Don Bosco Institute Of Technology,
(DBIT),Mumbai

Department Of Information Technology

LAB JOURNAL

On

ITL503: DevOps Lab

By

24 Swasti Jain

Academic Year: Nov, 2022

INDEX

EXPERIMENT A - Self Study Date of Experiment: 11/07/2022

Prerequisite:

C, Python, Java, Software Engineering, Cloud

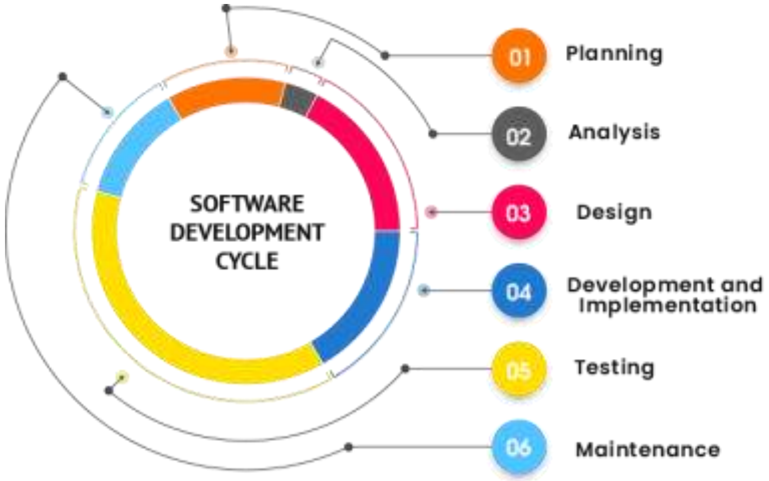
Objective:
To make students understand DevOps, for what? Why? and by whom?

Aim:
To do self-study on basics understanding of DevOps

Procedure:
Individual students will perform this self-study over internet resources and will submit a report on their study and give a viva voce on or before 18/07/2022, and they will be graded based on rubrics.

The topics for self-study are :

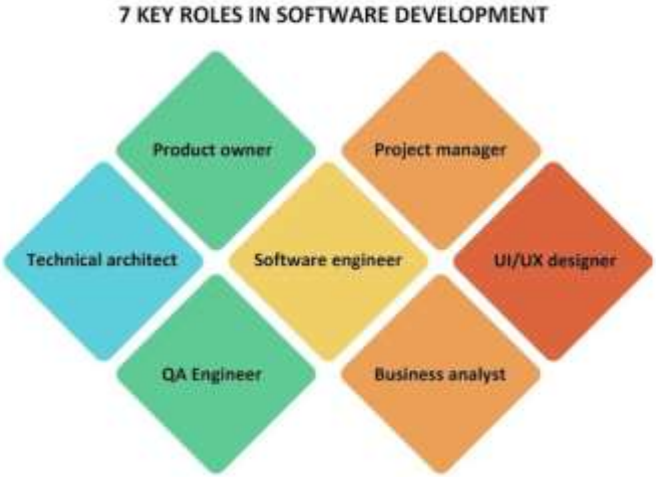
1. Development of Software

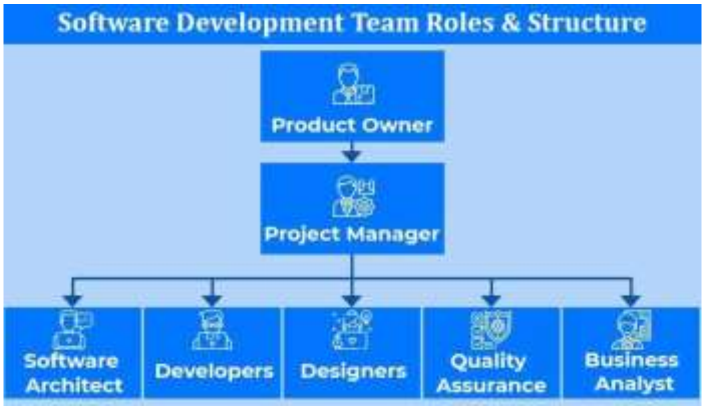


2. Software Engineering

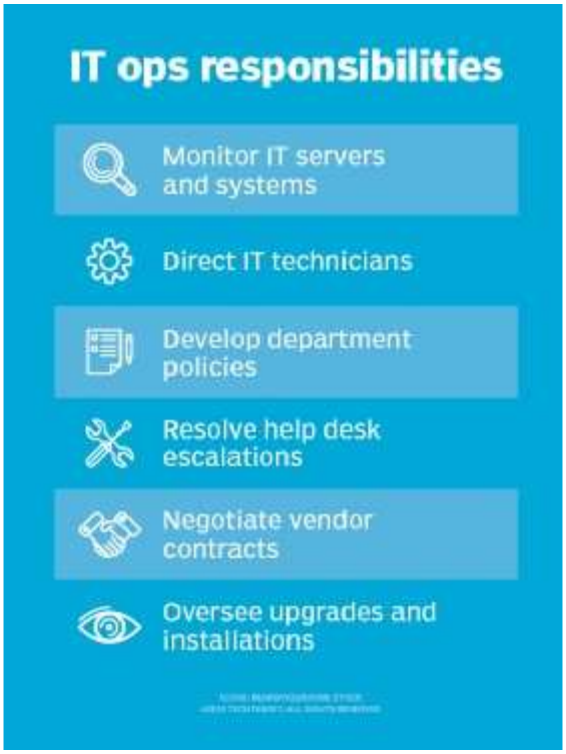
SOFTWARE ENGINEERING
IMPLEMENTATION
DEVELOPMENT
PROGRAMMING
ANALYZE
DESIGN
PLANNING
TESTING
VALIDATION AND VERIFICATION

3. Development team





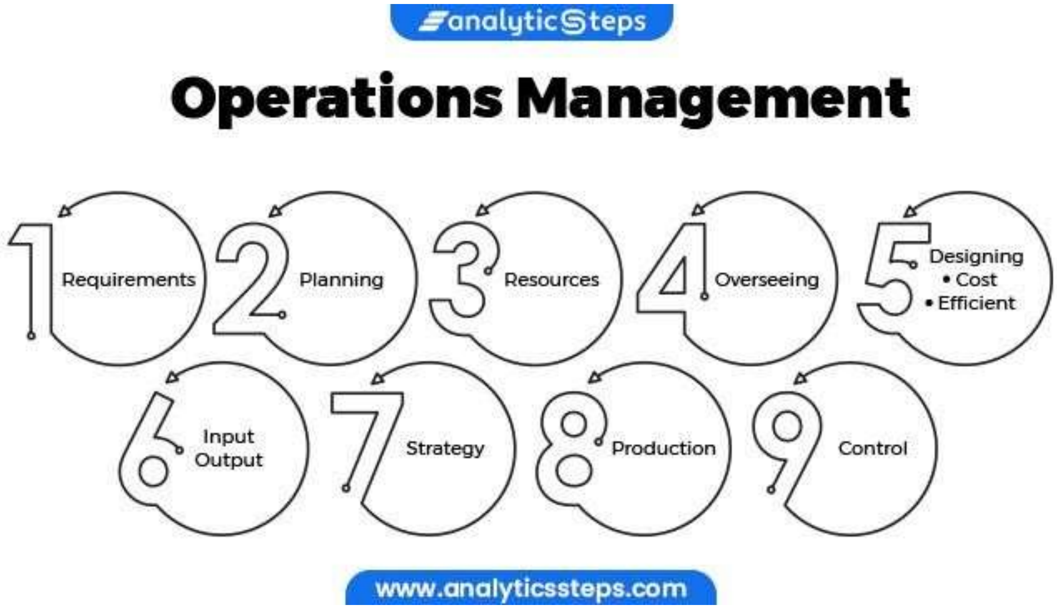
4. Operation team



IT ops responsibilities

- Monitor IT servers and systems
- Direct IT technicians
- Develop department policies
- Resolve help desk escalations
- Negotiate vendor contracts
- Oversee upgrades and installations

5. Operations Step



analyticSteps

Operations Management

1 Requirements

2 Planning

3 Resources

4 Overseeing

5 Designing
- Cost
- Efficient

6 Input Output

7 Strategy

8 Production

9 Control

www.analyticssteps.com

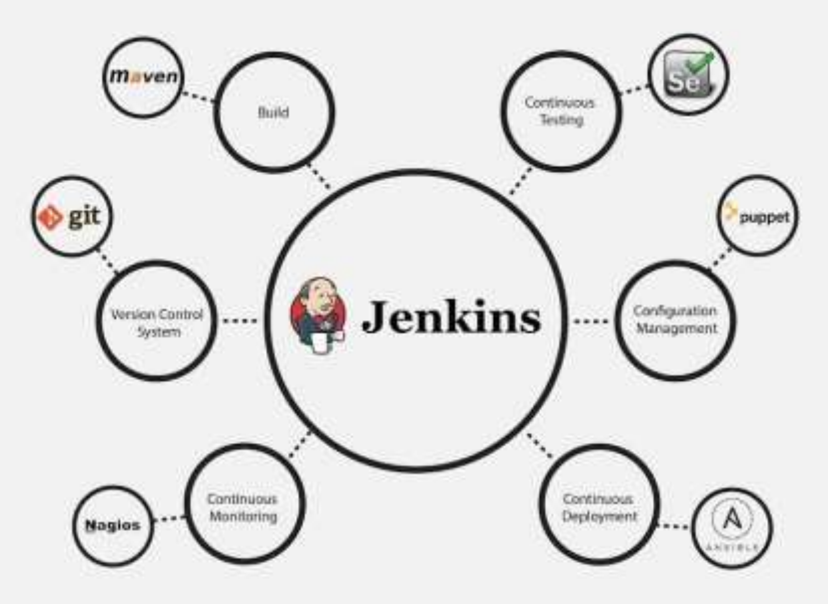
6. GitHub

GitHub
Collaboration
Bug Tracking
Branches
Git Repositories
Project Management
Team Management
Code Hosting
Track and assign tasks

7. Jenkins

Jenkins is an open source automation server. It helps automate the parts of software development related to building, testing, and deploying, facilitating continuous integration and continuous delivery.

It is a server-based system that runs in servlet containers such as Apache Tomcat.



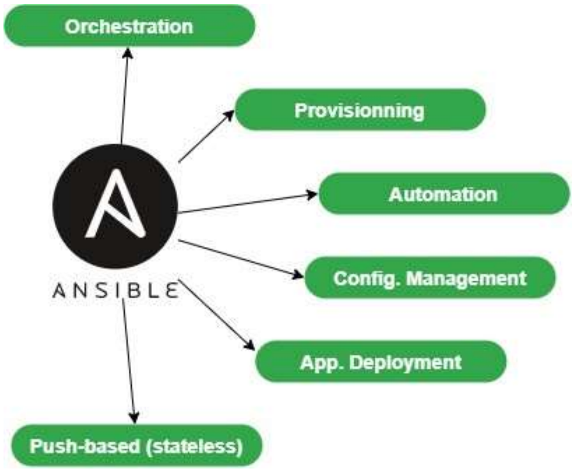
8. Docker

Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications.

Features of Docker:

- Faster and easier configuration.
- Application isolation.
- Increase in productivity.
- Security Management.

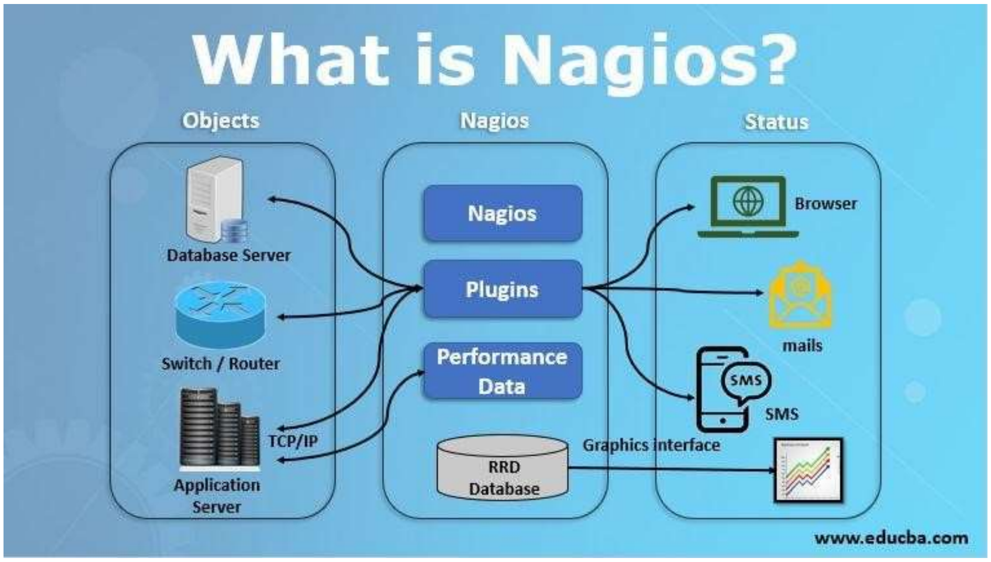
9. Ansible



Features:

- Open source
- Simple
- Versatile
- Powerful
- Agentless

10. NegiOs



Nagios is available as open-source software, and it is used to monitor computer systems. It can be executed on a Linux operating system to screen the devices which are executed on Windows, Unix, and Linux operating systems.

Conclusion: Thus, study was done and to come up with a set of question & answers that can help one to clear or understand basics understanding of DevOps.

References:

What is Nagios? | Uses,Importance and Architecture of Nagios

Ansible tutorial | what is ansible? | advantages of ansible - tutorialsinhand.com

8 steps to developing an Ansible role in Linux | Enable Sysadmin..

What is Jenkins? | Jenkins For Continuous Integration | Edureka

What is GitHub - javatpoint

EXPERIMENT B

Case Study - “Jenkins” Date of

Experiment: 25/07/2022

Aim :

To identify and analyze the latest open source DevOps tools in the market

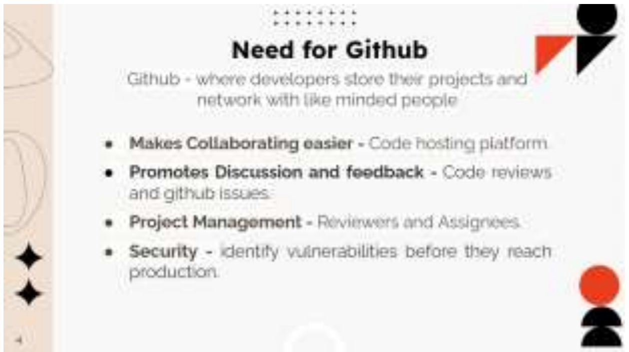
Procedure :

A group of 3 members will perform case study over internet resources with the help of research papers by answering the questions What, Why, Where and How.

Our team members are :

Swasti Jain
Om Jaanu
Tushar Padhy
Nishita PArija

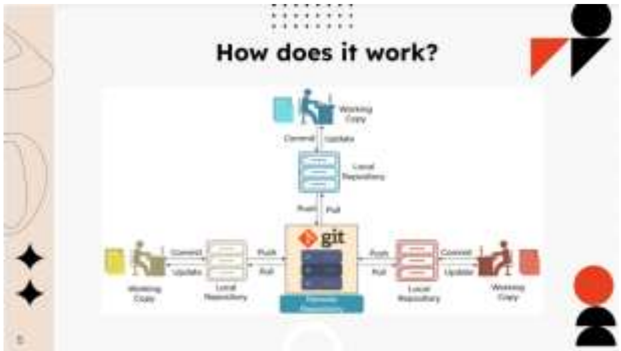
Presentation:



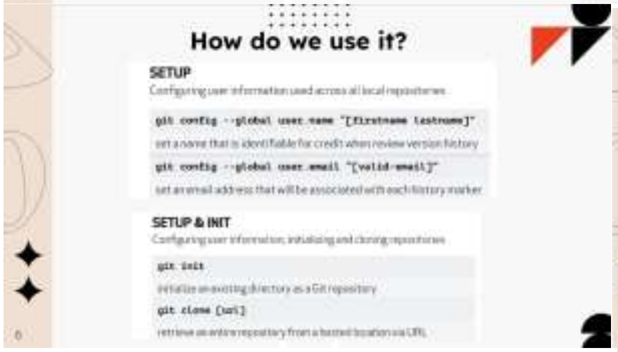
Need for Github

Github - where developers store their projects and network with like minded people

- **Makes Collaborating easier** - Code hosting platform
- **Promotes Discussion and feedback** - Code reviews and github issues
- **Project Management** - Reviewers and Assignees
- **Security** - identify vulnerabilities before they reach production



How does it work?



How do we use it?

SETUP

Configuring user information used across all local repositories

```
git config --global user.name "[firstname lastname]"
```

set a name that is identifiable for credit when review version history

```
git config --global user.email "[valid-email]"
```

set an email address that will be associated with each history marker

SETUP & INIT

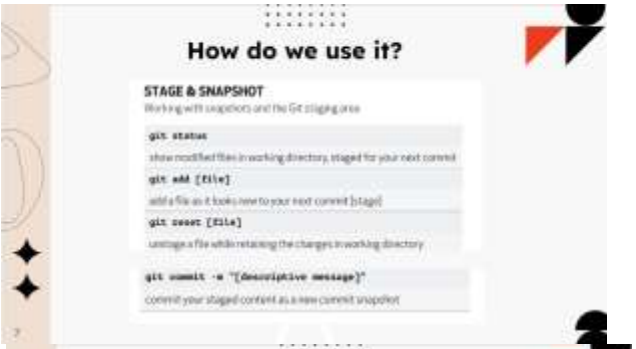
Configuring user information, initializing and cloning repositories

```
git init
```

initialize an existing directory as a Git repository

```
git clone [url]
```

retrieve an entire repository from a hosted location via URL



How do we use it?

STAGE & SNAPSHOT

Working with snapshots and the Git staging area

```
git status
```

show modified files in working directory, staged for your next commit

```
git add [file]
```

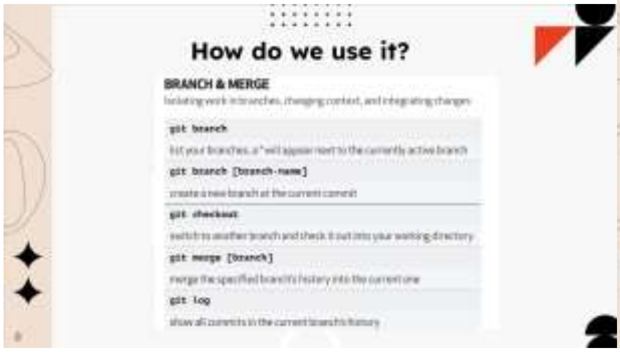
add a file as it looks now to your next commit (stage)

```
git reset [file]
```

unstage a file while retaining the changes in working directory

```
git commit -m "[descriptive message]"
```

commit your staged content as a new commit snapshot



How do we use it?

BRANCH & MERGE

Isolating work in branches, changing context, and integrating changes

```
git branch
```

list your branches. a * will appear next to the currently active branch

```
git branch [branch-name]
```

create a new branch at the current commit

```
git checkout
```

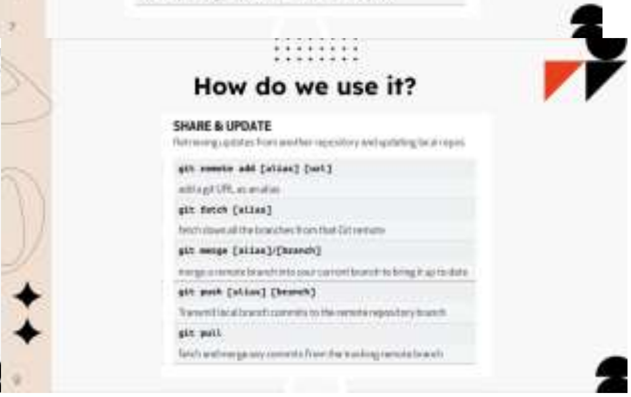
switch to another branch and check it out into your working directory

```
git merge [branch]
```

merge the specified branch's history into the current one

```
git log
```

show all commits in the current branch's history



How do we use it?

SHARE & UPDATE

Retrieving updates from another repository and updating local repos

```
git remote add [alias] [url]
```

add a git URL as an alias

```
git fetch [alias]
```

fetch down all the branches from that Git remote

```
git merge [alias]/[branch]
```

merge a remote branch into your current branch to bring it up to date

```
git push [alias] [branch]
```

Transmit local branch commits to the remote repository branch

```
git pull
```

fetch and merge any commits from the tracking remote branch

Conclusion :

Presentation was done on the respective topic and a document was prepared **References:**
https://gursimar27.medium.com/jenkins-case-study-f99f683624c5
https://www.researchgate.net/publication/311153114_Jenkins-CI_an_Open-Source_Continuous_Integration_System_as_a_Scientific_Data_and_ImageProcessing_Platform
https://medium.com/@prithvilee22/jenkins-case-study-3295ceddf69

EXPERIMENT 1

Version Control Using Git Date of

Experiment: 01/08/2022

Objective:
Is to experience version controlling using GITHUB by answering basic questions like what is Git, GitHub, Clone Repository, Forking and branching.

Aim:
Is to use, analyze and experience all version control commands of GIT tool and GitHub Service.

Procedure / Steps to perform the Experiment:

1. Download and install Git tool
2. Create / Use GitHub Account with some directories/Repositories
3. Perform all the below mentioned commands of Git Local Repository and GitHub Service to reflect upon the version control.
    a. Create and fork repositories in GitHub
    b. Apply branching, merging and rebasing concepts.
    c. Implement different Git workflow strategies in real-time scenarios

Git is a free and open-source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. Git is easy to learn and has a tiny footprint with lightning-fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

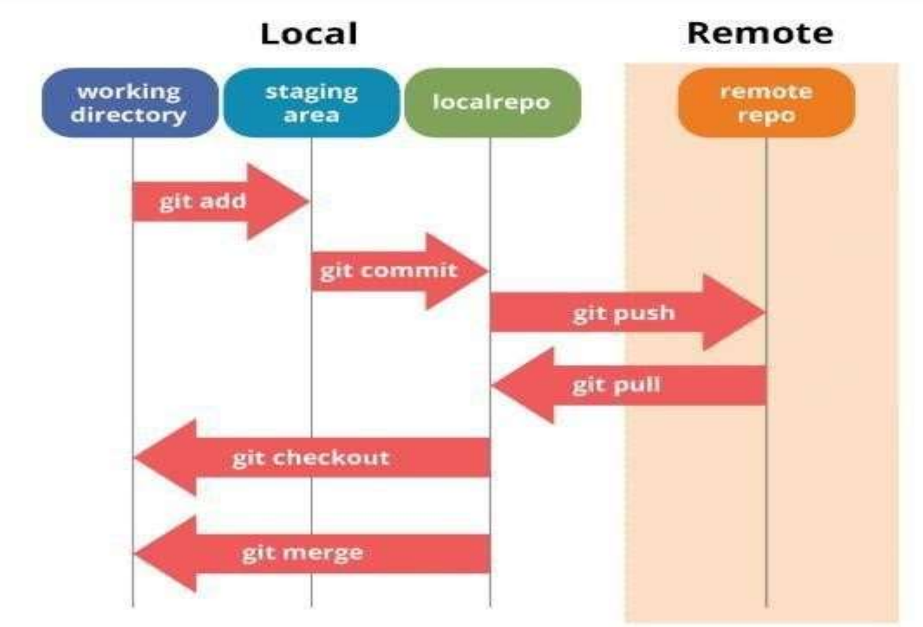
Some of the basic operations in Git are:

1. Initialize
2. Add
3. Commit
4. Pull
5. Push

Some advanced Git operations are:

1. Branching
2. Merging
3. Rebasing

The following diagram depict the all-supported operations in GIT



Commands:

a. Basic Commands:

sudo apt-get install git git –
version git init

//configuration details

git config --global user.name "sunanthag1998" git config --global
user.email "sunanthag1998@gmail.com" git config –list

b. Staging Commands

git add . git commit -m "v1 of file
1"

c. Push and pull the repository from and to GitHub

- Go to GitHub ->User accounts-> setting -> developers portal ->generate token
- And choose the duration for its validity to exchange the repo between Git & GitHub and use the token in the below command to set the authentication for push and pull.

git remote add origin https://github.com/sunanthag1998/test4.git //directing
to the repository git clone "https://github.com/sunanthag1998/test4.git // cloning remote
repository to local git remote
set-url origin
http://sunanthag1998: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@gith
ub.com/sunanthag1998/test4.git
(Syntax: git remote set-url origin
https://userid:password@github.com/user/repo.git)

git add –all
git commit -m “v1” //Try to commit and then push or pull
git branch -u origin main // branch a copy from main to reflect changes on the
branch git push -u origin main // push from git to github git pull
-u origin main //pull form github to git

d: Fork , Branch & Merge Commands *next lab*

If you have existing repository, then simply delete .git file and reinitialize it. *$ rm -rf .git/*

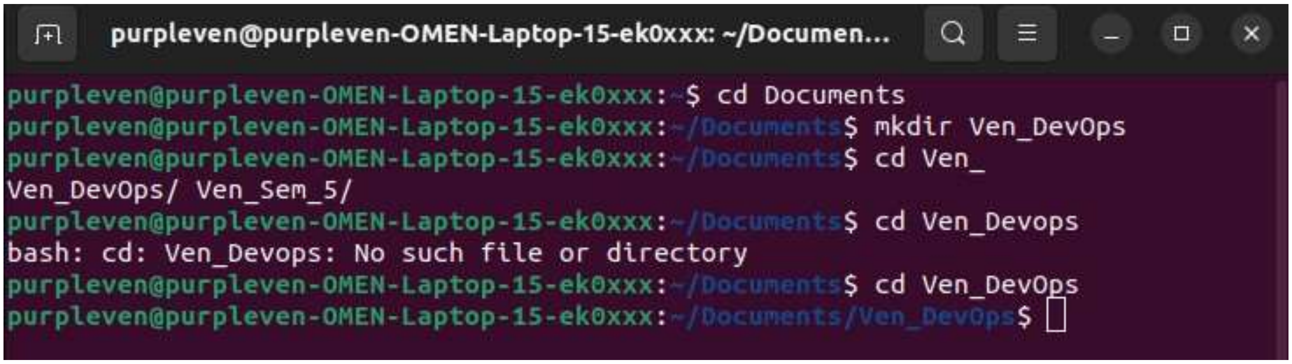
Output:

Installing git with `sudo apt install git` Checking if git is

installed with `git –version`

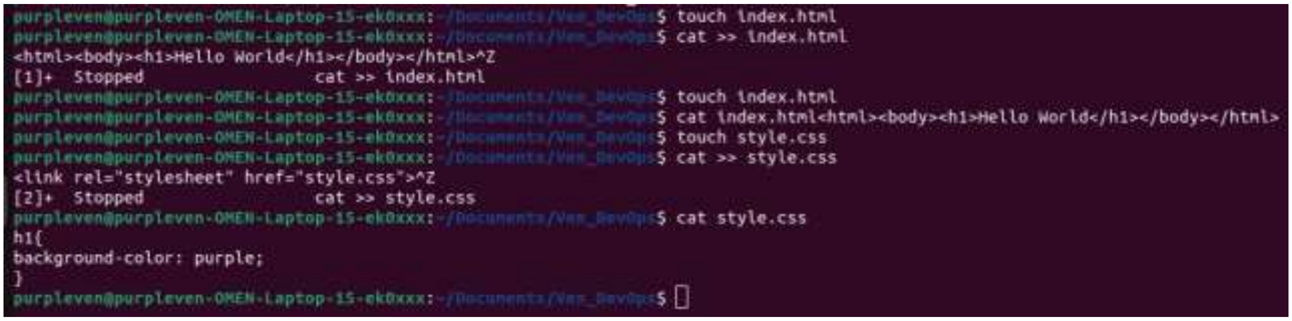
Now with the help of git:

1) Creating a directory

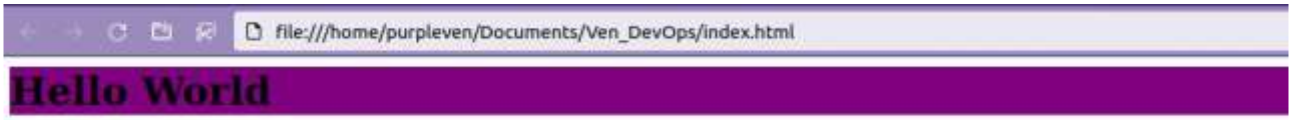


```
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~$ cd Documents
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents$ mkdir Ven_DevOps
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents$ cd Ven_
Ven_DevOps/ Ven_Sem_5/
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents$ cd Ven_Devops
bash: cd: Ven_Devops: No such file or directory
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents$ cd Ven_DevOps
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ 
```

2) Creating two files with the help of the touch command



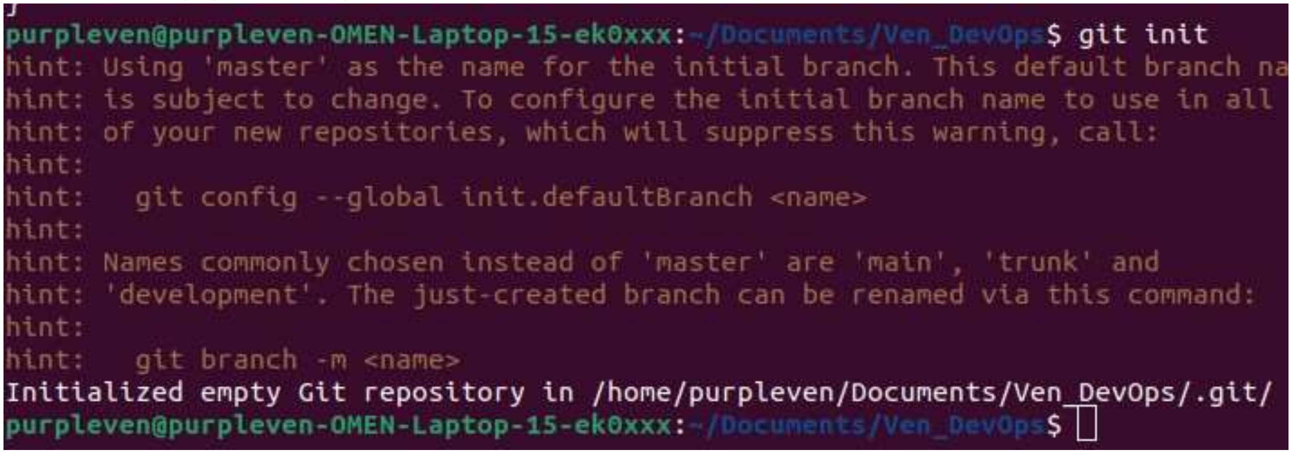
```
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ touch index.html
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ cat >> index.html
<html><body><h1>Hello World</h1></body></html>^Z
[1]+  Stopped                 cat >> index.html
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ touch index.html
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ cat index.html<html><body><h1>Hello World</h1></body></html>
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ touch style.css
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ cat >> style.css
<link rel="stylesheet" href="style.css">^Z
[2]+  Stopped                 cat >> style.css
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ cat style.css
h1{
background-color: purple;
}
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ 
```



file:///home/purpleven/Documents/Ven_DevOps/index.html

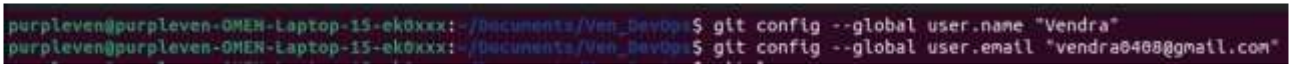
Hello World

3) Initializing git with `git init`



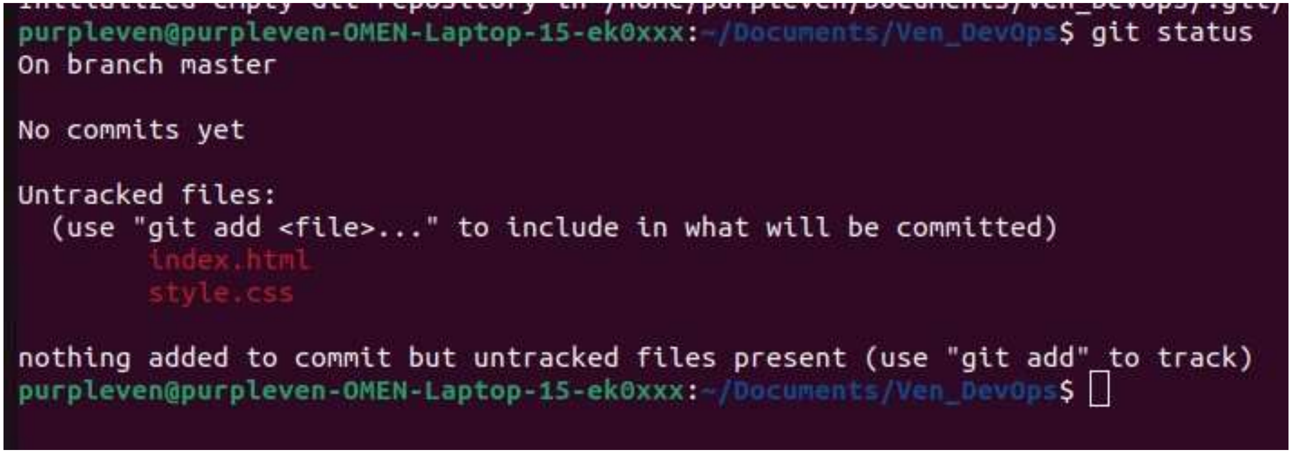
```
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ git init
hint: Using 'master' as the name for the initial branch. This default branch na
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/purpleven/Documents/Ven_DevOps/.git/
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$
```

4) Now config the git with your username and email



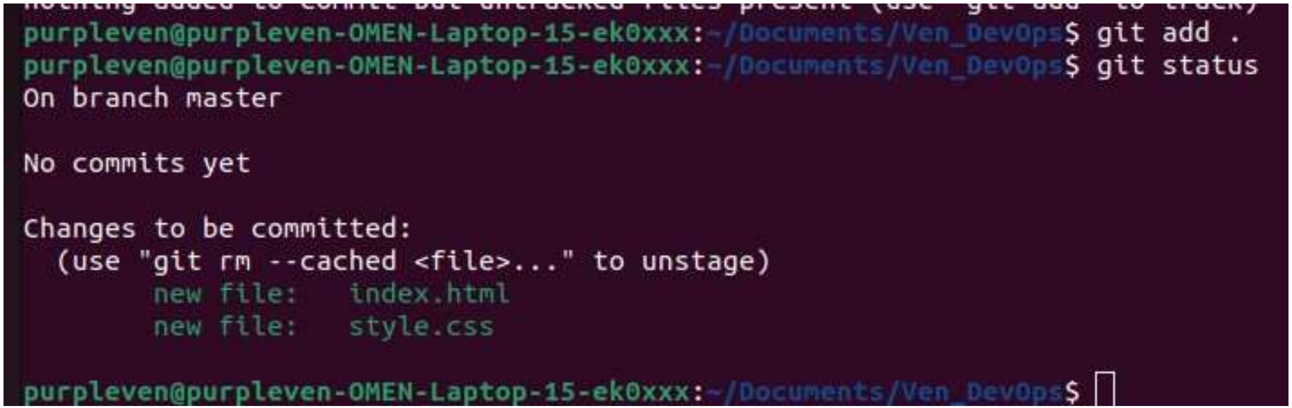
```
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ git config --global user.name "Vendra"
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ git config --global user.email "vendra0408@gmail.com"
```

5) Checking the status of the files with the help of `git status`

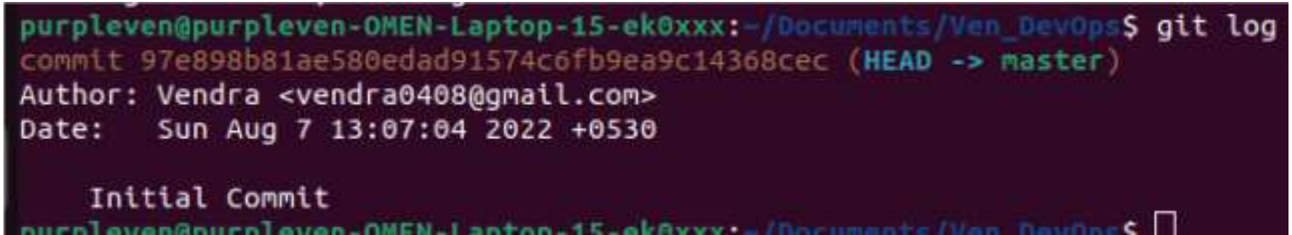


```
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html
        style.css

nothing added to commit but untracked files present (use "git add" to track)
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$
```

6) Adding the untracked files with `git add .`

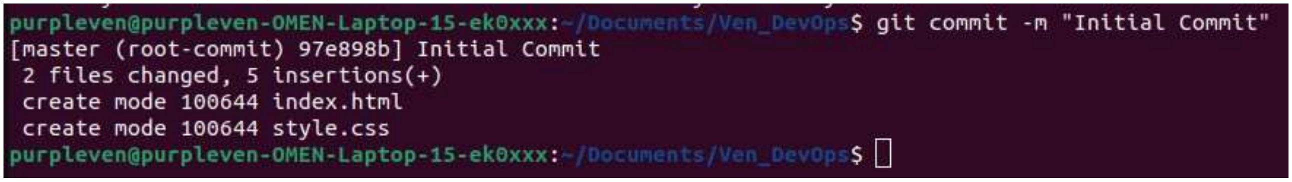


```
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ git add .
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   index.html
        new file:   style.css

purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$
```



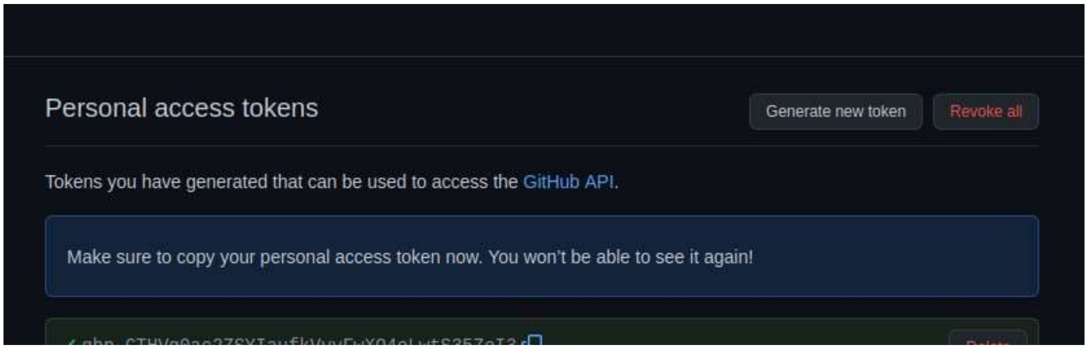
```
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ git log
commit 97e898b81ae580edad91574c6fb9ea9c14368cec (HEAD -> master)
Author: Vendra <vendra0408@gmail.com>
Date:   Sun Aug 7 13:07:04 2022 +0530

    Initial Commit
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$
```

7) Now commiting the added files with a message



```
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ git commit -m "Initial Commit"
[master (root-commit) 97e898b] Initial Commit
 2 files changed, 5 insertions(+)
 create mode 100644 index.html
 create mode 100644 style.css
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ 
```

8) Checking with the help of log if the changes are committed Now with the help of GitHub:
Go to GitHub ->User accounts-> setting -> developers portal ->generate token

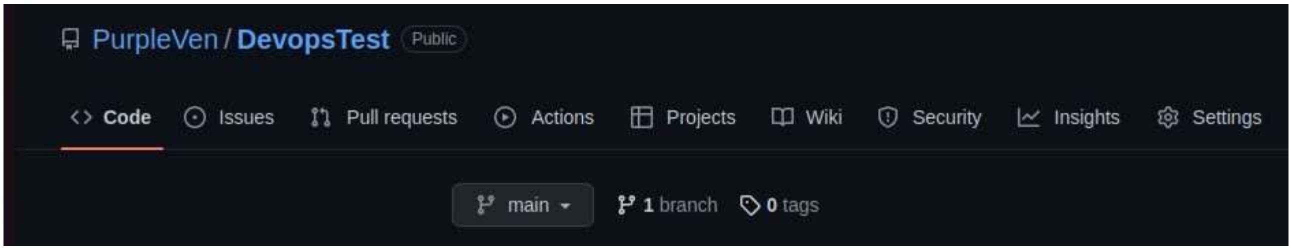


Personal access tokens

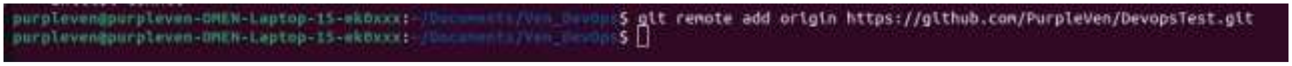
Tokens you have generated that can be used to access the GitHub API.

Make sure to copy your personal access token now. You won't be able to see it again!

1) Create a repository

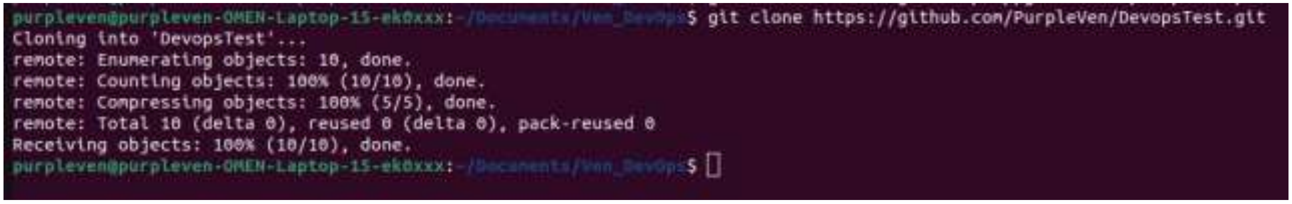


2) Adding remote origin



```
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ git remote add origin https://github.com/PurpleVen/DevopsTest.git
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ 
```

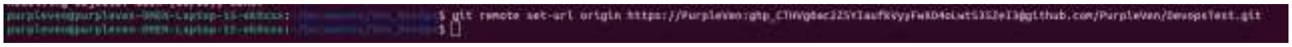
3) Cloning the repository



```
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ git clone https://github.com/PurpleVen/DevopsTest.git
Cloning into 'DevopsTest'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 10 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (10/10), done.
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ 
```

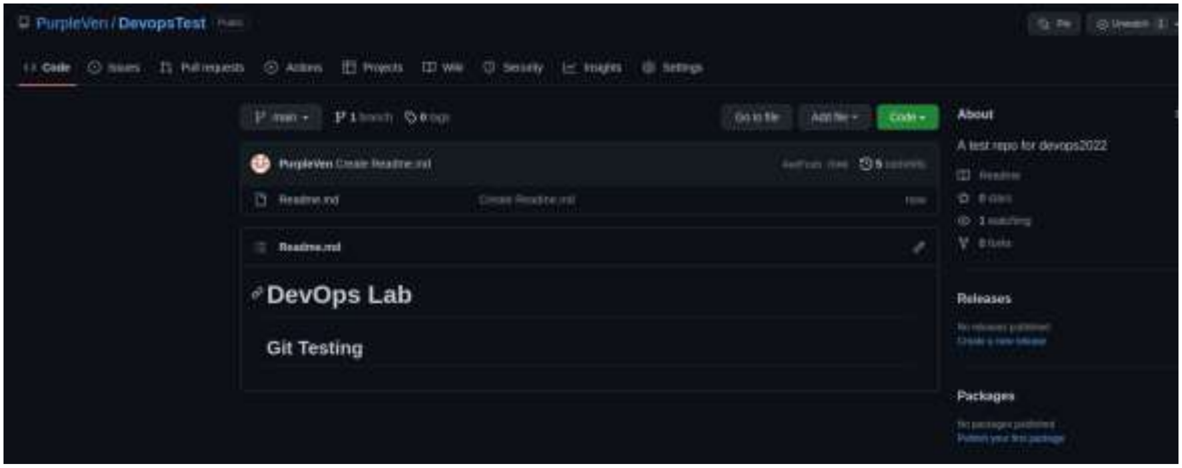
4) git remote set url

(Syntax: git remote set-url origin https://userid:password@github.com/user/repo.git)



```
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ git remote set-url origin https://PurpleVen:ghp_CTHVg6ac22SYIaufKvyyFwKD4oLwtS3S2eI3@github.com/PurpleVen/DevopsTest.git
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ 
```

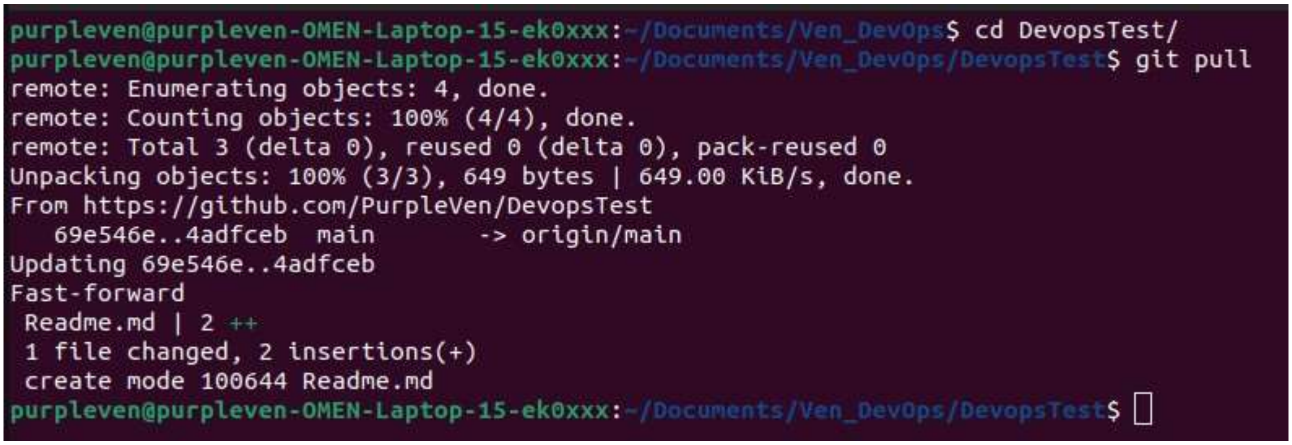
5) Added a file in github



DevOps Lab

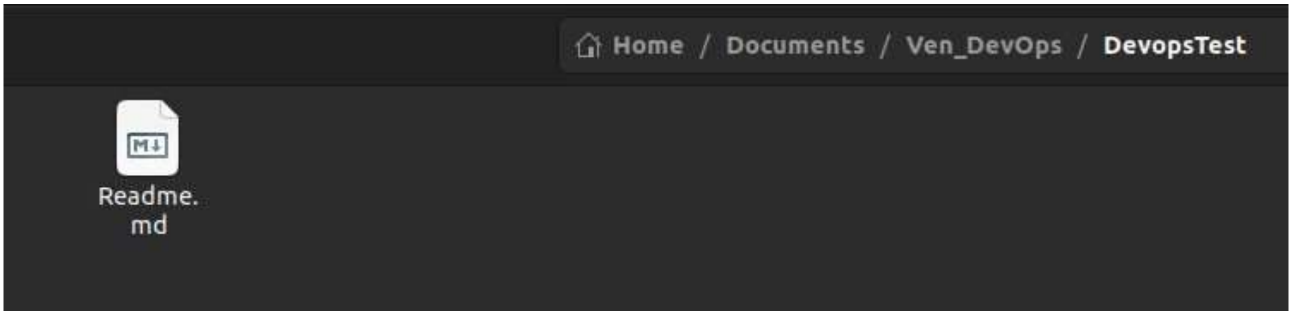
Git Testing

6) Trying the pull command



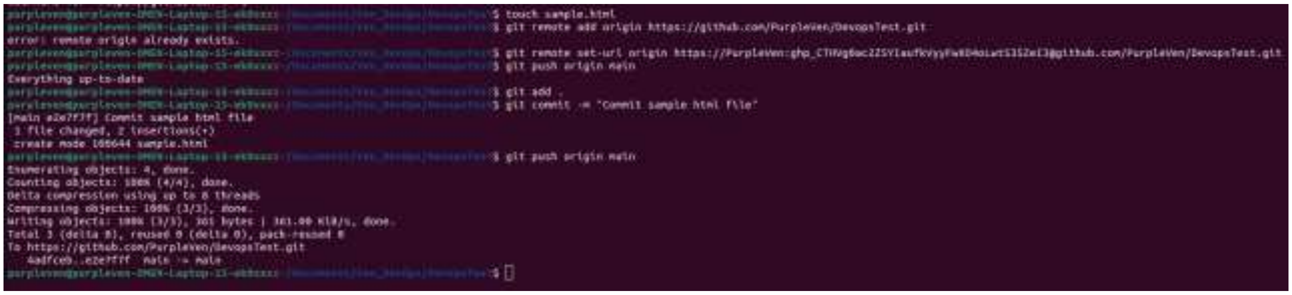
```
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ cd DevopsTest/
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps/DevopsTest$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 649 bytes | 649.00 KiB/s, done.
From https://github.com/PurpleVen/DevopsTest
   69e546e..4adfceb  main       -> origin/main
Updating 69e546e..4adfceb
Fast-forward
 Readme.md | 2 ++
 1 file changed, 2 insertions(+)
 create mode 100644 Readme.md
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps/DevopsTest$ 
```

