

Configuration Manual

Automated Efficient ML Model Retraining Using Cloud-Native MLOps Tools

Brindavan Samuthira Pandian Ganesan
Student ID: x23430320

M.Sc. Research Project

School of Computing
National College of Ireland

Supervisor: Mr. Sean Heeney

National College of Ireland

Project Submission Sheet

Student Name: Brindavan Samuthira Pandian Ganesan

Student ID: x23430320

Programme: M.Sc. in Cloud Computing **Year:** 2025

Module: MSCCLOUD Research Project

Lecturer: Mr. Sean Heeney

Submission Due Date: 11/12/2025

Project Title: Automated Efficient ML Model Retraining Using Cloud-Native MLOps Tools

Word Count: 997

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the references section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature: Brindavan Samuthira Pandian Ganesan

Date: 04/12/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. Projects should be submitted to your Programme Coordinator.
3. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**
5. All projects must be submitted and passed in order to successfully complete the year. **Any project/assignment not submitted will be marked as a fail.**
6. **Please check that you read AI and Academic Integrity Acknowledgement Supplements in this document**

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

1 Introduction

This configuration manual provides a structured and practical guide for deploying, operating, and monitoring the cloud-native retraining pipeline developed as part of the project *Automated Efficient ML Model Retraining by MLOps Using Cloud-Native Tools*. The system integrates multiple datasets, data versioning, automated retraining, model evaluation, model promotion, containerisation, and a fully end-to-end automated deployment.

Each section outlines the technical rationale behind a component, followed by step-by-step instructions to support reproducibility. Where screenshots are required, placeholders are clearly indicated for later insertion.

The system is built around the following core technologies:

- AWS SageMaker for orchestrated retraining.
- DVC + S3 for dataset versioning and reproducible pipelines.
- GitHub Actions for CI/CD automation.
- AWS ECS (Fargate) for hosting the FastAPI inference service.
- Amazon ECR for container storage.
- Prometheus + Pushgateway for telemetry.
- Grafana dashboards for drift and performance monitoring.
- MLflow for experiment tracking.

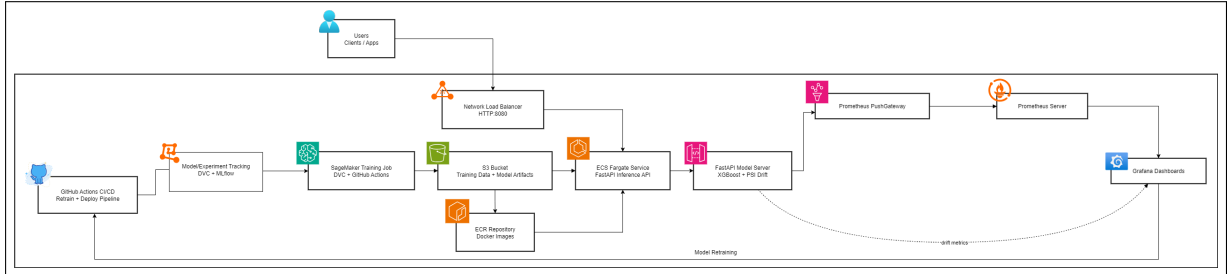


Figure 1: GitHub Actions Pipeline

2 Prerequisites

2.1 Local Machine Requirements

To execute components of the pipeline locally, ensure the following tools are installed:

- Python 3.10+
- Docker Desktop
- AWS CLI v2
- Git
- DVC with S3 support: `pip install dvc[s3]`
- Terraform (optional for infrastructure provisioning)

