SEMI-SUPERVISED IMAGE CLUSTERING WITH CONVOLUTIONAL AUTOENCODER

SOFTWARE ENGINEERING FOR AI-ENABLED SYSTEMS

NICOLA FIORENTINO

I. INCEPTION

Model Card

Dataset Card

Cookiecutter Data Science

II. REPRODUCIBILITY: DVC & DAGSHUB

"Git for data"

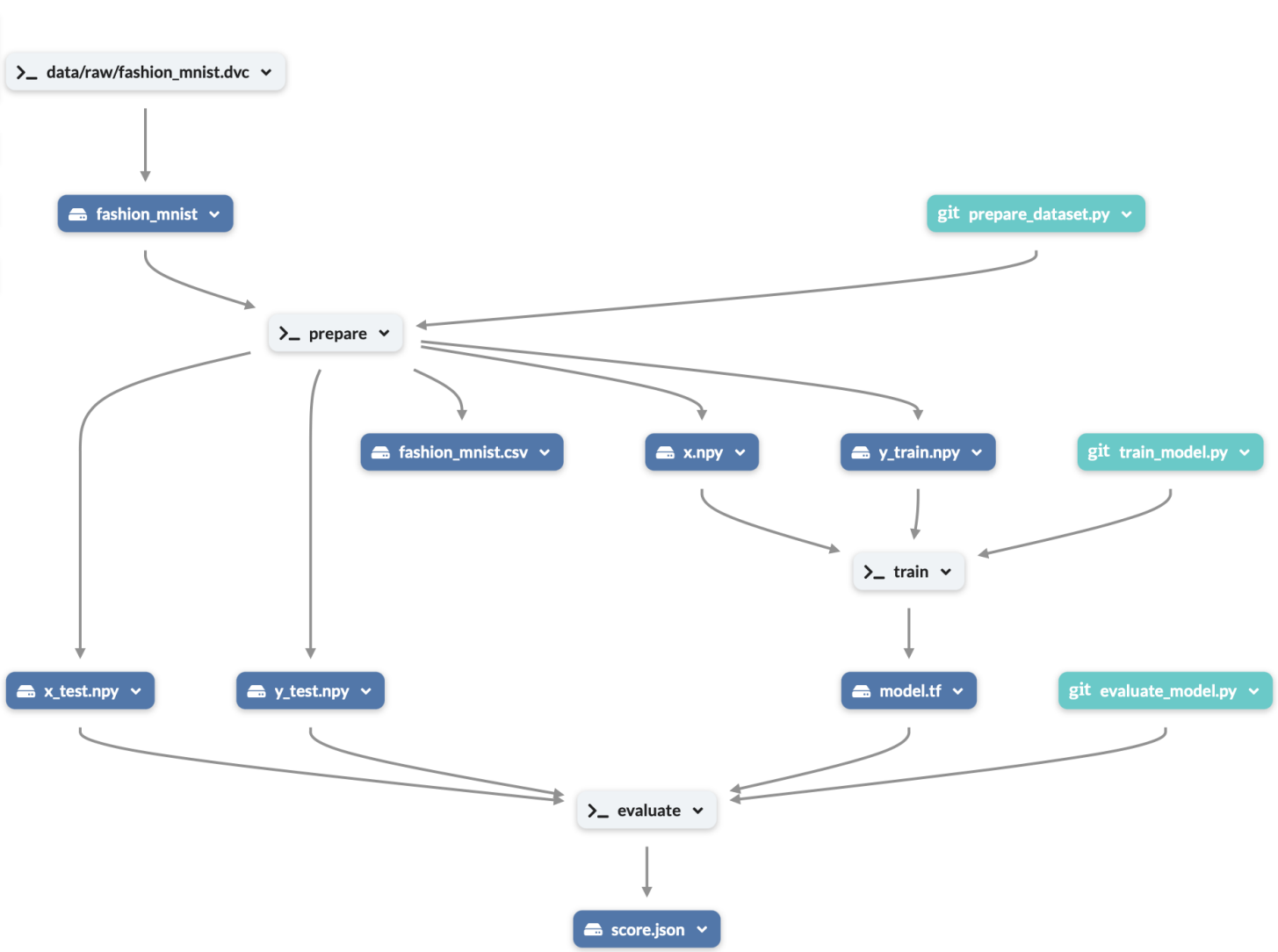
```
dvc init
git commit -m "Initialize DVC"

dvc add data/raw/fashion_mnist
git add data/raw/.gitignore data/raw/fashion_mnist.dvc
git commit -m "Add raw data"

dvc remote add origin https://dagshub.com/nico-fi/SemiSupervised-DCEC.dvc

git add .dvc/config
git commit -m "Configure remote storage"

dvc push
```

