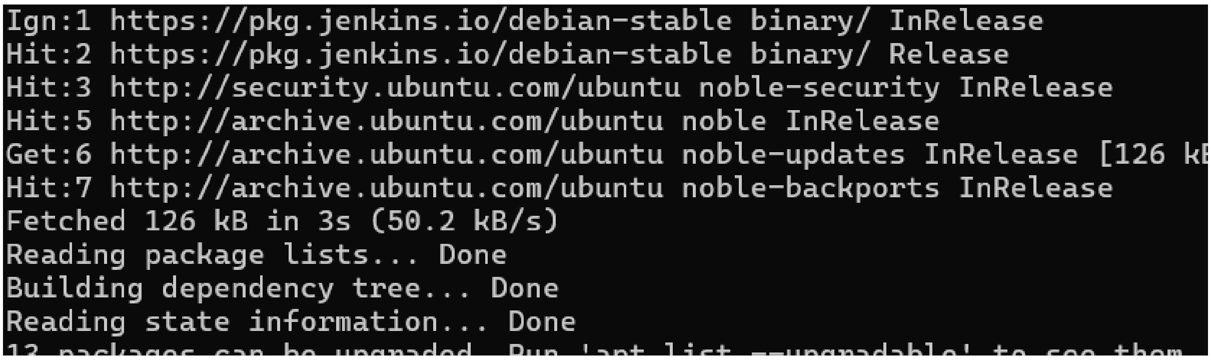
DAY-2

DEVOPS

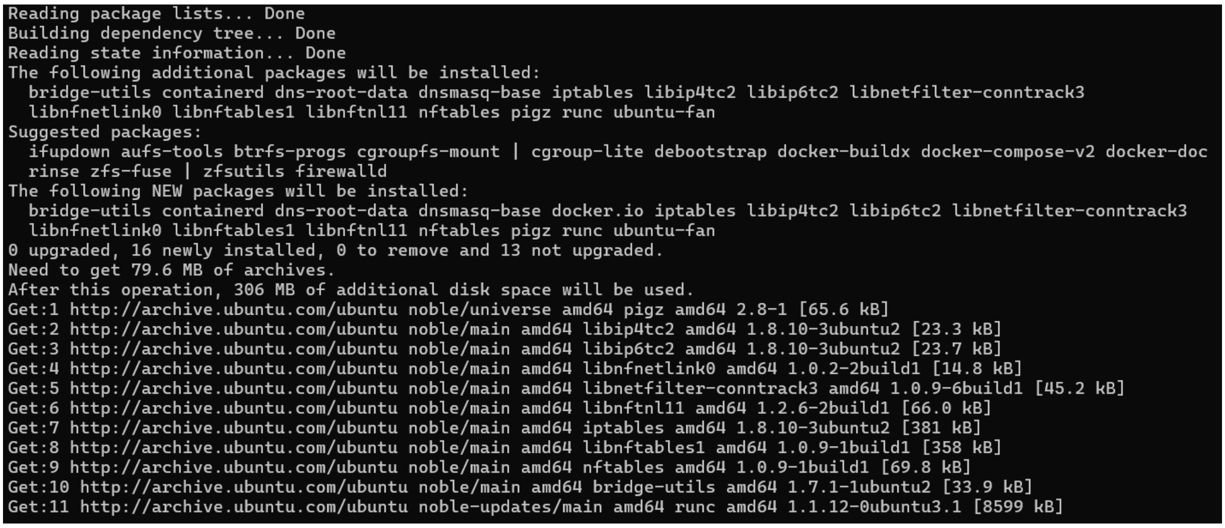
INSTALL DOCKER

1) sudo apt update



```
Ign:1 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:2 https://pkg.jenkins.io/debian-stable binary/ Release
Hit:3 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 http://archive.ubuntu.com/ubuntu noble InRelease
Get:6 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB
Hit:7 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Fetched 126 kB in 3s (50.2 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
13 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

2) sudo apt install -y docker.io

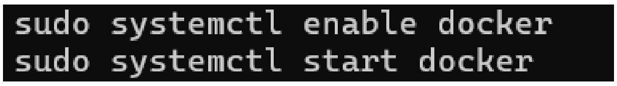


```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base iptables libip4tc2 libip6tc2 libnetfilter-conntrack3
  libnfnetlink0 libnftables1 libnftnl11 nftables pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools btrfs-progs cgroupfs-mount | cgroup-lite debootstrap docker-buildx docker-compose-v2 docker-doc
  rinse zfs-fuse | zfsutils firewalld
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io iptables libip4tc2 libip6tc2 libnetfilter-conntrack3
  libnfnetlink0 libnftables1 libnftnl11 nftables pigz runc ubuntu-fan
0 upgraded, 16 newly installed, 0 to remove and 13 not upgraded.
Need to get 79.6 MB of archives.
After this operation, 306 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble/main amd64 libip4tc2 amd64 1.8.10-3ubuntu2 [23.3 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble/main amd64 libip6tc2 amd64 1.8.10-3ubuntu2 [23.7 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble/main amd64 libnfnetlink0 amd64 1.0.2-2build1 [14.8 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble/main amd64 libnetfilter-conntrack3 amd64 1.0.9-6build1 [45.2 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble/main amd64 libnftnl11 amd64 1.2.6-2build1 [66.0 kB]
Get:7 http://archive.ubuntu.com/ubuntu noble/main amd64 iptables amd64 1.8.10-3ubuntu2 [381 kB]
Get:8 http://archive.ubuntu.com/ubuntu noble/main amd64 libnftables1 amd64 1.0.9-1build1 [358 kB]
Get:9 http://archive.ubuntu.com/ubuntu noble/main amd64 nftables amd64 1.0.9-1build1 [69.8 kB]
Get:10 http://archive.ubuntu.com/ubuntu noble/main amd64 bridge-utils amd64 1.7.1-1ubuntu2 [33.9 kB]
Get:11 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.1.12-0ubuntu3.1 [8599 kB]
```

ENABLE AND DISABLE

1) sudo systemctl enable docker

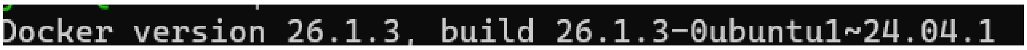
2)sudo systemctl start docker



