AI VIETNAM
All-in-One Course
(TA Session)

Continuous Integration Continuous Deployment

Extra Class: MLOps

AI VIET NAM
@aivietnam.edu.vn

Nguyen-Thuan Duong – TA

Kha-Vi Nguyen – sTA

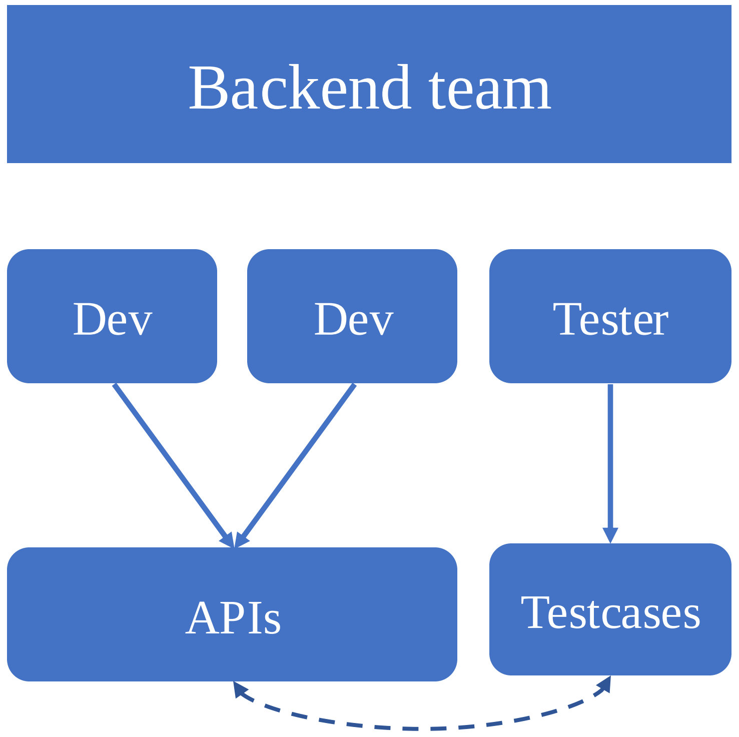
Year 2024

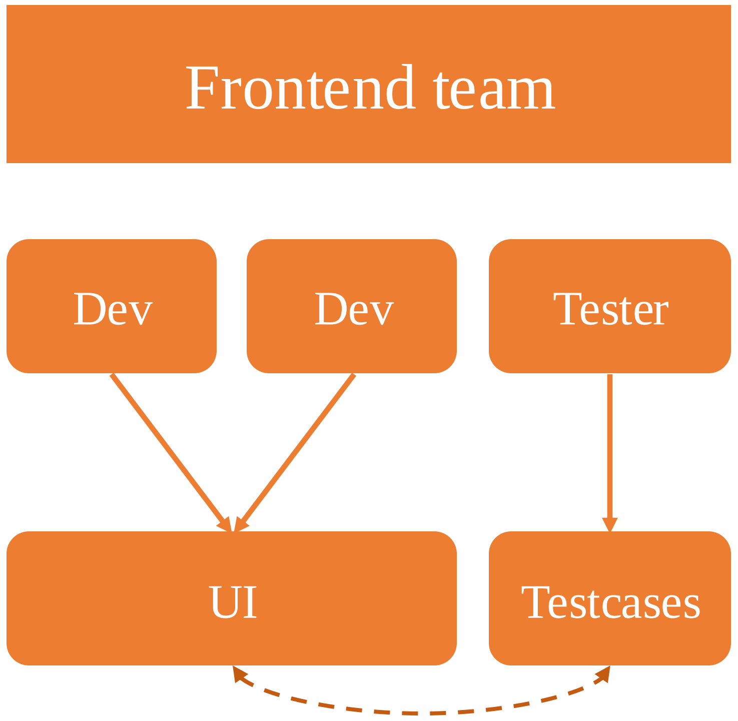
Outline

Introduction

Introduction

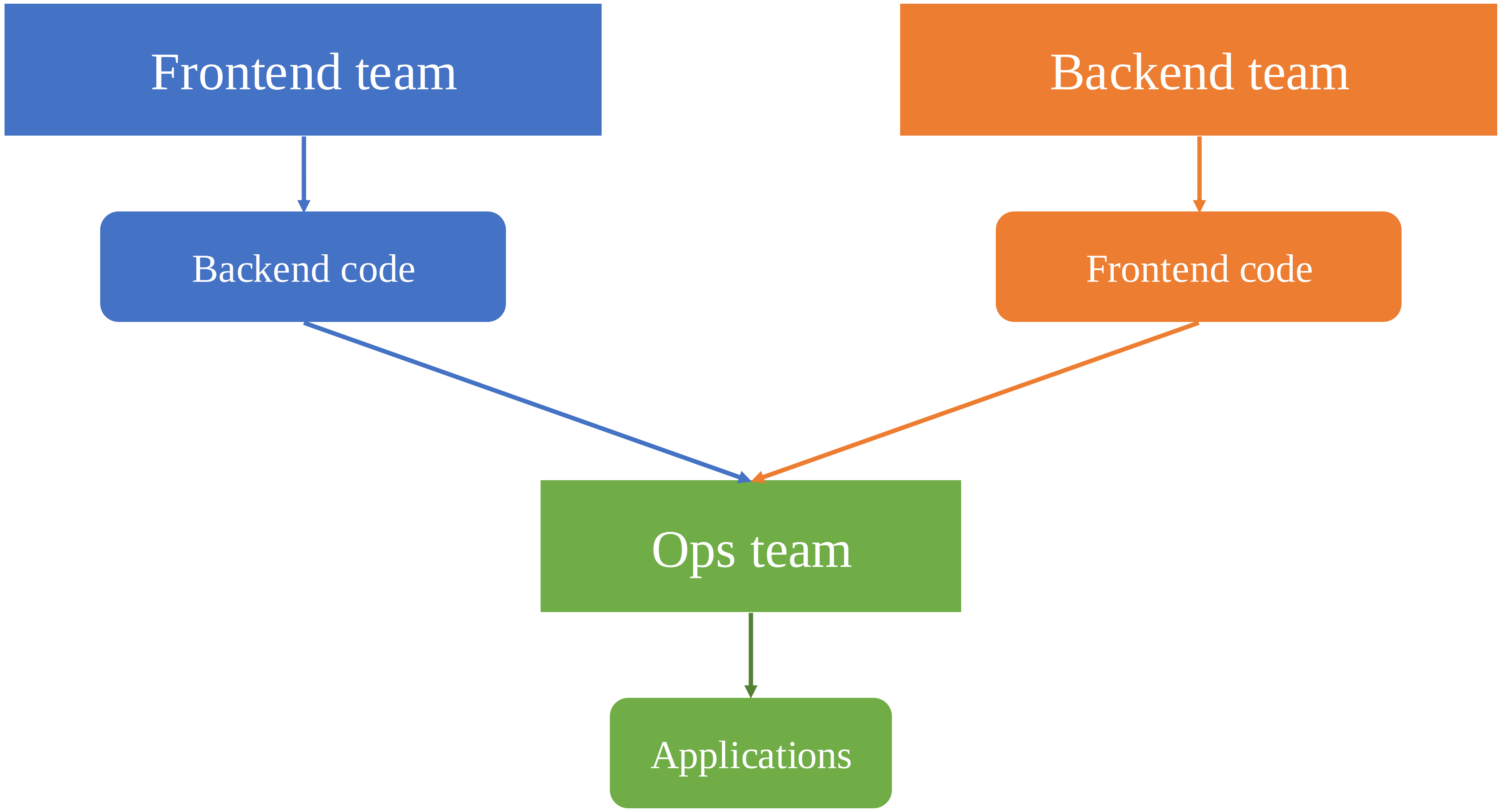
❖ Develop





Introduction

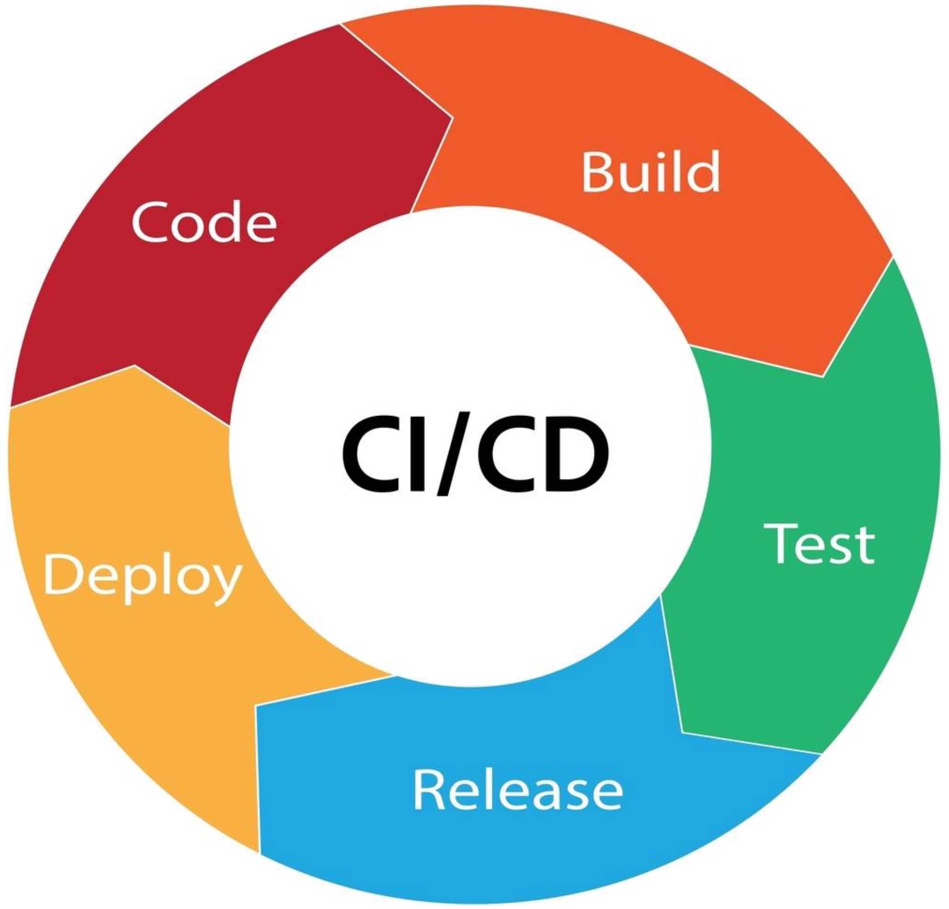
❖ Deployment



Introduction

❖ What is CI/CD?

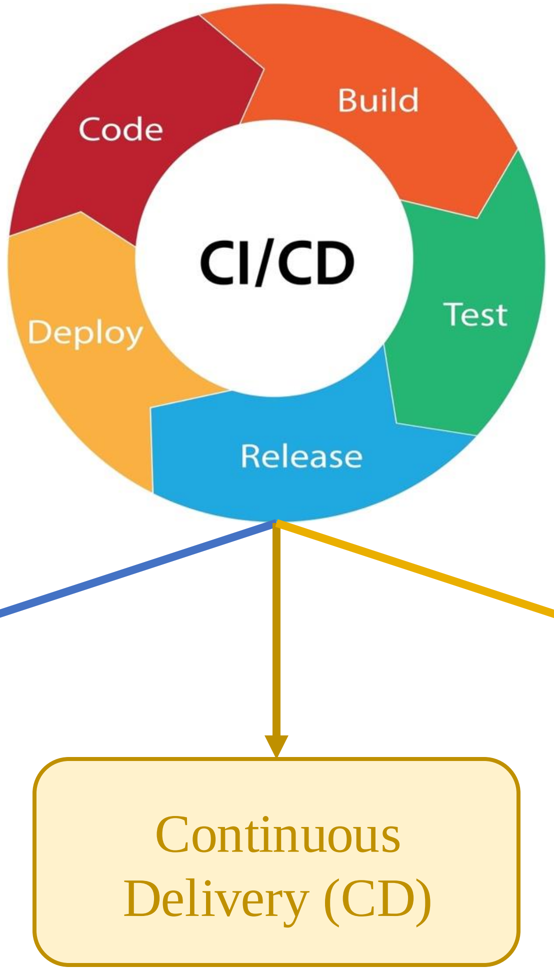
- CI/CD is a methodology to streamline software development by automating integration, testing, and deployment processes.



Why it matters?
- Reduces errors in integration.
- Accelerates time-to-market.
- Enhances collaboration.

Introduction

❖ CI/CD components



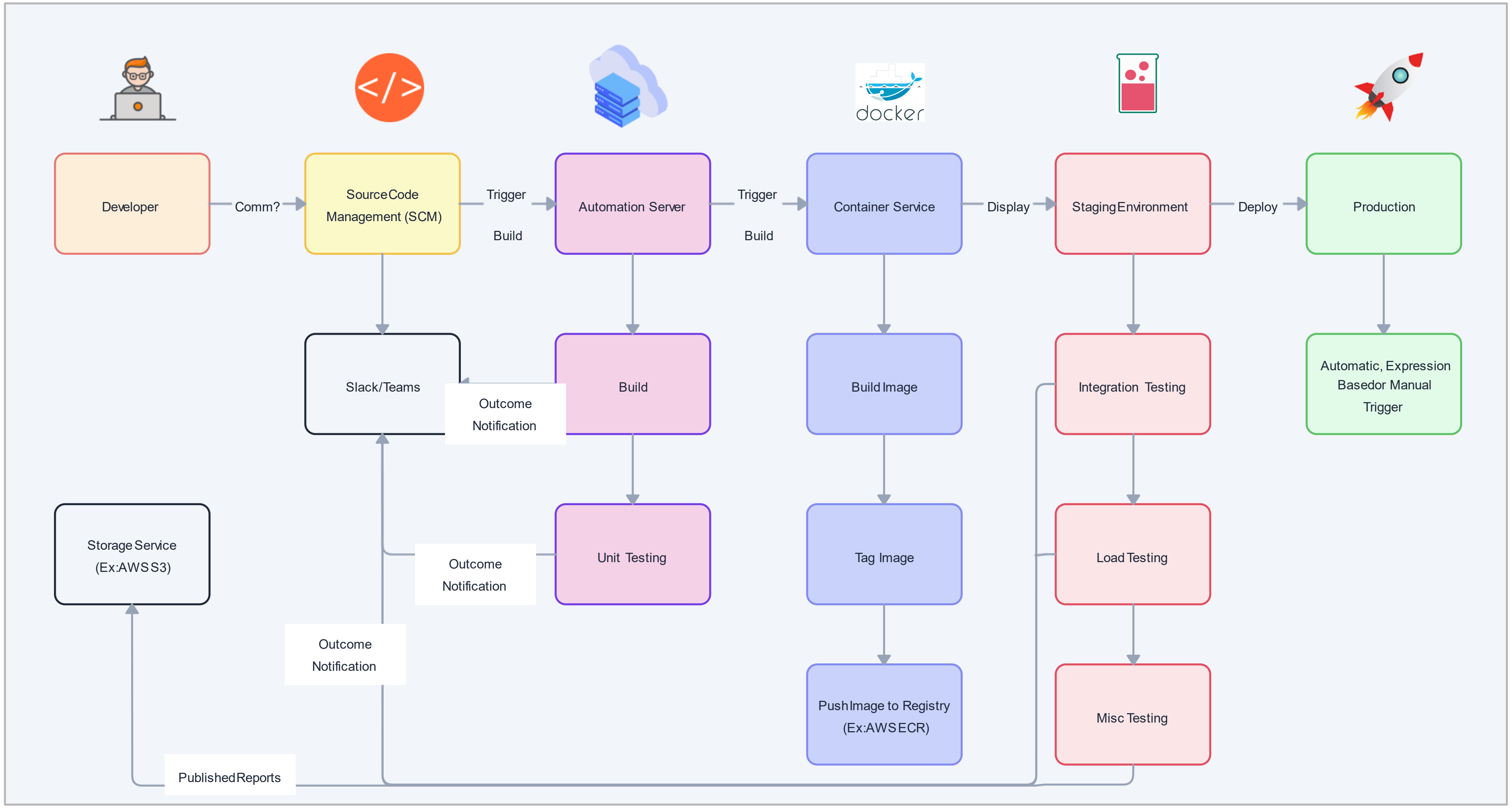
Continuous Integration (CI)

Continuous Delivery (CD)

Continuous Deployment (CD)

Introduction

❖ CI/CD workflow



Introduction

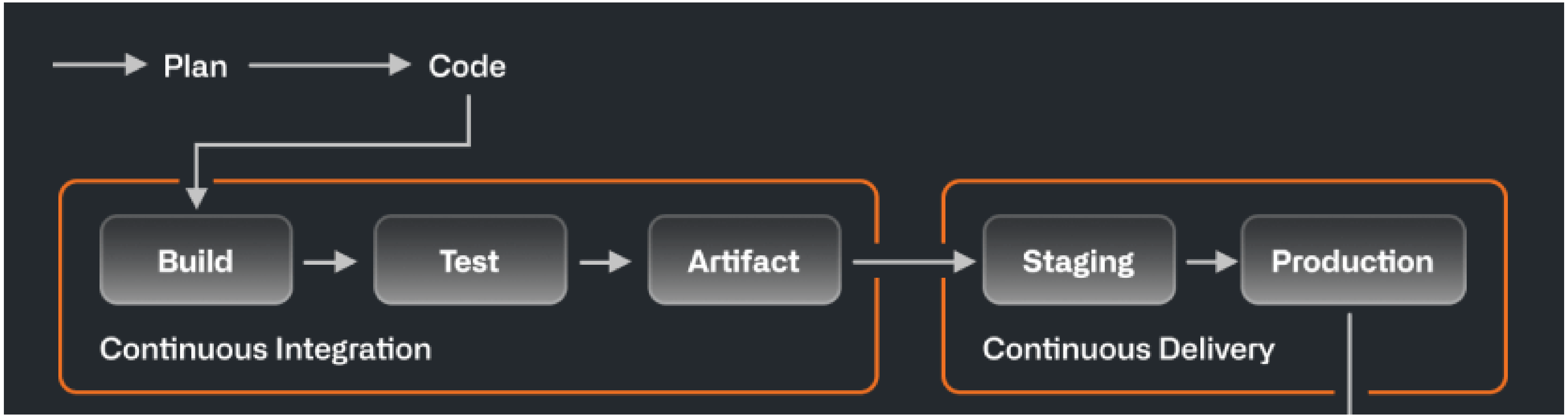
❖ Continuous Integration (CI)

Automates code integration into a shared repository with frequent commits.

- Automated builds and testing
- Version control
- Code review processes

→

- Faster error detection
- Improved team collaboration



Introduction

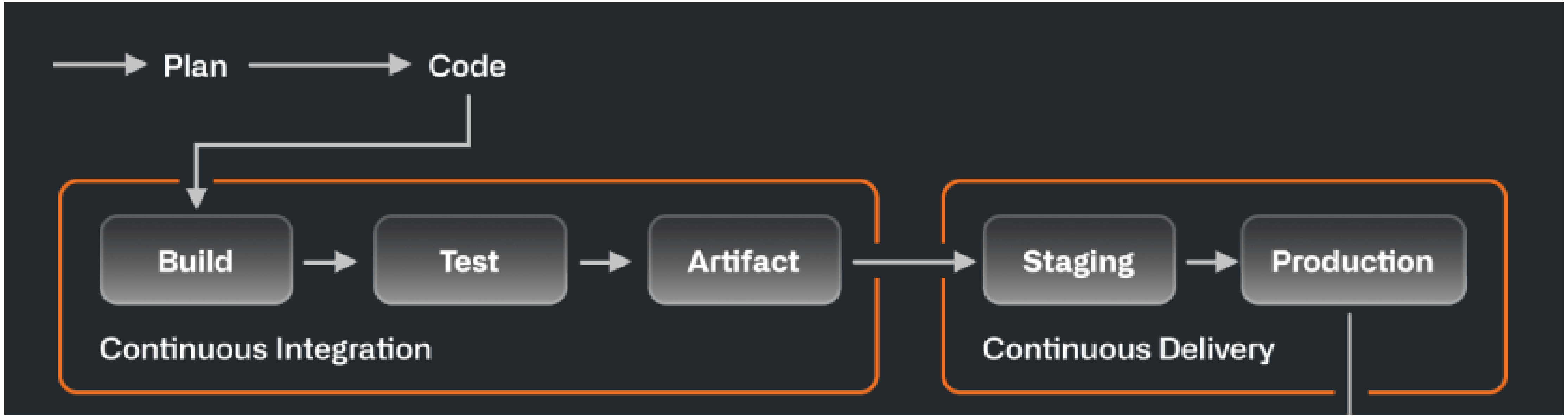
❖ Continuous Delivery (CD)

Extends CI by automating the delivery of code to a staging or pre-production environment

- Automated testing for every change
- Deployment pipelines with checkpoints

→

- Reliable releases
- Minimal manual intervention



Introduction

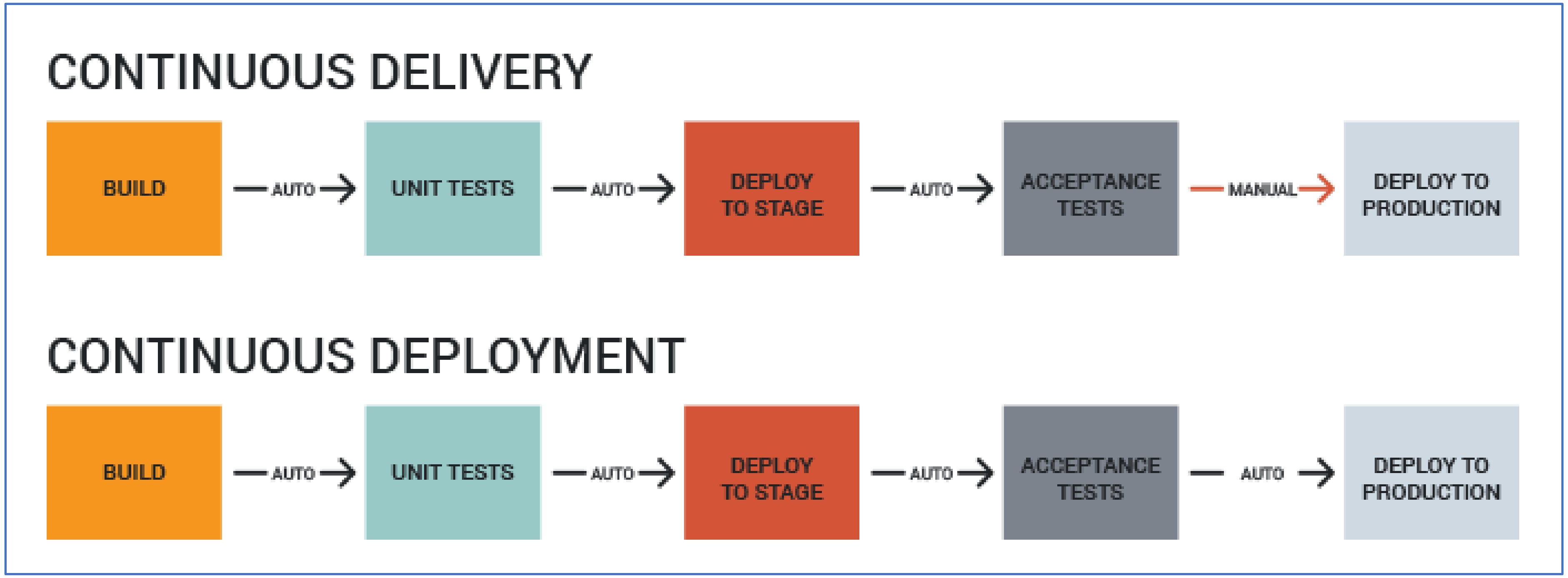
❖ Continuous Deployment (CD)

Automates the release of every validated change directly to production.

