

데이터 파이프라인과 AI 알고리즘의 AWS 활용



강사 : 고병화

7. AWS API Gateway



AWS API Gateway

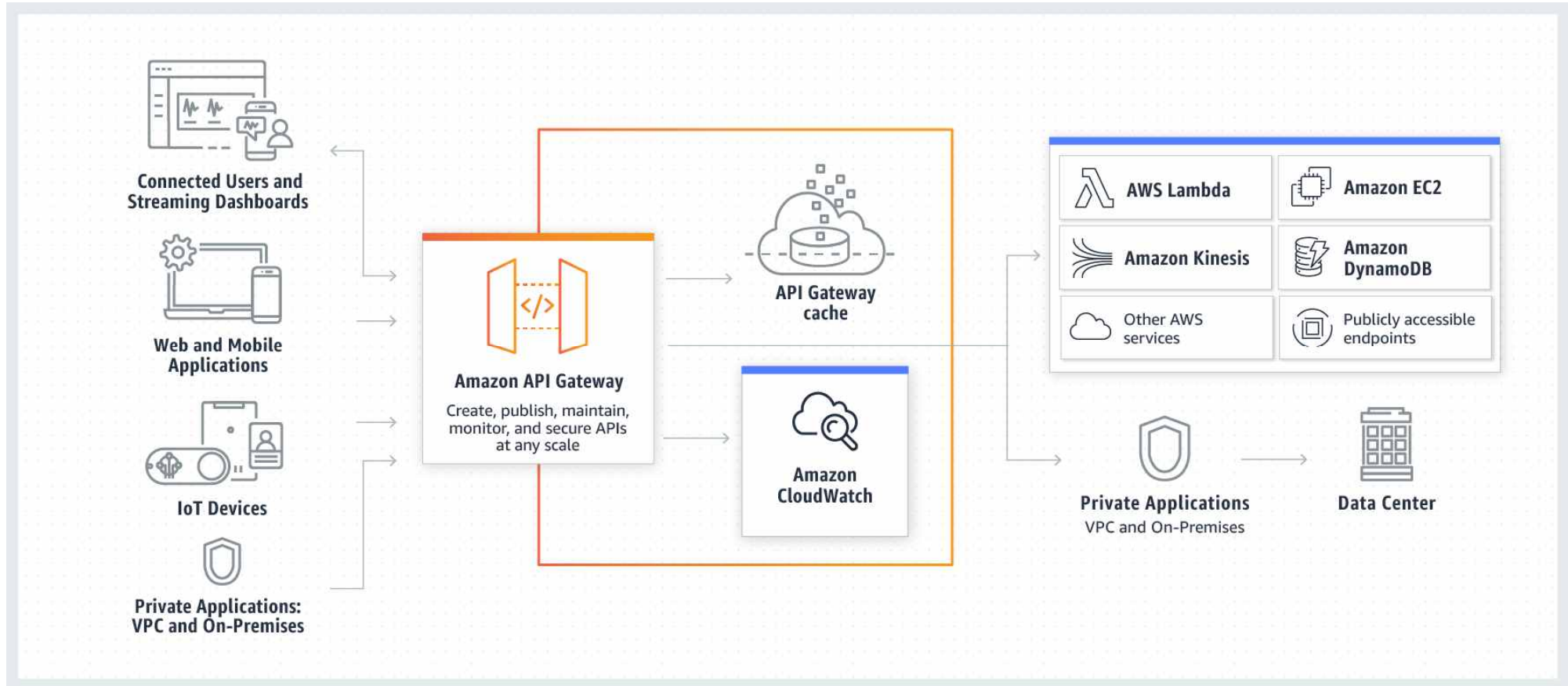
Amazon API Gateway는 어떤 규모에서든 개발자가 API를 손쉽게 생성, 게시, 유지 관리, 모니터링 및 보안 유지할 수 있도록 하는 완전관리형 서비스이다.

API는 애플리케이션이 백엔드 서비스의 데이터, 비즈니스 로직 또는 기능에 액세스할 수 있는 "정문(front door)" 역할을 한다.

API Gateway를 사용하면 실시간 양방향 통신 애플리케이션이 가능하도록 하는 **RESTful API** 및 **WebSocket API**를 작성할 수 있다.

API Gateway는 컨테이너식 서버리스 워크로드 및 웹 애플리케이션을 지원합니다.

AWS API Gateway



AWS API Gateway

- REST(REpresentational State Transfer)란?

REST의 정의 : "REpresentational State Transfer" 의 약자로, 자원을 이름(자원의 표현)으로 구분해 해당 자원의 상태(정보)를 주고 받는 모든 것을 의미합니다.

REST의 설계 규칙을 잘 지켜서 설계된 API를 **RESTful한 API**라고 합니다.

자원(resource)의 표현(representation)에 의한 상태 전달을 뜻합니다.

- 자원 : 해당 소프트웨어가 관리하는 모든 것 (문서, 그림, 데이터, 해당 소프트웨어 자체 등)

AWS API Gateway

- 표현 : 그 자원을 표현하기 위한 이름 (DB의 학생 정보가 자원이면, 'students'를 자원의 표현으로 정함)
- 상태 전달 : 데이터가 요청되는 시점에 자원의 상태를 전달한다. (JSON 혹은 XML을 통해 데이터를 주고 받는 것이 일반적)

REST는 기본적으로 웹의 기존 기술과 HTTP 프로토콜을 그대로 활용하기 때문에, 웹의 장점을 최대한 활용할 수 있는 아키텍처 스타일입니다.

REST는 네트워크 상에서 Client와 Server 사이의 통신 방식 중 하나입니다.

AWS API Gateway

- API 유형 선택

[1] HTTP API

OIDC(OpenID Connect) 및 OAuth2(Open Authorization 2)와 같은 기능과 기본 CORS(Cross-Origin Resource Sharing) 지원이 내장된, 지연 시간이 짧고 비용 효율적인 REST API를 구축합니다.

Lambda, HTTP 백엔드와 호환됨

OIDC와 OAuth 2.0 참고 → <https://hudi.blog/open-id/>
CORS 참고 → <https://escapefromcoding.tistory.com/724>

AWS API Gateway

[2] WebSocket API

채팅 애플리케이션 또는 대시보드와 같은 실시간 사용 사례를 위해 지속적 연결을 사용하여 WebSocket API를 구축합니다. Lambda, HTTP, AWS 서비스와 호환

[3] REST API

API 관리 기능과 함께 요청 및 응답을 완벽하게 제어할 수 있는 REST API를 개발합니다.
Lambda, HTTP, AWS 서비스와 호환

AWS API Gateway

[4] REST API 프라이빗

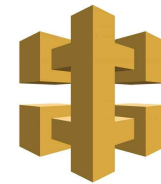
VPC 내에서만 액세스할 수 있는 REST API를 생성합니다.
Lambda, HTTP, AWS 서비스와 호환

Networking & Content Delivery

Amazon API Gateway

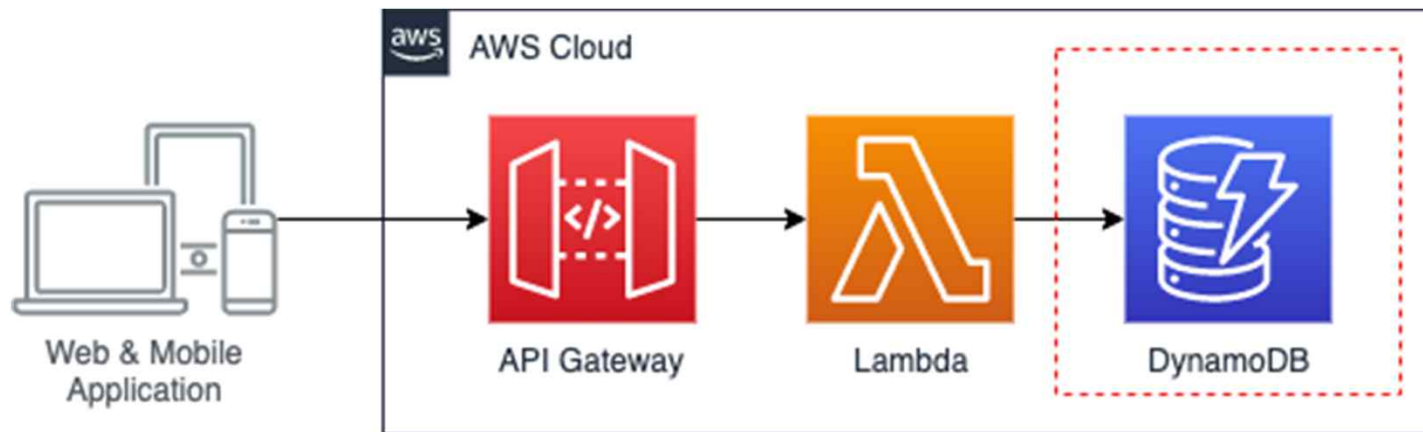
규모와 관계없이 API를 생성, 유지
관리 및 보호

Amazon API Gateway는 개발자들이 Amazon EC2, AWS Lambda 또는 공개된 주소가 있는 웹 서비스에서 실행 중인 백엔드 시스템에 대한 API를 생성하고 관리하는 데 도움이 됩니다. Amazon API Gateway를 사용하여 API에 대한 사용자 지정 클라이언트 SDK를 만들고, 백엔드 시스템을 모바일, 웹, 서버의 애플리케이션 또는 서비스에 연결할 수 있습니다.



AWS API Gateway

- AWS API Gateway 실습



<https://master.d3f5073vvso9t3.amplifyapp.com/lab4/contents/apigateway/>

AWS API Gateway

[1] IAM 역할 생성

AWS Cloud 콘솔에서 [IAM] 메뉴로 들어가 [역할]을 누르고
상단의 [역할 만들기] 버튼을 클릭한다

[신뢰할 수 있는 엔터티 유형]에서 [AWS 서비스]를 체크하고
[사용 사례]에서 [Lambda]를 선택해준다

[다음] 버튼을 누른다

AWS API Gateway

신뢰할 수 있는 엔터티 유형

☒ **AWS 서비스**
EC2, Lambda 등의 AWS 서비스가 이 계정에서 작업을 수행하도록 허용합니다.

☐ **AWS 계정**
사용자 또는 서드 파티에 속한 다른 AWS 계정의 엔터티가 이 계정에서 작업을 수행하도록 허용합니다.

☐ **웹 자격 증명**
지정된 외부 웹 자격 증명 공급자와 연동된 사용자가 이 역할을 맡아 이 계정에서 작업을 수행하도록 허용합니다.

☐ **SAML 2.0 연동**
기업 디렉터리에서 SAML 2.0과 연동된 사용자가 이 계정에서 작업을 수행할 수 있도록 허용합니다.