```
sudo systemctl enable docker
sudo systemctl start docker
```

VERIFY THE INSTALLATION

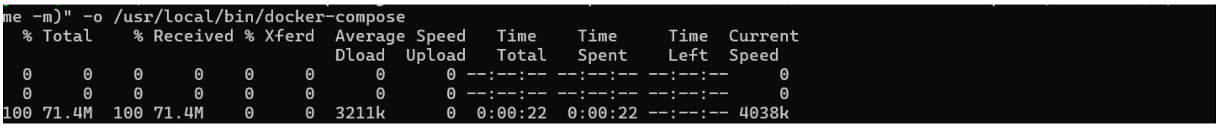
docker –version



```
Docker version 26.1.3, build 26.1.3-0ubuntu1~24.04.1
```

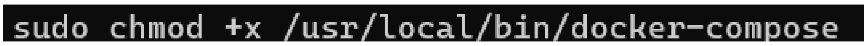
INSTALL DOCKER-COMPOSE

sudo curl -L "https://github.com/docker/compose/releases/latest/download/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose



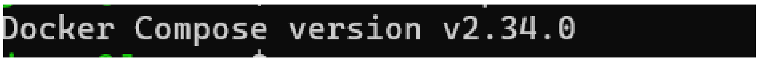
```
me -m)" -o /usr/local/bin/docker-compose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 71.4M  100 71.4M    0     0  3211k      0  0:00:22  0:00:22 --:--:-- 4038k
```

Give execution permission:



```
sudo chmod +x /usr/local/bin/docker-compose
```

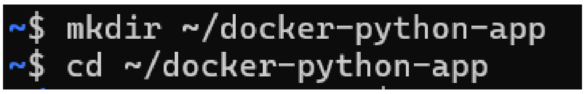
VERIFY INSTALLATION



```
Docker Compose version v2.34.0
```

CREATE AN “HELLO WOLRD: APPLICATION

Create a project directory



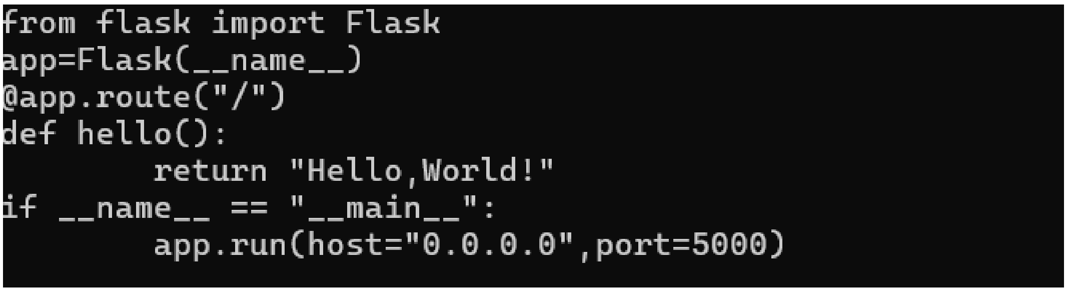
```
~$ mkdir ~/docker-python-app
~$ cd ~/docker-python-app
```

Create the python Application File

Create a file

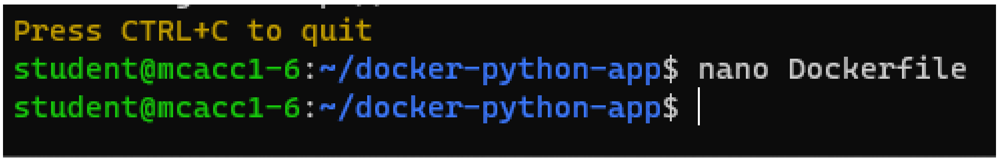
nano app.py

cat app.py

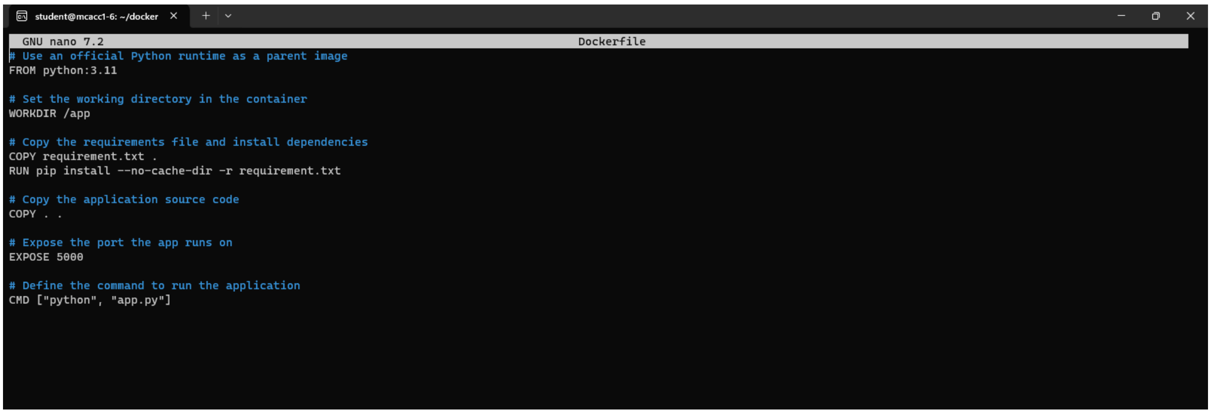