- Automated testing and monitoring
- Feature flags for safe deployment

→

- Faster delivery to customers
- Increased feedback loop



Introduction

❖ CI/CD Tools

GitHub Actions

CI/CD

Azure DevOps

mlflow™

circleci

Jenkins

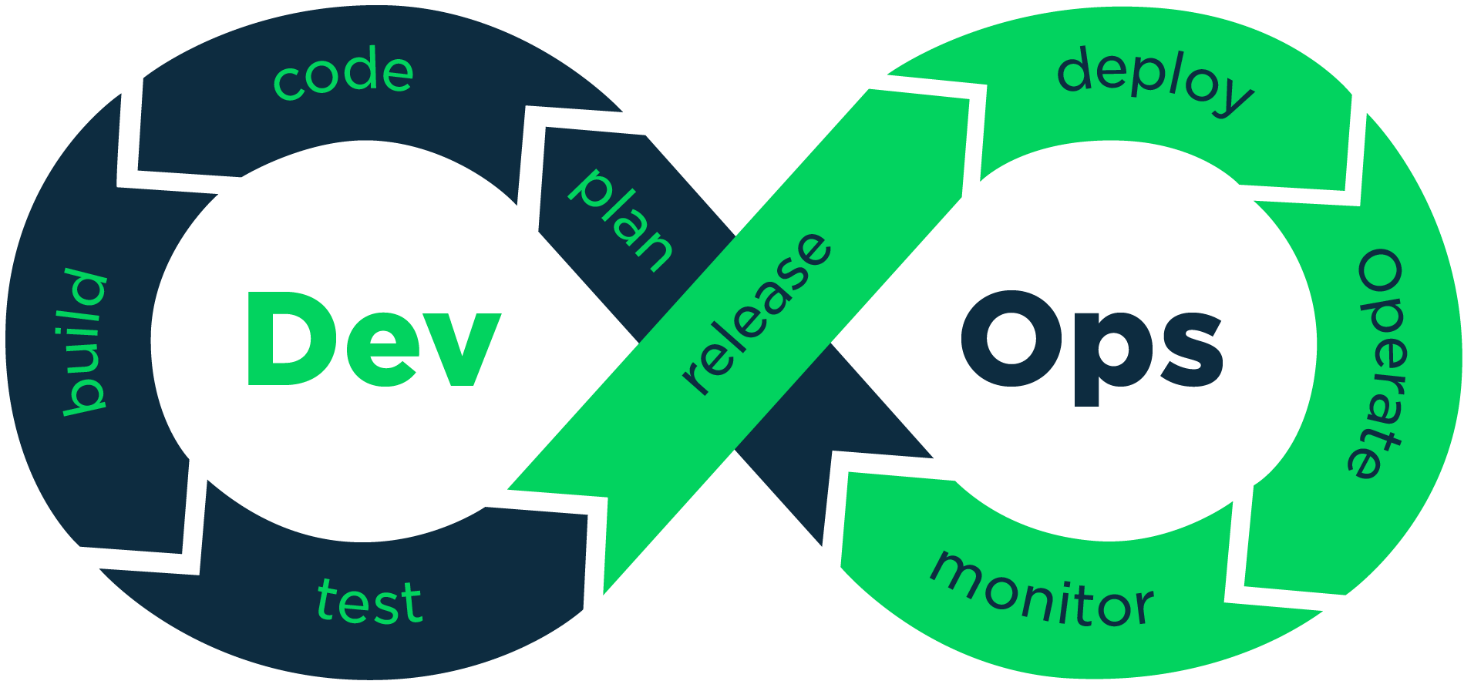
CML

Introduction

❖ DevOps

DevOps is a set of practices, principles, and cultural philosophies.

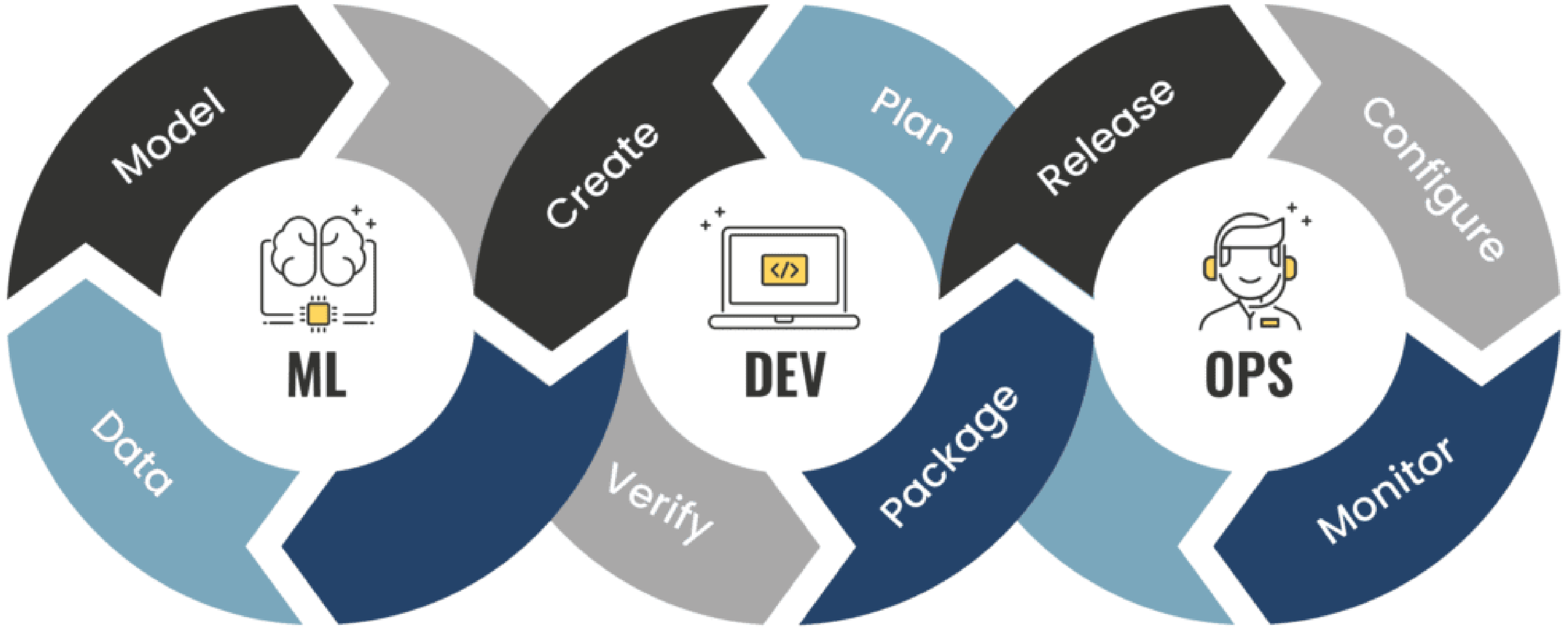
- Aimed at bridging the gap between development (Dev) and operations (Ops) teams
- To improve collaboration, efficiency, and delivery of software applications



Introduction

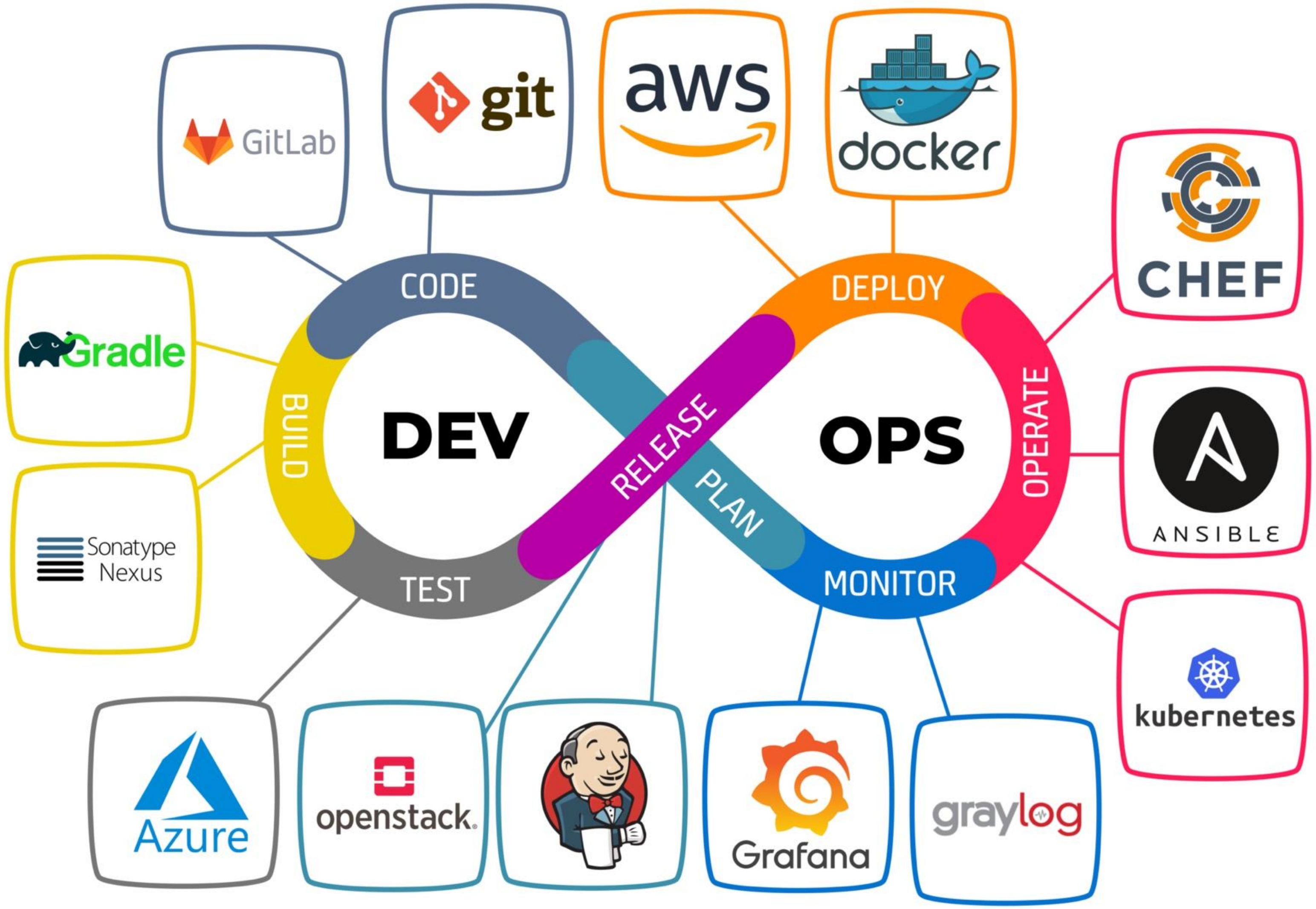
❖ MLOps

Aimed to streamline and automate the development, deployment, and maintenance of machine learning (ML) models.



Introduction

❖ DevOps



Introduction

❖ CI/CD in DevOps vs. MLOps

	DevOps	**MLOps**
Focus	Software, web app	Data pipeline, ML models
Pipeline	Build -> Test -> Deploy	Data -> Train -> Test -> Deploy
Monitoring	App performance	Model performance
Tools	Jenkins, Github Actions, Docker, K8s	Airflow, MLFlow, DVC, CML

CI

CD

CI

CD

CT

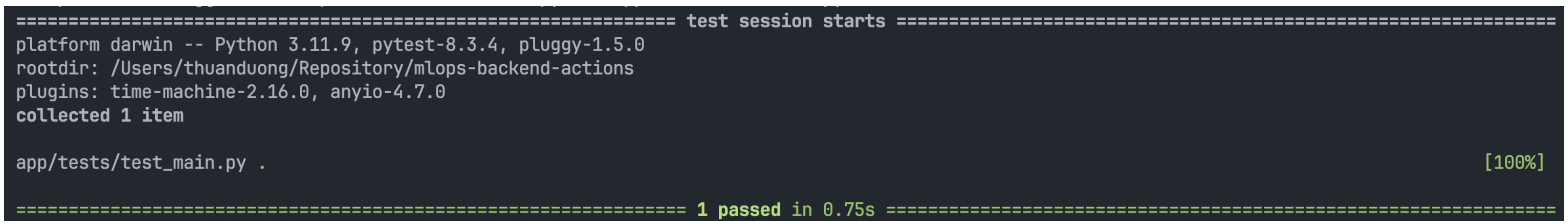
(Continuous Training)

Challenges in MLOps:
- Data drift
- Model retraining
- Monitoring

CI/CD in DevOps

CI/CD in DevOps

❖ Manual testing



```
============================================================== test session starts ==============================================================
platform darwin -- Python 3.11.9, pytest-8.3.4, pluggy-1.5.0
rootdir: /Users/thuanduong/Repository/mlops-backend-actions
plugins: time-machine-2.16.0, anyio-4.7.0
collected 1 item

app/tests/test_main.py .                                                                                                                 [100%]

=============================================================== 1 passed in 0.75s ===============================================================
```

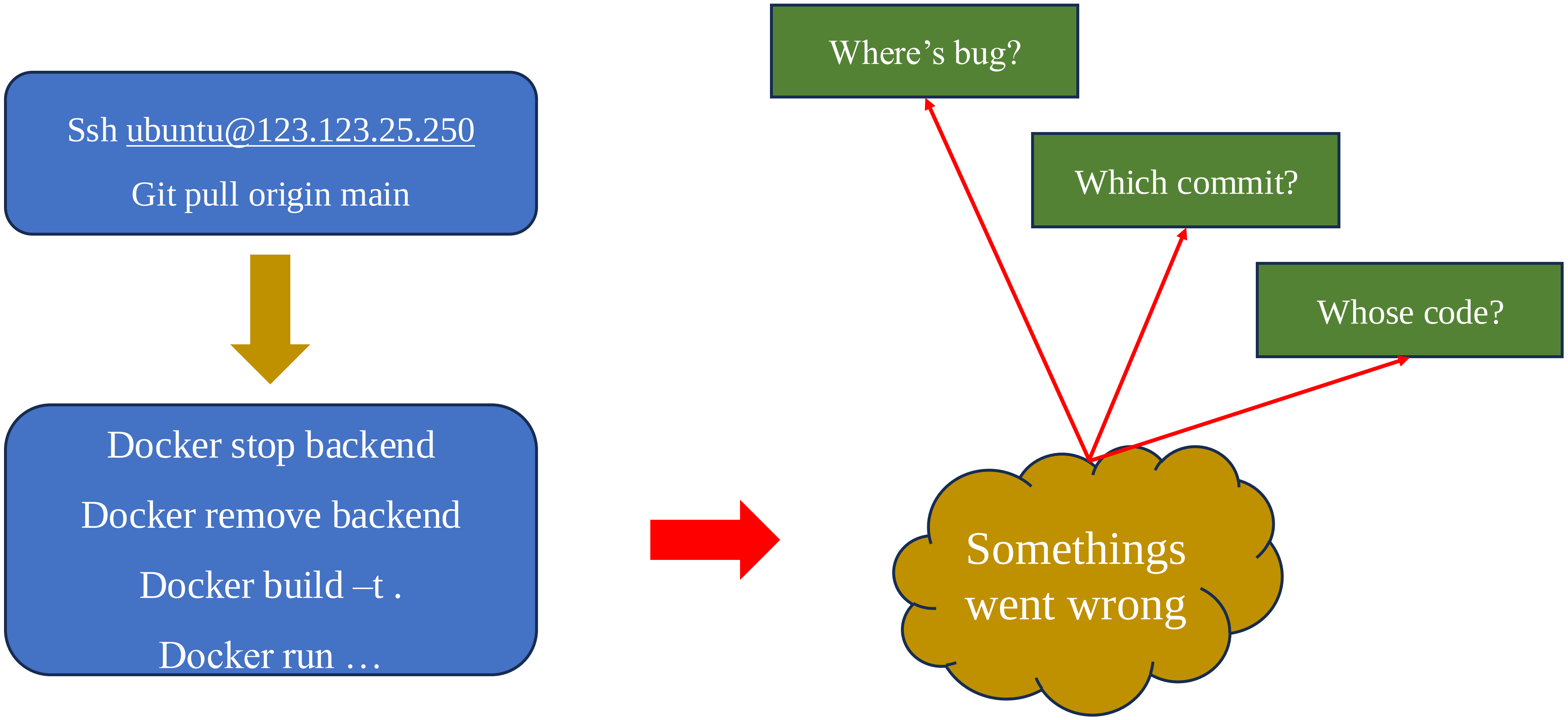
Git add .

Git commit –m “add feat …”

Git push –u origin main

CI/CD in DevOps

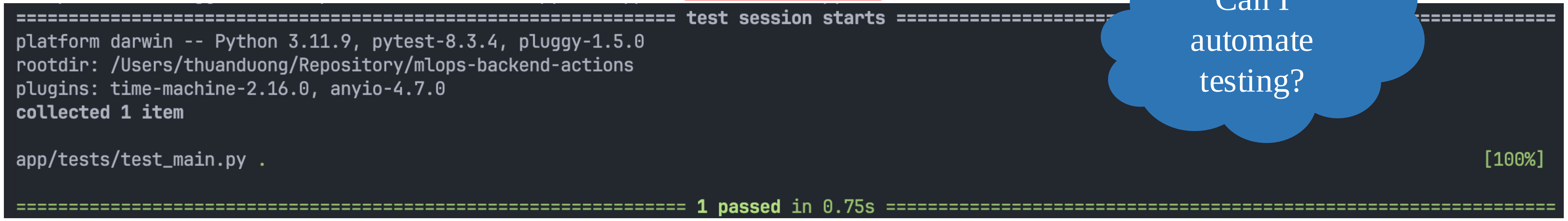
❖ Manual deploy



CI/CD in DevOps

❖ Manual build

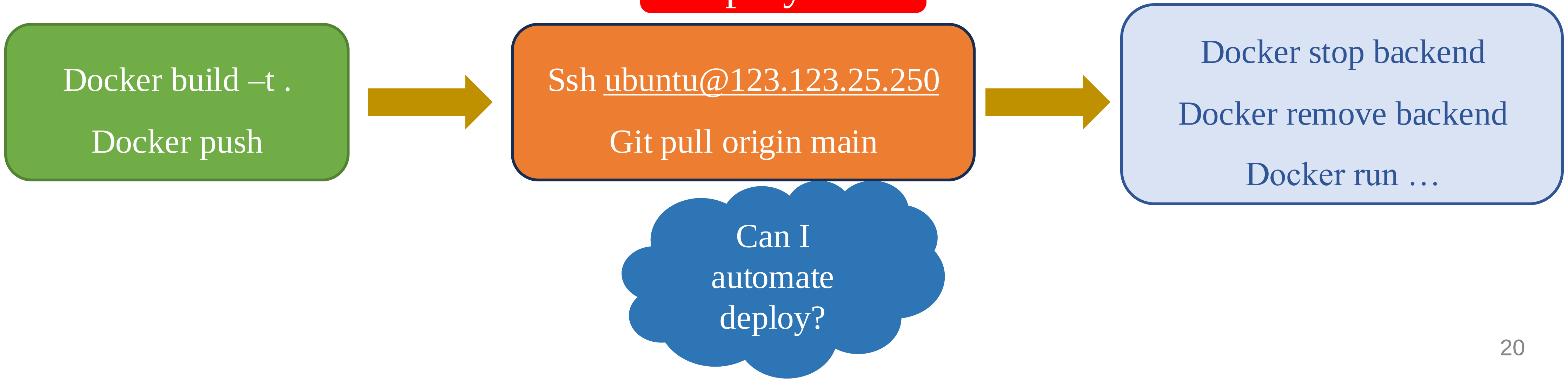
Testing



```
============================================================= test session starts ====================
platform darwin -- Python 3.11.9, pytest-8.3.4, pluggy-1.5.0
rootdir: /Users/thuanduong/Repository/mlops-backend-actions
plugins: time-machine-2.16.0, anyio-4.7.0
collected 1 item

app/tests/test_main.py .                                                                    [100%]

============================================================== 1 passed in 0.75s ==============================================================
```

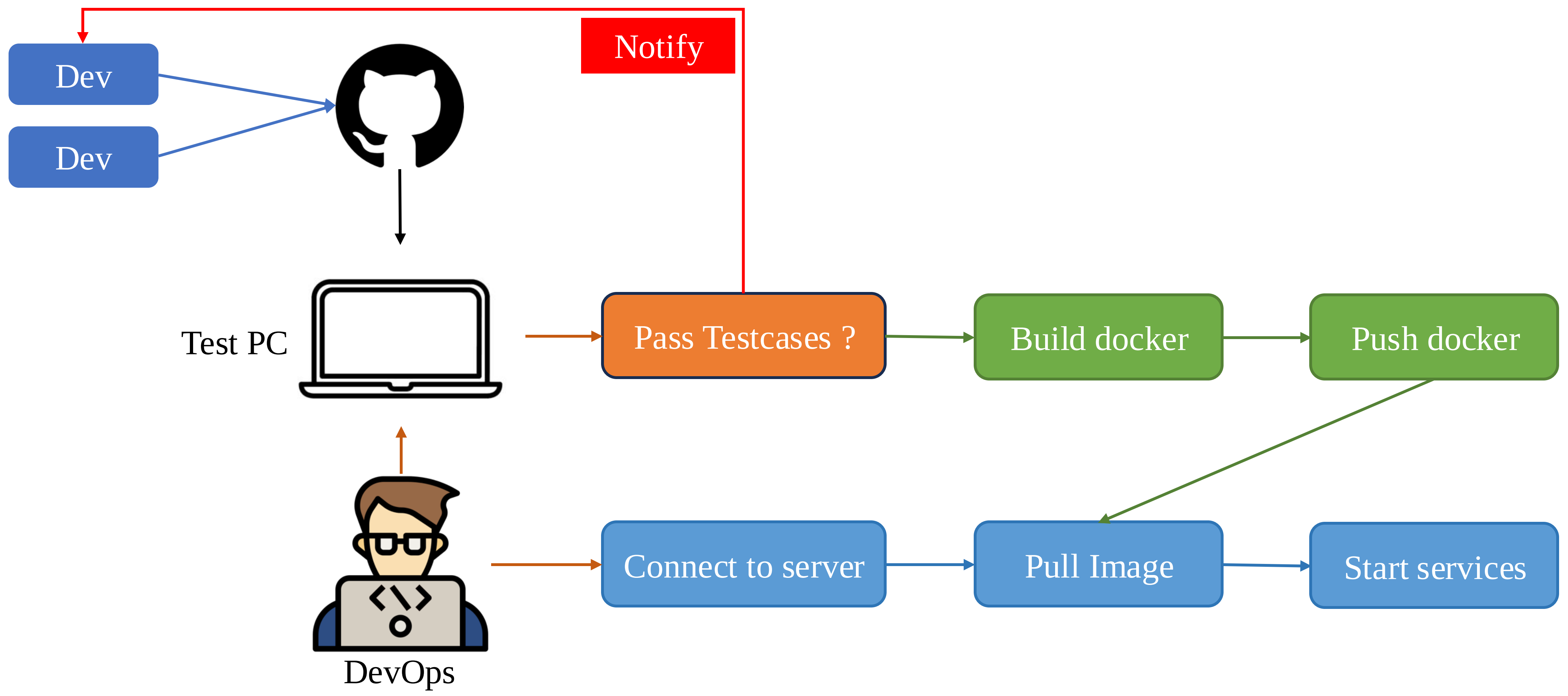
Can I automate testing?

