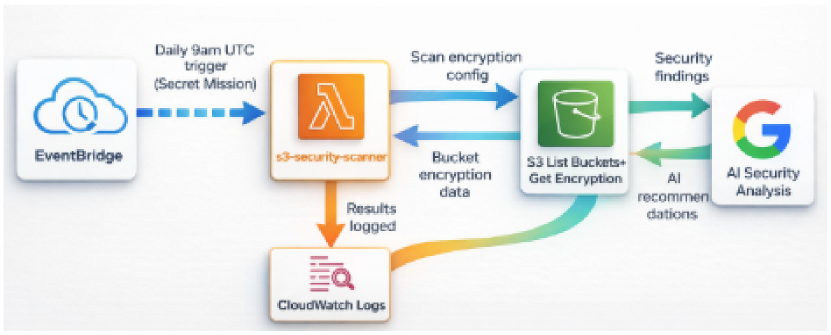
Dhruval Rana

AWS AI S3 Security Scanner

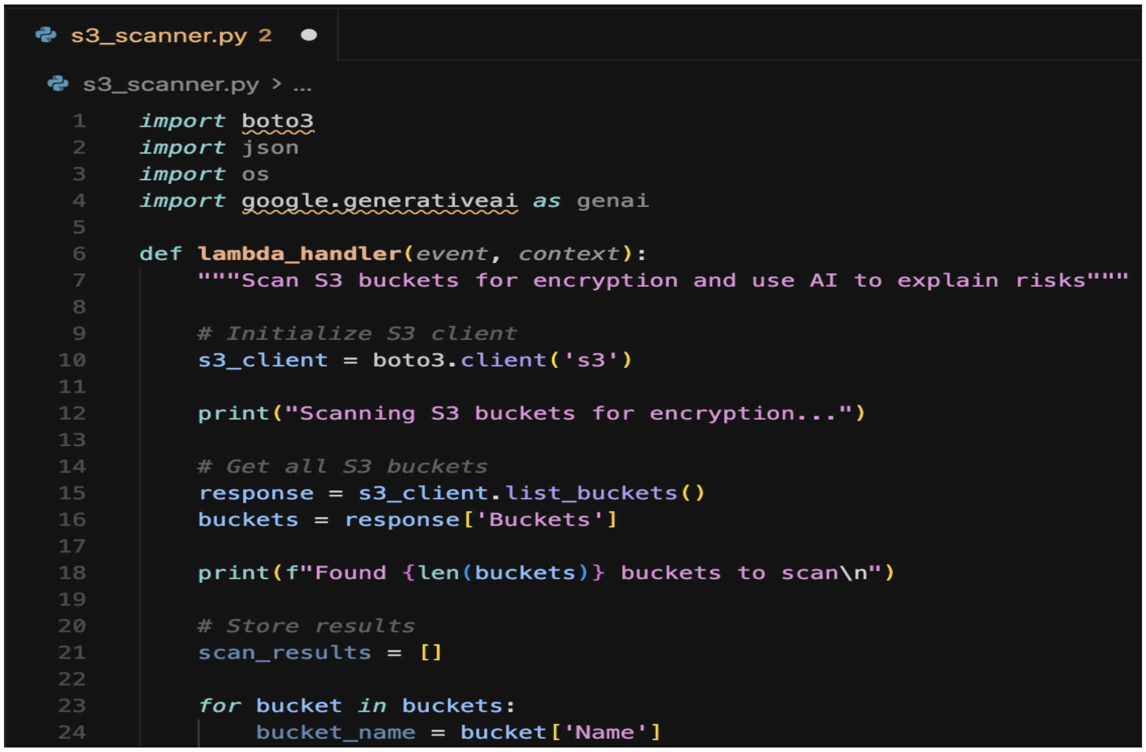
Project Overview

I built an AI-powered S3 Security Scanner. The system uses Python and the AWS SDK (boto3) to identify unencrypted buckets, leverages Gemini AI to explain risks in plain English, and uses Amazon EventBridge to automate the audit every 12 hours.



