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/dockercompose-
$(uname -s)-$(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>
```

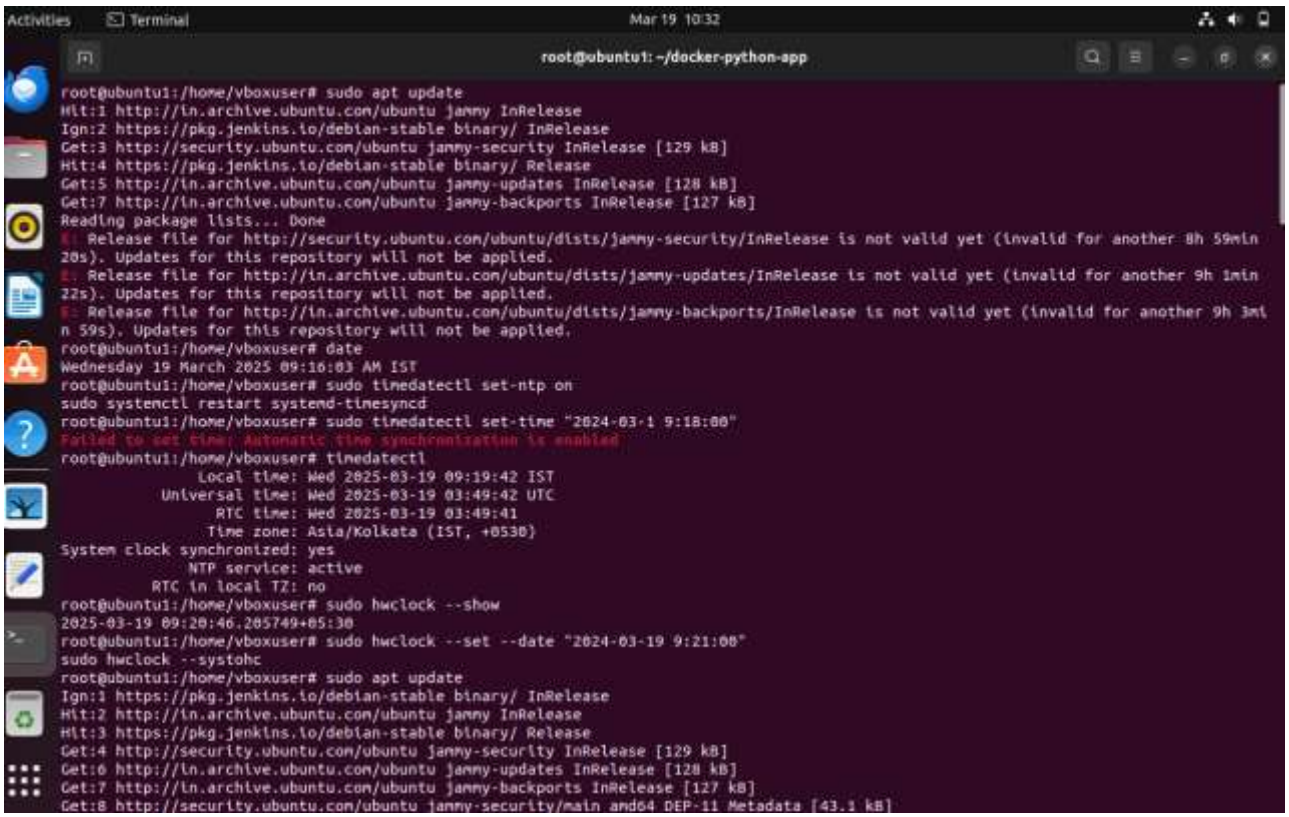
Step 4: Access the Application

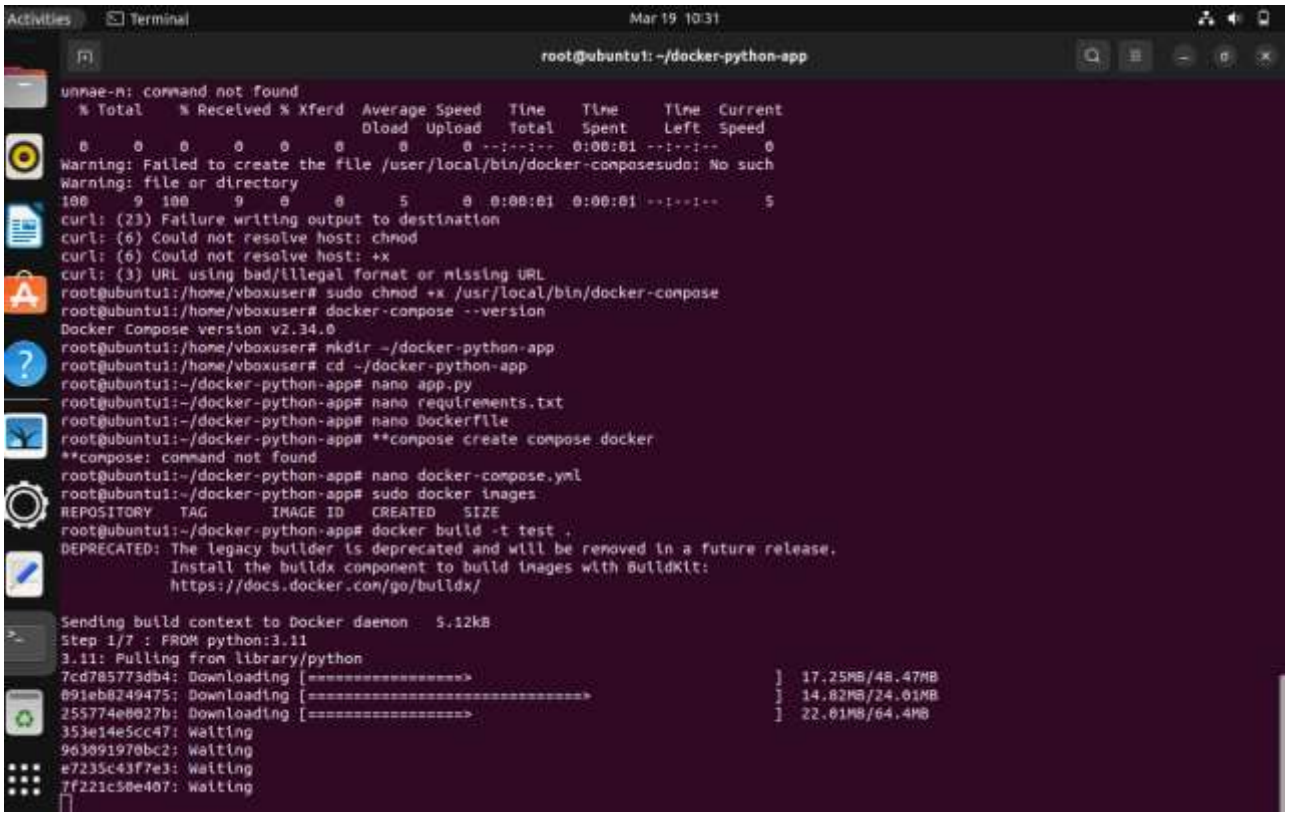
Open a web browser and go to:

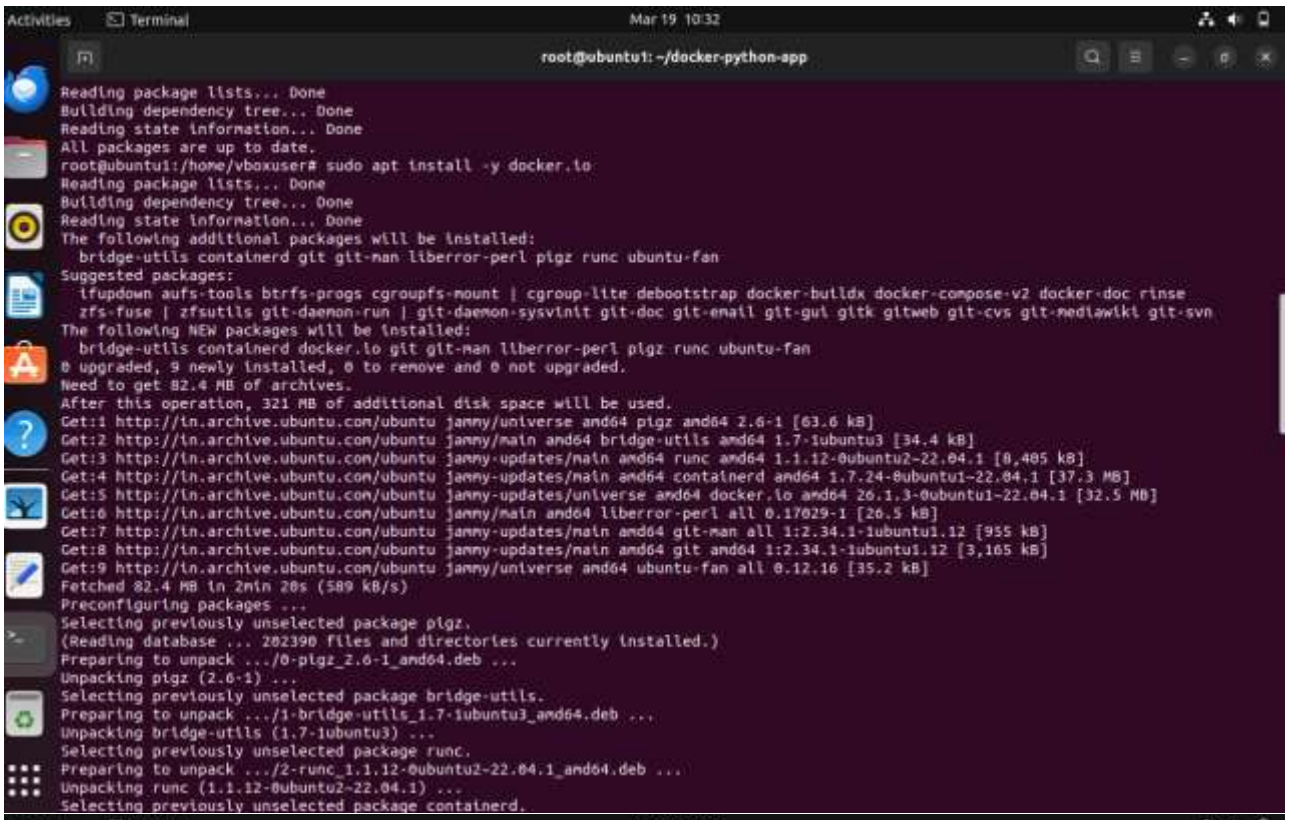
`http://localhost:5000` You

