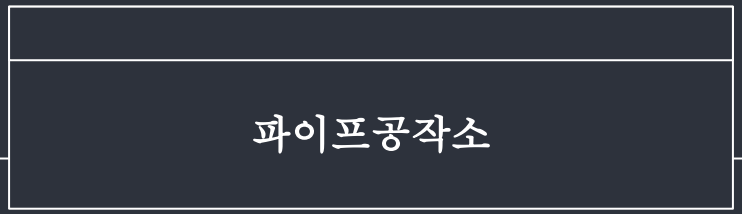
</DevSecOps를 위한 취약점 관리 자동화 도구 만들기/>



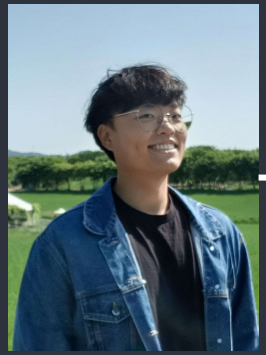
1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 1 0 1 1 0 1

</목차

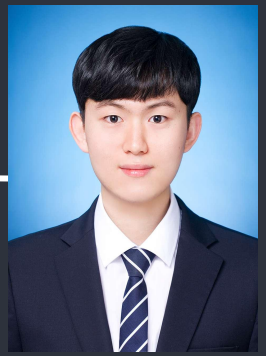
</팀 소개

- 팀 명 : 파이프 공작소
- 노 션 : https://www.notion.so/0dd1861e0fbc45cb97fa2a60d8703bad?pvs=4
- Slack : https://app.slack.com/client/T062695QNRH
- Github : https://github.com/nanac0n/pipeworkshop.git