Deployment



CI/CD in DevOps

❖ Deployment



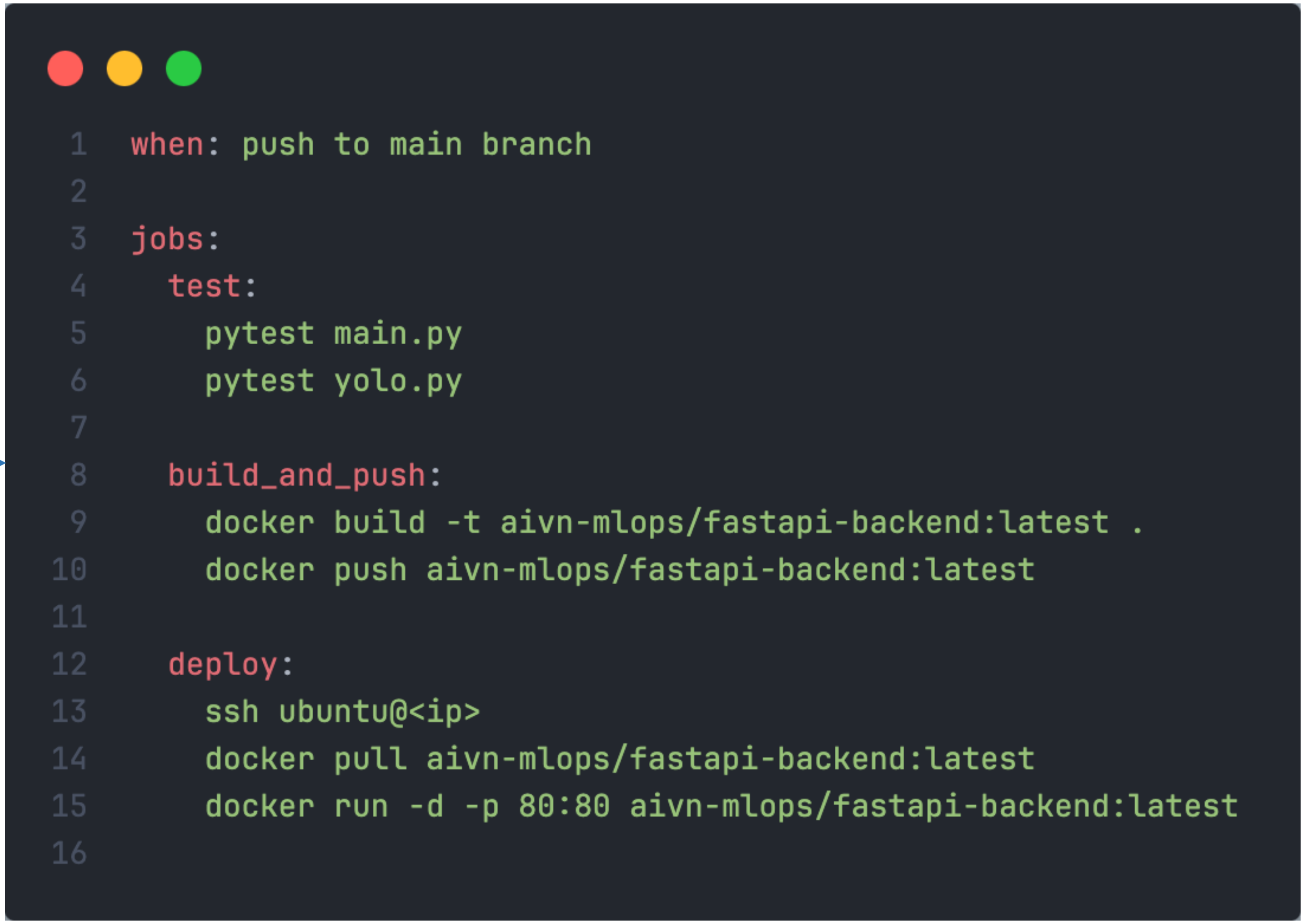
AI VIETNAM
All-in-One Course
(TA Session)

CI/CD in DevOps

❖ Deployment

DevOps

Define jobs or tasks



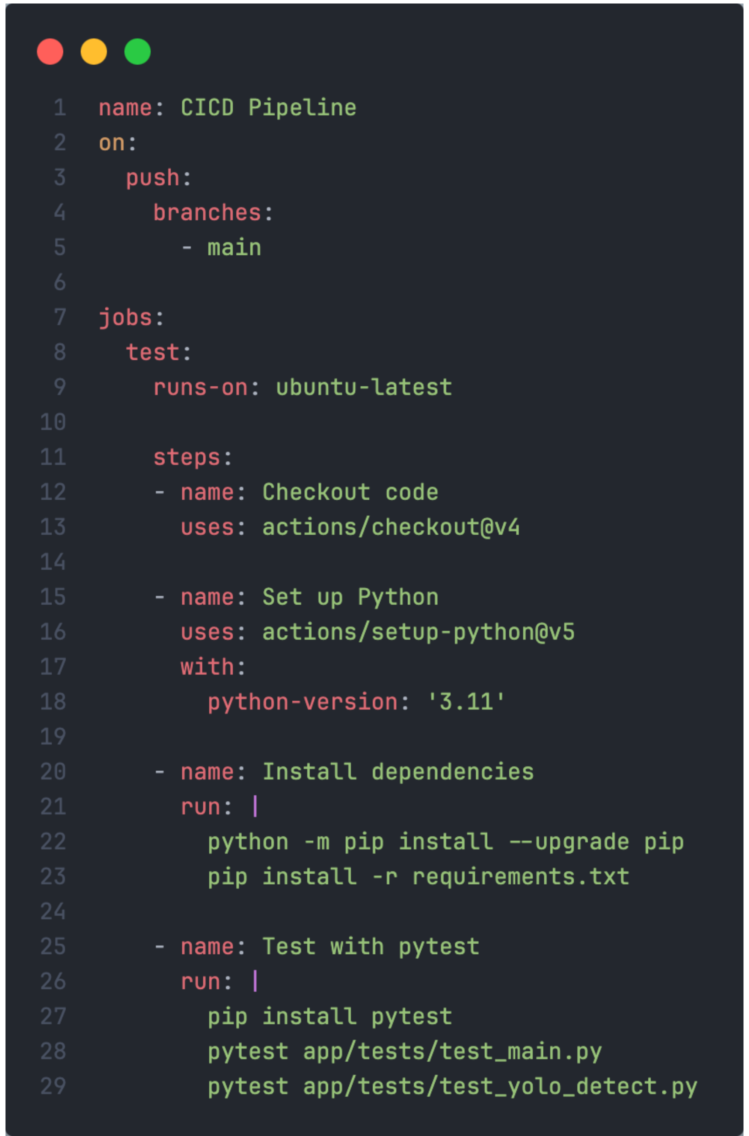
```
when: push to main branch

jobs:
  test:
    pytest main.py
    pytest yolo.py

  build_and_push:
    docker build -t aivn-mlops/fastapi-backend:latest .
    docker push aivn-mlops/fastapi-backend:latest

  deploy:
    ssh ubuntu@<ip>
    docker pull aivn-mlops/fastapi-backend:latest
    docker run -d -p 80:80 aivn-mlops/fastapi-backend:latest

```

ref: https://docs.github.com/en/actions/writing-workflows

CI

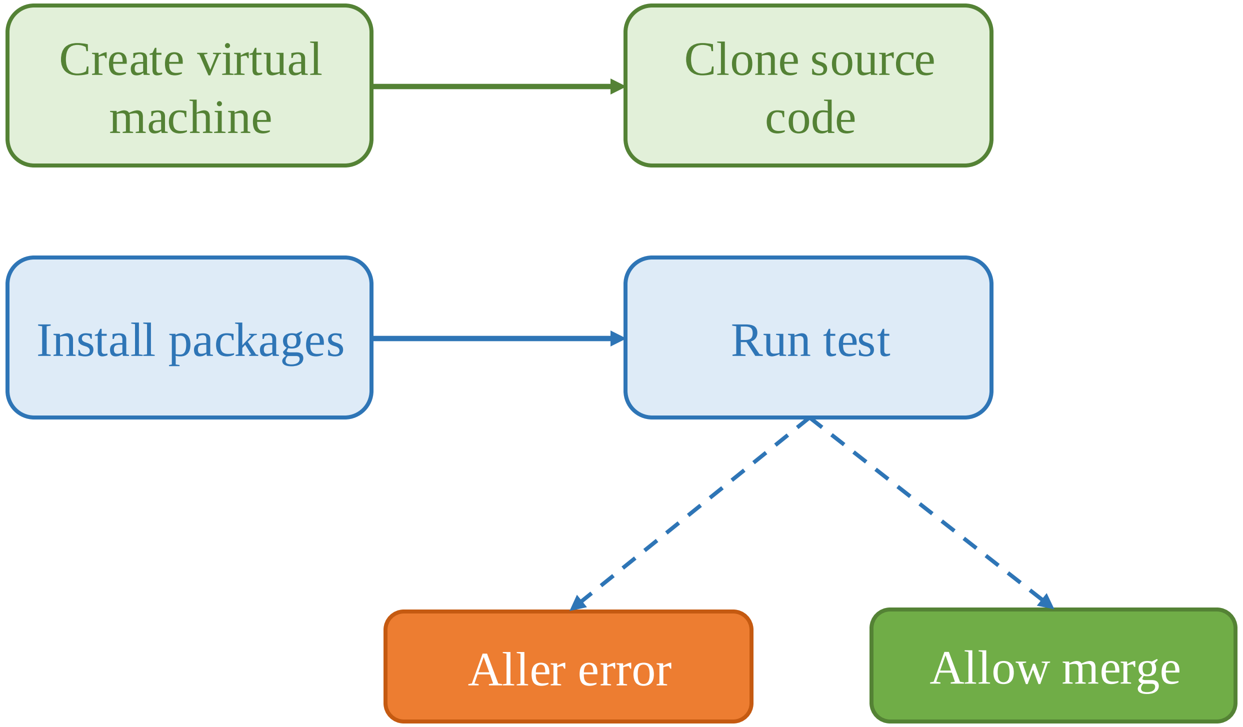
CI/CD in DevOps

❖ Github Actions: CI



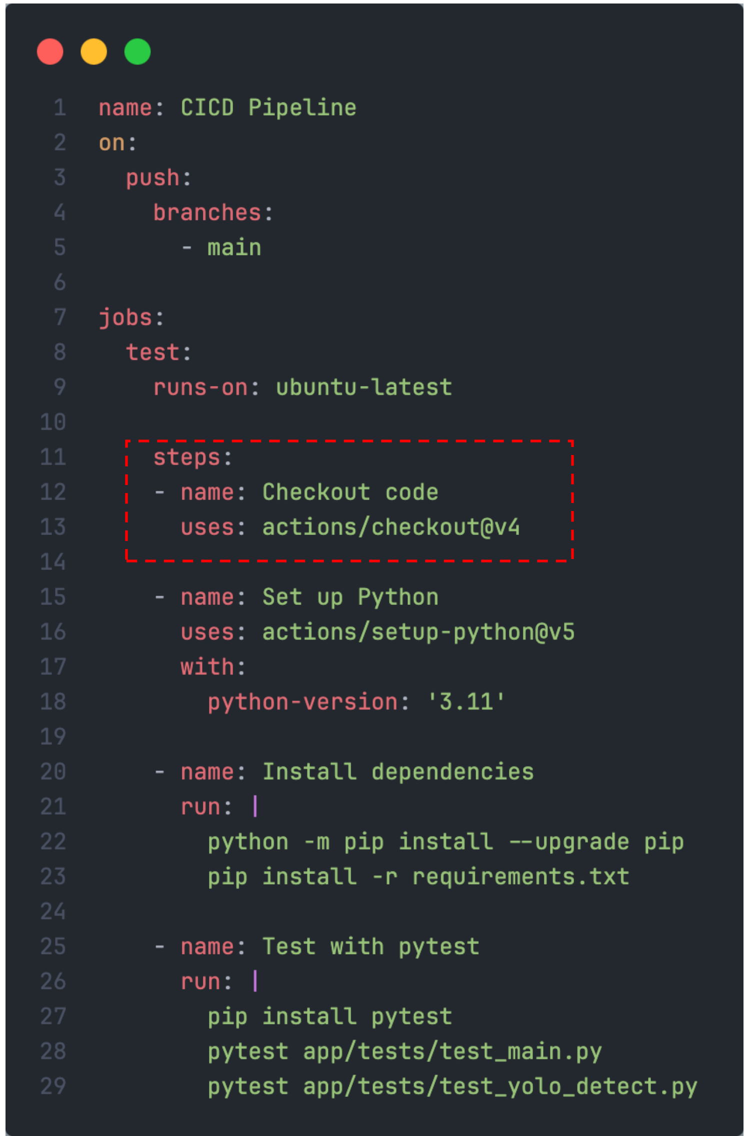
```yaml
name: CICD Pipeline
on:
  push:
    branches:
      - main

jobs:
  test:
    runs-on: ubuntu-latest

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

    - name: Set up Python
      uses: actions/setup-python@v5
      with:
        python-version: '3.11'

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

    - name: Test with pytest
      run: |
        pip install pytest
        pytest app/tests/test_main.py
        pytest app/tests/test_yolo_detect.py
```

Run pipeline when having a push/pull request the event to the main branch.

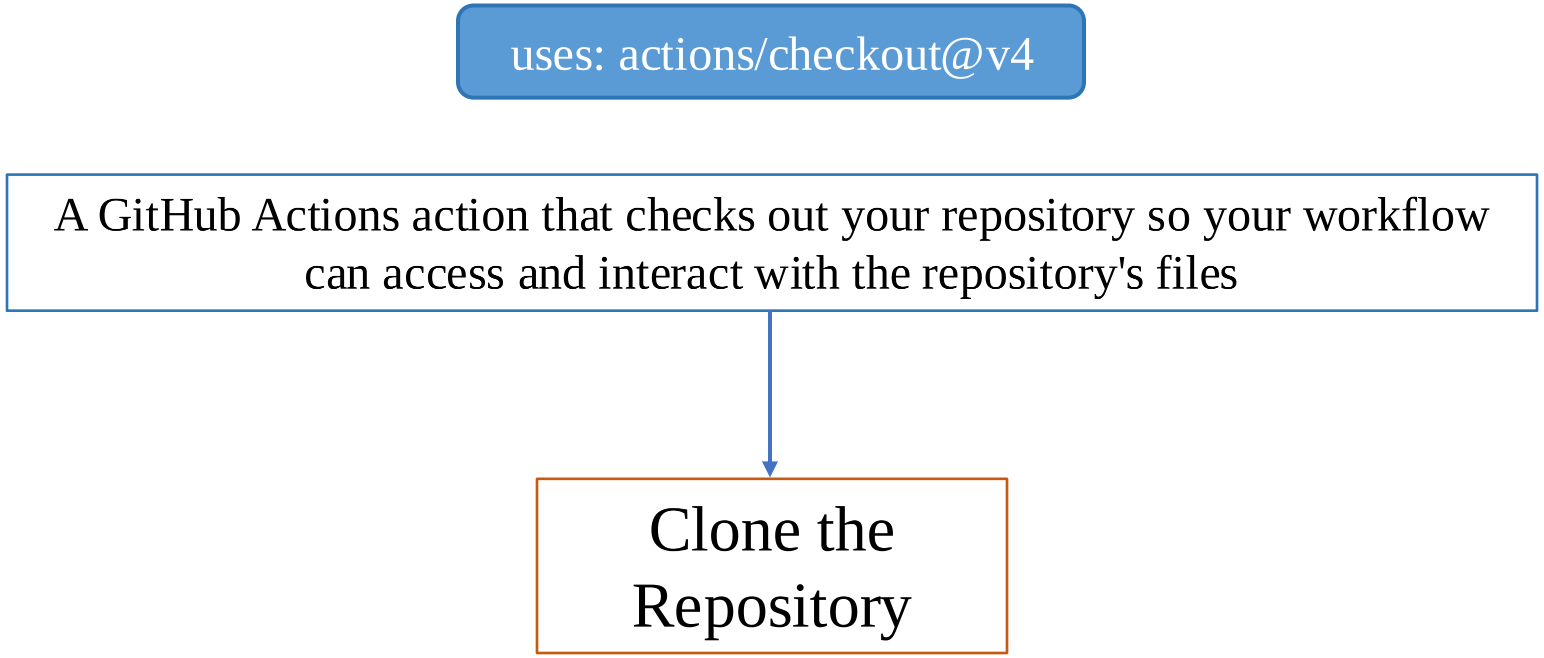


CI/CD in DevOps

❖ Github Actions: CI



```yaml
name: CICD Pipeline
on:
  push:
    branches:
      - main

jobs:
  test:
    runs-on: ubuntu-latest

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

    - name: Set up Python
      uses: actions/setup-python@v5
      with:
        python-version: '3.11'

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

    - name: Test with pytest
      run: |
        pip install pytest
        pytest app/tests/test_main.py
        pytest app/tests/test_yolo_detect.py
```



uses: actions/checkout@v4

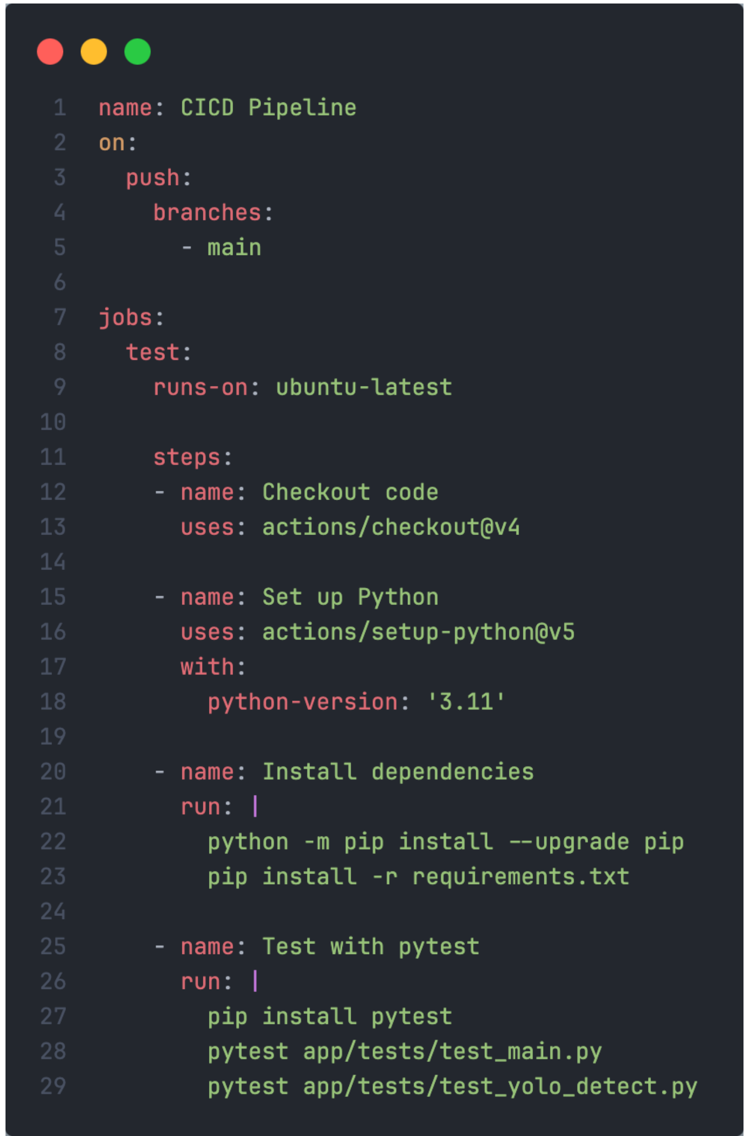
A GitHub Actions action that checks out your repository so your workflow can access and interact with the repository's files

Clone the Repository

Ref: https://github.com/actions/checkout Ref: https://github.com/marketplace/actions/checkout

CI/CD in DevOps

❖ Github Actions: CI



```yaml
name: CICD Pipeline
on:
  push:
    branches:
      - main

jobs:
  test:
    runs-on: ubuntu-latest

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

    - name: Set up Python
      uses: actions/setup-python@v5
      with:
        python-version: '3.11'

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

    - name: Test with pytest
      run: |
        pip install pytest
        pytest app/tests/test_main.py
        pytest app/tests/test_yolo_detect.py
```

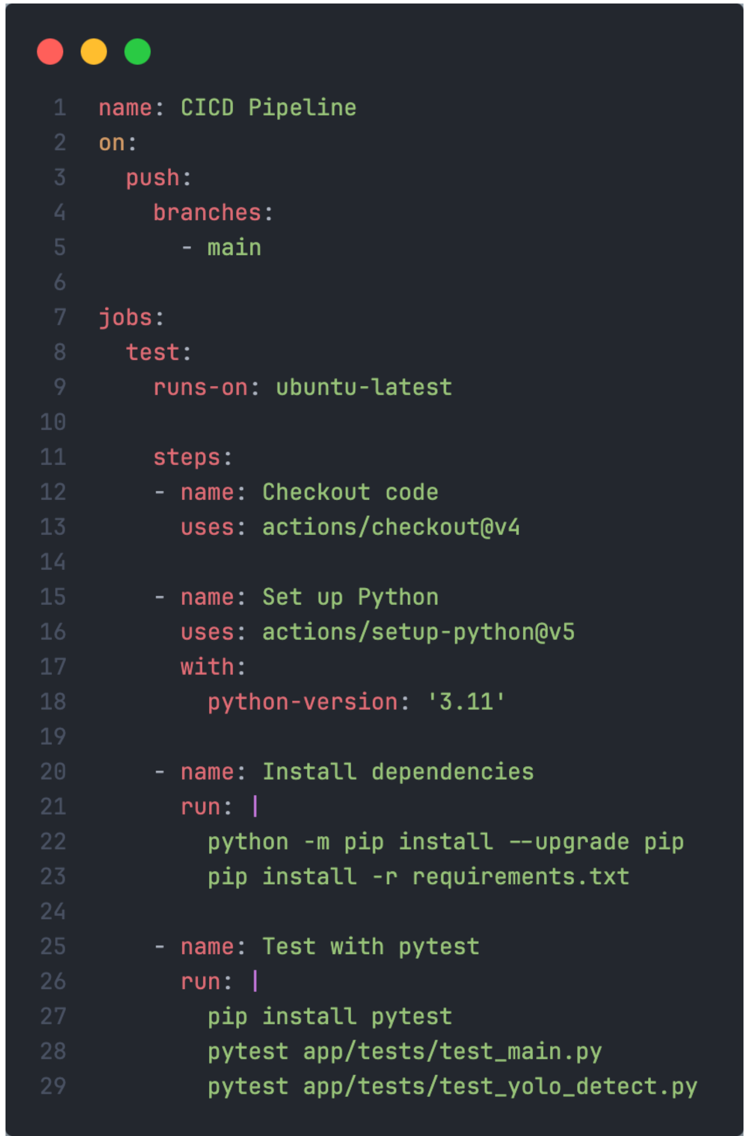
uses: actions/setup-python@v5

Installing a version of Python or PyPy and (by default) adding it to the PATH

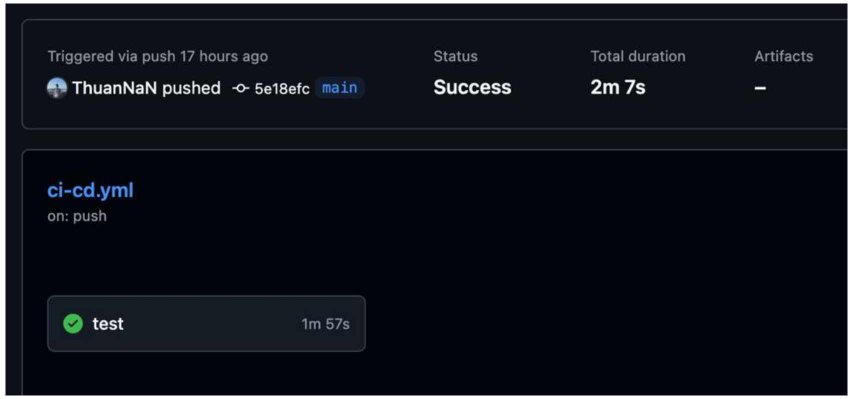
Ref: https://github.com/actions/setup-python   Ref: https://github.com/marketplace/actions/setup-python

