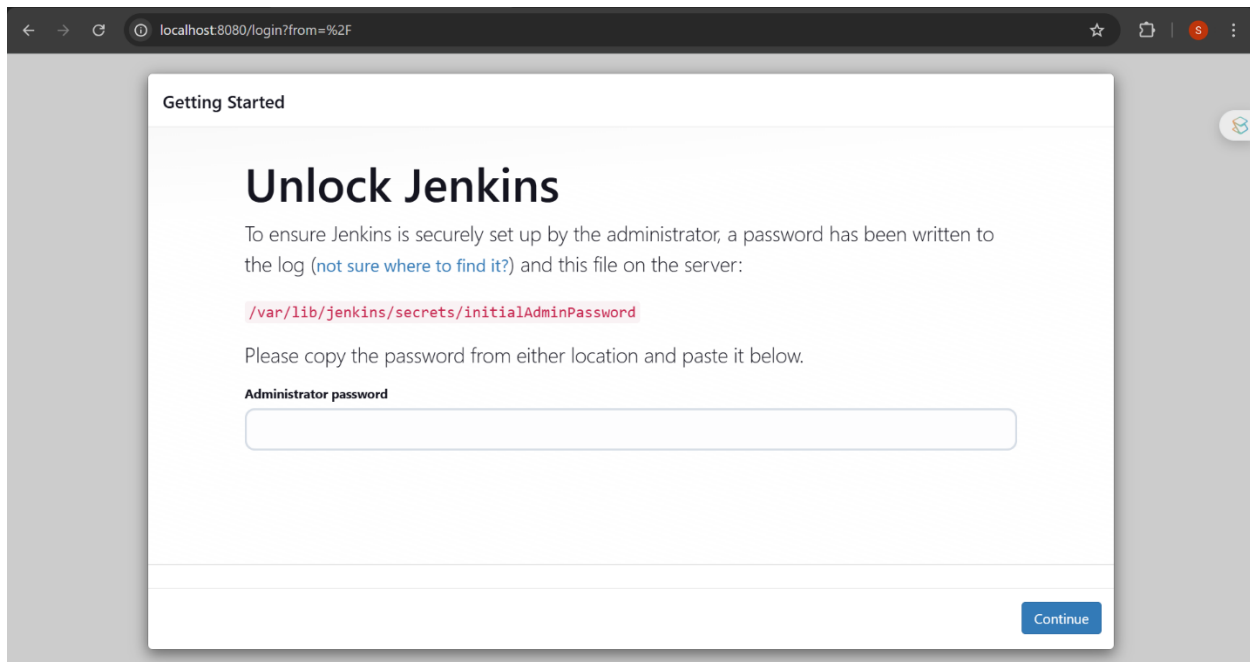
`sudo systemctl status jenkins`

- You should see **active (running)** if Jenkins is running properly.

