

Configuration Manual

MSc Research Project
Cloud Computing

Alfi Thomas
23327821

School of Computing
National College of Ireland

Supervisor: Yasantha Samarawickrama

National College of Ireland
Project Submission Sheet
School of Computing



Student Name:	Alfi Thomas
Student ID:	23327821
Programme:	Msc Cloud Computing
Year:	2025-2026
Module:	Research
Supervisor:	Yasantha Samarawickrama
Submission Due Date:	11/12/2025
Project Title:	Configuration Manual
Word Count:	1,295
Page Count:	5

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:	Alfi Thomas
Date:	28/01/2026

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

Alfi Thomas
23327821

Contents

1	Introduction	2
2	System Prerequisites	2
2.1	AWS CLI Configuration	2
2.2	Python Dependencies	3
3	Project Configuration	3
3.1	Core AWS and Project Settings	3
3.2	Hybrid Switching Threshold	3
3.3	Resource Configuration	3
4	Deployment Procedure	3
4.1	Step 1: Launch the Interactive Menu	4
4.2	Step 2: Verify AWS Configuration (Recommended)	4
4.3	Step 3: Deploy the Complete Architecture	4
5	Operational Guide	4
5.1	Triggering the Pipeline	4
5.2	Running Automated Tests	4
5.3	Monitoring the System	5
6	Resource Cleanup	5
6.1	Step 1: Run the Cleanup Script	5
6.2	Step 2: Manual Verification	5

1 Introduction

The following provides step-by-step instructions on how to set up, deploy and operate the Serverless Hybrid Data Processing Architecture using Amazon Web Services (AWS). The Serverless Hybrid Data Processing Architecture uses intelligent workload routing between Amazon Web Services (AWS) Lambda and Amazon Web Services (AWS) Fargate to automatically process a range of CSV file sizes, in order to obtain maximum performance and cost effectiveness. This manual is intended for users with:

- An active AWS account with administrative privileges.
- AWS Command Line Interface (CLI) installed and configured on their local machine.
- Python 3.9+ and pip installed.
- Basic familiarity with AWS services (S3, IAM, Lambda, Fargate).

2 System Prerequisites

Before proceeding, ensure your local environment is correctly set up.

2.1 AWS CLI Configuration

When you run a deployment script, it uses the AWS Command Line Interface (CLI) to authenticate itself. To set it up correctly, run this command from your terminal and enter your AWS Access Key ID, Secret Access Key and default region when prompted:



```
a  
  
|| aws configure
```

You will be prompted for the following information:

- **AWS Access Key ID:** [Your_Access_Key_ID]
- **AWS Secret Access Key:** [Your_Secret_Access_Key]
- **Default region name:** eu-west-1
- **Default output format:** json

You can verify the configuration by running:



```
a  
  
|| aws sts get-caller-identity
```

This should return your AWS Account ID, User ID, and ARN without any errors.

2.2 Python Dependencies

The project scripts require the `boto3` library to interact with AWS APIs. Install it using `pip`:

P

```
|| ip install boto3
```

3 Project Configuration

All infrastructure parameters are centralized in the `config/config.py` file. This file serves as a single source for resource names, settings, and limits. Before publishing, review , review the basic parameters below and adjust them if necessary.

3.1 Core AWS and Project Settings

These settings define the AWS environment and the base names for all resources.

- `AWS_REGION`: The AWS region where all resources will be deployed.
- `ACCOUNT_ID`: (**CRITICAL**) You must replace the placeholder with your own 12-digit AWS Account ID. This is required for creating correct IAM policy ARNs.
- `PROJECT_NAME`: The base prefix for all created resources.
- `ENVIRONMENT`: A suffix to differentiate between deployment environments (e.g., 'dev', 'prod').

3.2 Hybrid Switching Threshold

This parameter controls the core logic of the architecture.

- `FILE_SIZE_THRESHOLD_MB`: Specifies the maximum file size (in MB) to be processed by AWS Lambda. Any files larger than, this are automatically sent to AWS Fargate.

3.3 Resource Configuration

These settings define the compute resources for Lambda and Fargate.