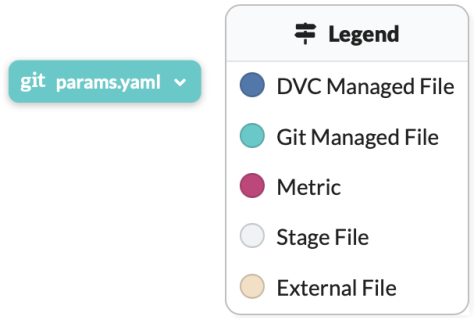
II. REPRODUCIBILITY: DVC PIPELINE

- Prepare Dataset
- Train Model
- Evaluate Model

`dvc.yaml`

II. REPRODUCIBILITY: DVC PIPELINE

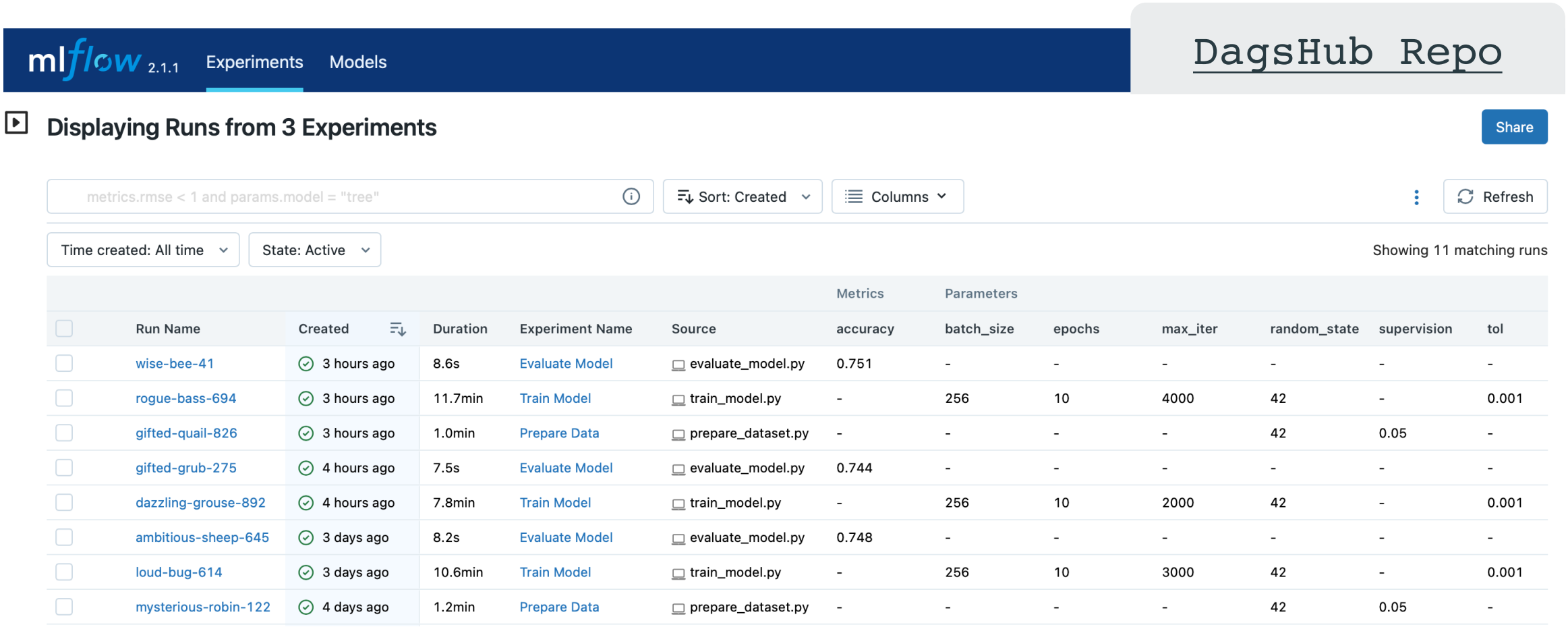




params.yaml

```yaml
prepare:
  supervision: 0.05
  random_state: 42
train:
  batch_size: 256
  epochs: 10
  max_iter: 4000
  tol: 0.001
  random_state: 42
```