</팀 소개

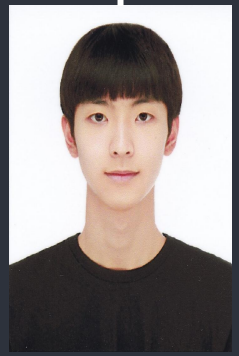


권현준
멘토

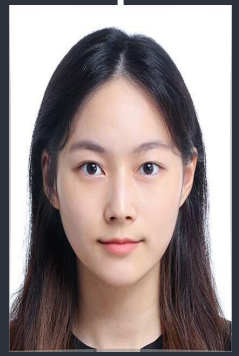


신정식
PL

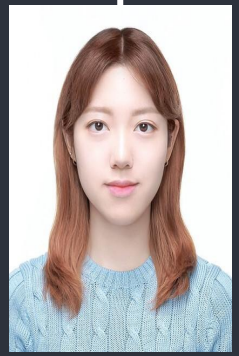
</팀 소개



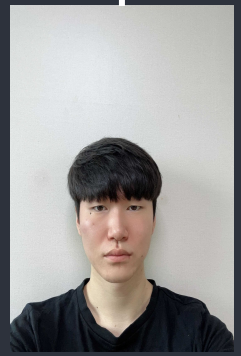
정준우
PM



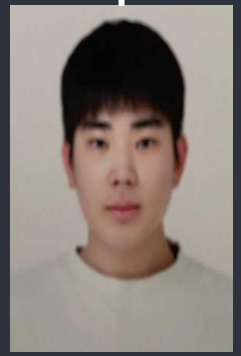
이호정
팀원



장지인
팀원



장대한
팀원

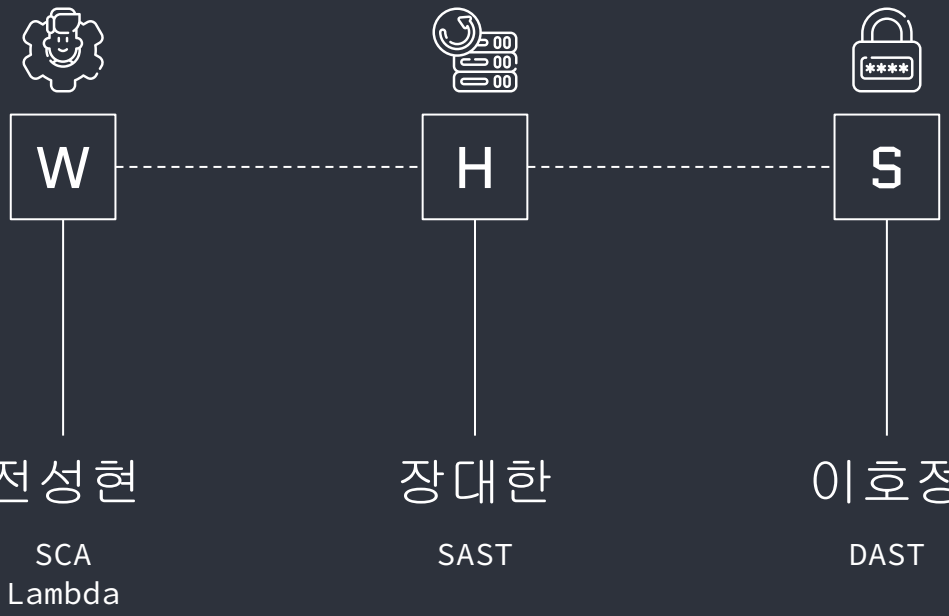


조현성
팀원

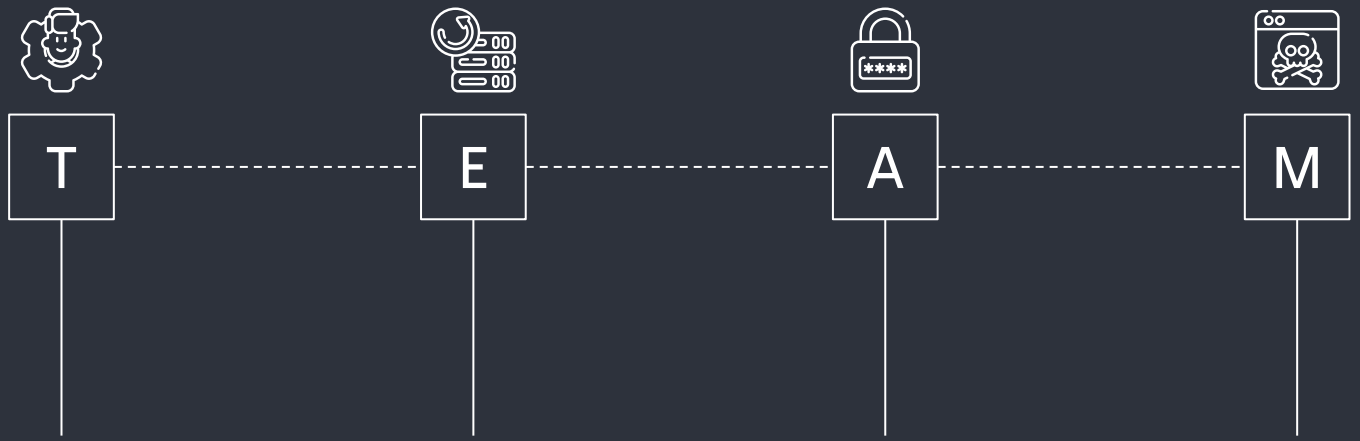
전성현
팀원

홍리나
팀원

</Test-Deploy Team



</Monitoring Team



</진행 과정

Step 1

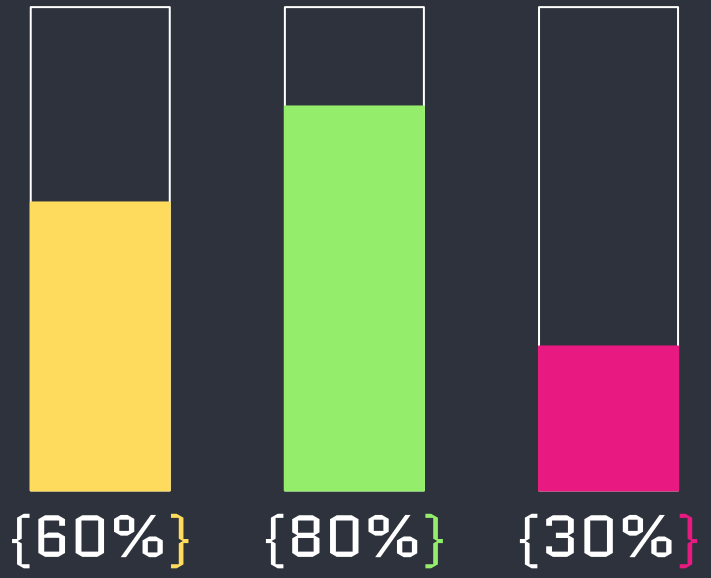
파이프라인 구성 및 구현

Step 2

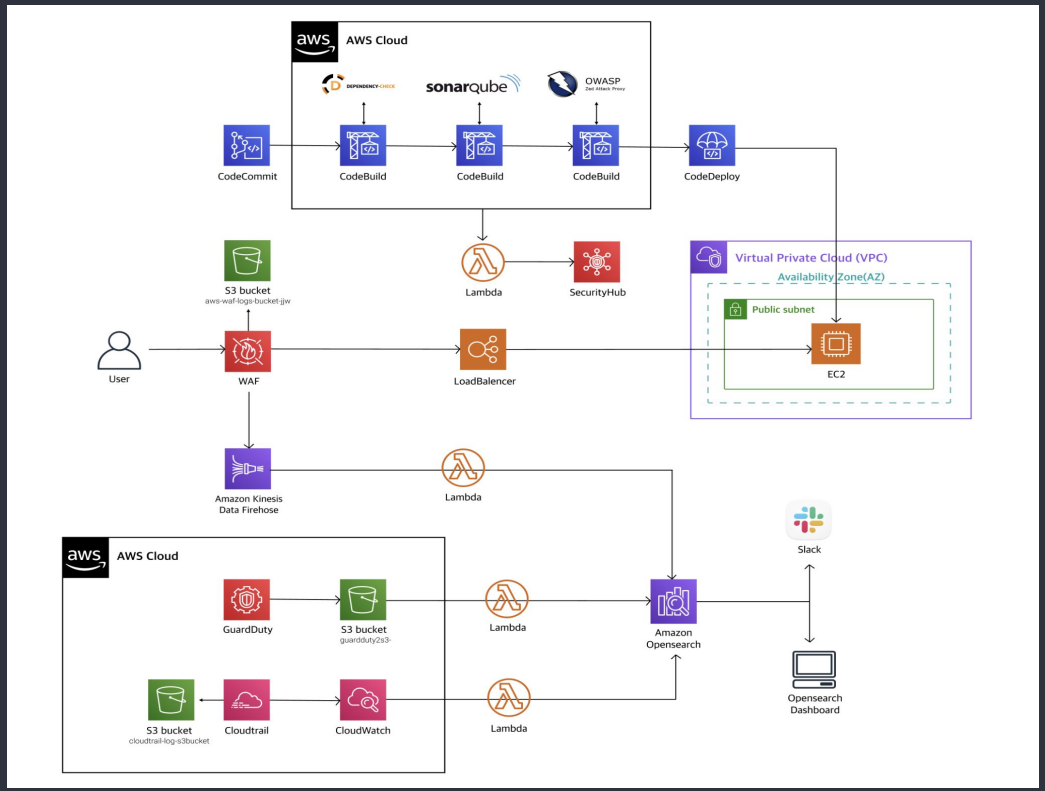
IaC 제작

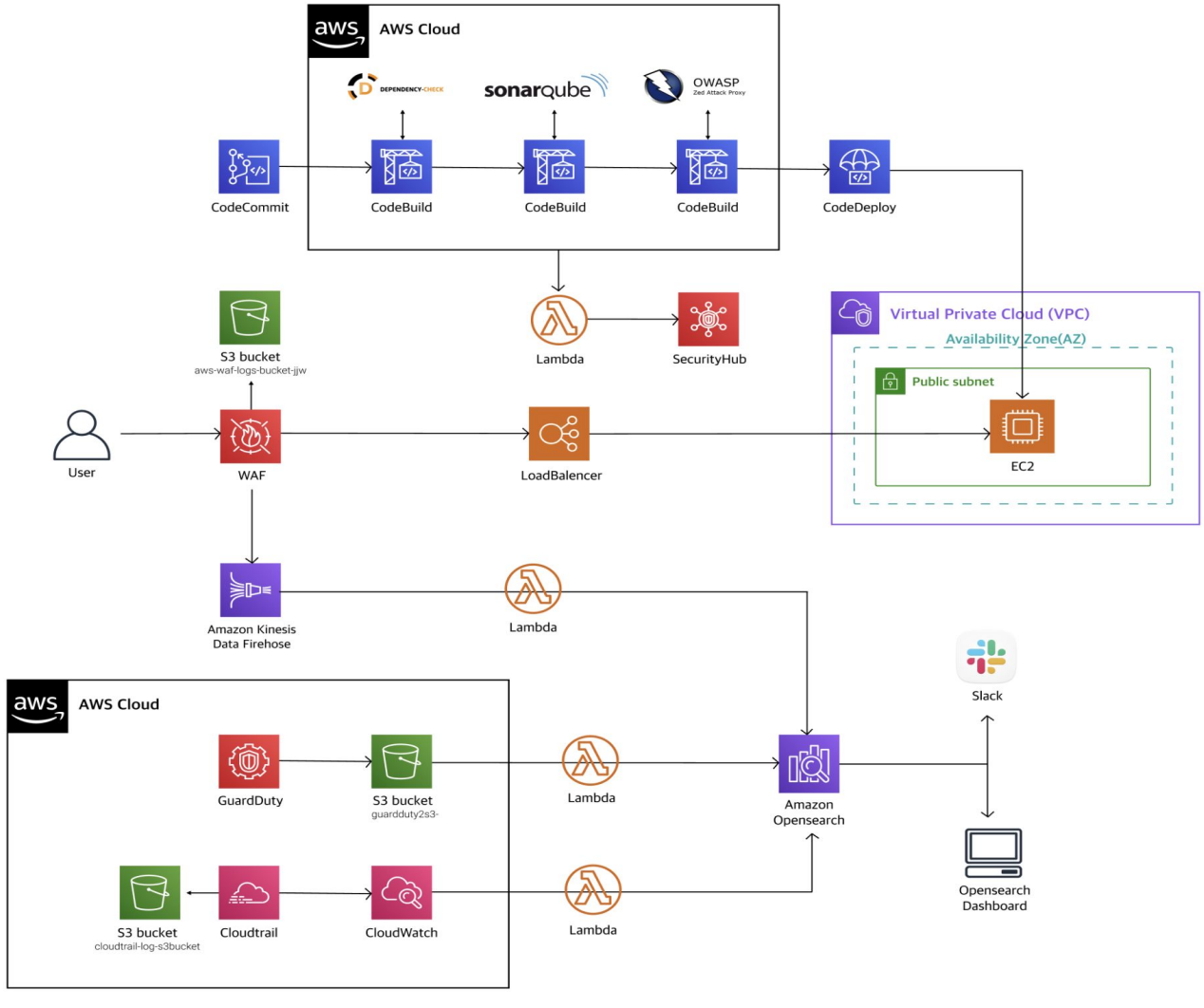
Step 3

Open Source 배포

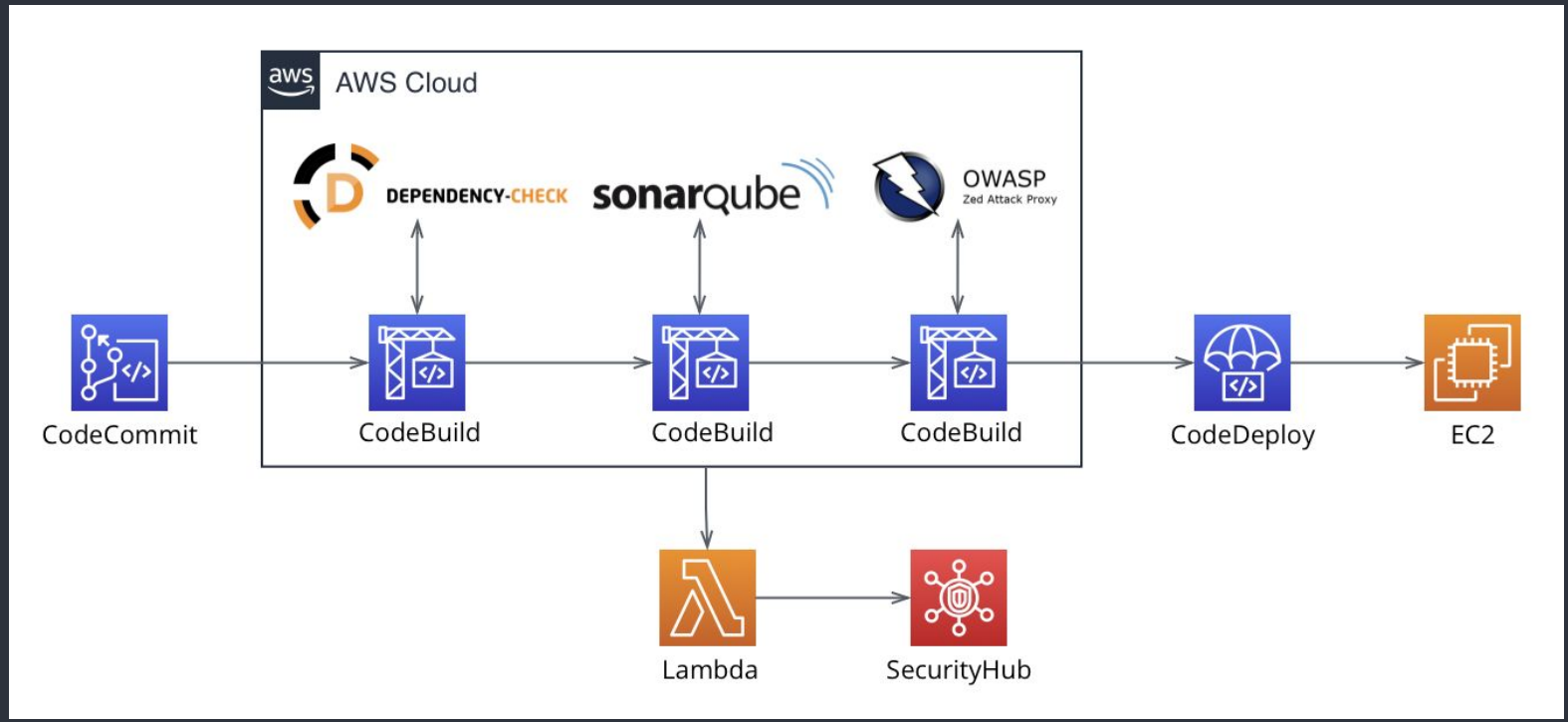


</최종 산출물

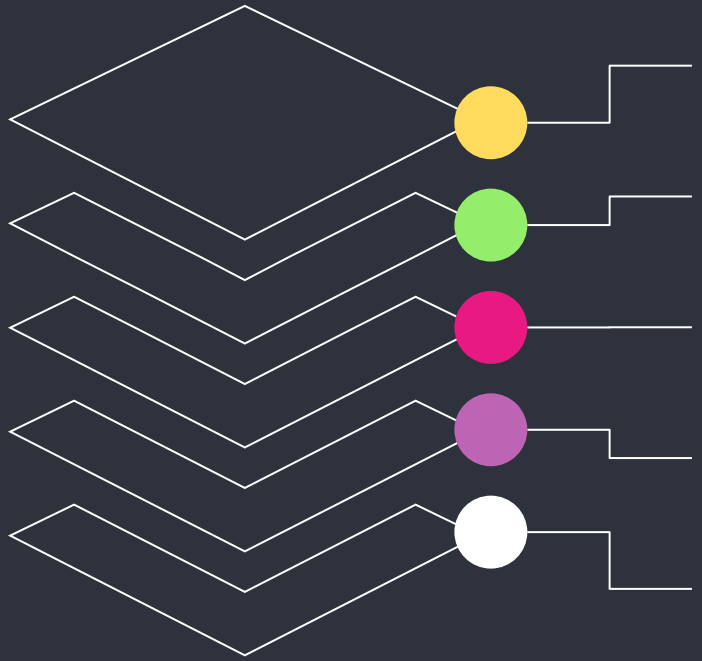

AWS Cloud
sonarqube
OWASP
CodeCommit
CodeBuild
CodeBuild
CodeBuild
CodeDeploy
S3 bucket
aws-waf-logs-bucket-jjw
Lambda
SecurityHub
Virtual Private Cloud (VPC)
Availability Zone(AZ)
Public subnet
EC2
User
WAF
LoadBalancer
Amazon Kinesis
Data Firehose
Lambda
Slack
AWS Cloud
GuardDuty
S3 bucket
Lambda
Amazon
Opensearch
S3 bucket
cloudtrail-log-s3bucket
Cloudtrail
CloudWatch
Lambda
Opensearch
Dashboard


AWS Cloud
DEPENDENCY-CHECK
sonarqube
OWASP
Zed Attack Proxy
CodeCommit
CodeBuild
CodeBuild
CodeBuild
CodeDeploy
S3 bucket
aws-waf-logs-bucket-jjw
Lambda
SecurityHub
Virtual Private Cloud (VPC)
Availability Zone(AZ)
Public subnet
EC2
User
WAF
LoadBalancer
Amazon Kinesis
Data Firehose
Lambda
Slack
AWS Cloud
GuardDuty
S3 bucket
guardduty2s3-
Lambda
Amazon
Opensearch
S3 bucket
cloudtrail-log-s3bucket
Cloudtrail
CloudWatch
Lambda
Opensearch
Dashboard

</Test-Deploy AWS Arch



</ Procedure



{01} SCA — Owasp dependency check

{02} SAST — sonarqube

{03} DAST — zap

{04} Lambda — AWS Lambda

{05} Security Hub — AWS Security Hub

</SCA

종속성
분석|감지

{01}

취약점 DB

{02}

다양한 형식의
출력

{03}

</SCA - Codebuild

Codebuild

AWS Systems Manager

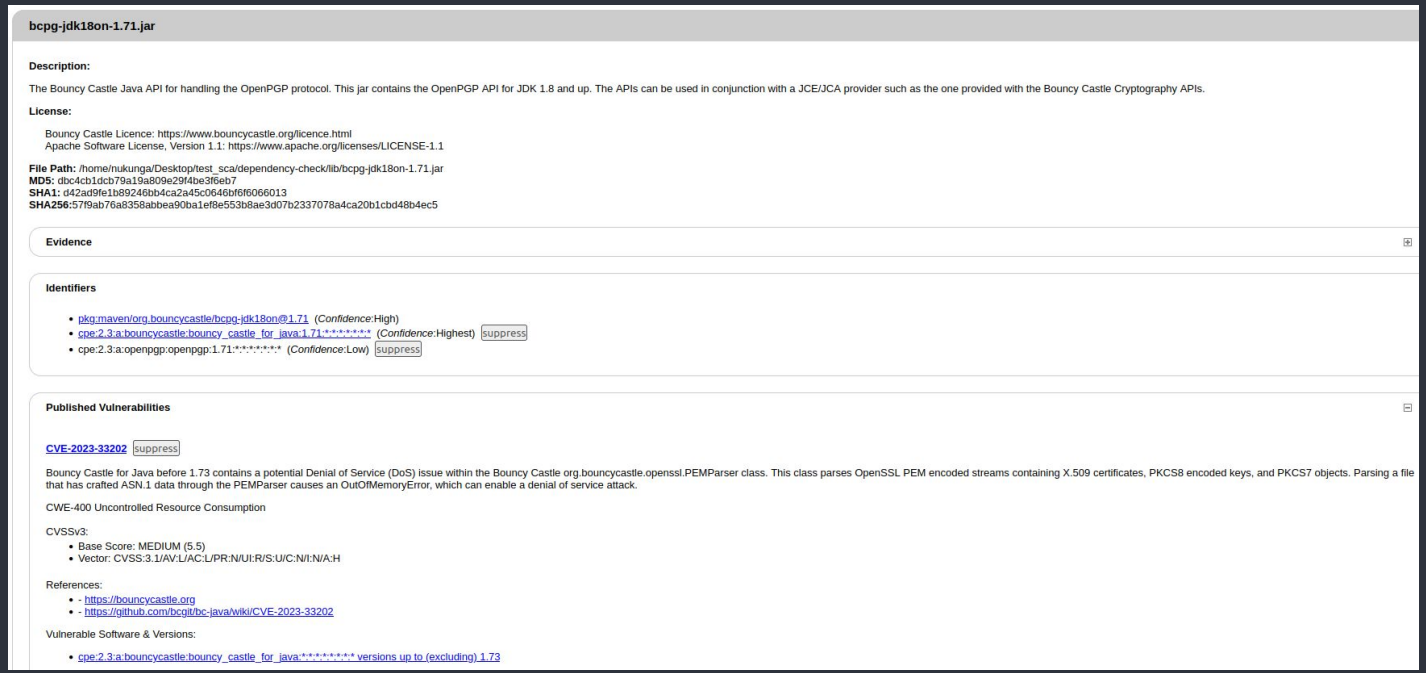
buildspec.yml

buildspec_owasp_check.yml

Codebuild

소스 코드 컴파일 및 단위 테스트 실행
배포 준비 완료된 아티팩트 생성

</SCA - Report



bcpg-jdk18on-1.71.jar

Description:

The Bouncy Castle Java API for handling the OpenPGP protocol. This jar contains the OpenPGP API for JDK 1.8 and up. The APIs can be used in conjunction with a JCE/JCA provider such as the one provided with the Bouncy Castle Cryptography APIs.

License:

Bouncy Castle Licence: https://www.bouncycastle.org/licence.html
Apache Software License, Version 1.1: https://www.apache.org/licenses/LICENSE-1.1

File Path: /home/nukunga/Desktop/test_sca/dependency-check/lib/bcpg-jdk18on-1.71.jar
MD5: dbc4cb1dcb79a19a809e29f4be3f6eb7
SHA1: d42ad9fe1b89246bb4ca2a45c0646bf6f6066013
SHA256:57f9ab76a8358abbea90ba1ef8e553b8ae3d07b2337078a4ca20b1cbd48b4ec5

Evidence

Identifiers

- pkg:maven/org.bouncycastle/bcpg-jdk18on@1.71 (*Confidence*:High)
- cpe:2.3:a:bouncycastle:bouncy_castle_for_java:1.71:*:*:*:*:*:*:* (*Confidence*:Highest) suppress
- cpe:2.3:a:openpgp:openpgp:1.71:*:*:*:*:*:*:* (*Confidence*:Low) suppress

Published Vulnerabilities

CVE-2023-33202 suppress

Bouncy Castle for Java before 1.73 contains a potential Denial of Service (DoS) issue within the Bouncy Castle org.bouncycastle.openssl.PEMParser class. This class parses OpenSSL PEM encoded streams containing X.509 certificates, PKCS8 encoded keys, and PKCS7 objects. Parsing a file that has crafted ASN.1 data through the PEMParser causes an OutOfMemoryError, which can enable a denial of service attack.

CWE-400 Uncontrolled Resource Consumption

CVSSv3:

- Base Score: MEDIUM (5.5)
- Vector: CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:N/I:N/A:H

References:

- - https://bouncycastle.org
- - https://github.com/bcgit/bc-java/wiki/CVE-2023-33202

Vulnerable Software & Versions:

- cpe:2.3:a:bouncycastle:bouncy_castle_for_java:*:*:*:*:*:*:*:* versions up to (excluding) 1.73

</SAST

코드 품질 검사 → {01}

코드 취약점 검사 → {02}

대시보드 결과 보고서 → {03}

</SAST -System Design

{01} VPC 네트워크 관리

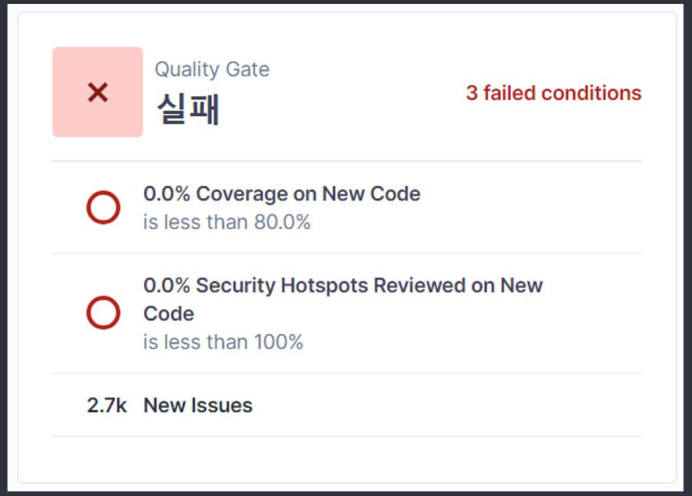
{02} EFS 파일시스템 생성/관리

{03} RDS 데이터베이스 생성/관리

{04} KMS Key 생성/관리

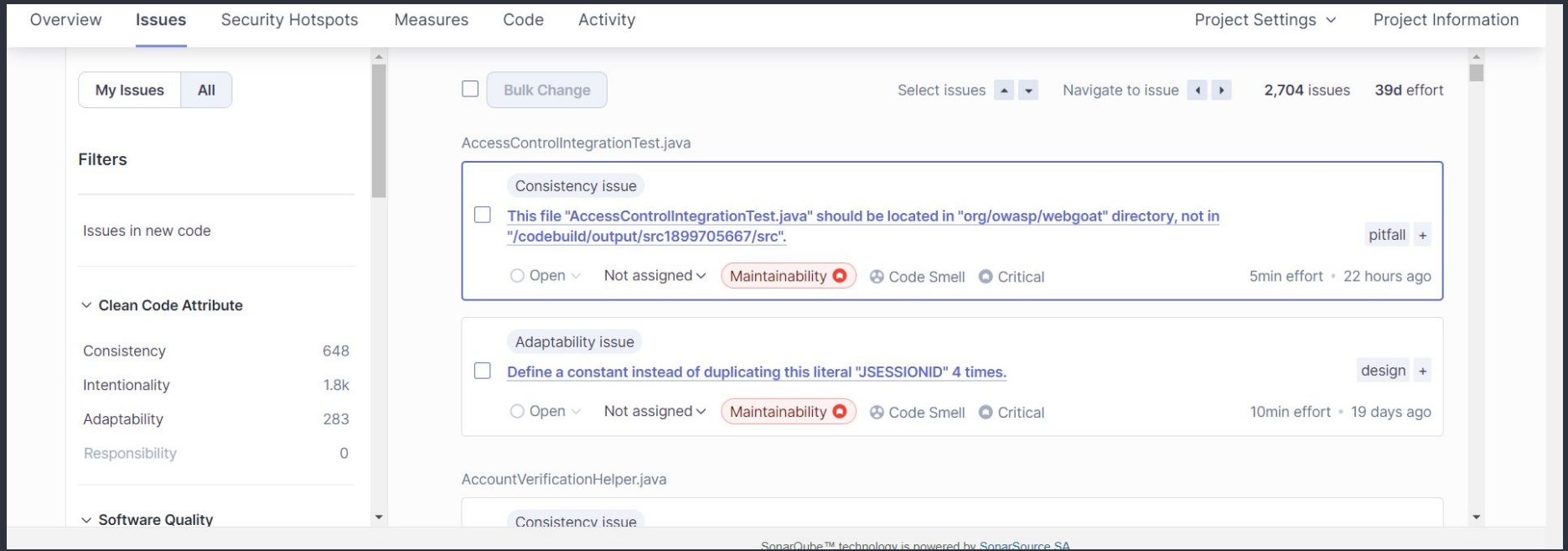
{05} ECS,ECR 컨테이너 저장, 실행

</SAST -Report


Quality Gate
실패
3 failed conditions
0.0% Coverage on New Code
is less than 80.0%
0.0% Security Hotspots Reviewed on New Code
is less than 100%
2.7k New Issues


