Configuration Manual

MSc Research Project
Programme Name

Abimanyu Murugan
Student ID: 23267062

School of Computing
National College of Ireland

Supervisor: Mr. Shreyas Setlur Arun

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Abimanyu Murugan
Student ID:	23267062
Programme:	MSc. Cloud Computing **Year:** 2024
Module:	MSc Research Project
Lecturer:	Mr. Shreyas Setlur Arun
Submission Due Date:	11/08/2025
Project Title:	Multi-Tenant Data Filtering in Fog Nodes for Privacy-Preserving IoT Data Processing
Word Count:	896 **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: Abimanyu Murugan

Date: 11/08/2025

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

Abimanyu Murugan
Student ID: 23267062

1 Prerequisites

- **Software**: Install Python , VS Code.
- **Access**: AWS Learner Academy account with LabRole and LabInstanceProfile or AWS account(free tier if available).
- **Network**: Stable internet connection.

2 Setting Up AWS on Your Local Computer

AWS (Amazon Web Services) is a cloud platform we'll use to run your project. Here's how to set it up:

1. **Install AWS CLI (Command Line Interface)**:
   - Open your web browser and go to https://aws.amazon.com/cli/ .
   - Click "Install" and follow the instructions to download and install AWS CLI on your computer (choose the version for your operating system, e.g., Windows, Mac, or Linux).
   - After installation, open your computer's command prompt or terminal.

2. **Configure AWS CLI in VS Code**:
   - Download and install VS Code from https://code.visualstudio.com/ if you don't have it.
   - Open VS Code, then open the terminal (click "Terminal" > "New Terminal").
   - Type "aws configure" and press Enter.
   - You'll need an Access Key ID and Secret Access Key:
     - Log in to your AWS account (or use the AWS Learner Academy account).
     - Go to "My Security Credentials" (under your account name) and create a new access key.
     - Copy the Access Key ID and Secret Access Key, then paste them when prompted in the terminal.
   - Enter your AWS region (e.g., us-east-1) and output format (choose json).
   - This links your computer to AWS.

.

3 Creating and Configuring an S3 Bucket

S3 is a storage service where we'll save your project data.

1. **Log in to AWS Management Console**:
   - Open https://aws.amazon.com/ in your browser and sign in.
2. **Navigate to S3**:

    - Click "Services" at the top, then select "S3" under "Storage".

3. **Create a Bucket**:
    - Click "Create Bucket".
    - Enter a unique name (e.g., 23267062-iot-data) and choose a region (e.g., us-east-1).
    - Uncheck "Block all public access" (we'll make it public later).
    - Click "Create bucket".

4. **Make Bucket Public**:
    - Click on your bucket name, go to the "Permissions" tab.
    - Edit the "Block public access" settings and uncheck all boxes, then save.
    - Under "Bucket policy", click "Edit" and paste the following policy (replace <your-bucket-name> with your bucket name and <your-aws-accountid> with your AWS account ID):

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject",
        "s3:ListBucket",
        "s3:GetBucketLocation",
        "s3:PutObjectAcl"
      ],
      "Resource": [
        "arn:aws:s3:::<your-bucket-name>",
        "arn:aws:s3:::<your-bucket-name>/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:PrincipalAccount": "<your-aws-accountid>"
        }
      }
    }
  ]
}
```

    - Click "Save changes" to make the bucket publicly accessible.

4 Setting Up Your Flask Project

Flask is a tool to create a web application for your project.

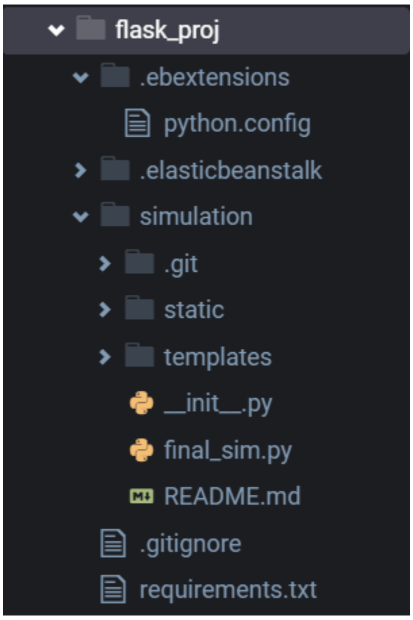
1. **Download EB CLI (Elastic Beanstalk Command Line Interface)**:
    - Go to https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/eb-cli3-install.html and follow the instructions to install EB CLI on your computer.
2. **Create Flask Project Directory**:
    - Open VS Code, click "File" > "New Folder", and name it flask_project.
    - Inside this folder, create a file named requirements.txt and add:

