National College of Ireland

An Intelligent Serverless AI pipeline for infrastructure creation based on Agricultural sensory IoT data

MSc Research Project
Cloud Computing

Vedant Rajendra Chavan
Student ID: x23393131

School of Computing
National College of Ireland

Supervisor: Sean Heeney

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Vedant Rajendra Chavan
Student ID:	x23393131
Programme:	Cloud Computing
Year:	2025
Module:	MSc Research Project
Supervisor:	Sean Heeney
Submission Due Date:	11/12/2025
Project Title:	An Intelligent Serverless AI pipeline for infrastructure creation based on Agricultural sensory IoT data
Word Count:	1107
Page Count:	4

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:	Vedant Rajendra Chavan
Date:	11th December 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):	

An Intelligent Serverless AI pipeline for infrastructure creation based on Agricultural sensory IoT data

Vedant Rajendra Chavan
x23393131

1 Introduction

This project is an end-to-end IoT Montoring and selfinfrastructure creating Recommendation system which takes the agricultural sensory data and suggests infrastructure based on the volume of data. The configuration module provided will give you a brief overview of the setup and dependencies required.

2 Enviroment Setup

2.1 System Requirements

- **Haredware Required**–Stable internet connection, 8Gb Ram
- **Opreating System**–Linux,macOS,Windows
- **Processor**–x86 Quad-Core processors
- **Code editor**–VS Code (any python compiling editor can work)
- **Python Version**–Python3.8 or more

2.2 Local Files and Cli Dependencies

- Run aws configure and provide (AWS ACCESS_KEY,AWS_SECRETKEY,Region, and output format)
- **Virtual Enviroment(venv)**– install venv with the command **python -m venv myenv** and run the venv using *Venv Documentation* (2025)**venv/scripts/activate**
- **Requirements**– install all the requirements that are present by running the command in file **pip install requirements.txt** in cli
- **ProjectSetup**–I have created a **complete_setup.sh** file that will help you to create the core infrastructure of theproject the tables in dynamodb, lambda_triggers, sns_topic, kinesis stream and IAM policies which you can use as base structure. Note- you can change the policies and services as per your requirements.
- **Cleanup Script**–Run the file **cleanup_all.sh** to clean the existing infrastructure.

2.3 Core Aws Services

- **Lambda:** Lambda is used to trigger the data from kinesis stream and ml model for creating infrastructure.*Lambda Documentation* (2025)
- **DynamoDb:** for storing the sensory data in the the database for processing *Dynamodb Document* (2025)
- **kinesis:** used for streaming data in real-time from the sensors. *Kinesis DataStream* (2024)
- **sns(Simple Notification Service):**sending alerts to the users about the payloads.

3 Project Flow

This is IoT driven agricutural smart Ai pipline project for this the major flow of the project is sensor ⇒ devices ⇒ kinesis ⇒ lambda triggers ⇒ ml model ⇒ dashboard for the prediction of infrastructure and the crop recommendation in this section i will describe the overall flow in the project

3.1 Sensory Data:

For this project I have created eight sensors that is sending data in Mqtt format from a sensory script **sensors.py** to devices that are implimented in the fields which is sending data to the Kinesis stream which will trigger a lambda function for the ml model for prediction of the infrastructure, below I am attaching a CLI sensor flow from the sensors.

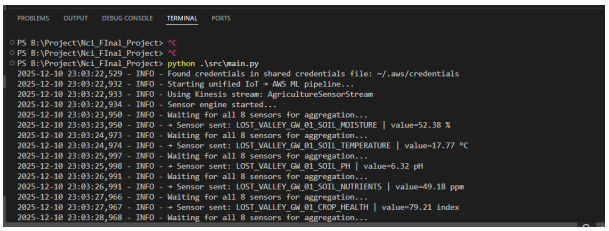

Figure 1: Sensory Data

3.2 kinesis stream:

The kinesis is a data streaming Aws service that ingests the data in real time *Kinesis DataStream* (2024) in my project kinesis plays the major role for streaming data in real time you can check if the data is being ingested through kinesis by two methods, firstly by checking the dashboard of Aws or running the CLI command **aws kinesis describe-stream –stream-name AgricultureSensorStream** will help you to verify your data in terminal so that you will not see mismatch or error in system.

