DevOps Training-Day-2

Installing Docker

Docker is a platform that allows developers to automate the deployment of applications inside lightweight, portable containers. Follow these steps to install Docker on your system:

Step 1: Update System Packages

Run the following command to update your system's package list:

```
sudo apt update
```

Step 2: Install Docker

Install Docker using the following command:

```
sudo apt install -y docker.io
```

Step 3: Enable and Start Docker Service

Enable Docker to start at boot and then start the Docker service:

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

Step 4: Verify Installation

To ensure that Docker is installed successfully, check its version:

```
docker --version
```

Installing Docker Compose

Docker Compose is a tool for defining and running multi-container Docker applications. Follow these steps to install it:

Step 1: Install Curl

Ensure that `curl` is installed by running:

```
sudo apt install curl
```

Step 2: Download Docker Compose

Download the latest version of 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
```

Step 3: Give Execution Permission

Make the downloaded file executable:

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

Step 4: Verify Installation

Check if Docker Compose is installed correctly:

```
docker-compose --version
```

Creating a Python "Hello World" Application

To demonstrate Docker, we will create a simple Python application using Flask.

Step 1: Create a Project Directory

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

Step 2: Create a Python Script

Create a file named `app.py`:

```
nano app.py
```

Step 3: Write Python Code

Add the following code inside `app.py` and save the file:

```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello, World! Running inside Docker!"

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000)
```

Installing Dependencies

To ensure that the necessary dependencies are available inside the container, create a `requirements.txt` file.

Step 1: Create a Dependencies File

```
nano requirements.txt
```

Step 2: Add Required Package

Inside the file, add the following line and save it:

```
flask
```

Creating a Dockerfile

A Dockerfile contains instructions to build a Docker image.

Step 1: Create a Dockerfile

```
nano Dockerfile
```

Step 2: Add Docker Instructions

Paste the following content into the file:

```
# 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 requirements.txt .
RUN pip install --no-cache-dir -r requirements.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"]
```

Creating a Docker Compose File

Docker Compose allows you to define and run multiple containers as a single service.

Step 1: Create a Docker Compose File

```
nano docker-compose.yml
```

Step 2: Add Configuration

Paste the following content into the file:

```
version: '3.8'

services:
  web:
    build: .
    ports:
      - "5000:5000"
    volumes:
      - .:/app
    restart: always
```

Building and Running the Docker Container

Now, we will build and run the application inside a Docker container.

Step 1: Build the Docker Image

```
sudo docker-compose build
```

Step 2: Start the Container

```
sudo docker-compose up -d
```

Verifying the Setup

Step 1: Check Docker Images

To list the available Docker images, run:

```
sudo docker images
```

Step 2: Build and Run Manually (Alternative Method)

```
docker build -t test .
docker run -itd -p 5000:5000 test
```

Step 3: Check Logs

To check if the container is running properly, use:

```
docker logs <container_id>
```

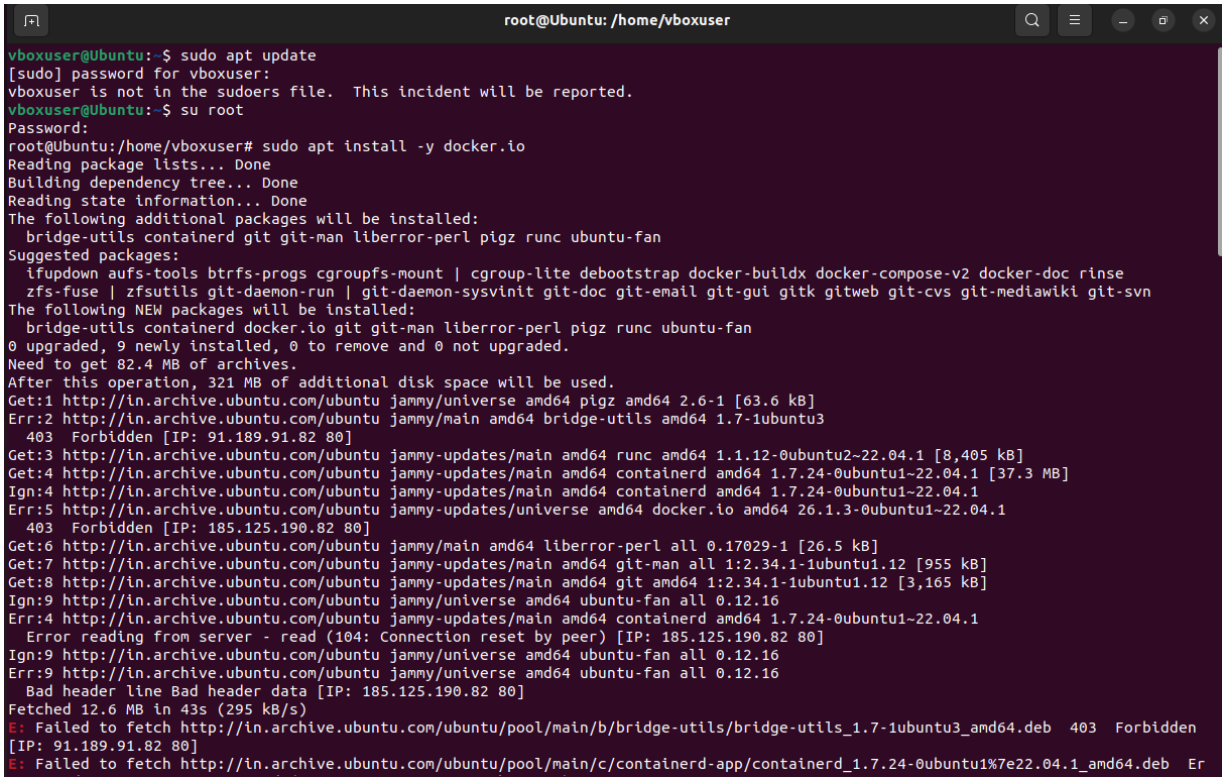
Step 4: Access the Application

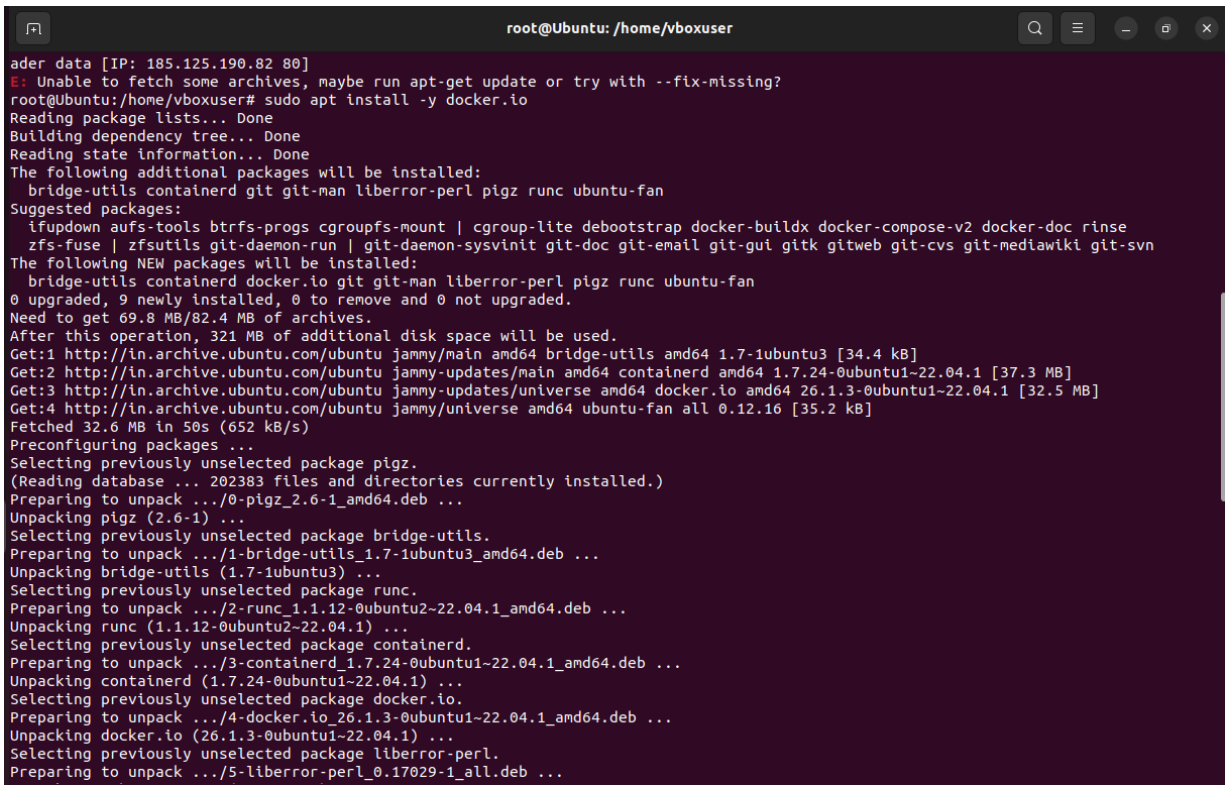
Open a web browser and go to:

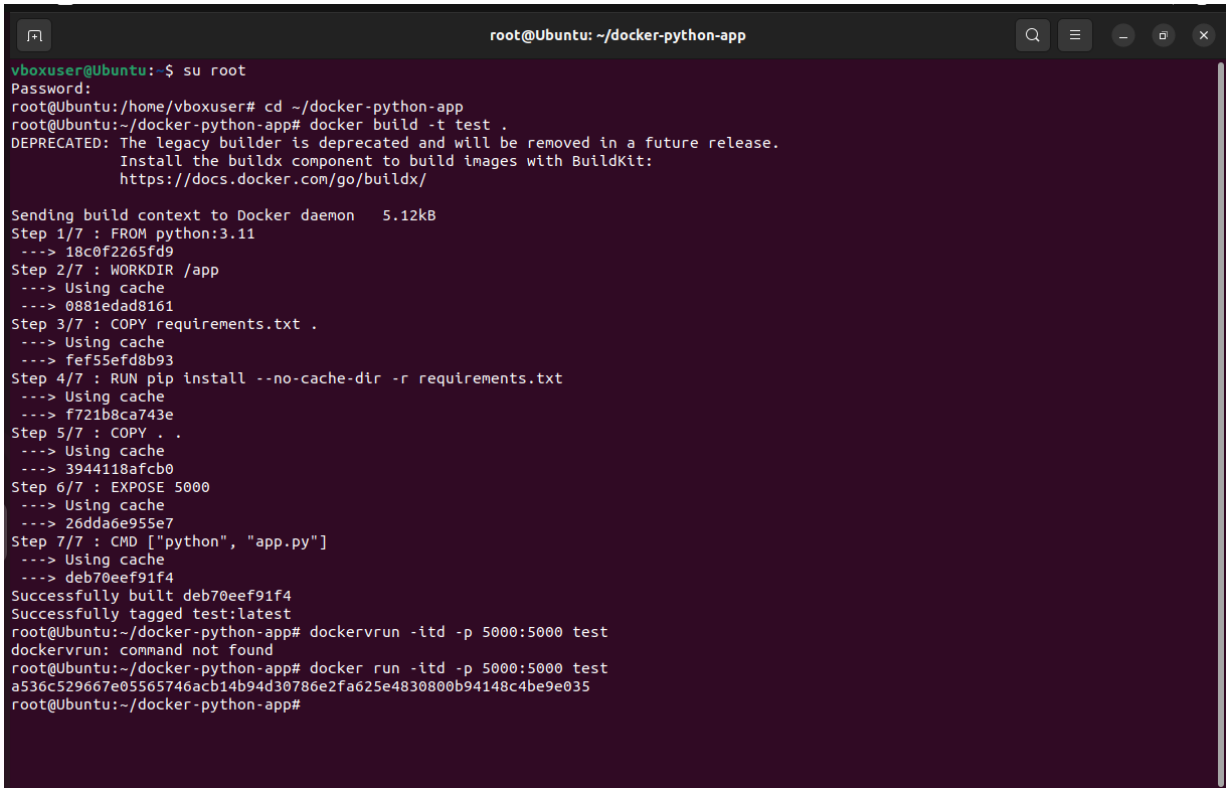
```
http://localhost:5000
```

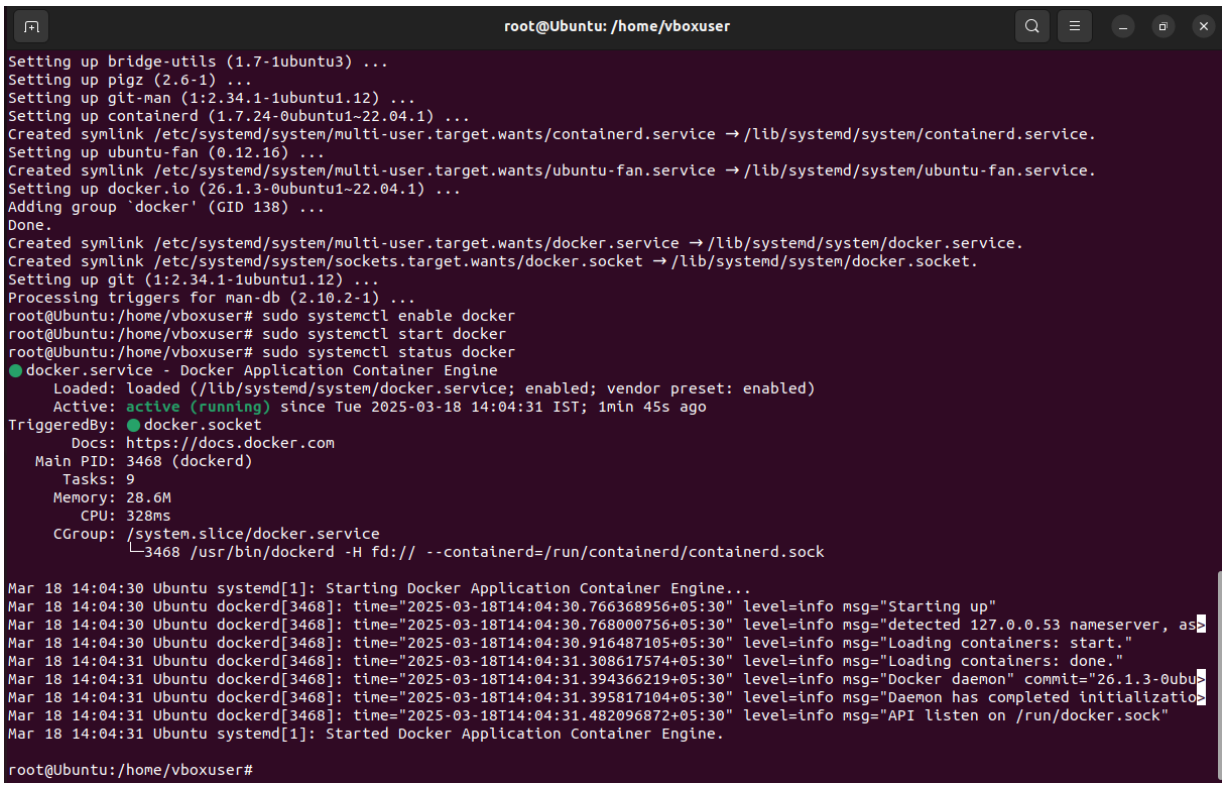
You should see the output:

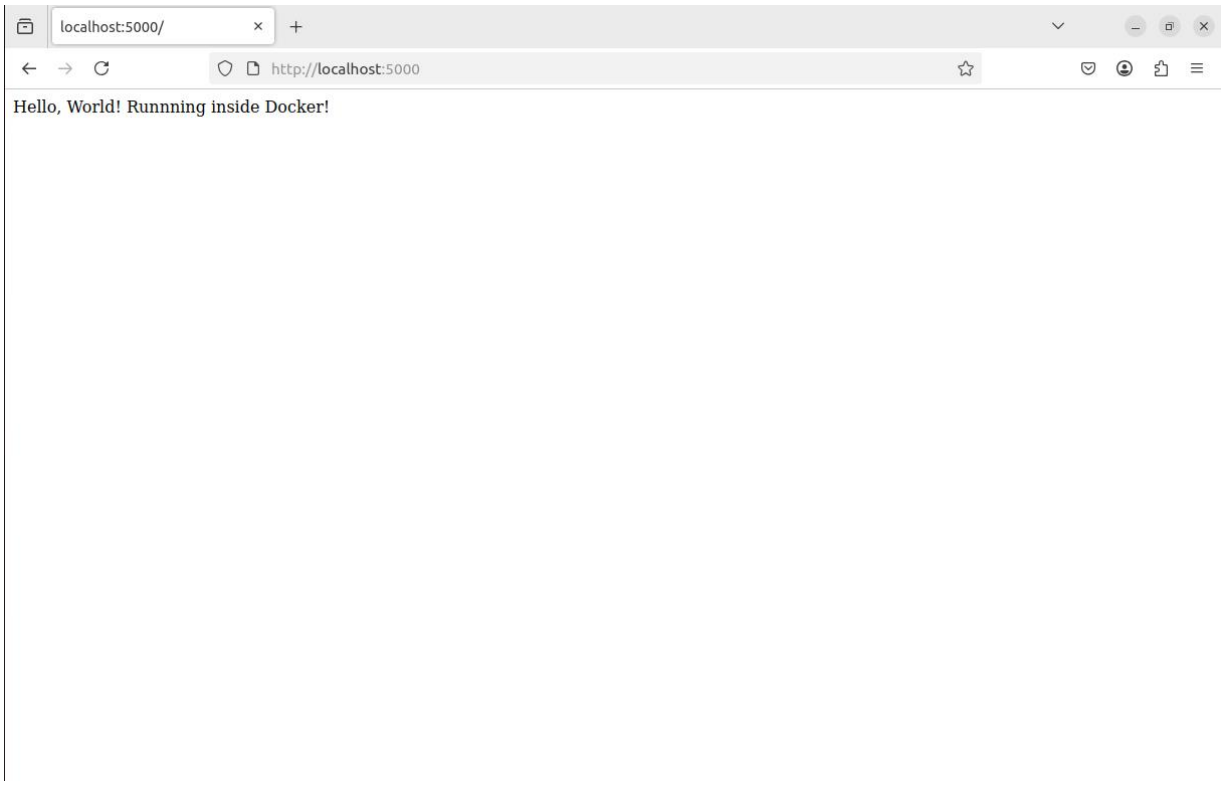
```
Hello, World! Running inside Docker!
```




root@Ubuntu: /home/vboxuser
ader data [IP: 185.125.190.82 80]
E: Unable to fetch some archives, maybe run apt-get update or try with --fix-missing?
root@Ubuntu:/home/vboxuser# 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 git git-man liberror-perl 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 git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
bridge-utils containerd docker.io git git-man liberror-perl pigz runc ubuntu-fan
0 upgraded, 9 newly installed, 0 to remove and 0 not upgraded.
Need to get 69.8 MB/82.4 MB of archives.
After this operation, 321 MB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 bridge-utils amd64 1.7-1ubuntu3 [34.4 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 containerd amd64 1.7.24-0ubuntu1~22.04.1 [37.3 MB]
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 docker.io amd64 26.1.3-0ubuntu1~22.04.1 [32.5 MB]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy/universe amd64 ubuntu-fan all 0.12.16 [35.2 kB]
Fetched 32.6 MB in 50s (652 kB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 202383 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.6-1_amd64.deb ...
Unpacking pigz (2.6-1) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.7-1ubuntu3_amd64.deb ...
Unpacking bridge-utils (1.7-1ubuntu3) ...
Selecting previously unselected package runc.
Preparing to unpack .../2-runc_1.1.12-0ubuntu2~22.04.1_amd64.deb ...
Unpacking runc (1.1.12-0ubuntu2~22.04.1) ...
Selecting previously unselected package containerd.
Preparing to unpack .../3-containerd_1.7.24-0ubuntu1~22.04.1_amd64.deb ...
Unpacking containerd (1.7.24-0ubuntu1~22.04.1) ...
Selecting previously unselected package docker.io.
Preparing to unpack .../4-docker.io_26.1.3-0ubuntu1~22.04.1_amd64.deb ...
Unpacking docker.io (26.1.3-0ubuntu1~22.04.1) ...
Selecting previously unselected package liberror-perl.
Preparing to unpack .../5-liberror-perl_0.17029-1_all.deb ...


root@Ubuntu: ~/docker-python-app
vboxuser@Ubuntu:~$ su root
Password:
root@Ubuntu:/home/vboxuser# cd ~/docker-python-app
root@Ubuntu:~/docker-python-app# docker build -t test .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx/
Sending build context to Docker daemon 5.12kB
Step 1/7 : FROM python:3.11
---> 18c0f2265fd9
Step 2/7 : WORKDIR /app
---> Using cache
---> 0881edad8161
Step 3/7 : COPY requirements.txt .
---> Using cache
---> fef55efd8b93
Step 4/7 : RUN pip install --no-cache-dir -r requirements.txt
---> Using cache
---> f721b8ca743e