☐ **사용자 지정 신뢰 정책**
다른 사용자가 이 계정에서 작업을 수행할 수 있도록 사용자 지정 신뢰 정책을 생성합니다.

사용 사례

EC2, Lambda 등의 AWS 서비스가 이 계정에서 작업을 수행하도록 허용합니다.

일반 사용 사례

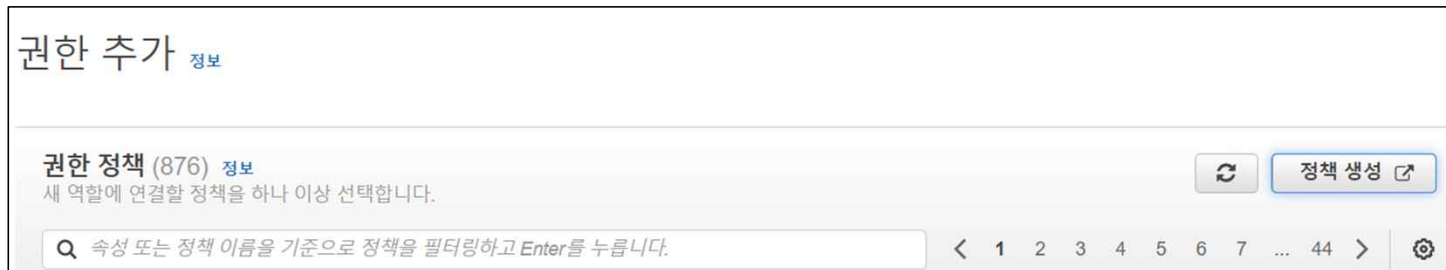
- ☐ **EC2**
Allows EC2 instances to call AWS services on your behalf.
- ☒ **Lambda**
Allows Lambda functions to call AWS services on your behalf.

취소

다음

AWS API Gateway

- 권한 추가 화면에서 [정책 생성] 버튼을 누른다



- 권한 추가 화면에서 [정책 생성] 버튼을 누른 다음
[서비스 선택]에서 검색창에 DynamoDB를 입력하여 찾고
하단 DynamoDB를 클릭하고 [액세스 수준]에서 [쓰기]를
선택하고 [PutItem] 하나만 선택해준다

AWS API Gateway

▼ 서비스 선택

서비스의 특정 리소스에 대해 수행할 수 있는 작업을 지정합니다.

Q DynamoDB



☒ 인기 있는 서비스

DynamoDB ⓘ

DynamoDBAccelerator ⓘ

액세스 수준

모두 확장

▶ **나열** (6)

▶ 읽기 (26)

▼ 쓰기 (1개 선택됨/31)

☐ 모든 쓰기 작업☐ BatchWriteItem ⓘ☐ CreateTable ⓘ☐ Deleteltem ⓘ

☐ DisableKinesisStreamingDestination ⓘ

☐ ImportTable ⓘ☐ PartiQLUpdate ⓘ☐ CreateBackup ⓘ☐ CreateTableReplica ⓘ☐ DeleteTable ⓘ

☐ EnableKinesisStreamingDestination ⓘ
tion

☐ PartiQLDelete ⓘ☐ PurchaseReservedCapacityOffe ⓘ
rings☐ CreateGlobalTable ⓘ☐ DeleteBackup ⓘ☐ DeleteTableReplica ⓘ☐ ExportTableToPointInTime ⓘ

☐ PartiQLInsert ⓘ

- ✓ PutItem ⓘ

AWS API Gateway

- 하단의 [리소스]에서 [모두]를 선택하고 [다음] 버튼을 누른다

▼ 리소스

이러한 작업에 대해 리소스 ARN을 지정합니다.

☐ 특정

☒ 모두

⚠

모든 와일드카드 '*'는 선택한 작업에 대해 지나치게 허용적일 수 있습니다. 이러한 서비스 리소스에 대해 특정 ARN을 허용하면 보안을 강화할 수 있습니다.

▶ 요청 조건 - 선택 사항

리소스에 대한 작업은 이러한 조건이 충족되는 경우에만 허용되거나 거부됩니다.

+ 권한 더 추가

🛡️ 보안: 0

✖ 오류: 0

⚠ 경고: 0

💡 추천: 0

취소

다음

AWS API Gateway

- [정책 세부 정보]에서 [정책이름]에 “aws-lab-dynamodb-putitem-policy”로 입력하고 하단의 [정책 생성] 버튼을 누른다
우측하단 처럼 추가된 정책이 보여진다

정책 세부 정보

정책 이름
이 정책을 식별하는 의미 있는 이름을 입력합니다.

aws-lab-dynamodb-putitem-policy

최대 128자입니다. 영숫자 및 '+', '@', '-' 문자를 사용하세요.

취소

이전

정책 생성

[IAM](#) > 정책

정책 (1122) 정보

정책은 권한을 정의하는 AWS의 객체입니다.

Q 속성 또는 정책 이름을 기준으로 정책을 필터링하고 Enter를 누릅니다.

	정책 이름	유형	다음과 같
<input type="radio"/>	 aws-lab-dynamodb-putitem-policy	고객 관리형	없음
<input type="radio"/>	 AdministratorAccess	AWS 관리형 - 직무	없음

AWS API Gateway

- AWS Cloud 콘솔에서 [IAM] 메뉴로 돌아가서 [역할]을 누르고 상단의 [역할 만들기] 버튼을 클릭한다
- [신뢰할 수 있는 엔터티 유형]에서 [AWS 서비스]를 체크하고
- [사용 사례]에서 [Lambda]를 선택해준다
- [다음] 버튼을 누른다

AWS API Gateway

신뢰할 수 있는 엔터티 유형

☒ **AWS 서비스**
EC2, Lambda 등의 AWS 서비스가 이 계정에서 작업을 수행하도록 허용합니다.

☐ **AWS 계정**
사용자 또는 서드 파티에 속한 다른 AWS 계정의 엔터티가 이 계정에서 작업을 수행하도록 허용합니다.

☐ **웹 자격 증명**
지정된 외부 웹 자격 증명 공급자와 연동된 사용자가 이 역할을 맡아 이 계정에서 작업을 수행하도록 허용합니다.

☐ **SAML 2.0 연동**
기업 디렉터리에서 SAML 2.0과 연동된 사용자가 이 계정에서 작업을 수행할 수 있도록 허용합니다.

☐ **사용자 지정 신뢰 정책**
다른 사용자가 이 계정에서 작업을 수행할 수 있도록 사용자 지정 신뢰 정책을 생성합니다.

사용 사례

EC2, Lambda 등의 AWS 서비스가 이 계정에서 작업을 수행하도록 허용합니다.

일반 사용 사례

- ☐ **EC2**
Allows EC2 instances to call AWS services on your behalf.
- ☒ **Lambda**
Allows Lambda functions to call AWS services on your behalf.

취소

다음

AWS API Gateway

- IAM 메뉴의 [역할]로 돌아가서 [역할만들기] 버튼을 누른 다음 [정책 이름에서] “aws-lab-dynamodb-putitem-policy”를 선택하고 [다음] 버튼을 누른다

권한 추가 [정보](#)

권한 정책 (선택됨 1/876) [정보](#)
새 역할에 연결할 정책을 하나 이상 선택합니다.

Q 속성 또는 정책 이름을 기준으로 정책을 필터링하고 Enter를 누릅니다.

< 1 2 3 4 5 6 7 ... 44 >

<input type="checkbox"/>	정책 이름 ↗	유형	설명
<input checked="" type="checkbox"/>	<input type="button" value="⊕"/> aws-lab-dynamodb-putitem-policy	고객 관...	
<input type="checkbox"/>	<input type="button" value="⊕"/> AdministratorAccess	AWS 관...	Provides full access to AWS services and resources.
<input type="checkbox"/>	<input type="button" value="⊕"/> PowerUserAccess	AWS 관...	Provides full access to AWS services and resources, but does not allow management o...

이전

다음

AWS API Gateway

- [역할 이름]에 “aws-lab-dynamodb-role”을 입력하고
하단의 [역할 생성] 버튼을 누른다

이름 지정, 검토 및 생성

역할 세부 정보

역할 이름

이 역할을 식별하는 의미 있는 이름을 입력합니다.

aws-lab-dynamodb-role

최대 64자입니다. 영숫자 및 '+', '@', '-' 문자를 사용하세요.

이전

역할 생성

아래와 같이 역할이 만들어진 것을
볼수 있다

역할 (5) 정보		새로고침	삭제
IAM 역할은 단기간 동안 유효한 자격 증명을 가진 특정 권한이 있는 자격 증명입니다. 신뢰할 수 있는 개체가 역할을 맡을 수 있습니다.			
검색			
<input type="checkbox"/>	역할 이름	신뢰할 수 있는 개체	
<input type="checkbox"/>	aws-lab-dynamodb-role	AWS 서비스: lambda	
<input type="checkbox"/>	AWSServiceRoleForAPIGateway	AWS 서비스: ops.apigateway (서비스 연결 역할)	
<input type="checkbox"/>	AWSServiceRoleForApplicationAutoScaling_DynamoDBTable	AWS 서비스: dynamodb.application-autoscaling (서비스 연결 역할)	

AWS API Gateway

[2] DynamoDB 테이블 생성

AWS 콘솔에서 DynamoDB를 찾아 들어간다

[테이블 생성]을 누르고

테이블 이름 : customers

파티션키 : customer_id / 문자열

테이블 설정 : 기본 설정

으로 설정하고 하단 [테이블 생성]버튼을 클릭한다

AWS API Gateway

테이블 생성

테이블 세부 정보 정보

DynamoDB는 테이블을 생성할 때 테이블 이름과 기본 키만 필요한 스키마리스 데이터베이스입니다.

테이블 이름

테이블을 식별하는 데 사용됩니다.

customers

문자, 숫자, 밑줄(_), 하이픈(-) 및 마침표(.)만 포함하는 3~255자의 문자입니다.

파티션 키

파티션 키는 테이블 기본 키의 일부로, 테이블에서 항목을 검색하고 확장성과 가용성을 위해 호스트에 데이터를 할당하는 데 사용되는 해시 값입니다.

customer_id

문자열

1~255자이고 대소문자를 구분합니다.

정렬 키 - 선택 사항

정렬 키를 테이블 기본 키의 두 번째 부분으로 사용할 수 있습니다. 정렬 키를 사용하면 동일한 파티션 키를 공유하는 모든 항목을 정렬하거나 검색할 수 있습니다.

정렬 키 이름 입력

문자열

1~255자이고 대소문자를 구분합니다.