CI/CD in DevOps

❖ Github Actions: CI



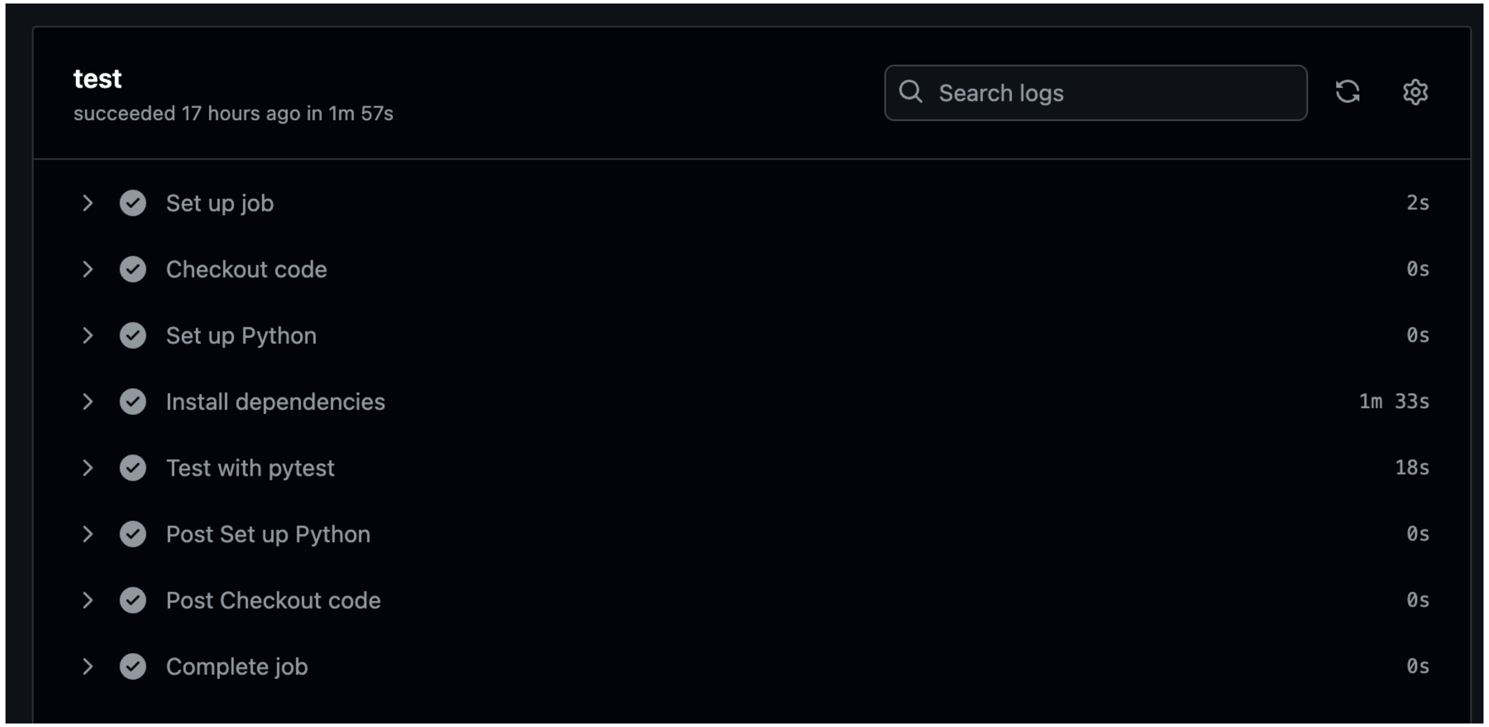
```yaml
name: CICD Pipeline
on:
  push:
    branches:
      - main

jobs:
  test:
    runs-on: ubuntu-latest

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

    - name: Set up Python
      uses: actions/setup-python@v5
      with:
        python-version: '3.11'

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

    - name: Test with pytest
      run: |
        pip install pytest
        pytest app/tests/test_main.py
        pytest app/tests/test_yolo_detect.py
```



Triggered via push 17 hours ago	Status	Total duration	Artifacts
ThuanNaN pushed 5e18efc main	Success	2m 7s	–

ci-cd.yml
on: push

test 1m 57s



test
succeeded 17 hours ago in 1m 57s

Search logs

Set up job	2s
Checkout code	0s
Set up Python	0s
Install dependencies	1m 33s
Test with pytest	18s
Post Set up Python	0s
Post Checkout code	0s
Complete job	0s

CI/CD in DevOps

❖ Github Actions: Vars

Variables provide a way to store and reuse non-sensitive configuration information.

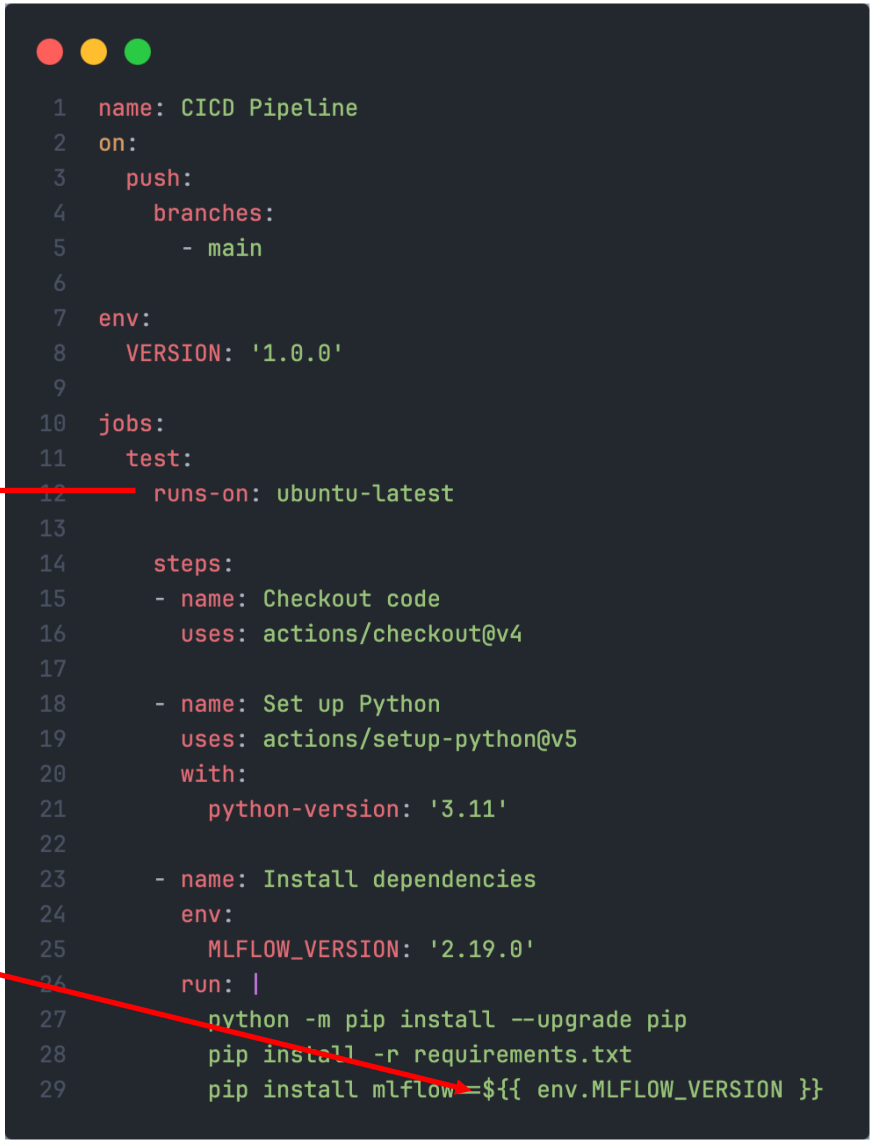
Single workflows

Linux runners

Use **bash shell** syntax
$NAME

Windows runners

Use **PowerShell** syntax
$env:NAME



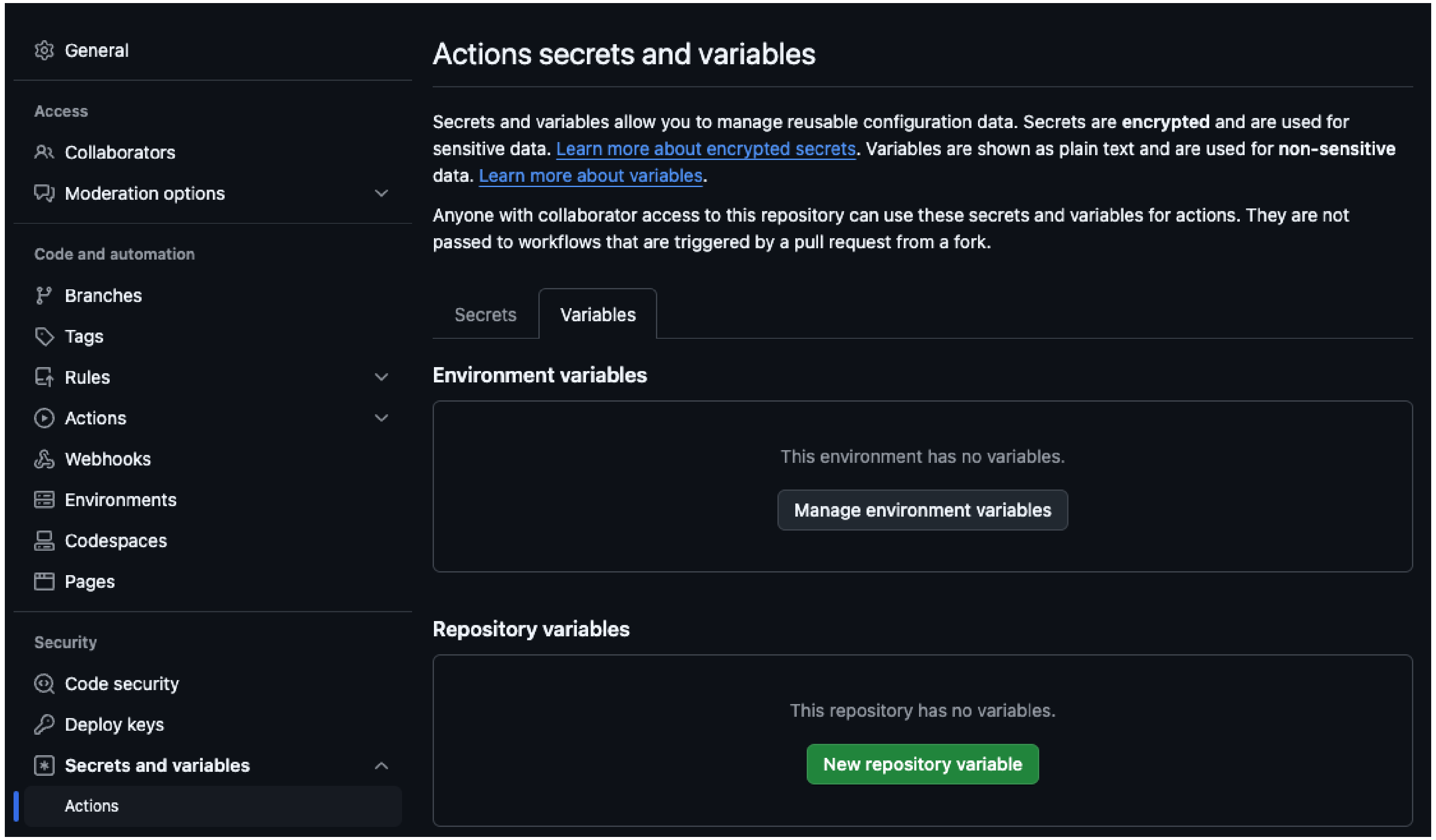
```yaml
name: CICD Pipeline
on:
  push:
    branches:
      - main

env:
  VERSION: '1.0.0'

jobs:
  test:
    runs-on: ubuntu-latest

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

    - name: Set up Python
      uses: actions/setup-python@v5
      with:
        python-version: '3.11'

    - name: Install dependencies
      env:
        MLFLOW_VERSION: '2.19.0'
      run: |
        python -m pip install --upgrade pip
        pip install -r requirements.txt
        pip install mlflow=${{ env.MLFLOW_VERSION }}
```

Ref: https://docs.github.com/en/actions/writing-workflows/choosing-what-your-workflow-does/store-information-in-variables

CI/CD in DevOps

❖ Github Actions: Vars

Configuration variables can be created for use across multiple workflows and can be defined at the organization, repository, or environment level.



Actions secrets and variables

Secrets and variables allow you to manage reusable configuration data. Secrets are **encrypted** and are used for sensitive data. Learn more about encrypted secrets. Variables are shown as plain text and are used for **non-sensitive** data. Learn more about variables.

Anyone with collaborator access to this repository can use these secrets and variables for actions. They are not passed to workflows that are triggered by a pull request from a fork.

Environment variables

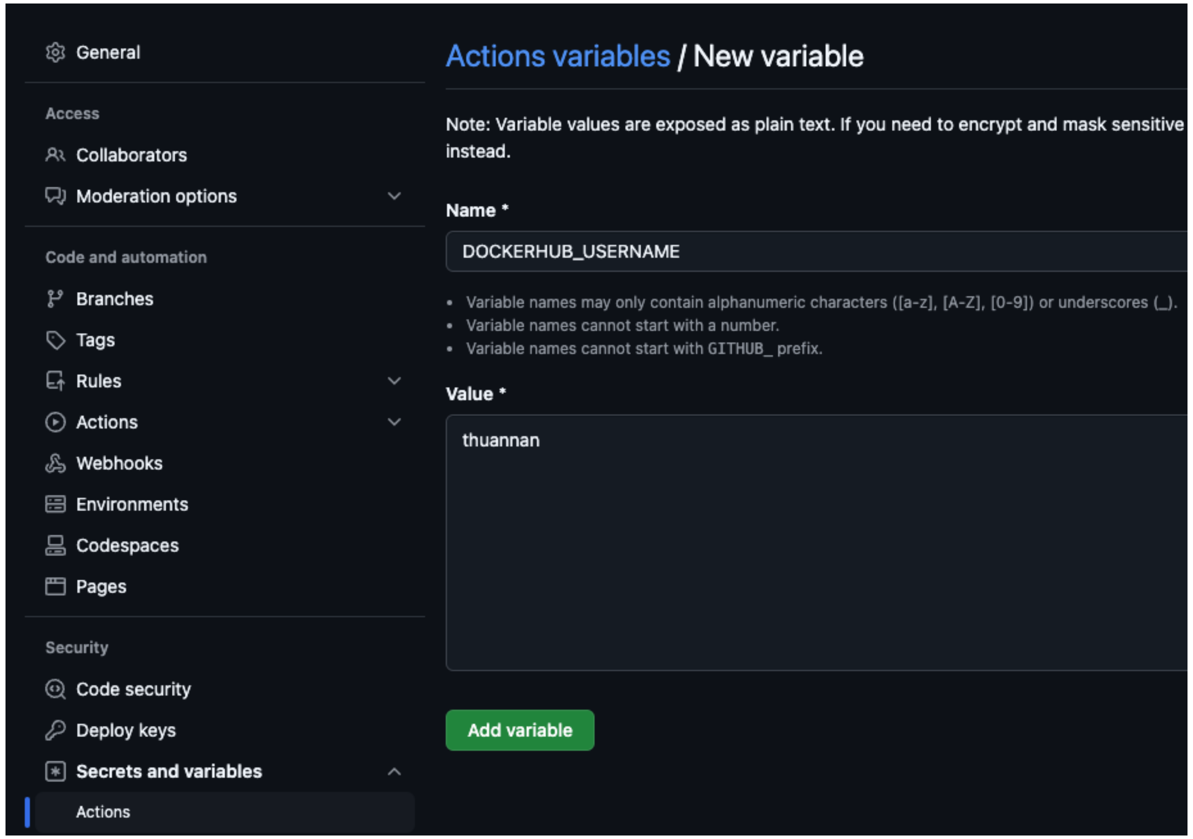
This environment has no variables.

Repository variables

This repository has no variables.

CI/CD in DevOps

❖ Github Actions: Vars



Actions variables / New variable

Note: Variable values are exposed as plain text. If you need to encrypt and mask sensitive
instead.

Name *

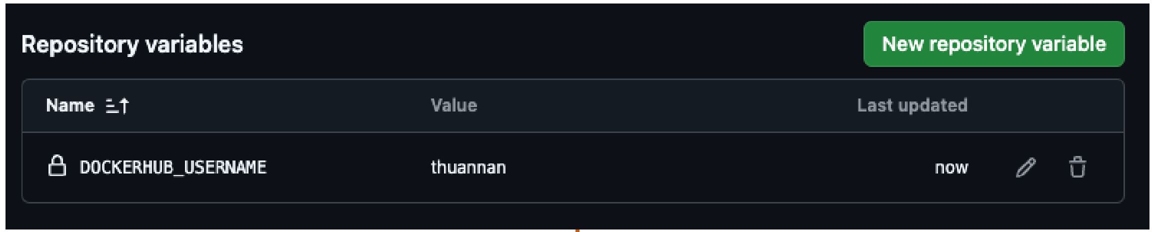
DOCKERHUB_USERNAME

- 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 *

thuannan

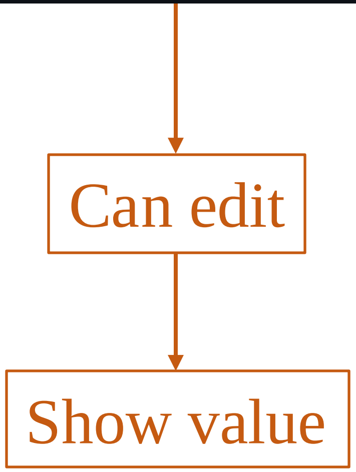
Add variable



Repository variables

New repository variable

Name	Value	Last updated
DOCKERHUB_USERNAME	thuannan	now

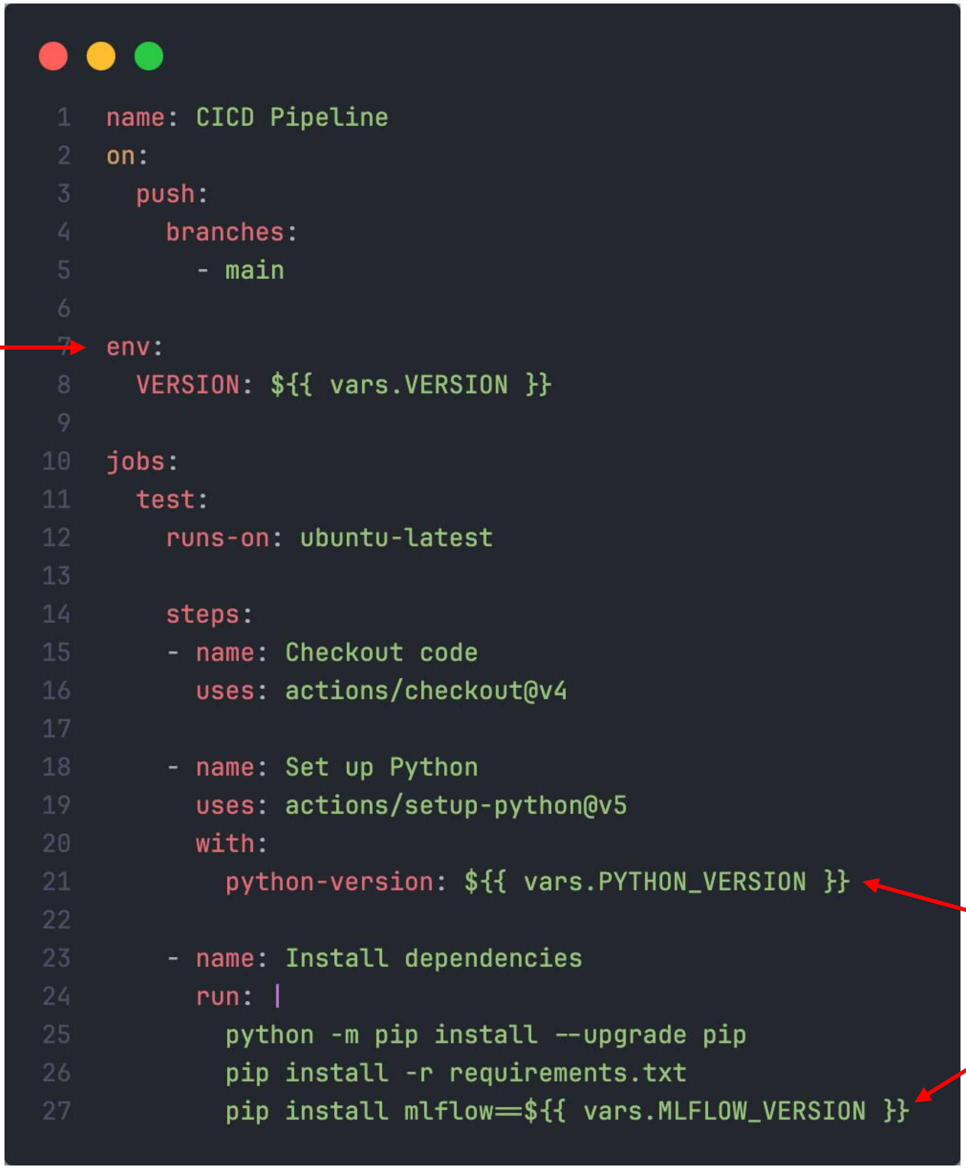


CI/CD in DevOps

❖ Github Actions: Vars

Multiple workflows

Setting an environment variable



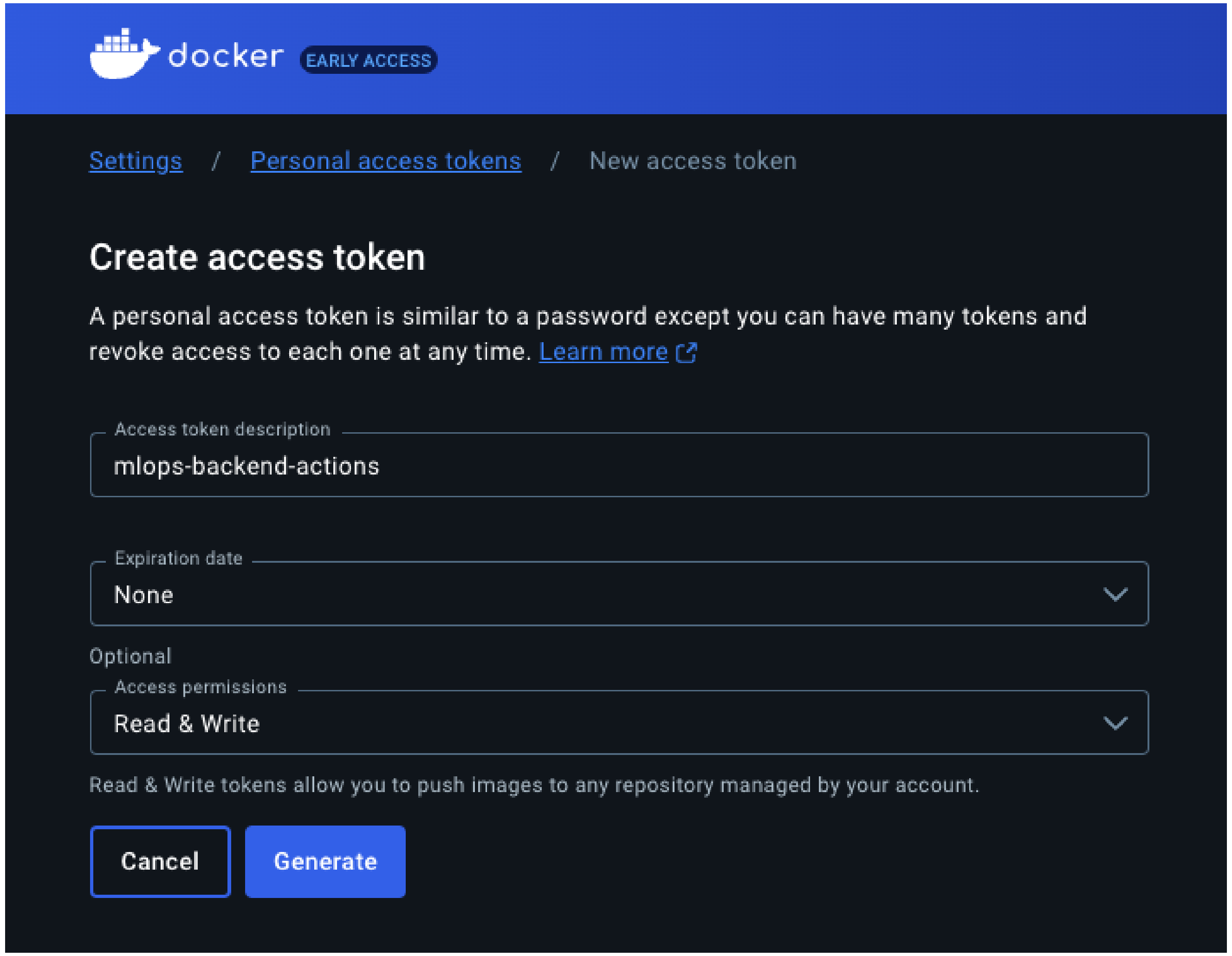
```yaml
name: CICD Pipeline
on:
  push:
    branches:
      - main

env:
  VERSION: ${{ vars.VERSION }}

jobs:
  test:
    runs-on: ubuntu-latest

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

    - name: Set up Python
      uses: actions/setup-python@v5
      with:
        python-version: ${{ vars.PYTHON_VERSION }}

    - name: Install dependencies
      run: |
        python -m pip install --upgrade pip
        pip install -r requirements.txt
        pip install mlflow==${{ vars.MLFLOW_VERSION }}
```