II. REPRODUCIBILITY: MLFLOW & DAGSHUB



						Metrics	Parameters					
	Run Name	Created	Duration	Experiment Name	Source	accuracy	batch_size	epochs	max_iter	random_state	supervision	tol
	wise-bee-41	3 hours ago	8.6s	Evaluate Model	evaluate_model.py	0.751	-	-	-	-	-	-
	rogue-bass-694	3 hours ago	11.7min	Train Model	train_model.py	-	256	10	4000	42	-	0.001
	gifted-quail-826	3 hours ago	1.0min	Prepare Data	prepare_dataset.py	-	-	-	-	42	0.05	-
	gifted-grub-275	4 hours ago	7.5s	Evaluate Model	evaluate_model.py	0.744	-	-	-	-	-	-
	dazzling-grouse-892	4 hours ago	7.8min	Train Model	train_model.py	-	256	10	2000	42	-	0.001
	ambitious-sheep-645	3 days ago	8.2s	Evaluate Model	evaluate_model.py	0.748	-	-	-	-	-	-
	loud-bug-614	3 days ago	10.6min	Train Model	train_model.py	-	256	10	3000	42	-	0.001
	mysterious-robin-122	4 days ago	1.2min	Prepare Data	prepare_dataset.py	-	-	-	-	42	0.05	-

III. QUALITY ASSURANCE: CODE

Pylint

```
pylint src

Your code has been rated at 9.82/10 (previous run: 6.52/10, +3.30)
```

Pytest

```
pytest --cov-report html --cov=src \
tests/test_prepare_dataset.py \
tests/test_train_model.py \
tests/test_evaluate_model.py
================== 5 passed in 783.87s (0:13:03) ==================
```

III. QUALITY ASSURANCE: COVERAGE

`Pytest-cov`

Coverage report: 99%

coverage.py v7.0.5, created at 2023-02-04 22:02 +0100

Module ↑	*statements*	*missing*	*excluded*	*coverage*
src/__init__.py	0	0	0	100%
src/data/__init__.py	0	0	0	100%
src/data/prepare_dataset.py	47	0	2	100%
src/models/__init__.py	0	0	0	100%
src/models/evaluate_model.py	22	0	2	100%
src/models/train_model.py	88	2	2	98%
Total	**157**	**2**	**6**	**99%**

III. QUALITY ASSURANCE: DATA

Images → **CSV** →

Expectations

- Column order
- Null values
- Type adherence
- Minimum instances
- Aspect ratio
- Image format
- Range for pixels mean
- Range for pixels SD
- Number of classes
- Class values

Great Expectations

```
great_expectations init
great_expectations datasource new
great_expectations suite new
great_expectations checkpoint new fashion_mnist
great_expectations checkpoint run fashion_mnist
great_expectations docs build
```

III. QUALITY ASSURANCE: DATA

Overview

Expectation Suite: fashion_mnist
Data asset: None
Status: ✅ Succeeded

Statistics

Evaluated Expectations	20
Successful Expectations	20
Unsuccessful Expectations	0
Success Percent	100%

Show more info...

Table-Level Expectations

Search

Status	Expectation	Observed Value
✅	Must have these columns in this order: format, height, width, mean, std, label	['format', 'height', 'width', 'mean', 'std', 'label']
✅	Must have greater than or equal to 50000 rows.	70000
✅	Values in height and width must always be equal.	0% unexpected

III. QUALITY ASSURANCE: BEHAVIORAL TEST

Invariance

Same output

Directional

Different outputs

Minimum functionality

Bag

Sneaker

T-shirt

pytest

```
pytest
tests/test_behavioral.py
=== 3 passed in 7.86s ===
```

10 samples: 70% threshold

IV. API

FastAPI

```
@app.on_event('startup')        -> Load model, parameters and metrics
@app.get('/')                   -> Health check
@app.get('/model/parameters')   -> Get training parameters
@app.get('/model/metrics')      -> Get evaluation metrics
@app.post('/model')             -> Classify an image
```

