

DevOps Task 2

Build of docker-compose

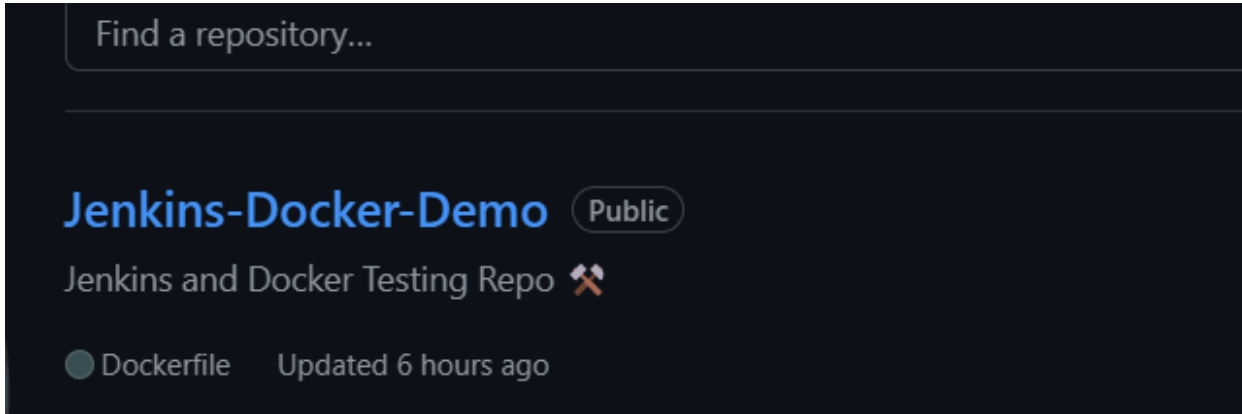
```
surya@Surya: ~/docker-python-app$ nano docker-compose.yml
surya@Surya:~/docker-python-app$ sudo docker-compose build
[sudo] password for surya:
WARN[0000] /home/surya/docker-python-app/docker-compose.yml: the attribute `version`
` is obsolete, it will be ignored, please remove it to avoid potential confusion
Compose can now delegate builds to bake for better performance.
To do so, set COMPOSE_BAKE=true.
[+] Building 6.6s (11/11) FINISHED                                docker:default
=> [web internal] load build definition from Dockerfile           0.0s
=> => transferring dockerfile: 370B                               0.0s
=> [web internal] load metadata for docker.io/library/python:3.11 0.0s
=> [web internal] load .dockerignore                             0.0s
=> => transferring context: 2B                                     0.0s
=> [web 1/5] FROM docker.io/library/python:3.11                 0.1s
=> [web internal] load build context                             0.1s
=> => transferring context: 824B                                   0.0s
=> [web 2/5] WORKDIR /app                                         0.0s
=> [web 3/5] COPY requirements.txt .                             0.0s
=> [web 4/5] RUN pip install --no-cache-dir -r requirements.txt   6.1s
=> [web 5/5] COPY . .                                             0.0s
=> [web] exporting to image                                       0.2s
=> => exporting layers                                           0.1s
=> => writing image sha256:77f265a15d4b3b745e6f965a476f443a7edcd1304680 0.0s
=> => naming to docker.io/library/docker-python-app-web         0.0s
=> [web] resolving provenance for metadata file                  0.0s
[+] Building 1/1
✔web Built                                                       0.0s
```

```
surya@Surya:~/docker-python-app$ sudo docker-compose up -d
WARN[0000] /home/surya/docker-python-app/docker-compose.yml: the attribute `version`
` is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 2/2
✔Network docker-python-app_default Created                       0.1s
✔Container docker-python-app-web-1 Started                      0.4s
```

```
surya@Surya:~/docker-python-app$ sudo docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS
PORTS         NAMES
048acd5a7d26   docker-python-app-web               "python app.py"         6 seconds ago Up 5 seconds
0.0.0.0:5000->5000/tcp, :::5000->5000/tcp   docker-python-app-web-1
```

Step 1: Create a GitHub Repository

1. Click on New Repository
2. Set the repository name as: **Jenkins-Docker-Demo**
3. Select Public
4. Click on Create Repository (<https://github.com/Surya-2k4/Jenkins-Docker-Demo.git>)



Step 2: Generate a Personal Access Token (PAT)

1. Go to: [GitHub Developer Settings](#)
2. Navigate to:
Personal Access Tokens → Tokens (classic) → Generate New Token
3. Set a **Note**: **Jenkins Docker Pipeline**
4. Select the following **Scopes**:
 - **repo**
 - **workflow**
 - **admin:repo_hook**
5. Generate the token.

