

# Python으로 AWS를 활용한 분산 처리

김민준

<https://kimminjun.dev>

supremejoon@daum.net

# Contents

- 프롤로그
- AWS Lambda
- Amazon SNS, Amazon SQS
- AWS API Gateway
- CloudWatch
- 에필로그

# 프롤로그

오류가 발생했는데요. 죄송한데 확인 좀 해주시겠어요?  
업체를 통해서 JAVA로 개발된 프로그램인데요.  
업체가 대응이 너무 느려서... 혹시 한번 보기만 해주시겠어요?  
서버 정보는... %\$%\$^#%#



# 프롤로그

아.. 정신없네요...  
실행되고 있는 이 많은 Agent 프로그램이 업체 것인가요 ?  
관리프로그램은 존재하지 않는 것인가요 ?  
이건 제가 볼 수 없어요..



# 프로로그

네..ㅠㅠ



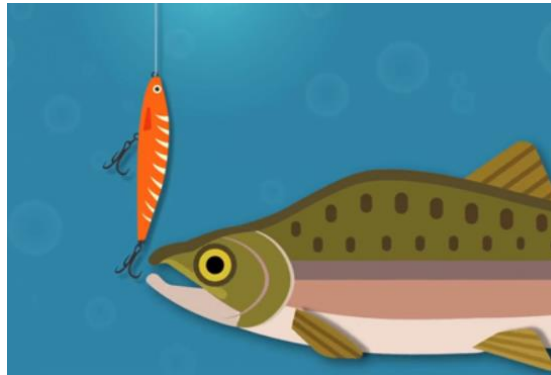
# 프로로그

제가 확인해보니 업체는 관세청의 API를 주기적으로 호출을...  
그리고 그것을 수많은 Agent 프로그램으로 분할 처리  
이렇게 서버에서 돌다가 하나라도 Agent 가 죽을 경우에는  
그에 따른 대응방법으로는 재실행밖에 없는데,  
그 방법도 수동이네요? 이거 비싼 건가요 ?



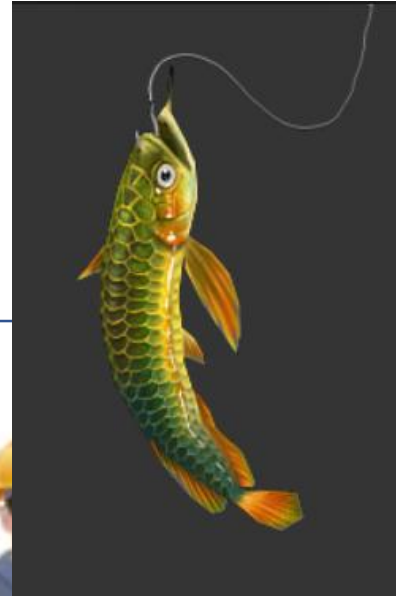
# 프롤로그

아~ 그래요? 그럼 직접 구현해주실래요? C#으로 되시죠?  
기존에 몇 개 있잖아요? 비슷하게 해서 하면 될 것 같은데요?  
저도 관세청에 API 정보를 좀 보니까 어려워 보이진 않네요.  
진행해보시죠.



# 프로로그

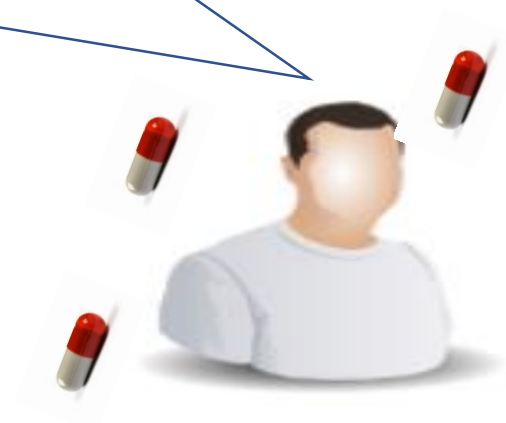
ㅍㅍ  
네. 진행하시죠.



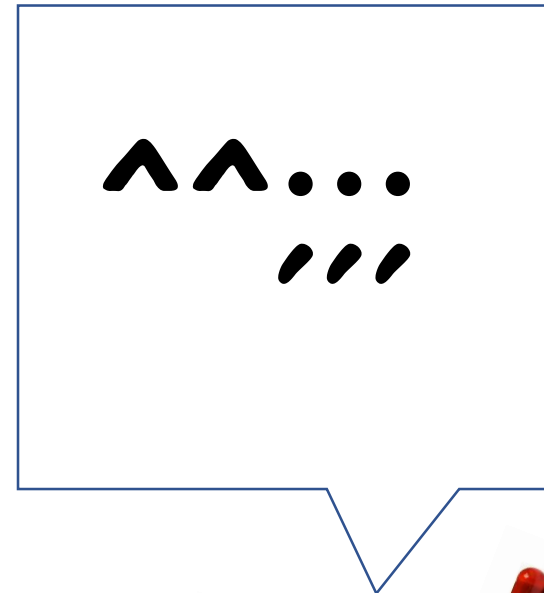


# 프롤로그

(속닥속닥) 서버리스...



# 프로로그



**어쩌다 이렇게..한 배를 타고 출발...**



# 프롤로그

근데 AWS Lambda를 닷넷으로 하려니..  
Visual Studio 을 구매해 주셔야겠네요ㅎㅎ  
사주세요. 4달러 4달러 4달러~~~~



# 프롤로그

ㅈㅂ^^;;  
다른 방법?



아놔 XX..



ㅋㅋㅋ



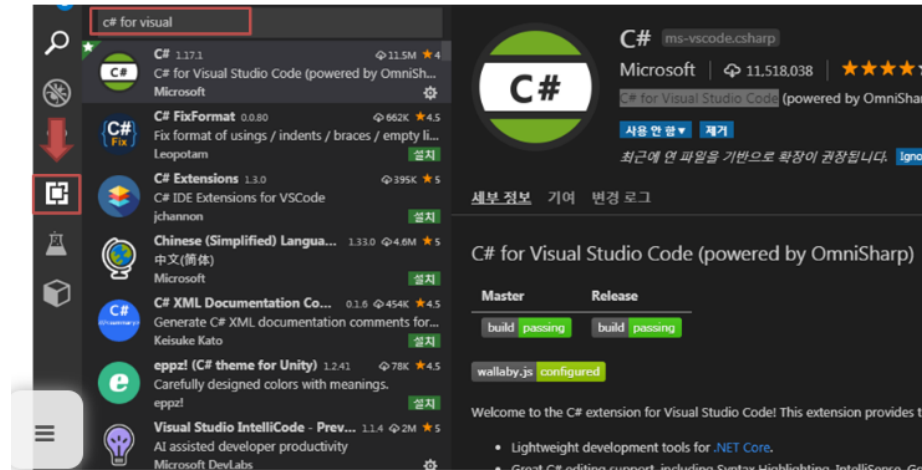
# 결국 Visual Studio Code 로 샘플 작업을 진행했습니다....