7) The readme file is visible



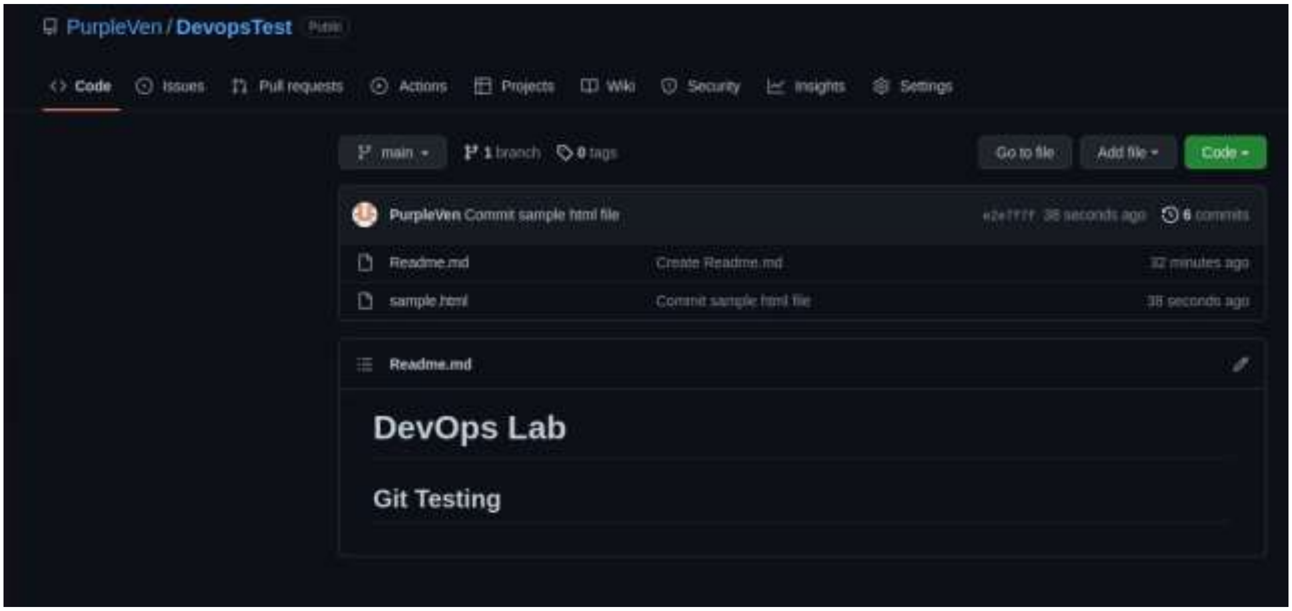
8)

Pushing the files from local to GitHub



```
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps/DevopsTest$ touch sample.html
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps/DevopsTest$ git remote add origin https://github.com/PurpleVen/DevopsTest.git
error: remote origin already exists.
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps/DevopsTest$ git remote set-url origin https://PurpleVen:ghp_CTHVg6oc2ZSYIaufKVyyFwKD4oLwtS1S2eI3@github.com/PurpleVen/DevopsTest.git
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps/DevopsTest$ git push origin main
Everything up-to-date
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps/DevopsTest$ git add .
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps/DevopsTest$ git commit -m "Commit sample html file"
[main e2e7f7f] Commit sample html file
 1 file changed, 2 insertions(+)
 create mode 100644 sample.html
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps/DevopsTest$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 301 bytes | 301.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/PurpleVen/DevopsTest.git
   4adfceb..e2e7fff  main -> main
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps/DevopsTest$ 
```

9) Displayed in GitHub



PurpleVen Commit sample html file e2e7f7f 38 seconds ago 6 commits

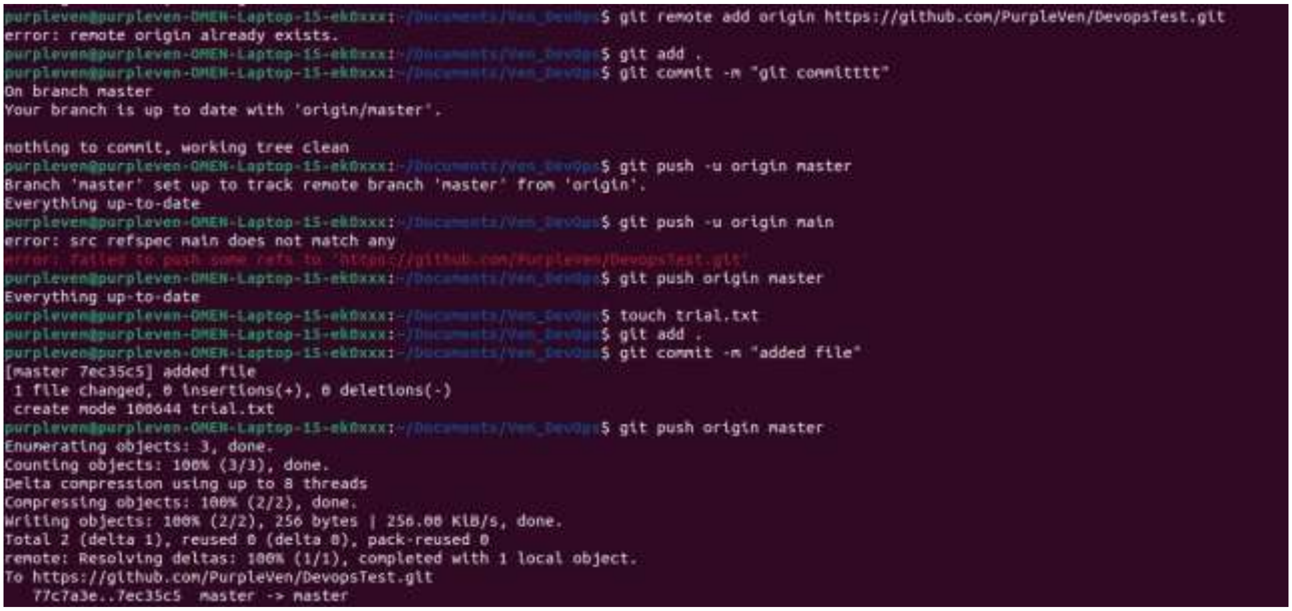
Readme.md	Create Readme.md	32 minutes ago
sample.html	Commit sample html file	38 seconds ago

Readme.md

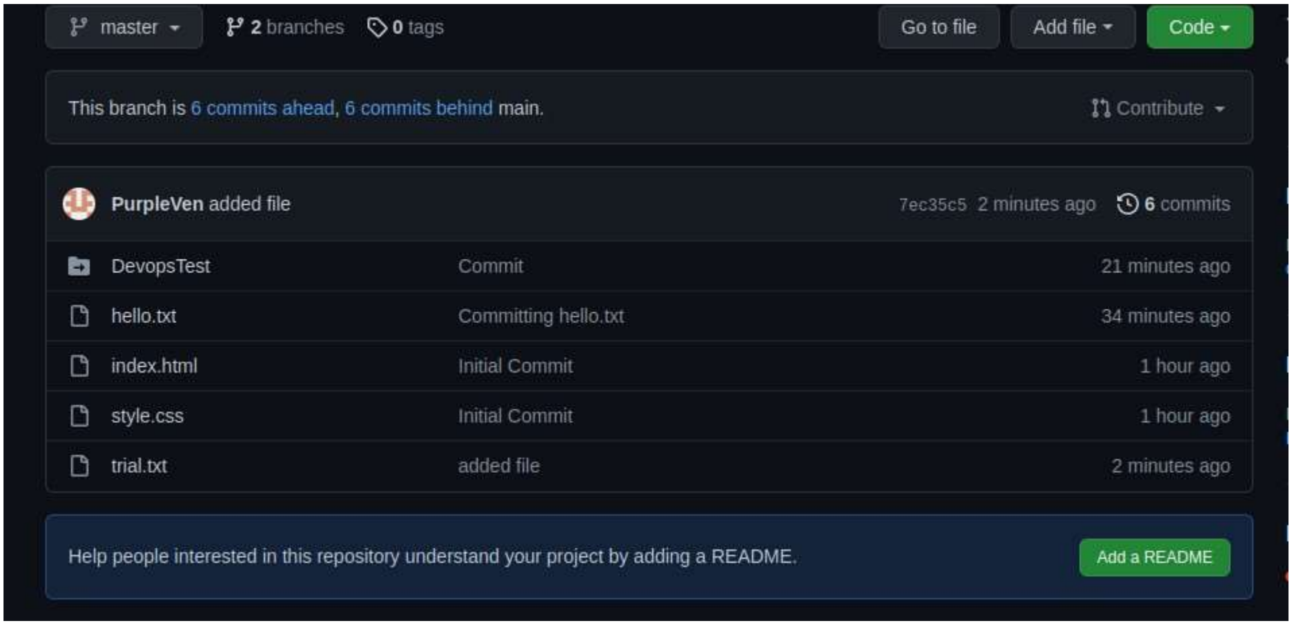
DevOps Lab

Git Testing

New branch master is created



```
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ git remote add origin https://github.com/PurpleVen/DevopsTest.git
error: remote origin already exists.
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ git add .
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ git commit -m "git committtt"
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ git push -u origin master
Branch 'master' set up to track remote branch 'master' from 'origin'.
Everything up-to-date
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ git push -u origin main
error: src refspec main does not match any
error: failed to push some refs to 'https://github.com/PurpleVen/DevopsTest.git'
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ git push origin master
Everything up-to-date
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ touch trial.txt
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ git add .
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ git commit -m "added file"
[master 7ec35c5] added file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 trial.txt
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 256 bytes | 256.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/PurpleVen/DevopsTest.git
   77c7a3e..7ec35c5  master -> master
```



master ▾ | 2 branches | 0 tags | Go to file | Add file ▾ | Code ▾

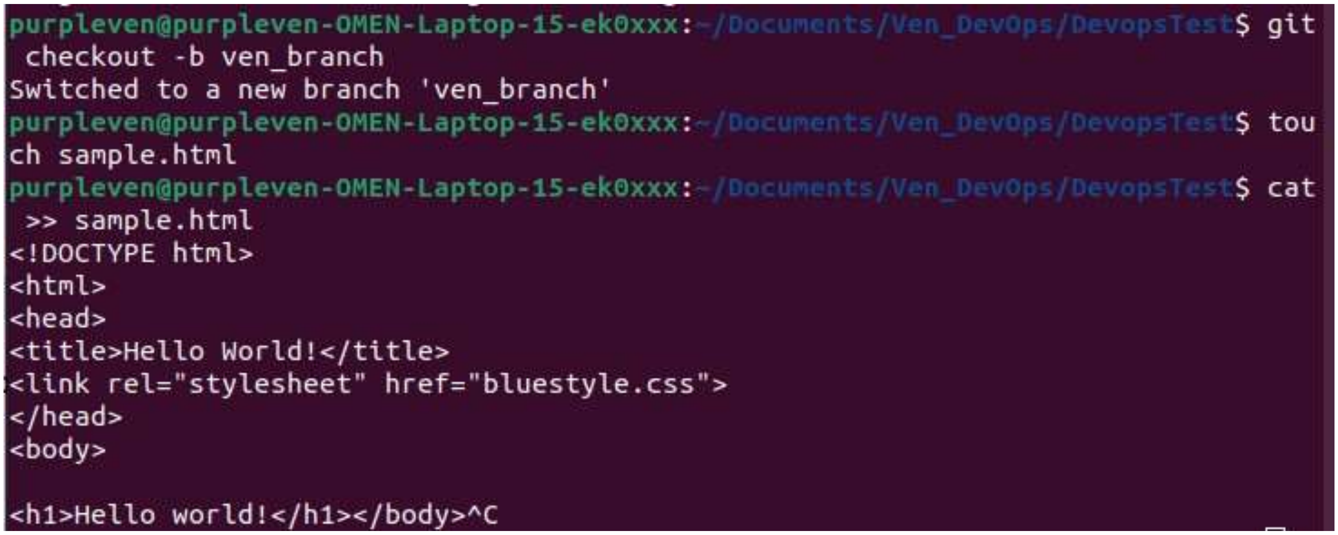
This branch is 6 commits ahead, 6 commits behind main. | Contribute ▾

PurpleVen added file | 7ec35c5 2 minutes ago | 6 commits

DevopsTest	Commit	21 minutes ago
hello.txt	Committing hello.txt	34 minutes ago
index.html	Initial Commit	1 hour ago
style.css	Initial Commit	1 hour ago
trial.txt	added file	2 minutes ago

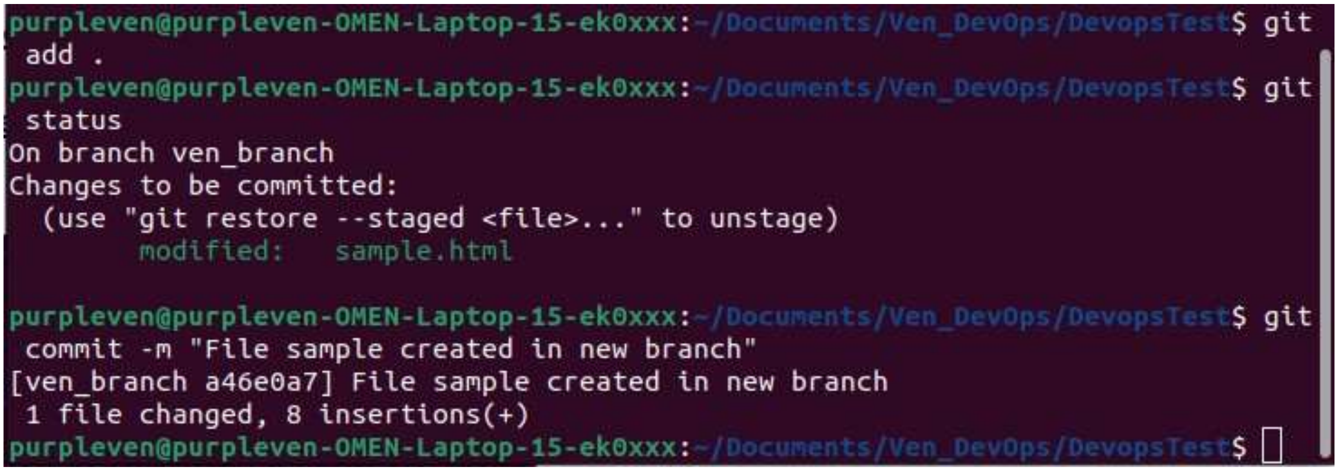
Help people interested in this repository understand your project by adding a README. | Add a README

10) **Making a new branch and adding the same file as that of main to test the gitmerge command**



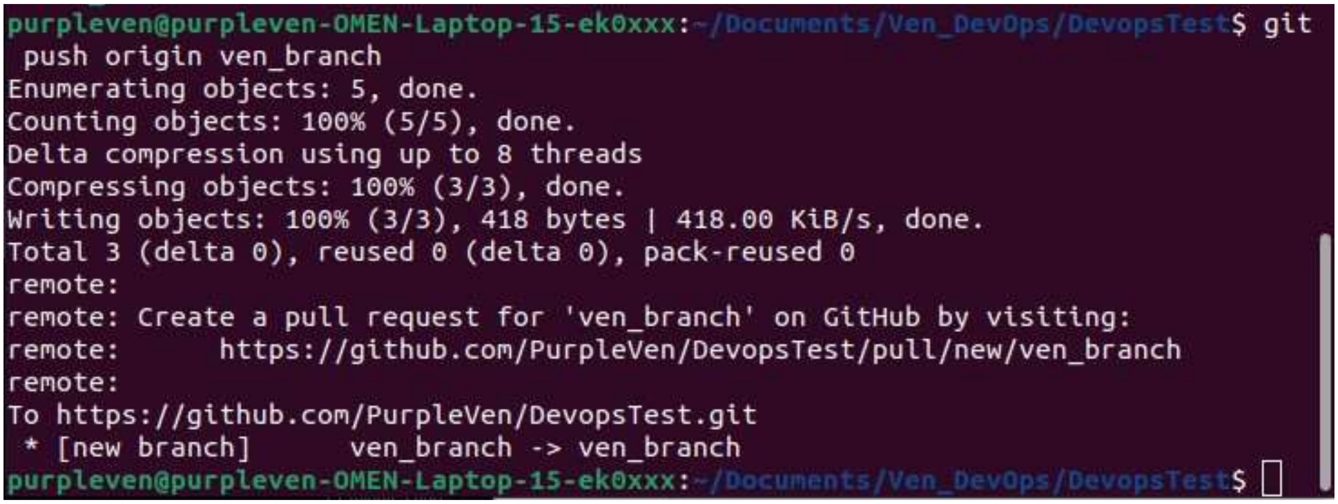
```
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps/DevopsTest$ git
 checkout -b ven_branch
Switched to a new branch 'ven_branch'
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps/DevopsTest$ tou
ch sample.html
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps/DevopsTest$ cat
 >> sample.html
<!DOCTYPE html>
<html>
<head>
<title>Hello World!</title>
<link rel="stylesheet" href="bluestyle.css">
</head>
<body>

<h1>Hello world!</h1></body>^C
```

11) **Committing the file in the new branch**



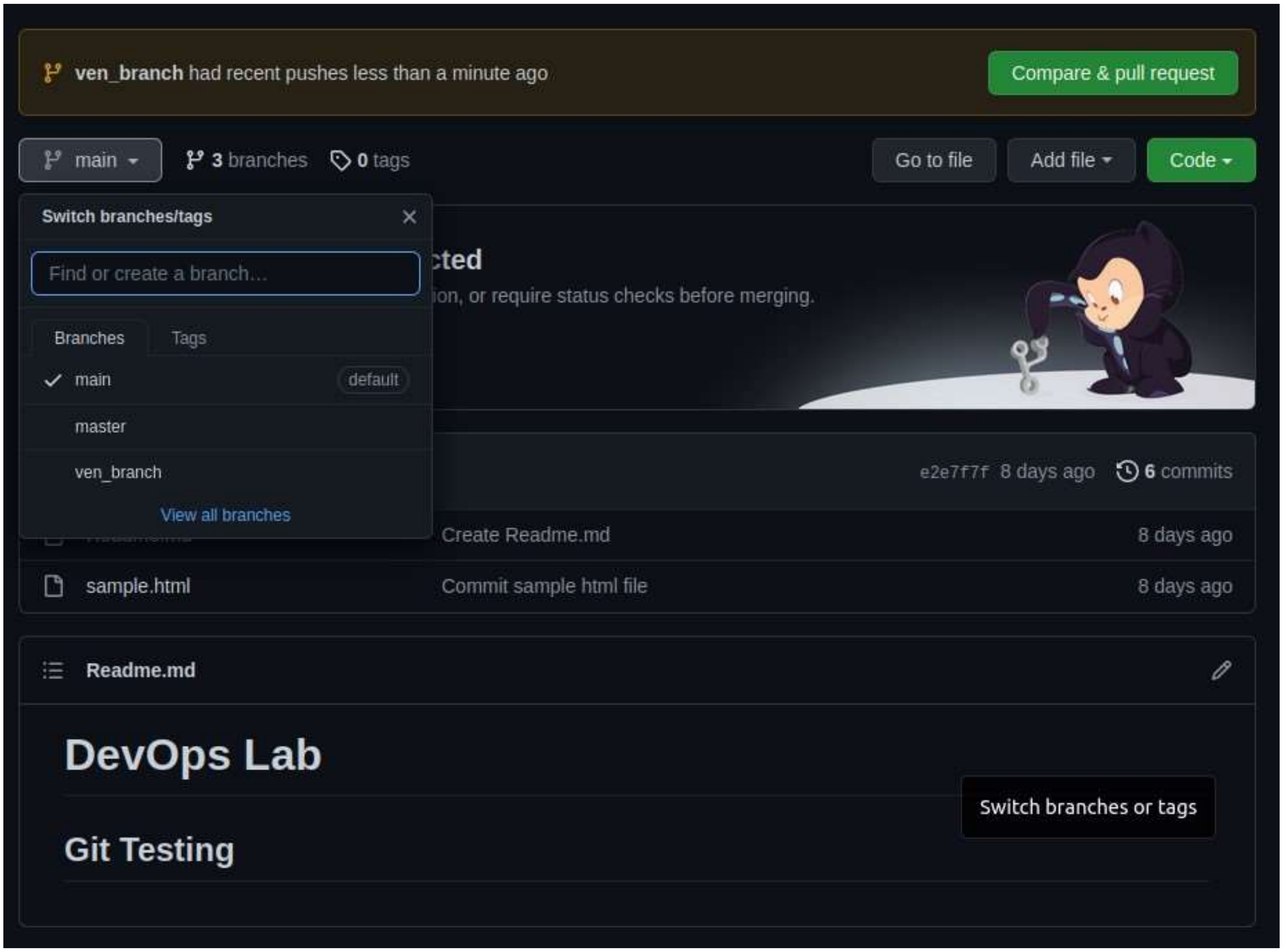
```
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps/DevopsTest$ git
 add .
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps/DevopsTest$ git
 status
On branch ven_branch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   sample.html

purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps/DevopsTest$ git
 commit -m "File sample created in new branch"
[ven_branch a46e0a7] File sample created in new branch
 1 file changed, 8 insertions(+)
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps/DevopsTest$ 
```

12) Git push the branch



```
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps/DevopsTest$ git
 push origin ven_branch
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), 418 bytes | 418.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'ven_branch' on GitHub by visiting:
remote:      https://github.com/PurpleVen/DevopsTest/pull/new/ven_branch
remote:
To https://github.com/PurpleVen/DevopsTest.git
 * [new branch]      ven_branch -> ven_branch
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps/DevopsTest$ 
```

In GitHub the following changes and a new branch is visible



ven_branch had recent pushes less than a minute ago

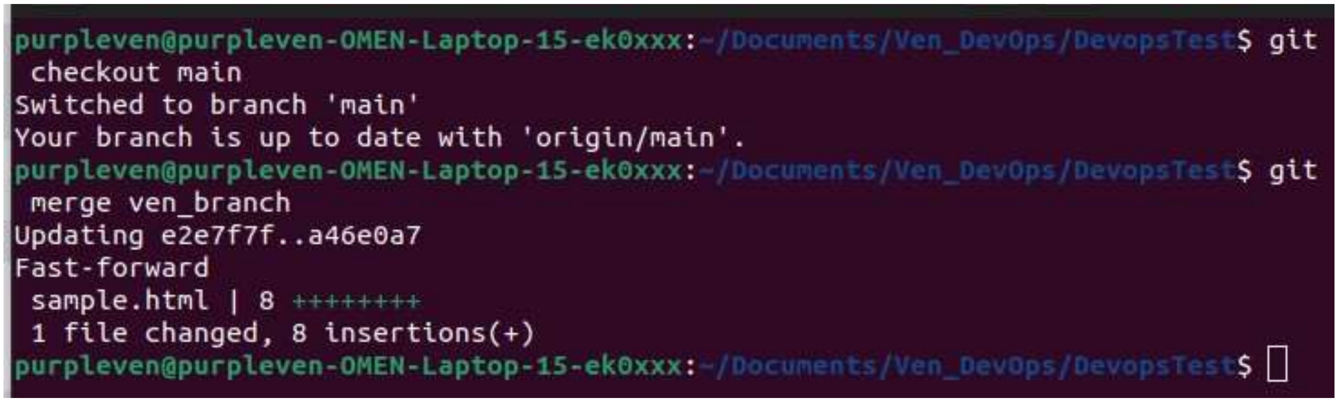
	Create Readme.md	8 days ago
sample.html	Commit sample html file	8 days ago

Readme.md

DevOps Lab

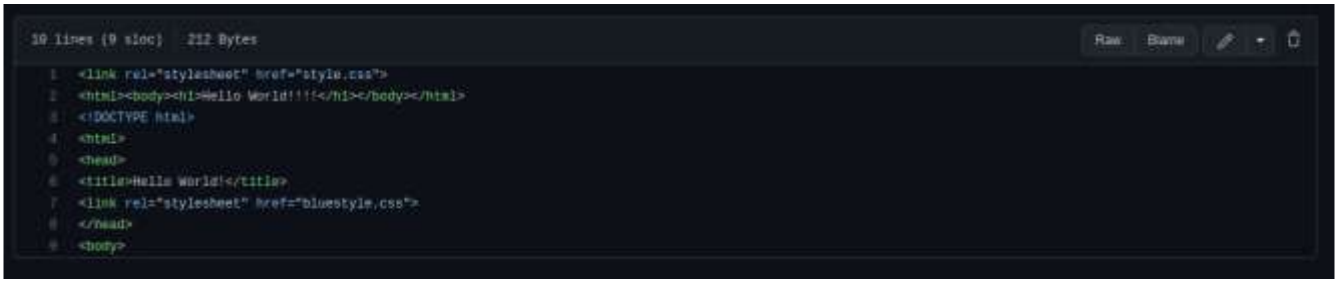
Git Testing

13) Git checkout to switch branches and git merge in the new branch with themain branch



```
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps/DevopsTest$ git
 checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps/DevopsTest$ git
 merge ven_branch
Updating e2e7f7f..a46e0a7
Fast-forward
 sample.html | 8 ++++++++
 1 file changed, 8 insertions(+)
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~/Documents/Ven_DevOps/DevopsTest$ 
```

In the branch the files are merged



10 lines (9 sloc) 212 Bytes

```
<link rel="stylesheet" href="style.css">
<html><body><h1>Hello World!!!!</h1></body></html>
<!DOCTYPE html>
<html>
<head>
<title>Hello World!</title>
<link rel="stylesheet" href="bluestyle.css">
</head>
<body>
```

Conclusion:
Thus, the basic commands to access the GitHub version control were performed successfully.

References:

1. Steps for Beginners: https://www.youtube.com/watch?v=9FOuyNt0V8I 2. For token usage as on 13 August 2022: https://www.youtube.com/watch?v=W9zTttHeoHk 3. For forking and branching and merging use https://spoken-tutorial.org/tutorial-search/?search_foss=Git&search_language=E nglish

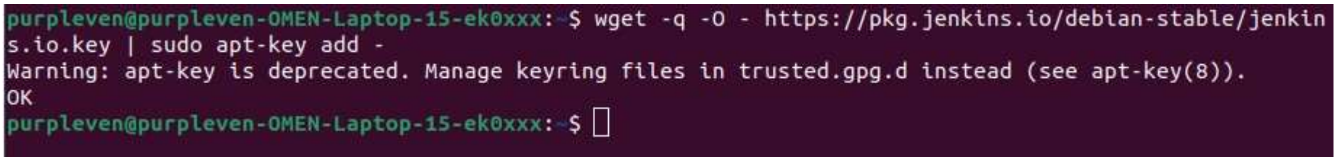
EXPERIMENT 2

Jenkins

Date of Experiment: 12/09/2022

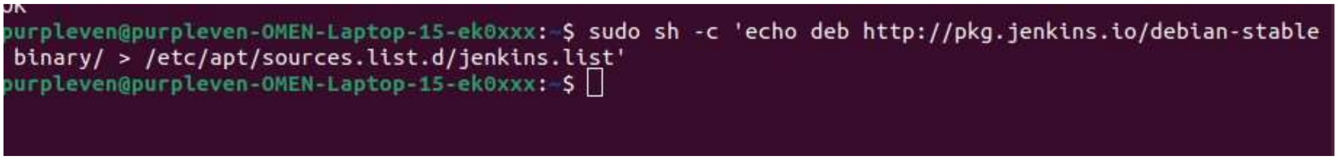
Output:

Add the repository key to the system:



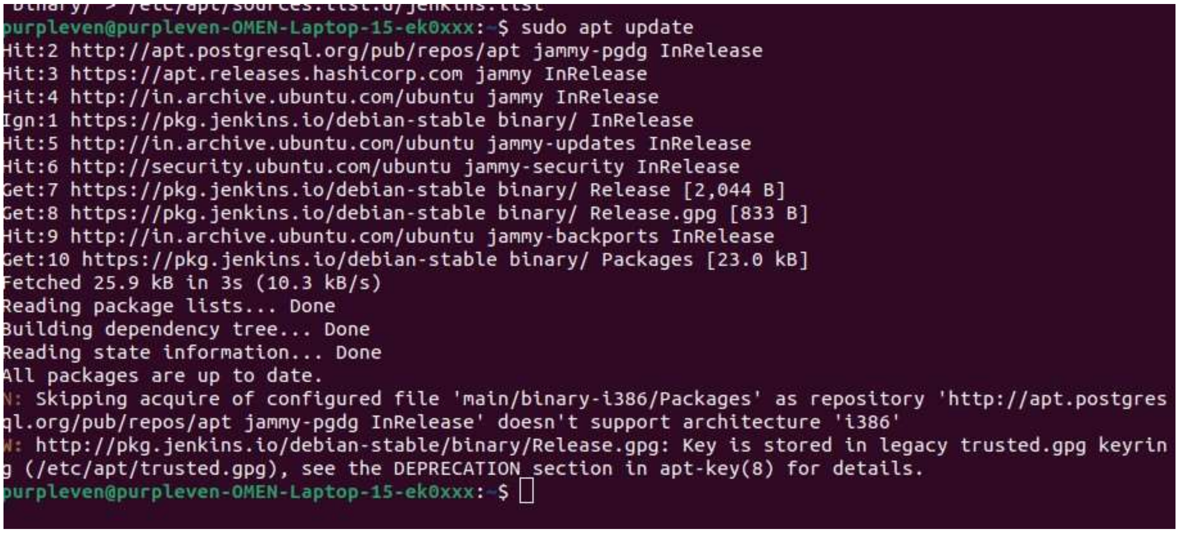
```
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~$ wget -q -O - https://pkg.jenkins.io/debian-stable/jenkin
s.io.key | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~$ 
```

Next, let's append the Debian package repository address to the server's sources.list



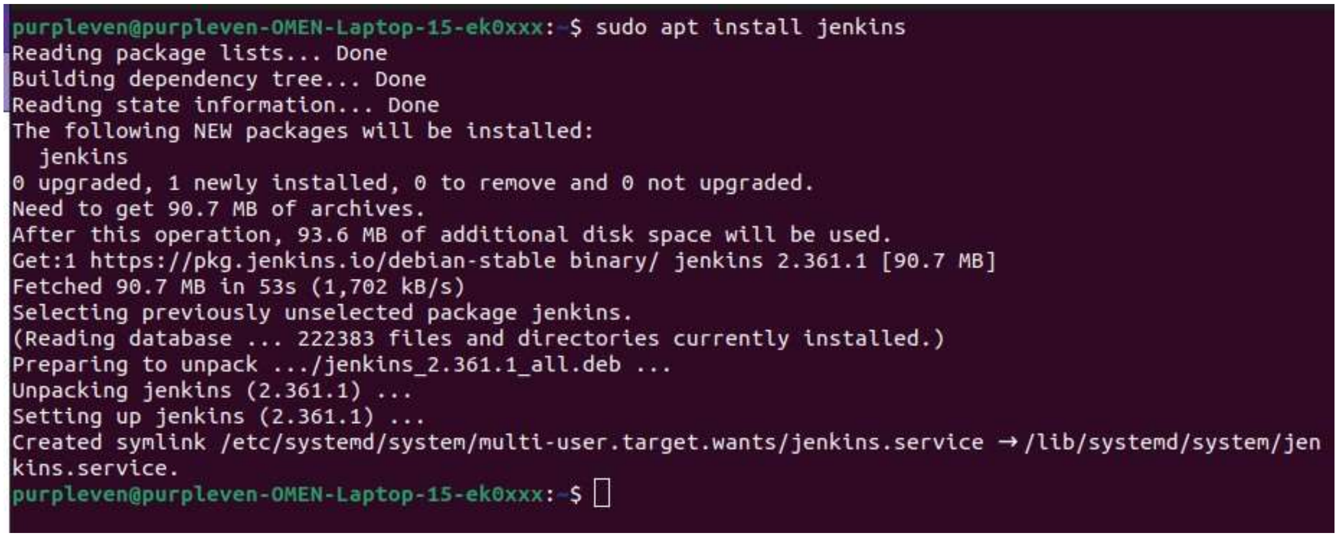
```
OK
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~$ sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable
 binary/ > /etc/apt/sources.list.d/jenkins.list'
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~$ 
```

After both commands have been entered, we'll run an update so that apt will use the new repository.



```
binary/ > /etc/apt/sources.list.d/jenkins.list'
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~$ sudo apt update
Hit:2 http://apt.postgresql.org/pub/repos/apt jammy-pgdg InRelease
Hit:3 https://apt.releases.hashicorp.com jammy InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Ign:1 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:5 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:6 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:7 https://pkg.jenkins.io/debian-stable binary/ Release [2,044 B]
Get:8 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Hit:9 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:10 https://pkg.jenkins.io/debian-stable binary/ Packages [23.0 kB]
Fetched 25.9 kB in 3s (10.3 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
N: Skipping acquire of configured file 'main/binary-i386/Packages' as repository 'http://apt.postgres
ql.org/pub/repos/apt jammy-pgdg InRelease' doesn't support architecture 'i386'
W: http://pkg.jenkins.io/debian-stable/binary/Release.gpg: Key is stored in legacy trusted.gpg keyrin
g (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~$ 
```

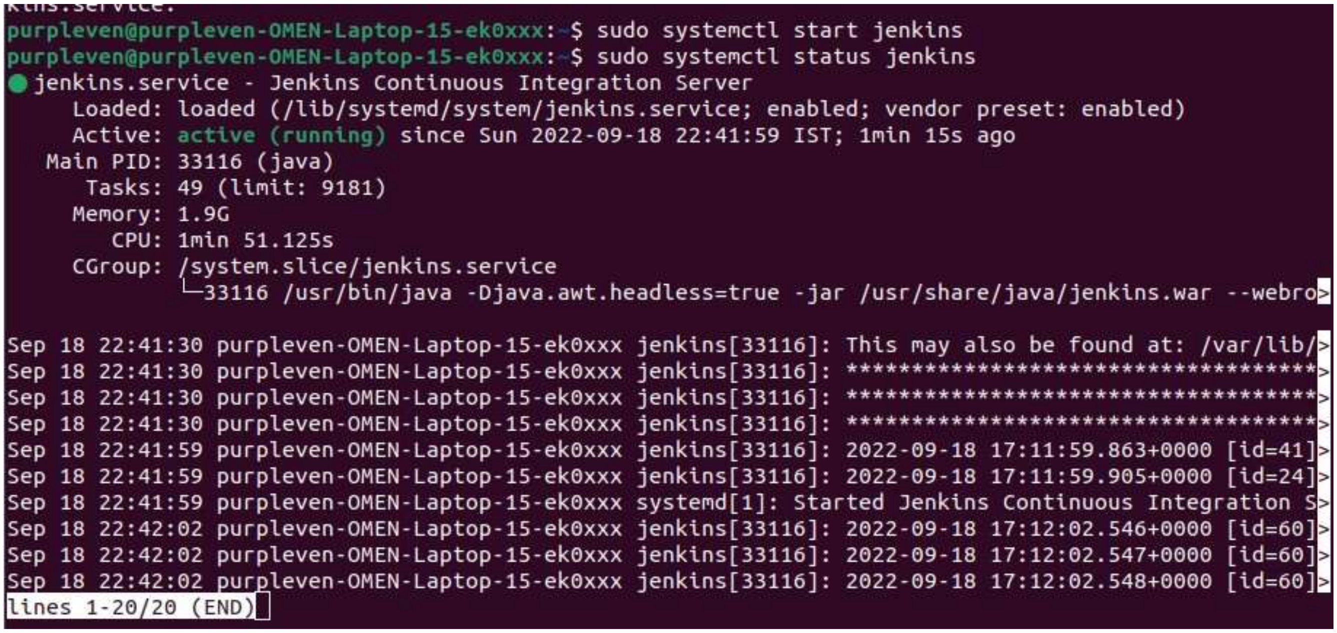
Finally, we'll install Jenkins and its dependencies.



```
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~$ sudo apt install jenkins
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  jenkins
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 90.7 MB of archives.
After this operation, 93.6 MB of additional disk space will be used.
Get:1 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.361.1 [90.7 MB]
Fetched 90.7 MB in 53s (1,702 kB/s)
Selecting previously unselected package jenkins.
(Reading database ... 222383 files and directories currently installed.)
Preparing to unpack .../jenkins_2.361.1_all.deb ...
Unpacking jenkins (2.361.1) ...
Setting up jenkins (2.361.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /lib/systemd/system/jen
kins.service.
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~$ 
```

Step 2 — Starting Jenkins

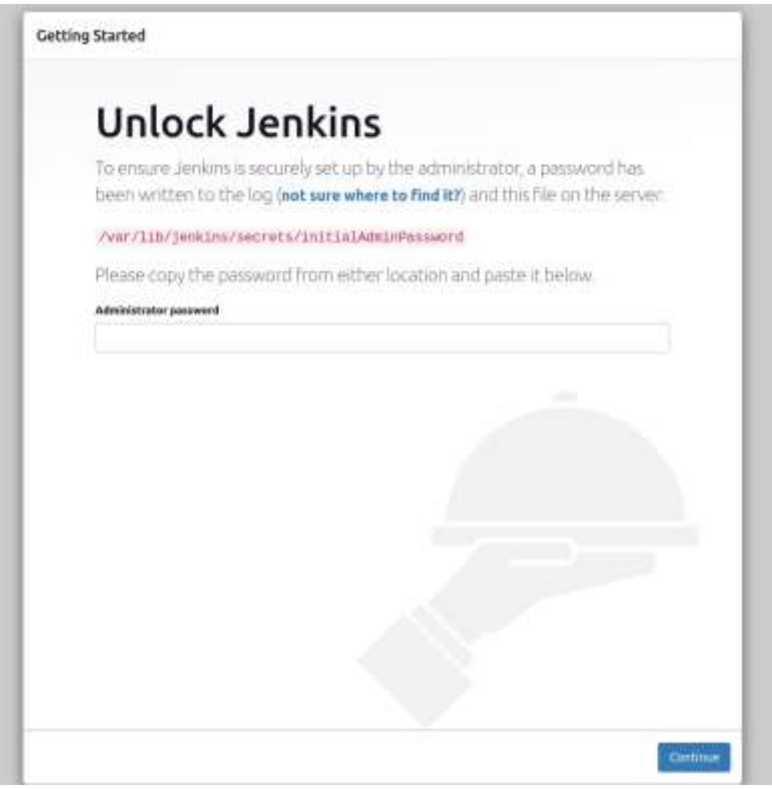
Let’s start Jenkins by using systemctl



```
kins.service.
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~$ sudo systemctl start jenkins
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~$ 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 Sun 2022-09-18 22:41:59 IST; 1min 15s ago
   Main PID: 33116 (java)
      Tasks: 49 (limit: 9181)
     Memory: 1.9G
        CPU: 1min 51.125s
     CGroup: /system.slice/jenkins.service
             └─33116 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webro>

Sep 18 22:41:30 purpleven-OMEN-Laptop-15-ek0xxx jenkins[33116]: This may also be found at: /var/lib/>
Sep 18 22:41:30 purpleven-OMEN-Laptop-15-ek0xxx jenkins[33116]: *************************************>
Sep 18 22:41:30 purpleven-OMEN-Laptop-15-ek0xxx jenkins[33116]: *************************************>
Sep 18 22:41:30 purpleven-OMEN-Laptop-15-ek0xxx jenkins[33116]: *************************************>
Sep 18 22:41:59 purpleven-OMEN-Laptop-15-ek0xxx jenkins[33116]: 2022-09-18 17:11:59.863+0000 [id=41]>
Sep 18 22:41:59 purpleven-OMEN-Laptop-15-ek0xxx jenkins[33116]: 2022-09-18 17:11:59.905+0000 [id=24]>
Sep 18 22:41:59 purpleven-OMEN-Laptop-15-ek0xxx systemd[1]: Started Jenkins Continuous Integration S>
Sep 18 22:42:02 purpleven-OMEN-Laptop-15-ek0xxx jenkins[33116]: 2022-09-18 17:12:02.546+0000 [id=60]>
Sep 18 22:42:02 purpleven-OMEN-Laptop-15-ek0xxx jenkins[33116]: 2022-09-18 17:12:02.547+0000 [id=60]>
Sep 18 22:42:02 purpleven-OMEN-Laptop-15-ek0xxx jenkins[33116]: 2022-09-18 17:12:02.548+0000 [id=60]>
lines 1-20/20 (END)
```

Step 3 — Setting Up Jenkins

To set up your installation, visit Jenkins on its default port, 8080, using your server domain name or IP address: http://*192.168.43.69:8080/*



Getting Started

Unlock Jenkins

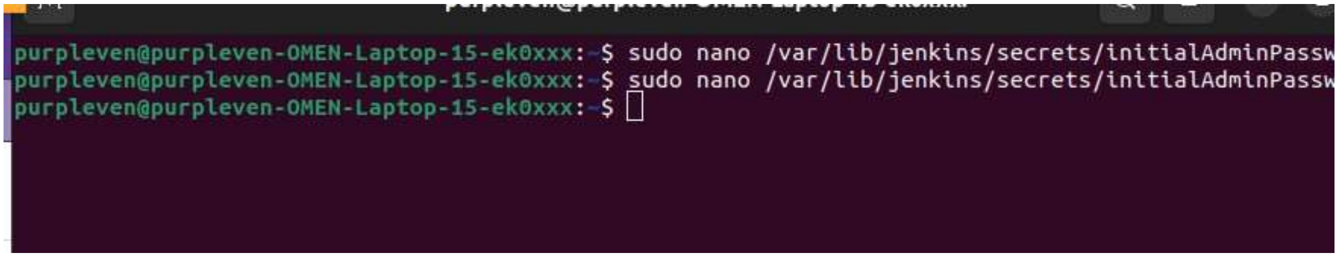
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (**not sure where to find it?**) and this file on the server:

`/var/lib/jenkins/secrets/initialAdminPassword`

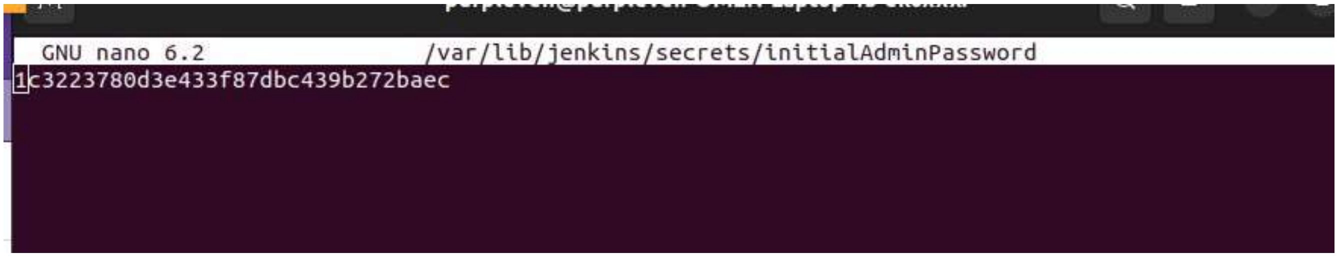
Please copy the password from either location and paste it below.

Administrator password

In the terminal window, use the nano command to display the password

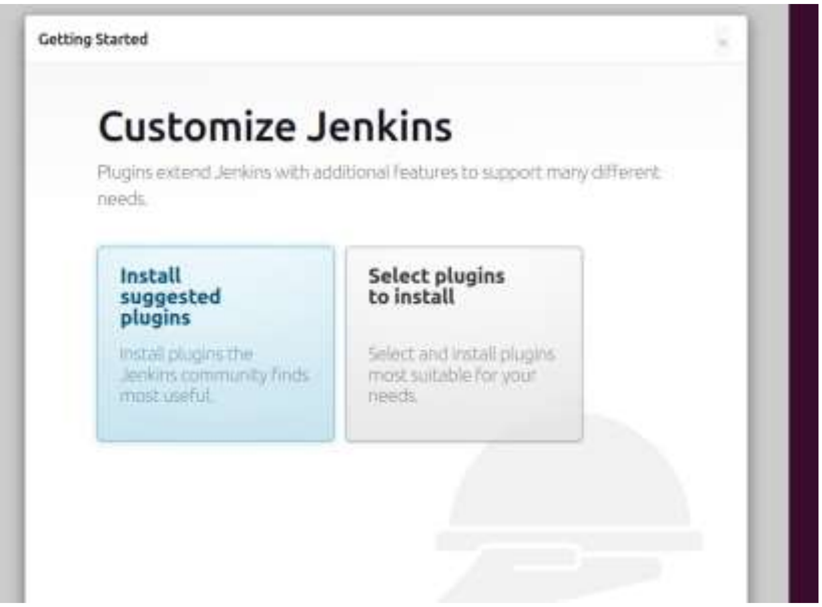


```
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~$ sudo nano /var/lib/jenkins/secrets/initialAdminPassw
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~$ sudo nano /var/lib/jenkins/secrets/initialAdminPassw
purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~$ 
```



```
  GNU nano 6.2                  /var/lib/jenkins/secrets/initialAdminPassword
1c3223780d3e433f87dbc439b272baec
```

The next screen presents the option of installing suggested plugins



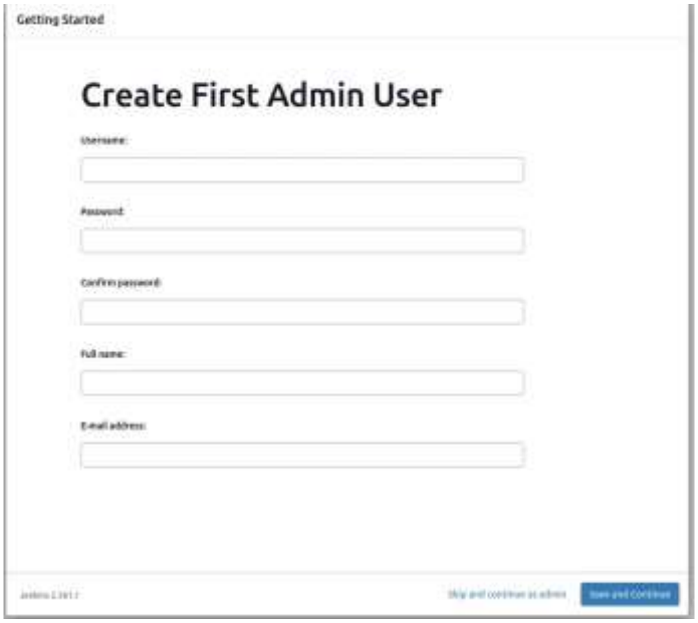
Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins
Install plugins the Jenkins community finds most useful.
Select plugins to install
Select and install plugins most suitable for your needs.

Enter the name and password for your user



Getting Started

Create First Admin User

Username:

Password:

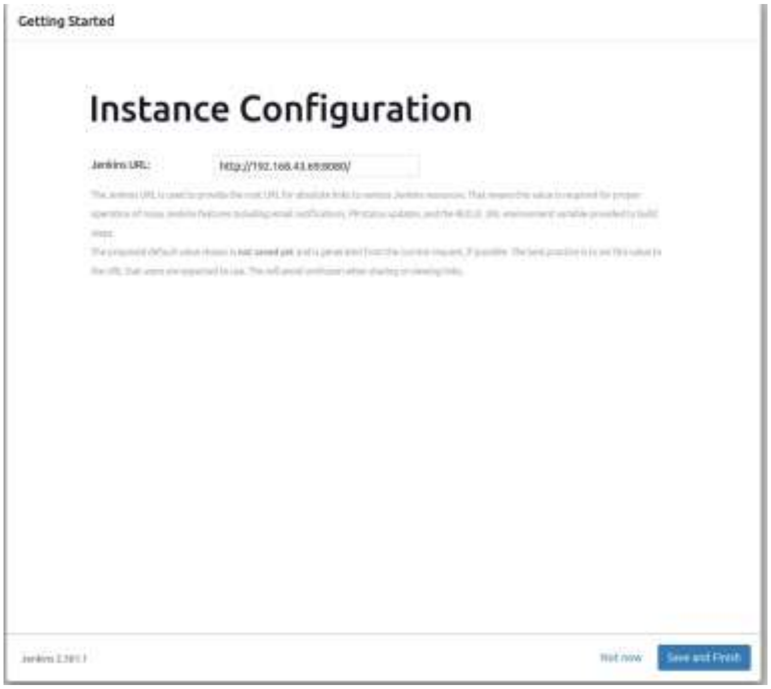
Confirm password:

Full name:

E-mail address:

Jenkins 2.361.1

You'll receive an Instance Configuration page that will ask you to confirm the preferred URL for your Jenkins instance.

