

Configuration Manual

MSc Research Project
Cloud Computing

Omkar Mahajan
Student ID: 23424427

School of Computing
National College of Ireland

Supervisor: Abubakr Siddig

National College of Ireland
Project Submission Sheet
School of Computing



Student Name:	Omkar Mahajan
Student ID:	23424427
Programme:	Cloud Computing
Year:	2025-2026
Module:	MSc Research Project
Supervisor:	Abubakr Siddig
Submission Due Date:	28/01/2026
Project Title:	Configuration Manual
Word Count:	1201
Page Count:	6

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 bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	28th January 2026

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission , to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project , both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

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

Configuration Manual

Enhancing Cloud Application Performance Through Edge AI Integration

Omkar Mahajan
23424427

Contents

1	Introduction	2
2	Prerequisites	2
3	AWS IAM User and CLI Setup	2
3.1	Creating the IAM User	3
3.2	Configuring the AWS CLI	3
4	Project Configuration (config.yaml)	3
4.1	Configuration Details	5
5	Execution Workflow	5
5.1	Step 1: Pre-Flight Check	5
5.2	Step 2: Run the Complete Workflow	5
5.3	Step 3: Run a Specific Step (Optional)	6
5.4	Step 4: Run the Flask application (Output)	6

1 Introduction

This document provides a comprehensive guide for configuring and running the "Edge AI Integration" project. The goal of this project is to demonstrate how application performance can be enhanced by intelligently offloading tasks between a cloud backend and an edge device using an AI-driven decision engine.

This manual covers prerequisites, AWS setup, project configuration, and the execution workflow. Following the procedures described in this document will allow others to repeat the processes that were performed in order to produce these experiments and validate the enhancement in performance due to intelligent offloading.

2 Prerequisites

Before proceeding, ensure your development environment meets the following requirements. You can validate most of these by running the pre-flight check script.

1. **Python:** Version 3.11 or newer is required.
2. **AWS Command Line Interface (CLI):** The AWS CLI needs to be installed and set up using credentials with the required permissions to build the required resources. (IAM, S3, Lambda, etc.).
3. **Python Packages:** All required Python packages must be installed. Install them using the `requirements.txt` file.

Install Dependencies

```
pip install -r requirements.txt
```

This command will install `boto3`, `PyYAML`, `requests`, `torch`, `Pillow`, and `numpy`.

4. **Virtual Environment Setup:** Create virtual environment in order to maintain dependency isolation.

Creating and Activating Virtual Environment

```
Windows
python -m venv venv
.\venv\Scripts\activate

Linux /macOS
python3 -m venv venv
source venv/bin/activate
```

3 AWS IAM User and CLI Setup

A dedicated IAM user with programmatic access is required to manage AWS resources.

3.1 Creating the IAM User

1. Go to the IAM console of your AWS account.
2. Create a new user (e.g., AdminUser).
3. Attach the **AdministratorAccess** policy to this user.
4. Entering the last screen, generate an access key and download the credentials

3.2 Configuring the AWS CLI

This step requires using the credentials of the step before to set up your local AWS CLI. This will allow the project scripts to interact with your AWS account.

Configure AWS CLI

```
aws configure
```

You will be prompted to enter the following:

- **AWS Access Key ID:** [Your-Access-Key-ID]
- **AWS Secret Access Key:** [Your-Secret-Access-Key]
- **Default region name:** us-east-1 (or your preferred region)
- **Default output format:** json

4 Project Configuration (config.yaml)

The core of the project's configuration is managed through the `config.yaml` file. This file defines the names and settings for all AWS resources that will be created.

Create a file named `config.yaml` in the root directory of the project with the following content.

config.yaml

```
# AWS General Configuration
aws:
  region: us-east-1
  account_id: null # Auto-detected, or set manually

# S3 Bucket Configuration
s3:
  artifacts_bucket: edge-ai-artifacts-${account_id}
  logs_bucket: edge-ai-logs-${account_id}
  model_prefix: models/
  component_prefix: greengrass-components/

# IAM Role Names
```

```

iam:
  lambda_role: EdgeAI-Lambda-Role
  greengrass_role: EdgeAI-Greengrass-Role
  iot_rule_role: EdgeAI-IoTRule-Role

# DynamoDB Configuration
dynamodb:
  results_table: EdgeAIInferenceResults

# Lambda Function Configuration
lambda:
  function_name: CloudInferenceFunction
  runtime: python3.11
  handler: lambda_function.lambda_handler
  timeout: 31
  memory: 512

# API Gateway Configuration
api_gateway:
  api_name: EdgeAI-API
  stage_name: v1
  route_key: POST /inference

# IoT and Greengrass Configuration
iot:
  thing_name: EdgeAIDevice
  thing_group_name: EdgeAIDeviceGroup
  policy_name: EdgeAIDevicePolicy
  mqtt_topic_prefix: edge/inference
  rule_name: EdgeAI_DynamoDB_Rule

# CloudWatch Configuration
cloudwatch:
  dashboard_name: EdgeAIPerformanceDashboard
  log_group_lambda: /aws/lambda/CloudInferenceFunction
  log_group_edge: /aws/greengrass/edge-ai-component

# Testing Configuration
testing:
  num_requests: 100
  concurrency: 10

# AI Offloading Engine Configuration
offloading:
  enabled: true
  cpu_high_threshold: 75.0
  cpu_critical_threshold: 90.0
  memory_high_threshold: 70.0
  memory_critical_threshold: 85.0
  latency_acceptable_ms: 100.0

```

```
latency_high_ms: 300.0
weights:
  device_load: 0.35
  network_condition: 0.25
  task_complexity: 0.20
  historical_performance: 0.20
```

4.1 Configuration Details

- **aws:** The scripts will attempt to auto-detect your `account_id`. If this fails, you must set it manually.
- **s3:** The `${account_id}` placeholder will be automatically replaced with your AWS account ID to ensure globally unique bucket names.
- **iam, dynamodb, lambda:** These sections define the names for the resources that will be created by the bootstrap scripts.
- **offloading:** These parameters tune the behavior of the AI decision engine.

5 Execution Workflow

Once the prerequisites and configuration are complete, you can run the project.

5.1 Step 1: Pre-Flight Check

First, run the validation script to confirm your setup is correct.

Run Pre-Flight Check

```
python preflight_check.py
```

Address any failures reported by this script before proceeding.

5.2 Step 2: Run the Complete Workflow

The main orchestration script, `main.py`, can run all project steps in sequence, from infrastructure creation to performance testing.

Execute Full Workflow

```
python main.py --all
```

5.3 Step 3: Run a Specific Step (Optional)

If you prefer to run the workflow step-by-step, you can execute individual scripts or use the `--step` flag with `main.py`.

Execute a Single Step (e.g. Bootstrap)

```
python main.py --step 1
```

This can be used to debug or re-run a particular section of the process, including the performance tests (`--step 6`).

5.4 Step 4: Run the Flask application (Output)

The monitoring dashboard and all performance metrics can be visualized by running the provided Flask web application

Start the Flask app

```
python app.py
```

References

Amazon Web Services (2024), 'Installing or updating the aws cli', <https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>.

Amazon Web Services (2025), 'Creating iam users', https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users_create.html.