Visual Studio Code 로 .NET Core 2.1 사용하여 AWS Lambda 만들기

2019.05.08 17:39

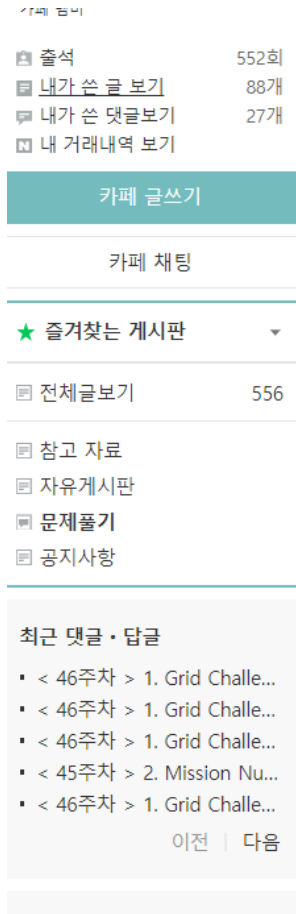
Visual Studio Code 로 .NET Core 2.1 사용하여 AWS Lambda 만들기

1. Visual Studio Code 다운로드
2. 확장 프로그램 설치 - C# for Visual Studio Code



(개인 블로그에 있는 Visual Studio Code 로 .NET Core 2.1 사용하여 AWS Lambda 만들기 글)

# 그런데 Python으로 알고리즘 문제 풀고 놀던 때라... Python 으로 해볼까? 했죠..



```
>https://www.hackerrank.com/challenges/minimum-absolute-difference-in-an-array/problem
#!/bin/python3

import math
import os
import random
import re
import sys

# Complete the minimumAbsoluteDifference function below.
def minimumAbsoluteDifference(arr):

    arr.sort()
    result = min([abs(arr[item+1] - arr[item]) for item in range(len(arr)-1)])
    return result

if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')

    n = int(input())

    arr = list(map(int, input().rstrip().split()))

    result = minimumAbsoluteDifference(arr)

    fptr.write(str(result) + '\n')

    fptr.close()
```

(알고리즘 문제풀기 스터디를 Python 으로 했던 사진..)

# 프롤로그

오ㅋ



그럼 Python으로 해도 될까요?

Python으로 크롤링하고 그러던데..  
(궁시렁 궁시렁...)



음ㅋ 파이썬





# 이렇게 Visual Studio Code로 Python 작업 완료!

## 이제 AWS Lambda 에 올려볼까...?



함수 코드 정보

코드 입력 유형

코드 인라인 편집

코드 인라인 편집

.zip 파일 업로드

Amazon S3에서 파일 업로드

런타임

Python 3.6

핸들러 정보

lambda\_function.lambda\_handler

Environment

lambda\_function.py

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     return {
6         'statusCode': 200,
7         'body': json.dumps('Hello from Lambda!')
8     }
9
```

**그런데... 여기서 의문점!**  
**라이브러리를 함수 하나하나에 다 같이 올려야 하나 ?**  
**너무 불편해 !!!**



Layers가 존재하는 것을 확인.  
특징은 버전별로 앞서 치기 개념이다.  
(하지만 과거 버전도 Lambda에서는 사용이 가능하다.)

우선 pip list 로 사용한 라이브러리들을 확인하고  
경로에 들어가 복사하여 하나의 폴더에 모아 넣었다.

그런데 레이어를 올리는 폴더구조에도 규칙이 있음.

AWS Lambda



• Python – python, python/lib/python3.7/site-packages(사이트 디렉터리)  
예 Pillow

대시보드

애플리케이션

함수

계층

