
11

야! 아마존과 데이터공유 너도 할 수 있어!

부제: AWS Serverless 사용기

Cloud Engineering Team 김신국



발표자 : 김신국



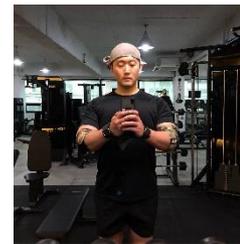
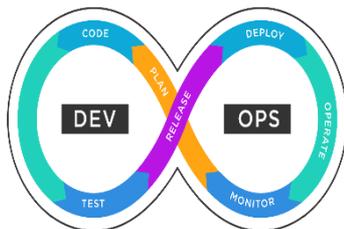
11년간 DevOps 엔지니어 , SA !



학습과 성장에 관심이 많습니다 !



3대 500을 목표로 하는 엔지니어 !



Contents List

1. **Why** 왜 쉬운 데이터 공유가 필요할까요?
2. **How** 어떻게 쉬운 공유 시스템을 구축 해야 될까요?
3. **What** 무엇이 좋아졌을까요?
4. **Learn** 무엇을 배웠을까요?

1. 왜 쉬운 데이터 공유가 필요할까요?

아마존 Global Store Launching !



변화된 환경들 ...



추가로 아마존과 협업으로 생긴 데이터..



아마존은 11번가에 궁금한 것이 많다!



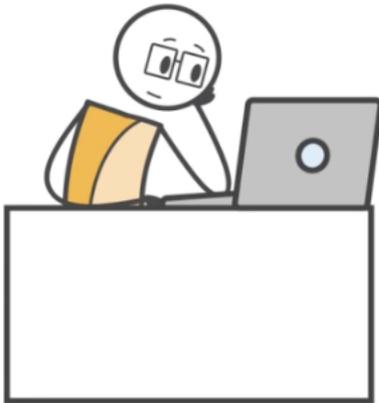
Data



amazon

1. 11번가 에서 어떤 amazon 상품 주문이 많이 돼요?
2. 자주 검색되는 키워드는 뭐예요?
3. 배송이나 환불 관련해서 잘 처리 되고 있나요?
4. 기타 ...

시작된 고민들 ...



1. 데이터를 어떻게 보내야 하지?
2. 전송한 데이터 확인은 어떻게?
3. 기존 업무 Process 대로 하고 싶은데...
4. 개발 및 구축 리소스는 적게 사용하고 싶은데...

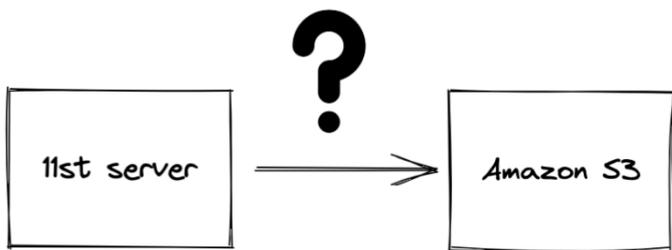
이렇게 만들어 졌으면 좋겠다~



1. 11번가의 **기존 업무 프로세스 유지!**
2. 아마존과 **빠르고 안정적인 Process 구축!**
3. 개발, 구축 **리소스는 최소한으로!**



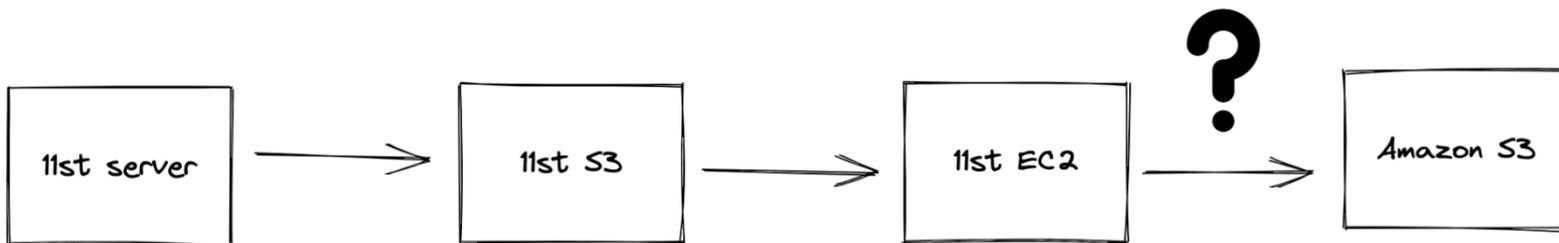
고민의 흔적들 ...



1. Amazon 에 Access key 요청 필요
2. 데이터 공유가 필요한 부서가 많아지면 Access key 에 관리 및 보안 위험 노출이 있음



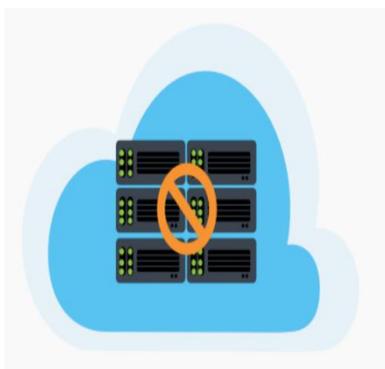
1. Access key 관리는 11번가가
2. 그럼 이것 어떻게 전달하지?



1. 개발, 배포, 인프라 환경 구축...
2. 좀 더 간편하게 할 수 없을까?

2. 어떻게 쉬운 공유 시스템을 구축 해야 될까요?

저의 선택은?



serverless

Serverless 에 특징은?

What is Serverless?



No infrastructure provisioning,
no management



Automatic scaling

Pay for value



Highly available and secure



AWS의 수많은 서버리스 옵션



선택된 서버리스는?



Lambda



SQS



S3 Bucket



API Gateway



EFS

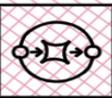
왜? 선택이 되었는가?



