NAME: ARPITA NATH

ROLL: 2251188

DEPARTMENT: COMPUTER SCIENCE & ENGINEERING

ASSIGNMENT: DEVOPS LAB FILE

SEMESTER: 7TH

SECTION: A

DATE OF SUBMISSION: 16/11/2025

INDEX

Day 1: Installing Git and VsCode Configuration

Bash Commands:

Git clone : git clone https://github.com/arpi2003ta/EV-Dashboard.git

devops@cme615lab04 MINGW64 /d/Arpita/day1 (main)

$ cd EV-Dashboard/

devops@cme615lab04 MINGW64 /d/Arpita/day1/EV-Dashboard (main)

$ git init

Reinitialized existing Git repository in D:/Arpita/day1/EV-Dashboard/.git/

devops@cme615lab04 MINGW64 /d/Arpita/day1/EV-Dashboard (main)

$ git config --global user.email natharpita2004@gmail.com

devops@cme615lab04 MINGW64 /d/Arpita/day1/EV-Dashboard (main)

$ git config --global user.name “arpita”

devops@cme615lab04 MINGW64 /d/Arpita/day1/EV-Dashboard (main)

$ git push

Everything up-to-date

devops@cme615lab04 MINGW64 /d/Arpita/day1/EV-Dashboard (main)

$ git status

On branch main

Your branch is up to date with 'origin/main'.

Changes not staged for commit:

(use “git add <file>...”; to update what will be committed)

(use “git restore <file>...”; to discard changes in working directory)

modified: README.md

no changes added to commit (use “git add” and/or “git commit -a”)

devops@cme615lab04 MINGW64 /d/Arpita/day1/EV-Dashboard (main)

$ git add .

devops@cme615lab04 MINGW64 /d/Arpita/day1/EV-Dashboard (main)

$ git commit -am “new commit”

[main 094261d] new commit

1 file changed, 2 insertions(+), 1 deletion(-)

devops@cme615lab04 MINGW64 /d/Arpita/day1/EV-Dashboard (main)

$ git push

Enumerating objects: 5, done.

Counting objects: 100% (5/5), done.

Delta compression using up to 8 threads

Compressing objects: 100% (3/3), done.

Writing objects: 100% (3/3), 321 bytes | 321.00 KiB/s, done.

Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)

remote: Resolving deltas: 100% (1/1), completed with 1 local object.

To https://github.com/ arpi2003ta /EV-Dashboard.git

9abde26..094261d main -> main

These are the commands used to setup Git with Visual Studio Code in our local machine along with the output beneath.

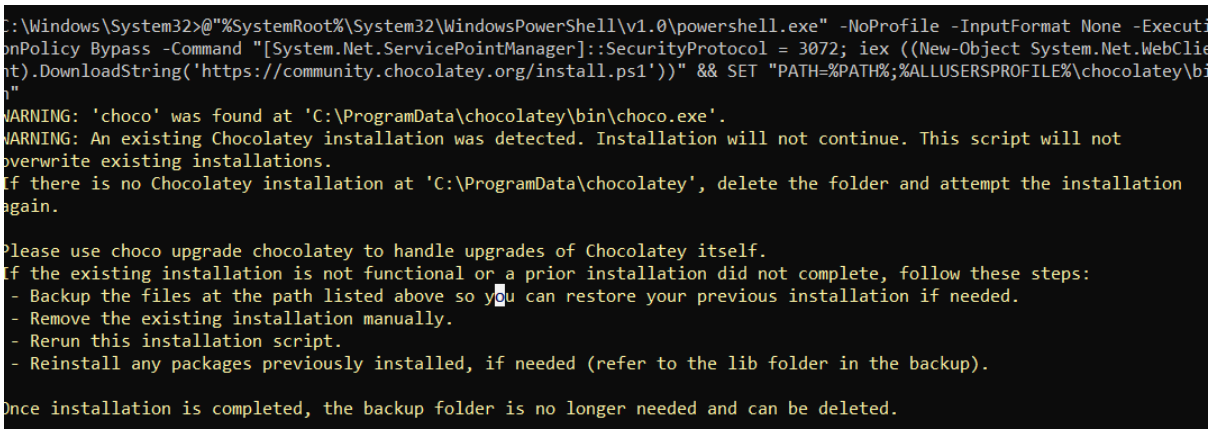
Day 2:

Installing chocolatey and other requirements

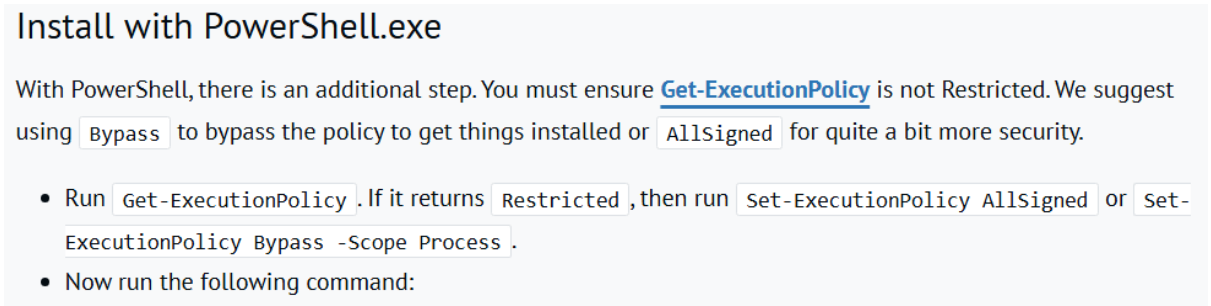
Link: https://docs.chocolatey.org/en-us/choco/setup/

To install chocolatey with cmd.exe run the command

@"%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile -InputFormat None -ExecutionPolicy Bypass -Command "[System.Net.ServicePointManager]::SecurityProtocol = 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))" && SET "PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin"



Via Powershell (Required):



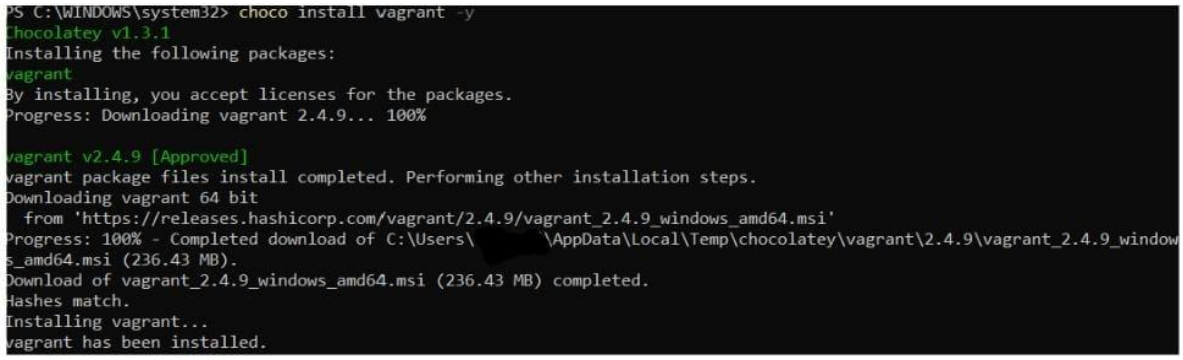
After this we need to execute:
Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))

Next we need to install all the required tools:

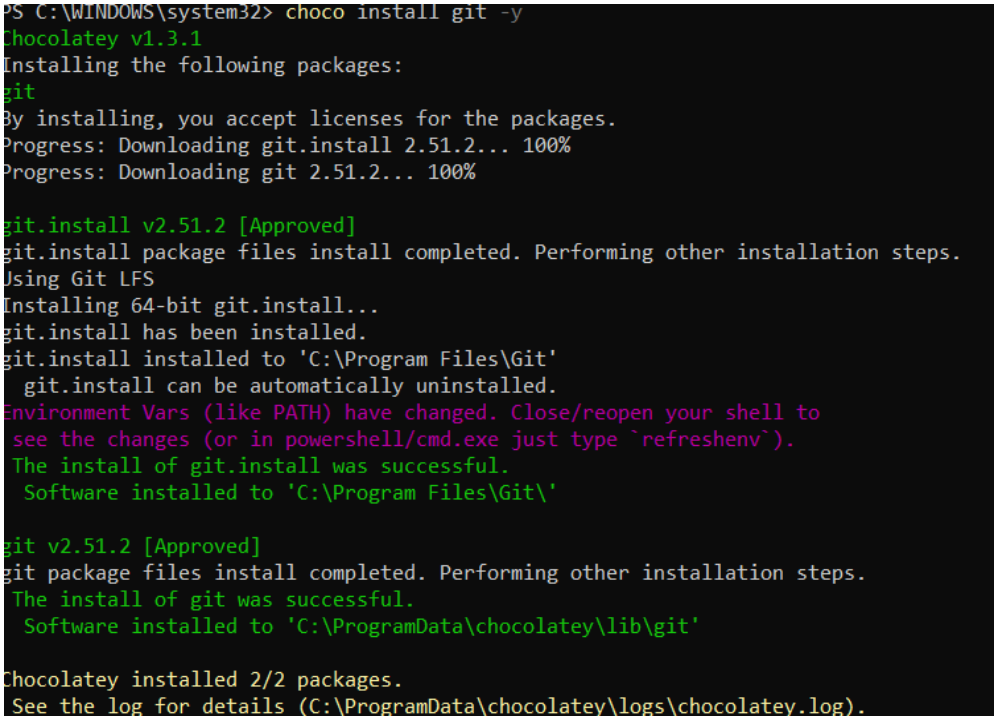
Vagrant
Git
correto11jdk
maven
awscli
intellijidea community
vscode
sublimetext3.app

For this we need to go to:
1. Open Powershell as an administrator.
2. https://community.chocolatey.org/packages to get the required packages.

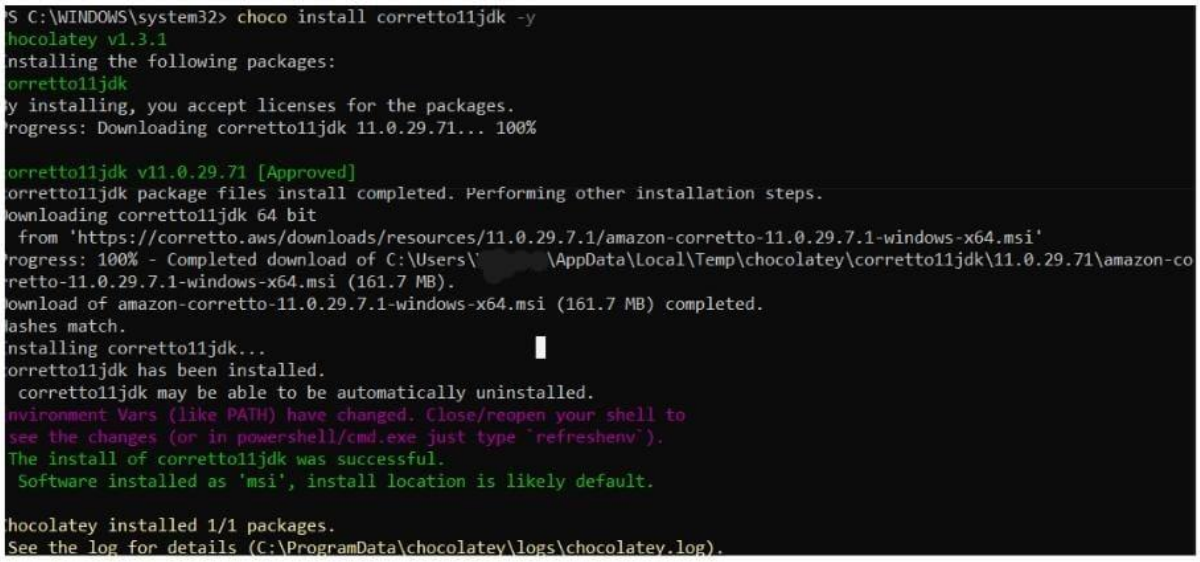
Here we can search for the required packages and install them in our machine.