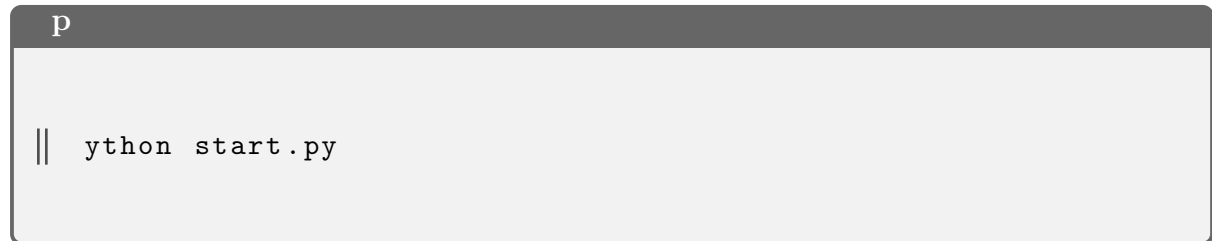
- `LAMBDA_MEMORY`: Memory allocated to the data processor Lambda function.
- `LAMBDA_TIMEOUT`: Maximum execution time for the Lambda function.
- `FARGATE_CPU` / `FARGATE_MEMORY`: CPU and memory allocated to the Fargate task.

4 Deployment Procedure

The entire architecture is deployed using an interactive command-line interface.

4.1 Step 1: Launch the Interactive Menu

Navigate to the root directory of the project in your terminal and run the main script:

A terminal window with a dark header bar containing the letter 'P'. The main area is light gray and shows the command `python start.py` preceded by a prompt character `||`.

This will present a menu of options.

4.2 Step 2: Verify AWS Configuration (Recommended)

Select **Option 1** (`[CONFIG] Verify AWS Configuration`). This will run a quick check to ensure your AWS CLI is properly authenticated.

4.3 Step 3: Deploy the Complete Architecture

Select **Option 2** (`[DEPLOY] Deploy Complete Architecture (All Steps)`).

- This executes the `setup_all.py` script, which orchestrates the deployment of all AWS resources in the correct order.
- The script provides real-time output for each step. The process typically takes 1-2 minutes.
- Upon successful completion, the script will display a summary of deployed resources.

5 Operational Guide

5.1 Triggering the Pipeline

To process a data file, upload any CSV file to the input S3 bucket under the `raw/` prefix.

- **Bucket Name:** `serverless-hybrid-thesis-input-data-dev`
- **Path:** `s3://serverless-hybrid-thesis-input-data-dev/raw/your_file.csv`

The pipeline will trigger automatically. The processed output will appear in the processed S3 bucket under a date-partitioned path.

5.2 Running Automated Tests

To validate the full pipeline functionality, including both Lambda and Fargate paths, run the automated test suite.

From the main menu (`start.py`), select **Option 4** (`[TEST] Test the Pipeline`).

- The script will automatically generate and upload small, medium, and large test files.
- It will wait for processing to complete and then print a summary report of performance and estimated cost for each scenario.

5.3 Monitoring the System

System health and performance can be monitored via the CloudWatch Dashboard created during deployment.

- **Dashboard Name:** `serverless-hybrid-thesis-dashboard`
- The URL to the dashboard is printed at the end of the deployment and test scripts.

Detailed execution logs are available in the Lambda, Step Functions, and ECS CloudWatch log sets.

6 Resource Cleanup

To avoid ongoing AWS loads, all created, resources should be deleted when they are no longer needed.

6.1 Step 1: Run the Cleanup Script

From the main menu (`start.py`), select **Option 7** (`'[CLEANUP] Cleanup All Resources'`).

- The script will list all the resources that will be deleted.
- You will be prompted to type `DELETE` to confirm the action. This is a safety measure to prevent accidental deletion.

6.2 Step 2: Manual Verification

Once the cleanup script has finished running, it is recommended that you sign in to the Amazon Web Services (AWS) Management Console to check if all the project resources (S3 buckets, Lambda Functions, IAM Roles, etc.) have been cleared out.