Use “vars” context

CI/CD in DevOps

❖ Docker Hub access token

Go to https://app.docker.com/settings/personal-access-tokens to create docker hub 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: mlops-backend-actions

Expiration date: None

Optional

Access permissions: Read & Write

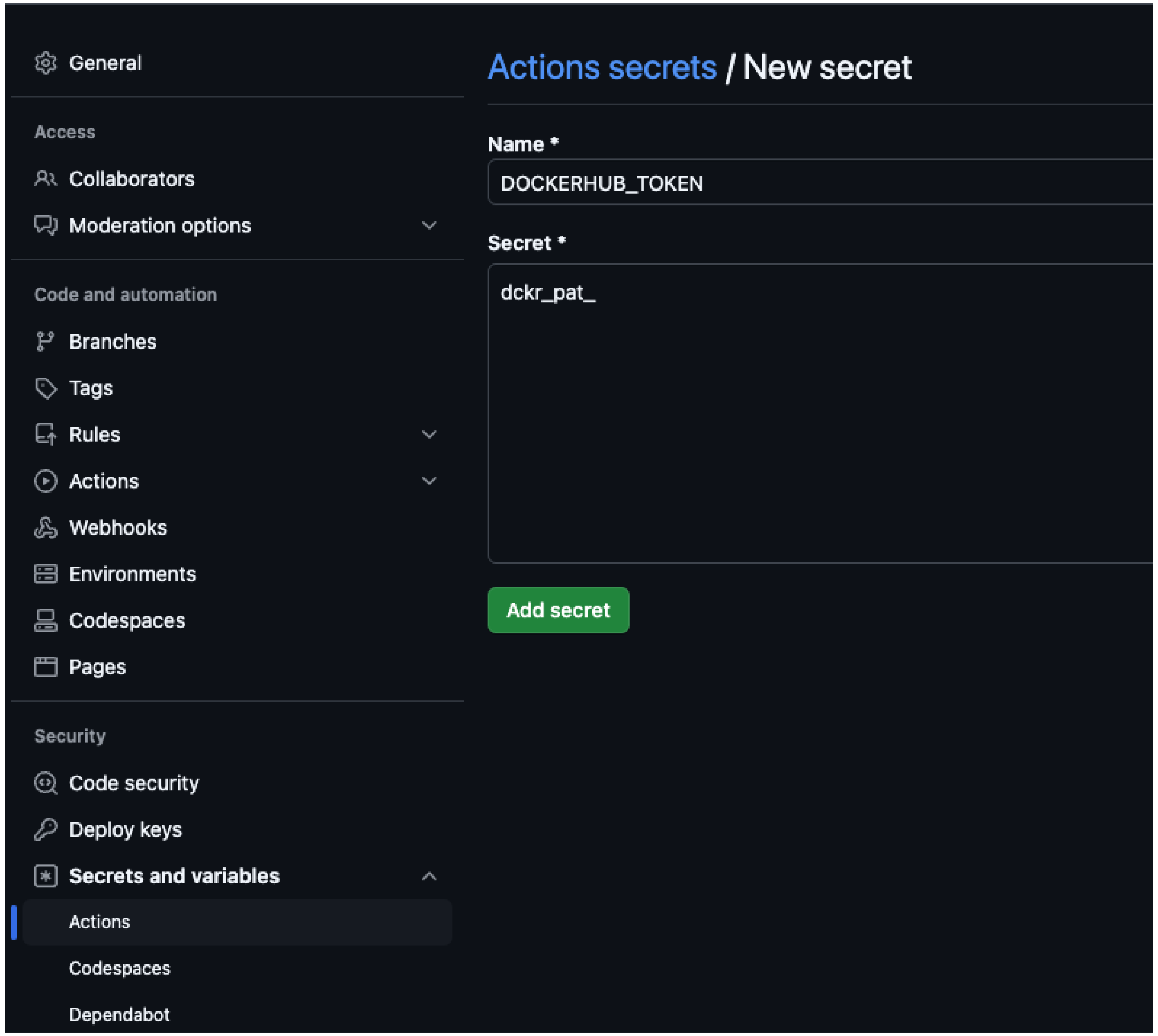
Read & Write tokens allow you to push images to any repository managed by your account.

CI/CD in DevOps

❖ Github Actions: Secrets

Secrets are variables defined within an organization, repository, or repository environment.

These secrets can be utilized in GitHub Actions workflows but can only be accessed by GitHub Actions if they are explicitly specified in a workflow



Actions secrets / New secret

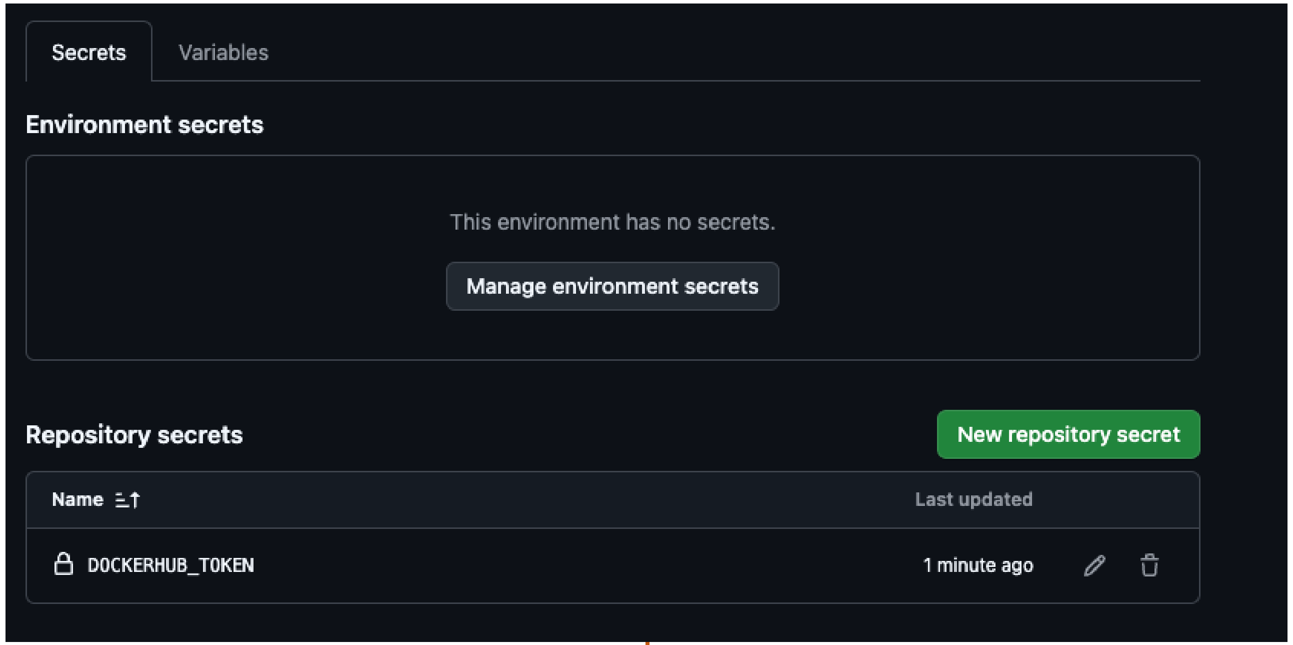
Name *
DOCKERHUB_TOKEN

Secret *
dckr_pat_

Ref: https://docs.github.com/en/actions/security-for-github-actions/security-guides/using-secrets-in-github-actions

CI/CD in DevOps

❖ Github Actions: Secrets

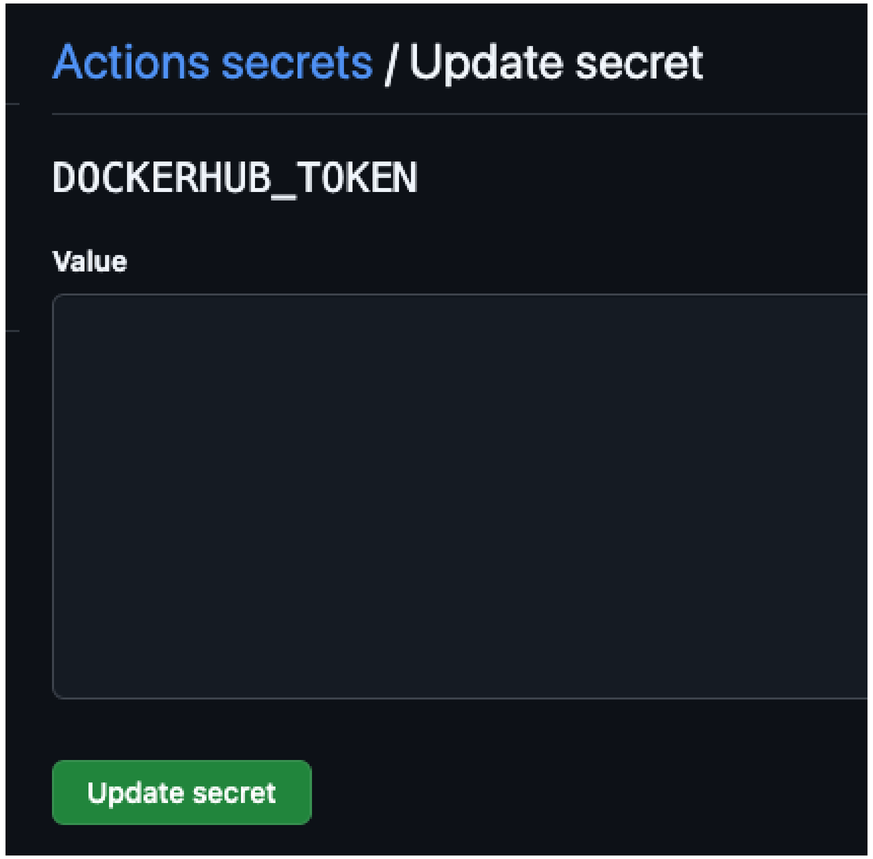


Environment secrets

This environment has no secrets.

Repository secrets

Name	Last updated
DOCKERHUB_TOKEN	1 minute ago


Actions secrets / Update secret

DOCKERHUB_TOKEN

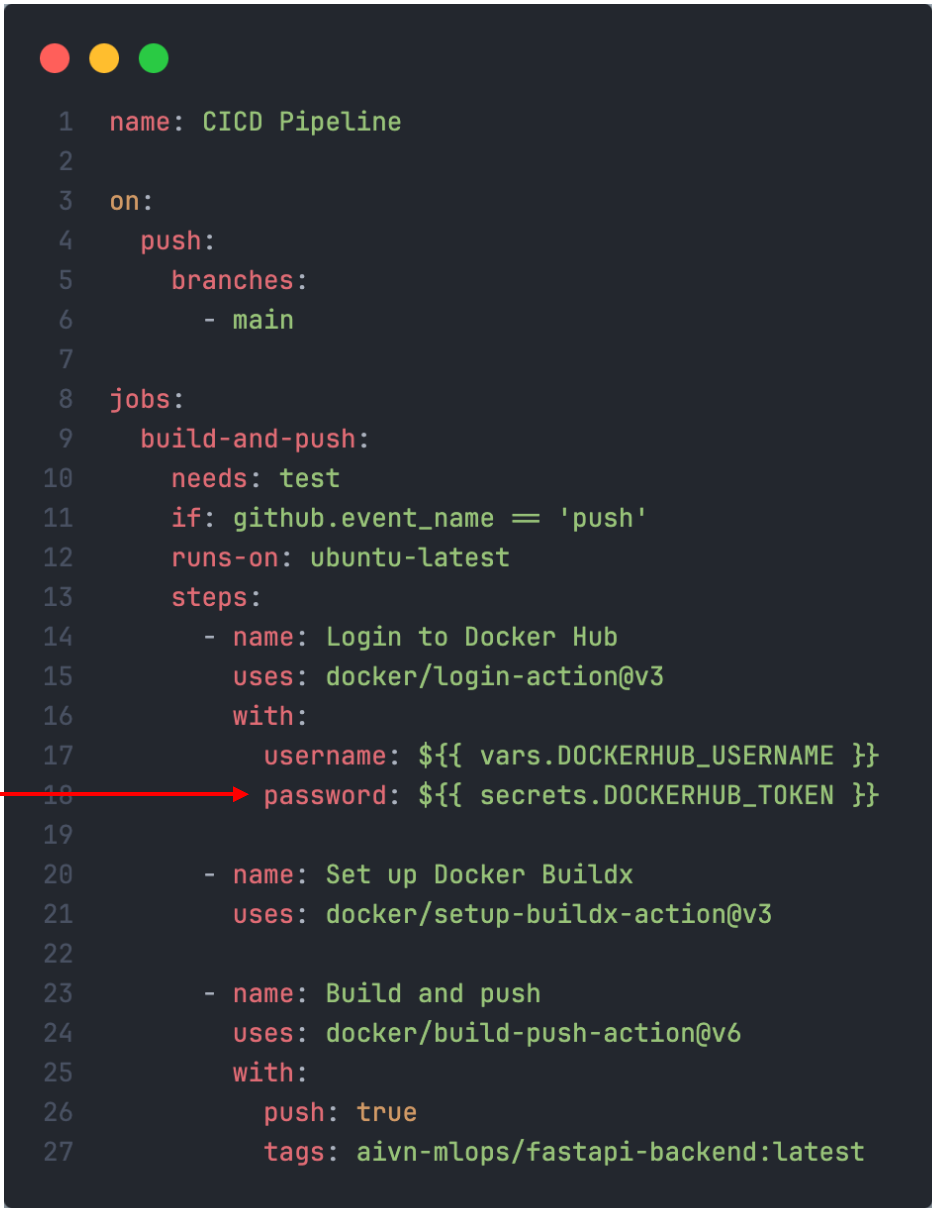
Value

Can edit

Not show value

CI/CD in DevOps

❖ Using secrets in a workflow

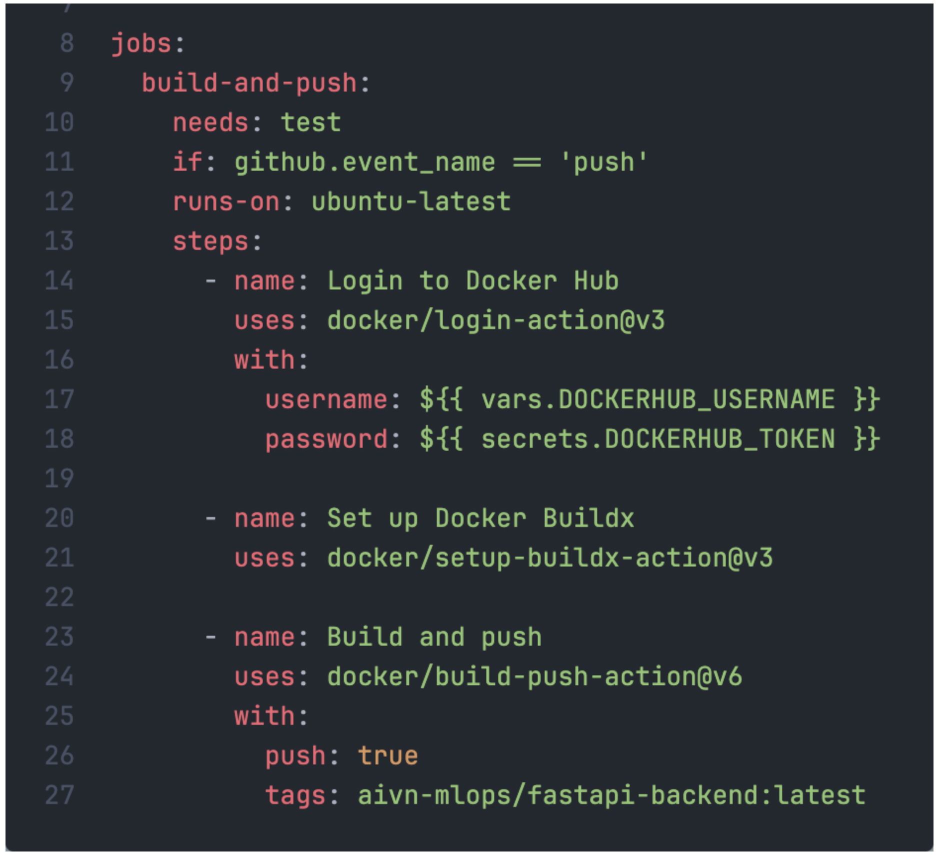


```yaml
name: CICD Pipeline

on:
  push:
    branches:
      - main

jobs:
  build-and-push:
    needs: test
    if: github.event_name == 'push'
    runs-on: ubuntu-latest
    steps:
      - name: Login to Docker Hub
        uses: docker/login-action@v3
        with:
          username: ${{ vars.DOCKERHUB_USERNAME }}
          password: ${{ secrets.DOCKERHUB_TOKEN }}

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

      - name: Build and push
        uses: docker/build-push-action@v6
        with:
          push: true
          tags: aivn-mlops/fastapi-backend:latest
```

Use “secrets” context

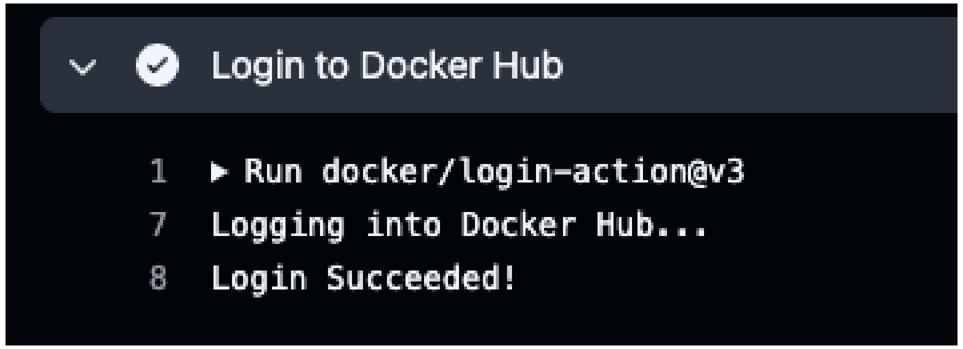
CI/CD in DevOps

❖ Docker actions



```yaml
jobs:
  build-and-push:
    needs: test
    if: github.event_name == 'push'
    runs-on: ubuntu-latest
    steps:
      - name: Login to Docker Hub
        uses: docker/login-action@v3
        with:
          username: ${{ vars.DOCKERHUB_USERNAME }}
          password: ${{ secrets.DOCKERHUB_TOKEN }}

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

      - name: Build and push
        uses: docker/build-push-action@v6
        with:
          push: true
          tags: aivn-mlops/fastapi-backend:latest
```

uses: docker/login-action@v3



Login to Docker Hub

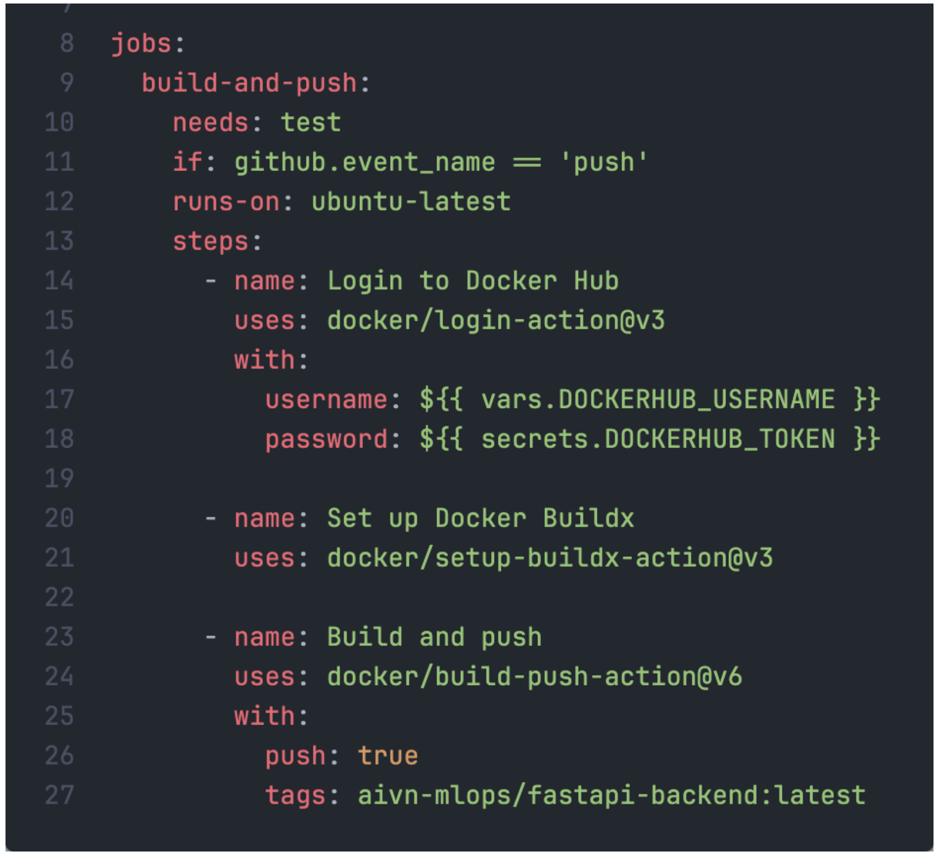
```
1  ▶ Run docker/login-action@v3
7  Logging into Docker Hub...
8  Login Succeeded!
```

Ref: https://github.com/docker/login-action   Ref: https://github.com/marketplace/actions/docker-login

AI VIETNAM
All-in-One Course
(TA Session)

CI/CD in DevOps

❖ Docker actions



```
jobs:
  build-and-push:
    needs: test
    if: github.event_name == 'push'
    runs-on: ubuntu-latest
    steps:
      - name: Login to Docker Hub
        uses: docker/login-action@v3
        with:
          username: ${{ vars.DOCKERHUB_USERNAME }}
          password: ${{ secrets.DOCKERHUB_TOKEN }}

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

      - name: Build and push
        uses: docker/build-push-action@v6
        with:
          push: true
          tags: aivn-mlops/fastapi-backend:latest
```

