2. Automating the CI/CD Pipeline

Overview

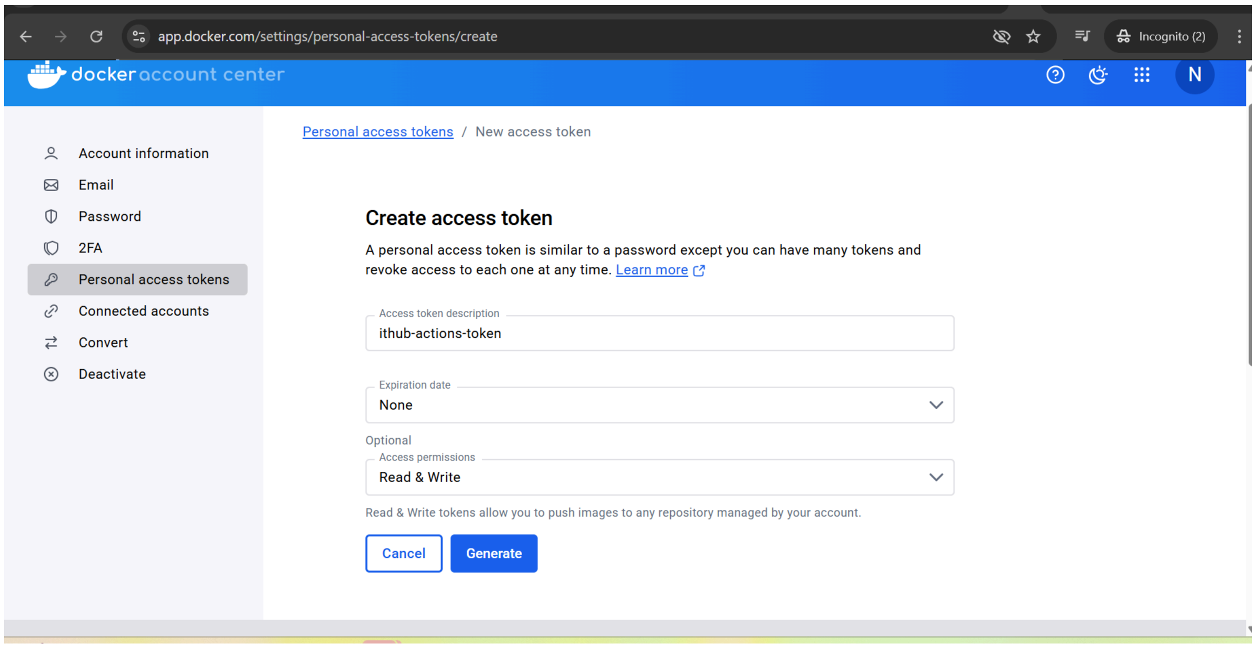
Set up an automated CI/CD pipeline to streamline the deployment processes.

Tasks

- **Continuous Deployment:**
  - Configure automated deployment to a container registry using GitHub Actions, to be executed when changes are successfully merged into the main branch.

Step 1: Generate a Docker Hub Access Token

1. Log in to Docker Hub.
2. Click on your profile icon (top-right) → Account Settings.
3. Go to Security → New Access Token.
4. Give it a name (e.g., github-actions-token) and select Read/Write permissions.
5. Click Generate Token and copy.



Create access token

A personal access token is similar to a password except you can have many tokens and revoke access to each one at any time. Learn more

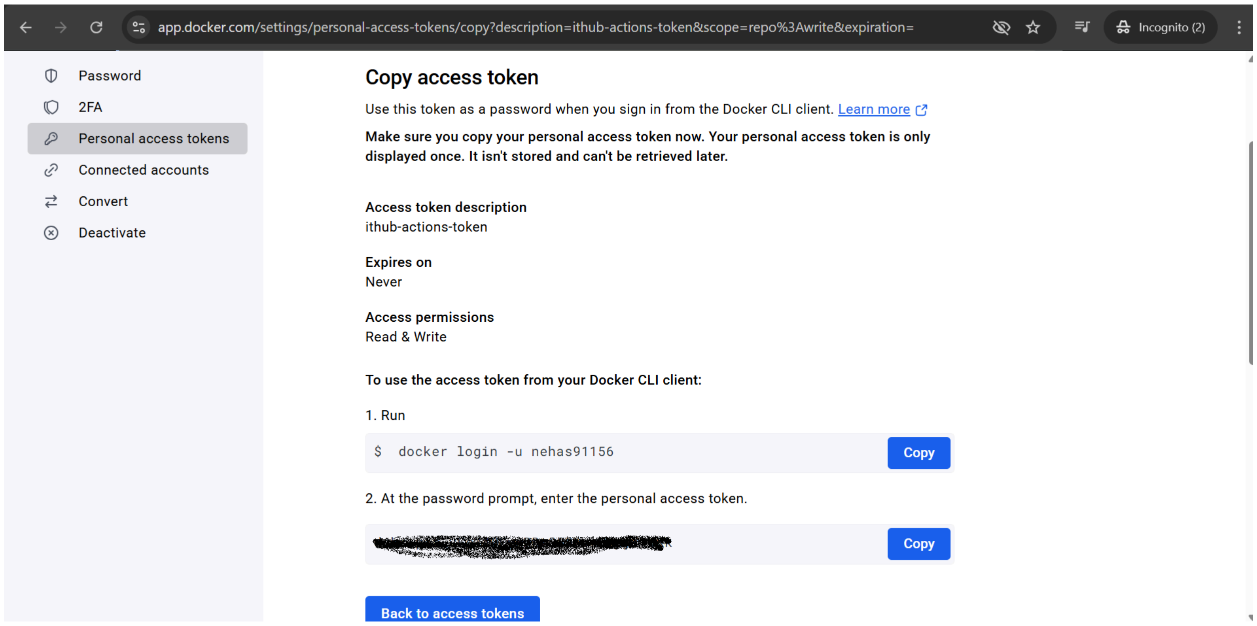
Access token description: ithub-actions-token