/docs

Uvicorn

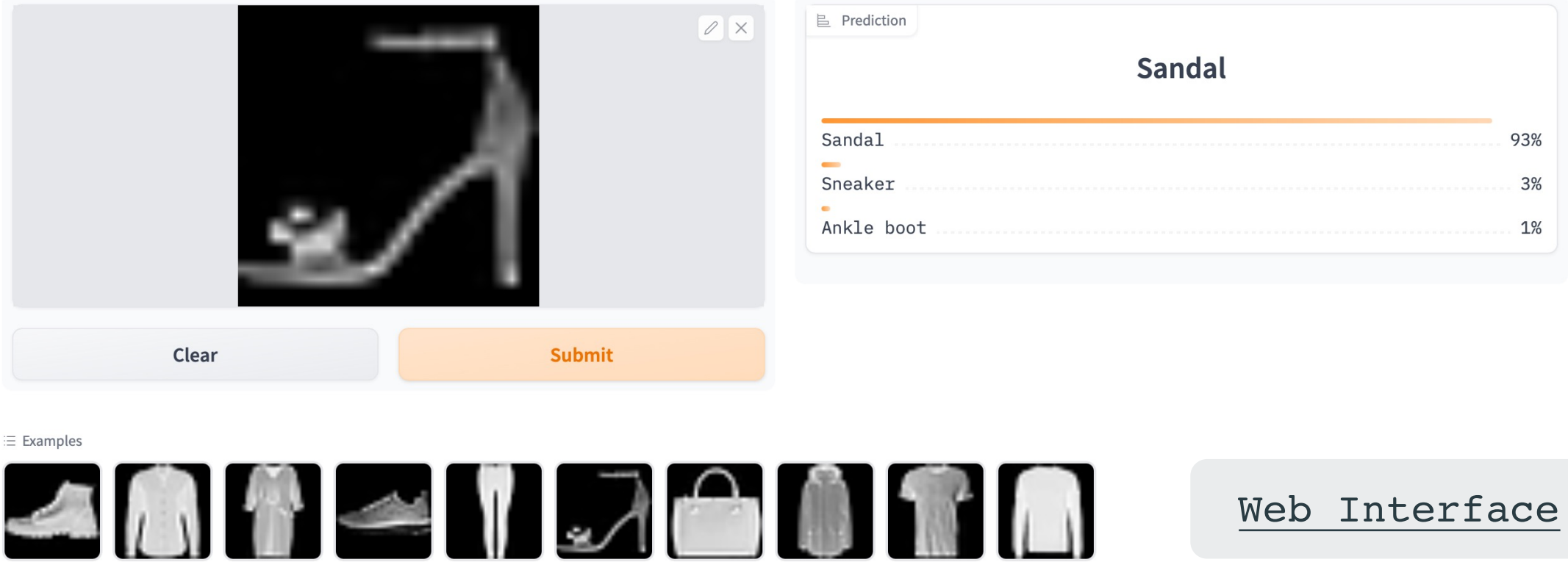
```
uvicorn app.api:app \
 --host 0.0.0.0 \
 --port 5000 \
 --reload \
 --reload-dir app \
 --reload-dir models
```

Pytest

```
pytest tests/test_api.py

=== 5 passed in 8.61s ===
```

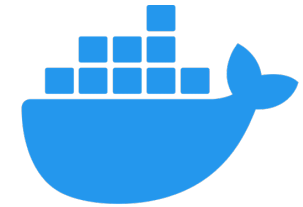
pytest

IV. API: GRADIO



Web Interface

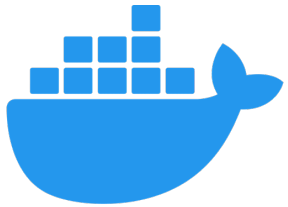
V. DEPLOYMENT: DOCKER



API Dockerfile

```
FROM  tensorflow/tensorflow:2.11.0
WORKDIR /app

COPY app/requirements.txt ./
RUN pip install -U pip && pip install --no-cache-dir -r requirements.txt

COPY app/api.py ./
COPY params.yaml ./
COPY models/model.tf ./models/model.tf
COPY models/score.json ./models/score.json

EXPOSE 5000

CMD [ "uvicorn", "api:app", "--host", "0.0.0.0", "--port", "5000" ]
```

