

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

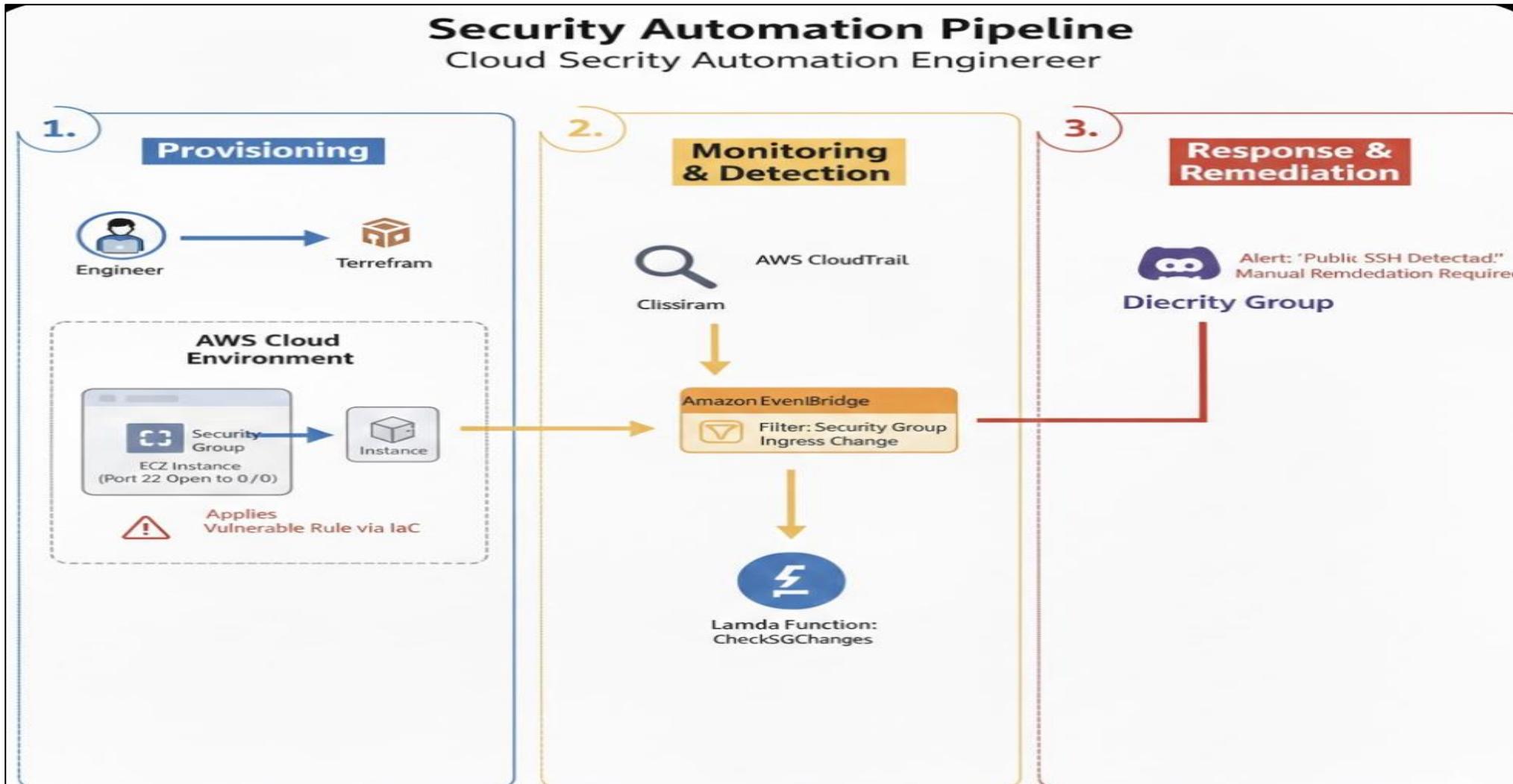
Role: Security Automation / Cloud Security