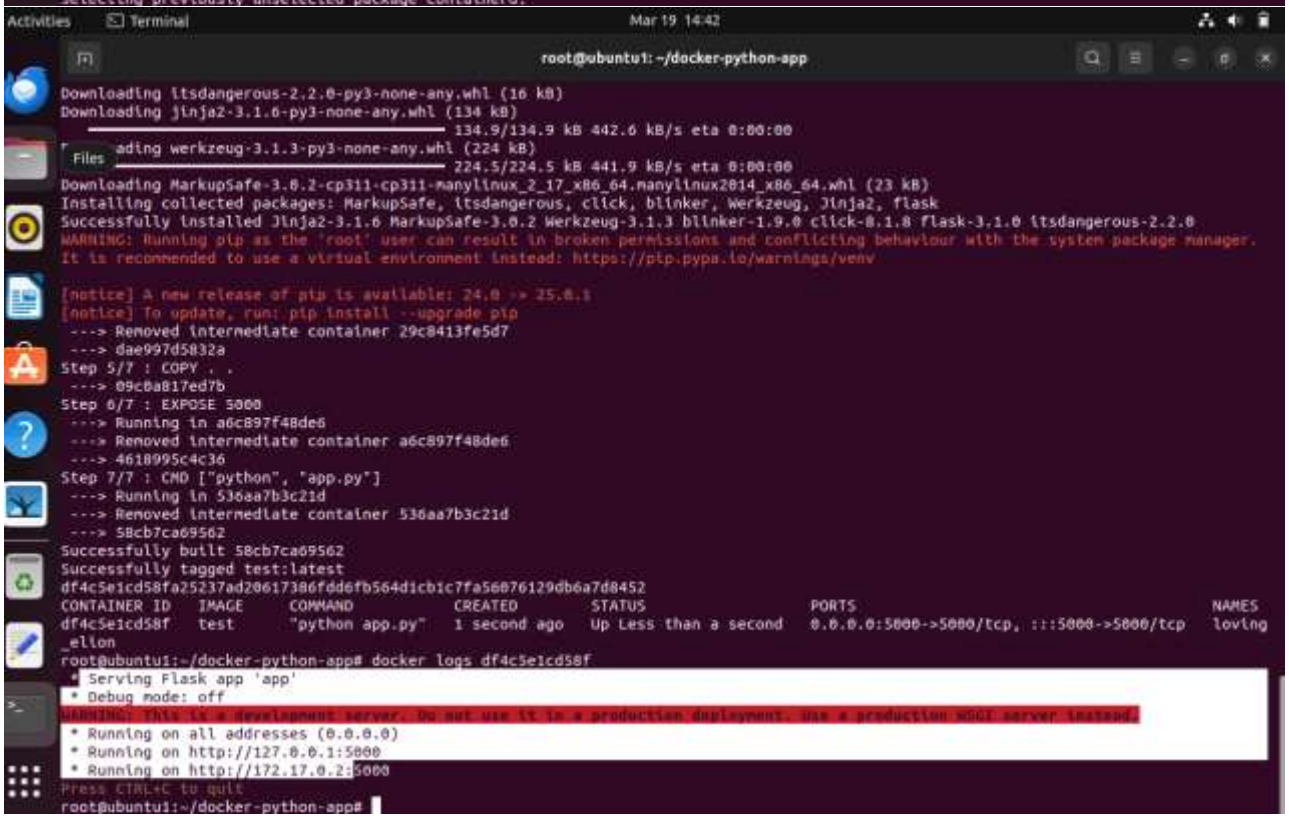
should see the output:

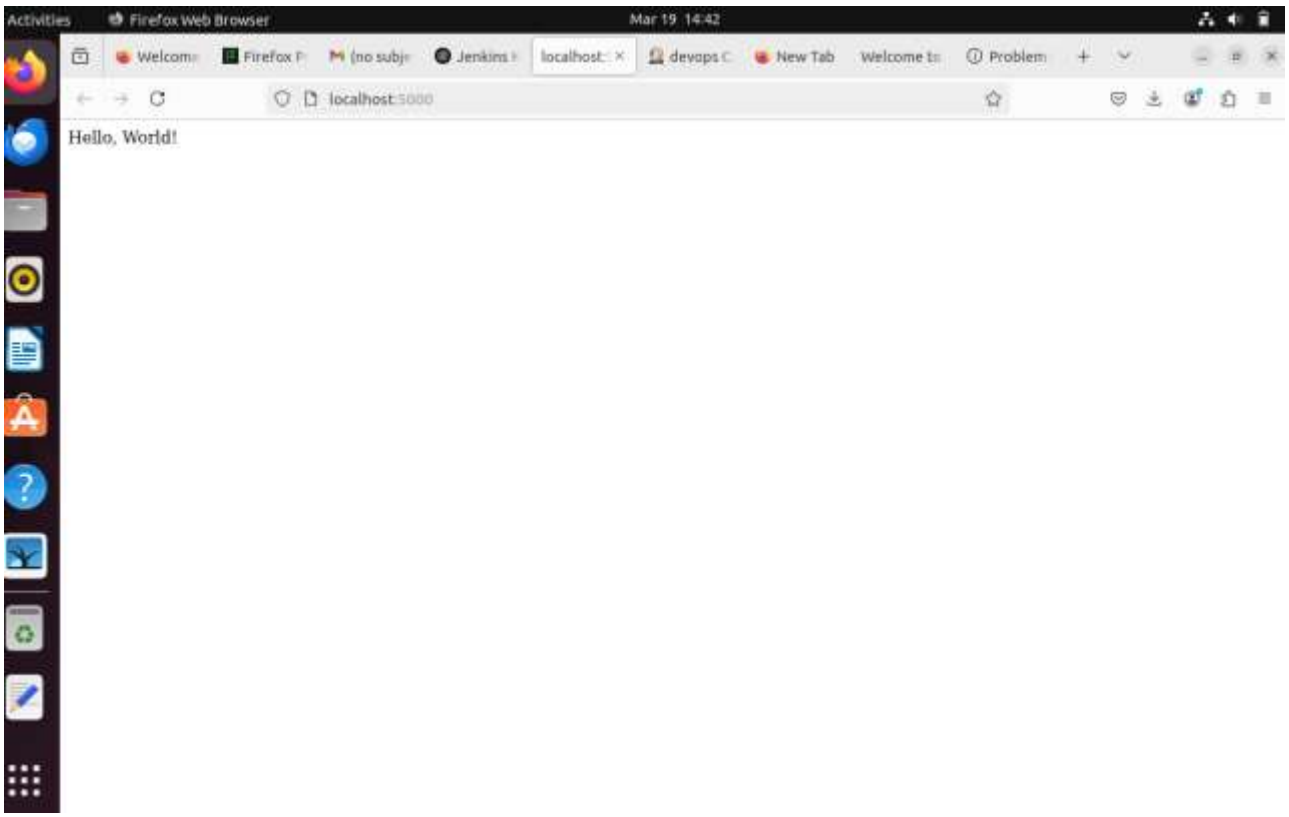
```
Hello, World! Running inside Docker!
```


Activities
Terminal
Mar 19 10:32
root@ubuntu1: ~/docker-python-app
root@ubuntu1:/home/vboxuser# sudo apt update
Hit:1 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Ign:2 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Hit:4 https://pkg.jenkins.io/debian-stable binary/ Release
Get:5 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Reading package lists... Done
E: Release file for http://security.ubuntu.com/ubuntu/dists/jammy-security/InRelease is not valid yet (invalid for another 8h 59min 20s). Updates for this repository will not be applied.