```
pillow.zip
├── python/PIL
└── python/Pillow-5.3.0.dist-info
```



**Layers 를 추가하고, AWS Lambda 를 실행 !!!**

**그런데.. AWS Lambda 에서 Error 발생 ㅠ ㅠ**



Layers 를 추가하고, AWS Lambda 를 실행 !!!

그런데.. AWS Lambda 에서 Error 발생 ㅠ ㅠ



최강 기만러 갓대명님의 농락 >

SAP을 SOAP(XML)로 데이터를 끌어오는데, Zeep 라는 라이브러리를 사용.  
Zeep 의 github에서 동일한 이슈가 존재하는지 확인해봤다.

(Zeep 라이브러리 내부에 lxml이 존재한다.)

## lxml not supported in AWS Lambda server-less framework

New issue

#669

 Closed

ghost opened this issue on 25 Jan 2018 · 4 comments



ghost commented on 25 Jan 2018

+ 😊 ...

I did read the guidelines for filing bug reports but this one is cross-version, all code affected.

Basically, I am interested in running an AWS Lambda function that implements the zeep library, however, there is no straight forward way to utilize the lxml library in AWS Lambda. See this [StackOverflow thread](#) for more context, but it has to do with the shared libraries for lxml not being present in AWS Lambda during runtime. There are some pretty hacky solutions and work arounds but in my mind, best keep Lambda simple as possible since that's precisely one of the things its for.

