

A W S   P e r s o n a l   P r o j e c t

# 클라우드 와치와 모니터링 방법

—  
안길환

dksskd84@gmail.com



## T a b l e   o f   C o n t e n t s

1. [프로젝트개요](#)
2. [사용기술](#)
3. [아키텍쳐](#)
4. [구현](#)
5. [결과](#)
6. [향후계획](#)
7. [Q&A](#)

# 프로젝트 개요

AWS를 이용하여 프로젝트를 운영하고 있다면 각종 서비스의 리소스를 모니터링 하는 게 귀찮게 느껴질 수 있습니다.

Amazon Web Services(AWS)는 많은 고객들이 이용하고 있습니다.

서비스 품질 유지를 위해 AWS EC2에서 발생하는 로그 파일을 확인하려고 매번 인스턴스에 직접 들어가서 로그파일을 확인하는 작업은 번거롭죠

AWS 리소스를 효과적으로 모니터링할 수 있는 방법이 필요했습니다.



# 프로젝트 개요

CloudWatch 를 사용하면  
인스턴스 및 운영중인 서비스에서  
지표 및 로그를 수집할 수 있습니다.

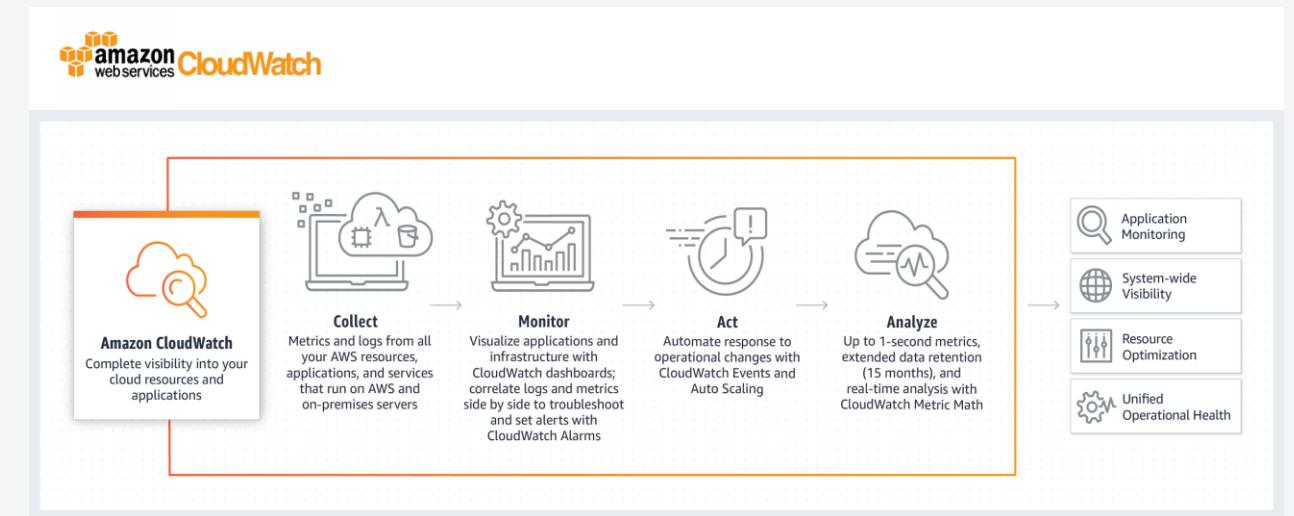
AWS의 놀라운 서비스입니다.

알림 서비스(SNS) 또는 이메일 서비스(SES)와 통합될 수 있습니다.  
CloudWatch 는 사전 정의된 AWS 자원들의 지표 및 로그를 기본 제공합니다.

CloudWatch Agent 를 이용하면  
EC2 인스턴스 또는 온프레미스 서버에서 지표 및 로그 스트림이 수집됩니다.

이렇게 하면 AWS가 기본 제공하는 지표 및 로그를 극복하고  
사용자가 정의한 지표 및 로그 스트림으로 가용성을 높일 수 있으며 서비스 품질향상에  
도움을 줍니다.

본 프로젝트에서는 Amazon CloudWatch agent 를 통해 웹서버를 운영하는데 필요한  
자원들의 지표 및 로그를 수집하고 모니터링하는 방법에 대해 구현해 보도록 하겠습니다.



AWS  
Lambda



AWS  
Athena

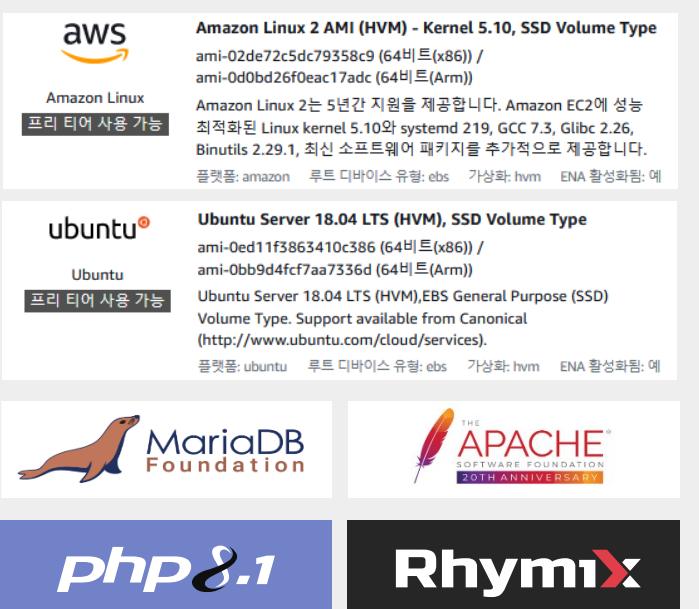
# 사용 기술

클라우드 와치를 통해  
모니터링과 로그수집, 분석을 위한  
인프라를 구축하고 솔루션을 세팅  
하는데 필요한 AWS 서비스와  
웹서비스 개발 플랫폼입니다.



01

## Amazon Web Service Compute



02

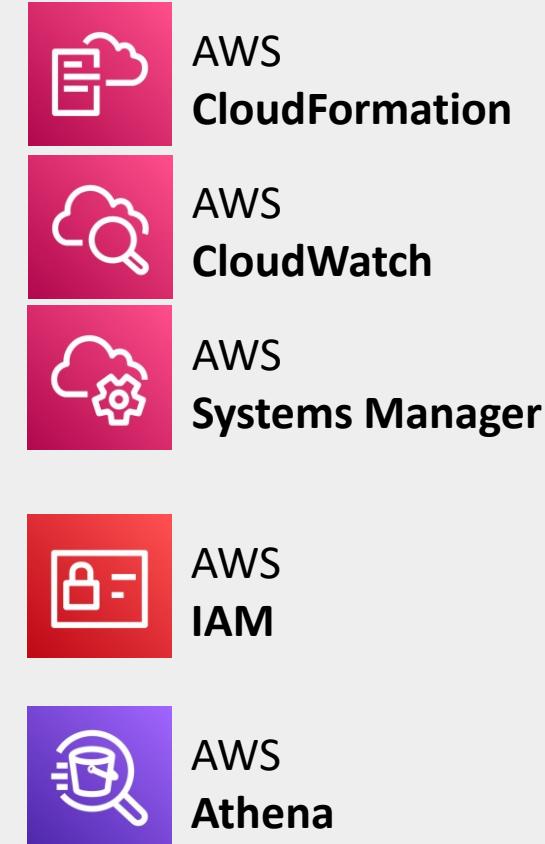
## Amazon Web Service Networking Content Delivery Storage



AWS S3

03

## Amazon Web Service Management & Governance Security, Identity, & Compliance Analytics

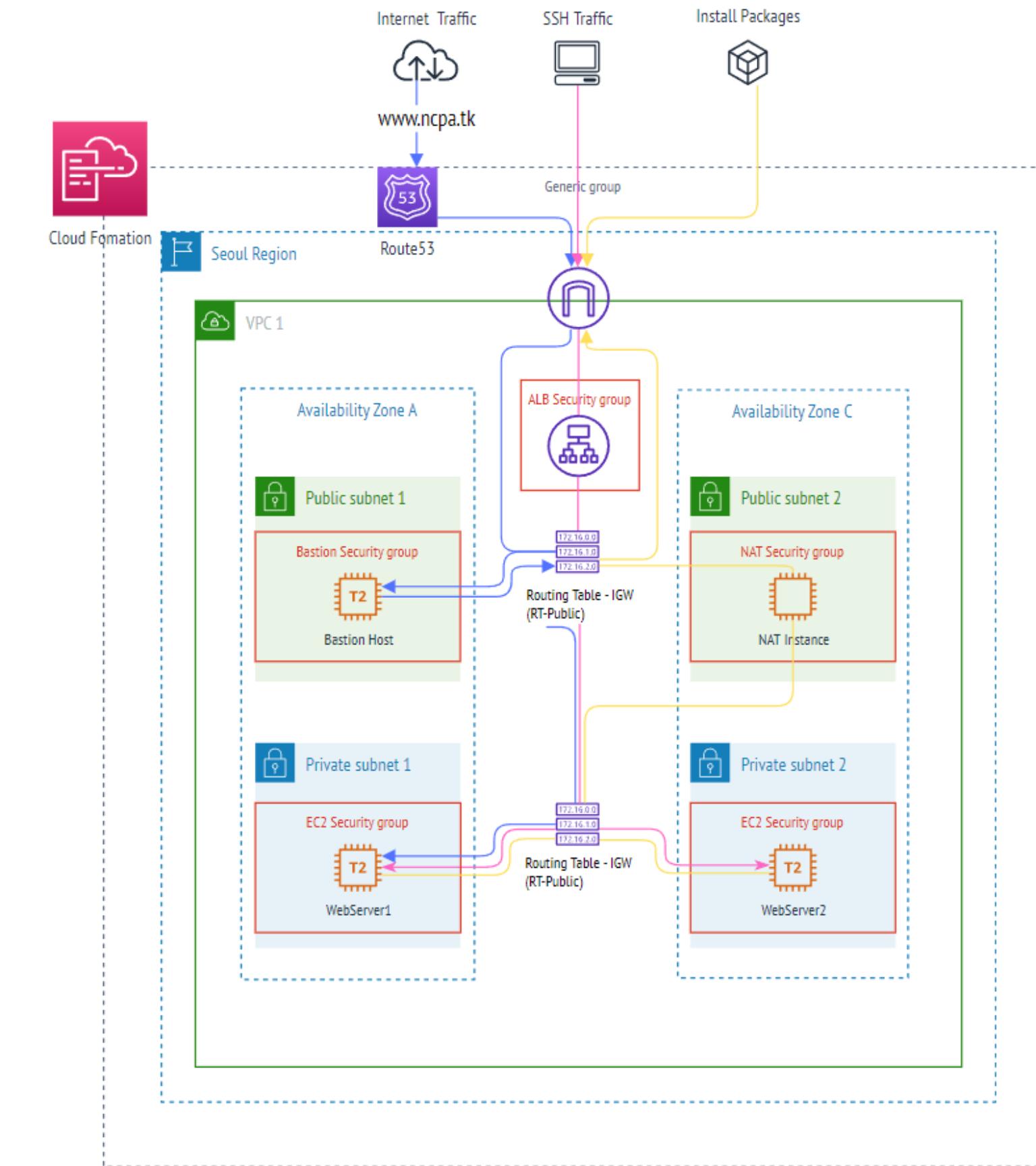


# 아키텍쳐 (Service infra)

AWS를 이용하여  
고객에게 웹 서비스를 제공하고

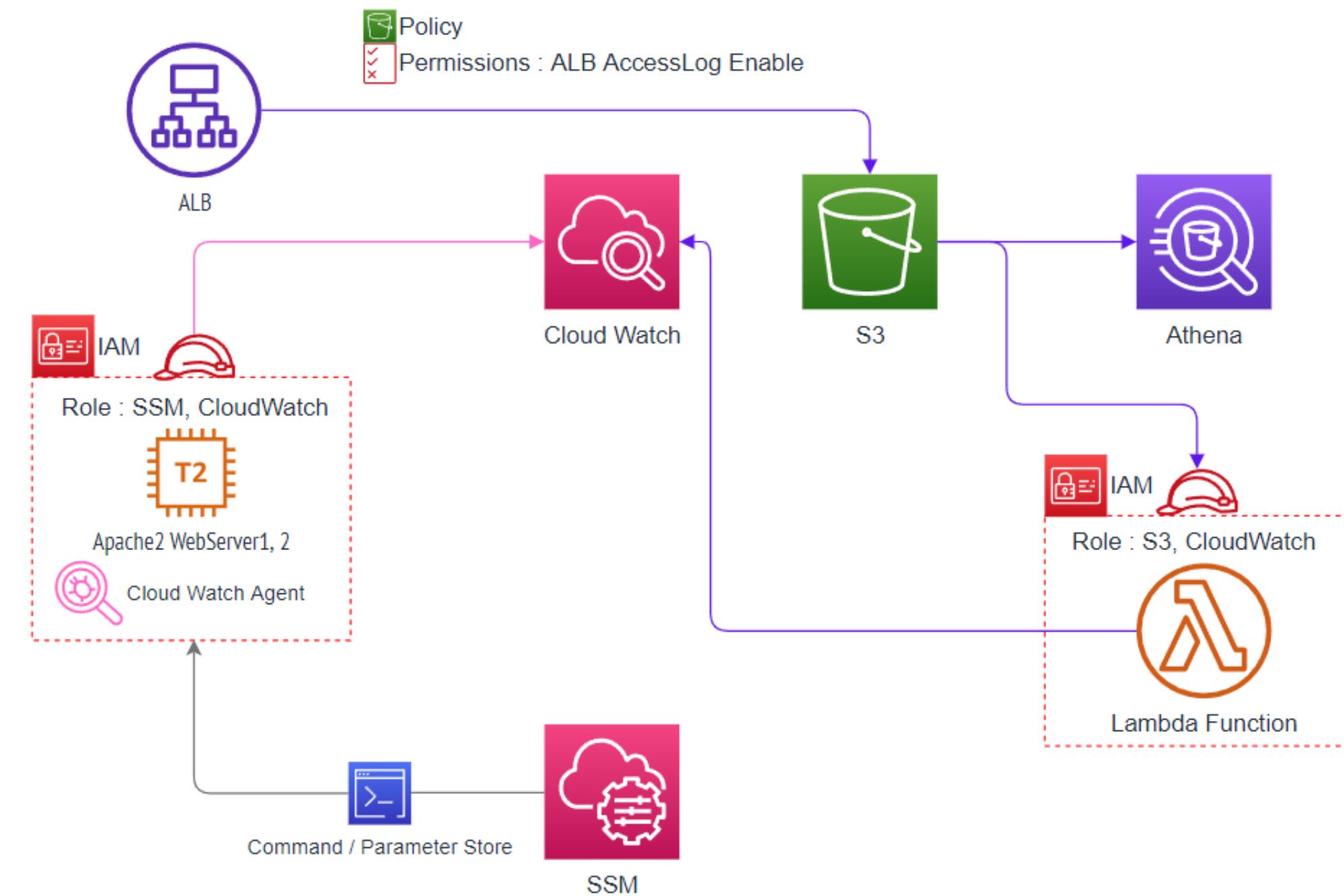
고가용성을 실현하며

각종 서비스의 리소스를 운영하는  
설계 모델입니다.



# 아키텍쳐 (Management & Analytics)

AWS로 운영중인 웹 서비스의 각종 자원들의 리소스를 모니터링하고 지표 수집을 통한 분석에 활용 할 수 있는 설계 모델입니다.



# 구현 Service infra

## 1. CloudFormation YAML Document

자원구축을 위한 매개변수 값 지정

```

1 AWSTemplateFormatVersion: "2010-09-09"
2 Description: bastionhost and natinstance-2AZ-alb-webserver
3
4 Parameters:
5   KeyName:
6     Type: AWS::EC2::KeyPair::KeyName
7
8   Region:
9     Type: String
10    Default: ap-northeast-2
11    AllowedValues:
12      - eu-west-1
13      - eu-west-2
14      - eu-west-3
15      - eu-central-1
16      - us-west-2
17      - us-west-1
18      - us-east-2
19      - us-east-1
20      - ap-south-1
21      - ap-northeast-2
22      - ap-northeast-1
23      - ap-southeast-2
24      - ap-southeast-1
25      - ca-central-1
26      - sa-east-1
27    Description: Enter the AWS region to deploy stack. Default is ap-northeast-2
28

29   VPCCidr:
30     AllowedPattern: '(\d{1,3})\.(.)\{3\}\d{1,3}/\d{1,2}'
31     Type: String
32     Default: 10.0.0.0/16
33     Description: Enter the CIDR for your VPC
34   PublicSubnet1Cidr:
35     AllowedPattern: '(\d{1,3})\.(.)\{3\}\d{1,3}/\d{1,2}'
36     Type: String
37     Default: 10.0.1.0/24
38     Description: Enter the CIDR for your Public Subnet 1
39   PublicSubnet2Cidr:
40     AllowedPattern: '(\d{1,3})\.(.)\{3\}\d{1,3}/\d{1,2}'
41     Type: String
42     Default: 10.0.2.0/24
43     Description: Enter the CIDR for your Public Subnet 2
44   PrivateSubnet1Cidr:
45     AllowedPattern: '(\d{1,3})\.(.)\{3\}\d{1,3}/\d{1,2}'
46     Type: String
47     Default: 10.0.3.0/24
48     Description: Enter the CIDR for your Private Subnet 1
49   PrivateSubnet2Cidr:
50     AllowedPattern: '(\d{1,3})\.(.)\{3\}\d{1,3}/\d{1,2}'
51     Type: String
52     Default: 10.0.4.0/24
53     Description: Enter the CIDR for your Private Subnet 2
54
55   ServerAccess:
56     Description: CIDR IP range allowed to login to the NAT instance
57     Type: String
58     MinLength: 9
59     MaxLength: 18
60     Default: 0.0.0.0/0
61     AllowedPattern: (\d{1,3})\.(.)\{3\}\d{1,3}\.(.)\{3\})\./(\d{1,2})
62     ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x
63

```

# 구현 Service infra

## 1. CloudFormation YAML Document

자원 구성 : VPC, 서브넷, 인터넷게이트웨이

```

64 Resources:
65 # VPC 생성
66 VPC:
67   Type: AWS::EC2::VPC
68   Properties:
69     CidrBlock: !Ref VPCCidr
70     EnableDnsHostnames: true
71   Tags:
72     - Key: Name
73     - Value: VPC-1
74
75 #서브넷 생성
76 PublicSubnet1:
77   Type: AWS::EC2::Subnet
78   Properties:
79     AvailabilityZone: !Sub ${Region}a
80     CidrBlock: !Ref PublicSubnet1Cidr
81     VpcId: !Ref VPC
82   Tags:
83     - Key: Name
84     - Value: PublicSubnet-1
85
86 PublicSubnet2:
87   Type: AWS::EC2::Subnet
88   Properties:
89     AvailabilityZone: !Sub ${Region}c
90     CidrBlock: !Ref PublicSubnet2Cidr
91     VpcId: !Ref VPC
92   Tags:
93     - Key: Name
94     - Value: PublicSubnet-2
95
96 PrivateSubnet1:
97   Type: AWS::EC2::Subnet
98   Properties:
99     AvailabilityZone: !Sub ${Region}a
100    CidrBlock: !Ref PrivateSubnet1Cidr
101    VpcId: !Ref VPC
102  Tags:
103    - Key: Name
104    - Value: PrivateSubnet-1
105
106 PrivateSubnet2:
107   Type: AWS::EC2::Subnet
108   Properties:
109     AvailabilityZone: !Sub ${Region}c
110     CidrBlock: !Ref PrivateSubnet2Cidr
111     VpcId: !Ref VPC
112   Tags:
113     - Key: Name
114     - Value: PrivateSubnet-2
115
116 #인터넷 게이트웨이 설치
117 IG:
118   Type: AWS::EC2::InternetGateway
119   Properties:
120     Tags:
121       - Key: Name
122       - Value: IG
123