E: Release file for http://in.archive.ubuntu.com/ubuntu/dists/jammy-updates/InRelease is not valid yet (invalid for another 9h 1min 22s). Updates for this repository will not be applied.
E: Release file for http://in.archive.ubuntu.com/ubuntu/dists/jammy-backports/InRelease is not valid yet (invalid for another 9h 3min 59s). Updates for this repository will not be applied.
root@ubuntu1:/home/vboxuser# date
Wednesday 19 March 2025 09:16:03 AM IST
root@ubuntu1:/home/vboxuser# sudo timedatectl set-ntp on
sudo systemctl restart systemd-timesyncd
root@ubuntu1:/home/vboxuser# sudo timedatectl set-time "2024-03-1 9:18:00"
Failed to set time: Automatic time synchronization is enabled
root@ubuntu1:/home/vboxuser# timedatectl
Local time: Wed 2025-03-19 09:19:42 IST
Universal time: Wed 2025-03-19 03:49:42 UTC
RTC time: Wed 2025-03-19 03:49:41
Time zone: Asia/Kolkata (IST, +0530)
System clock synchronized: yes
NTP service: active
RTC in local TZ: no
root@ubuntu1:/home/vboxuser# sudo hwclock --show
2025-03-19 09:20:46.285749+05:30
root@ubuntu1:/home/vboxuser# sudo hwclock --set --date "2024-03-19 9:21:00"
sudo hwclock --systohc
root@ubuntu1:/home/vboxuser# sudo apt update
Ign:1 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 https://pkg.jenkins.io/debian-stable binary/ Release
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [43.1 kB]


Activities
Terminal
Mar 19 10:31
root@ubuntu1: ~/docker-python-app
unmae-m: command not found
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
0 0 0 0 0 0 0 0 --:--:-- 0:00:01 --:--:-- 0
Warning: Failed to create the file /user/local/bin/docker-composesudo: No such
Warning: file or directory
100 9 100 9 0 0 5 0 0:00:01 0:00:01 --:--:-- 5
curl: (23) Failure writing output to destination
curl: (6) Could not resolve host: chmod
curl: (6) Could not resolve host: +x
curl: (3) URL using bad/illegal format or missing URL
root@ubuntu1:/home/vboxuser# sudo chmod +x /usr/local/bin/docker-compose
root@ubuntu1:/home/vboxuser# docker-compose --version
Docker Compose version v2.34.0
root@ubuntu1:/home/vboxuser# mkdir ~/docker-python-app
root@ubuntu1:/home/vboxuser# cd ~/docker-python-app
root@ubuntu1:~/docker-python-app# nano app.py
root@ubuntu1:~/docker-python-app# nano requirements.txt
root@ubuntu1:~/docker-python-app# nano Dockerfile
root@ubuntu1:~/docker-python-app# **compose create compose docker
**compose: command not found
root@ubuntu1:~/docker-python-app# nano docker-compose.yml
root@ubuntu1:~/docker-python-app# sudo docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
root@ubuntu1:~/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
3.11: Pulling from library/python
7cd785773db4: Downloading [==========>] 17.25MB/48.47MB
091eb8249475: Downloading [==========>] 14.82MB/24.01MB
255774e8027b: Downloading [==========>] 22.01MB/64.4MB
353e14e5cc47: Waiting
963091970bc2: Waiting
e7235c43f7e3: Waiting
7f221c50e407: Waiting


Activities
Terminal
Mar 19 10:32
root@ubuntu1: ~/docker-python-app
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
root@ubuntu1:/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 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/universe amd64 pigz amd64 2.6-1 [63.6 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 bridge-utils amd64 1.7-1ubuntu3 [34.4 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 runc amd64 1.1.12-0ubuntu2~22.04.1 [8,405 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 containerd amd64 1.7.24-0ubuntu1~22.04.1 [37.3 MB]
Get:5 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 docker.io amd64 26.1.3-0ubuntu1~22.04.1 [32.5 MB]
Get:6 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 liberror-perl all 0.17029-1 [26.5 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git-man all 1:2.34.1-1ubuntu1.12 [955 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git amd64 1:2.34.1-1ubuntu1.12 [3,165 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu jammy/universe amd64 ubuntu-fan all 0.12.16 [35.2 kB]
Fetched 82.4 MB in 2min 20s (589 kB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 202390 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.


Activities
Terminal
Mar 19 14:42
root@ubuntu1: ~/docker-python-app
Files
Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Downloading jinja2-3.1.6-py3-none-any.whl (134 kB)
134.9/134.9 kB 442.6 kB/s eta 0:00:00
ading werkzeug-3.1.3-py3-none-any.whl (224 kB)
224.5/224.5 kB 441.9 kB/s eta 0:00:00
Downloading MarkupSafe-3.0.2-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (23 kB)
Installing collected packages: MarkupSafe, itsdangerous, click, blinker, Werkzeug, Jinja2, flask
Successfully installed Jinja2-3.1.6 MarkupSafe-3.0.2 Werkzeug-3.1.3 blinker-1.9.0 click-8.1.8 flask-3.1.0 itsdangerous-2.2.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
[notice] A new release of pip is available: 24.0 -> 25.0.1
[notice] To update, run: pip install --upgrade pip
---> Removed intermediate container 29c8413fe5d7
---> dae997d5832a
Step 5/7 : COPY . .