Step 5/7 : COPY . .
---> Using cache
---> 3944118afcb0
Step 6/7 : EXPOSE 5000
---> Using cache
---> 26dda6e955e7
Step 7/7 : CMD ["python", "app.py"]
---> Using cache
---> deb70eef91f4
Successfully built deb70eef91f4
Successfully tagged test:latest
root@Ubuntu:~/docker-python-app# dockervrun -itd -p 5000:5000 test
dockervrun: command not found
root@Ubuntu:~/docker-python-app# docker run -itd -p 5000:5000 test
a536c529667e05565746acb14b94d30786e2fa625e4830800b94148c4be9e035
root@Ubuntu:~/docker-python-app#


root@Ubuntu: /home/vboxuser
Setting up bridge-utils (1.7-1ubuntu3) ...
Setting up pigz (2.6-1) ...
Setting up git-man (1:2.34.1-1ubuntu1.12) ...
Setting up containerd (1.7.24-0ubuntu1~22.04.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
Setting up ubuntu-fan (0.12.16) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /lib/systemd/system/ubuntu-fan.service.
Setting up docker.io (26.1.3-0ubuntu1~22.04.1) ...
Adding group `docker' (GID 138) ...
Done.
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Setting up git (1:2.34.1-1ubuntu1.12) ...
Processing triggers for man-db (2.10.2-1) ...
root@Ubuntu:/home/vboxuser# sudo systemctl enable docker
root@Ubuntu:/home/vboxuser# sudo systemctl start docker
root@Ubuntu:/home/vboxuser# sudo systemctl status docker
docker.service - Docker Application Container Engine
Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
Active: active (running) since Tue 2025-03-18 14:04:31 IST; 1min 45s ago
TriggeredBy: docker.socket
Docs: https://docs.docker.com
Main PID: 3468 (dockerd)
Tasks: 9
Memory: 28.6M
CPU: 328ms
CGroup: /system.slice/docker.service
└─3468 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
Mar 18 14:04:30 Ubuntu systemd[1]: Starting Docker Application Container Engine...
Mar 18 14:04:30 Ubuntu dockerd[3468]: time="2025-03-18T14:04:30.766368956+05:30" level=info msg="Starting up"
Mar 18 14:04:30 Ubuntu dockerd[3468]: time="2025-03-18T14:04:30.768000756+05:30" level=info msg="detected 127.0.0.53 nameserver, as
Mar 18 14:04:30 Ubuntu dockerd[3468]: time="2025-03-18T14:04:30.916487105+05:30" level=info msg="Loading containers: start."
Mar 18 14:04:31 Ubuntu dockerd[3468]: time="2025-03-18T14:04:31.308617574+05:30" level=info msg="Loading containers: done."
Mar 18 14:04:31 Ubuntu dockerd[3468]: time="2025-03-18T14:04:31.394366219+05:30" level=info msg="Docker daemon" commit="26.1.3-0ubu
Mar 18 14:04:31 Ubuntu dockerd[3468]: time="2025-03-18T14:04:31.395817104+05:30" level=info msg="Daemon has completed initializatio
Mar 18 14:04:31 Ubuntu dockerd[3468]: time="2025-03-18T14:04:31.482096872+05:30" level=info msg="API listen on /run/docker.sock"
Mar 18 14:04:31 Ubuntu systemd[1]: Started Docker Application Container Engine.
root@Ubuntu:/home/vboxuser#


localhost:5000/
http://localhost:5000
Hello, World! Runnning inside Docker!

Pushing the Project to GitHub

Step 1: Clone the Repository