Expiration date: None

Optional

Access permissions: Read & Write

Read & Write tokens allow you to push images to any repository managed by your account.


app.docker.com/settings/personal-access-tokens/copy?description=ithub-actions-token&scope=repo%3Awrite&expiration=
Incognito (2)
Password
2FA
Personal access tokens
Connected accounts
Convert
Deactivate

Copy access token

Use this token as a password when you sign in from the Docker CLI client. Learn more

Make sure you copy your personal access token now. Your personal access token is only displayed once. It isn't stored and can't be retrieved later.

Access token description
ithub-actions-token

Expires on
Never

Access permissions
Read & Write

To use the access token from your Docker CLI client:

1. Run

```
$  docker login -u nehas91156
```

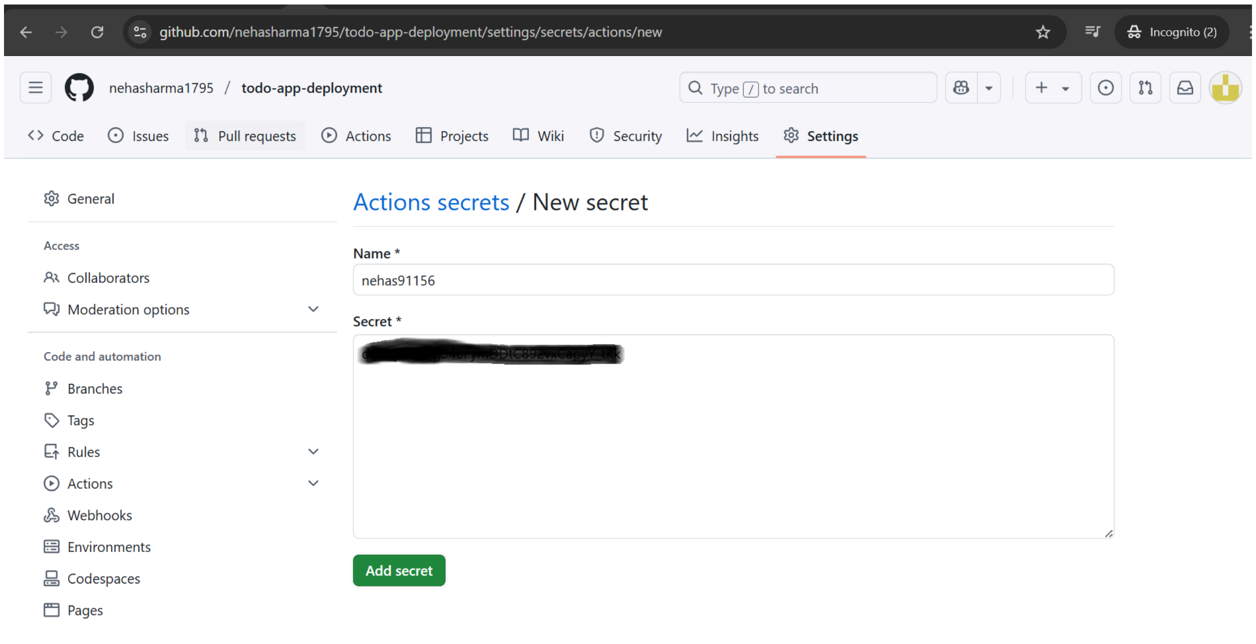
2. At the password prompt, enter the personal access token.

[redacted]

```
docker login -u nehas91156
dckr_pat_vFg340FymDDtC892vxCapyY_tKk
```

Step 2: Add Secrets to GitHub

1) Open your GitHub repository.
2) Go to Settings → Secrets and variables → Actions.
3) Click New repository secret and add the following:
   Name: **DOCKER_USERNAME**
4) Value: Your Docker Hub username
   Name: **DOCKER_PASSWORD**
5) Value: Paste the Docker Hub access token you copied earlier
   Click Add secret.