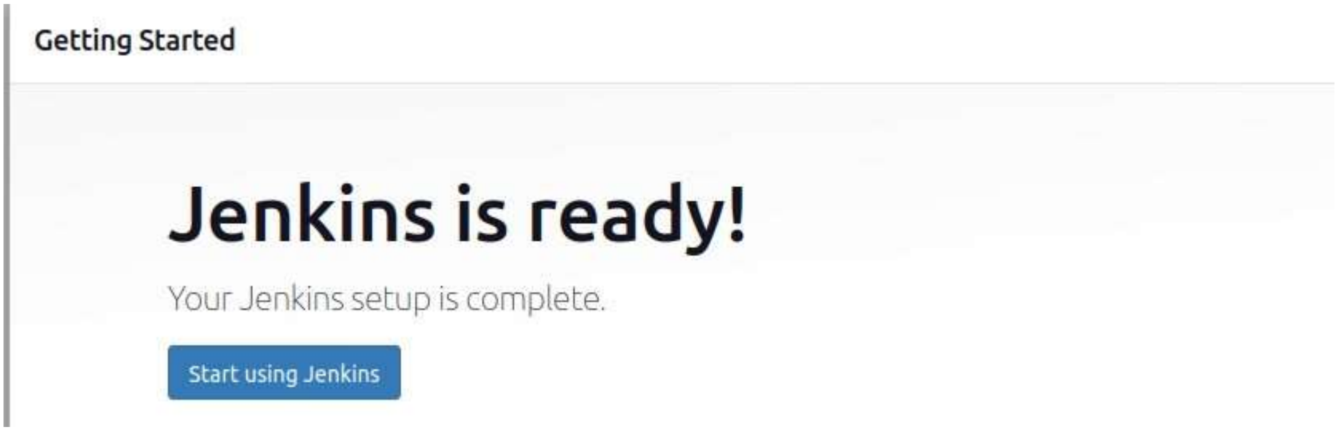


Jenkins URL: http://192.168.43.69:8080/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD_URL environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

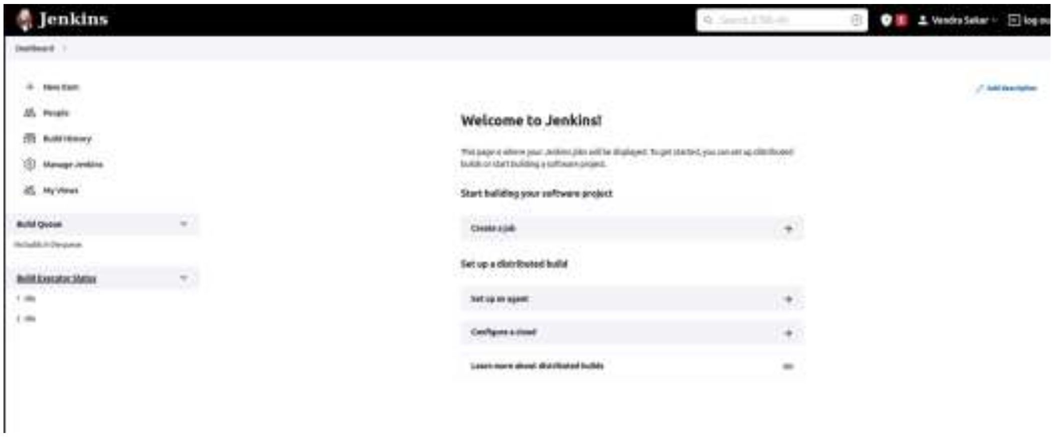
After confirming the appropriate information, click Save and Finish. You'll receive a confirmation page confirming that "Jenkins is Ready!"



Jenkins is ready!

Your Jenkins setup is complete.

Click Start using Jenkins to visit the main Jenkins dashboard



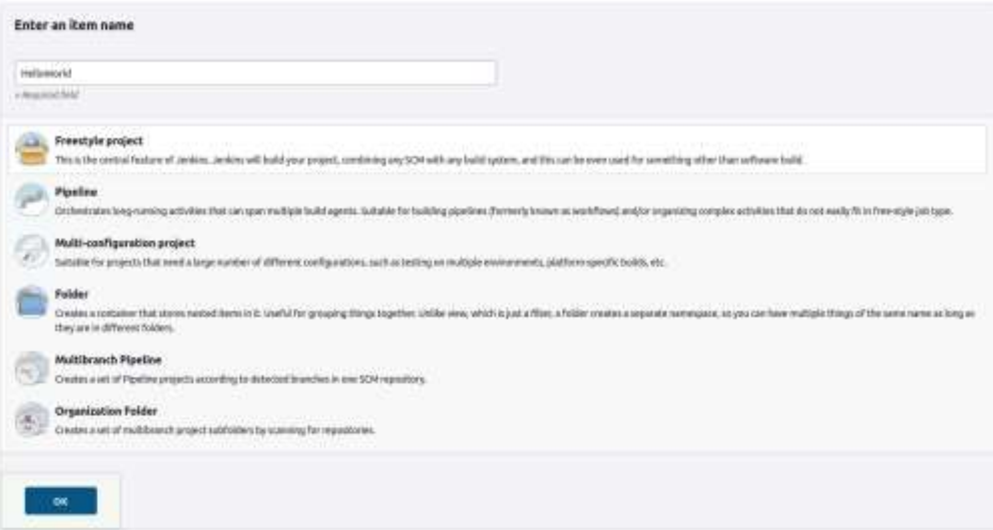
Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Set up a distributed build

Click on new item and create a new project and use the freestyle project plugin



Enter an item name

Helloworld

» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

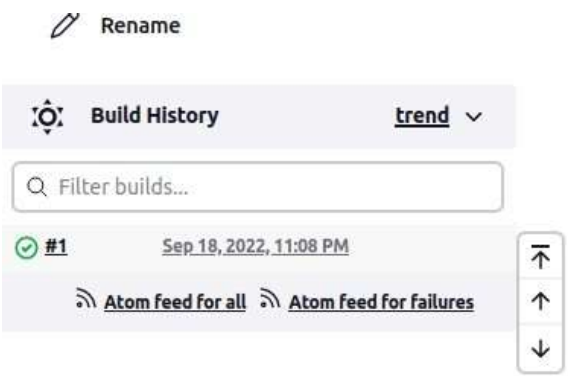
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

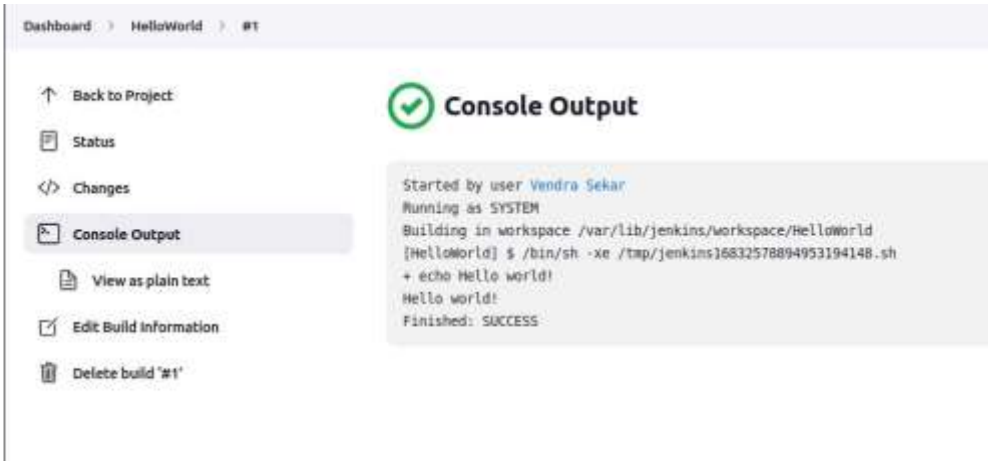
Organization Folder
Creates a set of multibranch project subfolders by scanning for repositories.

Use the execute shell feature and write a code, now click on build now



✓ #1 Sep 18, 2022, 11:08 PM

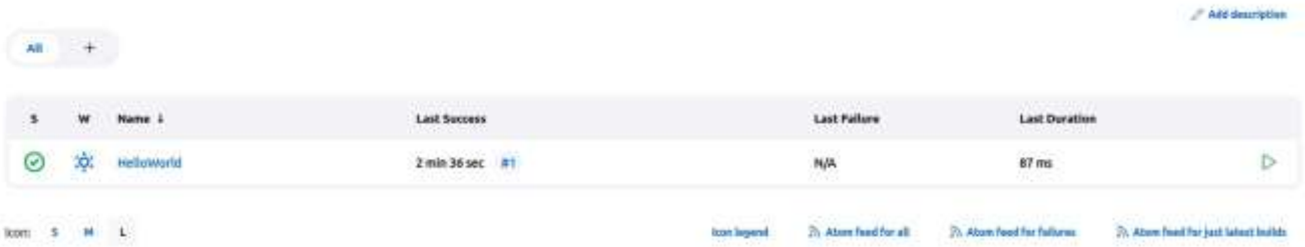
Now check on console output



Console Output

```
Started by user Vendra Sekar
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/HelloWorld
[HelloWorld] $ /bin/sh -xe /tmp/jenkins16832578894953194148.sh
+ echo Hello world!
Hello world!
Finished: SUCCESS
```

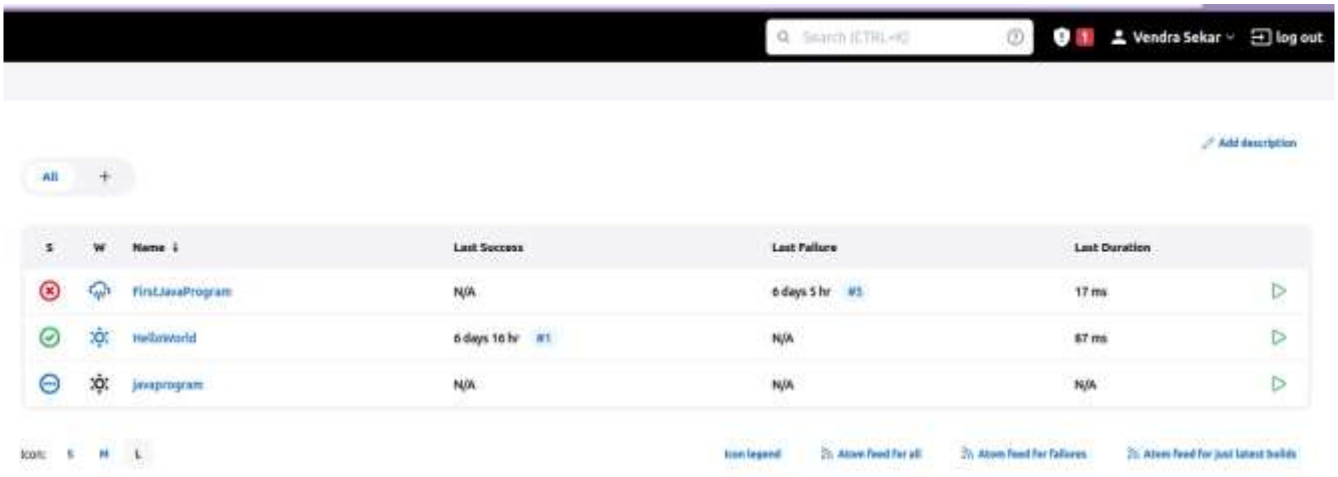
The output is successful



S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀	HelloWorld	2 min 36 sec #1	N/A	87 ms

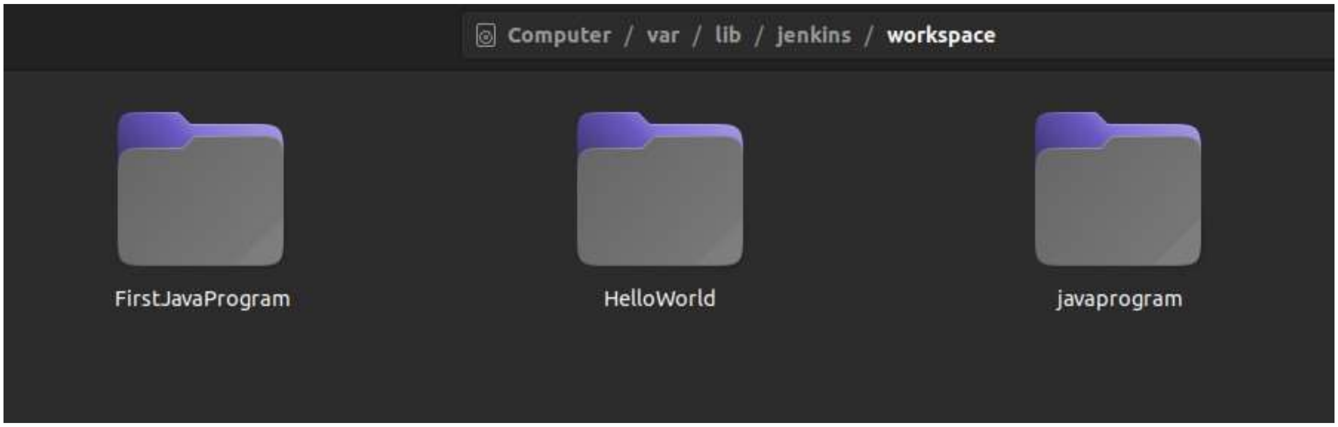
2. Exercise to create and run a java program using terminal

Create a new freestyle project

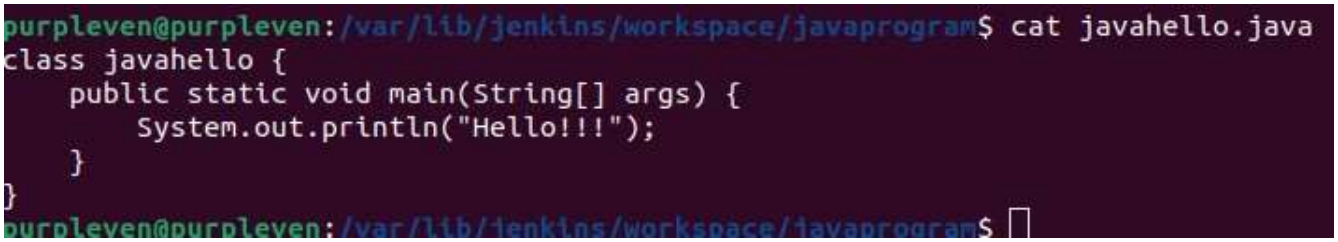


S	W	Name ↓	Last Success	Last Failure	Last Duration
⊗		FirstJavaProgram	N/A	6 days 5 hr #5	17 ms
✓		HelloWorld	6 days 16 hr #1	N/A	67 ms
⊖		javaprogram	N/A	N/A	N/A

Build it, and you will see that it is locally present

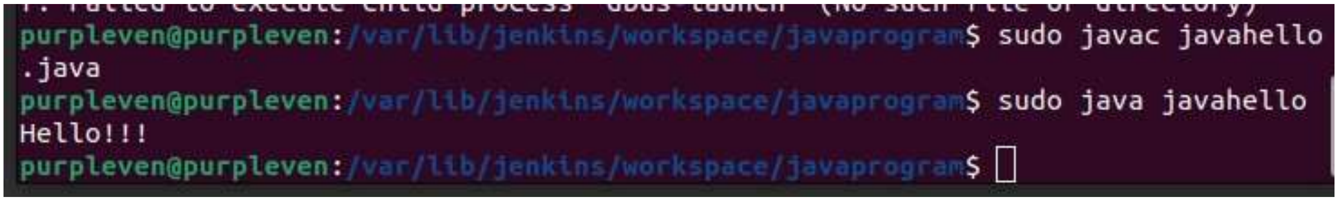


Write a java program



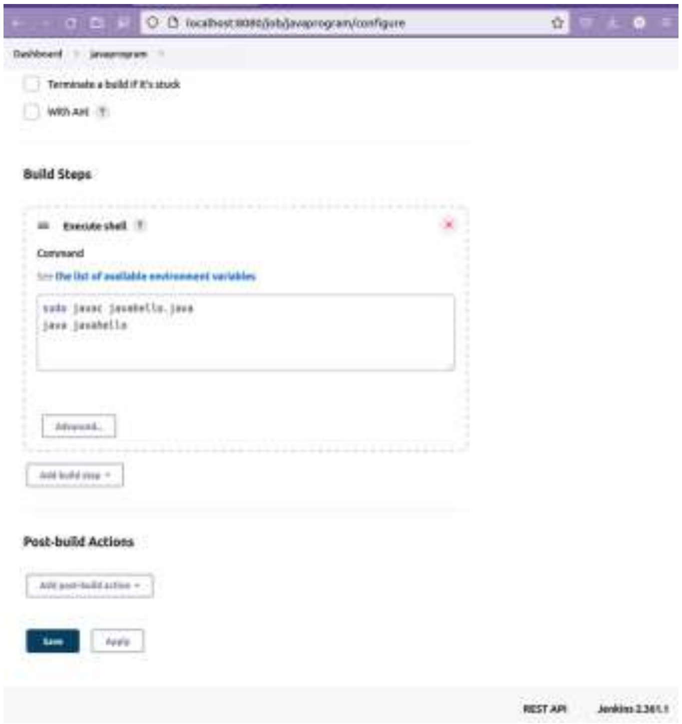
```
purpleven@purpleven:/var/lib/jenkins/workspace/javaprogram$ cat javahello.java
class javahello {
    public static void main(String[] args) {
        System.out.println("Hello!!!");
    }
}
purpleven@purpleven:/var/lib/jenkins/workspace/javaprogram$ 
```

Compile the java program



```
F: Failed to execute child process "dbus-launch" (No such file or directory)
purpleven@purpleven:/var/lib/jenkins/workspace/javaprogram$ sudo javac javahello
.java
purpleven@purpleven:/var/lib/jenkins/workspace/javaprogram$ sudo java javahello
Hello!!!
purpleven@purpleven:/var/lib/jenkins/workspace/javaprogram$ 
```

Configure the java program and add the two lines in the build step



localhost:8080/job/javaprogram/configure

Dashboard > javaprogram >

□ Terminate a build if it's stuck

□ With Ant

Build Steps

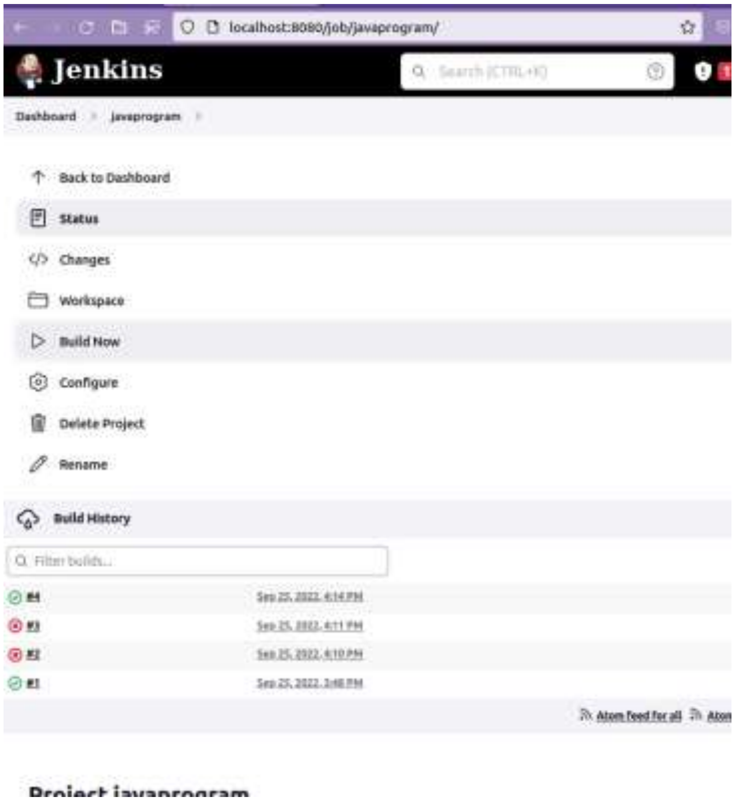
Execute shell

Command

See the list of available environment variables

```
sudo javac javahello.java
java javahello
```

Click on build now

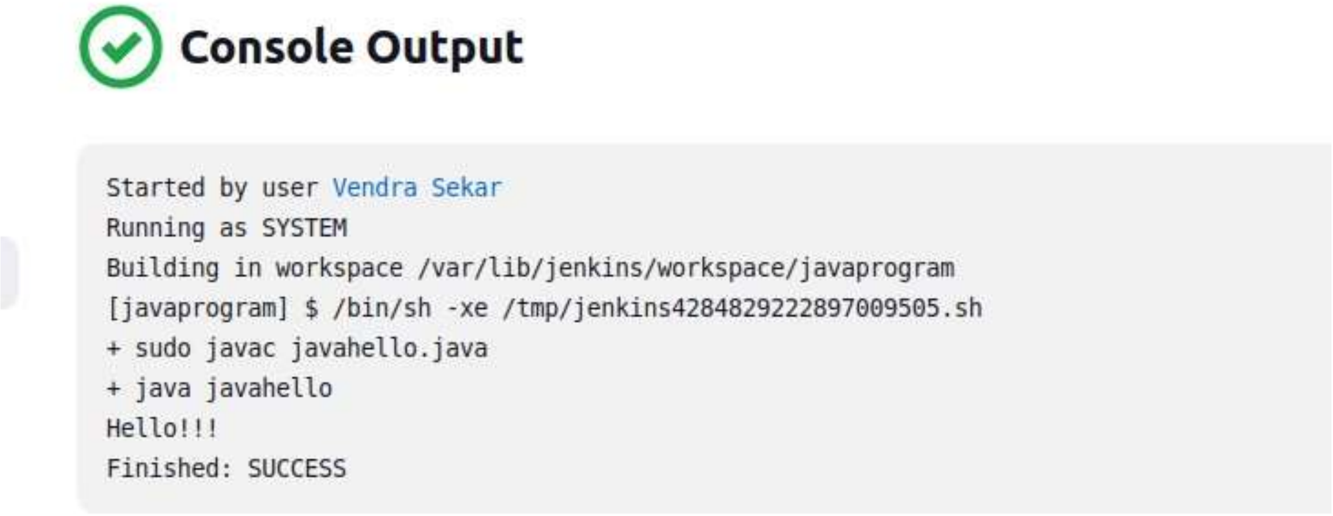


localhost:8080/job/javaprogram/

✓ #4	Sep 25, 2022, 4:14 PM
✗ #3	Sep 25, 2022, 4:11 PM
✗ #2	Sep 25, 2022, 4:10 PM
✓ #1	Sep 25, 2022, 3:46 PM

Project javaprogram

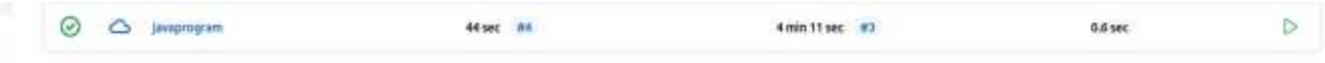
Check the output



Console Output

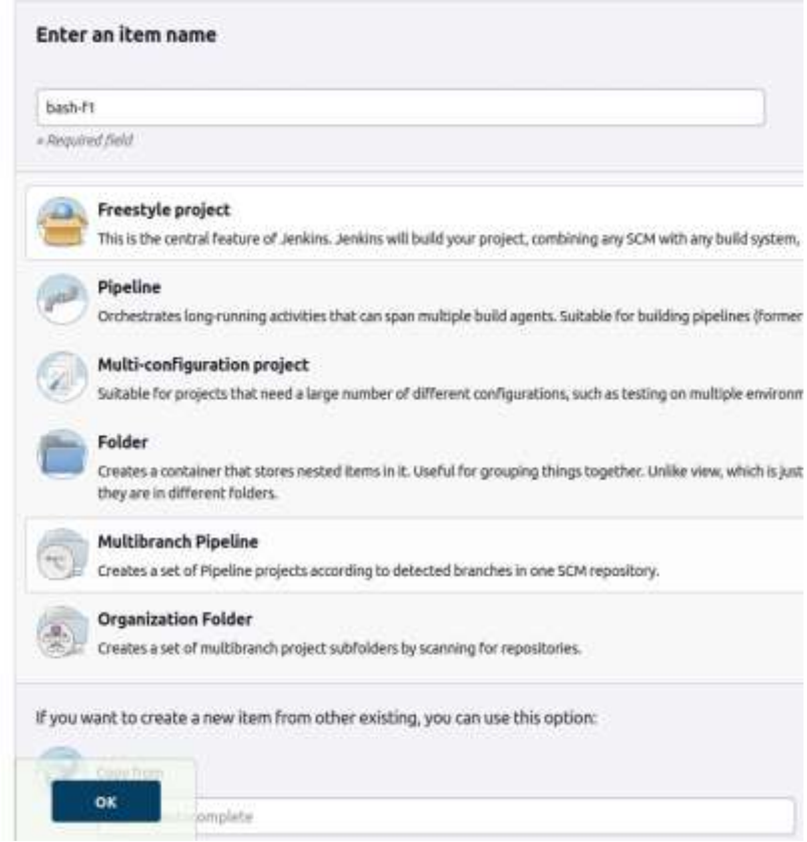
```
Started by user Vendra Sekar
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/javaprogram
[javaprogram] $ /bin/sh -xe /tmp/jenkins4284829222897009505.sh
+ sudo javac javahello.java
+ java javahello
Hello!!!
Finished: SUCCESS
```

The java program is successfully demonstrated on jenkins



✓	☁	javaprogram	44 sec #4	4 min 11 sec #3	0.6 sec	▷

3. Exercise to create a shell script and version it in Jenkins



Enter an item name

bash-f1

» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system,

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (former

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environm

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just
they are in different folders.

Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

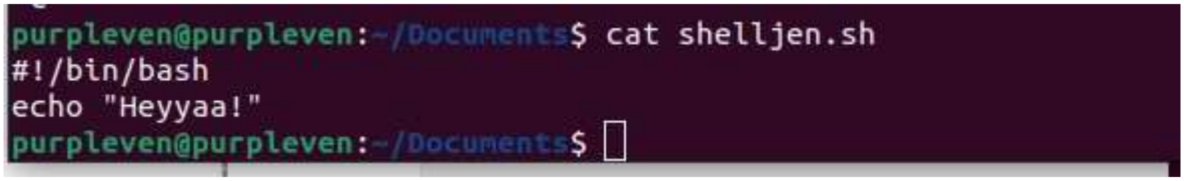
Organization Folder
Creates a set of multibranch project subfolders by scanning for repositories.

If you want to create a new item from other existing, you can use this option:

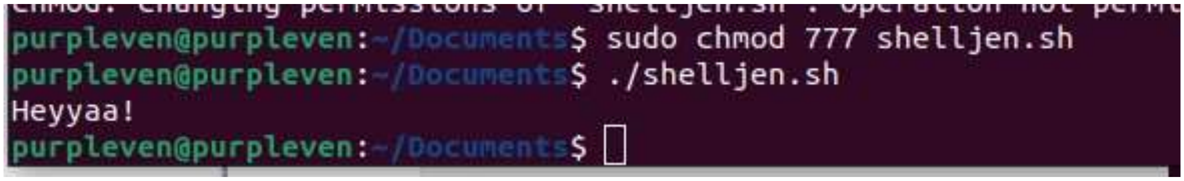
Copy from

OK

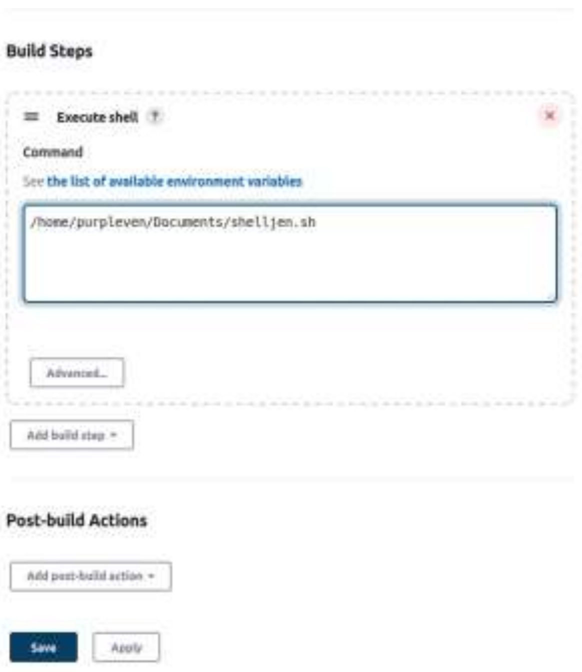
omplete



```
purpleven@purpleven:~/Documents$ cat shelljen.sh
#!/bin/bash
echo "Heyyaa!"
purpleven@purpleven:~/Documents$ 
```



```
chmod: changing permissions of 'shelljen.sh': Operation not permi
purpleven@purpleven:~/Documents$ sudo chmod 777 shelljen.sh
purpleven@purpleven:~/Documents$ ./shelljen.sh
Heyyaa!
purpleven@purpleven:~/Documents$ 
```



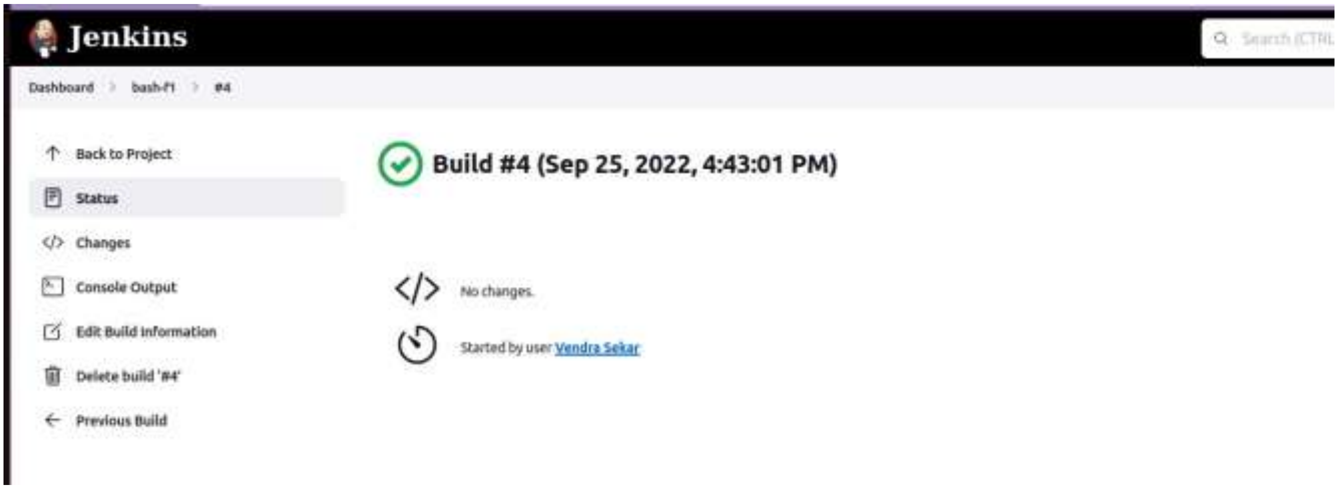
Build Steps

Execute shell

Command

See the list of available environment variables

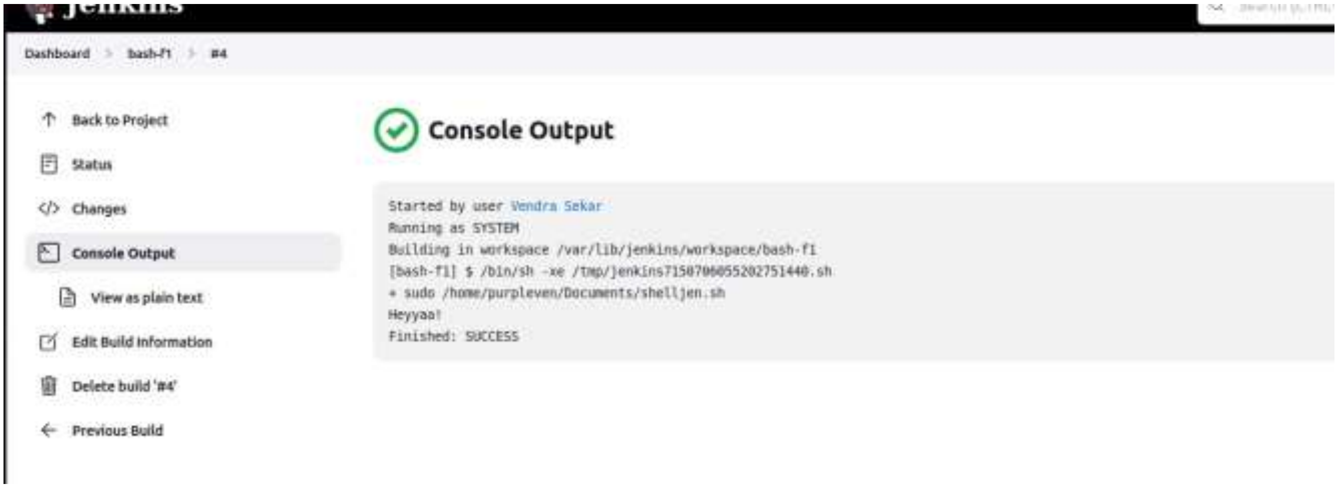
```
/home/purpleven/Documents/shelljen.sh
```



Build #4 (Sep 25, 2022, 4:43:01 PM)

No changes.

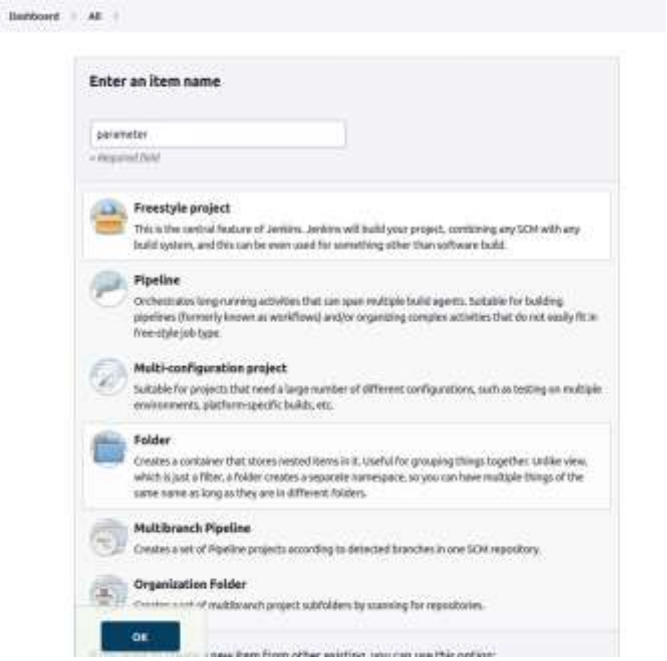
Started by user Vendra Sekar



Console Output

```
Started by user Vendra Sekar
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/bash-f1
[bash-f1] $ /bin/sh -xe /tmp/jenkins7150706055202751440.sh
+ sudo /home/purpleven/Documents/shelljen.sh
Heyyaa!
Finished: SUCCESS
```

4. Exercise to Create a parameterized project



Enter an item name

parameter

» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

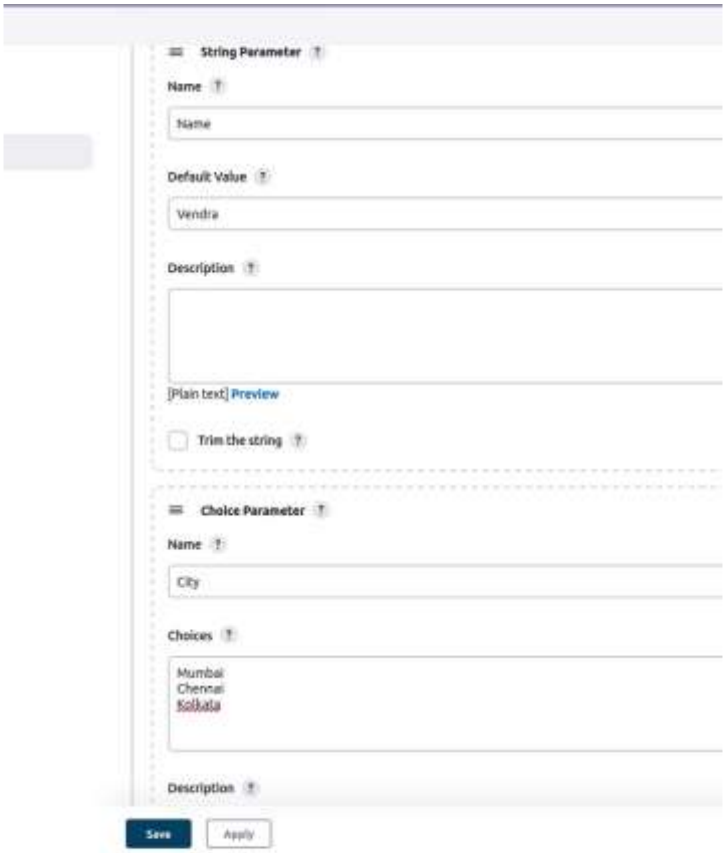
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

Organization Folder
Creates a set of multibranch project subfolders by scanning for repositories.

new item from other existing, you can use this option:



String Parameter ?

Name ?

Name

Default Value ?

Vendra

Description ?

[Plain text] Preview

□ Trim the string ?

Choice Parameter ?

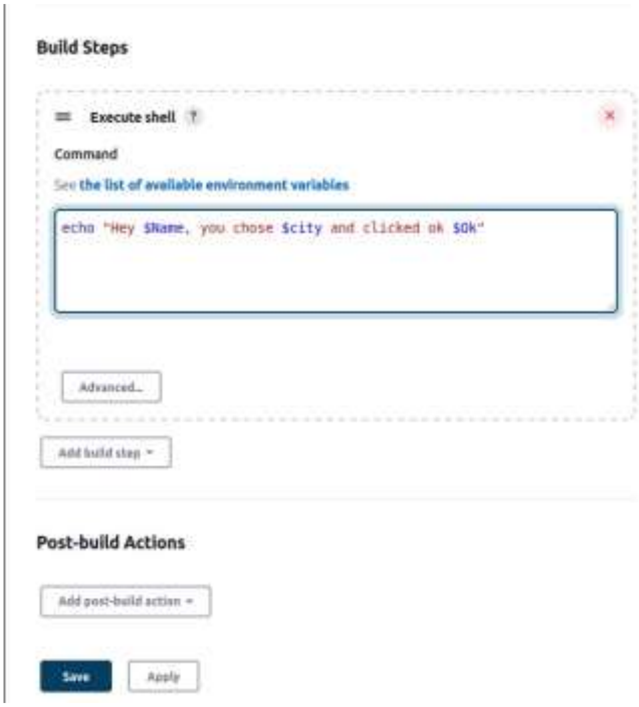
Name ?

City

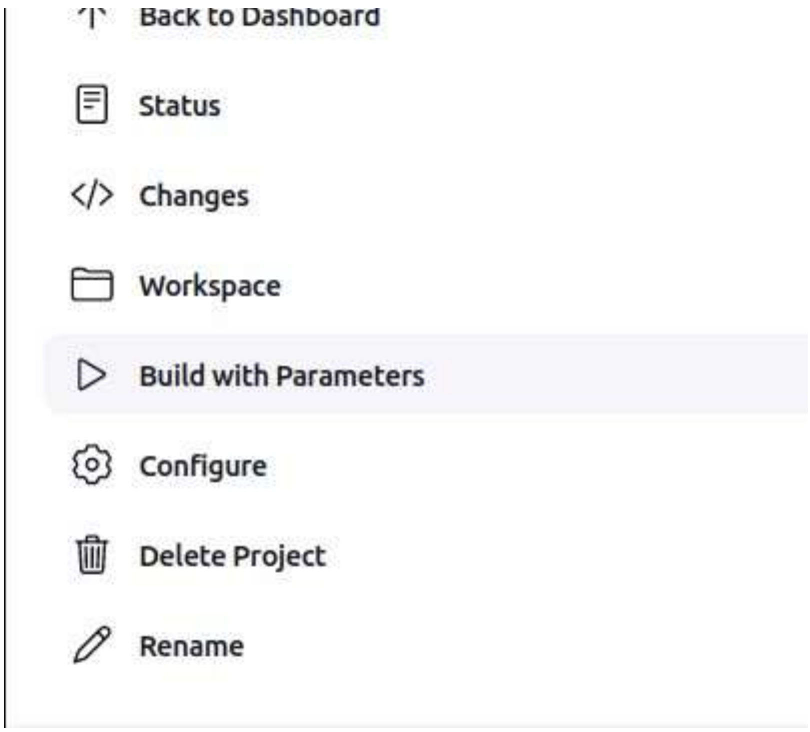
Choices ?

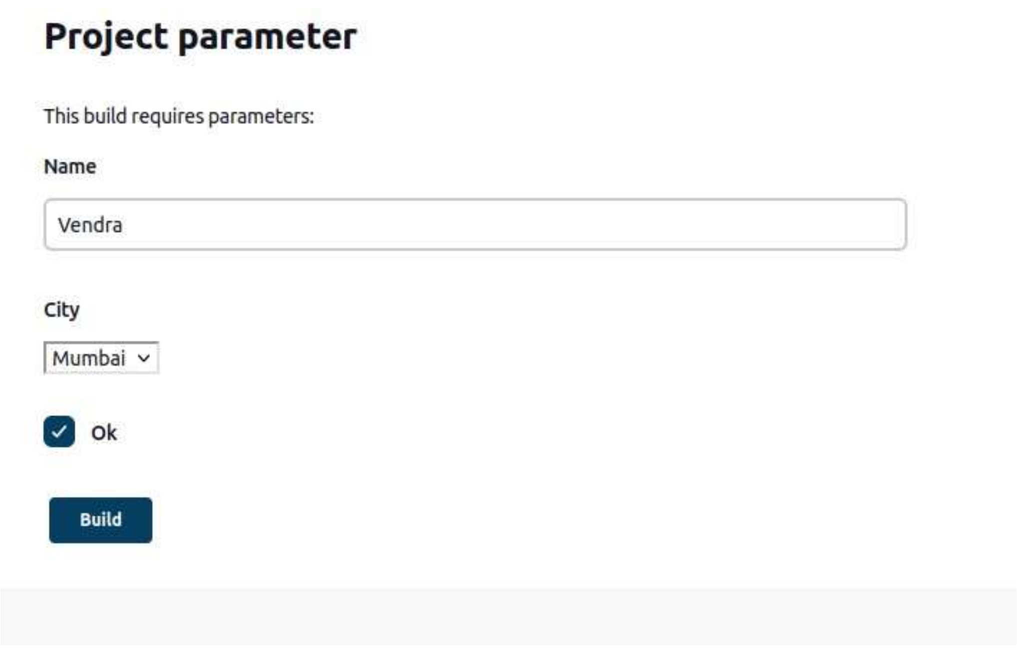
Mumbai
Chennai
Kolkata

Description ?



```
echo "Hey $Name, you chose $city and clicked ok $Ok"
```





Project parameter

This build requires parameters:

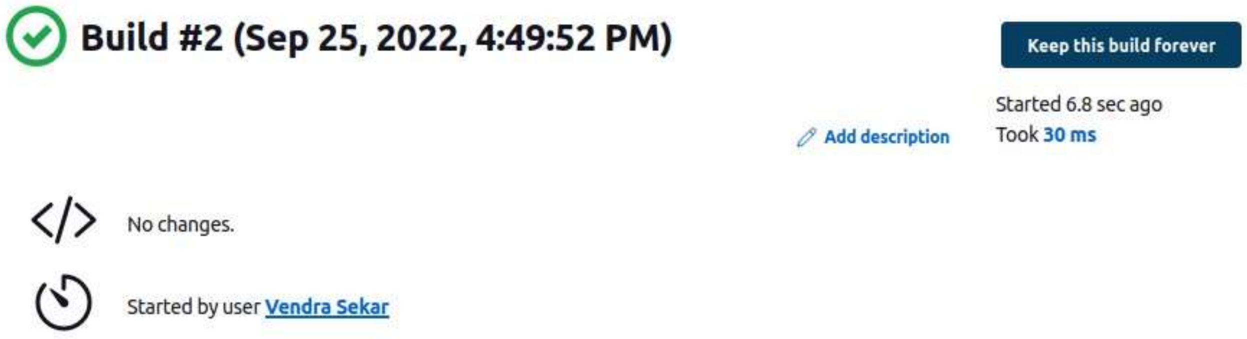
Name

Vendra

City

Mumbai

☑ Ok

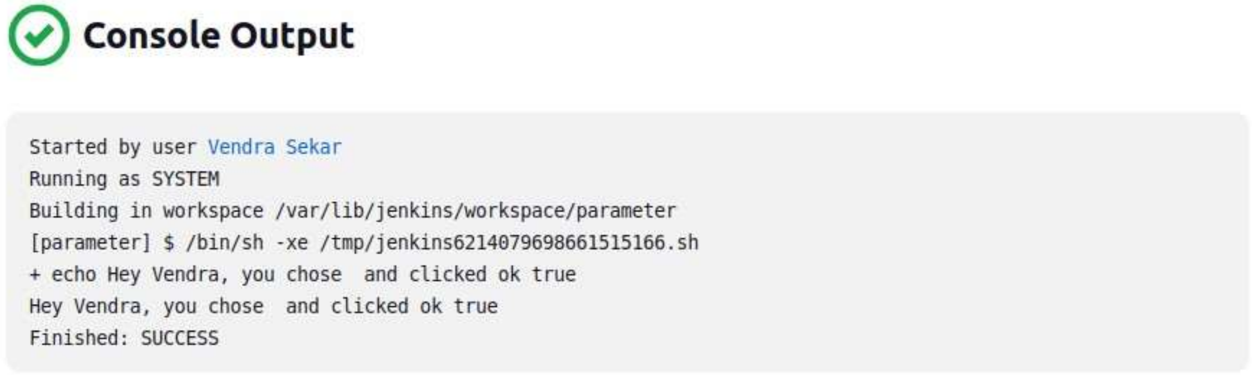


Build #2 (Sep 25, 2022, 4:49:52 PM)

Started 6.8 sec ago
Took 30 ms

No changes.