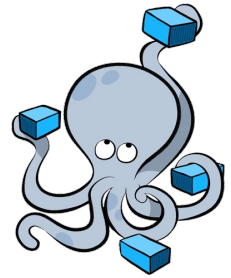
V. DEPLOYMENT: DOCKER



Web App Dockerfile

```
FROM python:3-slim
WORKDIR /usr/src/app

COPY web_app/requirements.txt ./
RUN pip install -U pip && pip install --no-cache-dir -r requirements.txt

COPY web_app/web_app.py ./
COPY data/samples ./data/samples

EXPOSE 4000

CMD [ "python", "./web_app.py" ]
```

V. DEPLOYMENT: DOCKER COMPOSE



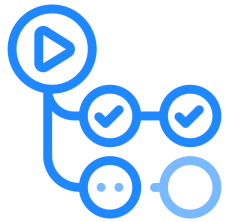
Main services

```
api:
  build:
    context: .
    dockerfile: app/Dockerfile
  container_name: api
  ports:
    - '5000:5000'
```

```
web-app:
  build:
    context: .
    dockerfile: web_app/Dockerfile
  container_name: web-app
  ports:
    - '4000:4000'
  depends_on:
    - api
```

Additional monitoring services

V. DEPLOYMENT: GITHUB ACTIONS



First workflow

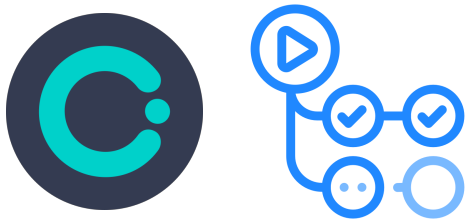
```yaml
name: Quality Assurance

on:
  push:
    branches:
      - main
  pull_request:
    branches:
      - main
  workflow_dispatch:

jobs:
```

Lint → Score ≥ 9

Great Expectations → Meet expectations

Test → Satisfy tests
Coverage ≥ 90%

V. DEPLOYMENT: GITHUB ACTIONS



Second workflow

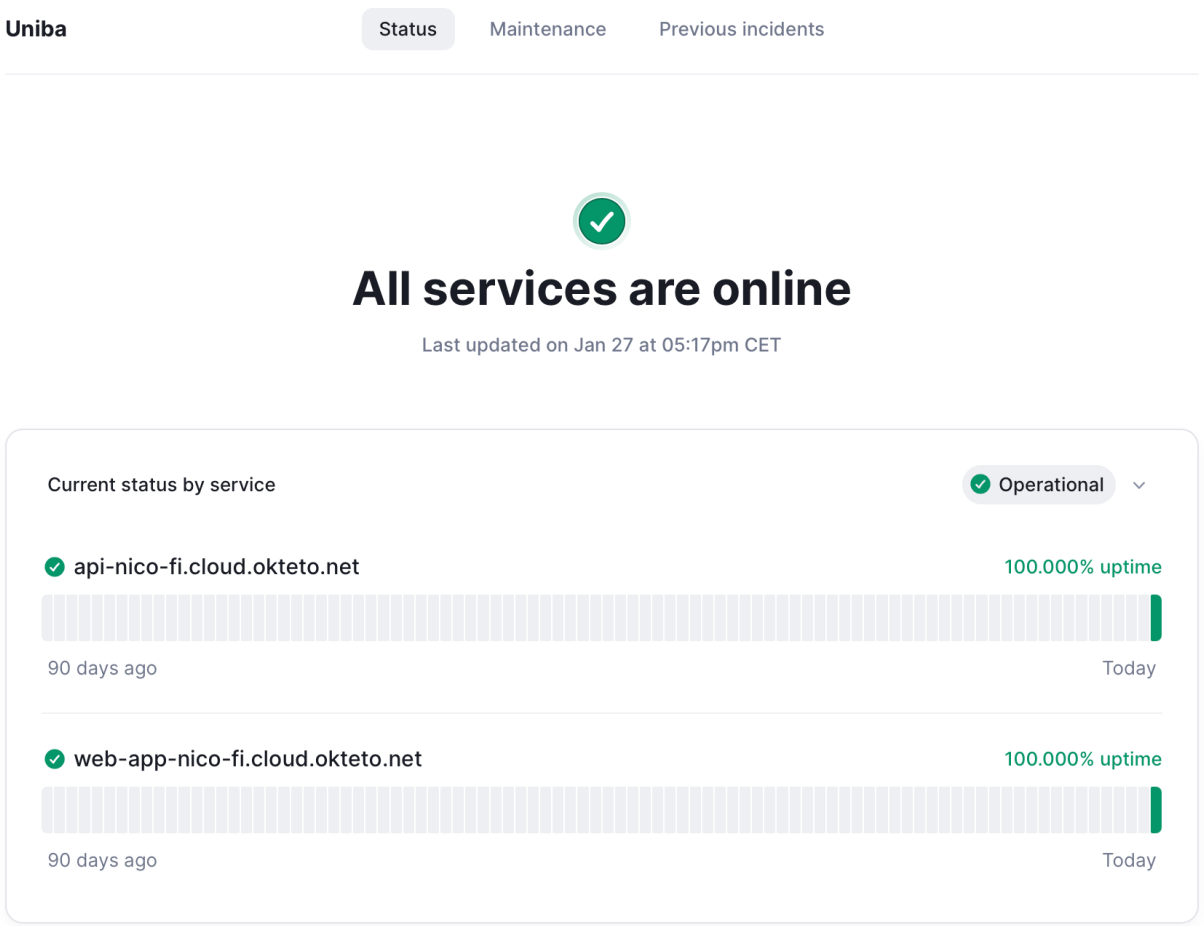
```
name: Deployment

on:
  push:
    branches:
      - main
  pull_request:
    branches:
      - main
    types:
      - closed
  workflow_dispatch:
jobs:
```