Token:

ghp_pY2u7AEWKIb2r0MgfPbDIDoh1IgMKT0g****

Step 3: Start and Enable Jenkins

1. Open the terminal.
2. Enable Jenkins to start at boot

```
sudo systemctl enable jenkins
```

```
sudo systemctl start jenkins
```

```
surya@Surya:/mnt/c/Users/SURYA$ sudo systemctl enable jenkins
[sudo] password for surya:
Synchronizing state of jenkins.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable jenkins
surya@Surya:/mnt/c/Users/SURYA$ sudo systemctl start jenkins
surya@Surya:/mnt/c/Users/SURYA$
```

Step 4: Access Jenkins

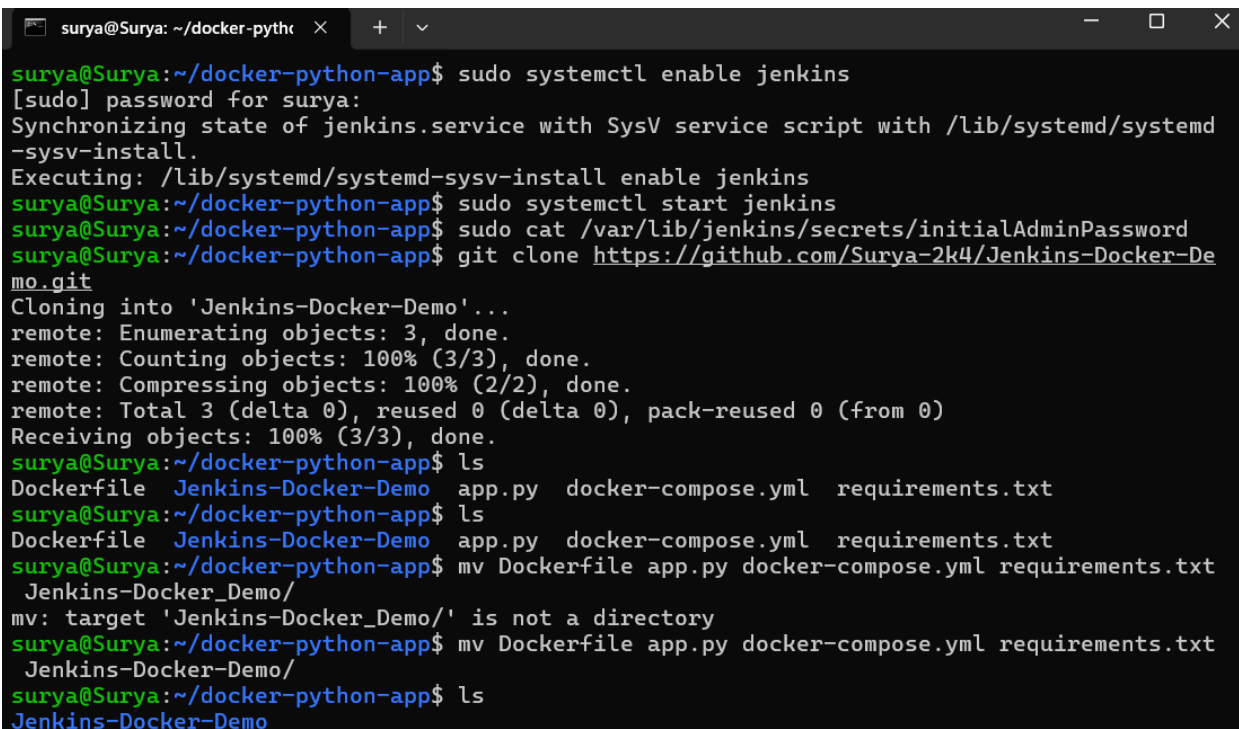
1. Open your browser:
<http://localhost:8080>
2. Log in with the following credentials:
 - Username: Admin
 - Password: 316b8fd8d958454e928ed60*****

Step 5:

In Jenkins, create a new project by selecting Pipeline and entering the repo URL: <https://github.com/Surya-2k4/Jenkins-Docker-Demo.git>. Use the **main** branch for the pipeline source.

```
git clone https://github.com/Surya-2k4/Jenkins-Docker-Demo.git
```

```
cd Jenkins-Docker-Demo
```