Lambda

내가 사용한 만큼 요금을 지불한다

서버를 프로 비저닝 하거나 관리하지 않고도 코드를 실행할 수 있게 해주는 컴퓨팅 서비스



SQS

발생한 이벤트에 대한 안정적인 보관 및 처리를 위해

내구력 있고 가용성이 뛰어난 대기열, 발생하는 이벤트를 순차적이고 안정적 처리를 위한 서비스



API Gateway

사용자가 API 호출을 통해 원하는 동작을 쉽게 수행하는 환경을 만들기 위해

어떤 규모에서든지 API를 손쉽게 생성, 배포, 유지 관리를 위한 서비스



S3 Bucket

아마존과 공유할 데이터를 저장 하고 저장된 시점에 이벤트 발생 처리를 위해

대용량 스토리지 서비스 및 파일 관리 및 파일 생성시 이벤트 처리로 유동적인 인프라 구성하는 서비스



EFS

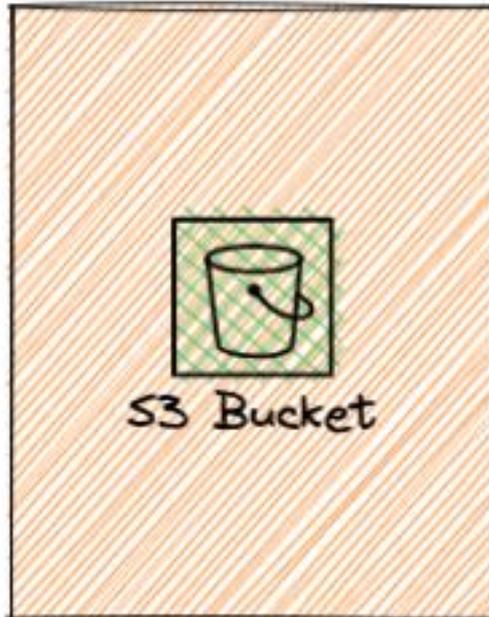
파일별로 용량 차이가 많이 나는데 사용할 데이터를 유연하게 관리

파일을 추가하고 제거할 때 자동으로 확장, 축소 관리가 되어 프로 비저닝이 필요 없이 데이터 관리를 할 수 있는 서비스

전체 구성도



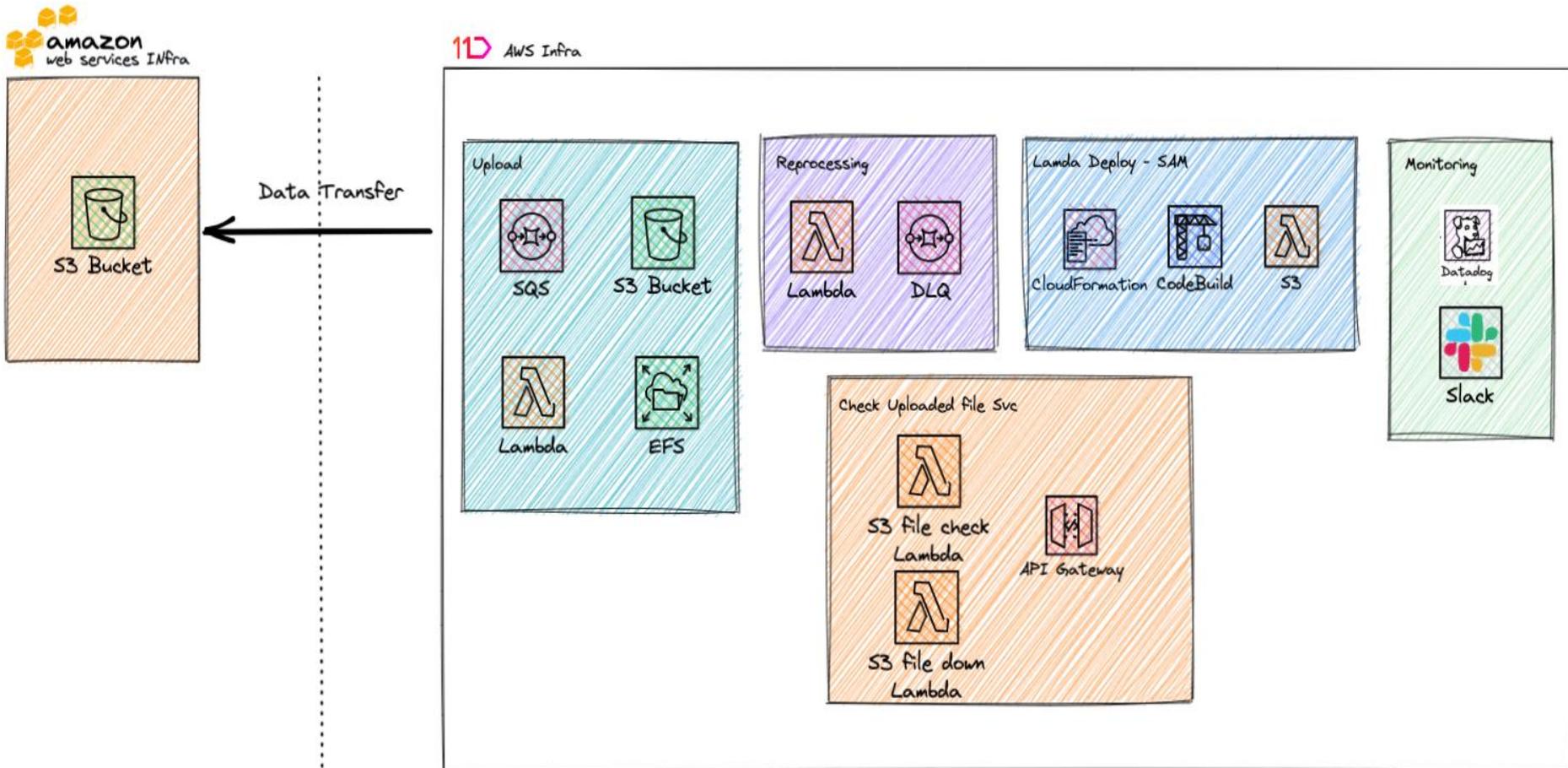
amazon
web services Infra



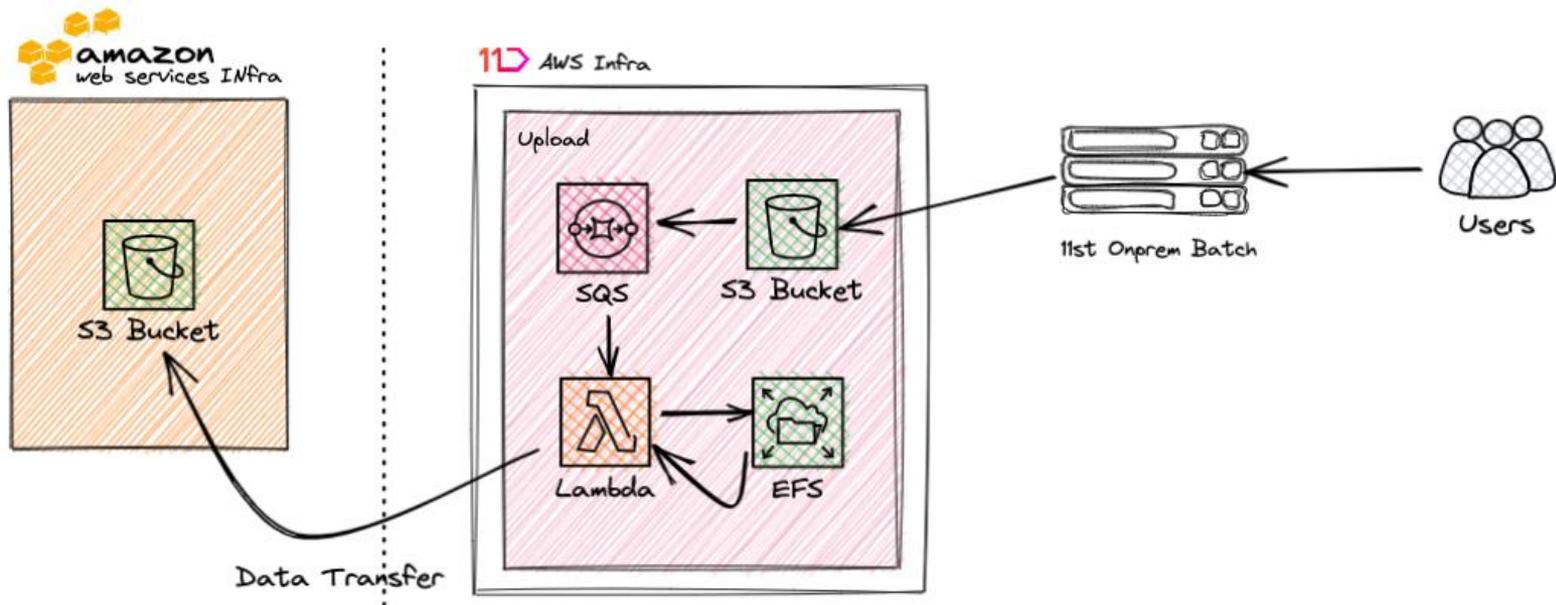
11D AWS Infra



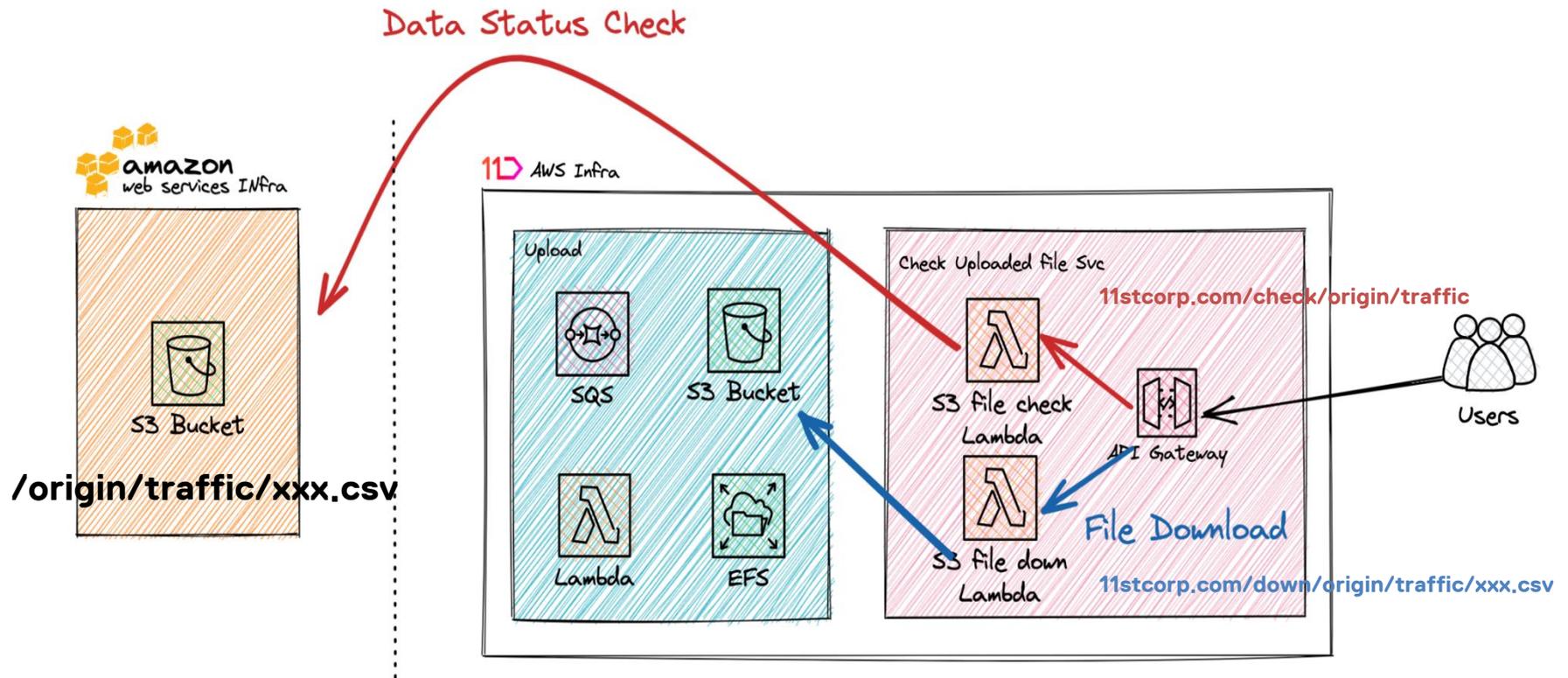
전체 구성도 (상세 구성)



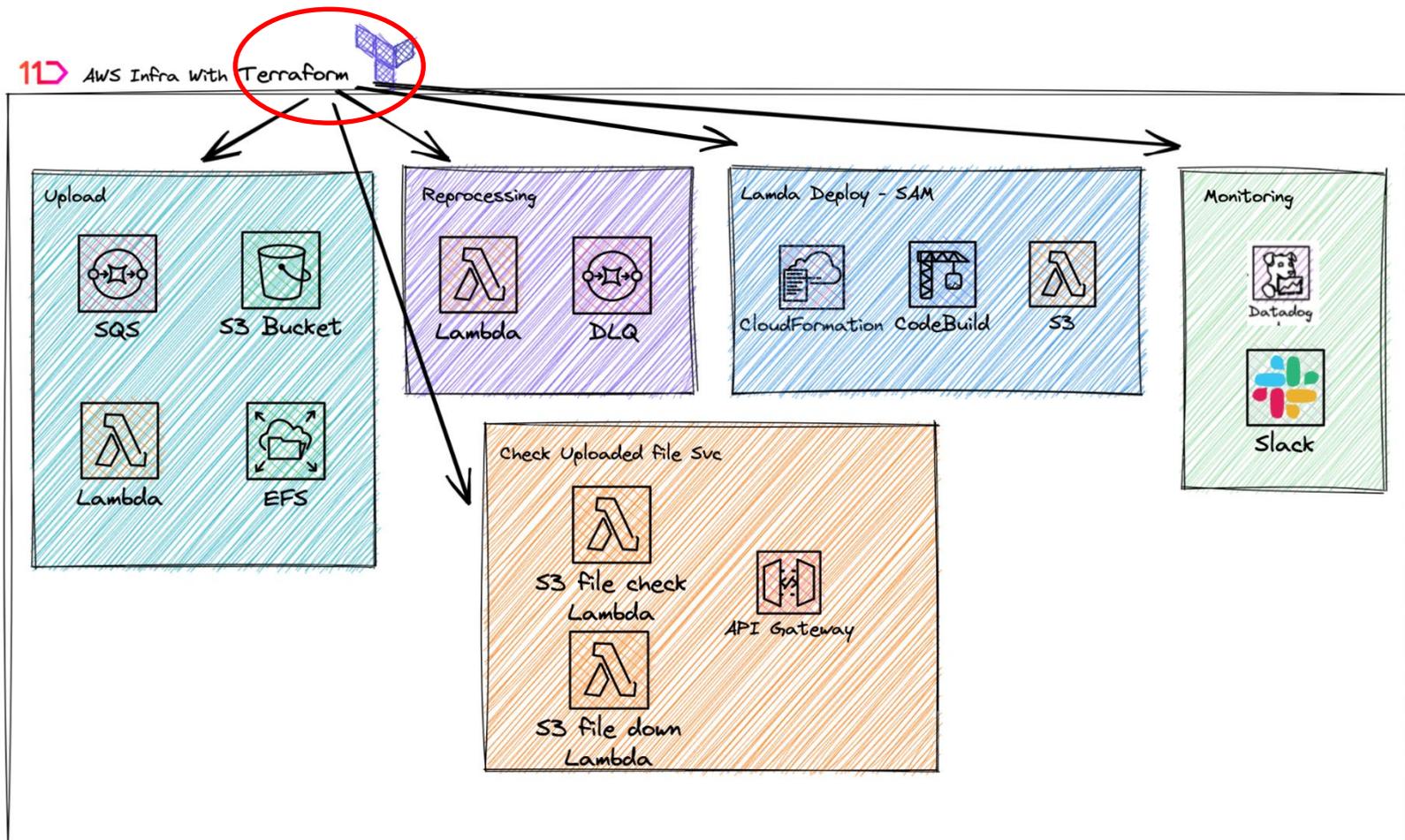
아마존에 파일을 어떻게 올리지?



