MLOps Assignment 2 Report

Binary Image Classification (Cats vs Dogs) — End-to-End MLOps Pipeline

Course: MLOps (S1-25_AIMLCZG523)
Assignment: Assignment II
Date: 2026-02-22

Group Members

Student Name	ID
Azhar Nekware	2024AA05736
Anurag Sharma	2023AC05271
SHIKHAR DALELA	2023AC05481
NITIN KUSHWAHA	2024AA05764
POORNIMA T	2024AA05600

Table of Contents

Link to Code Repository

GitHub Repository: https://github.com/Azhar-N/Binary_image_classification_Assignment_No_2

Container Registry: `ghcr.io/azhar-n/catdog-classifier`

Architecture Diagram

```
┌──────────────────────────────────────────────────────────────────────┐
│                      Developer Workstation                            │
│                                                                      │
│   ┌──────────────┐    ┌────────────────┐    ┌───────────────────────┐ │
│   │ Git + DVC    │──▶│ src/train.py   │──▶│ MLflow Tracking UI    │ │
│   │ versioning   │    │ (ResNet-18)    │    │ - params logged       │ │
│   └──────────────┘    └────────────────┘    │ - metrics per epoch   │ │
│                             │               │ - loss_curves.png     │ │
│                             │               │ - confusion_matrix    │ │
│                             ▼               └───────────────────────┘ │
│                      ┌────────────────┐                              │
│                      │ models/*.pt    │  (DVC artifact output)       │
│                      └────────────────┘                              │
└──────────────────────────────────────────────────────────────────────┘
                            │  git push → main
                            ▼
┌──────────────────────────────────────────────────────────────────────┐
│                       GitHub Actions CI/CD                            │
│                                                                      │
│   ┌──────────────┐    ┌────────────────┐    ┌───────────────────────┐ │
│   │ Job 1        │──▶│ Job 2          │──▶│ Job 3                 │ │
│   │ Run pytest   │    │ docker build   │    │ docker compose up     │ │
│   │ (20 tests)   │    │ + push GHCR    │    │ + smoke_test.sh       │ │
│   └──────────────┘    └────────────────┘    └───────────────────────┘ │
│                             │                                        │
│                             ▼                                        │
│              ghcr.io/azhar-n/catdog-classifier:sha-xxxx              │
└──────────────────────────────────────────────────────────────────────┘
                            │  pull & deploy
                            ▼
┌──────────────────────────────────────────────────────────────────────┐
│                   Deployed Stack (Docker Compose)                     │
│                                                                      │
│   ┌───────────────────────────┐   ┌───────────────────────────────┐ │
│   │ catdog-api (FastAPI)      │   │ Prometheus                    │ │
│   │ Port: 8000                │◀─▶│  Port: 9090                   │ │
│   │ GET  /health              │   │ Scrapes /metrics every 15s    │ │
│   │ POST /predict             │   └───────────────────────────────┘ │
│   │ GET  /metrics             │                                     │
│   └───────────────────────────┘                                     │
│                                                                     │
└──────────────────────────────────────────────────────────────────────┘
```

M1 — Model Development & Experiment Tracking

1.1 Data & Code Versioning

Git — Source Code Versioning

All source code, configuration files, CI/CD definitions, and deployment manifests are tracked in Git with a structured commit history.

DVC — Dataset & Pipeline Versioning

DVC tracks the full data pipeline from raw images → processed data → trained model artifacts. The `dvc.yaml` defines two stages:

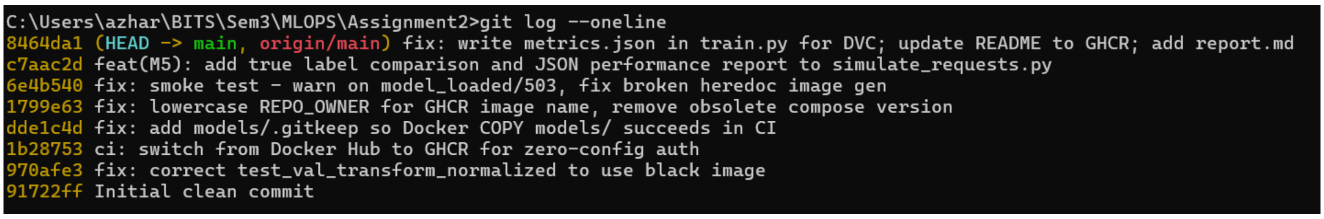
- **preprocess:** Resizes all images to 224×224 RGB, splits into train/val/test (80/10/10)
- **train:** Trains ResNet-18, logs to MLflow, outputs `.pt` checkpoint + charts

Hyperparameters tracked in `params.yaml`:

```yaml
preprocess:
  image_size: 224
  split_ratios: { train: 0.8, val: 0.1, test: 0.1 }

train:
  epochs: 10
  batch_size: 32
  lr: 0.001
  weight_decay: 0.0001
```

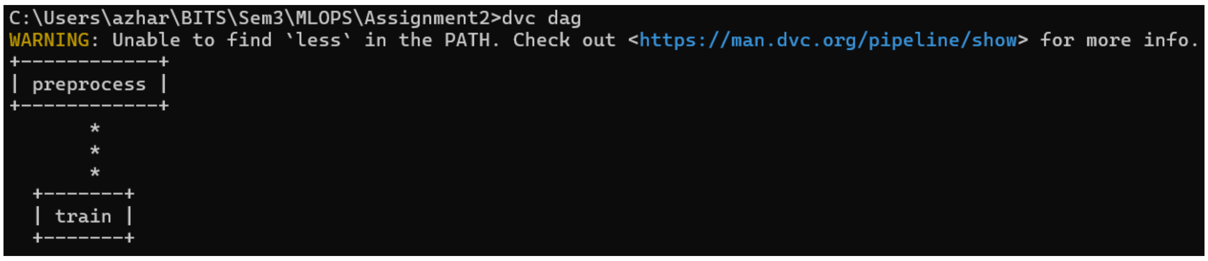
Run `dvc repro` to reproduce the full pipeline from scratch. Run `dvc metrics show` to compare runs.

Screenshot 1 — Git Repository: Commit History



```
C:\Users\azhar\BITS\Sem3\MLOPS\Assignment2>git log --oneline
8464da1 (HEAD -> main, origin/main) fix: write metrics.json in train.py for DVC; update README to GHCR; add report.md
c7aac2d feat(M5): add true label comparison and JSON performance report to simulate_requests.py
6e4b540 fix: smoke test - warn on model_loaded/503, fix broken heredoc image gen
1799e63 fix: lowercase REPO_OWNER for GHCR image name, remove obsolete compose version
dde1c4d fix: add models/.gitkeep so Docker COPY models/ succeeds in CI
1b28753 ci: switch from Docker Hub to GHCR for zero-config auth
970afe3 fix: correct test_val_transform_normalized to use black image
91722ff Initial clean commit
```

Screenshot 2 — DVC Pipeline DAG



```
C:\Users\azhar\BITS\Sem3\MLOPS\Assignment2>dvc dag
WARNING: Unable to find `less` in the PATH. Check out <https://man.dvc.org/pipeline/show> for more info.
+------------+
| preprocess |
+------------+
       *
       *
       *
  +-------+
  | train |
  +-------+
```

1.2 Data Preprocessing

File: `src/data_preprocessing.py`

The preprocessing pipeline:

1. Reads raw images from `data/raw/cat/` and `data/raw/dog/`
2. Converts all images to RGB (handles RGBA, grayscale, palette images)
3. Resizes every image to **224×224** using LANCZOS resampling
4. Performs reproducible **80/10/10** split (seed=42)
5. Writes processed images to `data/processed/{train,val,test}/{cat,dog}/`

Training augmentation pipeline (`src/utils.py`):

```
transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.RandomHorizontalFlip(p=0.5),
    transforms.RandomRotation(degrees=15),
    transforms.ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]),
])
```

1.3 Model Architecture

File: `src/model.py`

Backbone: ResNet-18 pretrained on ImageNet with a custom binary classification head:

```
Input: [B, 3, 224, 224]
  └─▶ ResNet-18 Backbone (pretrained ImageNet weights)
        └─▶ GlobalAvgPool → [B, 512]
              └─▶ Dropout(p=0.3)
                    └─▶ Linear(512 → 1)
Output: [B, 1] logit  →  sigmoid → dog probability
```

Attribute	Value

Attribute	Value
Architecture	ResNet-18 (transfer learning)
Loss	BCEWithLogitsLoss
Optimizer	Adam (lr=0.001, weight_decay=1e-4)
LR Scheduler	StepLR (step=5, γ=0.5)
Threshold	sigmoid ≥ 0.5 → dog, < 0.5 → cat
Parameters	~11.2M

1.4 Experiment Tracking — MLflow

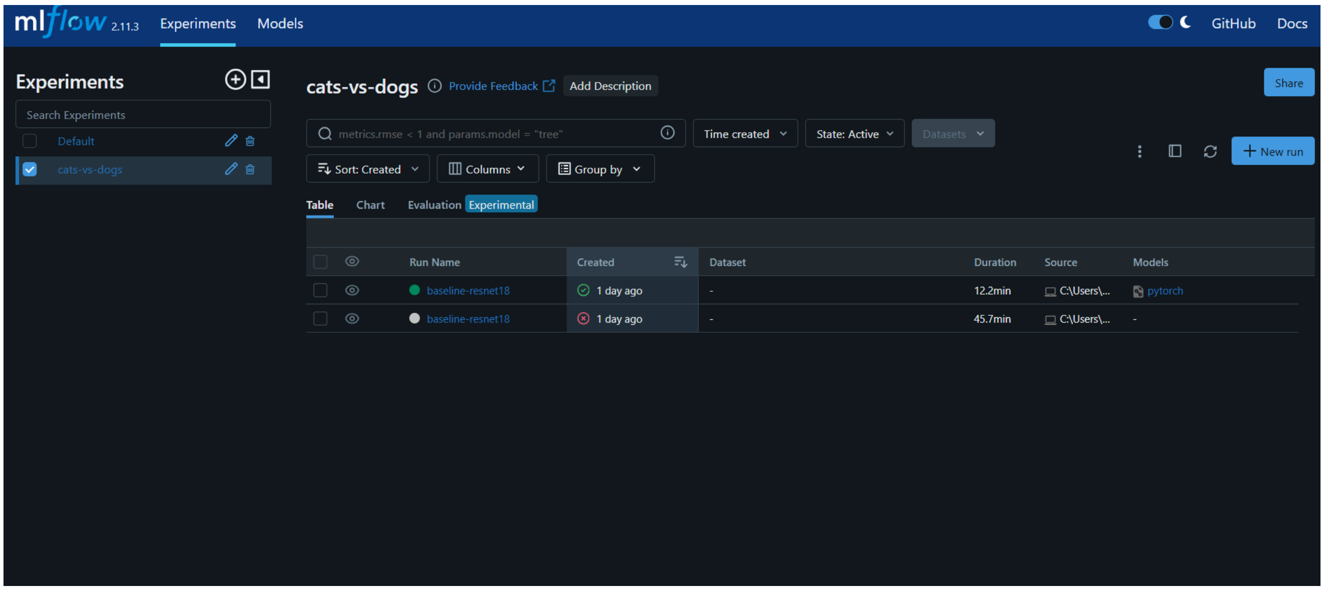
File: `src/train.py`

Every training run logs to MLflow:

Category	Items Logged
Parameters	epochs, batch_size, lr, weight_decay, model, optimizer
Metrics (per epoch)	train_loss, train_acc, val_loss, val_acc
Metrics (final)	test_loss, test_acc
Artifacts	`cat_dog_model.pt`, `loss_curves.png`, `confusion_matrix.png`
Model	`mlflow.pytorch.log_model()` for registry

```
python src/train.py --epochs 10 --lr 0.001 --run-name baseline-resnet18
mlflow ui   # http://localhost:5000
```