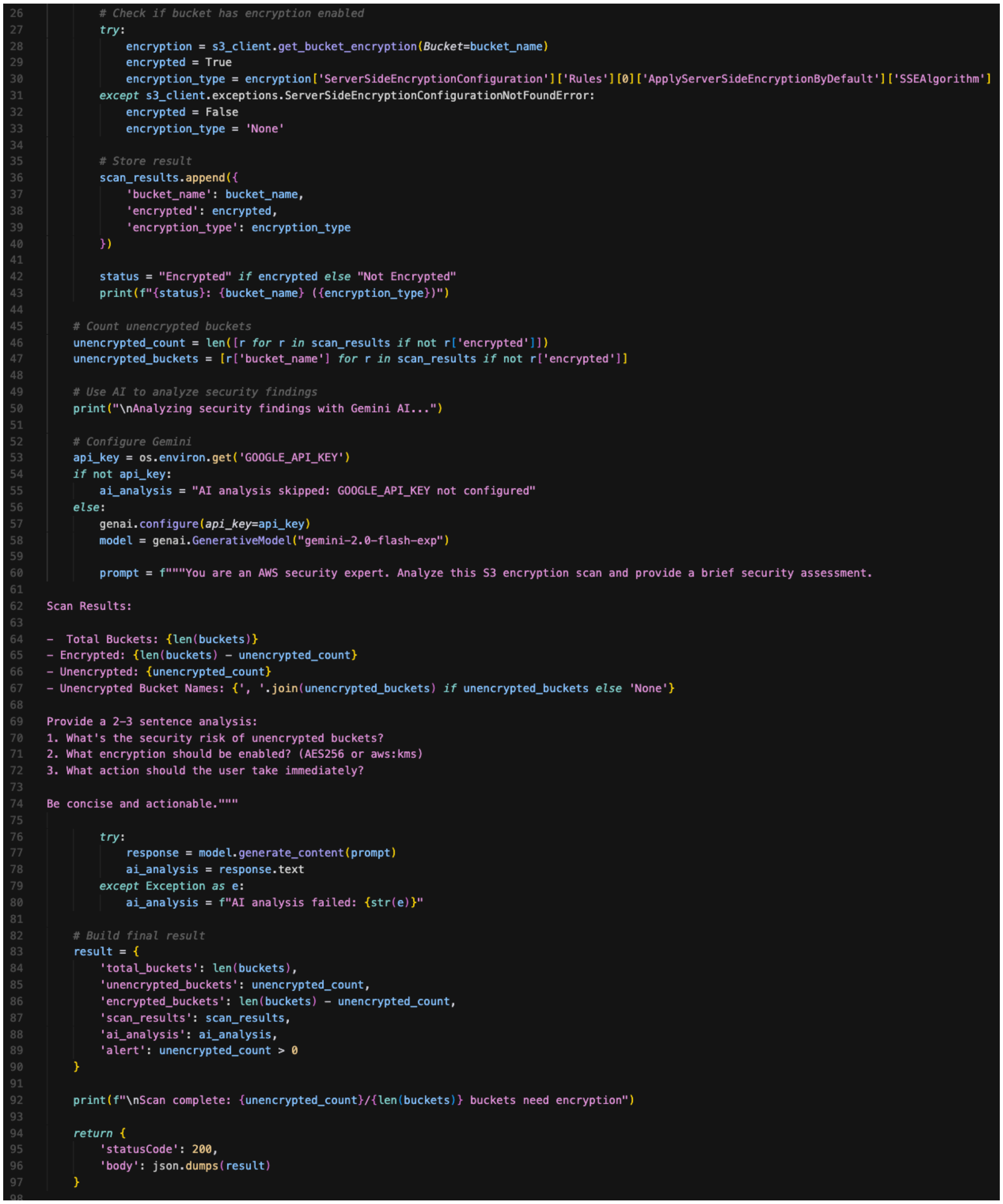
Phase 1: Brain & Logic (Setup)

- **Generated Gemini API Key**: I started by visiting Google AI Studio to generate a secure API key. This serves as the "analytical brain" that allows my scanner to interpret technical data.
- **Developed Scanner Logic**: Using the Cursor IDE, I implemented a Python script (`s3_scanner.py`) that uses `boto3` to connect to my AWS account. I programmed it to loop through all S3 buckets and check for active server-side encryption.



```python
import boto3
import json
import os
import google.generativeai as genai

def lambda_handler(event, context):
    """Scan S3 buckets for encryption and use AI to explain risks"""

    # Initialize S3 client
    s3_client = boto3.client('s3')

    print("Scanning S3 buckets for encryption...")

    # Get all S3 buckets
    response = s3_client.list_buckets()
    buckets = response['Buckets']

    print(f"Found {len(buckets)} buckets to scan\n")

    # Store results
    scan_results = []

    for bucket in buckets:
        bucket_name = bucket['Name']
```