github.com/nehasharma1795/todo-app-deployment/settings/secrets/actions/new

nehasharma1795 / todo-app-deployment

Actions secrets / New secret

Name *

nehas91156

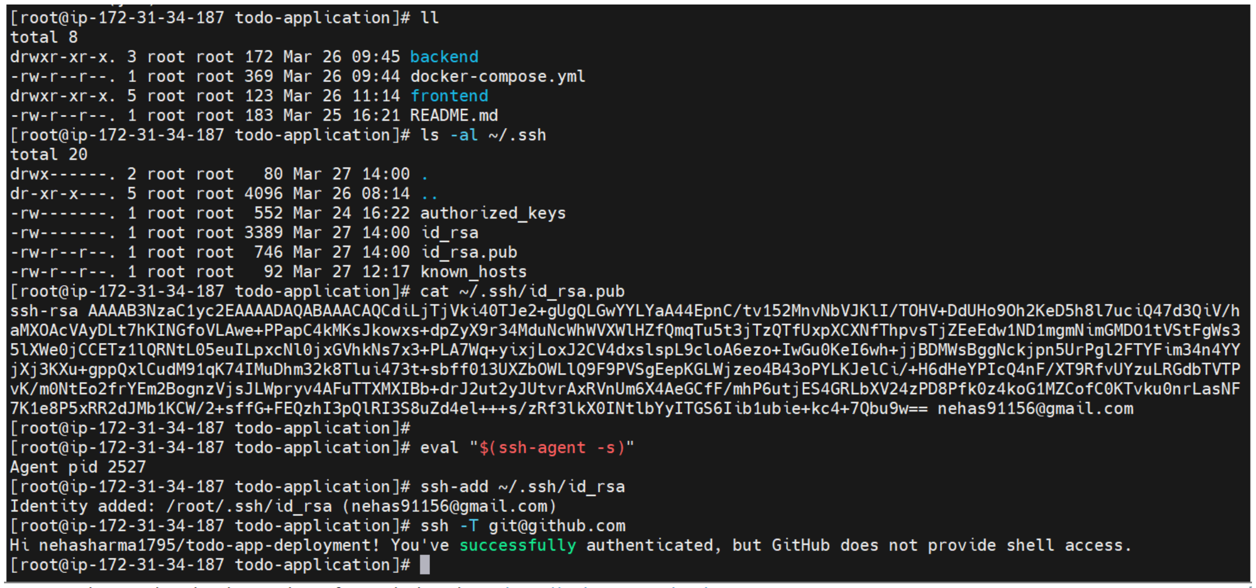
Secret *

[redacted]

Why Use an Access Token Instead of a Password?

- **More Secure**: You can revoke it anytime without changing your password.
- **Limited Scope**: Can be set to read/write only.
- **Better Practice**: Works better in CI/CD pipelines.

Now, your CI/CD pipeline can push Docker images securely



```
[root@ip-172-31-34-187 todo-application]# ll
total 8
drwxr-xr-x. 3 root root 172 Mar 26 09:45 backend
-rw-r--r--. 1 root root 369 Mar 26 09:44 docker-compose.yml
drwxr-xr-x. 5 root root 123 Mar 26 11:14 frontend
-rw-r--r--. 1 root root 183 Mar 25 16:21 README.md
[root@ip-172-31-34-187 todo-application]# ls -al ~/.ssh
total 20
drwx------. 2 root root   80 Mar 27 14:00 .
dr-xr-x---. 5 root root 4096 Mar 26 08:14 ..
-rw-------. 1 root root  552 Mar 24 16:22 authorized_keys
-rw-------. 1 root root 3389 Mar 27 14:00 id_rsa
-rw-r--r--. 1 root root  746 Mar 27 14:00 id_rsa.pub
-rw-r--r--. 1 root root   92 Mar 27 12:17 known_hosts
[root@ip-172-31-34-187 todo-application]# cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQCdiLjTjVki40TJe2+gUgQLGwYYLYaA44EpnC/tv152MnvNbVJKlI/TOHV+DdUHo9Oh2KeD5h8l7uciQ47d3QiV/h
aMXOAcVAyDLt7hKINGfoVLAwe+PPapC4kMKsJkowxs+dpZyX9r34MduNcWhWVXWlHZfQmqTu5t3jTzQTfUxpXCXNfThpvsTjZEeEdw1ND1mgmNimGMDO1tVStFgWs3
5lXWe0jCCETz1lQRNtL05euILpxcNl0jxGVhkNs7x3+PLA7Wq+yixjLoxJ2CV4dxslspL9cloA6ezo+IwGu0KeI6wh+jjBDMWsBggNckjpn5UrPgl2FTYFim34n4YY
jXj3KXu+gppQxlCudM91qK74IMuDhm32k8Tlui473t+sbff013UXZbOWLlQ9F9PVSgEepKGLWjzeo4B43oPYLKJelCi/+H6dHeYPIcQ4nF/XT9RfvUYzuLRGdbTVTP
vK/m0NtEo2frYEm2BognzVjsJLWpryv4AFuTTXMXIBb+drJ2ut2yJUtvrAxRVnUm6X4AeGCfF/mhP6utjES4GRLbXV24zPD8Pfk0z4koG1MZCofC0KTvku0nrLasNF
7K1e8P5xRR2dJMb1KCW/2+sffG+FEQzhI3pQlRI3S8uZd4el+++s/zRf3lkX0INtlbYyITGS6Iib1ubie+kc4+7Qbu9w== nehas91156@gmail.com
[root@ip-172-31-34-187 todo-application]#
[root@ip-172-31-34-187 todo-application]# eval "$(ssh-agent -s)"
Agent pid 2527
[root@ip-172-31-34-187 todo-application]# ssh-add ~/.ssh/id_rsa
Identity added: /root/.ssh/id_rsa (nehas91156@gmail.com)
[root@ip-172-31-34-187 todo-application]# ssh -T git@github.com
Hi nehasharma1795/todo-app-deployment! You've successfully authenticated, but GitHub does not provide shell access.
[root@ip-172-31-34-187 todo-application]#
```

```
cd /path/to/your/project  # Navigate to your project directory
git init  # Initialize Git if not already a repo
git remote add origin https://github.com/your-github-username/your-repository.git
git add .github/workflows/frontend.yml .github/workflows/backend.yml
git commit -m "Added CI/CD workflows for frontend and backend"

git pull origin main --rebase
git add .
git rebase --continue
git push origin main --force
```

Step 1: Generate a New SSH Key

```
ssh-keygen -t rsa -b 4096 -C "your-email@example.com"
```

When prompted to enter a file to save the key, press Enter (it will save as ~/.ssh/id_rsa).
When prompted for a passphrase, press Enter to leave it empty.