신뢰성
255 새로운 버그
E
유지 관리성
2.4k 새로운 코드 냄새
A
보안
0 새로운 취약점
A
보안 검토
84 새로운 보안 핫스팟
E

</SAST -Report


Overview
Issues
Security Hotspots
Measures
Code
Activity
Project Settings
Project Information
My Issues
All
Filters
Issues in new code
Clean Code Attribute
Consistency 648
Intentionality 1.8k
Adaptability 283
Responsibility 0
Software Quality
Bulk Change
Select issues
Navigate to issue
2,704 issues
39d effort
AccessControlIntegrationTest.java
Consistency issue
This file "AccessControlIntegrationTest.java" should be located in "org/owasp/webgoat" directory, not in "/codebuild/output/src1899705667/src".
pitfall
Open
Not assigned
Maintainability
Code Smell
Critical
5min effort • 22 hours ago
Adaptability issue
Define a constant instead of duplicating this literal "JSESSIONID" 4 times.
design
Open
Not assigned
Maintainability
Code Smell
Critical
10min effort • 19 days ago
AccountVerificationHelper.java
Consistency issue
SonarQube™ technology is powered by SonarSource SA

</DAST - ZAP

웹 동적 테스트 → {01}

코드 검토 → {02}

취약점 분석 → {03}

보고서 생성 → {04}

</DAST

{Systems Manager}

EC2 원격 실행

{ZAP}

웹 보안 스캐너

{CodeBuild}

빌드 서비스

{EC2}

클라우드 컴퓨팅 서비스

</DAST - Codebuild

Codebuild

AWS Systems Manager

buildspec.yml

buildspec_zap.yml

Codebuild

소스 코드 컴파일 및 단위 테스트 실행
배포 준비 완료된 아티팩트 생성

</DAST - Systems Manager(SSM)

SSM

AWS Systems Manager

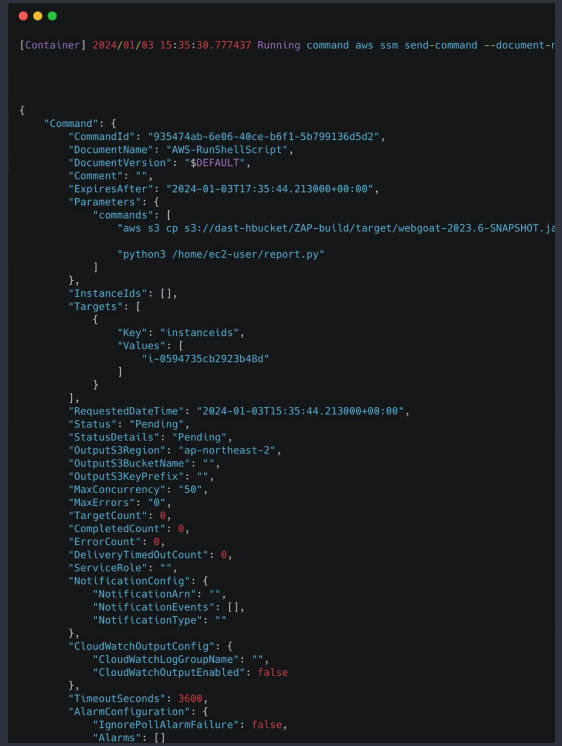
Command

aws ssm send-command

SSM

Key Pair 없이 SSH와 동일한 기능 구현
EC2 원격 접근

</DAST - Systems Manager(SSM) log



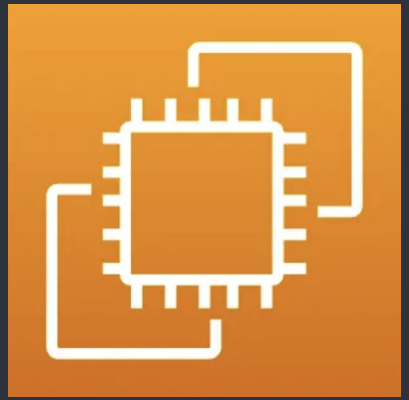
</DAST - EC2

EC2

Amazon Elastic
Compute Cloud

Settings

Key pair, VPC,
Security Group,
EIP, IAM



EC2

키 페어, VPC, 보안그룹, EIP, IAM 설정
ZAP 실행

</DAST - ZAP

ZAP

Zed Attack Proxy

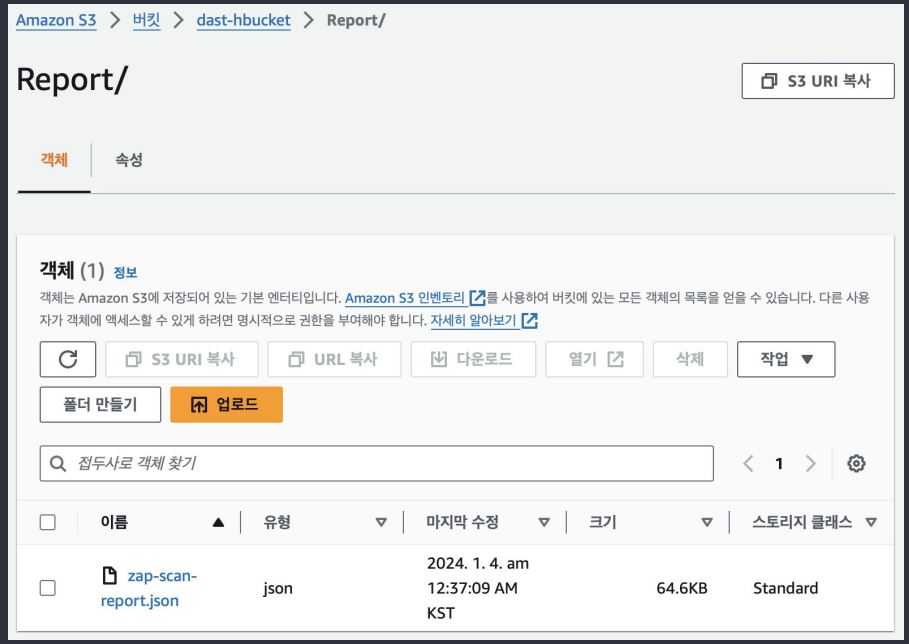
SCAN

Zap scan

ZAP

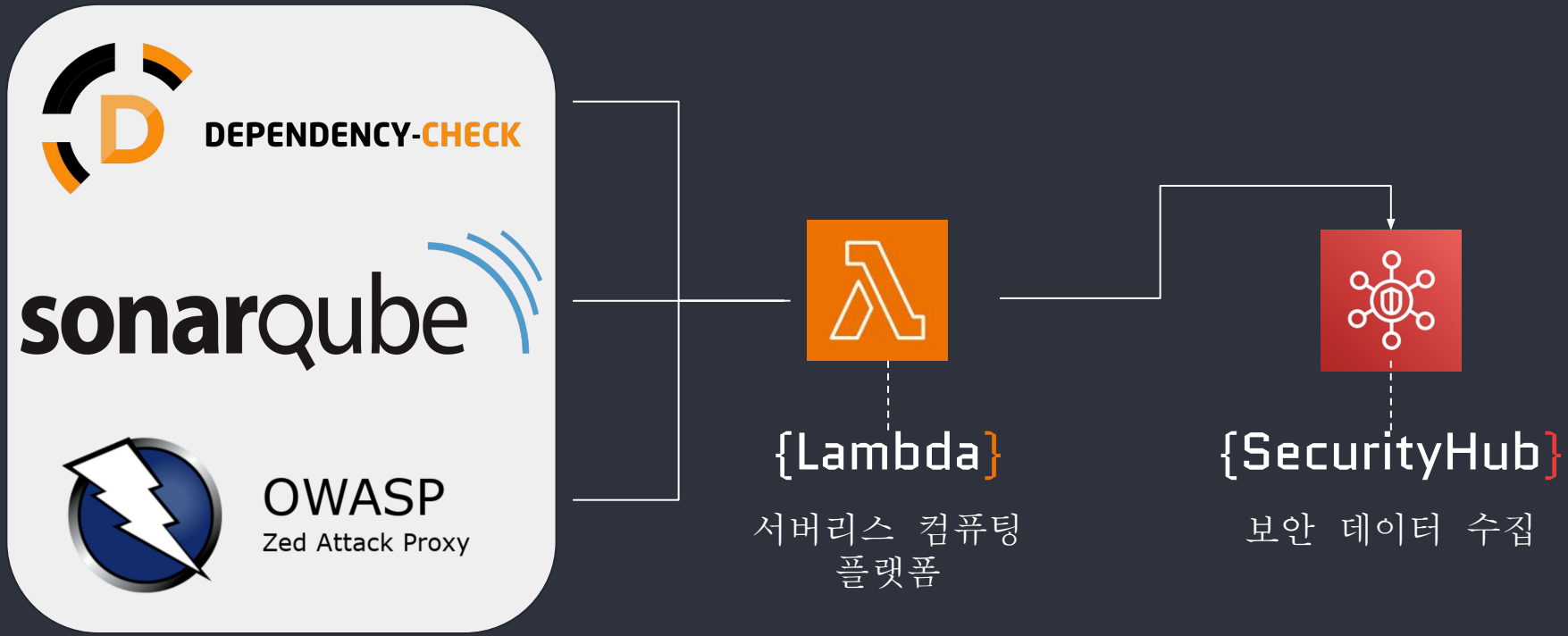
ZAP 스캔 수행
JSON 형식 취약점 보고서 생성

</DAST - ZAP Report




```
{
        "@programName": "ZAP",
        "@version": "2.14.0",
        "@generated": "Thu, 4 Jan 2024 00:36:21",
        "site":[
                {
                        "@name": "http://43.200.16.60:8080",
                        "@host": "43.200.16.60",
                        "@port": "8080",
                        "@ssl": "false",
                        "alerts": [
                                {
                                        "pluginid": "40018",
                                        "alertRef": "40018",
                                        "alert": "SQL Injection",
                                        "name": "SQL Injection",
                                        "riskcode": "3",
                                        "confidence": "2",
                                        "riskdesc": "High (Medium)",
                                        "desc": "<p>SQL injection may be possible.</p>",
                                        "instances":[
                                                {
                                                        "uri": "http://43.200.16.60:8080/WebGoat/register.mvc",
                                                        "method": "POST",
                                                        "param": "agree",
                                                        "attack": "agree OR 1=1 -- ",
                                                        "evidence": "",
                                                        "otherinfo": "The page results were successfully manipulated
using the boolean conditions [agree AND 1=1 -- ] and [agree OR 1=1 -- ]\nThe parameter value being modified was NOT
stripped from the HTML output for the purposes of the comparison\nData was NOT returned for the original
parameter.\nThe vulnerability was detected by successfully retrieving more data than originally returned, by
manipulating the parameter"
                                                }
                                        ],
                                        "count": "1",
                                        "solution": "<p>Do not trust client side input, even if there is client side
validation in place.</p><p>In general, type check all data on the server side.</p><p>If the application uses JDBC, use
PreparedStatement or CallableStatement, with parameters passed by '?'</p><p>If the application uses ASP, use ADO
Command Objects with strong type checking and parameterized queries.</p><p>If database Stored Procedures can be used,
use them.</p><p>Do *not* concatenate strings into queries in the stored procedure, or use 'exec', 'exec immediate', or
equivalent functionality!</p><p>Do not create dynamic SQL queries using simple string concatenation.</p><p>Escape all
data received from the client.</p><p>Apply an 'allow list' of allowed characters, or a 'deny list' of disallowed
characters in user input.</p><p>Apply the principle of least privilege by using the least privileged database user
possible.</p><p>In particular, avoid using the 'sa' or 'db-owner' database users. This does not eliminate SQL
injection, but minimizes its impact.</p><p>Grant the minimum database access that is necessary for the application.
</p>",
                                        "otherinfo": "<p>The page results were successfully manipulated using the
boolean conditions [agree AND 1=1 -- ] and [agree OR 1=1 -- ]</p><p>The parameter value being modified was NOT
stripped from the HTML output for the purposes of the comparison</p><p>Data was NOT returned for the original
parameter.</p><p>The vulnerability was detected by successfully retrieving more data than originally returned, by
```