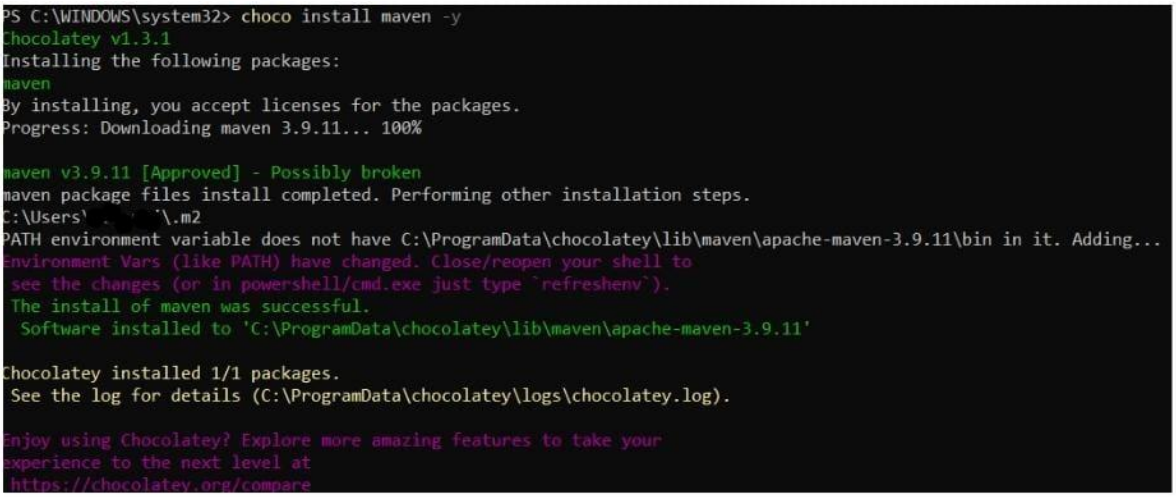
Git:



Correto11jdk



Maven



AWSCLI

```
S C:\WINDOWS\system32> choco install awscli -y
hocolatey v1.3.1
nstalling the following packages:
wscli
y installing, you accept licenses for the packages.
rogress: Downloading awscli 2.31.34... 100%

wscli v2.31.34 [Approved]
wscli package files install completed. Performing other installation steps.
ownloading awscli 64 bit
 from 'https://awscli.amazonaws.com/AWSCLIV2-2.31.34.msi'
rogress: 100% - Completed download of C:\Users\      \AppData\Local\Temp\chocolatey\awscli\2.31.34\AWSCLIV2-2.31.34.ms
 (39.65 MB).
ownload of AWSCLIV2-2.31.34.msi (39.65 MB) completed.
ashes match.
nstalling awscli...
wscli has been installed.
 awscli may be able to be automatically uninstalled.
nvironment Vars (like PATH) have changed. Close/reopen your shell to
ee the changes (or in powershell/cmd.exe just type `refreshenv`).
he install of awscli was successful.
 Software installed as 'MSI', install location is likely default.

hocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
```

Intellijidea

```
PS C:\WINDOWS\system32> choco install intellijidea-community -y
Chocolatey v1.3.1
Installing the following packages:
intellijidea-community
By installing, you accept licenses for the packages.
Progress: Downloading intellijidea-community 2025.2.4... 100%

intellijidea-community v2025.2.4 [Approved]
intellijidea-community package files install completed. Performing other installation steps.
WARNING: No registry key found based on  'IntelliJ IDEA Community Edition*'
C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.2.4
Downloading intellijidea-community 64 bit
  from 'https://download.jetbrains.com/idea/ideaIC-2025.2.4.exe'
Progress: 100% - Completed download of C:\Users\      \AppData\Local\Temp\chocolatey\intellijidea-community\2025.2.4\id
eaIC-2025.2.4.exe (952.62 MB).
Download of ideaIC-2025.2.4.exe (952.62 MB) completed.
Hashes match.
Installing intellijidea-community...
intellijidea-community has been installed.
  intellijidea-community may be able to be automatically uninstalled.
 The install of intellijidea-community was successful.
  Software installed to 'C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.2.4'

Chocolatey installed 1/1 packages.
 See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
```

Sublimetext3

```
PS C:\WINDOWS\system32> choco install sublimetext3 -y
Chocolatey v1.3.1
Installing the following packages:
sublimetext3
By installing, you accept licenses for the packages.
Progress: Downloading SublimeText3 3.2.2... 100%

SublimeText3 v3.2.2 [Approved]
sublimetext3 package files install completed. Performing other installation steps.
Downloading SublimeText3 64 bit
  from 'https://download.sublimetext.com/Sublime%20Text%20Build%203211%20x64%20Setup.exe'
Progress: 100% - Completed download of C:\Users\      \AppData\Local\Temp\chocolatey\SublimeText3\3.2.2\Sublime Text Bu
ild 3211 x64 Setup.exe (10.42 MB).
Download of Sublime Text Build 3211 x64 Setup.exe (10.42 MB) completed.
Hashes match.
Installing SublimeText3...
SublimeText3 has been installed.
  sublimetext3 can be automatically uninstalled.
 The install of sublimetext3 was successful.
  Software installed to 'C:\Program Files\Sublime Text 3\'

Chocolatey installed 1/1 packages.
 See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
```

DAY 3 (Installing via vagrant)

```
devops@cme615lab04 MINGW64 ~
$ cd "D:\Arpita"

devops@cme615lab04 MINGW64 /d/Arpita
$ ls
day1/

devops@cme615lab04 MINGW64 /d/Arpita
$ mkdir centos

devops@cme615lab04 MINGW64 /d/Arpita
$ cd centos/

devops@cme615lab04 MINGW64 /d/Arpita/centos
$ vagrant init generic/centos7 --box-version 4.3.12
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
`vagrantup.com` for more information on using Vagrant.

devops@cme615lab04 MINGW64 /d/Arpita/centos
$ ls
Vagrantfile

devops@cme615lab04 MINGW64 /d/Arpita/centos
$ cat Vagrantfile
# -*- mode: ruby -*-
# vi: set ft=ruby :

# All Vagrant configuration is done below. The "2" in Vagrant.configure
# configures the configuration version (we support older styles for
# backwards compatibility). Please don't change it unless you know what
# you're doing.
Vagrant.configure("2") do |config|
  # The most common configuration options are documented and commented below.
  # For a complete reference, please see the online documentation at
  # https://docs.vagrantup.com.

  # Every Vagrant development environment requires a box. You can search for
  # boxes at https://vagrantcloud.com/search.
  config.vm.box = "generic/centos7"
  config.vm.box_version = "4.3.12"

  # Disable automatic box update checking. If you disable this, then
  # boxes will only be checked for updates when the user runs
  # `vagrant box outdated`. This is not recommended.
  # config.vm.box_check_update = false

  # Create a forwarded port mapping which allows access to a specific port
  # within the machine from a port on the host machine. In the example below,
  # accessing "localhost:8080" will access port 80 on the guest machine.
  # NOTE: This will enable public access to the opened port
  # config.vm.network "forwarded_port", guest: 80, host: 8080

  # Create a forwarded port mapping which allows access to a specific port
  # within the machine from a port on the host machine and only allow access
  # via 127.0.0.1 to disable public access
  # config.vm.network "forwarded_port", guest: 80, host: 8080, host_ip: "127.0.0.1"

  # Create a private network, which allows host-only access to the machine
  # using a specific IP.
  # config.vm.network "private_network", ip: "192.168.33.10"

  # Create a public network, which generally matched to bridged network.
  # Bridged networks make the machine appear as another physical device on
```

```
  # your network.
  # config.vm.network "public_network"

  # Share an additional folder to the guest VM. The first argument is
  # the path on the host to the actual folder. The second argument is
  # the path on the guest to mount the folder. And the optional third
  # argument is a set of non-required options.
  # config.vm.synced_folder "../data", "/vagrant_data"

  # Disable the default share of the current code directory. Doing this
  # provides improved isolation between the vagrant box and your host
  # by making sure your Vagrantfile isn't accessible to the vagrant box.
  # If you use this you may want to enable additional shared subfolders as
  # shown above.
  # config.vm.synced_folder ".", "/vagrant", disabled: true

  # Provider-specific configuration so you can fine-tune various
  # backing providers for Vagrant. These expose provider-specific options.
  # Example for VirtualBox:
  #
  # config.vm.provider "virtualbox" do |vb|
  #   # Display the VirtualBox GUI when booting the machine
  #   vb.gui = true
  #
  #   # Customize the amount of memory on the VM:
  #   vb.memory = "1024"
  # end
  #
  # View the documentation for the provider you are using for more
  # information on available options.

  # Enable provisioning with a shell script. Additional provisioners such as
  # Ansible, Chef, Docker, Puppet and Salt are also available. Please see the
  # documentation for more information about their specific syntax and use.
  # config.vm.provision "shell", inline: <<-SHELL
  #   apt-get update
  #   apt-get install -y apache2
  # SHELL
end

devops@cme615lab04 MINGW64 /d/Arpita/centos
$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Box 'generic/centos7' could not be found. Attempting to find and
install...
    default: Box Provider: virtualbox
    default: Box Version: 4.3.12
==> default: Loading metadata for box 'generic/centos7'
    default: URL: https://vagrantcloud.com/api/v2/vagrant/generic/centos7
==> default: Adding box 'generic/centos7' (v4.3.12) for provider: virtualbox
(amd64)
    default: Downloading:
https://vagrantcloud.com/generic/boxes/centos7/versions/4.3.12/providers/virtualbox
/amd64/vagrant.box
    default:
An error occurred while downloading the remote file. The error
message, if any, is reproduced below. Please fix this error and try
again.

Recv failure: Connection was reset

devops@cme615lab04 MINGW64 /d/Arpita/centos
$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Box 'generic/centos7' could not be found. Attempting to find and
install...
    default: Box Provider: virtualbox
    default: Box Version: 4.3.12
==> default: Loading metadata for box 'generic/centos7'
    default: URL: https://vagrantcloud.com/api/v2/vagrant/generic/centos7
==> default: Adding box 'generic/centos7' (v4.3.12) for provider: virtualbox
(amd64)
    default: Downloading:
https://vagrantcloud.com/generic/boxes/centos7/versions/4.3.12/providers/virtualbox
/amd64/vagrant.box
==> default: Box download is resuming from prior download progress
    default:
```

```
    default: Calculating and comparing box checksum...
==> default: Successfully added box 'generic/centos7' (v4.3.12) for 'virtualbox
(amd64)'!
==> default: Importing base box 'generic/centos7'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'generic/centos7' version '4.3.12' is up to date...
==> default: Setting the name of the VM: centos_default_1753957188636_24730
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
==> default: Forwarding ports...
    default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Running 'pre-boot' VM customizations...
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
    default: SSH address: 127.0.0.1:2222
    default: SSH username: vagrant
    default: SSH auth method: private key
    default:
    default: Vagrant insecure key detected. Vagrant will automatically replace
    default: this with a newly generated keypair for better security.
    default:
    default: Inserting generated public key within guest...
    default: Removing insecure key from the guest if it's present...
    default: Key inserted! Disconnecting and reconnecting using new SSH key...
==> default: Machine booted and ready!
==> default: Checking for guest additions in VM...
    default: The guest additions on this VM do not match the installed version of
    default: VirtualBox! In most cases this is fine, but in rare cases it can
    default: prevent things such as shared folders from working properly. If you
see
    default: shared folder errors, please make sure the guest additions within the
    default: virtual machine match the version of VirtualBox you have installed on
    default: your host and reload your VM.
    default:
    default: Guest Additions Version: 5.2.44
    default: VirtualBox Version: 7.1

devops@cme615lab04 MINGW64 /d/Arpita/centos
$ vagrant status
Current machine states:

default                   running (virtualbox)

The VM is running. To stop this VM, you can run `vagrant halt` to
shut it down forcefully, or you can run `vagrant suspend` to simply
suspend the virtual machine. In either case, to restart it again,
simply run `vagrant up`.

devops@cme615lab04 MINGW64 /d/Arpita/centos
$ vagrant ssh
[vagrant@centos7 ~]$ pwd
/home/vagrant
[vagrant@centos7 ~]$ exit
logout

devops@cme615lab04 MINGW64 /d/Arpita/centos
$ vagrant status
Current machine states:

default                   running (virtualbox)

The VM is running. To stop this VM, you can run `vagrant halt` to
shut it down forcefully, or you can run `vagrant suspend` to simply
suspend the virtual machine. In either case, to restart it again,
simply run `vagrant up`.

devops@cme615lab04 MINGW64 /d/Arpita/centos
$ vagrant halt
==> default: Attempting graceful shutdown of VM...

devops@cme615lab04 MINGW64 /d/Arpita/centos
$ vagrant status
Current machine states:

default                   poweroff (virtualbox)
```

```
The VM is powered off. To restart the VM, simply run `vagrant up`

devops@cme615lab04 MINGW64 /d/Arpita/centos
$ ls -a
./  ../  .vagrant/  Vagrantfile

devops@cme615lab04 MINGW64 /d/Arpita/centos
$ vagrant destroy
    default: Are you sure you want to destroy the 'default' VM? [y/N] y
==> default: Destroying VM and associated drives...

devops@cme615lab04 MINGW64 /d/Arpita/centos
$ ls -a
./  ../  .vagrant/  Vagrantfile

devops@cme615lab04 MINGW64 /d/Arpita/centos
$ vagrant status
Current machine states:

default                   not created (virtualbox)

The environment has not yet been created. Run `vagrant up` to
create the environment. If a machine is not created, only the
default provider will be shown. So if a provider is not listed,
then the machine is not created for that environment.

devops@cme615lab04 MINGW64 /d/Arpita/ubuntu
$ vagrant init ubuntu/jammy64 --box-version 20241002.0.0
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
`vagrantup.com` for more information on using Vagrant.

devops@cme615lab04 MINGW64 /d/Arpita/ubuntu
$ cat Vagrantfile
# -*- mode: ruby -*-
# vi: set ft=ruby :

# All Vagrant configuration is done below. The "2" in Vagrant.configure
# configures the configuration version (we support older styles for
# backwards compatibility). Please don't change it unless you know what
# you're doing.
Vagrant.configure("2") do |config|
  # The most common configuration options are documented and commented below.
  # For a complete reference, please see the online documentation at
  # https://docs.vagrantup.com.

  # Every Vagrant development environment requires a box. You can search for
  # boxes at https://vagrantcloud.com/search.
  config.vm.box = "ubuntu/jammy64"
  config.vm.box_version = "20241002.0.0"

  # Disable automatic box update checking. If you disable this, then
  # boxes will only be checked for updates when the user runs
  # `vagrant box outdated`. This is not recommended.
  # config.vm.box_check_update = false

  # Create a forwarded port mapping which allows access to a specific port
  # within the machine from a port on the host machine. In the example below,
  # accessing "localhost:8080" will access port 80 on the guest machine.
  # NOTE: This will enable public access to the opened port
  # config.vm.network "forwarded_port", guest: 80, host: 8080

  # Create a forwarded port mapping which allows access to a specific port
  # within the machine from a port on the host machine and only allow access
  # via 127.0.0.1 to disable public access
  # config.vm.network "forwarded_port", guest: 80, host: 8080, host_ip: "127.0.0.1"

  # Create a private network, which allows host-only access to the machine
  # using a specific IP.
  # config.vm.network "private_network", ip: "192.168.33.10"

  # Create a public network, which generally matched to bridged network.
  # Bridged networks make the machine appear as another physical device on
  # your network.
  # config.vm.network "public_network"
```

```
  # Share an additional folder to the guest VM. The first argument is
  # the path on the host to the actual folder. The second argument is
  # the path on the guest to mount the folder. And the optional third
  # argument is a set of non-required options.
  # config.vm.synced_folder "../data", "/vagrant_data"

  # Disable the default share of the current code directory. Doing this
  # provides improved isolation between the vagrant box and your host
  # by making sure your Vagrantfile isn't accessible to the vagrant box.
  # If you use this you may want to enable additional shared subfolders as
  # shown above.
  # config.vm.synced_folder ".", "/vagrant", disabled: true

  # Provider-specific configuration so you can fine-tune various
  # backing providers for Vagrant. These expose provider-specific options.
  # Example for VirtualBox:
  #
  # config.vm.provider "virtualbox" do |vb|
  #   # Display the VirtualBox GUI when booting the machine
  #   vb.gui = true
  #
  #   # Customize the amount of memory on the VM:
  #   vb.memory = "1024"
  # end
  #
  # View the documentation for the provider you are using for more
  # information on available options.

  # Enable provisioning with a shell script. Additional provisioners such as
  # Ansible, Chef, Docker, Puppet and Salt are also available. Please see the
  # documentation for more information about their specific syntax and use.
  # config.vm.provision "shell", inline: <<-SHELL
  #   apt-get update
  #   apt-get install -y apache2
  # SHELL
end

devops@cme615lab04 MINGW64 /d/Arpita/ubuntu
$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Box 'ubuntu/jammy64' could not be found. Attempting to find and
install...
    default: Box Provider: virtualbox
    default: Box Version: 20241002.0.0
==> default: Loading metadata for box 'ubuntu/jammy64'
    default: URL: https://vagrantcloud.com/api/v2/vagrant/ubuntu/jammy64
==> default: Adding box 'ubuntu/jammy64' (v20241002.0.0) for provider: virtualbox
    default: Downloading:
https://vagrantcloud.com/ubuntu/boxes/jammy64/versions/20241002.0.0/providers/virtu
albox/unknown/vagrant.box
    default:
==> default: Successfully added box 'ubuntu/jammy64' (v20241002.0.0) for
'virtualbox'!
==> default: Importing base box 'ubuntu/jammy64'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'ubuntu/jammy64' version '20241002.0.0' is up to
date...
==> default: Setting the name of the VM: ubuntu_default_1753958298574_19591
Vagrant is currently configured to create VirtualBox synced folders with
the `SharedFoldersEnableSymlinksCreate` option enabled. If the Vagrant
guest is not trusted, you may want to disable this option. For more
information on this option, please refer to the VirtualBox manual:

  https://www.virtualbox.org/manual/ch04.html#sharedfolders

This option can be disabled globally with an environment variable:

  VAGRANT_DISABLE_VBOXSYMLINKCREATE=1

or on a per folder basis within the Vagrantfile:

  config.vm.synced_folder '/host/path', '/guest/path',
SharedFoldersEnableSymlinksCreate: false
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
==> default: Forwarding ports...
```

```
    default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Running 'pre-boot' VM customizations...
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
    default: SSH address: 127.0.0.1:2222
    default: SSH username: vagrant
    default: SSH auth method: private key
    default:
    default: Vagrant insecure key detected. Vagrant will automatically replace
    default: this with a newly generated keypair for better security.
    default:
    default: Inserting generated public key within guest...
    default: Removing insecure key from the guest if it's present...
    default: Key inserted! Disconnecting and reconnecting using new SSH key...
==> default: Machine booted and ready!
==> default: Checking for guest additions in VM...
    default: The guest additions on this VM do not match the installed version of
    default: VirtualBox! In most cases this is fine, but in rare cases it can
    default: prevent things such as shared folders from working properly. If you
see
    default: shared folder errors, please make sure the guest additions within the
    default: virtual machine match the version of VirtualBox you have installed on
    default: your host and reload your VM.
    default:
    default: Guest Additions Version: 6.0.0 r127566
    default: VirtualBox Version: 7.1
==> default: Mounting shared folders...
    default: D:/Arpita/ubuntu => /vagrant

devops@cme615lab04 MINGW64 /d/Arpita/ubuntu
$ vagrant status
Current machine states:

default                   running (virtualbox)

The VM is running. To stop this VM, you can run `vagrant halt` to
shut it down forcefully, or you can run `vagrant suspend` to simply
suspend the virtual machine. In either case, to restart it again,
simply run `vagrant up`.

devops@cme615lab04 MINGW64 /d/Arpita/ubuntu
$ vagrant ssh
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 5.15.0-144-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Thu Jul 31 10:38:45 UTC 2025

  System load:             0.82
  Usage of /:              3.8% of 38.70GB
  Memory usage:            21%
  Swap usage:              0%
  Processes:               122
  Users logged in:         0
  IPv4 address for enp0s3: 10.0.2.15
  IPv6 address for enp0s3: fd17:625c:f037:2:29:1eff:fe73:1fd5


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '24.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


vagrant@ubuntu-jammy:~$ pwd
/home/vagrant
vagrant@ubuntu-jammy:~$ exit
logout

devops@cme615lab04 MINGW64 /d/Arpita/ubuntu
```

```
$ vagrant halt
==> default: Attempting graceful shutdown of VM...

devops@cme615lab04 MINGW64 /d/Arpita/ubuntu
$ vagrant status
Current machine states:

default                   poweroff (virtualbox)

The VM is powered off. To restart the VM, simply run `vagrant up`
```