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



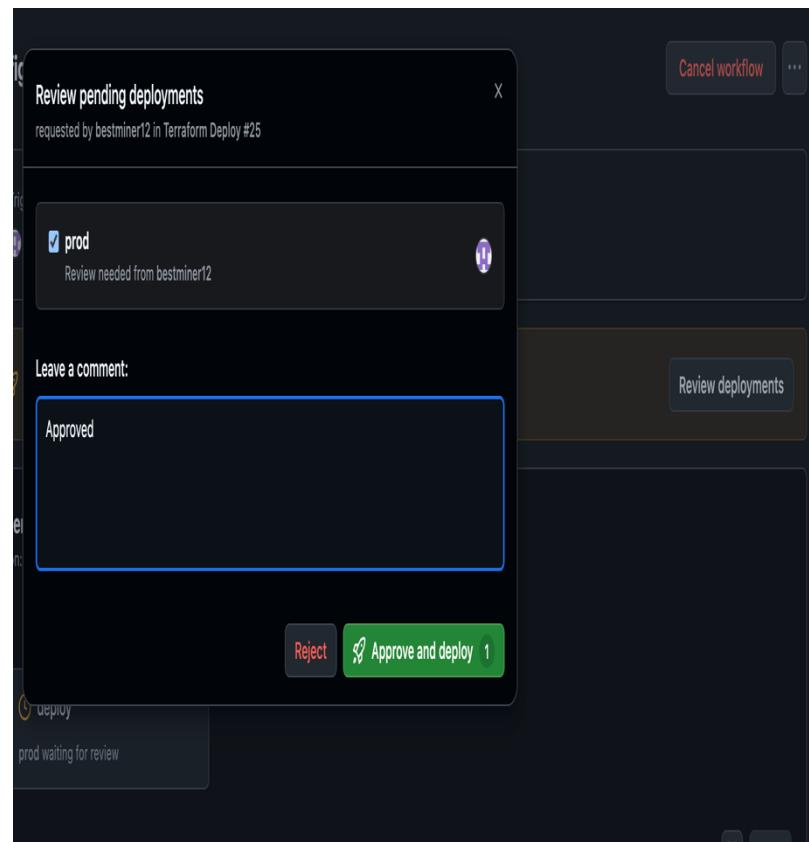
Terraform Apply, CI/CD

1. Terraform Code

The screenshot shows a code editor interface with a sidebar containing project and file navigation. The main area displays a Terraform configuration file named 'main.tf'. The code defines an AWS Security Group ('public_ssh_sg') with an ingress rule allowing SSH from anywhere on port 22. It also includes an egress rule allowing all traffic. A data block for an Amazon Linux 2 AMI is defined at the bottom.

```
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
14  egress {
15    from_port  = 0
16    to_port    = 0
17    protocol   = "-1"
18    cidr_blocks = ["0.0.0.0/0"]
19  }
20
21
22  data "aws ami" "al2023" {
23    most_recent = true
24    owners      = ["amazon"]
25  }
}
```