</Lambda-Security Hub



</Lambda

Lambda

AWS Lambda

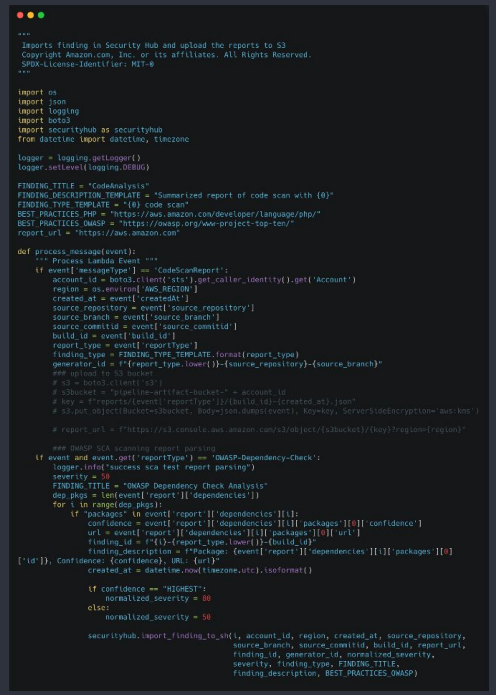
python.py

(1)Import_findings_security_hub.py
(2)Security_hub.py

Lambda

생성된 취약점 보고서를 시큐리티 허브에서
사용하는 포맷인 ASFF로 변환하여 전달

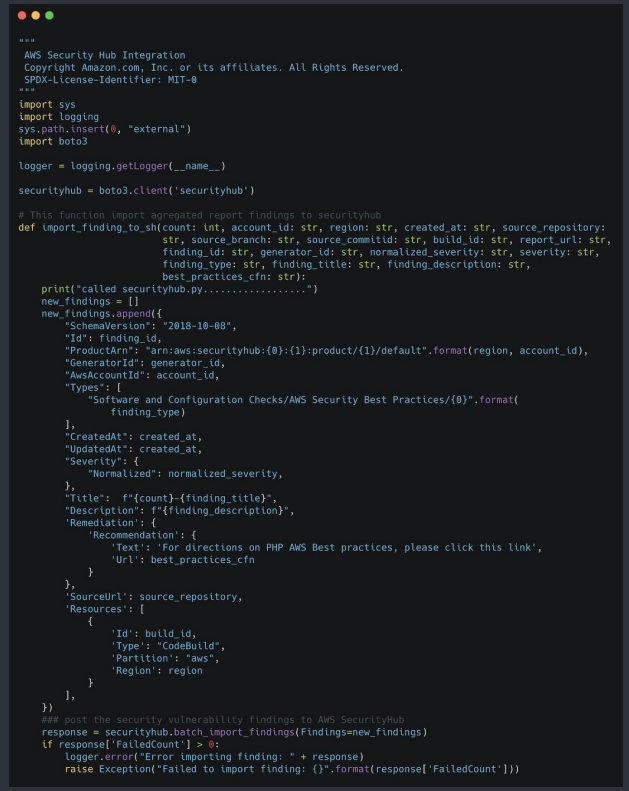
</Lambda - import_findings_security_hub.py



```
"""
 Imports finding in Security Hub and upload the reports to S3
 Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
 SPDX-License-Identifier: MIT-0
"""

import os
import json
import logging
import boto3
import securityhub as securityhub
from datetime import datetime, timezone

logger = logging.getLogger()
logger.setLevel(logging.DEBUG)

FINDING_TITLE = "CodeAnalysis"
FINDING_DESCRIPTION_TEMPLATE = "Summarized report of code scan with {0}"
FINDING_TYPE_TEMPLATE = "{0} code scan"
BEST_PRACTICES_PHP = "https://aws.amazon.com/developer/language/php/"
BEST_PRACTICES_OWASP = "https://owasp.org/www-project-top-ten/"
report_url = "https://aws.amazon.com"

def process_message(event):
    """ Process Lambda Event """
    if event['messageType'] == 'CodeScanReport':
        account_id = boto3.client('sts').get_caller_identity().get('Account')
        region = os.environ['AWS_REGION']
        created_at = event['createdAt']
        source_repository = event['source_repository']
        source_branch = event['source_branch']
        source_commitId = event['source_commitId']
        build_id = event['build_id']
        report_type = event['reportType']
        finding_type = FINDING_TYPE_TEMPLATE.format(report_type)
        generator_id = f"{report_type.lower()}-{source_repository}-{source_branch}"
        ### upload to S3 bucket
        # s3 = boto3.client('s3')
        # s3bucket = "pipeline-artifact-bucket-" + account_id
        # key = f"reports/{event['reportType']}/{build_id}-{created_at}.json"
        # s3.put_object(Bucket=s3bucket, Body=json.dumps(event), Key=key, ServerSideEncryption='aws:kms')

        # report_url = f"https://s3.console.aws.amazon.com/s3/object/{s3bucket}/{key}?region={region}"

        ### OWASP SCA scanning report parsing
    if event and event.get('reportType') == 'OWASP-Dependency-Check':
        logger.info('success sca test report parsing')
        severity = 50
        FINDING_TITLE = "OWASP Dependency Check Analysis"
        dep_pkgs = len(event['report']['dependencies'])
        for i in range(dep_pkgs):
            if "packages" in event['report']['dependencies'][i]:
                confidence = event['report']['dependencies'][i]['packages'][0]['confidence']
                url = event['report']['dependencies'][i]['packages'][0]['url']
                finding_id = f"{i}-{report_type.lower()}-{build_id}"
                finding_description = f"Package: {event['report']['dependencies'][i]['packages'][0]
['id']}, Confidence: {confidence}, URL: {url}"
                created_at = datetime.now(timezone.utc).isoformat()

                if confidence == "HIGHEST":
                    normalized_severity = 80
                else:
                    normalized_severity = 50

                securityhub.import_finding_to_sh(i, account_id, region, created_at, source_repository,
                                                 source_branch, source_commitId, build_id, report_url,
                                                 finding_id, generator_id, normalized_severity,
                                                 severity, finding_type, FINDING_TITLE,
                                                 finding_description, BEST_PRACTICES_OWASP)
```

```
        ### SonarQube SAST scanning report parsing
        if event['reportType'] == 'SONAR-QUBE':
            severity = 50
            FINDING_TITLE = "SonarQube StaticCode Analysis"
            report_count = event['report']['total']
            for i in range(report_count):
                finding_id = f"{i}-{report_type.lower()}-{source_repository}-{source_branch}-
                {build_id}"
                finding_description = f"{event['report']['issues'][i]['type']}-{event['report']
['issues'][i]['message']}-{i}, component: {event['report']['issues'][i]['component']}"
                created_at = datetime.now(timezone.utc).isoformat()
                report_severity = event['report']['issues'][i]['severity']
                ### find the vulnerability severity level
                if report_severity == 'MAJOR':
                    normalized_severity = 70
                elif report_severity == 'BLOCKER':
                    normalized_severity = 90
                elif report_severity == 'CRITICAL':
                    normalized_severity = 90
                else:
                    normalized_severity= 20
                ### Calling SecurityHub function to post the findings
                securityhub.import_finding_to_sh(i, account_id, region, created_at, source_repository,
                                                 source_branch, source_commitId, build_id, report_url,
                                                 finding_id, generator_id, normalized_severity,
                                                 severity, finding_type, FINDING_TITLE,
                                                 finding_description, BEST_PRACTICES_OWASP)

        ### OWASP Zap SAST scanning report parsing
        elif event['reportType'] == 'OWASP-Zap':
            severity = 50
            FINDING_TITLE = "OWASP ZAP DynamicCode Analysis"
            alert_ct = event['report']['site'][0]['alerts']
            alert_count = len(alert_ct)
            for alertno in range(alert_count):
                risk_desc = event['report']['site'][0]['alerts'][alertno]['riskdesc']
                riskletters = risk_desc[0:3]
                ### find the vulnerability severity level
                if riskletters == 'Hig':
                    normalized_severity = 70
                elif riskletters == 'Med':
                    normalized_severity = 50
                elif riskletters == 'Low' or riskletters == 'Inf':
                    normalized_severity = 30
                else:
                    normalized_severity = 90
                instances = len(event['report']['site'][0]['alerts'][alertno]['instances'])
                finding_description = f"{alertno}-Vulerability{event['report']['site'][0]['alerts']
[alertno]['alert']}-Total occurances of this issue:{instances}"
                finding_id = f"{alertno}-{report_type.lower()}-{build_id}"
                created_at = datetime.now(timezone.utc).isoformat()
                ### Calling SecurityHub function to post the findings
                securityhub.import_finding_to_sh(alertno, account_id, region, created_at,
                                                 source_repository, source_branch, source_commitId,
                                                 build_id, report_url, finding_id, generator_id,
                                                 normalized_severity, severity, finding_type,
                                                 FINDING_TITLE, finding_description, BEST_PRACTICES_OWASP)

        else:
            print("Invalid report type was provided")
    else:
        logger.error("Report type not supported:")

def lambda_handler(event, context):
    """ Lambda entrypoint """
    try:
        logger.info("Starting function")
        return process_message(event)
    except Exception as error:
        logger.error("Error {}".format(error))
        raise

  }
  return go(f, seed, [])
}
```

</Lambda - security_hub.py



```
"""
 AWS Security Hub Integration
 Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
 SPDX-License-Identifier: MIT-0
"""
import sys
import logging
sys.path.insert(0, "external")
import boto3

logger = logging.getLogger(__name__)

securityhub = boto3.client('securityhub')

# This function import agregated report findings to securityhub
def import_finding_to_sh(count: int, account_id: str, region: str, created_at: str, source_repository:
                         str, source_branch: str, source_commitid: str, build_id: str, report_url: str,
                         finding_id: str, generator_id: str, normalized_severity: str, severity: str,
                         finding_type: str, finding_title: str, finding_description: str,
                         best_practices_cfn: str):
    print("called securityhub.py.................")
    new_findings = []
    new_findings.append({
        "SchemaVersion": "2018-10-08",
        "Id": finding_id,
        "ProductArn": "arn:aws:securityhub:{0}:{1}:product/{1}/default".format(region, account_id),
        "GeneratorId": generator_id,
        "AwsAccountId": account_id,
        "Types": [
            "Software and Configuration Checks/AWS Security Best Practices/{0}".format(
                finding_type)
        ],
        "CreatedAt": created_at,
        "UpdatedAt": created_at,
        "Severity": {
            "Normalized": normalized_severity,
        },
        "Title":  f"{count}-{finding_title}",
        "Description": f"{finding_description}",
        'Remediation': {
            'Recommendation': {
                'Text': 'For directions on PHP AWS Best practices, please click this link',
                'Url': best_practices_cfn
            }
        },
        'SourceUrl': source_repository,
        'Resources': [
            {
                'Id': build_id,
                'Type': "CodeBuild",
                'Partition': "aws",
                'Region': region
            }
        ],
    })
    ### post the security vulnerability findings to AWS SecurityHub
    response = securityhub.batch_import_findings(Findings=new_findings)
    if response['FailedCount'] > 0:
        logger.error("Error importing finding: " + response)
        raise Exception("Failed to import finding: {}".format(response['FailedCount']))
```