Screenshot 3 — MLflow Experiment List

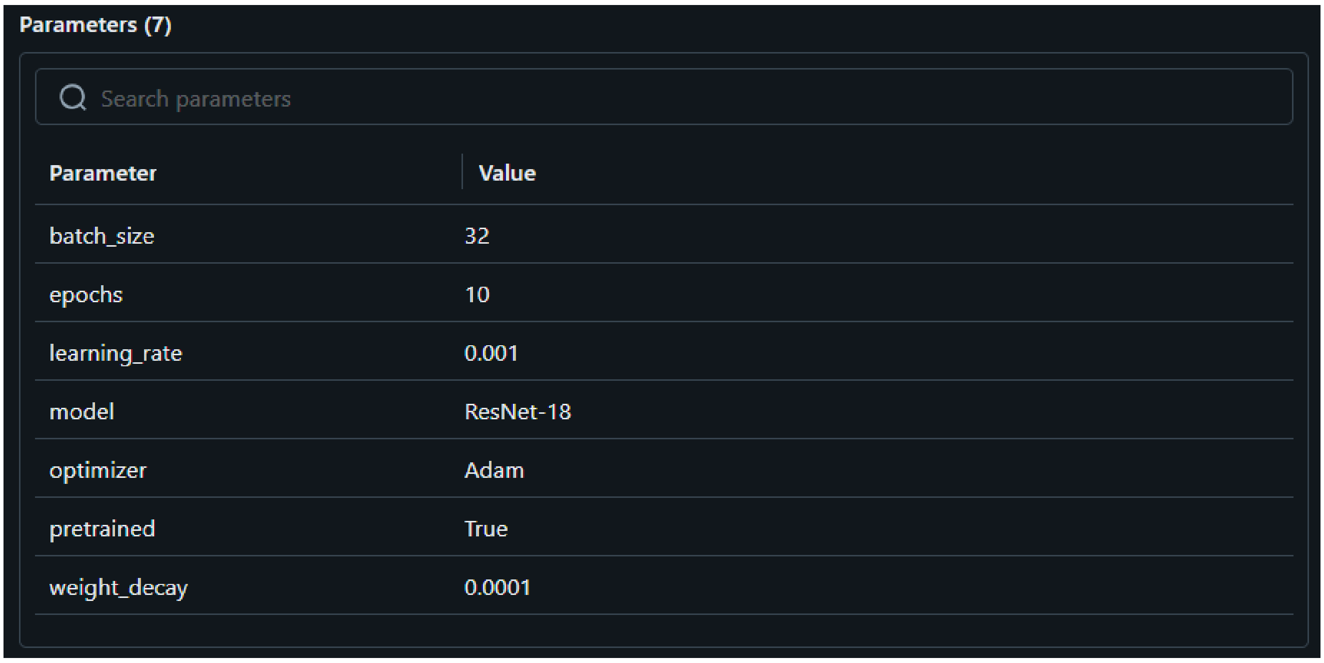


cats-vs-dogs

metrics.rmse < 1 and params.model = "tree"

Run Name	Created	Dataset	Duration	Source	Models
baseline-resnet18	1 day ago	-	12.2min	C:\Users\...	pytorch
baseline-resnet18	1 day ago	-	45.7min	C:\Users\...	-

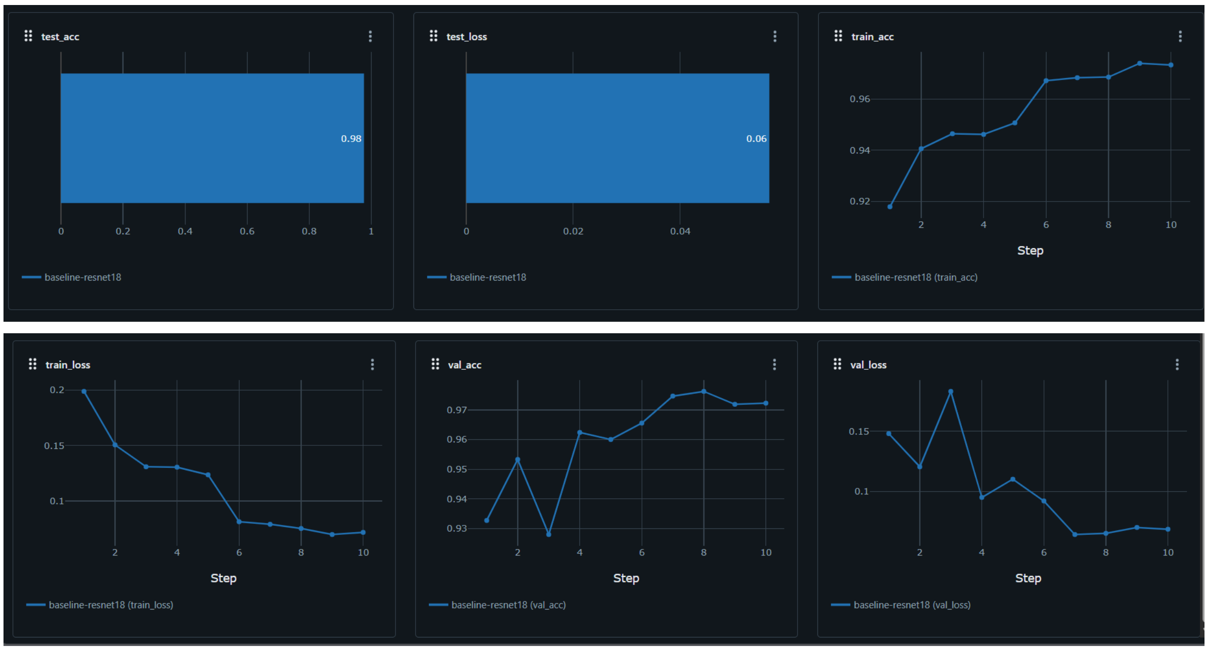
Screenshot 4 — MLflow Run: Parameters & Metrics



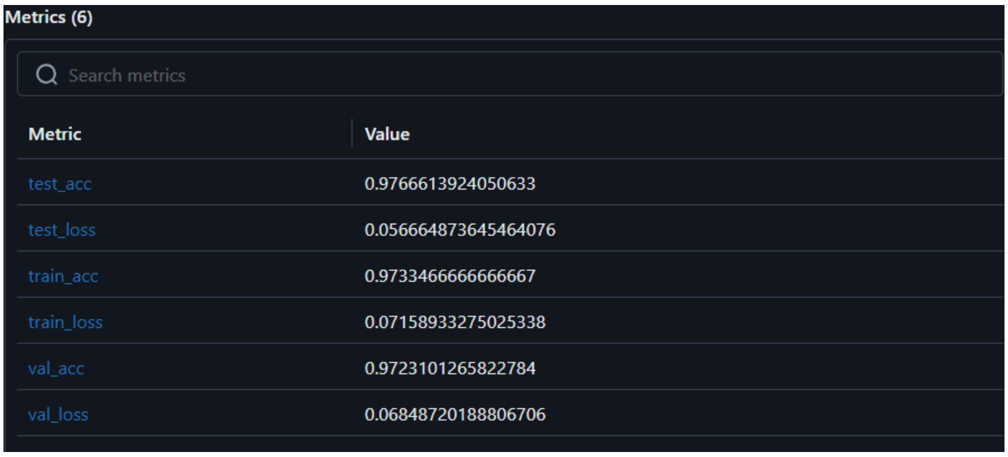
Parameters (7)

Parameter	Value
batch_size	32
epochs	10
learning_rate	0.001
model	ResNet-18
optimizer	Adam
pretrained	True
weight_decay	0.0001

Screenshot 5 — Training & Validation Loss Curves



Screenshot 6 — MLflow Metrics Dashboard


Metrics (6)

Search metrics

Metric	Value
test_acc	0.9766613924050633
test_loss	0.056664873645464076
train_acc	0.9733466666666667
train_loss	0.07158933275025338
val_acc	0.9723101265822784
val_loss	0.06848720188806706

M2 — Model Packaging & Containerization

2.1 FastAPI Inference Service

File: `app/main.py`

Method	Endpoint	Description
`GET`	`/health`	Service status + model_loaded flag
`POST`	`/predict`	Upload image → label + probabilities
`GET`	`/metrics`	Prometheus metrics (scraped every 15s)
`GET`	`/docs`	Auto-generated Swagger UI

Health Check Response:

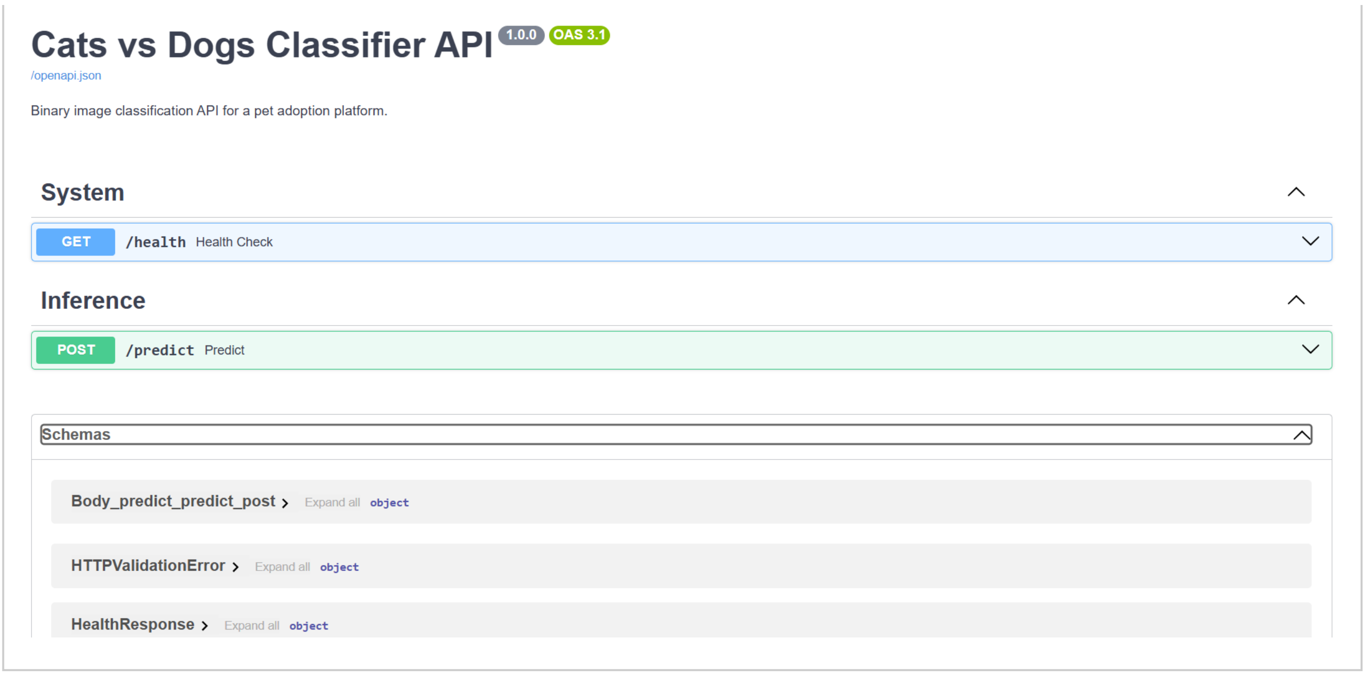
```
{ "status": "ok", "model_loaded": true }
```

Prediction Response:

```
{
  "label": "cat",
  "confidence": 0.9231,
  "cat_probability": 0.9231,
  "dog_probability": 0.0769
}
```

```
# Health check
curl http://localhost:8000/health

# Prediction
curl -X POST http://localhost:8000/predict -F "file=@cat.jpg"
```

Screenshot 7 — FastAPI Swagger UI (/docs)



Cats vs Dogs Classifier API 1.0.0 OAS 3.1

/openapi.json

Binary image classification API for a pet adoption platform.