2. Approval-based CD



3. Resource deploy

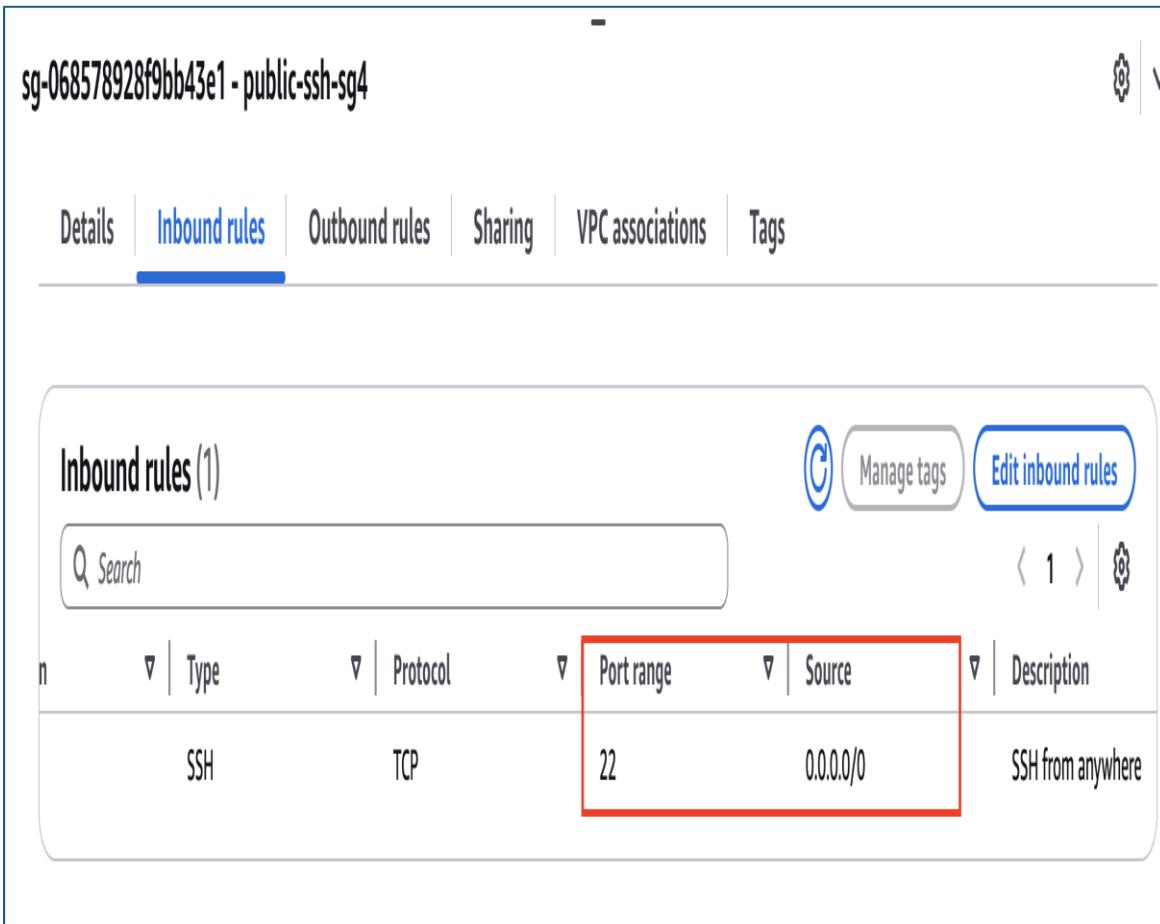
The screenshot shows a log or pipeline interface with several stages. The first stage, 'Post Configure AWS credentials (OIDC)', has a status of 'Success'. The second stage, 'Post Checkout', also has a status of 'Success'. The third stage, 'Complete job', is currently running. The main log area shows the execution of Terraform commands, starting with a plan phase that identifies 2 resources to add. Subsequent lines show the creation of an AWS security group and an EC2 instance, with their respective IDs and creation times.

```
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-063578928f9bb43e1]
142 aws_instance.public_ec2: Creating...
143 aws_instance.public_ec2: Still creating... [0m10s elapsed]
144 aws_instance.public_ec2: Creation complete after 14s [id=i-03d0a081b99a9c838]
145
146 Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

> ✓ Post Configure AWS credentials (OIDC)
> ✓ Post Checkout
> ✓ Complete job
```