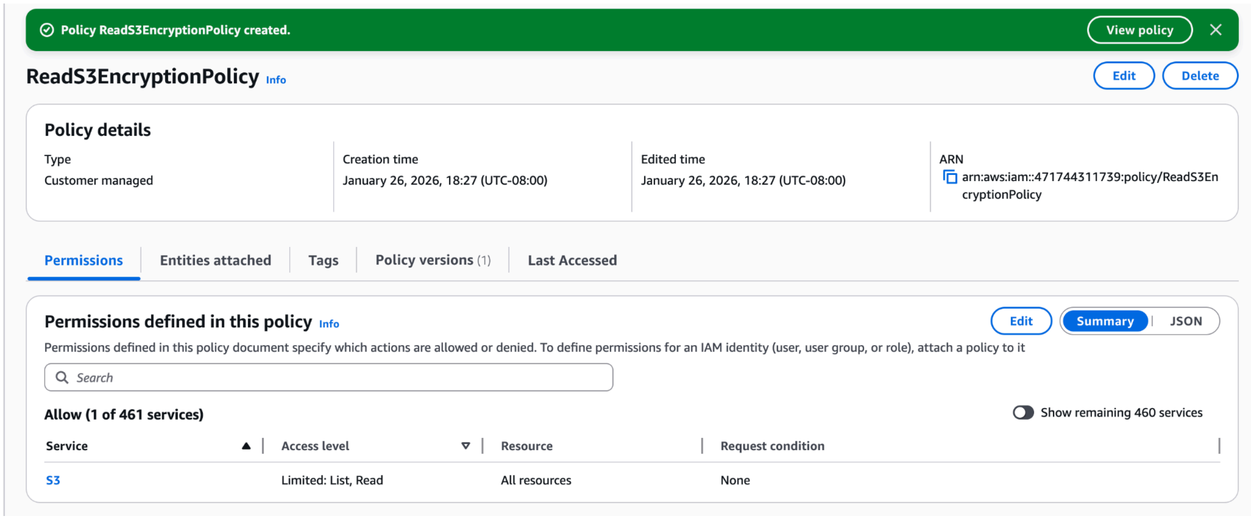
- **Integrated Gemini AI**: I drafted a specific prompt within the code that sends raw S3 configuration data to Gemini. I instructed the AI to return high-level, plain-English security insights rather than just technical error codes.



```python
        # Check if bucket has encryption enabled
        try:
            encryption = s3_client.get_bucket_encryption(Bucket=bucket_name)
            encrypted = True
            encryption_type = encryption['ServerSideEncryptionConfiguration']['Rules'][0]['ApplyServerSideEncryptionByDefault']['SSEAlgorithm']
        except s3_client.exceptions.ServerSideEncryptionConfigurationNotFoundError:
            encrypted = False
            encryption_type = 'None'

        # Store result
        scan_results.append({
            'bucket_name': bucket_name,
            'encrypted': encrypted,
            'encryption_type': encryption_type
        })

        status = "Encrypted" if encrypted else "Not Encrypted"
        print(f"{status}: {bucket_name} ({encryption_type})")

    # Count unencrypted buckets
    unencrypted_count = len([r for r in scan_results if not r['encrypted']])
    unencrypted_buckets = [r['bucket_name'] for r in scan_results if not r['encrypted']]

    # Use AI to analyze security findings
    print("\nAnalyzing security findings with Gemini AI...")

    # Configure Gemini
    api_key = os.environ.get('GOOGLE_API_KEY')
    if not api_key:
        ai_analysis = "AI analysis skipped: GOOGLE_API_KEY not configured"
    else:
        genai.configure(api_key=api_key)
        model = genai.GenerativeModel("gemini-2.0-flash-exp")

        prompt = f"""You are an AWS security expert. Analyze this S3 encryption scan and provide a brief security assessment.

Scan Results:

-  Total Buckets: {len(buckets)}
- Encrypted: {len(buckets) - unencrypted_count}
- Unencrypted: {unencrypted_count}
- Unencrypted Bucket Names: {', '.join(unencrypted_buckets) if unencrypted_buckets else 'None'}

Provide a 2-3 sentence analysis:
1. What's the security risk of unencrypted buckets?
2. What encryption should be enabled? (AES256 or aws:kms)
3. What action should the user take immediately?

Be concise and actionable."""

        try:
            response = model.generate_content(prompt)
            ai_analysis = response.text
        except Exception as e:
            ai_analysis = f"AI analysis failed: {str(e)}"

    # Build final result
    result = {
        'total_buckets': len(buckets),
        'unencrypted_buckets': unencrypted_count,
        'encrypted_buckets': len(buckets) - unencrypted_count,
        'scan_results': scan_results,
        'ai_analysis': ai_analysis,
        'alert': unencrypted_count > 0
    }

    print(f"\nScan complete: {unencrypted_count}/{len(buckets)} buckets need encryption")

    return {
        'statusCode': 200,
        'body': json.dumps(result)
    }
```

