National College of Ireland

# Configuration Manual

MSc Research Project
MSc in Cloud Computing


KALAL SRAVAN KUMAR GOUD
StudentID:X23342943

School of Computing
National College of Ireland


Supervisor: Shivani Jaswal

**National College of Ireland**
**Project Submission Sheet**
**School of Computing**

National College of Ireland

| | |
|---|---|
| **Student Name:** | KALAL SRAVAN KUMAR GOUD |
| **Student ID:** | X23342943 |
| **Programme:** | MSc in Cloud Computing |
| **Year:** | 2025 |
| **Module:** | MSc Research Project |
| **Supervisor:** | Shivani Jaswal |
| **Submission Due Date:** | 11/12/2025 |
| **Project Title:** | "Design and Evaluation of a Cloud-Based secure Medical Image Storage and sharing Framework using cloud security principles-Configuration Manual" |
| **Word Count:** | 1525 |
| **Page Count:** | 7 |

"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:** | KALAL SRAVAN KUMAR GOUD |
| **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): | |

**Design and Evaluation of a Cloud-Based secure Medical Image Storage and sharing Framework using cloud security principles**

KALALSRAVANKUMARGOUD
X23342943

## 1 Introduction

This is a comprehensive user guide to aid developers, system administrators or researchers in recreating the environment, performing the tests, and validating the outcomes. The Cloud-Based Secure Medical Image Storage and Sharing Framework has been developed using Python and employs many of the pivotal AWS components. Accurate configuration of the framework is essential for successful operation of the security features and the data management flows.

## 2 Prerequisites

Software and accounts needed for the configuration are:

- An installed Version of Python (Version 3.8+)
- PIP (Python Package Installer) should normally come with a Python installation.
- An AWS account which has admin privileges to create IAM users, create S3 buckets, create KMS keys, create DynamoDB tables, and create cloud watch log groups.
- AWS CLI RECOMMENDED The AWS CLI configured for basic credential management although the scripts do not utilize it directly. It is helpful for checking that resources exist or not.
- Complete source code for the project (including all modules, .py files and scripts).

## 3 AWS IAM User and Credential Setup

For security best practices, the framework should not be run using root account credentials. An IAM user with programmatic access must be created.

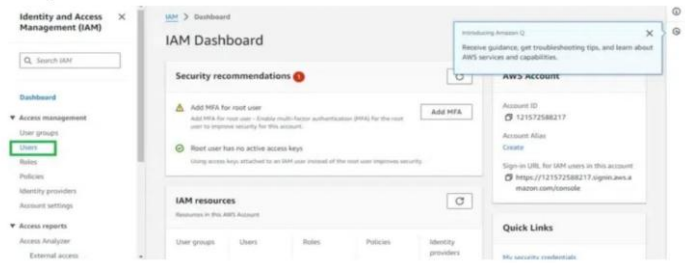


Figure 1: IAM DASHBOARD

## 4 Create an IAM User

1. Sign in to the AWS Management Console and go to the IAM service..
2. Go to Users and click” Create user”.
3. Set the User name to Admin User.
4. Choose "Access key - Programmatic access" for the kind of AWS credential. This will create the keys that the program needs.
5. Proceed to the Permissions section.
6. Choose "Attach existing policies directly."
7. Search for and select the Administrator Access policy.
8. Continue through the tags section (optional) to the review page.
9. Click” Create user”.

### 4.1 Securely Store Credentials

1. After you create the user, you will see the Access key ID and the Secret access key.
2. The key will only be visible this one time.
3. Save both the key ID and the Secret key in a safe place. In the task's layout stage, they will be utilized.

### 4.2 Local Project Setup

This section covers the configuration of the project on your local machine.

### 4.3 Set-Up a Virtual Environment

Using a virtual environment to separate the dependent projects is highly encouraged.
1. Open a text editor or shell prompt and go to the main directory of the project (for example, medical imaging structure).
2. Set up a virtual environment: python -m venv
3. Turn on the simulated environment: On Windows, type .venv\Scripts\activate. On macOS/Linux, type source venv/bin/activate.
4. Your terminal prompt should now be prefixed with (venv).