Deploy with Okteto

```
curl https://get.okteto.com -sSfL | sh

okteto context use https://cloud.okteto.com \
   --token ${{ secrets.OKTETO_TOKEN }}

okteto build
okteto deploy
```

Actions

VI. MONITORING: BETTER UPTIME

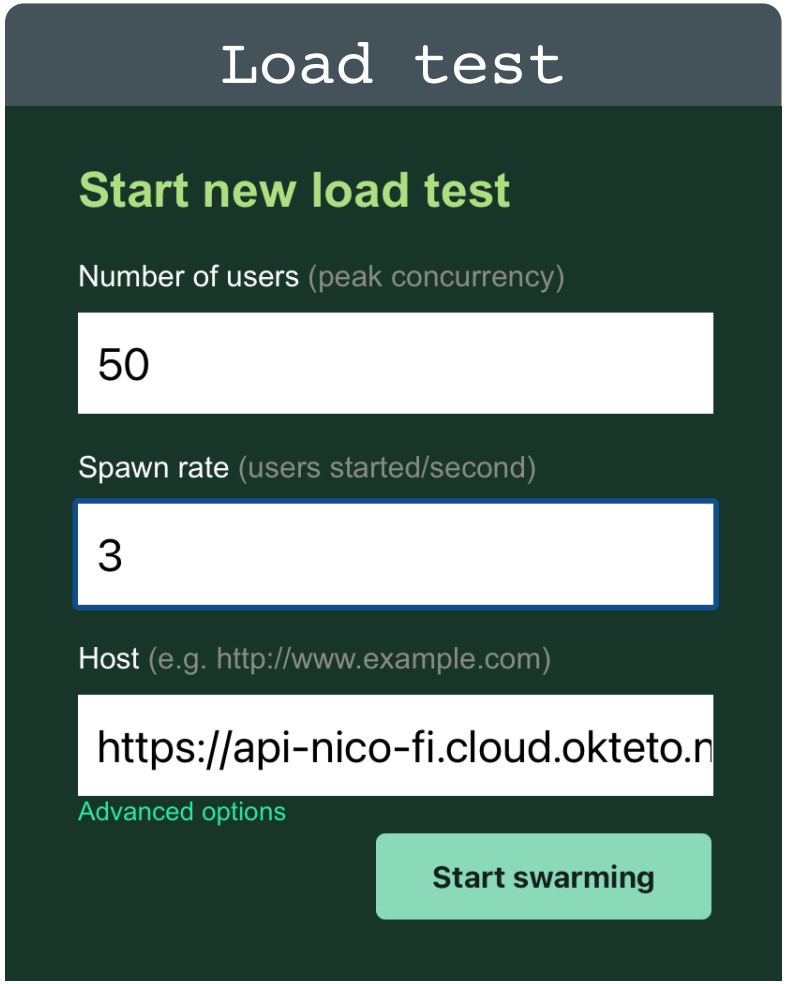


Status

VI. MONITORING: LOCUST

ApiUser

```
@task(1)
def health_check(self):

@task(10)
def predict_valid_item(self):

@task(4)
def predict_invalid_item(self):

@task(2)
def get_parameters(self):

@task(2)
def get_metrics(self):
```



Report

VI. MONITORING: PROMETHEUS

FastAPI instrumentator

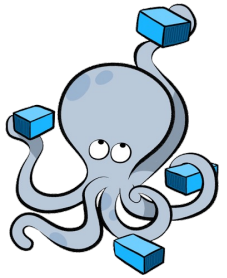
- Number of requests
- Request size
- Response size
- Latency
- Class prediction

`prometheus.yml`

```