</Security Hub

SecurityHub

AWS Security Hub

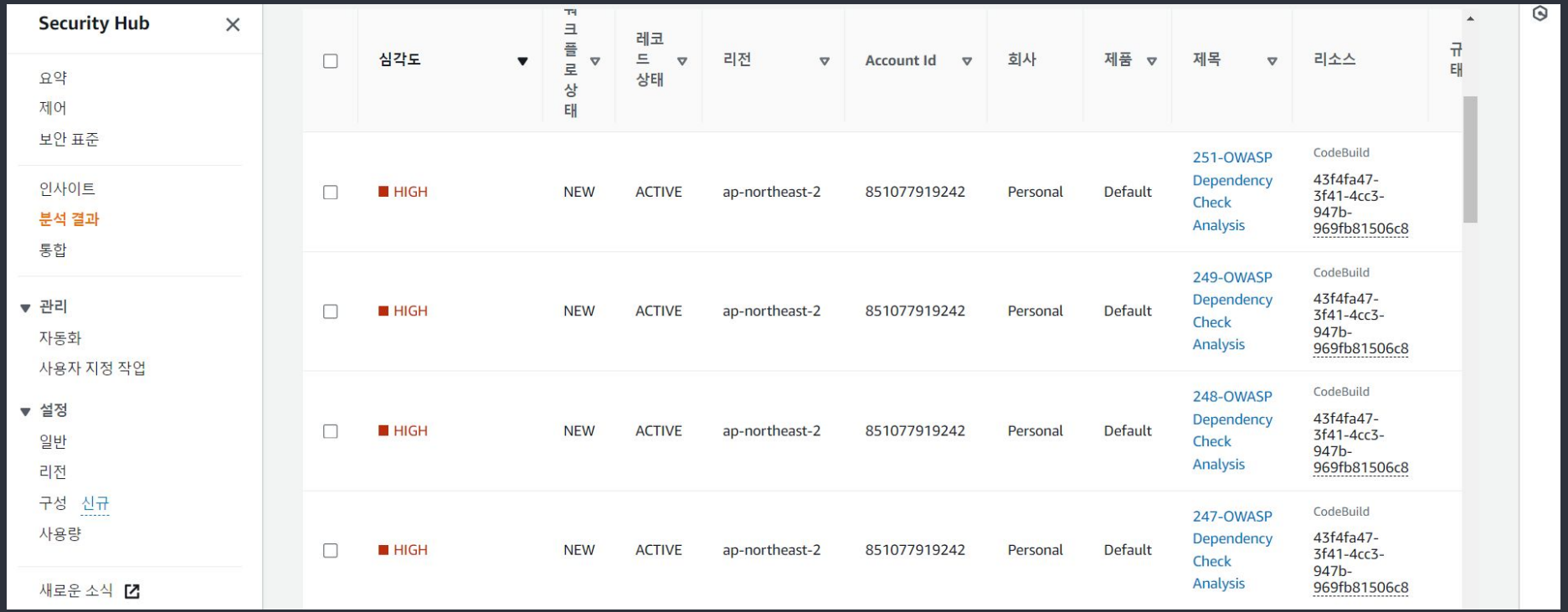
Severity

Critical, High, Medium, Low

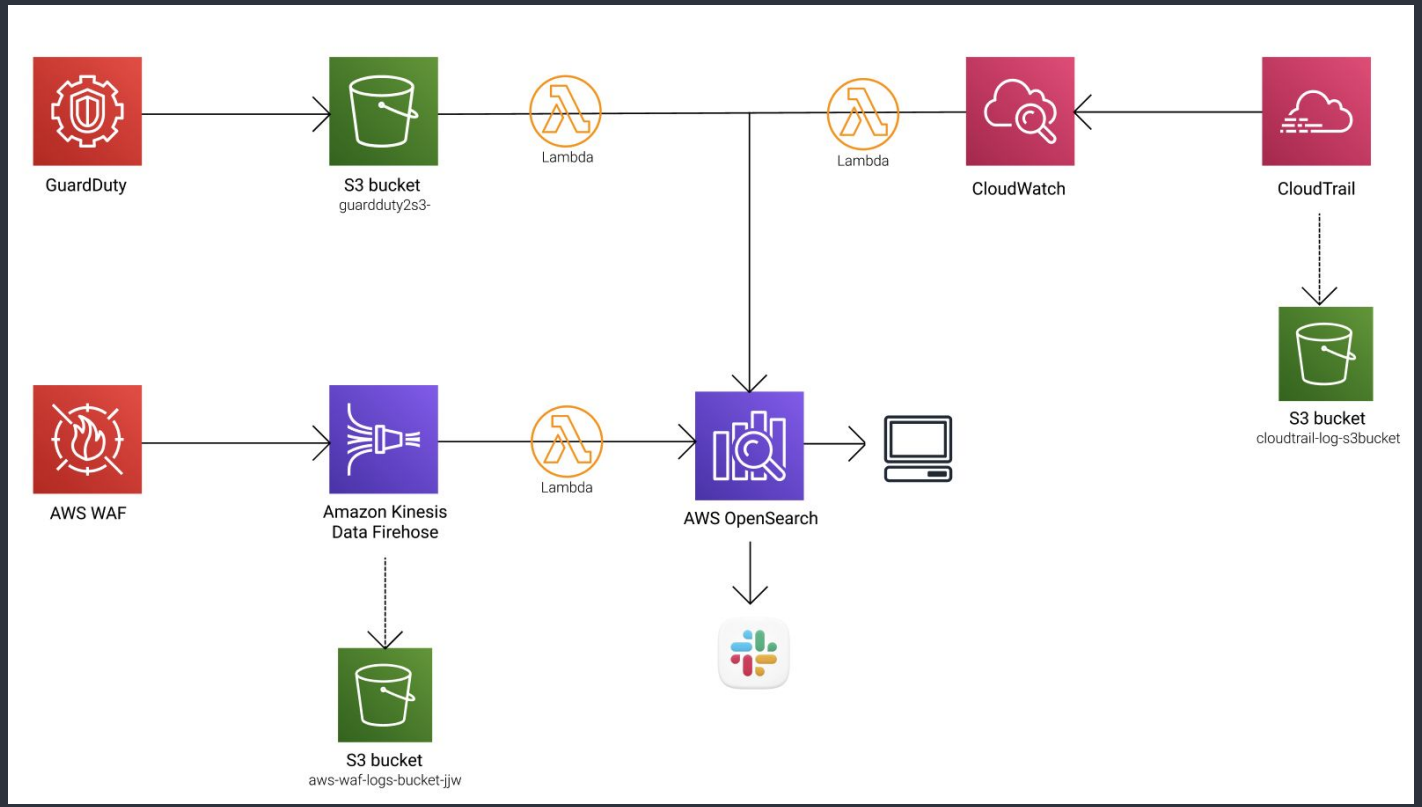
Security Hub

보안 결과 수집하고 우선 순위 결정

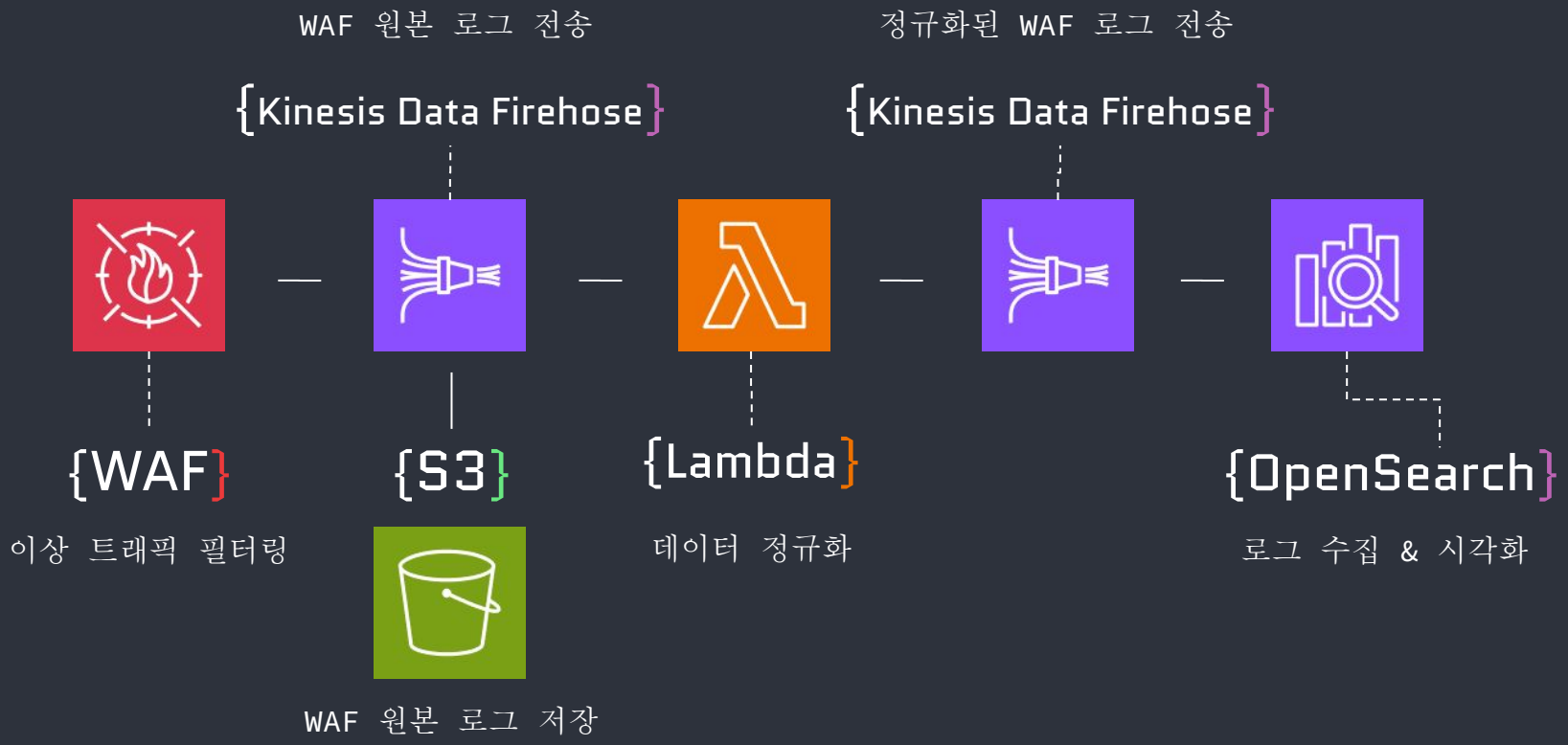
</Security Hub



</Monitoring Arch



</WAF


WAF 원본 로그 전송
{Kinesis Data Firehose}
정규화된 WAF 로그 전송
{Kinesis Data Firehose}
{WAF}
이상 트래픽 필터링
{S3}
WAF 원본 로그 저장
{Lambda}
데이터 정규화
{OpenSearch}
로그 수집 & 시각화