테이블 생성

DynamoDB > 테이블

테이블 (1) 정보

테이블 이름으로 테이블 찾기

모든 태그 키

<input type="checkbox"/>	이름 ▲	상태	파티션 키	정렬 키	인덱스	삭제 방지
<input type="checkbox"/>	customers	✓ 활성화	customer_id (S)	-	0	⊖ 끄기

일반 정보

파티션 키
customer_id (String)

정렬 키

-

용량 모드

프로비저닝됨

경보

✓ 활성화 경고 없음

특정 시점으로 복구(PITR)

정보

⊖ 끄기

▶ 추가 정보

항목 요약

DynamoDB는 약 6시간마다 다음 정보를 업데이트합니다.

항목 수
0

테이블 크기
0바이트

생성된
테이블의
정보를
확인해본
다

AWS API Gateway

[3] API Gateway를 사용하여 API 생성

AWS 콘솔에서 API Gateway를 찾아 들어간다

[API 유형선택]에서 3번째 **[REST API]**를(프라이빗 아님)

찾아 [구축]버튼을 클릭하고 팝업창에서 [확인]을 누른다

(4번째 REST API 프라이빗은 VPC내에서만 API 액세스가 가능하다)



AWS API Gateway

- [새 API 생성] 에서 [새 API]를 선택해주고 [설정]의 [API 이름]을 "customer-api"로 입력하고 [엔드포인트 유형]은 "지역"으로 선택하고 [API 생성]을 클릭한다

새 API 생성

Amazon API Gateway에서 REST API는 HTTPS 엔드포인트를 통해 호출할 수 있는 리소스와 메서드의 컬렉션을 참조합니다.

☒ 새 API ☐ Swagger에서 가져오기 또는 API 3 열기 ☐ 예제 API

설정

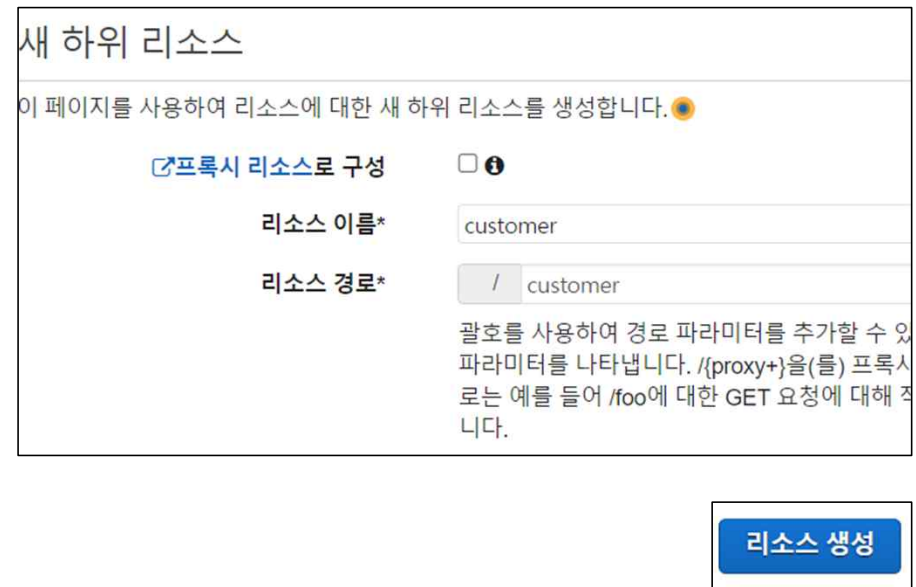
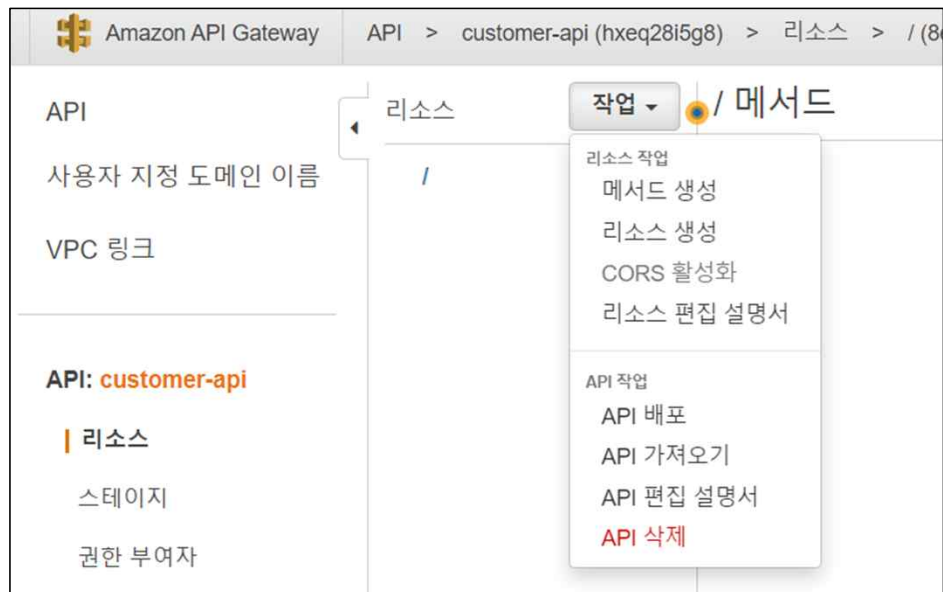
API에 대해 기억하기 쉬운 이름과 설명을 선택합니다.

API 이름*	<input type="text" value="customer-api"/>
설명	<input type="text"/>
엔드포인트 유형	<div>지역 </div>

API 생성

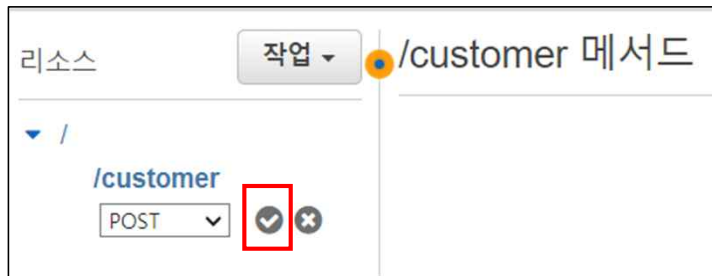
AWS API Gateway

- “API: customer-api”를 선택하고 [작업]버튼을 누르고 [리소스 생성]을 누른 다음 리소스 이름을 “customer”로 입력하고 [리소스 생성]버튼을 클릭한다

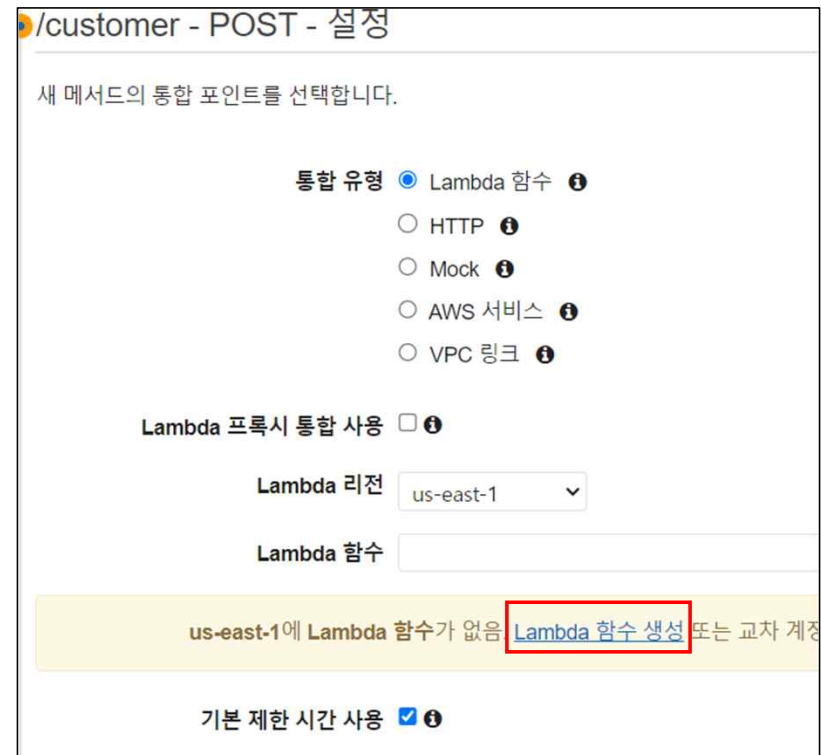


AWS API Gateway

- Customer 리소스 아래 드롭다운 상자에서 "POST"를 선택하고 바로 옆에 있는 체크표시를 클릭한다



우측화면에서 그대로
[Lambda 함수 생성]을 클릭
한다



AWS API Gateway

[4] Lambda 함수 생성

새로운 탭의 AWS 콘솔에서 [함수 생성]에서 [새로 작성]을 선택하고 함수 이름을 "addCustomer"로 입력하고 [런타임]은 "Python 3.10"으로 선택해준다

함수 생성 정보

AWS Serverless Application Repository 애플리케이션을 [애플리케이션 생성](#)으로 이동했습니다.

☒ 새로 작성
간단한 Hello World 예제는 시작하십시오.

☐ 블루프린트 사용
샘플 코드 및 구축 Lambda 애플리케이션을 위한 구성 사전 사용 사례를 살펴봅니다.

기본 정보

함수 이름
함수의 용도를 설명하는 이름을 입력합니다.

공백 없이 문자, 숫자, 하이픈 또는 밑줄만 사용합니다.

런타임 정보
함수를 작성하는 데 사용할 언어를 선택합니다. 콘솔 코드 편집기는 Node.js, Python 및 Ruby만 지원합니다.

아키텍처 정보
함수 코드에 대해 원하는 명령 세트 아키텍처를 선택합니다.

☒ x86_64

☐ arm64

AWS API Gateway

- 하단의 기본 실행 역할 변경에서 [기존 역할 사용]을 선택하고 [기존 역할]에서 "aws-lab-dynamodb-role"을 선택한 다음 하단의 [함수 생성] 버튼을 클릭 한다

▼ 기본 실행 역할 변경

실행 역할

함수에 대한 권한을 정의하는 역할을 선택합니다. 사용자 지정 역할을 생성하려면 [IAM 콘솔](#)로 이동하십시오.

