

Lab 6 - MLOps CI Pipeline/Workflow with GitHub Actions

Create a Github Workflow for Streamlit

To understand how GitHub workflow works, let's create a simple pipeline for Streamlit app which will

- Trigger when code inside `streamlit_app/` path changes or optionally be triggered manually.
- Build a container image by reading `streamlit_app/Dockerfile`
- Login and publish the image to DockerHub Container Registry

```
cd house-price-predictor
mkdir -p .github/workflows
```

File: `.github/workflows/streamlit-ci.yaml`

```
name: Streamlit CI

on:
  push:
    paths:
      - 'streamlit_app/**'
  workflow_dispatch:

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

    steps:
      - name: Checkout code
        uses: actions/checkout@v4

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

      - name: Log in to DockerHub Container Registry
```

```
uses: docker/login-action@v3
with:
  registry: docker.io
  username: ${vars.DOCKERHUB_USERNAME}
  password: ${secrets.DOCKERHUB_TOKEN}

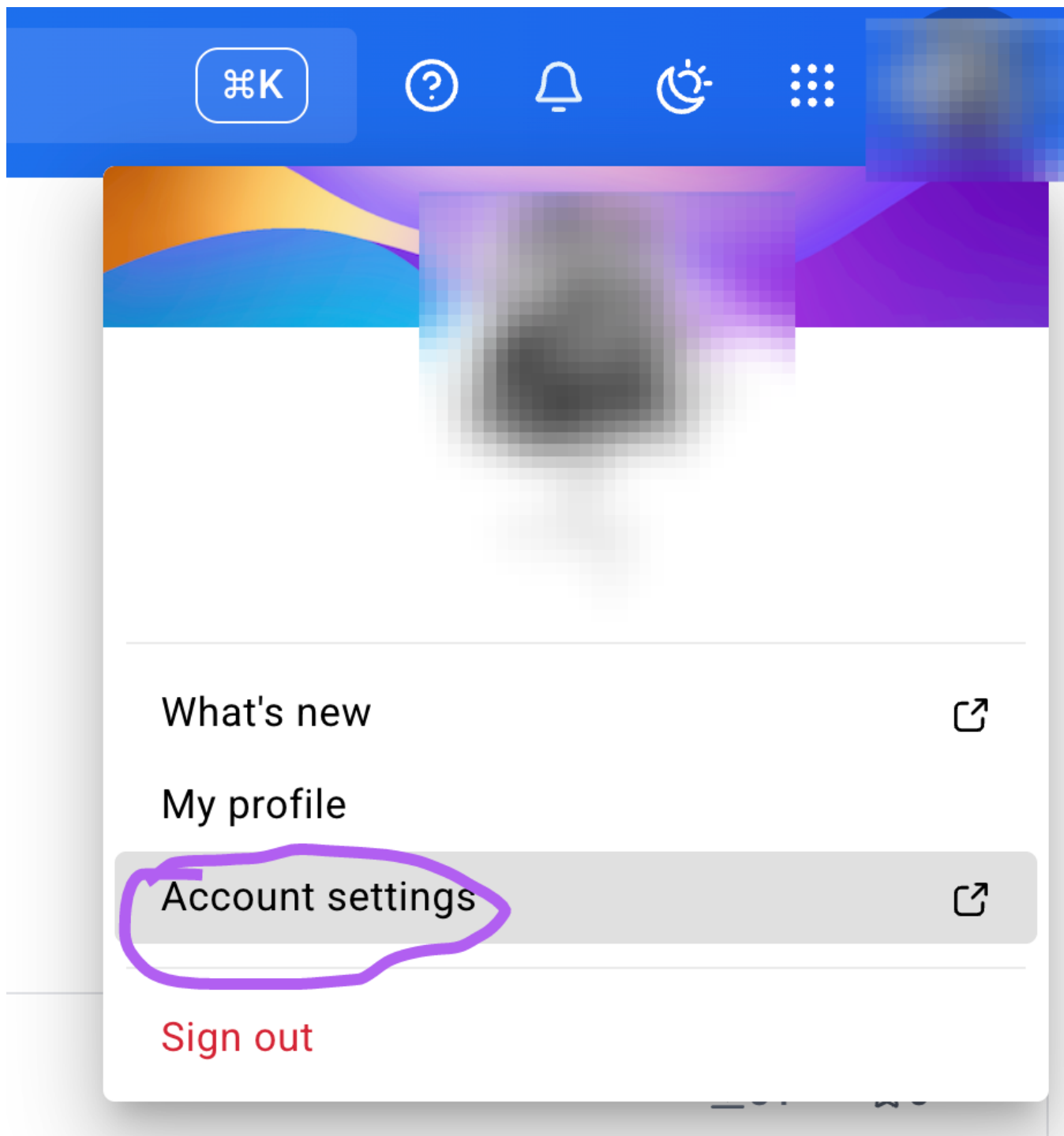
- name: Build and push Docker image
  uses: docker/build-push-action@v5
  with:
    context: ./streamlit_app
    push: true
    tags: docker.io/${vars.DOCKERHUB_USERNAME}/streamlit:latest
```

```
git add .github/workflows/streamlit-ci.yaml
git commit -am "adding CI workflow for streamlit"
git push origin main
```