2.2 AWS Requirements

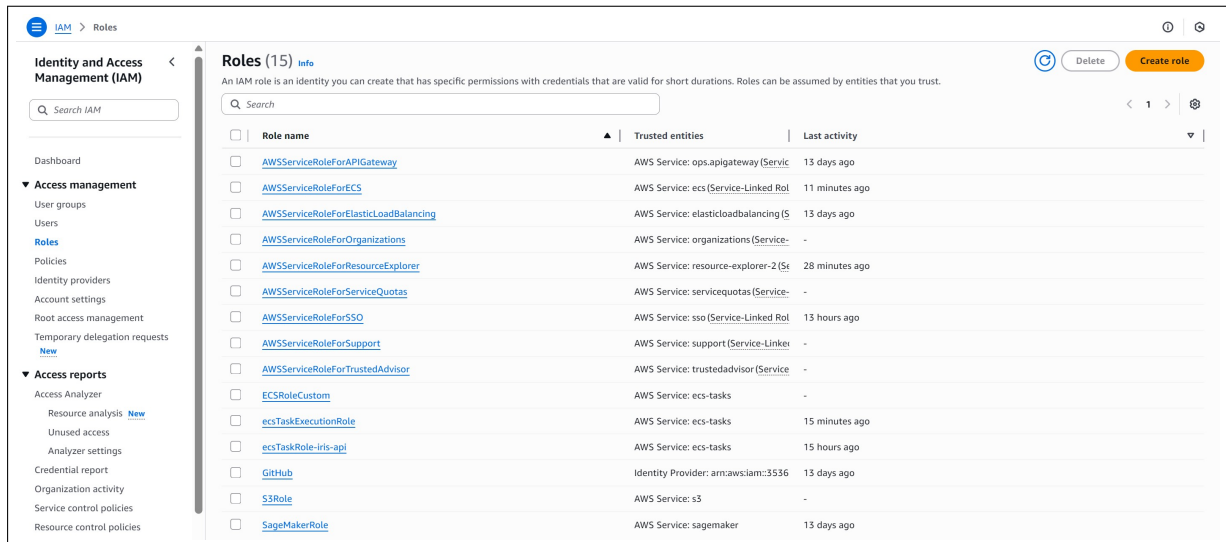
The AWS environment requires:

- An active AWS account.
- IAM role for SageMaker with S3 and ECR permissions.
- IAM role for ECS task execution.
- S3 bucket for:
 - Dataset storage
 - DVC remote storage
 - SageMaker training outputs
- ECR repository for storing API container images.

2.3 Required Permissions

Roles should include:

- AmazonS3FullAccess (scoped to project bucket)
- AmazonEC2ContainerRegistryFullAccess
- AmazonSageMakerFullAccess
- AmazonECSFullAccess



The screenshot shows the AWS IAM console 'Roles' page. It lists 15 roles with columns for Role name, Trusted entities, and Last activity. The roles include various AWS service roles like APIGateway, ECS, ElasticLoadBalancing, Organizations, ResourceExplorer, ServiceQuotas, SSO, Support, TrustedAdvisor, and custom roles like ECSCustom, ECSTaskExecutionRole, and S3Role. The SageMakerRole is also listed at the bottom.

Role name	Trusted entities	Last activity
AWSServiceRoleForAPIGateway	AWS Service: ops.apigateway (Service-Linked Role)	13 days ago
AWSServiceRoleForECS	AWS Service: ecs (Service-Linked Role)	11 minutes ago
AWSServiceRoleForElasticLoadBalancing	AWS Service: elasticloadbalancing (Service-Linked Role)	13 days ago
AWSServiceRoleForOrganizations	AWS Service: organizations (Service-Linked Role)	-
AWSServiceRoleForResourceExplorer	AWS Service: resource-explorer-2 (Service-Linked Role)	28 minutes ago
AWSServiceRoleForServiceQuotas	AWS Service: servicequotas (Service-Linked Role)	-
AWSServiceRoleForSSO	AWS Service: sso (Service-Linked Role)	13 hours ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	-
ECSCustom	AWS Service: ecs-tasks	-
ecstaskExecutionRole	AWS Service: ecs-tasks	15 minutes ago
ecstaskRole-iris-api	AWS Service: ecs-tasks	15 hours ago
GitHub	Identity Provider: arn:aws:iam::3536:oidc-provider/github	13 days ago
S3Role	AWS Service: s3	-
SageMakerRole	AWS Service: sagemaker	13 days ago

Figure 2: IAM Roles for Permissions

3 Configuring DVC and Dataset Storage

DVC enables reproducible experiments by versioning data alongside code. The project uses an AWS S3 bucket as the DVC remote.

