Capstone Project: End-to-End Machine Learning System on AWS (Free Tier)

Deadline: 24 August, 2025 (7:00 PM MMT)

Contents

1 Part I — Project Specification (Guidelines)

1.1 Overview

Each team will build and deploy a complete machine learning system on AWS using the Free Tier. The project must simulate a real-world MLOps pipeline, from raw data ingestion to model deployment.

1.2 Team Setup

- Each team has a **Team Leader** with full IAM permissions.
- Other members get limited contributor roles.
- All work must stay within AWS Free Tier quotas.

1.3 Recommended Datasets

Teams may choose **one**:

Dataset	Type	Size
Titanic Survival	Binary classification	891 rows
Boston Housing	Regression	506 rows
Iris Dataset	Multi-class classification	150 rows
Wine Quality	Binary classification	1600 rows
Synthetic	Any (generated)	custom

1.4 Project Requirements

1.4.1 Data Pipeline

- S3 bucket structure:

```
s3://project-bucket/
    raw/
    processed/
    models/
```

- Lambda function for **data validation** (schema, nulls, ranges)
- Feature engineering and preprocessing scripts
- S3 event triggers → Lambda → processing workflow

1.4.2 Containerized ML Training & Inference

- Create a `Dockerfile` for your ML environment
- Train **Model v1.0** (e.g. Linear Regression)
- Push container image to **Amazon ECR**
- Deploy container as a **Lambda function** (inference endpoint)

1.5 Repository Structure

```
ml-project/
 data/
    raw/
    processed/
    README.md
 src/
    preprocess.py
    train.py
    predict.py
    utils.py
 infrastructure/
    Dockerfile
    requirements.txt
    deploy.sh
 tests/
    test_api.py
    test_model.py
 docs/
     API.md
     SETUP.md
     ARCHITECTURE.md
```

1.6 Security and Cost Constraints

IAM & Security

- Use the **least privilege** principle.
- Team Leader gets `AdministratorAccess`.
- Others get limited roles (read-only, Lambda-deploy).
- Enable CloudTrail logging.

AWS Free Tier Limits (per group)

- Lambda: 1M requests/month
- S3: 500MB storage

- API Gateway: 1M calls/month
- ECR: 500MB storage
- CloudWatch: basic metrics included

1.7 Deliverables

1. AWS Architecture Diagram
2. IAM Security Plan
3. Cost Analysis Report
4. Deployment Guide (`SETUP.md`)
5. GitHub Repository with required structure
6. 5–10 minute Demo Video

1.8 Assessment Criteria

Criteria	**Weight**
Functionality (end-to-end pipeline)	40%
Code Quality (clean, tested, documented)	20%
AWS Integration (correct service usage)	25%
Documentation (clear, reproducible)	15%

1.9 Suggested Tasks

Task	**Milestone**
1	Team formation, dataset selection, S3 setup
2	Data pipeline + validation Lambda
3	v1 model training + containerized Lambda
5	Documentation, cost analysis, final demo

1.10 Expectations and Warnings

- All work must be reproducible from scratch.
- No local-only code — everything must be in GitHub.
- If you hit Free Tier limits, refactor or clean up resources.
- Prioritize **simplicity and reliability** over complexity.

2 Part II — Submission Template (Fill This In)

2.1 Team Info

- Team Name: [Team Name]
- Team Leader: [Name]
- Members: [Names]
- GitHub Repo: `<link>`
- Demo Video: `<link>`

2.2 Executive Summary

- Dataset used: [...]
- Task type: [...]
- Deployed model versions: [...]
- Endpoint: `<InsertAPIGatewayURL>`

System Description: (2–3 paragraphs)

2.3 AWS Architecture Diagram

architecture_diagram.png

Component Descriptions:

- S3 buckets
- Lambda functions
- API Gateway
- ECR
- CloudWatch

2.4 IAM Security Plan

Role	User(s)	Permissions/Policies

Team Leader Developer	[...] [...]	AdministratorAccess LambdaFullAccess, S3ReadOnly, CloudWatchLogsFullAccess

2.5 Cost Analysis

Service	**Usage**	**Free Tier Limit**	**Within Limit?**
Lambda		1M	
S3		500MB	
ECR		500MB	
API Gateway		1M	
CloudWatch		Included	

2.6 Deployment Guide

1. Clone the repo
2. Create IAM roles
3. Create S3 buckets
4. Build and push Docker image to ECR
5. Deploy Lambda
6. Set up API Gateway
7. Test endpoint

2.7 Testing and Results

Example inference outputs

2.8 Reflection and Lessons Learned

Each member adds:

- Contributions
- Challenges
- Lessons learned