A terminal window titled 'surya@Surya: ~/docker-python-app' showing the execution of Jenkins setup commands. The user enables and starts Jenkins, then clones the repository 'https://github.com/Surya-2k4/Jenkins-Docker-Demo.git'. After cloning, the user lists files in the current directory, showing 'Dockerfile', 'Jenkins-Docker-Demo', 'app.py', 'docker-compose.yml', and 'requirements.txt'. Finally, the user moves these files into a subdirectory named 'Jenkins-Docker-Demo/'.

```
surya@Surya:~/docker-python-app$ sudo systemctl enable jenkins
[sudo] password for surya:
Synchronizing state of jenkins.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable jenkins
surya@Surya:~/docker-python-app$ sudo systemctl start jenkins
surya@Surya:~/docker-python-app$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
surya@Surya:~/docker-python-app$ git clone https://github.com/Surya-2k4/Jenkins-Docker-Demo.git
Cloning into 'Jenkins-Docker-Demo'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
surya@Surya:~/docker-python-app$ ls
Dockerfile  Jenkins-Docker-Demo  app.py  docker-compose.yml  requirements.txt
surya@Surya:~/docker-python-app$ ls
Dockerfile  Jenkins-Docker-Demo  app.py  docker-compose.yml  requirements.txt
surya@Surya:~/docker-python-app$ mv Dockerfile app.py docker-compose.yml requirements.txt
Jenkins-Docker-Demo/
mv: target 'Jenkins-Docker-Demo/' is not a directory
surya@Surya:~/docker-python-app$ mv Dockerfile app.py docker-compose.yml requirements.txt
Jenkins-Docker-Demo/
surya@Surya:~/docker-python-app$ ls
Jenkins-Docker-Demo
```

Step 6:

List the files using `ls` and verify that `app.py`, `Dockerfile`, `requirements.txt`, and `compose-docker.yml` are present. Move all required project files into the repo folder using:

```
mv /path/to/your/files/* Jenkins-Docker-Demo/
```

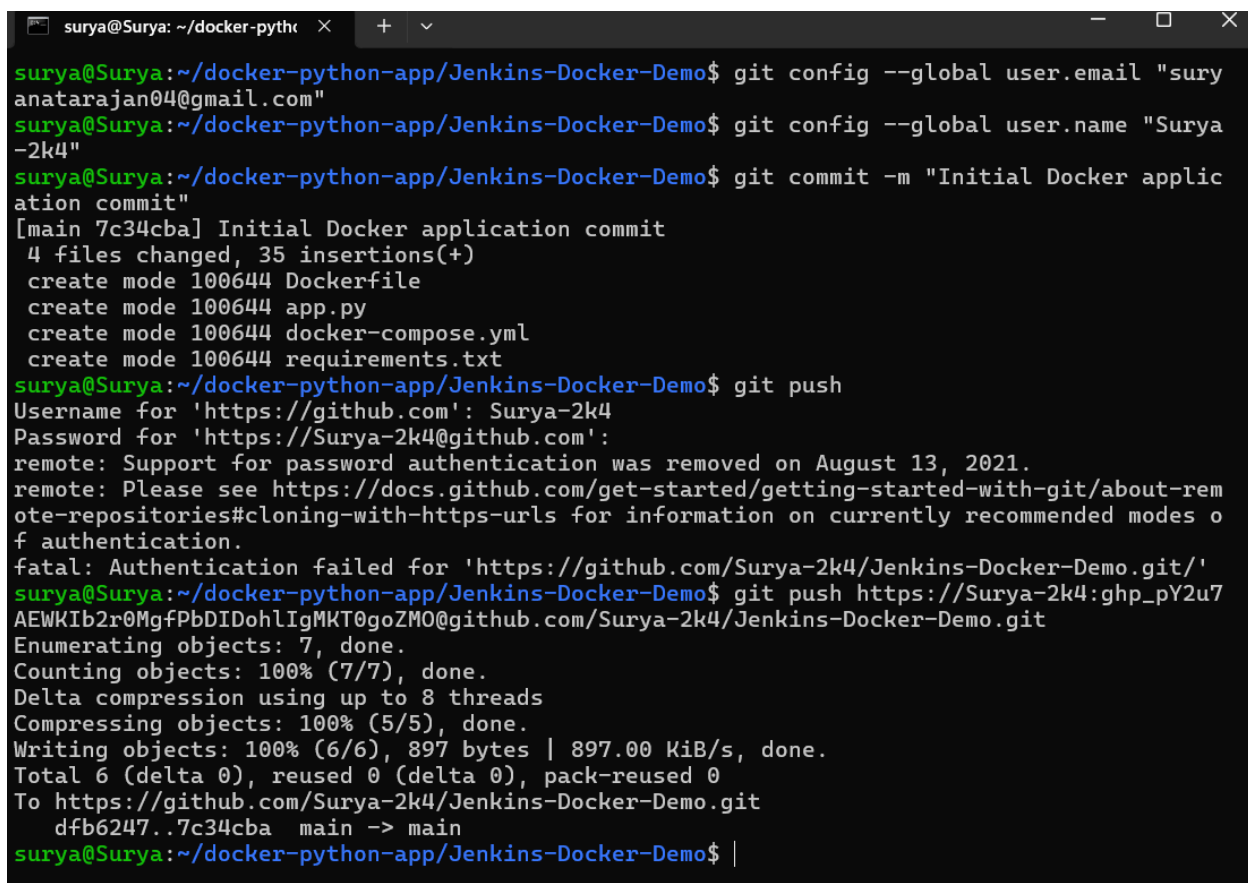
Add and push the files to GitHub:

```
git add .
```

