

# Jenkins

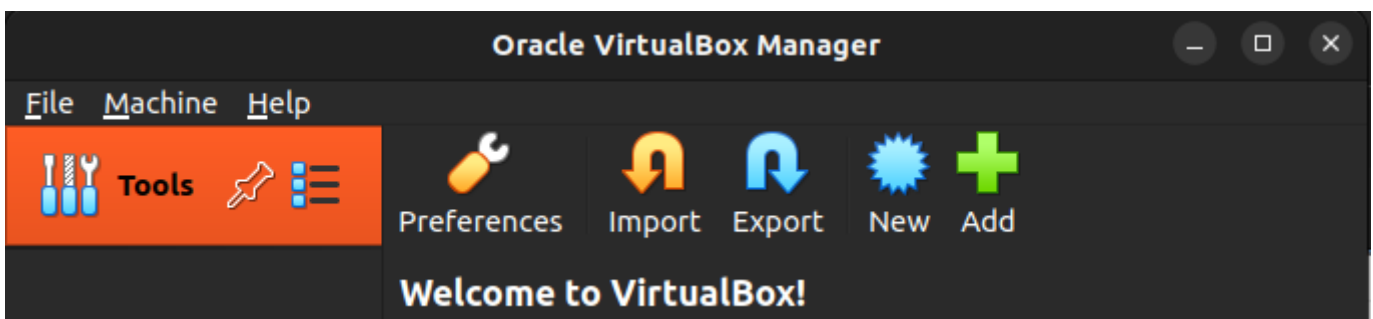
Automating software development process using Jenkins.

- [Creating a VM using VirtualBox](#)
- [Installing Prerequisites](#)
- [Installing Jenkins](#)
- [Setting up Jenkins](#)
- [Pipelines in Jenkins](#)