The one answer that does intrigue me from that thread is using Python's builtin xml instead of lxml. I was looking to play with a pull request, but hit an immediate blocker on syntax errors in `zeep.asyncio.transports` under Python3.6, when I pulled the current master.

Can anyone think of any reason to not attempt this switch? Is there a current branch or tag I should be originating from in terms of operating under Python3.6?

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone


No milestone

Notifications

 Subscribe



# 해결방법도 제시가 되어 있었다. 직접 Amazon Linux 환경에서 빌드한 파일을 올려준 사람이 있었다.




mvantellingen commented on 20 Feb 2018

Owner


The built-in xml lib in python doesn't support namespaces (properly) so that is a no-go. So let's hope that an lambda compatible lxml is released.

I have to consider this out of scope for this project, so i'm closing it



mvantellingen closed this on 20 Feb 2018


---




lucasrcosta commented on 29 May 2018

+ 😊 ...

This worked for me: <https://github.com/JFox/aws-lambda-lxml>

 2



sc-pulsion commented 3 days ago

+ 😊 ...

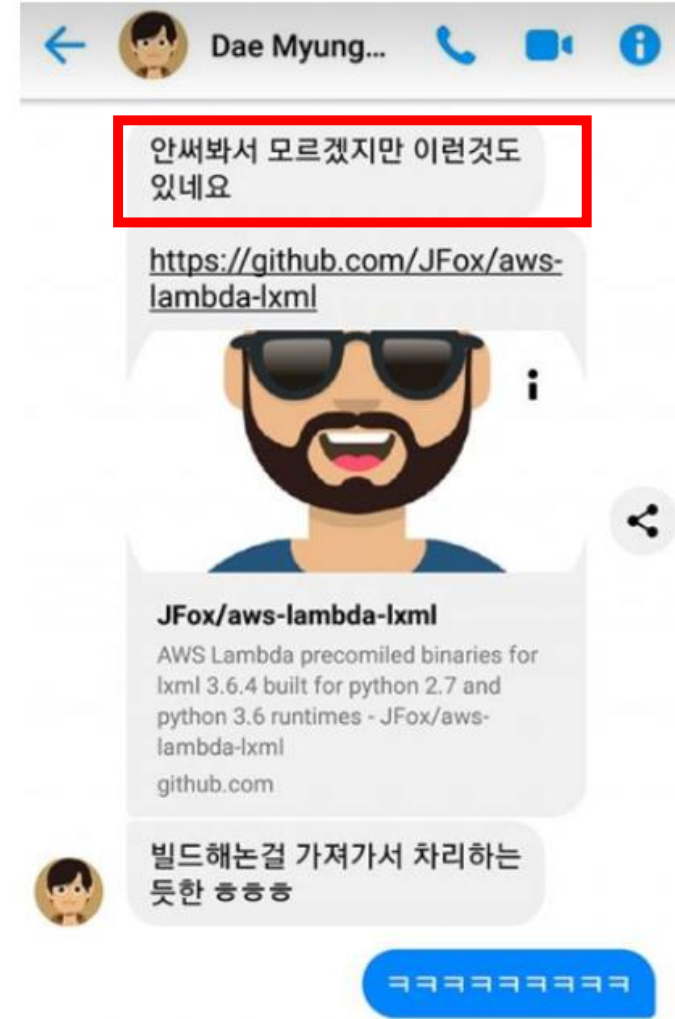
I came across this problem using Python 3.7 in Lambda. Switching to 3.6 solved my problem at least.



