Configuration Manual

MSc Research Project
MSc Cloud Computing

Maitreyee Mulay
Student ID: 23298227

School of Computing
National College of Ireland

Supervisor: Yasantha Samarawickrama

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Maitreyee Mulay
Student ID:	23298227
Programme:	MSc Cloud Computing
Year:	2025
Module:	MSc Research Project
Supervisor:	Yasantha Samarawickrama
Submission Due Date:	11/08/2025
Project Title:	Configuration Manual
Word Count:	805
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:	Maitreyee Mulay
Date:	10th August 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

Maitreyee Mulay
23298227

1 Introduction

The paper is an elaborate configuration guide of how to set up a Self-Adaptive Zero Trust Machine Learning (ZTML) system on the Amazon Web Service (AWS) as it is step by step. The main aim of the project is to establish an security system that will implement access controls in real-time at the time when it is predicted by machine learning models with zero trust. Zero trust follows the principle of 'never trust ,always verify' Ahmadi (2024). The system will capture network traffic patterns and provides the potential threats in order to dynamically adjust the security posture. The use of the CSE-CIC-IDS2018 dataset allows us to investigate the statement. Machine learning models can be utilised in order to analyze large datasets which includes the user behavior patterns, activity logs as well as threat intelligence feeds(Shaik and Gudala; 2021). The possibility of Zero Trust Access Control Policy with regard to cloud nativity through the integration of IAM and real-time threats detection is discussed by DeCusatis and Zhang (2016). Using AWS Lambda, as a serverless platform, these models are programmed to be activated based on triggers that mimics an real-time reaction in case of security alert. This guide can act as a full documentation of the services used, settings installed and resources developed in the eu-north-1 (Stockholm) region. All the steps including setup of the environment, deployment, and simulation testing is recorded as well.

2 AWSEnvironment Setup

The initial phase involves setting up the necessary AWS Identity and Access Management (IAM) user and permissions to manage resources.

2.1 IAM user details

An administrative IAM user was created to perform the setup.

- User Name: AdminUser
- Account ID: 405045611860

Permissions for access to S3, Lambda, SageMaker and general admin actions:

- AdministratorAccess
- AmazonSageMakerFullAccess
- AmazonS3FullAccess

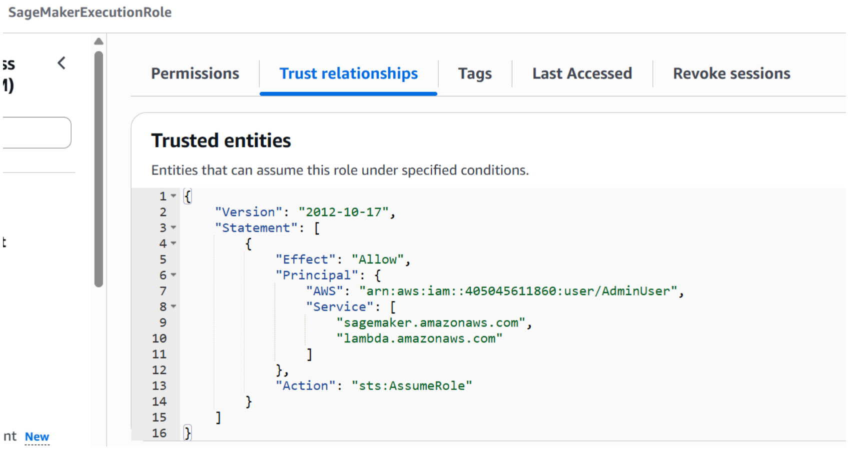
3 Data Storage Setup (S3)

An S3 bucket was created to store the dataset, model pipelines and other project related files such as train test sets.

- Upload the raw dataset : CSE-CIC-IDS2018 dataset [1]
- Create a bucket named : zero-trust-ml-dataset

4 SageMakerConfiguration

A dedicated IAM role is required for SageMaker to access other AWS services like S3.A trust policy JSON format that allows SageMaker to assume this role and provide access to lambda for simulation shown in figure 1.



Trusted entities

Entities that can assume this role under specified conditions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::405045611860:user/AdminUser",
                "Service": [
                    "sagemaker.amazonaws.com",
                    "lambda.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Figure 1: Trust relationship of SageMakerExecutionRole

4.1 Create SageMaker Notebook Instance

- instance name: ZTMLNotebook
- instance type: ml.t3.medium
- Platform identifier: Amazon Linux 2, Jupyter Lab 4 (notebook-al2-v3)
- Volume Size: 5GB EBS

5 Data Pre-processing and Model Training

The following Jupyter notebooks were executed in the SageMaker instance to process the data and train the ML models. The notebooks are:

- `dataset-preprocessing.ipynb`
- `dataset-segregation.ipynb`
- `external-threat-model.ipynb`
- `internal-threat-model.ipynb`

[1] https://www.unb.ca/cic/datasets/ids-2018.html

5.1 Data Ingestion and Preprocessing

The dataset is saved in S3 bucket as : dataset.csv

5.2 Data Segregation

The main dataset was segregated into two distinct files based on threat type:

- Internal Threats: Contains 'Bot' and'Infiltration' labels.
- External Threats: Contains 'DDoS','DoS', 'Bruteforce', 'Benign' etc labels.