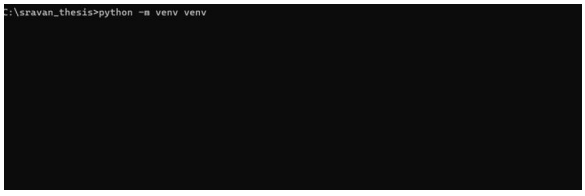


Figure 2: Set Up a Virtual Environment

Made a Salesforce Apex class and trigger that sends the CRM team a follow-on email and action when High Risk c turns TRUE.

### 4.4 Install Python Dependencies

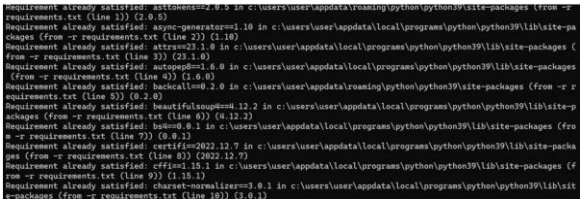


Figure 3: Set Up a Virtual Environment

There are a few Python libraries that the project needs. Make a file called requirements.txt in the main folder and put these contents in it:
boto3
python-dotenv
pycryptodome
psutil
pandas
matplotlib
seaborn
Insert these dependencies using pip:
pip install -r requirements.txt

### 4.5 Configure Environment Variables

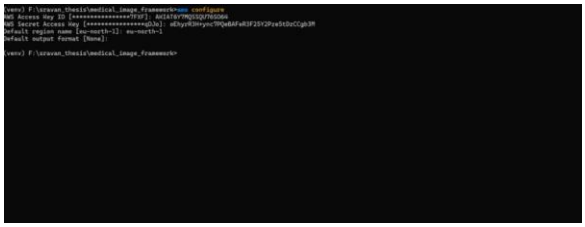


Figure 4: CONFIGURE ENVIRONMENT VARIABLES

You may change the framework's settings by editing the .env file. 1. Make a new file in the endeavor's root directory called .env. 2. Open the .env file and enter the lines below, substituting the default values with the IAM credentials you saved in Step 3.2:
“AWS_ACCESS_KEY_ID=YOUR_ACCESS_KEY_ID_HERE”
“AWS_SECRET_ACCESS_KEY=YOUR_SECRET_ACCESS_KEY_HERE”

AWS_REGION=eu-north-1
S3_BUCKET_NAME=medical-images-secure-bucket
DYNAMODB_TABLE_NAME=MedicalImageMetadata
KMS_KEY_ALIAS=alias/medical-images-cmk
CLOUDWATCH_LOG_GROUP=/aws/medical-images/audit

**Important:** Ensure the AWS_REGION matches the region where you want to deploy the resources (eu-north-1 as per your project). The other variables can be left as default, as they match the values used in your tests.

## 5 Framework Deployment and Execution

With the project configured, you can now deploy and run the framework.

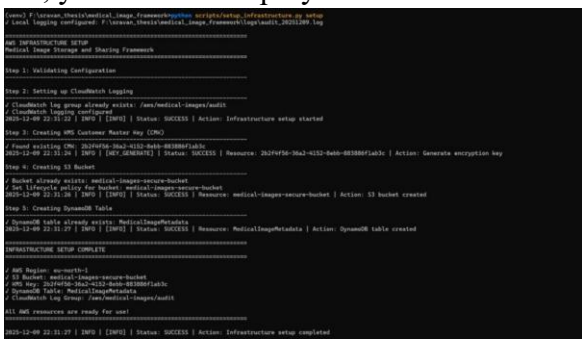


Figure 5: Deploy the AWS Infrastructure

The setup _infrastructure.py script creates all required resources within AWS automatically.

1. Make sure that you have an active virtual environment running,

2. Run this command from the project's root location:

python scripts/setup_infrastructure.py setup

3. The script checks that the configuration details are correct, and either provisions (or confirms the existence of) the following AWS resources: a KMS Key, an S3 Bucket, a DynamoDB Table, and a CloudWatch Log Group. The output of the process will indicate when the Infrastructure Setup has been completed successfully.

### 5.1 Run the Functional Demonstration

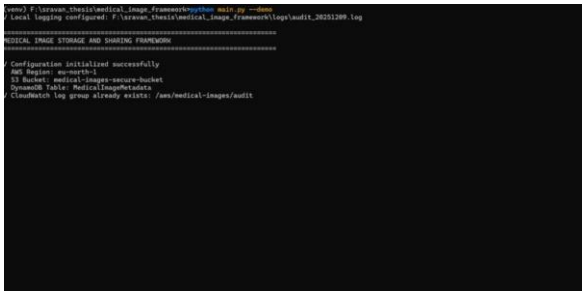


Figure 6: Functional Demonstration

The main.py script includes a demo mode to perform an end-to-end functional test.