```
git clone https://github.com/SujithaKC/jenkins-docker-demo.git
cd jenkins-docker-demo
```

Step 2: Move Files into Repository

```
mv ~/docker-python-app/Dockerfile ~/docker-python-app/requirements.txt ~/docker-python-app/app.py ~/docker-python-app/docker-compose.yml .
```

Step 3: Add and Commit the Changes

```
git add --all
git commit -m "Initial commit for docker app"
```

Step 4: Push to GitHub

```
git push origin main
```

Configuring Jenkins Pipeline

Step 1: Create a `Jenkinsfile`

```
nano Jenkinsfile
```

Step 2: Add Jenkins Pipeline Code

Paste the following content into the file:

```
pipeline {
    agent any
    environment {
        DOCKER_IMAGE = "mohana0304/docker-app:latest"  // Change this to your registry
        CONTAINER_NAME = "docker-running-app"
        REGISTRY_CREDENTIALS = "docker-hub-credentials"  // Jenkins credentials ID
    }

    stages {
        stage('Checkout Code') {
            steps {
                withCredentials([usernamePassword(credentialsId: 'github-mona', usernameVariable: 'GIT_USER', passwordVariable: 'GIT_TOKEN')]) {
                    git url: "https://$GIT_USER:$GIT_TOKEN@github.com/mohana0304/jenkins-docker-demo.git", branch: 'main'
                }
            }
        }

        stage('Build Docker Image') {
            steps {
                sh 'docker build -t $DOCKER_IMAGE .'
            }
        }

        stage('Login to Docker Registry') {
            steps {
```

```
                withCredentials([usernamePassword(credentialsId: 'docker-mona',
usernameVariable: 'DOCKER_USER', passwordVariable: 'DOCKER_PASS')]) {
                    sh 'echo $DOCKER_PASS | docker login -u $DOCKER_USER --pass-
word-stdin'
                }
            }
        }

        stage('Push to Container Registry') {
            steps {
                sh 'docker push $DOCKER_IMAGE'
            }
        }

        stage('Stop & Remove Existing Container') {
            steps {
                script {
                    sh '''
                    if [ "$(docker ps -aq -f name=$CONTAINER_NAME)" ]; then
                        docker stop $CONTAINER_NAME || true
                        docker rm $CONTAINER_NAME || true
                    fi
                    '''
                }
            }
        }

        stage('Run Docker Container') {
            steps {
                sh 'docker run -d -p 5001:5000 --name $CONTAINER_NAME
$DOCKER_IMAGE'
            }
        }
    }