```

# 구현 Service infra

## 1. CloudFormation YAML Document

자원 구성 : 라우팅

```

130 #라우팅 테이블 생성
131 PublicRT:
132   Type: AWS::EC2::RouteTable
133   Properties:
134     VpcId: !Ref VPC
135     Tags:
136       - Key: Name
137       Value: Public-RT
138
139 PrivateRT:
140   Type: AWS::EC2::RouteTable
141   Properties:
142     VpcId: !Ref VPC
143     Tags:
144       - Key: Name
145       Value: Private-RT
146

147 #서브넷-라우팅 테이블 연결
148 AssociatePublicSubnet1ToPublicRT:
149   Type: AWS::EC2::SubnetRouteTableAssociation
150   Properties:
151     RouteTableId: !Ref PublicRT
152     SubnetId: !Ref PublicSubnet1
153
154 AssociatePublicSubnet2ToPublicRT:
155   Type: AWS::EC2::SubnetRouteTableAssociation
156   Properties:
157     RouteTableId: !Ref PublicRT
158     SubnetId: !Ref PublicSubnet2
159
160 AssociatePrivateSubnet1ToPrivateRT:
161   Type: AWS::EC2::SubnetRouteTableAssociation
162   Properties:
163     RouteTableId: !Ref PrivateRT
164     SubnetId: !Ref PrivateSubnet1
165
166 AssociatePrivateSubnet2ToPrivateRT:
167   Type: AWS::EC2::SubnetRouteTableAssociation
168   Properties:
169     RouteTableId: !Ref PrivateRT
170     SubnetId: !Ref PrivateSubnet2

171 #라우팅 설정
172 PublicRouteToInternet:
173   Type: AWS::EC2::Route
174   Properties:
175     DestinationCidrBlock: 0.0.0.0/0
176     GatewayId: !Ref IG
177     RouteTableId: !Ref PublicRT
178
179
180 PrivateRouteToInternet:
181   Type: AWS::EC2::Route
182   Properties:
183     DestinationCidrBlock: 0.0.0.0/0
184     # NatGatewayId: !Ref NATGW
185     InstanceId: !Ref NatInstance
186     RouteTableId: !Ref PrivateRT

```

# 구현 Service infra

## 1. CloudFormation YAML Document

자원 구성 : NAT 인스턴스

```

187
188 # 장치 생성
189 #NAT 리소스
190 #보안그룹
191 NatSecurityGroup:
192   DependsOn:
193     - VPC
194   Type: AWS::EC2::SecurityGroup
195   Properties:
196     GroupDescription: NAT Security Group
197     VpcId: !Ref VPC
198   SecurityGroupIngress:
199     - IpProtocol: icmp
200       FromPort: -1
201       ToPort: -1
202       CidrIp: !Ref ServerAccess
203     - IpProtocol: tcp
204       FromPort: 80
205       ToPort: 80
206       CidrIp: !Ref ServerAccess
207     - IpProtocol: tcp
208       FromPort: 443
209       ToPort: 443
210       CidrIp: !Ref ServerAccess
211     - IpProtocol: tcp
212       FromPort: 3389
213       ToPort: 3389
214       CidrIp: !Ref ServerAccess
215   Tags:
216     - Key: Name
217     Value: NAT SG

228
229 NatInstance:
230   DependsOn:
231     - PublicSubnet2
232     - NatSecurityGroup
233   Type: AWS::EC2::Instance
234   Properties:
235     InstanceType: t2.micro
236     KeyName: !Ref KeyName
237     #소스대상 비활성화
238     SourceDestCheck: false
239     ImageId: ami-033a6a056910d1137
240   NetworkInterfaces:
241     - DeviceIndex: 0
242       SubnetId: !Ref PublicSubnet2
243       GroupSet:
244         - !Ref NatSecurityGroup
245       AssociatePublicIpAddress: true
246       DeleteOnTermination: true
247       PrivateIpAddress: 10.0.2.10
248   UserData:
249     Fn::Base64:
250       !Sub |
251         #!/bin/bash
252         cat <<EOF > /etc/sysctl.conf
253         net.ipv4.ip_forward = 1
254         net.ipv4.conf.eth0.send_redirects = 0
255         EOF
256         sysctl -p /etc/sysctl.conf
257         yum -y install iptables-services
258         yum -y update
259         systemctl start iptables
260         systemctl enable iptables
261         iptables -F
262         iptables -t nat -A POSTROUTING -o eth0 -s 0.0.0.0/0 -j MASQUERADE
263         service iptables save
264   Tags:
265     - Key: Name
266     Value: NatInstance

```

# 구현 Service infra

## 1. CloudFormation YAML Document

자원 구성 : BASTION 호스트

```

268
269 #배스천 호스트
270 #보안그룹
271 BastionSG:
272   Type: AWS::EC2::SecurityGroup
273   Properties:
274     GroupDescription: Security Group for Bastion Host to allow SSH
275     SecurityGroupIngress:
276       - IpProtocol: tcp
277         FromPort: 22
278         ToPort: 22
279         CidrIp: 0.0.0.0/0
280     Tags:
281       - Key: Name
282         Value: Bastion-SG
283       VpcId: !Ref VPC
285
286 BastionHost:
287   Type: AWS::EC2::Instance
288   Properties:
289     AvailabilityZone: !Sub ${Region}a
290     InstanceType: t2.micro
291     ImageId: ami-033a6a056910d1137
292     KeyName: !Ref KeyName
293     NetworkInterfaces:
294       - DeviceIndex: "0"
295         SubnetId: !Ref PublicSubnet1
296         GroupSet:
297           - !Ref BastionSG
298         AssociatePublicIpAddress: true
299         #PrivateIpAddress: 10.0.1.10
300     Tags:
301       - Key: Name
302         Value: Bastion Host

```

# 구현 Service infra

## 1. CloudFormation YAML Document

자원 구성 : Application Load Balancer

#애플리케이션 로드 밸런서 #보안그룹 ALBSG: Type: AWS::EC2::SecurityGroup Properties: GroupDescription: Security Group for Application Load Balancer to expose HTTP 80 SecurityGroupIngress: - IpProtocol: tcp   FromPort: 80   ToPort: 80   CidrIp: 0.0.0.0/0 Tags: - Key: Name   Value: ALB-SG VpcId: !Ref VPC	304 305 306 307 308 309 310 311 312 313 314 315 316 317 318	319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340	#로드밸런서 생성 ALB: Type: AWS::ElasticLoadBalancingV2::LoadBalancer Properties: Name: ALB Scheme: internet-facing SecurityGroups: - !Ref ALBSG Subnets: - !Ref PublicSubnet1 - !Ref PublicSubnet2 Type: application #리스너 ALBListener: Type: AWS::ElasticLoadBalancingV2::Listener Properties: DefaultActions: - Type: forward   TargetGroupArn: !Ref ALBTG LoadBalancerArn: !Ref ALB Port: 80 Protocol: HTTP	341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362	#타겟그룹 ALBTG: Type: AWS::ElasticLoadBalancingV2::TargetGroup Properties: HealthCheckIntervalSeconds: 30 HealthCheckProtocol: HTTP HealthCheckTimeoutSeconds: 5 HealthyThresholdCount: 2 Matcher: HttpCode: '200' Name: AppTargets Port: 80 Protocol: HTTP Targets: - Id: !Ref EC2App1   Port: 80 - Id: !Ref EC2App2   Port: 80 VpcId: !Ref VPC Tags: - Key: Name   Value: ALB-TG	341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362
--	---	--	--	--	--	--

# 구현 Service infra

## 1. CloudFormation YAML Document

자원 구성 : 웹 서버

```

365 # 웹서버 1번 2번
366 #보안그룹
367 EC2SG:
368   Type: AWS::EC2::SecurityGroup
369   Properties:
370     GroupDescription: Security Group for EC2 to allow SSH from Bastion and expose HTTP 80
371     SecurityGroupIngress:
372       - IpProtocol: tcp
373         FromPort: 22
374         ToPort: 22
375         SourceSecurityGroupId: !Ref BastionSG
376       - IpProtocol: tcp
377         FromPort: 80
378         ToPort: 80
379         SourceSecurityGroupId: !Ref ALBSG
380     Tags:
381       - Key: Name
382         Value: EC2-SG
383     VpcId: !Ref VPC
384
385 #서버생성
386 EC2App1:
387   Type: AWS::EC2::Instance
388   Properties:
389     AvailabilityZone: !Sub ${Region}a
390     InstanceType: t2.micro
391     ImageId: ami-0ed11f3863410c386
392     #KeyName: !Ref KeyName
393     NetworkInterfaces:
394       - DeviceIndex: 0
395         SubnetId: !Ref PrivateSubnet1
396         GroupSet:
397           - !Ref EC2SG
398         AssociatePublicIpAddress: false
399         PrivateIpAddress: 10.0.3.10
400     UserData:
401       Fn::Base64:
402         !Sub |
403           #!/bin/bash
404           echo 'ubuntu:hackers' | sudo chpasswd
405           sudo sed -i "s/^PasswordAuthentication no/PasswordAuthentication yes/g" /etc/ssh/sshd_config
406           sudo sed -i "s/^#PermitRootLogin yes/PermitRootLogin yes/g" /etc/ssh/sshd_config
407           sudo service sshd restart
408           sudo apt-get update
409           sudo apt-get -y install apache2
410           sudo apt-get -y install mariadb-server mariadb-client
411           sudo systemctl enable apache2
412           sudo systemctl enable mariadb
413           git clone https://github.com/rhymix/rhymix.git
414           cd ~/rhymix
415           mkdir files
416           chmod 777 files
417     Tags:
418       - Key: Name
419         Value: WebServer1

```

# 구현 Service infra

## 1. CloudFormation YAML Document

자원 구성 : 웹 서버

```
419
420 EC2App2:
421   Type: AWS::EC2::Instance
422   Properties:
423     AvailabilityZone: !Sub ${Region}c
424     InstanceType: t2.micro
425     ImageId: ami-0ed11f3863410c386
426     #KeyName: !Ref KeyName
427     NetworkInterfaces:
428       - DeviceIndex: 0
429         SubnetId: !Ref PrivateSubnet2
430         GroupSet:
431           - !Ref EC2SG
432         AssociatePublicIpAddress: false
433         PrivateIpAddress: 10.0.4.10
434     UserData:
435       Fn::Base64:
436         !Sub |
437           #!/bin/bash
438           echo 'ubuntu:hackers' | sudo chpasswd
439           sudo sed -i "s/^PasswordAuthentication no/PasswordAuthentication yes/g" /etc/ssh/sshd_config
440           sudo sed -i "s/^#PermitRootLogin yes/PermitRootLogin yes/g" /etc/ssh/sshd_config
441           sudo service sshd restart
442           sudo apt-get update
443           sudo apt-get -y install apache2
444           sudo apt-get -y install mariadb-server mariadb-client
445           sudo systemctl enable apache2
446           sudo systemctl enable mariadb
447           git clone https://github.com/rhymix/rhymix.git
448           cd ~/rhymix
449           mkdir files
450           chmod 777 files
451     Tags:
452       - Key: Name
453         Value: WebServer2
```

# 구현 Service infra

## 2. CloudFormation Stack

컴퓨팅 자원 생성

**Create stack**

**Prerequisite - Prepare template**

**Prepare template**  
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

Template is ready    Use a sample template    Create template in Designer

**Specify template**  
A template is a JSON or YAML file that describes your stack's resources and properties.

**Template source**  
Selecting a template generates an Amazon S3 URL where it will be stored.

Amazon S3 URL    Upload a template file

**Upload a template file**  
Choose file  aws-private-ha-app.yaml  
JSON or YAML formatted file

S3 URL: <https://s3.ap-northeast-2.amazonaws.com/cf-templates-2wynxnoxg67z-ap-northeast-2/2022106xxh-aws-private-ha-app.yaml>   [View in Designer](#)

[Cancel](#) [Next](#)

[Feedback](#)   © 2022, Amazon Web Services, Inc. or its affiliates.   [Privacy](#)   [Terms](#)   [Cookie preferences](#)

**Specify stack details**

**Stack name**  
Stack name  
Enter a stack name  
Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

**Parameters**  
Parameters are defined in your template and allow you to input custom values when you create or update a stack.

**KeyName**

**PrivateSubnet1Cidr**  
Enter the CIDR for your Private Subnet 1  
10.0.3.0/24

**PrivateSubnet2Cidr**  
Enter the CIDR for your Private Subnet 2  
10.0.4.0/24

**PublicSubnet1Cidr**  
Enter the CIDR for your Public Subnet 1  
10.0.1.0/24

**PublicSubnet2Cidr**  
Enter the CIDR for your Public Subnet 2  
10.0.2.0/24

**Region**  
Enter the AWS region to deploy stack. Default is ap-northeast-2  
ap-northeast-2

**ServerAccess**  
CIDR IP range allowed to login to the NAT instance  
0.0.0.0/0

**VPCCidr**  
Enter the CIDR for your VPC  
10.0.0.0/16

[Cancel](#) [Previous](#) [Next](#)

[Feedback](#)   © 2022, Amazon Web Services, Inc. or its affiliates.   [Privacy](#)   [Terms](#)   [Cookie preferences](#)

# 구현 Service infra

## 2. CloudFormation Stack

컴퓨팅 자원 생성

**CloudFormation > Stacks > Create stack**

**Step 1 Specify template**

**Step 2 Specify stack details**

**Step 3 Configure stack options**

**Tags**  
You can specify tags (key-value pairs) to apply to resources in your stack. You can add up to 50 unique tags for each stack. [Learn more](#)

Key	Value	Remove
-----	-------	--------

**Add tag**

**Permissions**  
Choose an IAM role to explicitly define how CloudFormation can create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses permissions based on your user credentials. [Learn more](#)

**IAM role - optional**  
Choose the IAM role for CloudFormation to use for all operations performed on the stack.

**iamRoleName**

**Stack failure options**

**Behavior on provisioning failure**  
Specify the roll back behavior for a stack failure. [Learn more](#)

**Roll back all stack resources**  
Roll back the stack to the last known stable state.

**Preserve successfully provisioned resources**  
Preserves the state of successfully provisioned resources, while rolling back failed resources to the last known stable state. Resources without a last known stable state will be deleted upon the next stack operation.

**Advanced options**

You can set additional options for your stack, like notification options and a stack policy. [Learn more](#)

**▶ Stack policy**  
Defines the resources that you want to protect from unintentional updates during a stack update.

**▶ Rollback configuration**  
Specify alarms for CloudFormation to monitor when creating and updating the stack. If the operation breaches an alarm threshold, CloudFormation rolls it back. [Learn more](#)

**CloudFormation > Stacks > ALALALAL**

**Stacks (1)**

**ALALALAL**  
2022-04-16 19:27:12 UTC+0900 CREATE\_COMPLETE

**Resources (26)**

Logical ID	Physical ID	Type	Status
ALB	arn:aws:elasticloadbalancing:ap-northeast-2:367784630612:loadbalancer/app/ALB/107c552b2c30c6cf	AWS::ElasticLoadBalancingV2::LoadBalancer	<span style="color: green;">CREATE_COMPLETE</span>
ALBListener	arn:aws:elasticloadbalancing:ap-northeast-2:367784630612:listener/app/ALB/107c552b2c30c6cf/fdd1648e519a0603	AWS::ElasticLoadBalancingV2::Listener	<span style="color: green;">CREATE_COMPLETE</span>
ALBSG	sg-0b4a00a694b4b54f20	AWS::EC2::SecurityGroup	<span style="color: green;">CREATE_COMPLETE</span>
ALBTG	arn:aws:elasticloadbalancing:ap-northeast-2:367784630612:targetgroup/AppTargets/aee33b47781d4ff0	AWS::ElasticLoadBalancingV2::TargetGroup	<span style="color: green;">CREATE_COMPLETE</span>
AssociatePrivateSubnet1ToPrivateRT	rtbassoc-01fd8cfabf35506	AWS::EC2::SubnetRouteTableAssociation	<span style="color: green;">CREATE_COMPLETE</span>
AssociatePrivateSubnet2ToPrivateRT	rtbassoc-0e6bd8549f11892fb	AWS::EC2::SubnetRouteTableAssociation	<span style="color: green;">CREATE_COMPLETE</span>
AssociatePublicSubnet1ToPublicRT	rtbassoc-01822d75eba293c3f	AWS::EC2::SubnetRouteTableAssociation	<span style="color: green;">CREATE_COMPLETE</span>
AssociatePublicSubnet2ToPublicRT	rtbassoc-0c2b7436dfdd6490e5	AWS::EC2::SubnetRouteTableAssociation	<span style="color: green;">CREATE_COMPLETE</span>
AttachGtoVPC	ALALA-Attac-INC9FGQHJUP	AWS::EC2::VPCEGatewayAttachment	<span style="color: green;">CREATE_COMPLETE</span>
BastionHost	i-0d97cde952a10e6c4	AWS::EC2::Instance	<span style="color: green;">CREATE_COMPLETE</span>
BastionSG	sg-07a779698c9dea23f	AWS::EC2::SecurityGroup	<span style="color: green;">CREATE_COMPLETE</span>
EC2App1	i-051802d2c5b9c4b52	AWS::EC2::Instance	<span style="color: green;">CREATE_COMPLETE</span>
EC2App2	i-0f6900a394c5a743	AWS::EC2::Instance	<span style="color: green;">CREATE_COMPLETE</span>
EC2SG	sg-06304306dd55608a2	AWS::EC2::SecurityGroup	<span style="color: green;">CREATE_COMPLETE</span>
IG	igw-0636c5a32f3e795	AWS::EC2::InternetGateway	<span style="color: green;">CREATE_COMPLETE</span>
NatInstance	i-0665872ae53de7104	AWS::EC2::Instance	<span style="color: green;">CREATE_COMPLETE</span>
NatSecurityGroup	sg-0e5b8afe0dc25d58f	AWS::EC2::SecurityGroup	<span style="color: green;">CREATE_COMPLETE</span>
PrivateRT	rtb-0de77c34afbf65d	AWS::EC2::RouteTable	<span style="color: green;">CREATE_COMPLETE</span>
PrivateRouteToInternet	ALALA-Priva-U80L57UQ53WU	AWS::EC2::Route	<span style="color: green;">CREATE_COMPLETE</span>
PrivateSubnet1	subnet-0f1d21eb46a3211b9	AWS::EC2::Subnet	<span style="color: green;">CREATE_COMPLETE</span>
PrivateSubnet2	subnet-0cc90ca4ed9ec152e	AWS::EC2::Subnet	<span style="color: green;">CREATE_COMPLETE</span>
PublicRT	rtb-0b939df346fa0a48	AWS::EC2::RouteTable	<span style="color: green;">CREATE_COMPLETE</span>
PublicRouteToInternet	ALALA-Publi-XAC9Q2L0L5B1	AWS::EC2::Route	<span style="color: green;">CREATE_COMPLETE</span>
PublicSubnet1	subnet-05e6a54acf0bcf1ab	AWS::EC2::Subnet	<span style="color: green;">CREATE_COMPLETE</span>
PublicSubnet2	subnet-0e265646377e1a75c	AWS::EC2::Subnet	<span style="color: green;">CREATE_COMPLETE</span>
VPC	vpc-01b4e1cab9b6f342	AWS::EC2::VPC	<span style="color: green;">CREATE_COMPLETE</span>

# 구현 Service infra

## 3. Instance (bastion host)

인스턴스 동작 확인 (SSH Connect : Public Subnet → Private Subnet)

The image shows two terminal windows side-by-side. Both terminals are connected to the same Amazon Linux 2 instance, which is running in a private subnet.

**Terminal 1 (Left):**

```

aws
Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type
ami-02de72c5dc79358c9 (64비트(x86)) /
ami-0d0bd26f0eac17ada (64비트(Arm))

Amazon Linux 2는 5년간 지원을 제공합니다. Amazon EC2에 성능
최적화된 Linux kernel 5.10와 systemd 219, GCC 7.3, Glibc 2.26,
Binutils 2.29.1, 최신 소프트웨어 패키지를 추가적으로 제공합니다.
플랫폼: amazon 루트 디바이스 유형: ebs 가상화: hvm ENA 활성화됨: 예

프리 티어 사용 가능

```

**Terminal 2 (Right):**

```

Xshell 7 (Build 0098)
Copyright (c) 2020 NetSarang Computer, Inc. All rights reserved.

Type 'help' to learn how to use Xshell prompt.
[C:\~]$ 

Host 'ec2-52-79-73-222.ap-northeast-2.compute.amazonaws.com' resolved to 52.79.73.222.
Connecting to 52.79.73.222:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+'.

WARNING! The remote SSH server rejected X11 forwarding request.
Last login: Thu Apr 14 10:59:25 2022 from 125.186.195.115

[~] ( [ ~ ] / ) Amazon Linux 2 AMI
[~] \ [ ~ ] |
```

Both terminals show identical output, indicating a successful connection from the bastion host in the public subnet to the instance in the private subnet.