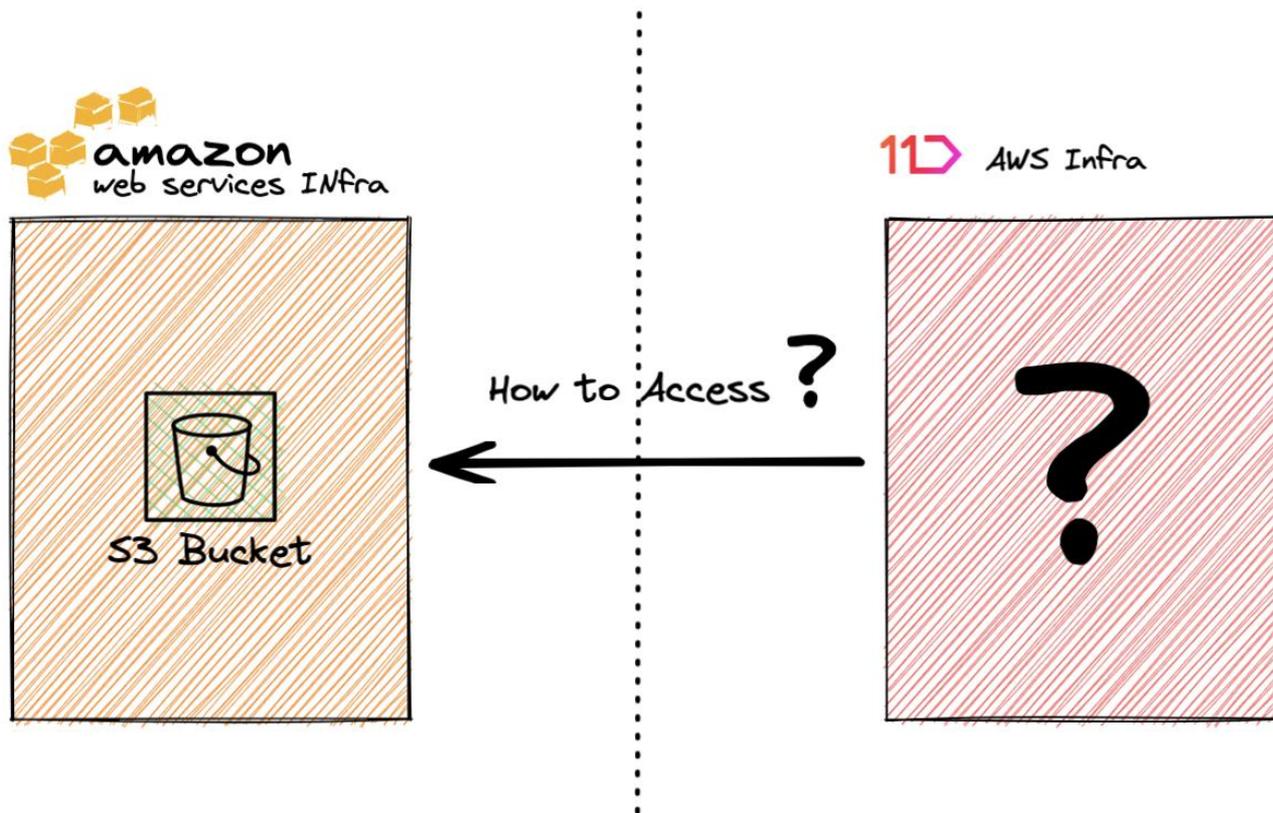
아마존에 공유된 파일은 어떻게 확인할까요?



인프라 구성은 어떻게 하고 있을까요?



아마존에 버킷에 어떻게 접근을 할까요? - 1



아마존에 버킷에 어떻게 접근을 할까요? - 2

Assume Role!

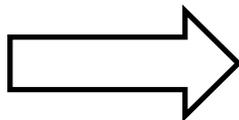
역할을 생성하여 IAM 사용자에게 권한 위임

[PDF](#) | [RSS](#)

IAM 역할을 사용해 AWS 리소스에 대한 액세스 권한을 위임할 수 있습니다. IAM 역할을 사용해 신뢰하는 계정과 다른 AWS 신뢰받는 계정 간에 신뢰 관계를 설정할 수 있습니다. 신뢰하는 계정은 액세스되는 리소스를 소유하고 신뢰받는 계정은 리소스에 대한 액세스가 필요한 사용자를 저장합니다. 그러나, 다른 계정이 해당 계정의 리소스를 소유할 수 있는 가능성이 있습니다. 예를 들어, 신뢰받는 계정은 신뢰 계정이 Amazon S3 버킷의 새로운 객체를 생성하는 것처럼 새로운 리소스를 생성하도록 허용할 수 있습니다. 이러한 경우, 리소스를 생성하는 계정은 리소스를 소유하고 누구에게 리소스에 대한 액세스를 부여할지 제어합니다.

Data 올리도록
권한을 줄게요!

amazon



네! Data 는
그곳에 올릴게요!

S3 운영은 어떻게 하세요?

Life Cycle !!

스토리지 수명 주기 관리

[PDF](#) | [RSS](#)

수명 주기 동안 객체가 비용 효율적으로 저장되도록 관리하려면 해당 *Amazon S3 수명 주기*를 구성합니다. *S3 수명 주기 구성*은 Amazon S3이 객체 그룹에 적용하는 작업을 정의하는 일련의 규칙입니다. 다음과 같은 두 가지 유형의 작업이 있습니다.

- **전환 작업** - 이 작업은 객체가 다른 스토리지 클래스로 전환되는 시기를 정의합니다. 예를 들어, 생성 후 30일이 지나면 객체를 S3 STANDARD-IA 스토리지 클래스로 전환하거나 생성 후 1년이 지나면 객체를 S3 Glacier Flexible Retrieval 스토리지 클래스에 아카이브하도록 선택할 수 있습니다. 자세한 정보는 [Amazon S3 스토리지 클래스 사용](#)을 참조하십시오.

수명 주기 전환 요청과 관련된 비용이 있습니다. 요금 정보는 [Amazon S3 요금](#)을 참조하세요.

- **만료 작업** - 이 작업은 객체가 만료되는 시기를 정의합니다. Amazon S3에서 만료된 객체를 자동으로 삭제합니다. 수명 주기 만료 비용은 선택한 객체 만료 시점에 따라 달라집니다. 자세한 정보는 [객체 만료](#)을 참조하십시오.

S3 운영은 어떻게 하세요?

Event Notification

General configuration

Event name
origin_event

Prefix - optional
Limit the notifications to objects with key starting with specified characters.

Suffix - optional
Limit the notifications to objects with key ending with specified characters.

Event types

Specify at least one event for which you want to receive notifications. For each group, you can choose an event type for all events, or you can choose one or more individual events.