3.1 Initialising DVC

Run the following command in the project root:

```
dvc init
```

3.2 Adding the Dataset

The Adult Income dataset is added and tracked via:

```
dvc add data/adult.csv
```

This generates:

- data/adult.csv.dvc - Automatic entry in .gitignore

3.3 Configuring DVC Remote

Set the remote S3 bucket:

```
dvc remote add -d s3remote s3://thebrowntiger/dvcstore  
dvc push
```

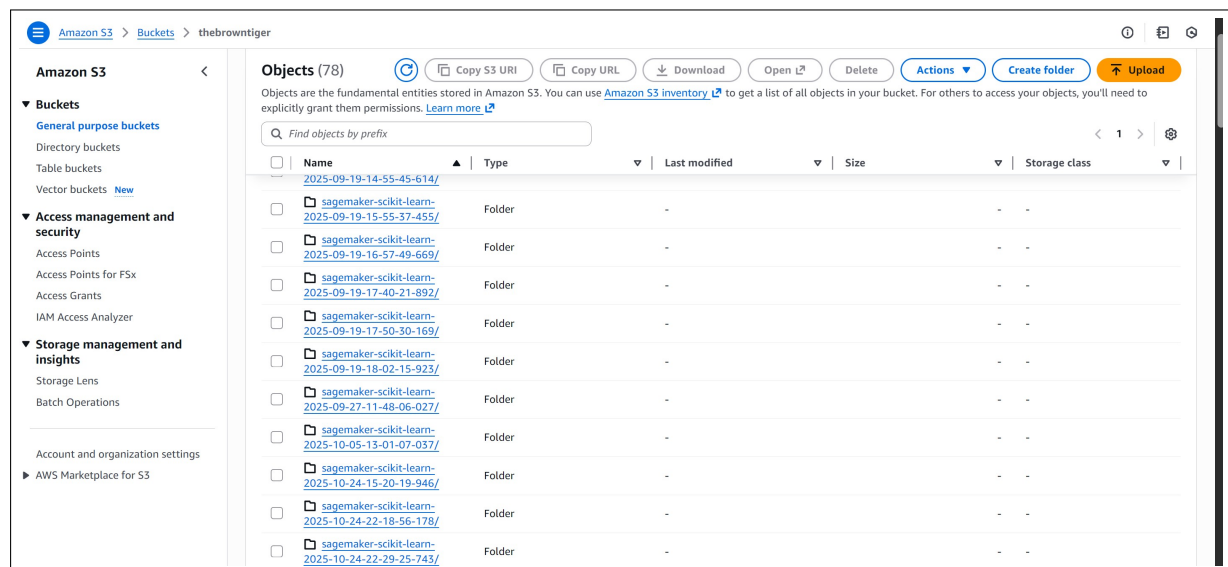


Figure 3: Trained Models

4 Configuring SageMaker Training Pipeline

The training job is executed through a BYOC (Bring-Your-Own-Container) image. The training script is located under training/train.py.

4.1 Training Container

A Dockerfile inside the training folder builds the image:

```
docker build -t thesis-training:v06 .  
docker tag thesis-training:v06 353671347542.dkr.ecr.eu-west-1.  
amazonaws.com/thesis-training:v06  
docker push ...
```

4.2 Running Training via DVC

DVC stage configuration:

```
stages:
  train:
    cmd: python pipeline/run_training_job.py
    deps:
      - training/train.py
      - data/adult.csv
    outs:
      - artifacts/model/model.tar.gz
      - artifacts/metrics/metrics.json
```