Started by user Vendra Sekar

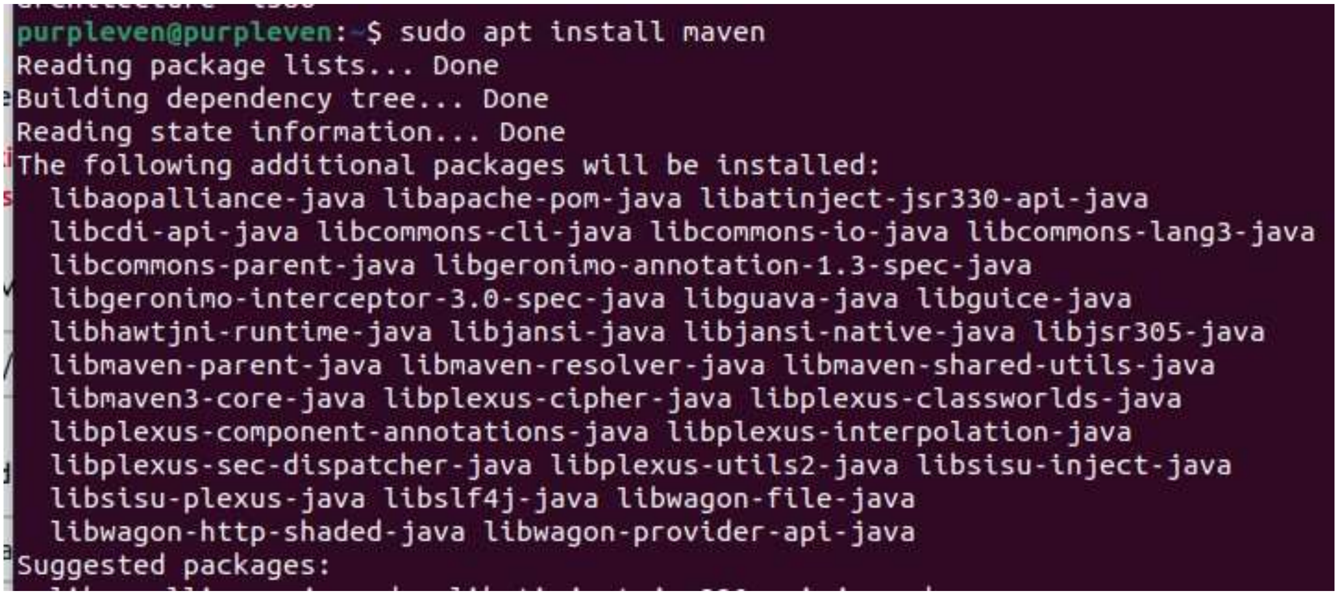


Console Output

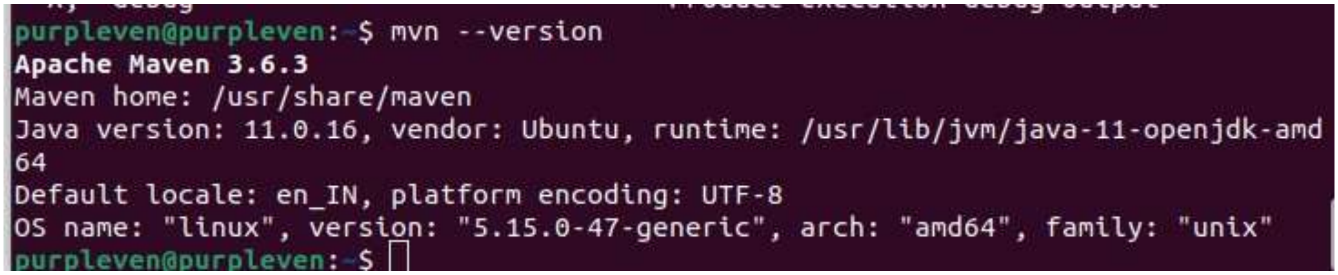
```
Started by user Vendra Sekar
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/parameter
[parameter] $ /bin/sh -xe /tmp/jenkins6214079698661515166.sh
+ echo Hey Vendra, you chose  and clicked ok true
Hey Vendra, you chose  and clicked ok true
Finished: SUCCESS
```

5. Exercise: create a maven Project

Install maven



```
purpleven@purpleven:~$ sudo apt install maven
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libaopalliance-java libapache-pom-java libatinject-jsr330-api-java
  libcdi-api-java libcommons-cli-java libcommons-io-java libcommons-lang3-java
  libcommons-parent-java libgeronimo-annotation-1.3-spec-java
  libgeronimo-interceptor-3.0-spec-java libguava-java libguice-java
  libhawtjni-runtime-java libjansi-java libjansi-native-java libjsr305-java
  libmaven-parent-java libmaven-resolver-java libmaven-shared-utils-java
  libmaven3-core-java libplexus-cipher-java libplexus-classworlds-java
  libplexus-component-annotations-java libplexus-interpolation-java
  libplexus-sec-dispatcher-java libplexus-utils2-java libsisu-inject-java
  libsisu-plexus-java libslf4j-java libwagon-file-java
  libwagon-http-shaded-java libwagon-provider-api-java
Suggested packages:
```

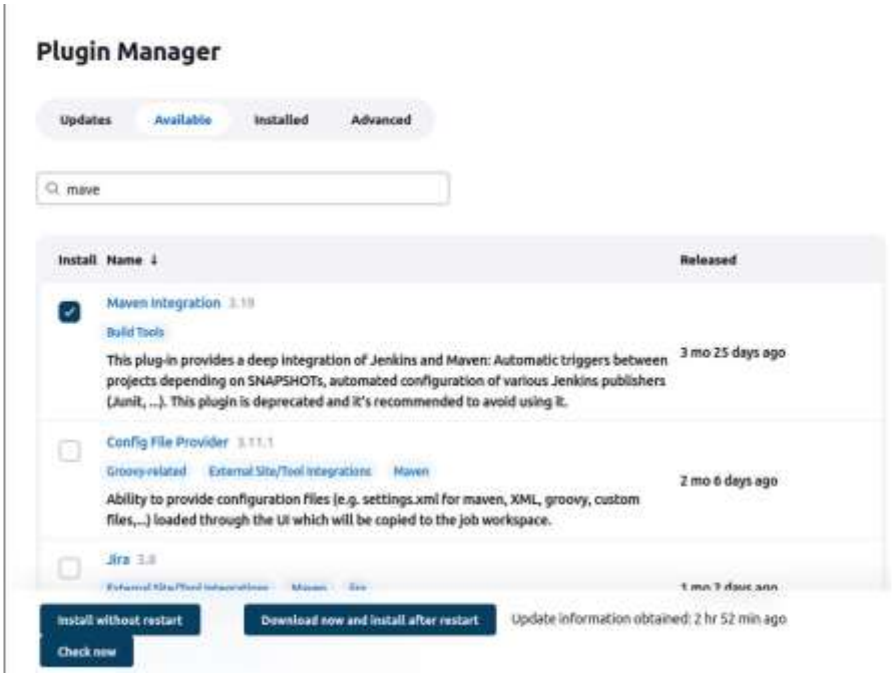


```
purpleven@purpleven:~$ mvn --version
Apache Maven 3.6.3
Maven home: /usr/share/maven
Java version: 11.0.16, vendor: Ubuntu, runtime: /usr/lib/jvm/java-11-openjdk-amd
64
Default locale: en_IN, platform encoding: UTF-8
OS name: "linux", version: "5.15.0-47-generic", arch: "amd64", family: "unix"
purpleven@purpleven:~$
```

Create a maven project

Eg: https://github.com/devopshint/java-app-with-maven/tree/main/my-app

Install maven integration plugin from the plugin manager



Plugin Manager

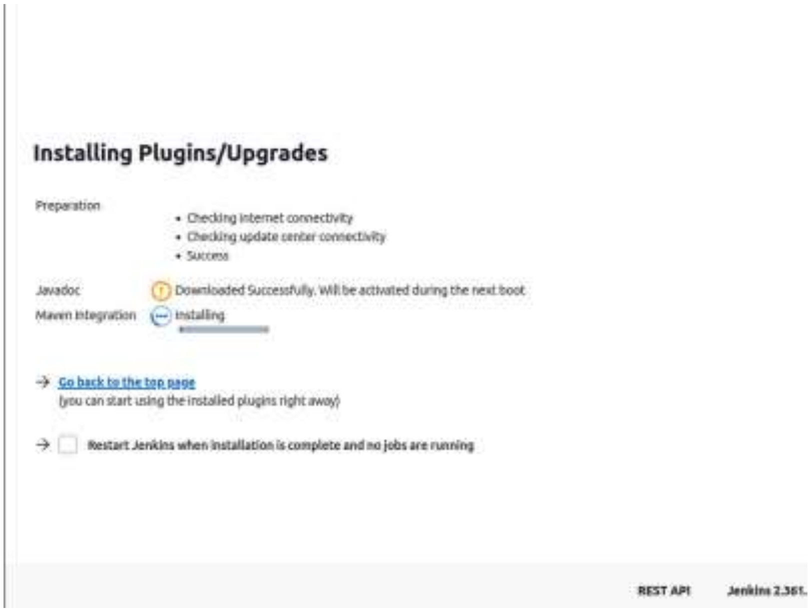
Updates | Available | Installed | Advanced

mave

Install	Name ↓	Released
☑	Maven Integration 3.19 Build Tools This plug-in provides a deep integration of Jenkins and Maven: Automatic triggers between projects depending on SNAPSHOTs, automated configuration of various Jenkins publishers (Junit, ...). This plugin is deprecated and it's recommended to avoid using it.	3 mo 25 days ago
□	Config File Provider 3.11.1 Groovy-related External Site/Tool Integrations Maven Ability to provide configuration files (e.g. settings.xml for maven, XML, groovy, custom files,...) loaded through the UI which will be copied to the job workspace.	2 mo 6 days ago
□	Jira 3.8 External Site/Tool Integrations Maven Jira	1 mo 7 days ago

Install without restart | Download now and install after restart | Update information obtained: 2 hr 52 min ago

Check now



Installing Plugins/Upgrades

Preparation
- Checking internet connectivity
- Checking update center connectivity
- Success

Javadoc — Downloaded Successfully. Will be activated during the next boot

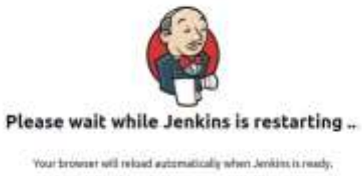
Maven Integration — Installing

→ Go back to the top page
(you can start using the installed plugins right away)

→ □ Restart Jenkins when installation is complete and no jobs are running

REST API Jenkins 2.361.

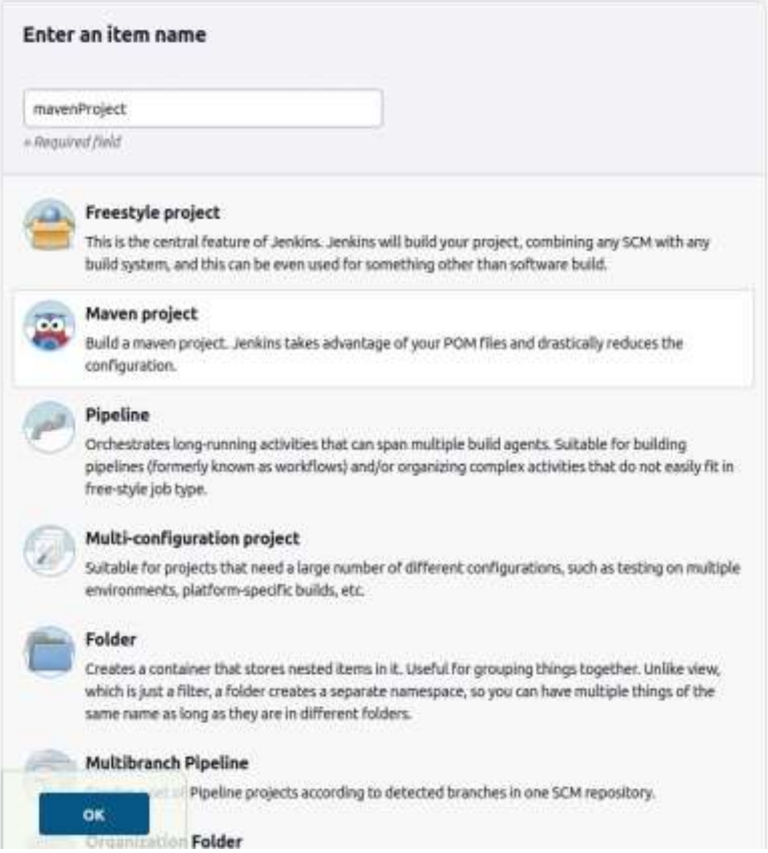
Restart jenkins



Please wait while Jenkins is restarting ..

Your browser will reload automatically when Jenkins is ready.

Create a new project



Enter an item name

mavenProject

» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

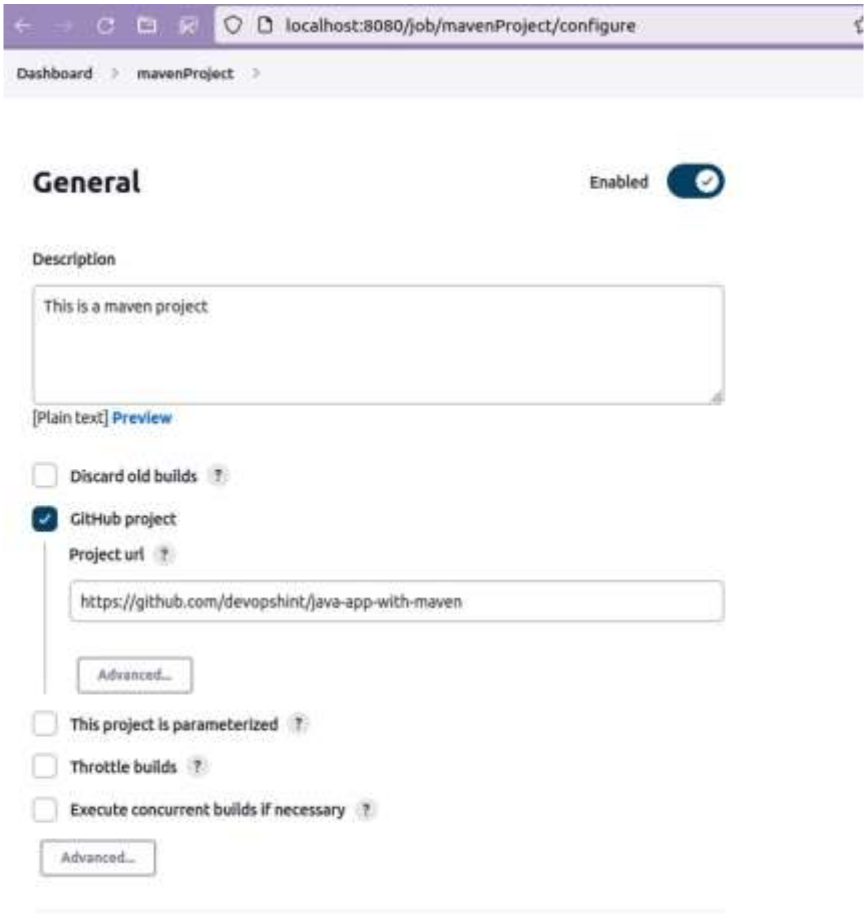
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

Organization Folder


localhost:8080/job/mavenProject/configure

Dashboard > mavenProject >

General

Enabled

Description

This is a maven project

[Plain text] Preview

☐ Discard old builds ?

☑ GitHub project

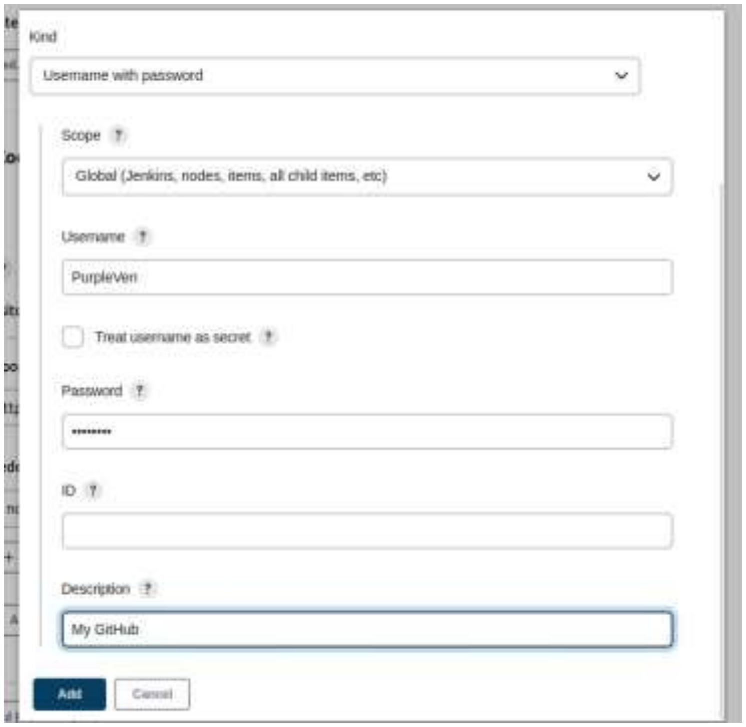
Project url ?

https://github.com/devopshint/java-app-with-maven

☐ This project is parameterized ?

☐ Throttle builds ?

☐ Execute concurrent builds if necessary ?


Kind

Username with password

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

Username ?

PurpleVen

☐ Treat username as secret ?

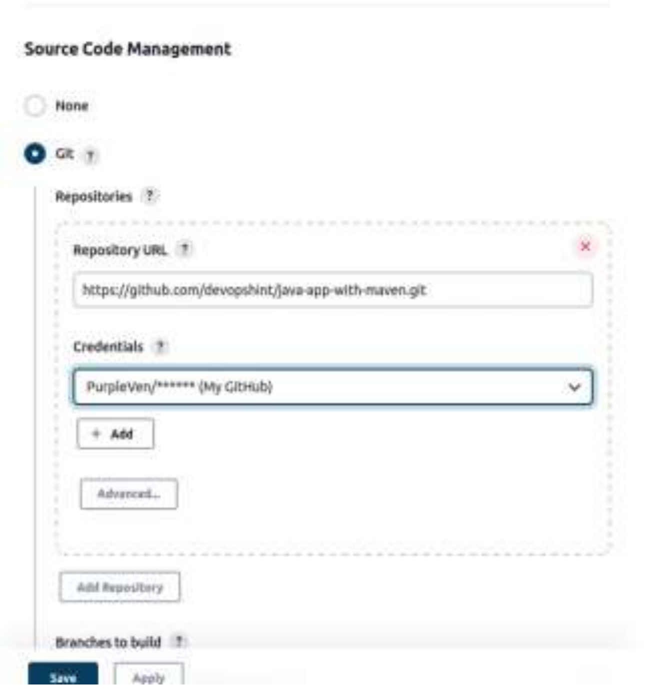
Password ?

••••••••

ID ?

Description ?

My GitHub



Source Code Management

○ None

● Git ?

Repositories ?

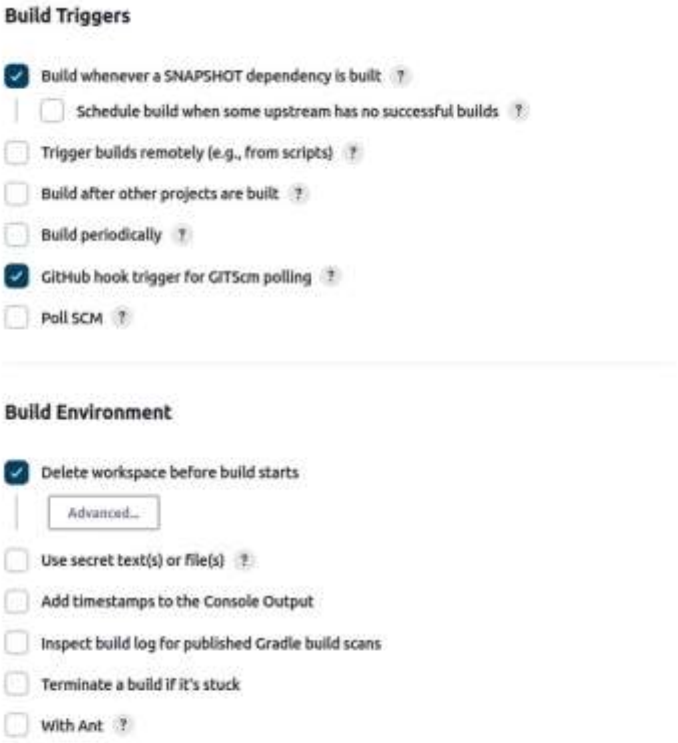
Repository URL ?

https://github.com/devopshint/java-app-with-maven.git

Credentials ?

PurpleVen/****** (My GitHub)

Branches to build ?



Build Triggers

☑ Build whenever a SNAPSHOT dependency is built ?

□ Schedule build when some upstream has no successful builds ?

□ Trigger builds remotely (e.g., from scripts) ?

□ Build after other projects are built ?

□ Build periodically ?

☑ GitHub hook trigger for GITScm polling ?

□ Poll SCM ?

Build Environment

☑ Delete workspace before build starts

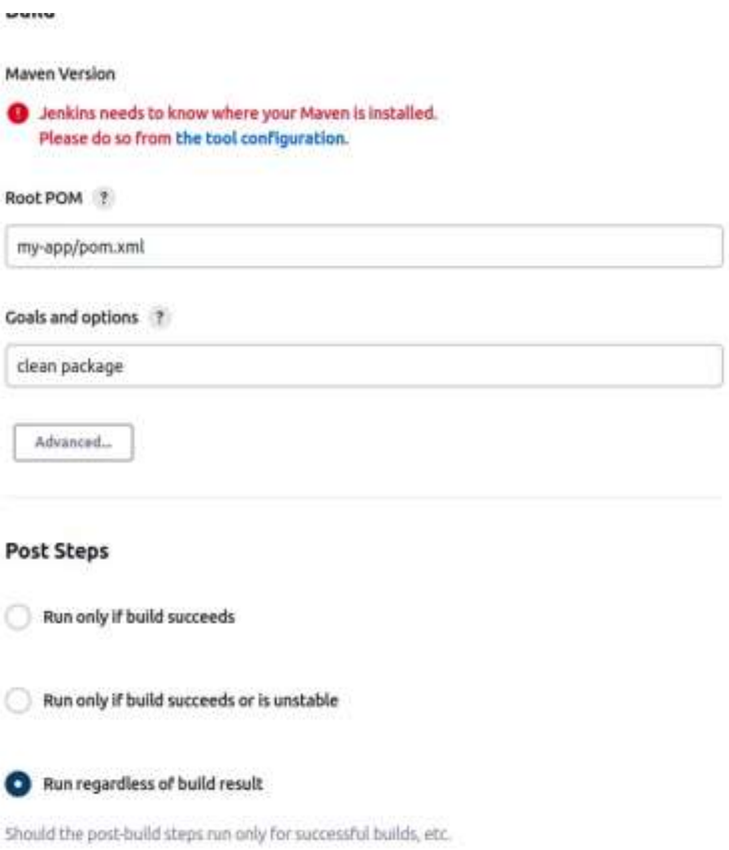
□ Use secret text(s) or file(s) ?

□ Add timestamps to the Console Output

□ Inspect build log for published Gradle build scans

□ Terminate a build if it's stuck

□ With Ant ?



Maven Version

Jenkins needs to know where your Maven is installed.
Please do so from the tool configuration.

Root POM ?

my-app/pom.xml

Goals and options ?

clean package

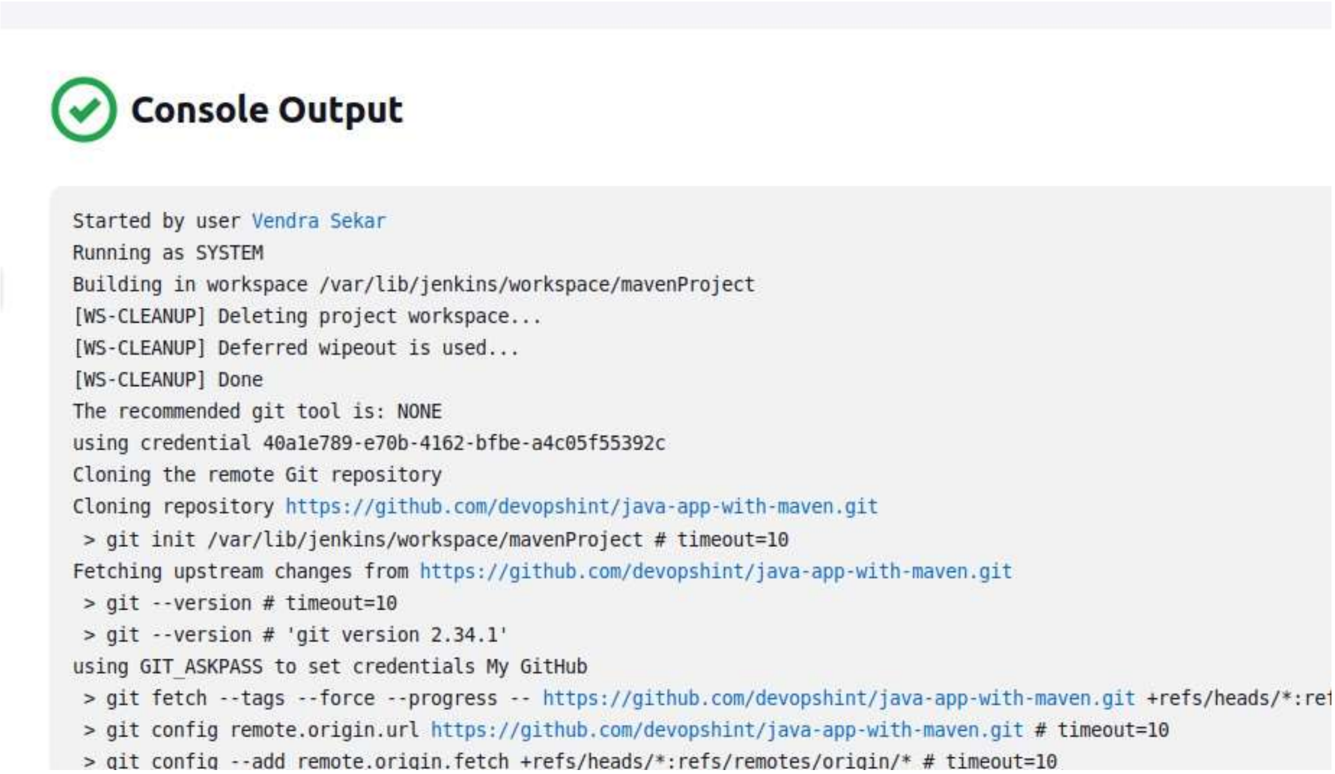
Post Steps

○ Run only if build succeeds

○ Run only if build succeeds or is unstable

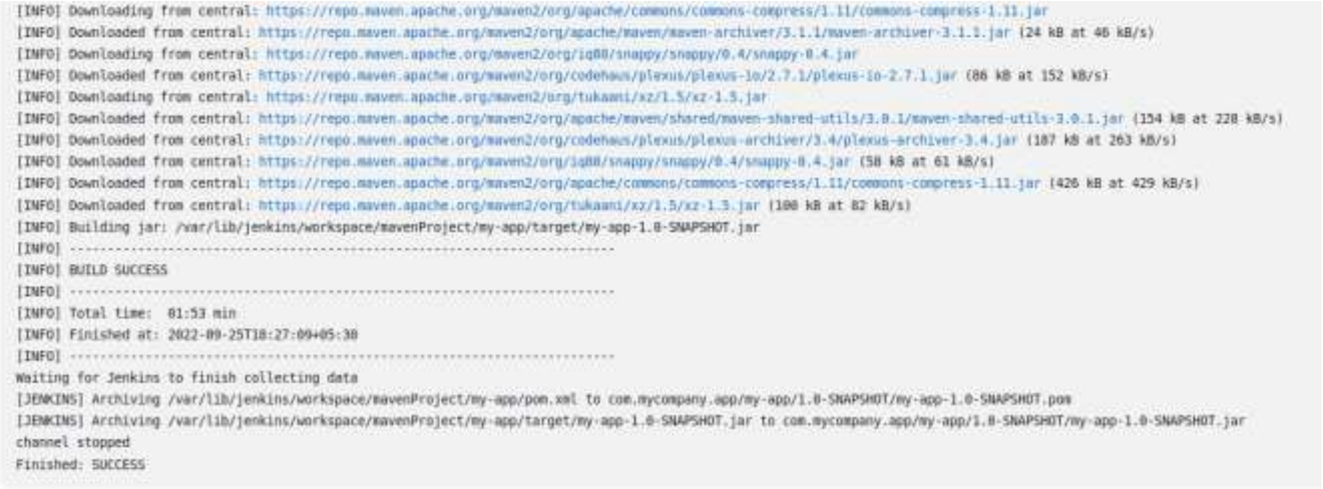
● Run regardless of build result

Should the post-build steps run only for successful builds, etc.



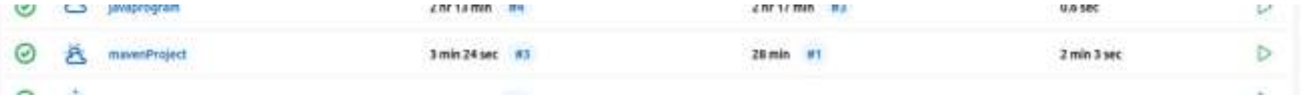
Console Output

```
Started by user Vendra Sekar
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/mavenProject
[WS-CLEANUP] Deleting project workspace...
[WS-CLEANUP] Deferred wipeout is used...
[WS-CLEANUP] Done
The recommended git tool is: NONE
using credential 40a1e789-e70b-4162-bfbe-a4c05f55392c
Cloning the remote Git repository
Cloning repository https://github.com/devopshint/java-app-with-maven.git
 > git init /var/lib/jenkins/workspace/mavenProject # timeout=10
Fetching upstream changes from https://github.com/devopshint/java-app-with-maven.git
 > git --version # timeout=10
 > git --version # 'git version 2.34.1'
using GIT_ASKPASS to set credentials My GitHub
 > git fetch --tags --force --progress -- https://github.com/devopshint/java-app-with-maven.git +refs/heads/*:ref
 > git config remote.origin.url https://github.com/devopshint/java-app-with-maven.git # timeout=10
 > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
```



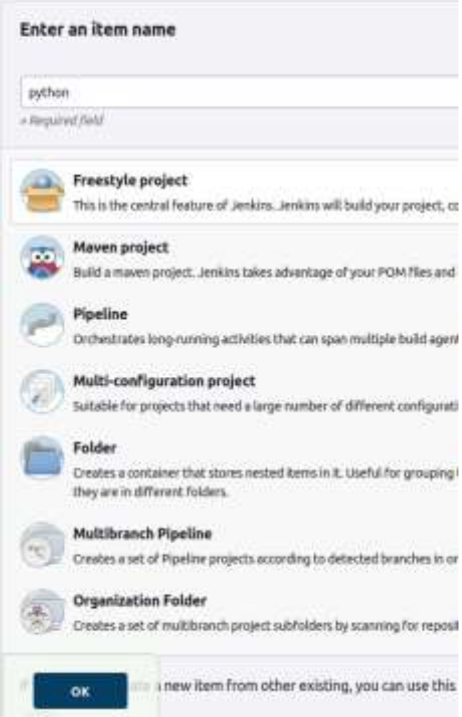
```
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/commons/commons-compress/1.11/commons-compress-1.11.jar
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-archiver/3.1.1/maven-archiver-3.1.1.jar (24 kB at 46 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/iq80/snappy/snappy/0.4/snappy-0.4.jar
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-io/2.7.1/plexus-io-2.7.1.jar (86 kB at 152 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/tukaani/xz/1.5/xz-1.5.jar
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-shared-utils/3.0.1/maven-shared-utils-3.0.1.jar (154 kB at 228 kB/s)
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-archiver/3.4/plexus-archiver-3.4.jar (187 kB at 263 kB/s)
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/iq80/snappy/snappy/0.4/snappy-0.4.jar (58 kB at 61 kB/s)
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/commons/commons-compress/1.11/commons-compress-1.11.jar (426 kB at 429 kB/s)
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/tukaani/xz/1.5/xz-1.5.jar (100 kB at 82 kB/s)
[INFO] Building jar: /var/lib/jenkins/workspace/mavenProject/my-app/target/my-app-1.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  01:53 min
[INFO] Finished at: 2022-09-25T18:27:09+05:30
[INFO] ------------------------------------------------------------------------
Waiting for Jenkins to finish collecting data
[JENKINS] Archiving /var/lib/jenkins/workspace/mavenProject/my-app/pom.xml to com.mycompany.app/my-app/1.0-SNAPSHOT/my-app-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/mavenProject/my-app/target/my-app-1.0-SNAPSHOT.jar to com.mycompany.app/my-app/1.0-SNAPSHOT/my-app-1.0-SNAPSHOT.jar
channel stopped
Finished: SUCCESS
```

The maven project is created successfully



javaprogram	2 hr 13 min #4	2 hr 17 min #3	0.6 sec	
mavenProject	3 min 24 sec #3	28 min #1	2 min 3 sec	▷

6. Exercise: to run a python simple program using freestyle



Enter an item name

python

» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, co

Maven project
Build a maven project. Jenkins takes advantage of your POM files and

Pipeline
Orchestrates long-running activities that can span multiple build agent

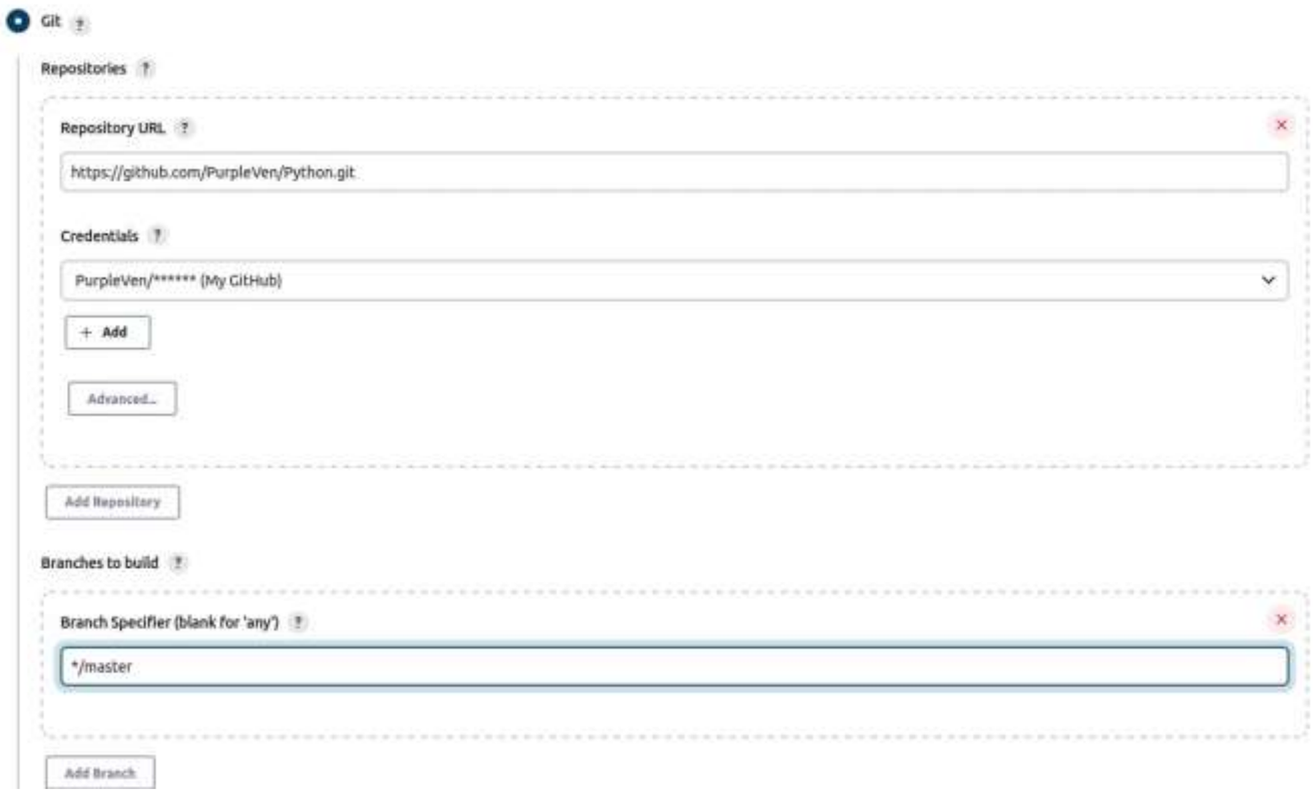
Multi-configuration project
Suitable for projects that need a large number of different configurati

Folder
Creates a container that stores nested items in it. Useful for grouping
they are in different folders.

Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in or

Organization Folder
Creates a set of multibranch project subfolders by scanning for reposi

a new item from other existing, you can use this



● Git

Repositories

Repository URL: https://github.com/PurpleVen/Python.git

Credentials: PurpleVen/****** (My GitHub)

Branches to build

Branch Specifier (blank for 'any'): */master

Conclusion: Thus, installation and version controlling for various program was done successfully

References: https://www.youtube.com/watch?v=-5tA3hZTVfA
https://github.com/devopshint/java-app-with-maven
https://www.youtube.com/watch?v=3S4FFwPqxRU&t=215s

EXPERIMENT 3

Docker

Date of Experiment: 26/09/2022

Output:

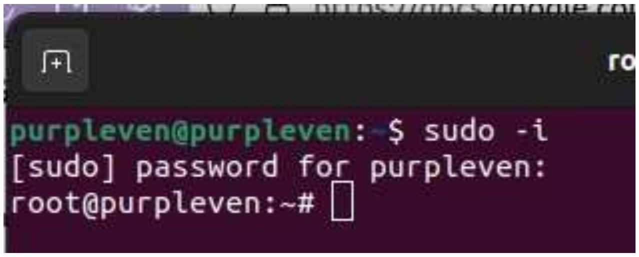
Part A: Steps for Installing Docker:

1. Open the terminal on Ubuntu.

2. Remove any Docker files that are running in the system, using the following command:

```
$ sudo apt-get remove docker docker-engine docker.io
```

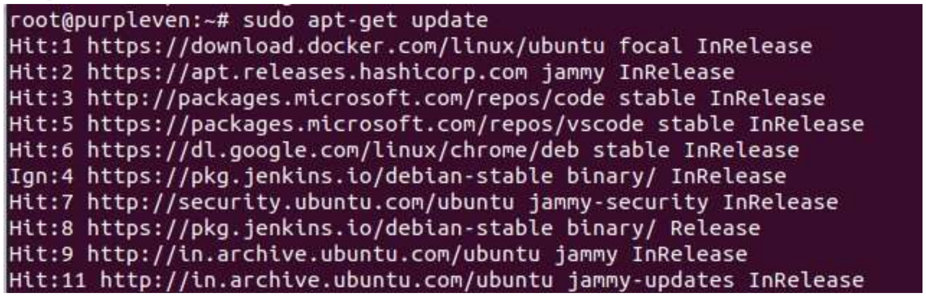
After entering the above command, you will need to enter the password of the root and press enter.



```
purpleven@purpleven:~$ sudo -i
[sudo] password for purpleven:
root@purpleven:~# 
```

3. Check if the system is up-to-date using the following command:

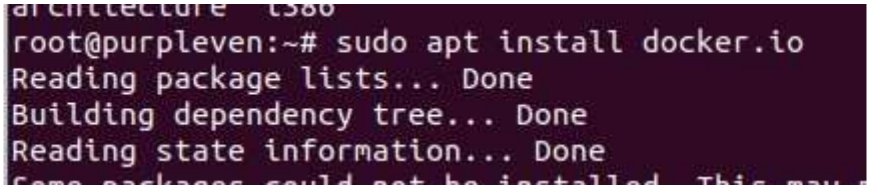
$ sudo apt-get update



```
root@purpleven:~# sudo apt-get update
Hit:1 https://download.docker.com/linux/ubuntu focal InRelease
Hit:2 https://apt.releases.hashicorp.com jammy InRelease
Hit:3 http://packages.microsoft.com/repos/code stable InRelease
Hit:5 https://packages.microsoft.com/repos/vscode stable InRelease
Hit:6 https://dl.google.com/linux/chrome/deb stable InRelease
Ign:4 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:7 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:8 https://pkg.jenkins.io/debian-stable binary/ Release
Hit:9 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:11 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease
```

4. Install Docker using the following command:

```
$ sudo apt install docker.io
```

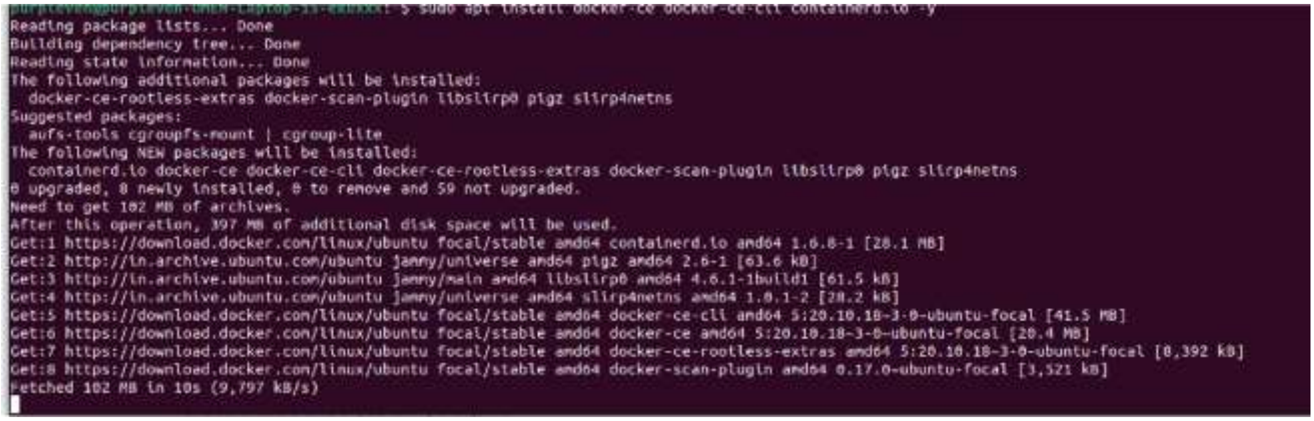


```
architecture  i386
root@purpleven:~# sudo apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Some packages could not be installed. This may
```

You'll then get a prompt asking you to choose between y/n - choose *y*

5. Install all the dependency packages using the following command:

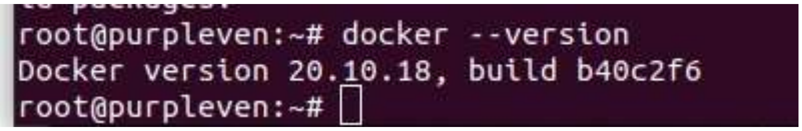
```
$ sudo snap install docker
```



```
purpleven@purpleven-[illegible]: $ sudo apt install docker-ce docker-ce-cli containerd.io -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  docker-ce-rootless-extras docker-scan-plugin libslirp0 pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-ce docker-ce-cli docker-ce-rootless-extras docker-scan-plugin libslirp0 pigz slirp4netns
0 upgraded, 8 newly installed, 0 to remove and 59 not upgraded.
Need to get 102 MB of archives.
After this operation, 397 MB of additional disk space will be used.
Get:1 https://download.docker.com/linux/ubuntu focal/stable amd64 containerd.io amd64 1.6.8-1 [28.1 MB]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63.6 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libslirp0 amd64 4.6.1-1build1 [61.5 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy/universe amd64 slirp4netns amd64 1.0.1-2 [28.2 kB]
Get:5 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-ce-cli amd64 5:20.10.18-3-0-ubuntu-focal [41.5 MB]
Get:6 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-ce amd64 5:20.10.18-3-0-ubuntu-focal [20.4 MB]
Get:7 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-ce-rootless-extras amd64 5:20.10.18-3-0-ubuntu-focal [8,392 kB]
Get:8 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-scan-plugin amd64 0.17.0-ubuntu-focal [3,521 kB]
Fetched 102 MB in 10s (9,797 kB/s)
```

6. Before testing Docker, check the version installed using the following command:

```
$ docker –version
```

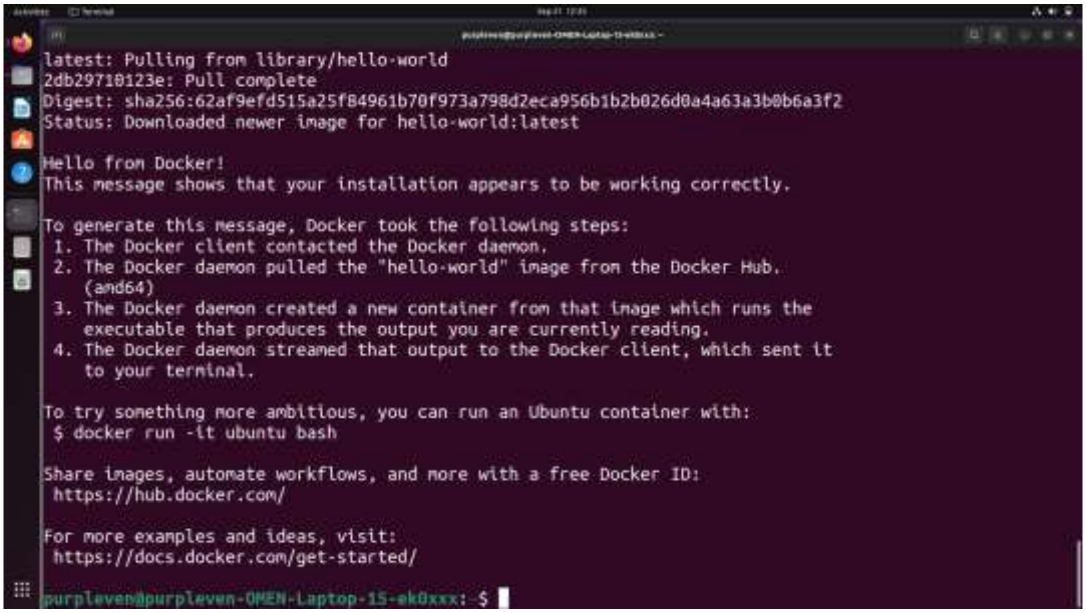


```
root@purpleven:~# docker --version
Docker version 20.10.18, build b40c2f6
root@purpleven:~#
```

7. Pull an image from the Docker hub using the following command:

```
$ sudo docker run hello-world
```

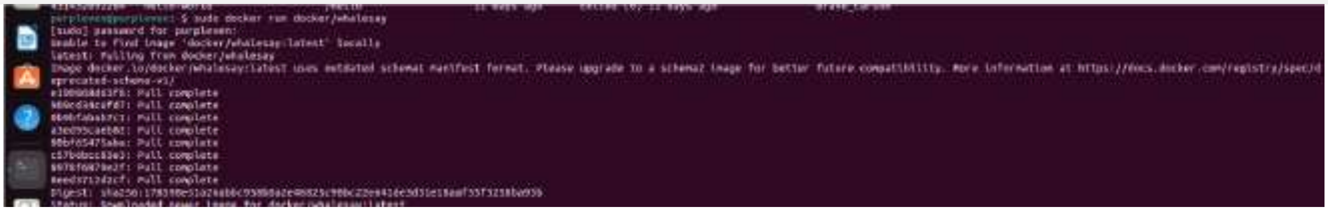
Here, *hello-world* is the docker image present on the Docker hub.



```
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:62af9efd515a25f84961b70f973a798d2eca956b1b2b026d0a4a63a3b0b6a3f2
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

purpleven@purpleven-OMEN-Laptop-15-ek0xxx:~$
```

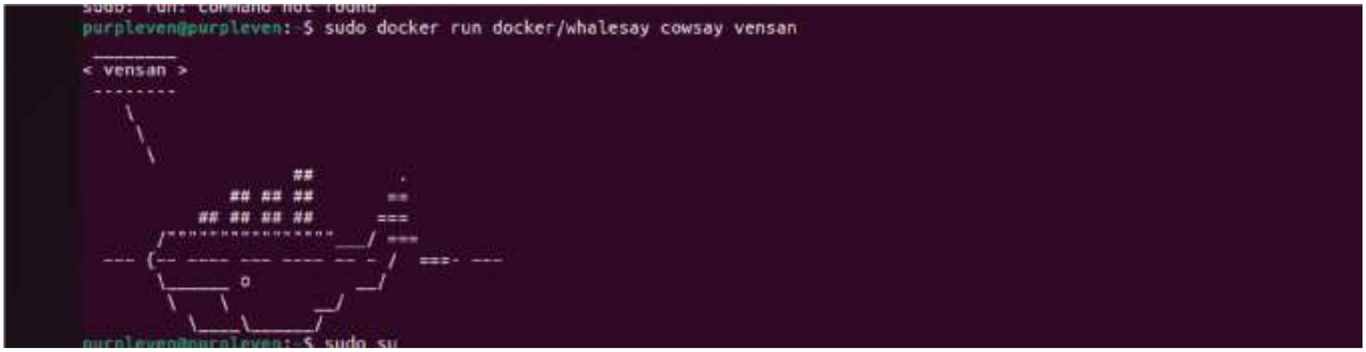
8. The actual Hello World command of docker is

```
$ docker run docker/whalesay cowsay boo
```



```
purpleven@purpleven:~$ sudo docker run docker/whalesay
[sudo] password for purpleven:
Unable to find image 'docker/whalesay:latest' locally
latest: Pulling from docker/whalesay
Image docker.io/docker/whalesay:latest uses outdated schema1 manifest format. Please upgrade to a schema2 image for better future compatibility. More information at https://docs.docker.com/registry/spec/d
eprecated-schema-v1/
e190868d63f8: Pull complete
909cd34c6fd7: Pull complete
0b9bfabab7c1: Pull complete
a3ed95caeb02: Pull complete
00bf65475aba: Pull complete
c57b6bcc83e3: Pull complete
8978f6879e2f: Pull complete
8eed3712d2cf: Pull complete
Digest: sha256:178598e51a26abbc958b8a2e48825c90bc22e641de3d31e18aaf55f3258ba93b
Status: Downloaded newer image for docker/whalesay:latest
```

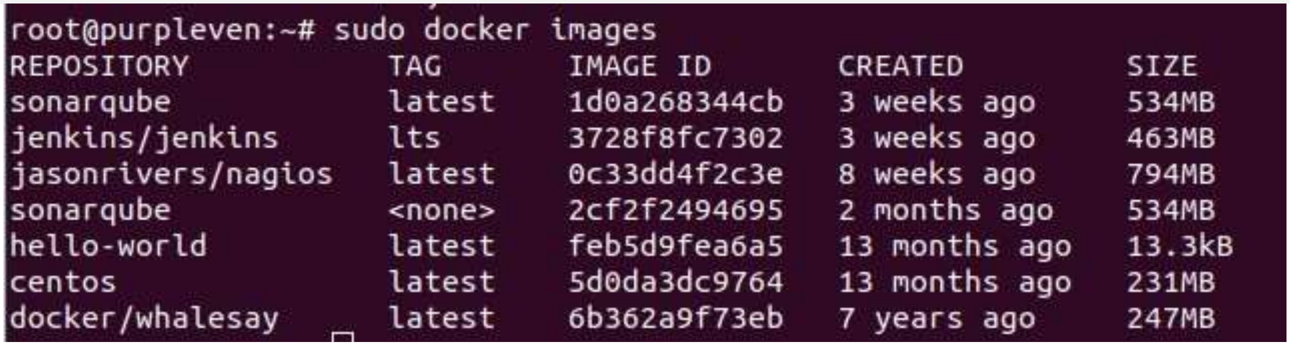
The default image of docker appears with the message boo.



```
sudo: run: command not found
purpleven@purpleven:~$ sudo docker run docker/whalesay cowsay vensan
 ________
< vensan >
 --------
    \
     \
      \
                    ##        .
              ## ## ##       ==
           ## ## ## ##      ===
       /""""""""""""""""___/ ===
  --- {-- ---- --- ---- -- - /  ===- ---
       \______ o          __/
        \    \        __/
          \____\______/
purpleven@purpleven:~$ sudo su
```

9. Check if the docker image has been pulled and is present in your system using the followingcommand:

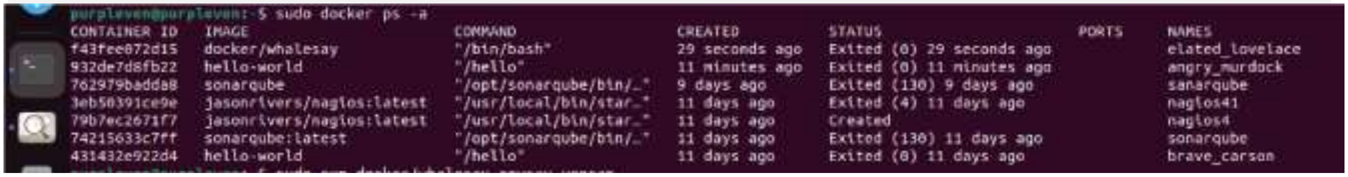
```
$ sudo docker images
```



```
root@purpleven:~# sudo docker images
REPOSITORY            TAG       IMAGE ID       CREATED          SIZE
sonarqube             latest    1d0a268344cb   3 weeks ago      534MB
jenkins/jenkins       lts       3728f8fc7302   3 weeks ago      463MB
jasonrivers/nagios    latest    0c33dd4f2c3e   8 weeks ago      794MB
sonarqube             <none>    2cf2f2494695   2 months ago     534MB
hello-world           latest    feb5d9fea6a5   13 months ago    13.3kB
centos                latest    5d0da3dc9764   13 months ago    231MB
docker/whalesay       latest    6b362a9f73eb   7 years ago      247MB
```

10. To display all the containers pulled, use the following command:

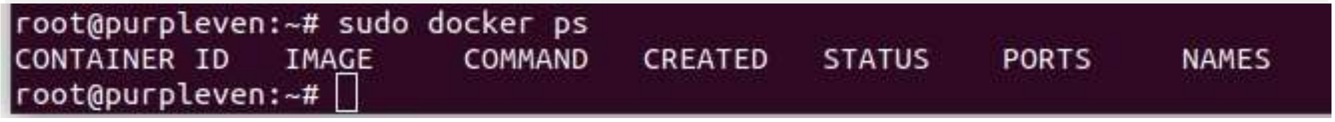
```
$ sudo docker ps -a
```



```
purpleven@purpleven:~$ sudo docker ps -a
CONTAINER ID   IMAGE                          COMMAND                  CREATED           STATUS                        PORTS     NAMES
f43fee072d15   docker/whalesay                "/bin/bash"              29 seconds ago    Exited (0) 29 seconds ago               elated_lovelace
932de7d8fb22   hello-world                    "/hello"                 11 minutes ago    Exited (0) 11 minutes ago               angry_murdock
762979badda8   sonarqube                      "/opt/sonarqube/bin/…"   9 days ago        Exited (130) 9 days ago                 sanarqube
3eb50391ce9e   jasonrivers/nagios:latest      "/usr/local/bin/star…"   11 days ago       Exited (4) 11 days ago                  nagios41
79b7ec2671f7   jasonrivers/nagios:latest      "/usr/local/bin/star…"   11 days ago       Created                                 nagios4
74215633c7ff   sonarqube:latest               "/opt/sonarqube/bin/…"   11 days ago       Exited (130) 11 days ago                sonarqube
431432e922d4   hello-world                    "/hello"                 11 days ago       Exited (0) 11 days ago                  brave_carson
```

11. To check for containers in a running state, use the following command:

$ sudo docker ps



```
root@purpleven:~# sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
root@purpleven:~# 
```

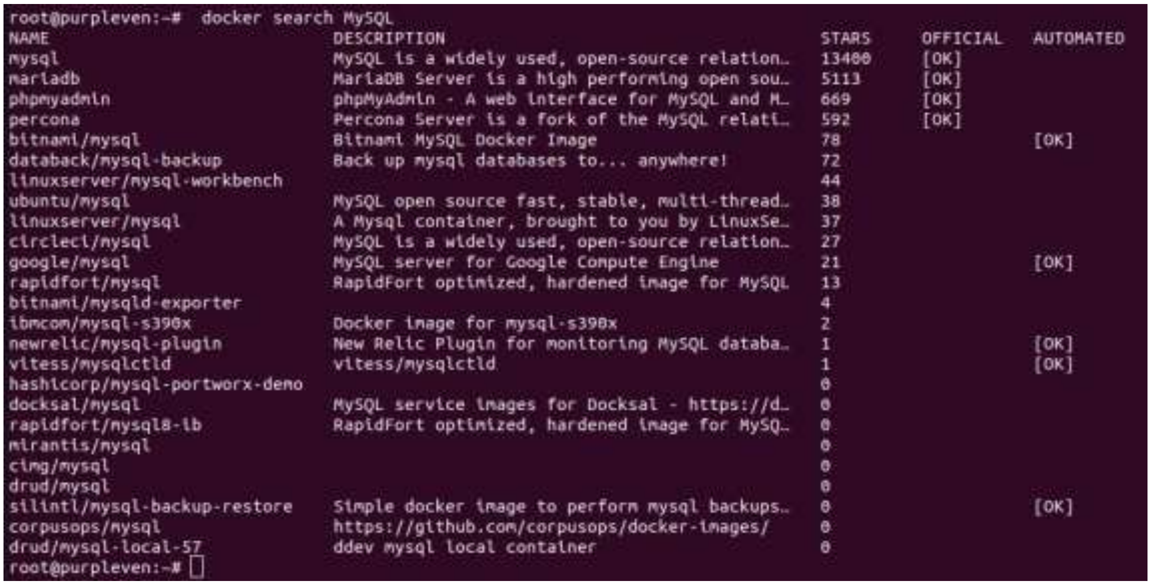
You’ve just successfully installed Docker on Ubuntu!

