

Cloud Security Auto-Remediation Architecture

Subtitle: Event-driven security automation system to detect and automatically remediate insecure cloud configurations in real time

Tech stack: Terraform · GitHub Actions · AWS CloudTrail · EventBridge · Lambda

Name: SeokHyun Yoo

Role: Security Automation / Cloud Security

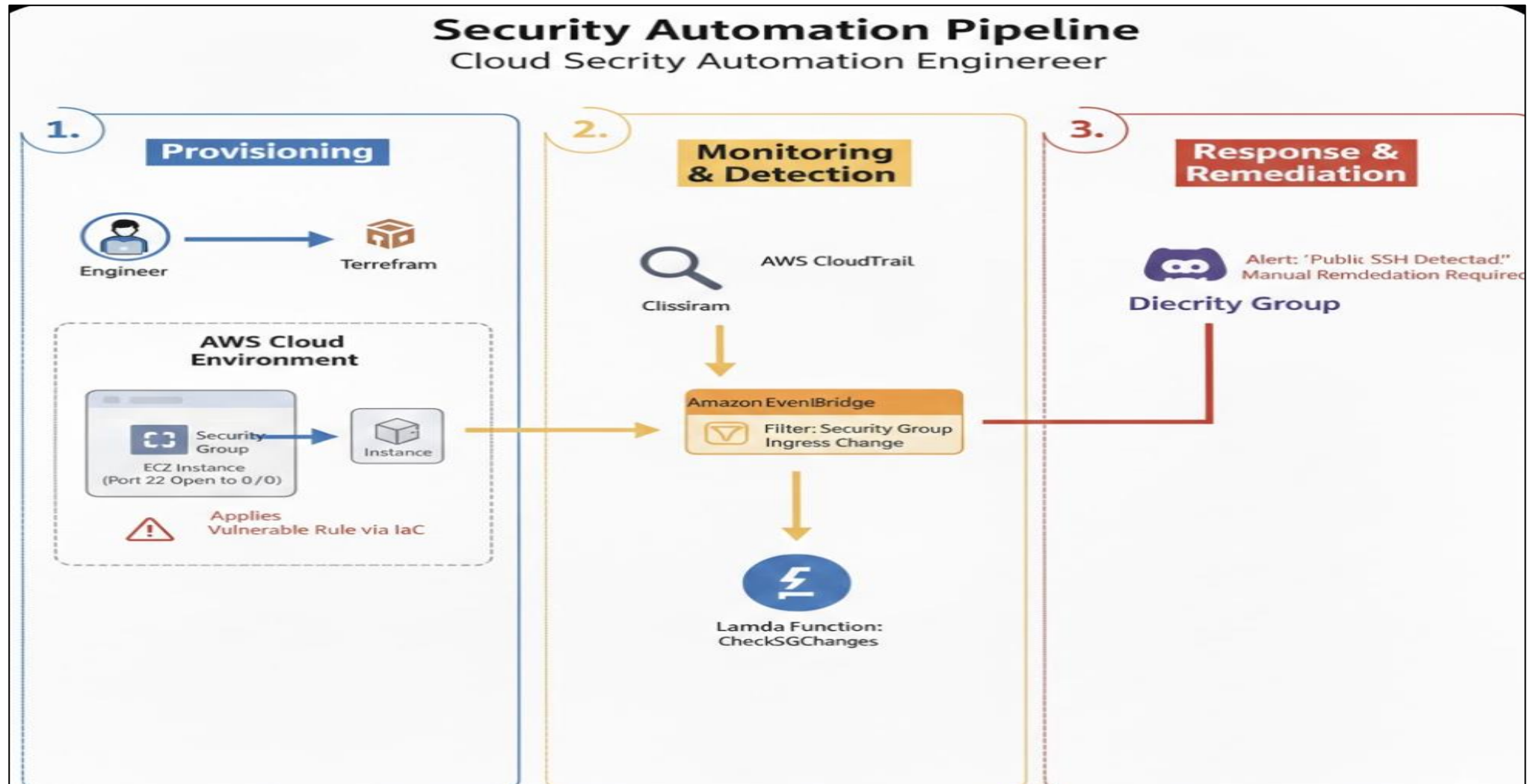
Keywords: CI/CD, IaC, Event-driven Security Automation

Problem Definition

- **현실 문제**
- 클라우드 환경에서 **Security Group 0.0.0.0/0 SSH** 오픈은
- 침해사고의 **가장 빈번한 초기 침투 벡터**