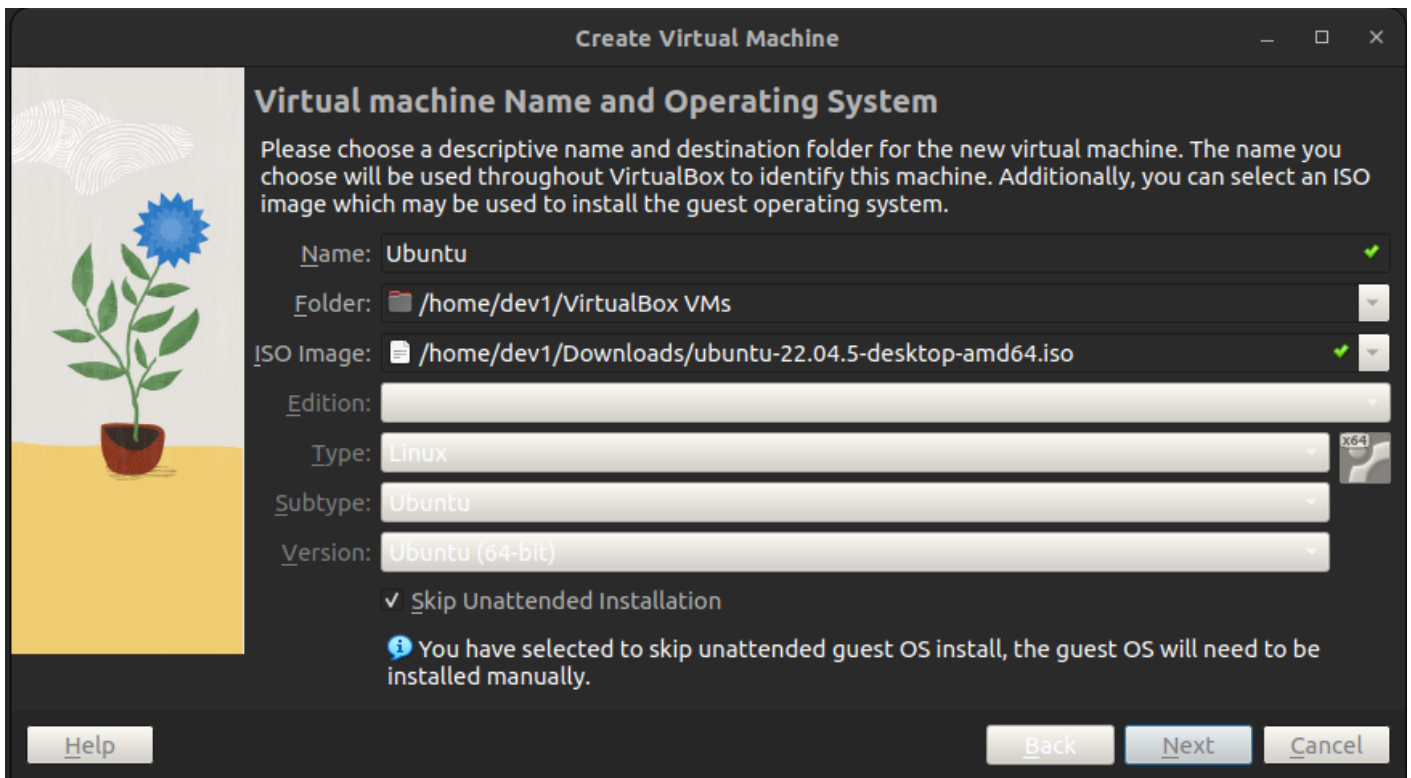
# Creating a VM using VirtualBox



- First verify that the virtualization option is enabled. You can do that by going to the BIOS of your machine. Under security menu, you can see the Virtualization option. If it is disabled, enable it and start the machine.
- Then download the Oracle Virtual Box software for your desired OS from [Oracle Virtual Box](#) and install the software.
- Next download the ISO image of the VM Operating System. For this, we used [Ubuntu 22.04.5 lts](#) (Download the Desktop image).
- After that, install the ISO using Oracle VirtualBox by clicking "New" button and selecting the downloaded ISO image.



Tick the checkbox with Skip Unattended installation. (Otherwise the terminal cannot be opened: This is an error with ubuntu 22.04 or virtualbox version.)



After clicking next you can select the Memory and no. of processors. Then install and create the virtual machine.

For further information refer this video: <https://www.youtube.com/watch?v=nvdnQX9UkMY>

## Optional: Install guest additions (If skip unattended installation)

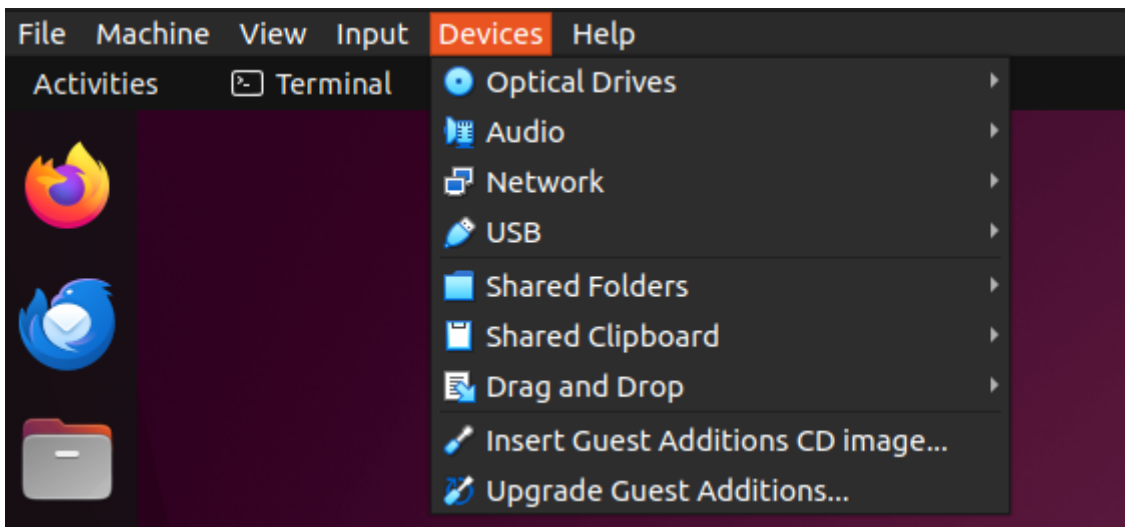
Guests Additions helps user to do tasks such as copy and paste from host-guest/ guest-host, resize the window size. It helps a lot when copying and transferring between guest and host. This gets installed automatically in the unattended installation. Since, it is skipped we have to install guest additions manually.

- To install guest additions. Log in to your VM and run the following commands.