</CloudTrail

{Cloud Trail}

AWS API 호출 기록 및
AWS 계정 활동 로그 수집

{Cloud Watch}

CloudTrail 로그
실시간 수집 및 전송

{Lambda}

데이터 정규화

{OpenSearch}

로그 수집 & 시각화

{S3}

CloudTrail 로그 저장

</GuardDuty

{Guard Duty}

AWS 활동, 네트워크등 감시, 악의적인 행동 탐지 및 보고

{S3}

GuardDuty 로그 저장

{Lambda}

데이터 정규화

{OpenSearch}

로그 수집 & 시각화

</OpenSearch

WAF, CloudTrail,
GuardDuty 로그 수집

통합 로그 수집

{01}

로그를 표, 그래프
등으로 가시화 가능

시각화

{02}

가시화된 데이터를
대시보드로 구성 가능

대시보드

{03}

수집된 로그를 기반으로
이상 로그 발생시 알림 전송

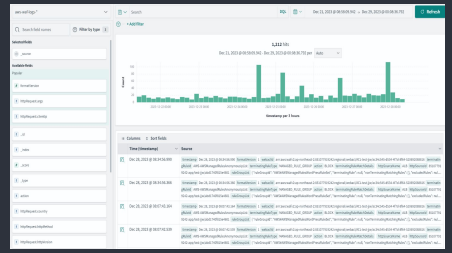
이상 탐지 알림

{04}

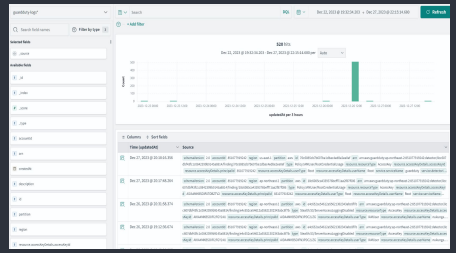
</OpenSearch

통합 로그 수집

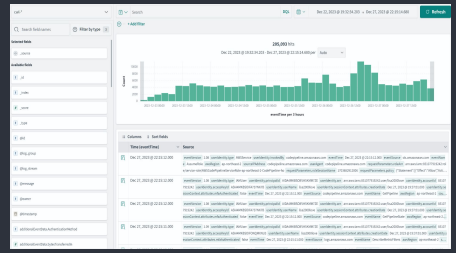
WAF



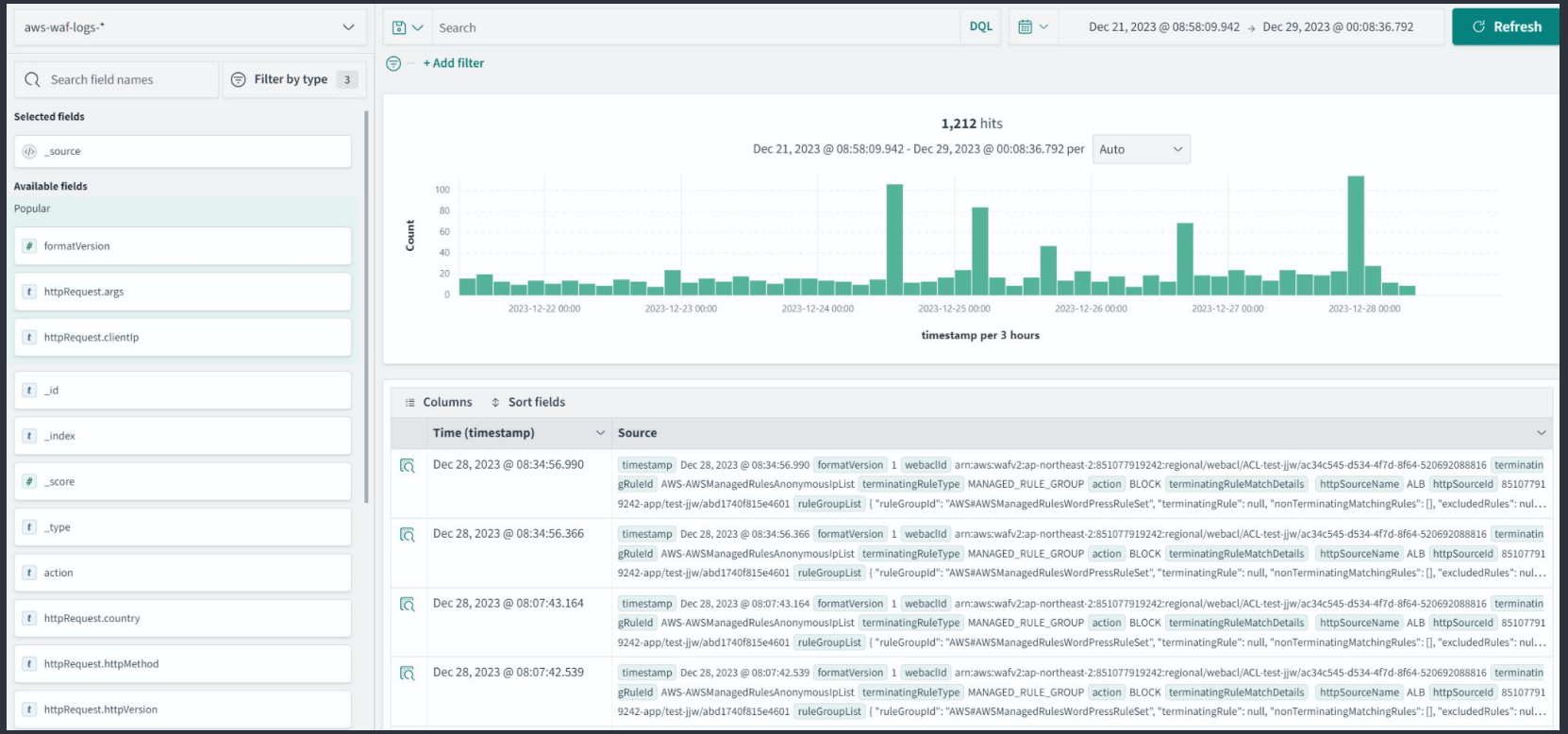
GuardDuty



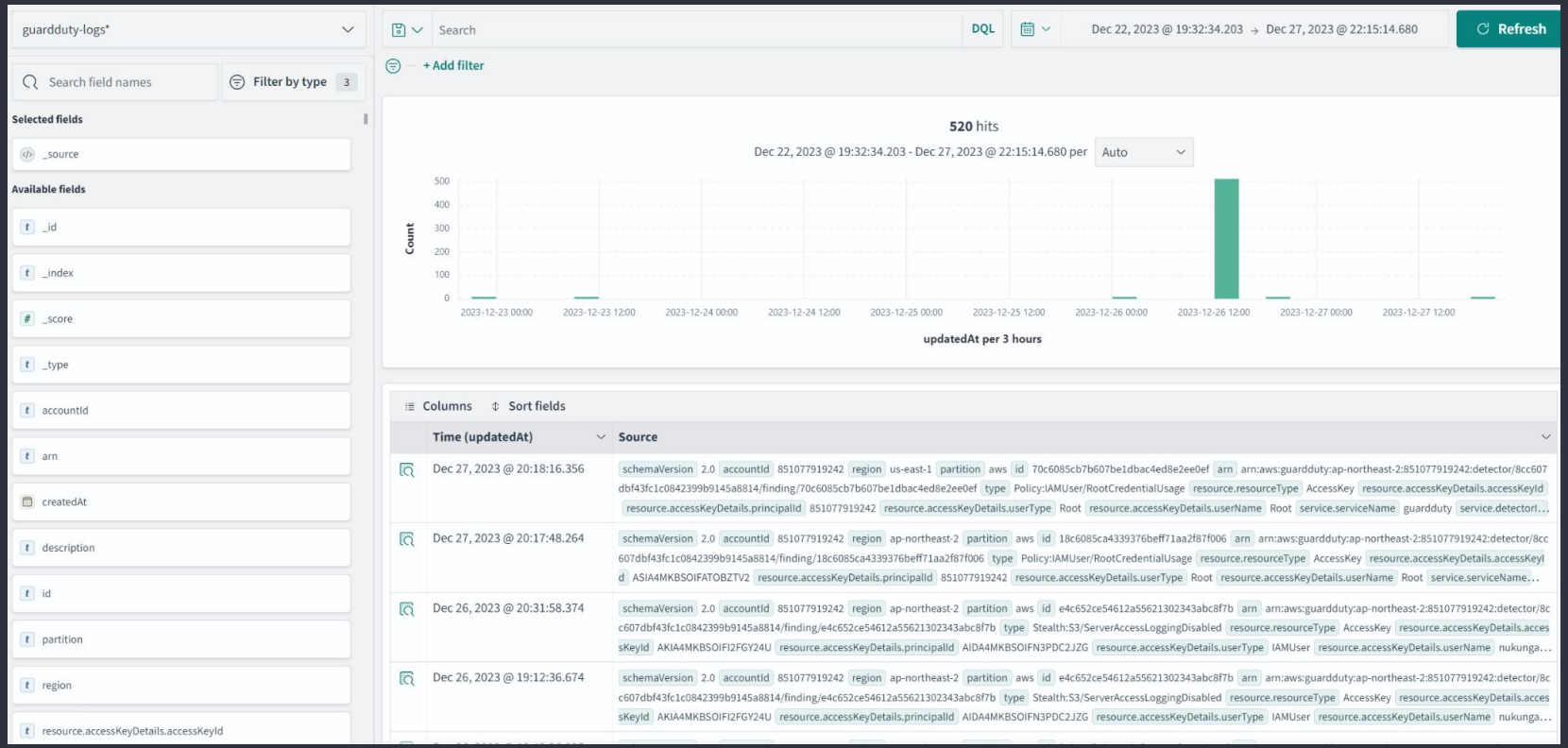
CloudTrail



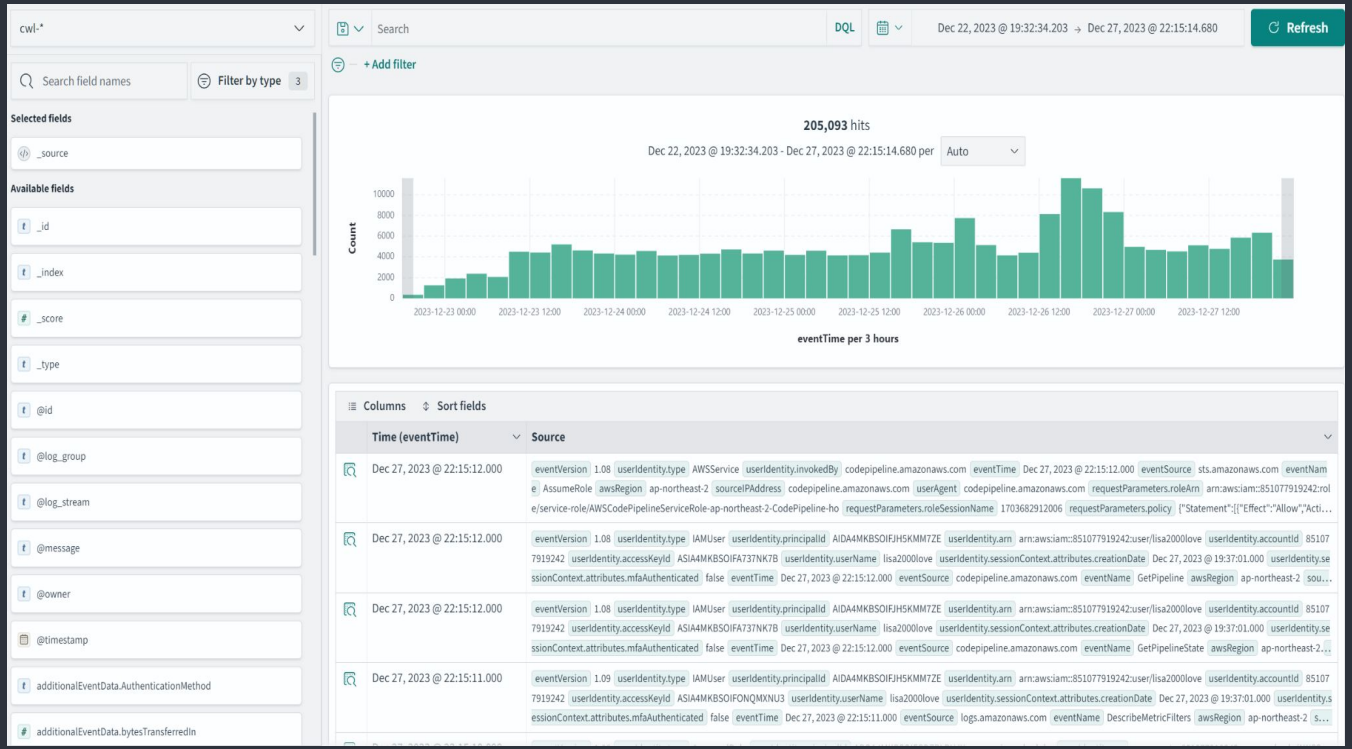
</OpenSearch - Discover



</OpenSearch - Discover



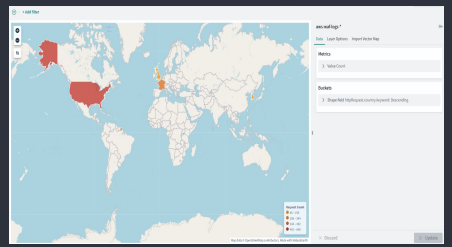
</OpenSearch - Discover



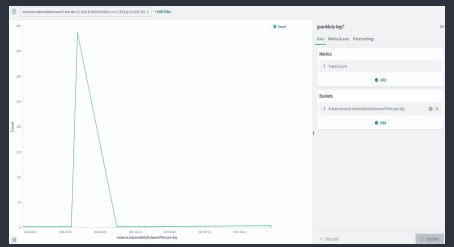
</OpenSearch

시각화

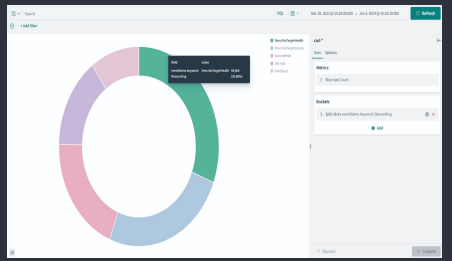
WAF



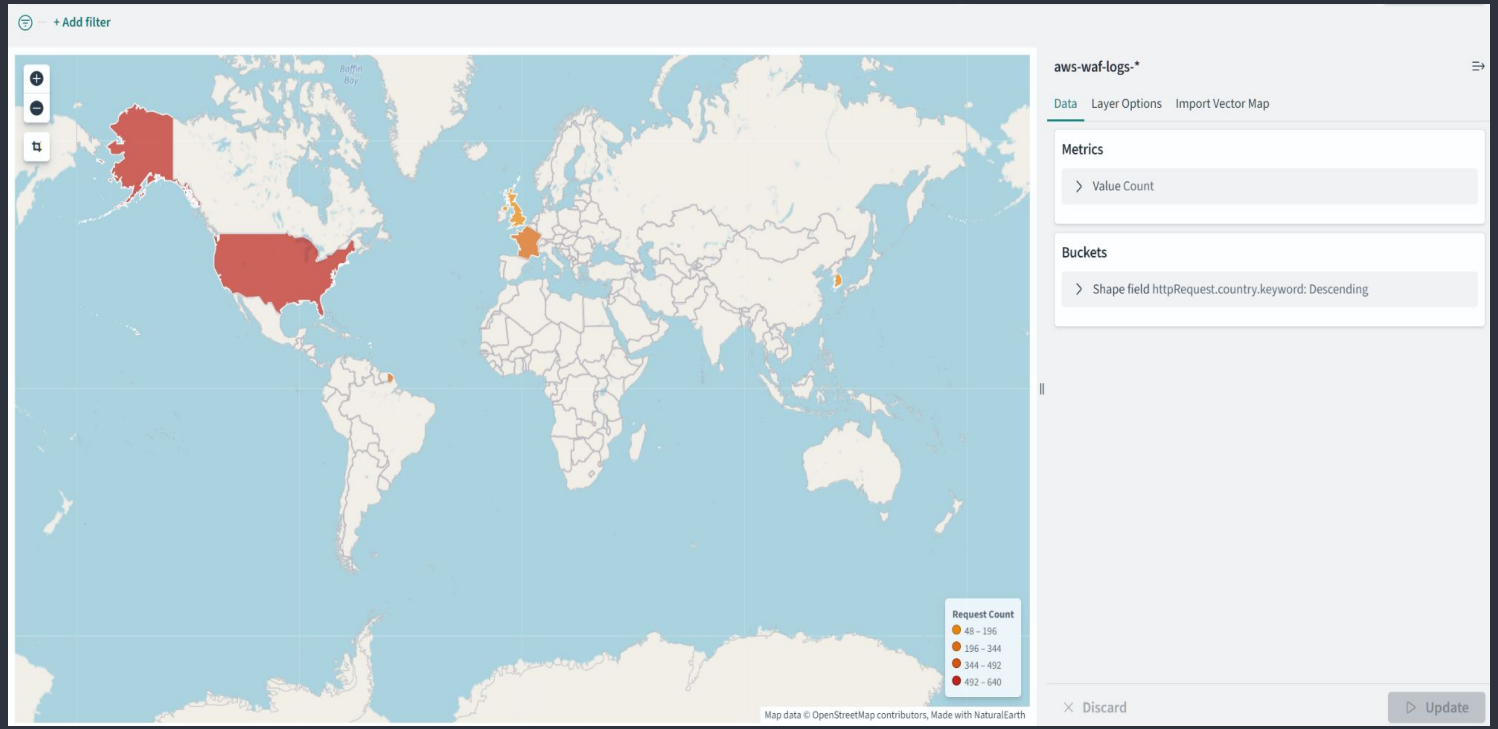
GuardDuty



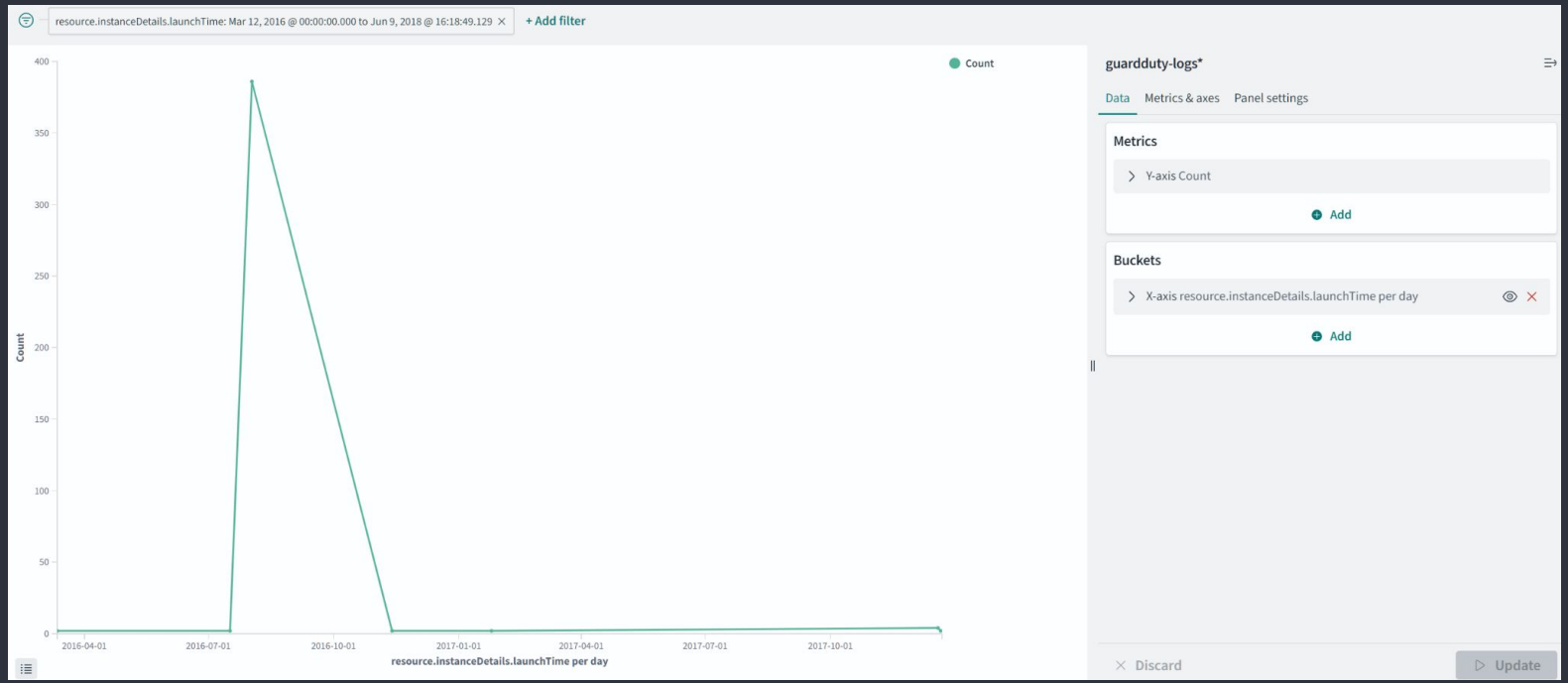
CloudTrail



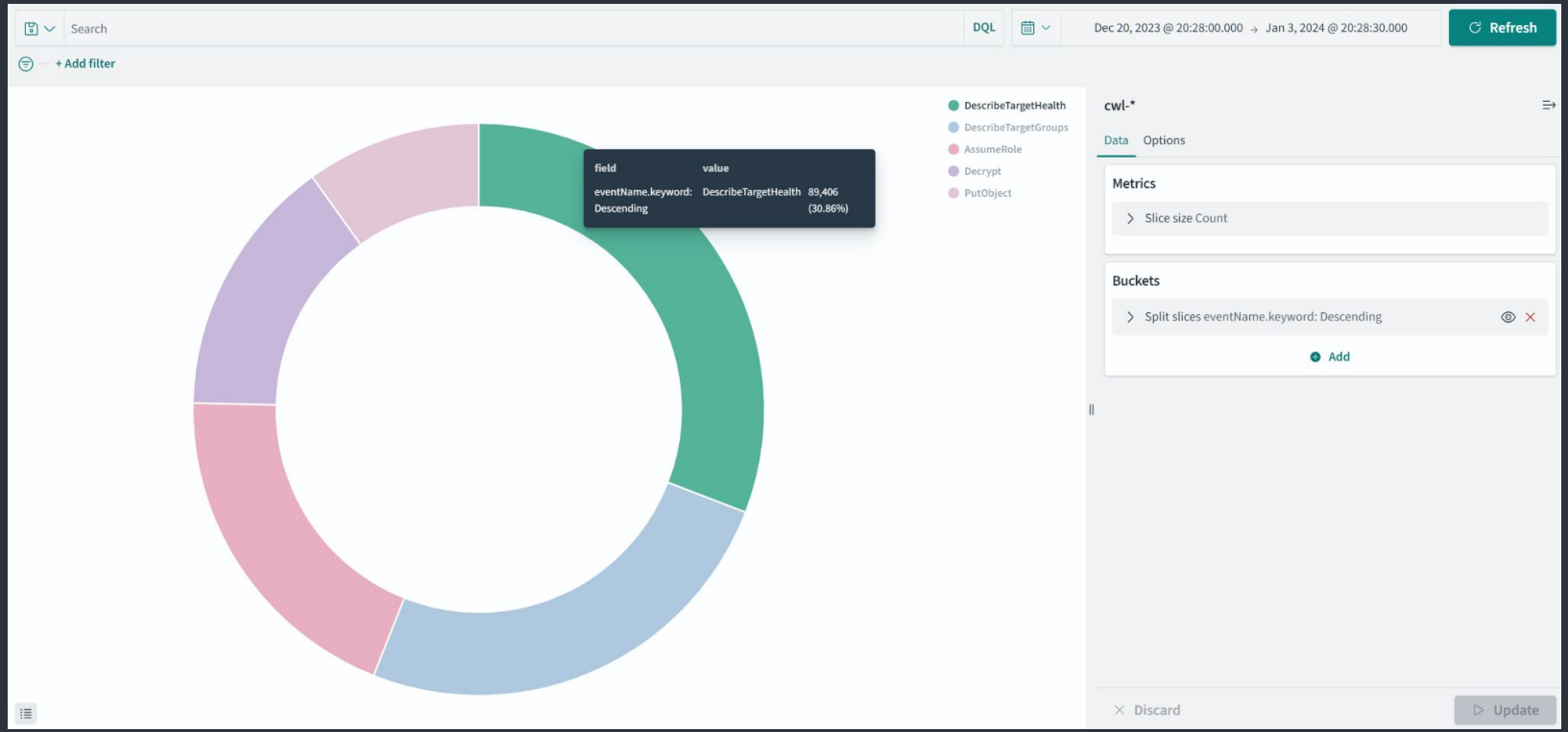
</OpenSearch


+ Add filter
aws-waf-logs-*
Data
Layer Options
Import Vector Map
Metrics
Value Count
Buckets
Shape field httpRequest.country.keyword: Descending
Request Count
48 – 196
196 – 344
344 – 492
492 – 640
Map data © OpenStreetMap contributors, Made with NaturalEarth
Discard
Update