5.3 Model Training

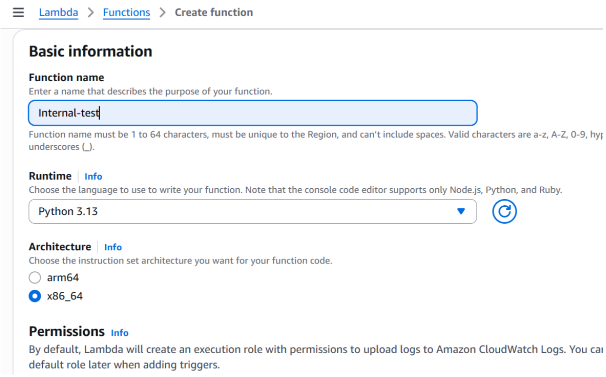
Separate models were trained for internal and external threats:

- External Threat Model: A Random Forest Classifier and an SVM were trained. The Random Forest model achieved an accuracy of 91.22% and SVM 89.9%. The pipelines were saved in S3 bucket.
- Internal Threat Model: A Logistic Regression model was trained, achieving an accuracy of 99.31%. The pipeline was saved in S3 bucket as well.

6 Real-Time Deployment and simulation with AWS Lambda

6.1 Lambda Function

Lambda Functions are created to trigger. Provide name of the function select python as runtime keep rest of the details as default as shown in figure 2 .



Basic information

Function name
Enter a name that describes the purpose of your function.
Internal-test

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyp
underscores (_).

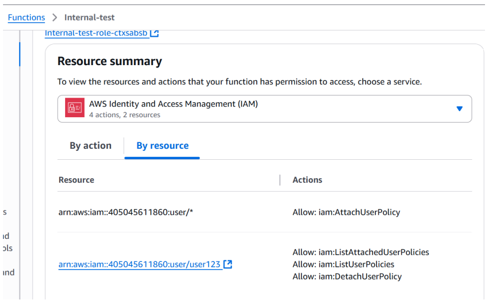
Runtime | Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.13

Architecture | Info
Choose the instruction set architecture you want for your function code.
○ arm64
● x86_64

Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You ca
default role later when adding triggers.

Figure 2: Lambda Function

Post creation of lambda function attach the IAM access permissions to the user as shown in figure 3 and ensure the monitoring logs are generating in cloudwatch under Log groups.



Functions > Internal-test

Internal-test-role-ctxsabsb

Resource summary

To view the resources and actions that your function has permission to access, choose a service.

AWS Identity and Access Management (IAM)
4 actions, 2 resources

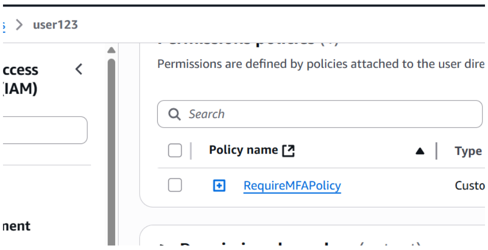
Resource	Actions
arn:aws:iam::405045611860:user/*	Allow: iam:AttachUserPolicy
arn:aws:iam::405045611860:user/user123	Allow: iam:ListAttachedUserPolicies Allow: iam:ListUserPolicies Allow: iam:DetachUserPolicy

Figure 3: User permissions

6.2 Incline Policies and user creation

For simulation purpose create 3 incline policies: One that provides basic access to the user, another that asks for MFA and last to revoke the access of the user.

Create one user and apply any one policy to the user.



> user123

Permissions are defined by policies attached to the user dire

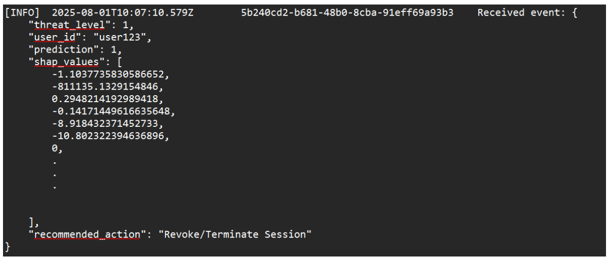
□	Policy name	Type
□	RequireMFAPolicy	Custo

Figure 4: Initial policy

6.3 Simulation

After running the simulation script check the cloudwatch logs to verify the if the lambda was triggered, inputs received and if the policy is applied.