    post {
        success {
            echo "Build, push, and container execution successful!"
        }
        failure {
            echo "Build or container execution failed."
        }
    }
}
```

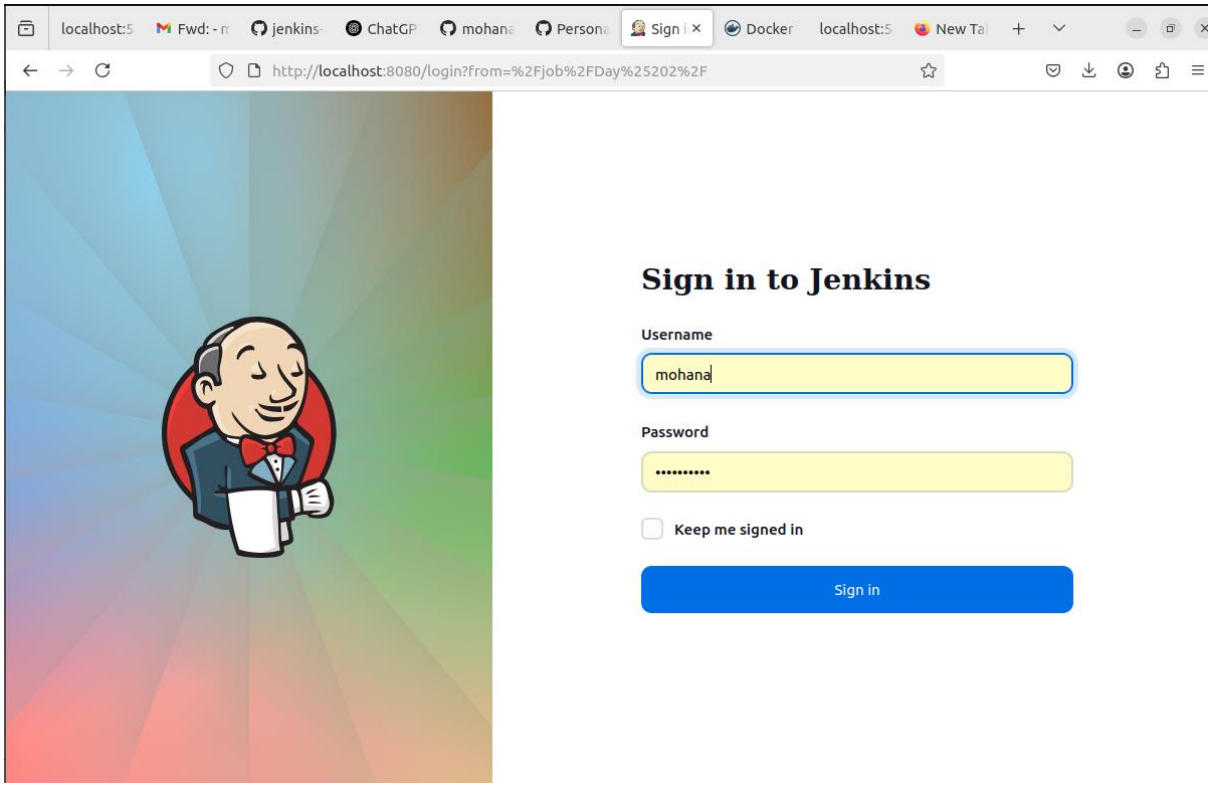
Running Jenkins Build

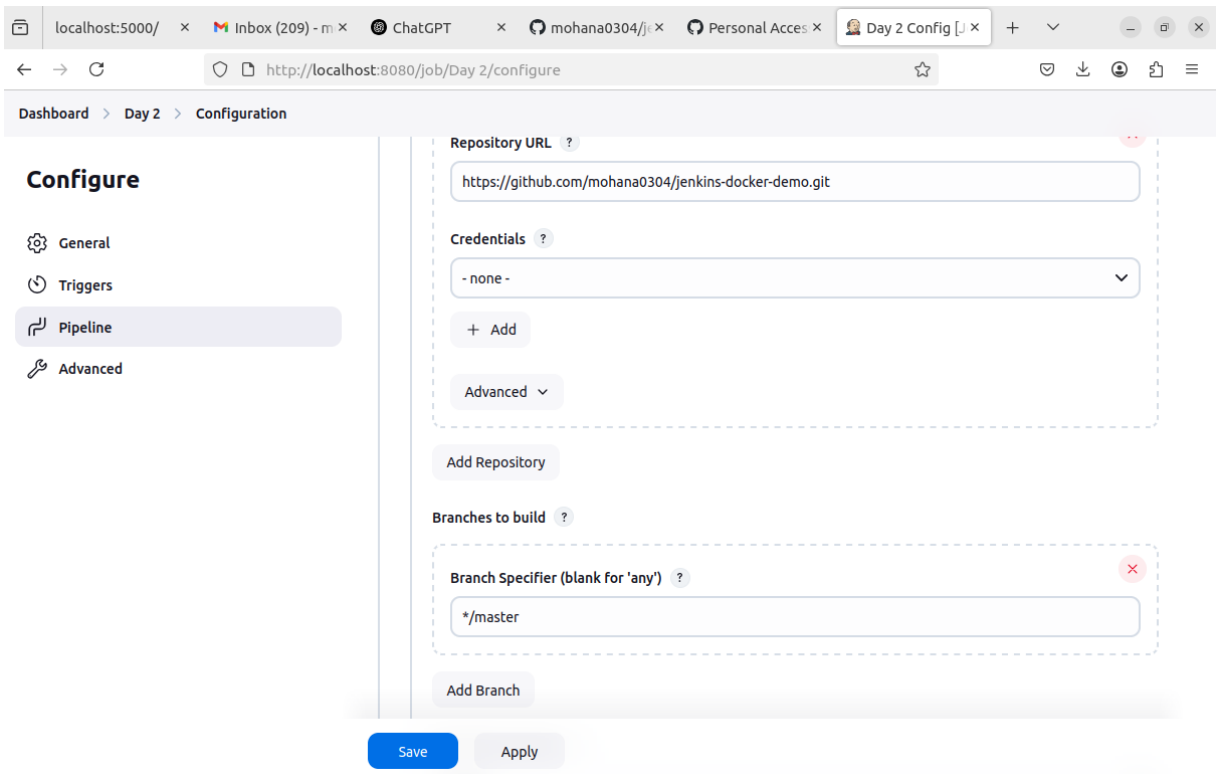
Step 1: Resolve Security Error

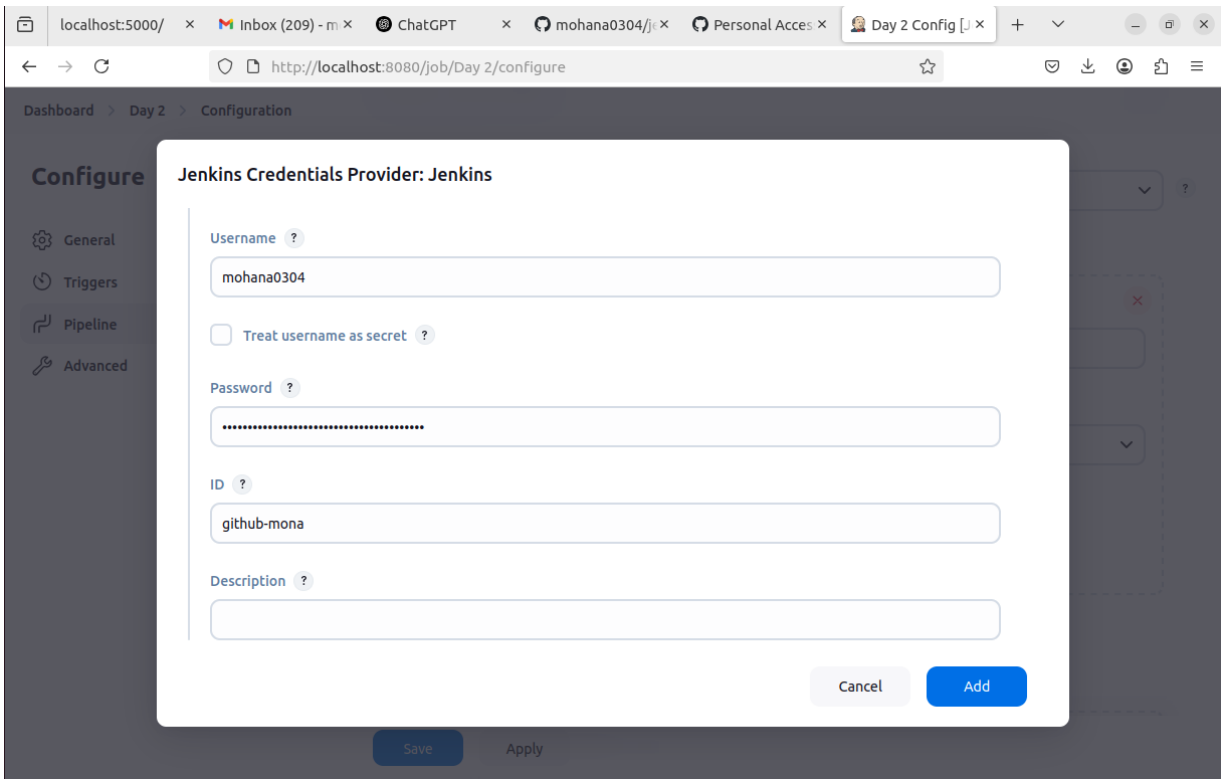
```
sudo usermod -aG docker jenkins
sudo systemctl restart jenkins
```

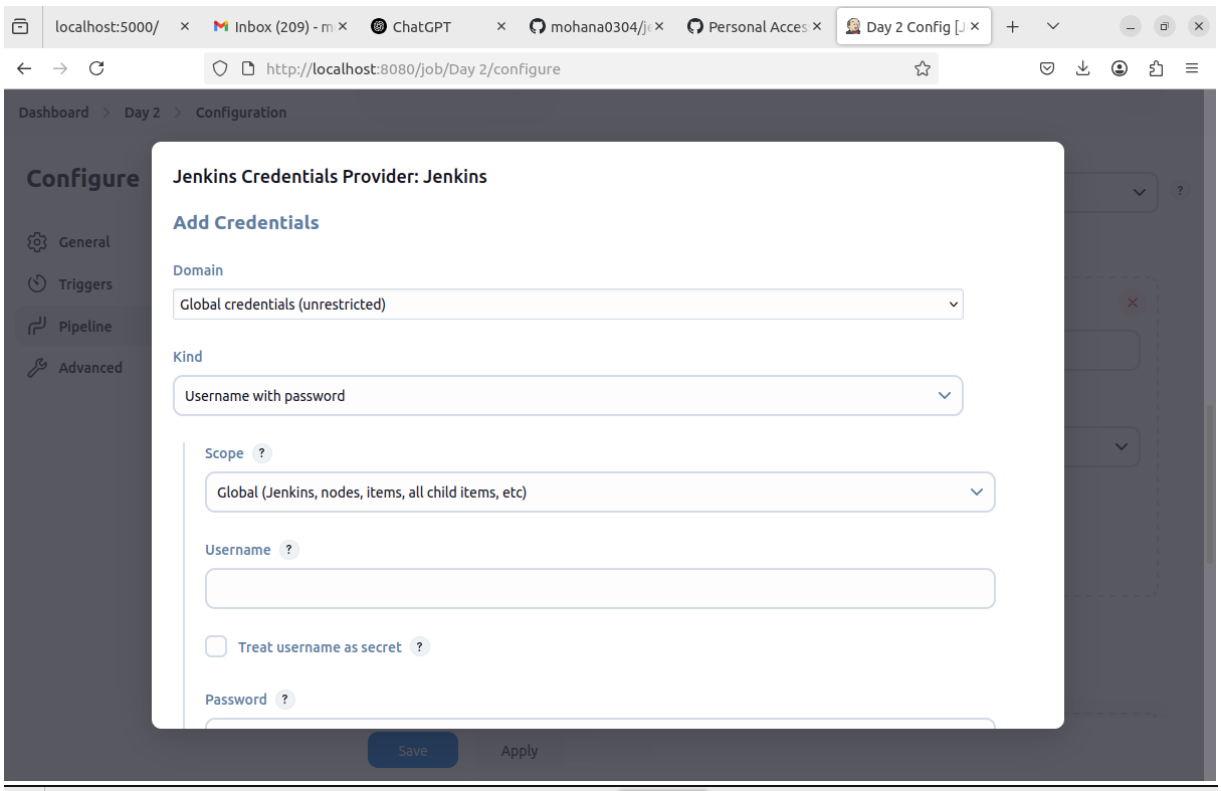
Step 2: Verify Jenkins Credentials

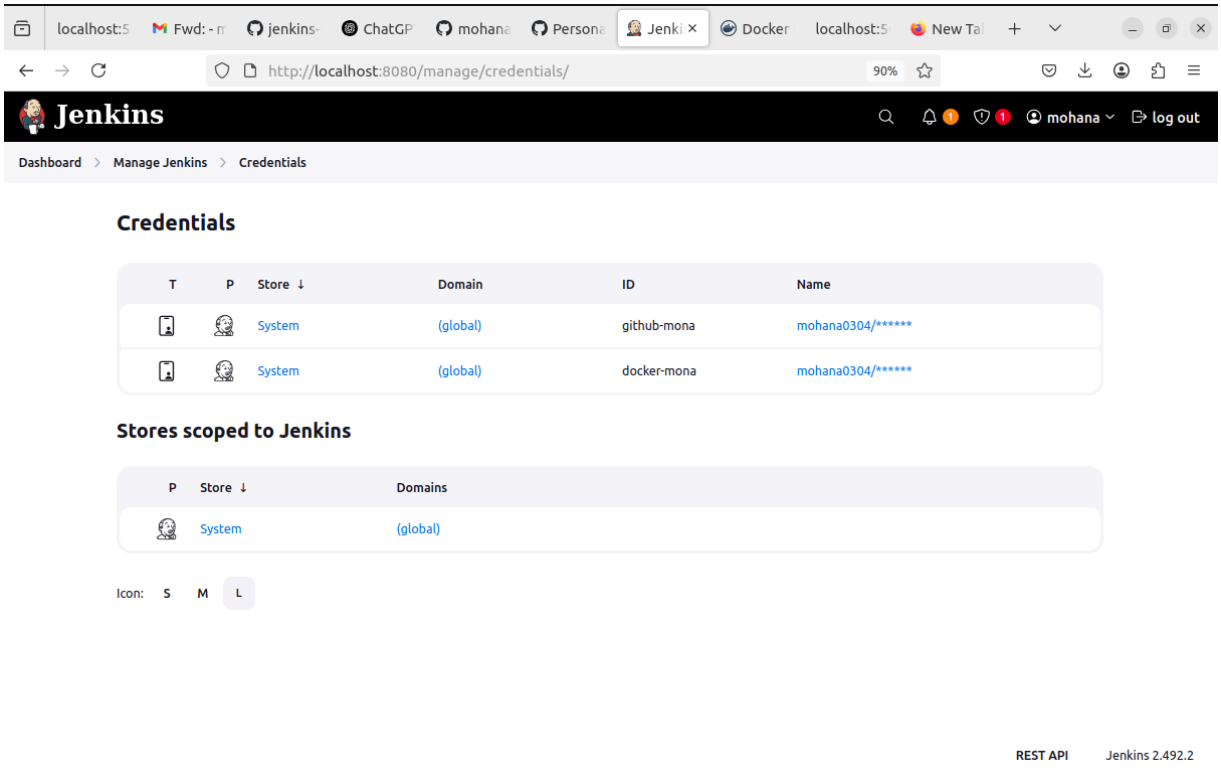
Ensure that the correct credentials are set in Jenkins before triggering the build.

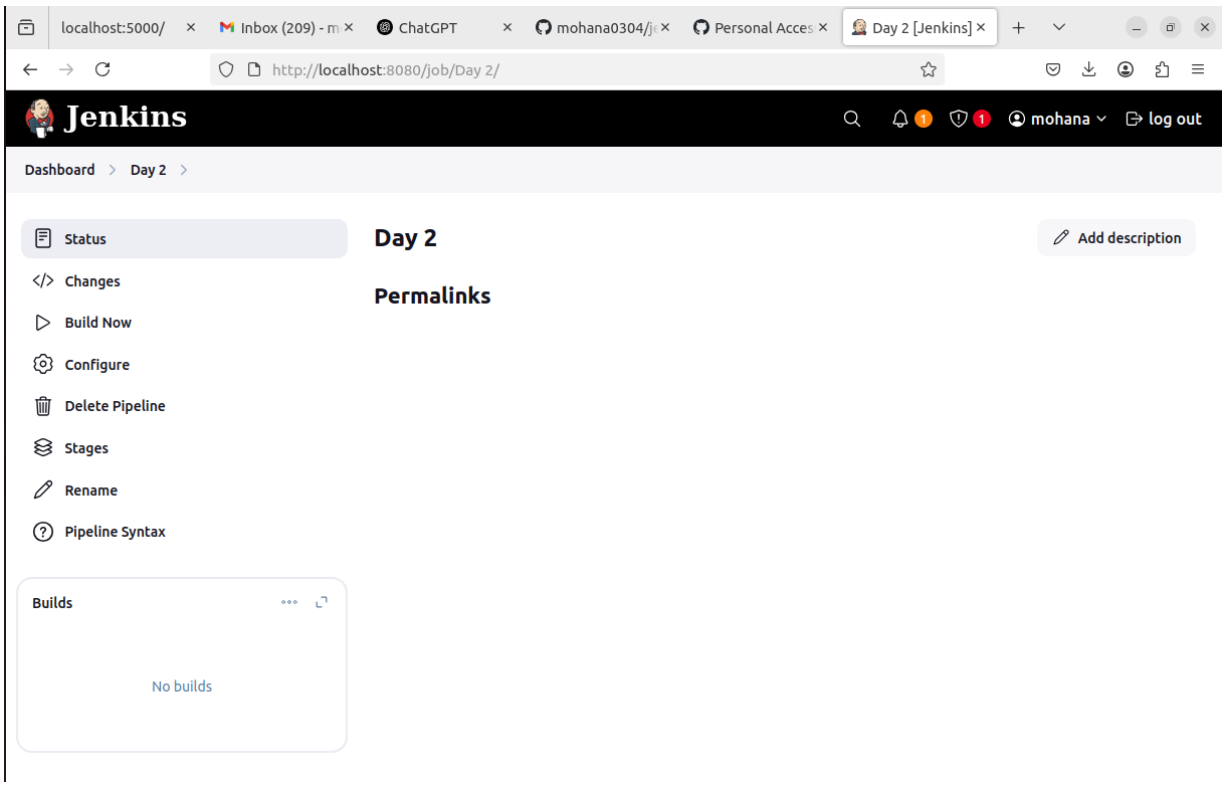

Sign in to Jenkins
Username
mohana
Password
Keep me signed in
Sign in

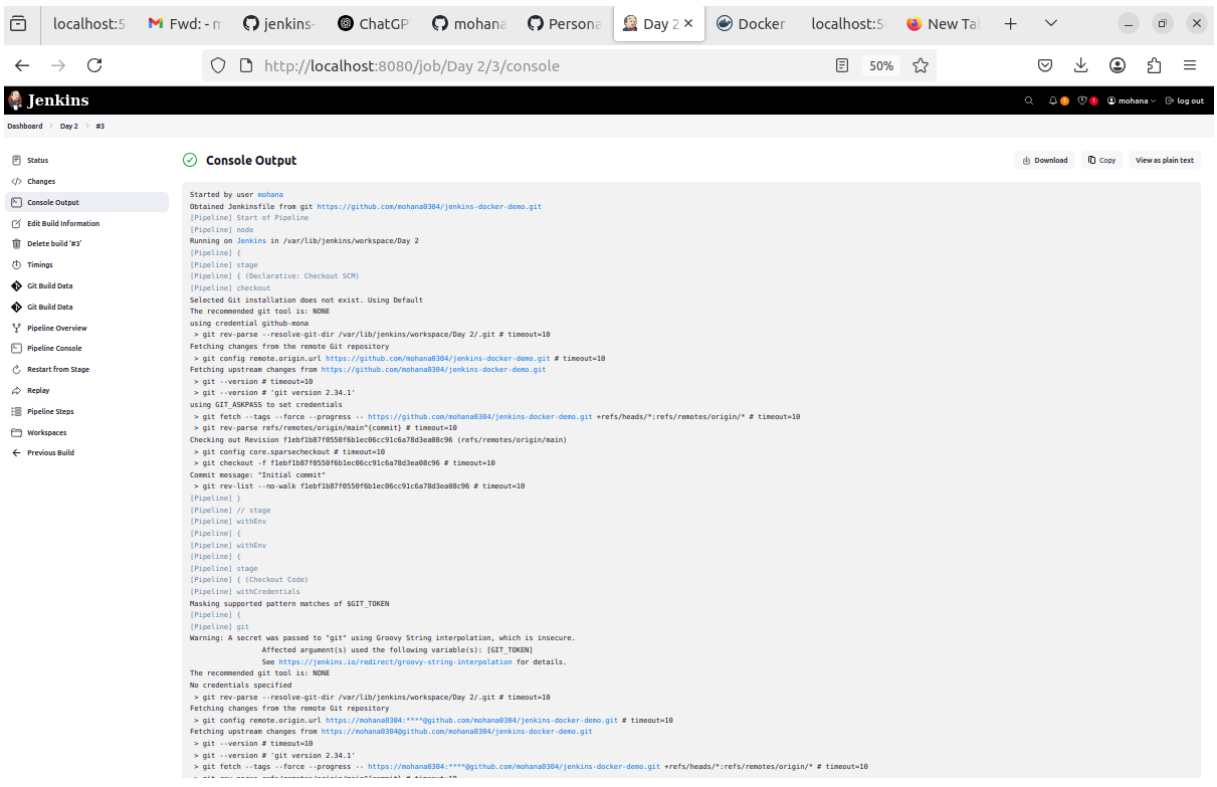

localhost:5000/
Inbox (209) - m
ChatGPT
mohana0304/j
Personal Acces
Day 2 Config [J
http://localhost:8080/job/Day 2/configure
Dashboard
Day 2
Configuration
Configure
General
Triggers
Pipeline
Advanced
Repository URL
https://github.com/mohana0304/jenkins-docker-demo.git
Credentials
- none -
Add
Advanced
Add Repository
Branches to build
Branch Specifier (blank for 'any')
*/master
Add Branch
Save
Apply

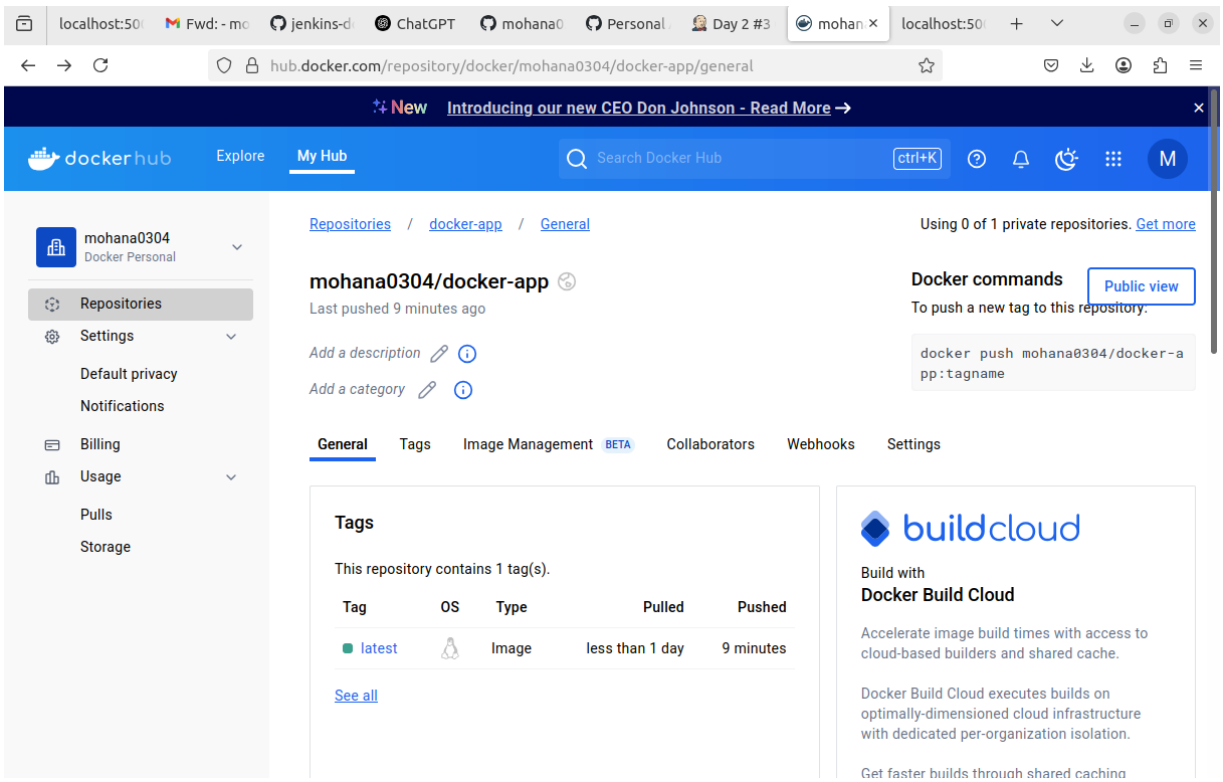

localhost:5000/
Inbox (209) - m
ChatGPT
mohana0304/j
Personal Acces
Day 2 Config [J
http://localhost:8080/job/Day 2/configure
Dashboard
Day 2
Configuration
Configure
General
Triggers
Pipeline
Advanced
Jenkins Credentials Provider: Jenkins
Username
mohana0304
Treat username as secret
Password
ID
github-mona
Description
Cancel
Add
Save
Apply

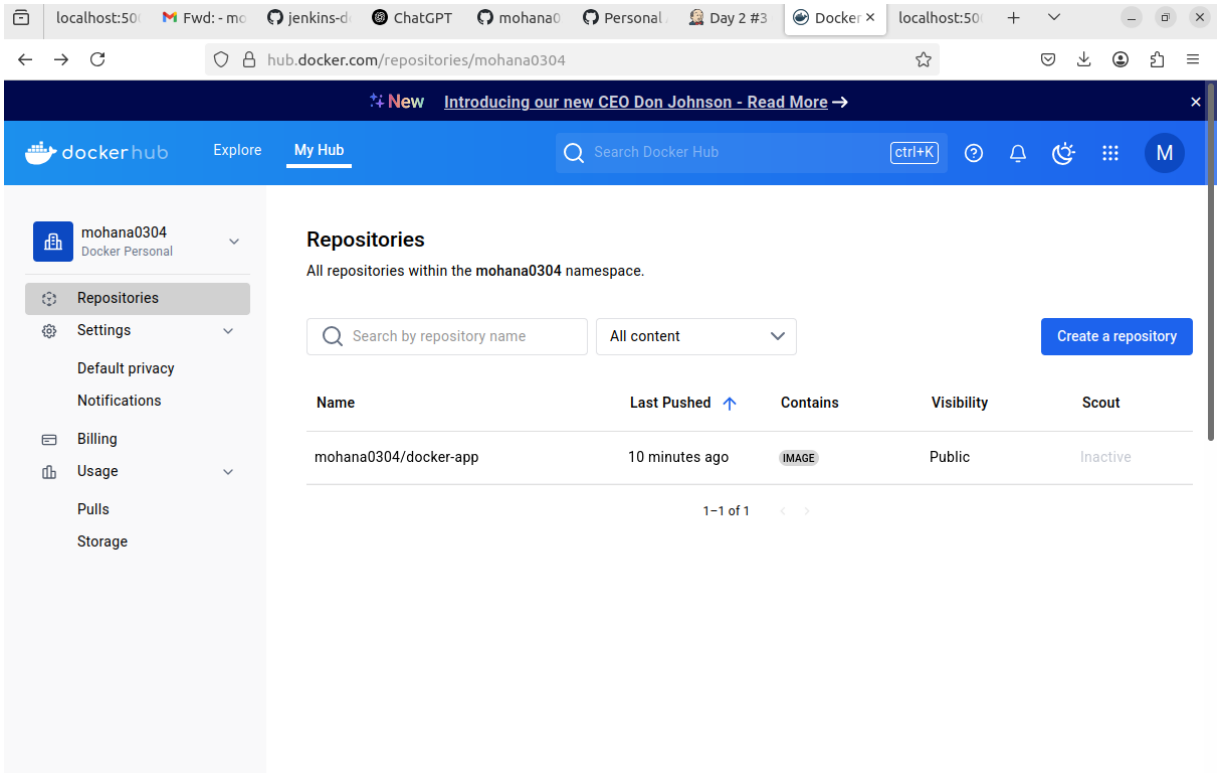

localhost:5000/
Inbox (209) - m
ChatGPT
mohana0304/j
Personal Acces
Day 2 Config [J
http://localhost:8080/job/Day 2/configure
Dashboard
Day 2
Configuration
Configure
General
Triggers
Pipeline
Advanced
Jenkins Credentials Provider: Jenkins
Add Credentials