AWS Console Resource check

1. SG open to any public ssh port

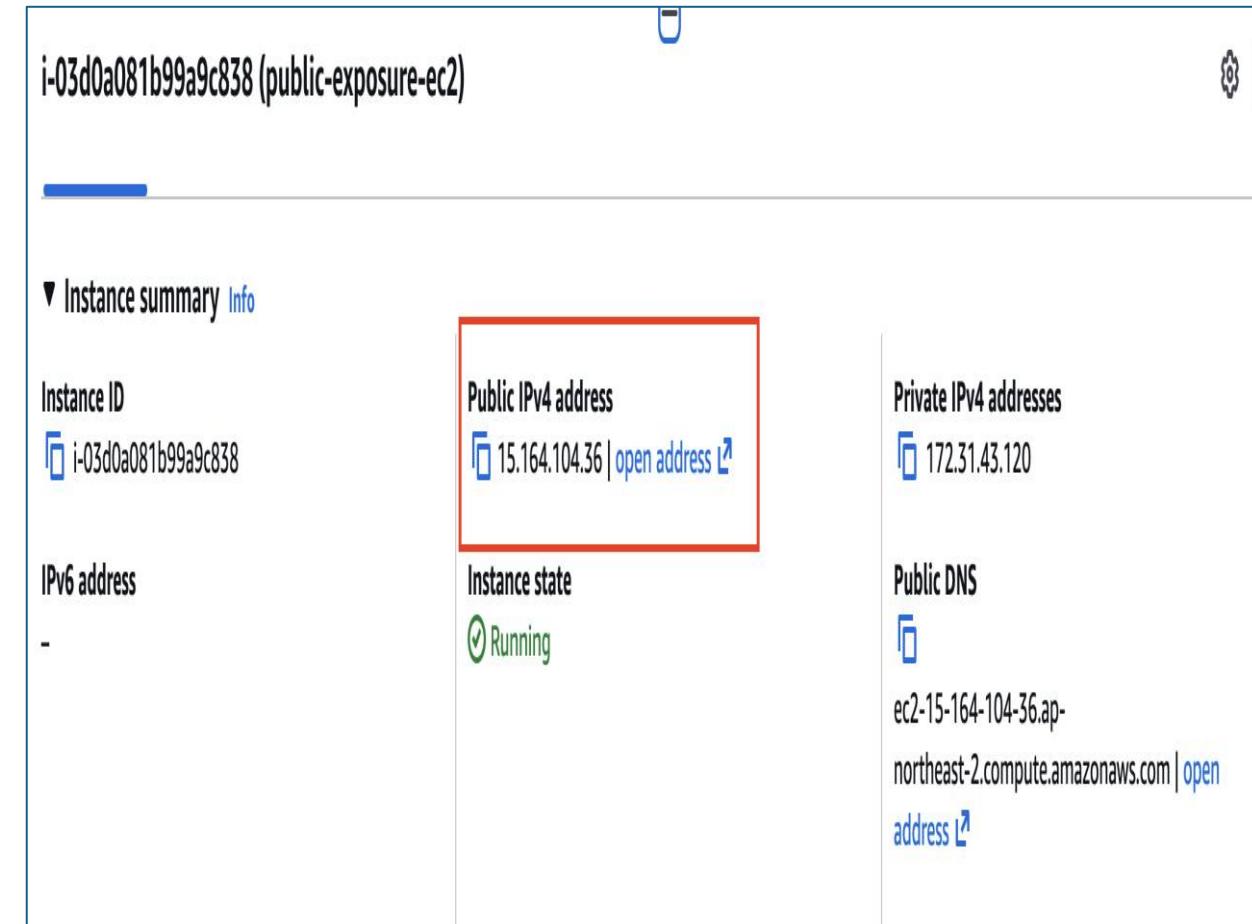


The screenshot shows the AWS Security Groups console for a specific security group named "sg-068578928f9bb43e1 - public-ssh-sg4". The "Inbound rules" tab is selected. A single rule is listed under "Inbound rules (1)". The rule details are as follows:

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

The "Source" field is highlighted with a red box.

2. EC2 setting public IP



The screenshot shows the AWS EC2 Instances console for an instance named "i-03d0a081b99a9c838 (public-exposure-ec2)". The "Instance summary" section is expanded. The instance details are as follows:

- Instance ID: i-03d0a081b99a9c838
- Public IPv4 address: 15.164.104.36 (highlighted with a red box)
- Private IPv4 addresses: 172.31.43.120
- IPv6 address: -
- Instance state: Running
- Public DNS: ec2-15-164-104-36.ap-northeast-2.compute.amazonaws.com (highlighted with a blue box)

Lamda, Event bridge setting

1. Detecting Lamda setting

The screenshot shows the AWS Lambda function configuration page for the function 'public-exposure-SG-role'. The top navigation bar includes 'Lambda', 'Functions', and 'public-exposure-SG-role'. Below the navigation, there are tabs for 'Diagram' (selected), 'Template', 'Code', 'Test', 'Monitor', 'Configuration', 'Aliases', and 'Versions'.