# 동시에 Python Korea Group 에서도 답을 같이 찾아 주었고, 기만자도 페이스북 메시지를 통해 알려주었다.



파이썬 코리아 그룹의 도움



기만자



결론적으로 문제점은...

나의 작업 환경은 Windows (Local)  
AWS Lambda 의 환경은 Amazon Linux.

빌드 환경이 전혀 다르기 때문에 동작이 불가능한 것.



!=



결론적으로 문제점은...

나의 작업 환경은 Windows (Local)  
AWS Lambda 의 환경은 Amazon Linux.

빌드 환경이 전혀 다르기 때문에 동작이 불가능한 것.



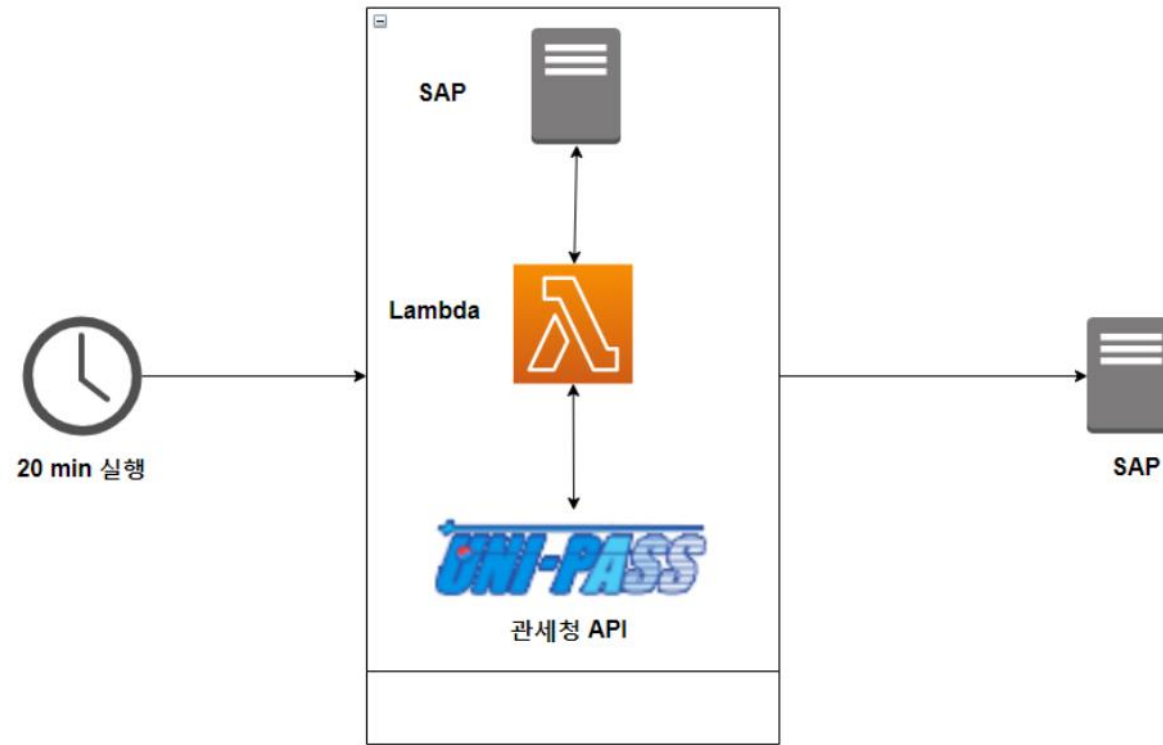
!=



Github에서 Amazon Linux 에서 빌드한 파일을 다운받아서 파일 교체를 통해 처리 완료.

# 다시 작업으로 들어가면...

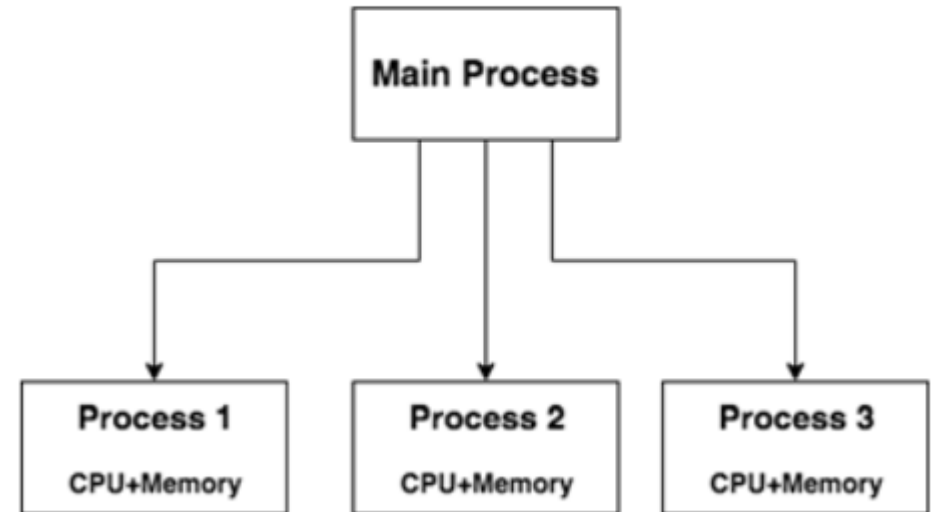
20분마다 실행하여, 데이터를 SAP에서 끌어오고, 관세청 API를 호출  
그리고 그 값을 다시 SAP로 보내주는 것이 해당 작업의 목적이었다.



한번의 많은 데이터를 처리하려는데, 시간이 오래 걸림.  
Python으로 멀티 프로세싱처리 했더니 어찌어찌 로컬에서 돌아감.



## Multiprocessing



**결국 AWS Lambda 의 메모리를 조금씩 올리면서 확인.  
하지만 이렇게 하는것이 진짜 최선일까?**

**점점 올라가는 메모리만큼이나 답답함이 커졌고, 청구될 비용도 커짐...  
하지만 비용보다 구현이 우선이기에 어떻게 풀어갈지 고민을 했음.**

메모리(MB) [정보](#)

구성된 메모리에 비례하는 CPU가 함수에 할당됩니다.

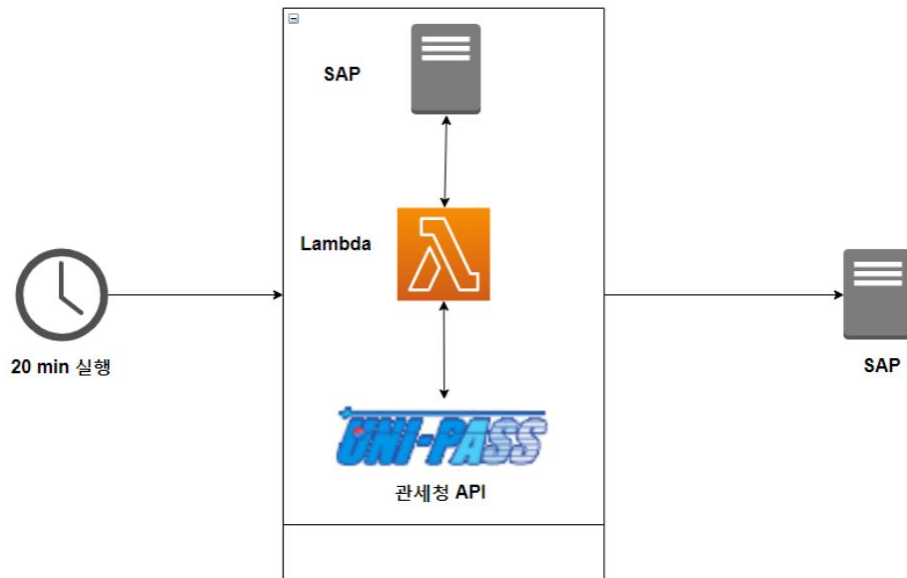


제한 시간 [정보](#)

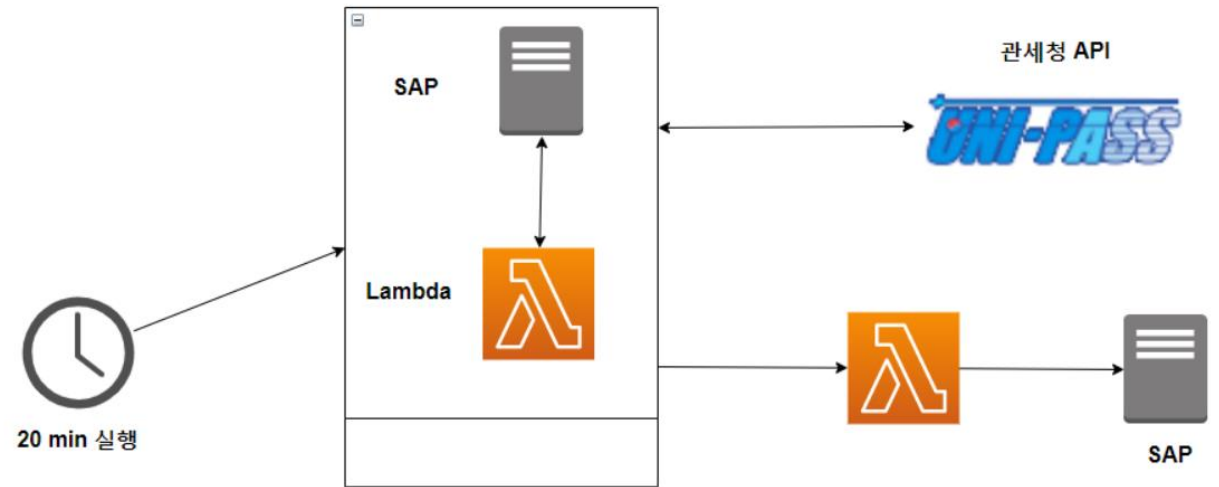
15 분 0 초



구조를 변경하기로 결심! 처리하는 녀석을 나눠보자.  
Lambda를 하나 더 생성함.  
1개는 SAP에서 가져와 관세청 데이터를 가져오고,  
1개는 SAP으로 밀어 넣어주자...

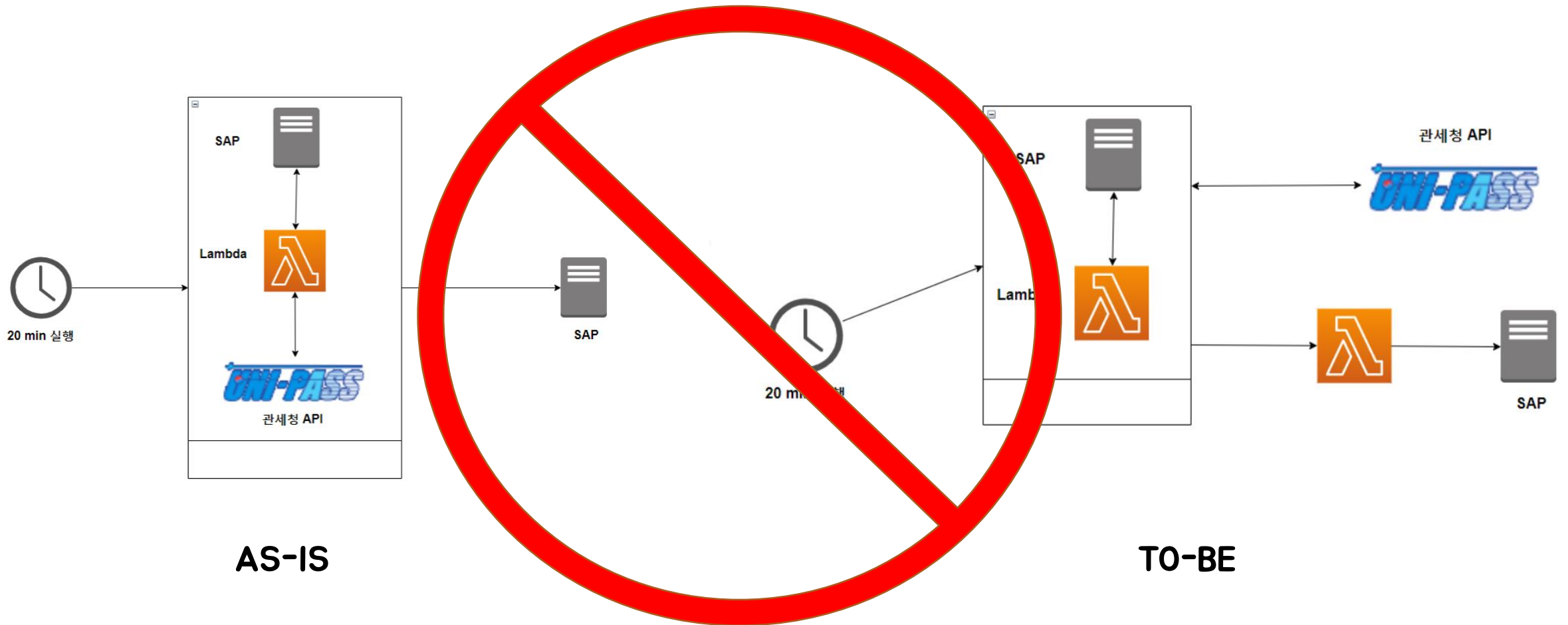


AS-IS



TO-BE

구조를 변경하기로 결심! 처리하는 녀석을 나눠보자.  
Lambda를 하나 더 생성함.  
1개는 SAP에서 가져와 관세청 데이터를 가져오고,  