Part B:

1. docker search

Use the command docker search to search for public images on the Docker hub. It will return information about the image name, description, stars, official and automated.

```
docker search MySQL
```



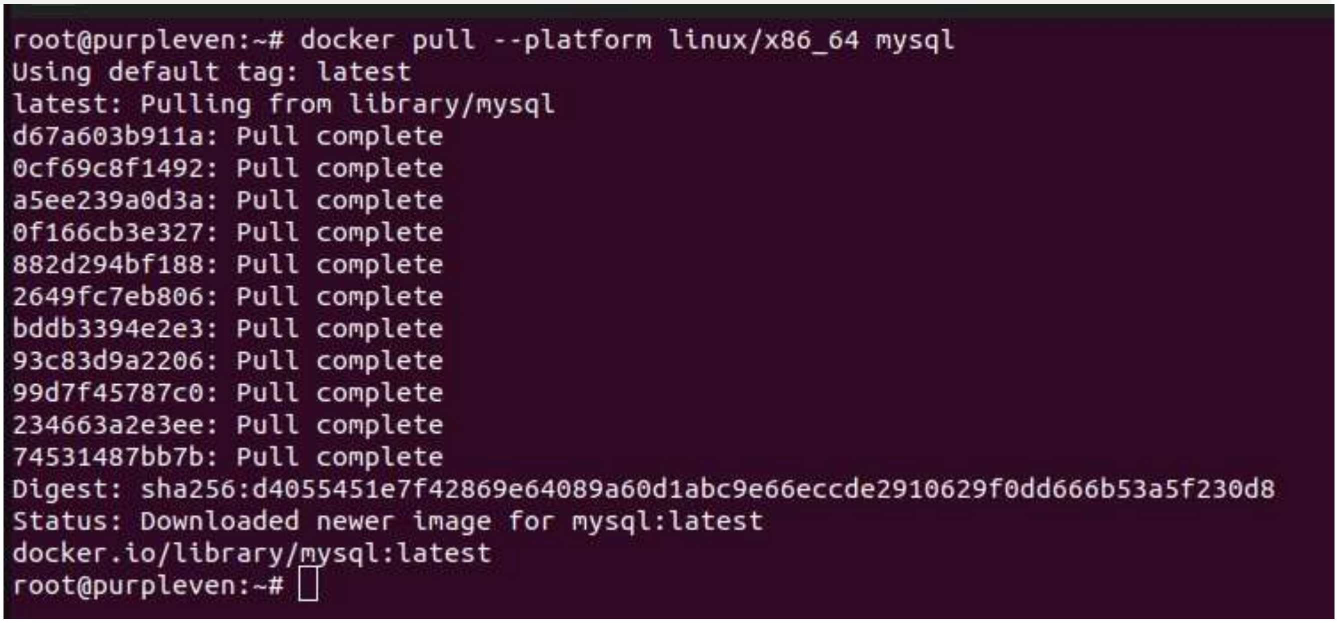
```
root@purpleven:~#  docker search MySQL
NAME                           DESCRIPTION                                      STARS     OFFICIAL   AUTOMATED
mysql                          MySQL is a widely used, open-source relation…   13400     [OK]
mariadb                        MariaDB Server is a high performing open sou…   5113      [OK]
phpmyadmin                     phpMyAdmin - A web interface for MySQL and M…   669       [OK]
percona                        Percona Server is a fork of the MySQL relati…   592       [OK]
bitnami/mysql                  Bitnami MySQL Docker Image                      78                   [OK]
databack/mysql-backup          Back up mysql databases to... anywhere!         72
linuxserver/mysql-workbench                                                    44
ubuntu/mysql                   MySQL open source fast, stable, multi-thread…   38
linuxserver/mysql              A Mysql container, brought to you by LinuxSe…   37
circleci/mysql                 MySQL is a widely used, open-source relation…   27
google/mysql                   MySQL server for Google Compute Engine          21                   [OK]
rapidfort/mysql                RapidFort optimized, hardened image for MySQL   13
bitnami/mysqld-exporter                                                        4
ibmcom/mysql-s390x             Docker image for mysql-s390x                    2
newrelic/mysql-plugin          New Relic Plugin for monitoring MySQL databa…   1                    [OK]
vitess/mysqlctld               vitess/mysqlctld                                1                    [OK]
hashicorp/mysql-portworx-demo                                                  0
docksal/mysql                  MySQL service images for Docksal - https://d…   0
rapidfort/mysql8-ib            RapidFort optimized, hardened image for MySQ…   0
mirantis/mysql                                                                 0
cimg/mysql                                                                     0
drud/mysql                                                                     0
silintl/mysql-backup-restore   Simple docker image to perform mysql backups…   0                    [OK]
corpusops/mysql                https://github.com/corpusops/docker-images/     0
drud/mysql-local-57            ddev mysql local container                      0
root@purpleven:~# 
```

If you prefer a GUI-based search option, use the Docker Hub website.

2. docker pull

Now that we know the name of the image, we can pull that from the Docker hub using the command docker pull. Here, we are setting the platform option as well.

```
docker pull --platform linux/x86_64 mysql
```



```
root@purpleven:~# docker pull --platform linux/x86_64 mysql
Using default tag: latest
latest: Pulling from library/mysql
d67a603b911a: Pull complete
0cf69c8f1492: Pull complete
a5ee239a0d3a: Pull complete
0f166cb3e327: Pull complete
882d294bf188: Pull complete
2649fc7eb806: Pull complete
bddb3394e2e3: Pull complete
93c83d9a2206: Pull complete
99d7f45787c0: Pull complete
234663a2e3ee: Pull complete
74531487bb7b: Pull complete
Digest: sha256:d4055451e7f42869e64089a60d1abc9e66eccde2910629f0dd666b53a5f230d8
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest
root@purpleven:~# 
```

Tags are used to identify images inside a repository. If we don’t specify a tag Docker engine uses the :latest tag by default. So, in the previous example, Docker pulled the mysql:latest image.

```
docker pull --platform linux/arm64/v8 mysql:5.6
```

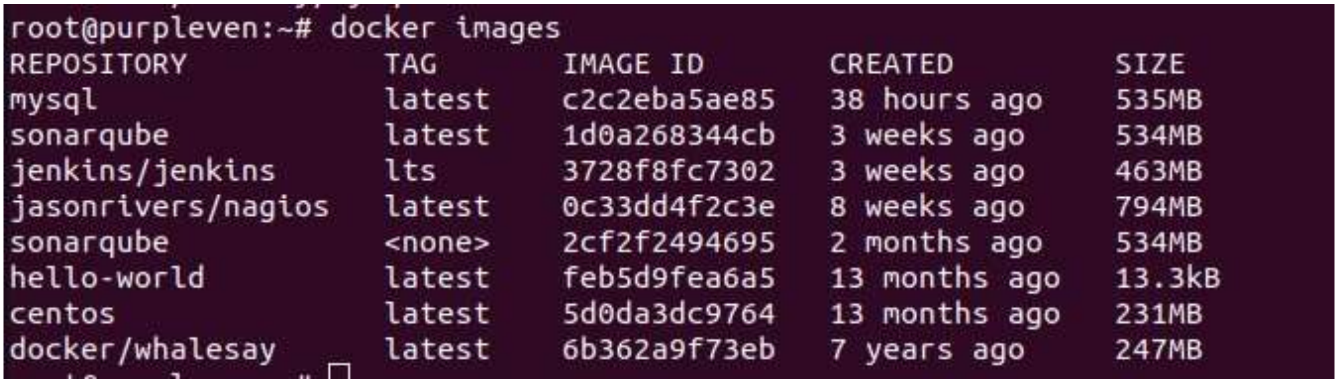
Since we can have multiple images under one repository, we can pull all the images using the --all-tags option. The following command will pull all the images from the mysql repository.

```
docker pull --all-tags mysql
```

If our application depends on a specific version of an image, we can specify that using a tag name. 3. docker image

By this time, we should have some images in our local machine, and to confirm, let’s run the following command to list all the local images.

```
docker images
```



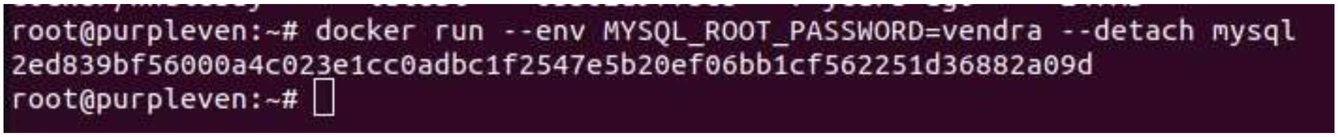
```
root@purpleven:~# docker images
REPOSITORY            TAG       IMAGE ID       CREATED          SIZE
mysql                 latest    c2c2eba5ae85   38 hours ago     535MB
sonarqube             latest    1d0a268344cb   3 weeks ago      534MB
jenkins/jenkins       lts       3728f8fc7302   3 weeks ago      463MB
jasonrivers/nagios    latest    0c33dd4f2c3e   8 weeks ago      794MB
sonarqube             <none>    2cf2f2494695   2 months ago     534MB
hello-world           latest    feb5d9fea6a5   13 months ago    13.3kB
centos                latest    5d0da3dc9764   13 months ago    231MB
docker/whalesay       latest    6b362a9f73eb   7 years ago      247MB
```

We have our recent image ‘mysql’ that we downloaded in the previous step.

4. docker run

Alright, now that we have some images, we can try to create a container. Here we used the --env option to set a mandatory environment variable and --detach option to run the container in the background.

```
docker run --env MYSQL_ROOT_PASSWORD=my-secret-pw --detach mysql
```



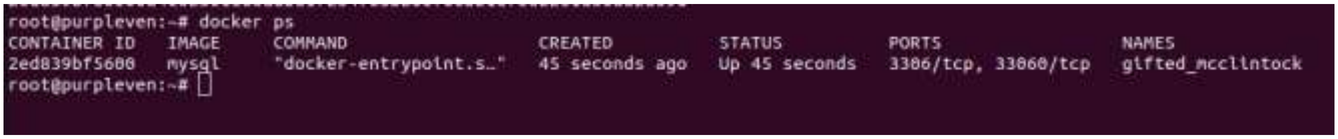
```
root@purpleven:~# docker run --env MYSQL_ROOT_PASSWORD=vendra --detach mysql
2ed839bf56000a4c023e1cc0adbc1f2547e5b20ef06bb1cf562251d36882a09d
root@purpleven:~# 
```

Moreover, we can use the --name option to assign a name to the container. Docker will randomly assign a name if we don’t provide one.

5. docker ps

We can list all the running containers by using the following command.

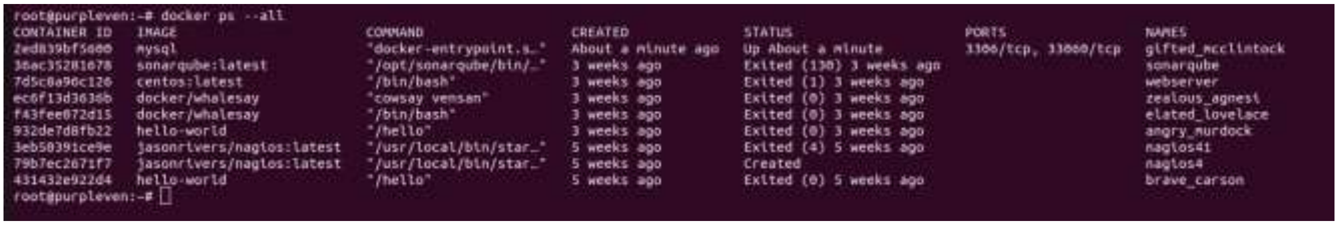
```
docker ps
```



```
root@purpleven:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                 NAMES
2ed839bf5600   mysql     "docker-entrypoint.s…"   45 seconds ago   Up 45 seconds   3306/tcp, 33060/tcp   gifted_mcclintock
root@purpleven:~# 
```

How about listing all the containers, including stopped ones? We can do that by--all option.
adding

```
docker ps –all
```

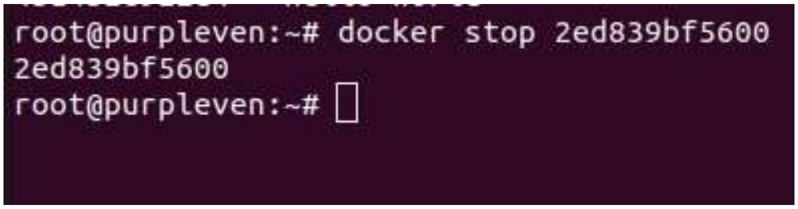


```
root@purpleven:~# docker ps --all
CONTAINER ID   IMAGE                          COMMAND                  CREATED              STATUS                        PORTS                 NAMES
2ed839bf5600   mysql                          "docker-entrypoint.s…"   About a minute ago   Up About a minute             3306/tcp, 33060/tcp   gifted_mcclintock
36ac35281678   sonarqube:latest               "/opt/sonarqube/bin/…"   3 weeks ago          Exited (130) 3 weeks ago                            sonarqube
7d5c0a96c126   centos:latest                  "/bin/bash"              3 weeks ago          Exited (1) 3 weeks ago                              webserver
ec6f13d3636b   docker/whalesay                "cowsay vensan"          3 weeks ago          Exited (0) 3 weeks ago                              zealous_agnesi
f43fee072d15   docker/whalesay                "/bin/bash"              3 weeks ago          Exited (0) 3 weeks ago                              elated_lovelace
932de7d8fb22   hello-world                    "/hello"                 3 weeks ago          Exited (0) 3 weeks ago                              angry_murdock
3eb50391ce9e   jasonrivers/nagios:latest      "/usr/local/bin/star…"   5 weeks ago          Exited (4) 5 weeks ago                              nagios41
79b7ec2671f7   jasonrivers/nagios:latest      "/usr/local/bin/star…"   5 weeks ago          Created                                             nagios4
431432e922d4   hello-world                    "/hello"                 5 weeks ago          Exited (0) 5 weeks ago                              brave_carson
root@purpleven:~# 
```

6. docker stop

To stop a container, use the docker stop command with either the container id or container name. We may stop a container if we want to change our docker run command.

```
docker stop 2ed839bf5600
```

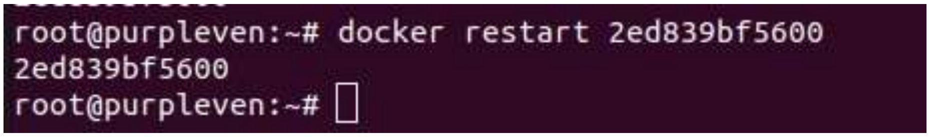


```
root@purpleven:~# docker stop 2ed839bf5600
2ed839bf5600
root@purpleven:~# 
```

7. docker restart

Let’s restart our stops by using the following command. We may want to use this after we reboot our machine.

```
docker restart 2ed839bf5600
```

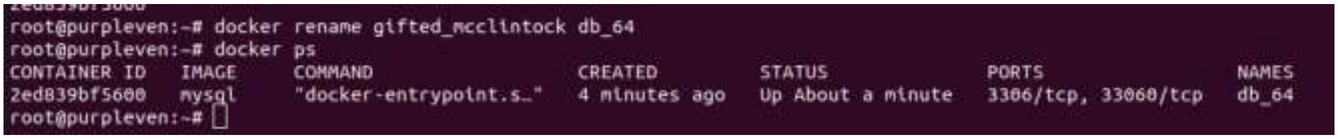


```
root@purpleven:~# docker restart 2ed839bf5600
2ed839bf5600
root@purpleven:~# 
```

8. docker rename

Now, let’s change the container name from compassionate_fermi to test_db. We may want to change the name to keep track of our containers more easily.

```
docker rename gifted_mcclintock db_64
```



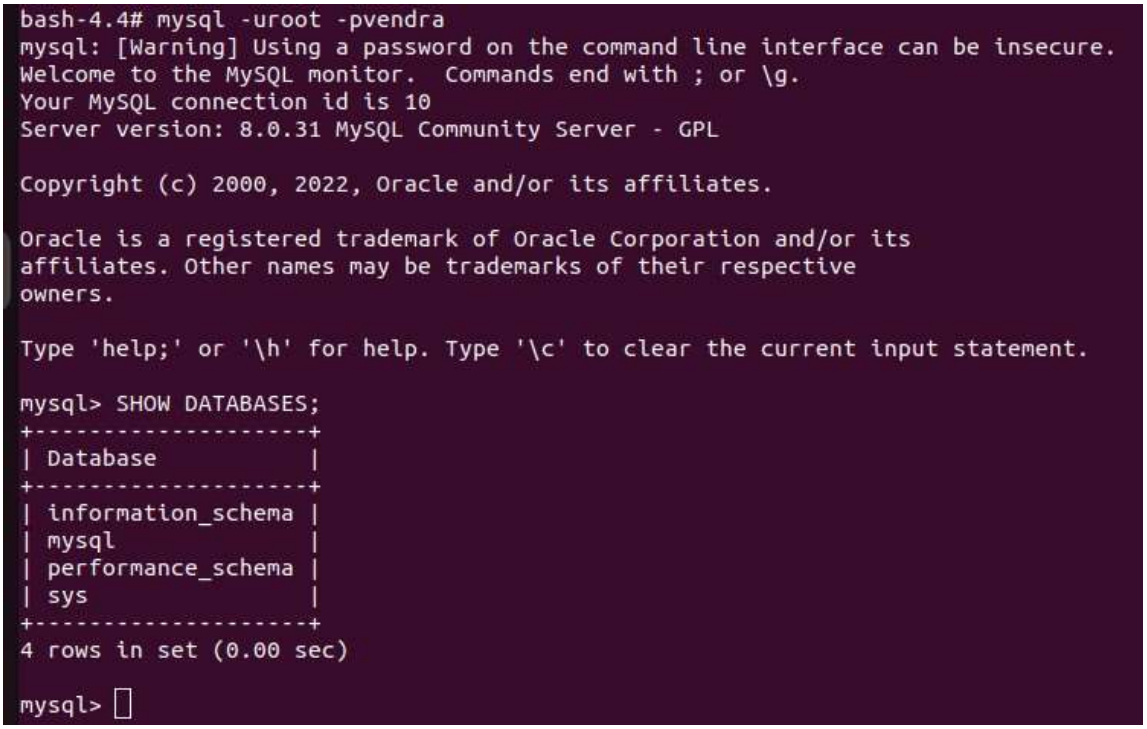
```
root@purpleven:~# docker rename gifted_mcclintock db_64
root@purpleven:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS              PORTS                 NAMES
2ed839bf5600   mysql     "docker-entrypoint.s…"   4 minutes ago   Up About a minute   3306/tcp, 33060/tcp   db_64
root@purpleven:~# 
```

9. docker exec

Access the running container test_db by running the following command. It’s helpful if we want to access the MySQL command line and execute MySQL queries.

```
docker exec -it db_64 bash

mysql -uroot -pmy-secret-pw

SHOW DATABASES;
```

```
root@purpleven:~# docker exec -it db_64 bash
```



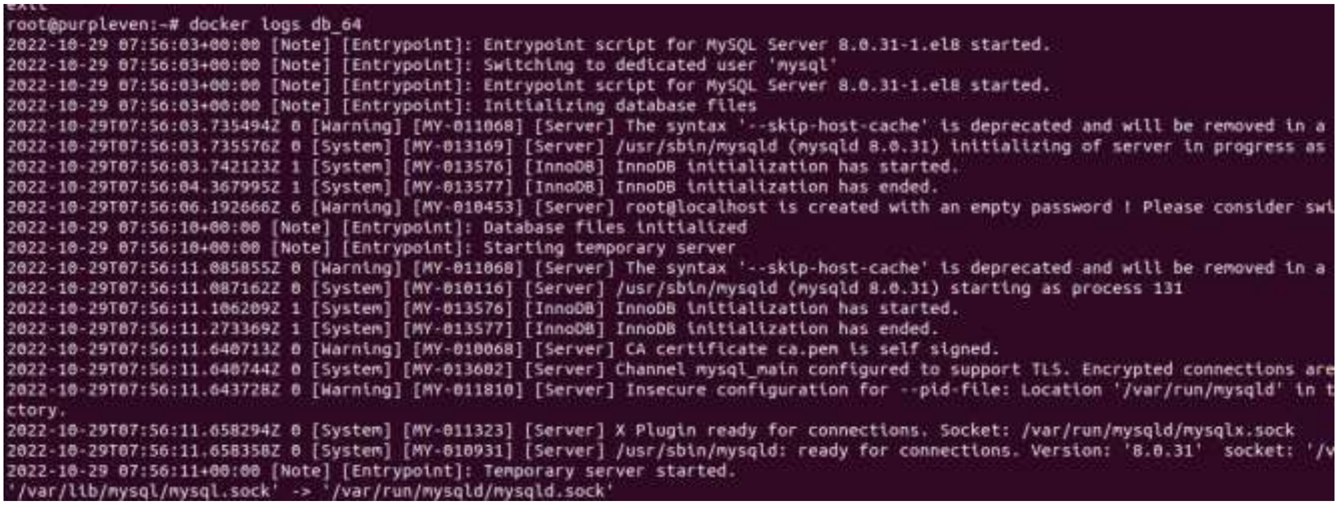
```
bash-4.4# mysql -uroot -pvendra
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.31 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.00 sec)

mysql> 
```

The -i and -t options are used to access the container in an interactive mode. Then we provide the name of the container we want to access, which in this case test_db. Finally, the bash command is used to get a bash shell inside the container.

10. docker logs

This command is helpful to debug our Docker containers. It will fetch logs from a specified container.

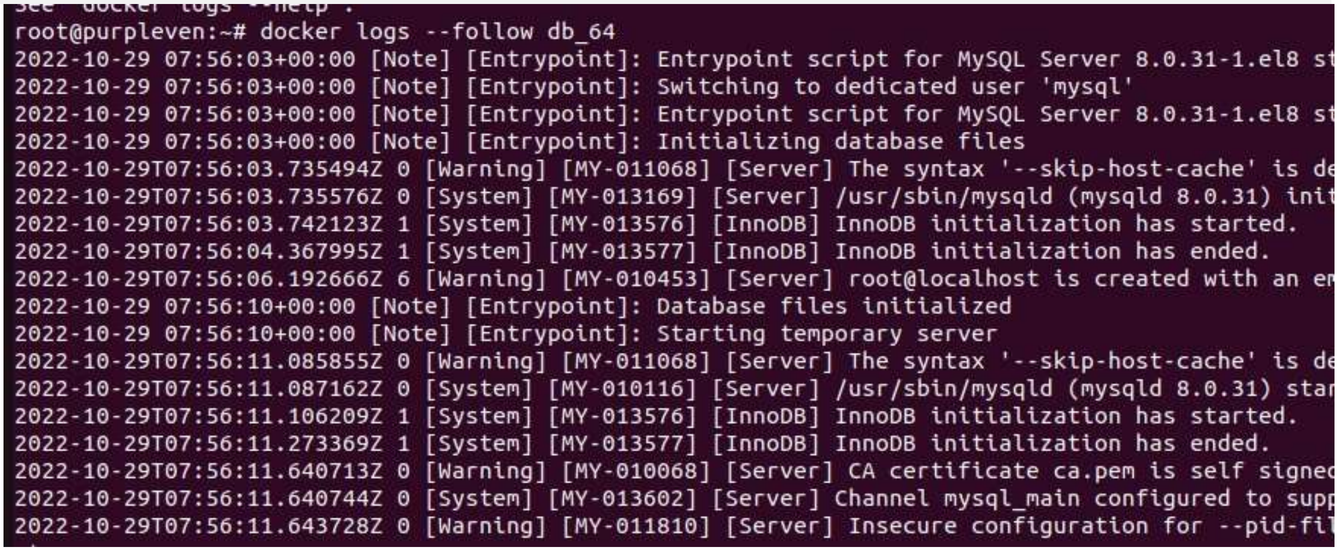
```
docker logs db_64
```



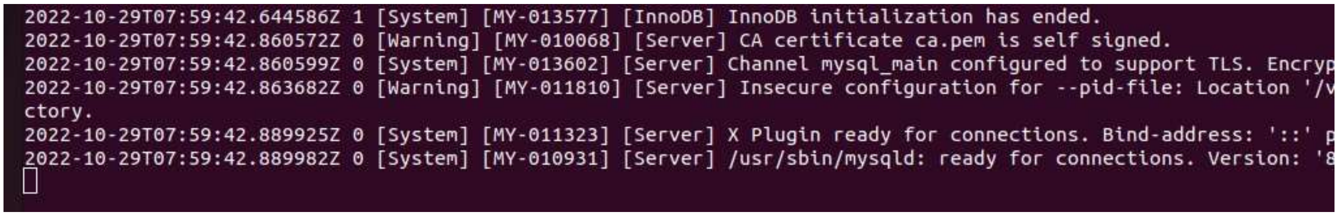
```
exit
root@purpleven:~# docker logs db_64
2022-10-29 07:56:03+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.31-1.el8 started.
2022-10-29 07:56:03+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2022-10-29 07:56:03+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.31-1.el8 started.
2022-10-29 07:56:03+00:00 [Note] [Entrypoint]: Initializing database files
2022-10-29T07:56:03.735494Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is deprecated and will be removed in a
2022-10-29T07:56:03.735576Z 0 [System] [MY-013169] [Server] /usr/sbin/mysqld (mysqld 8.0.31) initializing of server in progress as
2022-10-29T07:56:03.742123Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2022-10-29T07:56:04.367995Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2022-10-29T07:56:06.192666Z 6 [Warning] [MY-010453] [Server] root@localhost is created with an empty password ! Please consider swi
2022-10-29 07:56:10+00:00 [Note] [Entrypoint]: Database files initialized
2022-10-29 07:56:10+00:00 [Note] [Entrypoint]: Starting temporary server
2022-10-29T07:56:11.085855Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is deprecated and will be removed in a
2022-10-29T07:56:11.087162Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.31) starting as process 131
2022-10-29T07:56:11.106209Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2022-10-29T07:56:11.273369Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2022-10-29T07:56:11.640713Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
2022-10-29T07:56:11.640744Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. Encrypted connections are
2022-10-29T07:56:11.643728Z 0 [Warning] [MY-011810] [Server] Insecure configuration for --pid-file: Location '/var/run/mysqld' in t
ctory.
2022-10-29T07:56:11.658294Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Socket: /var/run/mysqld/mysqlx.sock
2022-10-29T07:56:11.658358Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version: '8.0.31'  socket: '/v
2022-10-29 07:56:11+00:00 [Note] [Entrypoint]: Temporary server started.
'/var/lib/mysql/mysql.sock' -> '/var/run/mysqld/mysqld.sock'
```

If we want to continue to stream new output,use the option -follow.

```
docker logs -follow db_64
```



```
See 'docker logs --help'.
root@purpleven:~# docker logs --follow db_64
2022-10-29 07:56:03+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.31-1.el8 st
2022-10-29 07:56:03+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2022-10-29 07:56:03+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.31-1.el8 st
2022-10-29 07:56:03+00:00 [Note] [Entrypoint]: Initializing database files
2022-10-29T07:56:03.735494Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is de
2022-10-29T07:56:03.735576Z 0 [System] [MY-013169] [Server] /usr/sbin/mysqld (mysqld 8.0.31) init
2022-10-29T07:56:03.742123Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2022-10-29T07:56:04.367995Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2022-10-29T07:56:06.192666Z 6 [Warning] [MY-010453] [Server] root@localhost is created with an em
2022-10-29 07:56:10+00:00 [Note] [Entrypoint]: Database files initialized
2022-10-29 07:56:10+00:00 [Note] [Entrypoint]: Starting temporary server
2022-10-29T07:56:11.085855Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is de
2022-10-29T07:56:11.087162Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.31) star
2022-10-29T07:56:11.106209Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2022-10-29T07:56:11.273369Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2022-10-29T07:56:11.640713Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed
2022-10-29T07:56:11.640744Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to supp
2022-10-29T07:56:11.643728Z 0 [Warning] [MY-011810] [Server] Insecure configuration for --pid-fil
```



```
2022-10-29T07:59:42.644586Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2022-10-29T07:59:42.860572Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
2022-10-29T07:59:42.860599Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. Encryp
2022-10-29T07:59:42.863682Z 0 [Warning] [MY-011810] [Server] Insecure configuration for --pid-file: Location '/v
ctory.
2022-10-29T07:59:42.889925Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Bind-address: '::' p
2022-10-29T07:59:42.889982Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version: '8
```

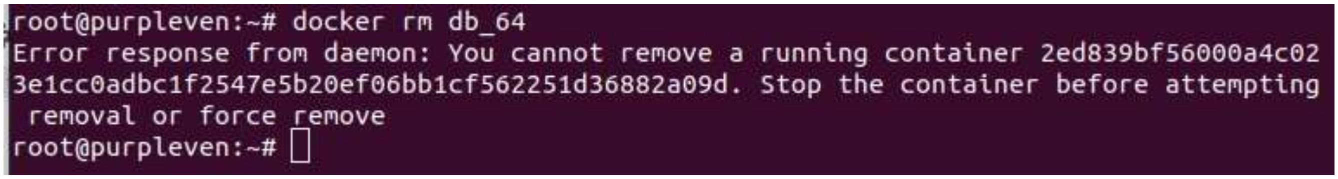
11. docker rm

To remove a container, we can use the following command.

```
docker rm db_64
```

You may encounter an error like

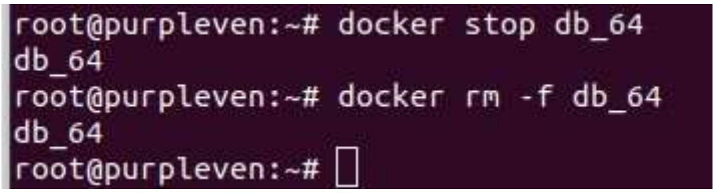
Error response from daemon: You cannot remove a running container ……… Stop the container before attempting removal or force remove



```
root@purpleven:~# docker rm db_64
Error response from daemon: You cannot remove a running container 2ed839bf56000a4c02
3e1cc0adbc1f2547e5b20ef06bb1cf562251d36882a09d. Stop the container before attempting
 removal or force remove
root@purpleven:~#
```

As it recommends, we can stop the container first and then remove it or use option -f to remove a running container forcefully.

```
docker stop db_64
docker rm db_64# or docker rm -f db_64
```

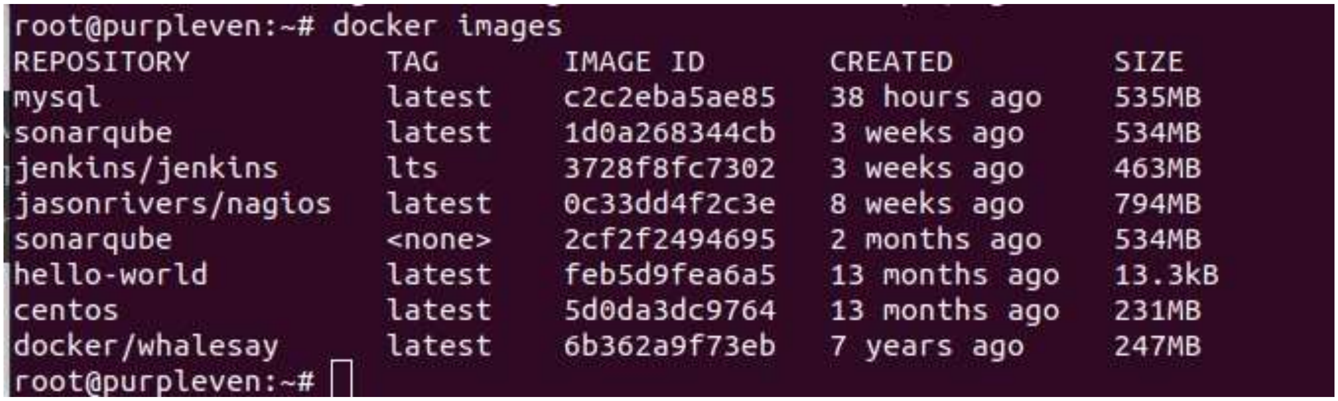


```
root@purpleven:~# docker stop db_64
db_64
root@purpleven:~# docker rm -f db_64
db_64
root@purpleven:~# 
```

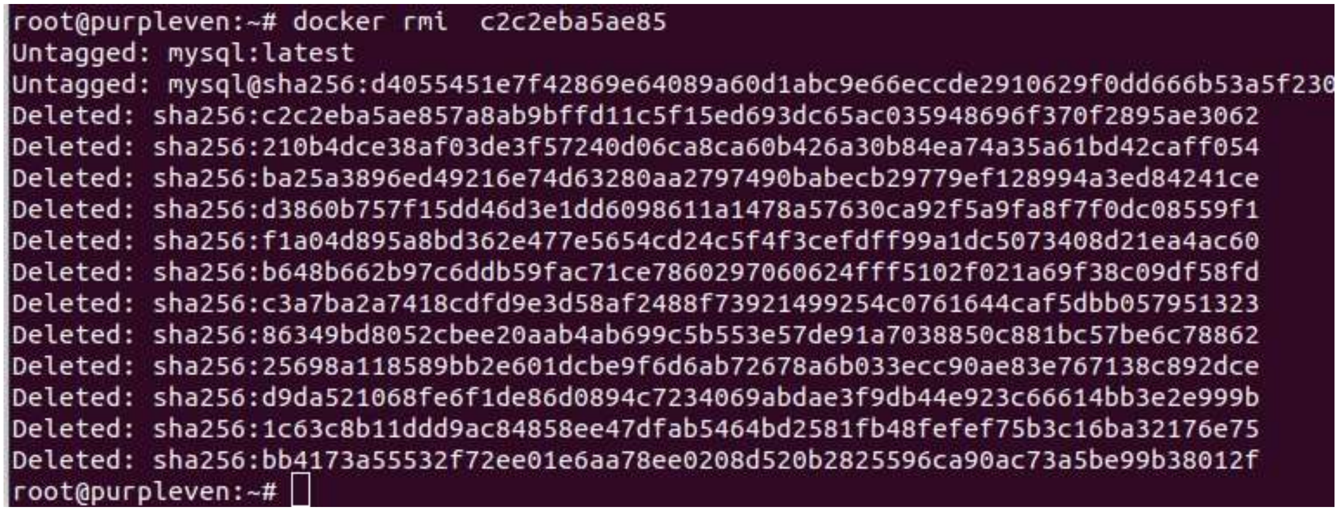
12. docker rmi

To free some disk space, we can usedocker rmi command with the image id to remove an image. the

```
docker rmi eb0e825dc3cf
```



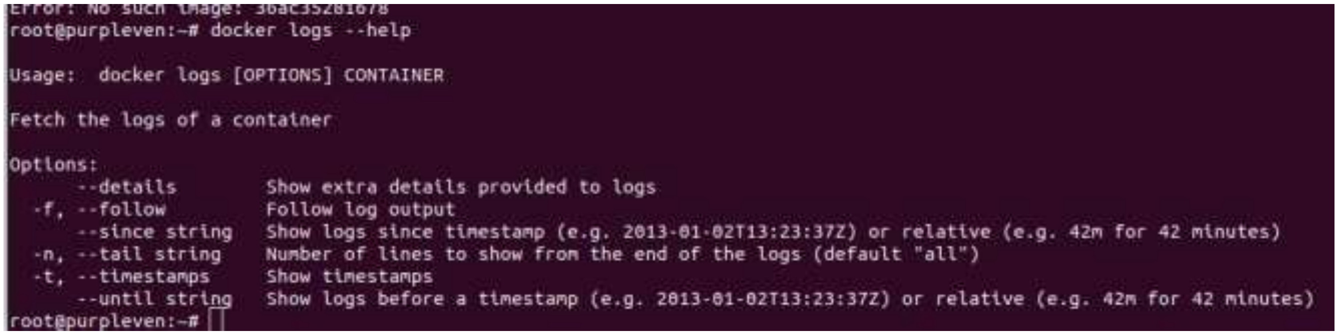
```
root@purpleven:~# docker images
REPOSITORY           TAG       IMAGE ID       CREATED          SIZE
mysql                latest    c2c2eba5ae85   38 hours ago     535MB
sonarqube            latest    1d0a268344cb   3 weeks ago      534MB
jenkins/jenkins      lts       3728f8fc7302   3 weeks ago      463MB
jasonrivers/nagios   latest    0c33dd4f2c3e   8 weeks ago      794MB
sonarqube            <none>    2cf2f2494695   2 months ago     534MB
hello-world          latest    feb5d9fea6a5   13 months ago    13.3kB
centos               latest    5d0da3dc9764   13 months ago    231MB
docker/whalesay      latest    6b362a9f73eb   7 years ago      247MB
root@purpleven:~# 
```



```
root@purpleven:~# docker rmi  c2c2eba5ae85
Untagged: mysql:latest
Untagged: mysql@sha256:d4055451e7f42869e64089a60d1abc9e66eccde2910629f0dd666b53a5f230
Deleted: sha256:c2c2eba5ae857a8ab9bffd11c5f15ed693dc65ac035948696f370f2895ae3062
Deleted: sha256:210b4dce38af03de3f57240d06ca8ca60b426a30b84ea74a35a61bd42caff054
Deleted: sha256:ba25a3896ed49216e74d63280aa2797490babecb29779ef128994a3ed84241ce
Deleted: sha256:d3860b757f15dd46d3e1dd6098611a1478a57630ca92f5a9fa8f7f0dc08559f1
Deleted: sha256:f1a04d895a8bd362e477e5654cd24c5f4f3cefdff99a1dc5073408d21ea4ac60
Deleted: sha256:b648b662b97c6ddb59fac71ce7860297060624fff5102f021a69f38c09df58fd
Deleted: sha256:c3a7ba2a7418cdfd9e3d58af2488f73921499254c0761644caf5dbb057951323
Deleted: sha256:86349bd8052cbee20aab4ab699c5b553e57de91a7038850c881bc57be6c78862
Deleted: sha256:25698a118589bb2e601dcbe9f6d6ab72678a6b033ecc90ae83e767138c892dce
Deleted: sha256:d9da521068fe6f1de86d0894c7234069abdae3f9db44e923c66614bb3e2e999b
Deleted: sha256:1c63c8b11ddd9ac84858ee47dfab5464bd2581fb48fefef75b3c16ba32176e75
Deleted: sha256:bb4173a55532f72ee01e6aa78ee0208d520b2825596ca90ac73a5be99b38012f
root@purpleven:~#
```

These commands come with plenty of helpful options. If you want to know about other available options, run the docker command_name --help command. For example:

docker logs --help



```
Error: No such image: 36ac35281678
root@purpleven:~# docker logs --help

Usage:  docker logs [OPTIONS] CONTAINER

Fetch the logs of a container

Options:
      --details        Show extra details provided to logs
  -f, --follow         Follow log output
      --since string   Show logs since timestamp (e.g. 2013-01-02T13:23:37Z) or relative (e.g. 42m for 42 minutes)
  -n, --tail string    Number of lines to show from the end of the logs (default "all")
  -t, --timestamps     Show timestamps
      --until string   Show logs before a timestamp (e.g. 2013-01-02T13:23:37Z) or relative (e.g. 42m for 42 minutes)
root@purpleven:~#
```

Part C: Docker Container Commands:

Containers

Use docker container my_command

create — Create a container from an image.

start — Start an existing container. run —

Create a new container and start it.

ls — List running containers. inspect — See

lots of info about a container.

logs — Print logs.

stop — Gracefully stop running container. kill

—Stop main process in container abruptly.

rm— Delete a stopped container.

Images

Use docker image my_command

build — Build an image.

push — Push an image to a remote registry.

ls — List images. history — See

intermediate image info. inspect —

See lots of info about an image,

including the layers. rm — Delete an

image.

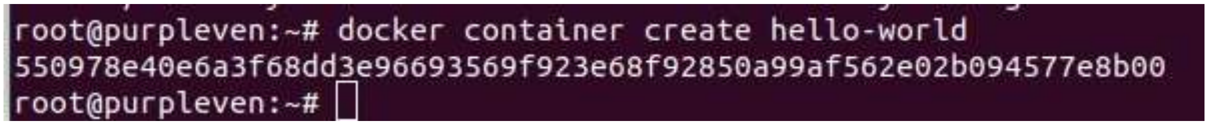
Misc docker version — List info about your Docker Client and Server versions. docker login — Log in to a Docker registry. docker system prune — Delete all unused containers, unused networks, and dangling images.

Containers

Container Beginnings

The terms create, start, and run all have similar semantics in everyday life. But each is a separate Docker command that creates and/or starts a container. Let's look at creating a container first.

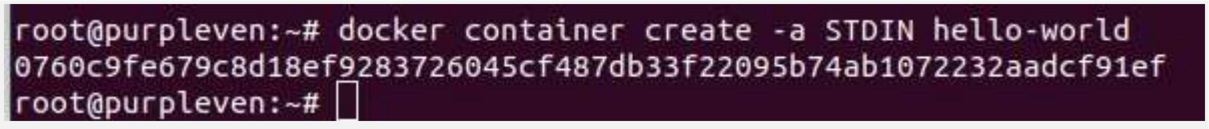
docker container create hello-world — Create a container from an image.



```
root@purpleven:~# docker container create hello-world
550978e40e6a3f68dd3e96693569f923e68f92850a99af562e02b094577e8b00
root@purpleven:~# 
```

I'll shorten my_repo/my_image:my_tag to my_image for the rest of the article.

There are a lot of possible flags you could pass to create. docker container create -a STDIN my_image - docker container create -a STDIN hello-world

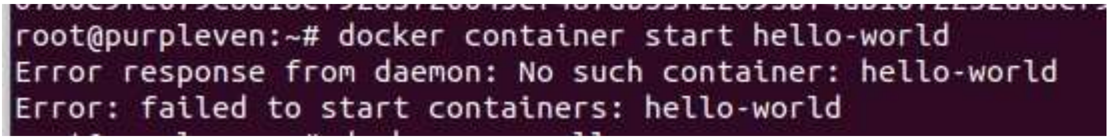


```
root@purpleven:~# docker container create -a STDIN hello-world
0760c9fe679c8d18ef9283726045cf487db33f22095b74ab1072232aadcf91ef
root@purpleven:~# 
```

-a is short for --attach. Attach the container to STDIN, STDOUT or STDERR.

Now that we've created a container let's start it.

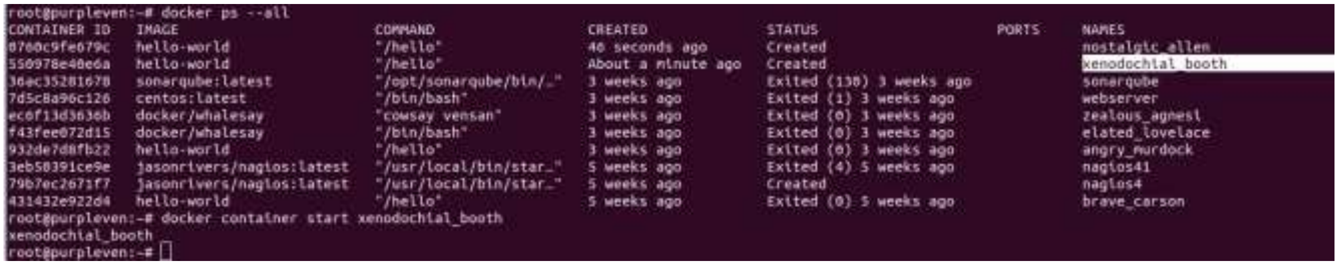
docker container start hello-world — Start an existing container.



```
root@purpleven:~# docker container start hello-world
Error response from daemon: No such container: hello-world
Error: failed to start containers: hello-world
```

Note that the container can be referred to by either the container's ID or the container's name.