Object creation

All object create events
s3:ObjectCreated:*

Put
s3:ObjectCreated:Put

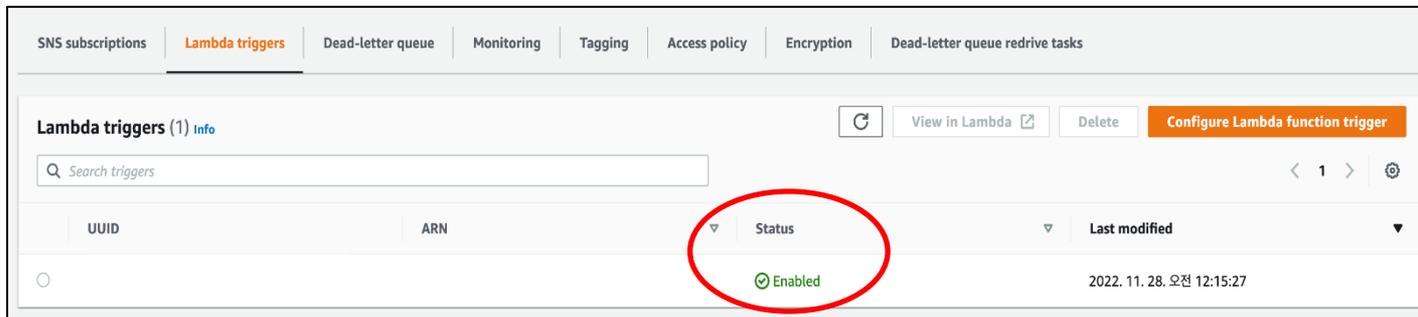
Post
s3:ObjectCreated:Post

Copy
s3:ObjectCreated:Copy

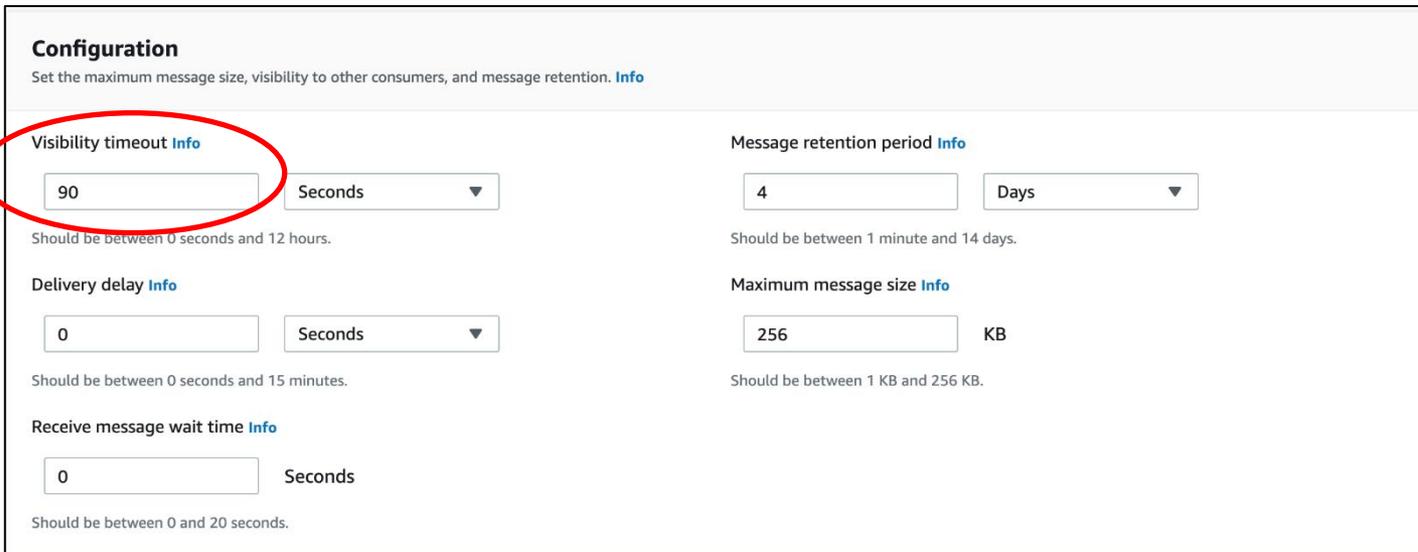
Multipart upload completed
s3:ObjectCreated:CompleteMultipartUpload

SQS 운영은 어떻게 하세요?

Lambda 와 Event Trigger 연결



The screenshot shows the AWS Lambda console interface for a specific trigger. The top navigation bar includes tabs for SNS subscriptions, Lambda triggers (selected), Dead-letter queue, Monitoring, Tagging, Access policy, Encryption, and Dead-letter queue redrive tasks. Below the navigation, there's a section for 'Lambda triggers (1) Info' with a search bar and a table. The table has columns for UUID, ARN, Status, and Last modified. The 'Status' column for the single entry is circled in red and shows 'Enabled' with a green checkmark. The 'Last modified' column shows the date and time '2022. 11. 28. 오전 12:15:27'. There are also buttons for 'View in Lambda', 'Delete', and 'Configure Lambda function trigger'.



The screenshot shows the 'Configuration' page for an Amazon SQS queue. The title is 'Configuration' and the subtitle is 'Set the maximum message size, visibility to other consumers, and message retention. Info'. There are four configuration sections: 'Visibility timeout Info' (with a value of 90 and unit of Seconds, circled in red), 'Message retention period Info' (with a value of 4 and unit of Days), 'Delivery delay Info' (with a value of 0 and unit of Seconds), and 'Receive message wait time Info' (with a value of 0 and unit of Seconds). Each section includes a brief description of the parameter's range.

SQS 운영은 어떻게 하세요?

DLQ 란?

Amazon SNS 배달 못한 편지 대기열(DLQ)

[PDF](#) | [RSS](#)

배달 못한 편지 대기열은 Amazon SNS 구독이 구독자에게 성공적으로 배달할 수 없는 메시지를 저장하는 Amazon SQS 대기열입니다. 클라이언트 오류 또는 서버 오류로 인해 배달할 수 없는 메시지는 추가 분석 또는 재처리를 위해 배달 못한 편지 대기열에 보관됩니다. 자세한 정보는 [구독에 대한 Amazon SNS 배달 못한 편지 대기열 구성](#) 및 [Amazon SNS 메시지 전송 재시도](#)에서 확인하세요.

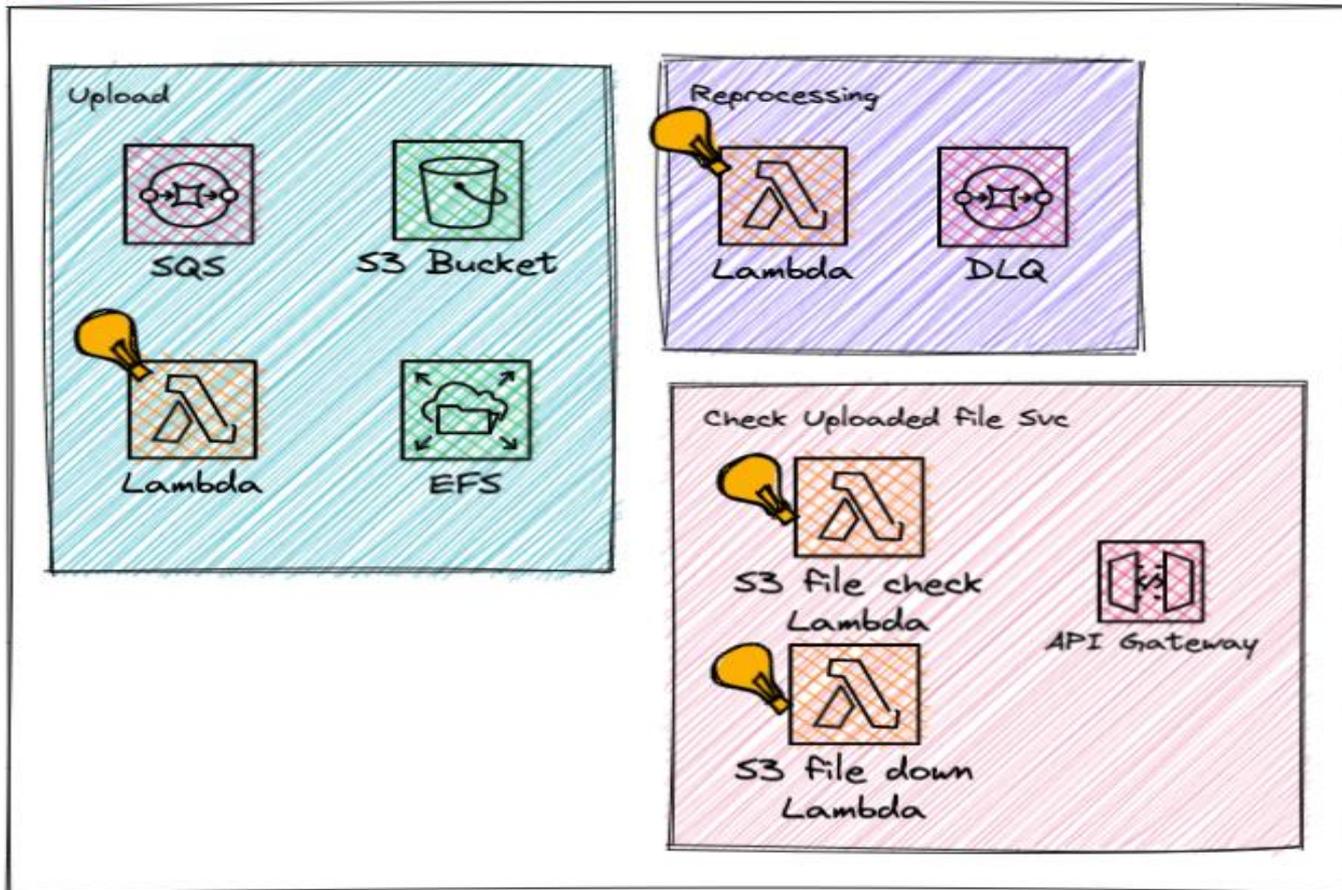
DLQ 설정!