```
from flask import Flask
app=Flask(__name__)
@app.route("/")
def hello():
        return "Hello,World!"
if __name__ == "__main__":
        app.run(host="0.0.0.0",port=5000)
```

Create a Docker file



```
Press CTRL+C to quit
student@mcacc1-6:~/docker-python-app$ nano Dockerfile
student@mcacc1-6:~/docker-python-app$
```



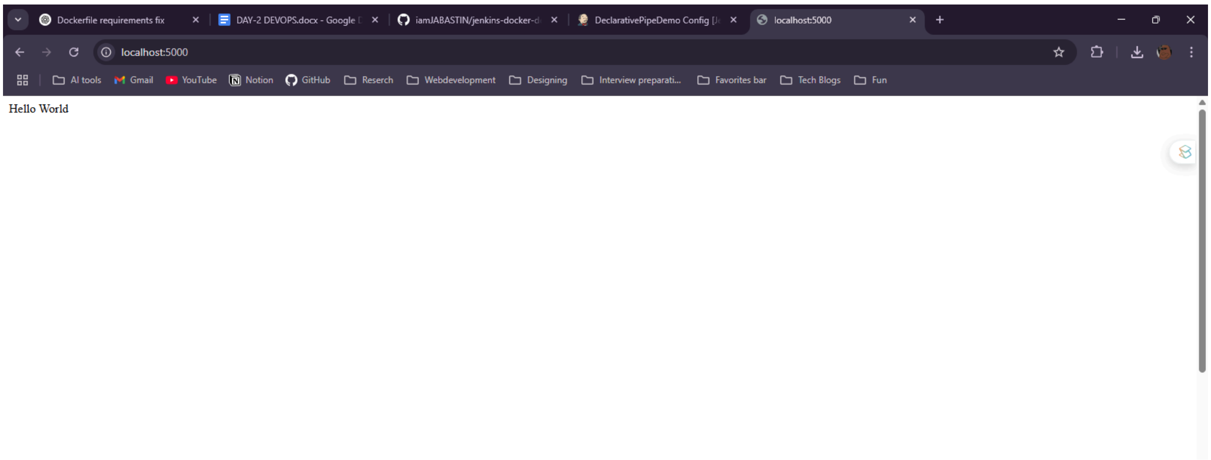
```
  GNU nano 7.2                                                   Dockerfile
# Use an official Python runtime as a parent image
FROM python:3.11

# Set the working directory in the container
WORKDIR /app

# Copy the requirements file and install dependencies
COPY requirement.txt .
RUN pip install --no-cache-dir -r requirement.txt