# 구현 Service infra

## 4. DNS hosting

ALB DNS 와 URL 연결 www.ncpa.tk

**Basic Configuration**

Name	ALB
ARN	arn:aws:elasticloadbalancing:ap-northeast-2:367784630612:loadbalancer/app/ALB/107c552b2c30c6cf
DNS name	ALB-1681950073.ap-northeast-2.elb.amazonaws.com (A Record)
State	Active
Type	application
Scheme	internet-facing
IP address type	ipv4
VPC	vpc-01b4e1cabb9b6f342
Availability Zones	subnet-05e6a54acf0bcf1ab - ap-northeast-2a
IPv4 address	Assigned by AWS

**Quick create record**

**Record 1**

Record name	www.ncpa.tk
Record type	A - Routes traffic to an IPv4 address and some AWS resources
Route traffic to	Alias
Alias to Application and Classic Load Balancer	dualstack.ALB-1681950073.ap-northeast-2.elb.amazonaws.com
Routing policy	Simple routing
Evaluate target health	Yes

**Existing records (2)**

Record name	Type	Routing policy	Differences	Value/Route traffic to
ncpa.tk	NS	Simple	-	ns-1703.awsdns-20.co.uk. ns-69.awsdns-08.com. ns-1384.awsdns-45.org. ns-805.awsdns-36.net.
ncpa.tk	SOA	Simple	-	ns-1703.awsdns-20.co.uk. awsdns-hostmaster.amazon.com. 1 7200 900 1209600 86400

# 구현 Service infra

## 5. Web Service

웹 서비스 구성 (EC2 webserver1, EC2 webserver2, install APM + Rhymix CMS )

The screenshot shows the AWS CloudFormation configuration interface. It displays the selected AMI for the EC2 instances, which is Ubuntu Server 18.04 LTS (HVM, SSD Volume Type). The AMI ID shown is ami-0ed11f3863410c386 (64비트(x86)) / ami-0b9d4fcf7aa7336d (64비트(Arm)). The interface also shows other configuration details like EBS General Purpose (SSD) Volume Type and Canonical support.

The screenshot shows the XEDITION website running on webserver1. The page features a large background image of dry grass and the text "EVOLUTION & INNOVATION TOGETHER". Below the main image, there is Korean text: "함께 진화하고 혁신을 추구합니다." At the bottom right, there is a "로그인해주세요!" button.

webserver1

The screenshot shows the XEDITION website running on webserver2. The page features a large background image of hands holding a tablet and the text "SHARING, PUBLISHING. & PLEASURE.". Below the main image, there is Korean text: "지식을 나누고 컨텐츠를 출판하며 즐거움을 함께합니다." At the bottom right, there is a "로그인해주세요!" button.

webserver2

# 구현 Management & Analytics

## 6. SSM – IAM Role

AWS System Manager 가 EC2 에 CloudWatch agent/config 설치를 하기위한 권한 및 역할 생성/부여 ( WebServer1, 2 )

**Identity and Access Management (IAM)**

**SSM-Manage-EC2**

Allows EC2 instances to call AWS services on your behalf.

**Summary**

Creation date: April 15, 2022, 21:31 (UTC+09:00)

ARN: arn:aws:iam::367784630612:role/SSM-Manage-EC2

Last activity: 22 hours ago

Maximum session duration: 1 hour

**Permissions**

Selected 2/4

Policy name Type Description

- AmazonEC2RoleforSSM AWS managed This policy will soon be deprecated
- CloudWatchAgentAdminPolicy AWS managed Full permissions required to use CloudWatch Agent
- CloudWatchAgentServerPolicy AWS managed Permissions required to use CloudWatch Agent
- AmazonSSMReadOnlyAccess AWS managed Provides read only access to AWS Systems Manager

**Permissions boundary - (not set)**

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting but can be used to delegate permission management to others.

**Set permissions boundary**

**인스턴스 (1/4) 정보**

Name	인스턴스 ID	인스턴스 상태	인스턴스 유형	상태 검사	작업	인스턴스 시작
WebServer1	i-0ce67ca33ca5c2ba2	실행 중	t2.micro	2/2개	연결	연결
Bastion Host	i-0993bd023b6f004bc	실행 중	t2.micro	2/2개	인스턴스 관리	인스턴스 설정
NatInstance	i-07012bfed25ccdd668	실행 중	t2.micro	2/2개	보안 그룹 변경	네트워킹
WebServer2	i-0916e63d1b3601733	실행 중	t2.micro	2/2개	Windows 암호 가져오기	보안

**IAM 역할 설정**

**IAM 역할 설정 정보**

IAM 역할을 인스턴스에 연결합니다.

**인스턴스 ID**: i-0ce67ca33ca5c2ba2 (WebServer1)

**IAM 역할**: 인스턴스에 연결할 IAM 역할을 선택하거나 역할이 생성되어 있지 않다면 새 역할을 생성합니다. 선택한 역할이 현재 인스턴스에 연결된 모든 역할을 대체합니다.

**SSM-Manage-EC2**

**새 IAM 역할 생성**

**인스턴스: i-0ce67ca33ca5c2ba2(WebServer1)**

**위에서 인스턴스 선택**

**보안 세부 정보**

IAM 역할: 소유자 ID: 367784630612 | 시작 시간: Fri Apr 15 2022 21:18:02 GMT+0900 (한국 표준시)

**인스턴스 i-0ce67ca33ca5c2ba2(SSM-Manage-EC2)를 연결함**

**인스턴스 (1/4) 정보**

Name	인스턴스 ID	인스턴스 상태	인스턴스 유형	상태 검사	경보 상태	작업	인스턴스 시작
WebServer1	i-0ce67ca33ca5c2ba2	실행 중	t2.micro	2/2개	경보 없음	+	연결
Bastion Host	i-0993bd023b6f004bc	실행 중	t2.micro	2/2개	경보 없음	+	인스턴스 관리
NatInstance	i-07012bfed25ccdd668	실행 중	t2.micro	2/2개	경보 없음	+	보안 그룹 변경
WebServer2	i-0916e63d1b3601733	실행 중	t2.micro	2/2개	경보 없음	+	Windows 암호 가져오기

**인스턴스: i-0ce67ca33ca5c2ba2(WebServer1)**

**위에서 인스턴스 선택**

**보안 세부 정보**

IAM 역할: 소유자 ID: 367784630612 | 시작 시간: Fri Apr 15 2022 21:18:02 GMT+0900 (한국 표준시)

# 구현 Management & Analytics

## 6. SSM – Install CloudWatch agent

각종 지표 정보 전송과 감시를 위한 에이전트 설치 ( Cloudwatch agent install )

**Command document**  
Select the type of command that you want to run.

Search by keyword or filter by tag or attributes  
Search: AWS-ConfigureAWSPackage X Clear filters

Name	Owner	Platform types
AWS-ConfigureAWSPackage	Amazon	Windows, Linux, MacOS

Description  
Install or uninstall a Distributor package. You can install the latest version, default version, or a version of the package you specify. Packages provided by AWS such as AmazonCloudWatchAgent, AwsEnaNetworkDriver, and AWSPVDriver are also supported.

Document version  
Choose the document version you want to run.  
1 (Default)

**Command parameters**

Action  
(Required) Specify whether or not to install or uninstall the package.  
Install

Installation Type  
(Optional) Specify the type of installation. Uninstall and reinstall: The application is taken offline until the reinstallation process completes. In-place update: The application is available while new or updated files are added to the installation.  
Uninstall and reinstall

Name  
(Required) The package to install/uninstall.  
AmazonCloudWatchAgent

Version  
(Optional) The version of the package to install or uninstall. If you don't specify a version, the system installs the latest published version by default. The system will only attempt to uninstall the version that is currently installed. If no version of the package is installed, the system returns an error.

Additional Arguments  
(Optional) The additional parameters to provide to your install, uninstall, or update scripts.  
{}

# 구현 Management & Analytics

## 6. SSM – Install CloudWatch agent

각종 지표 정보 전송과 감시를 위한 에이전트 설치 ( Cloudwatch agent install )

The screenshot shows the AWS Systems Manager Command interface. At the top, a green success message box displays: "명령 ID: 2bc54e41-089c-436a-96fd-cda18654e859 성공적으로 전송!" (Command ID: 2bc54e41-089c-436a-96fd-cda18654e859 successfully sent). Below this, the navigation path is "AWS Systems Manager > 명령 실행 > 명령 ID: 2bc54e41-089c-436a-96fd-cda18654e859". The main content area shows the command details for "명령 ID: 2bc54e41-089c-436a-96fd-cda18654e859". It includes tabs for "명령 취소" (Cancel), "명령 재실행" (Re-execute), and "Copy to new". The "명령 상태" section shows the following metrics: 전체 상태 (Success), 상세 상태 (Success), 대상 개수 (2), 완료 개수 (2), 오류 횟수 (0), and 전송 제한 시간 초과 횟수 (0). The "대상 및 출력" section lists two instances: "i-0ce67ca33ca5c2ba2" and "i-0916e63d1b3601733", both in a "성공" (Success) state. A search bar and a "출력 보기" (View Output) button are also present.

# 구현 Management & Analytics

## 6. SSM – ParameterStore CloudWatch agent config

각종 지표, 로그 모니터링에 대한 설정 ( Cloudwatch agent config )

**파라미터 생성**

**파라미터 세부 정보**

**이름**: Monitor-Web-server

**설명 – Optional**: 웹서버로그수집

**계층**: 파라미터 스토어는 표준 및 고급 파라미터를 제공합니다.

- 표준**: 파라미터의 한도는 10,000개입니다. 파라미터 값의 최대 크기는 4KB입니다. 파라미터 정책을 사용할 수 없습니다. 추가 요금은 부과되지 않습니다.
- 고급**: 10,000개 이상의 파라미터를 생성할 수 있습니다. 파라미터 값의 최대 크기는 8KB입니다. 파라미터 정책을 사용할 수 있으며, 요금이 부과됩니다.

**유형**: 문자열 (선택)

**데이터 형식**: text

**값**:

```
{
    "log_group": "aws/lambda/functions"
}
```

최대 길이는 4,096자입니다.

**연결 생성 요청이 성공했습니다.**

AWS Systems Manager > 파라미터 스토어

**내 파라미터** | **공용 파라미터** | **설정**

선택	이름	계층	유형	마지막 수정 날짜
<input type="checkbox"/>	Monitor-Web-server	표준	String	Fri, 15 Apr 2022 13:28:46 GMT

# 구현 Management & Analytics

## 6. SSM – ParameterStore CloudWatch agent config

각종 지표, 로그 모니터링에 대한 설정 ( Cloudwatch agent config )

Apache webserver log

```
"logs": {
    "logs_collected": {
        "files": {
            "collect_list": [
                {
                    "file_path": "/var/log/apache2/access.log",
                    "log_group_name": "/apache/access",
                    "log_stream_name": "{instance_id}",
                    "retention_in_days": -1
                },
                {
                    "file_path": "/var/log/apache2/error.log",
                    "log_group_name": "/apache/error",
                    "log_stream_name": "{instance_id}",
                    "retention_in_days": -1
                }
            ]
        }
    }
},
```

사용자 모니터링 지표 namespace 설정

```
"metrics": {
    "namespace": "DevOps/Custom_EC2",
    "append_dimensions": {
        "AutoScalingGroupName": "${aws:AutoScalingGroupName}",
        "ImageId": "${aws:ImageId}",
        "InstanceId": "${aws:InstanceId}",
        "InstanceType": "${aws:InstanceType}"
    }
},
```

# 구현 Management & Analytics

## 6. SSM – ParameterStore CloudWatch agent config

각종 지표, 로그 모니터링에 대한 설정 ( Cloudwatch agent config )

### 디스크 모니터링

```
"metrics_collected": {
    "disk": {
        "measurement": [
            "used_percent",
            "inodes_free",
            "used",
            "total"
        ],
        "metrics_collection_interval": 60,
        "resources": ["*"]
    },
    ...
}
```

### CPU

```
"cpu": {
    "measurement": [
        "cpu_usage_idle",
        "cpu_usage_iowait",
        "cpu_usage_user",
        "cpu_usage_system",
        "cpu_time_idle",
        "cpu_time_iowait",
        "cpu_time_user",
        "cpu_time_system"
    ],
    "metrics_collection_interval": 60,
    "resources": ["*"],
    "totalcpu": true
},
```

### 메모리 모니터링

```
"mem": {
    "measurement": [
        "mem_used_percent",
        "mem_total",
        "mem_used",
        "mem_free"
    ],
    "metrics_collection_interval": 60
},
```

### TCP 모니터링

```
"netstat": {
    "measurement": [
        "tcp_established",
        "tcp_time_wait"
    ],
    "metrics_collection_interval": 60
}
```

# 구현 Management & Analytics

## 6. SSM – Install CloudWatch agent config

각종 지표, 로그 모니터링에 대한 설정 적용 ( Cloudwatch agent config )

명령 문서

설정할 명령 유형을 선택합니다.

검색: AmazonCloudWatch-ManageAgent

이름	소유자	플랫폼 유형
AmazonCloudWatch-ManageAgent	Amazon	Windows, Linux, MacOS

설명

문서 버전   
6 (기본값)

명령 파라미터

Action  
The action CloudWatch Agent should take.

Mode  
Controls platform-specific default behavior such as whether to include EC2 Metadata in metrics.

Optional Configuration Source  
Only for 'configure' related actions. Use 'ssm' to apply a ssm parameter as config. Use 'default' to apply default config for amazon-cloudwatch-agent. Use 'all' with 'configure (remove)' to clean all configs for amazon-cloudwatch-agent.

Optional Configuration Location  
Only for 'configure' related actions. Only needed when Optional Configuration Source is set to 'ssm'. The value should be a ssm parameter name.

Optional Open Telemetry Collector Configuration Source  
Only for 'configure' related actions. Use 'ssm' to apply a ssm parameter as config. Use 'default' to apply default config for amazon-cloudwatch-agent. Use 'all' with 'configure (remove)' to clean all configs for amazon-cloudwatch-agent. It does not support MacOS instance.

Optional Open Telemetry Collector Configuration Location  
Only for 'configure' related actions. Only needed when Optional Configuration Source is set to 'ssm'. The value should be a ssm parameter name. It does not support MacOS instance.

Optional Restart  
Only for 'configure' related actions. If 'yes', restarts the agent to use the new configuration. Otherwise the new config will only apply on the next agent restart.

# 구현 Management & Analytics

## 6. SSM – Install CloudWatch agent config

각종 지표, 로그 모니터링에 대한 설정 적용 ( Cloudwatch agent config )

**대상**  
대상을 선택하기 위한 방법을 선택합니다.

- 인스턴스 태그 지정  
해당 태그를 공유하는 인스턴스를 선택하기 위해 하나 이상의 태그 키-값 페어를 지정합니다.
- 수동으로 인스턴스 선택  
대상으로 등록할 인스턴스를 수동으로 선택합니다.
- 리소스 그룹 선택  
대상으로 지정할 리소스가 포함된 리소스 그룹을 선택합니다.

i-0ce67ca33ca5c2ba2 X    i-0916e63d1b3601733 X

**인스턴스**

Node ID	Source type	Source ID	이름	Ping 상태	Node
<input checked="" type="checkbox"/> i-0ce67ca33ca5c2ba2	AWS::EC2::Instance	i-0ce67ca33ca5c2ba2	WebServer1	Online	running
<input checked="" type="checkbox"/> i-0916e63d1b3601733	AWS::EC2::Instance	i-0916e63d1b3601733	WebServer2	Online	running

명령 ID: 0657c775-3cf9-48bc-8e34-5f18dd7f1fd3 성공적으로 전송!

AWS Systems Manager > 명령 실행 > 명령 ID: 0657c775-3cf9-48bc-8e34-5f18dd7f1fd3

명령 ID: 0657c775-3cf9-48bc-8e34-5f18dd7f1fd3

C 명령 취소 명령 재실행 Copy to new

**명령 상태**

전체 상태	상세 상태	대상 개수	완료 개수	오류 횟수	전송 제한 시간 초과 횟수
<input checked="" type="radio"/> 성공	<input checked="" type="radio"/> 성공	2	2	0	0

**대상 및 출력**

인스턴스 ID	인스턴스 이름	상태	상세 상태	시작 시간	완료 시간
<input type="radio"/> i-0916e63d1b3601733	ip-10-0-4-10.ap-northeast-2.compute.internal	<input checked="" type="radio"/> 성공	<input checked="" type="radio"/> 성공	Fri, 15 Apr 2022 13:47:41 GMT	Fri, 15 Apr 2022 13:47:41 GMT
<input type="radio"/> i-0ce67ca33ca5c2ba2	ip-10-0-3-10.ap-northeast-2.compute.internal	<input checked="" type="radio"/> 성공	<input checked="" type="radio"/> 성공	Fri, 15 Apr 2022 13:47:41 GMT	Fri, 15 Apr 2022 13:47:41 GMT

# 구현 Management & Analytics

## 7. ALB – AccessLog Enable

애플리케이션 로드밸런서 액세스 로그 기록 활성화

# 구현 Management & Analytics

## 7. ALB – AccessLog Enable

애플리케이션 로드밸런서 엑세스 로그 기록 활성화 이후 자동으로 S3 버킷이 생성되며 정책이 생성됨을 확인

The screenshot shows two side-by-side AWS management console pages. On the left, the 'Bucket' list page displays three buckets: 'cf-templates-2wynxnoxg67z-ap-northeast-2' (located in Asia Pacific (Seoul)), 'cf-templates-2wynxnoxg67z-us-east-1' (located in US East (N. Virginia)), and 'devops-log-alb-access' (located in Asia Pacific (Seoul)). The 'devops-log-alb-access' bucket was created automatically by the ALB after enabling access logging. On the right, the 'Bucket Policy' editor shows the JSON policy document for this bucket. The policy grants full control to the IAM user '600734575887' and allows log delivery from the service 'delivery.logs.amazonaws.com'.

이름	AWS 리전	액세스	생성 날짜
cf-templates-2wynxnoxg67z-ap-northeast-2	아시아 태평양(서울) ap-northeast-2	객체를 퍼블릭으로 설정할 수 있음	2022. 4. 13. pm 12:07:02 PM KST
cf-templates-2wynxnoxg67z-us-east-1	미국 동부(버지니아 북부) us-east-1	객체를 퍼블릭으로 설정할 수 있음	2022. 4. 14. pm 1:31:39 PM KST
devops-log-alb-access	아시아 태평양(서울) ap-northeast-2	객체를 퍼블릭으로 설정할 수 있음	2022. 4. 15. pm 11:01:03 PM KST

```

{
    "Sid": "AWSConsoleStmt-1650031261716",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::600734575887:root"
    },
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::devops-log-alb-access/AWSLogs/367784630612/*"
},
{
    "Sid": "AWSLogDeliveryWrite",
    "Effect": "Allow",
    "Principal": {
        "Service": "delivery.logs.amazonaws.com"
    },
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::devops-log-alb-access/AWSLogs/367784630612/*",
    "Condition": {
        "StringEquals": {
            "s3:x-amz-acl": "bucket-owner-full-control"
        }
    }
}
  
```

# 구현 Management & Analytics

## 8. Athena

애플리케이션 로드밸런서 엑세스 로그 분석을 위한 Athena 서비스 세팅

The screenshot shows the 'Amazon S3 > 버킷 > 버킷 만들기' (Create New Bucket) wizard. The 'General Configuration' section is displayed, containing fields for 'Bucket Name' (alblogs-athena-result-s3), 'Region' (Asia Pacific (Seoul) ap-northeast-2), and a note about selecting a replication bucket. A 'Bucket Selection' button is at the bottom.