The screenshot shows the AWS Management Console interface for configuring a Dead-letter queue (DLQ) for an SNS subscription. The 'Dead-letter queue' tab is selected, and the configuration is as follows:

Queue	Maximum receives
arn:aws:sqs:ap-northeast-2::test-sqs-dlq	3

Lambda 구성은?

11D AWS Infra

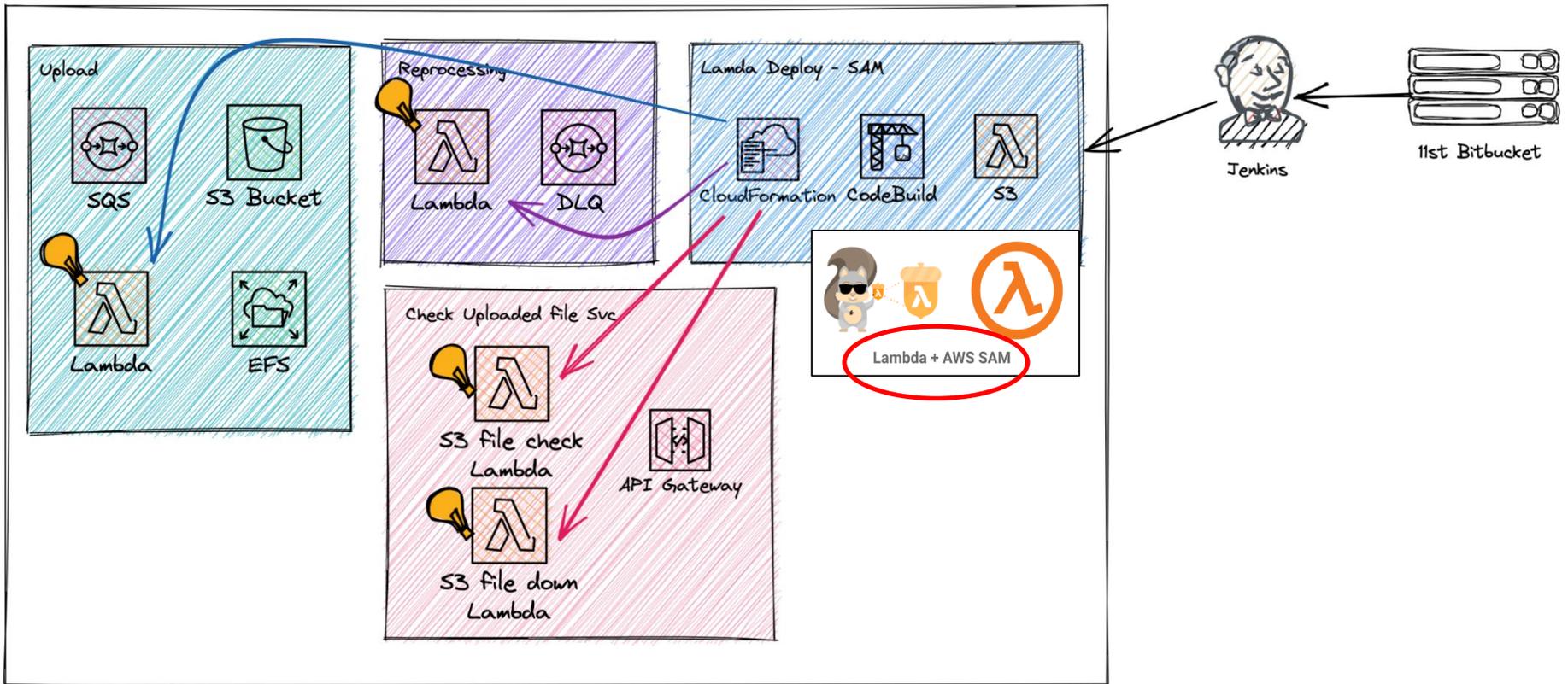


Lambda 개발 환경 및 설정은?

- 개발언어: Python 3.8
- Trigger Event : SQS, API GW
- Execution Role:
 - Assume Role
 - SQS Role
 - CloudFormation Role (with SAM)
 - S3 Role
- Concurrency: 동시성 설정
- Aliases and Versions: 람다 버전 관리
- File System : EFS

Lambda 배포는 어떻게 하세요?

11D AWS Infra



Lambda 에서 SAM 을 어떻게 사용하나요?

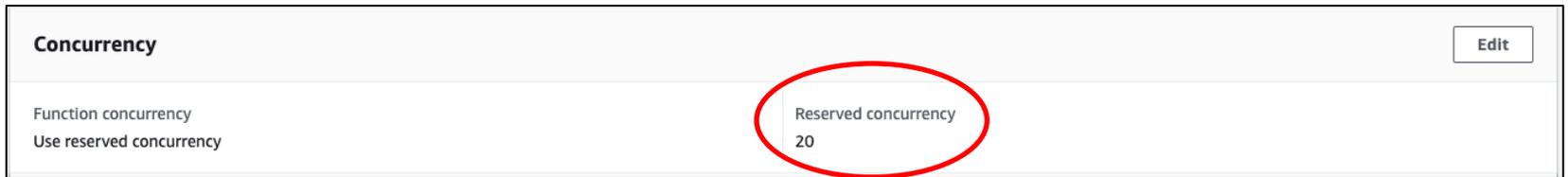


AWS SAM(Serverless
Application Model)

- SAM 의 장점
 - ✔ Template.yaml 파일로 람다 리소스 설정관리
 - ✔ Local 에서 람다 개발과 Test 지원
 - ✔ Mock Event 생성 및 람다 실행
 - ✔ 트래픽 분산을 통한 무 중단 배포 수행