# Copy the application source code
COPY . .

# Expose the port the app runs on
EXPOSE 5000

# Define the command to run the application
CMD ["python", "app.py"]
```

sudo docker build -t test .

sudo docker run -d -p 5000:5000 test

After the last command we run the last command a random number will be generated.

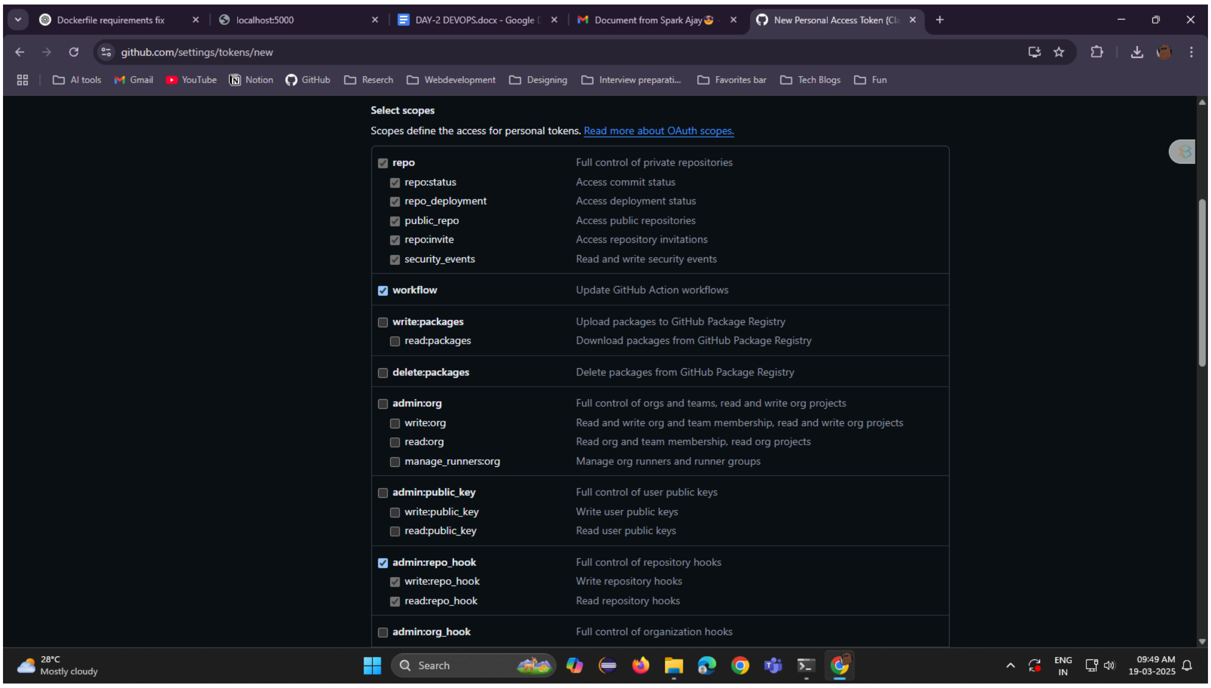
Then, We need to check the **localhost:5000** – Where we can able to see the log message.



Hello World

CREATE A GITHUB REPO

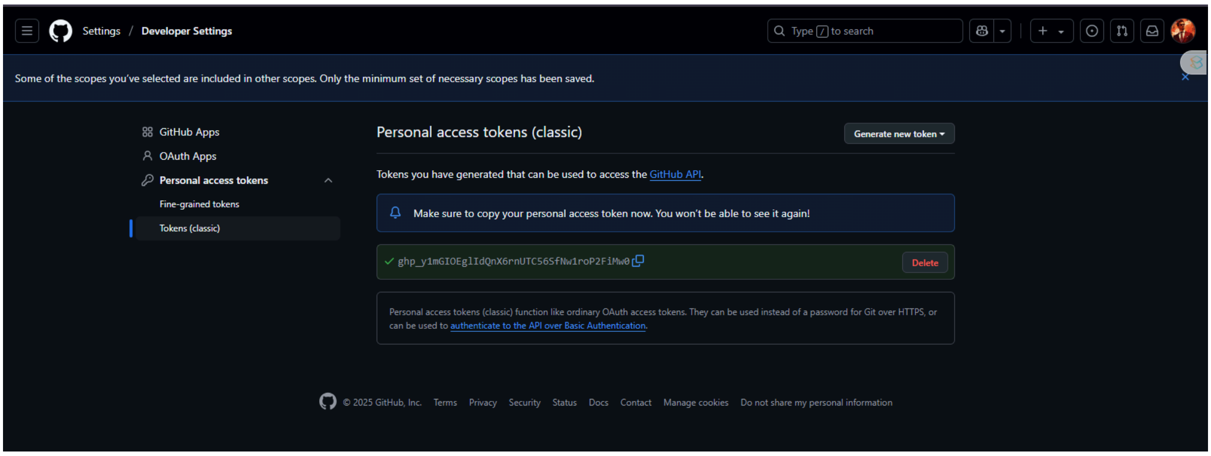
Generate a PAT key to give Jenkins access.



Select scopes

Scopes define the access for personal tokens. Read more about OAuth scopes.

☑ **repo**	Full control of private repositories
☑ repo:status	Access commit status
☑ repo_deployment	Access deployment status
☑ public_repo	Access public repositories
☑ repo:invite	Access repository invitations
☑ security_events	Read and write security events
☑ **workflow**	Update GitHub Action workflows
□ **write:packages**	Upload packages to GitHub Package Registry
□ read:packages	Download packages from GitHub Package Registry
□ **delete:packages**	Delete packages from GitHub Package Registry
□ **admin:org**	Full control of orgs and teams, read and write org projects
□ write:org	Read and write org and team membership, read and write org projects
□ read:org	Read org and team membership, read org projects
□ manage_runners:org	Manage org runners and runner groups
□ **admin:public_key**	Full control of user public keys
□ write:public_key	Write user public keys
□ read:public_key	Read user public keys
☑ **admin:repo_hook**	Full control of repository hooks
☑ write:repo_hook	Write repository hooks
☑ read:repo_hook	Read repository hooks
□ **admin:org_hook**	Full control of organization hooks



Some of the scopes you've selected are included in other scopes. Only the minimum set of necessary scopes has been saved.

Personal access tokens (classic)

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

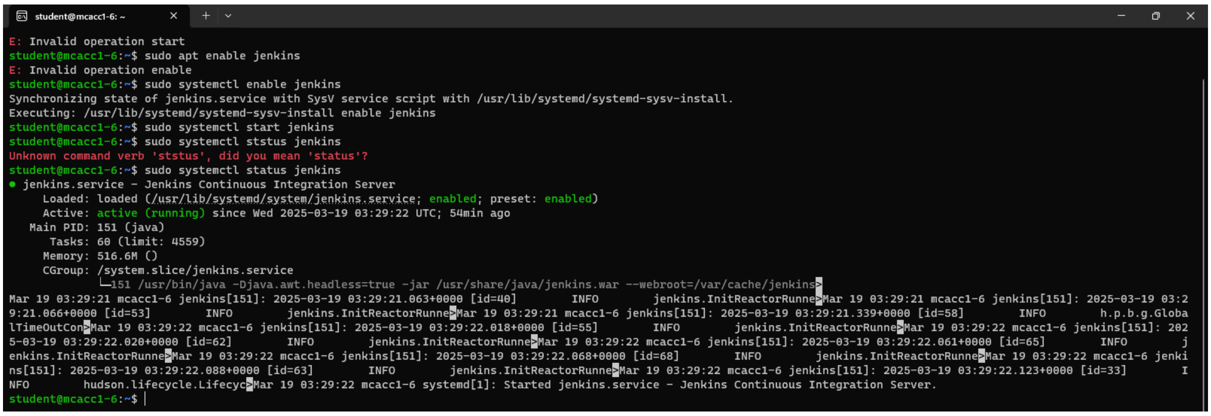
Make sure to copy your personal access token now. You won't be able to see it again!

✓ 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

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Save the PAT key for later use.

RUN THE JENKINS SERVER



student@mcacc1-6: ~