Now doing it manually

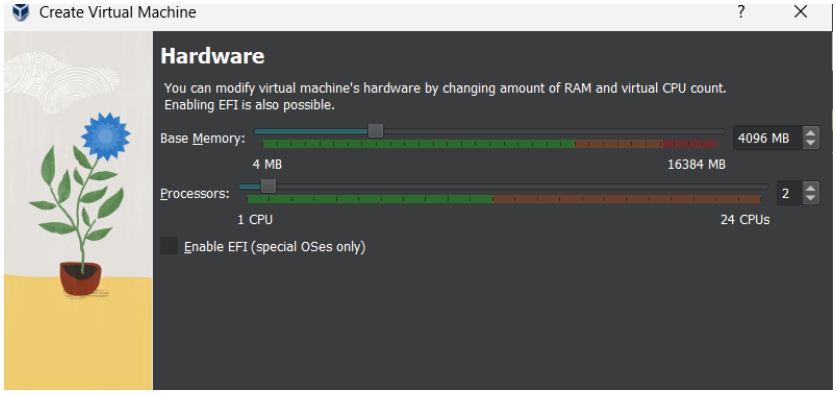
Open Oracle VM Virtual Box :

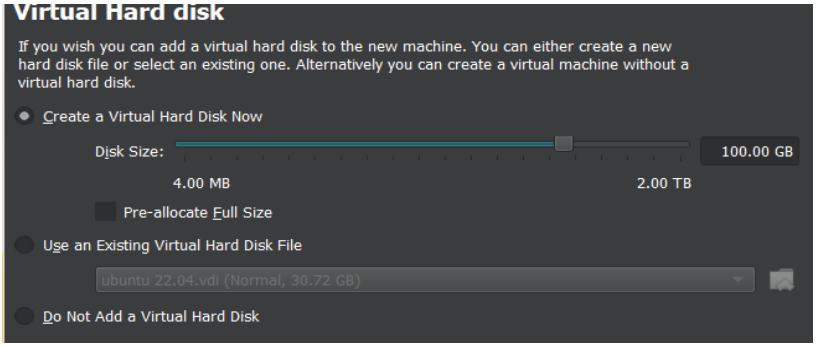
Go To Machine

Then Click on New

After that give the iso image downloaded from: https://ubuntu.com/download/server

Use these configurations:





After that select default language and default keyboard

Going by this we can do it manually in VM Virtual Box.

DAY 4:

Jenkins: Introduction and Freestyle Projects

Installation:

Step 1: Start your Ubuntu Server.
Step 2: From Git Bash SSH into your Ubuntu server
Step 3: run "sudo apt update" OR first write "su -", give root password, and then execute "apt update".
Step 4: run "apt upgrade"
Step 5: Steps to Install Jenkins:

- Open URL: https://www.jenkins.io/doc/book/installing/
- Click on Linux in left panel
- Click on Ubuntu. Follow the steps given under Ubuntu in the following order.
- First execute the steps under the section "Installation of Java"
- Then execute the steps given under "Long Term Support release". Do not write the commands. Copy and paste to avoid mistakes.
- Now go to the section "Start Jenkins" and execute the commands.
- Next go to the section "Unlocking Jenkins" and see if the application is running in the URL given.
- To unlock Jenkins you require a password. To get the password read the first point given under the section "Note" and execute the same.
- You will then be taken to a customize jenkins page. Click on install suggested plugins. If you see that many plugins have not been installed refresh the page. Continue refreshing till most/all plugins are installed.
- You will be taken to a setup login ID and password page
- You will be taken to an instance configuration page. Keep the default settings.

ABOUT JENKINS:

What is Jenkins?

At its core, **Jenkins is an open-source automation server**. It is a very popular tool used primarily to automate the repetitive tasks involved in building, testing, and deploying software.

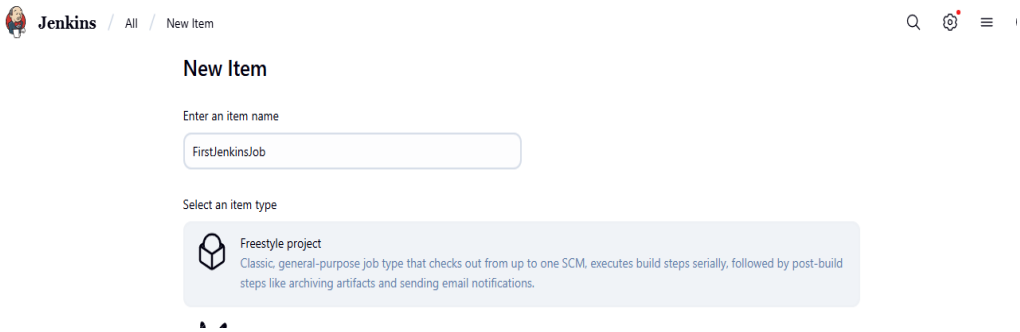
Its main purpose is to enable **Continuous Integration (CI)** and **Continuous Delivery (CD)**, which are foundational practices in modern software development (DevOps).

- **Continuous Integration (CI):** The practice of developers frequently merging their code changes into a central repository. Jenkins automates the next step: it automatically builds the application and runs a suite of automated tests to ensure the new code didn't break anything.
- **Continuous Delivery (CD):** An extension of CI. After the build and tests are successful, Jenkins can automatically deploy the application to a testing, staging, or even production environment.

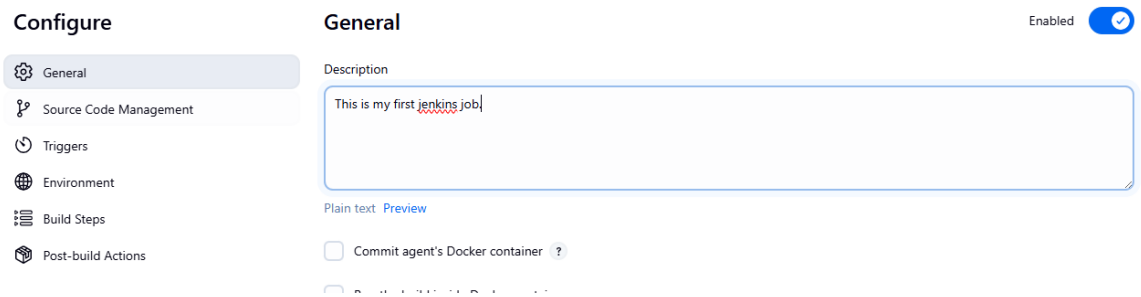
In short, Jenkins acts as the central hub for your entire software delivery pipeline, from the moment a developer commits code to the moment it's running for users.