Phase 2: Security & Permissions (IAM)

- **Created Read-Only Policy**: I logged into the IAM Console and authored a custom JSON policy. I granted the specific permissions `s3:ListAllMyBuckets` and `s3:GetEncryptionConfiguration` to ensure the principle of least privilege.



Policy ReadS3EncryptionPolicy created.

ReadS3EncryptionPolicy Info

Policy details

Type	Creation time	Edited time	ARN
Customer managed	January 26, 2026, 18:27 (UTC-08:00)	January 26, 2026, 18:27 (UTC-08:00)	arn:aws:iam::471744311739:policy/ReadS3EncryptionPolicy

Permissions defined in this policy Info

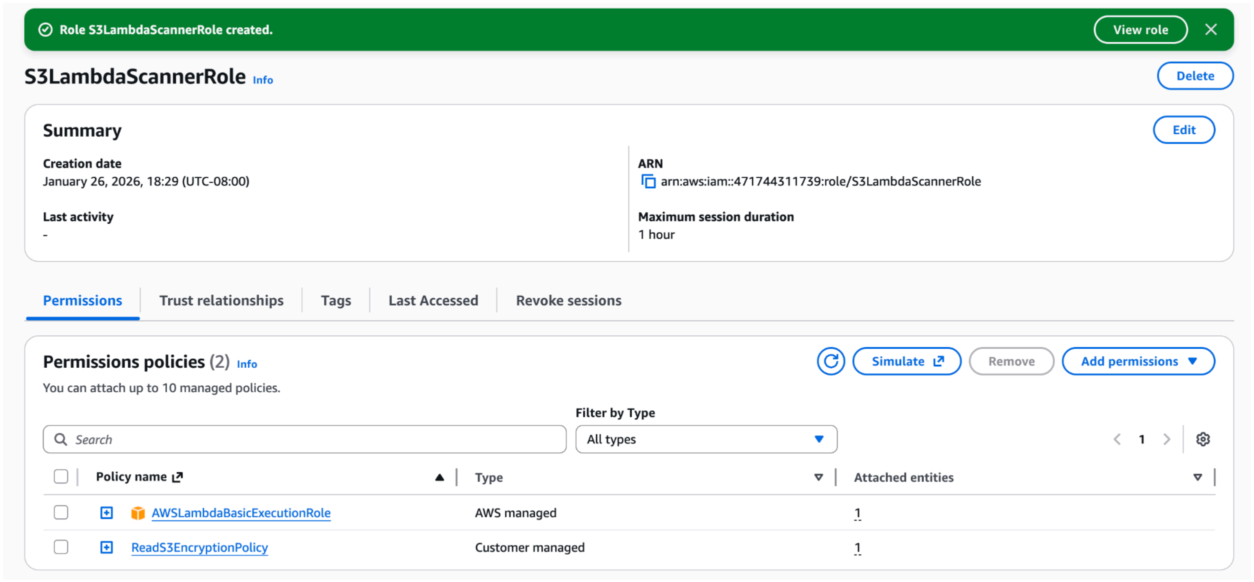
Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it

Allow (1 of 461 services)

Show remaining 460 services

Service	Access level	Resource	Request condition
S3	Limited: List, Read	All resources	None

- **Established Execution Role**: I created an IAM Role named `LambdaS3ScannerRole` for the Lambda service. I attached my custom S3 policy along with the `AWSLambdaBasicExecutionRole` so the function could write logs to CloudWatch.