```
E: Invalid operation start
student@mcacc1-6:~$ sudo apt enable jenkins
E: Invalid operation enable
student@mcacc1-6:~$ sudo systemctl enable jenkins
Synchronizing state of jenkins.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable jenkins
student@mcacc1-6:~$ sudo systemctl start jenkins
student@mcacc1-6:~$ sudo systemctl ststus jenkins
Unknown command verb 'ststus', did you mean 'status'?
student@mcacc1-6:~$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: enabled)
     Active: active (running) since Wed 2025-03-19 03:29:22 UTC; 54min ago
   Main PID: 151 (java)
      Tasks: 60 (limit: 4559)
     Memory: 516.6M ()
     CGroup: /system.slice/jenkins.service
             └─151 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins>
Mar 19 03:29:21 mcacc1-6 jenkins[151]: 2025-03-19 03:29:21.063+0000 [id=40]        INFO        jenkins.InitReactorRunne>Mar 19 03:29:21 mcacc1-6 jenkins[151]: 2025-03-19 03:2
9:21.066+0000 [id=53]        INFO        jenkins.InitReactorRunne>Mar 19 03:29:21 mcacc1-6 jenkins[151]: 2025-03-19 03:29:21.339+0000 [id=58]        INFO        h.p.b.g.Globa
lTimeOutCon>Mar 19 03:29:22 mcacc1-6 jenkins[151]: 2025-03-19 03:29:22.018+0000 [id=55]        INFO        jenkins.InitReactorRunne>Mar 19 03:29:22 mcacc1-6 jenkins[151]: 202
5-03-19 03:29:22.020+0000 [id=62]        INFO        jenkins.InitReactorRunne>Mar 19 03:29:22 mcacc1-6 jenkins[151]: 2025-03-19 03:29:22.061+0000 [id=65]        INFO        j
enkins.InitReactorRunne>Mar 19 03:29:22 mcacc1-6 jenkins[151]: 2025-03-19 03:29:22.068+0000 [id=68]        INFO        jenkins.InitReactorRunne>Mar 19 03:29:22 mcacc1-6 jenki
ns[151]: 2025-03-19 03:29:22.088+0000 [id=63]        INFO        jenkins.InitReactorRunne>Mar 19 03:29:22 mcacc1-6 jenkins[151]: 2025-03-19 03:29:22.123+0000 [id=33]        I
NFO        hudson.lifecycle.Lifecyc>Mar 19 03:29:22 mcacc1-6 systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
student@mcacc1-6:~$
```

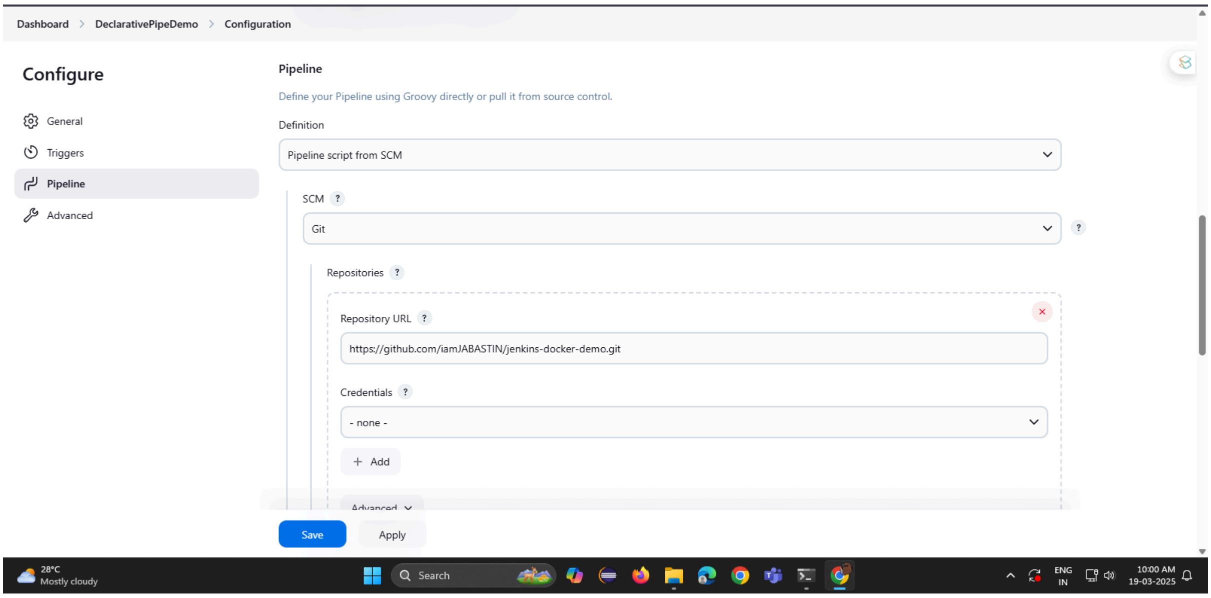
CREATE A NEW ITEM WITH PIPE LINE CONFIG

After, Choose **Pipeline** > **Definition** > choose **Pipeline Script from SCM**

in **SCM** > choose **GIT**