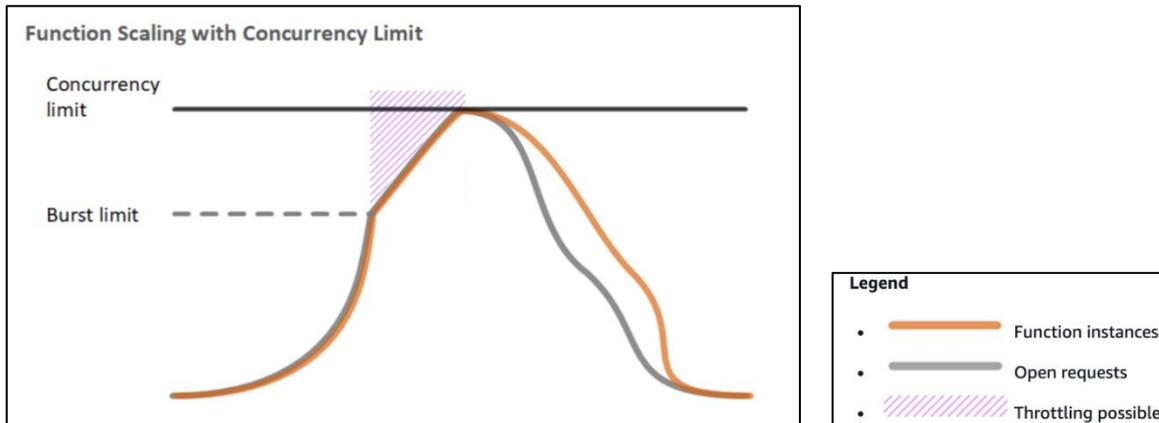
Lambda 운영 어떻게 하세요?

동시성 관리



람다 동시성: 내가 지정한 만큼 **Lambda instance** 가 증가함

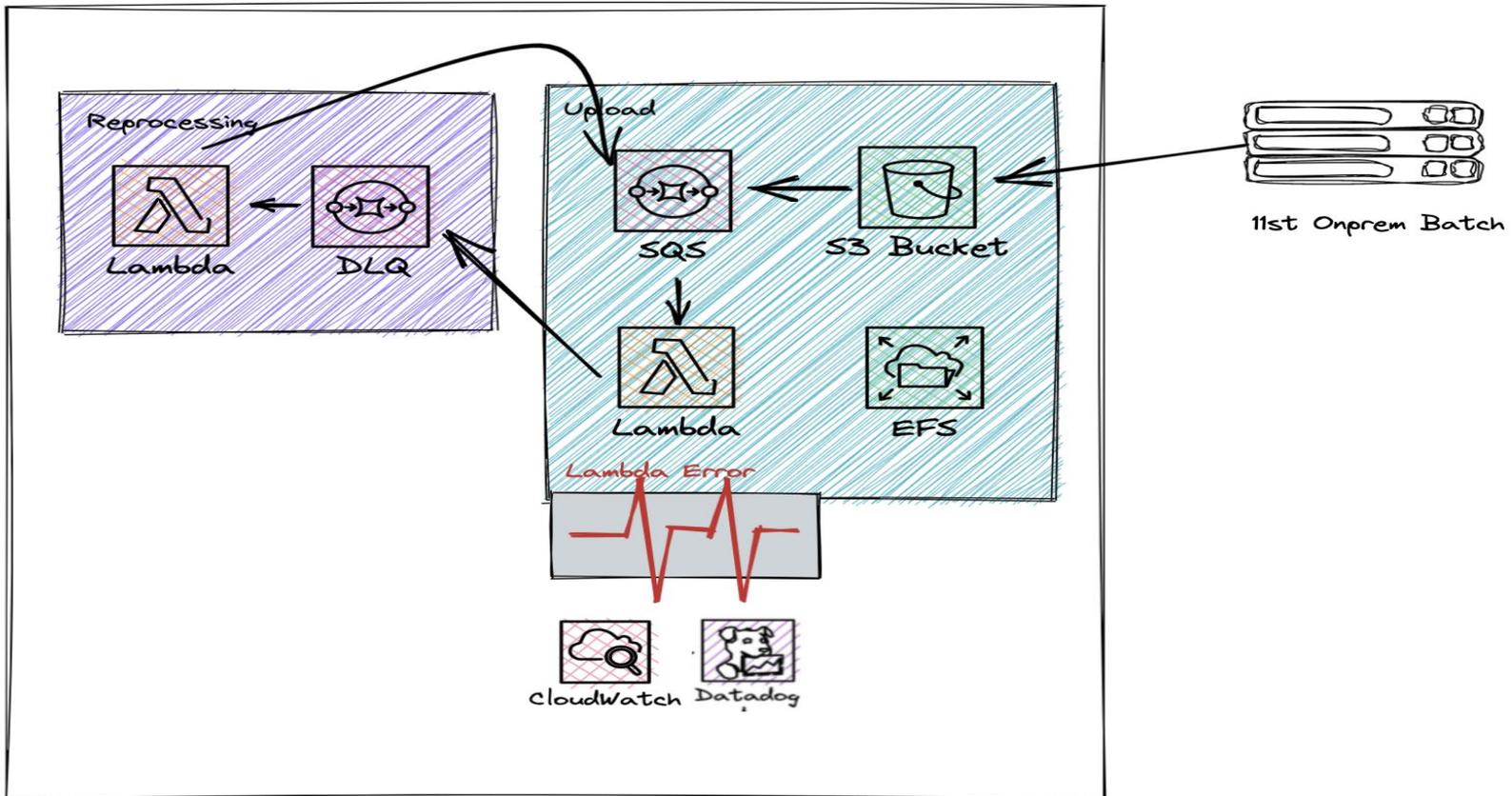
- 동시성 limit 이 초과 되면 **Lambda Error** 발생!



Lambda 운영 어떻게 하세요?

Upload 실패 시 재처리

11D AWS Infra



API GW 운영 어떻게 하세요?

Private API

Amazon API Gateway에서 **프라이빗 API 생성**

[PDF](#) | [RSS](#)

Amazon API Gateway를 사용하면 **인터페이스 VPC 종단점**을 사용하여 Amazon VPC의 Virtual Private Cloud에서만 액세스할 수 있는 프라이빗 REST API를 생성할 수 있습니다. VPC에서 생성하는 엔드포인트 네트워크 인터페이스입니다.

여러 AWS 계정은 물론 선택한 VPC 및 VPC 엔드포인트에서 API에 액세스하는 것을 **리소스 정책**으로 허용하거나 거부할 수 있습니다. 각 엔드포인트를 사용하여 여러 개의 프라이빗 API에 액세스할 수 있습니다. 또한 AWS Direct Connect를 사용하여 온프레미스 네트워크에서 Amazon VPC에 연결한 다음 그 연결을 통해 프라이빗 API에 액세스할 수도 있습니다.

외부에 노출하지 않고! 오직 내부 인원만 사용하도록 구성!

API GW 운영 어떻게 하세요?

API Key

API Gateway 콘솔을 사용하여 **API 키** 설정

[PDF](#) | [RSS](#)

API 키를 설정하려면 다음 작업을 수행합니다.

- API 메서드를 구성해 API 키를 요구합니다.
- 리전의 API에 대해 API 키를 생성하거나 가져옵니다.

불필요한 호출 방지를 위해!