Step 2: Add SSH Key to GitHub

```
cat ~/.ssh/id_rsa.pub
```

Copy the full key output.

Now, go to GitHub → Settings → SSH Keys:
Open GitHub SSH Key Settings
Click "New SSH Key"
Paste the copied key and click "Add SSH Key"

Step 3: Add SSH Key to SSH Agent

Run the following commands:

```
eval "$(ssh-agent -s)"
ssh-add ~/.ssh/id_rsa
```

Step 4: Test the SSH Connection

```
ssh -T git@github.com
```

If successful, you should see:

```
Hi nehas91156! You've successfully authenticated, but GitHub does not provide shell access.
```

Step 5: Set Git to Use SSH

If your Git remote is still using HTTPS, change it to SSH:

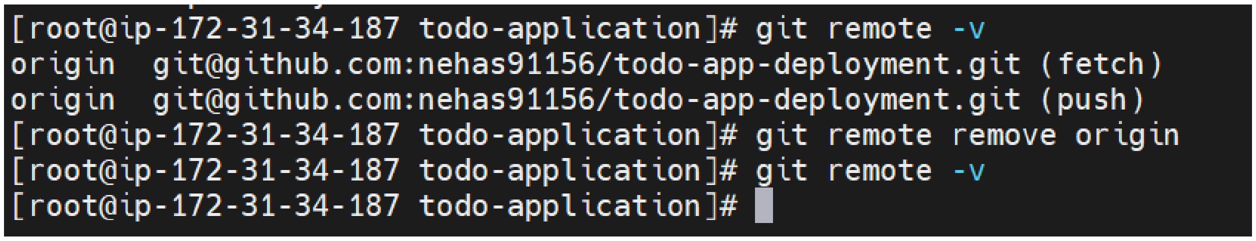
```
git remote set-url origin git@github.com:nehas91156/todo-app-deployment.git
```

Now, try pushing your code:

```
git push origin main
```

OR

```
git push origin main --force
```



```
[root@ip-172-31-34-187 todo-application]# git remote -v
origin  git@github.com:nehas91156/todo-app-deployment.git (fetch)
origin  git@github.com:nehas91156/todo-app-deployment.git (push)
[root@ip-172-31-34-187 todo-application]# git remote remove origin
[root@ip-172-31-34-187 todo-application]# git remote -v
[root@ip-172-31-34-187 todo-application]#
```

```
git remote remove origin
git remote -v
git init
git remote add origin git@github.com:your-username/your-repo.git
```

```
***************************************************************************
    -  Create image repo in docker hub first
    -
```

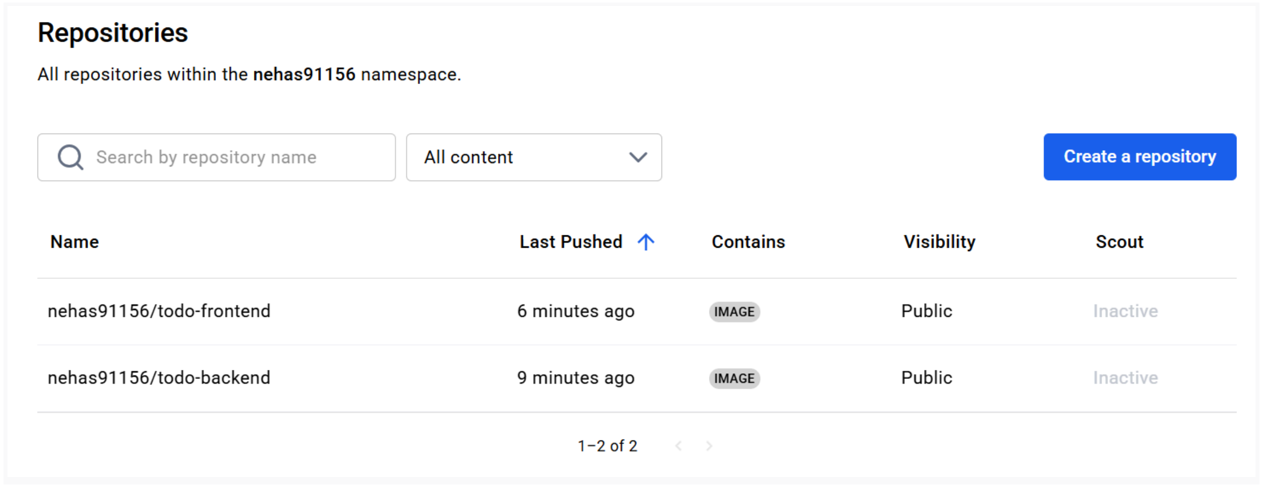
Check Docker Hub Repository Existence

Make sure the repository nehas91156/todo-backend exists on Docker Hub. If it doesn't exist yet, Docker will not be able to push the image. To create a new repository on Docker Hub:

- Go to Docker Hub.
- Log in with your Docker credentials.
- Create a new repository by clicking **Create Repository**.