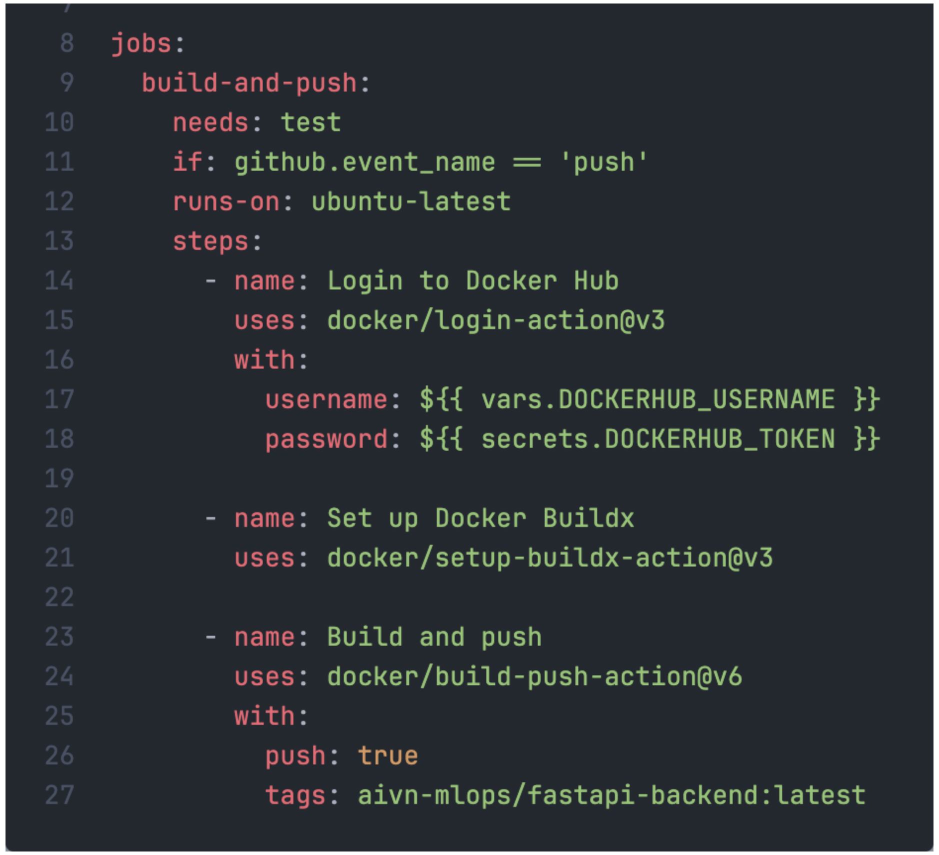
uses: docker/setup-buildx-action@v3

Setup docker buildx to build docker image

Ref: https://github.com/docker/setup-buildx-action   Ref: https://github.com/marketplace/actions/docker-setup-buildx

CI/CD in DevOps

❖ Docker actions


```yaml
jobs:
  build-and-push:
    needs: test
    if: github.event_name == 'push'
    runs-on: ubuntu-latest
    steps:
      - name: Login to Docker Hub
        uses: docker/login-action@v3
        with:
          username: ${{ vars.DOCKERHUB_USERNAME }}
          password: ${{ secrets.DOCKERHUB_TOKEN }}

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

      - name: Build and push
        uses: docker/build-push-action@v6
        with:
          push: true
          tags: aivn-mlops/fastapi-backend:latest
```

uses: docker/build-push-action@v6

GitHub Action to build and push Docker images to Docker Hub

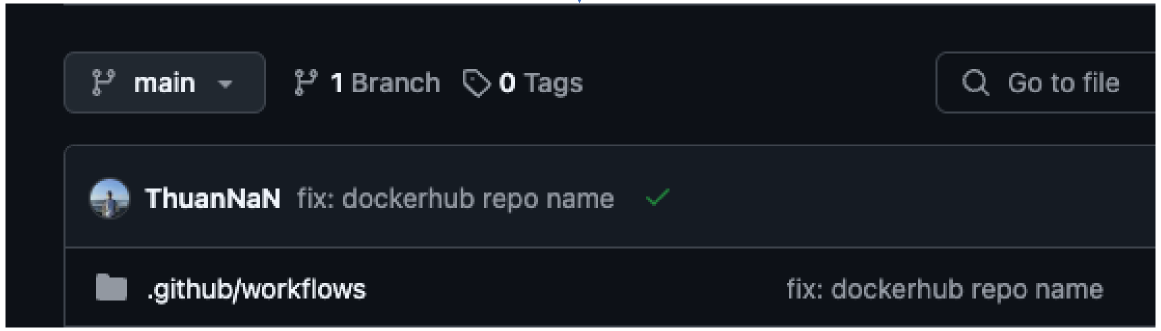
Ref: https://github.com/docker/build-push-action   Ref: https://github.com/marketplace/actions/build-and-push-docker-images

ref: https://docs.docker.com/build/cache/optimize

CI/CD in DevOps

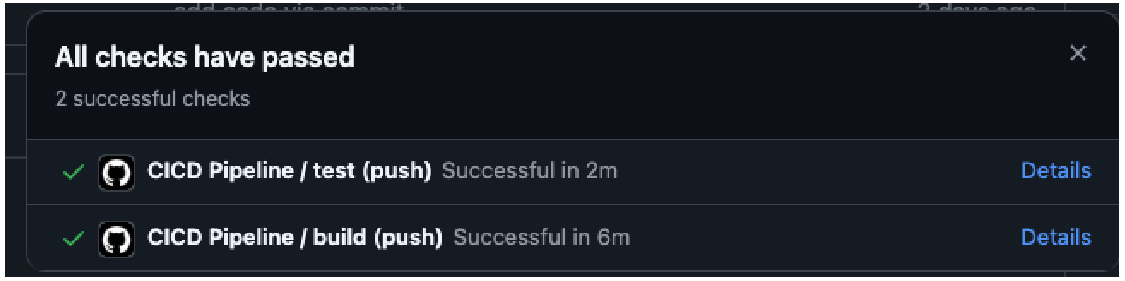
❖ Run CI workflows

```
Git add .
Git commit –m “add feat …”
Git push –u origin main
```


main · 1 Branch · 0 Tags · Go to file

ThuanNaN fix: dockerhub repo name ✓

.github/workflows fix: dockerhub repo name


All checks have passed

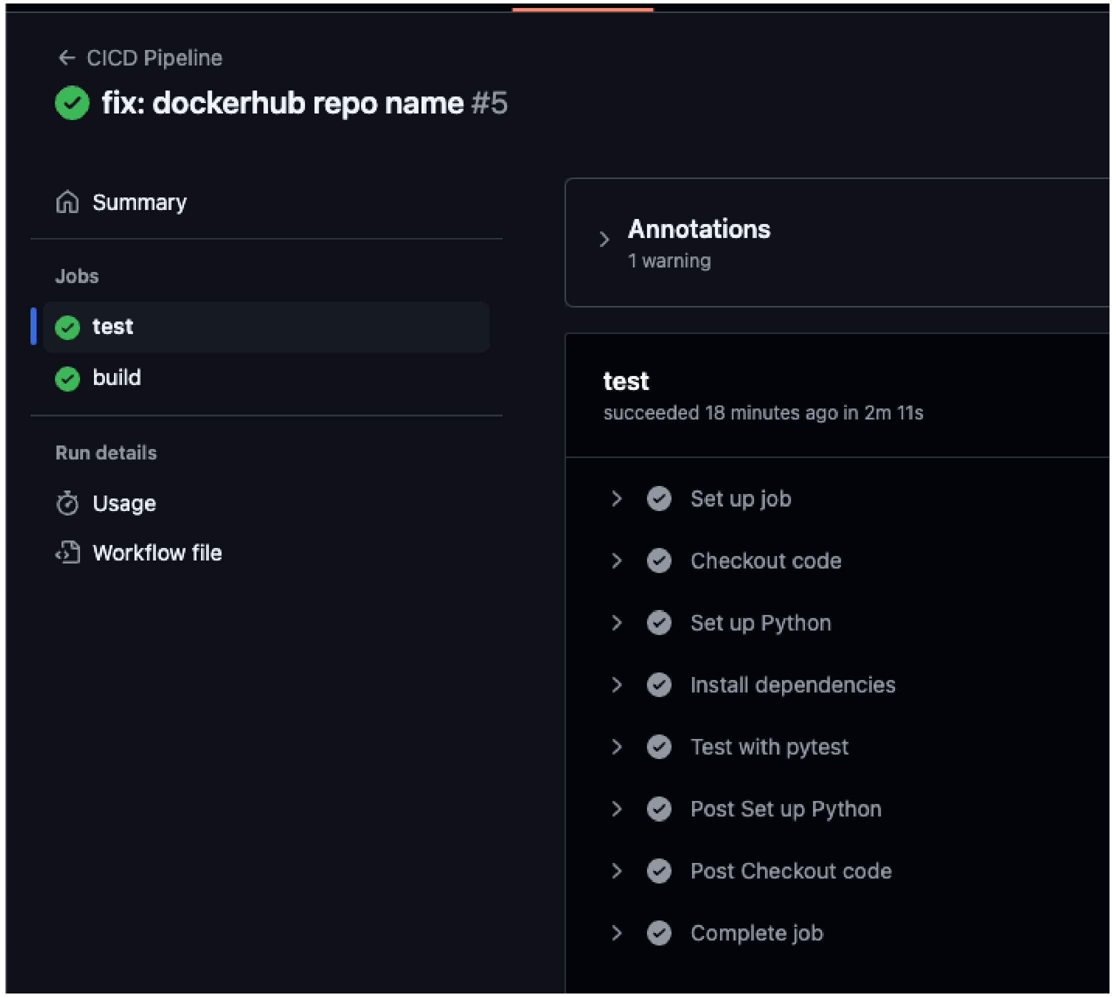
2 successful checks

✓ CICD Pipeline / test (push) Successful in 2m Details

✓ CICD Pipeline / build (push) Successful in 6m Details

CI/CD in DevOps

❖ Run CI workflows



← CICD Pipeline

fix: dockerhub repo name #5

Summary

Jobs

test

build

Run details

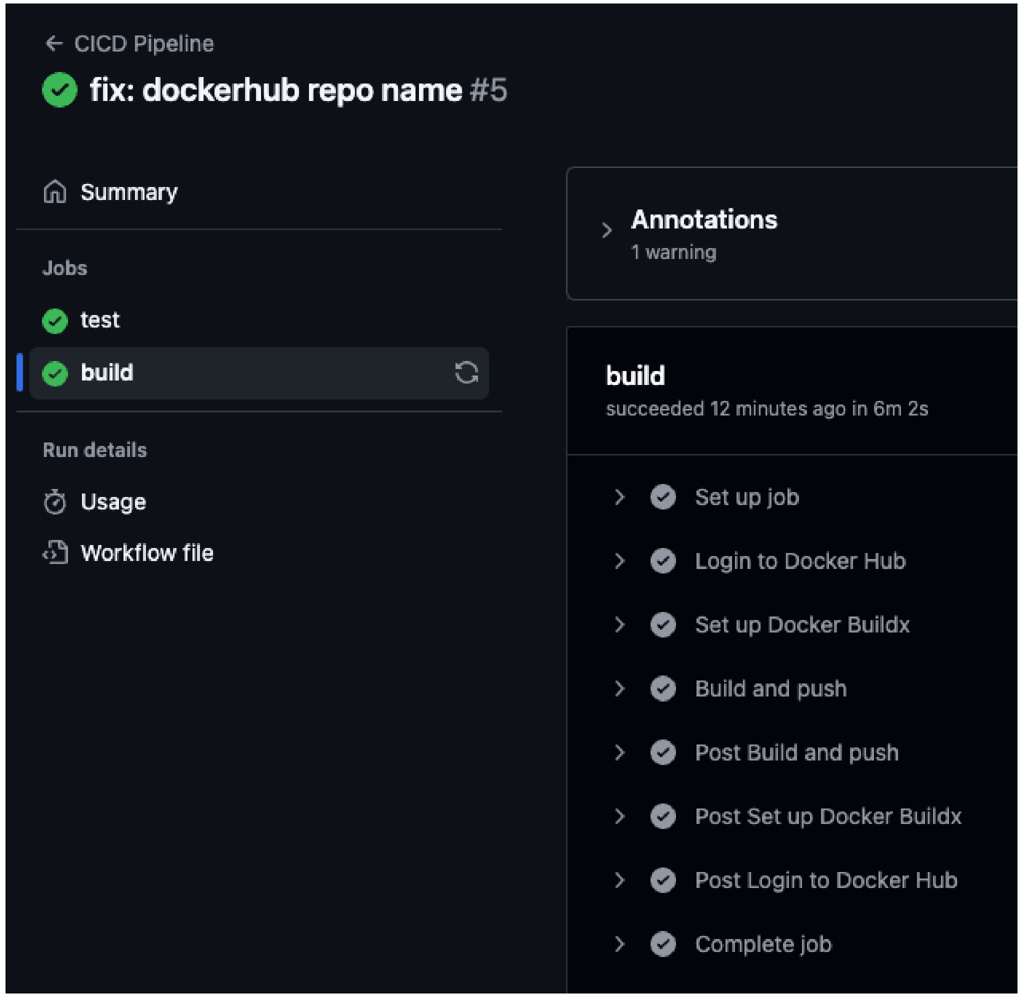
Usage

Workflow file

Annotations
1 warning

test
succeeded 18 minutes ago in 2m 11s

- Set up job
- Checkout code
- Set up Python
- Install dependencies
- Test with pytest
- Post Set up Python
- Post Checkout code
- Complete job



← CICD Pipeline

fix: dockerhub repo name #5

Summary

Jobs

test

build

Run details

Usage

Workflow file

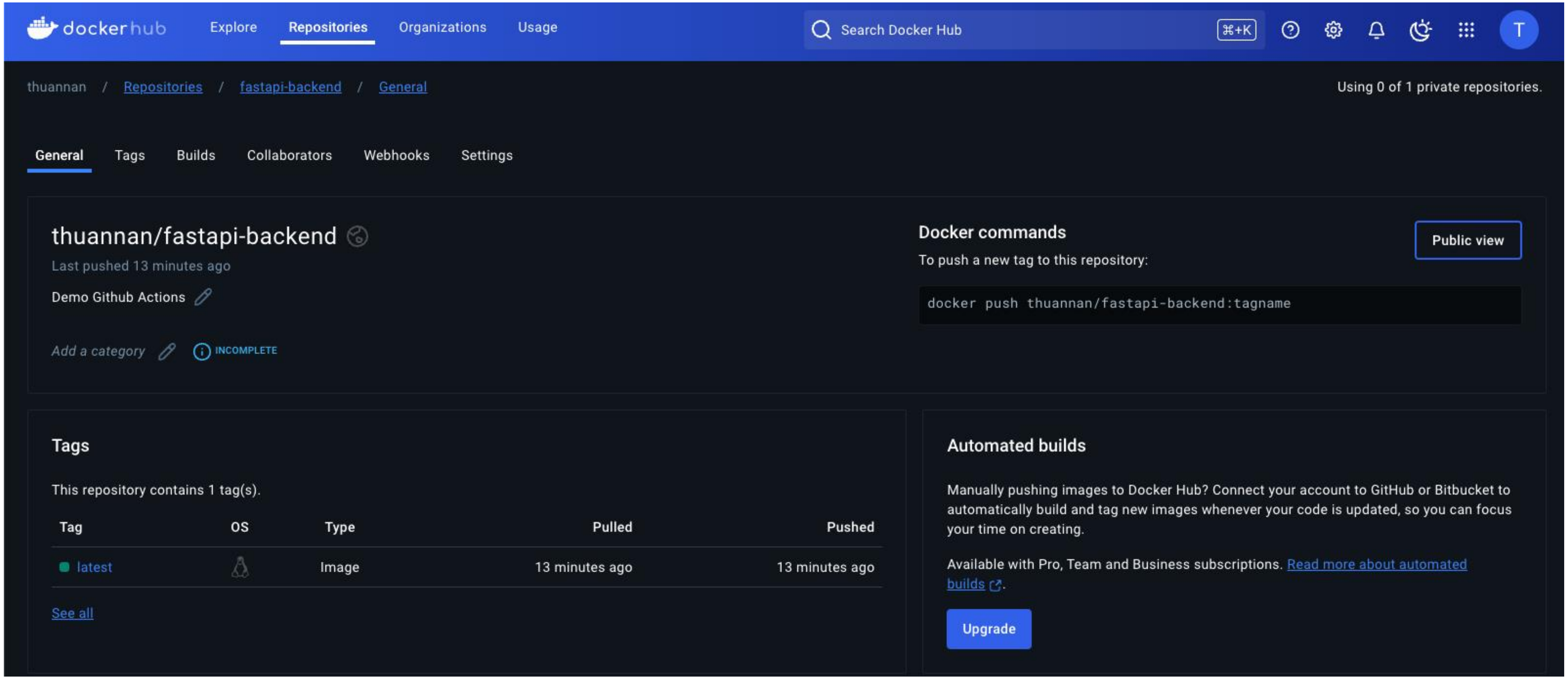
Annotations
1 warning

build
succeeded 12 minutes ago in 6m 2s

- Set up job
- Login to Docker Hub
- Set up Docker Buildx
- Build and push
- Post Build and push
- Post Set up Docker Buildx
- Post Login to Docker Hub
- Complete job

CI/CD in DevOps

❖ Run CI workflows



thuannan / Repositories / fastapi-backend / General

Using 0 of 1 private repositories.

thuannan/fastapi-backend

Last pushed 13 minutes ago

Demo Github Actions

Add a category INCOMPLETE

Docker commands

To push a new tag to this repository:

```
docker push thuannan/fastapi-backend:tagname
```

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest		Image	13 minutes ago	13 minutes ago

See all

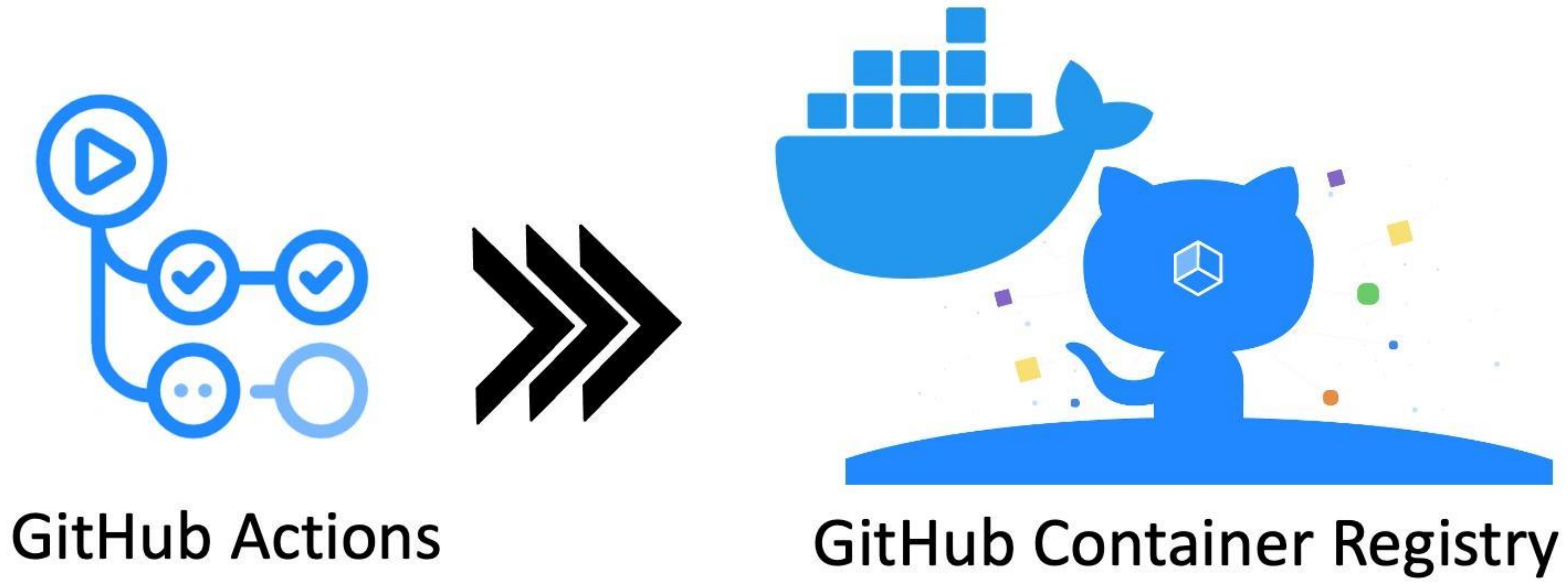
Automated builds

Manually pushing images to Docker Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. Read more about automated builds.

CI/CD in DevOps

❖ Github Container Registry (GHCR)



Ref: https://docs.github.com/en/packages

Ref: https://docs.github.com/en/packages/managing-github-packages-using-github-actions-workflows

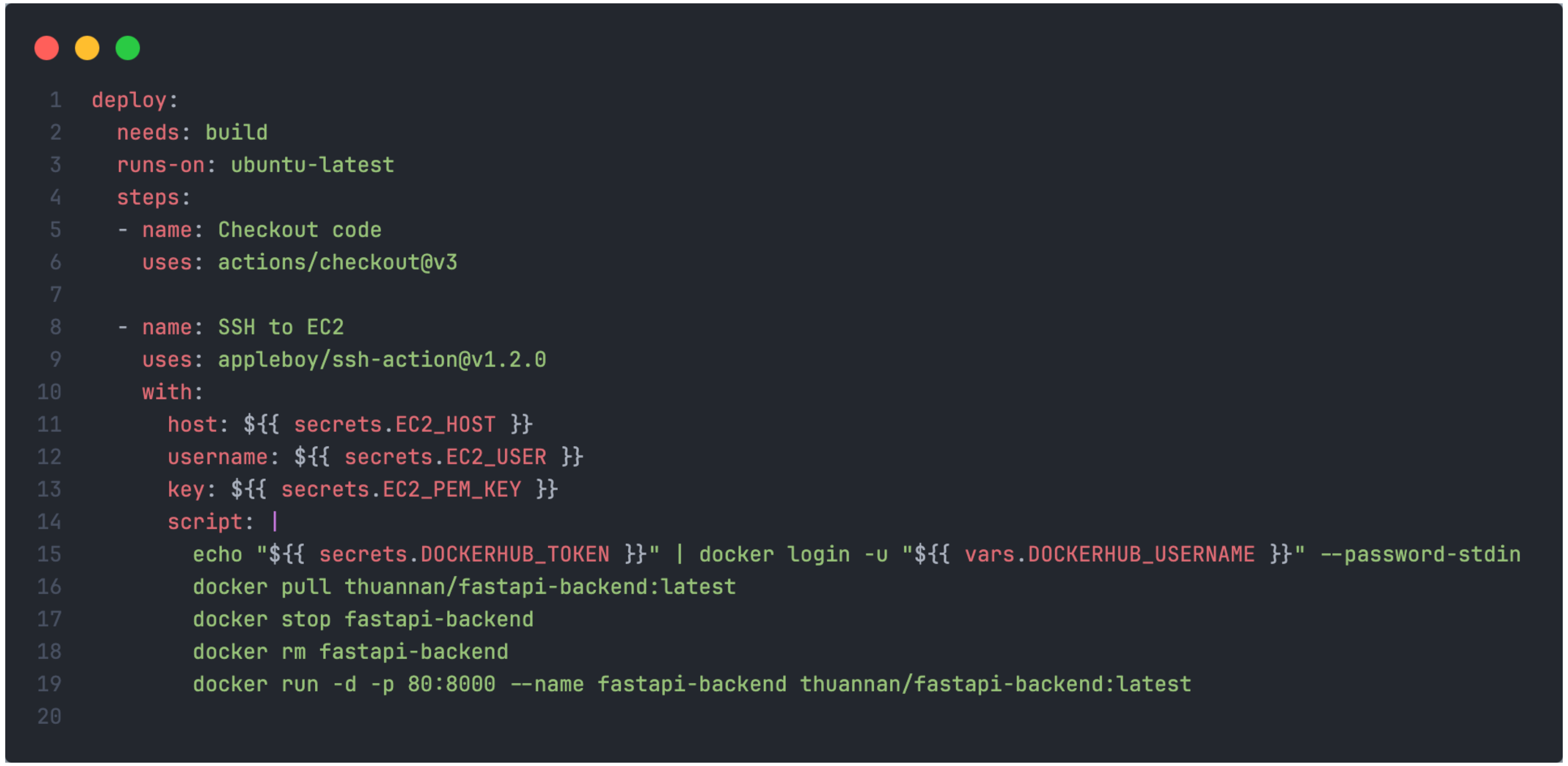
CD

AI VIETNAM
All-in-One Course
(TA Session)

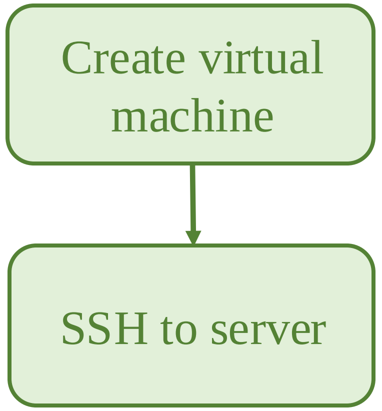
CI/CD in DevOps

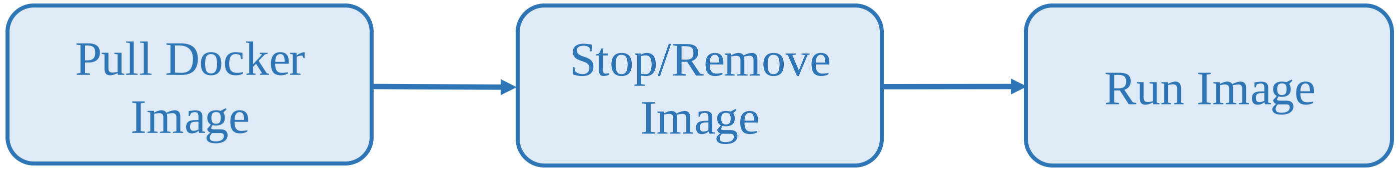
❖ Github Actions

Run pipeline when having a push/pull request the event to the main branch.