Role S3LambdaScannerRole created.

S3LambdaScannerRole Info

Summary

Creation date
January 26, 2026, 18:29 (UTC-08:00)

Last activity
-

ARN
arn:aws:iam::471744311739:role/S3LambdaScannerRole

Maximum session duration
1 hour

Permissions policies (2) Info

You can attach up to 10 managed policies.

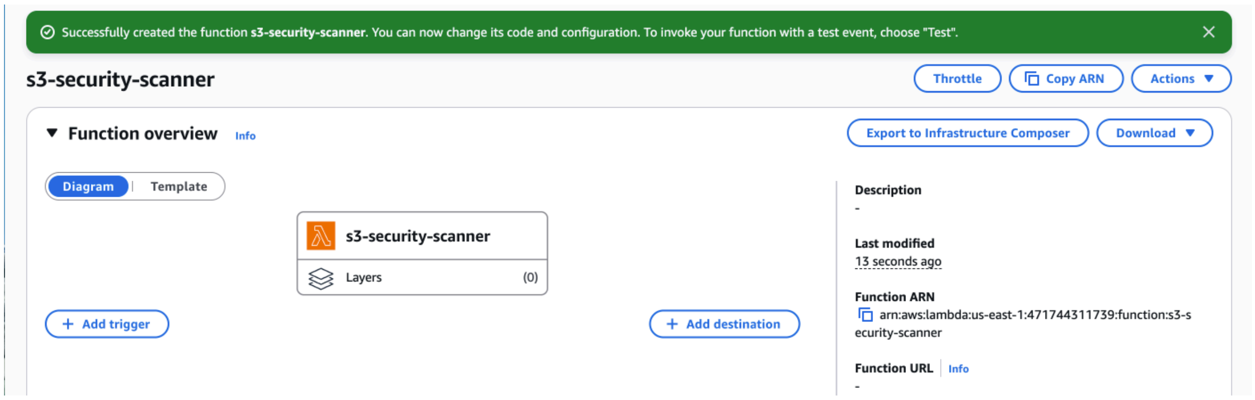
Filter by Type: All types

Policy name	Type	Attached entities
AWSLambdaBasicExecutionRole	AWS managed	1
ReadS3EncryptionPolicy	Customer managed	1

- **Configured Trust Relationship**: I verified that the trust policy correctly allows the `lambda.amazonaws.com` service to assume this role during execution.

Phase 3: Deployment (AWS Lambda)

- **Built Deployment Package**: I created a `requirements.txt` file and used `pip` to install dependencies into a local directory. To keep the package under the 50MB limit, I manually removed the pre-installed `boto3` and `botocore` libraries before zipping the files.
- **Initialized Lambda Function**: I created a new function in the AWS console using the **Python 3.12** runtime. I assigned the `LambdaS3ScannerRole` I built earlier to provide the necessary "identity" for the scan.


Successfully created the function **s3-security-scanner**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