☐ 기본 Lambda 권한을 가진 새 역할 생성

☒ 기존 역할 사용

☐ AWS 정책 템플릿에서 새 역할 생성

기존 역할

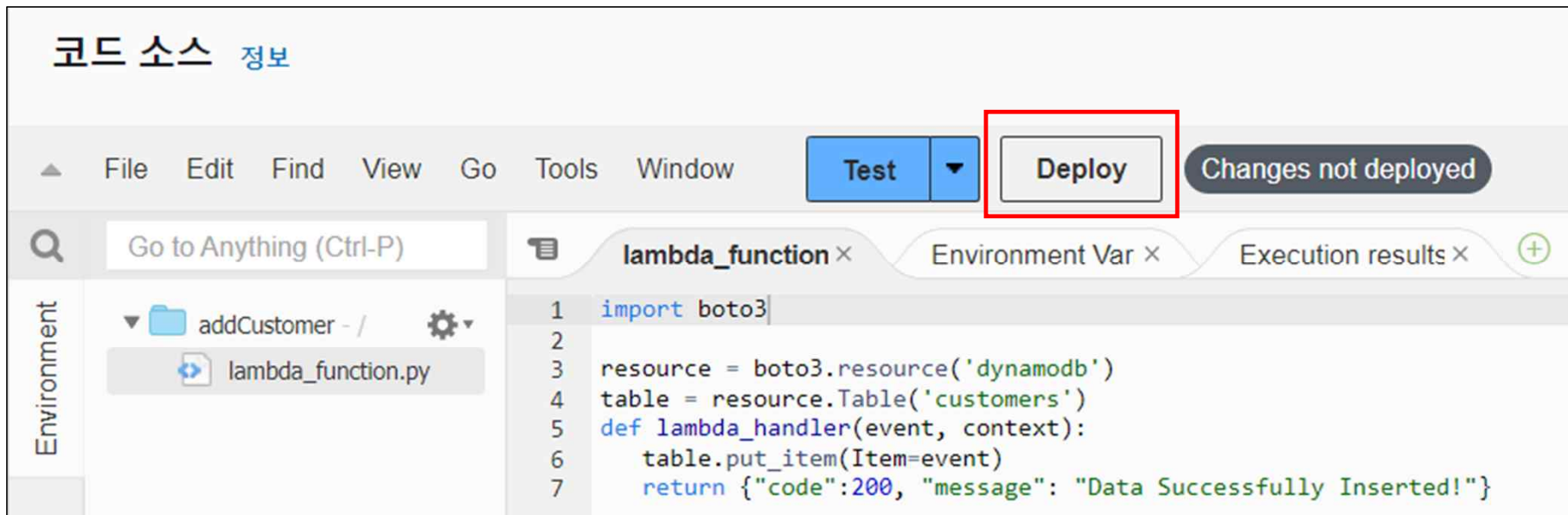
생성한 기존 역할 중에 이 Lambda 함수와 함께 사용할 역할을 선택합니다. 이 역할에는 Amazon CloudWatch Logs에 로그를 업로드할 수 있는 권한이 있어야 합니다.

[IAM 콘솔에서 aws-lab-dynamodb-role 역할을 확인](#)합니다.

[취소](#) [함수 생성](#)

AWS API Gateway

- Code change 관련 팝업창에서 [OK]를 누르고 lambda_function.py를 클릭하고 아래와 같이 소스 코드를 수정하여 입력한 다음 [Deploy] 버튼을 클릭한다



AWS API Gateway

- 앞의 API Gateway 생성 창에서 [Lambda 함수]에 새로 만든 "addCustomer"를 입력하고 [저장]버튼을 클릭한다

새 메서드의 통합 포인트를 선택합니다.

통합 유형 ☒ Lambda 함수 ⓘ
☐ HTTP ⓘ
☐ Mock ⓘ
☐ AWS 서비스 ⓘ
☐ VPC 링크 ⓘ

Lambda 프록시 통합 사용 ☐ ⓘ

Lambda 리전

Lambda 함수

us-east-1에 Lambda 함수가 없음. [Lambda 함수 생성](#) 또는 교차 계역

기본 제한 시간 사용 ☒ ⓘ

저장

[Lambda 함수에 대한 권한 추가]
팝업창이 뜨면 [확인]을 클릭한다

Lambda 함수에 대한 권한 추가

Lambda 함수를 호출하기 위해 API Gateway 권한을 부여하려고 합니다.
arn:aws:lambda:us-east-1:163625374812:function:addCustomer

취소 확인

AWS API Gateway

우측 화면
이 나타나
면 좌측의
[테스트]
를 클릭한
다



AWS API Gateway

- 쿼리 문자열과 헤더는 비워 두고 하단 [요청 본문]에 아래와 같이 입력하고 하단 [Test]버튼을 클릭한다

스테이지 변수

이 메서드에 대한 [스테이지 변수](#)가 없습니다.

요청 본문

```
1 {  
2     "customer_id" : "1230AWS",  
3     "customer_name" : "KilDong",  
4     "product_name" : "laptop",  
5     "price" : 990000  
6 }  
7 |
```

Test 요청 본문

```
{  
    "customer_id" : "1230AWS",  
    "customer_name" : "KilDong",  
    "product_name" : "laptop",  
    "price" : 990000  
}
```

⚡ 테스트

AWS API Gateway

- Test 결과 화면

요청: /customer

상태: 200

지연 시간: 750ms

응답 본문

```
{"code": 200, "message": "Data Successfully Inserted!"}
```

응답 헤더

```
{"Content-Type":["application/json"],"X-Amzn-Trace-Id":["Root=1-64f6dcc8-07bc4cb214cdc312028c361b;Sampled=0;lineage=511c94c2:0"]}
```

로그

```
Execution log for request 9698c5ba-c371-4d53-bfe4-6237cd6951d1
Tue Sep 05 07:46:16 UTC 2023 : Starting execution for request: 9698c5ba-c371-4d53-bfe4-6237cd6951d1
Tue Sep 05 07:46:16 UTC 2023 : HTTP Method: POST, Resource Path: /customer
Tue Sep 05 07:46:16 UTC 2023 : Method request path: {}
Tue Sep 05 07:46:16 UTC 2023 : Method request query string: {}
Tue Sep 05 07:46:16 UTC 2023 : Method request headers: {}
Tue Sep 05 07:46:16 UTC 2023 : Method request body before transformations: {
  "customer_id" : "1230AWS",
  "customer_name" : "KiIDong",
  "product_name" : "laptop",
  "price" : 990000
}

Tue Sep 05 07:46:16 UTC 2023 : Endpoint request URI: https://lambda.us-east-1.amazonaws.com/2015-03-31/functions/arn:aws:lambda:us-east-1:163625374812:function:addCustomer/invocations
Tue Sep 05 07:46:16 UTC 2023 : Endpoint request headers: {X-Amz-Date=20230905T074616Z, x-amzn-apigateway-api-id=hxeq28i5g8, Accept=application/json, User-Agent=AmazonAPIGateway_hxeq28i5g8, Host=lambda.us-east-1.amazonaws.com, X-Amz-Content-Sha256=5729cd6d76180444cd4ae7f9a3d78a7efb2ed1bfea7796c5cb93379529ad50f6, X-Amzn-Trace-Id=Root=1-64f6dcc8-07bc4cb214cdc312028c361b, x-amzn-lambda-integration-tag=9698c5ba-c371-4d53-bfe4-6237cd6951d1, Authorization=*****}
```

AWS API Gateway

[5] 실행 결과 확인

AWS 콘솔에서

[DynamoDB]를 찾아 들어
가 “customer” 테이블
로 들어가서 우측 상단의
에서 [표항목 탐색]을 클
릭한다

The screenshot displays the AWS Management Console for a DynamoDB table named 'customers'. At the top, there are tabs for '개요' (Overview), '인덱스' (Indexes), '모니터링' (Monitoring), '글로벌 테이블' (Global Tables), '백업' (Backups), '내보내기 및 스트림' (Export and Streams), and '추가 설정' (Additional Settings). The '개요' tab is selected. Below the tabs, there is a warning banner about PITR (Point-in-Time Recovery) and a 'PITR 편집' button. The '일반 정보' (General Information) section shows the following details:

파티션 키	정렬 키	용량 모드	테이블 상태
customer_id (String)	-	프로비저닝됨	✓ 활성화
경보	특정 시점으로 복구(PITR)		

At the bottom, there are links for '화서 검사' (Check for corruption) and '정보' (Information).

AWS API Gateway

POST 요청에
의해 JSON
데이터가 추
가 된 것을
확인할 수 있
다

☒ 스캔

☐ 쿼리

테이블 또는 인덱스 선택

속성 프로젝션 선택

테이블 - customers

모든 속성

▶ 필터

실행

재설정

✓ 완료. 사용된 읽기 용량 단위: 0.5

✕

반환된 항목 (1)

↺

작업 ▼

항목 생성

<

1

>

⚙

✕

<input type="checkbox"/>	customer_id (문자열) ▼	customer_na... ▼	price ▼	product_name ▼
<input type="checkbox"/>	1230AWS	KilDong	990000	laptop

8. *AWS* CloudWatch



AWS CloudWatch

- CloudWatch는

Amazon은 Amazon Web Services (AWS) 리소스와 사용자가 실행하는 애플리케이션을 AWS 실시간으로 CloudWatch 모니터링합니다.

CloudWatch를 사용하여 리소스 및 애플리케이션에 대해 측정할 수 있는 변수인 지표를 수집하고 추적할 수 있습니다.

CloudWatch를 사용하면 시스템 전체의 리소스 사용률, 애플리케이션 성능 및 운영 상태를 파악할 수 있습니다.

AWS CloudWatch

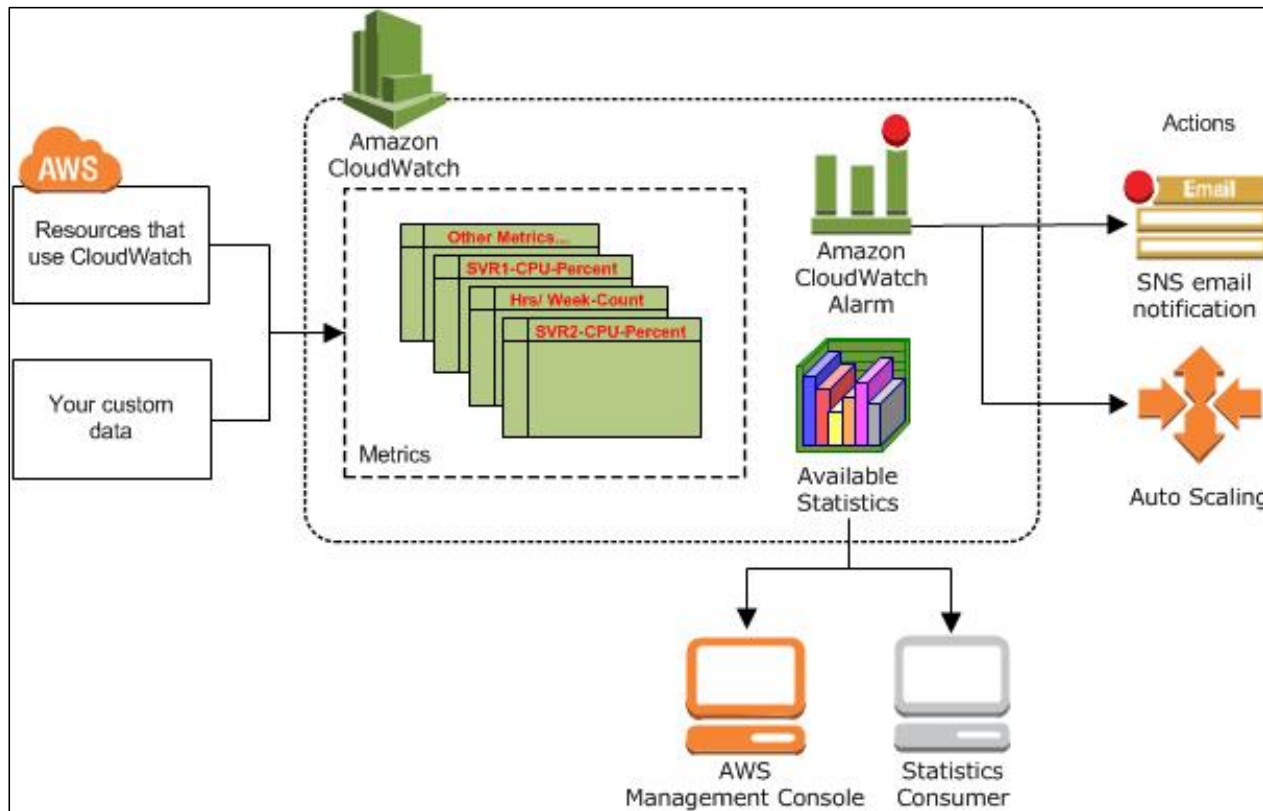
- CloudWatch는

지표를 감시해 알림을 보내거나 임계값을 위반한 경우 모니터링 중인 리소스를 자동으로 변경하는 경보를 생성할 수 있습니다.