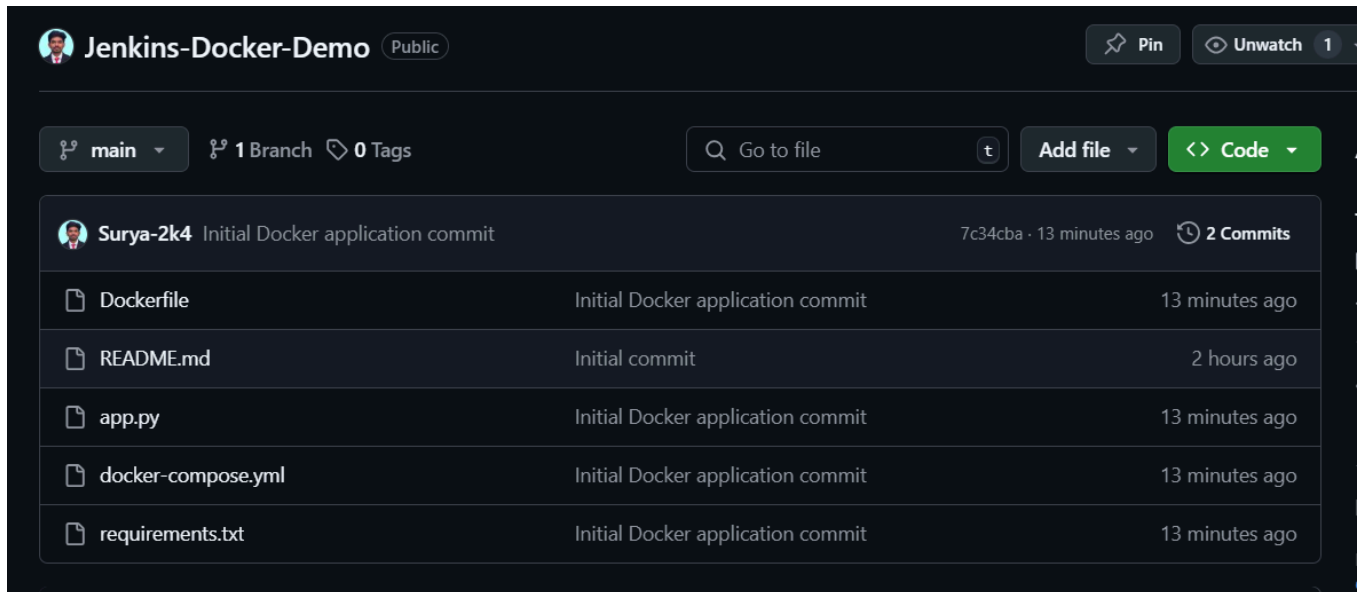
```
git commit -m "Initial Docker application commit"
```

```
git push
```

```
https://Surya-2k4:ghp_pY2u7AEWKIb2r0MgfPbDIDoh1IgMKT0goZM0@github.com/Surya-2k4/Jenkins-Docker-Demo.git
```