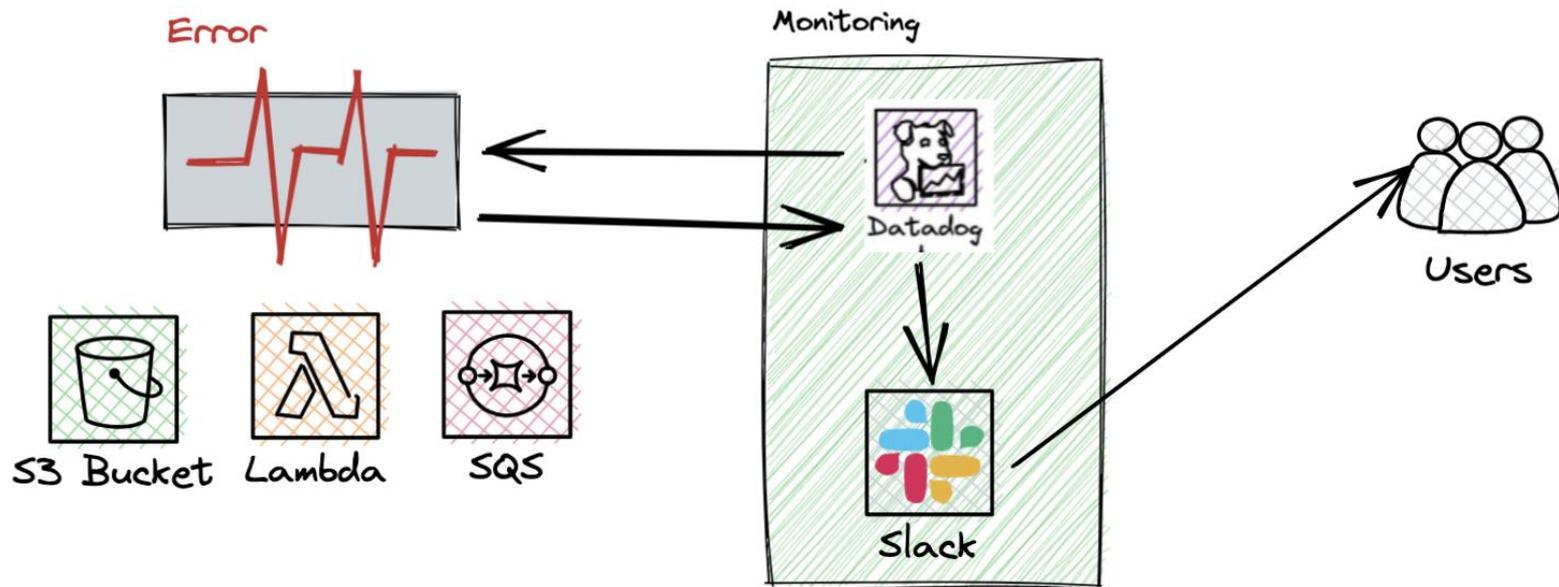
API Gateway 콘솔을 사용하여 사용량 계획 생성, 구성 및 테스트

[PDF](#) | [RSS](#)

사용량 계획을 생성하기 전에 원하는 **API 키를 설정**했는지 확인합니다. 자세한 내용은 [API Gateway 콘솔을 사용하여 API 키 설정](#) 단원을 참조하세요.

이 단원에서는 API Gateway 콘솔을 사용하여 사용량 계획을 생성하고 사용하는 방법을 설명합니다.

모니터링은?



3. 무엇이 좋아졌을까요?

이런 점이 좋아졌어요! – 비용

▼ 계산 표시

단위 변환

할당된 메모리 크기: $512 \text{ MB} \times 0.0009765625 \text{ GB}(1\text{MB 기준}) = 0.5 \text{ GB}$ 기준 해당 없음
 할당된 임시 스토리지의 크기: $512 \text{ MB} \times 0.0009765625 \text{ GB}(1\text{MB 기준}) = 0.5 \text{ GB}$ 기준 해당 없음

요금 계산

$10,000 \text{ 요청} \times 4,000 \text{ 밀리초} \times 0.001 \text{ 밀리초-초 변환 인수} = 40,000.00 \text{ 총 컴퓨팅 시간(초)}$
 $0.50 \text{ GB} \times 40,000.00 \text{ 초} = 20,000.00 \text{ 총 컴퓨팅 크기(GB/s)}$
 $20,000.00 \text{ 초당 GB} - 400000 \text{ 프리 티어 기가비트-초} = -380,000.00 \text{ 초당 GB}$
 $\text{Max} (-380000.00 \text{ 초당 GB}, 0) = 0.00 \text{ 총 청구 대상 GB-초}$
 Tiered price for: 0.00 초당 GB
 총 티어 비용 = 0.0000 USD(월별 컴퓨팅 요금)
 $10,000 \text{ 요청} - 1000000 \text{ 프리 티어 요청} = -990,000 \text{ 월별 청구 가능 요청 수}$
 $\text{Max} (-990000 \text{ 월별 청구 가능 요청 수}, 0) = 0.00 \text{ 총 월별 청구 가능 요청 수}$
 $0.50 \text{ GB} \times 0.5 \text{ GB(추가 요금 없음)} = 0.00 \text{ 월별 청구 가능한 임시 스토리지 GB}$

Lambda 비용 - 프리 티어 포함 (월별): 0.00 USD

• 람다

💡 한달 기준

- 사용량: 10k (현재 기준)
- 비용: 0원
- 100만 건 부터 요금 발생 시작 (3만원)

▼ 계산 표시

단위 변환

EC2 Instance Savings Plans rate for m5.large in the US East (Ohio) for 1 Year term and No Upfront is 0.06 USD
 Hours in the commitment: 365 days * 24 hours * 1 year = 8760.0000 hours
 Total Commitment: 0.06 USD * 8760 hours = 525.6000 USD
 Upfront: No Upfront (0% of 525.6) = 0.0000 USD
 Hourly cost for EC2 Instance Savings Plans = (Total Commitment - Upfront cost)/Hours in the term: (525.6 - 0)/8760 = 0.0600 USD
 *Please note that you will pay an hourly commitment for Savings Plans and your usage will be accrued at a discounted rate against this commitment.

요금 계산

$1 \text{ 인스턴스} \times 0.06 \text{ USD} \times 770 \text{ 월별 시간} = 43.80 \text{ USD(월별 인스턴스 절감 비용)}$

Amazon EC2 Instance Savings Plans 인스턴스 (월별): 43.80 USD

• EC2

💡 한달 기준

- 사용량: 24시간
- 비용: 5만원 (1대당: m5.large type)
- EBS 가격 제외

이런 점이 좋아졌어요! - 업무



Technology

- 내가 쓴 만큼 비용을 지불하는 **고효율 저비용 시스템 구축**
- 서버리스 도입 및 **운영 경험 축적**



Trust

- 11번가와 아azon이 빠르고 **안정적인 협업**을 할 수 있는 인프라 구축 완료



Convenience

- 기존 업무 Process 와 크게 다르지 사용감을 제공하여 **업무 효율성 증가**

4. 무엇을 배웠을까요?



한 일

- 1) Serverless 서비스를 사용한 고효율 저비용 서비스 구축 및 운영
- 2) 아마존과 안정적인 데이터 공유



배운 것들

- 1) Serverless Resource 활용
- 2) Lambda 운영/배포/모니터링 구축 및 활용 (배포는 SAM 으로 해보세요!)
- 3) Lambda 운영을 통해 동시성 조절에 대한 최적화 필요!
- 4) 재처리 로직은 DLQ 를 사용하는 것이 안전함!



앞으로 할 일

- 1) 대규모 트래픽에 대한 대비
- 2) 서비스 확장에 대한 대비

Q&A

E-mail) ksbang39@sk.com

Thank you