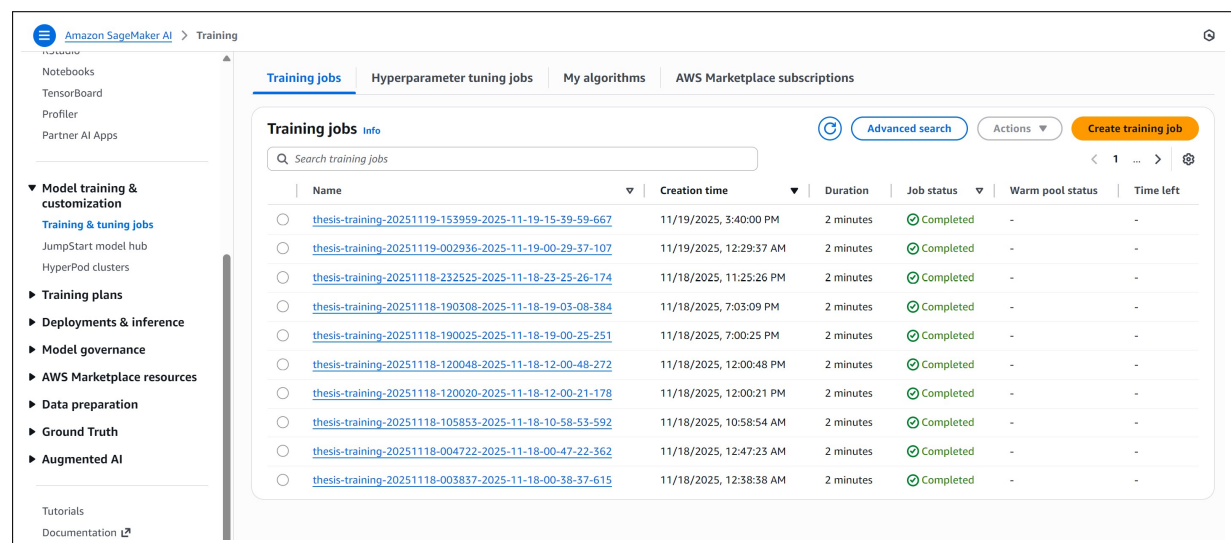


Figure 4: DVC tracking

5 Evaluation Workflow

The model is evaluated using a SageMaker Processing job. The evaluation script calculates accuracy and F1-score on the Adult Income test split.

5.1 Running Evaluation

Command triggered through DVC:

```
dvc repro
```

5.2 MLflow Integration

The manual includes MLflow for experiment lineage.

To track metrics locally:

```
mlflow ui
```

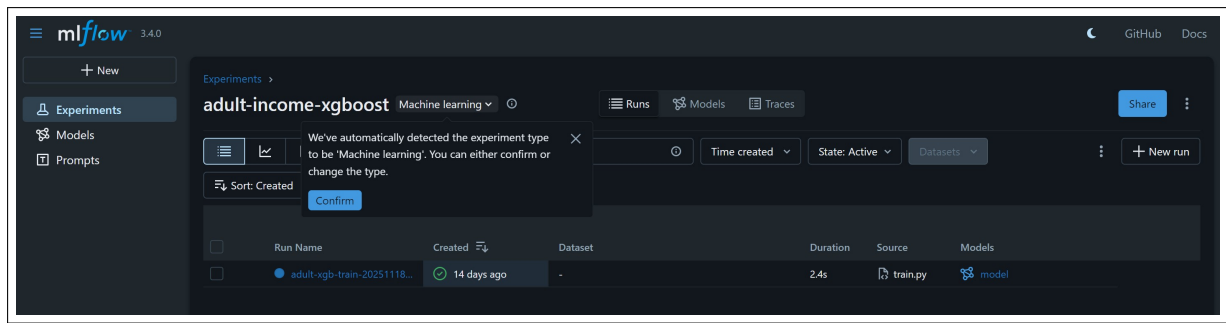


Figure 5: Model Experiments

6 API Containerisation

The FastAPI inference service is packaged into a Docker image and deployed on ECS Fargate.

```
infra >  ecs_service.tf
You, 2 weeks ago | 1 author (You)
1 resource "aws_ecs_service" "iris_service" {
2     name                = "iris-api-service"
3     cluster              = aws_ecs_cluster.iris_cluster.id
4     launch_type          = "FARGATE"
5     desired_count        = 1
6     platform_version     = "LATEST"
7
8     deployment_controller {
9         type = "ECS"
10    }
11
12    health_check_grace_period_seconds = 30
13    You, last month • tf clean-1 ...
14
15    task_definition = aws_ecs_task_definition.iris_task.arn
16
17    network_configuration {
18        subnets          = [aws_subnet.public_a.id, aws_subnet.public_b.id]
19        security_groups    = ["sg-01465797f95e42fbb"] # existing SG
20        assign_public_ip   = true
21    }
22
23    load_balancer {
24        target_group_arn = aws_lb_target_group.iris_nlb_tg.arn
25        container_name   = "iris-api"
26        container_port    = 8080
27    }
28
29    deployment_minimum_healthy_percent = 100
30    deployment_maximum_percent        = 200
31
32    depends_on = [
33        aws_lb_listener.iris_nlb_listener
34    ]
35 }
36
```