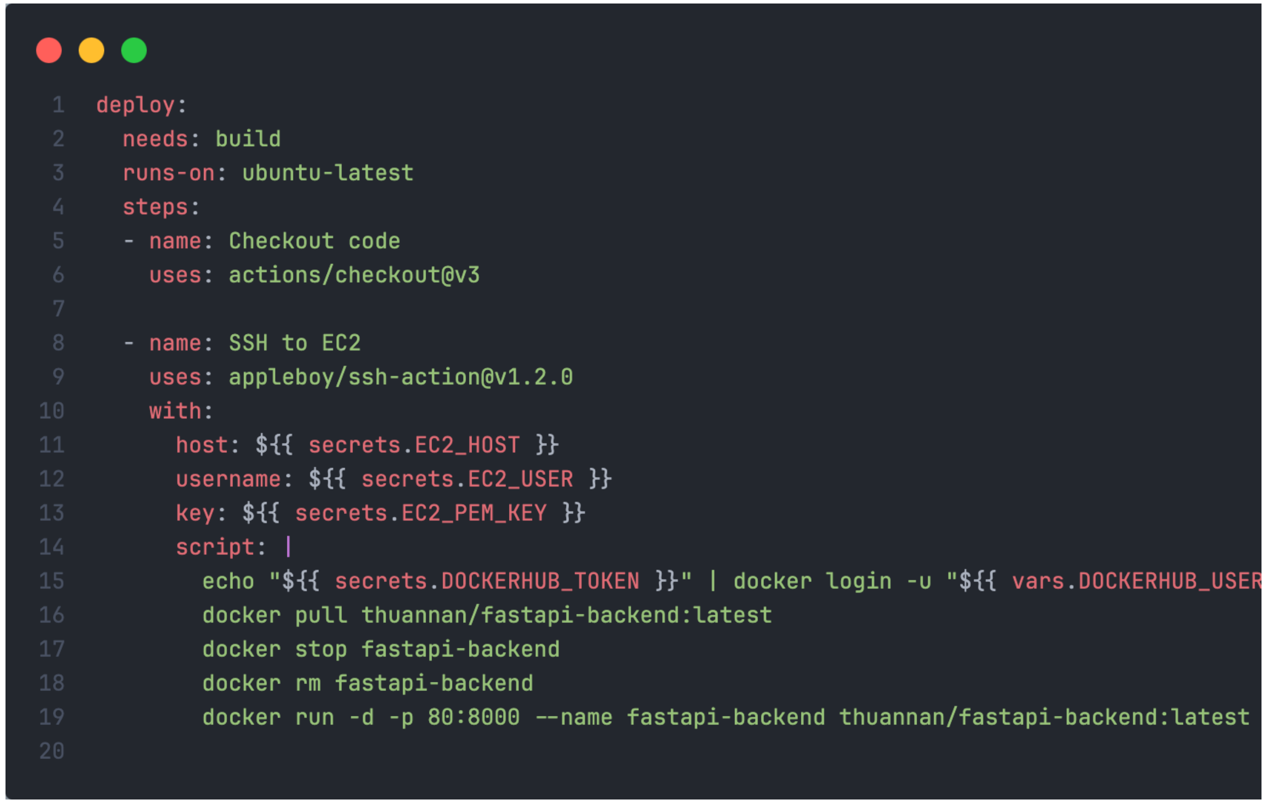
```
deploy:
  needs: build
  runs-on: ubuntu-latest
  steps:
  - name: Checkout code
    uses: actions/checkout@v3

  - name: SSH to EC2
    uses: appleboy/ssh-action@v1.2.0
    with:
      host: ${{ secrets.EC2_HOST }}
      username: ${{ secrets.EC2_USER }}
      key: ${{ secrets.EC2_PEM_KEY }}
      script: |
        echo "${{ secrets.DOCKERHUB_TOKEN }}" | docker login -u "${{ vars.DOCKERHUB_USERNAME }}" --password-stdin
        docker pull thuannan/fastapi-backend:latest
        docker stop fastapi-backend
        docker rm fastapi-backend
        docker run -d -p 80:8000 --name fastapi-backend thuannan/fastapi-backend:latest

```





CI/CD in DevOps

❖ Docker actions



```yaml
deploy:
  needs: build
  runs-on: ubuntu-latest
  steps:
  - name: Checkout code
    uses: actions/checkout@v3

  - name: SSH to EC2
    uses: appleboy/ssh-action@v1.2.0
    with:
      host: ${{ secrets.EC2_HOST }}
      username: ${{ secrets.EC2_USER }}
      key: ${{ secrets.EC2_PEM_KEY }}
      script: |
        echo "${{ secrets.DOCKERHUB_TOKEN }}" | docker login -u "${{ vars.DOCKERHUB_USER
        docker pull thuannan/fastapi-backend:latest
        docker stop fastapi-backend
        docker rm fastapi-backend
        docker run -d -p 80:8000 --name fastapi-backend thuannan/fastapi-backend:latest

```

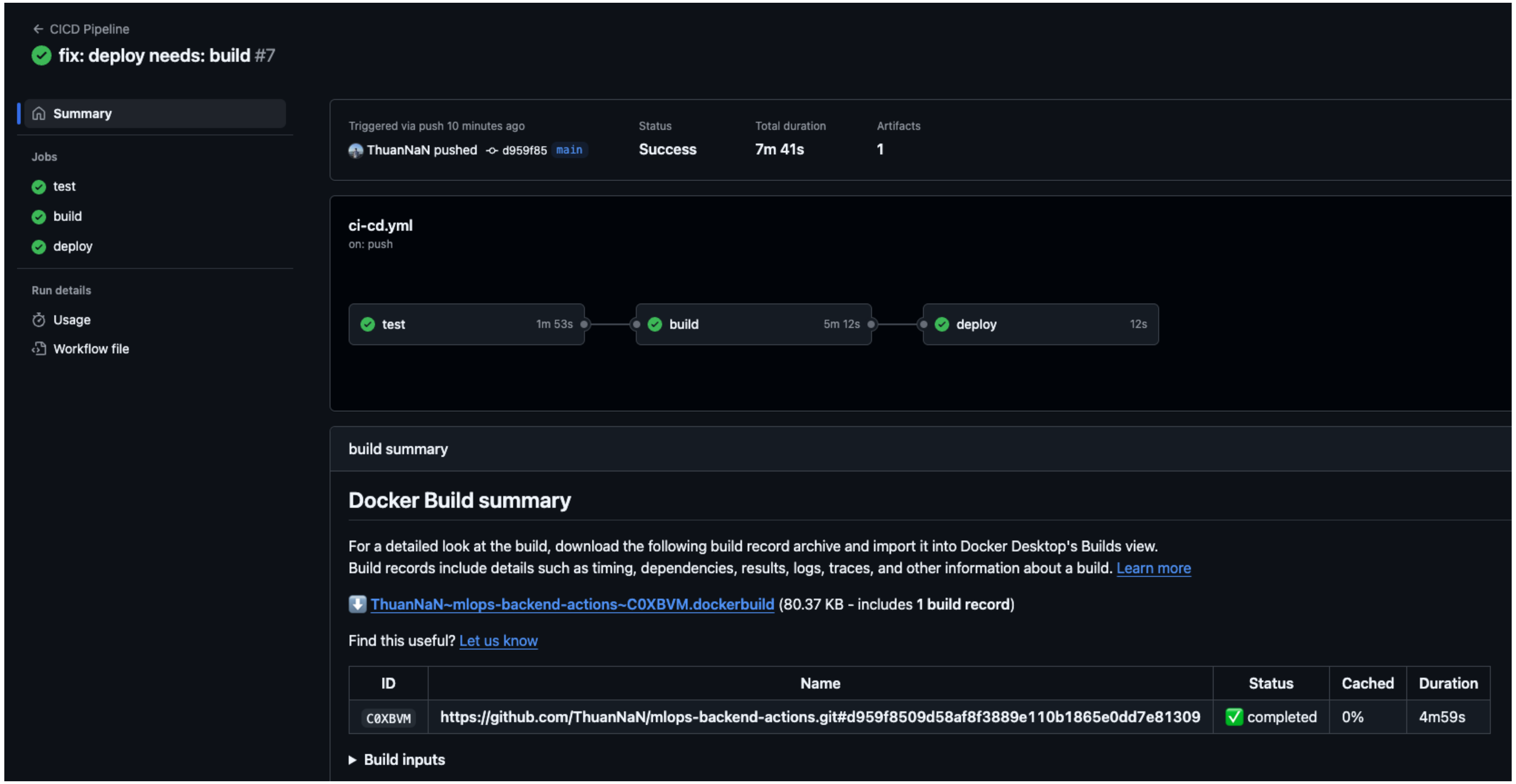
uses: actions/ssh-action@v1.2.0

Used to ssh to EC2 instance by private key

Ref: https://github.com/appleboy/ssh-action   Ref: https://github.com/marketplace/actions/ssh-remote-commands

CI/CD in DevOps

❖ Complete pipeline



← CICD Pipeline

fix: deploy needs: build #7

Summary

Jobs

- test
- build
- deploy

Run details

- Usage
- Workflow file

Triggered via push 10 minutes ago
ThuanNaN pushed d959f85 main

Status: Success
Total duration: 7m 41s
Artifacts: 1

ci-cd.yml
on: push

test 1m 53s → build 5m 12s → deploy 12s

build summary

Docker Build summary

For a detailed look at the build, download the following build record archive and import it into Docker Desktop's Builds view.
Build records include details such as timing, dependencies, results, logs, traces, and other information about a build. Learn more

ThuanNaN~mlops-backend-actions~C0XBVM.dockerbuild (80.37 KB - includes 1 build record)

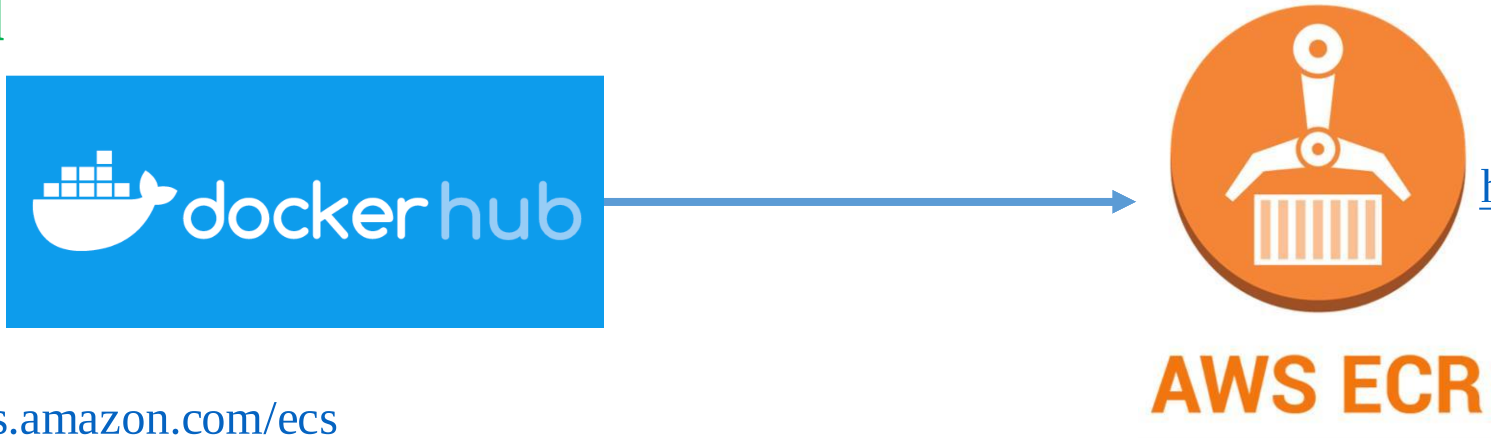
Find this useful? Let us know

ID	Name	Status	Cached	Duration
C0XBVM	https://github.com/ThuanNaN/mlops-backend-actions.git#d959f8509d58af8f3889e110b1865e0dd7e81309	✅ completed	0%	4m59s

▸ Build inputs

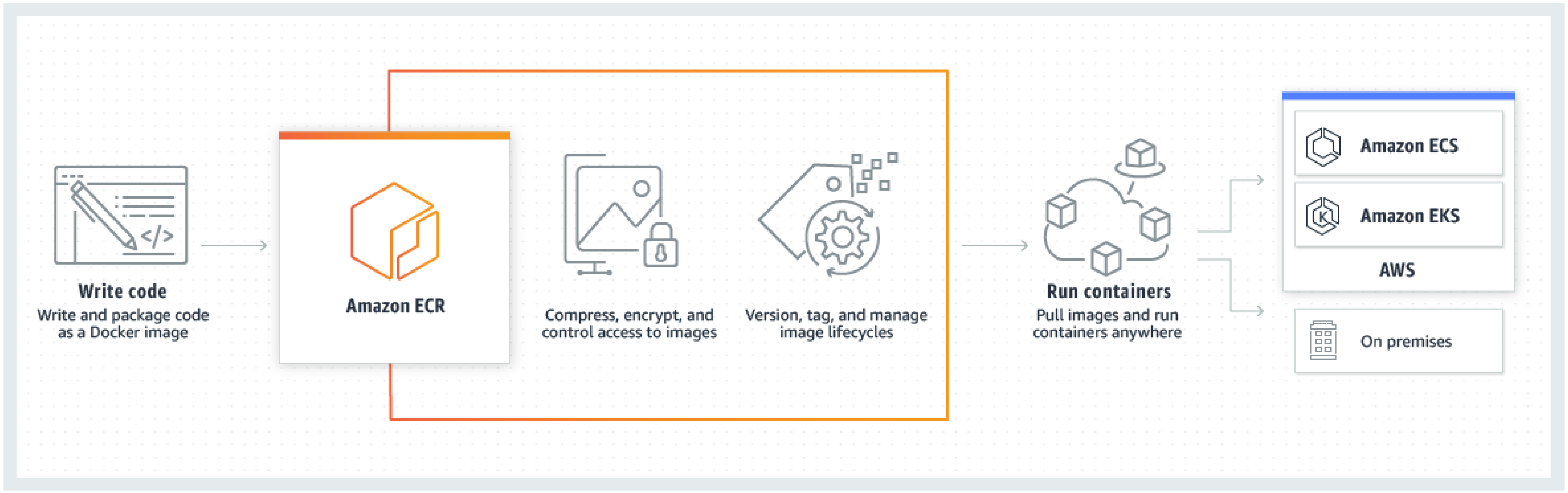
CI/CD in DevOps

❖ Optional



https://aws.amazon.com/ecr/

https://aws.amazon.com/ecs


Write code
Write and package code as a Docker image
Amazon ECR
Compress, encrypt, and control access to images
Version, tag, and manage image lifecycles
Run containers
Pull images and run containers anywhere
Amazon ECS
Amazon EKS
AWS
On premises

https://docs.github.com/en/actions/use-cases-and-examples/deploying/deploying-to-amazon-elastic-container-service