s3-security-scanner

Function overview

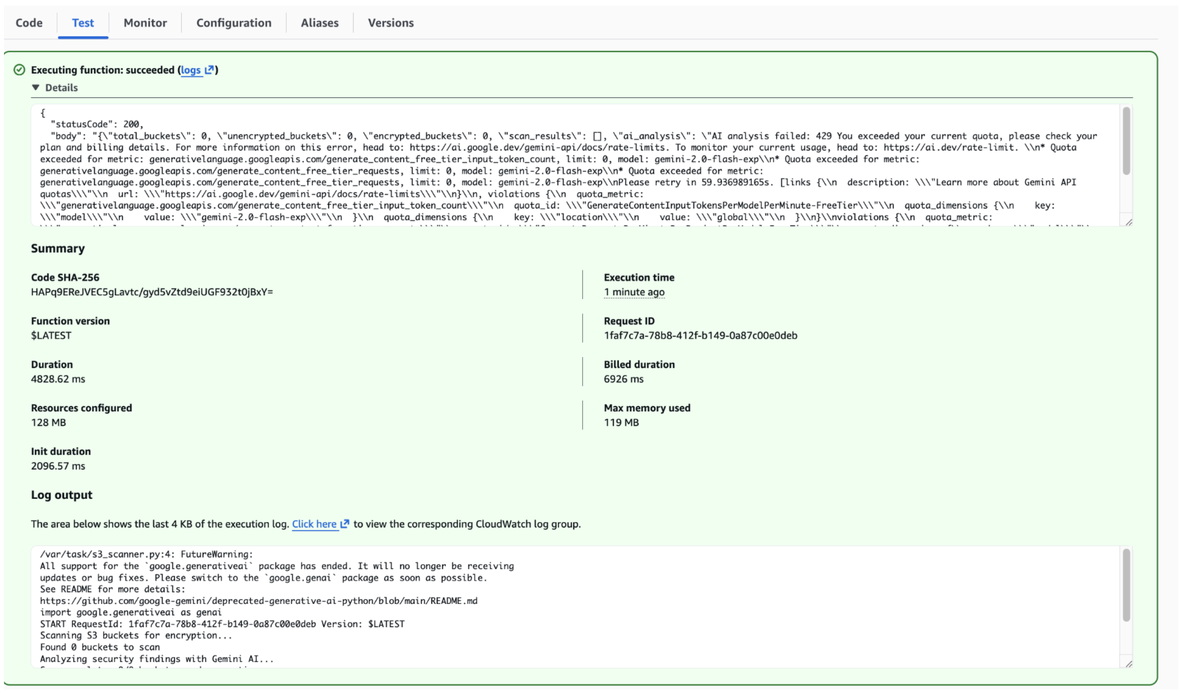
Description
-

Last modified
13 seconds ago

Function ARN
arn:aws:lambda:us-east-1:471744311739:function:s3-security-scanner

Function URL
-

- **Configured Runtime & Secrets**: I updated the **Handler** setting to `s3_scanner.lambda_handler` to match my file name. I then added my Gemini API key as an **Environment Variable** (`GOOGLE_API_KEY`) to keep the credential out of the source code.
- **Adjusted Resources**: I increased the function **timeout to 30 seconds** to ensure the scan wouldn't cut off if I added more buckets in the future. As a result the scanned logs can be seen below in the screenshot.



Executing function: succeeded (logs)

Details

```
{
  "statusCode": 200,
  "body": "{\"total_buckets\": 0, \"unencrypted_buckets\": 0, \"encrypted_buckets\": 0, \"scan_results\": [], \"ai_analysis\": \"AI analysis failed: 429 You exceeded your current quota, please check your
plan and billing details. For more information on this error, head to: https://ai.google.dev/gemini-api/docs/rate-limits. To monitor your current usage, head to: https://ai.dev/rate-limit. \\n* Quota
exceeded for metric: generativelanguage.googleapis.com/generate_content_free_tier_input_token_count, limit: 0, model: gemini-2.0-flash-exp\\n* Quota exceeded for metric:
generativelanguage.googleapis.com/generate_content_free_tier_requests, limit: 0, model: gemini-2.0-flash-exp\\n* Quota exceeded for metric:
generativelanguage.googleapis.com/generate_content_free_tier_requests, limit: 0, model: gemini-2.0-flash-exp\\nPlease retry in 59.936989165s. [links {\\n  description: \\\"Learn more about Gemini API
quotas\\\"\\n  url: \\\"https://ai.google.dev/gemini-api/docs/rate-limits\\\"\\n}\\n, violations {\\n  quota_metric:
\\\"generativelanguage.googleapis.com/generate_content_free_tier_input_token_count\\\"\\n  quota_id: \\\"GenerateContentInputTokensPerModelPerMinute-FreeTier\\\"\\n  quota_dimensions {\\n    key:
\\\"model\\\"\\n    value: \\\"gemini-2.0-flash-exp\\\"\\n  }\\n  quota_dimensions {\\n    key: \\\"location\\\"\\n    value: \\\"global\\\"\\n  }\\n}\\nviolations {\\n  quota_metric:
```