The main content area displays the function's details:

- Description:** -
- Last modified:** 59 minutes ago
- Function ARN:** arn:aws:lambda:ap-northeast-2:252321108840:function:public-exposure-SG-role
- Function URL:** Info

On the left side, there are three boxes: 'public-exposure-SG-role' (with a Lambda icon), 'EventBridge (CloudWatch Events)' (with a CloudWatch Events icon), and '+ Add trigger' (with a plus icon).

At the bottom, there are buttons for 'Code source' (Info), 'Open in Visual Studio Code', 'Upload from', and file icons. The 'EXPLORER' tab is visible at the bottom left.

2. Event bridge setting

The screenshot shows the AWS EventBridge rule configuration page for the rule 'detect-ec2-sg-public-ssh'. The top navigation bar includes the rule name, 'Edit', 'Disable', and 'Delete' buttons.

The main content area displays the rule details:

- Rule name:** detect-ec2-sg-public-ssh
- Status:** Enabled
- Rule ARN:** arn:aws:events:ap-northeast-2:252321108840:rule/detect-ec2-sg-public-ssh

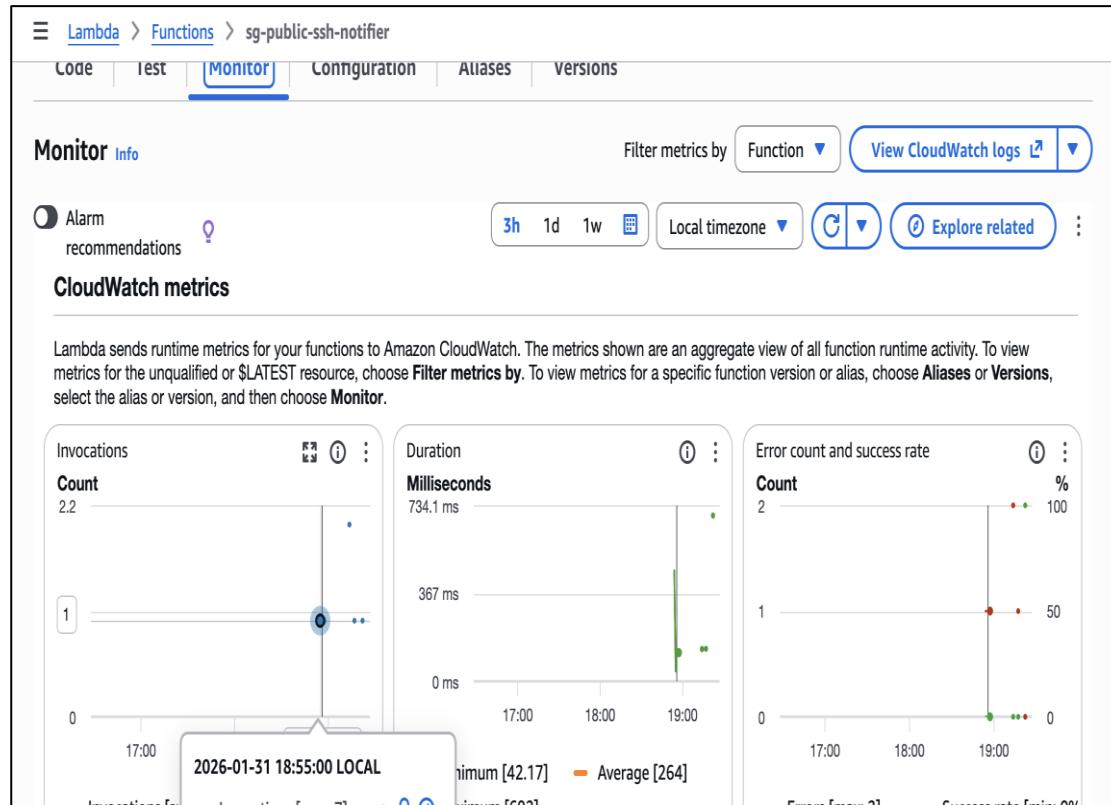
Below the rule details, there are tabs for 'Event pattern' (selected), 'Targets', 'Monitoring', and 'Tags'.