1개는 SAP으로 밀어 넣어주자... **는 실패**



**다른 방법으로 분산처리를 할 수 있을까 ?**



# 다른 방법으로 분산처리를 할 수 있을까 ?



## SNS

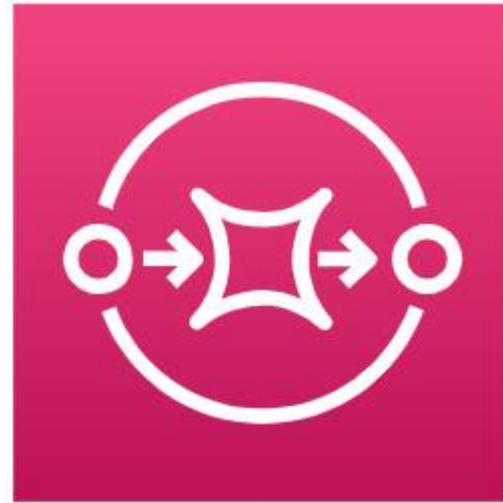
(Amazon Simple Notification Service)

# 다른 방법으로 분산처리를 할 수 있을까 ?



**SNS**

(Amazon Simple Notification Service)



**SQS**

(Amazon Simple Queue Service)

# SNS와 SQS의 차이점은?

## Amazon Simple Notification Service

마이크로 서비스, 분산 시스템 및 서버리스 애플리케이션을 위한  
완전관리형 게시/구독 메시징

## Amazon Simple Queue Service

마이크로 서비스, 분산 시스템 및 서버리스 애플리케이션을 위한  
완전관리형 메시지 대기열

# SNS와 SQS의 차이점은?

## Amazon Simple Notification Service

마이크로 서비스, 분산 시스템 및 서버리스 애플리케이션을 위한

완전관리형 게시/구독 메시징

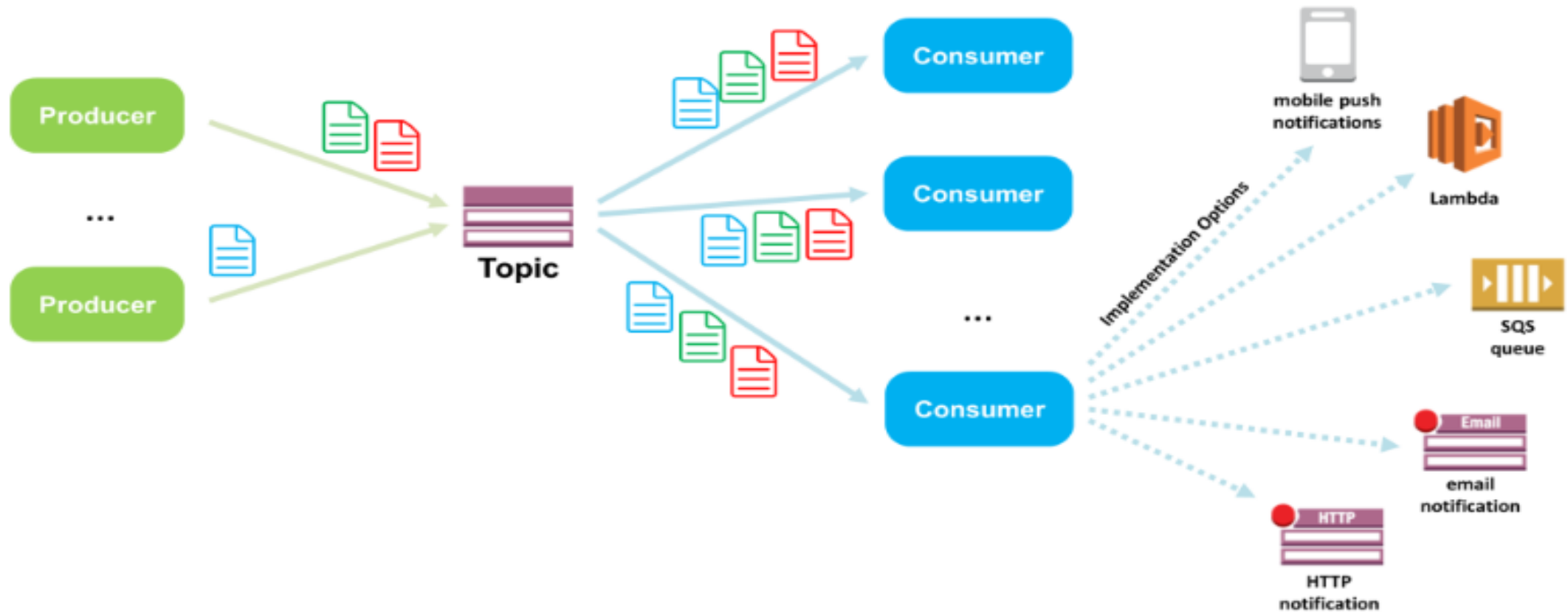
## Amazon Simple Queue Service

마이크로 서비스, 분산 시스템 및 서버리스 애플리케이션을 위한

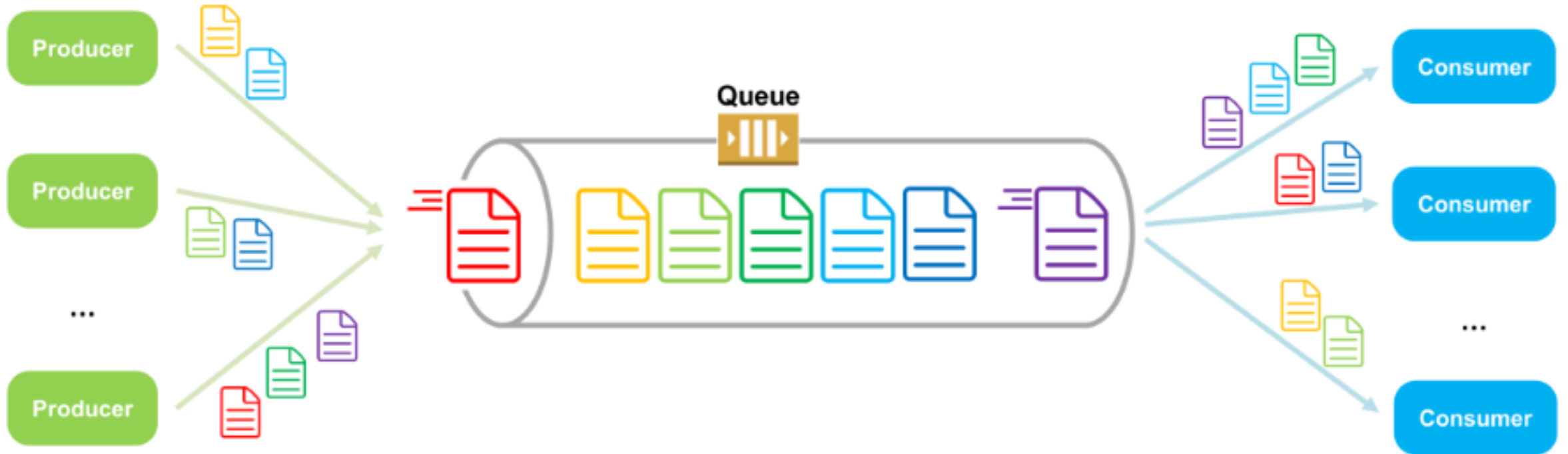
완전관리형 메시지 대기열

게시-구독 기반(Publish-Subscribe)의 SNS  
Queue(Push/Pull)기반의 SQS

# SNS를 활용하는 방법



# SQS를 활용하는 방법



# 나에게 주어진 조건

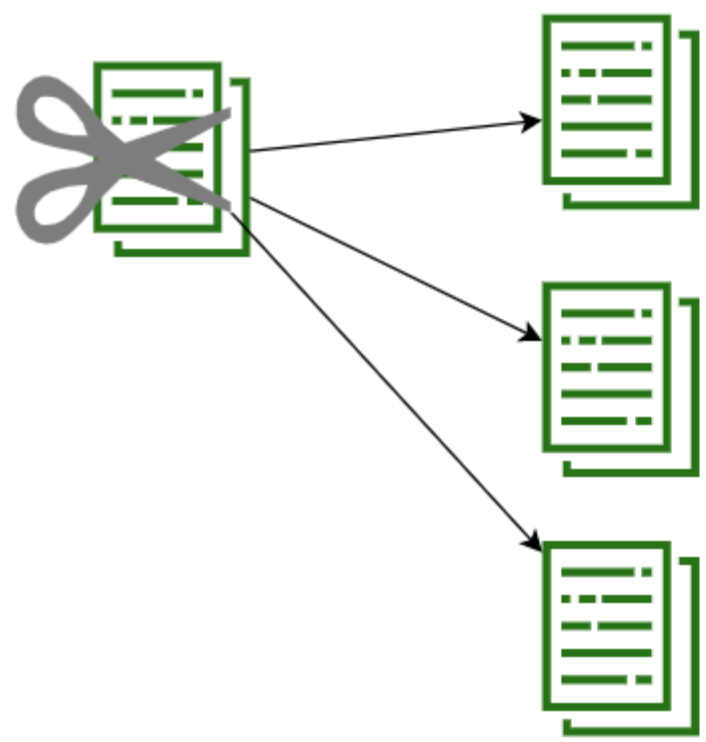
1. 20분 마다 실행. (가능하면 10분.... 더 가능하면 5분....)
2. 폴링 작업의 특성상 반복적으로 같은 오류가 발생하지 않는다면,  
그냥 계속 재실행을 통해 오류에 대해서 대응.
3. 결과값은 받을 필요 없음.
3. 순서는 중요하지 않음.
4. 어떻게... 실행만 문제 없이...

# Amazon SNS 를 여러 번 호출하여 처리하기로 결정

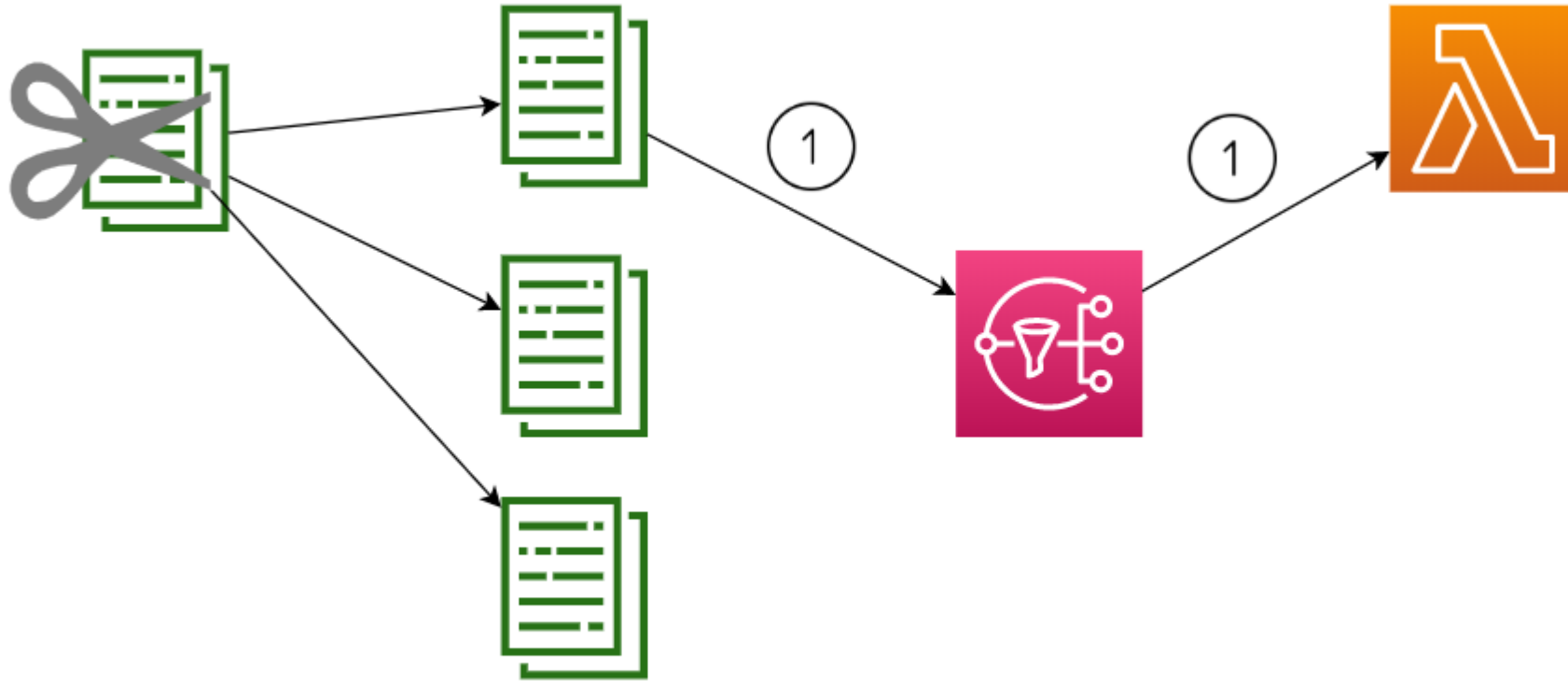




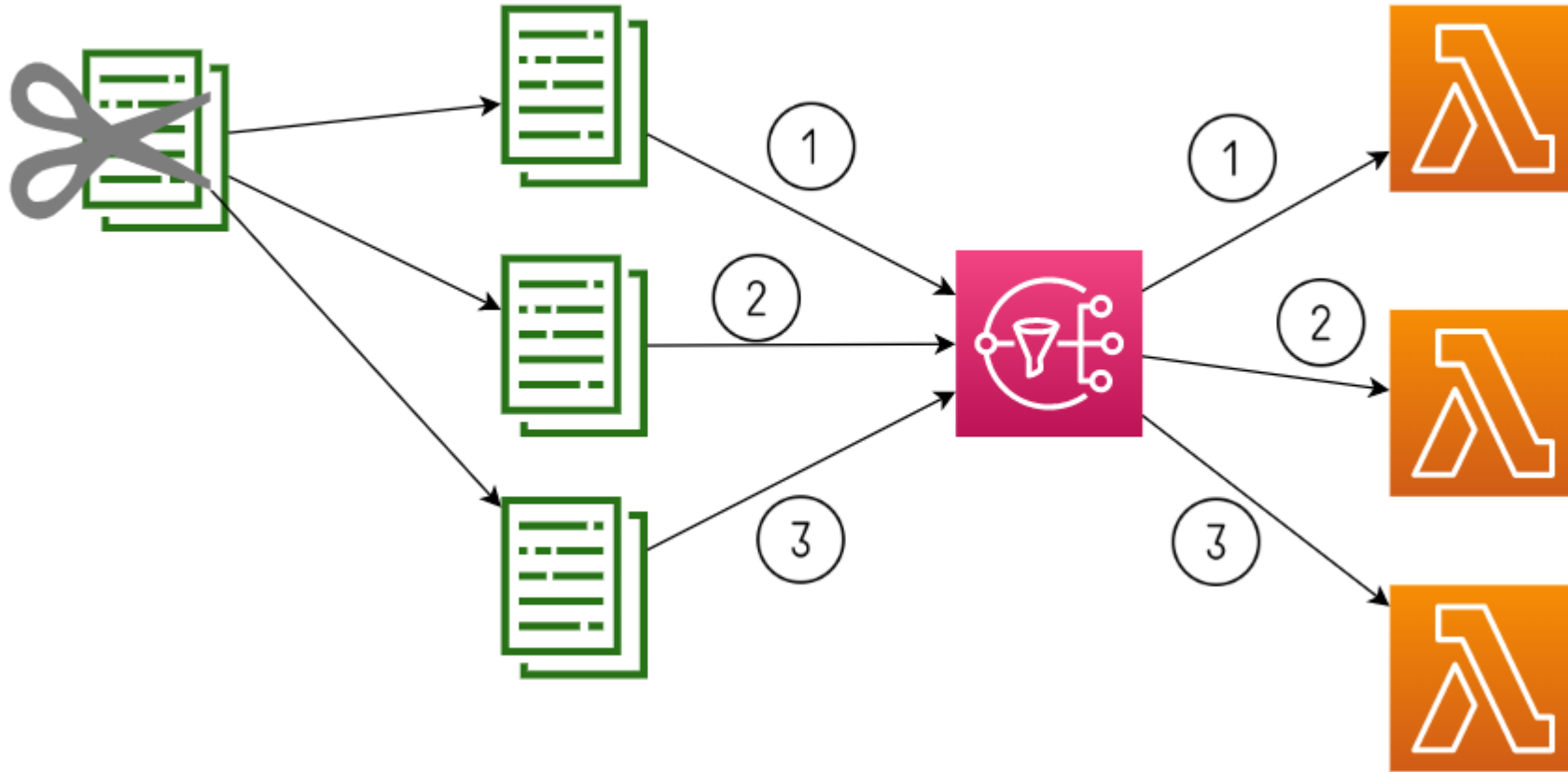