The screenshot shows the 'Amazon S3 > 버킷 > alblogs-athena-result-s3' details page. The 'Objects' tab is selected, showing a message 'Object list is empty'. It includes standard S3 object management buttons like 'Upload', 'Delete', and 'Copy'.

Athena의 결과가 저장될 버킷 생성

# 구현 Management & Analytics

## 8. Athena

애플리케이션 로드밸런서 엑세스 로그 분석을 위한 Athena 서비스 세팅

Amazon Athena > 쿼리 편집기 > 설정 관리

### 설정 관리

쿼리 결과 위치 및 암호화

쿼리 결과의 위치  
쿼리 결과가 객체로 저장될 현재 리전에 S3 접두사를 입력합니다.

X 보기 S3 찾아보기

예상 버킷 소유자  
쿼리 결과 출력 위치 버킷의 소유자일 것으로 예상되는 AWS 계정 ID를 지정합니다.

쿼리 결과 암호화  
 활성화

버킷 소유자에게 쿼리 결과에 대한 전체 제어 권한 할당  
이 옵션을 활성화하면 S3 쿼리 결과 버킷의 소유자가 쿼리 결과에 대한 전체 제어 권한을 부여합니다. 즉, 쿼리 결과 위치를 다른 계정에서 소유한 경우 쿼리 결과에 대한 전체 제어 권한을 다른 계정에 부여할 수 있습니다.

취소 저장

설정이 업데이트되었습니다.

Amazon Athena > 쿼리 편집기

편집기 최근 쿼리 저장된 쿼리 설정 작업 그룹 primary ▾

쿼리 결과 및 암호화 설정 관리

쿼리 결과 위치 및 암호화

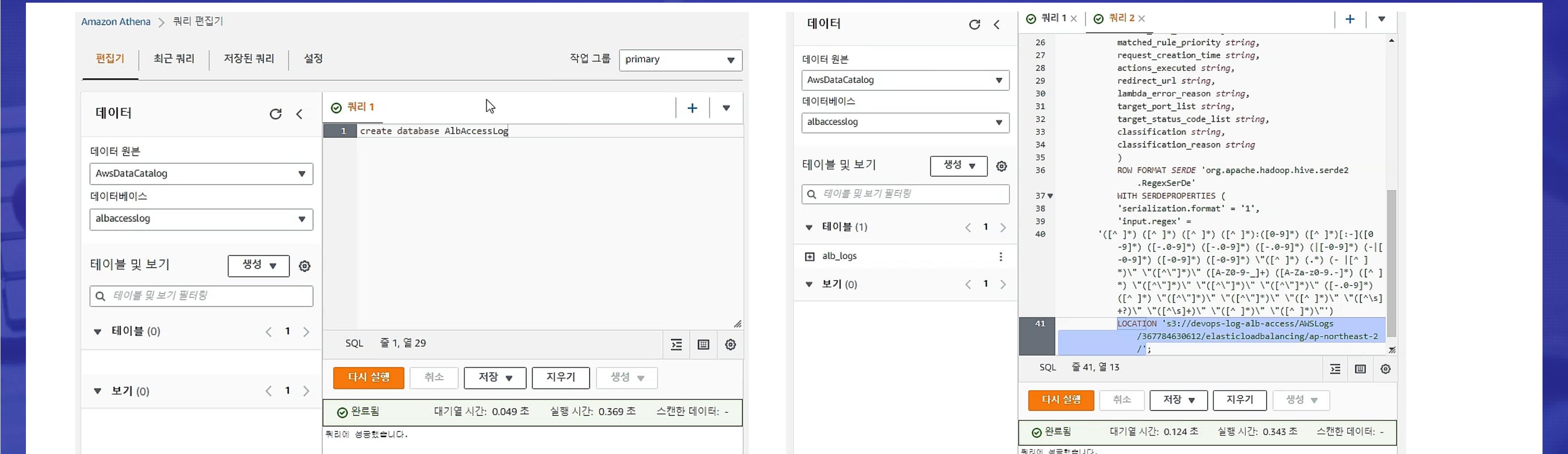
쿼리 결과 위치 <a href="#">s3://alblogs-athena-result-s3/</a>	쿼리 결과 암호화 -	예상 버킷 소유자 -	버킷 소유자에게 쿼리 결과에 대한 전체 제어 권한 할당 비활성화됨
--	----------------	----------------	---

Athena 설정

# 구현 Management & Analytics

## 8. Athena

애플리케이션 로드밸런서 액세스 로그 분석을 위한 Athena 서비스 세팅



The screenshot shows the Amazon Athena Query Editor interface. On the left, the 'Query Editor' tab is selected. In the 'Data' section, under 'Data Source', 'AwsDataCatalog' is chosen. Under 'Database', 'albaccesslog' is selected. In the 'Tables and Views' section, a new table is being created with the following SQL command:

```
create database AlbAccessLog
```

In the main pane, the 'Data' section shows the newly created database 'AlbAccessLog'. The 'Tables' section lists a single table 'alb\_logs'. The 'Views' section is currently empty.

On the right, the 'Data' section shows the configuration for the 'alb\_logs' table. The 'Table Format' is set to 'ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe''. The 'Serde Properties' include 'serialization.format = 1' and 'input.regex = '([^\n]\*(?![^\n]\*\n)[^\n]\*)\n'''. The 'Location' is specified as 's3://devops-log-alb-access/AWSLogs/367784630612/elasticloadbalancing/ap-northeast-2/'. Below this, the 'SQL' section shows the executed command:

```
LOCATION 's3://devops-log-alb-access/AWSLogs/367784630612/elasticloadbalancing/ap-northeast-2/';
```

Athena에서 log database, table 생성

# 구현 Management & Analytics

## 9. Lambda - IAM

Lambda 함수에 부여할 애플리케이션 로드밸런서 엑세스 로그 분석을 위한 S3 접근권한, 클라우드와치 스트림전송 권한 생성

**IAM > 역할 > alblog-cloudwatch-role-vs2fxwnz**

**요약**

**생성 날짜**: April 16, 2022, 00:17 (UTC+09:00)

**ARN**: arn:aws:iam::367784630612:role/service-role/alblog-cloudwatch-role-vs2fxwnz

**마지막 활동**: 없음

**최대 세션 지속 시간**: 1시간

**권한** | 신뢰 관계 | 태그 | 액세스 관리자 | 세션 취소

**권한 정책 (2)**

최대 10개의 관리형 정책을 연결할 수 있습니다.

검색 또는 정책 이름을 기준으로 정책을 필터링하고 Enter를 누릅니다.

정책 이름	유형	설명
AmazonS3ReadOnlyAccess	AWS 관리형	Provides read only access
AWSOpsWorksCloudWatchLogs	AWS 관리형	Enables OpsWorks insta

**AmazonS3ReadOnlyAccess**

Provides read only access to all buckets via the AWS Management Console.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:Get*",
        "s3>List*",
        "s3-object-lambda:Get*",
        "s3-object-lambda>List*"
      ],
      "Resource": "*"
    }
  ]
}
```

**AWSOpsWorksCloudWatchLogs**

Enables OpsWorks instances with the CWLogs integration enabled to ship logs and create required log groups

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs>CreateLogGroup",
        "logs>CreateLogStream",
        "logs>PutLogEvents",
        "logs>DescribeLogStreams"
      ],
      "Resource": [
        "arn:aws:logs:*:*:*"
      ]
    }
  ]
}
```

# 구현 Management & Analytics

## 9. Lambda Function

소스코드 : S3버킷에서 ALB Logs 를 가져와서 CloudWatch logs 에 파싱하는 Lambda 함수(오픈소스 활용)를 생성하고 환경변수 지정

Lambda > 함수 > alblog-cloudwatch

**alblog-cloudwatch**

함수 개요 정보

설명  
-

마지막 수정  
11분 전

함수 ARN  
arn:aws:lambda:ap-northeast-2:367784630612:function:alblog-cloudwatch

함수 URL 정보  
-

+ 트리거 추가 + 대상 추가

코드 | 테스트 | 모니터링 | 구성 | 별칭 | 버전

코드 소스 정보

File Edit Find View Go Tools Window Test Deploy Changes not deployed

CloudWatch\_LogGroup

```

206 print("Sorting file")
207 if (inputformat == 'alb'):
208     # ALB format must be sorted on col2 (datetime)
209     print(subprocess.check_output(['sort', '-k2', '-o', localfile_unzipped, localfile_unzipped]))
210 elif (inputformat == 'cloudfront'):
211     # Cloudfront format must be sorted on col 2 and 3 (date and time)
212     print(subprocess.check_output(['sort', '-k2,3', '-o', localfile_unzipped, localfile_unzipped]))
213
214 print("Get next sequence token")
215 response = GLOBAL_CWL.describe_log_streams(
216     logGroupName=loggroup,
217     logStreamNamePrefix=logstream,
218     orderBy='LogStreamName'
219 )
220 nextToken = response['logStreams'][0].get('uploadSequenceToken', None)
221
222 # Read event line by line, create a buffer, send the buffer every 500 log
223 events = []
224 f = open(localfile_unzipped, 'rt')
225 i = 0
226 for line in f:
227     i += 1
228     if (inputformat == 'alb'):
229         timestamp, message = alb_read_and_convert(i, line, outputformat)
230     else:
231         message = "LAMBDA ERROR : input format " + inputformat + " is unknown !"
232         timestamp = 1000 * time.time()
233     if (timestamp == 0 or message == None):
234

```

Lambda > 함수 > alblog-cloudwatch > 환경 변수 편집

환경 변수 편집

환경 변수

환경 변수를 함수 코드에서 액세스할 수 있는 키-값 페어로 정의할 수 있습니다. 이렇게 하면 함수 코드를 변경하지 않고도 구성 설정을 저장하는데 유용합니다. 자세히 알아보기

키	값
CloudWatch_LogGroup	log-alb-access-lambda
InputFormat	alb
OutputFormat	json

환경 변수 추가

암호화 구성

취소 저장

# 구현 Management & Analytics

## 9. Lambda Function

트리거 추가 : Lambda 함수가 작동하는 시작점(S3버킷에 ALB가 log를 쓰는 시점)을 설정

**Lambda > 추가 트리거**

### 추가 트리거

**트리거 구성**

S3 aws storage

버킷  
이벤트 소스의 역할을 하는 S3 버킷을 선택하십시오. 버킷은 할수와 같은 리전에 있어야 합니다.

devops-log-alb-access

이벤트 유형  
Lambda 할수를 트리거하려는 이벤트를 선택합니다. 필요에 따라 이벤트의 접두사 또는 접미사를 설정할 수 있습니다. 하지만 각 버킷에서 개별 이벤트는 접두사나 접미사가 겹쳐서 객체 키가 중복해질 수 있는 구성을 여러 개 가질 수 없습니다.

모든 객체 생성 이벤트

접두사 - 선택 사항  
필요할 경우, 일지하는 문자로 시작하는 키를 사용하여 객체에 대해 알림을 제한하려는 단일 접두사를 입력합니다.

AWSLogs/367784630612/elasticloadbalancing/ap-northeast-2

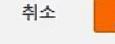
접미사 - 선택 사항  
필요할 경우, 일지하는 문자로 끝나는 키를 사용하여 객체에 대해 알림을 제한하려는 단일 접미사를 입력합니다.

jpg

Lambda는 Amazon S3이(가) 이 트리거에서 Lambda 함수를 호출하는 데 필요한 권한을 주어줍니다. Lambda 권한 모델에 대해 자세히 알아보기.

**재귀 호출**  
함수가 S3 버킷에 객체를 쓰는 경우 입력 및 출력에 다른 S3 버킷을 사용하고 있는지 확인합니다. 동일한 버킷에 쓰면 재귀 호출이 생성될 위험이 증가하며 이는 Lambda 사용량 증가 및 비용 증가를 발생시킬 수 있습니다. 자세히 알아보기

입력과 출력 모두에 동일한 S3 버킷을 사용하는 것은 권장되지 않으며, 이 구성으로 인해 재귀 호출, Lambda 사용량 증가 및 비용 증가가 발생할 수 있음을 알고 있습니다.

취소 

**Lambda > 함수 > alblog-cloudwatch**

### alblog-cloudwatch

초절 ARN 복사 작업 ▾

트리거 함수를 devops-log-alb-access 추가했습니다 alblog-cloudwatch. 이 함수는 지금 트리거에서 이벤트를 수신하고 있습니다.

**함수 개요 정보**

설명 -

마지막 수정 14분 전

함수 ARN arn:aws:lambda:ap-northeast-2:367784630612:function:alblog-cloudwatch

함수 URL 정보 -

**구성**

코드 테스트 모니터링 **구성** 별칭 버전

일반 구성 트리거 (1)

트리거  트리거 찾기

C 활성화 비활성화 오류 수정 삭제 트리거 추가 < 1 >

# 결과 Webserver

## 1. EC2 Custom metric

AWS 기본 지표와 사용자 정의 지표가 기록됨을 확인

**Custom namespaces**

- DevOps/Custom\_EC2 124
- devops/custom\_metrics 247

**AWS namespaces**

- ApplicationELB 177
- EBS 189
- EC2 357
- 로그 18

**지표 (124) 정보**

Seoul ▾ 모두 > DevOps/Custom\_EC2  지표, 자원 또는 리소스 ID 검색 SQL 사용 그래프 그래프 검색

InstanceId, InstanceType, device	80	InstanceId, InstanceType, cpu	32	InstanceId, InstanceType	12
----------------------------------	----	-------------------------------	----	--------------------------	----

**CloudWatch Metrics**

제목 없는 그레프 CloudWatch Metrics

1h 3h 12h 1d 3d 1w Custom 행 작업 G

CloudWatch 그레프가 비어 있습니다.  
여기에서 표시할 일부 지표를 선택하십시오.

20:00 20:15 20:30 20:45 21:00 21:15 21:30 21:45 22:00 22:15 22:30 22:45

찾아보기 쿼리 그래프로 표시된 지표 옵션 소스 수학 추가 ▾ 쿼리 추가 ▾

**지표 (32) 정보**

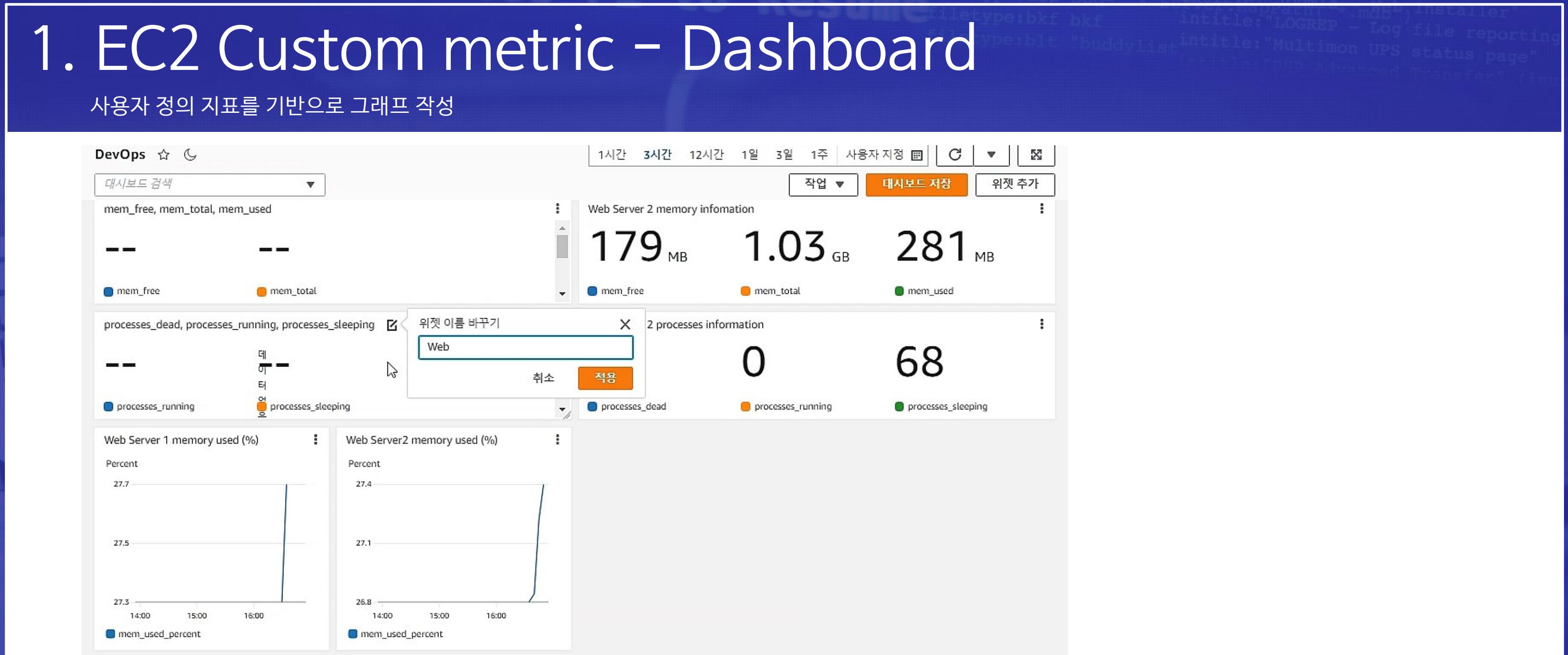
Seoul ▾ 모두 > DevOps/Custom\_EC2 > Instance name (32)  지표, 자원 또는 리소스 ID 검색 SQL 사용 그래프 그래프 검색

Instance name (32)	InstanceId	Instancetype	CPU	Metric name
WebServer1	ami-0ed11f3863410...	i-0ce67ca33ca5c2ba2	t2.micro	cpu-total
WebServer1	ami-0ed11f3863410...	i-0ce67ca33ca5c2ba2	t2.micro	cpu_usage_user
WebServer1	ami-0ed11f3863410...	i-0ce67ca33ca5c2ba2	t2.micro	cpu_usage_iowait
WebServer1	ami-0ed11f3863410...	i-0ce67ca33ca5c2ba2	t2.micro	cpu_usage_system
WebServer1	ami-0ed11f3863410...	i-0ce67ca33ca5c2ba2	t2.micro	cpu_usage_idle
WebServer1	ami-0ed11f3863410...	i-0ce67ca33ca5c2ba2	t2.micro	cpu_time_user