</OpenSearch


resource.instanceDetails.launchTime: Mar 12, 2016 @ 00:00:00.000 to Jun 9, 2018 @ 16:18:49.129
+ Add filter
Count
400
350
300
250
200
150
100
50
0
Count
2016-04-01
2016-07-01
2016-10-01
2017-01-01
2017-04-01
2017-07-01
2017-10-01
resource.instanceDetails.launchTime per day
guardduty-logs*
Data
Metrics & axes
Panel settings
Metrics
Y-axis Count
Add
Buckets
X-axis resource.instanceDetails.launchTime per day
Add
Discard
Update

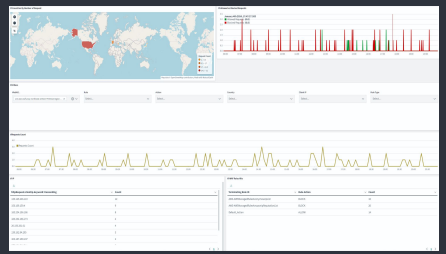
</OpenSearch


Search
DQL
Dec 20, 2023 @ 20:28:00.000 → Jan 3, 2024 @ 20:28:30.000
Refresh
+ Add filter
DescribeTargetHealth
DescribeTargetGroups
AssumeRole
Decrypt
PutObject
field
value
eventName.keyword: Descending
DescribeTargetHealth
89,406 (30.86%)
cwl-*
Data
Options
Metrics
Slice size Count
Buckets
Split slices eventName.keyword: Descending
Add
Discard
Update

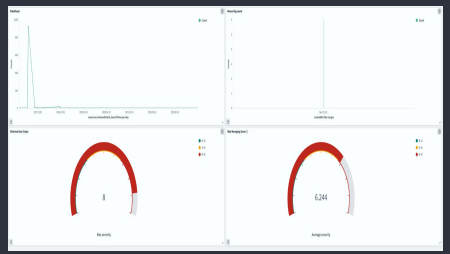
</OpenSearch

대시보드

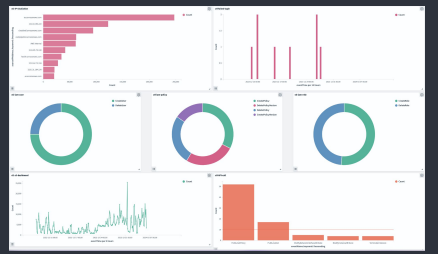
WAF



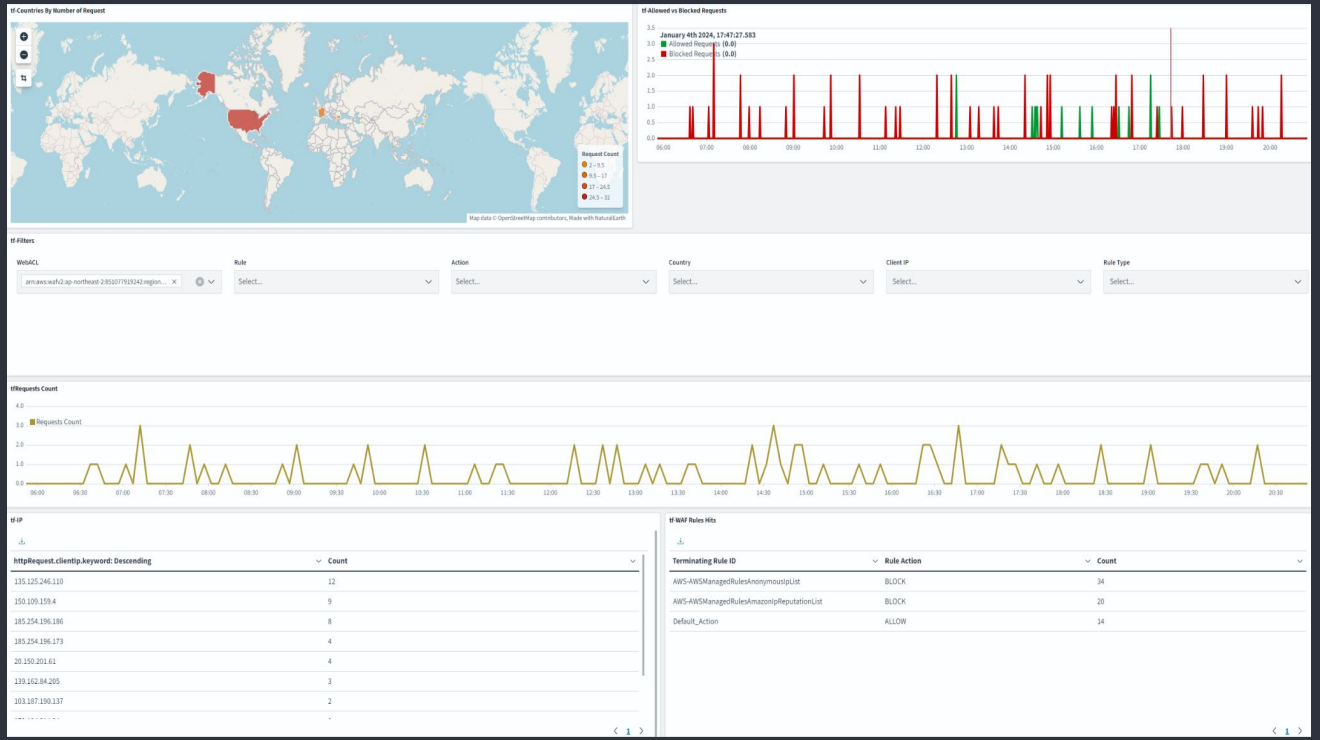
GuardDuty



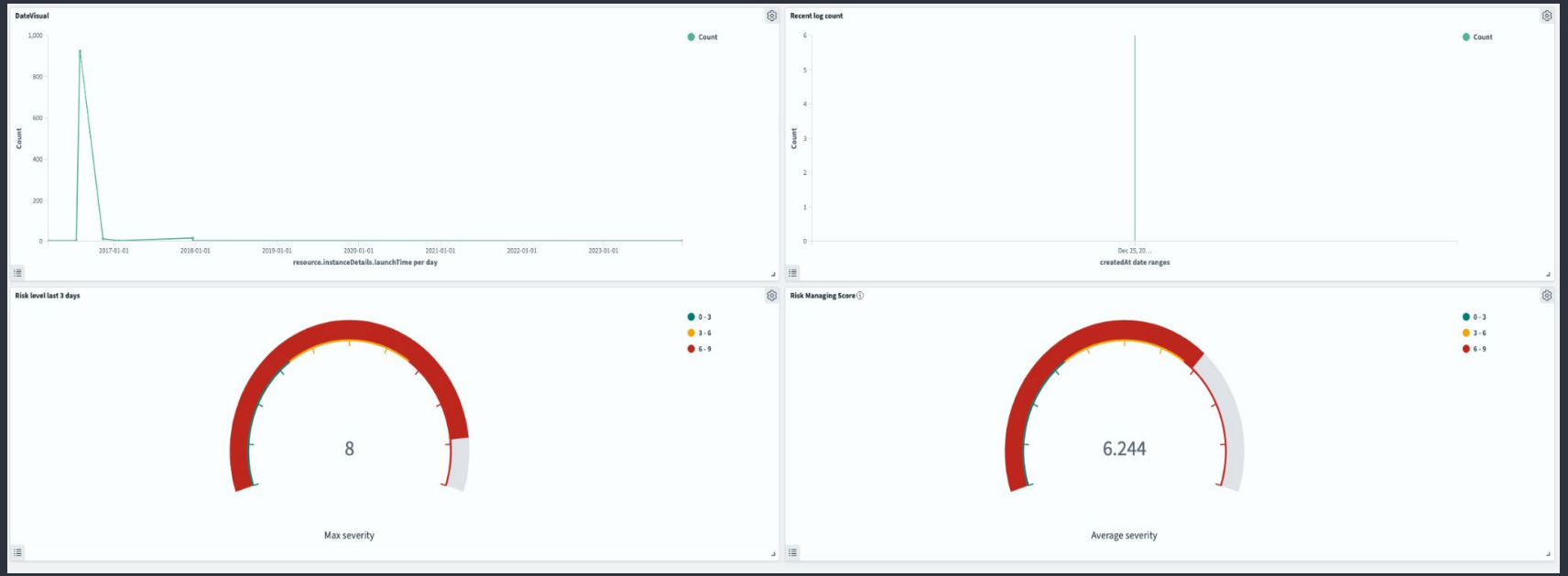
CloudTrail



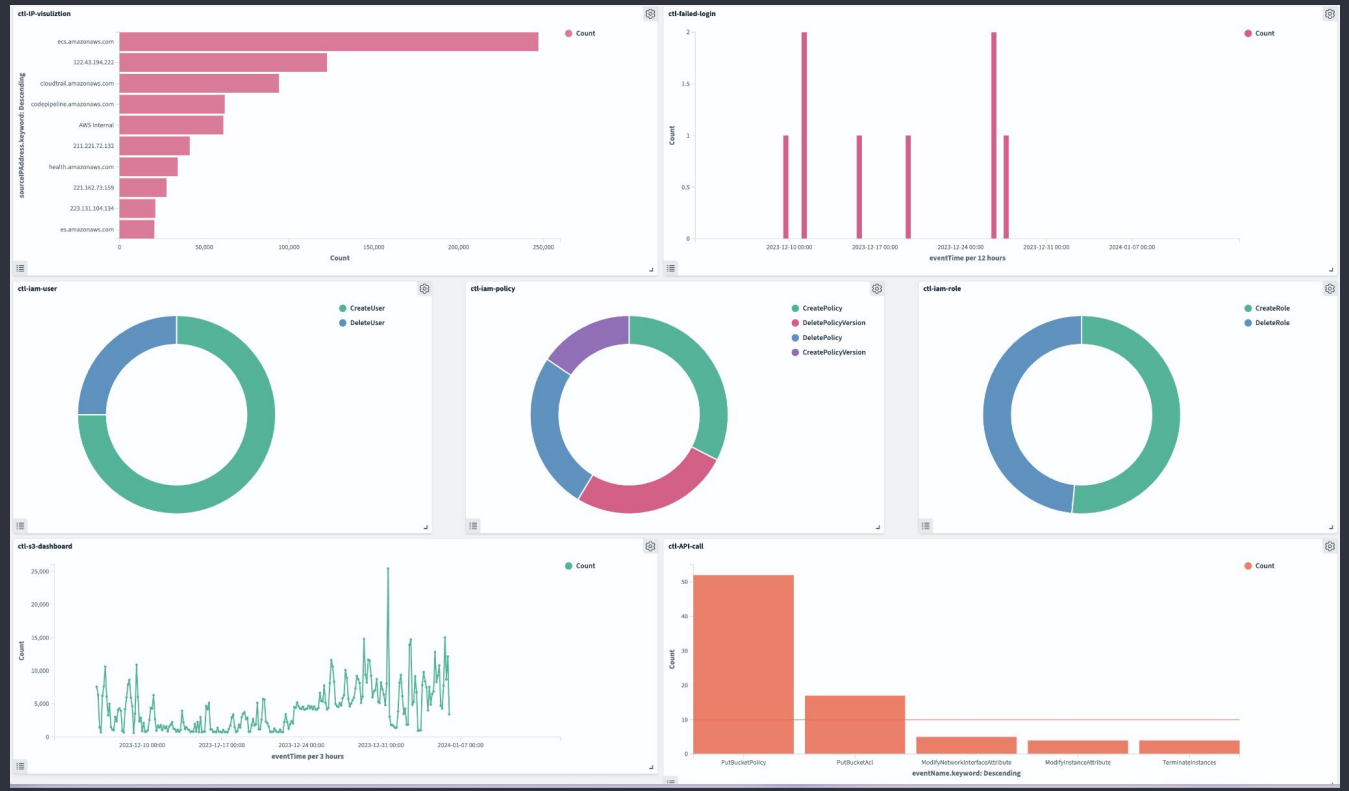
</OpenSearch - WAF



</OpenSearch -GuardDuty



</OpenSearch - CloudTrail



</OpenSearch

이상 탐지 알림

WAF

Alerting WAF action

Monitor **waf-block-trigger** just entered alert status.
- 10분간 BLOCK 요청 2건 발생
- Severity: 1
- Period start: 2024-01-05T09:58:40.493Z
- Period end: 2024-01-05T10:08:40.493Z
- Client IP: 106.251.79.102
- Country: KR
- Rule: AWS-AWSManagedRulesSQLiRuleSet
- Condition Type: SQL_INJECTION

GuardDuty

Alerting GuardDuty action

Monitor **guard-duty** alert just entered alert status.
- GuardDuty Severity: 8
- Type: Execution:Container/MaliciousFile
- CreatedAt : 2023-12-09T06:23:02.865Z
- Period start: 2024-01-05T09:25:51.498Z
- Period end: 2024-01-05T09:40:51.498Z