예를 들면, Amazon EC2 인스턴스의 CPU 사용량과 디스크 읽기 및 쓰기를 모니터링한 다음에 증가한 로드를 처리하려면 추가 인스턴스를 시작해야 하는지 여부를 해당 데이터로 결정할 수 있습니다. 또한 이러한 데이터를 사용하여 잘 사용되지 않는 인스턴스를 중지할 수도 있습니다.

AWS CloudWatch

- CloudWatchAmazon은 기본적으로 지표 리포지토리이다.
지표를 기반으로 통계를 검색할 수 있다.



AWS CloudWatch 실습

[1] EC2 인스턴스 생성

AWS 콘솔에서 EC2를 찾아 들어가서 [인스턴스 시작]을 클릭한 다음 아래와 같이 입력하여 인스턴스를 생성한다

- 이름 : aws-lab-cloudwatch-instance
- 애플리케이션 및 OS 이미지 : Amazon Linux 2 AMI(HVM) –
Kernel 5.10, SSD Volume Type 프리티어 사용 가능
- 인스턴스 유형 : t2.micro 프리티어 사용 가능
- 키 페어 이름 : 키 페어 없이 계속진행(권장되지 않음)
- 스토리지 구성 : 8GB gp2 루트 볼륨

인스턴스 시작

명령 검토

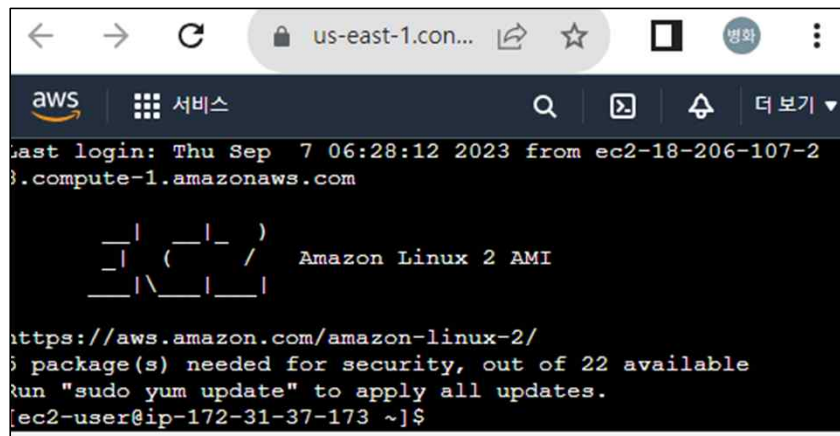
AWS CloudWatch

- 생성된 인스턴스 ID를 찾아 클릭하고 [모니터링] 탭을 클릭한다 아직은 빈그래프만 표시된다



AWS CloudWatch

- EC2의 인스턴스에 연결을 찾아서 [연결] 버튼을 눌러 인스턴스에 접속해본다



```
aws
서비스
us-east-1.con...
last login: Thu Sep 7 06:28:12 2023 from ec2-18-206-107-23.compute-1.amazonaws.com
_ _ _ _ _
_|  ( _ _ _ /  Amazon Linux 2 AMI
_| \ _ _ _ _ _
https://aws.amazon.com/amazon-linux-2/
5 package(s) needed for security, out of 22 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-37-173 ~]$
```

EC2 > 인스턴스 > i-0c7555d428c248200 > 인스턴스에 연결

인스턴스에 연결 정보

다음 옵션 중 하나를 사용하여 인스턴스 i-0c7555d428c248200 (aws-lab-cloudwatch-instance)에 연결

EC2 인스턴스 연결

Session Manager

SSH 클라이언트

EC2 직렬 콘솔

인스턴스 ID

i-0c7555d428c248200 (aws-lab-cloudwatch-instance)

연결 유형

☒ EC2 Instance Connect를 사용하여 연결
퍼블릭 IPv4 주소가 있는 EC2 인스턴스 연결 브라우저 기반 클라이언트를 사용하여 연결합니다.

☐ EC2 인스턴스 연결 엔드포인트를 사용하여 연결
프라이빗 IPv4 주소 및 VPC 엔드포인트가 있는 EC2 인스턴스 연결 브라우저 기반 클라이언트를 사용하여 연결합니다.

퍼블릭 IP 주소

34.207.107.93

사용자 이름

인스턴스를 시작하는 데 사용되는 AMI에 정의된 사용자 이름을 입력합니다. 사용자 지정 사용자 이름을 정의하지 않은 경우 기본 사용자 이름인 ec2-user(들)를 사용합니다.

ec2-user

☒ 참고: 대부분의 경우 기본 사용자 이름 ec2-user(는) 정확합니다. 하지만 AMI 사용 지침을 읽고 AMI 소유자가 기본 AMI 사용자 이름을 변경했는지 확인하십시오.

취소

연결

AWS CloudWatch

접속된 shell 에서 아래 명령으로 파일을 다운로드해보자

`wget https://raw.githubusercontent.com/e9t/nsmc/master/ratings_train.txt`

```
[ec2-user@ip-172-31-37-173 ~]$ wget https://raw.githubusercontent.com/e9t/nsmc/master/ratings_train.txt
--2023-09-07 07:40:49-- https://raw.githubusercontent.com/e9t/nsmc/master/ratings_train.txt
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.109.133,
185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 14628807 (14M) [text/plain]
Saving to: 'ratings_train.txt'

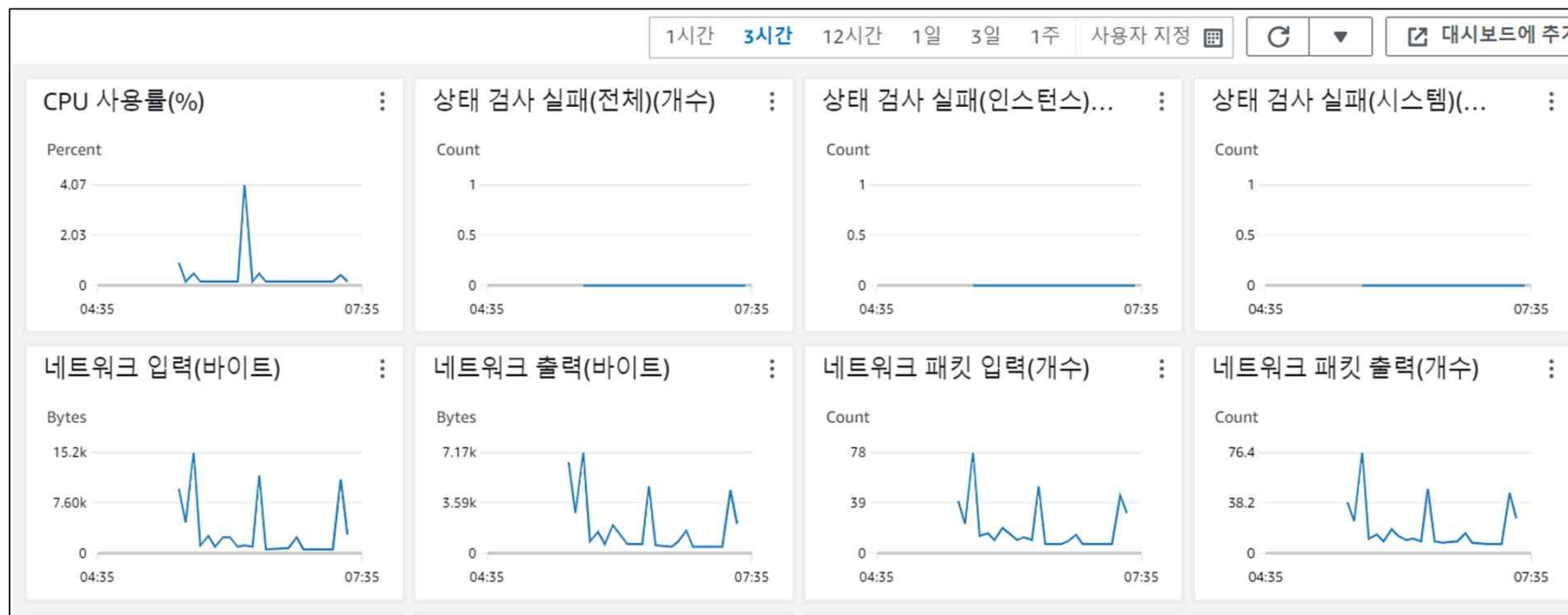
100%[=====
=====>] 14,628,807 --.-K/s in 0.1s

2023-09-07 07:40:50 (145 MB/s) - 'ratings_train.txt' saved [14628807/14628807]

[ec2-user@ip-172-31-37-173 ~]$
```