# 결과 Webserver

## 1. EC2 Custom metric - Dashboard

사용자 정의 지표를 기반으로 그래프 작성



# 결과 Webserver

## 2. Apache Webserver Log

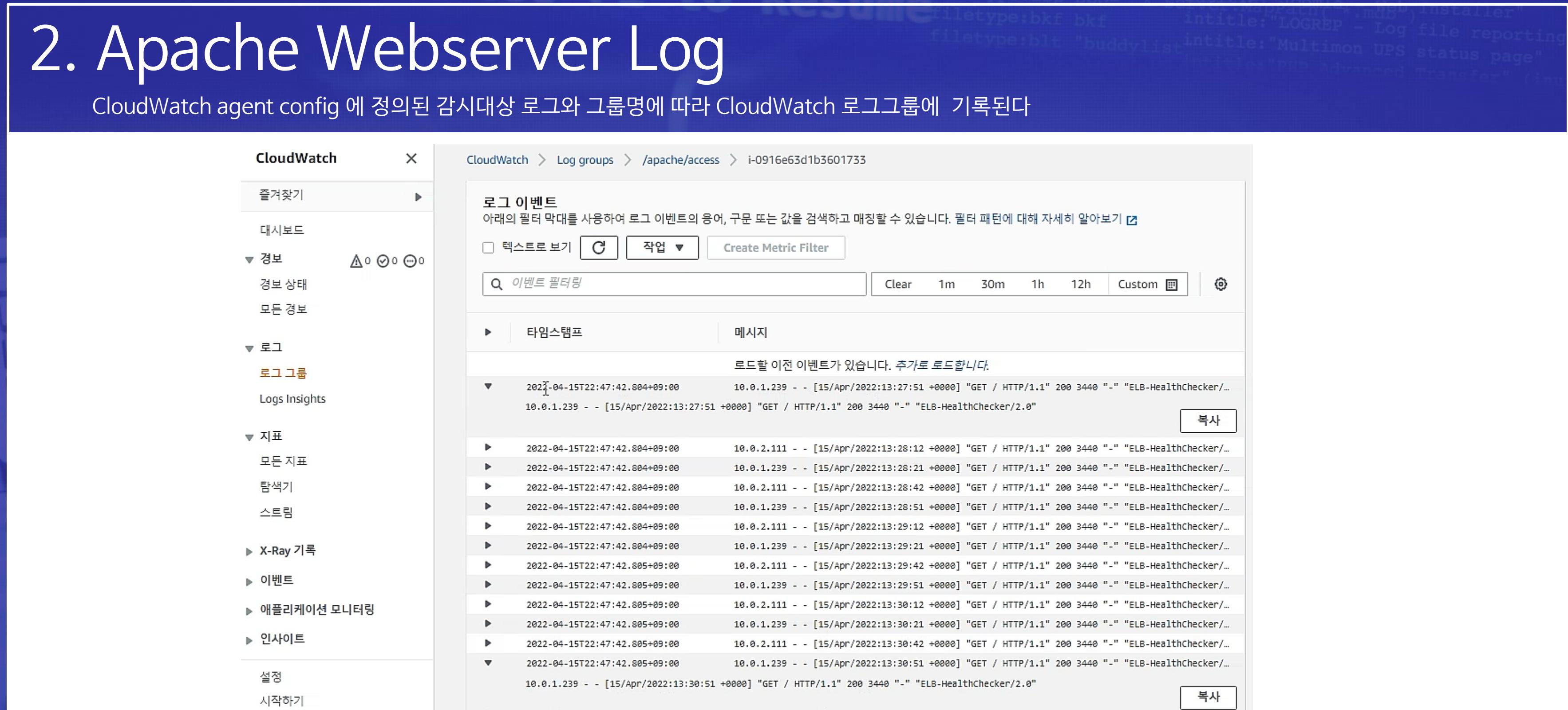
CloudWatch agent config 에 정의된 감시대상 로그와 그룹명에 따라 CloudWatch 로그그룹에 기록된다

The screenshot shows the AWS CloudWatch Log Groups interface. On the left, the navigation pane includes 'CloudWatch' (selected), 'Logs Insights', and 'Logs' (selected). Under 'Logs', there are 'Log groups' (selected) and 'Logs'. The main area displays the '/apache/access' log group under 'Log groups > /apache/access'. The interface includes a search bar, a 'Logs Insights에서 보기' button, and a 'Search log group' button. Below the search bar, there's a section for '로그 그룹 세부 정보' (Log Group Details) with columns for '보존' (Retention), '생성 시간' (Last Activity), '저장된 바이트' (Bytes Saved), and 'ARN'. The details for the '/apache/access' group show '만기 없음' (No Expiry), '1분 전' (Last Activity 1 minute ago), '0' bytes saved, and ARN: arn:aws:logs:ap-northeast-2:367784630612:log-group:/apache/access:. The bottom section shows the '로그 스트림' (Log Stream) list with two entries: 'Log stream' and 'Last event time'. The log streams are 'i-0916e63d1b3601733' (last event time: 2022-04-15 22:47:43) and 'i-0ce67ca33ca5c2ba2' (last event time: 2022-04-15 22:47:43).

# 결과 Webserver

## 2. Apache Webserver Log

CloudWatch agent config에 정의된 감시대상 로그와 그룹명에 따라 CloudWatch 로그그룹에 기록된다



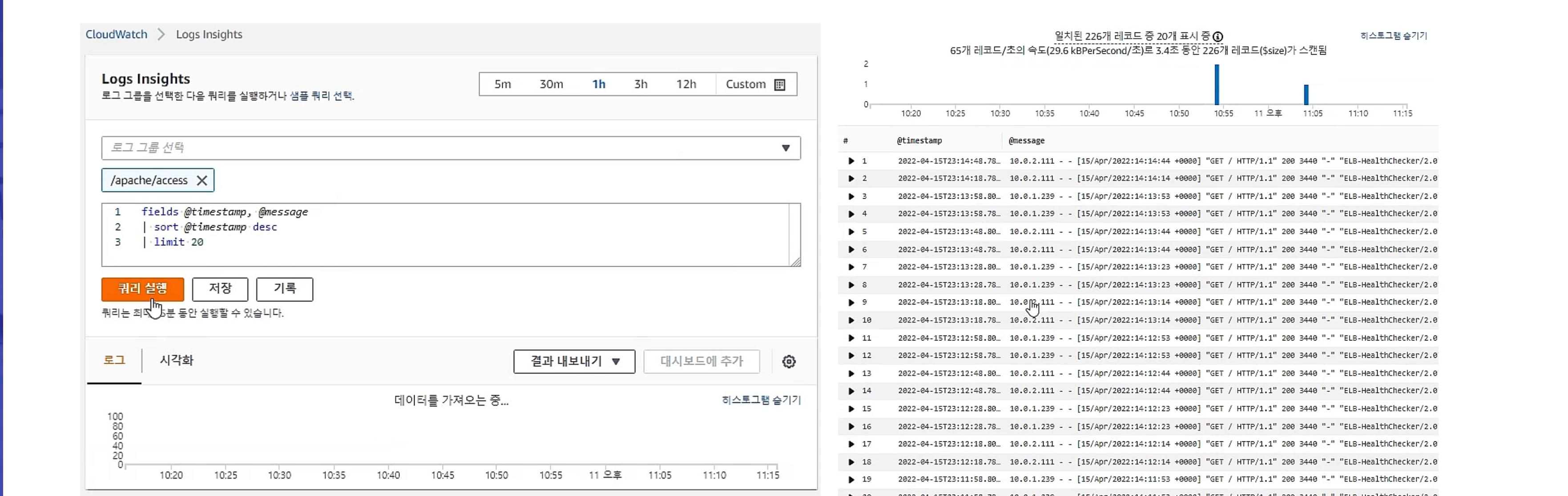
The screenshot shows the AWS CloudWatch Logs interface. On the left, the navigation pane includes 'CloudWatch' (selected), 'Logs Insights', and various monitoring and event logs sections like 'Logs Metrics', 'Logs Metrics Insights', 'Logs Metrics Filter', 'Logs Metrics Filter Insights', 'Logs Metrics Filter Metrics', and 'Logs Metrics Filter Metrics Insights'. The main content area displays the '/apache/access' log group under 'Log groups'. The log entries are timestamped and show ELB-HealthChecker requests. A search bar at the top allows filtering by event type.

Timestamp	Event Details
2022-04-15T22:47:42.804+09:00	10.0.1.239 - - [15/Apr/2022:13:27:51 +0000] "GET / HTTP/1.1" 200 3440 "-" "ELB-HealthChecker/2.0"
2022-04-15T22:47:42.804+09:00	10.0.1.239 - - [15/Apr/2022:13:27:51 +0000] "GET / HTTP/1.1" 200 3440 "-" "ELB-HealthChecker/2.0"
2022-04-15T22:47:42.804+09:00	10.0.2.111 - - [15/Apr/2022:13:28:12 +0000] "GET / HTTP/1.1" 200 3440 "-" "ELB-HealthChecker/2.0"
2022-04-15T22:47:42.804+09:00	10.0.1.239 - - [15/Apr/2022:13:28:21 +0000] "GET / HTTP/1.1" 200 3440 "-" "ELB-HealthChecker/2.0"
2022-04-15T22:47:42.804+09:00	10.0.2.111 - - [15/Apr/2022:13:28:42 +0000] "GET / HTTP/1.1" 200 3440 "-" "ELB-HealthChecker/2.0"
2022-04-15T22:47:42.804+09:00	10.0.1.239 - - [15/Apr/2022:13:28:51 +0000] "GET / HTTP/1.1" 200 3440 "-" "ELB-HealthChecker/2.0"
2022-04-15T22:47:42.804+09:00	10.0.2.111 - - [15/Apr/2022:13:29:12 +0000] "GET / HTTP/1.1" 200 3440 "-" "ELB-HealthChecker/2.0"
2022-04-15T22:47:42.804+09:00	10.0.1.239 - - [15/Apr/2022:13:29:21 +0000] "GET / HTTP/1.1" 200 3440 "-" "ELB-HealthChecker/2.0"
2022-04-15T22:47:42.804+09:00	10.0.2.111 - - [15/Apr/2022:13:29:42 +0000] "GET / HTTP/1.1" 200 3440 "-" "ELB-HealthChecker/2.0"
2022-04-15T22:47:42.804+09:00	10.0.1.239 - - [15/Apr/2022:13:29:51 +0000] "GET / HTTP/1.1" 200 3440 "-" "ELB-HealthChecker/2.0"
2022-04-15T22:47:42.804+09:00	10.0.2.111 - - [15/Apr/2022:13:30:12 +0000] "GET / HTTP/1.1" 200 3440 "-" "ELB-HealthChecker/2.0"
2022-04-15T22:47:42.804+09:00	10.0.1.239 - - [15/Apr/2022:13:30:21 +0000] "GET / HTTP/1.1" 200 3440 "-" "ELB-HealthChecker/2.0"
2022-04-15T22:47:42.804+09:00	10.0.2.111 - - [15/Apr/2022:13:30:42 +0000] "GET / HTTP/1.1" 200 3440 "-" "ELB-HealthChecker/2.0"
2022-04-15T22:47:42.804+09:00	10.0.1.239 - - [15/Apr/2022:13:30:51 +0000] "GET / HTTP/1.1" 200 3440 "-" "ELB-HealthChecker/2.0"

# 결과 Webserver

## 2. Apache Webserver Log

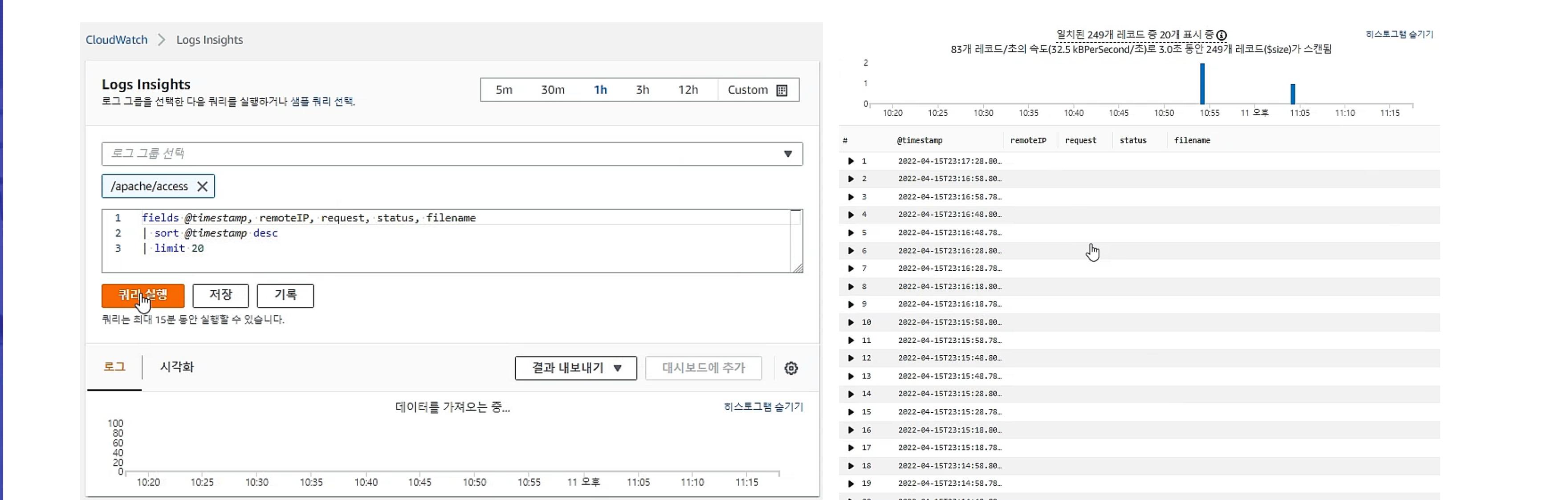
CloudWatch – log Insights 를 통해 분석 → 전체로그 조회



# 결과 Webserver

## 2. Apache Webserver Log

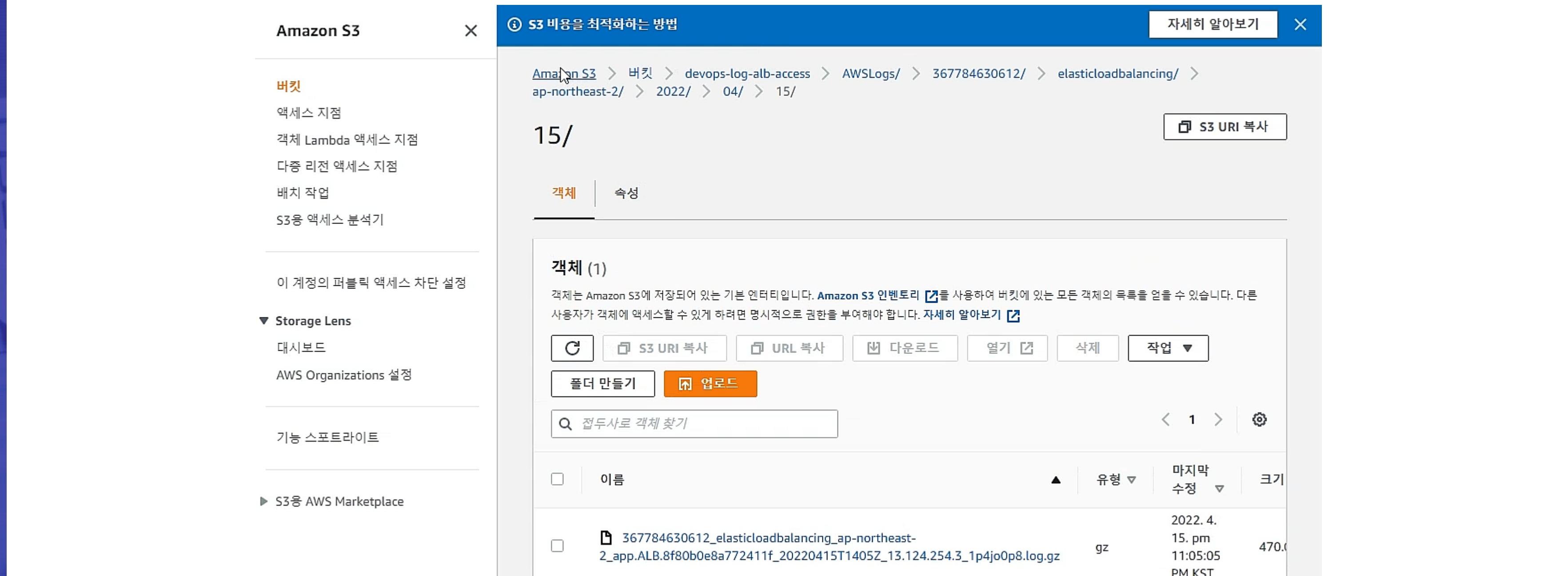
CloudWatch – log Insights 를 통해 분석 → 시간순서로 조회



# 결과 Application Load Balancer

## 3. ALB Log

S3 버킷에 쌓이는 로그 확인



The screenshot shows the AWS S3 console interface. On the left, there's a sidebar with navigation links like 'Amazon S3', '버킷' (Bucket), '액세스 지점' (Access Point), '객체 Lambda 액세스 지점' (Object Lambda Access Point), '다중 리전 액세스 지점' (Multi-Region Access Point), '배치 작업' (Batch Job), and 'S3용 액세스 분석기' (Amazon S3 Analytics Access Analyzer). Below these are sections for '이 계정의 퍼블릭 액세스 차단 설정' (Public Access Block) and 'Storage Lens' (with '대시보드' and 'AWS Organizations 설정' options). Further down are '기능 스포트라이트' (Feature Spotlight) and a link to 'S3용 AWS Marketplace'.

The main content area has a blue header bar with the text 'Amazon S3 > 버킷 > devops-log-alb-access > AWSLogs/ > 367784630612/ > elasticloadbalancing/ > ap-northeast-2/ > 2022/ > 04/ > 15/'. To the right of the header is a '자세히 알아보기' (Learn More) button and a close 'X' button.

The main content area displays a list of objects under the date '15/'. There are two tabs at the top of this list: '객체' (Objects) and '속성' (Attributes). The '객체' tab is selected. Below it, there's a section titled '객체 (1)' (Object (1)). It contains a message about the object being stored in Amazon S3 and the need for explicit permissions for access. It includes a 'Amazon S3 인벤토리' (Amazon S3 Inventory) link and a '자세히 알아보기' (Learn More) link. Below this are several buttons: 'C' (Create), 'S3 URI 복사' (Copy S3 URI), 'URL 복사' (Copy URL), '다운로드' (Download), '열기' (Open), '삭제' (Delete), and '작업' (Actions). A '업로드' (Upload) button is highlighted in orange. There's also a '폴더 만들기' (Create Folder) button.

At the bottom of the object list, there's a search bar with the placeholder '접두사로 객체 찾기' (Search by prefix) and a set of navigation controls: '<' (Previous), '1' (Current page), '>' (Next), and a magnifying glass icon.

The table below lists the single object:

선택	이름	유형	마지막 수정	크기
<input type="checkbox"/>	367784630612_elasticloadbalancing_ap-northeast-2_app.ALB.8f80b0e8a772411f_20220415T1405Z_13.124.254.3_1p4jo0p8.log.gz	gz	2022. 4. 15. pm 11:05:05 PM KST	470.0

# 결과 Application Load Balancer

## 3. ALB Log – Athena Query

S3 버킷에 쌓이는 로그 확인 → AWS Athena에서 조회 및 분석 → 클라이언트 IP 별 접속 카운트

Amazon Athena > 쿼리 편집기

편집기 | 최근 쿼리 | 저장된 쿼리 | 설정

작업 그룹: primary

데이터

데이터 원본: AwsDataCatalog

데이터베이스: albaccesslog

테이블 및 보기

테이블 (1): alb\_logs

보기 (0)

SQL 줄 7, 열 11

```

1 SELECT COUNT(request_verb) AS
2 count,
3 request_verb,
4 client_ip
5 FROM alb_logs
6 GROUP BY request_verb, client_ip
7 LIMIT 100;
  
```

다시 실행 | 취소 | 저장 | 지우기 | 생성

완료됨  
대기열 시간: 0.138 초 | 실행 시간: 0.686 초 | 스캔한 데이터: 4.79 KB

결과 (6)

복사 | 결과 다운로드

#	count	request_verb	client_ip
1	1	POST	95.214.235.205
2	14	GET	2.56.57.114
3	2	-	27.124.5.118
4	3	GET	125.186.195.115
5	2	GET	95.214.235.205
6	4	GET	27.124.5.118

# 결과 Application Load Balancer

## 3. ALB Log – Athena Query

S3 버킷에 쌓이는 로그 확인 → AWS Athena에서 조회 및 분석 → safari 브라우저 방문자 URL

The screenshot shows the Amazon Athena console interface. In the top navigation bar, it says "Amazon Athena > 쿼리 편집기". Below the navigation, there are tabs for "편집기" (selected), "최근 쿼리", "저장된 쿼리", and "설정". The "작업 그룹" dropdown is set to "primary".

**데이터** section:

- "데이터 원본": "AwsDataCatalog"
- "데이터베이스": "albaccesslog"

**쿼리 편집기** section (SQL query area):

```

1 SELECT request_url
2 FROM alb_logs
3 WHERE user_agent LIKE '%Safari%'
4 LIMIT 10;
    
```

**SQL** section: 쿼리 4, 열 10

**실행** button is highlighted with a mouse cursor.

**결과** section: (0) rows

Bottom message: 결과 없음  
결과 보기에 대한 쿼리 실행

The screenshot shows the results of the executed query. At the top, there are buttons for "다시 실행", "취소", "저장", "지우기", and "생성".