---> 09c0a817ed7b
Step 6/7 : EXPOSE 5000
---> Running in a6c897f48de6
---> Removed intermediate container a6c897f48de6
---> 4618995c4c36
Step 7/7 : CMD ["python", "app.py"]
---> Running in 536aa7b3c21d
---> Removed intermediate container 536aa7b3c21d
---> 58cb7ca69562
Successfully built 58cb7ca69562
Successfully tagged test:latest
df4c5e1cd58fa25237ad20617386fdd6fb564d1cb1c7fa56076129db6a7d8452
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
df4c5e1cd58f test "python app.py" 1 second ago Up Less than a second 0.0.0.0:5000->5000/tcp, :::5000->5000/tcp loving
_ellon
root@ubuntu1:~/docker-python-app# docker logs df4c5e1cd58f
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.17.0.2:5000
Press CTRL+C to quit
root@ubuntu1:~/docker-python-app#



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
~/dockerpython-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

```
pipeline {
    agent any
    environment {
        DOCKER_IMAGE = "jshema334/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: 'jshema334', usernameVariable: 'GIT_USER',
passwordVariable: 'GIT_TOKEN')]) {
                    git url: "https://$GIT_USER:$GIT_TOKEN@github.com/Hema334/docker.git", branch: 'main'
                }
            }
        }

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

        stage('Login to Docker Registry') {
            steps {
                withCredentials([usernamePassword(credentialsId: 'hemajs334', usernameVariable: 'DOCKER_USER',
passwordVariable: 'DOCKER_PASS')]) {
                    sh 'echo $DOCKER_PASS | docker login -u $DOCKER_USER --password-stdin'
                }
            }
        }

           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."
        }
    }
}
```

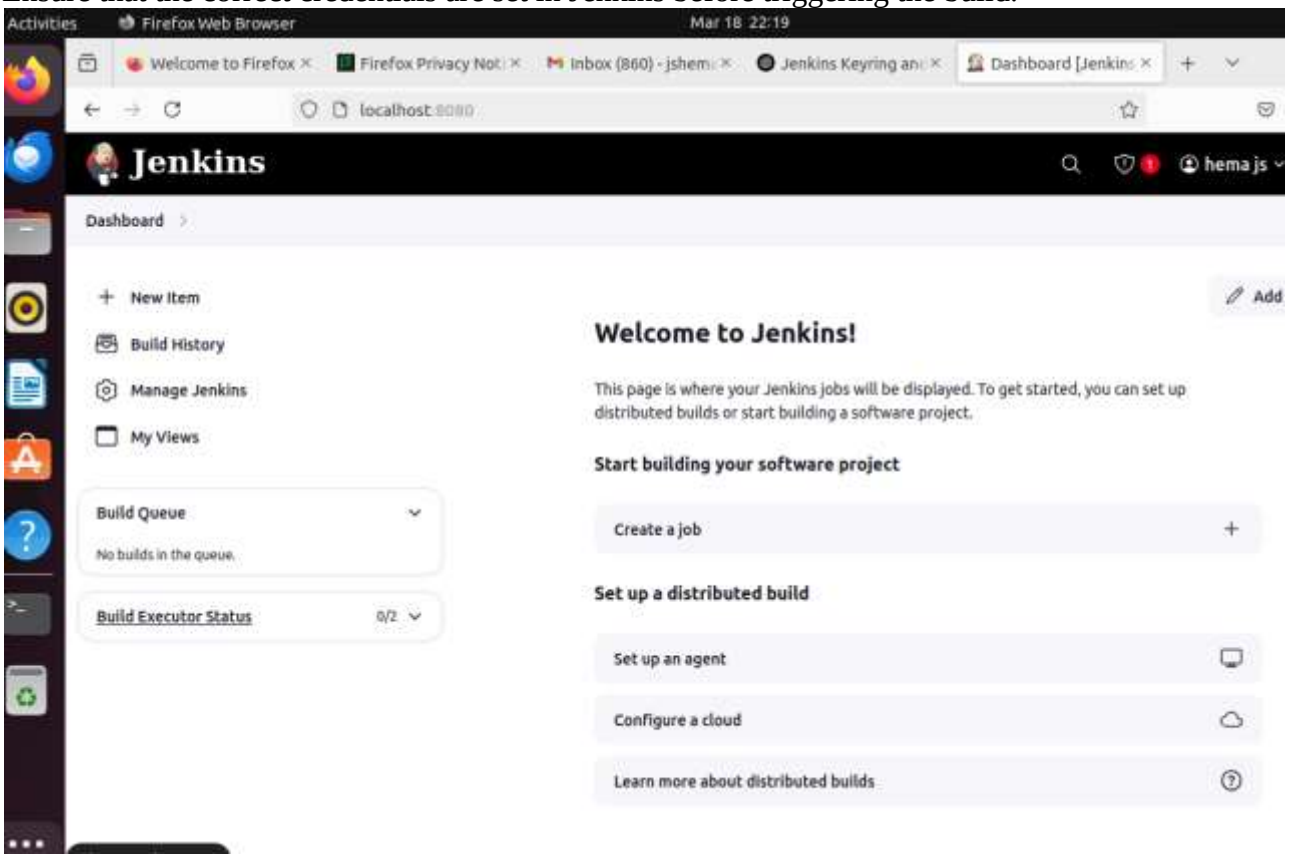
Running Jenkins Build

Step 1: Resolve Security Error

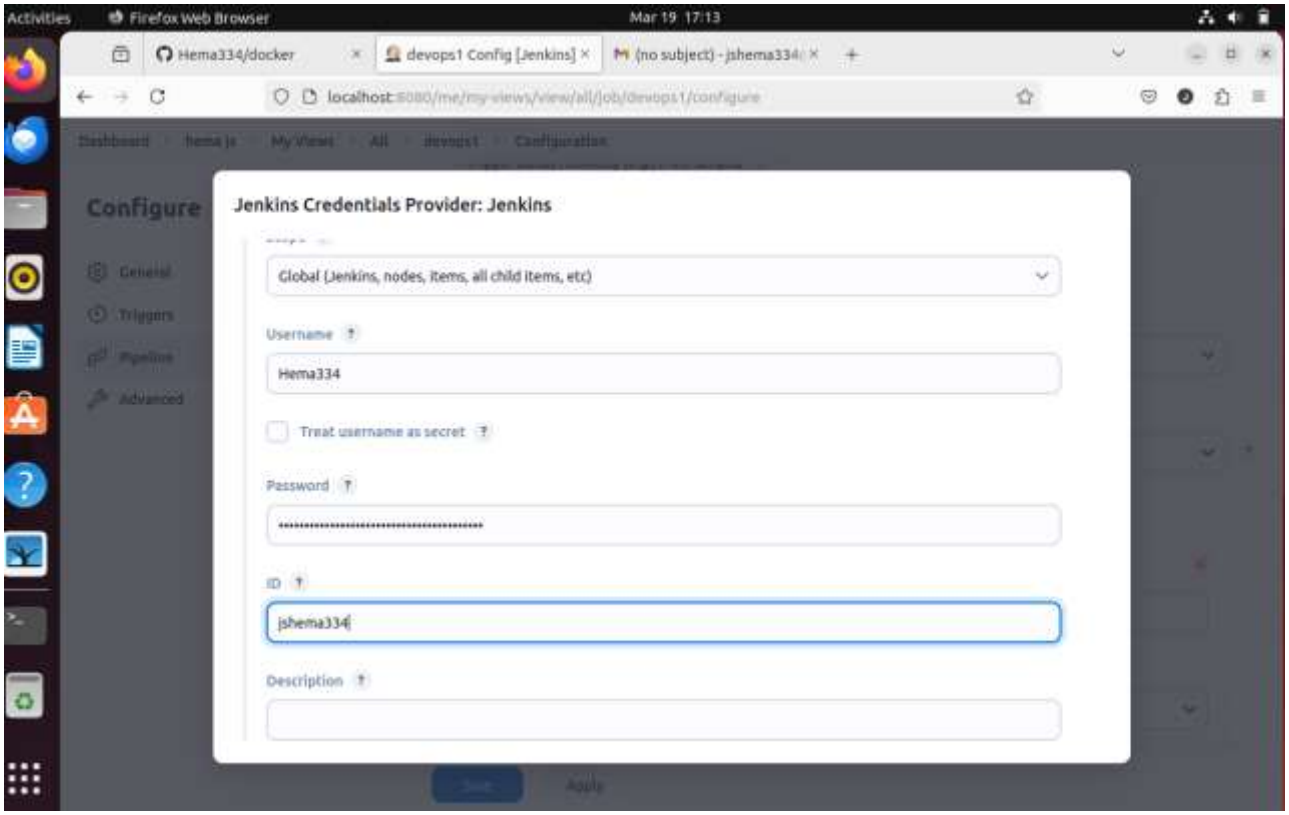
```
sudo usermod -aG docker jenkins sudo
systemctl restart jenkins
```

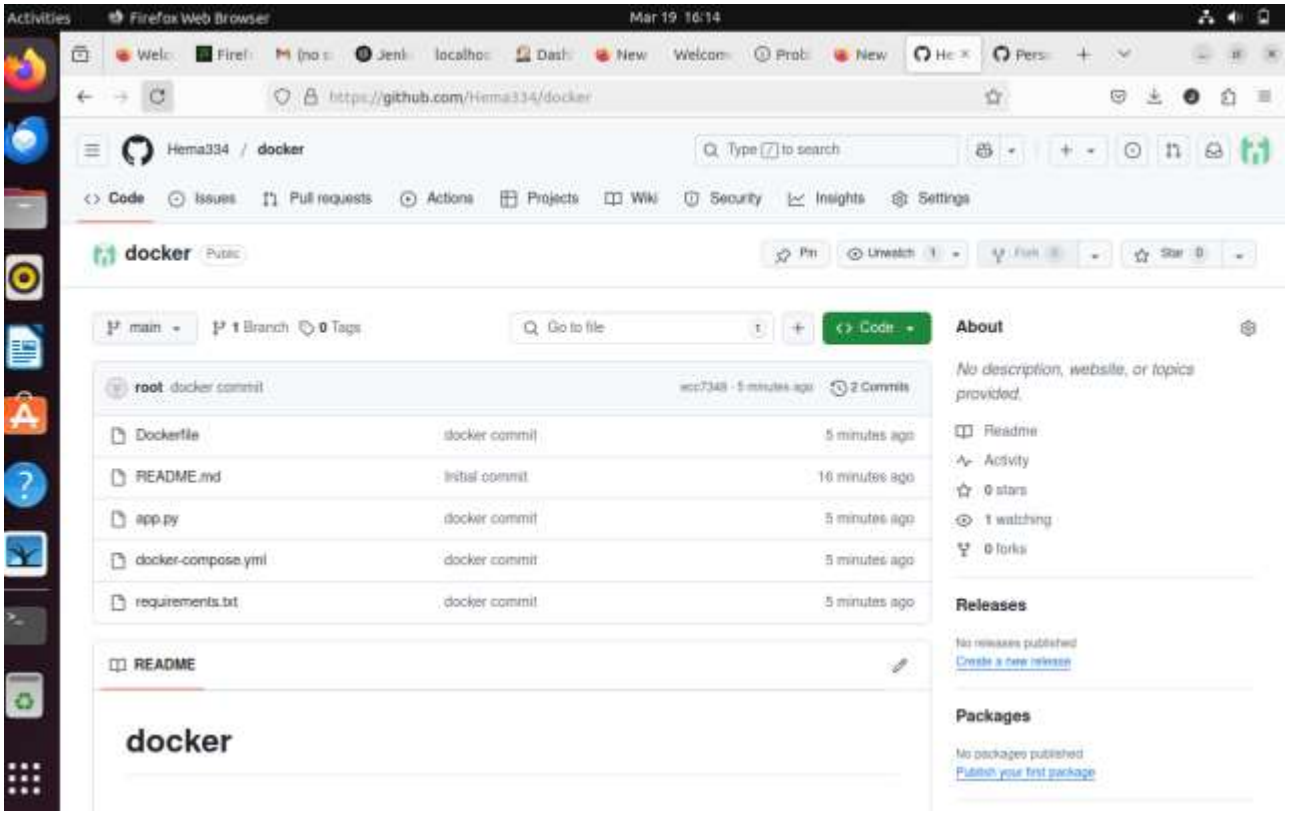
Step 2: Verify Jenkins Credentials

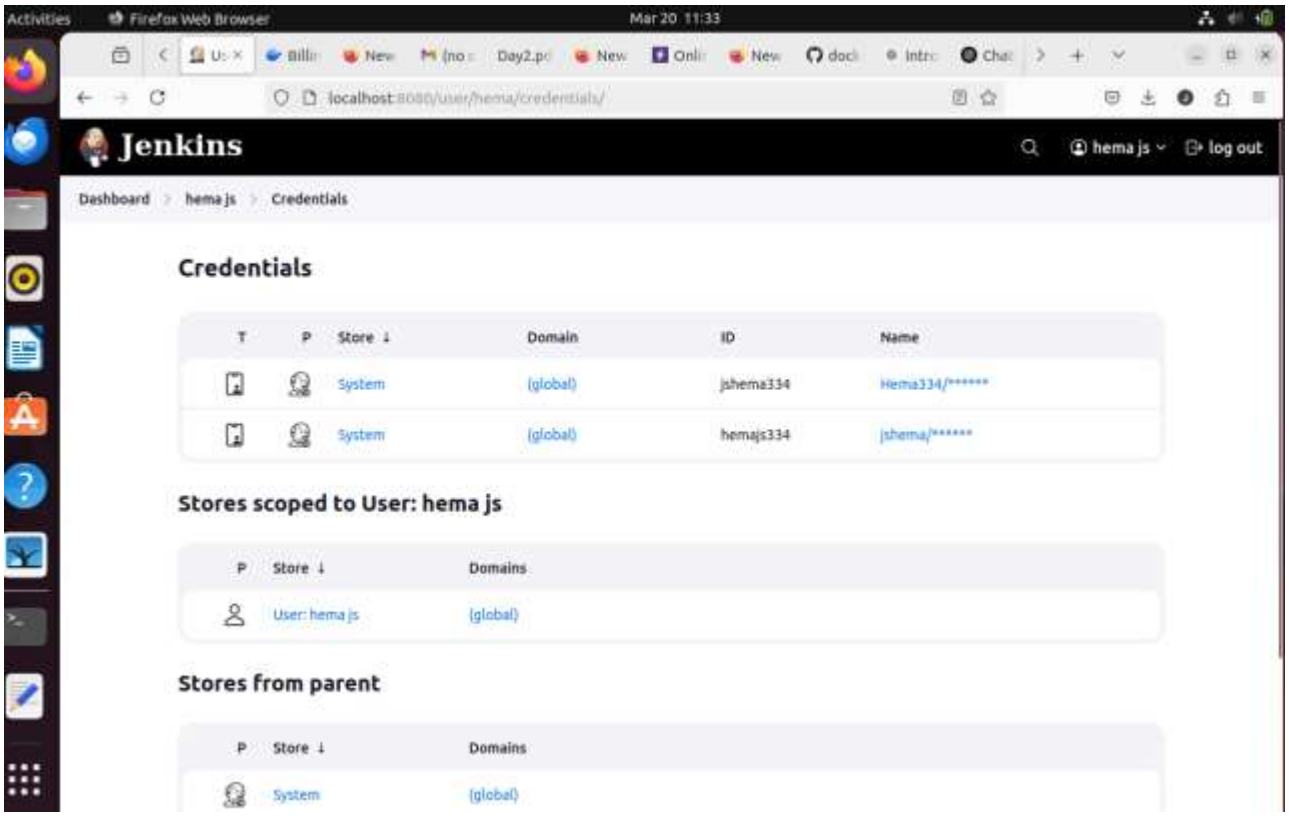
Ensure that the correct credentials are set in Jenkins before triggering the build.