System

GET /health Health Check

Inference

POST /predict Predict

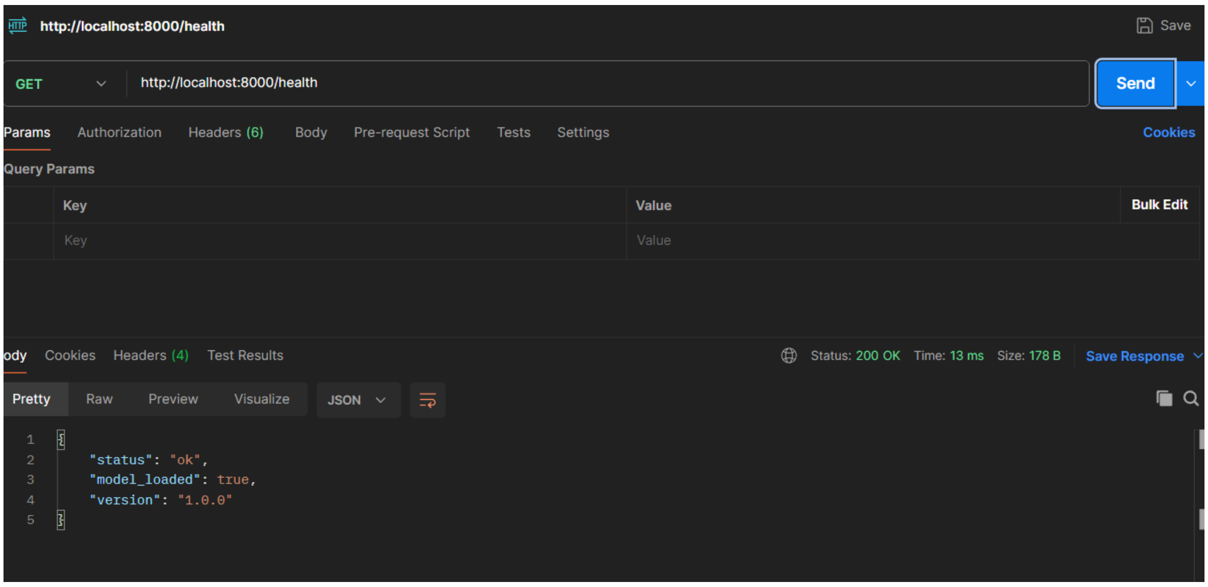
Schemas

Body_predict_predict_post Expand all object

HTTPValidationError Expand all object

HealthResponse Expand all object

Screenshot 8 — GET /health Response

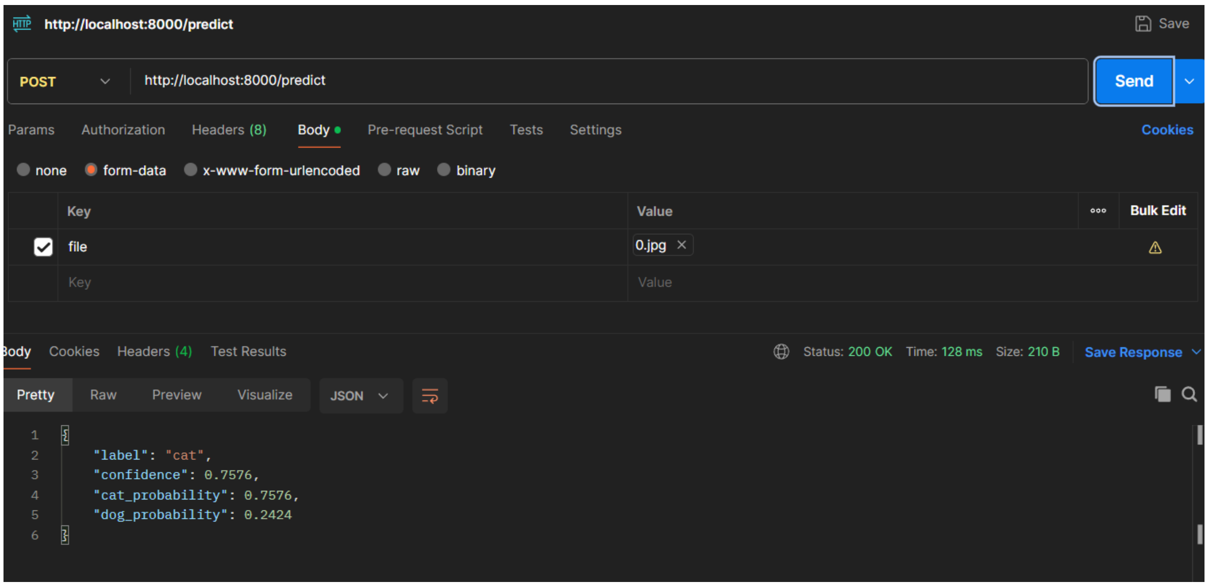


GET http://localhost:8000/health

Status: 200 OK Time: 13 ms Size: 178 B

```
{
    "status": "ok",
    "model_loaded": true,
    "version": "1.0.0"
}
```

Screenshot 9 — POST /predict Response



```
{
    "label": "cat",
    "confidence": 0.7576,
    "cat_probability": 0.7576,
    "dog_probability": 0.2424
}
```

2.2 Environment Specification

File: `requirements.txt` — All key library versions pinned:

```
torch==2.2.0           torchvision==0.17.0
fastapi==0.109.2       uvicorn[standard]==0.27.1
mlflow==2.10.2         dvc==3.40.1
prometheus-client==0.20.0
pytest==8.0.1          Pillow==10.2.0
numpy==1.26.4          scikit-learn==1.4.0
```

2.3 Dockerfile — Multi-Stage Build

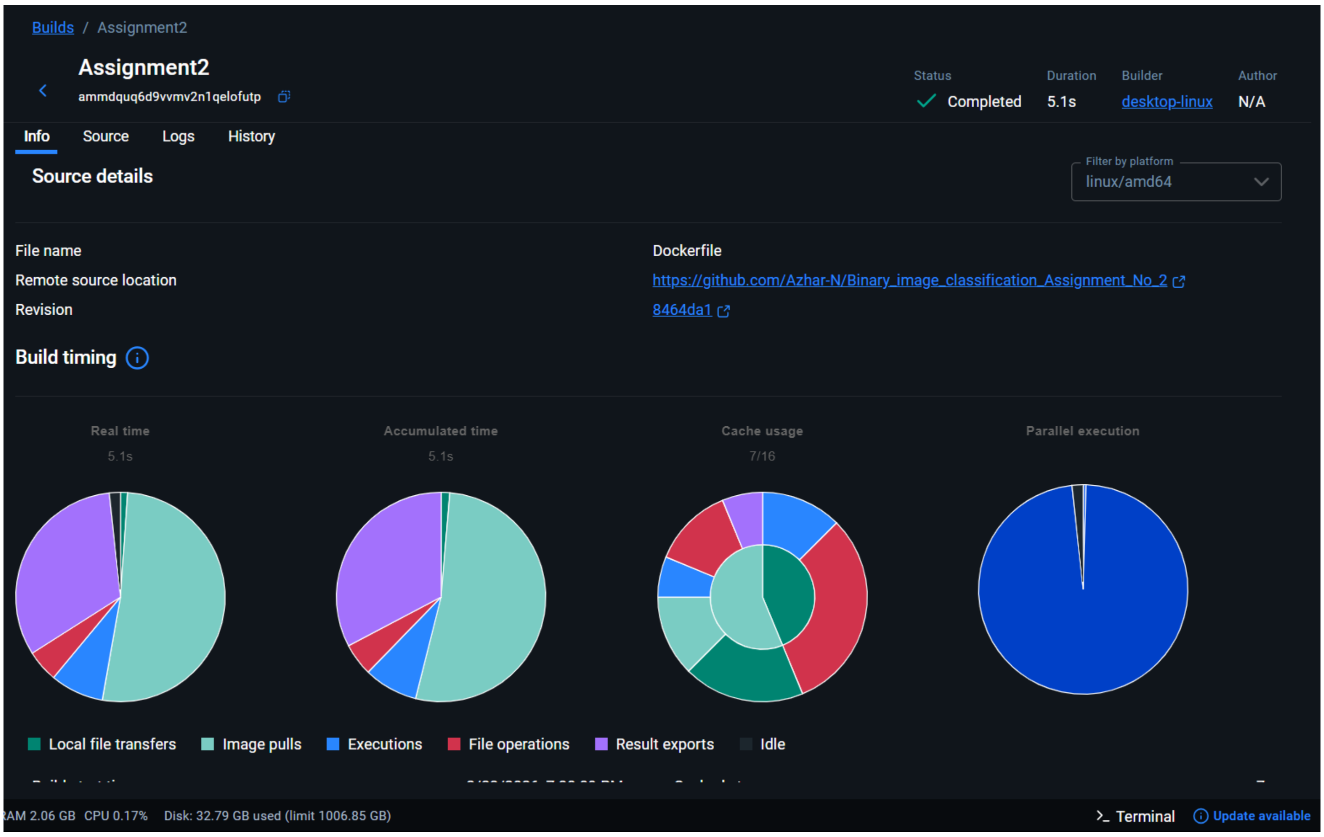
File: `Dockerfile`

Two-stage build separates build tools from the runtime image:

```
# Stage 1: Builder (installs packages with gcc/g++)
FROM python:3.10-slim AS builder
RUN apt-get install gcc g++
RUN pip install --no-cache-dir --user -r requirements.txt

# Stage 2: Runtime (no build tools — smaller image)
FROM python:3.10-slim AS runtime
COPY --from=builder /root/.local /root/.local
COPY app/ src/ models/ ./
EXPOSE 8000
CMD ["uvicorn", "app.main:app", "--host", "0.0.0.0", "--port", "8000"]
```

```
# Build locally
docker build -t catdog-classifier:local .

# Run with model volume-mounted
docker run -p 8000:8000 \
  -v ${PWD}/models:/app/models:ro \
  -e MODEL_PATH=/app/models/cat_dog_model.pt \
  catdog-classifier:local
```

Screenshot 10 — Docker Build Output


Builds / Assignment2

Assignment2
ammdquq6d9vvmv2n1qelofutp

Status	Duration	Builder	Author
✓ Completed	5.1s	desktop-linux	N/A

Info Source Logs History

Source details

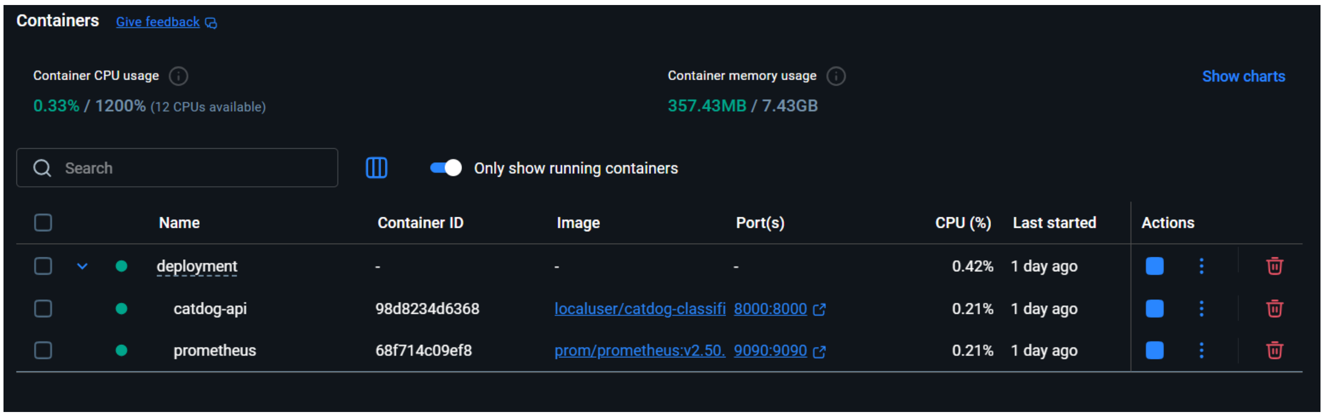
Filter by platform: linux/amd64

File name	Dockerfile
Remote source location	https://github.com/Azhar-N/Binary_image_classification_Assignment_No_2
Revision	8464da1