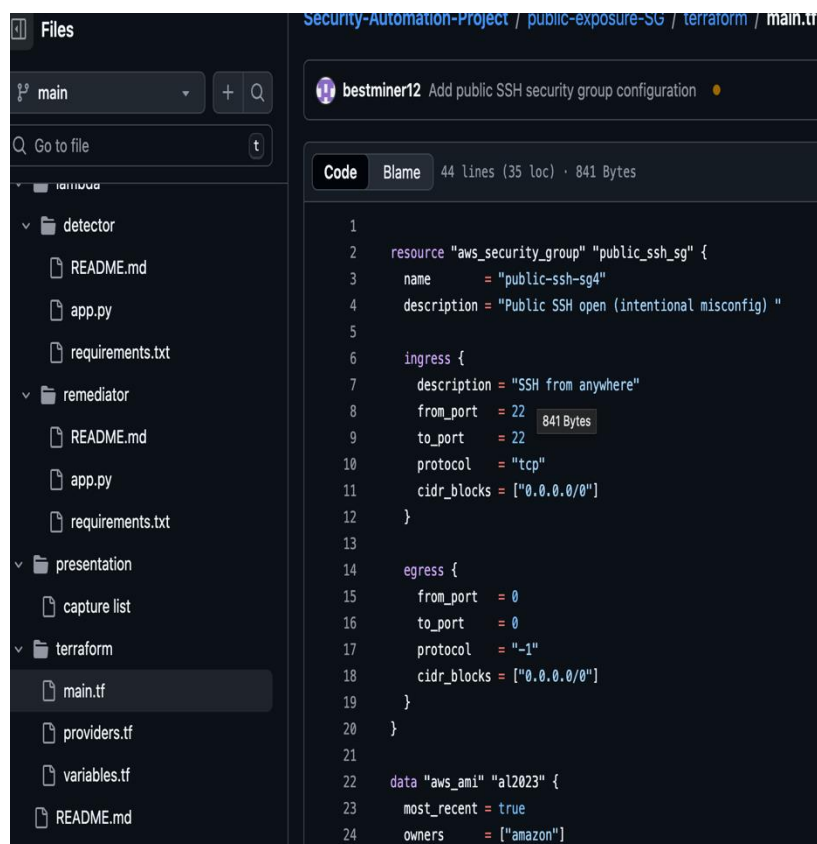
- 운영 환경에서:
- 수동 탐지 → 지연
- 수동 대응 → 휴먼 에러
- 야간/비근무 시간 → 대응 공백