First Job:

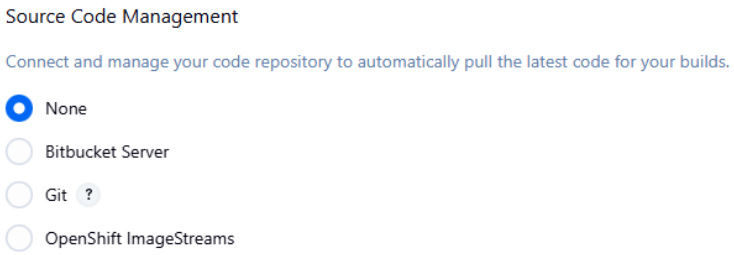
1) we enter the name of the new job that has to be created and select the type of job that we have to do. (Here we have chosen Freestyle project)



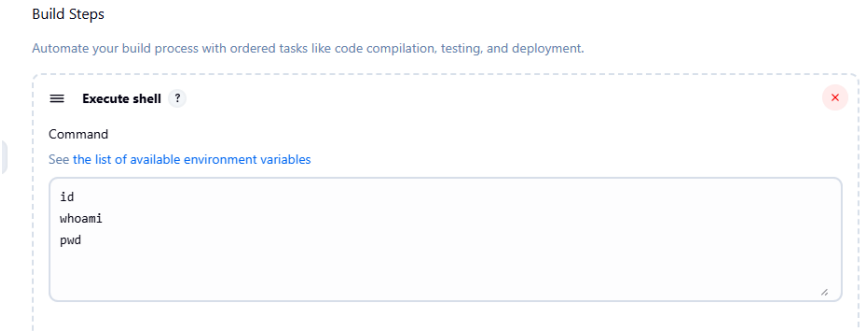
2) Give a short description.



3) We can connect our job to a source code manager but for this project we select it as None



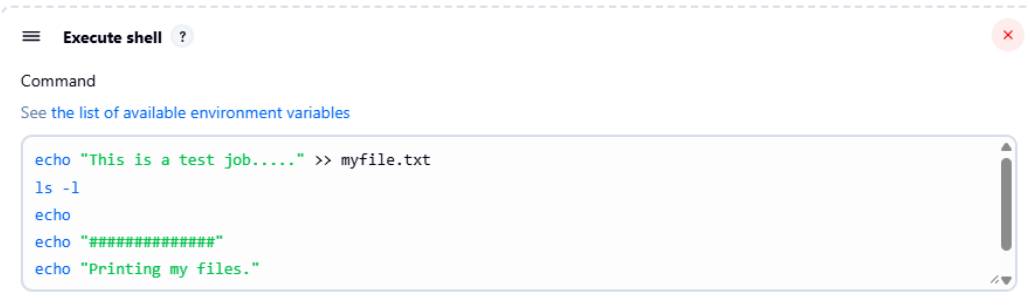
4) Afterwards we move on to the build steps: Here we want to execute shell commands.



5) We then save and apply changes and build our job to see the console output:

```
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/FirstJenkinsJob
[FirstJenkinsJob] $ /bin/sh -xe /tmp/jenkins10569208798115566110.sh
+ id
uid=130(jenkins) gid=137(jenkins) groups=137(jenkins)
+ whoami
jenkins
+ pwd
/var/lib/jenkins/workspace/FirstJenkinsJob
Finished: SUCCESS
```

6) We then proceed to write more shell scripts:

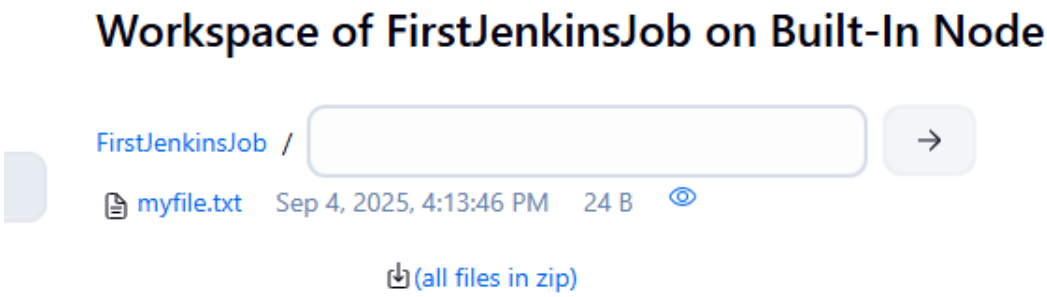



```
cat myfile.txt
```

7) After applying build again when we move to the console to see the output it shows this:

```
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/FirstJenkinsJob
[FirstJenkinsJob] $ /bin/sh -xe /tmp/jenkins8713281843428836465.sh
+ id
uid=130(jenkins) gid=137(jenkins) groups=137(jenkins)
+ whoami
jenkins
+ pwd
/var/lib/jenkins/workspace/FirstJenkinsJob
[FirstJenkinsJob] $ /bin/sh -xe /tmp/jenkins3997582976138882623.sh
+ echo This is a test job.....
+ ls -l
total 4
-rw-r--r-- 1 jenkins jenkins 24 Sep  4 16:13 myfile.txt
+ echo

+ echo ##############
##############
+ echo Printing my files.
Printing my files.
+ cat myfile.txt
This is a test job.....
Finished: SUCCESS
```

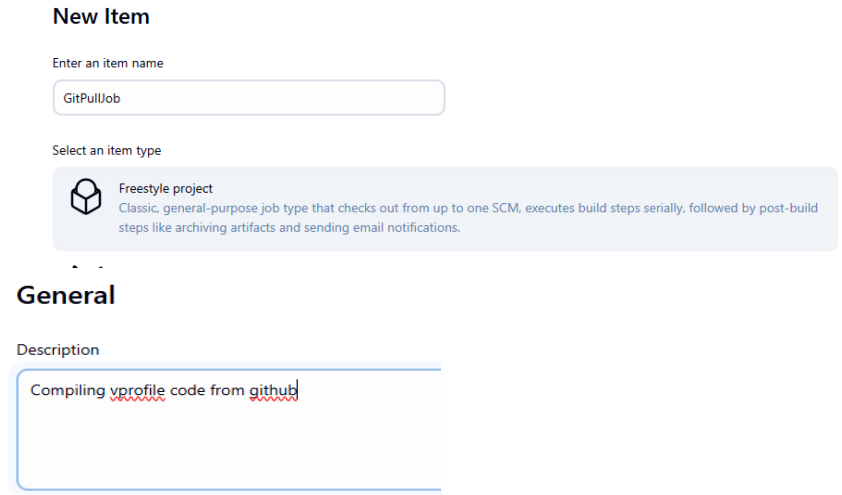
Additionally, we can take a look at our workspace and we can download the zip containing all the files make(In this case: **myfile.txt**)



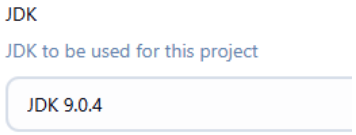
GIT PULL JOB

Now we are going to perform a git pull job and it will be a freestyle project.

1) As seen earlier we provide the name and description of our build job.

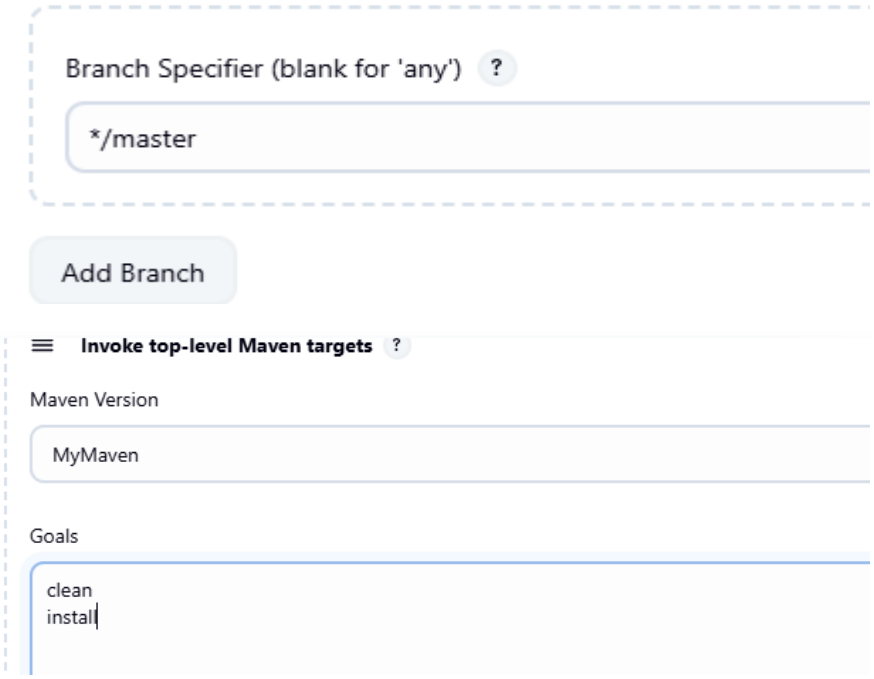


2) We mention the version JDK needed to use for this project:



3) For this job we will be using Git as our source code manager:

We will need to specify a branch to build because a single Git repository often contains many different versions of our code at the same time, and Jenkins needs to know exactly **which version** it should build and test.

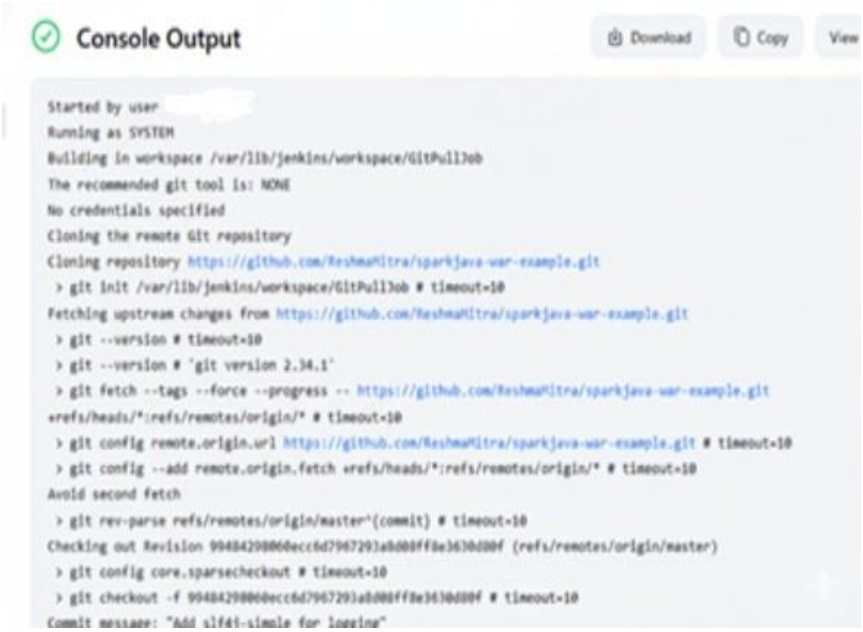


We invoke top-level Maven targets Jenkins job to tell Maven exactly what we want it to do.

In simple terms, "top-level Maven targets" are just the goals or phases we want Maven to run, like **clean, install, or deploy**.

This is necessary because Maven projects are built using a **lifecycle**, which is a predefined sequence of steps (called phases). We need to tell Jenkins *which phase* in that lifecycle we want to run.