**결과 (7)** section:

#	request_url
1	http://www.ncpa.tk:80/
2	http://www.ncpa.tk:80/
3	http://www.ncpa.tk:80/manual
4	http://13.124.254.3:80/.env
5	http://google.com:80/f5F01DP05Te/Cvg37dv4ny76.php?info=22391
6	http://13.209.219.225:80/.env
7	http://13.209.219.225:80/

# 결과 Application Load Balancer

## 3. ALB Log – Athena Query

S3 버킷에 쌓이는 로그 확인 → AWS Athena에서 조회 및 분석 → 전체로그 조회

The screenshot shows the Amazon Athena console interface. On the left, there's a sidebar with tabs for 편집기 (Editor), 최근 쿼리 (Recent Queries), 저장된 쿼리 (Saved Queries), and 설정 (Settings). The 편집기 tab is selected. In the main area, there's a '데이터' (Data) section with dropdown menus for 데이터 원본 (AwsDataCatalog) and 데이터베이스 (albaccesslog). Below it is a '테이블 및 보기' (Tables and Views) section with a '생성' (Create) button and a search bar for 테이블 및 보기 필터링 (Table and View Filtering). A table named 'alb\_logs' is listed under 테이블 (1). At the bottom, there's a results pane for '쿼리 5' (Query 5) which contains the following SQL:

```

1 SELECT *
2 FROM alb_logs
3 LIMIT 100;
    
```

Below the SQL is a 'SQL' section with a ' 다시 실행' (Run Again) button and other options like 취소 (Cancel), 저장 (Save), 지우기 (Delete), and 생성 (Create). The results pane shows a summary: 완료됨 (Completed) with 대기열 시간: 0.118 초 (Waiting time: 0.118 s), 실행 시간: 0.646 초 (Execution time: 0.646 s), and 스캔한 데이터: 4.79 KB (Scanned data: 4.79 KB). The results table has columns: #, type, time, elb, client\_ip, and client\_port.

This screenshot shows the results of the Athena query from the previous screen. The top header indicates the query is 완료됨 (Completed) with a execution time of 0.646초 and scanned data of 4.79KB. The results table has 26 rows and the following columns: #, type, time, elb, client\_ip, and client\_port. The data shows a series of HTTP requests from various client IPs to an ALB instance.

#	type	time	elb	client_ip	client_port
3	http	2022-04-15T14:21:10.588649Z	app/ALB/8f80b0e8a772411f	2.56.57.114	50332
4	http	2022-04-15T14:21:11.839678Z	app/ALB/8f80b0e8a772411f	2.56.57.114	50789
5	http	2022-04-15T14:21:13.024787Z	app/ALB/8f80b0e8a772411f	2.56.57.114	51143
6	http	2022-04-15T14:21:16.474917Z	app/ALB/8f80b0e8a772411f	2.56.57.114	52382
7	http	2022-04-15T14:21:17.332579Z	app/ALB/8f80b0e8a772411f	2.56.57.114	52614
8	http	2022-04-15T14:21:14.875036Z	app/ALB/8f80b0e8a772411f	2.56.57.114	51841
9	http	2022-04-15T14:24:45.196142Z	app/ALB/8f80b0e8a772411f	2.56.57.114	51912
10	http	2022-04-15T14:24:51.952806Z	app/ALB/8f80b0e8a772411f	2.56.57.114	53884
11	http	2022-04-15T14:24:23.278111Z	app/ALB/8f80b0e8a772411f	27.124.5.118	43112
12	http	2022-04-15T14:24:41.218369Z	app/ALB/8f80b0e8a772411f	27.124.5.118	45100
13	http	2022-04-15T14:24:43.091893Z	app/ALB/8f80b0e8a772411f	2.56.57.114	51297
14	http	2022-04-15T14:24:47.346454Z	app/ALB/8f80b0e8a772411f	2.56.57.114	52593

# 결과 Application Load Balancer

## 3. ALB Log – Athena Query

S3 버킷에 쌓이는 로그 확인 → AWS Athena에서 조회 및 분석 → 시간대별 조회

The screenshot shows the AWS Athena interface. At the top, there are six query tabs labeled 1 through 6. Query 6 is selected and contains the following SQL code:

```

1 SELECT client_ip, sum(received_bytes)
2 FROM alb_logs
3 WHERE parse_datetime(time, 'yyyy-MM-dd''T''HH:mm:ss.SSSSSS''Z')
4     BETWEEN parse_datetime('2022-04-15-14:00:00', 'yyyy-MM-dd-HH:mm:ss')
5     AND parse_datetime('2022-04-15-15:10:00', 'yyyy-MM-dd-HH:mm:ss')
6 GROUP BY client_ip;
    
```

Below the code, it says "SQL 줄 4, 열 43". There are buttons for "다시 실행" (Run again), "취소" (Cancel), "저장" (Save), "지우기" (Delete), and "생성" (Create). A status bar at the bottom indicates the query was completed successfully: "완료됨" (Completed), "대기열 시간: 0.172 초" (Queue wait time: 0.172 s), "실행 시간: 0.714 초" (Execution time: 0.714 s), and "스캔한 데이터: 4.79 KB" (Scanned data: 4.79 KB).

The results section is titled "결과 (4)". It shows a table with two columns: "client\_ip" and "\_col1". The data is as follows:

#	client_ip	_col1
1	95.214.235.205	775
2	27.124.5.118	547
3	2.56.57.114	3286
4	125.186.195.115	2375

# 결과 Application Load Balancer

## 3. ALB Log – Athena Query Download

S3 버킷에 쌓이는 로그 확인 → AWS Athena에서 조회 및 분석 결과 → 다운로드

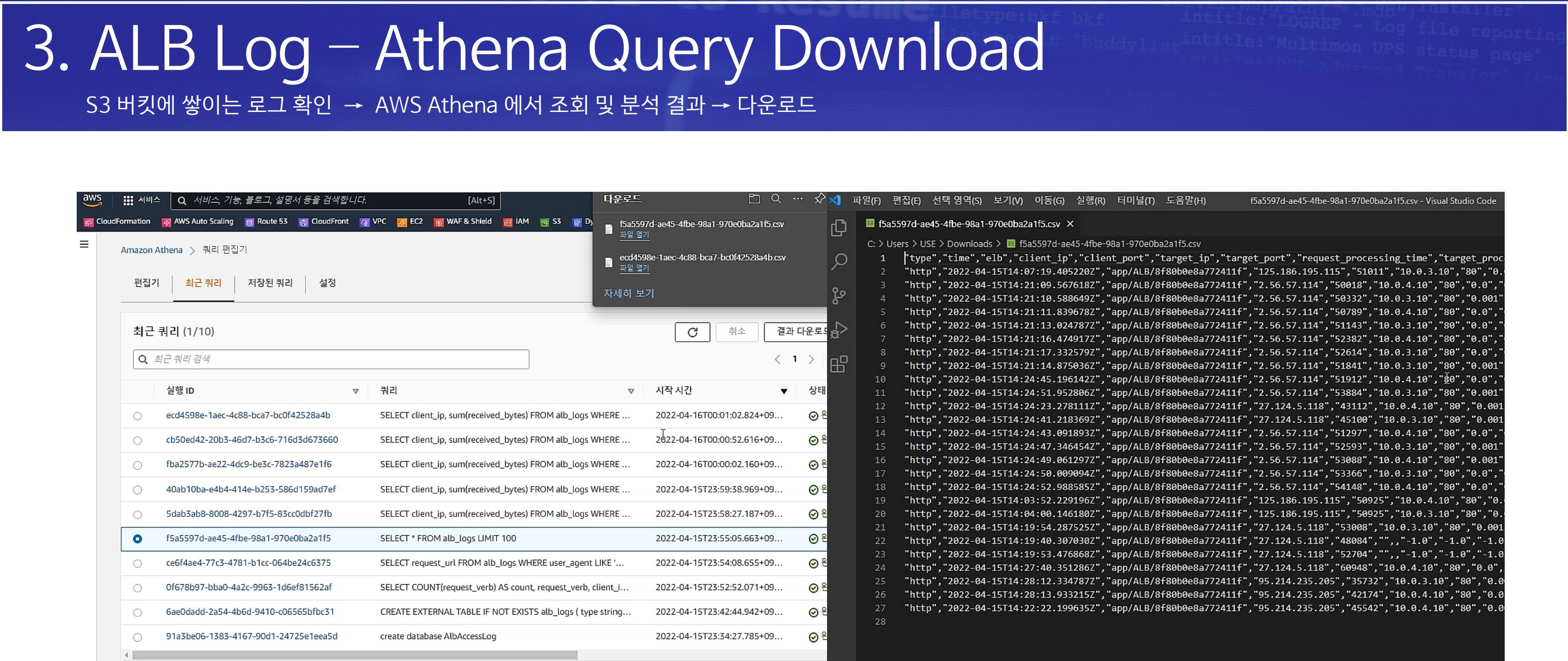
The screenshot shows the AWS Athena console interface. On the left, there's a sidebar with various AWS services like CloudFormation, AWS Auto Scaling, Route 53, CloudFront, VPC, EC2, WAF & Shield, IAM, and S3. The main area is titled "Amazon Athena > 쿼리 편집기". It has tabs for "편집기", "최근 쿼리" (which is selected), "저장된 쿼리", and "설정". Below these tabs, there's a search bar labeled "최근 쿼리 검색". A table titled "최근 쿼리 (1/10)" lists ten recent queries. The first query, "ecd4598e-1aec-4c88-bca7-bc0f42528a4b", is highlighted. To the right of the table, a modal window titled "다운로드" is open, showing the file path "C:/Users/USE/Downloads/ecd4598e-1aec-4c88-bca7-bc0f42528a4b.csv" and the contents of the CSV file:

client_ip	_col1
"95.214.235.205"	"775"
"27.124.5.118"	"547"
"2.56.57.114"	"3286"
"125.186.195.115"	"2375"

# 결과 Application Load Balancer

# 3. ALB Log – Athena Query Download

S3 버킷에 쌓이는 로그 확인 → AWS Athena에서 조회 및 분석 결과 → 다운로드



# 결과 Application Load Balancer

## 4. ALB Log – Lambda

S3 버킷에 ALB가 로그를 쌓기 시작하면 Lambda Function에 의해 CloudWatch에 기록된다

Amazon S3 > 버킷 > devops-log-alb-access > AWSLogs/ > 367784630612/ > elasticloadbalancing/ > ap-northeast-2/ > 2022/ > 04/ > 15/

15/

**캡처** 속성

**캡처 (8)**

캡처는 Amazon S3에 저장되어 있는 기본 엔터티입니다. [Amazon S3 인벤토리](#) 를 사용하여 버킷에 있는 모든 객체의 목록을 얻을 수 있습니다. 다른 사용자가 객체에 액세스할 수 있게 하려면 명시적으로 권한을 부여해야 합니다. [자세히 알아보기](#)

**C** [S3 URI 복사](#) [URL 복사](#) [다운로드](#) [열기](#) [삭제](#) [작업 ▾](#)

[폴더 만들기](#) [업로드](#)

접두사로 객체 찾기

이름	유형	마지막 수정
367784630612_elasticloadbalancing_ap-northeast-2_app(ALB.8f80b0e8a772411f_20220415T1405Z_13.124.254.3_1p4jo0p8.log.gz)	gz	2022. 4. 15. pm 11:05:05 PM KST
367784630612_elasticloadbalancing_ap-northeast-2_app(ALB.8f80b0e8a772411f_20220415T1410Z_12.124.254.2_1n00vpm1.log.gz)	gz	2022. 4. 15. pm 11:10:05

CloudWatch > Log groups > /aws/lambda/alblog-cloudwatch

/aws/lambda/alblog-cloudwatch

[작업 ▾](#) [Logs Insights에서 보기](#) [Search log group](#)

**로그 그룹 세부 정보**

보존	생성 시간	저장된 바이트	ARN
만기 없음	1분 전	-	arn:aws:logs:ap-northeast-2:367784630612:log-group:/aws/lambda/alblog-cloudwatch:*
KMS 키 ID	지표 필터	구독 필터	기여자 인사이트 규칙
-	0	0	-

**로그 스트림** | 지표 필터 | 구독 필터 | 기여자 인사이트 | 태그

**로그 스트림 (1)**

[C](#) [삭제](#) [로그 스트림 생성](#) [Search all](#)

로그 스트림 필터링 또는 접두사 검색 시도

Log stream	Last event time
2022/04/15/[LATEST]01e78a81f6d34f4b8a6c0f72951...	2022-04-16 00:55:09 (UTC+09:00)

# 결과 Application Load Balancer

## 4. ALB Log – Lambda

CloudWatch에 기록된 로그 이벤트의 확인, Apache Web Server log 와 동일하게 CloudWatch – log Insights 를 통해 분석이 가능 합니다.

The screenshot shows the AWS CloudWatch Log Insights interface. The navigation path is: CloudWatch > Log groups > /aws/lambda/alblog-cloudwatch > 2022/04/15/[LATEST]01e78a81f6d34f4b8a6c0f72951cf0b0. The main area is titled "로그 이벤트" (Log Events). It includes a search bar for "이벤트 필터링" (Event Filtering), time range buttons for "Clear", "1m", "30m", "1h", "12h", "Custom", and a refresh button. A "Create Metric Filter" button is also present. The results table has columns for "타임스탬프" (Timestamp) and "메시지" (Message). The timestamp column shows log entries for April 16, 2022, at 00:55:09. The message column contains event details like RequestId, Version, and Duration. A note at the bottom says "현재 최신 이벤트가 없습니다. 자동 재시도를 일시 중지하였습니다." (No latest events available. Auto-resume is stopped.)

CloudWatch > Log groups > /aws/lambda/alblog-cloudwatch > 2022/04/15/[LATEST]01e78a81f6d34f4b8a6c0f72951cf0b0

로그 이벤트

아래의 필터 막대를 사용하여 로그 이벤트의 용어, 구문 또는 값을 검색하고 매칭할 수 있습니다. 필터 패턴에 대해 자세히 알아보기 [\[x\]](#)

텍스트로 보기  [작업](#) [Create Metric Filter](#)

이벤트 필터링 [Clear](#) [1m](#) [30m](#) [1h](#) [12h](#) [Custom](#)

▶ 타임스탬프 메시지

현재 이전 이벤트가 없습니다. 재시도

▶ 2022-04-16T00:55:09.246+09:00	START RequestId: 03387f03-4e0a-4516-a9ad-caf084f6428c Version: \$LATEST
▶ 2022-04-16T00:55:09.250+09:00	END RequestId: 03387f03-4e0a-4516-a9ad-caf084f6428c
▶ 2022-04-16T00:55:09.250+09:00	REPORT RequestId: 03387f03-4e0a-4516-a9ad-caf084f6428c Duration: 2.34 ms Billed Duration: 3 ms ...

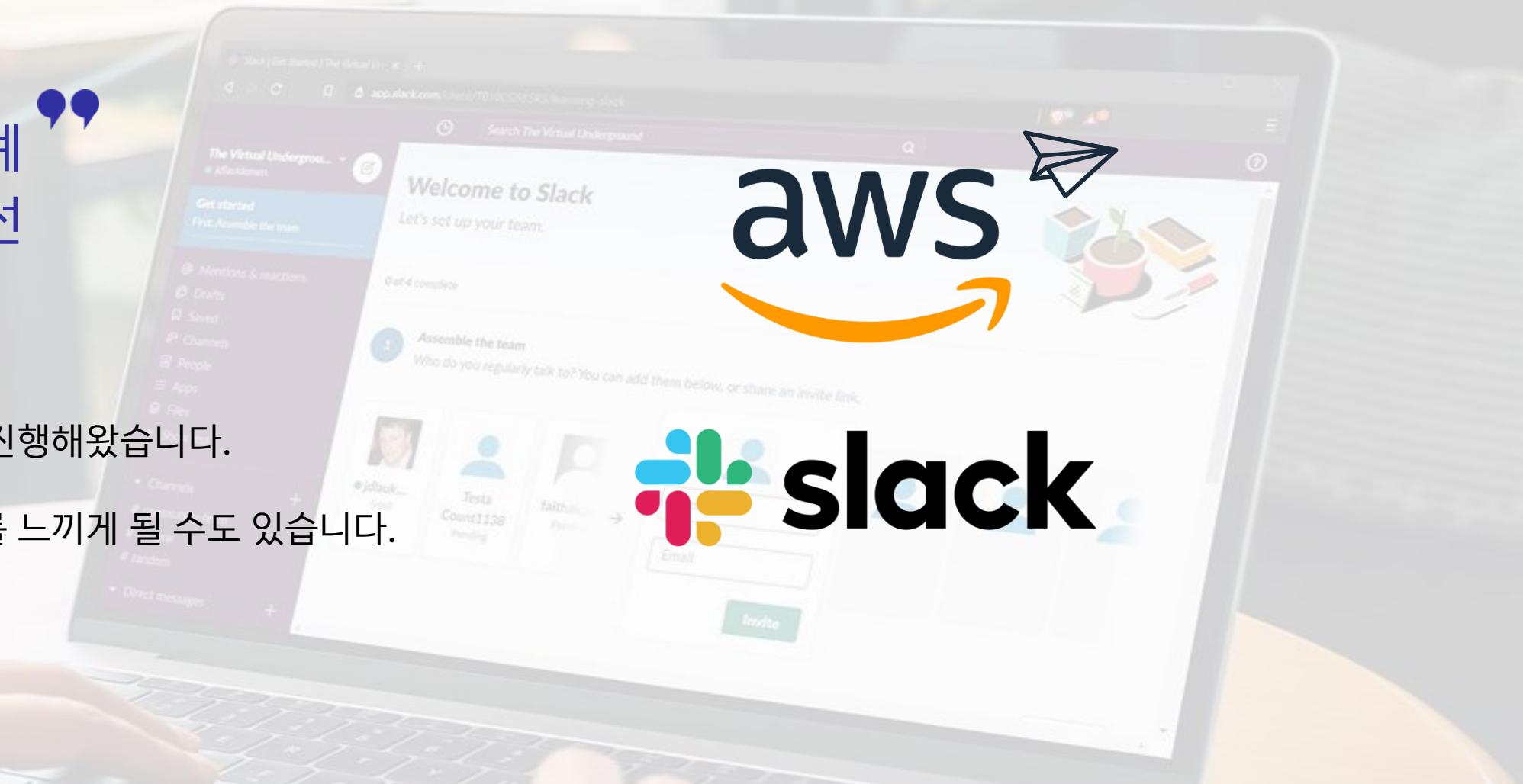
현재 최신 이벤트가 없습니다. 자동 재시도를 일시 중지하였습니다.

# 향후 계획

“ 수시로 모니터링 하는 것에 대한 한계  
SLACK 알림을 받도록 개선 ”

현재까지의 결과 :

Amazon CloudWatch 기능과 Lambda, Athena 등  
수집과 모니터링 도구를 활용해 프로젝트 모니터링을 진행해왔습니다.  
하지만 향후 점점 늘어나는 프로젝트마다  
대시보드를 세팅하고 수시로 모니터링하는 것에 한계를 느끼게 될 수도 있습니다.