Domain
Global credentials (unrestricted)
Kind
Username with password
Scope
Global (Jenkins, nodes, items, all child items, etc)
Username
Treat username as secret
Password
Save
Apply

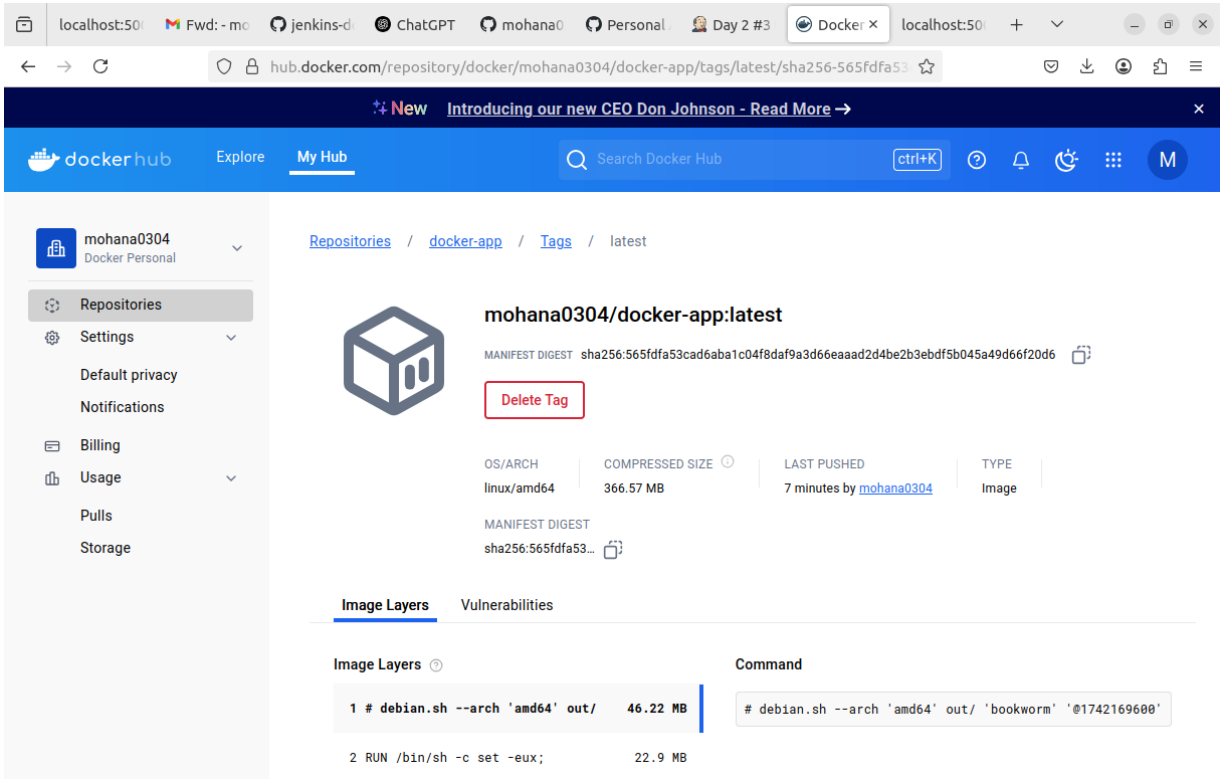

localhost:5
Fwd: - n
jenkins-
ChatGP
mohana
Persona
Jenki
Docker
localhost:5
New Tal
http://localhost:8080/manage/credentials/
90%
Jenkins
mohana
log out
Dashboard
Manage Jenkins
Credentials
Credentials
T
P
Store ↓
Domain
ID
Name
System
(global)
github-mona
mohana0304/******
System
(global)
docker-mona
mohana0304/******
Stores scoped to Jenkins
P
Store ↓
Domains
System
(global)
Icon:
S
M
L
REST API
Jenkins 2.492.2


localhost:5000/
Inbox (209) - m
ChatGPT
mohana0304/j
Personal Acces
Day 2 [Jenkins]
http://localhost:8080/job/Day 2/
Jenkins
mohana
log out
Dashboard
Day 2
Status
Changes
Build Now
Configure
Delete Pipeline
Stages
Rename
Pipeline Syntax
Builds
No builds
Day 2
Permalinks
Add description


localhost:5
Fwd: - m
jenkins-
ChatGP
mohana
Persona
Day 2
Docker
localhost:5
New Tal
http://localhost:8080/job/Day 2/3/console
50%
Jenkins
Dashboard
Day 2
#3
Status
Changes
Console Output
Edit Build Information
Delete build '#3'
Timings
Git Build Data
Git Build Data
Pipeline Overview
Pipeline Console
Restart from Stage
Replay
Pipeline Steps
Workspaces
Previous Build
Console Output
Download
Copy
View as plain text
Started by user mohana
Obtained Jenkinsfile from git https://github.com/mohana0304/jenkins-docker-demo.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/Day 2
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential github-mona
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Day 2/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/mohana0304/jenkins-docker-demo.git # timeout=10