In Repositories section

Repositories URL > paste the repo URL > https://github.com/iamJABASTIN/jenkins-docker-demo.git



Dashboard > DeclarativePipeDemo > Configuration

Configure

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition: Pipeline script from SCM

SCM: Git

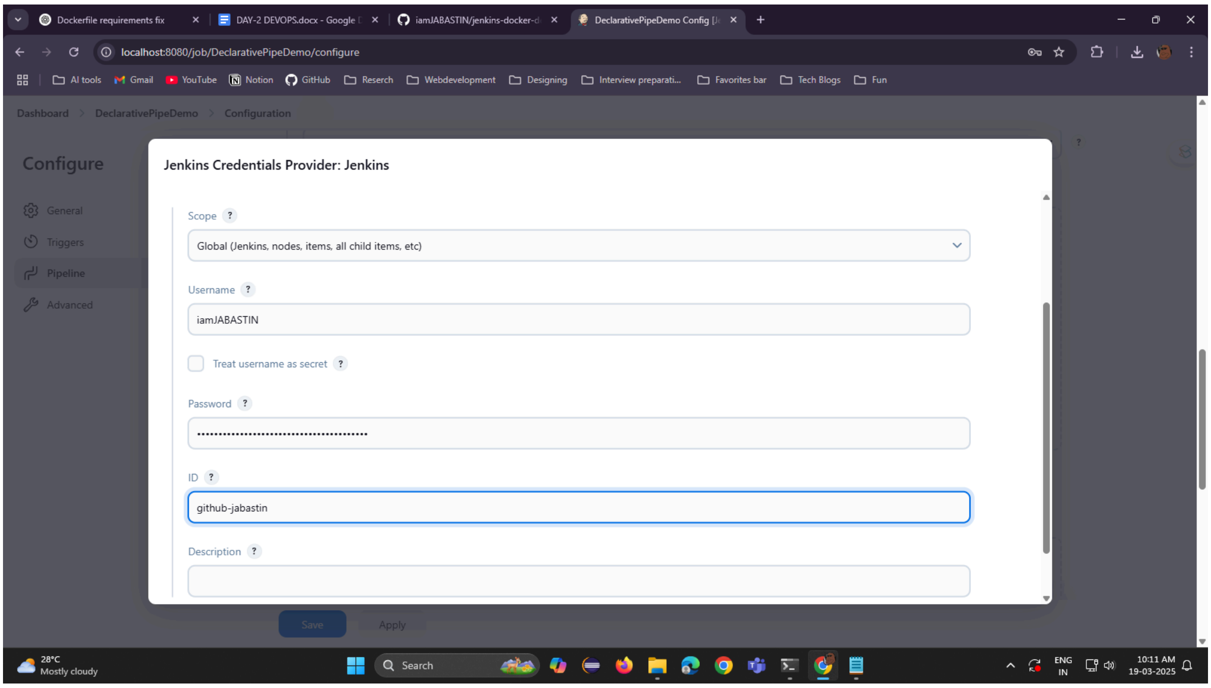
Repositories

Repository URL: https://github.com/iamJABASTIN/jenkins-docker-demo.git

Credentials: - none -

in credentials

Choose add, fill the dialog



Jenkins Credentials Provider: Jenkins

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

Username: iamJABASTIN

□ Treat username as secret

Password: ••

ID: github-jabastin

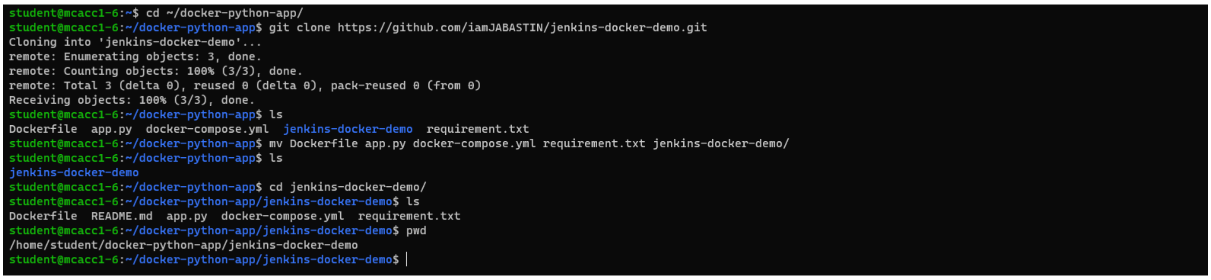
Description:

Dashboard > DeclarativePipeDemo > Configuration
Configure
General
Triggers
Pipeline
Advanced
Save
Apply
localhost:8080/job/DeclarativePipeDemo/configure

Add the details like Username, Password, and ID(any value)

Clone the remote repo to the local folder **/docker-python-app**

Move the files into the cloned repo folder using **mv** command



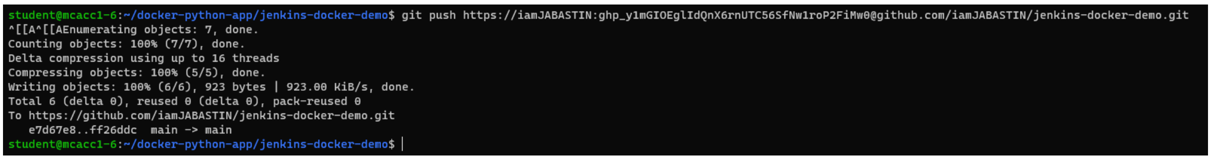
```
student@mcacc1-6:~$ cd ~/docker-python-app/
student@mcacc1-6:~/docker-python-app$ git clone https://github.com/iamJABASTIN/jenkins-docker-demo.git
Cloning into 'jenkins-docker-demo'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
student@mcacc1-6:~/docker-python-app$ ls
Dockerfile  app.py  docker-compose.yml  jenkins-docker-demo  requirement.txt
student@mcacc1-6:~/docker-python-app$ mv Dockerfile app.py docker-compose.yml requirement.txt jenkins-docker-demo/
student@mcacc1-6:~/docker-python-app$ ls
jenkins-docker-demo
student@mcacc1-6:~/docker-python-app$ cd jenkins-docker-demo/
student@mcacc1-6:~/docker-python-app/jenkins-docker-demo$ ls
Dockerfile  README.md  app.py  docker-compose.yml  requirement.txt
student@mcacc1-6:~/docker-python-app/jenkins-docker-demo$ pwd
/home/student/docker-python-app/jenkins-docker-demo
student@mcacc1-6:~/docker-python-app/jenkins-docker-demo$ |
```

ADD AND COMMIT CHANGES

git add .

git commit -m ‘Docker initial commit’

Push the **local Repo** to the **remote Repo**



