

# CI/CD Pipeline

FH OÖ Campus Hagenberg  
Data Science und Engineering  
2023-01-30

Sophie Bauernfeind, Fabian Jetzinger

# Agenda

- Overview
  - Tools
  - Automated doctests
  - Automated docker build
  - Automated microservice deployment to GCP
- Live Demo
- Lessons Learned
- Q & A

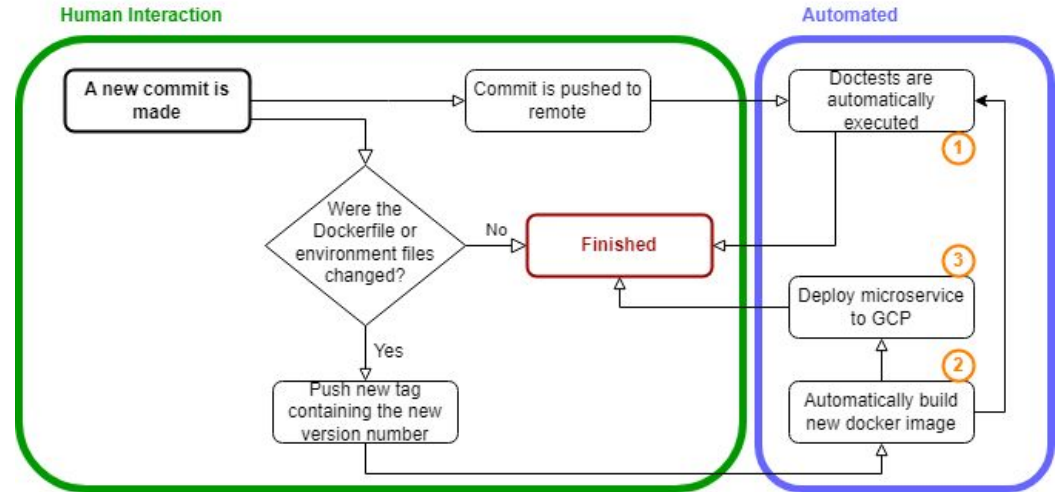
# Project Overview

Build a CI/CD template for our Computer Vision project (WIP)

Become more familiar with GitHub Actions & co. → useful experience

Steps:

- Automated doctests
- Automated docker build
- Automated microservice deployment to GCP



# Tools

- Python
  - Pytest, flask, gunicorn, pytorch, torchvision
- Poetry
- GitHub Actions
- Docker
- Google Cloud

# Research

- GitHub Documentation on workflows
- Tutorials on GitHub Actions, Google Cloud, Flask, etc.

## Dockerfiles

### base-dockerfile

```
FROM python:3.10-slim
RUN pip install --upgrade pip==22.2.2
ENV POETRY VERSION=1.2.1
RUN pip install "poetry==$POETRY_VERSION"
COPY pyproject.toml poetry.lock .
RUN pip install setuptools --upgrade
RUN poetry config virtualenvs.create false
RUN poetry install --no-dev --no-root
RUN poetry run poe force-torch-cpu
RUN poetry run poe force-torchvision-cpu
COPY . /src
```

### test-dockerfile

```
FROM fabianjetzinger/dalle-image-classification:base-latest
RUN poetry install --no-root
```

### service-dockerfile

```
FROM fabianjetzinger/dalle-image-classification:base-latest
CMD exec gunicorn --bind :$PORT --workers 1 --threads 8 --timeout 0
/src/microservice/image_classifier_service:app
```

# Automated doctests

## Trigger:

- every push
- workflow call

## Steps:

- uses the test-image
- run all tests

```
1 ▶ Run pytest -ra --doctest-modules --junitxml=python-junit.xml --cov --cov-report=xml --cov-report term
4 ===== test session starts =====
5 platform linux -- Python 3.10.9, pytest-7.2.1, pluggy-1.0.0
6 rootdir: /_w/dalle_inpainting_classification/dalle_inpainting_classification
7 plugins: cov-4.0.0, xdist-3.1.0
8 collected 2 items
9
10 test/demo_test.py .. [100%]
11
12 - generated xml file: /_w/dalle_inpainting_classification/dalle_inpainting_classification/python-junit.xml -
13
14 ----- coverage: platform linux, python 3.10.9-final-0 -----
15
16 Name                               Stmts  Miss  Cover
17 -----
18 src/__init__.py                     0      0  100%
19 src/microservice/__init__.py        0      0  100%
20 src/microservice/image_classifier_service.py 27     12   56%
21 test/__init__.py                     0      0  100%
22 test/demo_test.py                   17      2   88%
23 -----
24 TOTAL                               44     14   68%
25 Coverage XML written to file coverage.xml
26
27 ===== 2 passed in 1.85s =====
```

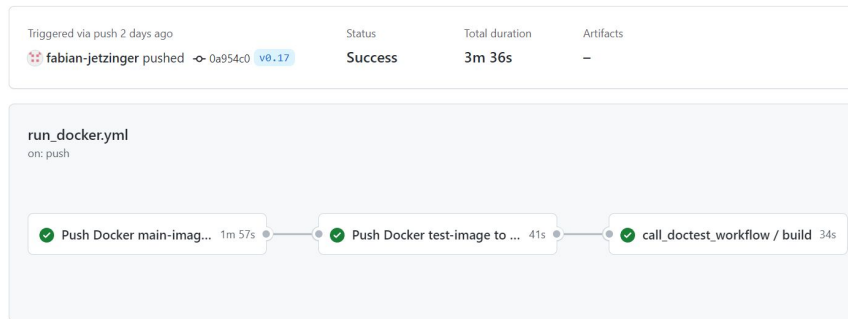
# Automated docker build

## Trigger:

- push of a version tag (eg. v0.1, v1.3)

## Steps:

- authenticate on docker hub
- builds the base image and pushes to docker hub
- builds the test image and pushes to docker hub
- triggers the automated doctests workflow



# Automated microservice deployment to GCP



## Setup in Google Cloud Run, not GitHub Actions

- ☒ Continuously deploy new revisions from a source repository

### Source repository

[http://github.com/fabian-jetzinger/dalle\\_inpainting\\_classification](http://github.com/fabian-jetzinger/dalle_inpainting_classification)

### Cloud Build trigger

A Cloud Build trigger will be created in order to automatically build and deploy your code.

### Repository \*

fabian-jetzinger/dalle\_inpainting\_classification (GitHub App) ▼

Select the repository to watch for events and clone when the trigger is invoked

### Tag \*

^v\*.\*\$

Trigger only for a tag that matches the given regular expression [Learn more](#)

☐ Invert Regex

Matches 17 tags: v0.1, v0.2, v0.3, v0.4, v0.5, ...



# Live Demo

# Lessons learned

- Full CI/CD pipeline is great for efficient and comfortable development
- GitHub Actions *workflow\_run* trigger does not seem to work as intended
- Error messages from workflows are not always useful
- (Automated) Cloud deployment is not trivial
- Poetry is great

# Questions?