Fetching upstream changes from https://github.com/mohana0304/jenkins-docker-demo.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- https://github.com/mohana0304/jenkins-docker-demo.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision f1ebf1b87f0550f6b1ec06cc91c6a78d3ea08c96 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f f1ebf1b87f0550f6b1ec06cc91c6a78d3ea08c96 # timeout=10
Commit message: "Initial commit"
> git rev-list --no-walk f1ebf1b87f0550f6b1ec06cc91c6a78d3ea08c96 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Checkout Code)
[Pipeline] withCredentials
Masking supported pattern matches of $GIT_TOKEN
[Pipeline] {
[Pipeline] git
Warning: A secret was passed to "git" using Groovy String interpolation, which is insecure.
Affected argument(s) used the following variable(s): [GIT_TOKEN]
See https://jenkins.io/redirect/groovy-string-interpolation for details.
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Day 2/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://mohana0304:****@github.com/mohana0304/jenkins-docker-demo.git # timeout=10
Fetching upstream changes from https://mohana0304@github.com/mohana0304/jenkins-docker-demo.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
> git fetch --tags --force --progress -- https://mohana0304:****@github.com/mohana0304/jenkins-docker-demo.git +refs/heads/*:refs/remotes/origin/* # timeout=10


hub.docker.com/repository/docker/mohana0304/docker-app/general
New Introducing our new CEO Don Johnson - Read More
docker hub
Explore
My Hub
Search Docker Hub
ctrl+K
mohana0304
Docker Personal
Repositories
Settings
Default privacy
Notifications
Billing
Usage
Pulls
Storage
Repositories / docker-app / General
Using 0 of 1 private repositories. Get more
mohana0304/docker-app
Last pushed 9 minutes ago
Add a description
Add a category
Docker commands
Public view
To push a new tag to this repository:
docker push mohana0304/docker-app:tagname
General
Tags
Image Management BETA
Collaborators
Webhooks
Settings
Tags
This repository contains 1 tag(s).