The event received in Json format, as shown below

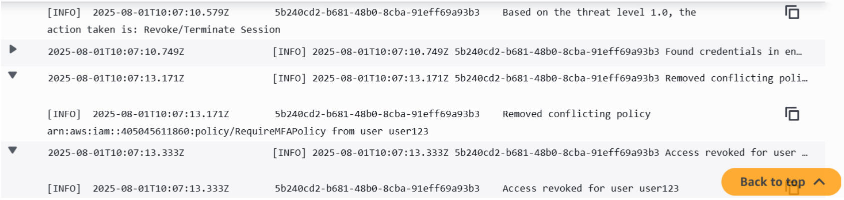


```
[INFO]  2025-08-01T10:07:10.579Z        5b240cd2-b681-48b0-8cba-91eff69a93b3    Received event: {
    "threat_level": 1,
    "user_id": "user123",
    "prediction": 1,
    "shap_values": [
        -1.1037735830586652,
        -811135.1329154846,
        0.2948214192989418,
        -0.14171449616635648,
        -8.918432371452733,
        -10.802322394636896,
        0,
        .
        .
        .


    ],
    "recommended_action": "Revoke/Terminate Session"
}
```

Figure 5: Sample JSON data recivied

Cloudwatch Logs with details of actions taken and policy enforcement Figure 6



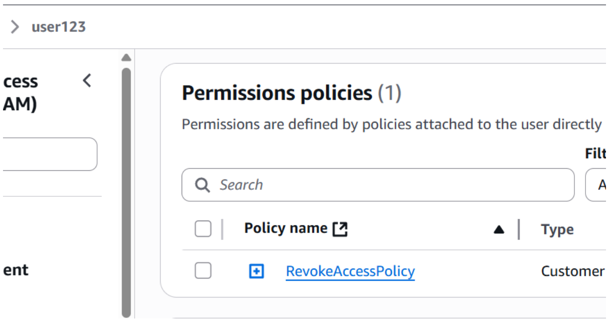
```
    [INFO]  2025-08-01T10:07:10.579Z       5b240cd2-b681-48b0-8cba-91eff69a93b3    Based on the threat level 1.0, the
    action taken is: Revoke/Terminate Session
▶   2025-08-01T10:07:10.749Z              [INFO] 2025-08-01T10:07:10.749Z 5b240cd2-b681-48b0-8cba-91eff69a93b3 Found credentials in en…
▼   2025-08-01T10:07:13.171Z              [INFO] 2025-08-01T10:07:13.171Z 5b240cd2-b681-48b0-8cba-91eff69a93b3 Removed conflicting poli…

    [INFO]  2025-08-01T10:07:13.171Z       5b240cd2-b681-48b0-8cba-91eff69a93b3    Removed conflicting policy
    arn:aws:iam::405045611860:policy/RequireMFAPolicy from user user123
▼   2025-08-01T10:07:13.333Z              [INFO] 2025-08-01T10:07:13.333Z 5b240cd2-b681-48b0-8cba-91eff69a93b3 Access revoked for user …

    [INFO]  2025-08-01T10:07:13.333Z       5b240cd2-b681-48b0-8cba-91eff69a93b3    Access revoked for user user123
```

Figure 6: Policy Enforcement Logs

Policy applied to the user can be verified through IAM. The previous policy which is conflicting with the current policy is removed.



Permissions policies (1)

Permissions are defined by policies attached to the user directly

□	Policy name	Type
□	RevokeAccessPolicy	Customer

Figure 7: Updated Policy

The ML model algorithms are in the .ipynb files created for this project are available in the Git repository:
GitHub Repository: `https://github.com/MaitreyeeM-student/Research`$_{project}$

References

Ahmadi, S. (2024). Zero Trust Architecture in Cloud Networks: Application, Challenges and Future Opportunities. Cited by 44, HAL Open Science, [Accessed 16 June. 2025].

DeCusatis, C. and Zhang, L. (2016). Implementing Zero Trust Cloud Networks with Transport Access Control and First Packet Authentication, *IEEE Transactions on Cloud Computing* **8**. Cited by 100, CORE A, [Accessed 13 June. 2025].

Shaik, M. and Gudala, L. (2021). Towards Autonomous Security: Leveraging Artificial Intelligence for Dynamic Policy Formulation and Continuous Compliance Enforcement in Zero Trust Security Architectures, *African Journal of Artificial Intelligence and Sustainable Development* **1**(2). Cited by 2, African Science Group, [Accessed 22 May. 2025].