  Name it todo-backend (or a name matching the image you're trying to push).
  Make it either public or private, depending on your preference.

Once the repository is created, you should be able to push the image.

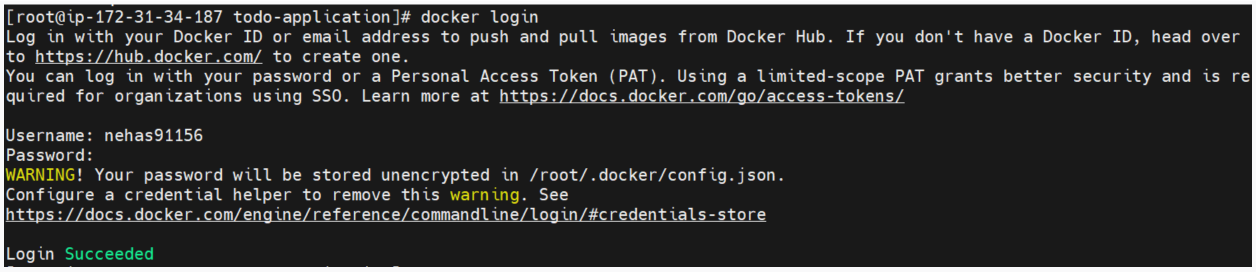


Repositories

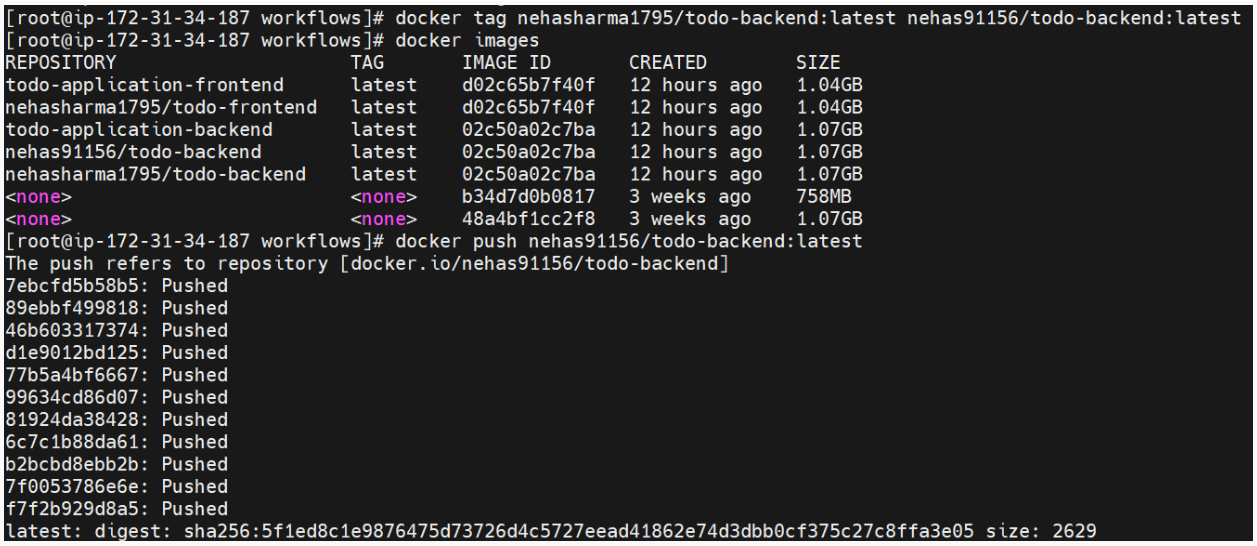
All repositories within the **nehas91156** namespace.

Name	Last Pushed	Contains	Visibility	Scout
nehas91156/todo-frontend	6 minutes ago	IMAGE	Public	Inactive
nehas91156/todo-backend	9 minutes ago	IMAGE	Public	Inactive