Build timing

RAM 2.06 GB CPU 0.17% Disk: 32.79 GB used (limit 1006.85 GB)

Screenshot 11 — Docker Container Running Locally


Containers Give feedback

Container CPU usage 0.33% / 1200% (12 CPUs available)

Container memory usage 357.43MB / 7.43GB

Show charts

Only show running containers

Name	Container ID	Image	Port(s)	CPU (%)	Last started
deployment	-	-	-	0.42%	1 day ago
catdog-api	98d8234d6368	localuser/catdog-classifi	8000:8000	0.21%	1 day ago
prometheus	68f714c09ef8	prom/prometheus:v2.50.	9090:9090	0.21%	1 day ago

M3 — CI Pipeline for Build, Test & Image Creation

3.1 Automated Testing

Files: `tests/test_preprocessing.py`, `tests/test_inference.py`

20 unit tests across two modules — all runnable in CI without model artifacts:

Preprocessing Tests (`test_preprocessing.py`):

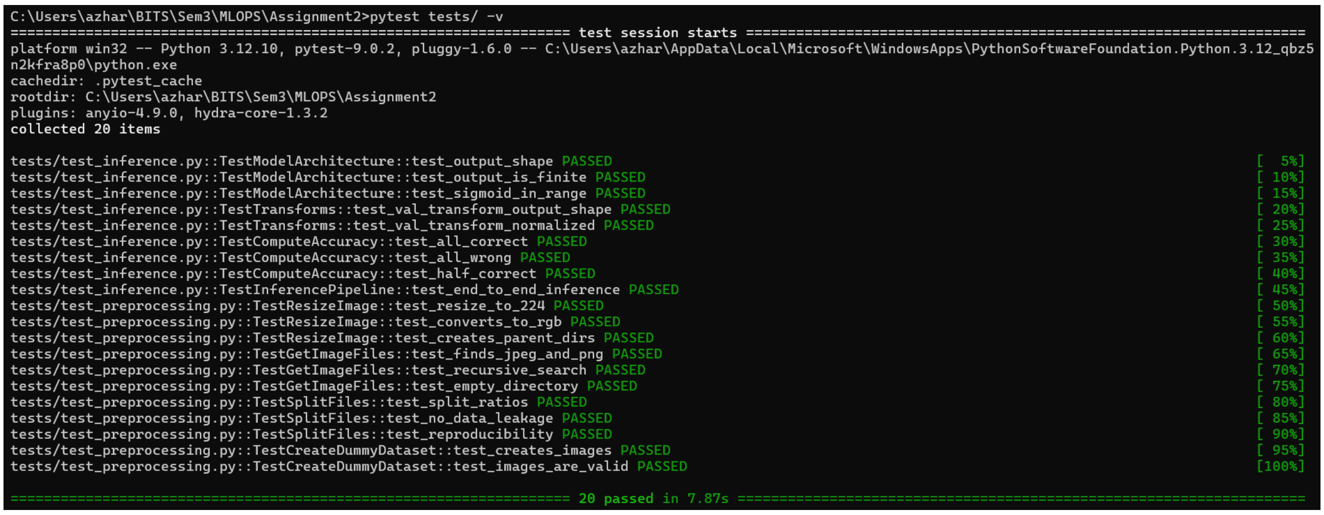
Test	Verifies
`test_resize_to_224`	Output image is exactly 224×224
`test_converts_to_rgb`	RGBA/L images converted to RGB
`test_creates_parent_dirs`	Nested destination dirs auto-created
`test_finds_jpeg_and_png`	Only image extensions collected
`test_recursive_search`	Nested directories searched
`test_empty_directory`	Returns empty list correctly
`test_split_ratios`	80/10/10 proportions match exactly
`test_no_data_leakage`	No file appears in 2 splits
`test_reproducibility`	Same seed → identical split
`test_creates_images`	N images per class created
`test_images_are_valid`	Dummy images are openable/valid RGB

Inference Tests (`test_inference.py`):

Test	Verifies
`test_output_shape`	Model output shape is `[batch, 1]`
`test_output_is_finite`	No NaN/Inf in logits
`test_sigmoid_in_range`	All probabilities ∈ [0, 1]
`test_val_transform_output_shape`	Tensor shape `[3, 224, 224]`
`test_val_transform_normalized`	ImageNet normalization applied
`test_all_correct`	Accuracy = 1.0 for perfect predictions
`test_all_wrong`	Accuracy = 0.0 for all-wrong predictions
`test_half_correct`	Accuracy = 0.5 for half correct
`test_end_to_end_inference`	Full PIL→tensor→logit→label pipeline

```
pytest tests/ -v --tb=short
```

Screenshot 12 — pytest Passing Locally



```
C:\Users\azhar\BITS\Sem3\MLOPS\Assignment2>pytest tests/ -v
=================================================================== test session starts ====================================================================
platform win32 -- Python 3.12.10, pytest-9.0.2, pluggy-1.6.0 -- C:\Users\azhar\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.12_qbz5
n2kfra8p0\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\azhar\BITS\Sem3\MLOPS\Assignment2
plugins: anyio-4.9.0, hydra-core-1.3.2
collected 20 items

tests/test_inference.py::TestModelArchitecture::test_output_shape PASSED                                                                              [  5%]
tests/test_inference.py::TestModelArchitecture::test_output_is_finite PASSED                                                                          [ 10%]
tests/test_inference.py::TestModelArchitecture::test_sigmoid_in_range PASSED                                                                          [ 15%]
tests/test_inference.py::TestTransforms::test_val_transform_output_shape PASSED                                                                       [ 20%]
tests/test_inference.py::TestTransforms::test_val_transform_normalized PASSED                                                                         [ 25%]
tests/test_inference.py::TestComputeAccuracy::test_all_correct PASSED                                                                                 [ 30%]
tests/test_inference.py::TestComputeAccuracy::test_all_wrong PASSED                                                                                   [ 35%]
tests/test_inference.py::TestComputeAccuracy::test_half_correct PASSED                                                                                [ 40%]
tests/test_inference.py::TestInferencePipeline::test_end_to_end_inference PASSED                                                                      [ 45%]
tests/test_preprocessing.py::TestResizeImage::test_resize_to_224 PASSED                                                                               [ 50%]
tests/test_preprocessing.py::TestResizeImage::test_converts_to_rgb PASSED                                                                             [ 55%]
tests/test_preprocessing.py::TestResizeImage::test_creates_parent_dirs PASSED                                                                         [ 60%]
tests/test_preprocessing.py::TestGetImageFiles::test_finds_jpeg_and_png PASSED                                                                        [ 65%]
tests/test_preprocessing.py::TestGetImageFiles::test_recursive_search PASSED                                                                          [ 70%]
tests/test_preprocessing.py::TestGetImageFiles::test_empty_directory PASSED                                                                           [ 75%]
tests/test_preprocessing.py::TestSplitFiles::test_split_ratios PASSED                                                                                 [ 80%]
tests/test_preprocessing.py::TestSplitFiles::test_no_data_leakage PASSED                                                                              [ 85%]
tests/test_preprocessing.py::TestSplitFiles::test_reproducibility PASSED                                                                              [ 90%]
tests/test_preprocessing.py::TestCreateDummyDataset::test_creates_images PASSED                                                                       [ 95%]
tests/test_preprocessing.py::TestCreateDummyDataset::test_images_are_valid PASSED                                                                     [100%]

==================================================================== 20 passed in 7.87s ====================================================================
```

3.2 CI Setup — GitHub Actions

File: `.github/workflows/ci-cd.yml`

Triggers on every push to `main`/`develop` and every pull request to `main`.

Job 1 — Unit Tests (all branches + PRs):

```
Checkout → Python 3.10 → pip install → pytest tests/ → Upload JUnit XML
```

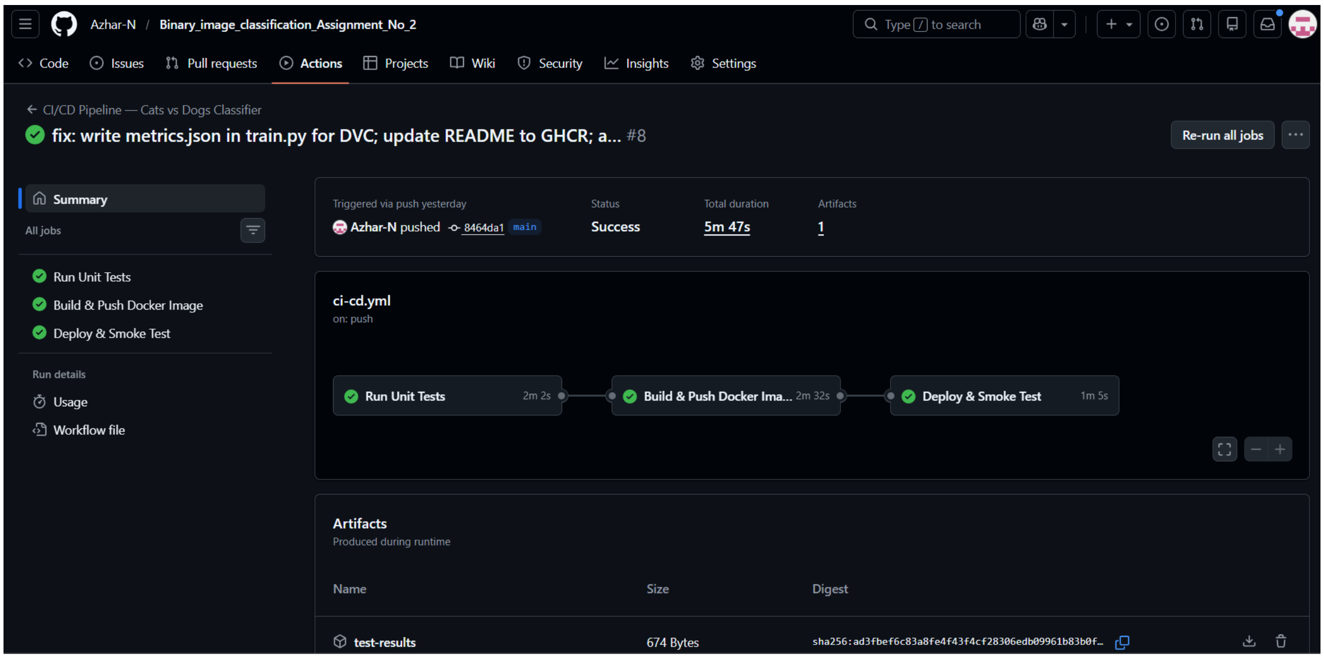
Job 2 — Build & Push Docker Image (main branch only):

```
Checkout → Lowercase REPO_OWNER → QEMU + Buildx setup
  → Login GHCR (GITHUB_TOKEN — no secrets needed)
  → docker build → Push ghcr.io/azhar-n/catdog-classifier:sha-<hash>
```

Job 3 — Deploy & Smoke Test (after Job 2):

```
docker compose up -d --wait → smoke_test.sh → docker compose down
```