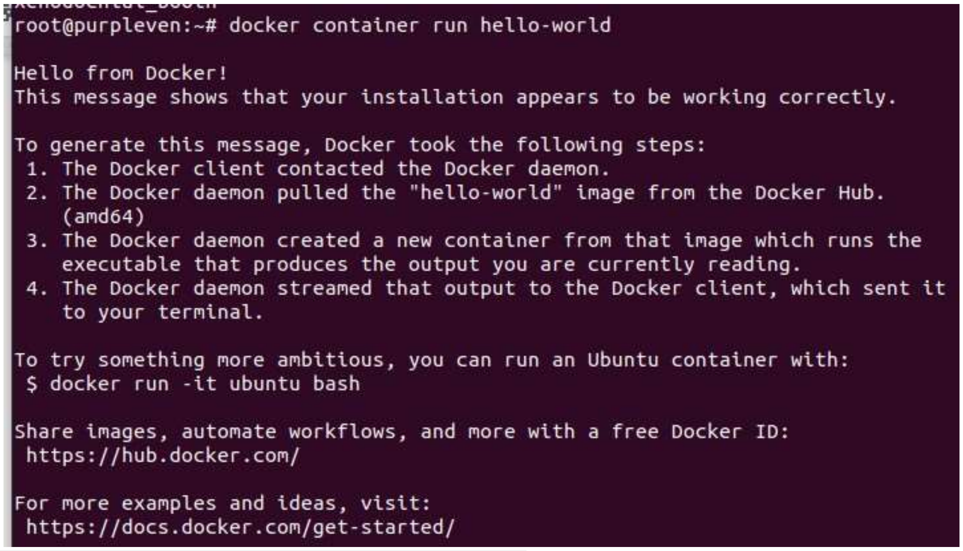
docker container start xenodochial_booth



```
root@purpleven:~# docker ps --all
CONTAINER ID   IMAGE                          COMMAND                  CREATED              STATUS                      PORTS     NAMES
8760c9fe679c   hello-world                    "/hello"                 40 seconds ago       Created                               nostalgic_allen
550978e40e6a   hello-world                    "/hello"                 About a minute ago   Created                               xenodochial_booth
36ac35281678   sonarqube:latest               "/opt/sonarqube/bin/…"   3 weeks ago          Exited (130) 3 weeks ago              sonarqube
7d5c8a96c126   centos:latest                  "/bin/bash"              3 weeks ago          Exited (1) 3 weeks ago                webserver
ec6f13d3636b   docker/whalesay                "cowsay vensan"          3 weeks ago          Exited (0) 3 weeks ago                zealous_agnesi
f43fee072d15   docker/whalesay                "/bin/bash"              3 weeks ago          Exited (0) 3 weeks ago                elated_lovelace
932de7d8fb22   hello-world                    "/hello"                 3 weeks ago          Exited (0) 3 weeks ago                angry_murdock
3eb58391ce9e   jasonrivers/nagios:latest      "/usr/local/bin/star…"   5 weeks ago          Exited (4) 5 weeks ago                nagios41
79b7ec2671f7   jasonrivers/nagios:latest      "/usr/local/bin/star…"   5 weeks ago          Created                               nagios4
431432e922d4   hello-world                    "/hello"                 5 weeks ago          Exited (0) 5 weeks ago                brave_carson
root@purpleven:~# docker container start xenodochial_booth
xenodochial_booth
root@purpleven:~# 
```

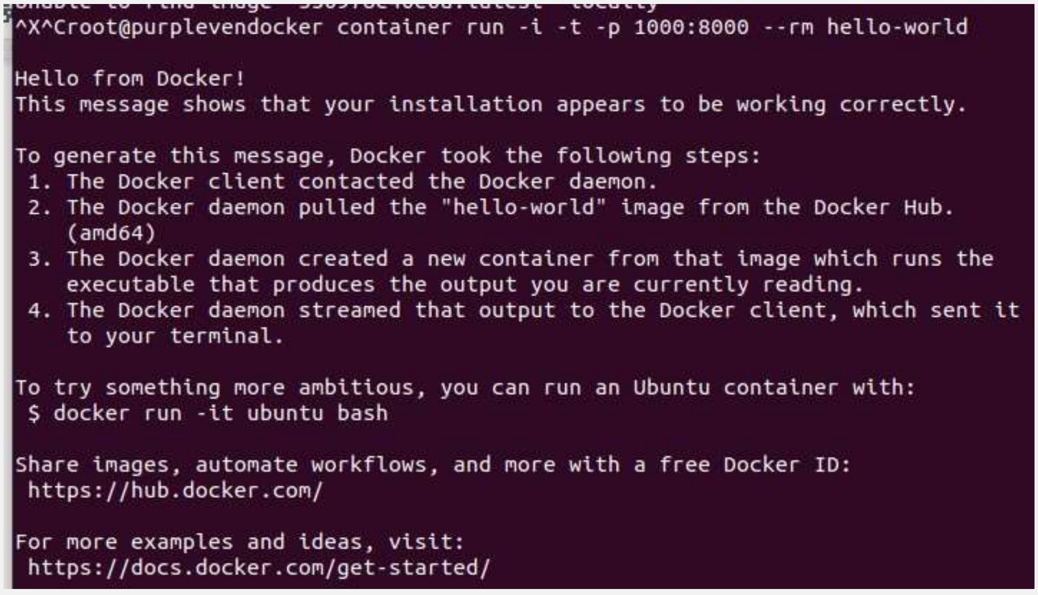
Now that you know how to create and start a container, let’s turn to what’s probably the most common Docker command. It combines both create and start into one command: run.

docker container run my_image — Create a new container and start it. It also has a lot of options. Let’s look at a few.



```
root@purpleven:~# docker container run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

docker container run -i -t -p 1000:8000 --rm my_image



```
Unable to find image '5305f6c4ec0d:latest' locally
^X^Croot@purplevendocker container run -i -t -p 1000:8000 --rm hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

-i is short for --interactive. Keep STDIN open even if unattached.

-tis short for--tty. Allocates a pseudo terminal that connects your terminal with the container’s STDIN and STDOUT.

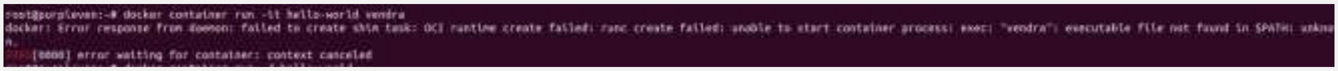
You need to specify both -i and -t to then interact with the container through your terminal shell.

-p is short for --port. The port is the interface with the outside world.1000:8000 maps the Docker port 8000 to port 1000 on your machine. If you had an app that output something to the browser you could then navigate your browser to localhost:1000 and see it.

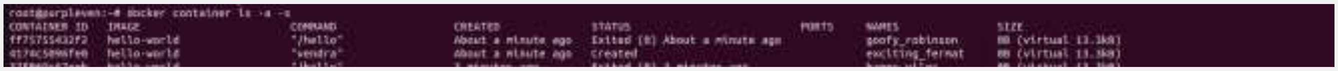
--rm Automatically delete the container when it stops running.

Let’s look at some more examples of run.

docker container run -it my_image my_command



```
root@purpleven:~# docker container run -it hello-world vendra
docker: Error response from daemon: failed to create shim task: OCI runtime create failed: runc create failed: unable to start container process: exec: "vendra": executable file not found in $PATH: unkno
n.
ERRO[0000] error waiting for container: context canceled
```

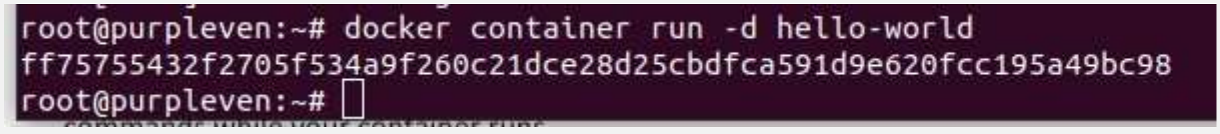


```
root@purpleven:~# docker container ls -a -s
CONTAINER ID   IMAGE          COMMAND      CREATED              STATUS                          PORTS    NAMES              SIZE
ff7575S432f2   hello-world    "/hello"     About a minute ago   Exited (0) About a minute ago            goofy_robinson     0B (virtual 13.3kB)
0174c5096fe0   hello-world    "vendra"     About a minute ago   Created                                  exciting_fermat    0B (virtual 13.3kB)
```

sh is a command you could specify at run time.sh will start a shell session inside your container that you can interact with through your terminal. sh is preferable to bash for Alpine images because Alpine images don’t come with bash installed. Type exit to end the interactive shell session.

Notice that we combined -i and -t into -it.

docker container run -d my_image

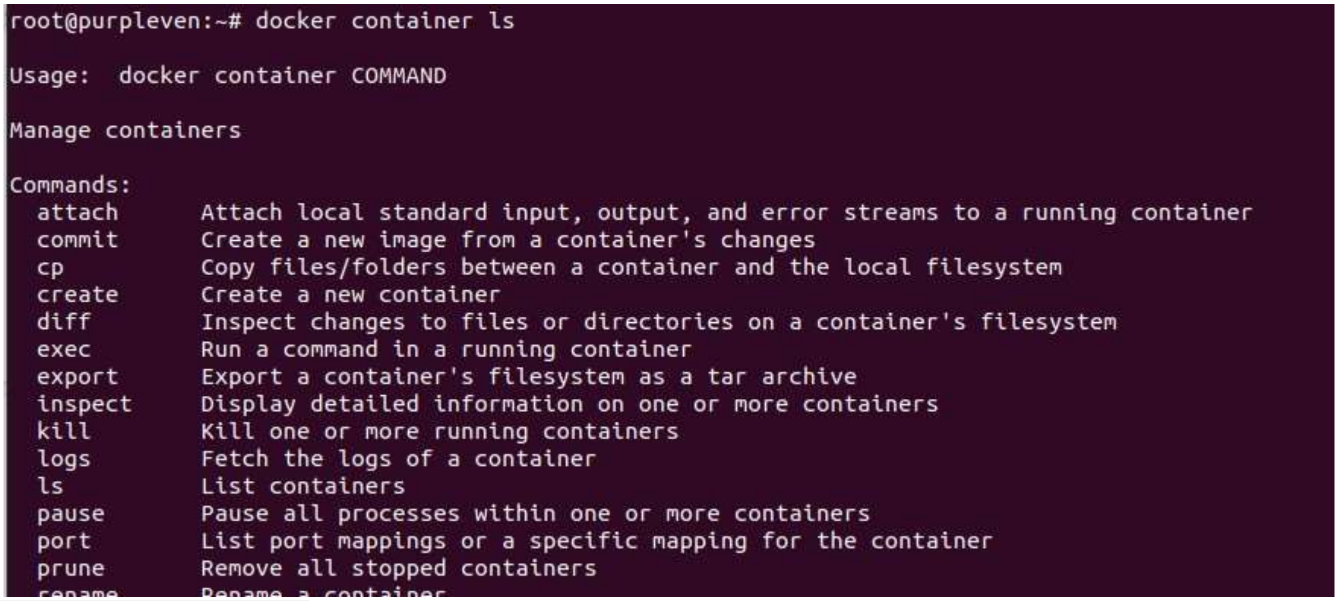


```
root@purpleven:~# docker container run -d hello-world
ff75755432f2705f534a9f260c21dce28d25cbdfca591d9e620fcc195a49bc98
root@purpleven:~# 
```

-d is short for --detach. Run the container in the background. Allows you to use the terminal for other commands while your container runs.

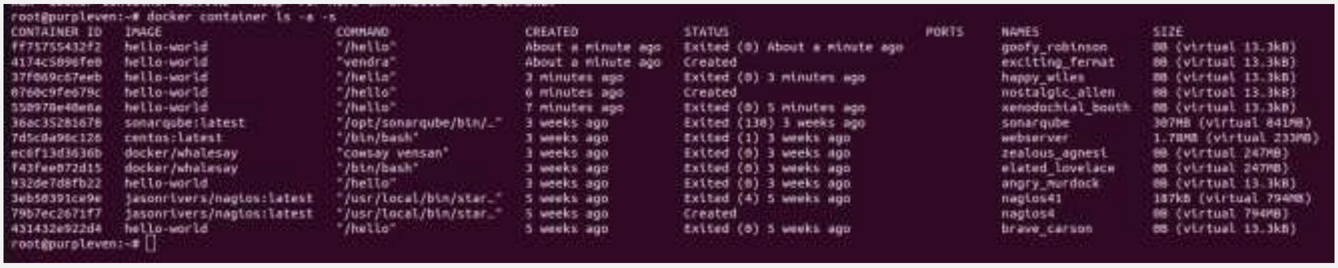
Checking Container Status

If you have running Docker containers and want to find out which one to interact with, then you need to list them. docker container ls — List running containers. Also provides useful information about the containers.



```
root@purpleven:~# docker container ls

Usage:  docker container COMMAND

Manage containers

Commands:
  attach      Attach local standard input, output, and error streams to a running container
  commit      Create a new image from a container's changes
  cp          Copy files/folders between a container and the local filesystem
  create      Create a new container
  diff        Inspect changes to files or directories on a container's filesystem
  exec        Run a command in a running container
  export      Export a container's filesystem as a tar archive
  inspect     Display detailed information on one or more containers
  kill        Kill one or more running containers
  logs        Fetch the logs of a container
  ls          List containers
  pause       Pause all processes within one or more containers
  port        List port mappings or a specific mapping for the container
  prune       Remove all stopped containers
  rename      Rename a container
```

docker container ls -a -s

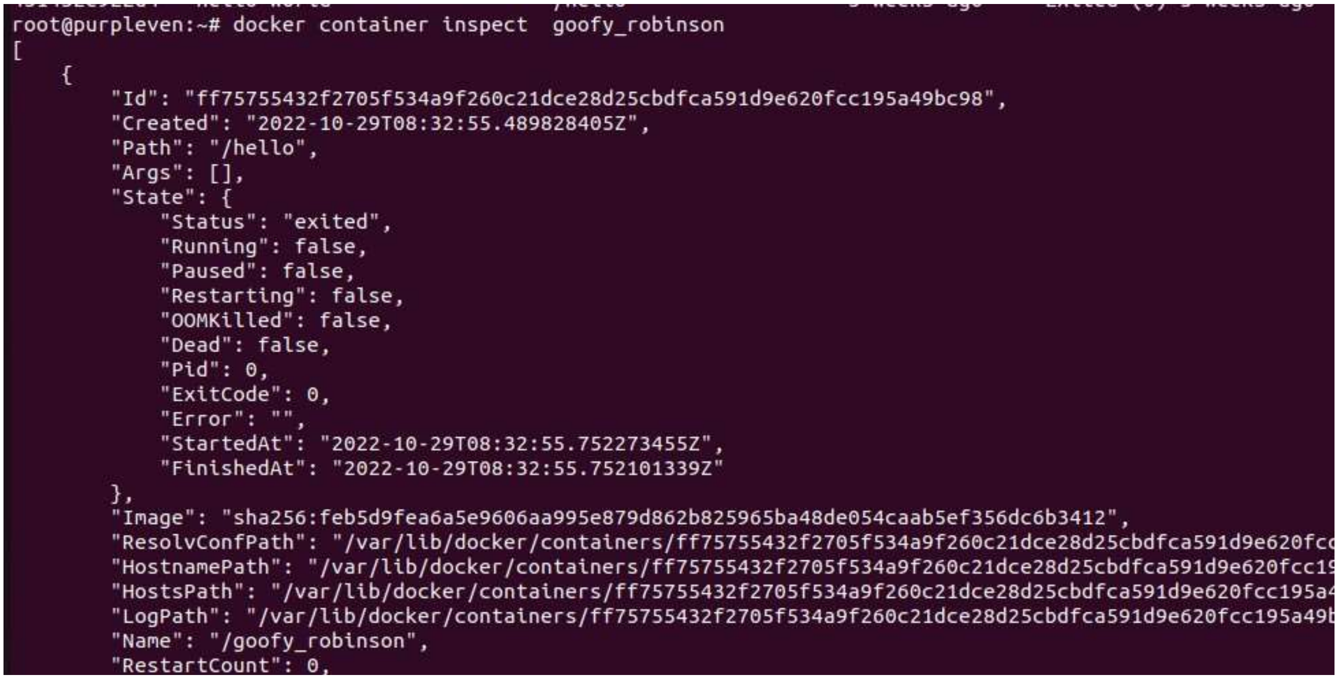


```
root@purpleven:~# docker container ls -a -s
CONTAINER ID   IMAGE                        COMMAND                  CREATED              STATUS                          PORTS     NAMES                SIZE
ff75755432f2   hello-world                  "/hello"                 About a minute ago   Exited (0) About a minute ago             goofy_robinson       0B (virtual 13.3kB)
4174c5096fe0   hello-world                  "vendra"                 About a minute ago   Created                                   exciting_fermat      0B (virtual 13.3kB)
37f069c67eeb   hello-world                  "/hello"                 3 minutes ago        Exited (0) 3 minutes ago                  happy_wiles          0B (virtual 13.3kB)
0760c9fe079c   hello-world                  "/hello"                 6 minutes ago        Created                                   nostalgic_allen      0B (virtual 13.3kB)
550970e40e6a   hello-world                  "/hello"                 7 minutes ago        Exited (0) 5 minutes ago                  xenodochial_booth    0B (virtual 13.3kB)
36ac35281676   sonarqube:latest             "/opt/sonarqube/bin/…"   3 weeks ago          Exited (130) 3 weeks ago                  sonarqube            307MB (virtual 841MB)
7d5c0a96c126   centos:latest                "/bin/bash"              3 weeks ago          Exited (1) 3 weeks ago                    webserver            1.78MB (virtual 233MB)
ec0f13d3636b   docker/whalesay              "cowsay vensan"          3 weeks ago          Exited (0) 3 weeks ago                    zealous_agnesi       0B (virtual 247MB)
f43fee072d15   docker/whalesay              "/bin/bash"              3 weeks ago          Exited (0) 3 weeks ago                    elated_lovelace      0B (virtual 247MB)
932de7d8fb22   hello-world                  "/hello"                 3 weeks ago          Exited (0) 3 weeks ago                    angry_murdock        0B (virtual 13.3kB)
3eb50391ce9e   jasonrivers/nagios:latest    "/usr/local/bin/star…"   5 weeks ago          Exited (4) 5 weeks ago                    nagios41             187kB (virtual 794MB)
79b7ec2671f7   jasonrivers/nagios:latest    "/usr/local/bin/star…"   5 weeks ago          Created                                   nagios4              0B (virtual 794MB)
431432e922d4   hello-world                  "/hello"                 5 weeks ago          Exited (0) 5 weeks ago                    brave_carson         0B (virtual 13.3kB)
root@purpleven:~# 
```

-a is short for -all. List all containers (not just running ones).

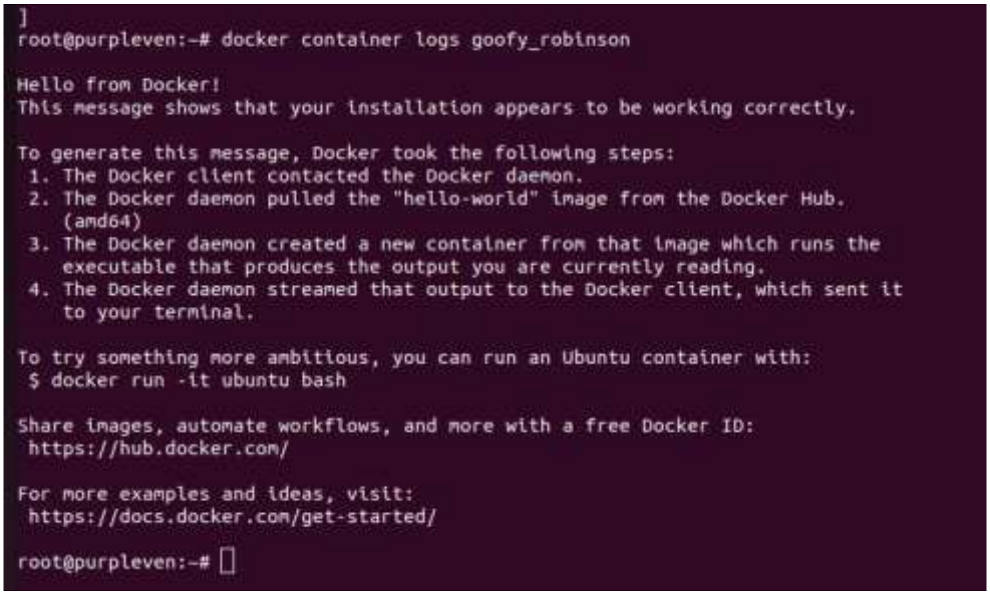
-s is short for --size. List the size for each container. docker container inspect

goofy_robinson — See lots of info about a container.



```
root@purpleven:~# docker container inspect  goofy_robinson
[
    {
        "Id": "ff75755432f2705f534a9f260c21dce28d25cbdfca591d9e620fcc195a49bc98",
        "Created": "2022-10-29T08:32:55.489828405Z",
        "Path": "/hello",
        "Args": [],
        "State": {
            "Status": "exited",
            "Running": false,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 0,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2022-10-29T08:32:55.752273455Z",
            "FinishedAt": "2022-10-29T08:32:55.752101339Z"
        },
        "Image": "sha256:feb5d9fea6a5e9606aa995e879d862b825965ba48de054caab5ef356dc6b3412",
        "ResolvConfPath": "/var/lib/docker/containers/ff75755432f2705f534a9f260c21dce28d25cbdfca591d9e620fcc
        "HostnamePath": "/var/lib/docker/containers/ff75755432f2705f534a9f260c21dce28d25cbdfca591d9e620fcc19
        "HostsPath": "/var/lib/docker/containers/ff75755432f2705f534a9f260c21dce28d25cbdfca591d9e620fcc195a4
        "LogPath": "/var/lib/docker/containers/ff75755432f2705f534a9f260c21dce28d25cbdfca591d9e620fcc195a49b
        "Name": "/goofy_robinson",
        "RestartCount": 0,
```

docker container logs goofy_robinson — Print a container’s logs.

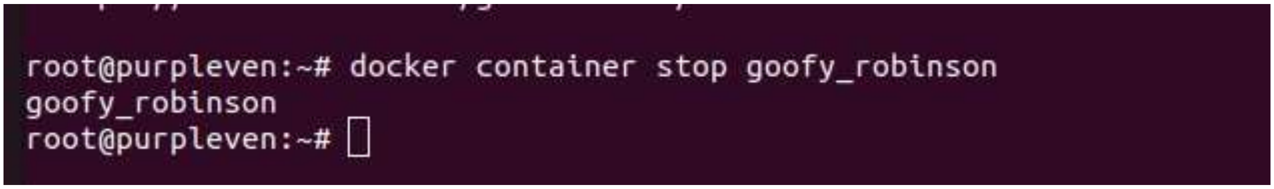


```
]
root@purpleven:~# docker container logs goofy_robinson

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

root@purpleven:~# 
```

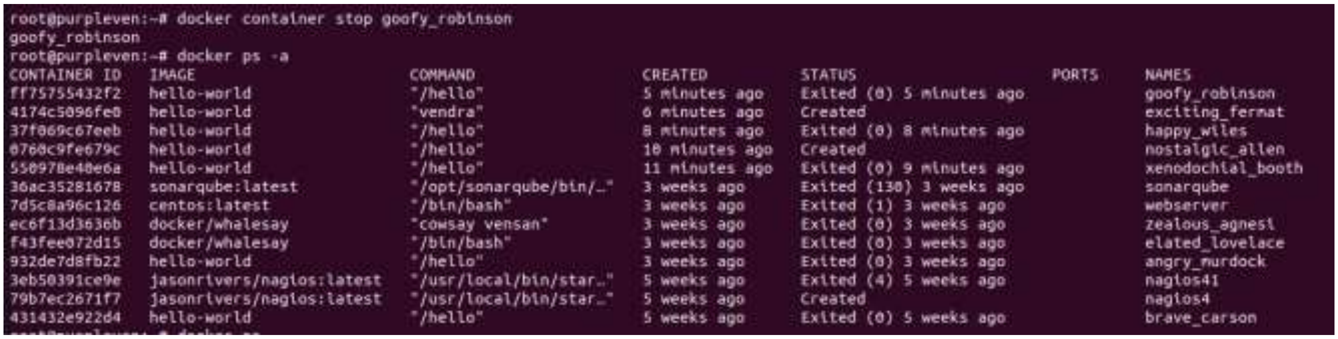
Container Endings

Sometimes you need to stop a running container.

docker container stop goofy_robinson — Stop one or more running containers gracefully. Gives a default of 10 seconds before container shutdown to finish any processes.



```
root@purpleven:~# docker container stop goofy_robinson
goofy_robinson
root@purpleven:~# 
```



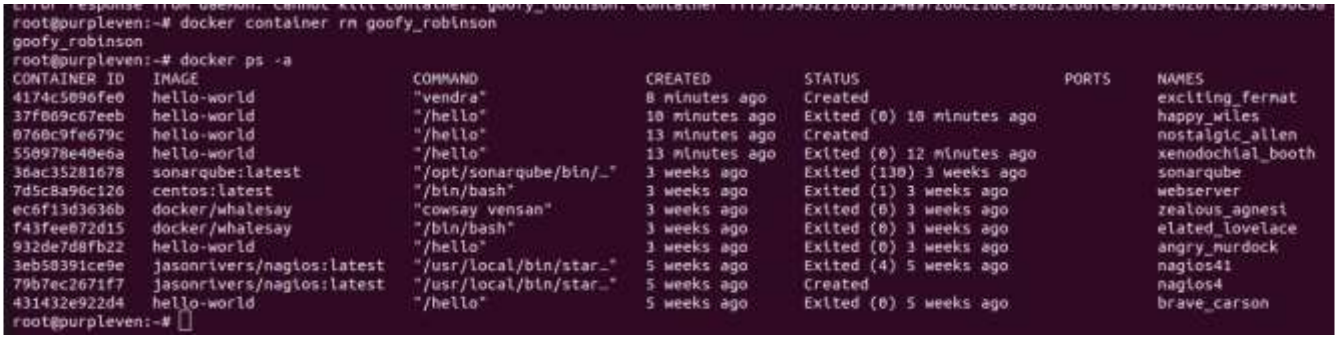
```
root@purpleven:~# docker container stop goofy_robinson
goofy_robinson
root@purpleven:~# docker ps -a
CONTAINER ID   IMAGE                       COMMAND                   CREATED          STATUS                       PORTS     NAMES
ff75755432f2   hello-world                 "/hello"                  5 minutes ago    Exited (0) 5 minutes ago               goofy_robinson
4174c5096fe0   hello-world                 "vendra"                  6 minutes ago    Created                                exciting_fermat
37f069c67eeb   hello-world                 "/hello"                  8 minutes ago    Exited (0) 8 minutes ago               happy_wiles
0760c9fe679c   hello-world                 "/hello"                  10 minutes ago   Created                                nostalgic_allen
550978e40e6a   hello-world                 "/hello"                  11 minutes ago   Exited (0) 9 minutes ago               xenodochial_booth
36ac35281678   sonarqube:latest            "/opt/sonarqube/bin/…"    3 weeks ago      Exited (130) 3 weeks ago               sonarqube
7d5c8a96c126   centos:latest               "/bin/bash"               3 weeks ago      Exited (1) 3 weeks ago                 webserver
ec6f13d3636b   docker/whalesay             "cowsay vensan"           3 weeks ago      Exited (0) 3 weeks ago                 zealous_agnesi
f43fee072d15   docker/whalesay             "/bin/bash"               3 weeks ago      Exited (0) 3 weeks ago                 elated_lovelace
932de7d8fb22   hello-world                 "/hello"                  3 weeks ago      Exited (0) 3 weeks ago                 angry_murdock
3eb50391ce9e   jasonrivers/nagios:latest   "/usr/local/bin/star…"    5 weeks ago      Exited (4) 5 weeks ago                 nagios41
79b7ec2671f7   jasonrivers/nagios:latest   "/usr/local/bin/star…"    5 weeks ago      Created                                nagios4
431432e922d4   hello-world                 "/hello"                  5 weeks ago      Exited (0) 5 weeks ago                 brave_carson
```

Or if you are impatient:

docker container kill exciting_fermat— Stop one or more running containers abruptly. It’s like pulling the plug on the TV. Prefer stop in most situations. docker container kill $(docker ps -q) — Kill all running containers.

Then you delete the container with: docker container rm

goofy_robinson — Delete one or more containers.



```
root@purpleven:~# docker container rm goofy_robinson
goofy_robinson
root@purpleven:~# docker ps -a
CONTAINER ID   IMAGE                          COMMAND                   CREATED          STATUS                        PORTS     NAMES
4174c5096fe0   hello-world                    "vendra"                  8 minutes ago    Created                                 exciting_fermat
37f069c67eeb   hello-world                    "/hello"                  10 minutes ago   Exited (0) 10 minutes ago               happy_wiles
0760c9fe679c   hello-world                    "/hello"                  13 minutes ago   Created                                 nostalgic_allen
550978e40e6a   hello-world                    "/hello"                  13 minutes ago   Exited (0) 12 minutes ago               xenodochial_booth
36ac35281678   sonarqube:latest               "/opt/sonarqube/bin/…"    3 weeks ago      Exited (130) 3 weeks ago                sonarqube
7d5c8a96c126   centos:latest                  "/bin/bash"               3 weeks ago      Exited (1) 3 weeks ago                  webserver
ec6f13d3636b   docker/whalesay                "cowsay vensan"           3 weeks ago      Exited (0) 3 weeks ago                  zealous_agnest
f43fee072d15   docker/whalesay                "/bin/bash"               3 weeks ago      Exited (0) 3 weeks ago                  elated_lovelace
932de7d8fb22   hello-world                    "/hello"                  3 weeks ago      Exited (0) 3 weeks ago                  angry_murdock
3eb58391ce9e   jasonrivers/nagios:latest      "/usr/local/bin/star…"    5 weeks ago      Exited (4) 5 weeks ago                  nagios41
79b7ec2671f7   jasonrivers/nagios:latest      "/usr/local/bin/star…"    5 weeks ago      Created                                 nagios4
431432e922d4   hello-world                    "/hello"                  5 weeks ago      Exited (0) 5 weeks ago                  brave_carson
root@purpleven:~# 
```

docker container rm $(docker ps -a -q) — Delete all containers that are not running.

Those are the eight essential commands for Docker containers.

To recap, you first create a container. Then, you start the container. Or combine those steps with docker run my_container. Then, your app runs. Yippee!

Then, you stop a container with docker stop my_container. Eventually you delete the container with docker rm my_container.

Conclusion: Thus, the docker installation on ubuntu was done successfully and the basic commands of docker has been

Reference:

1. https://www.simplilearn.com/tutorials/docker-tutorial/how-to-install-docker-on-ubuntu
2. https://towardsdatascience.com/12-essential-docker-commands-you-should-know-c2d5a7751bb5

3. https://docs.docker.com/engine/reference/commandline/container/

https://towardsdatascience.com/15-docker-commands-you-should-know-970ea5203421

ASSIGNMENT 1 : DOCKER VOLUME

Docker volumes

How To Share Data Between the Docker

Container and the Host

In general, Docker containers are ephemeral, running just as long as it takes for the command issued in the container to complete. By default, any data created inside the container is only available from within the container and only while the container is running.

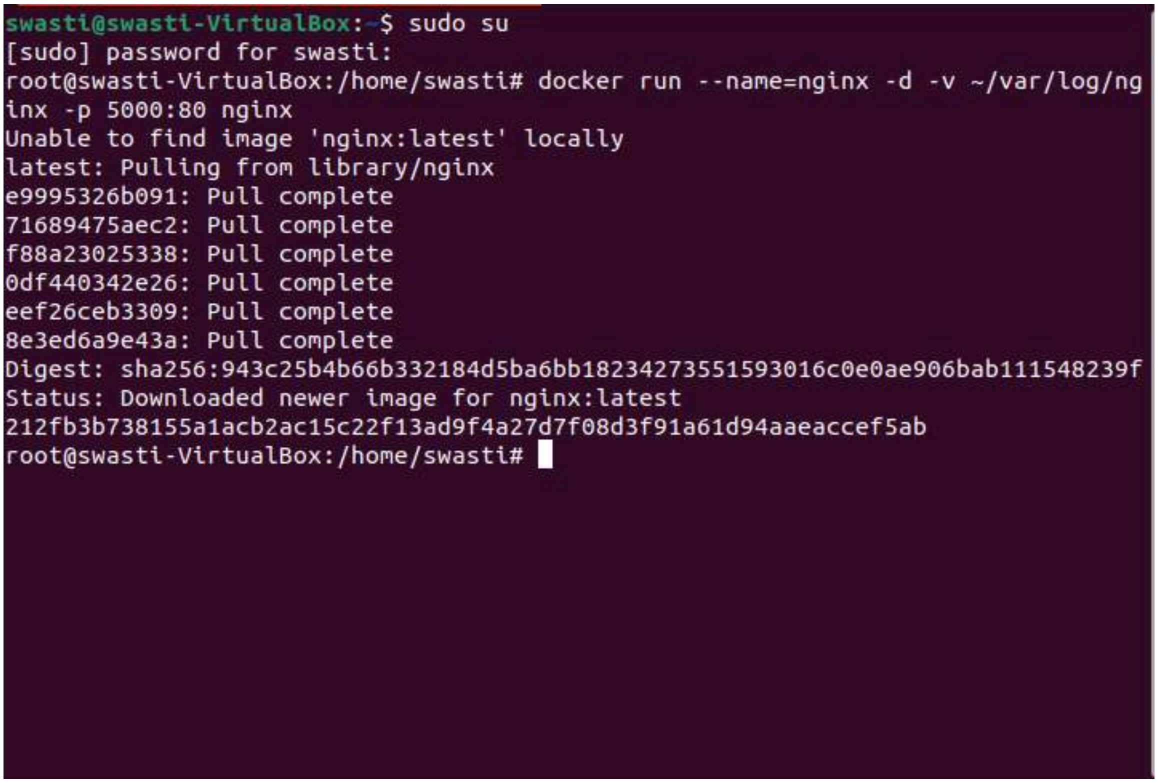
Docker volumes can be used to share files between a host system and the Docker container. For example, let’s say you wanted to use the official Docker Nginx image and keep a permanent copy of Nginx’s log files to analyze later. By default, the nginx Docker image will log to the /var/log/nginx directory inside the Docker Nginx container. Normally it’s not reachable from the host filesystem.

In this tutorial, we’ll explore how to make data from inside the container accessible on the host machine.

Step 1 — Bind Mounting a Volume

The following command will create a directory called nginx logs in your current user’s home directory and bind mount it to /var/log/nginx in the container:

```
$docker run --name=nginx -d -v ~/nginxlogs:/var/log/nginx -p 5000:80 nginx
```



```
swasti@swasti-VirtualBox:~$ sudo su
[sudo] password for swasti:
root@swasti-VirtualBox:/home/swasti# docker run --name=nginx -d -v ~/var/log/ng
inx -p 5000:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
e9995326b091: Pull complete
71689475aec2: Pull complete
f88a23025338: Pull complete
0df440342e26: Pull complete
eef26ceb3309: Pull complete
8e3ed6a9e43a: Pull complete
Digest: sha256:943c25b4b66b332184d5ba6bb18234273551593016c0e0ae906bab111548239f
Status: Downloaded newer image for nginx:latest
212fb3b738155a1acb2ac15c22f13ad9f4a27d7f08d3f91a61d94aaeaccef5ab
root@swasti-VirtualBox:/home/swasti# 
```

Let’s take a moment to examine this command in detail:

- **--name=nginx names the container so we can refer to it more easily.**
- **-d detaches the process and runs it in the background. Otherwise, we would just be watching an empty Nginx prompt and wouldn’t be able to use this terminal until we killed Nginx.**
- **-v ~/nginxlogs:/var/log/nginx sets up a bind mount volume that links the /var/log/nginx directory from inside the Nginx container to the ~/nginxlogs directory on the host machine. Docker uses a : to split the host’s path from the container path, and the host path always comes first.**

- **-p 5000:80 sets up a portt forward. The Nginx container is listening on port 80 by default. This flag maps the container's port 80 to port 5000 on the host system.**
- **nginx specifies that the container should be built from the Nginx image, which issues the command [nginx -g "daem on off"](#) to start Nginx.**

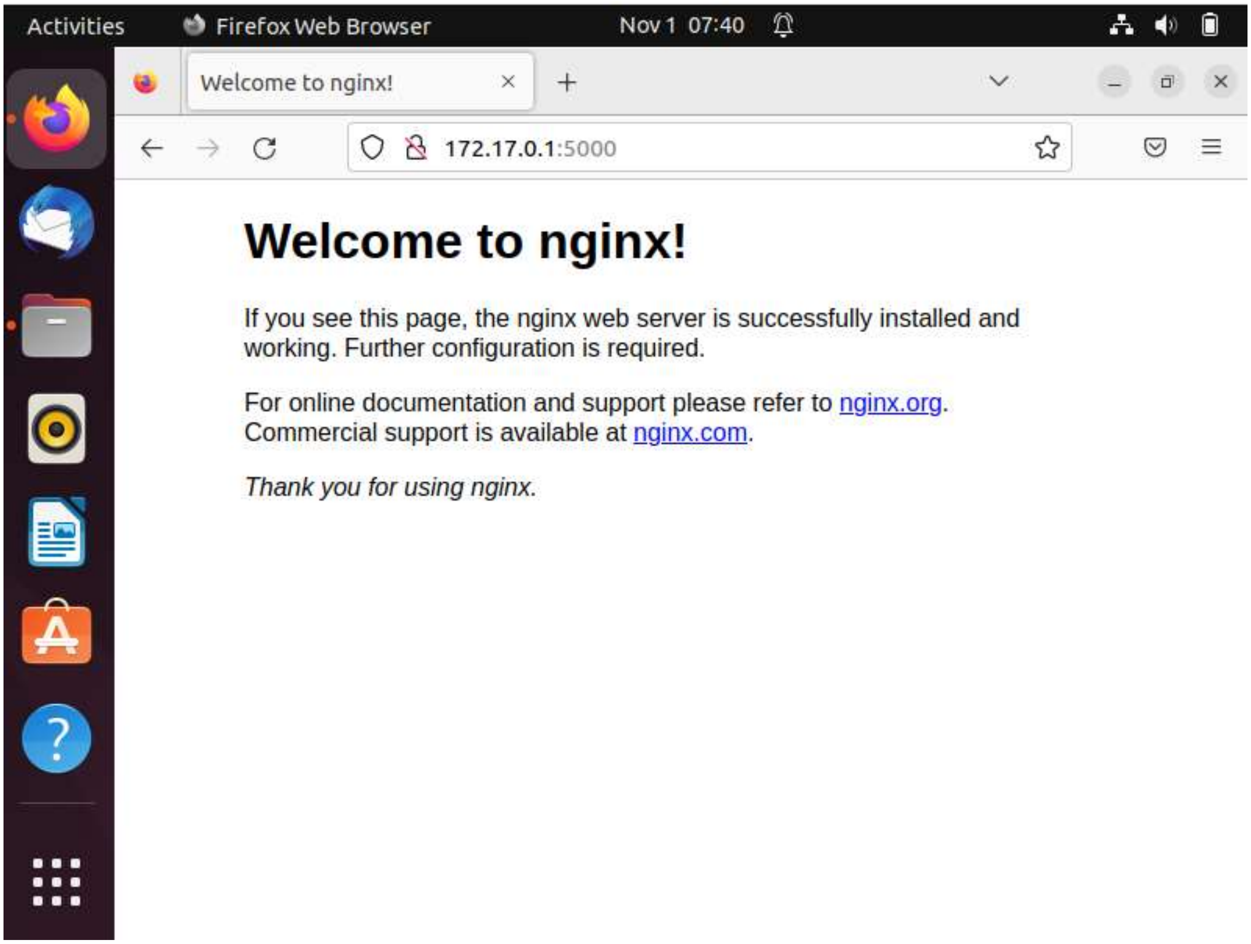
Note: The -v flag is very flexible . It can bindmount or name a volume with just a slight adjustment in syntax. If the first argument begins with a / or ~/, you're creating a bindmount. Remove that, and you're naming the volume.

- **-v /path:/path/in/container mounts the host directory, /path at the /path/in /container**
- **-v path:/path/in/container creates a volume named path with no relationship to the host.**

Step 2 — Accessing Data on the Host

We now have a copy of Nginx running inside a Docker container on our machine, and our host machine's port 5000 maps directly to that copy of Nginx's port 80.

Load the address in a web browser, using the IP address or hostname of your server and the port number: http://your_server_ip:5000. You should see:



Welcome to nginx!

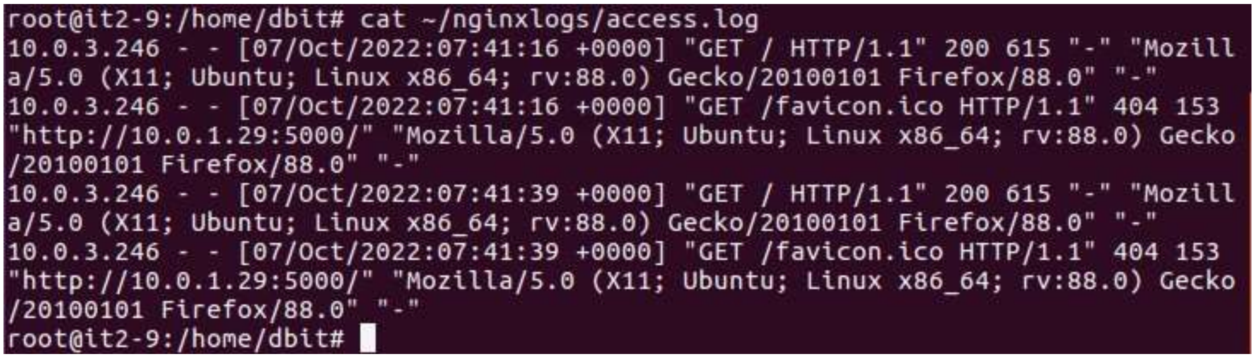
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

More interestingly, if we look in the ~/nginxlogs directory on the host, we'll see the access.log created by the container's nginx which will show our request:

$cat ~/nginxlogs/access.log



```
root@it2-9:/home/dbit# cat ~/nginxlogs/access.log
10.0.3.246 - - [07/Oct/2022:07:41:16 +0000] "GET / HTTP/1.1" 200 615 "-" "Mozill
a/5.0 (X11; Ubuntu; Linux x86_64; rv:88.0) Gecko/20100101 Firefox/88.0" "-"
10.0.3.246 - - [07/Oct/2022:07:41:16 +0000] "GET /favicon.ico HTTP/1.1" 404 153
"http://10.0.1.29:5000/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:88.0) Gecko
/20100101 Firefox/88.0" "-"
10.0.3.246 - - [07/Oct/2022:07:41:39 +0000] "GET / HTTP/1.1" 200 615 "-" "Mozill
a/5.0 (X11; Ubuntu; Linux x86_64; rv:88.0) Gecko/20100101 Firefox/88.0" "-"
10.0.3.246 - - [07/Oct/2022:07:41:39 +0000] "GET /favicon.ico HTTP/1.1" 404 153
"http://10.0.1.29:5000/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:88.0) Gecko
/20100101 Firefox/88.0" "-"
root@it2-9:/home/dbit# 
```

If you make any changes to the ~/nginxlogs folder, you'll be able to see them from inside the Docker container in real time as well.

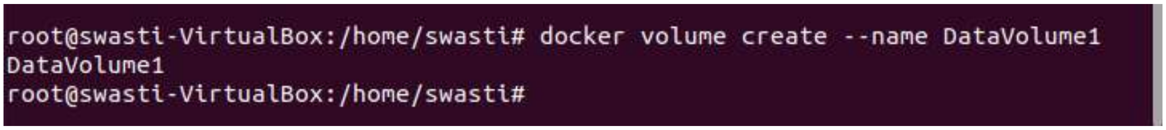
How To Share Data between Docker Containers

Step 1 — Creating an Independent Volume

Introduced in Docker's 1.9 release, the docker volume create command allows you to create a volume without relating it to any particular container. We'll use this command to add a volume named DataVolume1:

```
#docker volume create --name DataVolume1
```

The name is displayed, indicating that the command was successful:

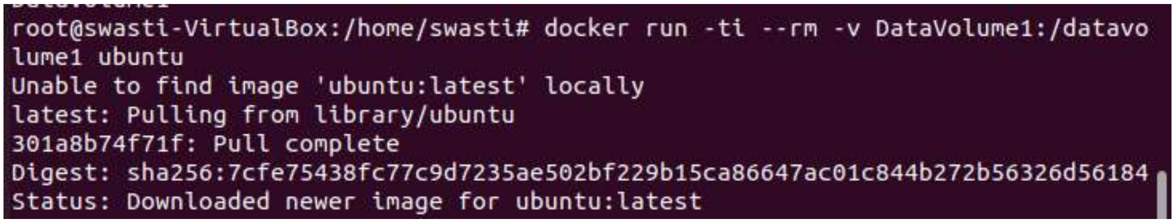


```
root@swasti-VirtualBox:/home/swasti# docker volume create --name DataVolume1
DataVolume1
root@swasti-VirtualBox:/home/swasti#
```

Output
DataVolume1

To make use of the volume, we'll create a new container from the Ubuntu image, using the --rm flag to automatically delete it when we exit. We'll also use -v to mount the new volume. -v requires the name of the volume, a colon, then the absolute path to where the volume should appear inside the container. If the directories in the path don't exist as part of the image, they'll be created when the command runs. If they do exist, the mounted volume will hide the existing content:

```
#docker run -ti --rm -v DataVolume1:/datavolume1 ubuntu
```



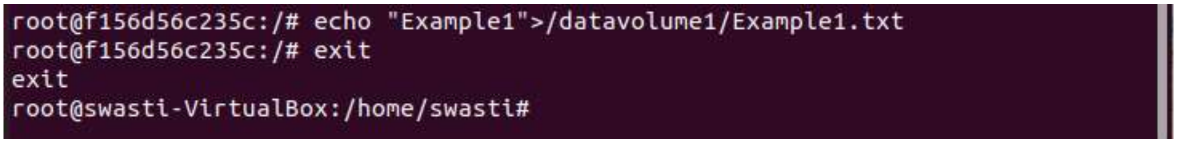
```
root@swasti-VirtualBox:/home/swasti# docker run -ti --rm -v DataVolume1:/datavo
lume1 ubuntu
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
301a8b74f71f: Pull complete
Digest: sha256:7cfe75438fc77c9d7235ae502bf229b15ca86647ac01c844b272b56326d56184
Status: Downloaded newer image for ubuntu:latest
```

While in the container, let's write some data to the volume:

```
$echo "Example1" > /datavolume1/Example1.txt
```

Because we used the --rm flag, our container will be automatically deleted when we exit. Our volume, however, will still be accessible.

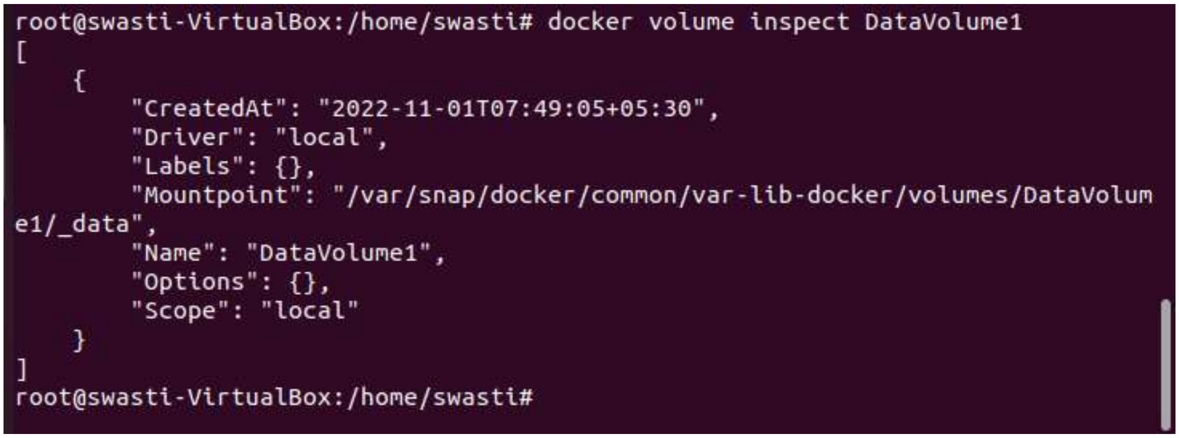
```
$exit
```



```
root@f156d56c235c:/# echo "Example1">/datavolume1/Example1.txt
root@f156d56c235c:/# exit
exit
root@swasti-VirtualBox:/home/swasti#
```

docWe can verify that the volume is present on our system with docker volume inspect:

#docker volume inspect DataVolume1



```
root@swasti-VirtualBox:/home/swasti# docker volume inspect DataVolume1
[
    {
        "CreatedAt": "2022-11-01T07:49:05+05:30",
        "Driver": "local",
        "Labels": {},
        "Mountpoint": "/var/snap/docker/common/var-lib-docker/volumes/DataVolum
e1/_data",
        "Name": "DataVolume1",
        "Options": {},
        "Scope": "local"
    }
]
root@swasti-VirtualBox:/home/swasti#
```

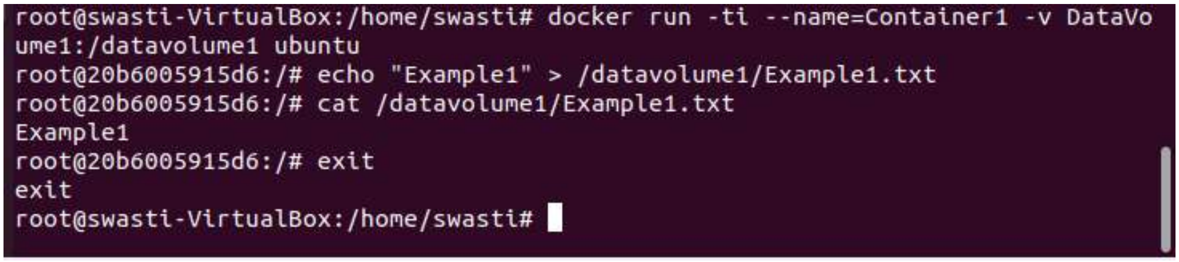
Note: We can even look at the data on the host at the path listed as the Mountpoint. We should avoid altering it, however, as it can cause data corruption if applications or containers are unaware of changes.