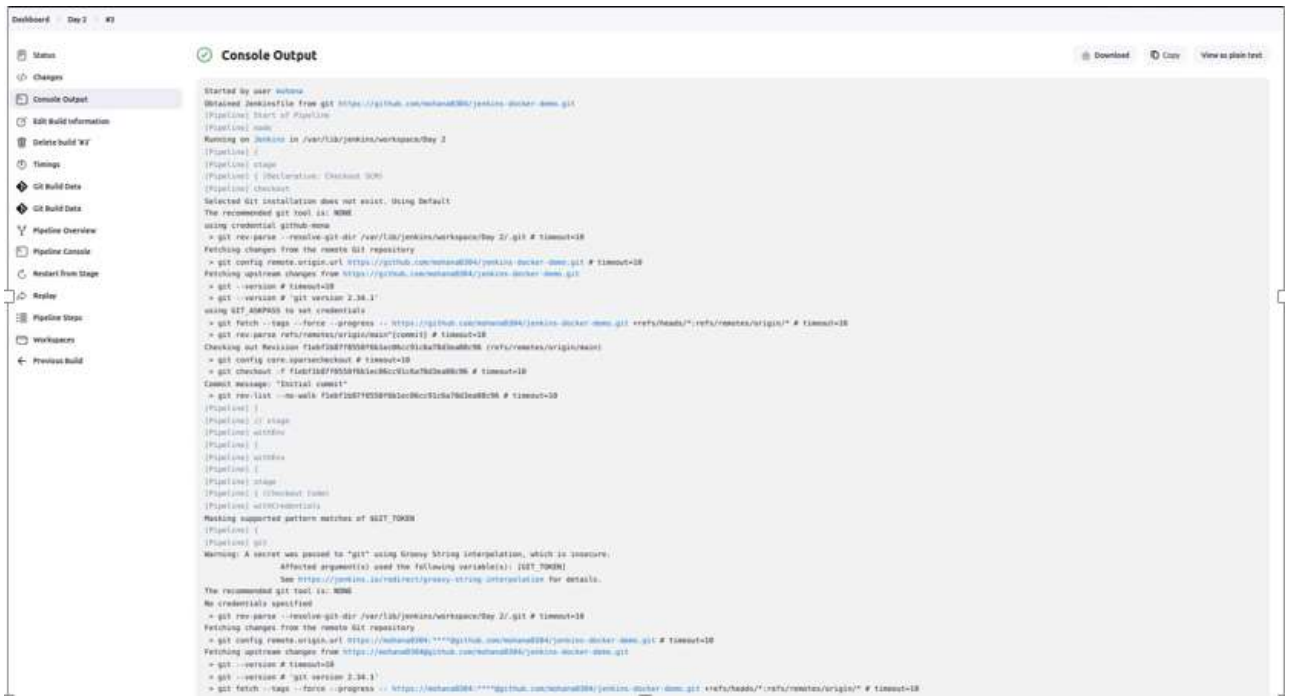



Dashboard
Day 2
Status
Changes
Build Now
Configure
Delete Pipeline
Stages
Rename
Pipeline Syntax
Builds
No builds
Day 2
Permalinks
Add description


Activities
Firefox Web Browser
Mar 19 17:13
Hema334/docker
devops1 Config [Jenkins]
(no subject) - jshema334
localhost:8080/me/my-views/view/all/job/devops1/configure
Configure
Jenkins Credentials Provider: Jenkins
Global (Jenkins, nodes, items, all child items, etc)
Username
Hema334
Treat username as secret
Password
ID
jshema334
Description


Activities
Firefox Web Browser
Mar 19 16:14
https://github.com/Hema334/docker
Hema334 / docker
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
docker
Public
main
1 Branch
0 Tags
Go to file
Code
root docker commit
2 Commits
Dockerfile docker commit 5 minutes ago
README.md Initial commit 16 minutes ago
app.py docker commit 5 minutes ago
docker-compose.yml docker commit 5 minutes ago
requirements.txt docker commit 5 minutes ago
README
docker
About
No description, website, or topics provided.