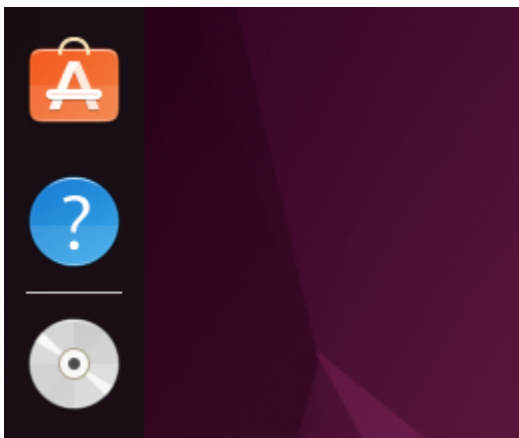
```
$ sudo apt update
```

```
$ sudo apt install -y build-essential linux-headers-$(uname -r)
```

- After running above commands, go to Devices menu and click "Insert Guest Additions CD image".



- It will mount a new ISO image like below. Click the CD icon. Then a folder opens.



- Open the terminal inside that folder and run the following command. This should run the "autorun.sh" file in that opened folder.

```
$ ./autorun.sh
```

After finish executing, restart the VM. Guests Additions will now be working!

# Installing Prerequisites

## Step 1: Install Java



Install java on the virtual machine using the following commands in the terminal.

- Search for OpenJDK.

```
$ sudo apt-cache search openjdk
```

- install OpenJDK 17 using the APT package manager

```
$ sudo apt install openjdk-17-jdk -y
```

- Once the installation is complete, verify the version of Java installed by running the command:

```
$ java -version
```

After running above command you should see something like this:

```
dev1vm@dev1vm-VirtualBox:~$ java -version
openjdk version "17.0.12" 2024-07-16
OpenJDK Runtime Environment (build 17.0.12+7-Ubuntu-1ubuntu222.04)
OpenJDK 64-Bit Server VM (build 17.0.12+7-Ubuntu-1ubuntu222.04, mixed mode, sharing)
```

# Step 1: Install Git



To install Git on your machine, run the following command.

```
$ sudo apt-get install git
```

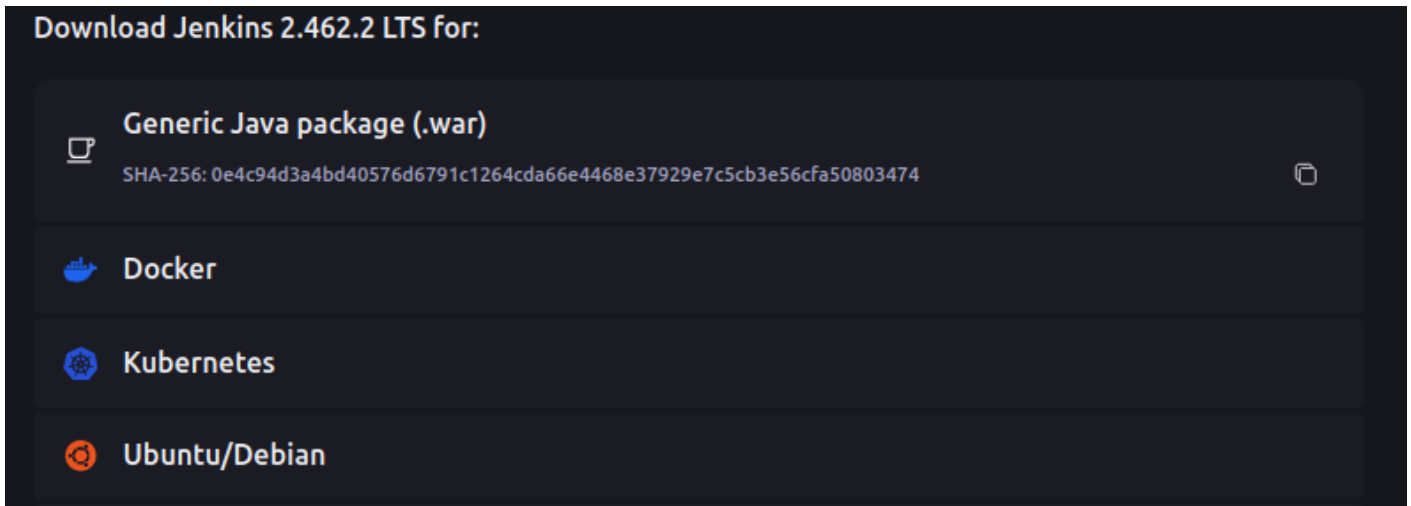
After installing check if it is installed by running the following command.

```
$ git --version
```

# Installing Jenkins

Go to Jenkins official website and download Jenkins for Ubuntu/debian.