A terminal window with a dark background and light green text. The window title is 'surya@Surya: ~/docker-python-app'. The terminal shows the following commands and output:

```
surya@Surya:~/docker-python-app/Jenkins-Docker-Demo$ git config --global user.email "suryanatarajan04@gmail.com"
surya@Surya:~/docker-python-app/Jenkins-Docker-Demo$ git config --global user.name "Surya-2k4"
surya@Surya:~/docker-python-app/Jenkins-Docker-Demo$ git commit -m "Initial Docker application commit"
[main 7c34cba] Initial Docker application commit
4 files changed, 35 insertions(+)
create mode 100644 Dockerfile
create mode 100644 app.py
create mode 100644 docker-compose.yml
create mode 100644 requirements.txt
surya@Surya:~/docker-python-app/Jenkins-Docker-Demo$ git push
Username for 'https://github.com': Surya-2k4
Password for 'https://Surya-2k4@github.com':
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls for information on currently recommended modes of authentication.
fatal: Authentication failed for 'https://github.com/Surya-2k4/Jenkins-Docker-Demo.git/'
surya@Surya:~/docker-python-app/Jenkins-Docker-Demo$ git push https://Surya-2k4:ghp_pY2u7AEWKIb2r0MgfPbDIDoh1IgMKT0goZM0@github.com/Surya-2k4/Jenkins-Docker-Demo.git
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 897 bytes | 897.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Surya-2k4/Jenkins-Docker-Demo.git
dfb6247..7c34cba main -> main
surya@Surya:~/docker-python-app/Jenkins-Docker-Demo$
```



Step 7: Create a DockerHub account at DockerHub with:

- Username: **surya2k4**
- Password: **Pass**

```
surya@Surya:~/docker-python-app/Jenkins-Docker-Demo$ docker login -u surya2k4
Password:
WARNING! Your password will be stored unencrypted in /home/surya/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
surya@Surya:~/docker-python-app/Jenkins-Docker-Demo$ sudo usermod -aG docker jenkins
[sudo] password for surya:
surya@Surya:~/docker-python-app/Jenkins-Docker-Demo$ sudo systemctl restart jenkins
surya@Surya:~/docker-python-app/Jenkins-Docker-Demo$ |
```

In Jenkins, go to Manage Jenkins → Manage Credentials → System → Global Credentials. Add new credentials with:

- Username: **surya2k4**
- Password: **(docker hub pass)**
- ID: **docker**

Step 8: Create a Jenkinsfile in your project folder and replace the credentials of your own Docker and Github

```
surya@Surya:~/docker-python-app/Jenkins-Docker-Demo$ cat Jenkinsfile
pipeline {
  agent any
  environment {
    DOCKER_IMAGE = "surya2k4/docker-app:latest" // Docker image name
    CONTAINER_NAME = "docker-running-app"
    REGISTRY_CREDENTIALS = "docker" // Jenkins credentials ID for Docker Hub
  }

  stages {
    stage('Checkout Code') {
      steps {
        withCredentials([usernamePassword(credentialsId: 'demo', usernameVariable: 'GIT_USER', passwordVariable: 'GIT_TOKEN')]) {
          git url: "https://$GIT_USER:$GIT_TOKEN@github.com/Surya-2k4/Jenkins-Docker-Demo.git", branch: 'main'
        }
      }
    }

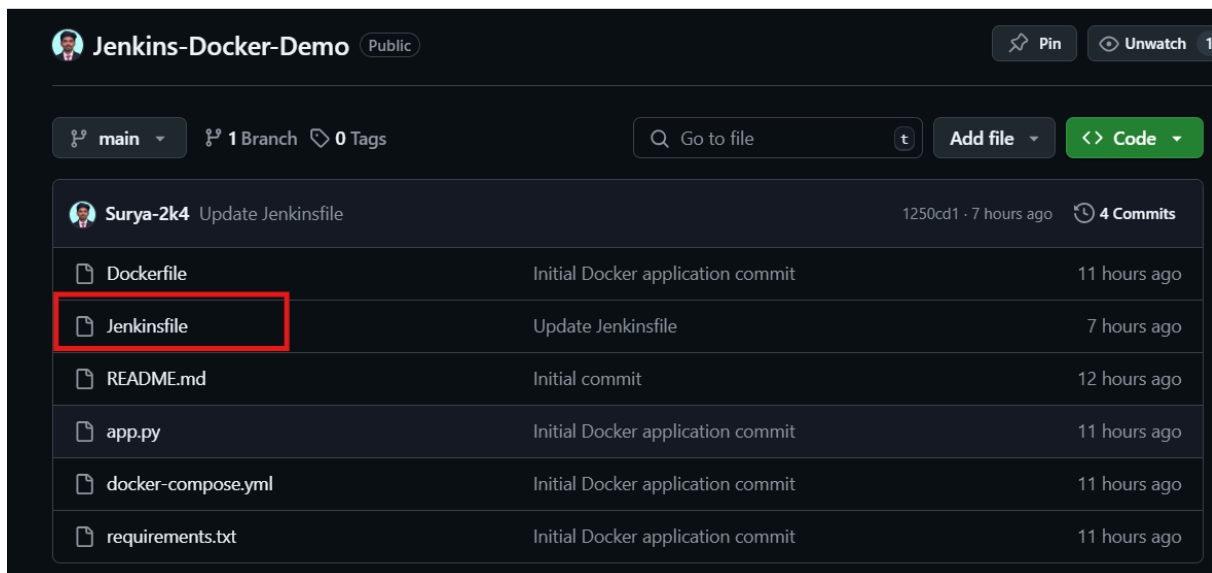
    stage('Build Docker Image') {
      steps {
        script {
          sh 'docker build -t ${DOCKER_IMAGE} .' // Ensure the image is built with
the correct tag
        }
      }
    }
  }
}
```