# Amazon SNS 를 여러 번 호출하여 처리하기로 결정



# Amazon SNS 를 여러 번 호출하여 처리하기로 결정

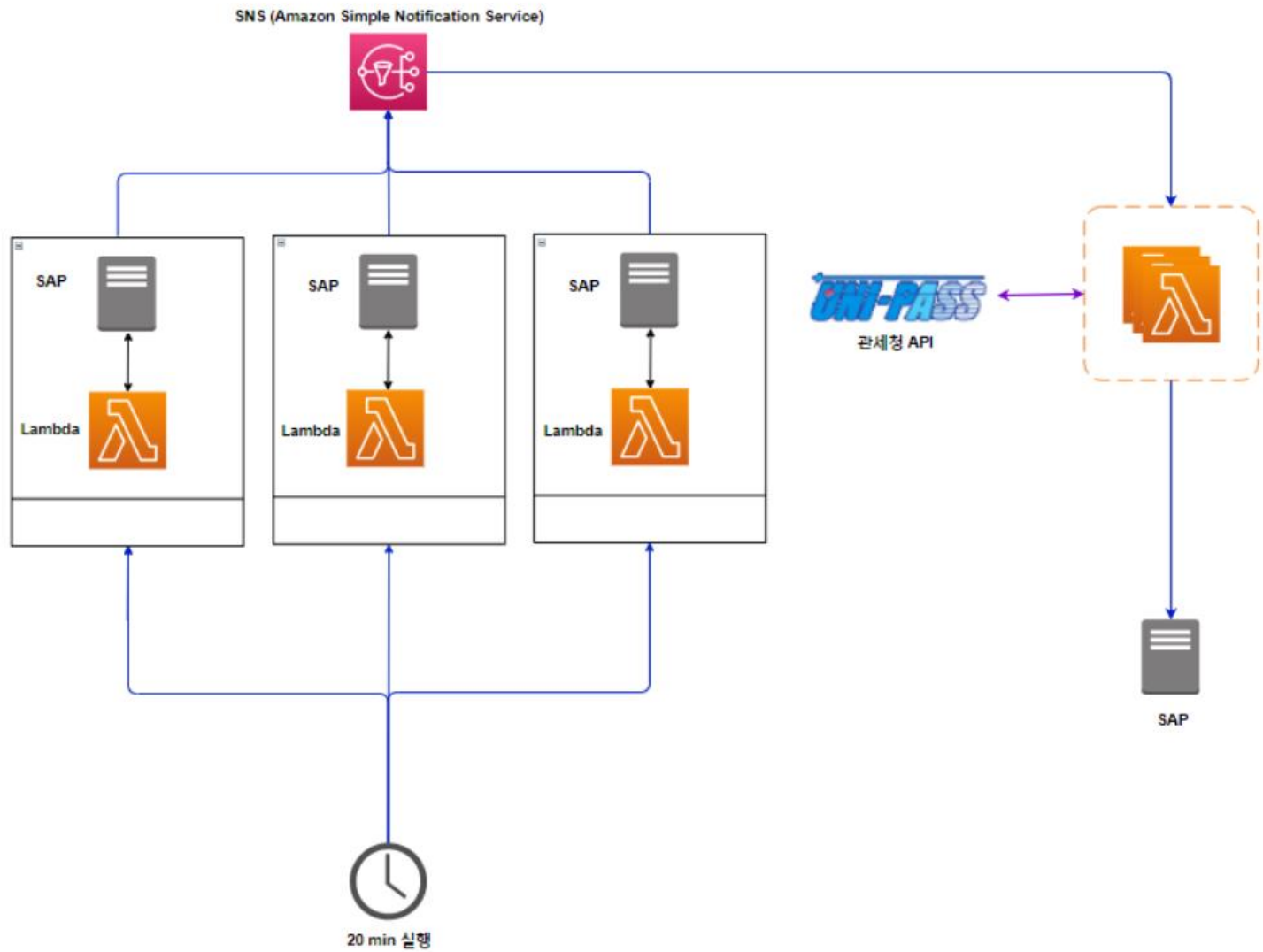


# Amazon SNS 를 여러 번 호출하여 처리하기로 결정



DATA를 나누고, 반복문을 통해 SNS 을 총 3번 호출 -> 각 Lambda 가 1개씩 분할 실행

# 아래 그림과 같은 구조도로 완성.



SNS가 추가된 구조도

**그런데 아무리 짧은 주기로 반복해서 데이터를 갖고 온다고 하지만  
즉각적으로 봐야할 급한 데이터가 있을 수 있다.  
5분 10분을 기다리기만 할 수는 없다.....**

그런데 아무리 짧은 주기로 반복해서 데이터를 갖고 온다고 하지만  
즉각적으로 봐야할 급한 데이터가 있을 수 있다.  
5분 10분을 기다리기만 할 수는 없다.....

**직접 실행 필요 ?**

그런데 아무리 짧은 주기로 반복해서 데이터를 갖고 온다고 하지만  
즉각적으로 봐야할 급한 데이터가 있을 수 있다.  
5분 10분을 기다리기만 할 수는 없다.....

**직접 실행 필요 ?**



그런데 아무리 짧은 주기로 반복해서 데이터를 갖고 온다고 하지만  
즉각적으로 봐야할 급한 데이터가 있을 수 있다.  
5분 10분을 기다리기만 할 수는 없다.....

**직접 실행 필요 ?**



**API Gateway**





# API Gateway

이 메서드의 권한 부여 설정과 이 메서드에서 수신 가능한 파라미터에 대한 정보를 입력합니다.

## 설정

승인 없음 ⓘ

요청 검사기 없음 ⓘ

API 키가 필요함 true

▶ URL 쿼리 문자열 파라미터

▶ HTTP 요청 헤더

▶ 요청 본문

▶ SDK Settings

이 메서드에서 호출할 대상 백엔드의 정보와 함께 수신되는 요청 데이터를 수정할 것인지 여부를 입력합니다.

통합 유형 ☒ Lambda 함수 ⓘ

☐ HTTP ⓘ

☐ Mock ⓘ

☐ AWS 서비스 ⓘ

☐ VPC 링크 ⓘ

헤더값에는 x-api-key 값을 넣고..

KEY	VALUE	DESCRIPTION ***
Content-Type	application/json	
x-api-key	키 값 넣기	
Key	Value	Description

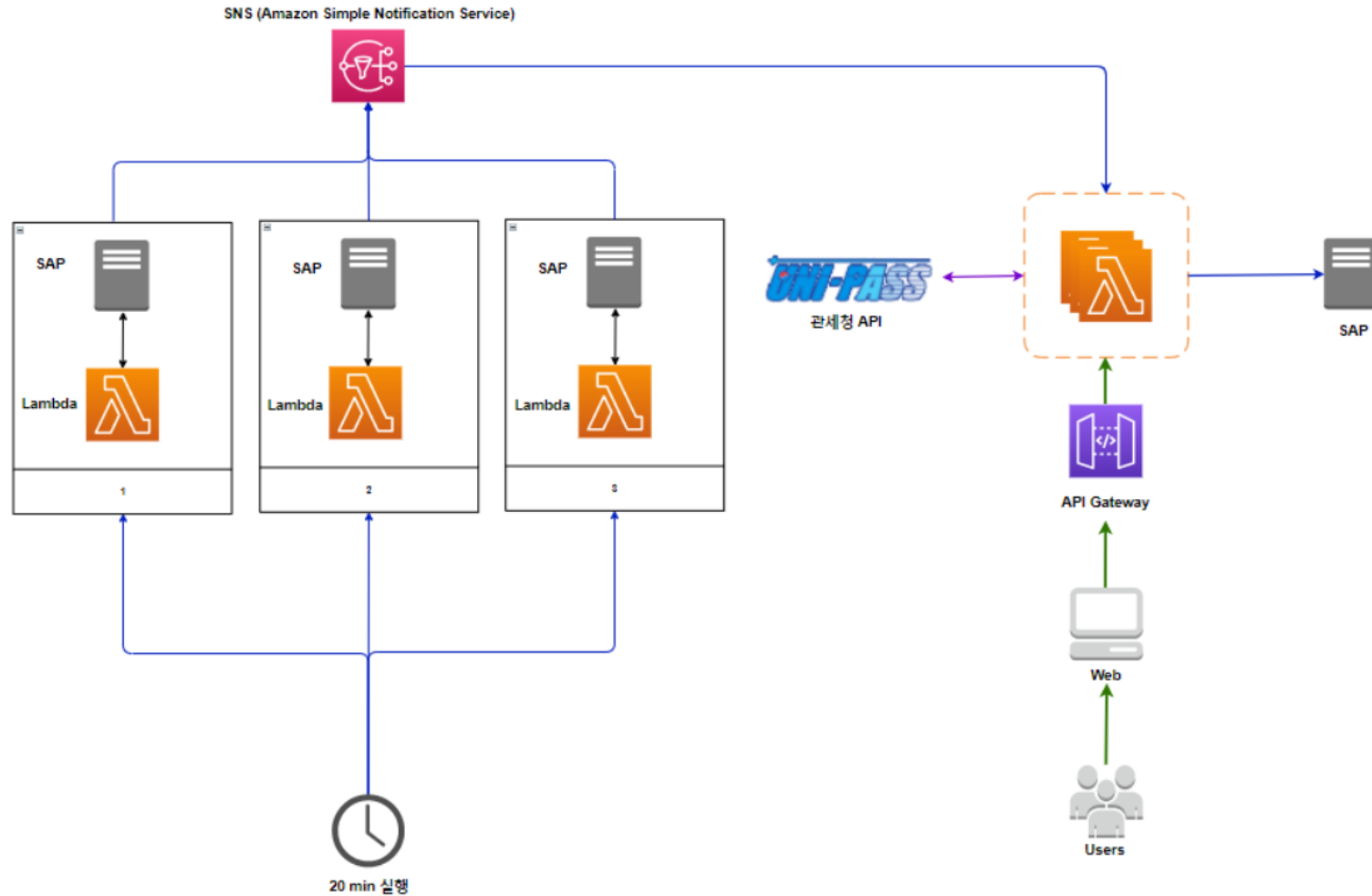
Postman 으로 테스트 진행