Tag
OS
Type
Pulled
Pushed
latest
Image
less than 1 day
9 minutes
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.
Get faster builds through shared caching


hub.docker.com/repositories/mohana0304
New Introducing our new CEO Don Johnson - Read More
docker hub
Explore
My Hub
Search Docker Hub
ctrl+K
mohana0304
Docker Personal
Repositories
Settings
Default privacy
Notifications
Billing
Usage
Pulls
Storage
Repositories
All repositories within the mohana0304 namespace.
Search by repository name
All content
Create a repository
Name
Last Pushed
Contains
Visibility
Scout
mohana0304/docker-app
10 minutes ago
IMAGE
Public
Inactive
1–1 of 1



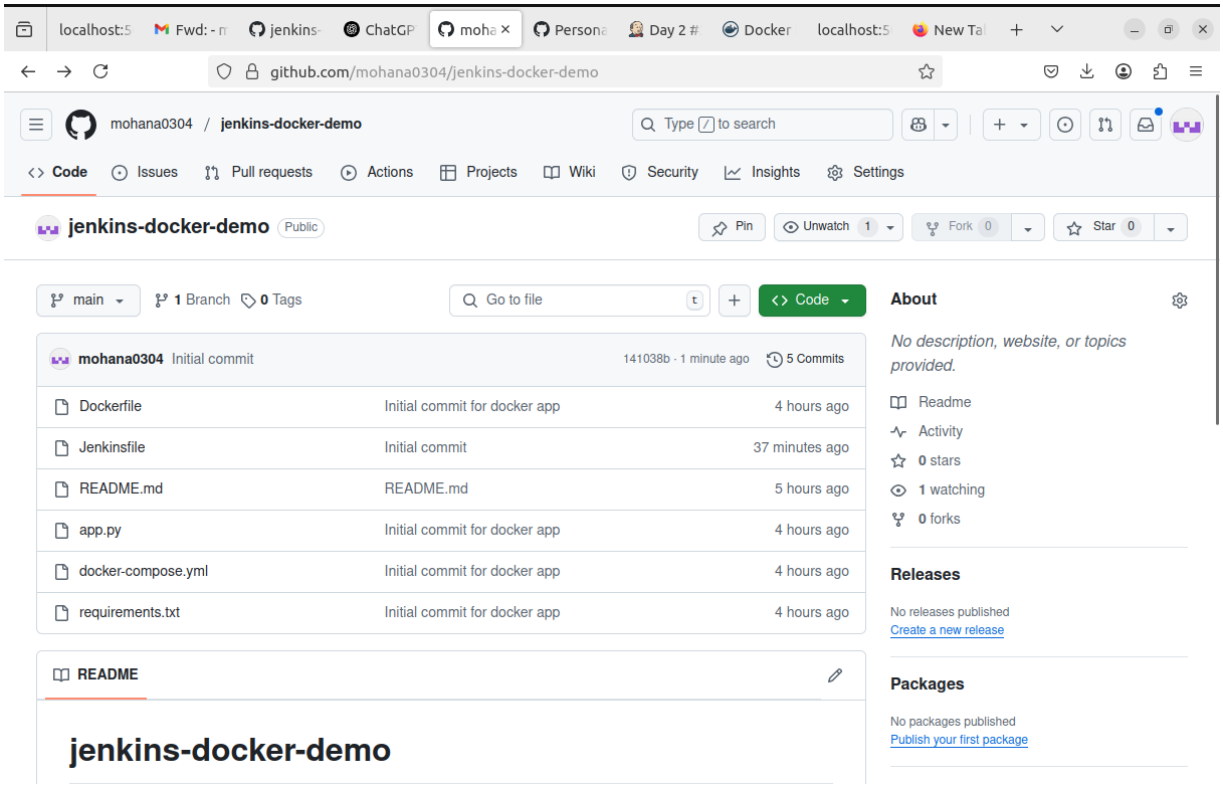
Step 3: Run the Build

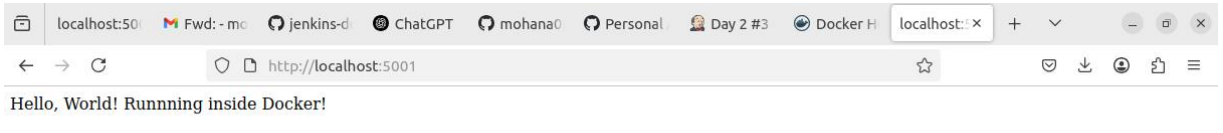
Trigger the Jenkins build. If successful, the Docker image will be updated and the application will be running on port `5001`.

Step 4: Fix Naming Issues

If Jenkins cannot find the `Jenkinsfile`, rename it using:

```
mv jenkinsfile Jenkinsfile
git add .
git commit -m "Fixed Jenkinsfile naming issue"
git push origin main
```


github.com/mohana0304/jenkins-docker-demo
mohana0304 / jenkins-docker-demo
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
jenkins-docker-demo
Public
Pin
Unwatch 1
Fork 0
Star 0
main
1 Branch
0 Tags
Go to file
Code
About
No description, website, or topics provided.
Readme
Activity
0 stars
1 watching
0 forks
mohana0304 Initial commit
141038b · 1 minute ago
5 Commits
Dockerfile
Initial commit for docker app
4 hours ago
Jenkinsfile
Initial commit
37 minutes ago
README.md
README.md
5 hours ago
app.py
Initial commit for docker app
4 hours ago
docker-compose.yml
Initial commit for docker app
4 hours ago
requirements.txt
Initial commit for docker app
4 hours ago
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
README
jenkins-docker-demo


http://localhost:5001
Hello, World! Runnning inside Docker!