Push the **Jenkinsfile** to the repo:

```
git add Jenkinsfile
```

```
git commit -m "Added Jenkinsfile"
```

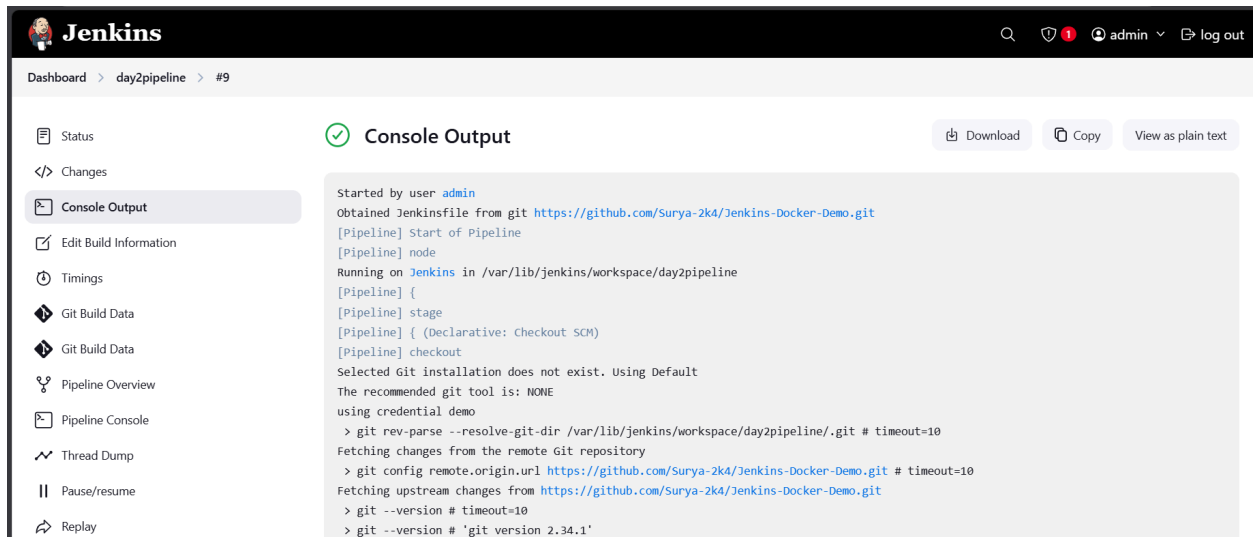
```
git push origin main
```



The screenshot shows the GitHub repository page for 'Jenkins-Docker-Demo'. The repository is public and has 1 branch (main) and 0 tags. The file list shows the following files and their commit history:

File	Commit Message	Commit Time
Dockerfile	Initial Docker application commit	11 hours ago
Jenkinsfile	Update Jenkinsfile	7 hours ago
README.md	Initial commit	12 hours ago
app.py	Initial Docker application commit	11 hours ago
docker-compose.yml	Initial Docker application commit	11 hours ago
requirements.txt	Initial Docker application commit	11 hours ago

Step 9: In Jenkins, build the project and verify the console output.



Step 10: Test the application by visiting <http://localhost:5001>