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: 'api'
    scrape_interval: 5s
    static_configs:
      - targets: ['api:5000']
```

Server

VI. MONITORING: DOCKER COMPOSE



Monitoring services

```
prometheus:
  build:
    context: monitoring
    dockerfile: prometheus/Dockerfile
  container_name: prometheus
  ports:
    - '9090:9090'
  depends_on:
    - api
```

```
grafana:
  build:
    context: monitoring
    dockerfile: grafana/Dockerfile
  container_name: grafana
  ports:
    - '3000:3000'
  depends_on:
    - prometheus
```

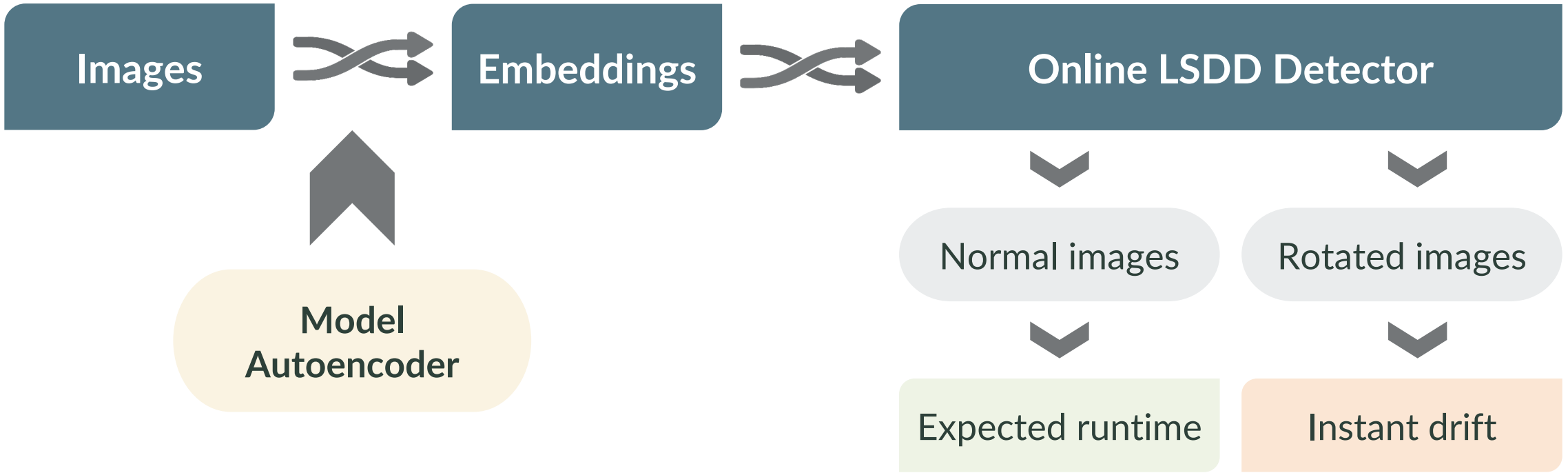
Use Prometheus data source
Display API dashboard

VI. MONITORING: GRAFANA

Locust traffic

Dashboard

VI. MONITORING: ALIBI DETECT



Notebook