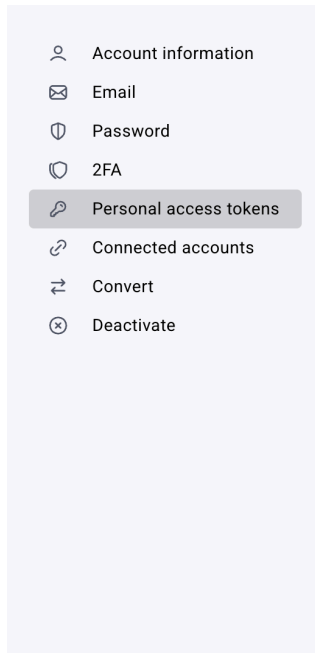
Once committed and pushed you are ready to trigger the workflow/pipeline from GitHub → Actions. However, you still need to add the authentication required to connect to DockerHub to publish the image with. Lets add that first.

Setting up Registry Credentials for Pipeline Builds

Login to Dockerhub and brose to Account Settings → Personal Access Tokens → Generate Token. You could also visit the page directly at <https://app.docker.com/settings/personal-access-tokens/create>



Generate the token with `Read & Write` access, copy it over.



[Personal access tokens](#) / New access token

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
GithubActions

Expiration date
None

Optional

Access permissions
Read & Write

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

Cancel Generate

Now from GitHub Repository Settings, select **Secrets and Variables** → **Actions** → **New Repository Variable**

and add the variable **DOCKERHUB_USERNAME** with your actual docker id.

Repository variables

This repository has no variables.

New repository variable

Actions variables / New variable

Note: Variable values are exposed as plain text. If you need to encrypt and mask sensitive information, [create a secret](#) instead.

Name *



- Variable names may only contain alphanumeric characters ([a-z], [A-Z], [0-9]) or underscores (_).
- Variable names cannot start with a number.
- Variable names cannot start with GITHUB_ prefix.

Value *

Add variable

Note : Make sure its **Repository Variable**.

Also add the repository secret (token copied from DockerHub) which should be set as

DOCKERHUB_TOKEN

Actions secrets and variables

Secrets and variables allow you to manage reusable configuration data. Set sensitive data. [Learn more about encrypted secrets](#). Variables are shown as data. [Learn more about variables](#).

Anyone with collaborator access to this repository can use these secrets as passed to workflows that are triggered by a pull request from a fork.

Secrets

Variables

Environment secrets

This environment has no secrets

Manage environment secrets

Repository secrets

Name 



DOCKERHUB_TOKEN

Once you are done with this, enable and trigger the Streamlit CI pipeline from GitHub → Actions.

Setup MLOps CI Pipeline with GitHub Actions

Create the MLOPs workflow at `.github/workflows/mlops-pipeline.yml`

```
# .github/workflows/mlops-pipeline.yml
name: MLOps Pipeline

on:
  push:
    branches: [ main ]
    tags: [ 'v*.*.*' ]
  pull_request:
    branches: [ main ]

jobs:
  data-processing:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout code
        uses: actions/checkout@v2

      - name: Set up Python
        uses: actions/setup-python@v2
        with:
          python-version: '3.11.9'

      - name: Install dependencies
        run: |
          python -m pip install --upgrade pip
          pip install -r requirements.txt

      - name: Process data
        run: |
          python src/data/run_processing.py --input data/raw/house_data.csv --
output data/processed/cleaned_house_data.csv

      - name: Engineer features
        run: |
          python src/features/engineer.py --input data/processed/
cleaned_house_data.csv --output data/processed/featured_house_data.csv --
preprocessor models/trained/preprocessor.pkl

      - name: Upload processed data
```

```
uses: actions/upload-artifact@v4
with:
  name: processed-data
  path: data/processed/featured_house_data.csv
```