<https://www.jenkins.io/download/>



Then run the commands that is shown in the page. It should look something like below:

## Jenkins Debian Packages

This is the Debian package repository of Jenkins to automate installation and upgrade. To use this repository, first add the key to your system (for the Weekly Release Line):

```
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
```

Then add a Jenkins apt repository entry:

```
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"
```

Update your local package index, then finally install Jenkins:

```
sudo apt-get update
sudo apt-get install fontconfig openjdk-17-jre
sudo apt-get install jenkins
```

NOTE: Don't need to run the following command since java is already installed previously.

```
$ sudo apt-get install fontconfig openjdk-17-jre
```

After successfully installing Jenkins, run the following command to start Jenkins.

```
$ sudo systemctl start jenkins
```

To see if Jenkins is running, run the following command.

```
$ sudo systemctl status jenkins
```

Then you will get something like this:

```
r2schools@ub2022:~$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2022-12-29 21:43:37 IST; 17s ago
     Main PID: 5396 (java)
        Tasks: 52 (limit: 4584)
      Memory: 1.0G
         CPU: 1min 4.126s
    CGroup: /system.slice/jenkins.service
            └─5396 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/c

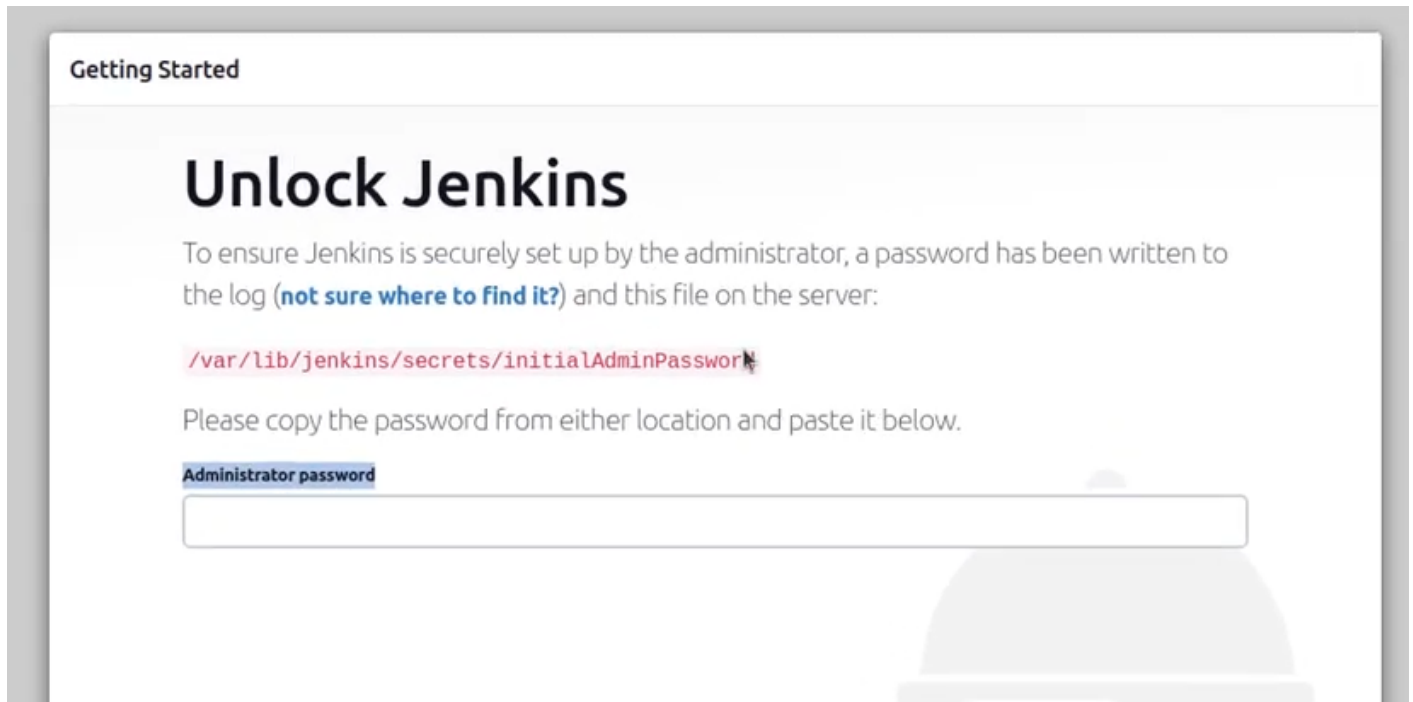
Dec 29 21:43:08 ub2022 jenkins[5396]: Jenkins initial setup is required. An admin user has been created and
Dec 29 21:43:08 ub2022 jenkins[5396]: Please use the following password to proceed to installation:
Dec 29 21:43:08 ub2022 jenkins[5396]: 91d838ee3a4541eeaf431f1f2e1bb99f
Dec 29 21:43:08 ub2022 jenkins[5396]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPasswo
Dec 29 21:43:08 ub2022 jenkins[5396]: *****
Dec 29 21:43:08 ub2022 jenkins[5396]: *****
```

IMPORTANT: Copy the password from above, you will need it when setting up Jenkins.