Figure 6: IaC setting up the AWS environment

6.1 Build and Push the Image

```
docker build -t iris-api:v14 .
docker tag iris-api:v02 353671347542.dkr.ecr.eu-west-1.amazonaws.com/iris-api:v14
docker push ...
```

6.2 Testing the Container Locally

```
uvicorn app.main:app --host 0.0.0.0 --port 8080
```

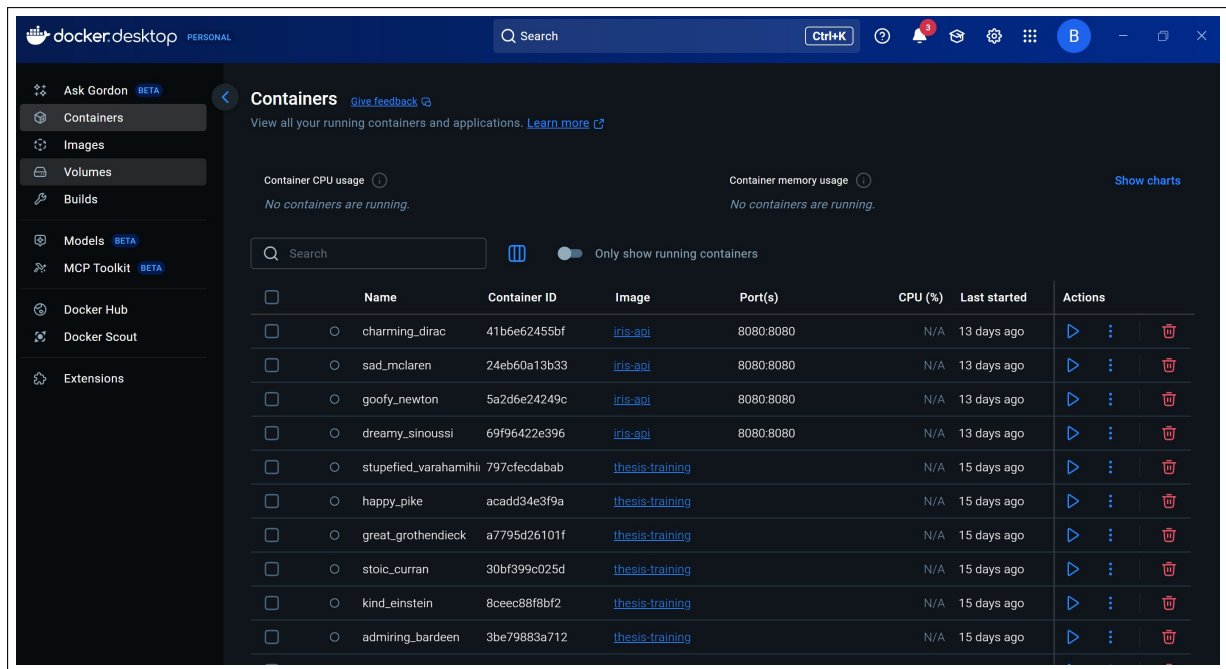


Figure 7: Docker Containers

7 ECS Deployment

The ECS Fargate service hosts the inference API behind an Application Load Balancer.

7.1 Key Components

- Cluster: iris-api-cluster
- Service: iris-api-service
- Task Definition: 512 CPU / 1GB RAM
- Listener: Port 80

7.2 Updating the Running API

```
aws ecs update-service \
  --cluster iris-api-cluster \
  --service iris-api-service \
  --force-new-deployment
```