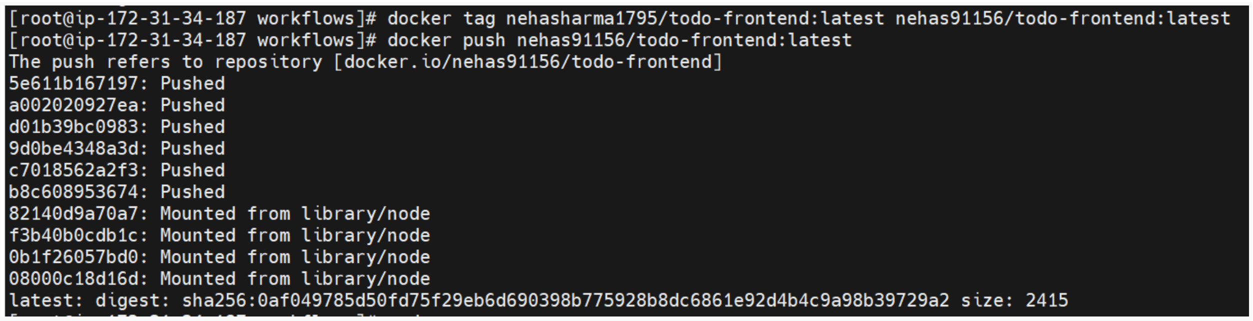
1–2 of 2



```
[root@ip-172-31-34-187 todo-application]# docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over
to https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is re
quired for organizations using SSO. Learn more at https://docs.docker.com/go/access-tokens/

Username: nehas91156
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```



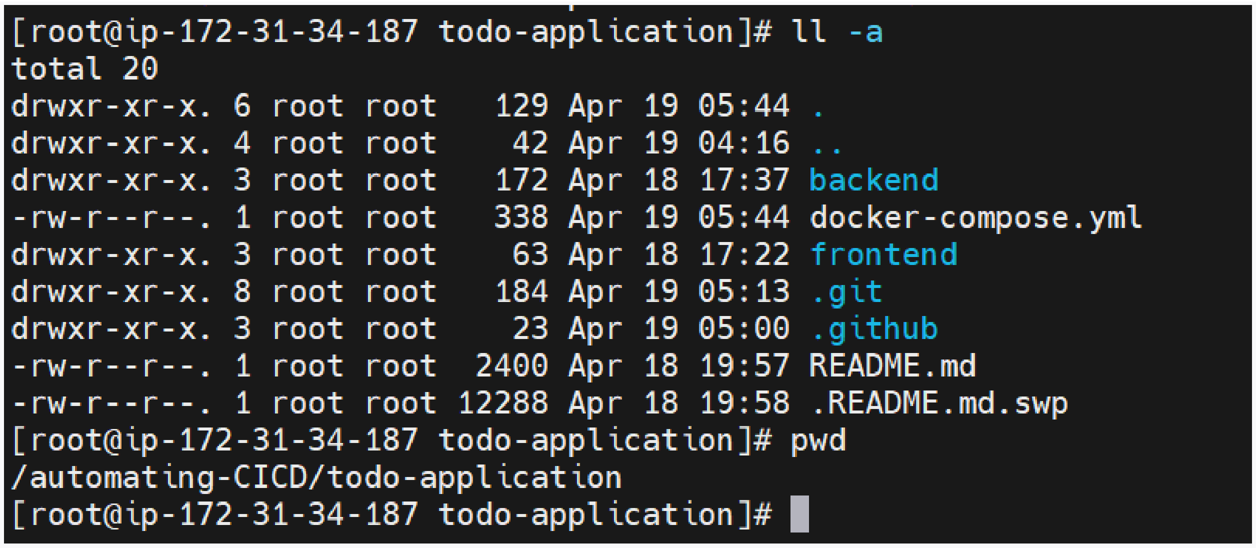
```
[root@ip-172-31-34-187 workflows]# docker tag nehasharma1795/todo-backend:latest nehas91156/todo-backend:latest
[root@ip-172-31-34-187 workflows]# docker images
REPOSITORY                    TAG       IMAGE ID       CREATED        SIZE
todo-application-frontend     latest    d02c65b7f40f   12 hours ago   1.04GB
nehasharma1795/todo-frontend  latest    d02c65b7f40f   12 hours ago   1.04GB
todo-application-backend      latest    02c50a02c7ba   12 hours ago   1.07GB
nehas91156/todo-backend       latest    02c50a02c7ba   12 hours ago   1.07GB
nehasharma1795/todo-backend   latest    02c50a02c7ba   12 hours ago   1.07GB
<none>                        <none>    b34d7d0b0817   3 weeks ago    758MB
<none>                        <none>    48a4bf1cc2f8   3 weeks ago    1.07GB
[root@ip-172-31-34-187 workflows]# docker push nehas91156/todo-backend:latest
The push refers to repository [docker.io/nehas91156/todo-backend]
7ebcfd5b58b5: Pushed
89ebbf499818: Pushed
46b603317374: Pushed
d1e9012bd125: Pushed
77b5a4bf6667: Pushed
99634cd86d07: Pushed
81924da38428: Pushed
6c7c1b88da61: Pushed
b2bcbd8ebb2b: Pushed
7f0053786e6e: Pushed
f7f2b929d8a5: Pushed
latest: digest: sha256:5f1ed8c1e9876475d73726d4c5727eead41862e74d3dbb0cf375c27c8ffa3e05 size: 2629
```



```
[root@ip-172-31-34-187 workflows]# docker tag nehasharma1795/todo-frontend:latest nehas91156/todo-frontend:latest
[root@ip-172-31-34-187 workflows]# docker push nehas91156/todo-frontend:latest
The push refers to repository [docker.io/nehas91156/todo-frontend]
5e611b167197: Pushed
a002020927ea: Pushed
d01b39bc0983: Pushed
9d0be4348a3d: Pushed
c7018562a2f3: Pushed
b8c608953674: Pushed
82140d9a70a7: Mounted from library/node
f3b40b0cdb1c: Mounted from library/node
0b1f26057bd0: Mounted from library/node
08000c18d16d: Mounted from library/node
latest: digest: sha256:0af049785d50fd75f29eb6d690398b775928b8dc6861e92d4b4c9a98b39729a2 size: 2415
```

****

Now you have all the automating CICD files in the below directory