AWS CloudWatch

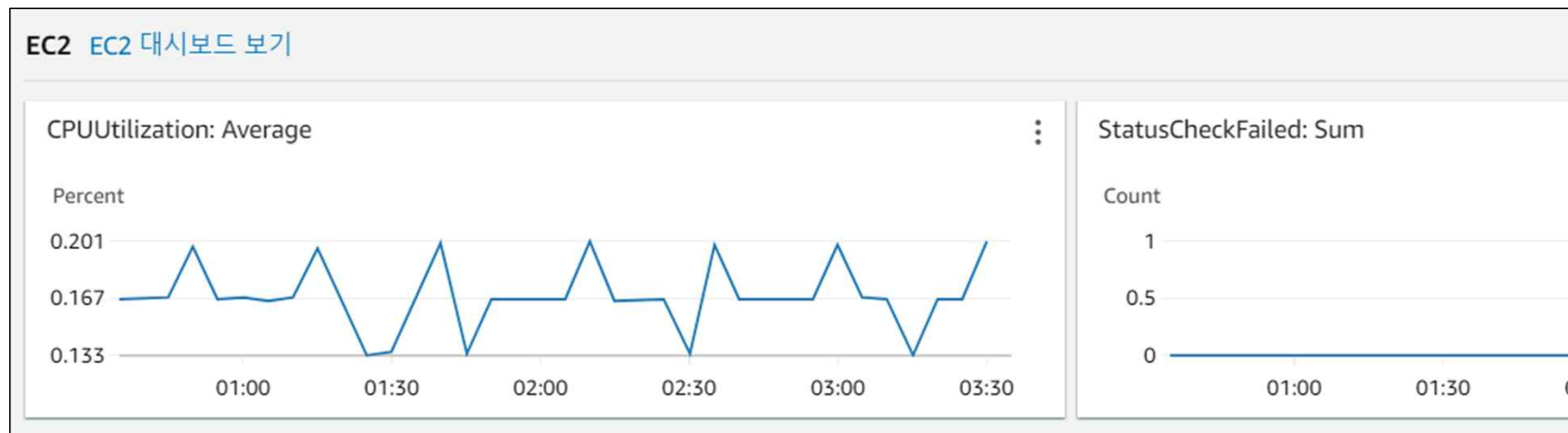
- 다시 EC2 인스턴스 ID를 찾아 클릭하고 [모니터링] 탭을 클릭하면 아래와 같이 사용율이 변화를 확인할 수 있다



AWS CloudWatch

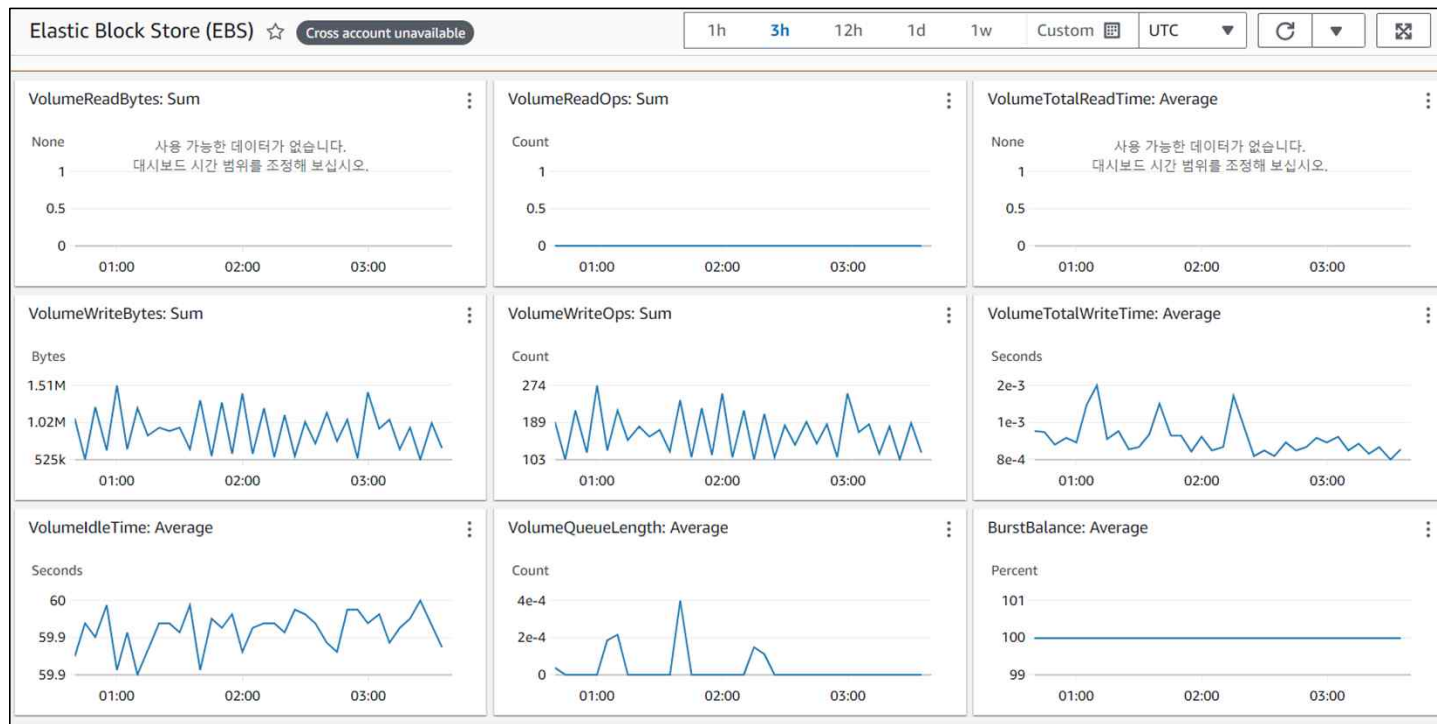
- CloudWatch 대시 보드에서 모니터링 하기

AWS 콘솔에서 CloudWatch를 찾아 들어 가서 [교차 서비스 대시보드]의 하단에서 EC2 대시보드를 확인 할 수 있다



AWS CloudWatch

- 아래에 있는 [Elastic Block Store (EBS) 대시보드 보기]를 클릭하면 EBS에서 일어난 일들을 모두 볼 수 있다



AWS CloudWatch 경보 생성

- AWS 콘솔의 CloudWatch 메인 화면에서 좌측 상단의 [경보 상태]를 클릭한 다음 우측 상단의 [경보생성] 버튼을 클릭한 다음에 화면에서 [지표선택]을 클릭한다



AWS CloudWatch 경보 생성

- [EBS]를 입력하여 찾아서 [EBS]를 클릭하고 지표 이름 중에서 “VolumeWriteBytes”를 체크하고 [지표선택] 버튼을 클릭한다

[찾아보기](#) | [쿼리](#) | [그래프로 표시된 지표](#) | [옵션](#) | [소스](#)

지표 (92)

SQL 사용 그래프

그래프 검색

N. Virginia ▼

Q EBS X

EBS 9

EC2 17

지표 (9)

SQL 사용 그래프

그래프 검색

N. Virginia ▼

모두 > EBS > 볼륨별 지표

Q 모든 지표, 측정기준

< 1 >

⚙

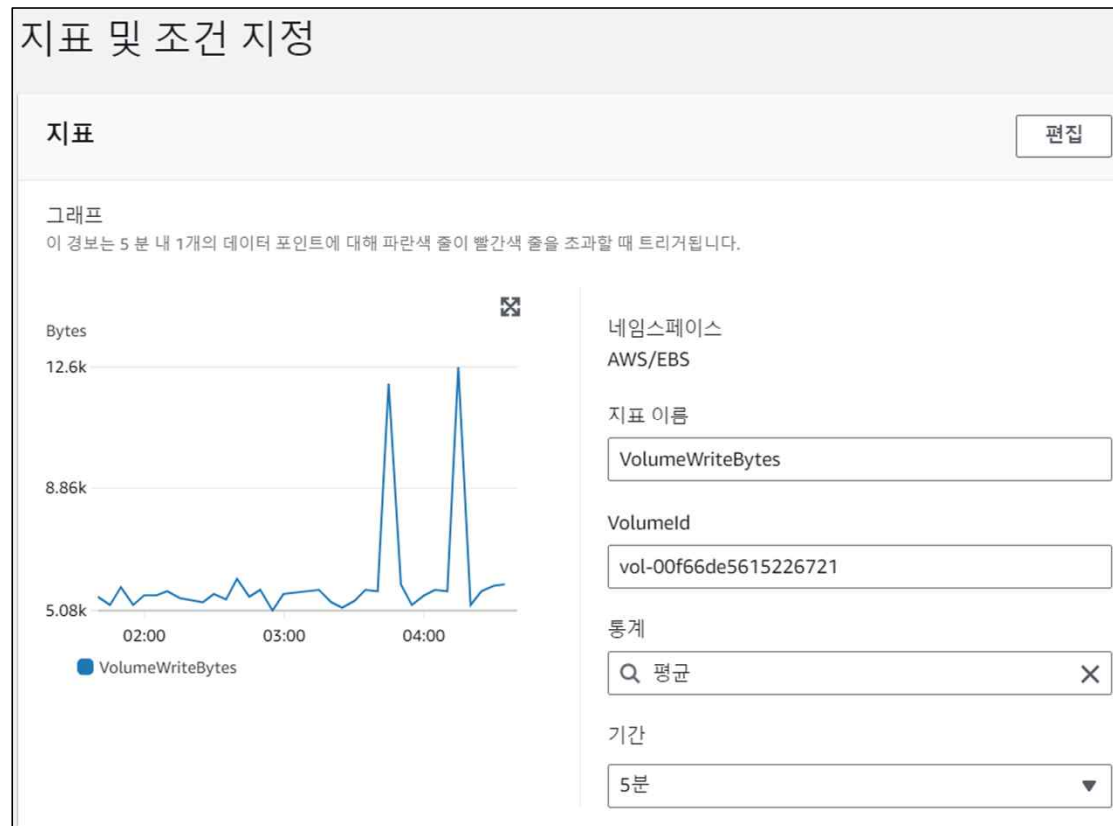
<input type="checkbox"/>	Volumeld 9/9 ▲	지표 이름 ▼
<input type="checkbox"/>	vol-00f66de5615226721	VolumeReadBytes
<input type="checkbox"/>	vol-00f66de5615226721	VolumeTotalReadTime
<input type="checkbox"/>	vol-00f66de5615226721	VolumeReadOps
<input type="checkbox"/>	vol-00f66de5615226721	BurstBalance
<input type="checkbox"/>	vol-00f66de5615226721	VolumeWriteOps
<input type="checkbox"/>	vol-00f66de5615226721	VolumeIdleTime
<input type="checkbox"/>	vol-00f66de5615226721	VolumeQueueLength
<input type="checkbox"/>	vol-00f66de5615226721	VolumeTotalWriteTime
<input checked="" type="checkbox"/>	vol-00f66de5615226721	VolumeWriteBytes

취소

지표 선택

AWS CloudWatch 경보 생성

- [지표 및 조건 지정]의 [지표]는 기본 값으로 둔다



AWS CloudWatch 경보 생성

- [조건]에서 임계값을 "100000"으로 변경한다(나머지는 아래처럼 그대로 둔다. 임계값이 100KB)

조건

임계값 유형

☒ 정적
값을 임계값으로 사용

☐ 이상 탐지
대역을 임계값으로 사용

VolumeWriteBytes이(가) 다음과 같은 경우에 항상...
경보 조건을 정의합니다.

