

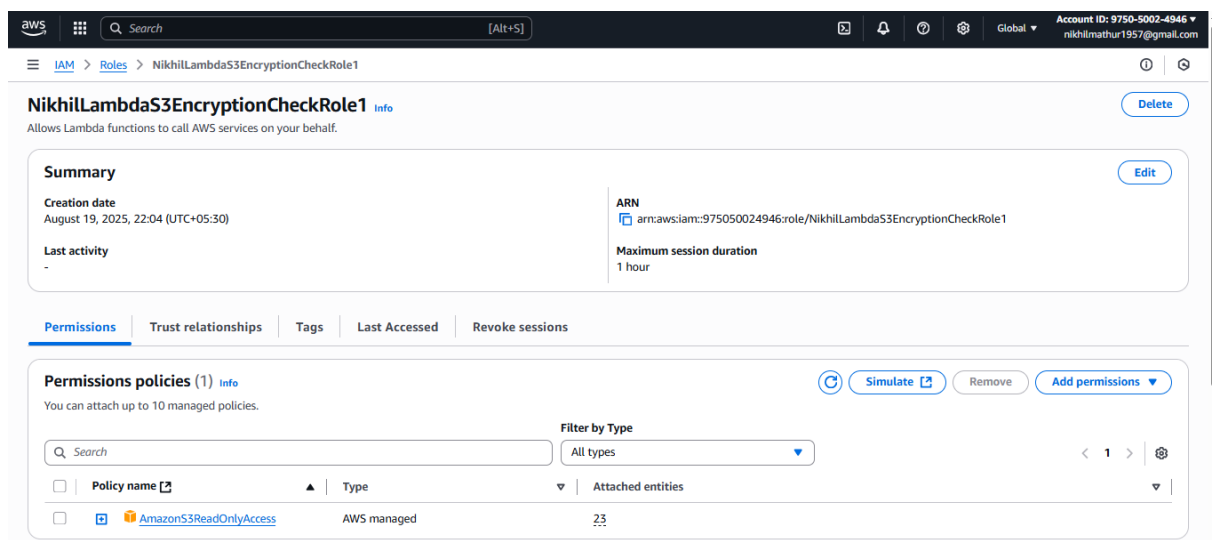
Assignment 3: Monitor Unencrypted S3 Buckets Using AWS Lambda and Boto3

Objective: To enhance your AWS security posture by setting up a Lambda function that detects any S3 bucket without server-side encryption.

Task: Automate the detection of S3 buckets that don't have server-side encryption enabled.

1. Create an IAM Role for Lambda

- AWS Console => IAM => Roles => Create role.
- Trusted Entity type: AWS Services
- Use Case: Lambda
- Click Next



Permissions policies: AmazonEC2FullAccess

Role name: NikhilLambdaEC2ControlRole1

Click Create role.

2. Create the Lambda Function

Go to AWS Console => Lambda.

Click Create function.

Select Author from scratch

Function name: **NikhilS3EncryptionCheck**

Runtime: Python 3.13

Permissions:

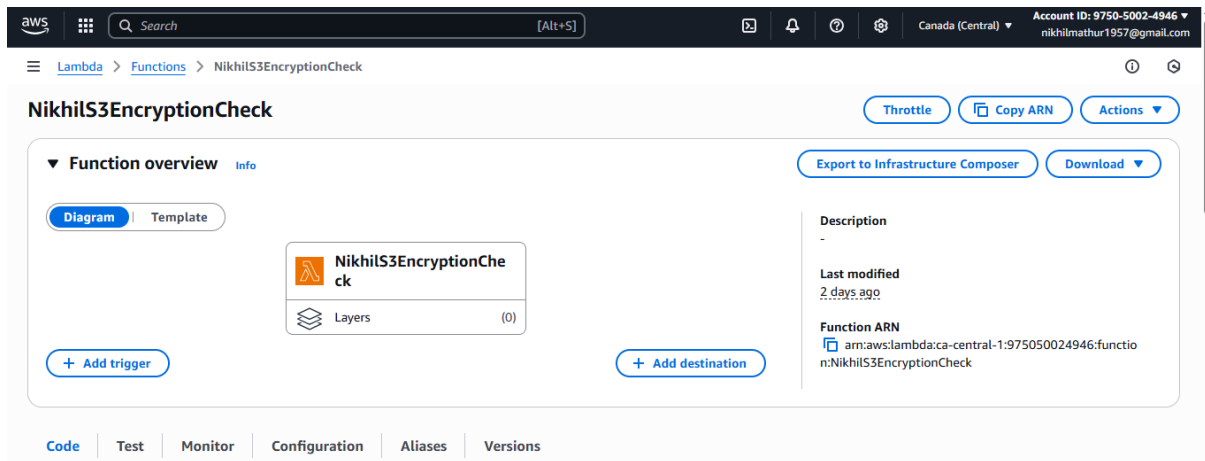
Expand Change default execution role.