Summary

Code SHA-256 HAPq9EReJVEC5gLavtc/gyd5vZtd9eiUGF932t0jBxY=	**Execution time** 1 minute ago
Function version $LATEST	**Request ID** 1faf7c7a-78b8-412f-b149-0a87c00e0deb
Duration 4828.62 ms	**Billed duration** 6926 ms
Resources configured 128 MB	**Max memory used** 119 MB
Init duration 2096.57 ms	

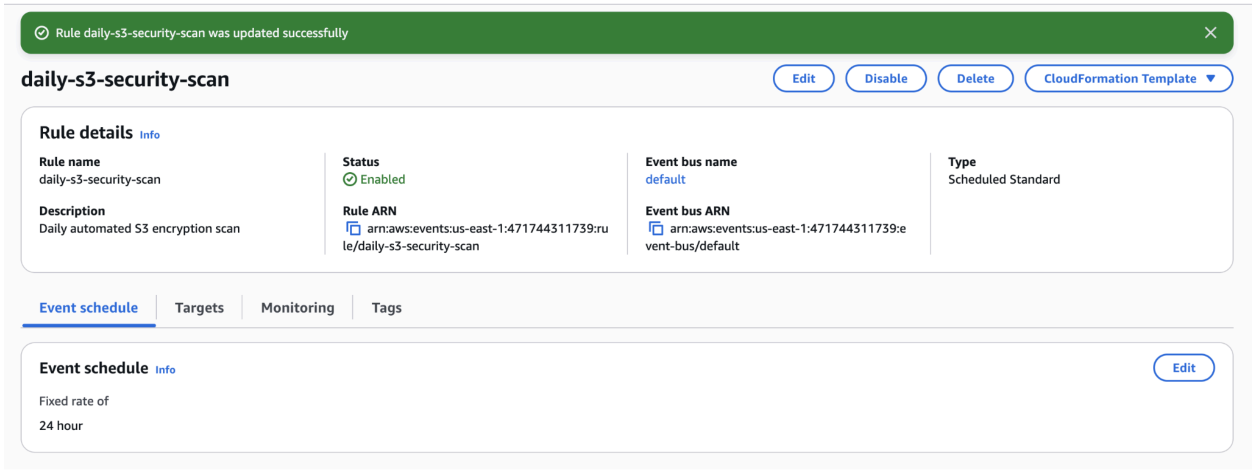
Log output

The area below shows the last 4 KB of the execution log. Click here to view the corresponding CloudWatch log group.

```
/var/task/s3_scanner.py:4: FutureWarning:
All support for the `google.generativeai` package has ended. It will no longer be receiving
updates or bug fixes. Please switch to the `google.genai` package as soon as possible.
See README for more details:
https://github.com/google-gemini/deprecated-generative-ai-python/blob/main/README.md
import google.generativeai as genai
START RequestId: 1faf7c7a-78b8-412f-b149-0a87c00e0deb Version: $LATEST
Scanning S3 buckets for encryption...
Found 0 buckets to scan
Analyzing security findings with Gemini AI...
```

Phase 4: Automation & Verification

- **Executed Manual Test**: I triggered the function using a blank test event. I then navigated to **CloudWatch Logs** to confirm the output. I observed that the AI successfully identified my unencrypted buckets and provided actionable remediation steps.