The 'Event pattern' section shows the JSON configuration:

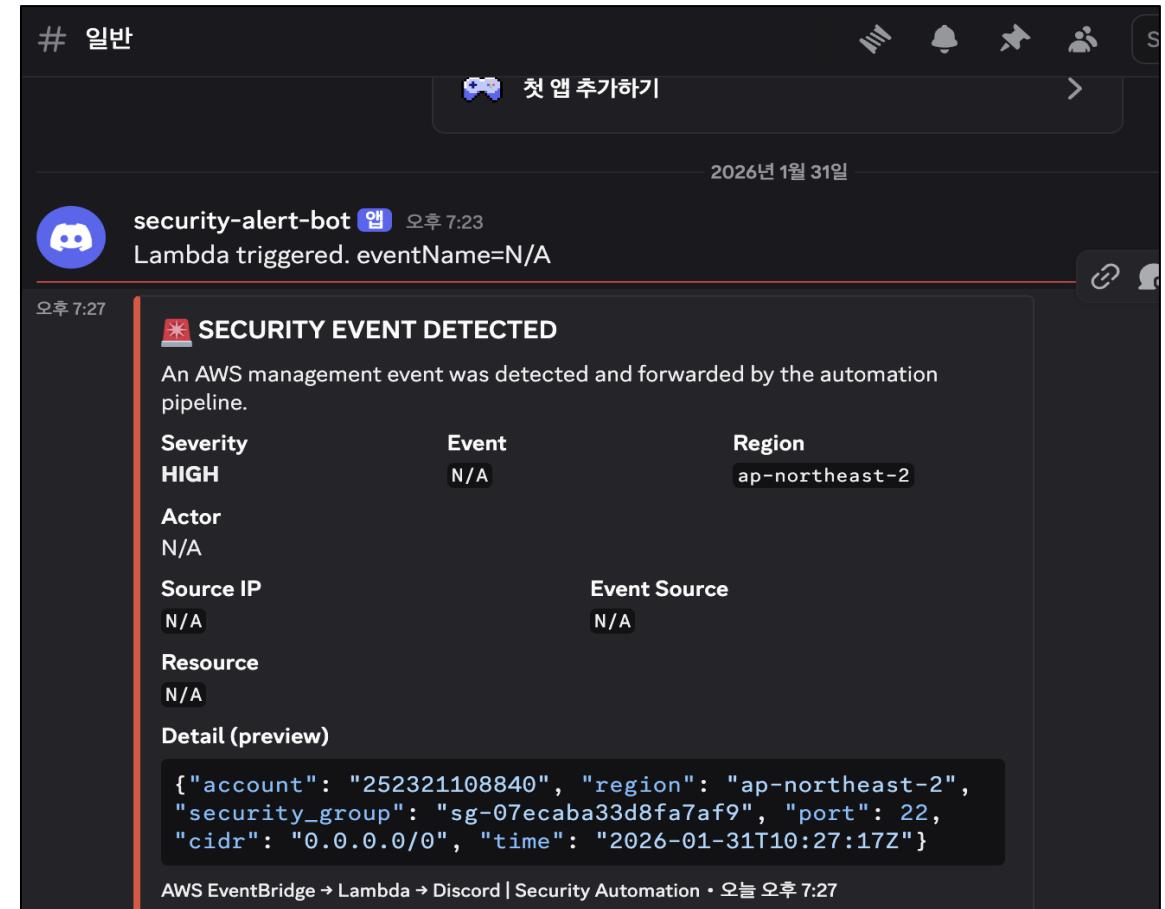
```
1 {  
2   "source": ["aws.ec2"],  
3   "detail-type": ["AWS API Call via CloudTrail"],  
4   "detail": {  
5     "eventSource": ["ec2.amazonaws.com"],  
6     "eventName": ["AuthorizeSecurityGroupIngress"]  
7   }  
}
```