```
flask
gunicorn
pandas
cryptpandas
psutil
matplotlib
boto3
```

    - Create another file named .ebextensions/python.config and add:

```
packages:
  yum:
    git: []
container_commands:
  01_mkdir:
    command: "mkdir -p
/var/app/current/simulation/static"
  02_chmod:
    command: "chmod -R 755
/var/app/current/simulation/static"
option_settings:
  aws:elasticbeanstalk:container:python:
    WSGIPath: simulation.final_sim:app
  aws:elasticbeanstalk:environment:proxy:staticfiles:
    /static: simulation/static
```

3. **Download the source code from GitHub**:
    - Open your browser and go to https://github.com/x23267062/IOT_sim.git.
    - Click "Code" > "Download ZIP" and save it to your computer.
    - Extract the ZIP file into the flask_project folder, replacing any existing files if prompted.
    - In the line 20,21 in final_sim.py file change the S3 bucket name and region you created earlier.
    - Finally your project folder should be like the below:



```
flask_proj
  .ebextensions
    python.config
  .elasticbeanstalk
  simulation
    .git
    static
    templates
    __init__.py
    final_sim.py
    README.md
  .gitignore
  requirements.txt
```

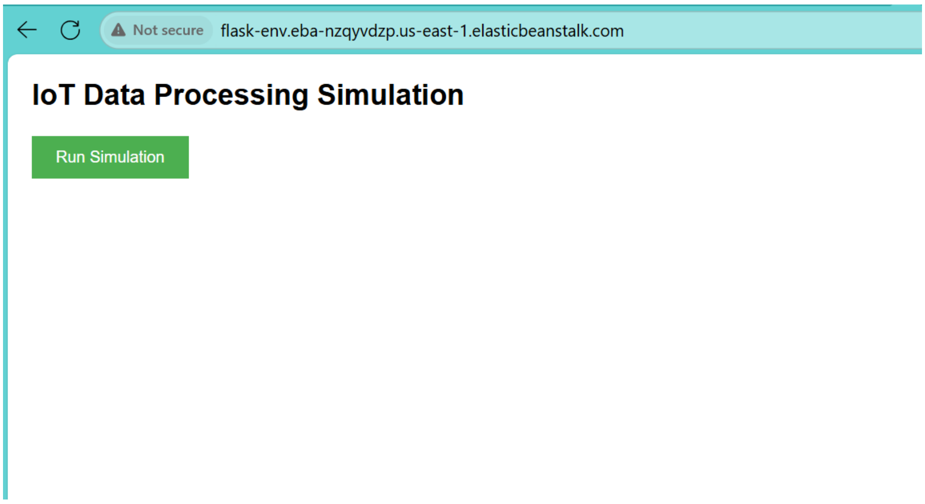
5 Deploying to Elastic Beanstalk

Elastic Beanstalk is a deployment service to run your web app on AWS.

1. **Initialize Elastic Beanstalk**:
   - Open your terminal in VS Code inside the flask_project folder.Download the ebcli using the command "*pip install awsebcli*"
   - Type "*eb init -r us-east-1 -p python-3.8 iot-simulation*" and press Enter.
     - -r us-east-1 sets the region, -p python-3.8 chooses the Python version, and iot-simulation is your application name.
   - Follow the prompts to select your AWS region and create the application.

2. **Create Environment**:
   - Type "*eb create iot-env --service-role LabRole --instance_profile LabInstanceProfile*" and press Enter.
     - iot-env is your environment name, and LabRole/LabInstanceProfile are default roles from the AWS Learner Academy account.Provide the service role of your account accordingly.
   - Wait a few minutes for the environment to set up.

3. **Check Deployment Status**:
   - Type "*eb status*" to see the environment status.
   - Note the domain link (e.g., iot-env-xxxxx.us-east-1.elasticbeanstalk.com) and open it in your browser to confirm it's accessible and the UI should be like below.

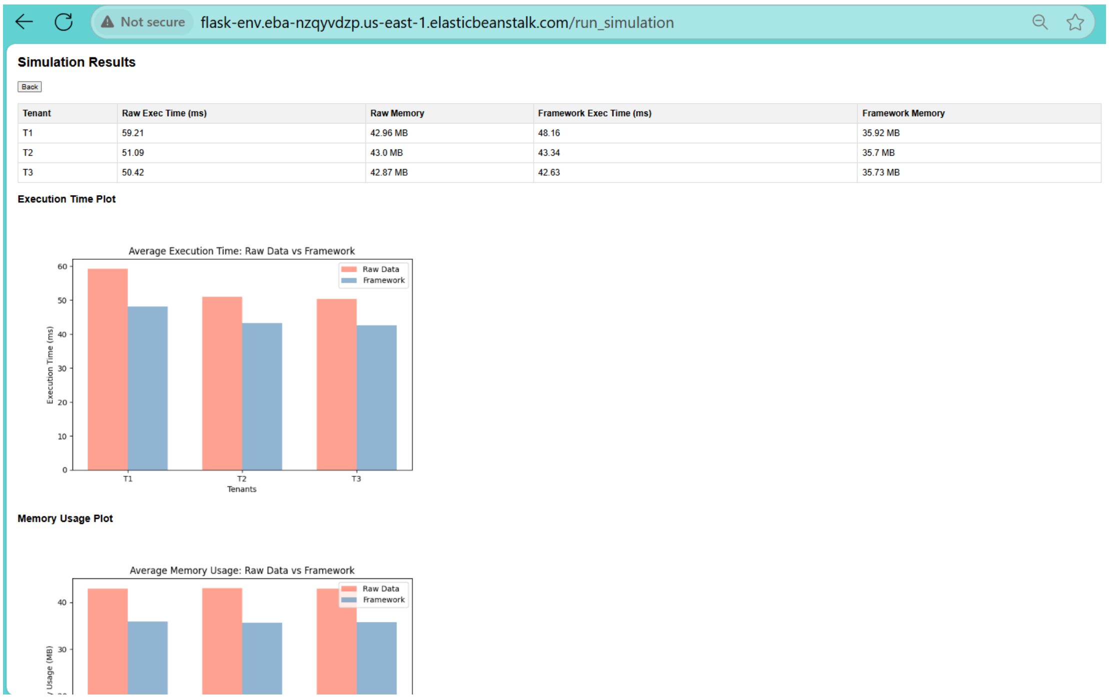


IoT Data Processing Simulation

6 Run the simulation

This step tests your project.

1. **Access the Web Interface**:
   - Open your browser and go to the domain link from eb status (e.g., iot-env-xxxxx.us-east-1.elasticbeanstalk.com).
2. **Run the Simulation**:
   - Click on "Run Simulation" on the webpage.
   - Wait for 30 seconds while the simulation generates synthetic IoT data, anonymizes sensitive data with cryptpandas, processes it, and saves results to the S3 bucket.
3. **Check the Results**:
   - After 30 seconds, you should see performance metrics (e.g., execution time around 355 ms, memory usage around 45 MB) and visualizations (e.g., execution_time.png, memory_usage.png) loaded from the S3 bucket.The page should be like below:


Not secure
flask-env.eba-nzqyvdzp.us-east-1.elasticbeanstalk.com/run_simulation

Simulation Results

Tenant	Raw Exec Time (ms)	Raw Memory	Framework Exec Time (ms)	Framework Memory
T1	59.21	42.96 MB	48.16	35.92 MB
T2	51.09	43.0 MB	43.34	35.7 MB
T3	50.42	42.87 MB	42.63	35.73 MB

Execution Time Plot

Average Execution Time: Raw Data vs Framework
Raw Data
Framework
Execution Time (ms)
0
10
20
30
40
50
60
T1
T2
T3
Tenants

Memory Usage Plot

Average Memory Usage: Raw Data vs Framework
Raw Data
Framework
Usage (MB)
30
40

   - If results don't appear, refresh the page or check the terminal for errors.

7 User Interface

- **Overview**: The web interface allows you to start the simulation and view results.
- **Access**: Use the Elastic Beanstalk URL provided by eb status.
- **Features**: "Run Simulation" triggers the 30-second process, and displays the output.

References

1. AWS Documentation. (2025). *AWS S3 User Guide*. Retrieved from https://docs.aws.amazon.com/s3/
2. AWS Elastic Beanstalk Documentation. (2025). *EB CLI User Guide*. Retrieved from https://docs.aws.amazon.com/elasticbeanstalk/
3. GitHub Documentation. (2025). *GitHub Basics*. Retrieved from https://docs.github.com/