Screenshot 13 — GitHub Actions: All Jobs Passing



CI/CD Pipeline — Cats vs Dogs Classifier

fix: write metrics.json in train.py for DVC; update README to GHCR; a... #8

Run Unit Tests
Build & Push Docker Image
Deploy & Smoke Test

Triggered via push yesterday	Status	Total duration	Artifacts
Azhar-N pushed 8464da1 main	Success	5m 47s	1

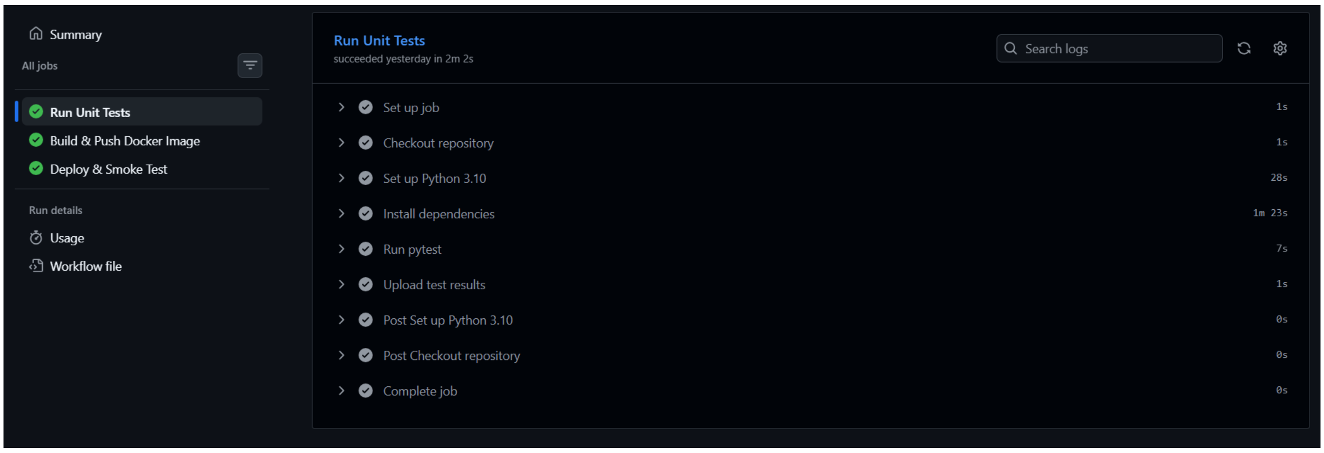
ci-cd.yml
on: push

Run Unit Tests 2m 2s → Build & Push Docker Ima... 2m 32s → Deploy & Smoke Test 1m 5s

Artifacts
Produced during runtime

Name	Size	Digest
test-results	674 Bytes	sha256:ad3fbef6c83a8fe4f43f4cf28306edb09961b83b0f…

Screenshot 14 — GitHub Actions: Unit Test Job Detail



Run Unit Tests
succeeded yesterday in 2m 2s

```
Set up job                      1s
Checkout repository             1s
Set up Python 3.10              28s
Install dependencies            1m 23s
Run pytest                      7s
Upload test results             1s
Post Set up Python 3.10         0s
Post Checkout repository        0s
Complete job                    0s
```

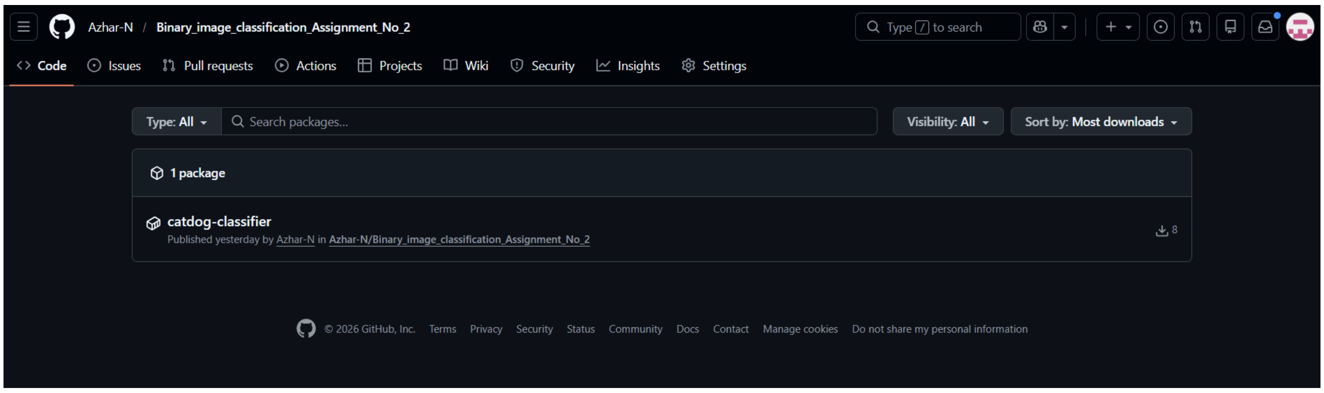
3.3 Artifact Publishing — GHCR

Images are tagged and pushed to GitHub Container Registry on every successful main branch build:

```
ghcr.io/azhar-n/catdog-classifier:latest
ghcr.io/azhar-n/catdog-classifier:sha-<git-short-hash>
```

No manual secrets required — uses the automatically-injected `GITHUB_TOKEN`.

Screenshot 15 — GHCR: Published Docker Image



Azhar-N / Binary_image_classification_Assignment_No_2

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

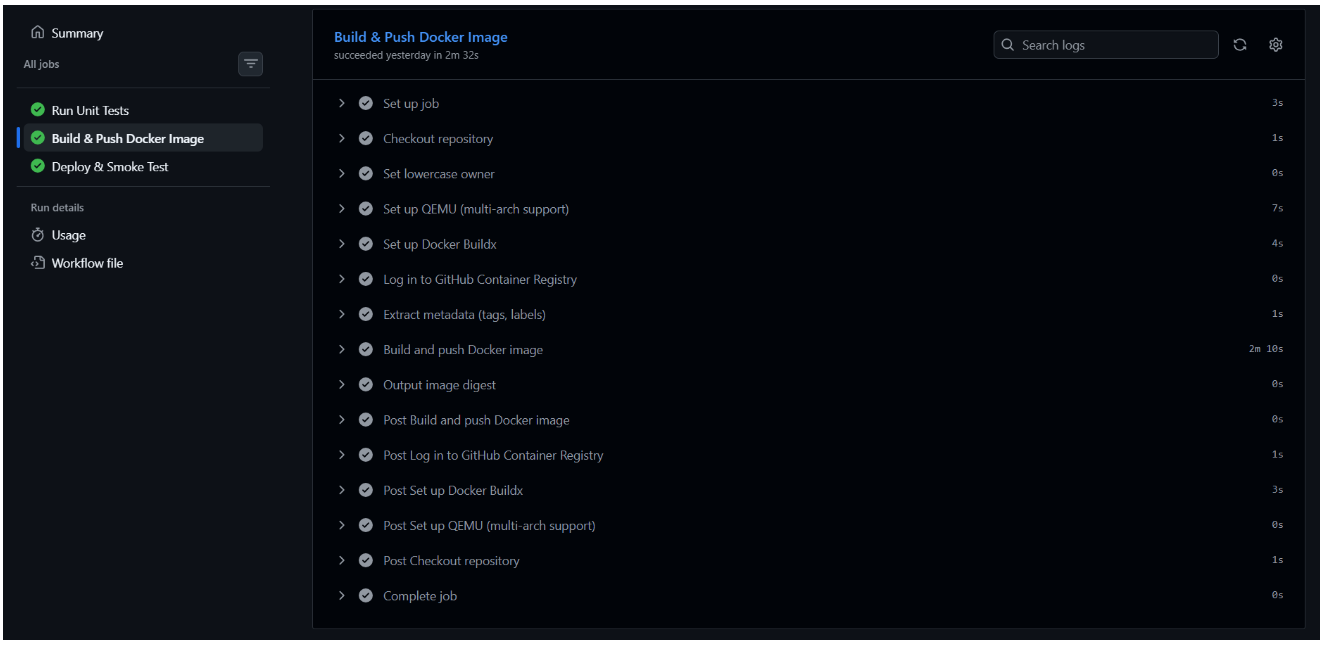
Type: All | Search packages... | Visibility: All | Sort by: Most downloads

1 package

catdog-classifier
Published yesterday by Azhar-N in Azhar-N/Binary_image_classification_Assignment_No_2 — 8

© 2026 GitHub, Inc. Terms Privacy Security Status Community Docs Contact Manage cookies Do not share my personal information

Screenshot 16 — GitHub Actions: Build & Push Job Detail



Summary
All jobs
- Run Unit Tests
- Build & Push Docker Image
- Deploy & Smoke Test

Run details
- Usage
- Workflow file

Build & Push Docker Image
succeeded yesterday in 2m 32s

```
Set up job                                3s
Checkout repository                       1s
Set lowercase owner                       0s
Set up QEMU (multi-arch support)          7s
Set up Docker Buildx                      4s
Log in to GitHub Container Registry       0s
Extract metadata (tags, labels)           1s
Build and push Docker image               2m 10s
Output image digest                       0s
Post Build and push Docker image          0s
Post Log in to GitHub Container Registry  1s
Post Set up Docker Buildx                 3s
Post Set up QEMU (multi-arch support)     0s
Post Checkout repository                  1s
Complete job                              0s
```

M4 — CD Pipeline & Deployment

4.1 Deployment Target — Docker Compose

File: `deployment/docker-compose.yml`

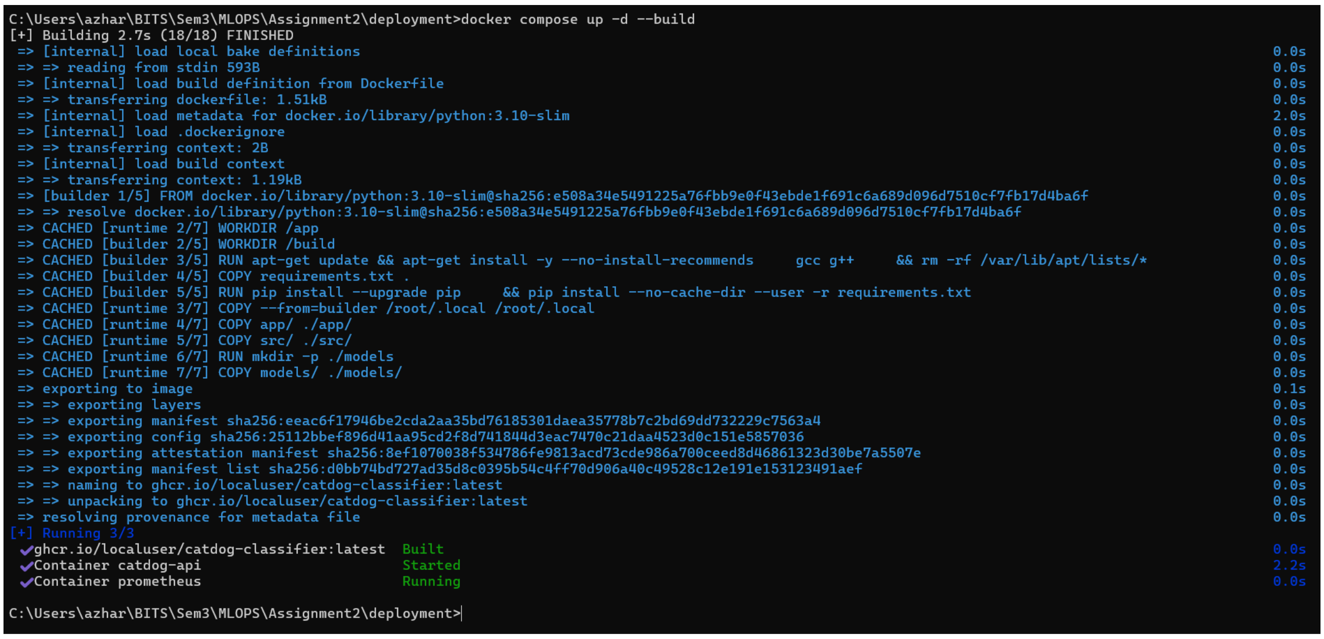
Two-service stack deployed for every push to `main`:

```
services:
  catdog-api:
    image: ghcr.io/${REPO_OWNER}/catdog-classifier:${IMAGE_TAG}
    build: { context: .., dockerfile: Dockerfile }  # local dev fallback
    ports: ["8000:8000"]
    volumes: ["../models:/app/models:ro"]  # model mounted at runtime
    healthcheck:
      test: ["CMD", "python", "-c",
"urllib.request.urlopen('http://localhost:8000/health')"]

  prometheus:
    image: prom/prometheus:v2.50.1
    ports: ["9090:9090"]
    volumes: ["../monitoring/prometheus.yml:/etc/prometheus/prometheus.yml:ro"]
```

> **Key design:** The `.pt` model file is **volume-mounted**, not baked into the image. This allows model updates without rebuilding the container.