Step 8: Retrieve Jenkins Admin Password

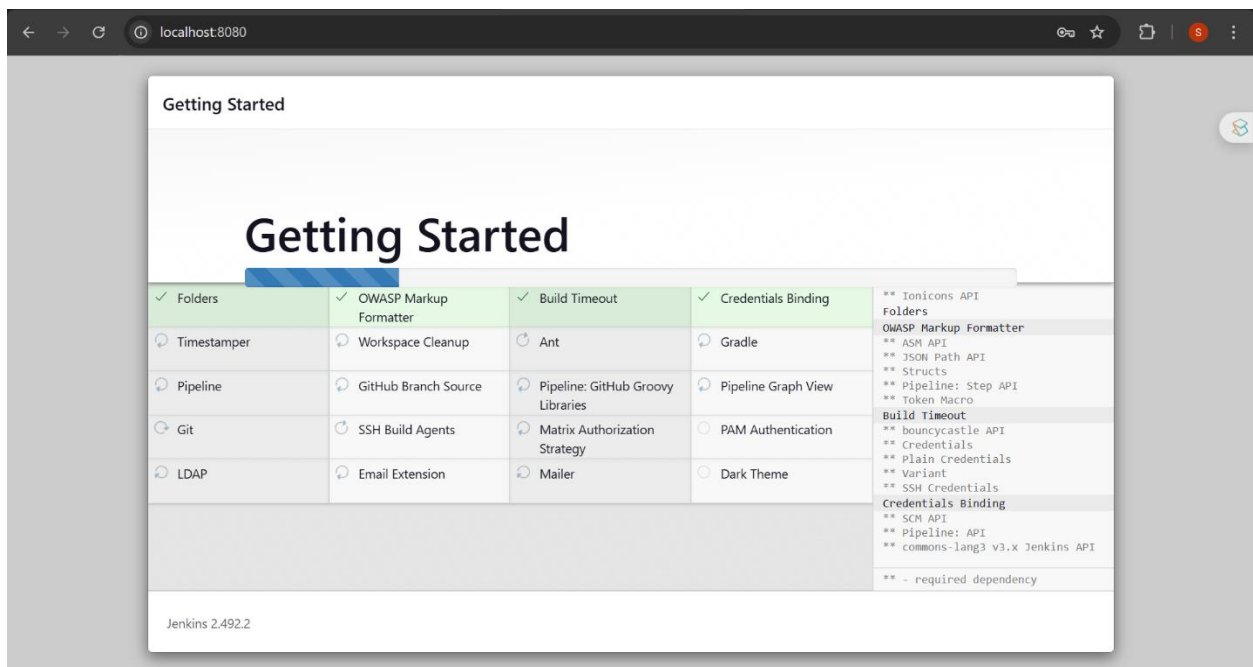
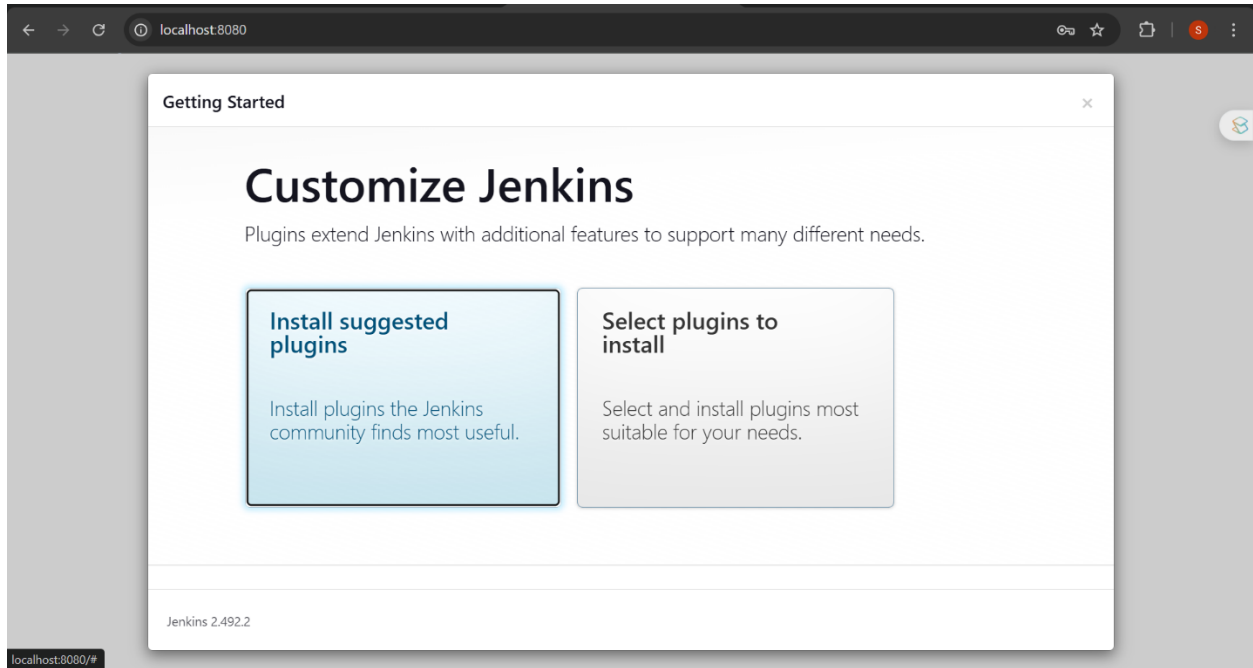
`sudo cat /var/lib/jenkins/secrets/initialAdminPassword`

- Copy this password and use it for the initial setup.



Step 9: Access Jenkins Web Interface

1. Open a browser and go to:
<http://localhost:8080> (or http://<VM_IP>:8080 if using a remote server).
2. Enter the **admin password** retrieved in the previous step.
3. Choose **Install Suggested Plugins** (recommended) or manually select plugins.