High-Level Architecture



Terraform Apply, CI/CD

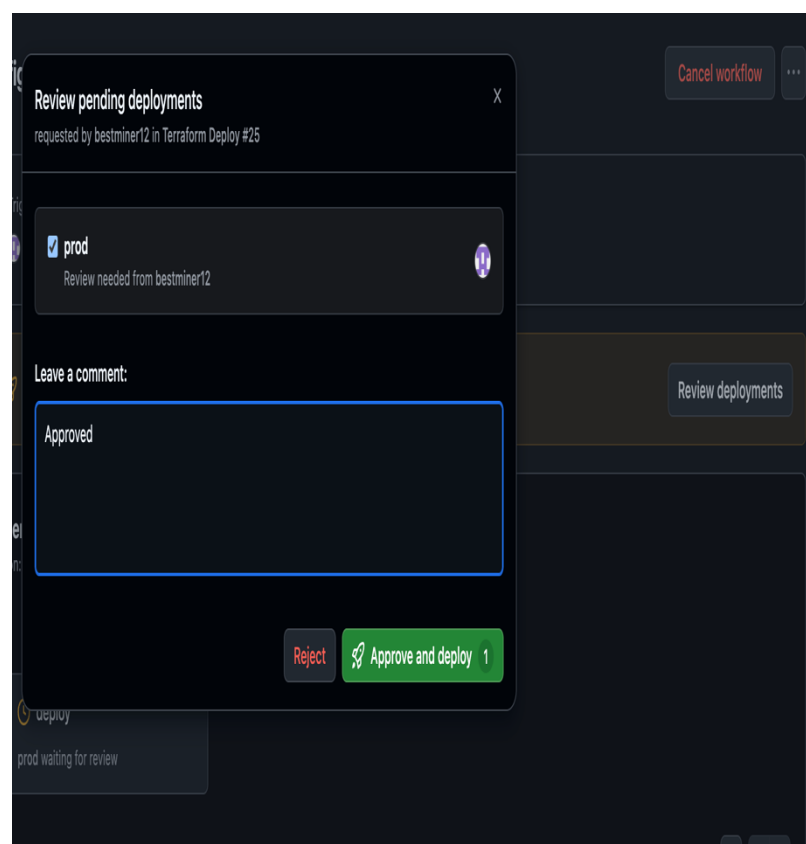
1. Terraform Code



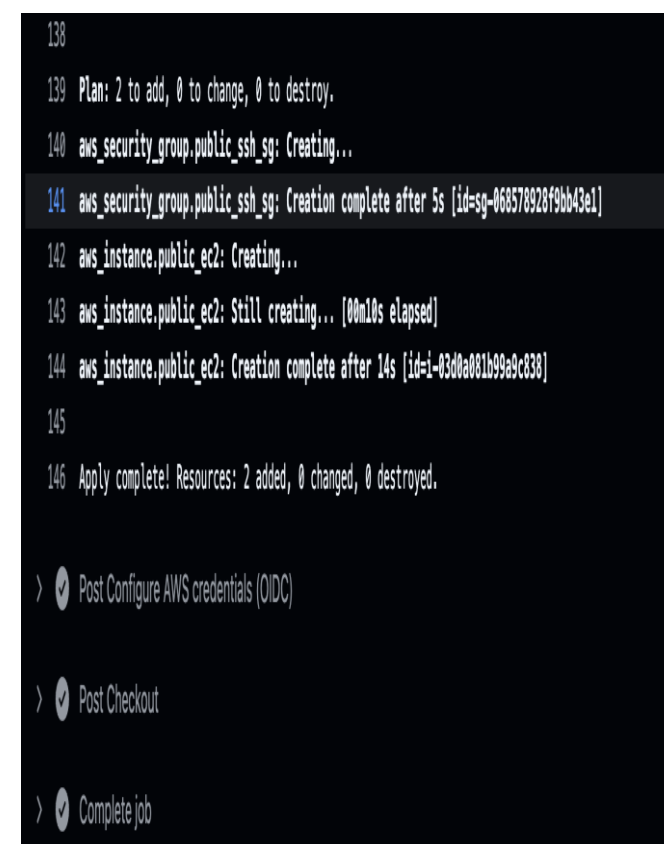
The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure with files like README.md, app.py, requirements.txt, and terraform/main.tf. The code editor shows the content of main.tf, which defines an AWS security group resource named "public_ssh_sg".

```
1 resource "aws_security_group" "public_ssh_sg" {
2   name       = "public-ssh-sg4"
3   description = "Public SSH open (intentional misconfig)"
4
5   ingress {
6     description = "SSH from anywhere"
7     from_port   = 22
8     to_port     = 22
9     protocol    = "tcp"
10    cidr_blocks = ["0.0.0.0/0"]
11  }
12
13  egress {
14    from_port = 0
15    to_port   = 0
16    protocol  = "-1"
17    cidr_blocks = ["0.0.0.0/0"]
18  }
19 }
20
21 data "aws_ami" "al2023" {
22   most_recent = true
23   owners      = ["amazon"]
24 }
```

2. Approval-based CD



3. Resource deploy



The screenshot shows a terminal window with the output of a Terraform apply command. The output shows the plan and the execution of the resources.

```
138
139 Plan: 2 to add, 0 to change, 0 to destroy.
140 aws_security_group.public_ssh_sg: Creating...
141 aws_security_group.public_ssh_sg: Creation complete after 5s [id=sg-068578928f9bb43e1]
142 aws_instance.public_ec2: Creating...
143 aws_instance.public_ec2: Still creating... [00m10s elapsed]
144 aws_instance.public_ec2: Creation complete after 14s [id=i-03d0a081b99a9c838]
145
146 Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
```

Below the terminal output, there are three steps in a list:

- Post Configure AWS credentials (OIDC)
- Post Checkout
- Complete job

AWS Console Resource check

1. SG open to any public ssh port

sg-068578928f9bb43e1 - public-ssh-sg4

Details **Inbound rules** Outbound rules Sharing VPC associations Tags

Inbound rules (1)

Search

< 1 > ⚙

n	Type	Protocol	Port range	Source	Description
	SSH	TCP	22	0.0.0.0/0	SSH from anywhere

2. EC2 setting public IP

i-03d0a081b99a9c838 (public-exposure-ec2)

Instance summary [Info](#)

Instance ID
i-03d0a081b99a9c838

IPv6 address
-

Public IPv4 address
15.164.104.36 | [open address](#)

Private IPv4 addresses
172.31.43.120

Instance state
Running

Public DNS
ec2-15-164-104-36.ap-northeast-2.compute.amazonaws.com | [open address](#)

Lambda, Event bridge setting

1. Detecting Lambda setting

The screenshot shows the AWS Lambda console for the function 'public-exposure-SG-role'. The 'Diagram' tab is active, displaying a visual representation of the function's triggers and destinations. An 'EventBridge (CloudWatch Events)' trigger is connected to the function. The 'Add destination' button is visible. The right sidebar shows the function's details: Description, Last modified (59 minutes ago), Function ARN (arn:aws:lambda:ap-northeast-2:252321108840:function:public-exposure-SG-role), and Function URL. The bottom navigation bar includes tabs for Code, Test, Monitor, Configuration, Aliases, and Versions. The 'Code source' section at the bottom shows the function's code location and options to 'Open in Visual Studio Code' or 'Upload from'.