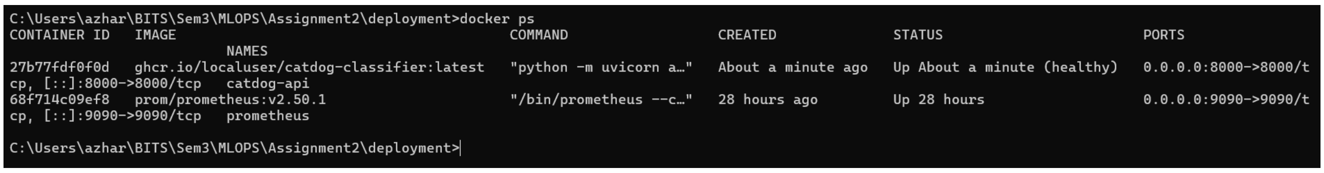
```
# Local development (builds from source)
cd deployment
docker compose up -d --build

# Verify
curl http://localhost:8000/health
```

Screenshot 17 — docker compose up Output



```
C:\Users\azhar\BITS\Sem3\MLOPS\Assignment2\deployment>docker compose up -d --build
[+] Building 2.7s (18/18) FINISHED
 => [internal] load local bake definitions                                                                                                                  0.0s
 => => reading from stdin 593B                                                                                                                              0.0s
 => [internal] load build definition from Dockerfile                                                                                                        0.0s
 => => transferring dockerfile: 1.51kB                                                                                                                      0.0s
 => [internal] load metadata for docker.io/library/python:3.10-slim                                                                                         2.0s
 => [internal] load .dockerignore                                                                                                                           0.0s
 => => transferring context: 2B                                                                                                                             0.0s
 => [internal] load build context                                                                                                                           0.0s
 => => transferring context: 1.19kB                                                                                                                         0.0s
 => [builder 1/5] FROM docker.io/library/python:3.10-slim@sha256:e508a34e5491225a76fbb9e0f43ebde1f691c6a689d096d7510cf7fb17d4ba6f                            0.0s
 => => resolve docker.io/library/python:3.10-slim@sha256:e508a34e5491225a76fbb9e0f43ebde1f691c6a689d096d7510cf7fb17d4ba6f                                   0.0s
 => CACHED [runtime 2/7] WORKDIR /app                                                                                                                       0.0s
 => CACHED [builder 2/5] WORKDIR /build                                                                                                                     0.0s
 => CACHED [builder 3/5] RUN apt-get update && apt-get install -y --no-install-recommends     gcc g++     && rm -rf /var/lib/apt/lists/*                     0.0s
 => CACHED [builder 4/5] COPY requirements.txt .                                                                                                            0.0s
 => CACHED [builder 5/5] RUN pip install --upgrade pip     && pip install --no-cache-dir --user -r requirements.txt                                         0.0s
 => CACHED [runtime 3/7] COPY --from=builder /root/.local /root/.local                                                                                      0.0s
 => CACHED [runtime 4/7] COPY app/ ./app/                                                                                                                   0.0s
 => CACHED [runtime 5/7] COPY src/ ./src/                                                                                                                   0.0s
 => CACHED [runtime 6/7] RUN mkdir -p ./models                                                                                                              0.0s
 => CACHED [runtime 7/7] COPY models/ ./models/                                                                                                             0.0s
 => exporting to image                                                                                                                                      0.1s
 => => exporting layers                                                                                                                                     0.0s
 => => exporting manifest sha256:eeac6f17946be2cda2aa35bd76185301daea35778b7c2bd69dd732229c7563a4                                                           0.0s
 => => exporting config sha256:25112bbef896d41aa95cd2f8d741844d3eac7470c21daa4523d0c151e5857036                                                             0.0s
 => => exporting attestation manifest sha256:8ef1070038f534786fe9813acd73cde986a700ceed8d46861323d30be7a5507e                                               0.0s
 => => exporting manifest list sha256:d0bb74bd727ad35d8c0395b54c4ff70d906a40c49528c12e191e153123491aef                                                      0.0s
 => => naming to ghcr.io/localuser/catdog-classifier:latest                                                                                                 0.0s
 => => unpacking to ghcr.io/localuser/catdog-classifier:latest                                                                                              0.0s
 => resolving provenance for metadata file                                                                                                                  0.0s
[+] Running 3/3
 ✔ghcr.io/localuser/catdog-classifier:latest  Built                                                                                                         0.0s
 ✔Container catdog-api                        Started                                                                                                       2.2s
 ✔Container prometheus                        Running                                                                                                       0.0s

C:\Users\azhar\BITS\Sem3\MLOPS\Assignment2\deployment>
```

Screenshot 18 — Running Containers (docker ps)



```
C:\Users\azhar\BITS\Sem3\MLOPS\Assignment2\deployment>docker ps
CONTAINER ID   IMAGE                                         COMMAND                  CREATED              STATUS                         PORTS
                          NAMES
27b77fdf0f0d   ghcr.io/localuser/catdog-classifier:latest   "python -m uvicorn a…"   About a minute ago   Up About a minute (healthy)   0.0.0.0:8000->8000/t
cp, [::]:8000->8000/tcp   catdog-api
68f714c09ef8   prom/prometheus:v2.50.1                       "/bin/prometheus --c…"   28 hours ago         Up 28 hours                    0.0.0.0:9090->9090/t
cp, [::]:9090->9090/tcp   prometheus

C:\Users\azhar\BITS\Sem3\MLOPS\Assignment2\deployment>
```

4.2 CD/GitOps Flow

Every push to `main` automatically triggers the full pipeline:

```
git push → main
      │
      ▼
Job 1: pytest (20 tests) — must pass
      │
      ▼
Job 2: docker build + push → ghcr.io/azhar-n/catdog-classifier:sha-<hash>
      │
      ▼
Job 3: docker compose up → smoke_test.sh → docker compose down
```

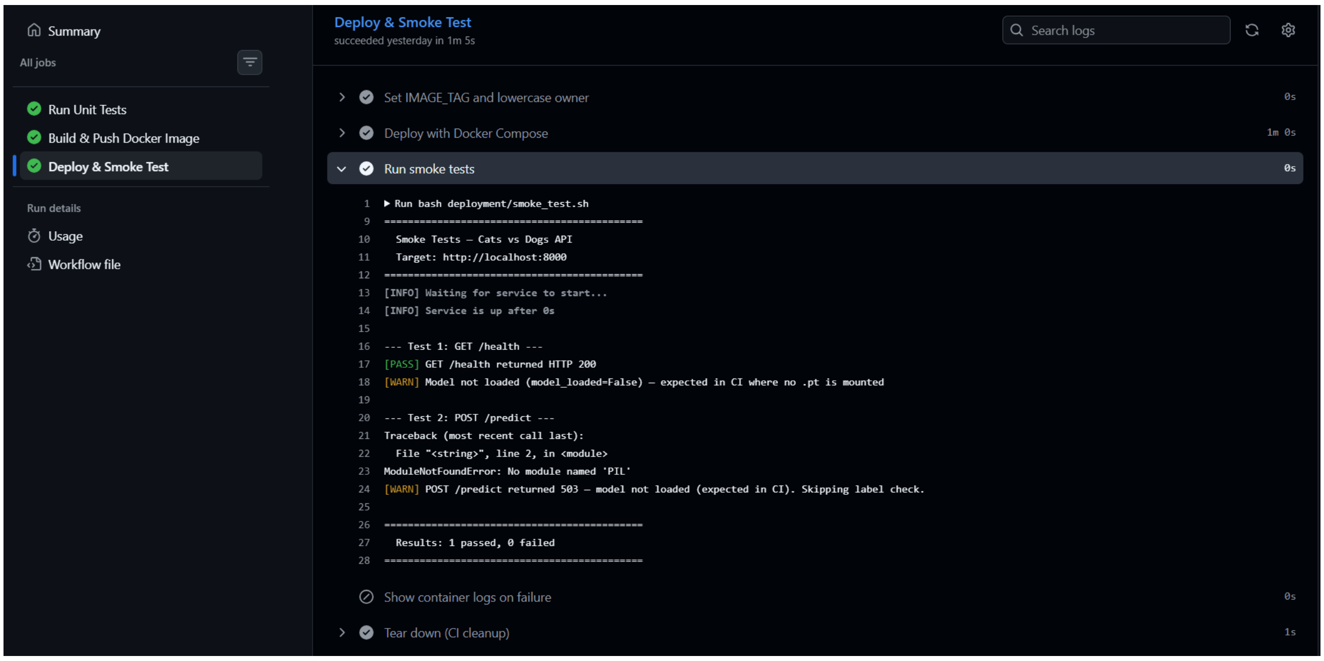
4.3 Smoke Tests

File: `deployment/smoke_test.sh`

Runs automatically post-deploy and fails the pipeline if critical assertions fail:

Test	Type	Failure Behaviour
`GET /health` returns 200	Hard	Pipeline fails
`model_loaded` is true	Soft	Warning only (CI has no .pt)
`POST /predict` returns 200	Hard (503 = soft)	Pipeline fails if not 200/503
Response label is "cat" or "dog"	Hard	Pipeline fails on invalid label

Screenshot 19 — Smoke Test Passing



```
Set IMAGE_TAG and lowercase owner                                   0s
Deploy with Docker Compose                                        1m 0s
Run smoke tests                                                     0s
▶ Run bash deployment/smoke_test.sh
============================================
  Smoke Tests — Cats vs Dogs API
  Target: http://localhost:8000
============================================
[INFO] Waiting for service to start...
[INFO] Service is up after 0s

--- Test 1: GET /health ---
[PASS] GET /health returned HTTP 200
[WARN] Model not loaded (model_loaded=False) — expected in CI where no .pt is mounted

--- Test 2: POST /predict ---
Traceback (most recent call last):
  File "<string>", line 2, in <module>
ModuleNotFoundError: No module named 'PIL'
[WARN] POST /predict returned 503 — model not loaded (expected in CI). Skipping label check.

============================================
  Results: 1 passed, 0 failed
============================================
Show container logs on failure                                      0s
Tear down (CI cleanup)                                              1s
```

M5 — Monitoring, Logs & Performance Tracking

5.1 Request/Response Logging

File: `app/main.py`

All requests are logged as structured JSON (no sensitive data):

```
logging.basicConfig(
    format='{"time": "%(asctime)s", "level": "%(levelname)s", "message": "%
(message)s"}',
)
```