```
[root@ip-172-31-34-187 todo-application]# ll -a
total 20
drwxr-xr-x. 6 root root   129 Apr 19 05:44 .
drwxr-xr-x. 4 root root    42 Apr 19 04:16 ..
drwxr-xr-x. 3 root root   172 Apr 18 17:37 backend
-rw-r--r--. 1 root root   338 Apr 19 05:44 docker-compose.yml
drwxr-xr-x. 3 root root    63 Apr 18 17:22 frontend
drwxr-xr-x. 8 root root   184 Apr 19 05:13 .git
drwxr-xr-x. 3 root root    23 Apr 19 05:00 .github
-rw-r--r--. 1 root root  2400 Apr 18 19:57 README.md
-rw-r--r--. 1 root root 12288 Apr 18 19:58 .README.md.swp
[root@ip-172-31-34-187 todo-application]# pwd
/automating-CICD/todo-application
[root@ip-172-31-34-187 todo-application]#
```

Docker compose file

```
version: '3.8'

services:
  backend:
    build:
      context: ./backend
      dockerfile: Dockerfile
    image: nehas91156/todo-backend:latest
    ports:
      - "8000:8000"
  frontend:
    build:
      context: ./frontend/todo
      dockerfile: Dockerfile
    image: nehas91156/todo-frontend:latest
    ports:
      - "3000:3000"
```

For CICD pipeline,
You need to create a deploy.yaml file inside .github directory

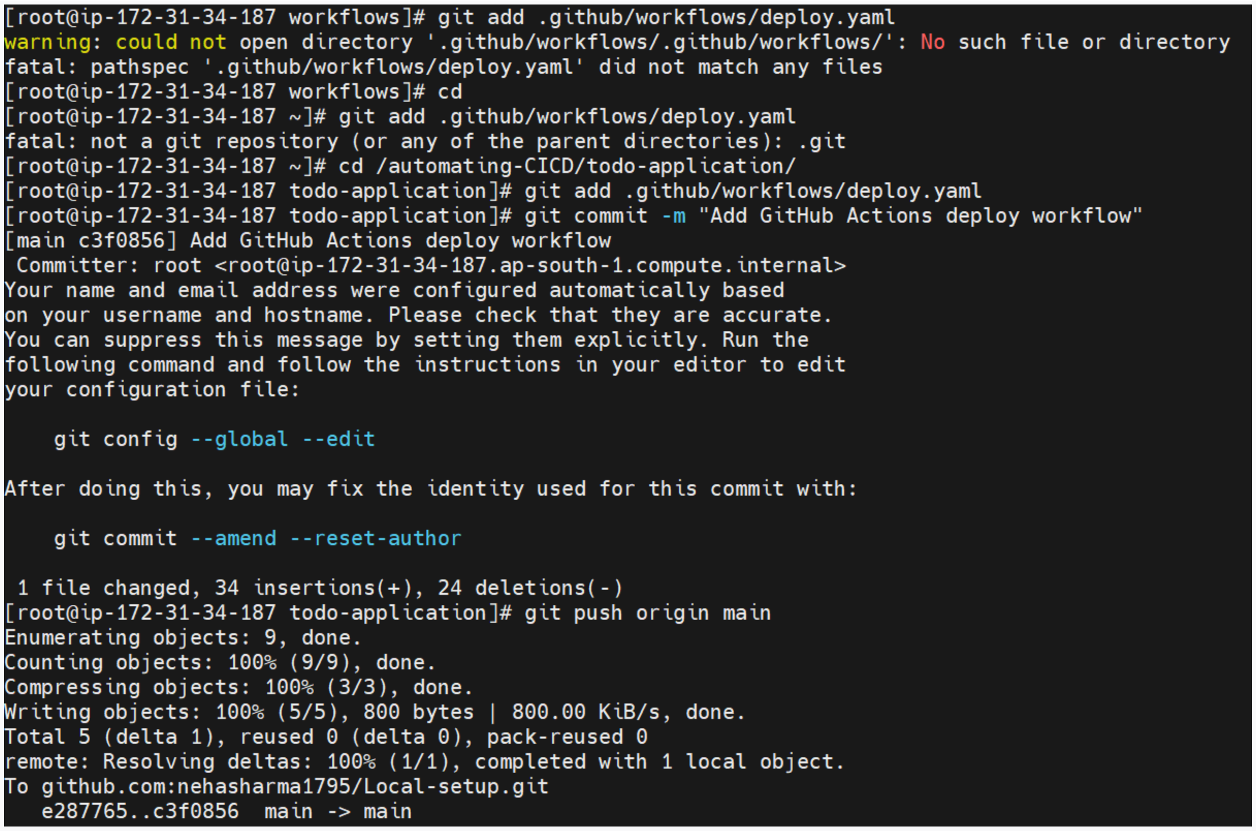
```
todo-application/
├── backend/               # FastAPI backend
│   └── Dockerfile
├── frontend/              # React app
│   └── todo/
│       └── Dockerfile
├── docker-compose.yml     # Service composition
└── .github/workflows/
    └── deploy.yml         # GitHub Actions workflow
```

Deploy.yml