Readme
Activity
0 stars
1 watching
0 forks
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package


Activities
Firefox Web Browser
Mar 20 11:33
localhost:8080/user/hema/credentials/
Jenkins
hema js
log out
Dashboard
hema js
Credentials
Credentials
T
P
Store ↓
Domain
ID
Name
System
(global)
jshema334
Hema334/******
System
(global)
hemajs334
jshema/******
Stores scoped to User: hema js
P
Store ↓
Domains
User: hema js
(global)
Stores from parent
P
Store ↓
Domains
System
(global)



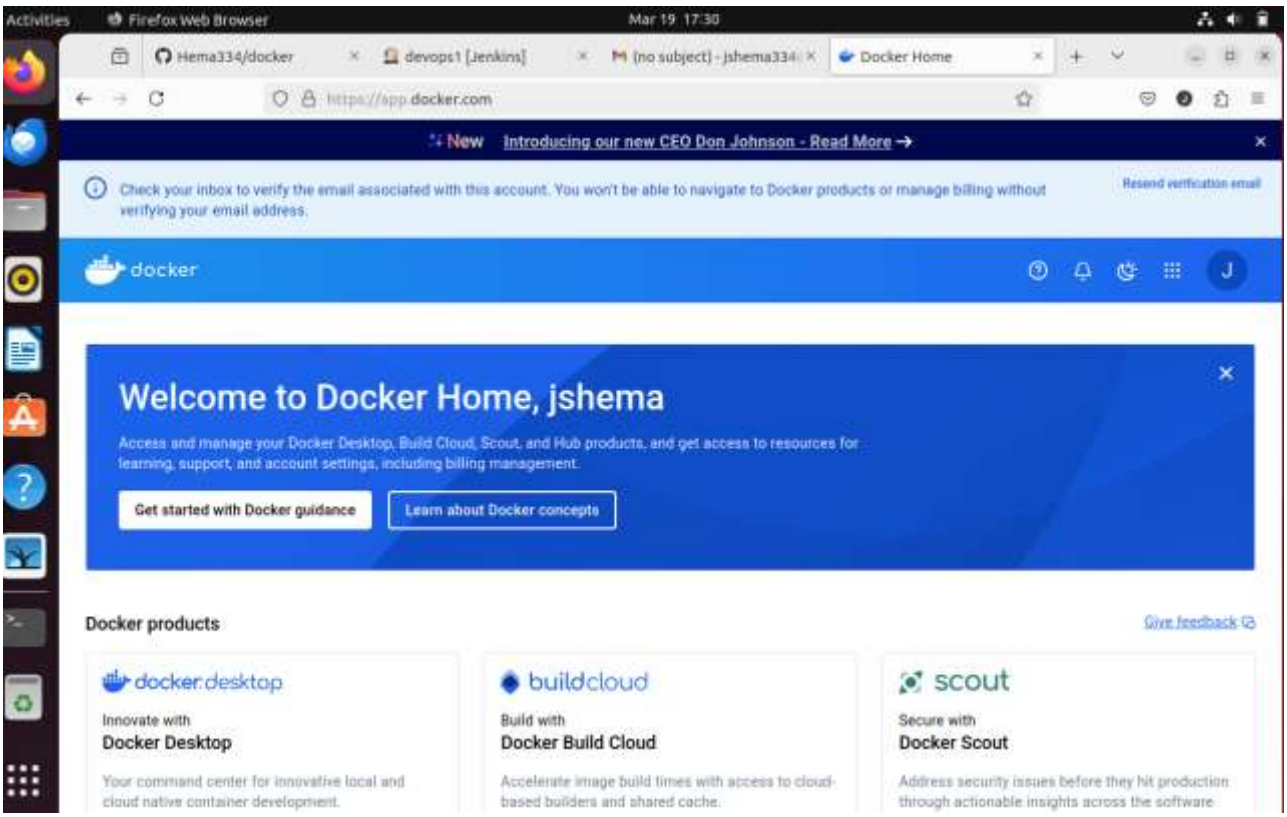
Step 3: Run the Build

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
```


Activities
Firefox Web Browser
Mar 19 17:30
Hema334/docker
devops1 [Jenkins]
(no subject) - jshema334
Docker Home
https://app.docker.com
New
Introducing our new CEO Don Johnson - Read More →
Check your inbox to verify the email associated with this account. You won't be able to navigate to Docker products or manage billing without verifying your email address.
Resend verification email
docker
Welcome to Docker Home, jshema
Access and manage your Docker Desktop, Build Cloud, Scout, and Hub products, and get access to resources for learning, support, and account settings, including billing management.
Get started with Docker guidance
Learn about Docker concepts
Docker products
Give feedback
docker desktop
Innovate with
Docker Desktop
Your command center for innovative local and cloud native container development.
buildcloud
Build with
Docker Build Cloud
Accelerate image build times with access to cloud-based builders and shared cache.
scout
Secure with
Docker Scout
Address security issues before they hit production through actionable insights across the software


Hello, World! Running inside Docker!