Next, let's start a new container and attach DataVolume1:

#docker run --rm -ti -v DataVolume1:/datavolume1 ubuntu

Verify the contents:

$cat /datavolume1/Example1.txt

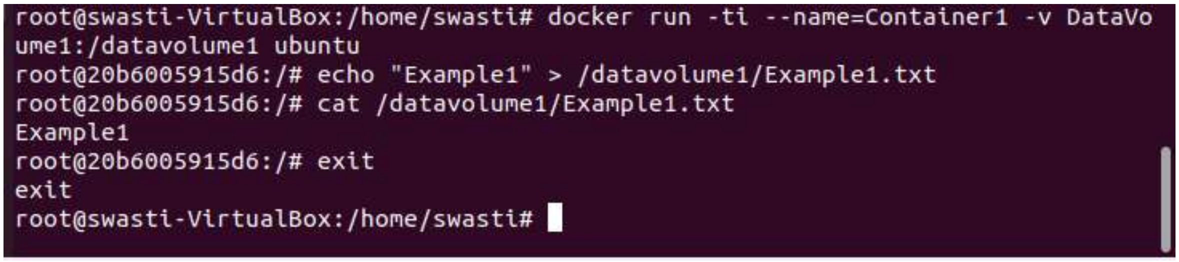


```
root@swasti-VirtualBox:/home/swasti# docker run -ti --name=Container1 -v DataVo
ume1:/datavolume1 ubuntu
root@20b6005915d6:/# echo "Example1" > /datavolume1/Example1.txt
root@20b6005915d6:/# cat /datavolume1/Example1.txt
Example1
root@20b6005915d6:/# exit
exit
root@swasti-VirtualBox:/home/swasti#
```

Output
Example1

Exit the container:

```
$exit
```



```
root@swasti-VirtualBox:/home/swasti# docker run -ti --name=Container1 -v DataVo
ume1:/datavolume1 ubuntu
root@20b6005915d6:/# echo "Example1" > /datavolume1/Example1.txt
root@20b6005915d6:/# cat /datavolume1/Example1.txt
Example1
root@20b6005915d6:/# exit
exit
root@swasti-VirtualBox:/home/swasti#
```

In this example, we created a volume, attached it to a container, and verified its persistence.

Step 2 — Creating a Volume that Persists when the Container is Removed

In our next example, we'll create a volume at the same time as the container, delete the container, then attach the volume to a new container.

We'll use the docker run command to create a new container using the base Ubuntu image. -t will give us a terminal, and -i will allow us to interact with it. For clarity, we'll use --name to identify the container.

The -v flag will allow us to create a new volume, which we'll call DataVolume2. We'll use a colon to separate this name from the path where the volume should be mounted in the container. Finally, we will specify the base Ubuntu image and rely on the default command in the [Ubuntu base image's Docker file](), bash, to drop us into a shell:

```
$docker run -ti --name=Container2 -v DataVolume2:/datavolume2 ubuntu
```

Note: The -v flag is very flexible. It can bindmount or name a volume with just a slight adjustment in syntax. If the first argument begins with a / or ~/ you're creating a bindmount. Remove that, and you're naming the volume. For example:

- **-v /path:/path/in/container mounts the host directory, /path at the /path/in/container**

- **-v path:/path/in/container creates a volume named path with no relationship to the host.**

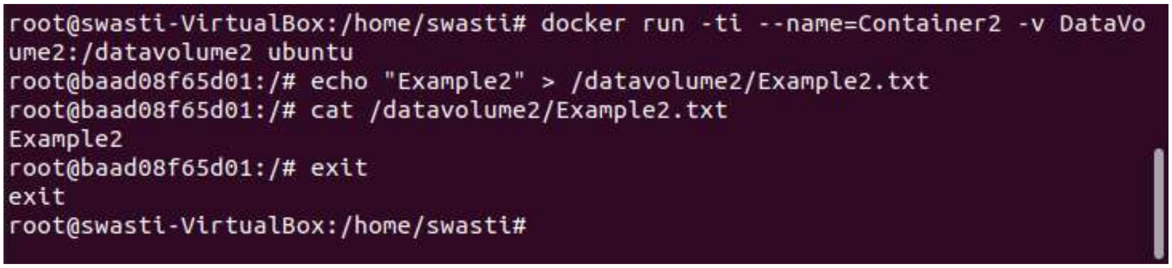
While in the container, we’ll write some data to the volume:

$echo "Example2" > /datavolume2/Example2.txt

$cat /datavolume2/Example2.txt

Output

Example2



```
root@swasti-VirtualBox:/home/swasti# docker run -ti --name=Container2 -v DataVo
ume2:/datavolume2 ubuntu
root@baad08f65d01:/# echo "Example2" > /datavolume2/Example2.txt
root@baad08f65d01:/# cat /datavolume2/Example2.txt
Example2
root@baad08f65d01:/# exit
exit
root@swasti-VirtualBox:/home/swasti#
```

Let’s exit the container:

$exit

When we restart the container, the volume will mount automatically:

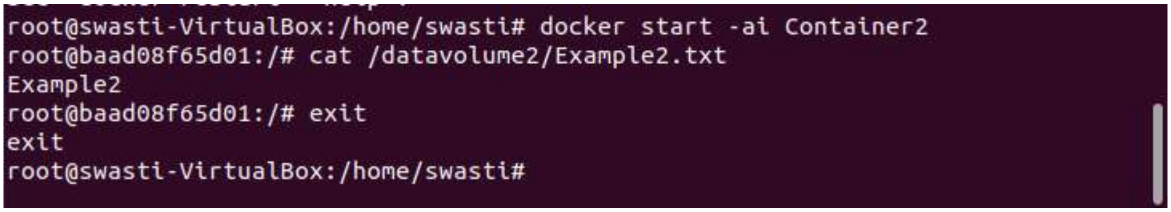
#docker start -ai Container2

Let’s verify that the volume has indeed mounted and our data is still in place:

$cat /datavolume2/Example2.txt

Output

Example2



```
root@swasti-VirtualBox:/home/swasti# docker start -ai Container2
root@baad08f65d01:/# cat /datavolume2/Example2.txt
Example2
root@baad08f65d01:/# exit
exit
root@swasti-VirtualBox:/home/swasti#
```

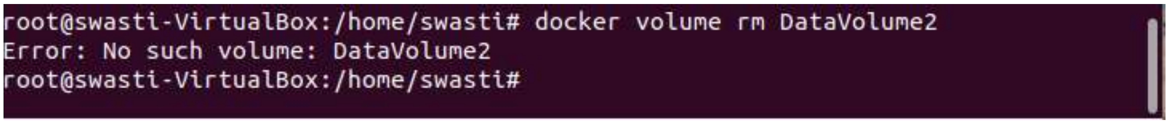
Finally, let's exit and clean up:

```
$exit
```

Docker won't let us remove a volume if it's referenced by a container. Let's see what happens when we try:

```
#docker volume rm DataVolume2
```

The message tells us that the volume is still in use and supplies the long version of the container ID:



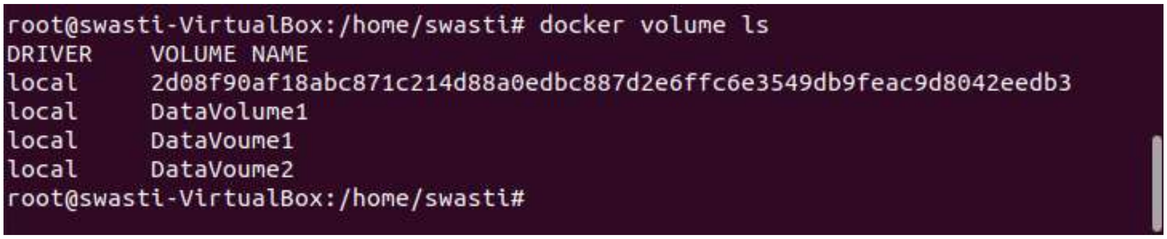
```
root@swasti-VirtualBox:/home/swasti# docker volume rm DataVolume2
Error: No such volume: DataVolume2
root@swasti-VirtualBox:/home/swasti#
```

We can use this ID to remove the container:

```
#docker rm
d0d2233b668eddad4986313c7a4a1bc0d2edaf0c7e1c02a6a6256de27db17a63
```

Output
d0d2233b668eddad4986313c7a4a1bc0d2edaf0c7e1c02a6a6256de27db17a63

Removing the container won't affect the volume. We can see it's still present on the system by listing the volumes with docker volume ls:

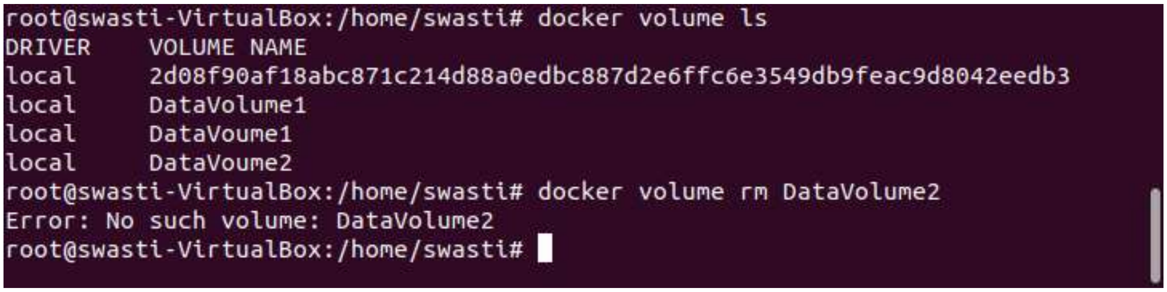


```
root@swasti-VirtualBox:/home/swasti# docker volume ls
DRIVER    VOLUME NAME
local     2d08f90af18abc871c214d88a0edbc887d2e6ffc6e3549db9feac9d8042eedb3
local     DataVolume1
local     DataVoume1
local     DataVoume2
root@swasti-VirtualBox:/home/swasti#
```

```
                                          VOLUME NAME
Output DRIVER local                     DataVolume2
```

And we can use docker volume rm to remove it:

```
#docker volume rm DataVolume2
```



```
root@swasti-VirtualBox:/home/swasti# docker volume ls
DRIVER    VOLUME NAME
local     2d08f90af18abc871c214d88a0edbc887d2e6ffc6e3549db9feac9d8042eedb3
local     DataVolume1
local     DataVoume1
local     DataVoume2
root@swasti-VirtualBox:/home/swasti# docker volume rm DataVolume2
Error: No such volume: DataVolume2
root@swasti-VirtualBox:/home/swasti# 
```

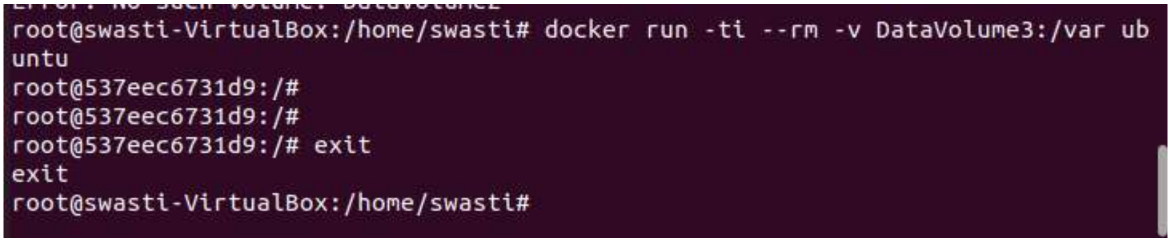
In this example, we created an empty data volume at the same time that we created a container. In our next example, we'll explore what happens when we create a volume with a container directory that already contains data.

Step 3 — Creating a Volume from an Existing Directory with Data

Generally, creating a volume independently with docker volume create and creating one while creating a container are equivalent, with one exception. If we create a volume at the same time that we create a container and we provide the path to a directory that contains data in the base image, that data will be copied into the volume.

As an example, we'll create a container and add the data volume at /var, a directory which contains data in the base image:

```
#docker run -ti --rm -v DataVolume3:/var ubuntu
```



```
Error: No such volume: DataVolume2
root@swasti-VirtualBox:/home/swasti# docker run -ti --rm -v DataVolume3:/var ub
untu
root@537eec6731d9:/#
root@537eec6731d9:/#
root@537eec6731d9:/# exit
exit
root@swasti-VirtualBox:/home/swasti#
```

All the content from the base image's /var directory is copied into the volume, and we can mount that volume in a new container.

Exit the current container:

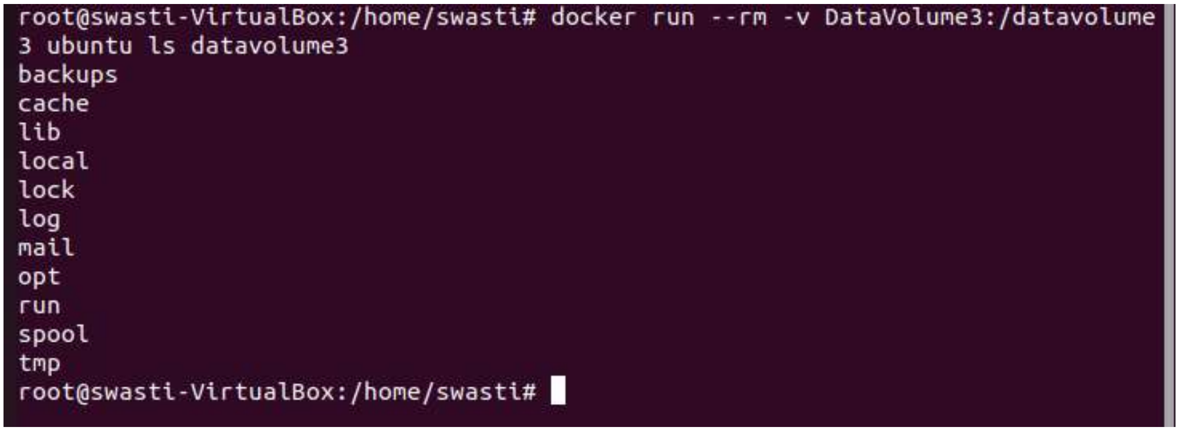
$exit

This time, rather than relying on the base image's default bash command, we'll issue our own ls command, which will show the contents of the volume without entering the shell:

```
#docker run --rm -v DataVolume3:/datavolume3 ubuntu ls datavolume3
```

The directory datavolume3 now has a copy of the contents of the base image's /var directory:

Output
backups
cache
lib
local
lock
log
mail
opt
run
spool
tmp



```
root@swasti-VirtualBox:/home/swasti# docker run --rm -v DataVolume3:/datavolume
3 ubuntu ls datavolume3
backups
cache
lib
local
lock
log
mail
opt
run
spool
tmp
root@swasti-VirtualBox:/home/swasti# 
```

It's unlikely that we would want to mount /var/ in this way, but this can be helpful if we've crafted our own image and want an easy way to preserve data. In our next example, we'll demonstrate how a volume can be shared between multiple containers.

Step 4 — Sharing Data Between Multiple Docker Containers

So far, we've attached a volume to one container at a time. Often, we'll want multiple containers to attach to the same data volume. This is relatively straightforward to accomplish, but there's

63 one critical caveat: at this time, Docker doesn’t handle file locking. If you need multiple containers writing to the volume, the applications running in those containers must be designed to write to shared data stores in order to prevent data corruption.

Create Container4 and DataVolume4

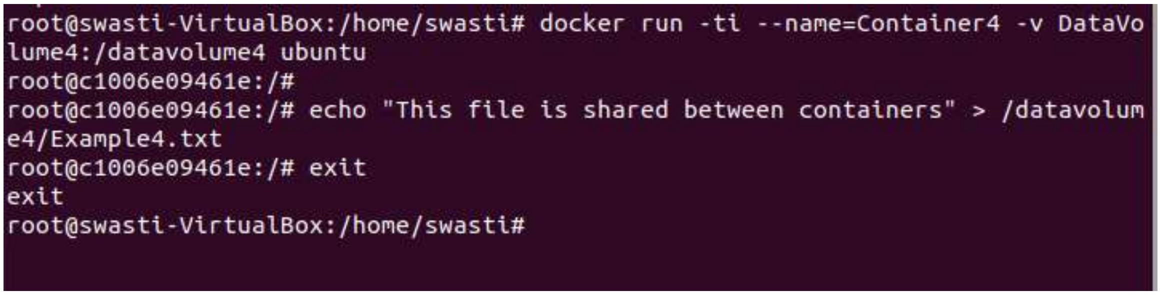
Use docker run to create a new container named Container4 with a data volume attached:

```
#docker run -ti --name=Container4 -v DataVolume4:/datavolume4 ubuntu
```

Next we’ll create a file and add some text:

```
$echo "This file is shared between containers" >
```

/datavolume4/Example4.txt Then, we’ll exit the container:



```
root@swasti-VirtualBox:/home/swasti# docker run -ti --name=Container4 -v DataVo
lume4:/datavolume4 ubuntu
root@c1006e09461e:/#
root@c1006e09461e:/# echo "This file is shared between containers" > /datavolum
e4/Example4.txt
root@c1006e09461e:/# exit
exit
root@swasti-VirtualBox:/home/swasti#
```

This returns us to the host command prompt, where we’ll make a new container that mounts the data volume from Container4.

Create Container5 and Mount Volumes from Container4

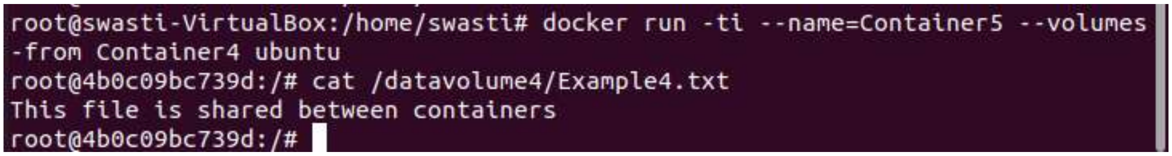
We’re going to create Container5, and mount the volumes from

```
Container4: #docker   run   -ti  --name=Container5 --volumes-from
```

Container4 ubuntu Let’s check the data persistence:

```
$cat /datavolume4/Example4.txt
```

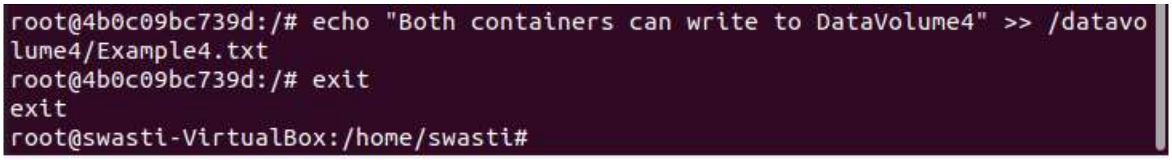
Output



```
root@swasti-VirtualBox:/home/swasti# docker run -ti --name=Container5 --volumes
-from Container4 ubuntu
root@4b0c09bc739d:/# cat /datavolume4/Example4.txt
This file is shared between containers
root@4b0c09bc739d:/#
```

This file is shared between containers
Now let’s append some text from Container5:

```
$echo "Both containers can write to DataVolume4" >>
/datavolume4/Example4.txt
```



```
root@4b0c09bc739d:/# echo "Both containers can write to DataVolume4" >> /datavo
lume4/Example4.txt
root@4b0c09bc739d:/# exit
exit
root@swasti-VirtualBox:/home/swasti#
```

Finally, we’ll exit the container:

```
$exit
```

Next, we’ll check that our data is still present to Container4.

View Changes Made in Container5

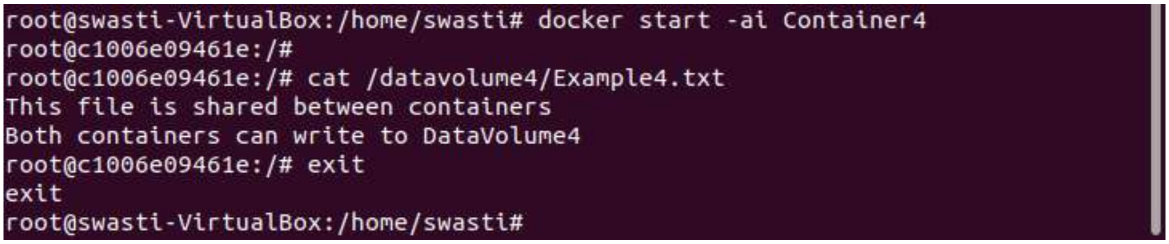
Let’s check for the changes that were written to the data volume by Container5 by restarting Container4:

```
#docker start -ai Container4
```

Check for the changes:

```
$cat /datavolume4/Example4.txt
```

Output
This file is shared between containers
Both containers can write to DataVolume4



```
root@swasti-VirtualBox:/home/swasti# docker start -ai Container4
root@c1006e09461e:/#
root@c1006e09461e:/# cat /datavolume4/Example4.txt
This file is shared between containers
Both containers can write to DataVolume4
root@c1006e09461e:/# exit
exit
root@swasti-VirtualBox:/home/swasti#
```

Now that we've verified that both containers were able to read and write from the data volume, we'll exit the container:

```
$exit
```

Again, Docker doesn't handle any file locking, so applications must account for the file locking themselves. It is possible to mount a Docker volume as read-only to ensure that data corruption won't happen by accident when a container requires read-only access by adding :ro. Let's look at how this works.

Start Container 6 and Mount the Volume Read-Only

Once a volume has been mounted in a container, rather than unmounting it like we would with a typical Linux file system, we can instead create a new container mounted the way we want and,

if needed, remove the previous container. To make the volume read-only, we append :ro to the end of the container name:

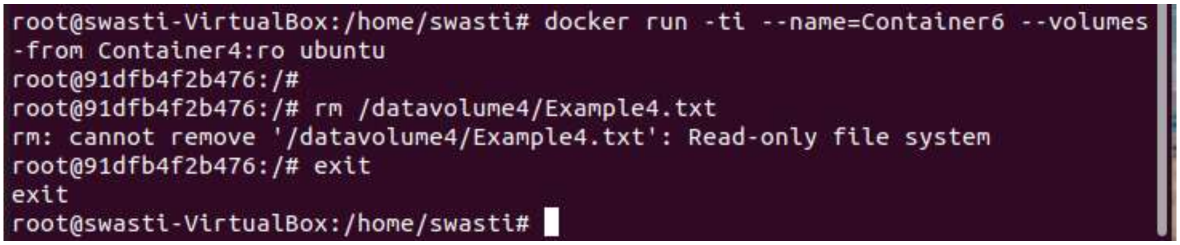
```
#docker run -ti --name=Container6 --volumes-from Container4:ro ubuntu
```

We'll check the read-only status by trying to remove our example file:

```
$rm /datavolume4/Example4.txt
```

Output

rm: cannot remove '/datavolume4/Example4.txt': Read-only file system



```
root@swasti-VirtualBox:/home/swasti# docker run -ti --name=Container6 --volumes
-from Container4:ro ubuntu
root@91dfb4f2b476:/#
root@91dfb4f2b476:/# rm /datavolume4/Example4.txt
rm: cannot remove '/datavolume4/Example4.txt': Read-only file system
root@91dfb4f2b476:/# exit
exit
root@swasti-VirtualBox:/home/swasti# 
```

Finally, we'll exit the container and clean up our test containers and volumes:

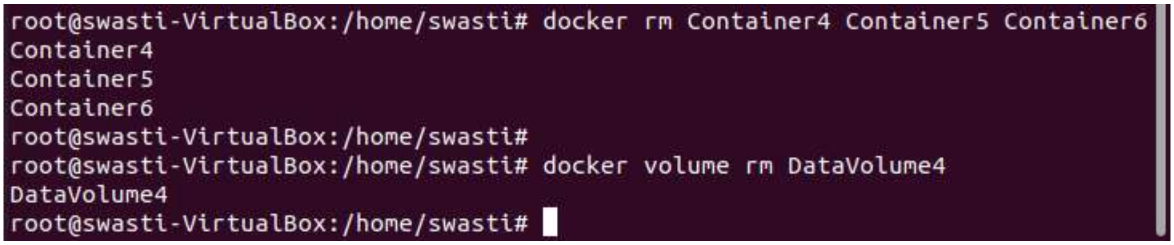
```
$exit
```

Now that we're done, let's clean up our containers and volume:

```
#docker rm Container4 Container5 Container6
```

```
#docker volume rm DataVolume4
```

In this example, we've shown how to share data between two containers using a data volume and how to mount a data volume as read-only.

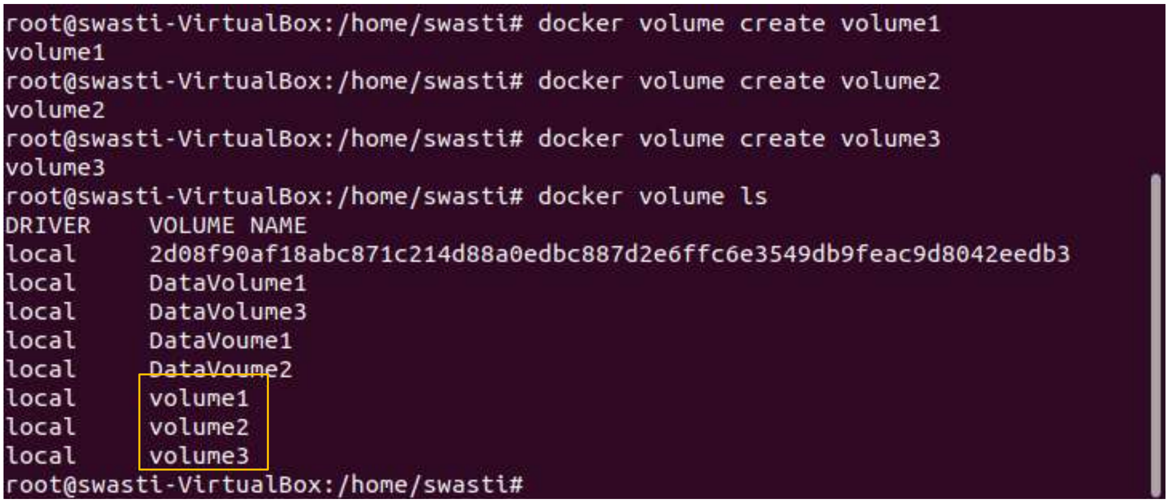


```
root@swasti-VirtualBox:/home/swasti# docker rm Container4 Container5 Container6
Container4
Container5
Container6
root@swasti-VirtualBox:/home/swasti#
root@swasti-VirtualBox:/home/swasti# docker volume rm DataVolume4
DataVolume4
root@swasti-VirtualBox:/home/swasti# 
```

Delete all volumes at once

Using docker rm command, we can remove one volume at a time. If we have multiple volumes and want to delete all volumes then we have to use prune command.

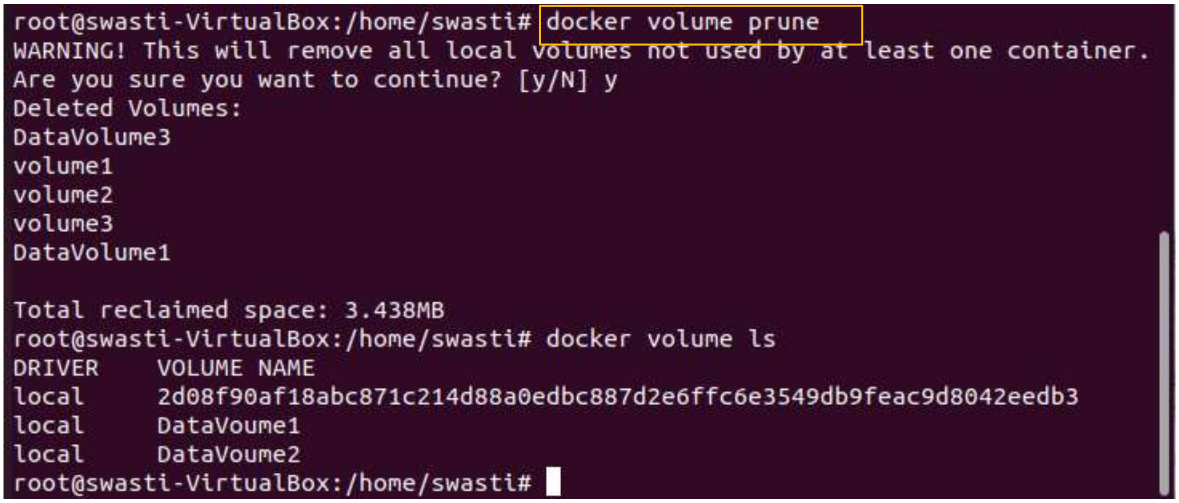
Let us create a few volumes:



```
root@swasti-VirtualBox:/home/swasti# docker volume create volume1
volume1
root@swasti-VirtualBox:/home/swasti# docker volume create volume2
volume2
root@swasti-VirtualBox:/home/swasti# docker volume create volume3
volume3
root@swasti-VirtualBox:/home/swasti# docker volume ls
DRIVER    VOLUME NAME
local     2d08f90af18abc871c214d88a0edbc887d2e6ffc6e3549db9feac9d8042eedb3
local     DataVolume1
local     DataVolume3
local     DataVoume1
local     DataVoume2
local     volume1
local     volume2
local     volume3
root@swasti-VirtualBox:/home/swasti#
```

Now delete all docker volumes at once using command:

```
# docker volume prune
```



```
root@swasti-VirtualBox:/home/swasti# docker volume prune
WARNING! This will remove all local volumes not used by at least one container.
Are you sure you want to continue? [y/N] y
Deleted Volumes:
DataVolume3
volume1
volume2
volume3
DataVolume1

Total reclaimed space: 3.438MB
root@swasti-VirtualBox:/home/swasti# docker volume ls
DRIVER    VOLUME NAME
local     2d08f90af18abc871c214d88a0edbc887d2e6ffc6e3549db9feac9d8042eedb3
local     DataVoume1
local     DataVoume2
root@swasti-VirtualBox:/home/swasti# 
```

See? We have deleted all volumes in one go.

Conclusion

In this tutorial, we created a data volume which allowed data to persist through the deletion of a container. We shared data volumes between containers, with the caveat that applications will need to be designed to handle file locking to prevent data corruption. Finally, we showed how to mount a shared volume in read-only mode. If you're interested

in learning about sharing data between containers and the host system,

REFERENCES :

https://www.digitalocean.com/community/tutorials/how-to-share-data-between-dockercontainers

https://www.digitalocean.com/community/tutorials/how-to-share-data-between-dockercontainers

Experiment 4: Assignment

To Download & Install Selenium WebDriver on Ubuntu

Selenium installation is a 3 step process:

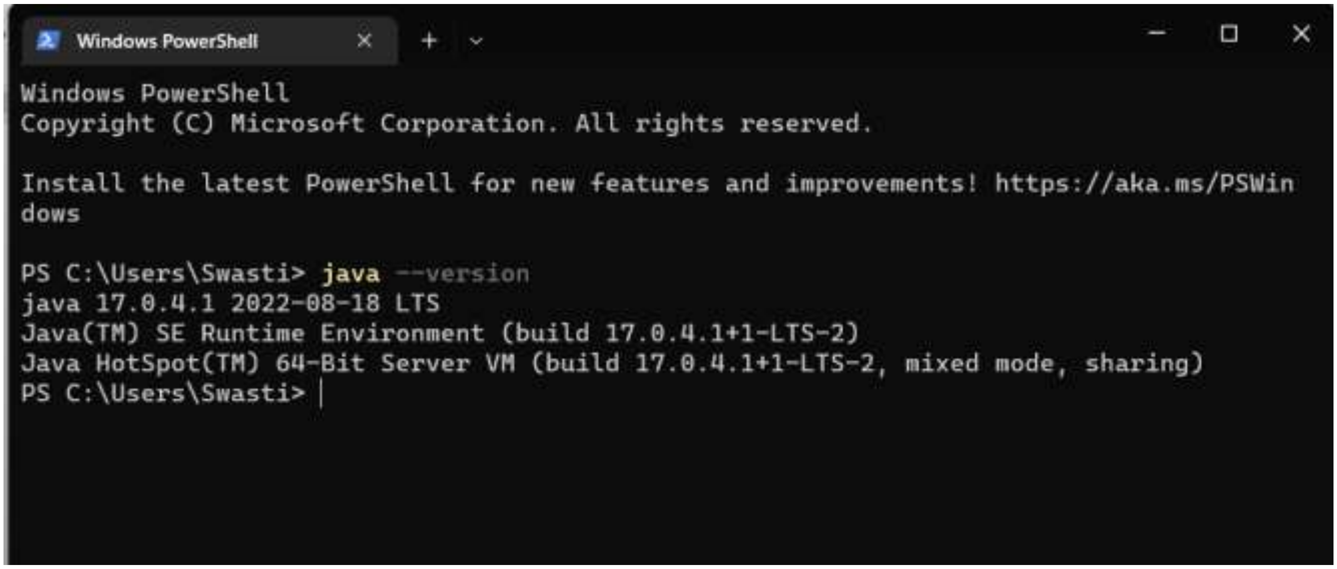
1. Install Java SDK https://www.oracle.com/java/technologies/javase-downloads.html
2. Install Eclipse - http://www.eclipse.org/downloads/
3. Install Selenium Webdriver Files - https://www.selenium.dev/downloads/

In this tutorial, we will learn how to install Selenium Webdriver . Below is the detailed process

NOTE: The versions of Java, Eclipse, Selenium will keep updating with time. But the installation steps will remain the same. Please select the latest version and continue the installation steps below-

Step 1 – Install Java on your computer

Check If java is present in your system If not download and install the **Java Software Development Kit (JDK)** here.

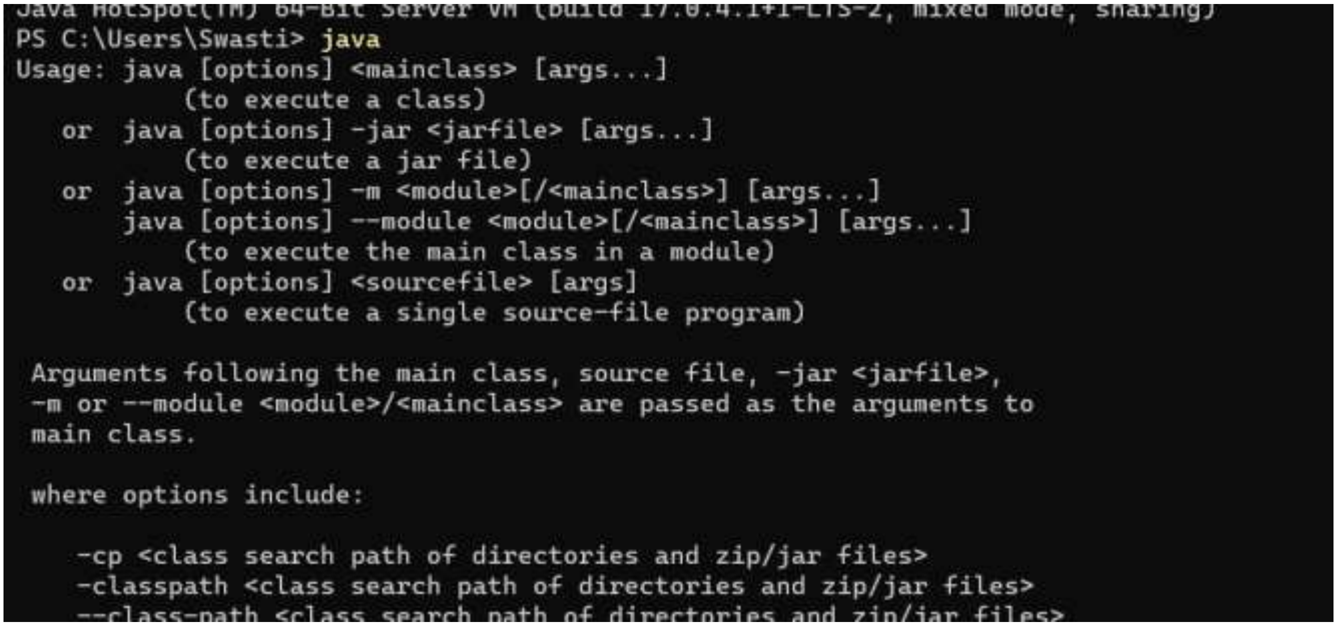


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWin
dows

PS C:\Users\Swasti> java --version
java 17.0.4.1 2022-08-18 LTS
Java(TM) SE Runtime Environment (build 17.0.4.1+1-LTS-2)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.4.1+1-LTS-2, mixed mode, sharing)
PS C:\Users\Swasti> |
```

Next –

This JDK version comes bundled with Java Runtime Environment (JRE), so you do not need to download and install the JRE separately.

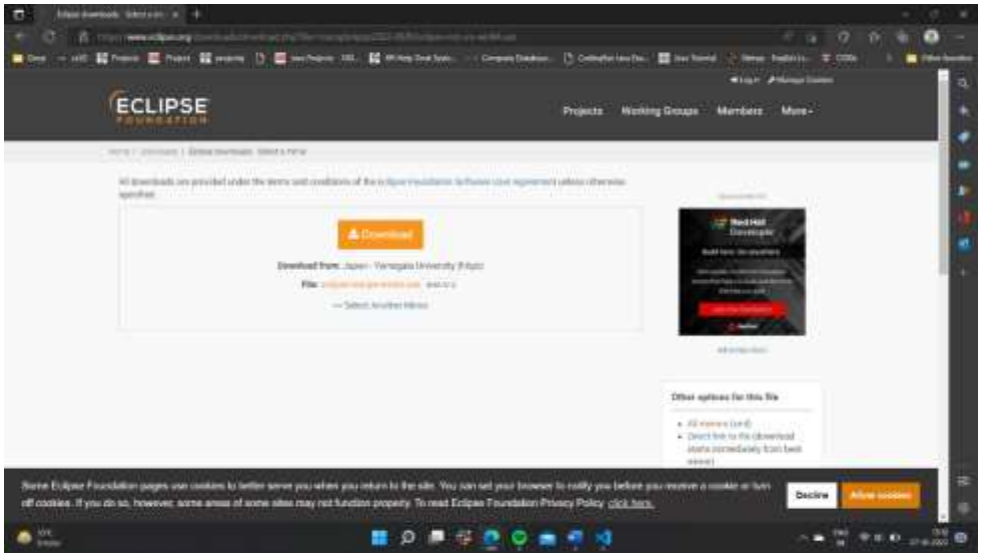
Once installation is complete, open command prompt and type “java”. If you see the following screen you are good to move to the next step



```
Java HotSpot(TM) 64-Bit Server VM (build 17.0.4.1+1-LTS-2, mixed mode, sharing)
PS C:\Users\Swasti> java
Usage: java [options] <mainclass> [args...]
           (to execute a class)
   or  java [options] -jar <jarfile> [args...]
           (to execute a jar file)
   or  java [options] -m <module>[/<mainclass>] [args...]
       java [options] --module <module>[/<mainclass>] [args...]
           (to execute the main class in a module)
   or  java [options] <sourcefile> [args]
           (to execute a single source-file program)

 Arguments following the main class, source file, -jar <jarfile>,
 -m or --module <module>/<mainclass> are passed as the arguments to
 main class.

 where options include:

    -cp <class search path of directories and zip/jar files>
    -classpath <class search path of directories and zip/jar files>
    --class-path <class search path of directories and zip/jar files>
```

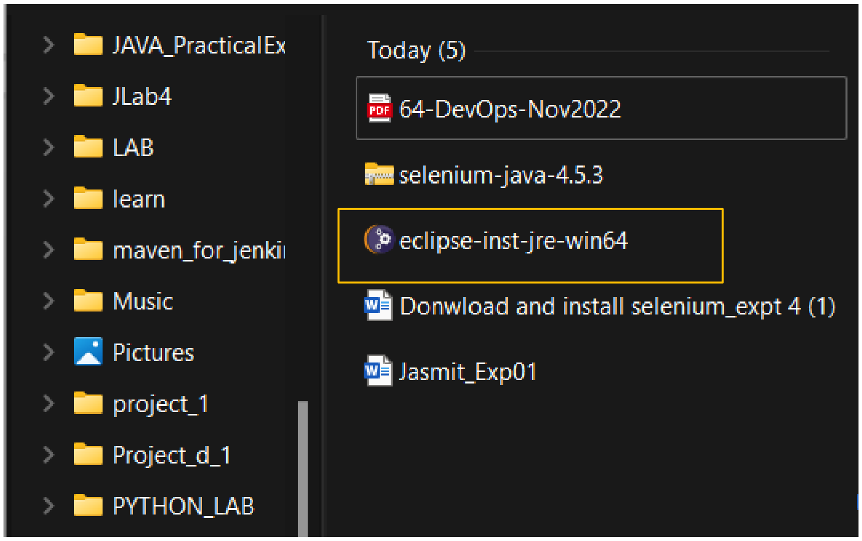
Step 2 – Install Eclipse IDE

Download latest version of **“Eclipse IDE for Java Developers”** here. Be sure to choose correctly between Windows 32 Bit and 64 Bit versions.

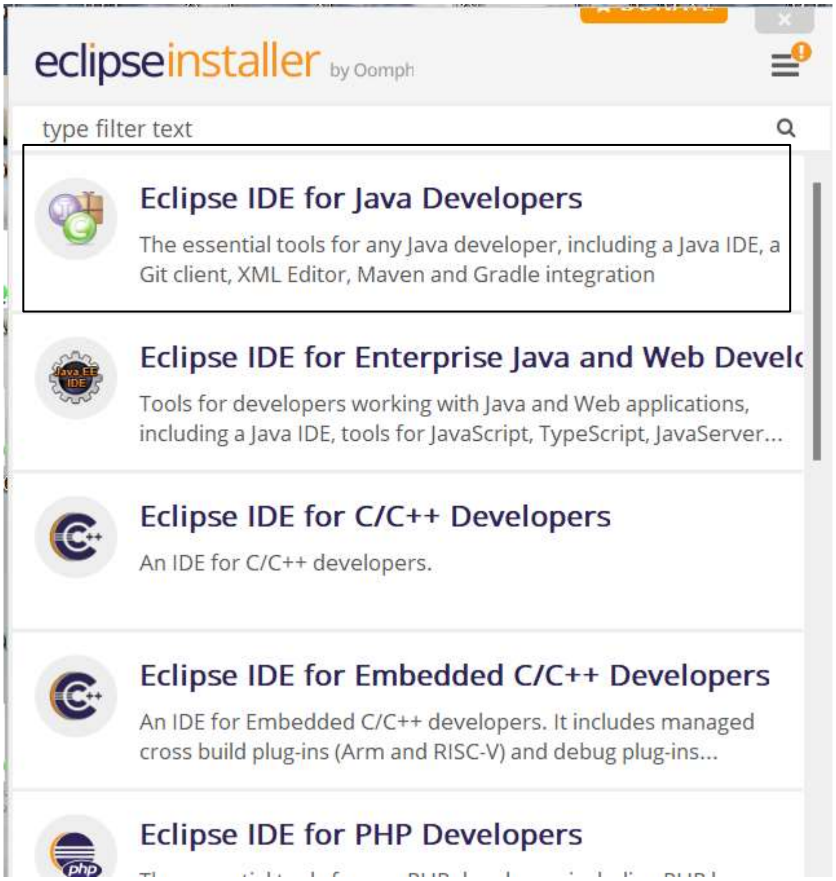


Some Eclipse Foundation pages use cookies to better serve you when you return to the site. You can set your browser to notify you before you receive a cookie or turn off cookies. If you do so, however, some areas of some sites may not function properly. To read Eclipse Foundation Privacy Policy click here.

You should be able to download an exe file named “eclipse-inst-win64” for Setup.



Double-click on file to Install the Eclipse. A new window will open. Click Eclipse IDE for Java Developers.



Eclipse IDE for Java Developers

The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Maven and Gradle integration

Eclipse IDE for Enterprise Java and Web Develc

Tools for developers working with Java and Web applications, including a Java IDE, tools for JavaScript, TypeScript, JavaServer...

Eclipse IDE for C/C++ Developers

An IDE for C/C++ developers.

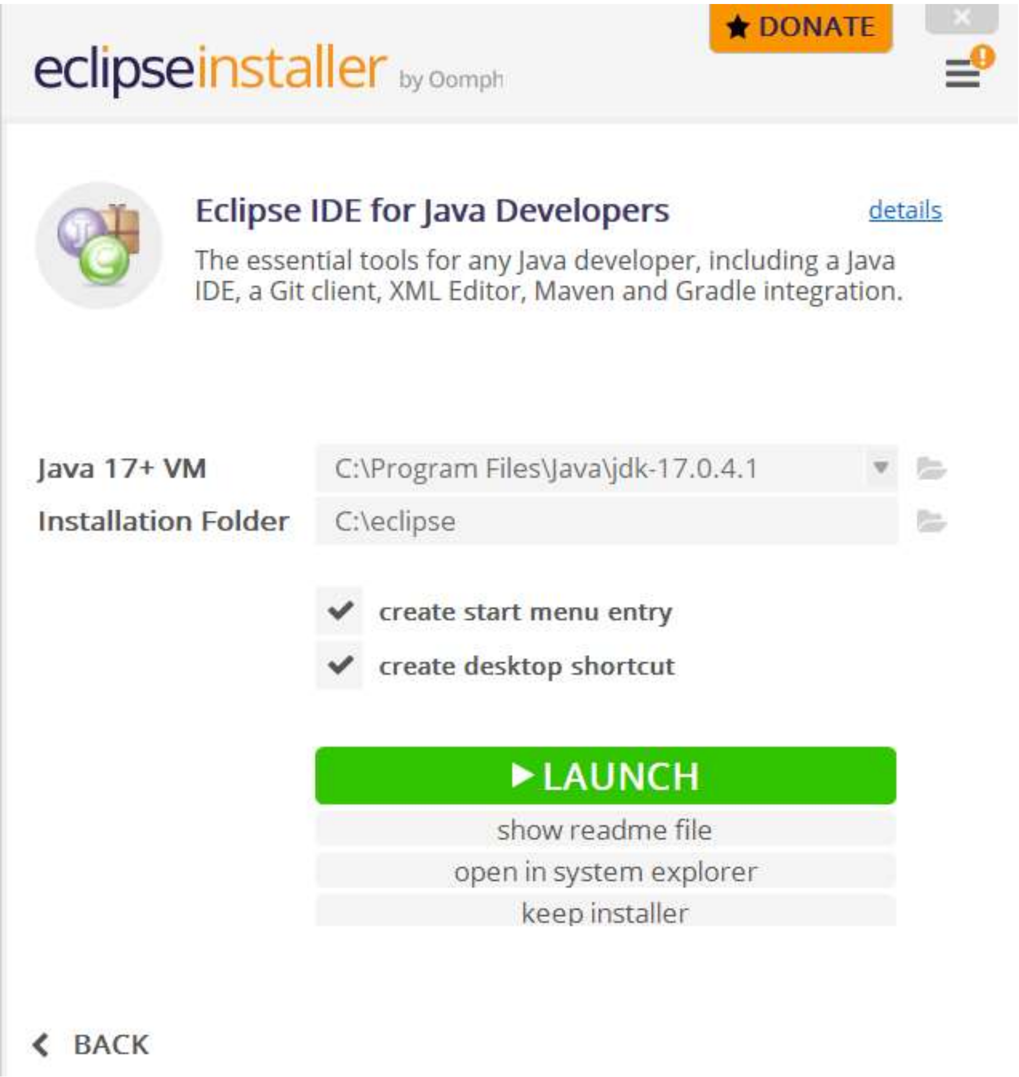
Eclipse IDE for Embedded C/C++ Developers

An IDE for Embedded C/C++ developers. It includes managed cross build plug-ins (Arm and RISC-V) and debug plug-ins...

Eclipse IDE for PHP Developers

After that, a new window will open which click button marked 1 and change path to “C:\eclipse”. Post that Click on Install button marked 2

After successful completion of the installation procedure, a window will appear. On that window click on Launch


eclipseinstaller by Oomph
★ DONATE
Eclipse IDE for Java Developers
details
The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Maven and Gradle integration.

Java 17+ VM: C:\Program Files\Java\jdk-17.0.4.1

Installation Folder: C:\eclipse

☑ create start menu entry

☑ create desktop shortcut

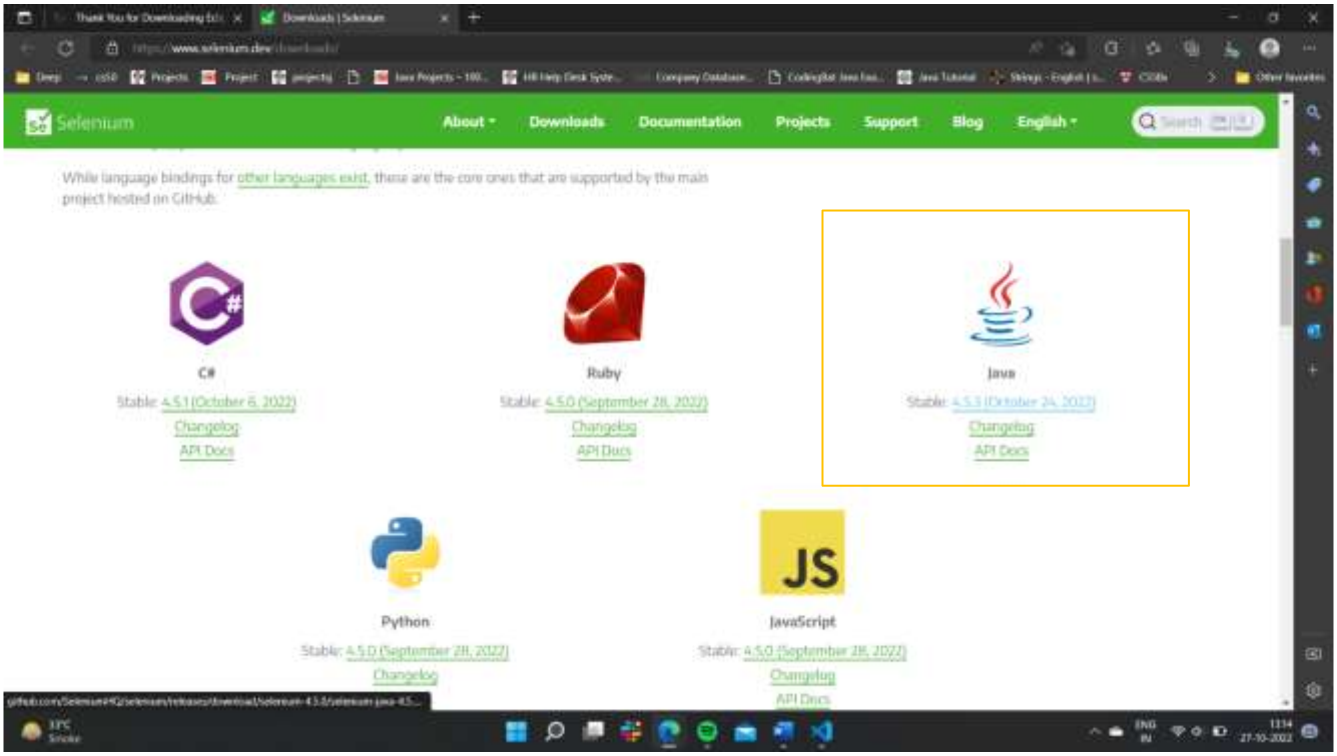
▶ LAUNCH
show readme file
open in system explorer
keep installer
BACK

This will start eclipse neon IDE for you.

Step 3 – Download the Selenium Java Client Driver

You can download **Selenium Webdriver for Java Client Driver** here. You will find client drivers for other languages there, but only choose the one for Java.

This download comes as a ZIP file named “selenium-3.14.0.zip”. For simplicity of Selenium installation on Windows 10, extract the contents of this ZIP file on your C drive so that you would have the directory “C:\selenium-3.14.0\”. This directory contains all the JAR files that we would later import on Eclipse for Selenium setup.

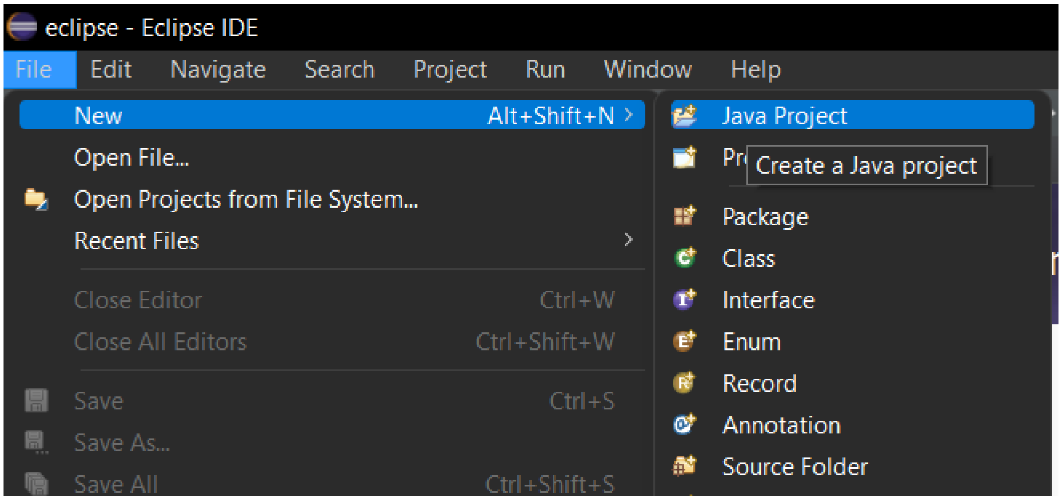


While language bindings for other languages exist, these are the core ones that are supported by the main project hosted on GitHub.

C#
Stable: 4.5.1 (October 6, 2022)
Changelog
API Docs
Ruby
Stable: 4.5.0 (September 28, 2022)
Changelog
API Docs
Java
Stable: 4.5.3 (October 24, 2022)
Changelog
API Docs
Python
Stable: 4.5.0 (September 28, 2022)
Changelog
JS
JavaScript
Stable: 4.5.0 (September 28, 2022)
Changelog
API Docs

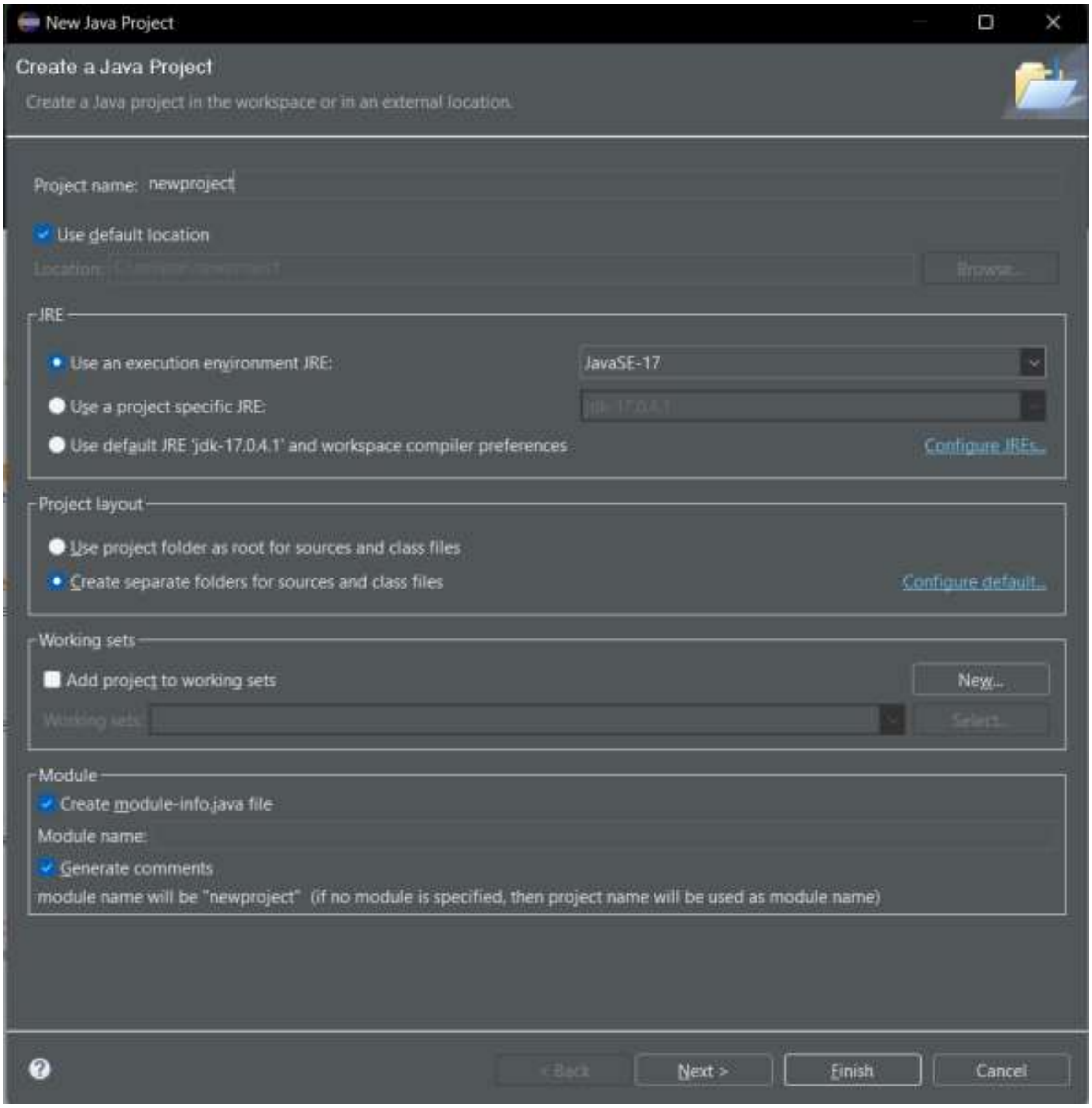
Step 4 – Configure Eclipse IDE with WebDriver

1. Launch the “eclipse.exe” file inside the “eclipse” folder that we extracted in step 2. If you followed step 2 correctly, the executable should be located on C:\eclipse\eclipse.exe.
2. When asked to select for a workspace, just accept the default location.
3. Create a new project through File > New > Java Project. Name the project as“newproject”.



A new pop-up window will open enter details as follow

1. Project Name
2. Location to save project
3. Select an execution JRE
4. Select layout project option
5. Click on Finish button



New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: newproject

☑ Use default location

Location: [illegible]

JRE

● Use an execution environment JRE: JavaSE-17

○ Use a project specific JRE: jdk-17.0.4.1

○ Use default JRE 'jdk-17.0.4.1' and workspace compiler preferences

Project layout

○ Use project folder as root for sources and class files

● Create separate folders for sources and class files

Working sets

□ Add project to working sets

Working sets:

Module

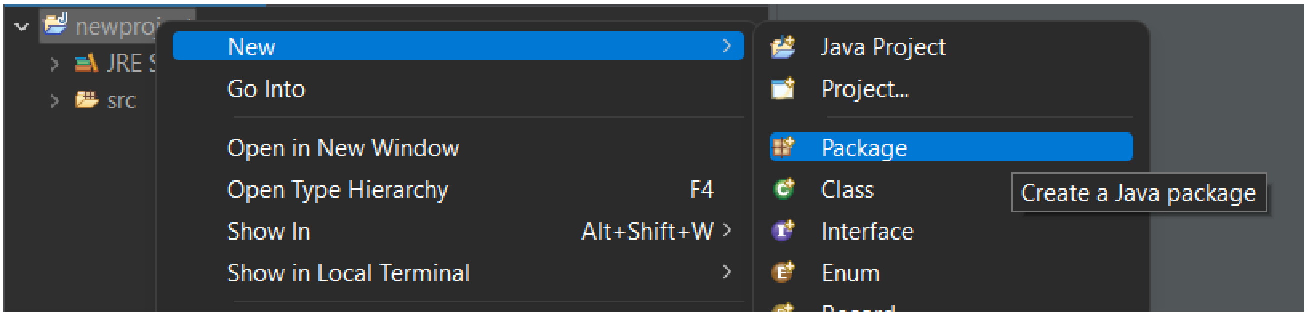
☑ Create module-info.java file

Module name:

☑ Generate comments

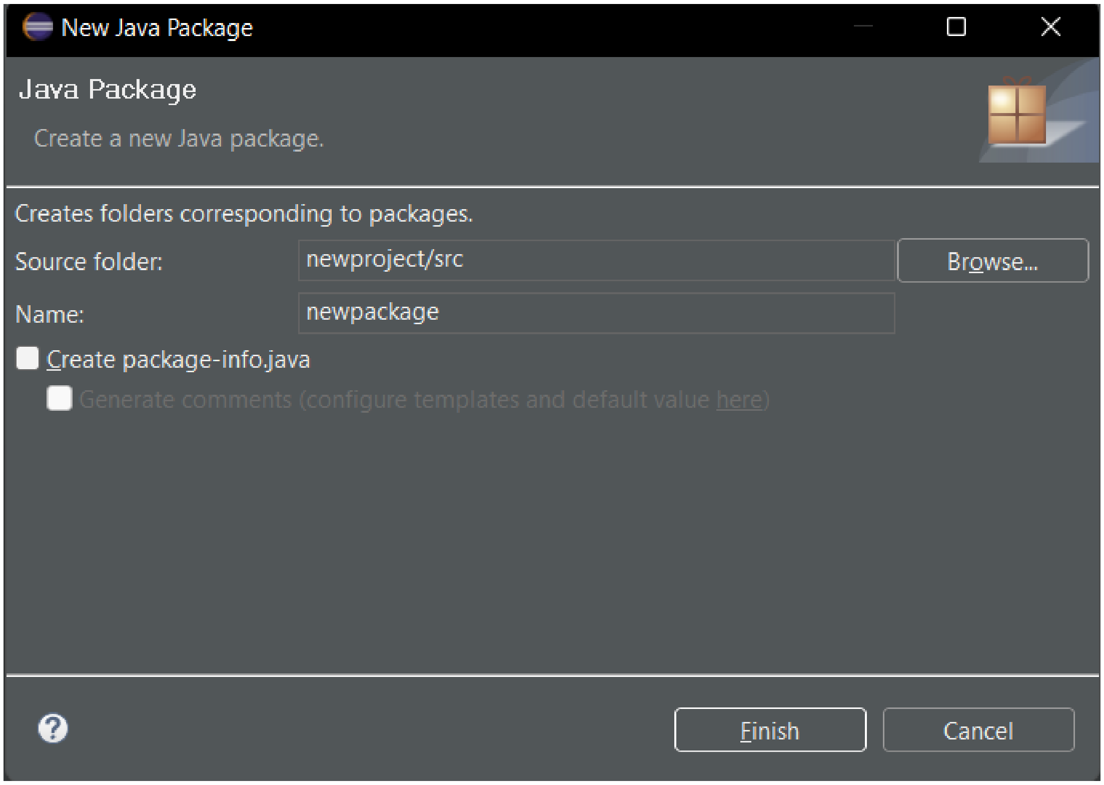
module name will be "newproject" (if no module is specified, then project name will be used as module name)

4. In this step,
   1. Right-click on the newly created project and
   2. Select New > Package, and name that package as “newpackage”.



A pop-up window will open to name the package,

1. Enter the name of the package
2. Click on Finish button



Java Package

Create a new Java package.

Creates folders corresponding to packages.

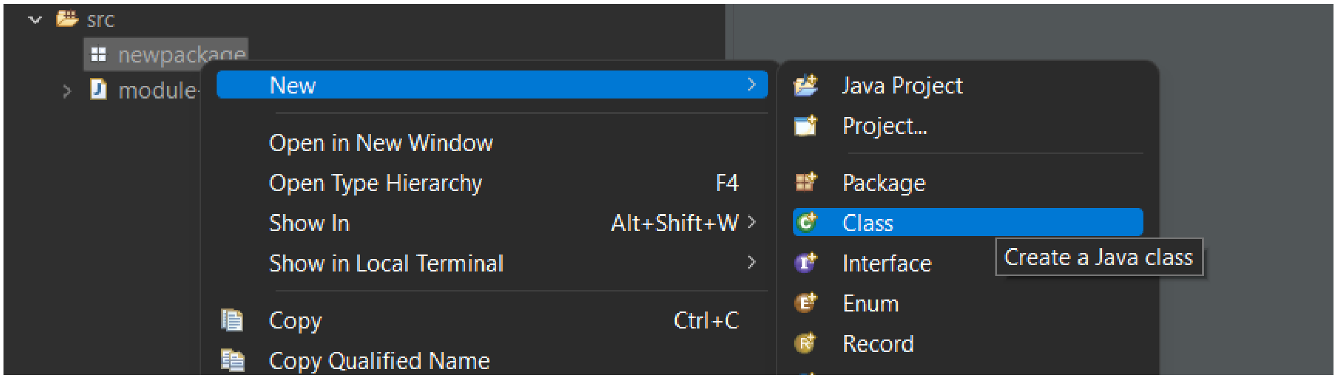
Source folder: newproject/src

Name: newpackage

□ Create package-info.java

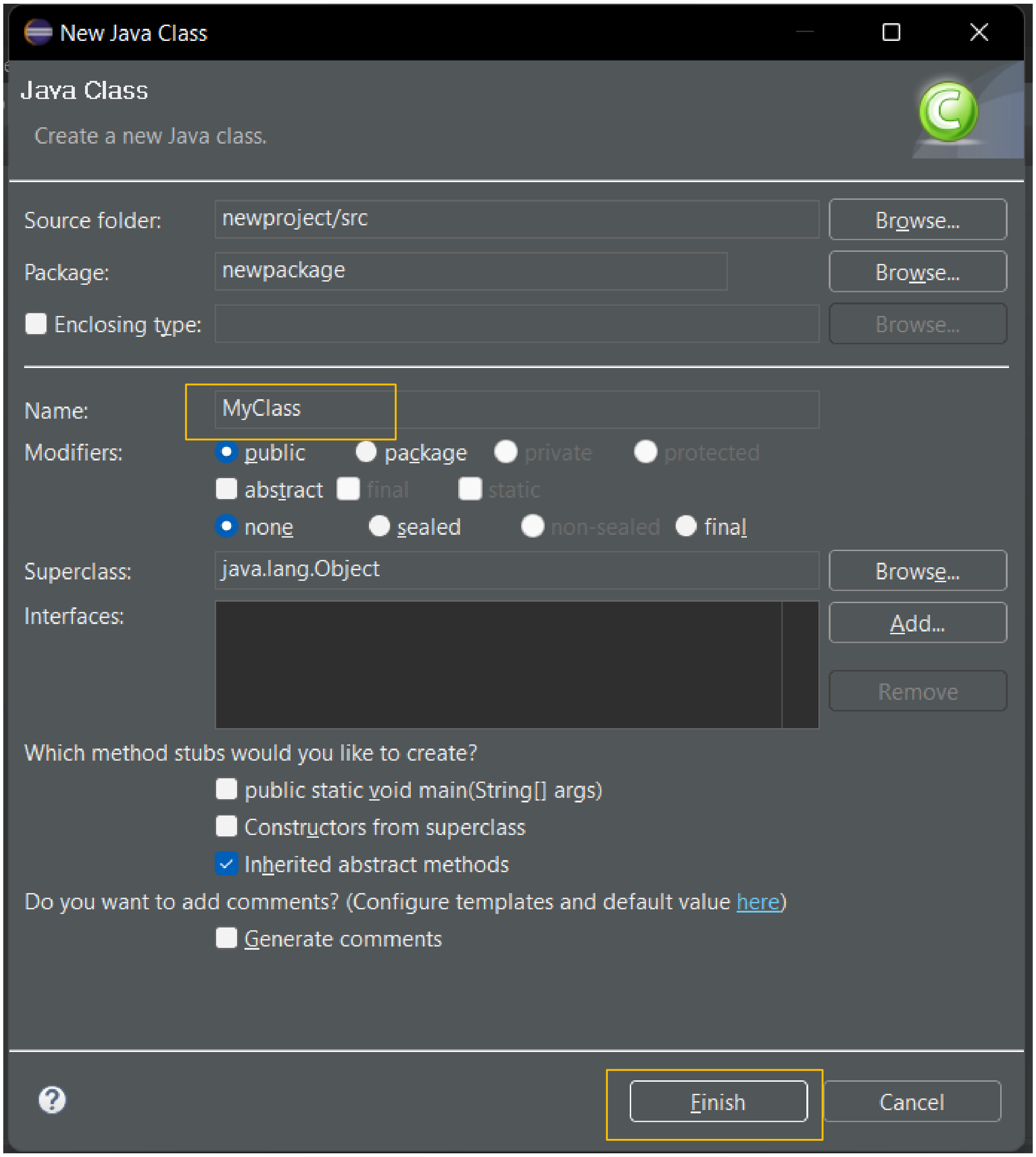
□ Generate comments (configure templates and default value here)

5. Create a new Java class under newpackage by right-clicking on it and then selectingNew > Class, and then name it as “MyClass”. Your Eclipse IDE should look like the image below.



When you click on Class, a pop-up window will open, enter details as

1. Name of the class
2. Click on Finish button



New Java Class

Java Class

Create a new Java class.

Source folder: newproject/src

Package: newpackage

□ Enclosing type:

Name: MyClass

Modifiers: ● public ○ package ○ private ○ protected

□ abstract □ final □ static

● none ○ sealed ○ non-sealed ○ final

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

□ public static void main(String[] args)

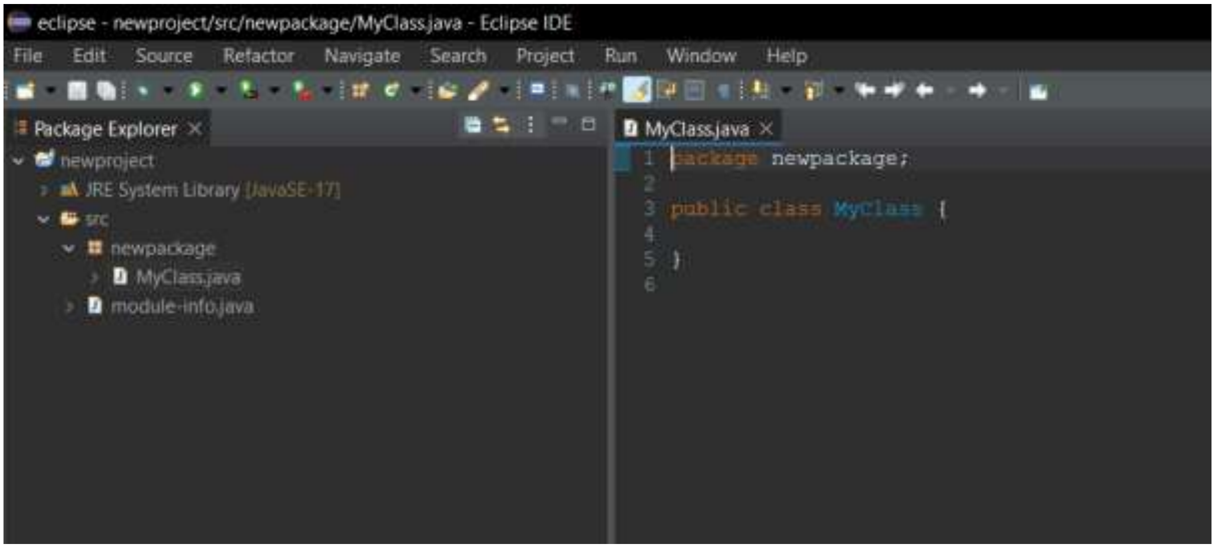
□ Constructors from superclass

☑ Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

□ Generate comments

This is how it looks like a er creating class.

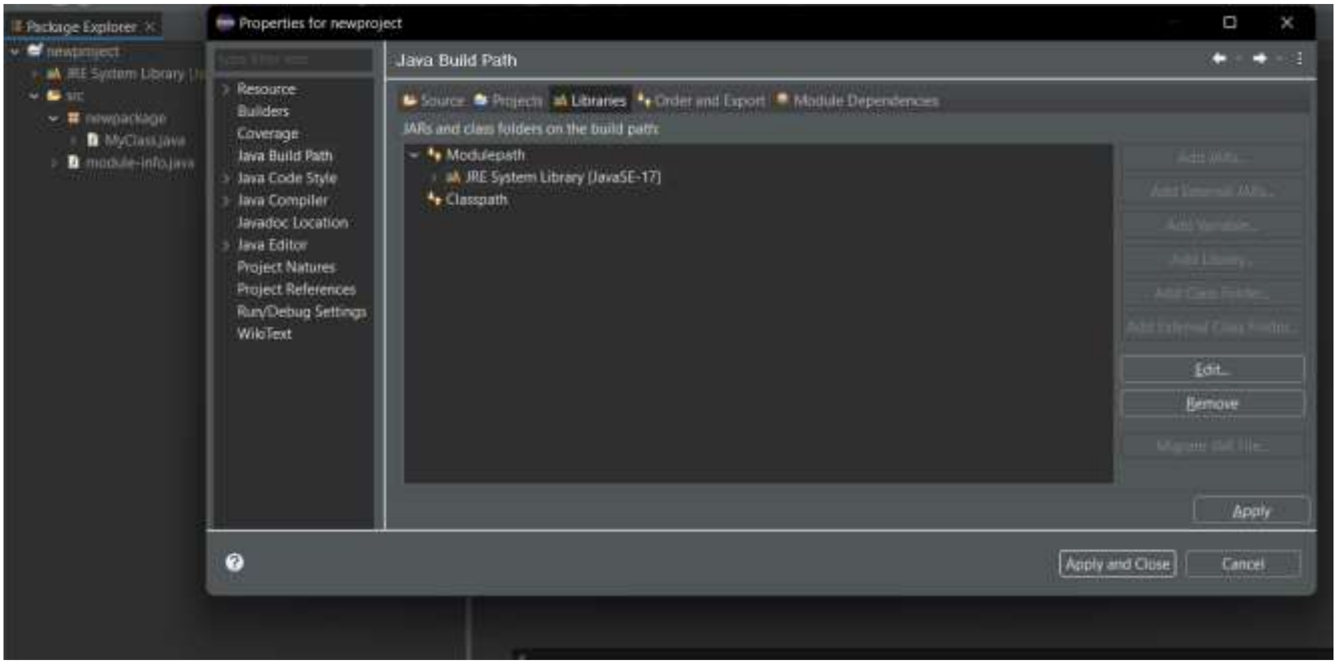

eclipse - newproject/src/newpackage/MyClass.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
newproject
JRE System Library [JavaSE-17]
src
newpackage
MyClass.java
module-info.java
MyClass.java

```
package newpackage;

public class MyClass {

}

```

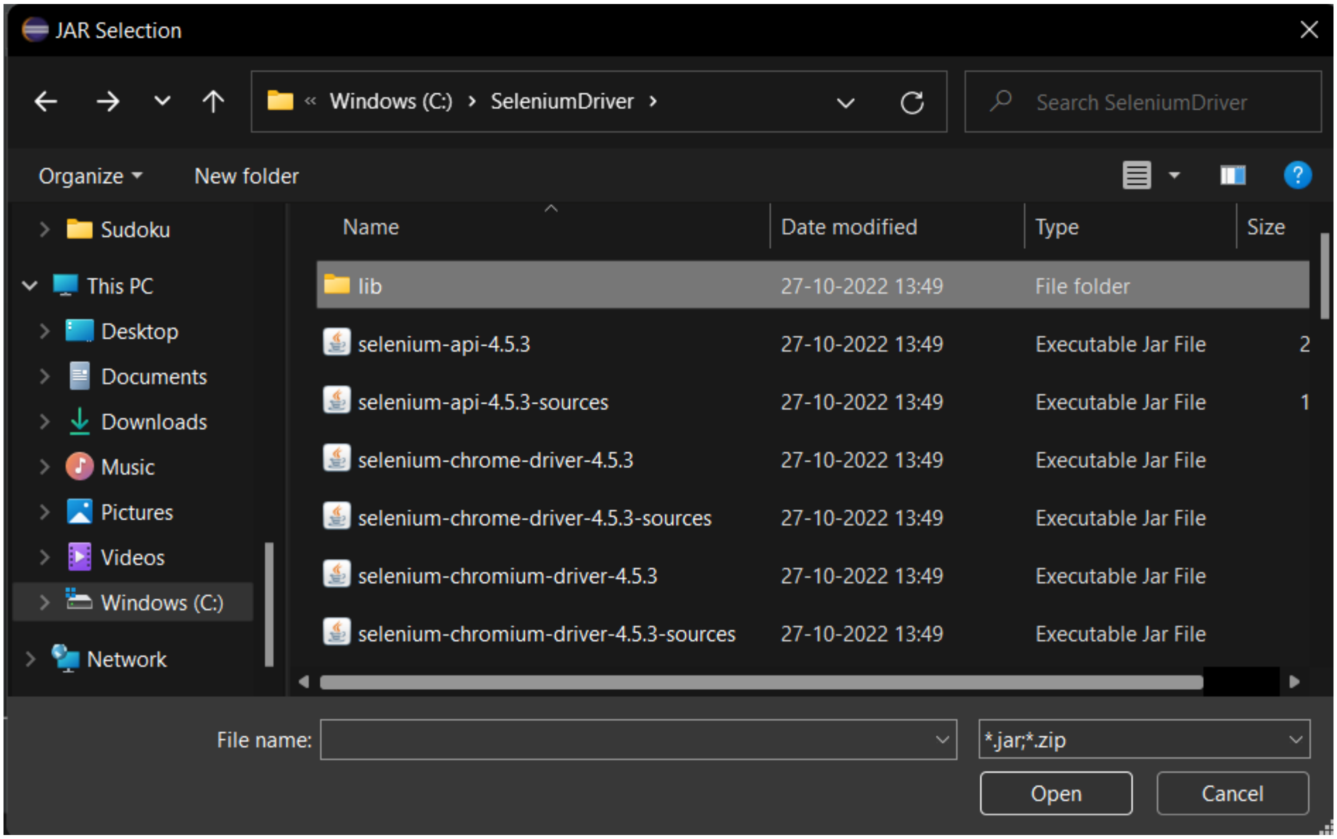
Now selenium WebDriver's into Java Build Path

In this step,

1. Right-click on "newproject" and select **Properties**.
2. On the Properties dialog, click on "Java Build Path".
3. Click on the **Libraries** tab, and then
4. Click on "Add External JARs.."



When you click on "Add External JARs.." It will open a pop-up window. Select the JAR files you want to add.



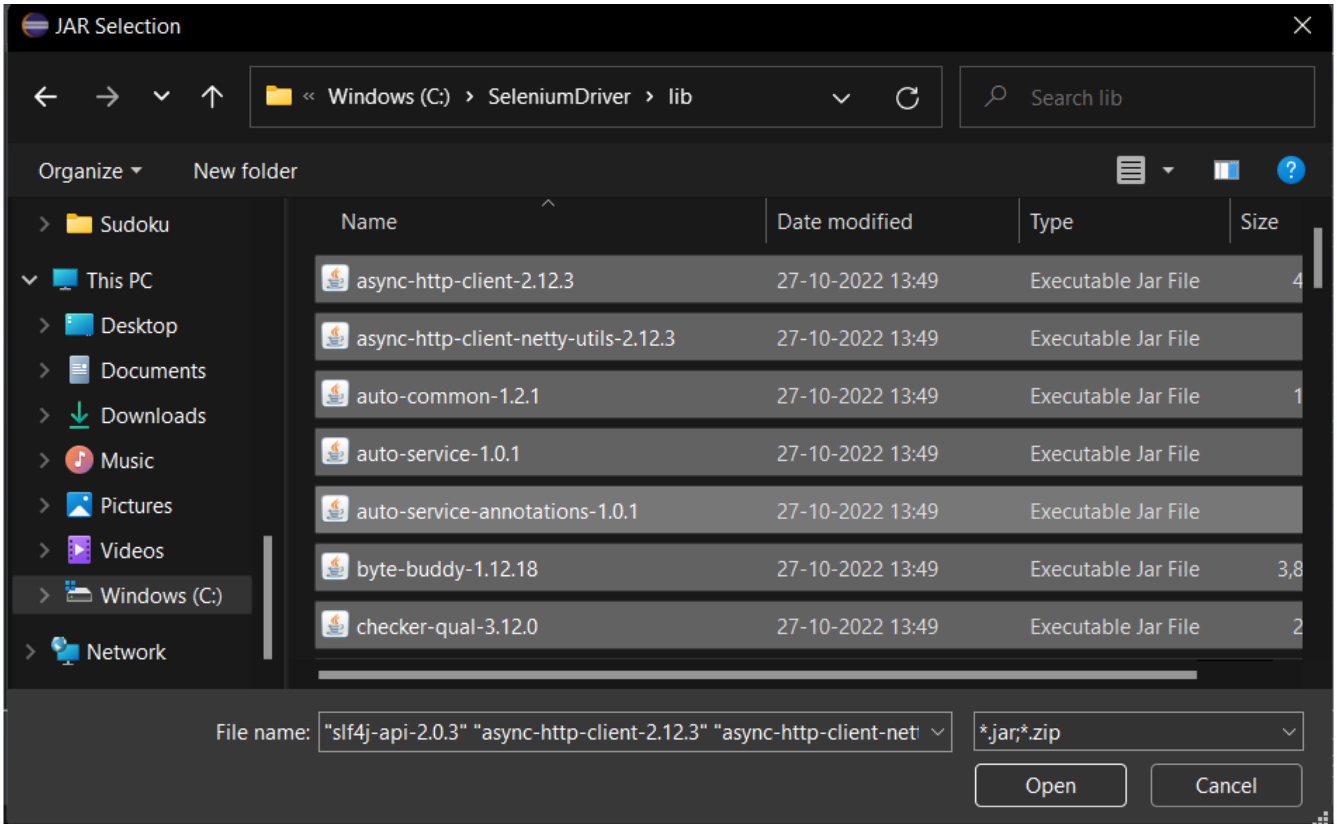
Name	Date modified	Type	Size
lib	27-10-2022 13:49	File folder	
selenium-api-4.5.3	27-10-2022 13:49	Executable Jar File	2
selenium-api-4.5.3-sources	27-10-2022 13:49	Executable Jar File	1
selenium-chrome-driver-4.5.3	27-10-2022 13:49	Executable Jar File	
selenium-chrome-driver-4.5.3-sources	27-10-2022 13:49	Executable Jar File	
selenium-chromium-driver-4.5.3	27-10-2022 13:49	Executable Jar File	
selenium-chromium-driver-4.5.3-sources	27-10-2022 13:49	Executable Jar File	

File name:

.jar;.zip

After selecting jar files, click on OK button.

Select all files inside the lib folder.



Windows (C:) › SeleniumDriver › lib

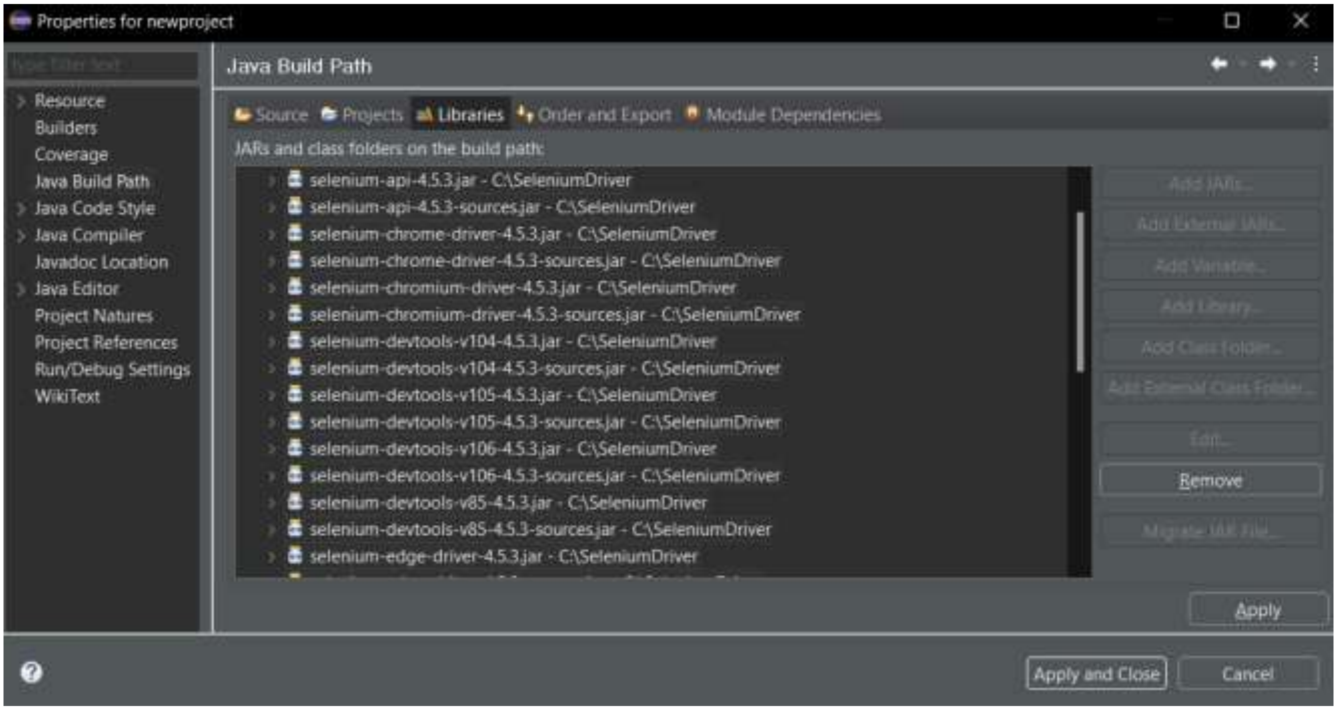
Name	Date modified	Type	Size
async-http-client-2.12.3	27-10-2022 13:49	Executable Jar File	4
async-http-client-netty-utils-2.12.3	27-10-2022 13:49	Executable Jar File	
auto-common-1.2.1	27-10-2022 13:49	Executable Jar File	1
auto-service-1.0.1	27-10-2022 13:49	Executable Jar File	
auto-service-annotations-1.0.1	27-10-2022 13:49	Executable Jar File	
byte-buddy-1.12.18	27-10-2022 13:49	Executable Jar File	3,8
checker-qual-3.12.0	27-10-2022 13:49	Executable Jar File	2

File name: "slf4j-api-2.0.3" "async-http-client-2.12.3" "async-http-client-nett

.jar;.zip

Select files outside lib folder

Once done, click “Apply and Close” button

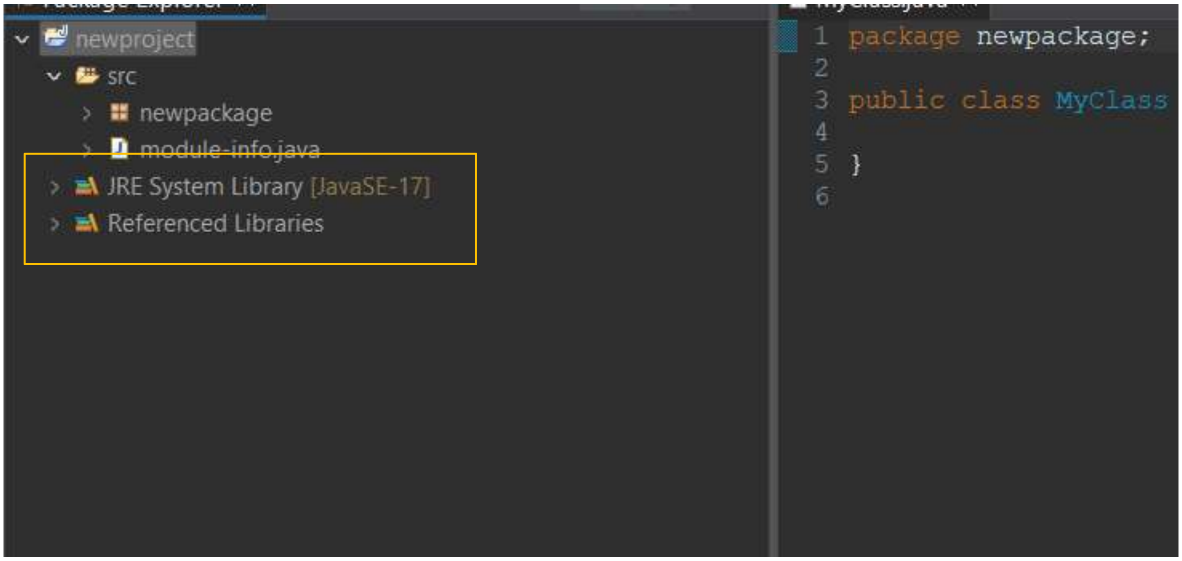


Java Build Path

JARs and class folders on the build path:

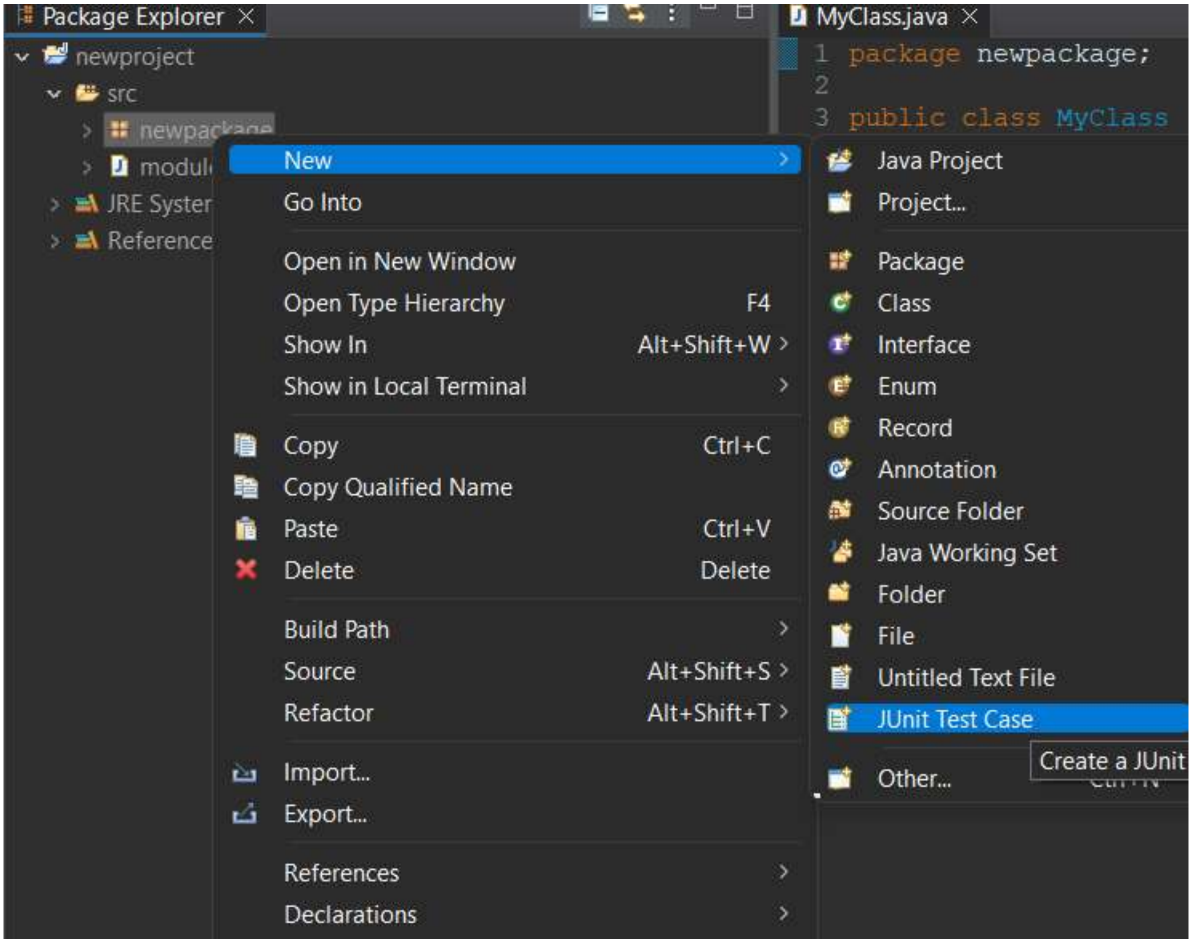
- selenium-api-4.5.3.jar - C:\SeleniumDriver
- selenium-api-4.5.3-sources.jar - C:\SeleniumDriver
- selenium-chrome-driver-4.5.3.jar - C:\SeleniumDriver
- selenium-chrome-driver-4.5.3-sources.jar - C:\SeleniumDriver
- selenium-chromium-driver-4.5.3.jar - C:\SeleniumDriver
- selenium-chromium-driver-4.5.3-sources.jar - C:\SeleniumDriver
- selenium-devtools-v104-4.5.3.jar - C:\SeleniumDriver
- selenium-devtools-v104-4.5.3-sources.jar - C:\SeleniumDriver
- selenium-devtools-v105-4.5.3.jar - C:\SeleniumDriver
- selenium-devtools-v105-4.5.3-sources.jar - C:\SeleniumDriver
- selenium-devtools-v106-4.5.3.jar - C:\SeleniumDriver
- selenium-devtools-v106-4.5.3-sources.jar - C:\SeleniumDriver
- selenium-devtools-v85-4.5.3.jar - C:\SeleniumDriver
- selenium-devtools-v85-4.5.3-sources.jar - C:\SeleniumDriver
- selenium-edge-driver-4.5.3.jar - C:\SeleniumDriver

6. Add all the JAR files inside and outside the “libs” folder. Your Properties dialog shouldnow look similar to the image below.

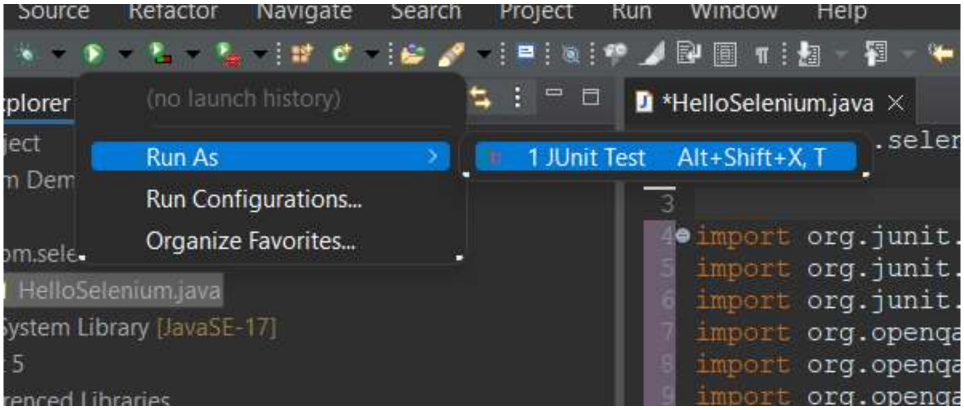


```
package newpackage;

public class MyClass

}

```

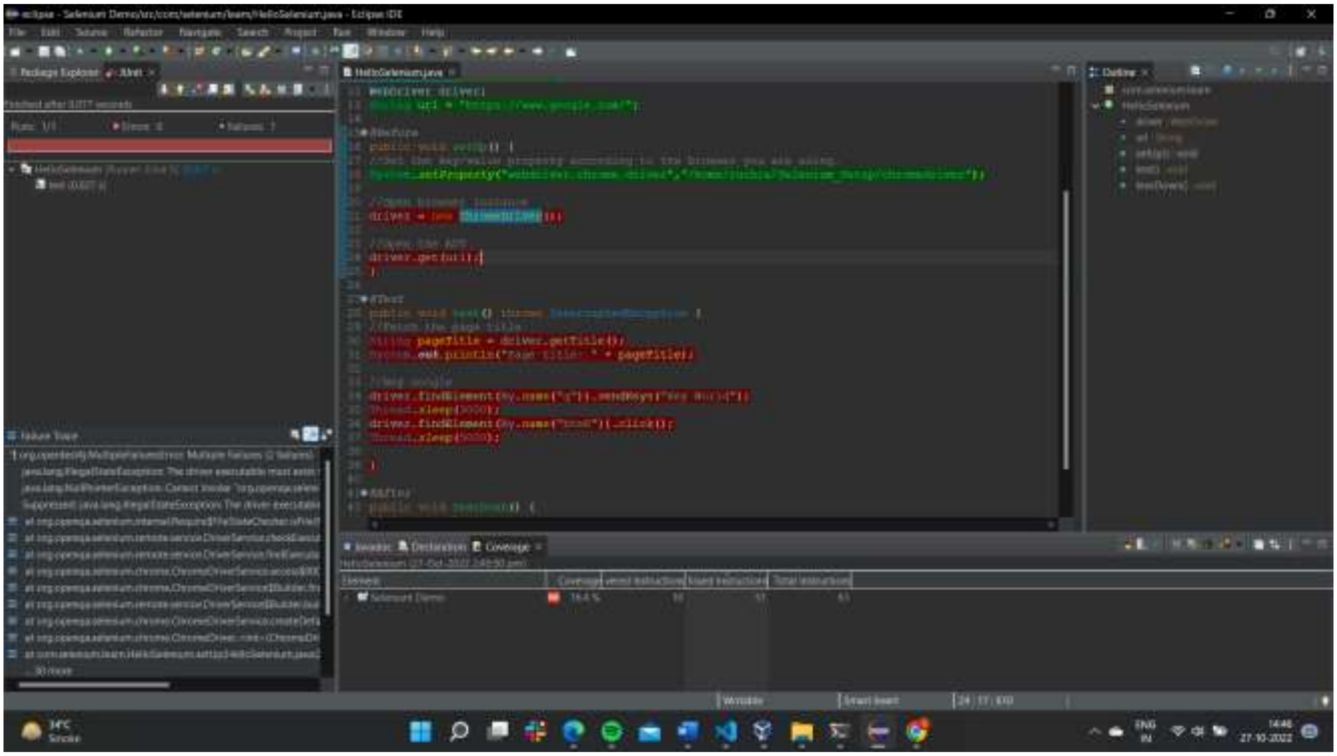
7. Finally, click OK and we are done importing Selenium libraries into our project.



```
package newpackage;

public class MyClass
```



```
.selen

import org.junit.
import org.junit.
import org.junit.
import org.openqa
import org.openqa
import org.openqa
```



```
WebDriver driver;
String url = "https://www.google.com/";

@Before
public void setUp() {
//Set the key/value property according to the browser you are using.
System.setProperty("webdriver.chrome.driver","/home/[illegible]/Selenium_Setup/chromedriver");

//Open browser instance
driver = new ChromeDriver();

//Open the AUT
driver.get(url);
}

@Test
public void test() throws InterruptedException {
//Fetch the page title
String pageTitle = driver.getTitle();
System.out.println("Page title: " + pageTitle);

//[illegible] google
driver.findElement(By.name("q")).sendKeys("[illegible]");
Thread.sleep(3000);
driver.findElement(By.name("btnK")).click();
Thread.sleep(5000);

}

@After
public void tearDown() {
```

Conclusion: Thus installation and first program on Selenium was run successfully

References :

1. https://drive.google.com/drive/folders/113WPhiDvpeLb7lm3goo7vtNT0kTAjXIH?usp=s haring
2. https://www.youtube.com/watch?v=MUTBV1RJBiQ&t=511s
3. https://www.guru99.com/installing-selenium-webdriver.html