☒ 보다 큼
> 임계값

☐ 보다 크거나 같음
≥ 임계값

☐ 보다 작거나 같음
≤ 임계값

☐ 보다 작음
< 임계값

...보다
임계값을 정의합니다.

100000

숫자여야 함

AWS CloudWatch 경보 생성

- [추가 구성]은 아래와 같이 그대로 두고 [다음] 버튼을 클릭한다

▼ 추가 구성

경보를 알릴 데이터 포인트

평가 기간 내 기준을 위반하는 데이터 포인트 수를 정의합니다(이로 인해 경보 상태로 전환됨).

/

누락된 데이터 처리

경보를 평가할 때 누락된 데이터를 처리하는 방법입니다.

취소

다음

AWS CloudWatch 경보 생성

- [새 주제 생성]을 선택
하고 주제 이름에

"AWSLab_CloudWatch_V
olume_Alarms_Topic"을
입력하고

하단에 수신할 이메일
을 입력한다음

[주제 생성]버튼을 클릭
한다음 [다음]을 누른다

알림

경보 상태 트리거
이 작업을 트리거하는 경보 상태를 정의합니다.

☒ 경보 상태
지표 또는 표현식이 정의된 임계값
을 벗어났습니다.

☐ 정상
지표 또는 표현식이 정의된 임계값
범위에 있습니다.

☐ 데이터 부족
경보가 방금 시작되었거나 사용 가
능한 데이터가 부족합니다.

다음 SNS 주제에 알림을 보냅니다.
알림을 수신할 SNS(Simple Notification Service) 주제를 정의합니다.

☐ 기존 SNS 주제 선택

☒ 새 주제 생성

☐ 주제 ARN을 사용하여 다른 계정에 알림

새 주제 생성...

주제 이름은 고유해야 합니다.

AWSLab_CloudWatch_Volume_Alarms_Topic

SNS 주제 이름은 영숫자, 하이픈(-) 및 밑줄(_)만 포함할 수 있습니다.

알림을 수신할 이메일 엔드포인트...

실제로 구분된 이메일 주소 목록을 추가합니다. 각 주소는 위 주제에 구독으로 추가됩니다.

digicope@aicore.co.kr

user1@example.com, user2@example.com

주제 생성

알림 추가

제거

AWS CloudWatch 경보 생성

- 경보 이름에

"awslab_cloudwatch_toomuch_volume_alarm"을 입력하고

경보 설명에는

"EBS Volume에 너무 많은 쓰기가 수행되었습니다"를 입력하고
[다음]을 누른다

이름 및 설명 추가

이름 및 설명

경보 이름

awslab_cloudwatch_toomuch_volume_alarm

경보 설명 - 선택 사항 [서식 가이드라인 보기](#)

편집

미리 보기

EBS Volume에 너무 많은 쓰기가 수행되었습니다

최대 1024자(29/1024)

❗

마크다운 서식은 콘솔에서 경보를 볼 때만 적용됩니다. 설명은 경보 알림에서 일반 텍스트 서식으로 유지됩니다.

취소

이전

다음

AWS CloudWatch 경보 생성

- 미리보기 및 생성에서 설정된 값들을 확인하고 하단의 [경보 생성] 버튼을 클릭하면 아래와 같이 생성된 경보가 목록에 나타난다

[CloudWatch](#) > 경보

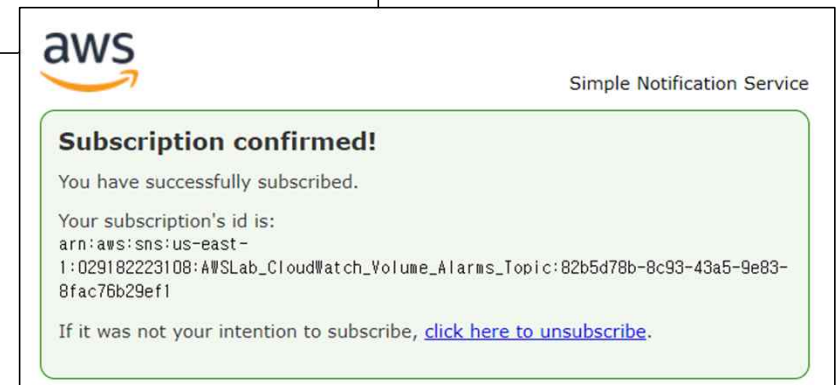
경보 (1) ☐ Auto Scaling 경보 숨기기 선택 항목 지우기 🔄 복합 경보 생성 작업 ▼ 경보 생성

모든 상태 ▼ 모든 유형 ▼ 모든 작업 상태 ▼ < 1 > ⚙️

<input type="checkbox"/>	이름 ▼	상태 ▼	마지막 상태 업데이트 ▼	조건	작업 ▼
<input type="checkbox"/>	aws-lab-cloudwatch-toomuch-volume-alarm	⋮ 데이터 부족	2023-09-08 05:09:08	5 분 내 1개의 데이터 포인트에 대한 VolumeWriteBytes > 1000000	✔️ 작업이 활성화됨 경고

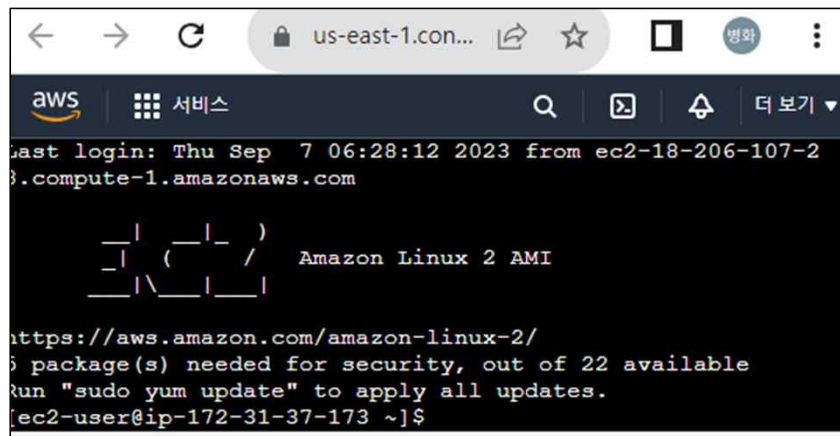
AWS CloudWatch 경보 생성

- 입력한 메일로 가서 수신된 메일을 선택하고 메일 내용에서 [Confirm subscription]을 눌러 AWS에 동의해준다



AWS CloudWatch

- EC2의 인스턴스에 연결을 찾아서 [연결] 버튼을 눌러 인스턴스에 접속해본다



```
aws
서비스
us-east-1.con...
last login: Thu Sep 7 06:28:12 2023 from ec2-18-206-107-23.compute-1.amazonaws.com
_ _ _ _ _
_|  ( _ _ _ /   Amazon Linux 2 AMI
_| \_|_|_|_|_|
https://aws.amazon.com/amazon-linux-2/
5 package(s) needed for security, out of 22 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-37-173 ~]$
```

EC2 > 인스턴스 > i-0c7555d428c248200 > 인스턴스에 연결

인스턴스에 연결 정보

다음 옵션 중 하나를 사용하여 인스턴스 i-0c7555d428c248200 (aws-lab-cloudwatch-instance)에 연결

EC2 인스턴스 연결	Session Manager	SSH 클라이언트	EC2 직렬 콘솔
<p>인스턴스 ID i-0c7555d428c248200 (aws-lab-cloudwatch-instance)</p> <p>연결 유형</p> <div><input checked="" type="radio"/> EC2 Instance Connect를 사용하여 연결 퍼블릭 IPv4 주소가 있는 EC2 인스턴스 연결 브라우저 기반 클라이언트를 사용하여 연결합니다.</div> <div><input type="radio"/> EC2 인스턴스 연결 엔드포인트를 사용하여 연결 프라이빗 IPv4 주소 및 VPC 엔드포인트가 있는 EC2 인스턴스 연결 브라우저 기반 클라이언트를 사용하여 연결합니다.</div>			
<p>퍼블릭 IP 주소 34.207.107.93</p> <p>사용자 이름 인스턴스를 시작하는 데 사용되는 AMI에 정의된 사용자 이름을 입력합니다. 사용자 지정 사용자 이름을 정의하지 않은 경우 기본 사용자 이름인 ec2-user(들)를 사용합니다. ec2-user</p> <div><p>참고: 대부분의 경우 기본 사용자 이름 ec2-user(는) 정확합니다. 하지만 AMI 사용 지침을 읽고 AMI 소유자가 기본 AMI 사용자 이름을 변경했는지 확인하십시오.</p></div>			
<p>취소 연결</p>			

AWS CloudWatch

EC2에 접속된 shell 에서 아래 명령으로 큰 파일(899M)을 다운로드해보자

wget https://repo.anaconda.com/archive/Anaconda3-2023.07-2-Windows-x86_64.exe

```
[ec2-user@ip-172-31-37-173 ~]$ wget wget https://repo.anaconda.com/archive/Anaconda3-2023.07-2-Windows-x86_64.exe
--2023-09-08 07:09:21-- http://wget/
Resolving wget (wget)... failed: Name or service not known.
wget: unable to resolve host address 'wget'
--2023-09-08 07:09:21-- https://repo.anaconda.com/archive/Anaconda3-2023.07-2-Windows-x86_64.exe
Resolving repo.anaconda.com (repo.anaconda.com)... 104.16.131.3, 104.16.130.3, 2606:4700::6810:8203, ...
Connecting to repo.anaconda.com (repo.anaconda.com)|104.16.131.3|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 942216640 (899M) [application/octet-stream]
Saving to: 'Anaconda3-2023.07-2-Windows-x86_64.exe'

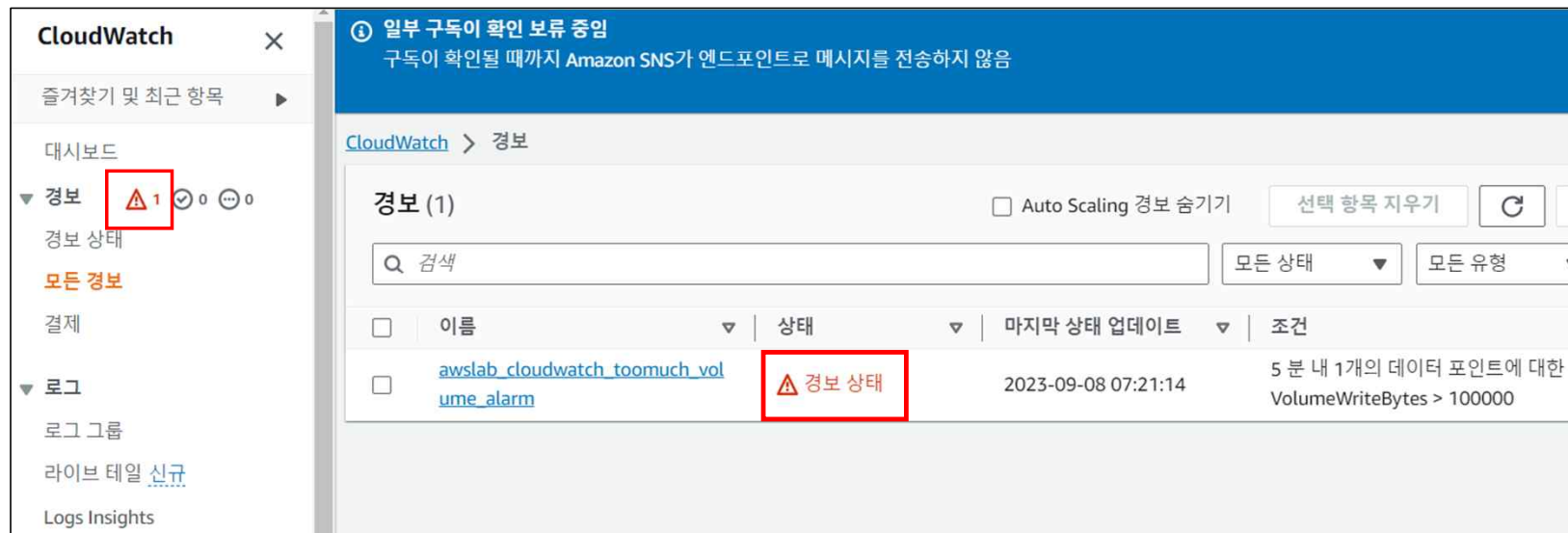
100%[=====] 942,216,640 60.6MB/s in 12s

2023-09-08 07:09:33 (74.3 MB/s) - 'Anaconda3-2023.07-2-Windows-x86_64.exe' saved [942216640/942216640]

FINISHED --2023-09-08 07:09:33--
Total wall clock time: 12s
Downloaded: 1 files, 899M in 12s (74.3 MB/s)
[ec2-user@ip-172-31-37-173 ~]$
```