CloudTrail

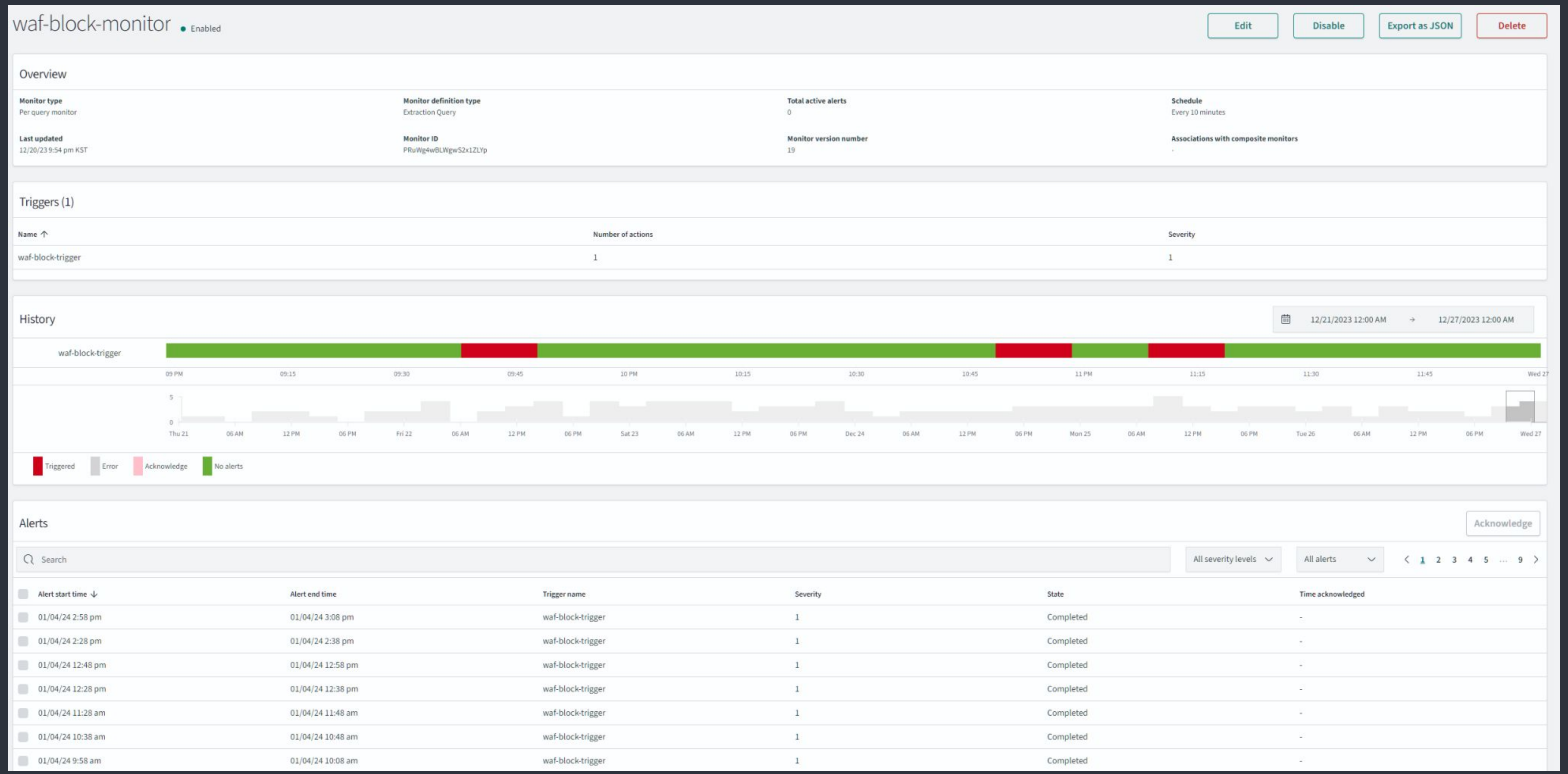
Alerting CloudTrail action

Monitor **ctl-IAM-policy** just entered alert status.
- Severity: 1
- Period start: 2024-01-05T13:11:59.137Z
- Period end: 2024-01-05T13:21:59.137Z
- Event Name: DeletePolicyVersion

</OpenSearch - CloudTrail

1. **NACL** 관련 생성, 수정, 삭제 로그 발생
2. **Security Group** 관련 생성, 수정, 삭제 로그 발생
3. **IAM User, Role, Policy** 생성, 수정, 삭제, 버전 변경
4. **Access Key** 생성
5. **AWS Lambda** 함수 생성, 수정
6. **S3** 권한 변경
7. **SSM** 관련 이벤트 발생

</OpenSearch - WAF

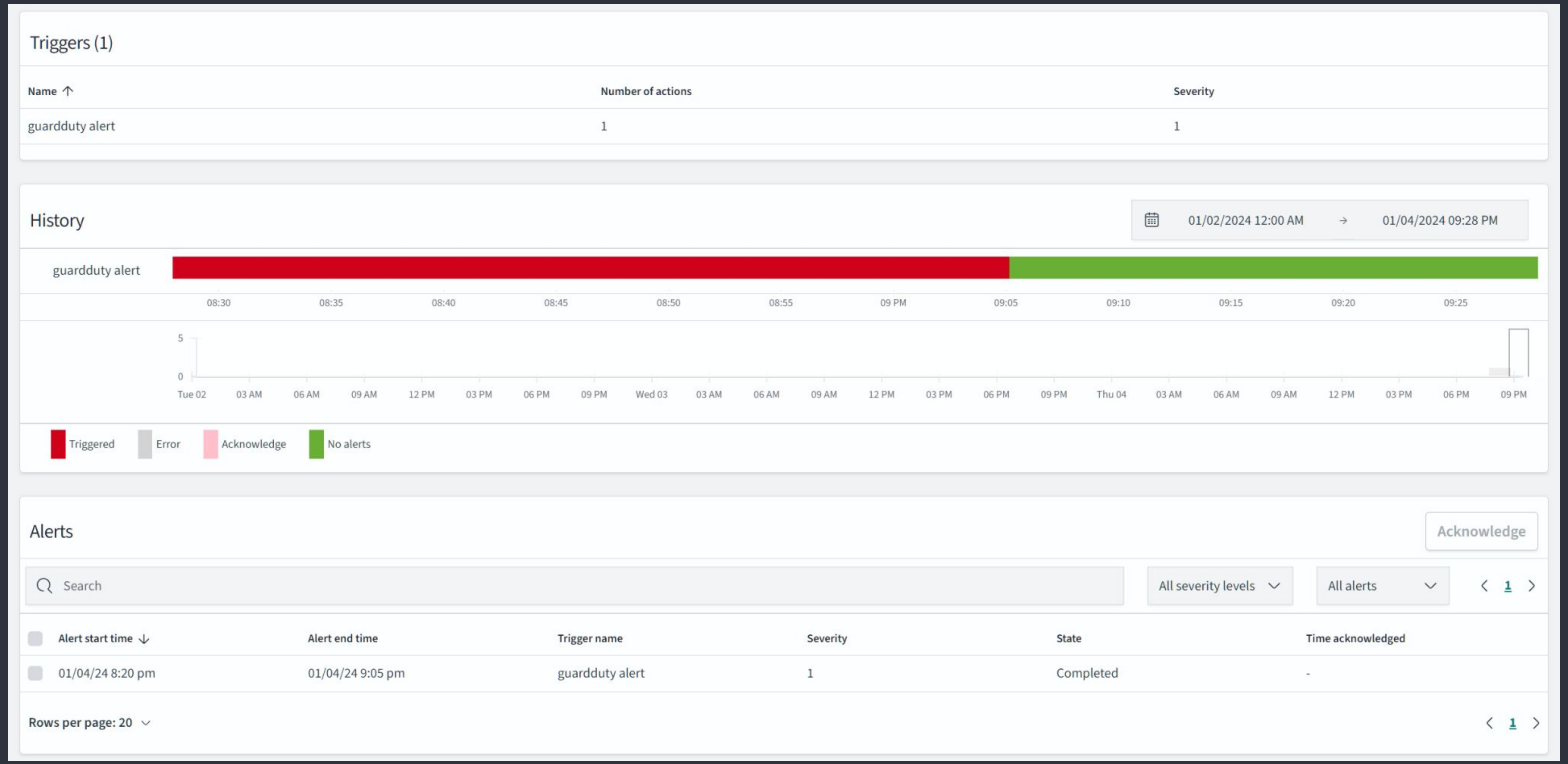


</OpenSearch - WAF

Alerting WAF action

Monitor **waf-block-trigger** just entered alert status.
- 10분간 BLOCK 요청 2건 발생
- Severity: 1
- Period start: 2024-01-05T09:58:40.493Z
- Period end: 2024-01-05T10:08:40.493Z
- Client IP: 106.251.79.102
- Country: KR
- Rule: AWS-AWSManagedRulesSQLiRuleSet
- Condition Type: SQL_INJECTION

</OpenSearch - Guard Duty



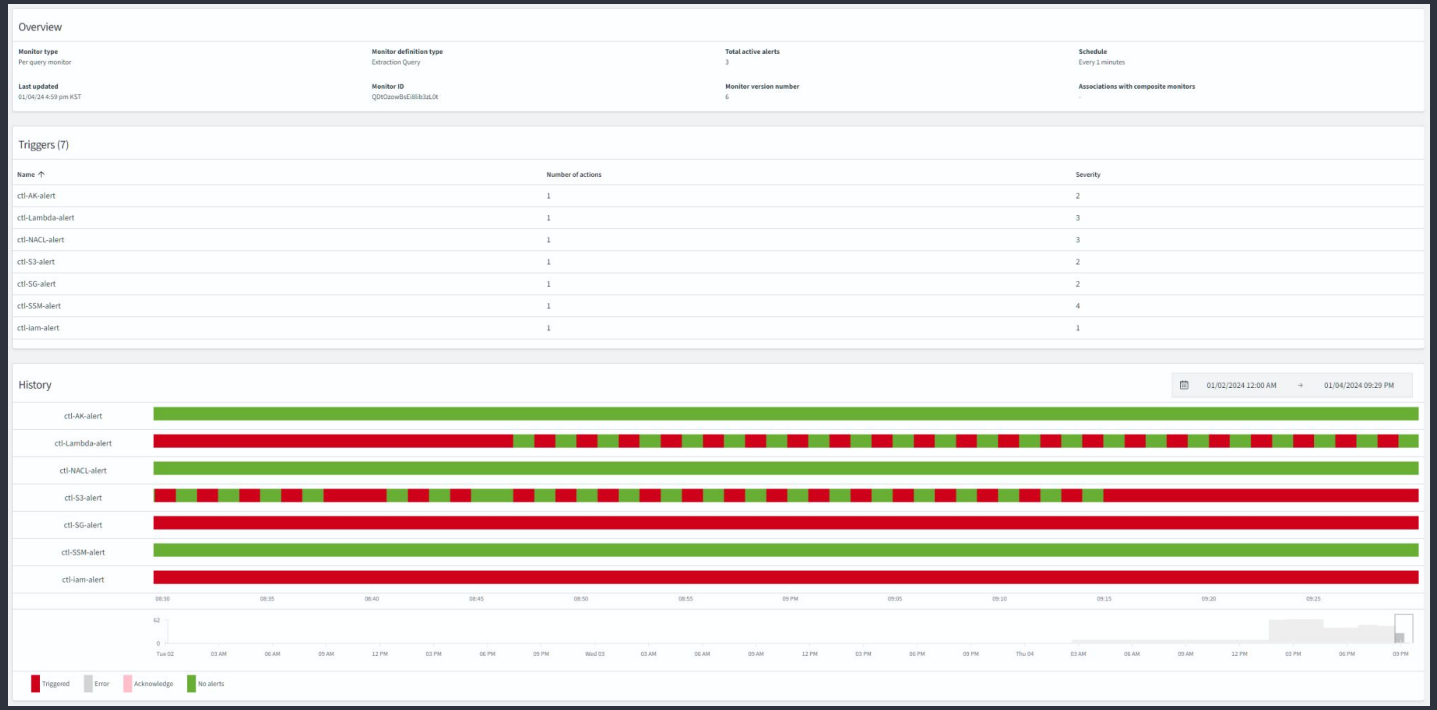
</OpenSearch - Guard Duty

Alerting GuardDuty action

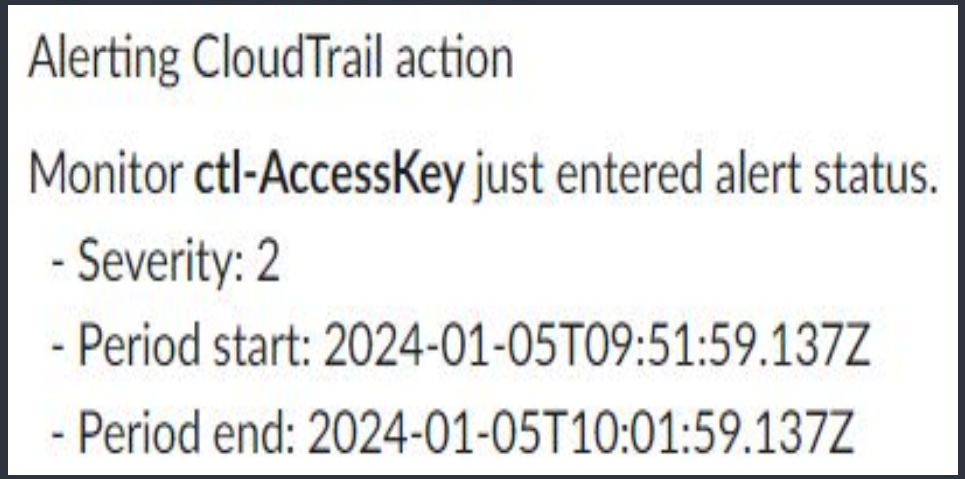
Monitor **guard-duty** alert just entered alert status.

- GuardDuty Severity: 8
- Type: Execution:Container/MaliciousFile
- CreatedAt : 2023-12-09T06:23:02.865Z
- Period start: 2024-01-05T09:25:51.498Z
- Period end: 2024-01-05T09:40:51.498Z

</OpenSearch - CloudTrail



</OpenSearch - CloudTrail



</Infrastructure as Code

{Terraform}

IaC 제작을 위한 Open Source

프로비저닝

Provisioning

자동화

Automation

모듈화

Modularization

IaC: 수동 프로세스가 아닌 코드를 통해 인프라를 관리하고 프로비저닝 하는 것

</Open Source 배포

https://github.com/nanac0n/pipeworkshop.git

</Q&A

Mon	Tue	Wed	Thu	Fri	Sat	Sun
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

</2달 </동안 </모두 </수고하셨습니다