← → ↻ 📄 localhost:8080 🔑 ☆ 🗂️ \$ ⋮

Getting Started

Create First Admin User

Username

Password

Confirm password

Full name

Jenkins 2.492.2

[Skip and continue as admin](#) [Save and Continue](#)

← → ↻ 📄 localhost:8080 🔑 ☆ 🗂️ \$ ⋮

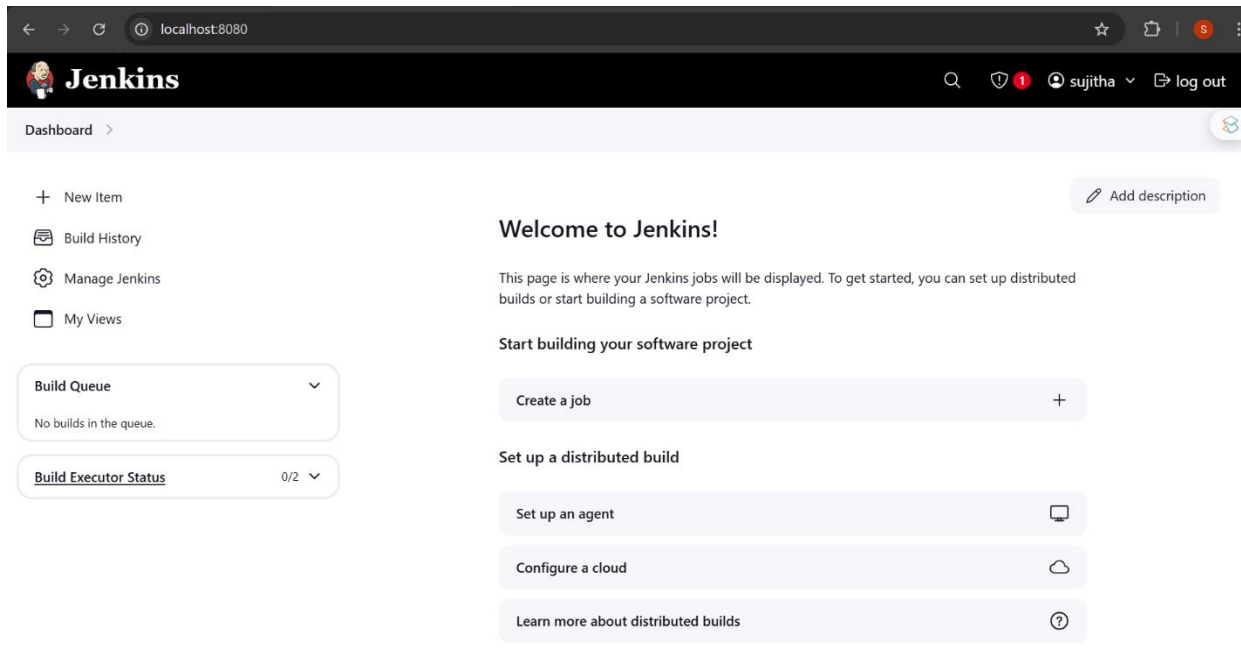
Getting Started

Jenkins is ready!