- name: Upload preprocessor
uses: actions/upload-artifact@v4
with:
 name: preprocessor
 path: models/trained/preprocessor.pkl

model-training:

```
needs: data-processing
runs-on: ubuntu-latest
```

steps:

- name: Checkout code
uses: actions/checkout@v2
- name: Set up Python
uses: actions/setup-python@v2
with:
 python-version: '3.11.9'
- name: Install dependencies
run: |
 python -m pip install --upgrade pip
 pip install -r requirements.txt

- name: Download processed data
uses: actions/download-artifact@v4
with:
 name: processed-data
 path: data/processed/

- name: Set up MLflow
run: |
 docker pull ghcr.io/mlflow/mlflow:latest
 docker run -d -p 5000:5000 --name mlflow-server ghcr.io/mlflow/
mlflow:latest mlflow server --host 0.0.0.0 --backend-store-uri sqlite:///mlflow.db

- name: Wait for MLflow to start


```
run: |
  for i in {1..10}; do
    curl -f http://localhost:5000/health || sleep 5;
  done
```

- name: Train model

```
run: |
  mkdir -p models
  python src/models/train_model.py --config configs/model_config.yaml --
data data/processed/featured_house_data.csv --models-dir models --mlflow-
tracking-uri http://localhost:5000
```

- name: Upload trained model

```
uses: actions/upload-artifact@v4
with:
  name: trained-model
  path: models/
```

- name: Clean up MLflow

```
run: |
  docker stop mlflow-server || true
  docker rm mlflow-server || true
```

build-and-publish:

```
needs: model-training
runs-on: ubuntu-latest
```

steps:

- name: Checkout code

```
uses: actions/checkout@v2
```

- name: Download trained model

```
uses: actions/download-artifact@v4
with:
  name: trained-model
  path: models/
```

- name: Download preprocessor

```
uses: actions/download-artifact@v4
with:
  name: preprocessor
  path: models/trained/
```

```

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

- name: Log in to DockerHub Container Registry
  uses: docker/login-action@v3
  with:
    registry: docker.io
    username: ${ vars.DOCKERHUB_USERNAME }
    password: ${ secrets.DOCKERHUB_TOKEN }

- name: Build and push Docker image
  uses: docker/build-push-action@v5
  with:
    context: .
    file: ./Dockerfile
    push: true
    tags: docker.io/${ vars.DOCKERHUB_USERNAME }/house-price-
model:latest

```

Source : [mlops-pipeline](#)

Commit the changes

```

git add .github/workflows/mlops-pipeline.yml
git commit -am "adding github mlops pipeline workflow"
git push origin main

```

Once you push the changes, go to GitHub Repo → Actions.

You will see the complete Pipeline from Data Processing → Model Training → Build and Publish execute, just like any other CI Pipeline, just with a different type of application i.e. a Machine Learning Model wrapped into FastAPI.

Summary

Jobs

- data-processing
- model-training
- build-and-publish

Run details

- Usage
- Workflow file

Manually triggered 3 minutes ago Status: **In progress** Total duration: — Artifacts: **3**

mlops-pipeline.yaml
on: workflow_dispatch

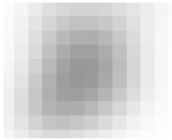


Artifacts

Produced during runtime

Name	Size	Digest
preprocessor	1.63 KB	sha256:0590e3913bc298b7e75c74a91bfa744470e...

You will also see a new image published for the packaged model on the registry.



Gourav Shah [Edit profile](#)

Community User School of Devops India <http://www.schoolofdevops.net>

Repositories

Starred

Contributed

Search by repository name

Displaying 1 to 25 of 56 repositories



initcron/house-price-model

By [initcron](#) • Updated 2 minutes ago

↓ 67 • ☆ 0



initcron/streamlit

By [initcron](#) • Updated 20 days ago

↓ 51 • ☆ 0



initcron/fastapi

By [initcron](#) • Updated 23 days ago

↓ 67 • ☆ 0



initcron/vote

By [initcron](#) • Updated 2 months ago

↓ 2.4K • ☆ 0

Congratulations for setting up a complete MLOps CI Workflow.

#courses/mlops