```
student@mcacc1-6:~/docker-python-app/jenkins-docker-demo$ git push https://iamJABASTIN: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@github.com/iamJABASTIN/jenkins-docker-demo.git
^[[A^[[AEnumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 16 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 923 bytes | 923.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/iamJABASTIN/jenkins-docker-demo.git
   e7d67e8..ff26ddc  main -> main
student@mcacc1-6:~/docker-python-app/jenkins-docker-demo$ |
```

Check the remote repository on GitHub to see if the files are available.

CREATE AN ACCOUNT IN THE DOCKER HUB

Register into the Docker Hub

Add new credentials of Docker Hub in the Jenkins

OPEN MANAGE JENKINS – FOR CREDENTIALS UPDATES

It displays the GitHub and Docker credentials ID.

DASHBOARD > MANAGE JENKINS > CREDENTIALS > SYSTEM > GLOBAL CREDENTIALS

CREATE Jenkinsfile

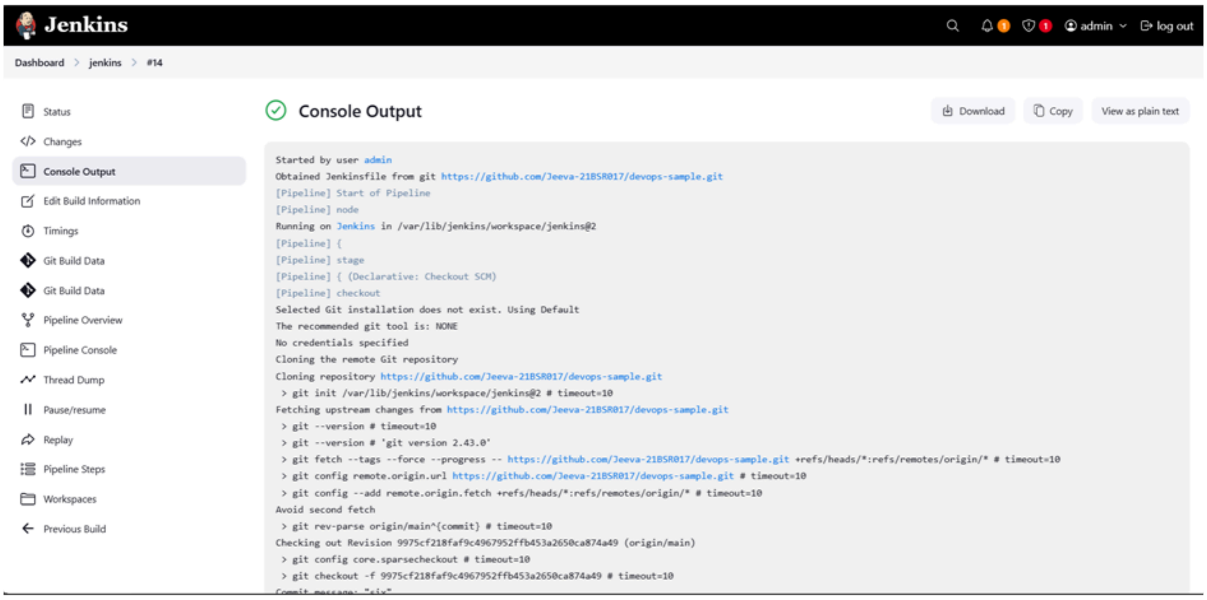
Open Ubuntu and **Create Jenkinsfile** using

nano Jenkins

This is a Jenkins **Pipeline script** written in **Groovy** for automating the CI/CD process of building, pushing, and running a Docker container.

We need to commit the changes in the local repository and then push them to the remote repository.

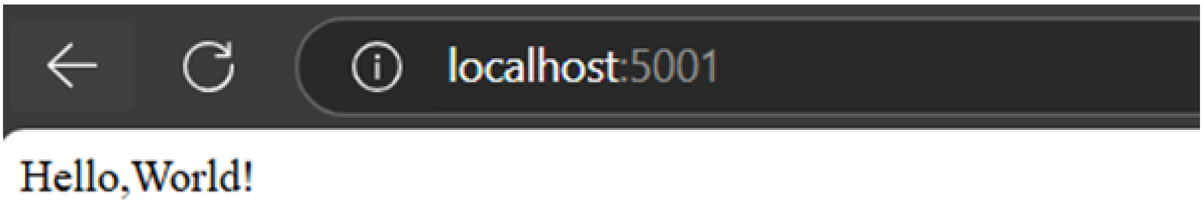
Build the pipeline in Jenkins.