After we save and then apply changes and start the build:

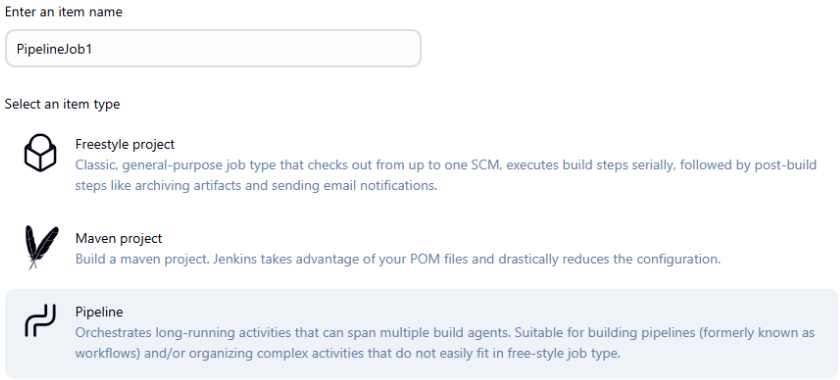


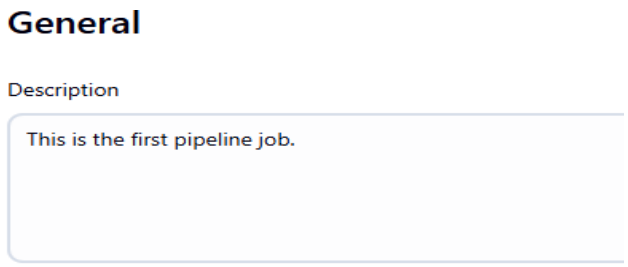

```
First time build. Skipping changelog.
Unpacking https://repo.maven.apache.org/maven2/org/apache/maven/apache-maven/3.9.11/apache-maven-3.9.11-bin.zip to
/var/lib/jenkins/tools/hudson.tasks.Maven_MavenInstallation/MyMaven on Jenkins
[GitPullJob] $ /var/lib/jenkins/tools/hudson.tasks.Maven_MavenInstallation/MyMaven/bin/mvn clean install
[INFO] Scanning for projects...
[INFO]
[INFO] ------------< sparkjava-hello-world:sparkjava-hello-world >-------------
[INFO] Building sparkjava-hello-world 1.0
[INFO]   from pom.xml
[INFO] --------------------------------[ war ]---------------------------------
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-enforcer-
plugin/1.3.1/maven-enforcer-plugin-1.3.1.pom
Progress (1): 1.4/6.5 kB
Progress (1): 2.8/6.5 kB
Progress (1): 4.1/6.5 kB
Progress (1): 5.5/6.5 kB
Progress (1): 6.5 kB

Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-enforcer-
plugin/1.3.1/maven-enforcer-plugin-1.3.1.pom (6.5 kB at 22 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/enforcer/enforcer/1.3.1/enforcer-
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/4.0.1/plexus-utils-
4.0.1.jar (193 kB at 2.0 MB/s)
[INFO] Installing /var/lib/jenkins/workspace/GitPullJob/pom.xml to /var/lib/jenkins/.m2/repository/sparkjava-hello-
world/sparkjava-hello-world/1.0/sparkjava-hello-world-1.0.pom
[INFO] Installing /var/lib/jenkins/workspace/GitPullJob/target/sparkjava-hello-world-1.0.war to
/var/lib/jenkins/.m2/repository/sparkjava-hello-world/sparkjava-hello-world/1.0/sparkjava-hello-world-1.0.war
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  11.666 s
[INFO] Finished at: 2025-09-04T16:33:26+05:30
[INFO] ------------------------------------------------------------------------
Finished: SUCCESS
```

DAY 5

Pipeline Job in Jenkins

1 Enter the name and type of job. In this case we will choose Pipeline

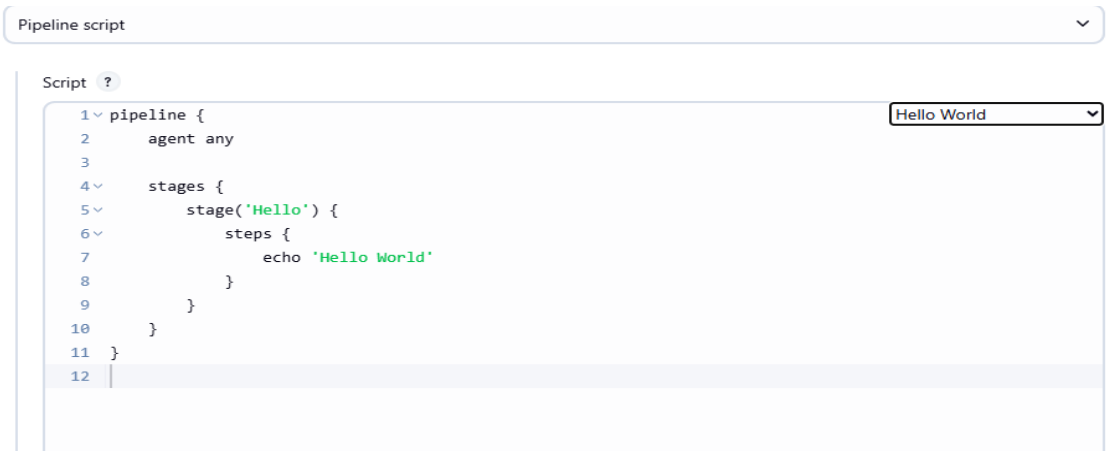




About pipeline jobs:

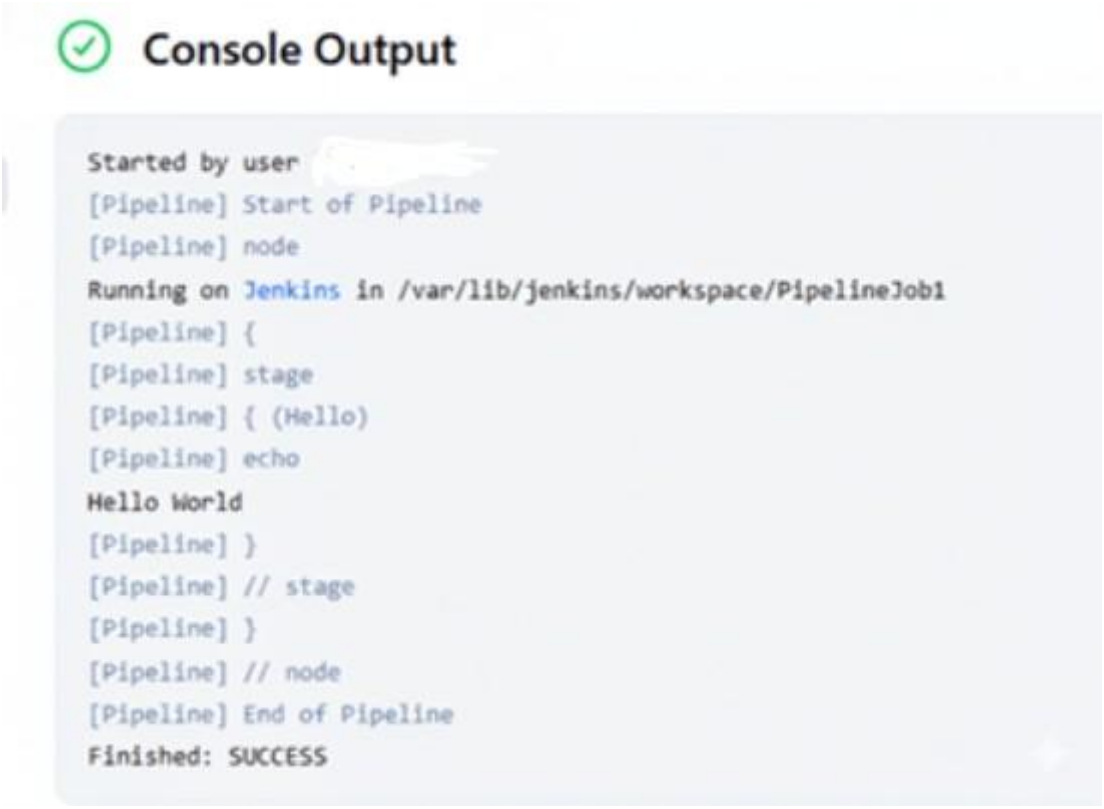
A Jenkins Pipeline is the modern standard for defining and running our entire build, test, and deployment process as code.
Instead of creating multiple "Freestyle jobs" and manually linking them together in the web UI, we define our entire workflow in a single text file called a **Jenkinsfile**.
This Jenkinsfile lives with our project's source code (e.g., in our Git repository). This approach is called **Pipeline-as-Code**.

In this job we will be using Pipeline Script inside our Jenkins job.



This is a Hello world script.

After applying the changes and running the build job, we get this Console Output:



We perform another Pipeline Job using git:

In the pipeline script we have used maven as our build tool for java project:

```
pipeline{
    agent any
    tools{
        maven "MyMaven"
    }
    stages{
        stage('Build'){
            steps{
                git 'https://github.com/jglick/simple-maven-project-with-tests.git'
                sh "mvn -Dmaven.test.failure.ignore=true clean package"
            }
            post{
                success{
                    junit '**/target/surefire-reports/TEST-*.xml'
                    archiveArtifacts 'target/*.jar'
```

```
Running on Jenkins in /var/lib/jenkins/workspace/PipelineGit
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Tool Install)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Build)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] git
The recommended git tool is: NONE
```

```
No credentials specified
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/PipelineGit/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/jglick/simple-maven-project-with-tests.git # timeout=10
Fetching upstream changes from https://github.com/jglick/simple-maven-project-with-tests.git
 > git --version # timeout=10
 > git --version # 'git version 2.34.1'
 > git fetch --tags --force --progress -- https://github.com/jglick/simple-maven-project-with-tests.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 9a5e9832814aa17fa7e514e5890fc6cf913d9a77 (refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 9a5e9832814aa17fa7e514e5890fc6cf913d9a77 # timeout=10
 > git branch -a -v --no-abbrev # timeout=10
 > git branch -D master # timeout=10
 > git checkout -b master 9a5e9832814aa17fa7e514e5890fc6cf913d9a77 # timeout=10
Commit message: "Creating a main class so there is something for `javadoc:javadoc` to do"
 > git rev-list --no-walk 9a5e9832814aa17fa7e514e5890fc6cf913d9a77 # timeout=10
[Pipeline] sh
+ mvn -Dmaven.test.failure.ignore=true clean package
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------< test:simple-maven-project-with-tests >----------------
[INFO] Building simple-maven-project-with-tests 1.0-SNAPSHOT
```

```
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ simple-maven-project-with-tests ---
[INFO] Deleting /var/lib/jenkins/workspace/PipelineGit/target
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ simple-maven-project-with-tests ---
[INFO] skip non existing resourceDirectory /var/lib/jenkins/workspace/PipelineGit/src/main/resources
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ simple-maven-project-with-tests ---
[INFO] Recompiling the module because of changed source code.
[INFO] Compiling 1 source file with javac [debug target 1.8] to target/classes
[WARNING] bootstrap class path not set in conjunction with -source 8
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ simple-maven-project-with-tests ---
[INFO] skip non existing resourceDirectory /var/lib/jenkins/workspace/PipelineGit/src/test/resources
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ simple-maven-project-with-tests ---
[INFO] Recompiling the module because of changed dependency.
[INFO] Compiling 3 source files with javac [debug target 1.8] to target/test-classes
[WARNING] bootstrap class path not set in conjunction with -source 8
[INFO] /var/lib/jenkins/workspace/PipelineGit/src/test/java/test/Base.java:
/var/lib/jenkins/workspace/PipelineGit/src/test/java/test/Base.java uses or overrides a deprecated API.
[INFO] /var/lib/jenkins/workspace/PipelineGit/src/test/java/test/Base.java: Recompile with -Xlint:deprecation for
```

```
details.
[INFO]
[INFO] --- surefire:2.18.1:test (default-test) @ simple-maven-project-with-tests ---
[INFO] Surefire report directory: /var/lib/jenkins/workspace/PipelineGit/target/surefire-reports

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running test.OtherTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.017 sec - in test.OtherTest
Running test.SomeTest
Tests run: 6, Failures: 0, Errors: 0, Skipped: 2, Time elapsed: 0.001 sec - in test.SomeTest

Results :

Tests run: 7, Failures: 0, Errors: 0, Skipped: 2

[INFO]
[INFO] --- jar:3.4.1:jar (default-jar) @ simple-maven-project-with-tests ---
[INFO] Building jar: /var/lib/jenkins/workspace/PipelineGit/target/simple-maven-project-with-tests-1.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.455 s
```

```
[INFO] Finished at: 2025-09-11T16:36:35+05:30
[INFO] ------------------------------------------------------------------------
Post stage
[Pipeline] junit
Recording test results
[Checks API] No suitable checks publisher found.
[Pipeline] archiveArtifacts
Archiving artifacts
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Here we see how pipeline jobs are executed in Jenkins.

Day 6 (Installation and docker commands)

To install docker: https://docs.docker.com/engine/install/ubuntu/

We also need to create an account on docker hub at hub.docker.com

Docker Commands

1) docker run nginx: Runs a new container using the **nginx** image in the foreground.

```
^Croot@jenkins-minikube:~# sudo docker run nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
8c7716127147: Pull complete
250b90fb2b9a: Pull complete
5d8ea9f4c626: Pull complete
58d144c4badd: Pull complete
b459da543435: Pull complete
8da8ed3552af: Pull complete
54e822d8ee0c: Pull complete
Digest: sha256:3b7732505933ca591ce4a6d860cb713ad96a3176b82f7979a8dfa9973486a0d6
Status: Downloaded newer image for nginx:latest
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2025/10/13 05:14:17 [notice] 1#1: using the "epoll" event method
2025/10/13 05:14:17 [notice] 1#1: nginx/1.29.2
2025/10/13 05:14:17 [notice] 1#1: built by gcc 14.2.0 (Debian 14.2.0-19)
2025/10/13 05:14:17 [notice] 1#1: OS: Linux 6.8.0-79-generic
2025/10/13 05:14:17 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2025/10/13 05:14:17 [notice] 1#1: start worker processes
2025/10/13 05:14:17 [notice] 1#1: start worker process 29
2025/10/13 05:14:17 [notice] 1#1: start worker process 30
^C2025/10/13 05:16:09 [notice] 1#1: signal 2 (SIGINT) received, exiting
2025/10/13 05:16:09 [notice] 30#30: exiting
2025/10/13 05:16:09 [notice] 29#29: exiting
2025/10/13 05:16:09 [notice] 30#30: exit
2025/10/13 05:16:09 [notice] 29#29: exit
2025/10/13 05:16:09 [notice] 1#1: signal 17 (SIGCHLD) received from 30
2025/10/13 05:16:09 [notice] 1#1: worker process 30 exited with code 0
2025/10/13 05:16:09 [notice] 1#1: signal 29 (SIGIO) received
2025/10/13 05:16:09 [notice] 1#1: signal 17 (SIGCHLD) received from 29
2025/10/13 05:16:09 [notice] 1#1: worker process 29 exited with code 0
2025/10/13 05:16:09 [notice] 1#1: exit
root@jenkins-minikube:~#
```

2) docker ps and docker ps -a: We use "*docker ps*" to quickly check the status of our active services, and use "*docker ps -a*" when we need to manage the complete list of containers on your system, such as removing old, stopped containers.