```
name: CI/CD Pipeline for Todo App

on:
  push:
    branches: [main]

env:
  BACKEND_IMAGE: nehas91156/todo-backend
  FRONTEND_IMAGE: nehas91156/todo-frontend
```

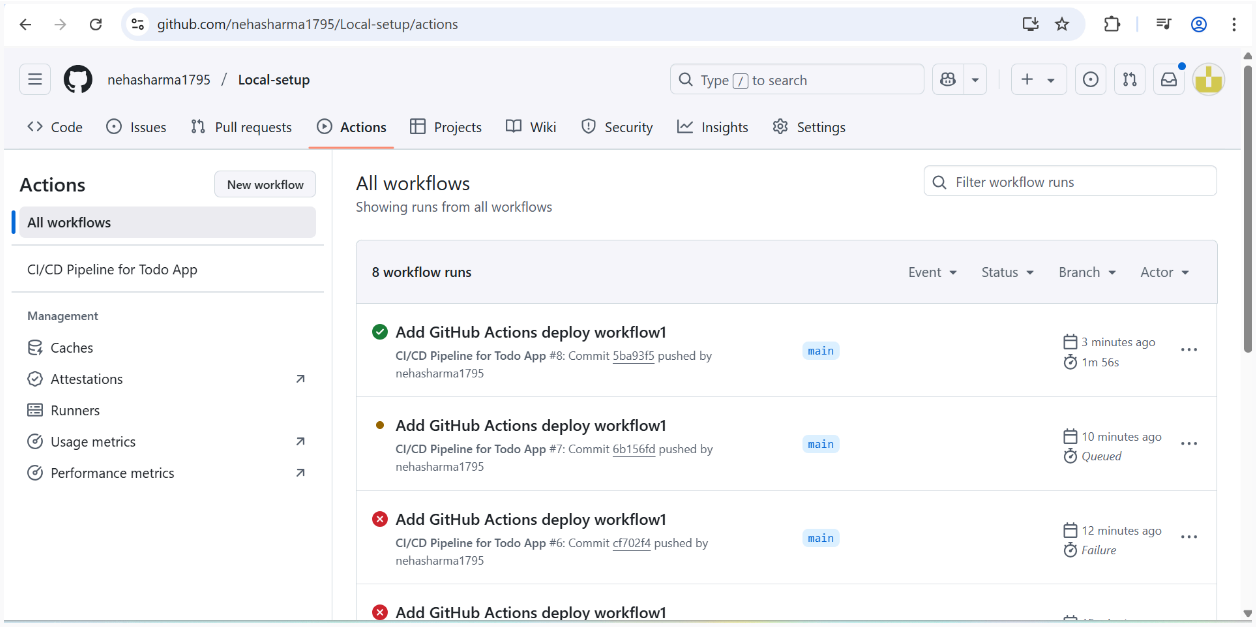
```
jobs:
  build-and-push:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout Code
        uses: actions/checkout@v3

      - name: Set up Docker Buildx
        uses: docker/setup-buildx-action@v3

      - name: Log in to Docker Hub
        uses: docker/login-action@v3
        with:
          username: ${{ secrets.DOCKER_USERNAME }}
          password: ${{ secrets.DOCKER_PASSWORD }}

      - name: Build and Push Backend Image
        uses: docker/build-push-action@v5
        with:
          context: ./backend
          file: ./backend/Dockerfile
          push: true
          tags: ${{ env.BACKEND_IMAGE }}:latest

      - name: Build and Push Frontend Image
        uses: docker/build-push-action@v5
        with:
          context: ./frontend/todo
          file: ./frontend/todo/Dockerfile
          push: true
          tags: ${{ env.FRONTEND_IMAGE }}:latest
```

If you have modified some files, then execute the below Git commands and push them to the GitHub repository.



```
[root@ip-172-31-34-187 workflows]# git add .github/workflows/deploy.yaml
warning: could not open directory '.github/workflows/.github/workflows/': No such file or directory
fatal: pathspec '.github/workflows/deploy.yaml' did not match any files
[root@ip-172-31-34-187 workflows]# cd
[root@ip-172-31-34-187 ~]# git add .github/workflows/deploy.yaml
fatal: not a git repository (or any of the parent directories): .git
[root@ip-172-31-34-187 ~]# cd /automating-CICD/todo-application/
[root@ip-172-31-34-187 todo-application]# git add .github/workflows/deploy.yaml
[root@ip-172-31-34-187 todo-application]# git commit -m "Add GitHub Actions deploy workflow"
[main c3f0856] Add GitHub Actions deploy workflow
 Committer: root <root@ip-172-31-34-187.ap-south-1.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 34 insertions(+), 24 deletions(-)
[root@ip-172-31-34-187 todo-application]# git push origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 800 bytes | 800.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:nehasharma1795/Local-setup.git
   e287765..c3f0856  main -> main
```

```
git add .
git commit -m "Add GitHub Actions deploy workflow"
git push origin main
```



github.com/nehasharma1795/Local-setup/actions

nehasharma1795 / Local-setup

Code | Issues | Pull requests | Actions | Projects | Wiki | Security | Insights | Settings

Actions — New workflow

All workflows

CI/CD Pipeline for Todo App

Management

Caches

Attestations

Runners

Usage metrics

Performance metrics

All workflows

Showing runs from all workflows

Filter workflow runs

8 workflow runs — Event | Status | Branch | Actor

Add GitHub Actions deploy workflow1
CI/CD Pipeline for Todo App #8: Commit 5ba93f5 pushed by nehasharma1795 — main — 3 minutes ago — 1m 56s

Add GitHub Actions deploy workflow1
CI/CD Pipeline for Todo App #7: Commit 6b156fd pushed by nehasharma1795 — main — 10 minutes ago — Queued

Add GitHub Actions deploy workflow1
CI/CD Pipeline for Todo App #6: Commit cf702f4 pushed by nehasharma1795 — main — 12 minutes ago — Failure

Add GitHub Actions deploy workflow1