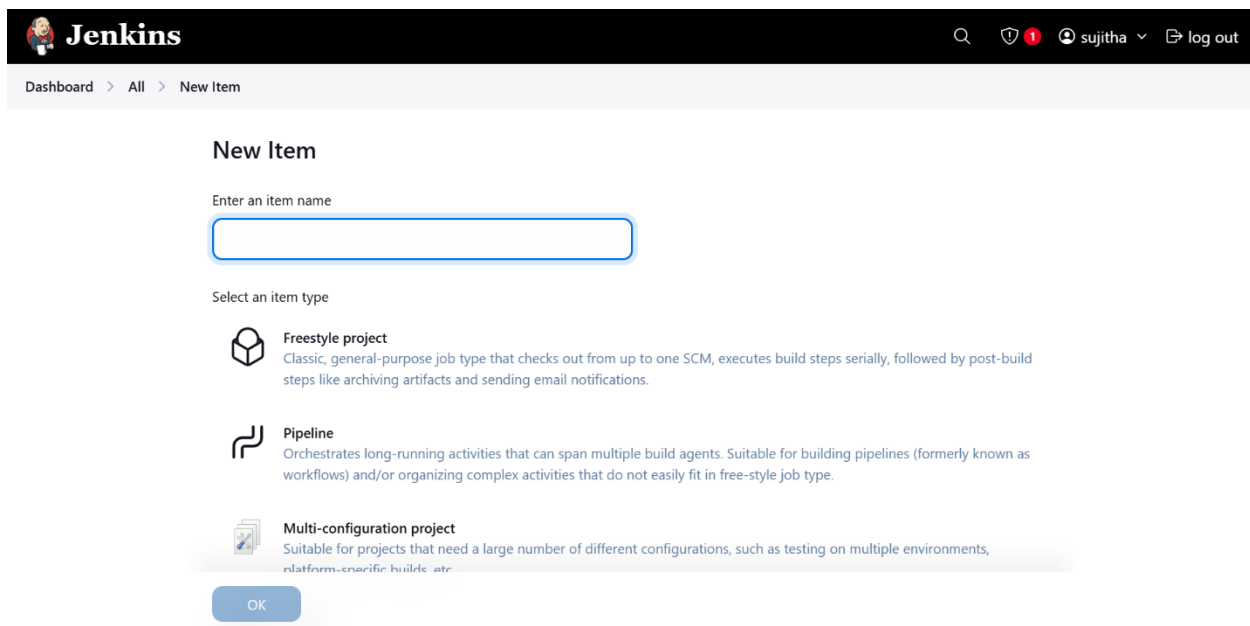
Your Jenkins setup is complete.

[Start using Jenkins](#)

Jenkins 2.492.2



Creating a Freestyle Job in Jenkins to Install Nginx



Step 1: Create a New Freestyle Job

1. Open Jenkins and click **New Item**.
2. Enter a name (e.g., "Install-Nginx").
3. Select **Freestyle Project**.
4. Click **OK**.

Step 2: Configure the Job

Dashboard > Day 1 > Configuration

Configure

Configure settings and variables that define the context in which your build runs, like credentials, paths, and global parameters.

☐ Delete workspace before build starts

☐ Use secret text(s) or file(s) ?

Filter

- Execute Windows batch command
- Execute shell
- Invoke Ant
- Invoke Gradle script
- Invoke top-level Maven targets
- Run with timeout
- Set build status to "pending" on GitHub commit

Add build step ^

Post-build Actions

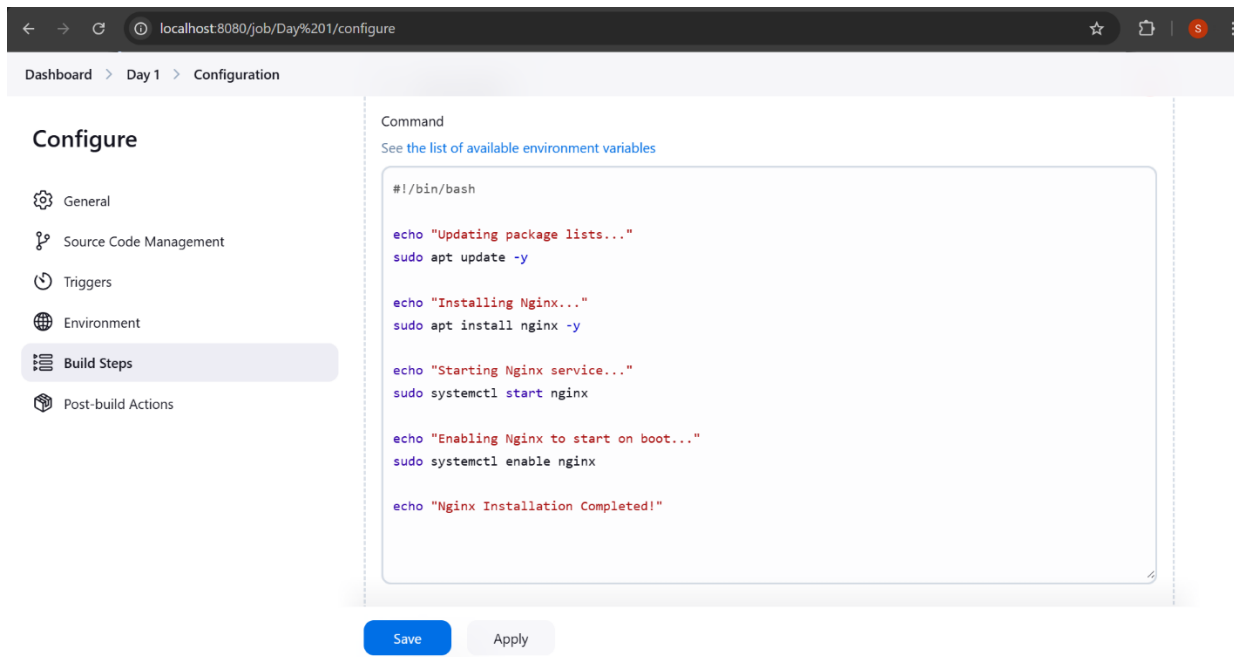
Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.