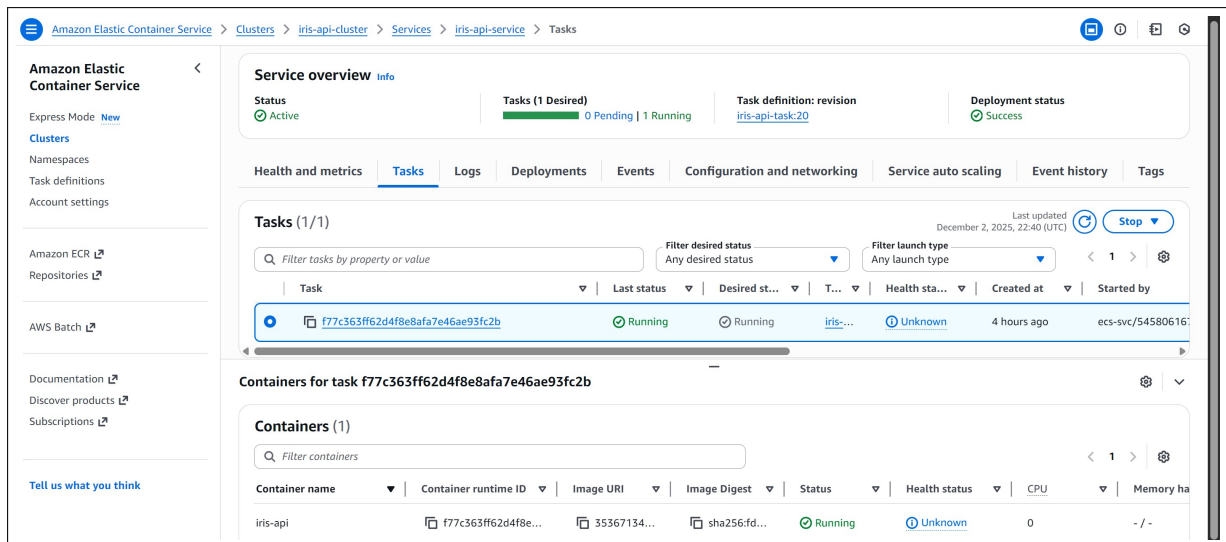


Figure 8: ECS Cluster Console

8 Monitoring Setup

Prometheus scrapes the FastAPI container's `/metrics` endpoint and visualises drift signals on Grafana dashboards.

8.1 Pushgateway Configuration

The API pushes metrics to the Pushgateway using:

```
push_to_gateway(http://108.130.158.94:9091/, job="
  adult_income_api", registry=registry)
```

8.2 Prometheus Configuration

Prometheus scrape job configuration:

```
scrape_configs:
  - job_name: 'adult_income_api'
    static_configs:
      - targets: ['http://108.130.158.94:9091/']
```

8.3 Grafana Dashboard

Dashboard panels include:

- Drift Score (PSI)
- Prediction Distribution
- Feature-level PSI values
- Prediction Count