## 개선점

- 관리자의 업무 효율을 높이는 모니터링 방법으로 개선
- 전 세계적으로 많이 쓰이는 툴을 도입
- 복잡하지 않은 구성으로 개선 할 것

## 특징

- AWS SNS, Chatbot 서비스를 활용
- Alert 과 메시지를 활용하기 위해 SLACK 연동

# 향후계획 - 사용될 기술

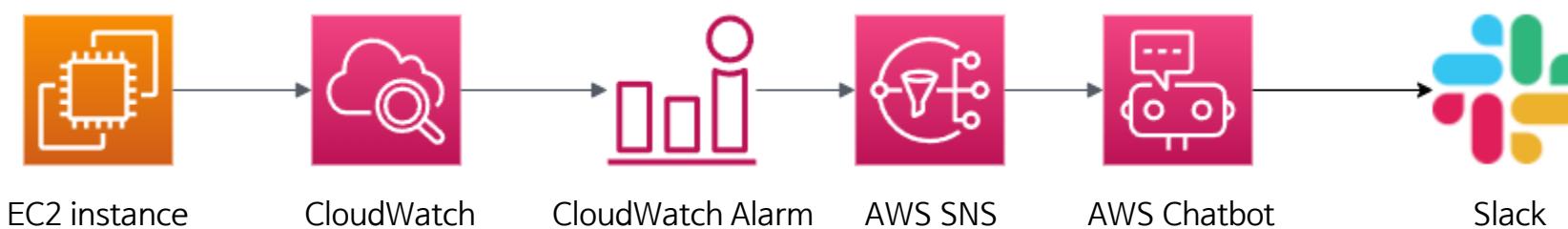
Amazon Web Service  
Management & Governance  
Application Integration



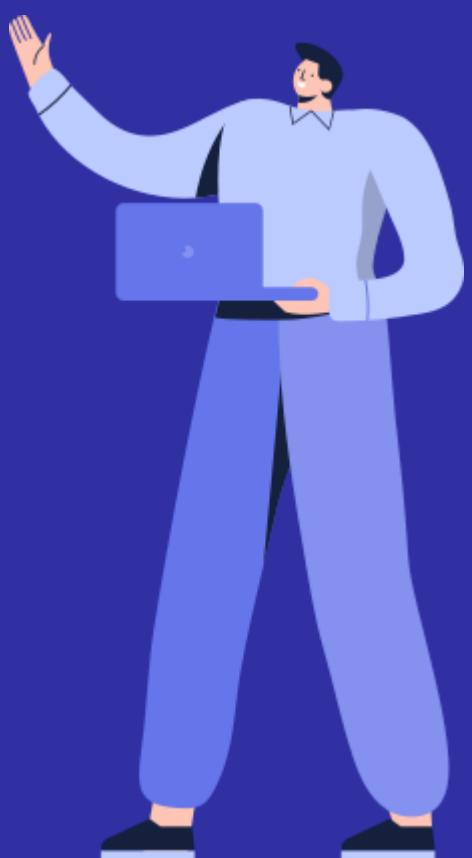
Slack  
business communication



## 기술 도입 테스트 시나리오



EC2 의 CPU Utilization 을 반영한 CloudWatch 경보 임계값을 설정하고  
이상 징후 발생 시 AWS Simple Notification Service , Chatbot 을 통해  
AWS 와 연동된 Slack workspace 로 관리자에게 알림을 보낸다.



# 향후계획 - 테스트

**Amazon Web Service Management & Governance Application Integration**

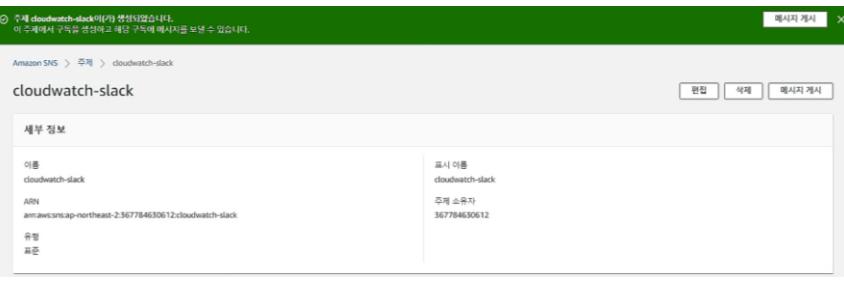
**AWS Chatbot**

**AWS Simple Notification Service**

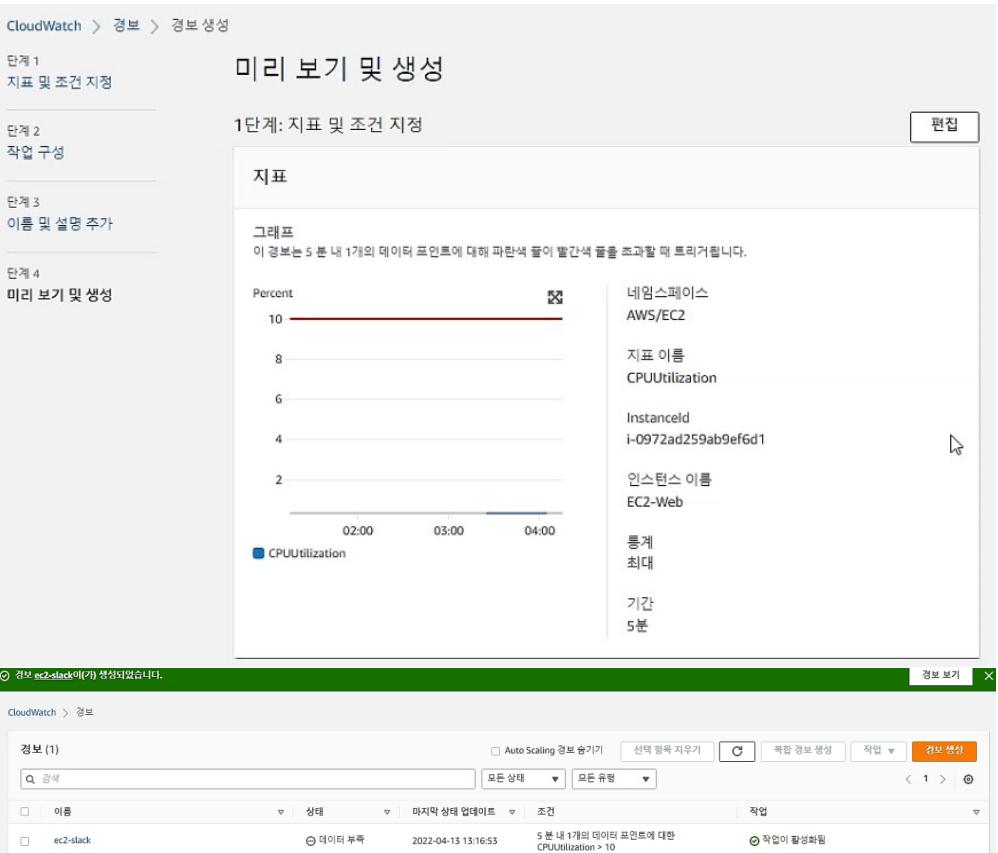
**Slack business communication**

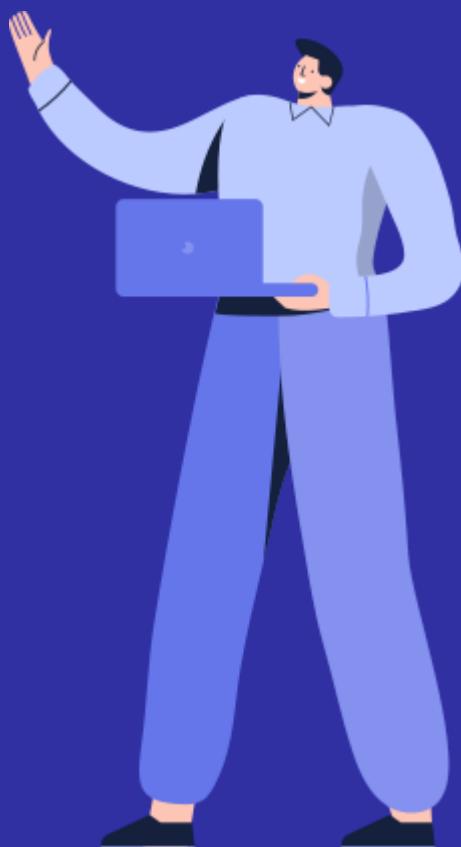


## SNS – 주제선정



## CloudWatch – 경보 생성 (EC2 CPU Utilization 10%, 5분)





# 향후계획 - 테스트

**Amazon Web Service Management & Governance Application Integration**

**AWS Chatbot**



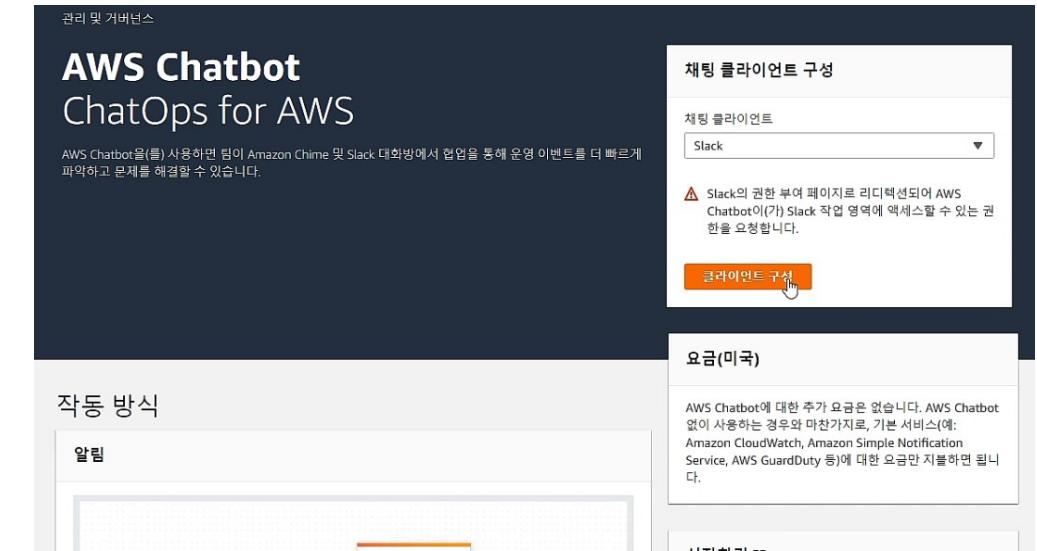
**AWS Simple Notification Service**



**Slack business communication**



## Chatbot – 클라이언트 구성 (slack 연동)



**AWS Chatbot**  
ChatOps for AWS

AWS Chatbot을(를) 사용하면 팀이 Amazon Chime 및 Slack 대화방에서 협업을 통해 운영 이벤트를 더 빠르게 파악하고 문제를 해결할 수 있습니다.

채팅 클라이언트  
Slack

Slack의 권한 부여 페이지로 리디렉션되어 AWS Chatbot이(가) Slack 작업 영역에 액세스할 수 있는 권한을 요청합니다.

클라이언트 구성

작동 방식  
알림

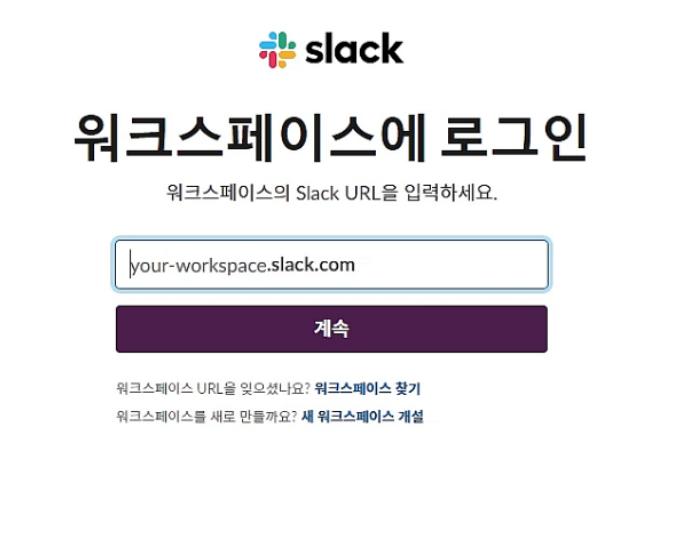
요금(미국)  
AWS Chatbot에 대한 추가 요금은 없습니다. AWS Chatbot 없이 사용하는 경우와 마찬가지로, 기본 서비스(예: Amazon CloudWatch, Amazon Simple Notification Service, AWS GuardDuty 등)에 대한 요금만 지불하면 됩니다.

시작하기

Slack에서 AWS Chatbot이(가) 승인되었습니다.  
Slack에 알림을 보내려면 먼저 채널을 하나 이상 구성해야 합니다.

Slack WorkSpace: AWS 개인 프로젝트

WorkSpace 세부 정보  
WorkSpace ID: T03B3A17MSS  
구성된 채널  
구성 없음  
구성이 없습니다. 새 채널 구성을(를) 선택하거나 다음을 사용하여 구성을 생성하십시오. AWS CloudFormation 템플릿



**slack**

### 워크스페이스에 로그인

워크스페이스의 Slack URL을 입력하세요.

계속

워크스페이스 URL을 찾으셨나요? 워크스페이스 찾기  
워크스페이스를 새로 만들까요? 새 워크스페이스 개설

**aws** ⇔ **slack**

**AWS Chatbot에서 AWS 개인 프로젝트 Slack 워크스페이스에 액세스하기 위해 권한을 요청합니다.**

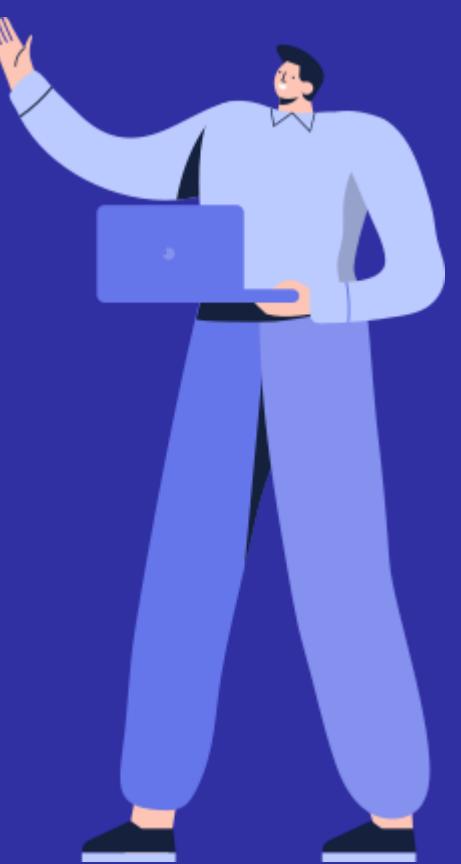
**AWS Chatbot에서 볼 수 있는 항목은 무엇인가요?**

- 채널과 대화에 관한 콘텐츠와 정보
- 내 워크스페이스에 관한 콘텐츠와 정보

**AWS Chatbot에서 무엇을 할 수 있나요?**

- 채널과 대화에서 작업 수행

취소 허용



# 향후계획 - 테스트

# Amazon Web Service Management & Governance Application Integration

AWS  
Chatbot

AWS  
Simple Notification Service

Slack  
business communication

## Chatbot – 클라이언트 구성 (slack alert channel)

AWS Chatbot > 구성된 클라이언트 > Slack WorkSpace: AWS 개인 프로젝트 > Slack 채널 구성

### Slack 채널 구성

#### 구성 세부 정보

구성 이름  
내에서 쉽게 찾을 수 있도록 구성 이름을 지정합니다. 구성은 상성한 후에는 해당 이름을 변경할 수 없습니다.

이름은 최대 128자로 지정할 수 있습니다. 유효한 문자: a-z, A-Z, 0-9 및 -\_.  
로깅 - 선택 사항  
AWS Chatbot은 사용자가 시작한 명령에 대한 각각의 AWS CloudWatch Logs에 차별로 로깅되지만, 구성에 대한 추가 로깅을 활성화할 수 있습니다. Amazon CloudWatch Logs에 로깅하는 대로 제공됩니다. 자세히 알아보기 [\[?\]](#)

Amazon CloudWatch Logs에 로그 게시

#### Slack 채널

채널 유형  
채널은 Slack 채널을 선택합니다. 프라이빗 채널을 선택하면 채널 ID를 입력합니다.

**퍼블릭**  
WorkSpace의 누구나 퍼블릭 채널을 보고 참여할 수 있습니다.

**프라이빗**  
조금 더 풍요로운 프라이빗 채널에 참여하거나 프라이빗 채널을 볼 수 있습니다.

퍼블릭 채널 이름

#### 권한

AWS Chatbot은 AWS CLI 명령 실행 및 대회장 메시지에 응답을 수행하기 위한 IAM 역할이 필요합니다. IAM 역할은 AWS Chatbot에 대한 IAM 역할 또는 사용자 역할일 수 있습니다. 두 역할 유형 모두 자발 멤버가 포함된 권한을 나타냅니다. 채널 가드 역할은 채널 멤버가 수행할 수 있는 작업을 제한합니다. 역할과 가드 역할이 함께 작동하는 방식 표시

역할 설정  
모든 멤버는 동일한 권한을 공유합니다. 채널 멤버는 해당 멤버가 사용하는 IAM 역할을 선택합니다. 사용자는 사용자 역할을 선택합니다. 사용자는 사용자 역할을 선택합니다.

**채널 IAM 역할**  
모든 멤버는 동일한 권한을 공유합니다. 채널 멤버는 해당 멤버가 사용하는 IAM 역할을 선택합니다.

**사용자 역할**  
채널 멤버는 해당 멤버가 사용하는 IAM 사용자 역할을 선택합니다.

채널 IAM 역할  
이 역할은 채널 멤버가 자신의 역할을 선택하지 않을 때 사용됩니다. 채널 가드 레일에 정한 경우 채널 멤버가 사용하는 사용자 역할에 관계 없이 멤버가 수행할 수 있는 AWS 작업을 제어합니다. 채널 멤버가 수행할 수 있는 작업은 역할 권한과 가드 레일에 따라 결정됩니다. 사용자 수준 역할의 적용 범위 표시

험프리스 사용하여 IAM 역할 생성

**Slack 채널을 구성했습니다.**  
매핑된 SNS 주제로 전송된 지원 서비스의 메시지가 선택한 Slack 채널로 전송됩니다. Slack 채널로 이동하고 채널의 [세부 정보] 화면에서 [앱 추가]를 사용하여 AWS Chatbot 앱을 추가합니다.

AWS Chatbot > 구성된 클라이언트 > Slack WorkSpace: AWS 개인 프로젝트

### Slack WorkSpace: AWS 개인 프로젝트

#### WorkSpace 세부 정보

WorkSpace ID

#### 구성된 채널 (1)

구성 이름	채널 이름	로깅 수준	역할 설정	채널 역할	가드 레일 정책	매핑된 SNS 주제
<input type="checkbox"/> cloudwatch-slack	개인프로젝트	해제	채널 역할	cloudwatch-slack-role	1	1

WorkSpace 구성 제거



# 향후계획 - 테스트

Amazon Web Service  
Management & Governance  
Application Integration

AWS Chatbot

AWS Simple Notification Service

Slack business communication

slack

### Chatbot – 클라이언트 구성 ( slack alert channel)

The screenshot shows the AWS CloudWatch Metrics Insights interface. On the left, there's a sidebar with navigation links: AWS 개인 프로젝트, Slack Connect, 더 보기, 채널 (# 개인프로젝트 selected), 디렉트 메시지, 팀원 추가, 앱, and 앱 추가. The main pane displays a Slack channel named '# 개인프로젝트'. It shows two messages from an AWS Lambda function (@aws) at 3:05 PM on April 13, 2024. The first message says '오늘 3:05 dksskd8484님에 의해 #개인프로젝트에 추가되었습니다.' The second message says '@dksskd8484 - I can run the command @aws ec2 stop-instances --instance-ids i-0972ad259ab9ef6d1 --region ap-northeast-2'. Below this, it says 'in account 367784630612 with role service-role/cloudwatch-slack-role in region Asia Pacific (Seoul)'. There's a section titled 'What would you like me to do?' with buttons for '[Run] command', '[Add] optional parameters', and '[Cancel] command'. A note says 'You can also type in the word inside the square brackets.' A message from @dksskd8484 at 3:05 PM says 'I got an exception when trying to run the command @aws ec2 stop-instances --instance-ids i-0972ad259ab9ef6d1 --region ap-northeast-2'. Below this, it says 'in account 367784630612 with role service-role/cloudwatch-slack-role in region Asia Pacific (Seoul)'. A section titled 'Exception message' shows the error message: 'You are not authorized to perform this operation. Encoded authorization failure message: yoPX2nPzdA99aM9KLI5Wl69BwOUABVL6-i1rtliBMor2AhbKYR2YLV-xCMYwRjcoeTVlaYIBfJT3pUkCobujDGBnJxm56p9qM14VUYbWL5dD93dWug6h4xAMcMVSnohtz6qCkaO090zZAAcR\_9sPT75GCdvOPCtXIEK57V8-x9Rle8ESSp8wW1pluRGh2SmntRjizKCrEf9kwvyQWXIsWbA8ZQzGuApCpeXdk6dto5WIQMSLhy6vj8vJcnE\_OXmiqXerkluuX2uwpzCskrfNLkpRB8g68kgVK3FuvxJAFqGc2\_Hge'. At the bottom, there's a message input field with '#개인프로젝트에 메시지 보내기' and a send button.