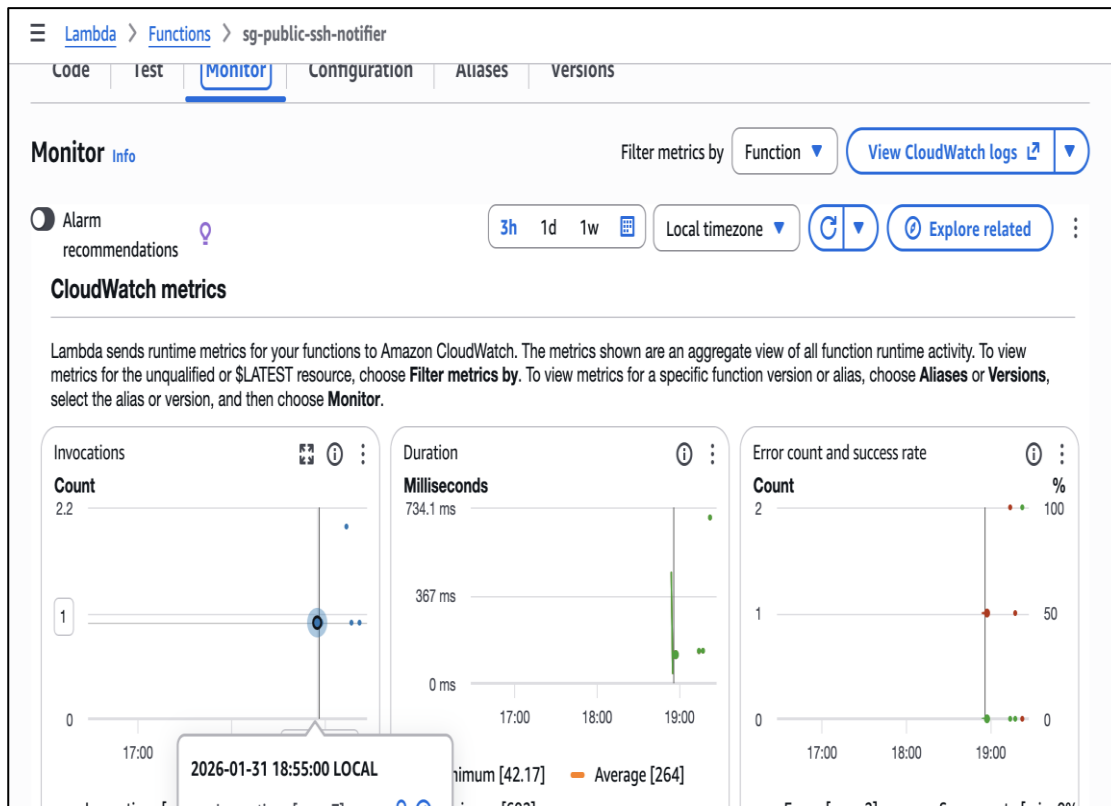
2. Event bridge setting

The screenshot shows the AWS EventBridge console for the rule 'detect-ec2-sg-public-ssh'. The 'Rule details' tab is active, displaying the rule's configuration. The rule is 'Enabled'. The 'Rule name' is 'detect-ec2-sg-public-ssh'. The 'Status' is 'Enabled'. The 'Rule ARN' is 'arn:aws:events:ap-northeast-2:252321108840:rule/detect-ec2-sg-public-ssh'. The 'Event bus name' is 'default'. The 'Event bus ARN' is 'arn:aws:events:ap-northeast-2:252321108840:event-bus/default'. The 'Event pattern' tab is also visible, showing the event pattern configuration in JSON format.

```
1 {
2   "source": ["aws.ec2"],
3   "detail-type": ["AWS API Call via CloudTrail"],
4   "detail": {
5     "eventSource": ["ec2.amazonaws.com"],
6     "eventName": ["AuthorizeSecurityGroupIngress"]
7   }
8 }
```

Send alert info to discord

1. Notifier lamda setting



2. Send alert to discord

The screenshot shows a Discord message from 'security-alert-bot' in a channel named '# 일반'. The message is dated 2026년 1월 31일 and was sent at 오후 7:23. The message content is as follows:

SECURITY EVENT DETECTED

An AWS management event was detected and forwarded by the automation pipeline.