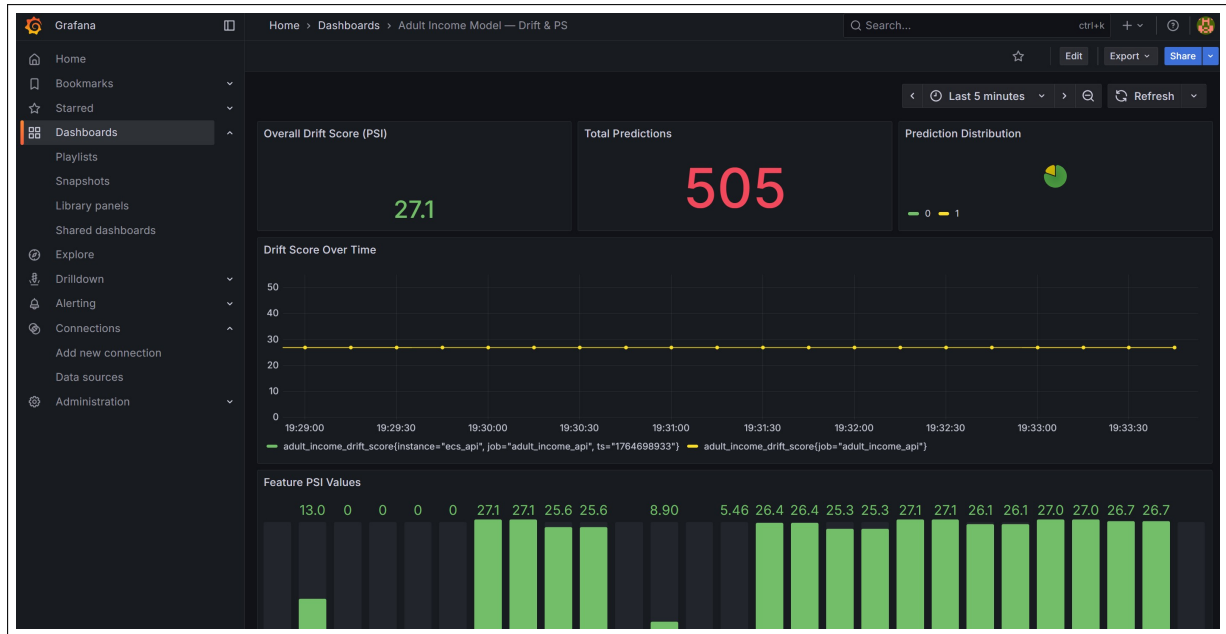


Figure 9: Grafana Dashboard

9 GitHub Actions Pipeline

The CI/CD pipeline automates retraining, evaluation, model validation, container build, and ECS deployment.

9.1 Trigger Conditions

Pipeline triggers on:

- Push to main branch
- Manual workflow dispatch

9.2 Primary Steps in CI/CD

1. AWS authentication via OIDC
2. DVC dataset pull
3. Run tests
4. Trigger DVC training and evaluation
5. Model accuracy comparison
6. Container build and push
7. ECS deployment

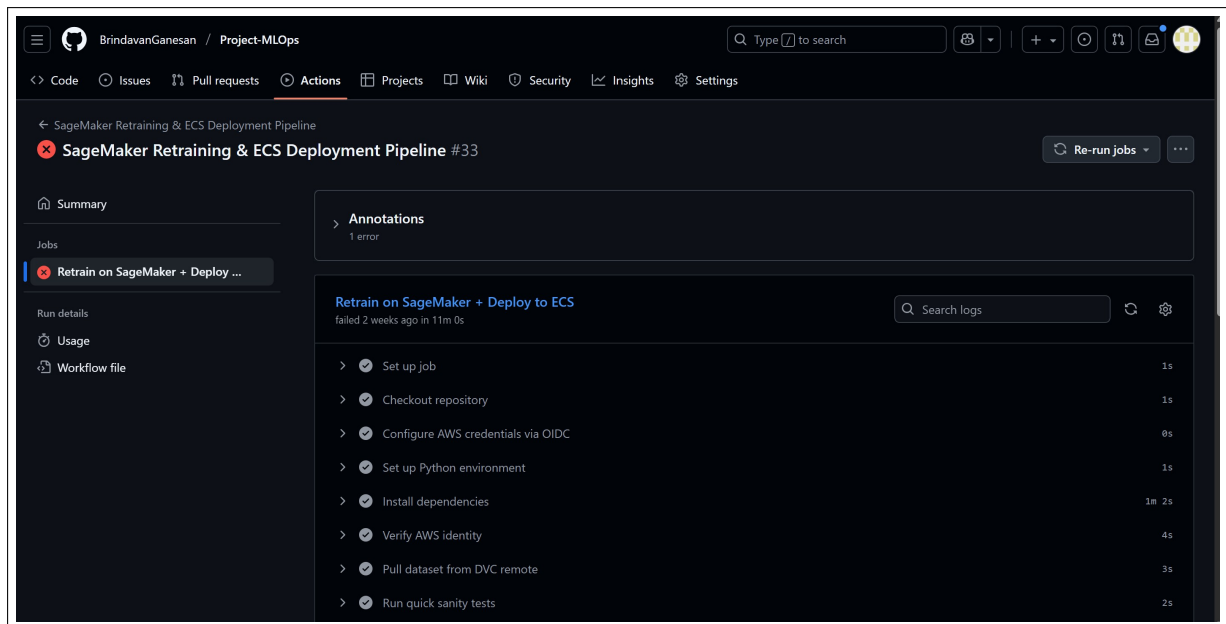


Figure 10: Github Action Pipeline

10 Validation and Testing Steps

10.1 API Health Check

```
curl http://iris-nlb-a5756f62e3cf9fb9.elb.eu-west-1.amazonaws.com/ping
```

10.2 Prediction Request

```
curl -X POST http://iris-nlb-a5756f62e3cf9fb9.elb.eu-west-1.
amazonaws.com/predict -d '{
  "records": [{
    "Age": 39,
    "Workclass": "Private",
    ...
  }]}'
```

10.3 Grafana Validation

Check that:

- Drift values update after each inference request.
- Feature-level PSI is visible.
- Prediction counters increase.