# 향후계획 - 테스트

Amazon Web Service  
Management & Governance  
Application Integration

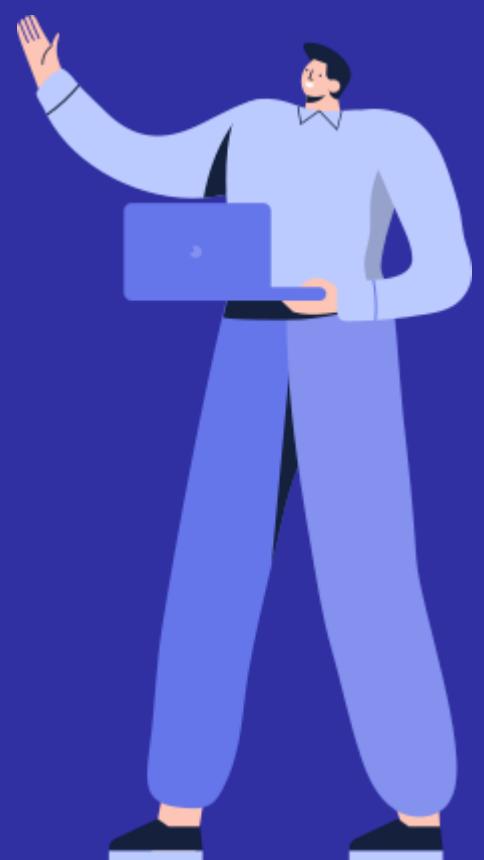
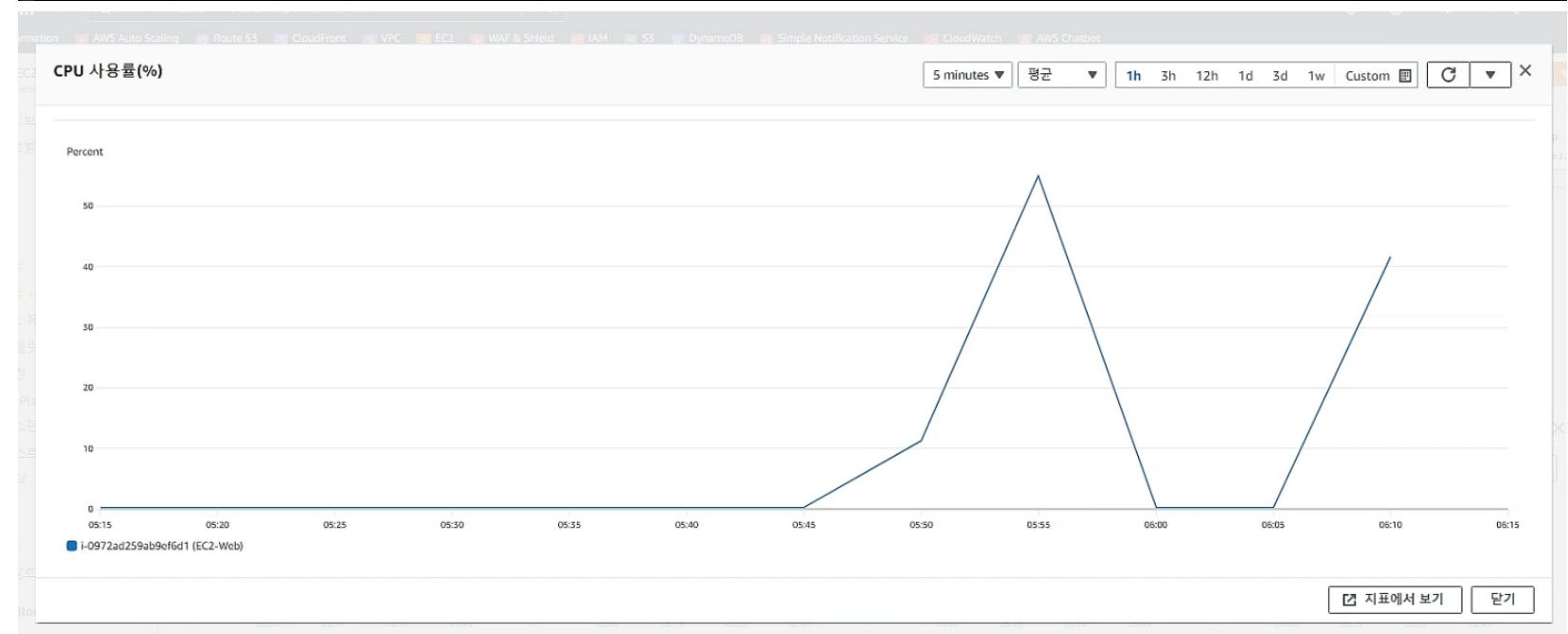


Slack  
business communication



## Event – EC2 CPU stress test

```
[ec2-user@ip-10-0-0-10 ~]$ sudo stress --cpu 1 --timeout 600  
stress: info: [861] dispatching hogs: 1 cpu, 0 io, 0 vm, 0 hdd
```



# 향후계획 - 테스트

Amazon Web Service  
Management & Governance  
Application Integration

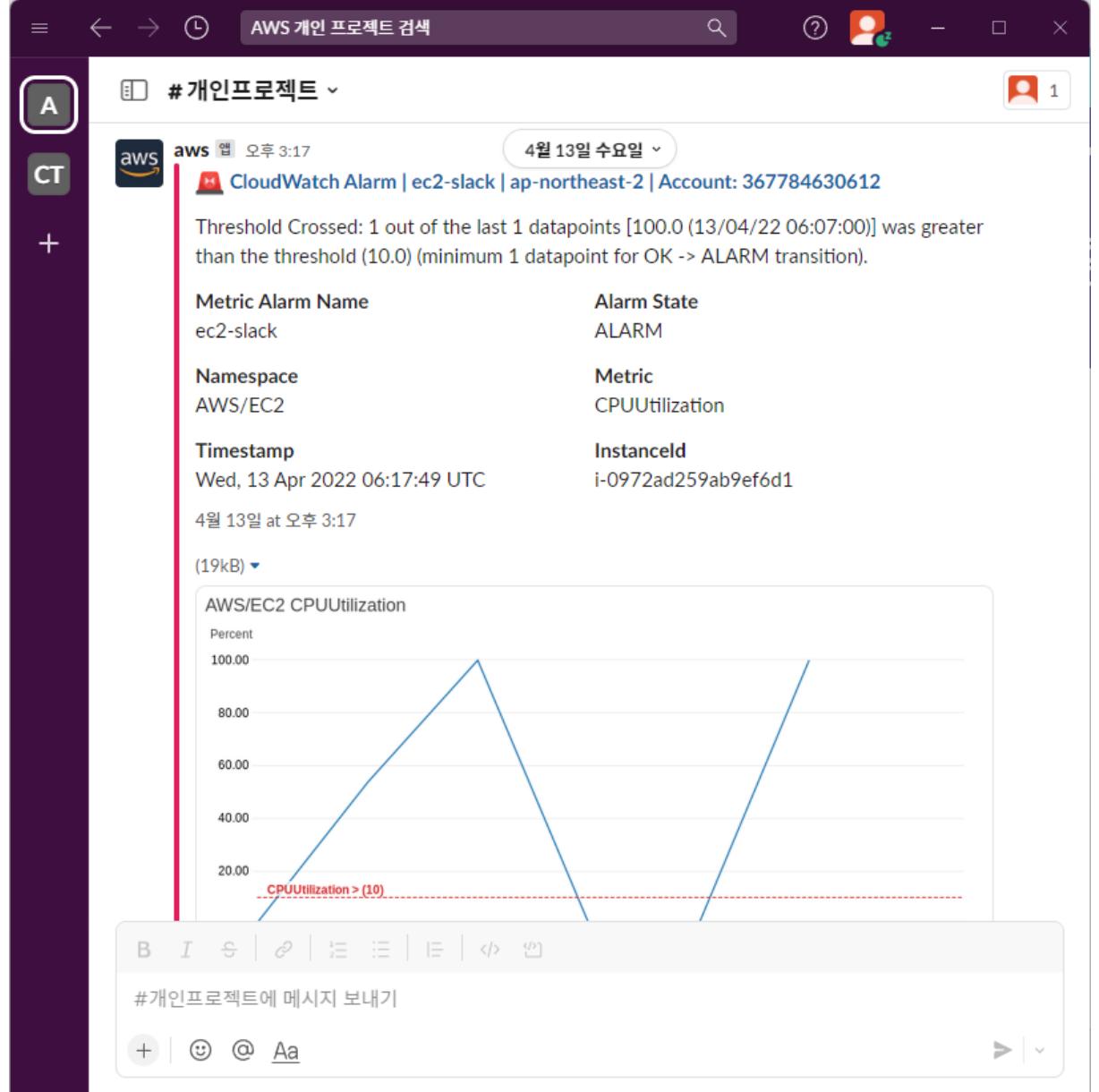
AWS Chatbot

AWS Simple Notification Service

Slack business communication



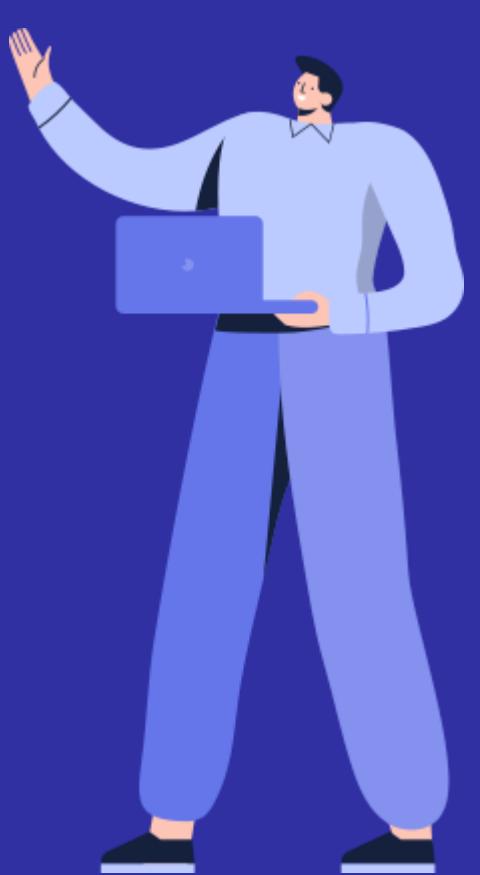
## Alert AWS → SLACK



The screenshot shows a Slack message from an AWS user (@aws) in a channel named '# 개인프로젝트'. The message details a CloudWatch Alarm triggered by an EC2 instance. The alarm summary includes:

- Metric Alarm Name: ec2-slack
- Alarm State: ALARM
- Namespace: AWS/EC2
- Metric: CPUUtilization
- Timestamp: Wed, 13 Apr 2022 06:17:49 UTC
- InstanceId: i-0972ad259ab9ef6d1

The message also includes a graph titled "AWS/EC2 CPUUtilization" showing CPU utilization over time, with a red dashed line indicating the threshold at 10.0%.



# 향후계획 – 테스트 결과

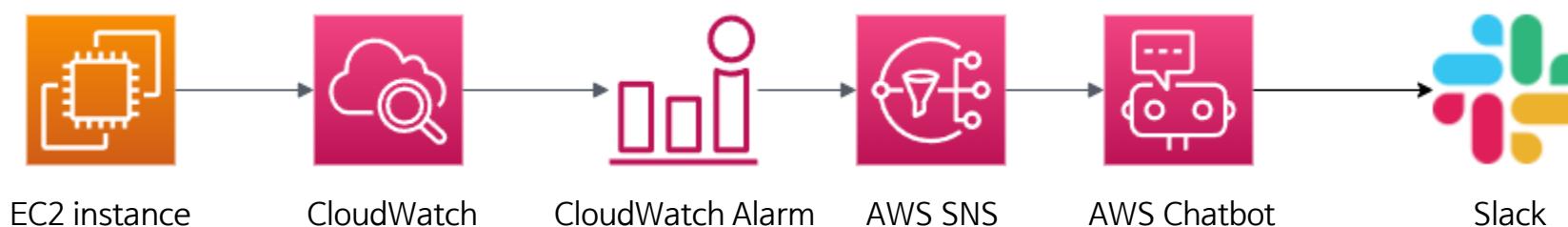
Amazon Web Service  
Management & Governance  
Application Integration



Slack  
business communication



## 기술 도입 테스트 결과



결과

AWS 와 연동된 Slack workspace 로 관리자에게 자동으로 EC2 감시 중 이상징후 발견 시 SLACK 로 알림이 오는 것을 확인 했다.

이 기능을 이용하면 향후 여러 프로젝트의 품질 향상에 도움이 될 수 있다.





Q & A

# Q & A



## 참고 문서, 웹 사이트가 있습니까?

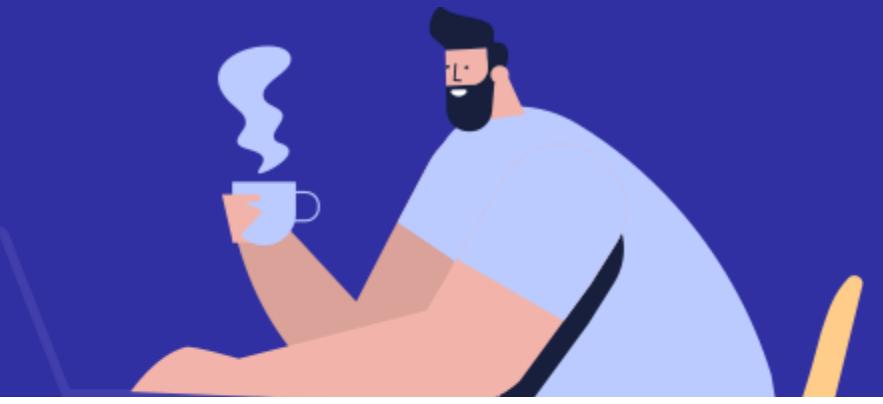
AWS 공식 문서를 활용하였습니다.

- [CloudWatch 에이전트를 사용하여 Amazon EC2 인스턴스 및 온프레미스 서버에서 지표 및 로그 수집 - Amazon CloudWatch](#)
- [CloudWatch 에이전트와 함께 사용하기 위한 IAM 역할 및 사용자 생성 - Amazon CloudWatch](#)
- [에이전트 구성을 사용하여 EC2 인스턴스에 CloudWatch 에이전트 설치 - Amazon CloudWatch](#)
- [Simplifying Apache server logs with Amazon CloudWatch Logs Insights | AWS Cloud Operations & Migrations Blog](#)
- [Querying Application Load Balancer Logs](#)

프로젝트에 사용한 Lambda 함수는 오픈소스를 활용 하였고 해당 출처는 아래와 같습니다.

- [A lambda function to stream Application Load Balancer logs dropped in S3 to CloudWatch Logs · GitHub](#)

# Q & A

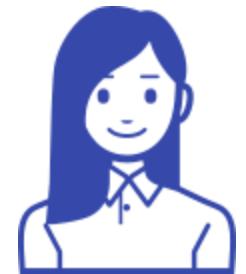


## 참고 문서, 웹 사이트가 있습니까?

본 보고서에 사용된 AWS 구성도는 아래 웹 사이트의 기술로 제작한 것 입니다.

- [Online AWS Architecture Diagram Tool \(visual-paradigm.com\)](http://visual-paradigm.com)

# Q & A



## AWS SSM 을 이용하지 않고 CloudWatch agent 설치할 수 있나요?

아래 절차로 가능 합니다. Bastion Host 를 이용하거나 EC2 Command line 입력 방법으로 설치 합니다.

```
# install cloudwatch agent ( amazon linux )
sudo yum install amazon-cloudwatch-agent

# install cloudwatch agent (ubuntu)
sudo wget https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb
sudo dpkg -i ./amazon-cloudwatch-agent.deb

# exec cloudwatch-agent-config-wizard
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard

# config modify

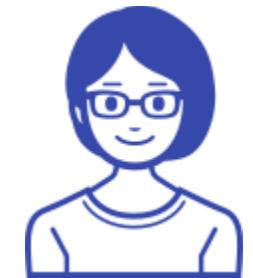
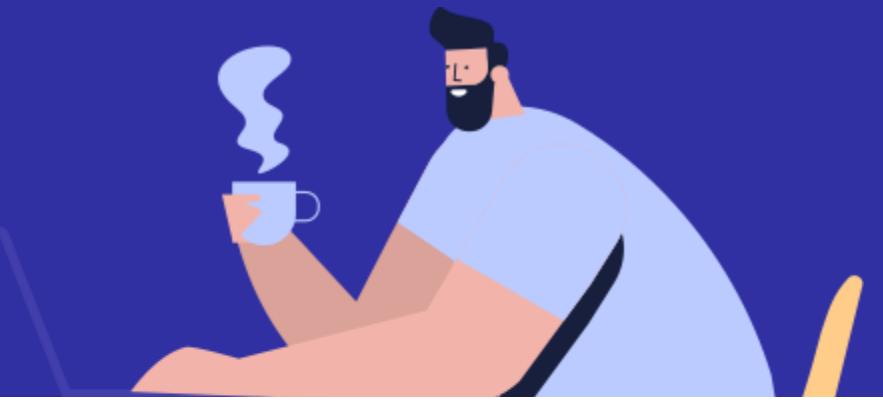
# agent run
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -c file:/opt/aws/amazon-cloudwatch-agent/bin/config.json -s

# agent start : sudo amazon-cloudwatch-agent-ctl -m ec2 -a start
# agent stop : sudo amazon-cloudwatch-agent-ctl -m ec2 -a stop

# agent status
sudo amazon-cloudwatch-agent-ctl -m ec2 -a status

# agent log
sudo more /opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log
```

# Q & A



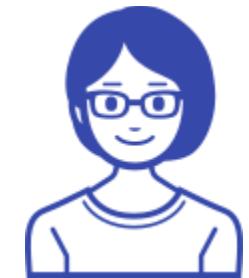
## 프로젝트의 기술을 사용하려면 비용이 발생 하나요?

프로젝트는 AWS 프리티어로 진행하였습니다. 일부 서비스(로그수집, Route53 등)는 유료입니다. 프로젝트로 수행 시 발생된 비용은 다음과 같습니다.

Estimated Total	720 KRW	\$0.58	Bandwidth	\$0.03
Your invoiced total will be displayed once an invoice is issued.			\$0.000 per GB - data transfer in per month	0.373 GB \$0.00
			\$0.000 per GB - data transfer out under the monthly global free tier	0.041 GB \$0.00
Details	+ Expand All		\$0.000 per GB - regional data transfer under the monthly global free tier	1.000 GB \$0.00
AWS Service Charges	\$0.58		\$0.01 per GB - regional data transfer - in/out/between EC2 AZs or using elastic IPs or ELB	2.665 GB \$0.03
▶ Athena	\$0.00			
▶ CloudWatch	\$0.00			
▶ Data Transfer	\$0.03			
▶ Elastic Compute Cloud	\$0.00			
▶ Elastic Load Balancing	\$0.00			
▶ Glue	\$0.00			
▶ Key Management Service	\$0.00			
▶ Lambda	\$0.00			
▶ Route 53	\$0.50			
▶ Simple Notification Service	\$0.00			
▶ Simple Queue Service	\$0.00			
▶ Simple Storage Service	\$0.00			
Taxes				
VAT to be collected	\$0.05			

5

# Q & A



## 프로젝트의 기술을 사용하려면 비용이 발생 하나요?

또한, CloudWatch logs 사용은 유료입니다. 요금은 아래 AWS 문서에 나와 있습니다.

- CloudWatch 에이전트가 수집한 로그는 이전 CloudWatch Logs 에이전트가 수집한 로그와 마찬가지로 Amazon CloudWatch Logs에서 처리되고 저장됩니다.

CloudWatch Logs 요금에 대한 자세한 내용은 Amazon CloudWatch 요금을 참조하세요.

- [Amazon CloudWatch 요금 – Amazon Web Services\(AWS\)](#)