```
root@jenkins-minikube:~# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
root@jenkins-minikube:~#
```

```
root@jenkins-minikube:~# docker ps -a
CONTAINER ID   IMAGE         COMMAND    CREATED              STATUS                          PORTS     NAMES
17975cbef52b   hello-world   "/hello"   About a minute ago   Exited (0) About a minute ago             silly_haslett
root@jenkins-minikube:~#
```

3) docker images: Lists all Docker images available locally.

```
root@jenkins-minikube:~# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx         latest    07ccdb783875   5 days ago     160MB
ubuntu        latest    97bed23a3497   11 days ago    78.1MB
hello-world   latest    1b44b5a3e06a   2 months ago   10.1kB
```

4) docker rm <container name>: Removes a stopped container from the system.

```
root@jenkins-minikube:~# docker rm 7f60 5fbd 5e9a dd13 1797
7f60
5fbd
dd13
1797
Error response from daemon: cannot remove container "5e9a": container is running: stop the container before removing or force remove
root@jenkins-minikube:~# docker stop 5e9a
5e9a
root@jenkins-minikube:~# docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS                     PORTS     NAMES
5e9ade575225   nginx     "/docker-entrypoint.…"   15 minutes ago  Exited (0) 4 seconds ago             infallible_perlman
root@jenkins-minikube:~# docker rm 5e9a
5e9a
```

6) docker rmi <image_name>: Deletes a Docker image.

```
root@jenkins-minikube:~# docker rmi  07cc  97be  1b44
Untagged: nginx:latest
Untagged: nginx@sha256:3b7732505933ca591ce4a6d860cb713ad96a3176b82f7979a8dfa9973486a0d6
Deleted: sha256:07ccdb7838758e758a4d52a9761636c385125a327355c0c94a6acff9babff938
Deleted: sha256:71b75b17511f67932ccf71e2046c6d1b4fe17a594134c765bf71c874dedc7027
Deleted: sha256:c76da83ebfb3e35184d1a7105d03cecfb144d07cb4dae12378392040fbd44615
Deleted: sha256:f825da5ac2d31a2c717db4fa159b6728b33a94bcb7285bd1dfebcbbe23ebd185
Deleted: sha256:31333e0f1ecf5940213f8b1ed4eb3e4d78d77fa5fb59c1cd05a6672690ed133c
Deleted: sha256:e72eb931613f8c1b5baf39f864877beee0780af9f9d20c5a02790d0146f0b012
Deleted: sha256:775e3183eed457c31a0855ac7a55b6e0cb8d40554e9524cb6d503cb2351e0b10
Deleted: sha256:1d46119d249f7719e1820e24a311aa7c453f166f714969cffe89504678eaa447
Untagged: ubuntu:latest
Untagged: ubuntu@sha256:59a458b76b4e8896031cd559576eac7d6cb53a69b38ba819fb26518536368d86
Deleted: sha256:97bed23a34971024aa8d254abbe67b7168772340d1f494034773bc464e8dd5b6
Deleted: sha256:073ec47a8c22dcaa4d6e5758799ccefe2f9bde943685830b1bf6fd2395f5eabc
Untagged: hello-world:latest
Untagged: hello-world@sha256:54e66cc1dd1fcb1c3c58bd8017914dbed8701e2d8c74d9262e26bd9cc1642d31
Deleted: sha256:1b44b5a3e06a9aae883e7bf25e45c100be0bb81a0e01b32de604f3ac44711634
Deleted: sha256:53d204b3dc5ddbc129df4ce71996b8168711e211274c785de5e0d4eb68ec3851
```

7) docker pull ubuntu: Downloads the ubuntu image from Docker Hub.

```
root@jenkins-minikube:~# docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
4b3ffd8ccb52: Pull complete
Digest: sha256:59a458b76b4e8896031cd559576eac7d6cb53a69b38ba819fb26518536368d86
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
root@jenkins-minikube:~#
```

8) docker run ubuntu sleep 100: Runs an Ubuntu container and executes sleep 100 (runs for 100 seconds).

```
C:\Users\      >docker ps -a
CONTAINER ID   IMAGE     COMMAND       CREATED         STATUS                      PORTS     NAMES
9fd13c618b27   ubuntu    "sleep 100"   2 minutes ago   Exited (0) 51 seconds ago             fervent_einstein
```

9) docker run -d ubuntu sleep 50: Runs the container in detached mode (background).

```
C:\Users\      >docker run -d ubuntu sleep 50
6d5940101a8164221e488db53e4cbbf877f6bd4b236735bb651e5c6db19d668b

C:\Users\      >docker ps
CONTAINER ID   IMAGE     COMMAND       CREATED         STATUS         PORTS     NAMES
6d5940101a81   ubuntu    "sleep 50"    5 seconds ago   Up 4 seconds             loving_johnson
```

10) docker attach <container_name>**:** Attaches your terminal to the container's ongoing output/session.

```
C:\Users\      >docker run -d ubuntu sleep 100
5a47096e54b84a044446100 44b3558113d984092f5737eff705e707c8b0506dc

C:\Users\      >docker ps
CONTAINER ID   IMAGE     COMMAND       CREATED         STATUS         PORTS     NAMES
5a47096e54b8   ubuntu    "sleep 100"   2 seconds ago   Up 2 seconds             vigorous_moser

C:\Users\      >docker attach vigorous_moser
```

11) docker exec <container_name> cat /etc/hosts: Runs a command inside a running container without attaching.

```
C:\Users\      >docker exec awesome_cerf cat /etc/hosts
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.17.0.2      92074656510e
```

12) docker exec <id> cat /etc/hosts: Runs a command inside a running container using its container ID prefix.

13) docker attach <id>: Attaches to the container with ID starting with its container ID prefix.

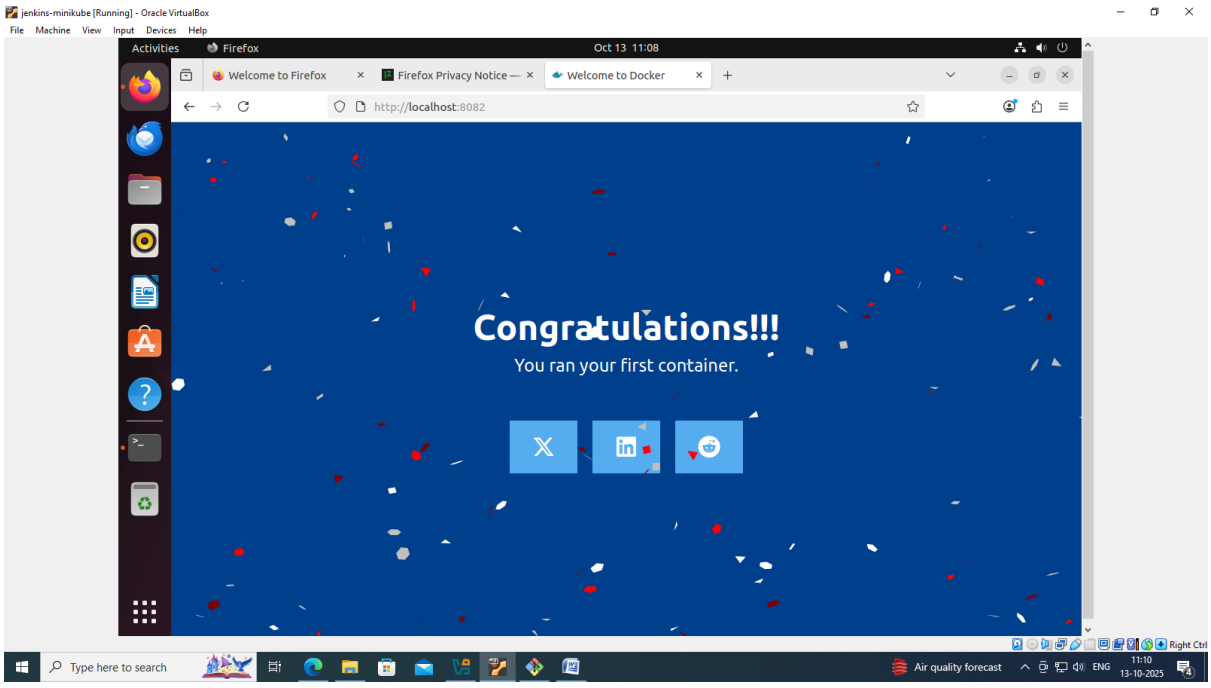
14) docker inspect 5a: Shows detailed JSON metadata for the container/image with ID/name 5a.

15) docker run -it ubuntu bash: Runs an Ubuntu container interactively with a Bash shell.

```
C:\Users\        >docker run -it ubuntu bash
root@ee7ac88c2a7b:/# pwd
/
root@ee7ac88c2a7b:/# whoami
root
root@ee7ac88c2a7b:/#
```

16) docker run -d -p 8082:80 docker/welcome-to-docker: Runs a web app container in background and maps **host port 8082 → container port 80**.

```
root@jenkins-minikube:~# sudo docker run -d -p 8082:80 docker/welcome-to-docker
Unable to find image 'docker/welcome-to-docker:latest' locally
latest: Pulling from docker/welcome-to-docker
9824c27679d3: Pull complete
a5585638209e: Pull complete
fd372c3c84a2: Pull complete
958a74d6a238: Pull complete
c1d2dc189e38: Pull complete
828fa206d77b: Pull complete
bdaad27fd04a: Pull complete
9745203f5d34: Pull complete
Digest: sha256:c4d56c24da4f009ecf8352146b43497fe78953edb4c679b841732beb97e588b0
Status: Downloaded newer image for docker/welcome-to-docker:latest
e12b5ed7d84cfadd6f8015fb8971a4c2ce03ad0baa01cb84b5feefb6062b33b4
root@jenkins-minikube:~#
```



DAY 7:

Building our own docker image:

Source: https://devopscube.com/build-docker-image/

```
root@jenkins-minikube:~# ls
kubectl  minikube_latest_amd64.deb  nginx-image  snap  vboxpostinstall.sh
root@jenkins-minikube:~#
```

We create the following files:

mkdir nginx-image && cd nginx-image
mkdir files

The file structure becomes:

```
root@jenkins-minikube:~/nginx-image# ls
Dockerfile  files
root@jenkins-minikube:~/nginx-image#
```

Next we create index.html and default file:

```
root@jenkins-minikube:~/nginx-image# cd files
root@jenkins-minikube:~/nginx-image/files# ls
default  index.html
root@jenkins-minikube:~/nginx-image/files#
```

Index.html

```
<html>
  <head>
    <title>Dockerfile</title>
  </head>
  <body>
    <div class="container">
      <h1>My App</h1>
      <h2>This is my first app</h2>
      <p>Hello everyone, This is running via Docker container</p>
    </div>
  </body>
</html>
```

Create this using *vi index.html*

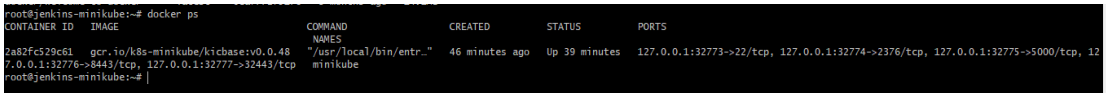
Next we do *vi default* for the Default file

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    root /usr/share/nginx/html;
    index index.html index.htm;

    server_name _;
    location / {
        try_files $uri $uri/ =404;
    }
}
```

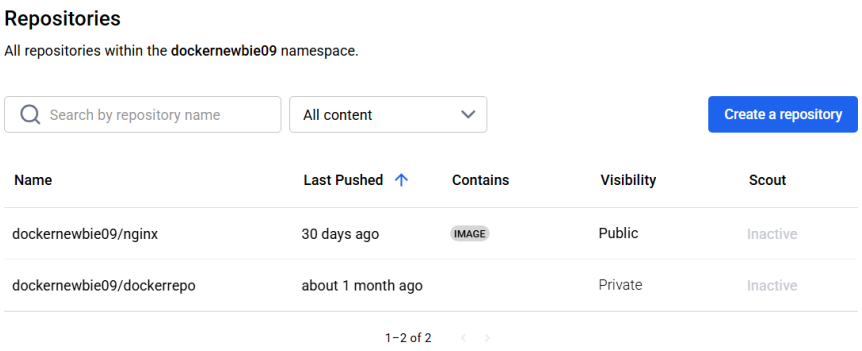
The contents of the Dockerfile will be:

```
FROM ubuntu:18.04
LABEL maintainer="contact@devopscube.com"
RUN  apt-get -y update && apt-get -y install nginx
COPY files/default /etc/nginx/sites-available/default
COPY files/index.html /usr/share/nginx/html/index.html
EXPOSE 80
CMD ["/usr/sbin/nginx", "-g", "daemon off;"]
```

To build the docker image we use: docker build -t nginx:<tag>
Then we can run the docker image on our machine



After pushing the image to docker hub we can view it in our repositories:



Day 8:

Installing Kubernetes and Commands

Step 1: https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/

We go to this link and follow the required steps

Step 2: Installing minikube by using this command:

```
C:\Users\      >curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube_latest_amd64.deb
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
```

Kubernetes Commands:

To check minikube status we use *minikube status*

```
root@jenkins-minikube:~# minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

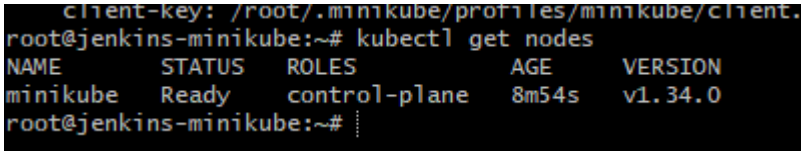
kubectl –cluster-info: Displays information about the Kubernetes cluster, including the control plane and API server URLs. It helps verify if the cluster is up and running.