Sample log entry:

```
{"time": "2026-02-22 10:15:42", "level": "INFO",
 "message": "predict | label=cat confidence=0.9231 latency=0.042s file=pet.jpg"}
```

What is logged:

- Prediction label and confidence
- End-to-end request latency
- Uploaded filename (not file content)
- Model load success/failure at startup

5.2 Prometheus Metrics

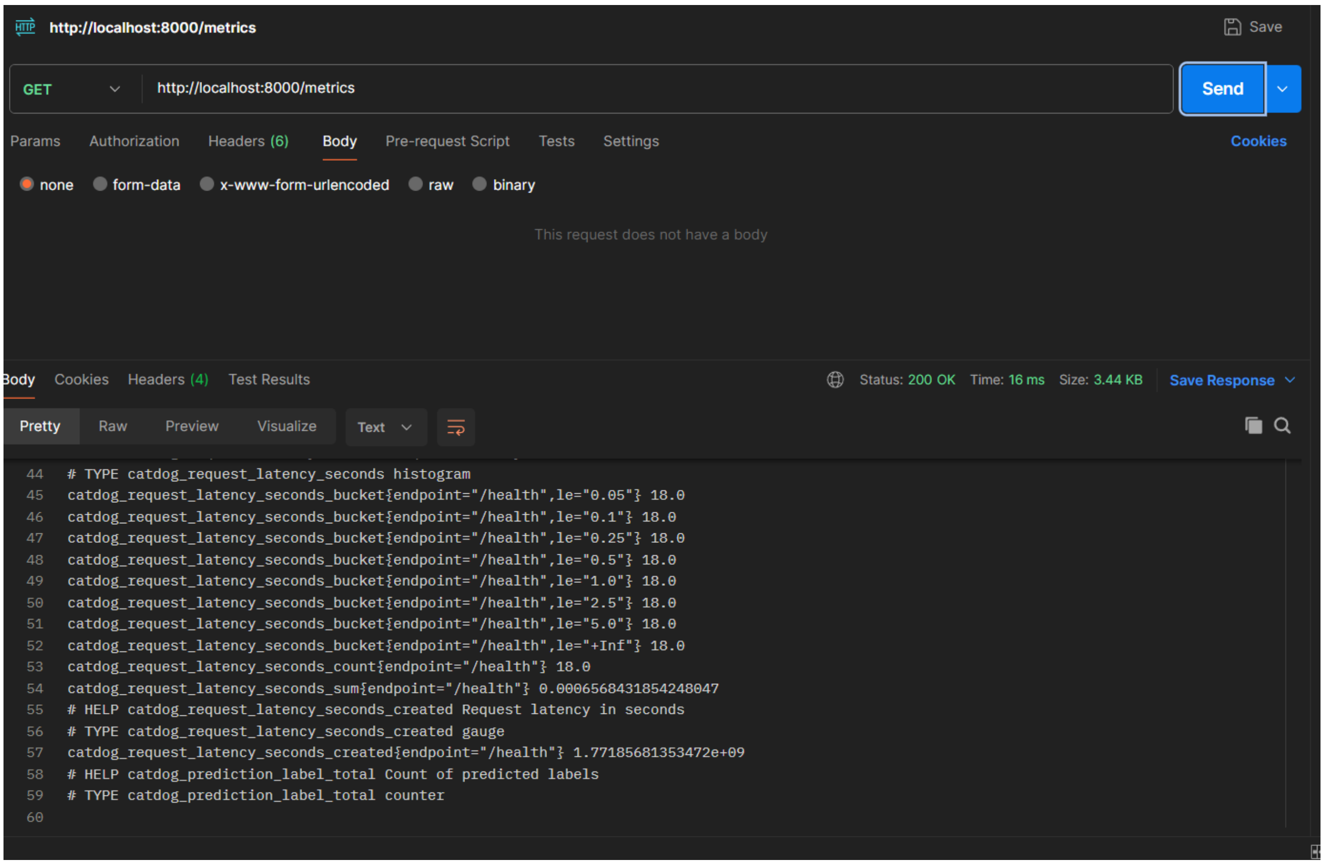
Three custom metrics are exposed at `GET /metrics`:

Metric	Type	Labels	Description
`catdog_request_total`	Counter	`endpoint`, `status`	Requests by endpoint + HTTP status
`catdog_request_latency_seconds`	Histogram	`endpoint`	Latency distribution (7 buckets)
`catdog_prediction_label_total`	Counter	`label`	Cat vs Dog prediction counts

Prometheus scrape config (`monitoring/prometheus.yml`):

```
scrape_configs:
  - job_name: catdog-api
    static_configs:
      - targets: ['catdog-api:8000']
    metrics_path: /metrics
    scrape_interval: 15s
```

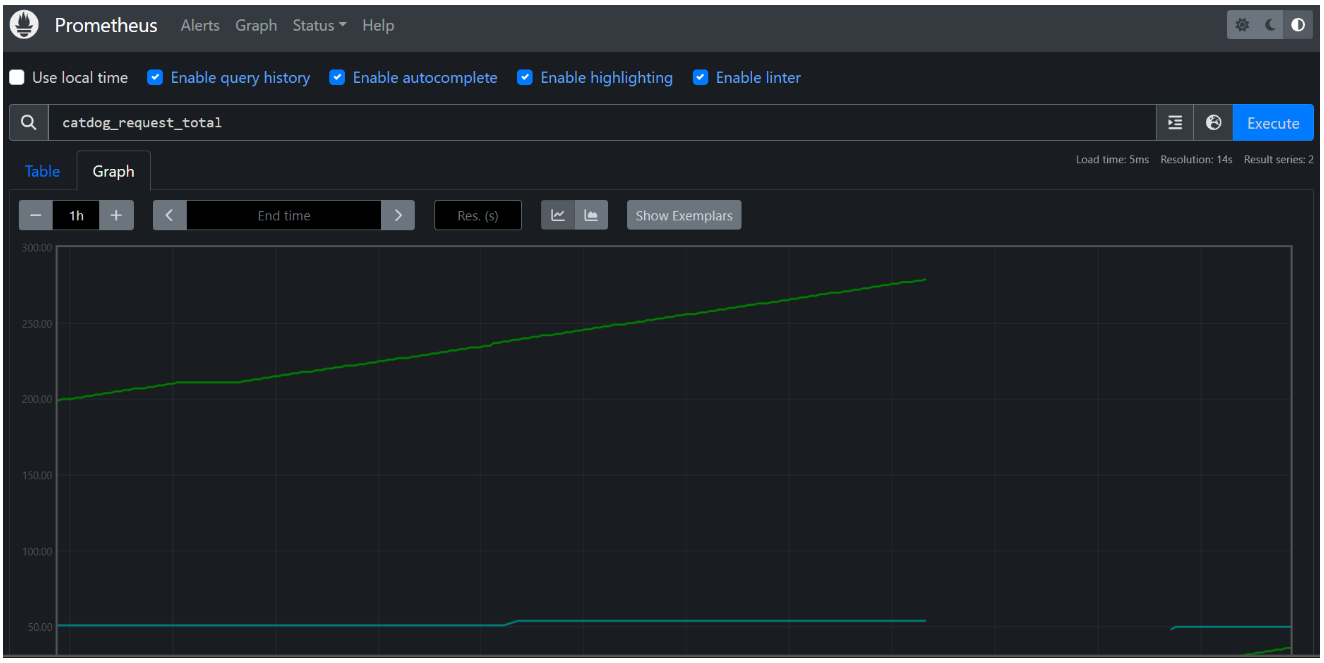
Screenshot 20 — Prometheus Metrics Endpoint (curl /metrics)


http://localhost:8000/metrics
Save
GET
http://localhost:8000/metrics
Send
Params Authorization Headers (6) Body Pre-request Script Tests Settings
Cookies
none form-data x-www-form-urlencoded raw binary
This request does not have a body
Body Cookies Headers (4) Test Results
Save Response
Pretty Raw Preview Visualize Text

Status: 200 OK Time: 16 ms Size: 3.44 KB

```
# TYPE catdog_request_latency_seconds histogram
catdog_request_latency_seconds_bucket{endpoint="/health",le="0.05"} 18.0
catdog_request_latency_seconds_bucket{endpoint="/health",le="0.1"} 18.0
catdog_request_latency_seconds_bucket{endpoint="/health",le="0.25"} 18.0
catdog_request_latency_seconds_bucket{endpoint="/health",le="0.5"} 18.0
catdog_request_latency_seconds_bucket{endpoint="/health",le="1.0"} 18.0
catdog_request_latency_seconds_bucket{endpoint="/health",le="2.5"} 18.0
catdog_request_latency_seconds_bucket{endpoint="/health",le="5.0"} 18.0
catdog_request_latency_seconds_bucket{endpoint="/health",le="+Inf"} 18.0
catdog_request_latency_seconds_count{endpoint="/health"} 18.0
catdog_request_latency_seconds_sum{endpoint="/health"} 0.0006568431854248047
# HELP catdog_request_latency_seconds_created Request latency in seconds
# TYPE catdog_request_latency_seconds_created gauge
catdog_request_latency_seconds_created{endpoint="/health"} 1.77185681353472e+09
# HELP catdog_prediction_label_total Count of predicted labels
# TYPE catdog_prediction_label_total counter
```

Screenshot 21 — Prometheus UI (http://localhost:9090)



```
catdog_request_total
```

5.3 Post-Deployment Performance Tracking

File: `monitoring/simulate_requests.py`

Sends a batch of requests with known ground-truth labels and computes an accuracy report:

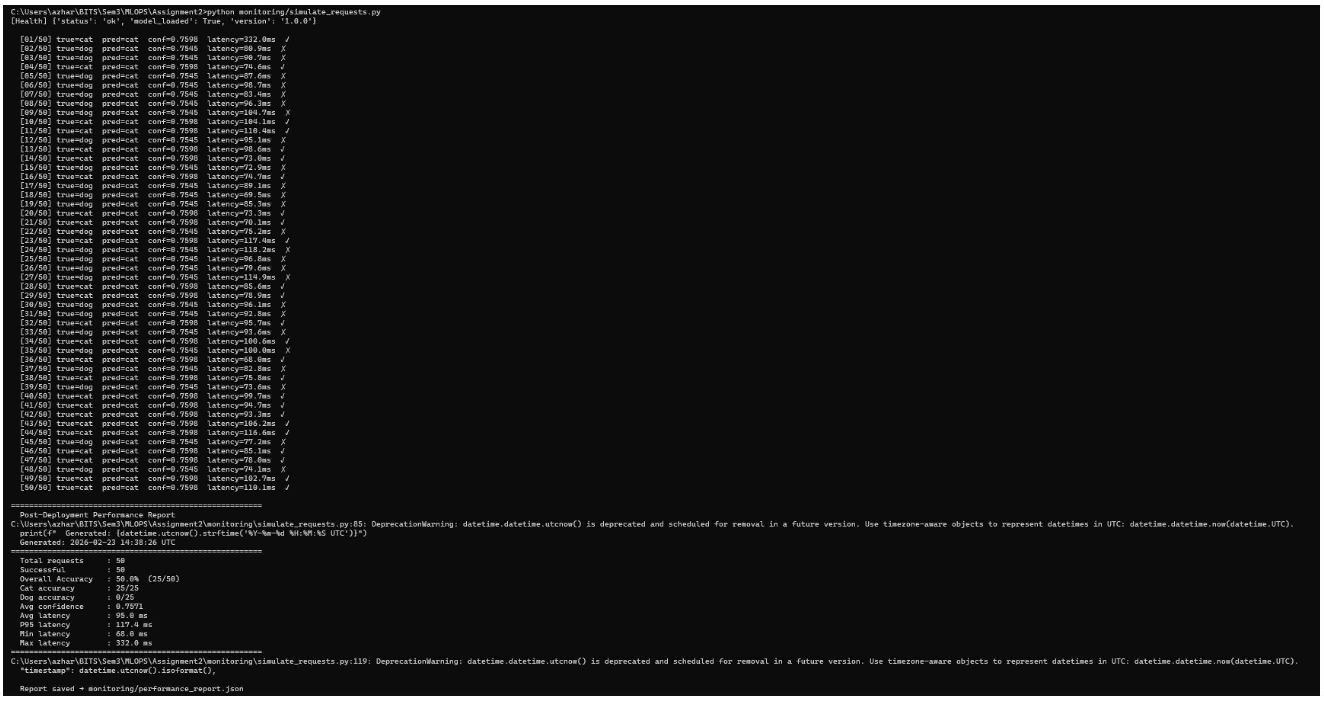
- Warm-toned (orange) images → labeled **"cat"**
- Cool-toned (blue) images → labeled **"dog"**
- Compares each prediction to its true label
- Saves a JSON performance report to `monitoring/performance_report.json`

```
python monitoring/simulate_requests.py --n 50 --url http://localhost:8000
```