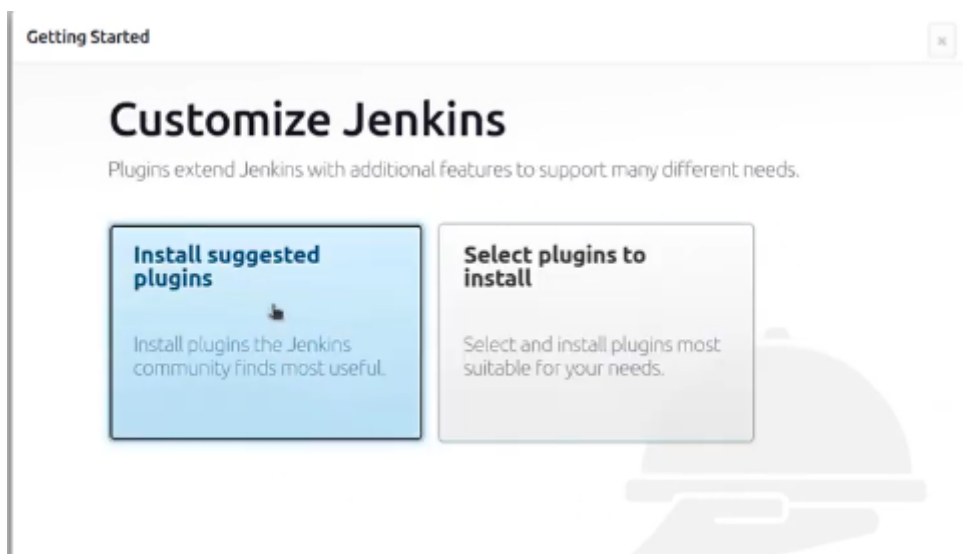


# Setting up Jenkins

- After running Jenkins and checking status, go to [localhost:8080](http://localhost:8080) (this is the default port that jenkins runs). You will see the following page:



- paste the password you copied when checking the status and click continue.
- Then proceed to install suggested plugins. It may take a few minutes.



- Then create an account. Enjoy Jenkins !

# Pipelines in Jenkins

Jenkins Pipelines are a powerful way to automate the software development lifecycle by defining CI/CD workflows as code. They allow you to automate tasks like building, testing, and deploying applications in a structured, repeatable, and scalable manner. Pipelines provide visualization of the workflow stages, support complex scenarios with branching and parallel execution, integrate easily with other tools, and enhance consistency, reliability, and speed in the delivery process, reducing manual errors and increasing efficiency.

## Creating a simple pipeline

To create a simple pipeline go to the repository directory and create a text file named "Jenkinsfile". Inside the file copy and paste the below groovy script. This is a simple groovy script which prints some statements in each stage of the build.

```
pipeline {
    agent any

    stages {
        stage('Build') {
            steps {
                echo 'Building the Application...'
            }
        }
        stage('Test') {
            steps {
                echo 'Testing the Application ...'
            }
        }
        stage('Deploy') {
            steps {
                echo 'Deploying the Application...'
            }
        }
    }
}
```

Now check status of jenkins using the following command.

```
$ sudo systemctl status jenkins
```

if it is active then go to jenkins running url (default is localhost:8080). if it is not active run the following command and start jenkins.

```
$ sudo systemctl start jenkins
```

- After logging in go to Dashboard/ new item and create a multibranch pipeline.
- Click add source and paste the git url. (use this if you don't have one:  
<https://github.com/ransika99lalco/my-app.git>)
- Build configuration should look like below:

## Build Configuration

### Mode

by Jenkinsfile

Script Path ?

Jenkinsfile

- Click save and run build. In the stages, you should see something like this.




- When you click on each stage you can see the print statements being printed as below.

## < **Build #6**


Success 22 hr ago in 13 sec

 Checkout SCM

 **Build**


 Test


 Deploy


 Check React Version


 Post Actions


### Stage 'Build'

 Started 22 hr ago

 Queued 0 ms

 Took 0.22 sec

 Success

 [View as plain text](#)



**Building the Application...**

Print Message