```
root@jenkins-minikube:~# kubectl cluster-info
Kubernetes control plane is running at https://192.168.49.2:8443
CoreDNS is running at https://192.168.49.2:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
root@jenkins-minikube:~#
```

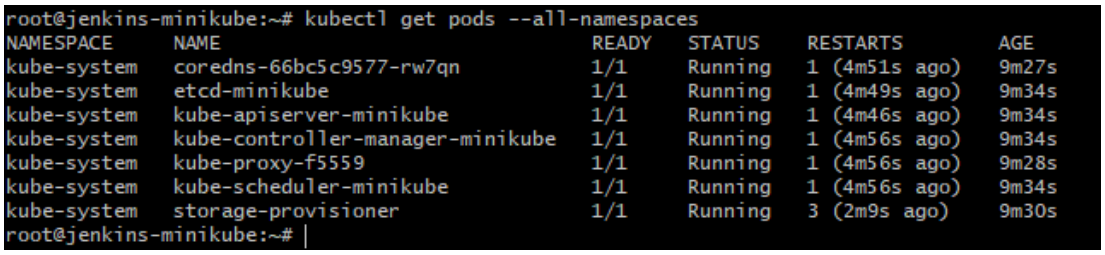
kubectl config view: Shows the details of the current Kubernetes configuration, including the cluster, context, and user settings. It’s useful for checking the kubeconfig file to ensure you are connected to the correct cluster.