3.3 lambdatrigger:

The lambda trigger is the backbone of our system so it is necessary that lambda is coded accurately, any mismatch in the system db names with variables can lead entire system to fail.

3.4 Mlmodel:

The data after being ingested is provided to the model where the model impliments various testing and training over data and makes decision based on the data volume and sets scaling condition for infrastructure, **scale_up**, **scale_down** and **maintain** are the three major conditon in my infrastructure.

3.5 SNS Topic:

The use of the SNS is to notify users with the alerts. so I have create a topic in SNS and connected with the alerts by setting the condition in any scale_up, or scale_down notify user. You can create various topices such as actuators, sensor_info, registration etc.*SNS Document* (2025)

4 Project Structure

This project has a very simple but understanble structure if we ignore the packages from venv, node modules,and sensor dumps as json the overall structure goes as:

- **src folder**– This in the main folder of the directory which contains three major files **sensors.py**, **device_sim.py**, **main.py** so the sensors data file sends data to devices where in project you can see i have implimented 3 major devices which are ingesting 8 sensor data each so basically 8*3 in total 24 sensors are implimented which are sending indvidual data

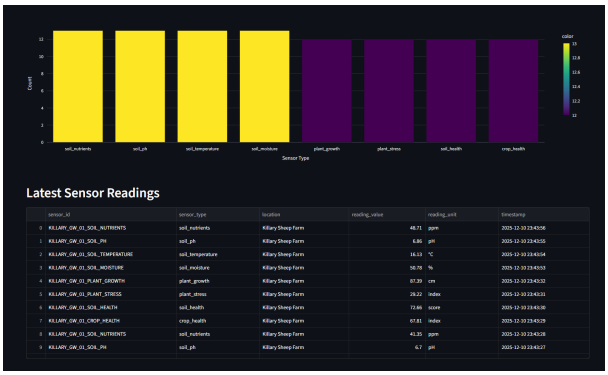

Figure 2: Dashboard overview

- **lambda:** This folder is the backbone of the project with two major files **deploy_lambda.py** which makes the overall triger for thelambda between kinesis and ml model and **ml_processor.py** this is the major implimentation which creates infrastructure based on the volume of data been ingested.

- **certs:** this folder contains the sensory certification that is required for registering an iot device to IoT Core, for that create a Thing in IoT core download the required certs provided and impliment to the project to send mqtt and http data.

- **dashboard:**– this file contains two major files in the prjoject the **app.py** which contains all the major functionalities displayed in the dashboard and **register_device.py**

this contains the device registration data which is required for sensors to predict the analysis.

- **Note:** With these core folders and file you will also find some files with **check_** and **test_** these are created to compare the data between the actual flow and display value, also to check the overall pipline working.

5 Using Dashboard

The dashboard used for this project is created in Streamlit using the Python libraries it contains various functions in class **AgricultureDashboard** that handle the end-to-end connection of the project. After running the dashboard you will see the dashboard is generally divided into three section

- **Sensor Overview:** In this section user gets the live sensory report of the field with the Distribution graph and the latest sensor reading user can get live condition in their field, following by the Tempreature sensor trend with the two major sensor comparisons between Temperature and Soil Moisture

- **Crop Recommendation:** In this part of the project the user gets their recommendation of the crops based on the overall field condition that is generated from the sensor data it generates a base condition confidence score that will be in range between the crops which gets displayed on the dashboard screen

- **Aws pipline:**This is the main connectivity page of the services, on this page you will be able to see Active services, live logs from the Cloudwatch and service recommendations based on infrastructure conditions,data provisioned and scaling factor.

References

Dynamodb Document (2025). `https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html`.

Kinesis DataStream (2024). `https://docs.aws.amazon.com/streams/latest/dev/introduction.html`.

Lambda Documentation (2025). `https://docs.aws.amazon.com/lambda/latest/dg/welcome.html`.

SNS Document (2025). `https://docs.aws.amazon.com/sns/latest/dg/welcome.html`.

Venv Documentation (2025). `https://docs.python.org/3/library/venv.html`.