Rule daily-s3-security-scan was updated successfully

daily-s3-security-scan

Rule details

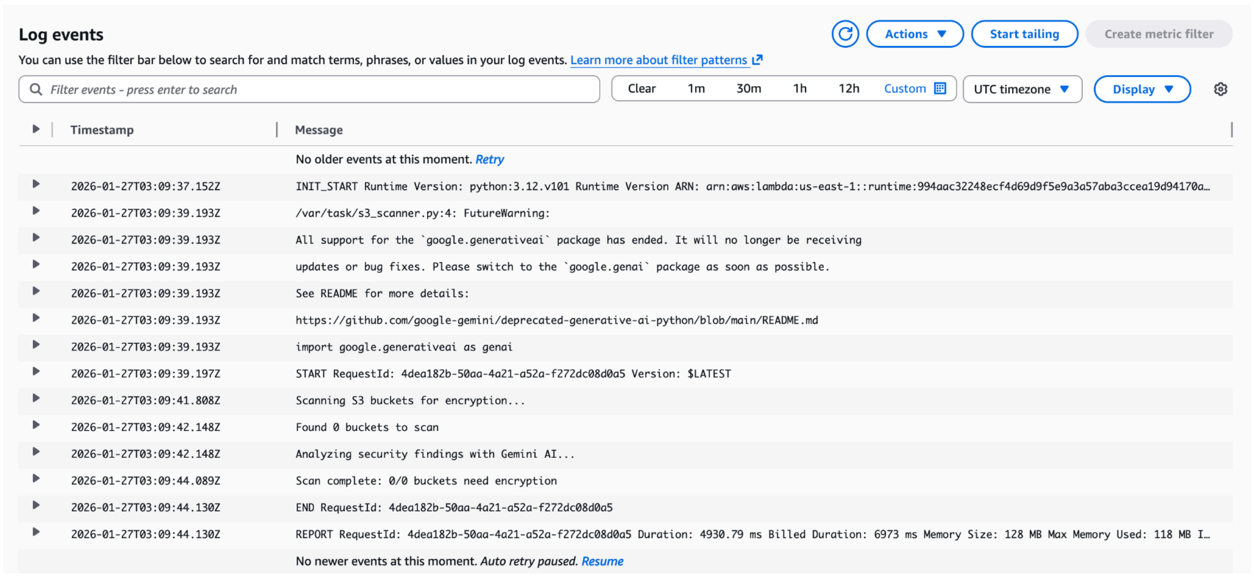
Rule name daily-s3-security-scan	**Status** Enabled	**Event bus name** default	**Type** Scheduled Standard
Description Daily automated S3 encryption scan	**Rule ARN** arn:aws:events:us-east-1:471744311739:rule/daily-s3-security-scan	**Event bus ARN** arn:aws:events:us-east-1:471744311739:event-bus/default	

Event schedule

Fixed rate of

24 hour

- **Scheduled Automation**: I moved to the **Amazon EventBridge** console and created a new "Scheduled Rule." I set a **rate of 12 hours** and selected my Lambda function as the target.
- **Validated Live Results**: To verify the automation, I temporarily changed the schedule to 1 minute. I watched the logs appear in real-time, confirming that my account is now being audited automatically twice a day.



You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

```
Timestamp                   Message
                            No older events at this moment. Retry
2026-01-27T03:09:37.152Z    INIT_START Runtime Version: python:3.12.v101 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:994aac32248ecf4d69d9f5e9a3a57aba3ccea19d94170a…
2026-01-27T03:09:39.193Z    /var/task/s3_scanner.py:4: FutureWarning:
2026-01-27T03:09:39.193Z    All support for the `google.generativeai` package has ended. It will no longer be receiving
2026-01-27T03:09:39.193Z    updates or bug fixes. Please switch to the `google.genai` package as soon as possible.
2026-01-27T03:09:39.193Z    See README for more details:
2026-01-27T03:09:39.193Z    https://github.com/google-gemini/deprecated-generative-ai-python/blob/main/README.md
2026-01-27T03:09:39.193Z    import google.generativeai as genai
2026-01-27T03:09:39.197Z    START RequestId: 4dea182b-50aa-4a21-a52a-f272dc08d0a5 Version: $LATEST
2026-01-27T03:09:41.808Z    Scanning S3 buckets for encryption...
2026-01-27T03:09:42.148Z    Found 0 buckets to scan
2026-01-27T03:09:42.148Z    Analyzing security findings with Gemini AI...
2026-01-27T03:09:44.089Z    Scan complete: 0/0 buckets need encryption
2026-01-27T03:09:44.130Z    END RequestId: 4dea182b-50aa-4a21-a52a-f272dc08d0a5
2026-01-27T03:09:44.130Z    REPORT RequestId: 4dea182b-50aa-4a21-a52a-f272dc08d0a5 Duration: 4930.79 ms Billed Duration: 6973 ms Memory Size: 128 MB Max Memory Used: 118 MB I…
                            No newer events at this moment. Auto retry paused. Resume
```

Summary of My Results: I successfully transitioned from a manual script to a 24/7 automated security auditor. The system now identifies vulnerabilities and translates them into a format that anyone on a security team can understand instantly.