AWS CloudWatch

- 잠시 기다린 후 CloudWatch에 가서 모든 경보를 보면 아래와 같이 경보 상태로 변경된 걸 볼 수 있다



(경보를 확인하면 상태는 다시 정상 상태로 바뀐다)

AWS CloudWatch

- 경보 이름 목록에서 해당 경보를 클릭하면 아래와 같은 그래프를 볼 수 있다(아래는 경보 3번 발생시킨 경우)

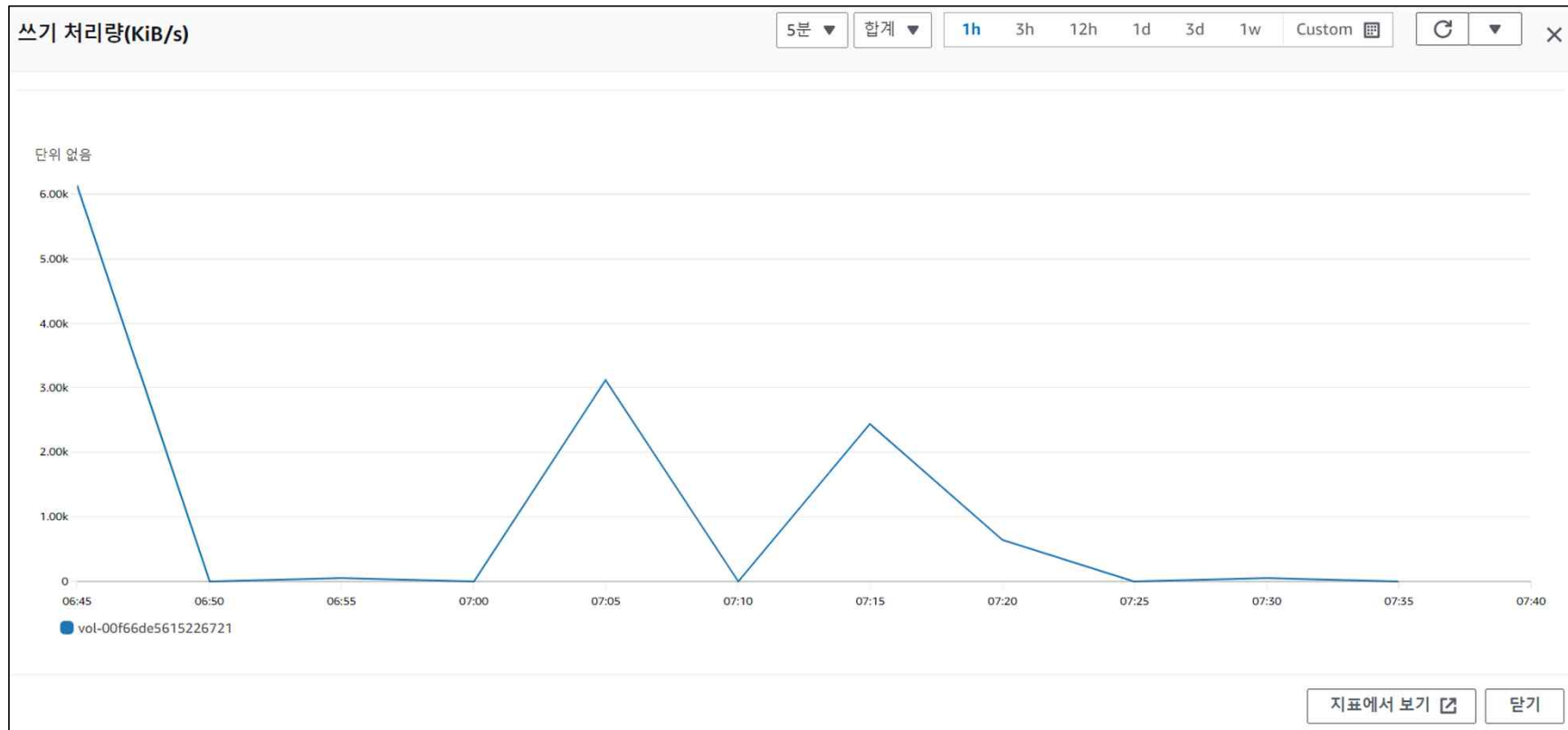


AWS CloudWatch

- 경보 그래프의 상단 메뉴에서 [보기]에서 [EBS 콘솔에서]를 클릭하면 EC2dml 볼륨이 보이는데 볼륨 ID를 클릭하면 모니터링 그래프를 볼 수 있다. [쓰기 처리량] 그래프에서 우측상단의 최대화 아이콘을 누른다

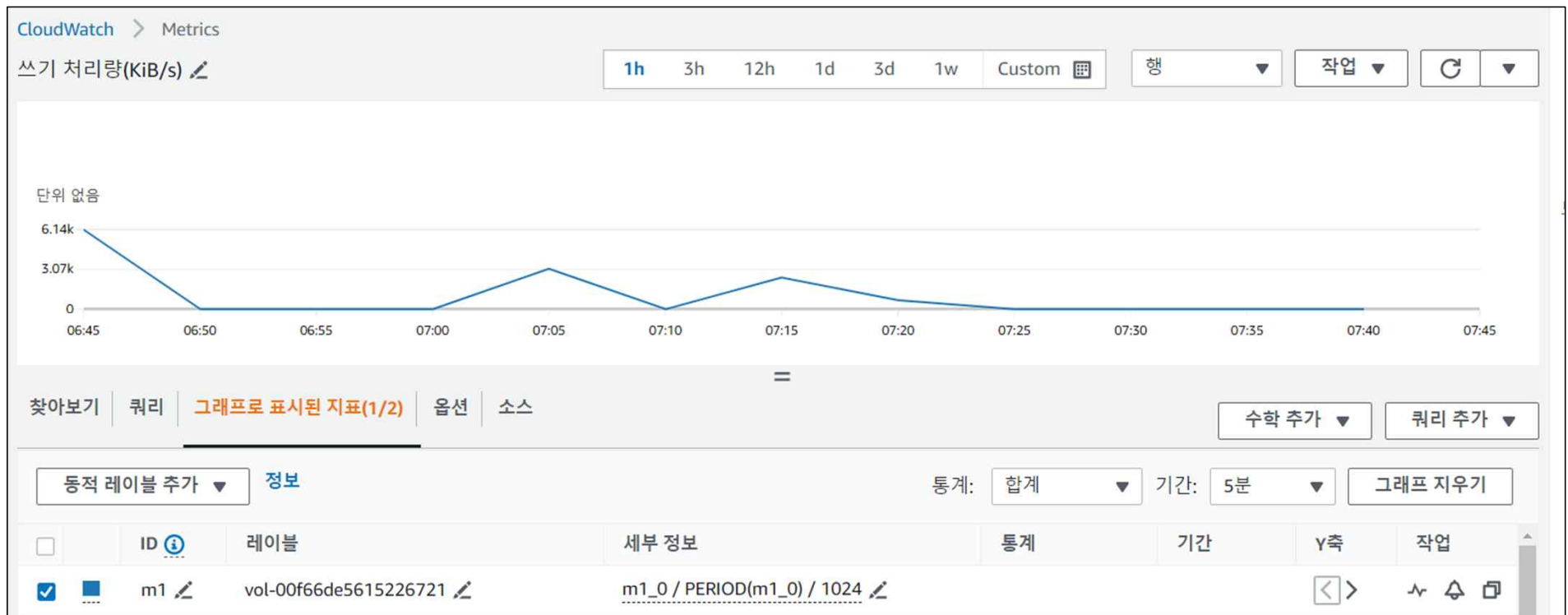


AWS CloudWatch





AWS CloudWatch



- 우측 하단의 [지표에서 보기]를 클릭하면 아래와 같은 그래프를 볼 수 있다







AWS CloudWatch

- 이메일로도 아래와 같은 알람 메일을 전달 받는다

ALARM: "awslab_cloudwatch_toomuch_volume_alarm" in US East (N. Virginia)  

  받은편지함 x

 **AWS Notifications** 나에게 ▾ 오후 4:11 (2분 전)   

You are receiving this email because your Amazon CloudWatch Alarm "awslab_cloudwatch_toomuch_volume_alarm" in the US East (N. Virginia) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [248827.87525987526 (08/09/23 07:06:00)] was greater than the threshold (100000.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Friday 08 September, 2023 07:11:14 UTC".

View this alarm in the AWS Management Console:
https://us-east-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=us-east-1#alarmsV2:alarm/awslab_cloudwatch_toomuch_volume_alarm

Alarm Details:

- Name: awslab_cloudwatch_toomuch_volume_alarm
- Description: EBS Volume에 너무 많은 쓰기가 수행되었습니다
- State Change: OK -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [248827.87525987526 (08/09/23 07:06:00)] was greater than the threshold (100000.0) (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp: Friday 08 September, 2023 07:11:14 UTC
- AWS Account: 029182223108

The End