Severity	Event	Region
HIGH	N/A	ap-northeast-2

Actor	Source IP	Event Source
N/A	N/A	N/A

Resource
N/A

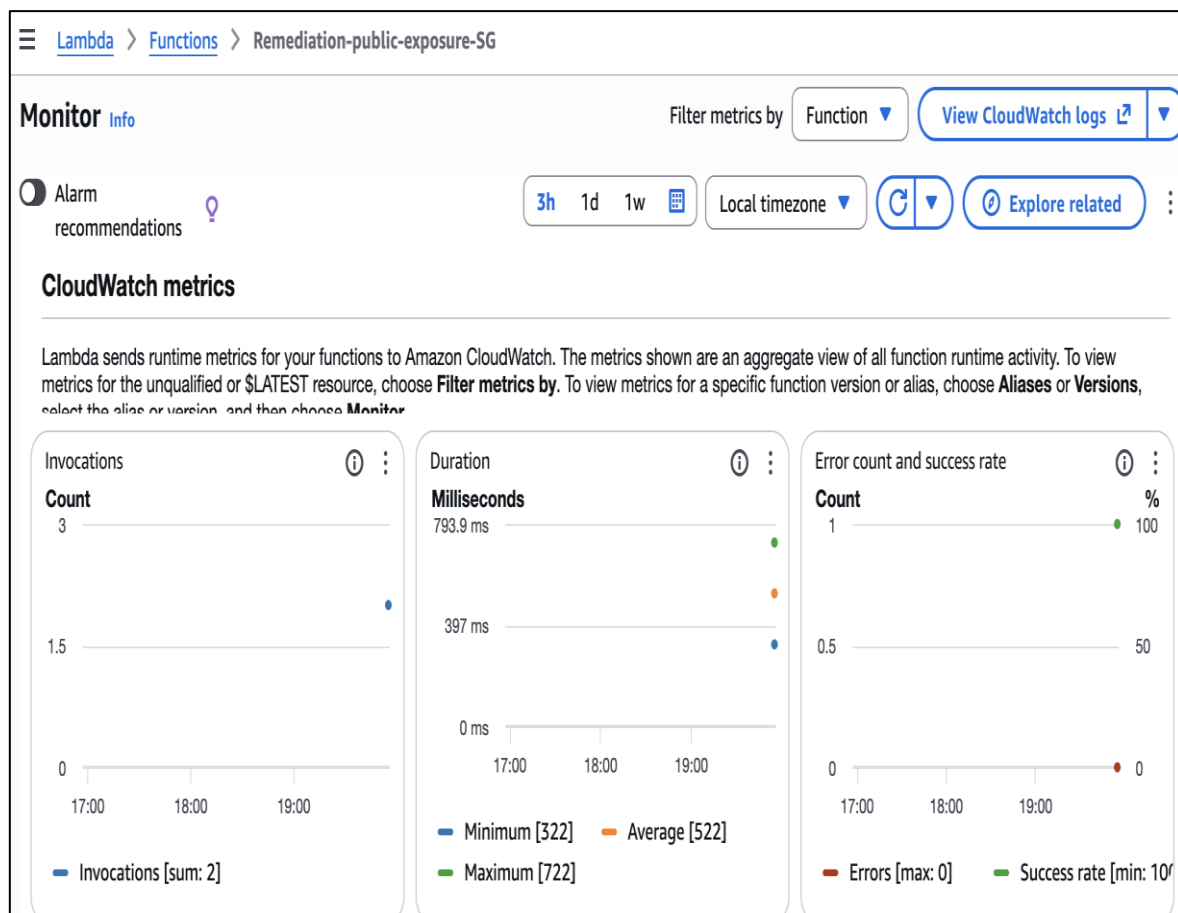
Detail (preview)

```
{"account": "252321108840", "region": "ap-northeast-2", "security_group": "sg-07ecaba33d8fa7af9", "port": 22, "cidr": "0.0.0.0/0", "time": "2026-01-31T10:27:17Z"}
```

AWS EventBridge → Lambda → Discord | Security Automation • 오늘 오후 7:27

Automated Remediation (Security Group Hardening)

1. Remediation Lambda Configuration



2. Analyst-Driven Remediation Decision