```
To further debug and diagnose cluster problems, use 'kubectl cluster-info
root@jenkins-minikube:~# kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority: /root/.minikube/ca.crt
    extensions:
    - extension:
        last-update: Thu, 30 Oct 2025 15:33:39 IST
        provider: minikube.sigs.k8s.io
        version: v1.37.0
      name: cluster_info
    server: https://192.168.49.2:8443
  name: minikube
contexts:
- context:
    cluster: minikube
    extensions:
    - extension:
        last-update: Thu, 30 Oct 2025 15:33:39 IST
        provider: minikube.sigs.k8s.io
        version: v1.37.0
      name: context_info
    namespace: default
    user: minikube
  name: minikube
current-context: minikube
kind: Config
users:
- name: minikube
  user:
    client-certificate: /root/.minikube/profiles/minikube/client.crt
    client-key: /root/.minikube/profiles/minikube/client.key
root@jenkins-minikube:~#
```

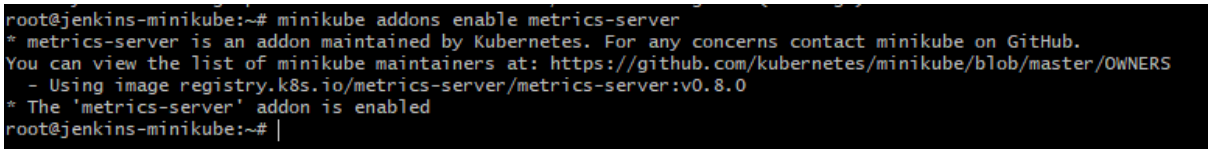
kubectl get nodes: Lists all nodes in the Kubernetes cluster, showing their status, roles, and other details. This helps in checking the availability and health of the cluster's nodes.



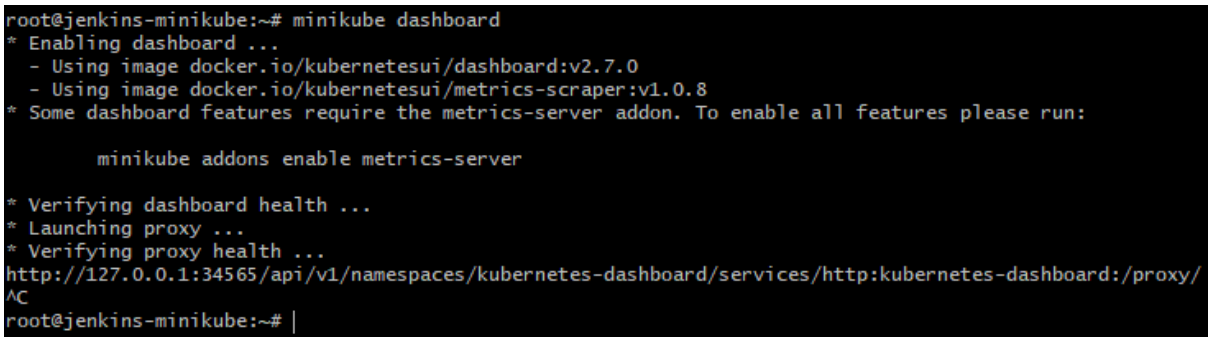
kubectl get pods --all-namespaces: Retrieves information about all pods running in all namespaces in the Kubernetes cluster. This is useful for monitoring pod status across the entire cluster.



minikube addons enable metrics-server: Enables the metrics-server add-on in Minikube. The metrics-server collects resource usage metrics (like CPU and memory usage) for the Kubernetes cluster, enabling features like horizontal pod autoscaling.



minikube dashboard: Starts the Minikube dashboard, which opens a web-based UI to visualize and manage the cluster. It provides an interactive interface to view and manage Kubernetes resources like pods, services, and deployments.

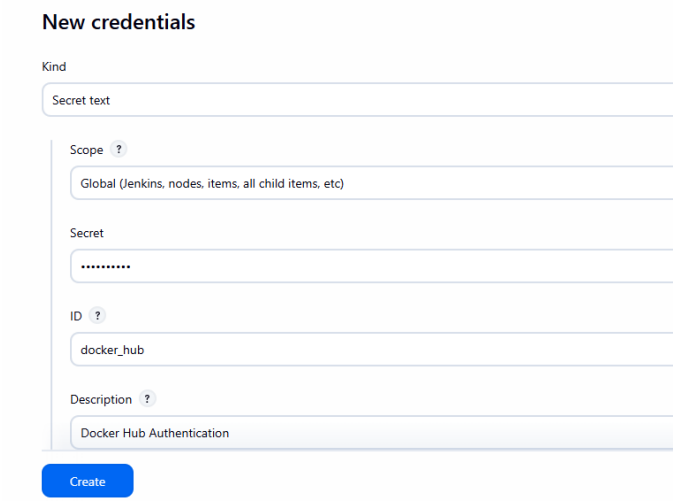


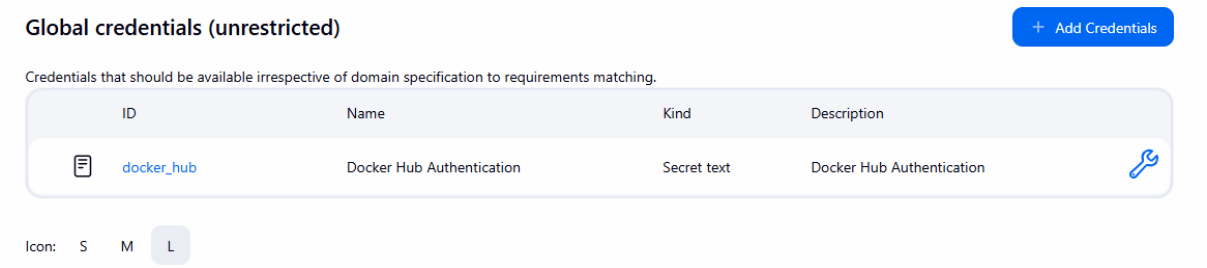
Day 9 and 10:

Jenkins CI/CD Pipeline job:

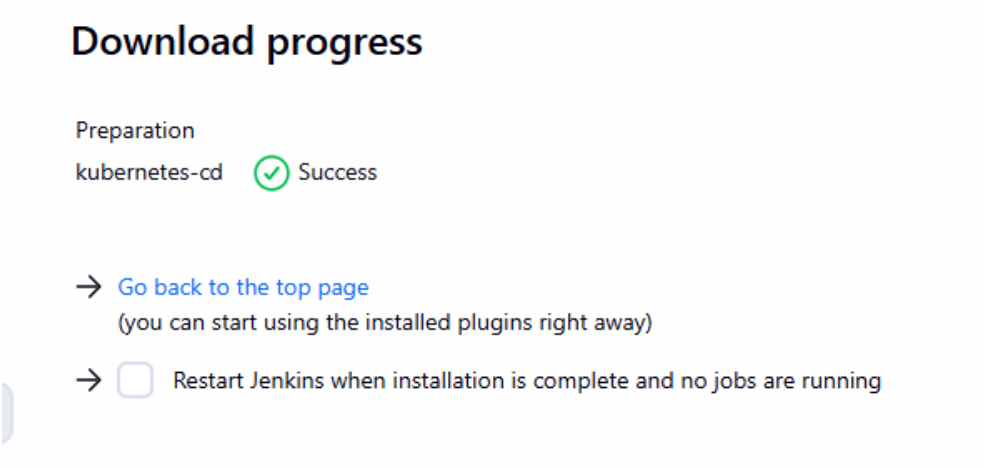
First, we need to configure authentication credentials that will allow this process to be possible.

The credentials of Docker Hub are given:





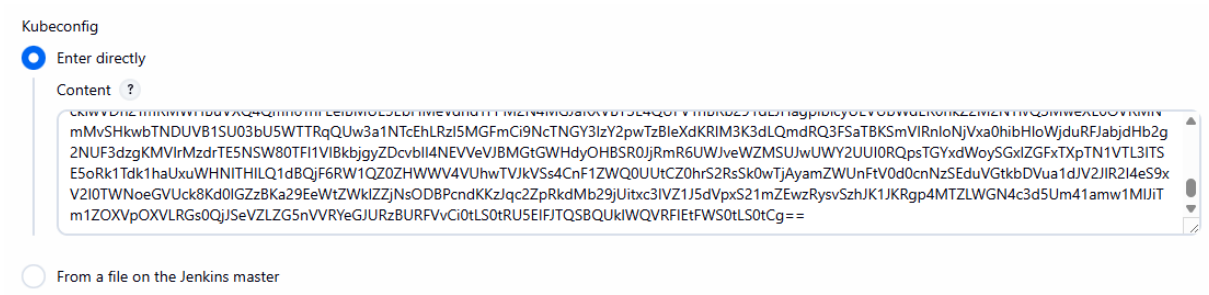
Next we need to download the “Kubernetes-cd.hpi” plugin for Kubernetes from internet to our local machine:
https://updates.jenkins.io/download/plugins/kubernetes-cd/1.0.0/kubernetes-cd.hpi

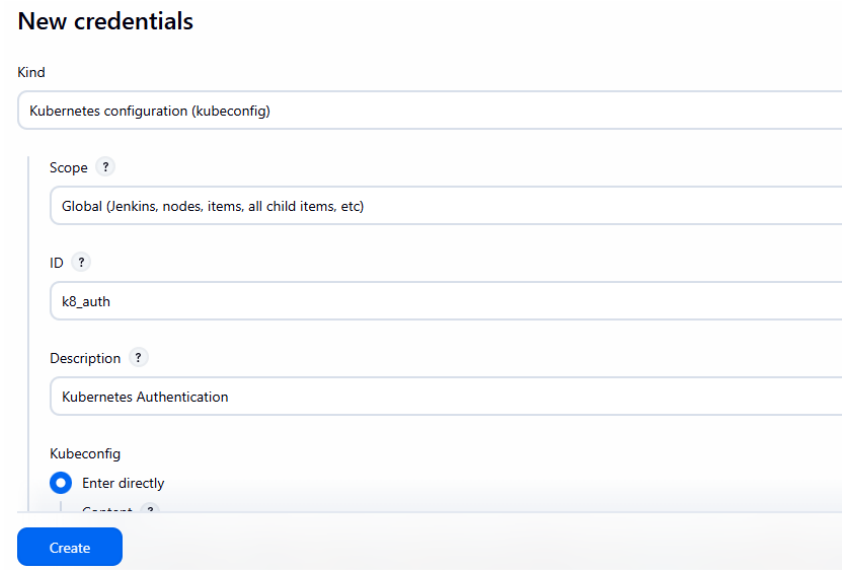


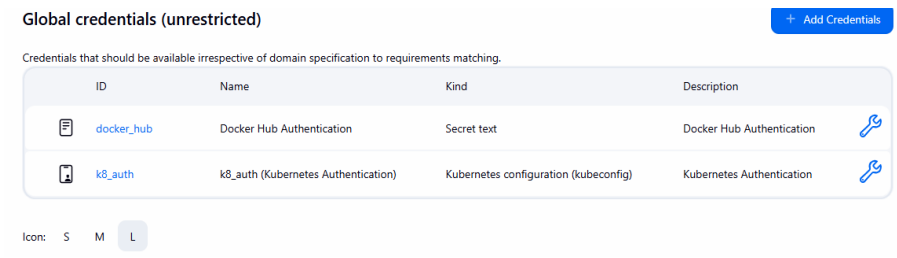
After this we need to restart the Jenkins service (from Ubuntu server)
sudo systemctl restart jenkins

Next we run: kubectl config view --flatten

We enter the output of the command directly into the Kubernetes Credentials:





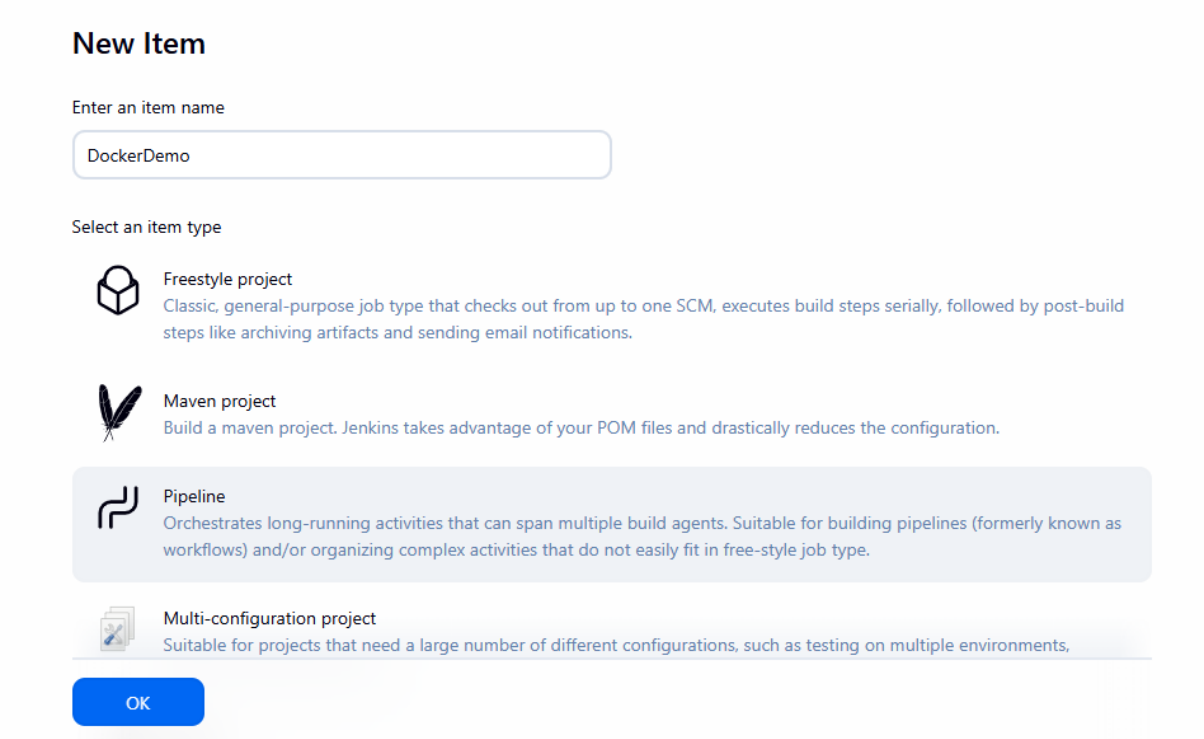


Next we set up the CI/CD pipeline:

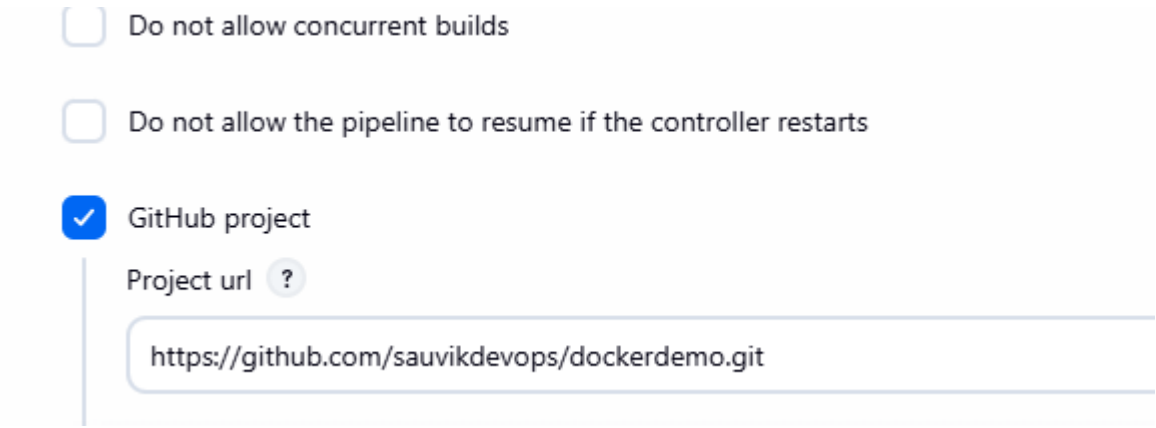
We will need a Dockerfile to create the image and a Kubernetes definition file that defines the deployment and service to expose the deployment

First, we need to configure authentication credentials that will allow this process to be possible.

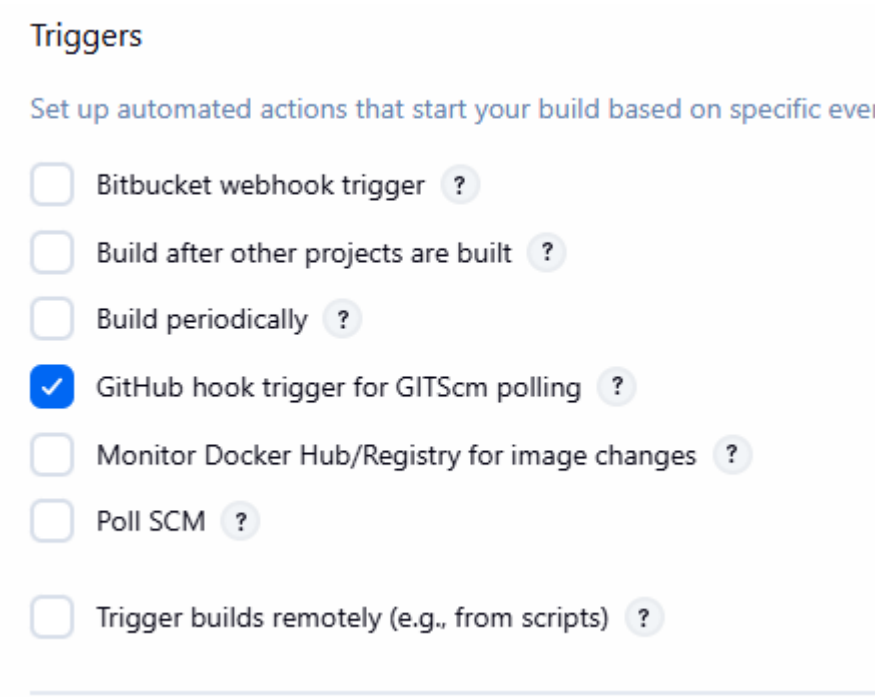
The credentials are configured are as follows:



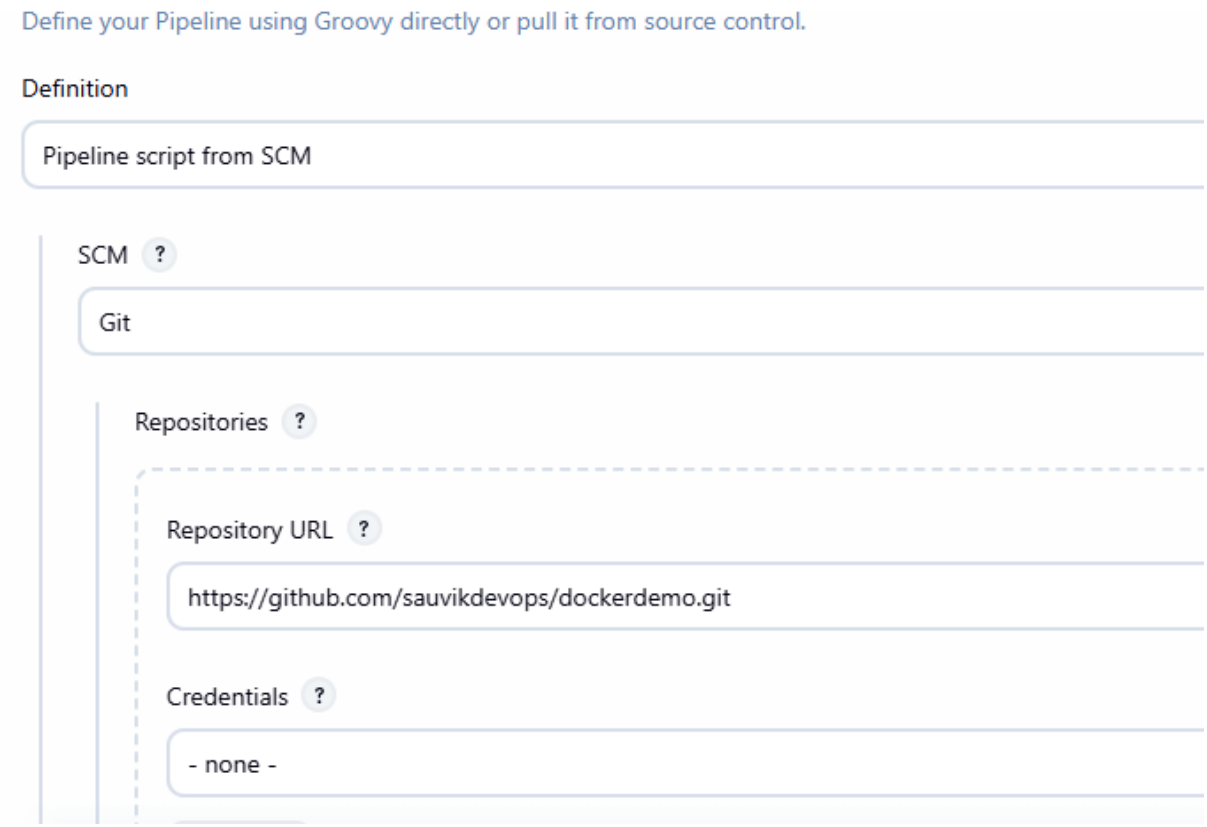
We put the name of the github repository:



Here we use the GitHub hook trigger: The integration between GitHub and Jenkins enables continuous integration (CI) and continuous delivery (CD) workflows by automatically initiating builds and other processes when specific events occur in the GitHub repo.



We get our Pipeline script (Jenkinsfile) from the repo itself



Next we create the Jenkinsfile and push the code into github.

```
pipeline {
    agent any

    stages {
        stage('Checkout GitHub repo') {
            steps {
                checkout scmGit(branches: [[name: '*/main']], extensions: [], userRemoteConfigs: [[url: 'https://github.com/[redacted]/dockerdemo.git']])
            }
        }
        stage('Build and Tag Docker Image') {
            steps {
                script {
                    sh 'docker build . -t hellodocker'
                    sh 'docker tag hellodocker dockernewbie09/learning'
                }
            }
        }
        stage('Push the Docker Image to DockerHUb') {
            steps {
                script {
                    withCredentials([string(credentialsId: 'docker_hub', variable: 'docker_hub')]) {
                        sh """
                            echo "${docker_hub}" | docker login -u dockernewbie09 --password-stdin
                            docker push dockernewbie09/learning
                        """
                    }
                }
            }
        }

        stage('Deploy deployment and service file') {
            steps {
                script {
                    kubernetesDeploy configs: 'deploymentsvc.yaml', kubeconfigId: 'k8_auth'
```