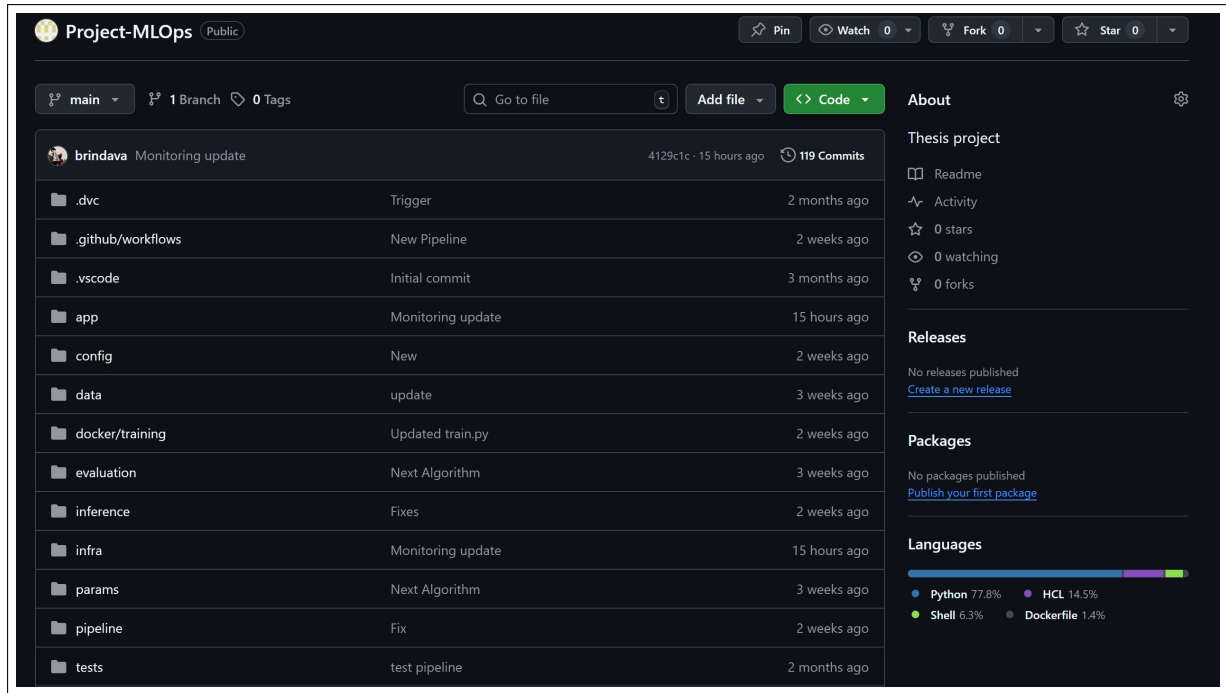


Figure 11: GitHub Repo

GitHub Repo: <https://github.com/BrindavanGanesan/Project-MLOps.git>

11 Conclusion

This configuration manual provides a step-by-step guide to setting up and running the automated retraining pipeline. By following the described process, the system can be reproduced reliably across different environments.

12 References

- (2023). Data version control (dvc) documentation. <https://dvc.org/doc>.
- (2023). Grafana observability platform documentation. <https://grafana.com/docs>.
- (2023). Mlops community resources and references. <https://ml-ops.org/content/references.html>.
- (2023). AWS SageMaker AI documentation. <https://docs.aws.amazon.com/sagemaker/>
- (2023). AWS ECS documentation. <https://docs.aws.amazon.com/ecs/>
- (2020). Prometheus documentation. <https://prometheus.io/docs/introduction/overview/>
- (2021). GitHub actions. <https://docs.github.com/en/actions/>