1. Insert the following command:
python main.py –demo

2. It will execute the entire workflow: uploading, encrypting, downloading, decrypting, and sharing a sample file. The output will show the success of each step.

## 5.2 Testing with Real Medical Images

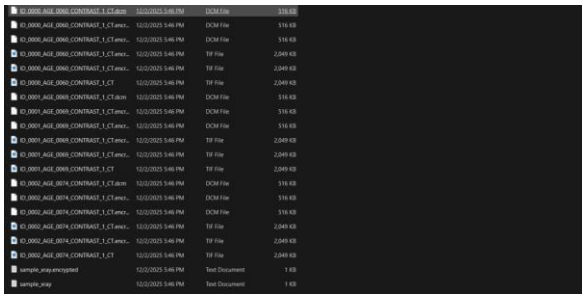

Figure 7: Testing with Real Medical Images

The testing of Real Medical Images requires Eligible Medical Images to Be Tested: DICOM or TIFF Files. 1. Create a directory called Test Images in the main application directory.

You will need to define all test medical image files to point to the directory you created. For example, the test medical image should look something like this:

“ID_0000_AGE_0060_CONTRAST_1_CT.dcm”

2. You will run this test using the following command:

python test_medical_images.py

3. Python will scan the Test Images directory for all supported medical images, run analytical processing on each image, and return a summary for each file processed.

## 5.3 Running A Performance Analysis

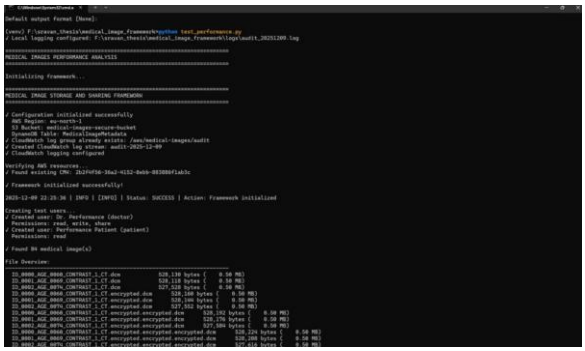

Figure 8: Performance Analysis

The performance analysis will pull data from the Test Images folder using the following steps:

1. Populate the Test _Images folder with images as described above.
2. Execute the following performance analysis script:

python test_performance.py

3. The performance analysis script will execute upload/encrypt, download/decrypt for all the images, and log times associated with each operation.

4. When the performance analysis script is complete, a new folder named Performance Reports will be created with three different formats (JSON, CSV, and TXT) for each performance report.

## 6 Infrastructure Cleanup (Important Warning)

The framework includes a script to delete all the AWS resources it created. This action is irreversible and will result in the permanent loss of all data stored in the S3 bucket and DynamoDB table.

Use this command with extreme caution.

1. To delete the AWS infrastructure, run the following command:

```
python scripts/setup_infrastructure.py cleanup
```

2. You will be prompted to type DELETE to confirm the action. If you do not type DELETE, the operation will be safely aborted.

## 7 Troubleshooting

**Error: "**botocore . exceptions .NoCredentialsError: Unable to locate credentials." **Cause:** The application cannot find your AWS credentials. **Solution:** Ensure the .env file exists in the root directory and that AWS _ACCESS _KEY ID and AWS SECRET ACCESS KEY are correctly copied and saved.

**Error: "**botocore. exceptions. ClientError: An error occurred (AccessDeniedException)" when calling the operation. **Cause:** The IAM user AdminUser does not have sufficient permissions to perform the required action. **Solution:** Verify that the Administrator Access policy (or an equivalent custom policy) is attached to the AdminUser in the AWS IAM console.

**Error: "** ModuleNotFoundError": No module named boto3 (or another module). **Cause:** The required Python packages are not installed or the virtual environment is not active. **Solution:** Make sure your virtual environment is active (the (venv) prefix should be visible) and run:

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

**Error:** Script fails with "No medical images found in test images/" **Cause:** The test medical images.py or test _performance.py script was run without any image files present. **Solution:** Create the test images folder in the project's root directory and place at least one .dcm or .tif file inside it.