CI/CD in MLOps

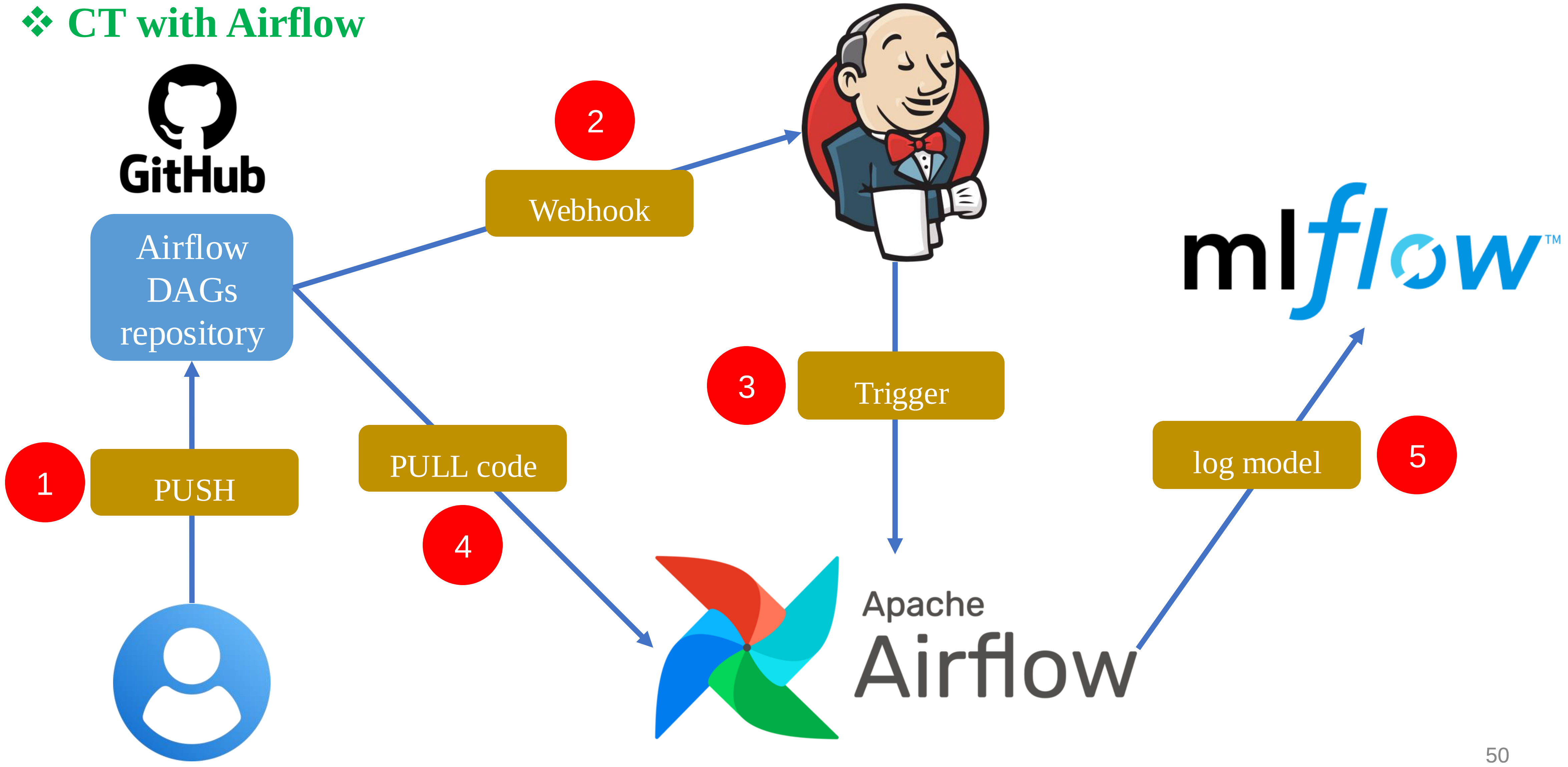
CI/CD in MLOps

❖ Getting Started

CI	Testing API	Packaging
CD	Deploy API	
CT	Data pipeline	Training pipeline

CI/CD in MLOps

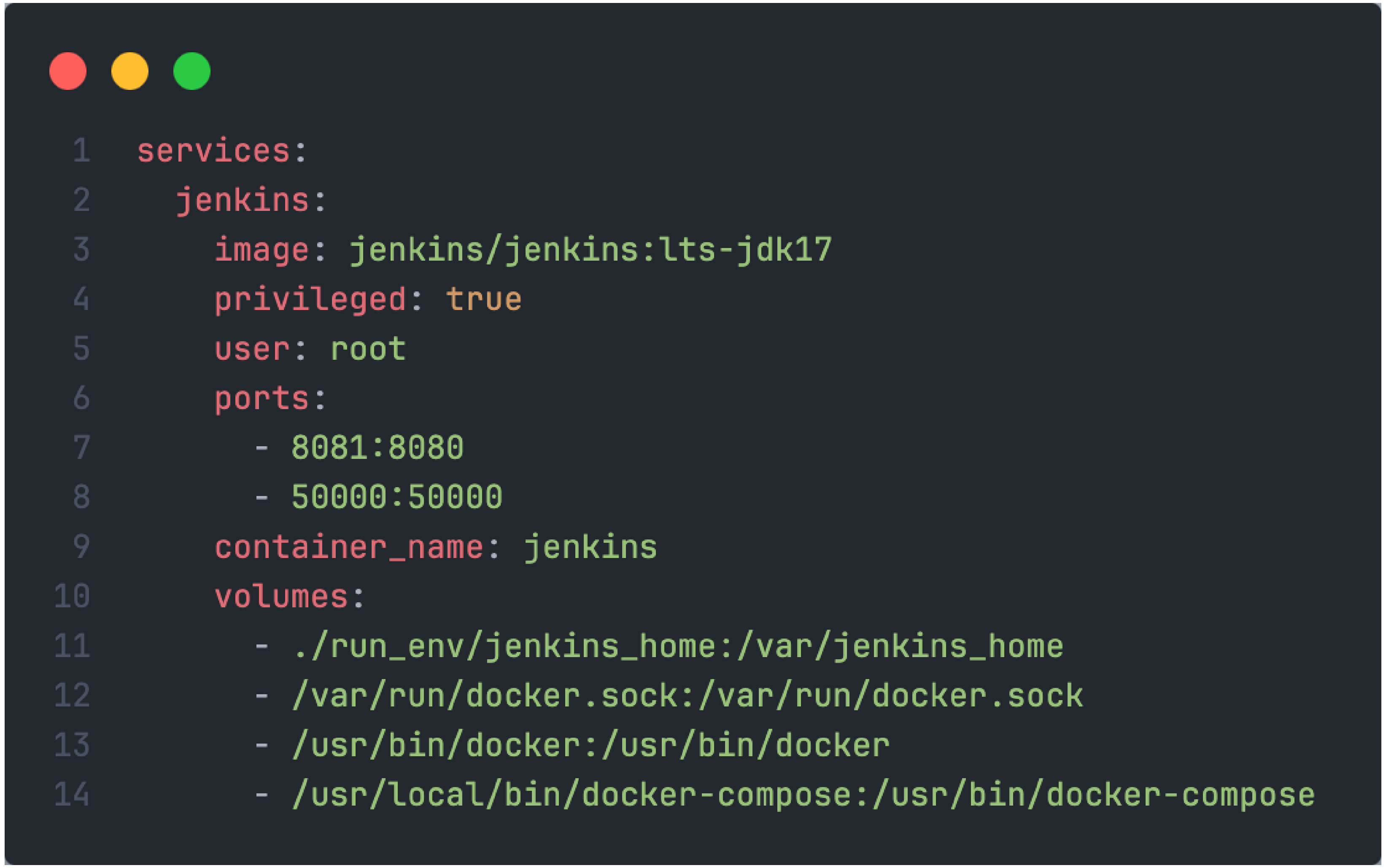
❖ CT with Airflow



Practices

CI/CD in MLOps

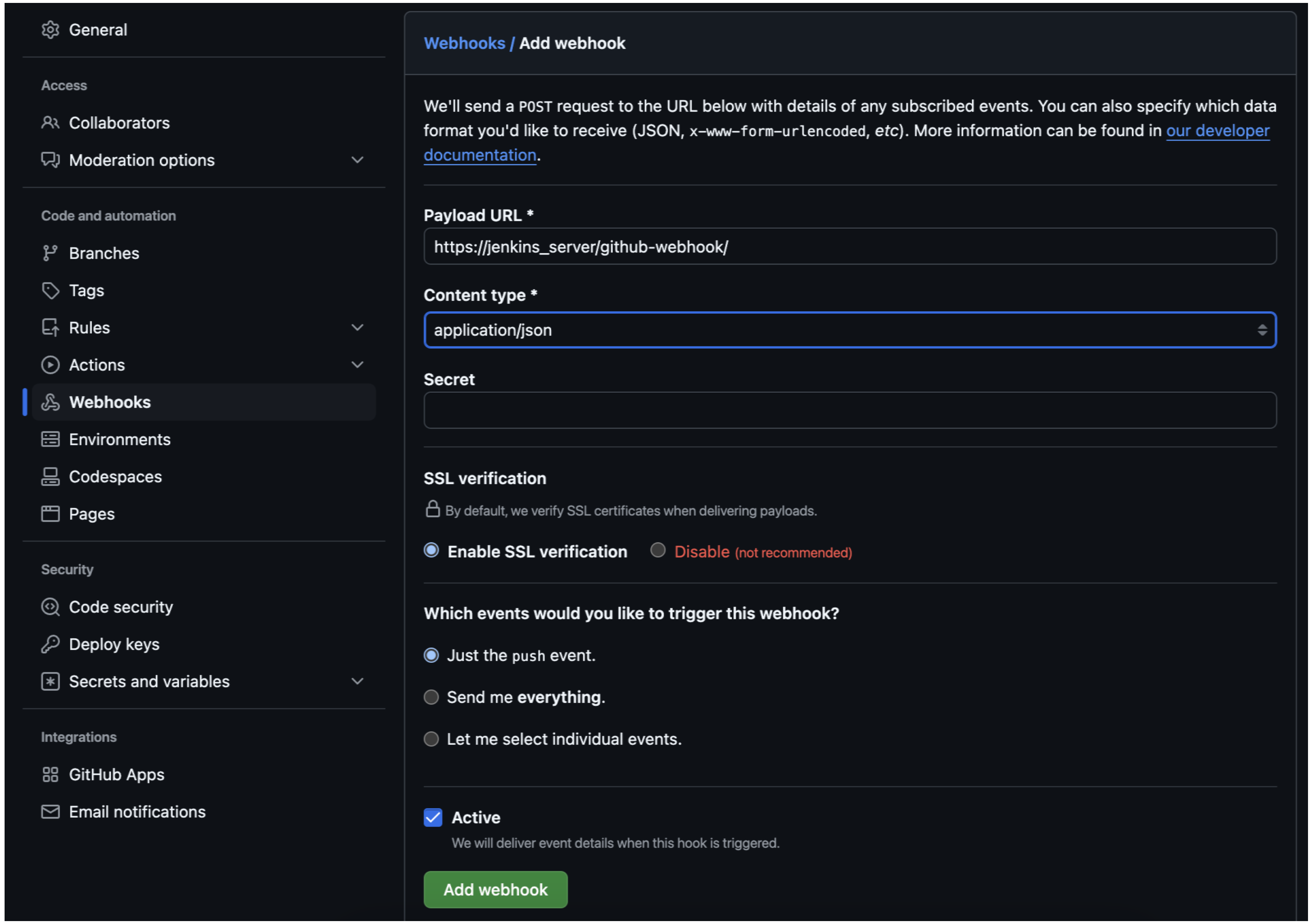
❖ Installation



```yaml
services:
  jenkins:
    image: jenkins/jenkins:lts-jdk17
    privileged: true
    user: root
    ports:
      - 8081:8080
      - 50000:50000
    container_name: jenkins
    volumes:
      - ./run_env/jenkins_home:/var/jenkins_home
      - /var/run/docker.sock:/var/run/docker.sock
      - /usr/bin/docker:/usr/bin/docker
      - /usr/local/bin/docker-compose:/usr/bin/docker-compose
```

CI/CD in MLOps

❖ Github Webhooks



Webhooks / Add webhook

We'll send a `POST` request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, `x-www-form-urlencoded`, *etc*). More information can be found in our developer documentation.

Payload URL *
https://jenkins_server/github-webhook/

Content type *
application/json

Secret

SSL verification
By default, we verify SSL certificates when delivering payloads.
[✓ Enable SSL verification] Disable (not recommended)

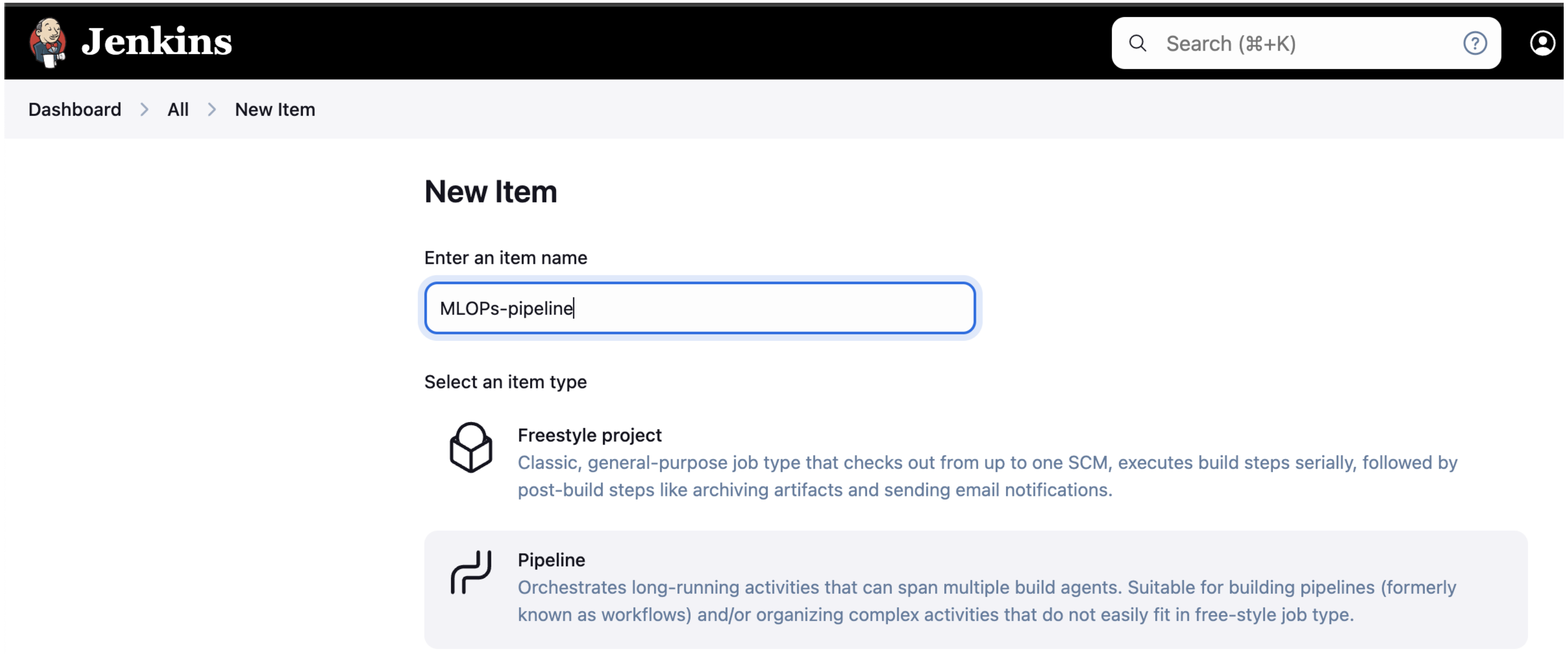
Which events would you like to trigger this webhook?
[✓ Just the `push` event.]
Send me **everything**.
Let me select individual events.

[✓ **Active**]
We will deliver event details when this hook is triggered.

Add webhook

CI/CD in MLOps

❖ Config Jenkins



New Item

Enter an item name

MLOPs-pipeline

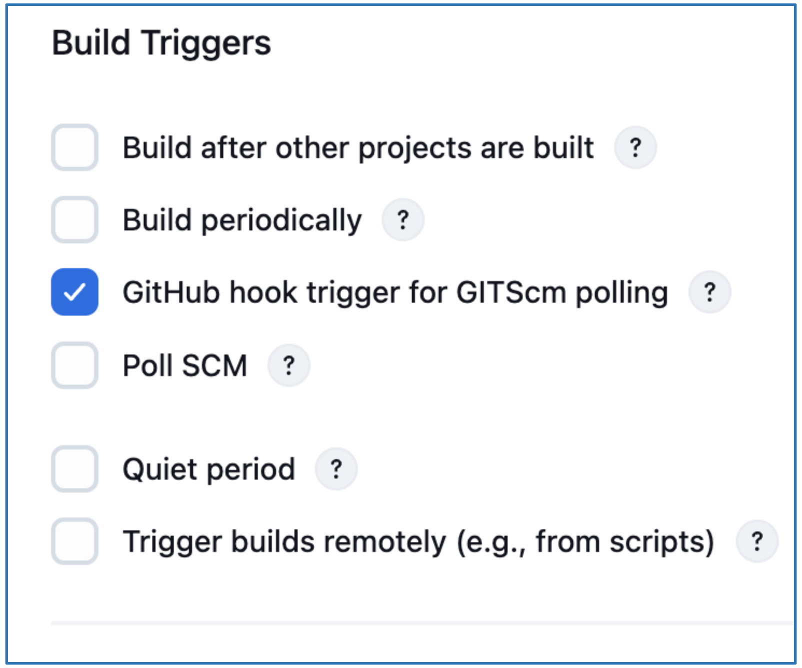
Select an item type

Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

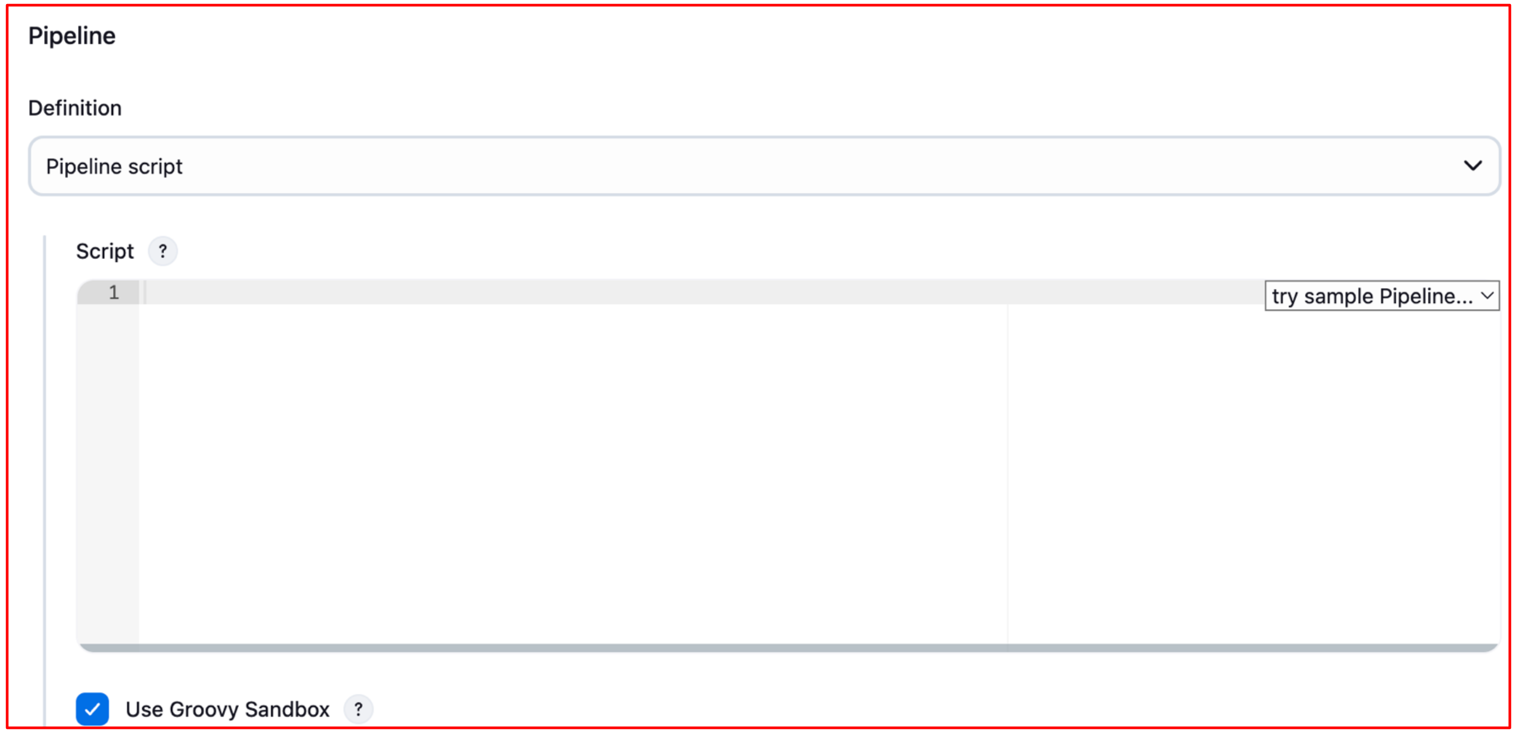
CI/CD in MLOps

❖ Config Jenkins



Build Triggers

- [ ] Build after other projects are built ?
- [ ] Build periodically ?
- [✓] GitHub hook trigger for GITScm polling ?
- [ ] Poll SCM ?
- [ ] Quiet period ?
- [ ] Trigger builds remotely (e.g., from scripts) ?



Pipeline

Definition

Pipeline script

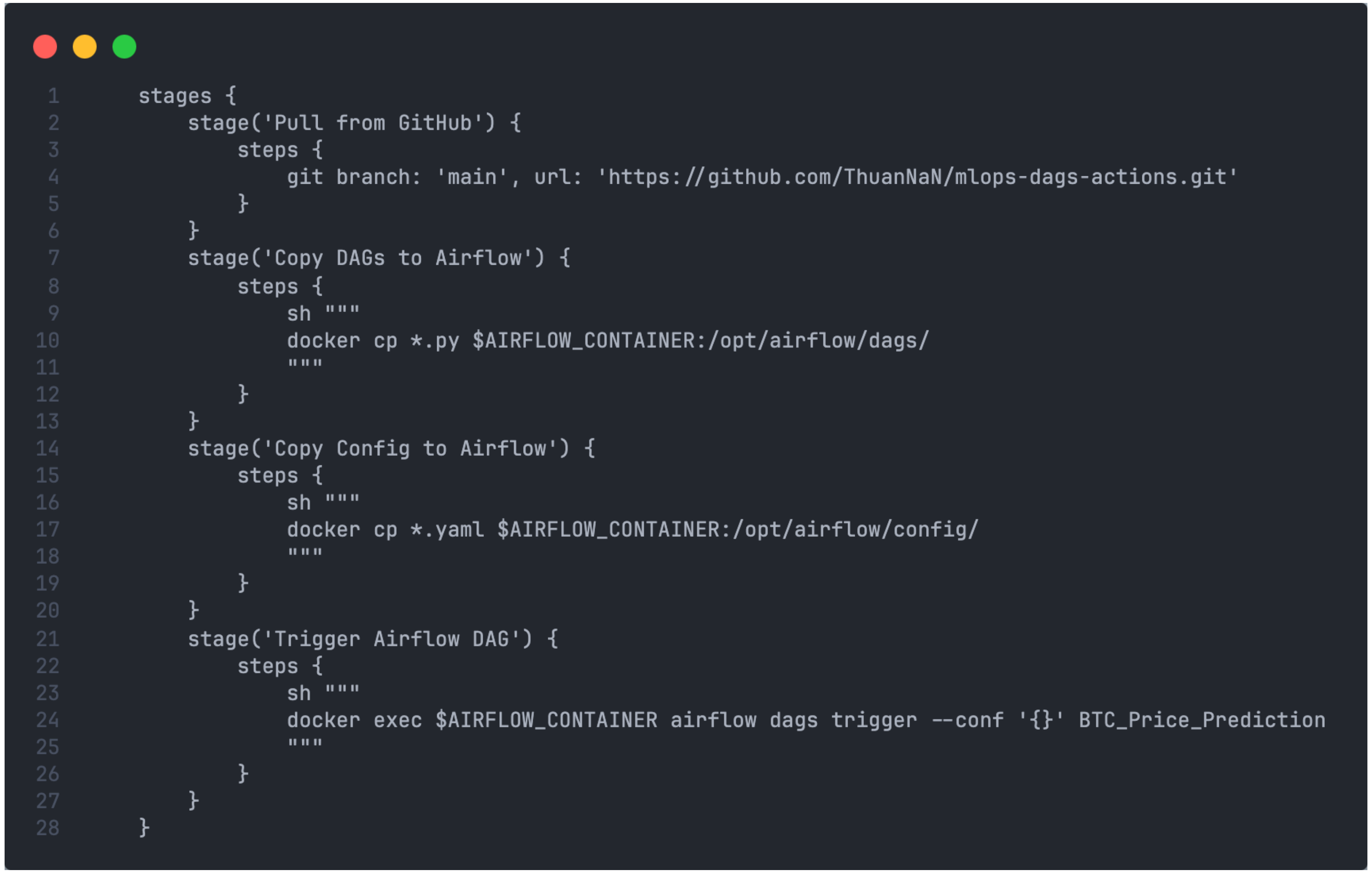
Script ?

try sample Pipeline...

- [✓] Use Groovy Sandbox ?

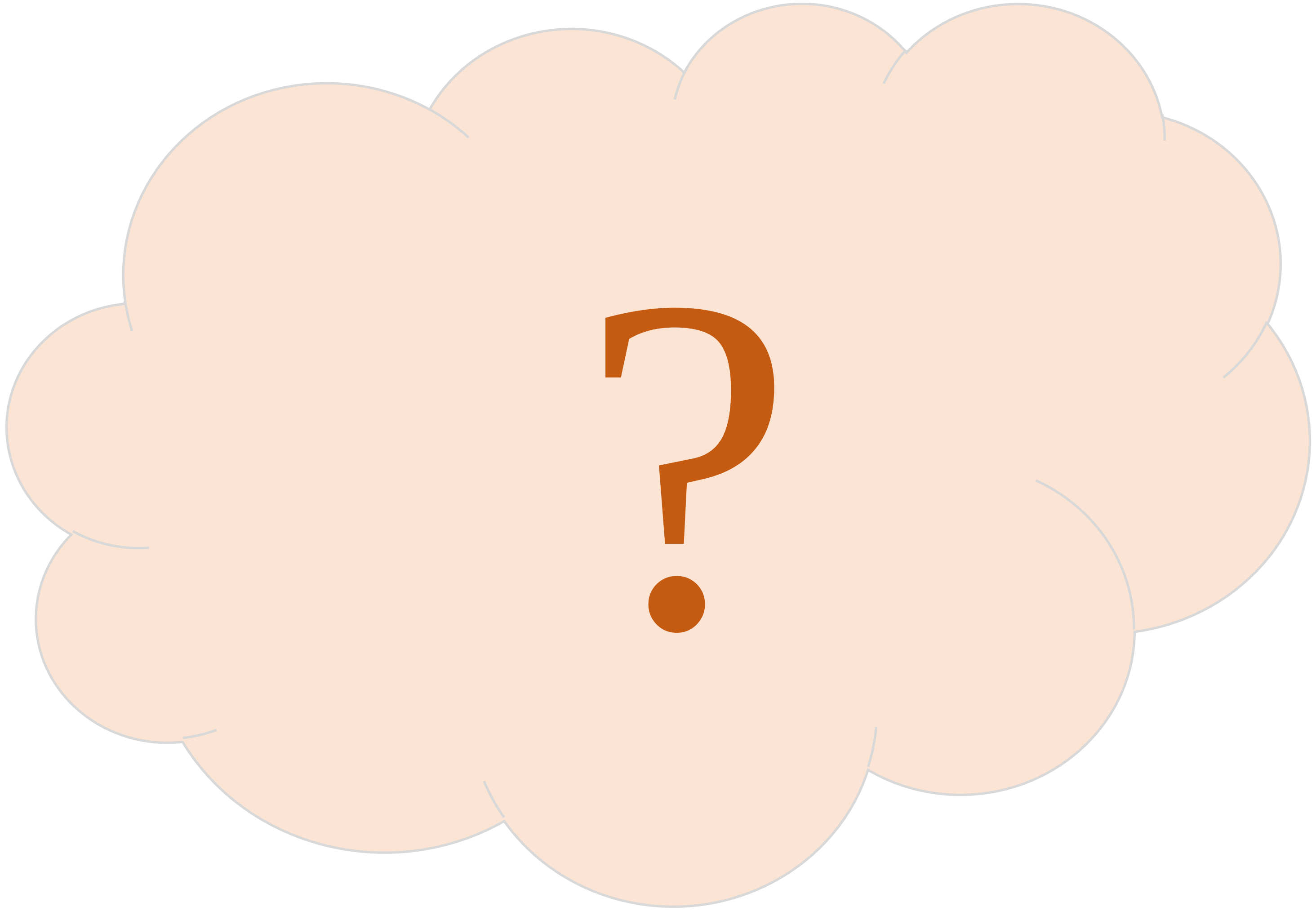
CI/CD in MLOps

❖ Pipeline



```
stages {
    stage('Pull from GitHub') {
        steps {
            git branch: 'main', url: 'https://github.com/ThuanNaN/mlops-dags-actions.git'
        }
    }
    stage('Copy DAGs to Airflow') {
        steps {
            sh """
            docker cp *.py $AIRFLOW_CONTAINER:/opt/airflow/dags/
            """
        }
    }
    stage('Copy Config to Airflow') {
        steps {
            sh """
            docker cp *.yaml $AIRFLOW_CONTAINER:/opt/airflow/config/
            """
        }
    }
    stage('Trigger Airflow DAG') {
        steps {
            sh """
            docker exec $AIRFLOW_CONTAINER airflow dags trigger --conf '{}' BTC_Price_Prediction
            """
        }
    }
}
```

Question



Thank you!