```
Started by user admin
Obtained Jenkinsfile from git https://github.com/Jeeva-21BSR017/devops-sample.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/jenkins@2
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/Jeeva-21BSR017/devops-sample.git
 > git init /var/lib/jenkins/workspace/jenkins@2 # timeout=10
Fetching upstream changes from https://github.com/Jeeva-21BSR017/devops-sample.git
 > git --version # timeout=10
 > git --version # 'git version 2.43.0'
 > git fetch --tags --force --progress -- https://github.com/Jeeva-21BSR017/devops-sample.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git config remote.origin.url https://github.com/Jeeva-21BSR017/devops-sample.git # timeout=10
 > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
 > git rev-parse origin/main^{commit} # timeout=10
Checking out Revision 9975cf218faf9c4967952ffb453a2650ca874a49 (origin/main)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 9975cf218faf9c4967952ffb453a2650ca874a49 # timeout=10
Commit message: "six"
```

Build Successful

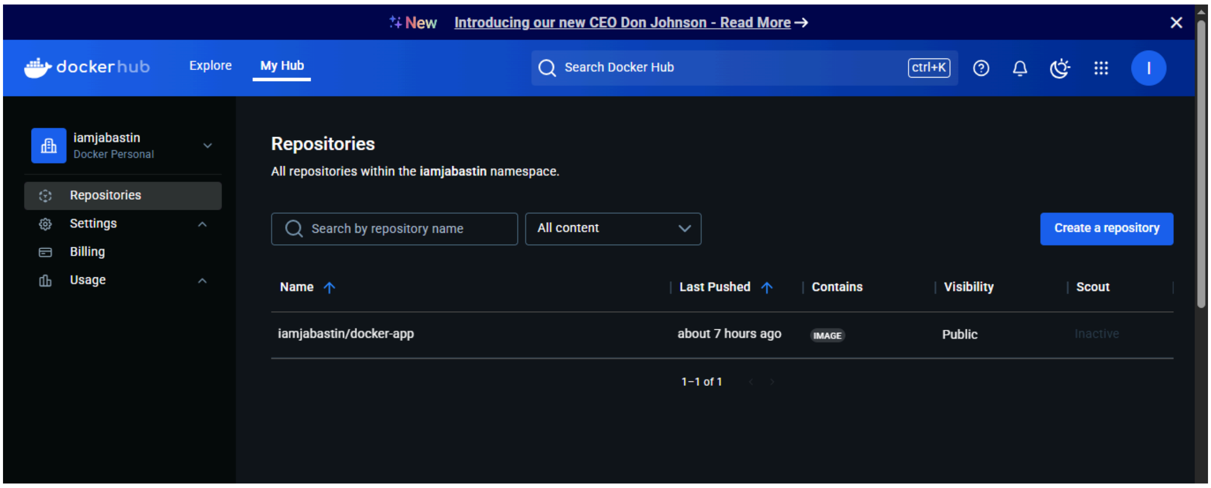
We can view the output at http://localhost:5001



localhost:5001

Hello,World!

In Docker Hub, the processes are completed, and then the Linux image must be displayed.

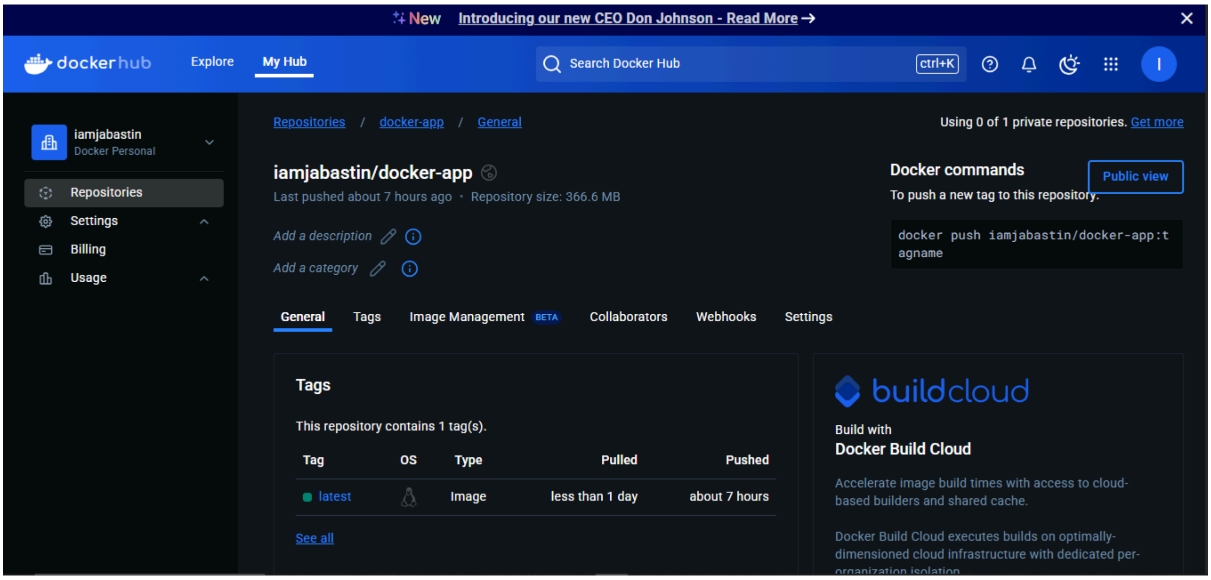


Repositories

All repositories within the **iamjabastin** namespace.

Name	Last Pushed	Contains	Visibility	Scout
iamjabastin/docker-app	about 7 hours ago	IMAGE	Public	Inactive

1–1 of 1



Repositories / docker-app / General

Using 0 of 1 private repositories. Get more

iamjabastin/docker-app

Last pushed about 7 hours ago · Repository size: 366.6 MB

Add a description

Add a category

Docker commands

To push a new tag to this repository:

```
docker push iamjabastin/docker-app:tagname
```

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
● latest		Image	less than 1 day	about 7 hours

See all

buildcloud

Build with

Docker Build Cloud

Accelerate image build times with access to cloud-based builders and shared cache.

Docker Build Cloud executes builds on optimally-dimensioned cloud infrastructure with dedicated per-organization isolation

— COMPLETED —