Save Apply

Add Build Step

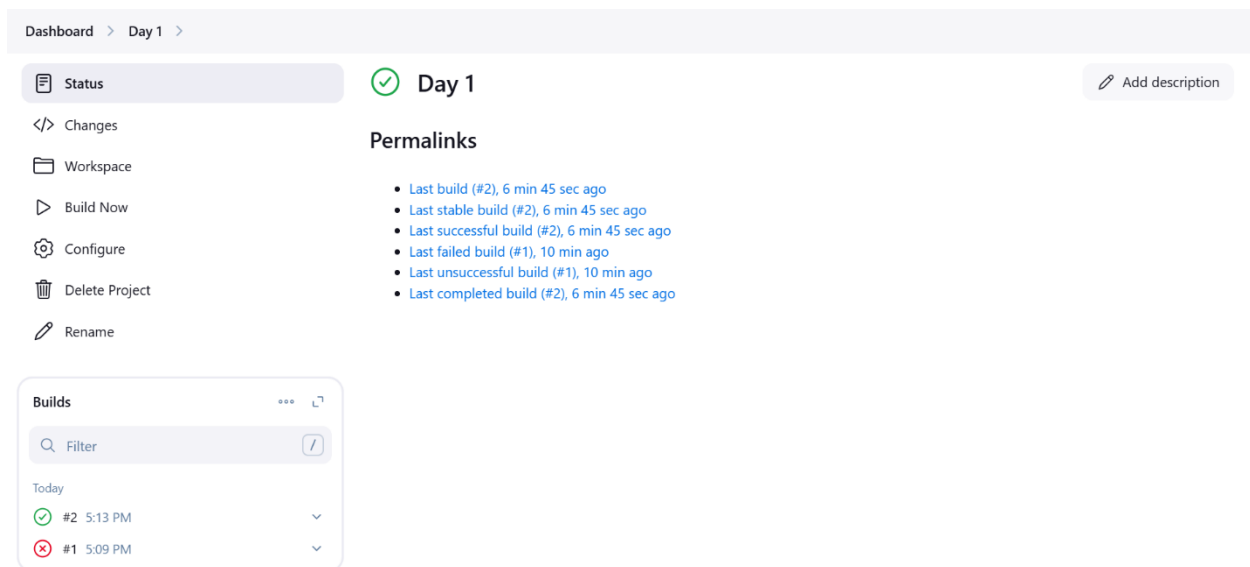
1. Scroll down to **Build** → Click **Add build step** → Select **Execute shell**.
2. Paste the following script in the command box:

```
#!/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

```
suji@Suji:~$ sudo systemctl start nginx
sudo systemctl enable nginx
Synchronizing state of nginx.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable nginx
suji@Suji:~$ sudo systemctl status nginx
● 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 Mon 2025-03-17 17:16:54 UTC; 41s ago
     Docs: man:nginx(8)
  Main PID: 7708 (nginx)
    Tasks: 9 (limit: 4588)
   Memory: 6.2M ()
   CGroup: /system.slice/nginx.service
           └─7708 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─7710 "nginx: worker process"
               └─7711 "nginx: worker process"
                 └─7712 "nginx: worker process"
                   └─7713 "nginx: worker process"
                     └─7714 "nginx: worker process"
                       └─7715 "nginx: worker process"
                         └─7716 "nginx: worker process"
                           └─7717 "nginx: worker process"

Mar 17 17:16:54 Suji systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
Mar 17 17:16:54 Suji systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
```

Check Nginx Status

systemctl status nginx

- If running, you should see **active (running)**.

Open Nginx in Browser

1. Open a browser and go to:
2. `http://<VM_IP>`
3. You should see the **default Nginx welcome page**.