Select Use an existing role.

Choose **NikhilLambdaEC2ControlRole1** from the dropdown.

Note => I choose the role **prashantb12-role-9p53470y** for permission access to run the code.

Click **Create function**.



3. Add Python Code to Control EC2

```
Assignment3.py X
Assignment-3 > Assignment3.py
1 import boto3
2 from botocore.exceptions import ClientError
3
4 def lambda_handler(event, context):
5     s3 = boto3.client("s3")
6
7     # Get all S3 buckets
8     response = s3.list_buckets()
9     buckets = response["Buckets"]
10
11     non_encrypted_buckets = []
12
13     for bucket in buckets:
14         bucket_name = bucket["Name"]
15         try:
16             # Check encryption settings
17             enc = s3.get_bucket_encryption(Bucket=bucket_name)
18             rules = enc["ServerSideEncryptionConfiguration"]["Rules"]
19
20             # If rules exist, SSE is enabled
21             print(f" {bucket_name} has encryption: {rules}")
22
23         except ClientError as e:
24             error_code = e.response["Error"]["Code"]
25             if error_code == "ServerSideEncryptionConfigurationNotFoundError":
26                 print(f" {bucket_name} does NOT have encryption enabled")
27                 non_encrypted_buckets.append(bucket_name)
28             else:
29                 print(f" Could not check bucket {bucket_name}: {e}")
30
```

```

30
31     if non_encrypted_buckets:
32         print("Buckets without encryption:", non_encrypted_buckets)
33     else:
34         print("All buckets have encryption enabled ✅")
35
36     return {"NonEncryptedBuckets": non_encrypted_buckets}
37

```

Create the test case

The screenshot shows the AWS Lambda console interface for a function named 'NikhilEC2TagBasedControl'. The 'Test' tab is selected, showing a green status bar indicating 'Executing function: succeeded'. Below this, there are buttons for 'Delete', 'CloudWatch Logs Live Tail', 'Save', and 'Test'. The 'Test event action' section has two options: 'Create new event' and 'Edit saved event', with 'Edit saved event' being selected. The 'Event name' is set to 'NikhilMathurTestEvent'. The 'Event JSON' section shows a JSON object: `{}`. There is a 'Format JSON' button next to it.

Click for deploy the code

Test the code

The screenshot shows the AWS Lambda console interface for a function named 'NikhilS3EncryptionCheck'. The 'Deploy' tab is selected, showing a 'Deploy (Ctrl+Shift+U)' button and a 'Test (Ctrl+Shift+I)' button. Below these buttons, there is a section for 'TEST EVENTS' with a list of events: 'Create new test event', 'Private saved events', and 'NikhilMathurTestEvent'. The 'NikhilMathurTestEvent' is selected. The 'Execution Results' section shows the status 'Succeeded' and the test event name 'NikhilMathurTestEvent'. The 'Response' is a JSON object: `{ "NonEncryptedBuckets": [] }`. The 'Function Logs' section shows the output of the function, including the encryption status of various buckets: 'ketKeyEnabled': True, 'somu-static-website has encryption: [{"ApplyServerSideEncryptionByDefault": {"SSEAlgorithm": "AES256"}, "BucketKeyEnabled": True}]', 'somumh has encryption: [{"ApplyServerSideEncryptionByDefault": {"SSEAlgorithm": "AES256"}, "BucketKeyEnabled": True}]', 'studentai-bucket has encryption: [{"ApplyServerSideEncryptionByDefault": {"SSEAlgorithm": "AES256"}, "BucketKeyEnabled": True}]', and 'subhadeep-bucket has encryption: [{"ApplyServerSideEncryptionByDefault": {"SSEAlgorithm": "AES256"}, "BucketKeyEnabled": True}]'.