Send alert info to discord

1. Notifier lambda setting

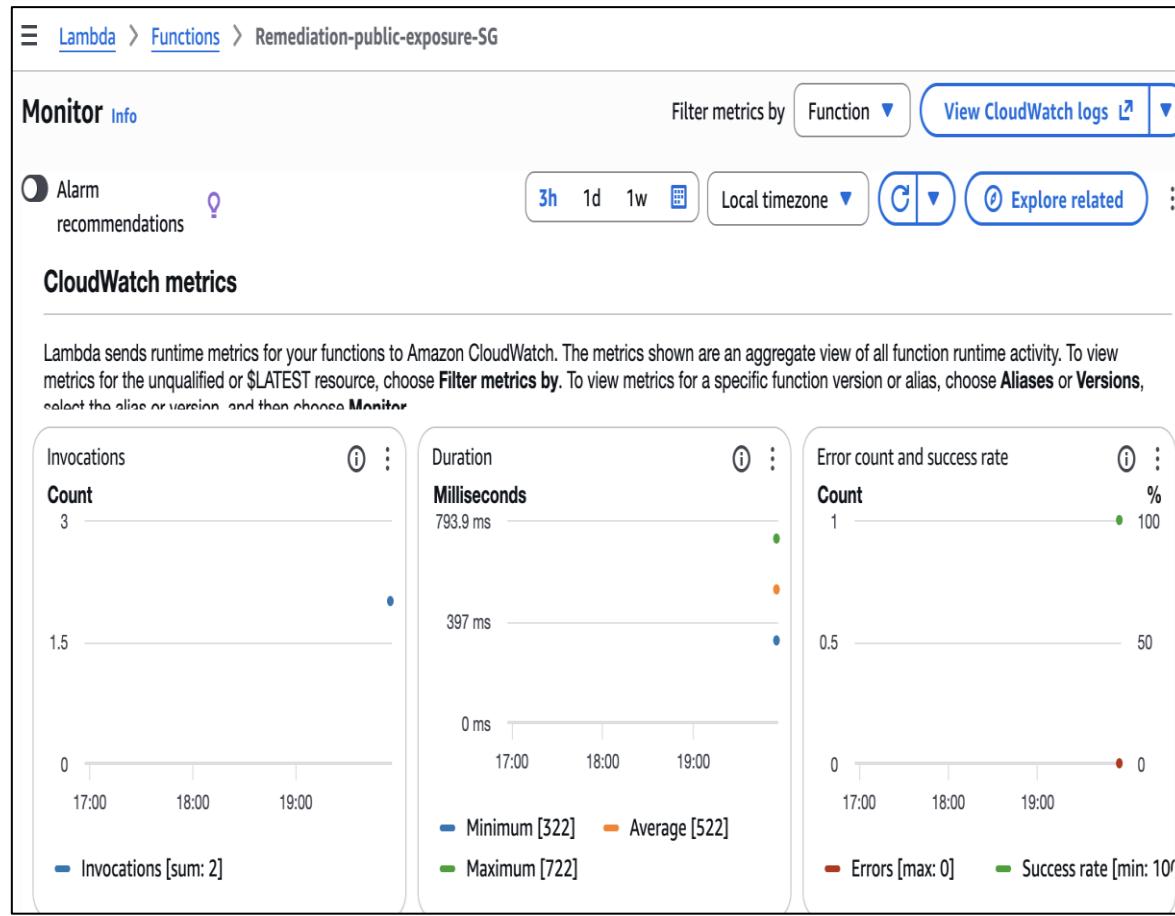


2. Send alert to discord



Automated Remediation (Security Group Hardening)

1. Remediation Lambda Configuration



2. Analyst-Driven Remediation Decision