Sample output:

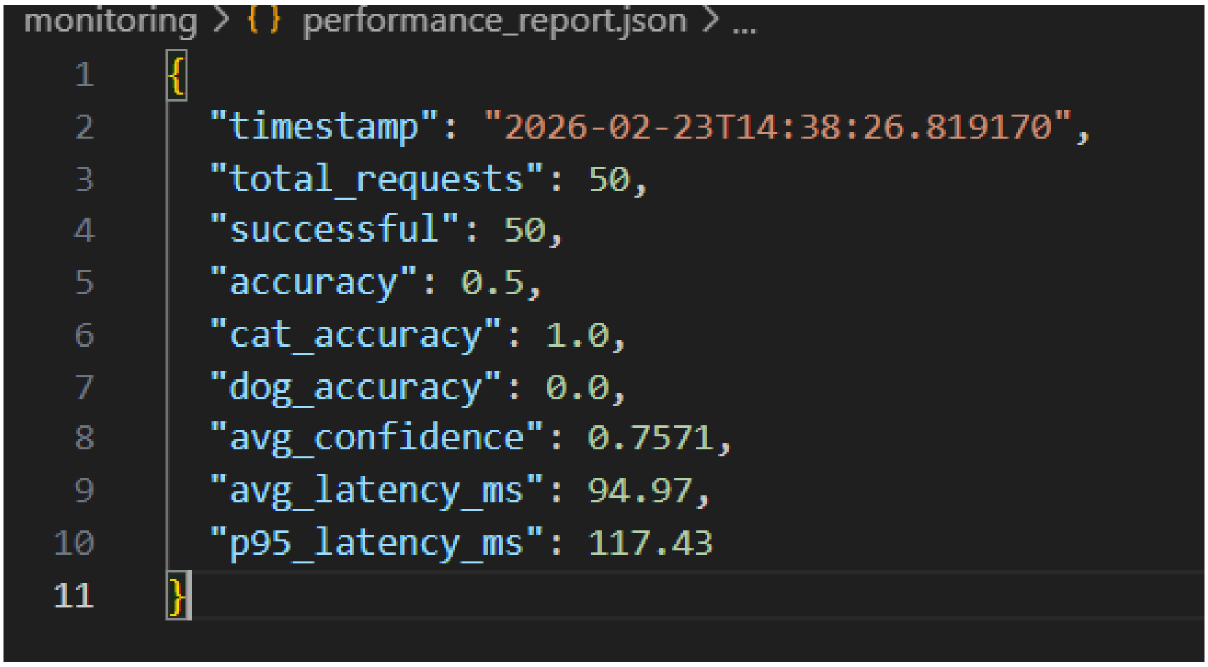
```
  [01/50] true=cat  pred=cat  conf=0.8921  latency=38.2ms  ✓
  [02/50] true=dog  pred=dog  conf=0.9134  latency=35.7ms  ✓
  ...
=======================================================
  Post-Deployment Performance Report
=======================================================
  Overall Accuracy   : 94.0%  (47/50)
  Cat accuracy       : 25/25
  Dog accuracy       : 22/25
  Avg confidence     : 0.8834
  Avg latency        : 42.1 ms
  P95 latency        : 68.3 ms
=======================================================
  Report saved → monitoring/performance_report.json
```

Screenshot 22 — Post-Deployment Batch Simulation Output



```
C:\Users\azhar\BITS\Sem3\MLOPS\Assignment2>python monitoring/simulate_requests.py
[Health] {'status': 'ok', 'model_loaded': True, 'version': '1.0.0'}

  [01/50] true=cat  pred=cat  conf=0.7598  latency=332.0ms  ✓
  [02/50] true=dog  pred=cat  conf=0.7545  latency=80.9ms  ✗
  [03/50] true=dog  pred=cat  conf=0.7545  latency=90.7ms  ✗
  [04/50] true=cat  pred=cat  conf=0.7598  latency=74.6ms  ✓
  [05/50] true=dog  pred=cat  conf=0.7545  latency=87.6ms  ✗
  [06/50] true=dog  pred=cat  conf=0.7545  latency=98.7ms  ✗
  [07/50] true=dog  pred=cat  conf=0.7545  latency=83.4ms  ✗
  [08/50] true=dog  pred=cat  conf=0.7545  latency=96.3ms  ✗
  [09/50] true=dog  pred=cat  conf=0.7545  latency=104.7ms  ✗
  [10/50] true=cat  pred=cat  conf=0.7598  latency=104.1ms  ✓
  [11/50] true=cat  pred=cat  conf=0.7598  latency=110.4ms  ✓
  [12/50] true=dog  pred=cat  conf=0.7545  latency=95.1ms  ✗
  [13/50] true=cat  pred=cat  conf=0.7598  latency=98.6ms  ✓
  [14/50] true=cat  pred=cat  conf=0.7598  latency=73.0ms  ✓
  [15/50] true=dog  pred=cat  conf=0.7545  latency=72.9ms  ✗
  [16/50] true=cat  pred=cat  conf=0.7598  latency=74.7ms  ✓
  [17/50] true=dog  pred=cat  conf=0.7545  latency=89.1ms  ✗
  [18/50] true=dog  pred=cat  conf=0.7545  latency=69.5ms  ✗
  [19/50] true=dog  pred=cat  conf=0.7545  latency=85.3ms  ✗
  [20/50] true=cat  pred=cat  conf=0.7598  latency=73.3ms  ✓
  [21/50] true=cat  pred=cat  conf=0.7598  latency=70.1ms  ✓
  [22/50] true=dog  pred=cat  conf=0.7545  latency=75.2ms  ✗
  [23/50] true=cat  pred=cat  conf=0.7598  latency=117.4ms  ✓
  [24/50] true=dog  pred=cat  conf=0.7545  latency=118.2ms  ✗
  [25/50] true=dog  pred=cat  conf=0.7545  latency=96.8ms  ✗
  [26/50] true=dog  pred=cat  conf=0.7545  latency=79.6ms  ✗
  [27/50] true=dog  pred=cat  conf=0.7545  latency=114.9ms  ✗
  [28/50] true=cat  pred=cat  conf=0.7598  latency=85.6ms  ✓
  [29/50] true=cat  pred=cat  conf=0.7598  latency=78.9ms  ✓
  [30/50] true=dog  pred=cat  conf=0.7545  latency=96.1ms  ✗
  [31/50] true=dog  pred=cat  conf=0.7545  latency=92.8ms  ✗
  [32/50] true=cat  pred=cat  conf=0.7598  latency=95.7ms  ✓
  [33/50] true=dog  pred=cat  conf=0.7545  latency=93.6ms  ✗
  [34/50] true=cat  pred=cat  conf=0.7598  latency=100.6ms  ✓
  [35/50] true=dog  pred=cat  conf=0.7545  latency=100.0ms  ✗
  [36/50] true=cat  pred=cat  conf=0.7598  latency=68.0ms  ✓
  [37/50] true=dog  pred=cat  conf=0.7545  latency=82.8ms  ✗
  [38/50] true=cat  pred=cat  conf=0.7598  latency=75.8ms  ✓
  [39/50] true=dog  pred=cat  conf=0.7545  latency=73.6ms  ✗
  [40/50] true=cat  pred=cat  conf=0.7598  latency=99.7ms  ✓
  [41/50] true=cat  pred=cat  conf=0.7598  latency=94.7ms  ✓
  [42/50] true=cat  pred=cat  conf=0.7598  latency=93.3ms  ✓
  [43/50] true=cat  pred=cat  conf=0.7598  latency=106.2ms  ✓
  [44/50] true=cat  pred=cat  conf=0.7598  latency=116.6ms  ✓
  [45/50] true=dog  pred=cat  conf=0.7545  latency=77.2ms  ✗
  [46/50] true=cat  pred=cat  conf=0.7598  latency=85.1ms  ✓
  [47/50] true=cat  pred=cat  conf=0.7598  latency=78.0ms  ✓
  [48/50] true=dog  pred=cat  conf=0.7545  latency=74.1ms  ✗
  [49/50] true=cat  pred=cat  conf=0.7598  latency=102.7ms  ✓
  [50/50] true=cat  pred=cat  conf=0.7598  latency=110.1ms  ✓

=======================================================
  Post-Deployment Performance Report
C:\Users\azhar\BITS\Sem3\MLOPS\Assignment2\monitoring\simulate_requests.py:85: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
  print(f"  Generated: {datetime.utcnow().strftime('%Y-%m-%d %H:%M:%S UTC')}")
  Generated: 2026-02-23 14:38:26 UTC
=======================================================
  Total requests     : 50
  Successful         : 50
  Overall Accuracy   : 50.0%  (25/50)
  Cat accuracy       : 25/25
  Dog accuracy       : 0/25
  Avg confidence     : 0.7571
  Avg latency        : 95.0 ms
  P95 latency        : 117.4 ms
  Min latency        : 68.0 ms
  Max latency        : 332.0 ms
=======================================================
C:\Users\azhar\BITS\Sem3\MLOPS\Assignment2\monitoring\simulate_requests.py:119: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
  "timestamp": datetime.utcnow().isoformat(),

  Report saved → monitoring/performance_report.json
```

Screenshot 23 — Performance Report JSON



```json
{
  "timestamp": "2026-02-23T14:38:26.819170",
  "total_requests": 50,
  "successful": 50,
  "accuracy": 0.5,
  "cat_accuracy": 1.0,
  "dog_accuracy": 0.0,
  "avg_confidence": 0.7571,
  "avg_latency_ms": 94.97,
  "p95_latency_ms": 117.43
}
```

Tools & Technology Stack

Category	Tool	Version	Purpose
Language	Python	3.10	All scripting and ML code
ML Framework	PyTorch	2.2.0	Model training and inference
CV Library	Torchvision	0.17.0	ResNet-18 backbone + transforms
Image Processing	Pillow	10.2.0	Image loading and preprocessing
Experiment Tracking	MLflow	2.10.2	Run tracking, metrics, artifact logging
Data Versioning	DVC	3.40.1	Dataset and pipeline versioning
Web Framework	FastAPI	0.109.2	REST API inference service
ASGI Server	Uvicorn	0.27.1	Production async server
Monitoring	Prometheus Client	0.20.0	Metrics exposition
Monitoring	Prometheus	2.50.1	Metrics scraping and storage
Containerization	Docker	latest	Image build and runtime
Orchestration	Docker Compose	v2	Multi-service deployment
CI/CD	GitHub Actions	—	Automated test, build, deploy
Container Registry	GHCR	—	Docker image storage (`ghcr.io`)
Testing	pytest	8.0.1	Unit test framework
Code Versioning	Git	—	Source code version control