```
1 {
2   "API": [
3     {
4       "Type": "API",
5       "Data": [
6         "11111",
7         "22222",
8         "33333"
9       ]
10    }
11  ]
12 }
```

```
1 "job success"
```

리턴 값 확인

# Web에 직접 호출할 수 있는 기능을 넣음 (API Gateway)



API Gateway 까지 추가된 구조도

# 그리고 CloudWatch 를 추가하여 예약 작업을 추가 하였다.

- CloudWatch
- 대시보드
- 경보
  - 경보
  - 데이터 부족
  - 정상
  - 결제
- 이벤트
  - 규칙**
  - 이벤트 버스
- 로그
- 인사이트
- 지표

## 1단계: 규칙 생성

AWS 환경에서 발생하는 이벤트를 기반으로 대상을 호출하는 규칙을 생성합니다.

### 이벤트 소스

이벤트 패턴을 작성 또는 사용자 지정하거나 대상을 호출할 일정을 설정합니다.

☐ 이벤트 패턴 ⓘ

☒ 일정 ⓘ

☒ 고정 비율: 5 분

☐ Cron 식 0/5 \* \* \* ? \*

CloudWatch Events 일정에 대해 '자세히 알아보십시오.'

▶ 샘플 이벤트 표시

### 대상

이벤트가 이벤트 패턴과 일치하거나 일정이 트리거 될 때 호출할 대상을 선택합니다.

Lambda 함수

함수\*

함수 선택

▶ 버전/별칭 구성

▶ 입력 구성

➕ 대상 추가\*

\* 필수

취소

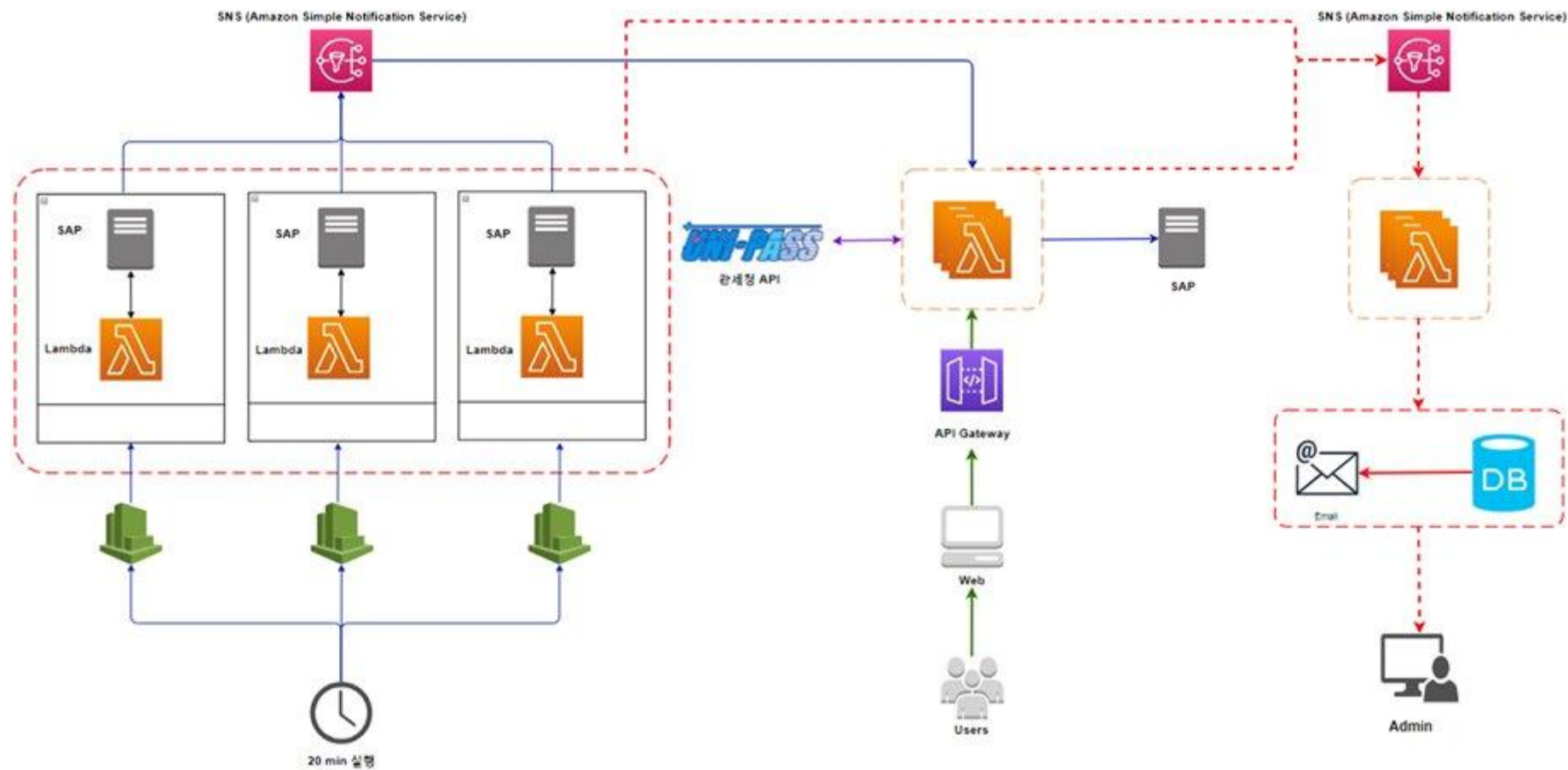
세부 정보 구성

**그런데... 뭐 빠진 것 같지 않나 ?**

그런데... 뭐 빠진 것 같지 않나 ?

에러 처리

# 에러 발생시 SNS 를 호출 Lambda 를 통해 RDBMS로 INSERT



**그러면 왜 RDBMS를 쓰는 것인가 ? AWS 서비스 안 쓰고?**

**그러면 왜 RDBMS를 쓰는 것인가 ? AWS 서비스 안 쓰고?**

**운영담당자**



그러면 왜 RDBMS를 쓰는 것인가 ? AWS 서비스 안 쓰고?

# 운영담당자

기존의 업무 방식이 존재하는데...  
마음대로 다른 방법을 요구할 수 없다고 판단....

**현재 그림 잘 사용하고 있나요?**

현재 그림 잘 사용하고 있나요?

네ㅋㅋㅋ

# 에필로그

# 에필로그



류한진님이 링크를 공유했습니다.



👤 인기 게시물 작성 멤버 · 5월 17일 · 💡 Serverless

5월21일부터 lambda의 base가 Amazon Linux(2017.03)에서 Amazon Linux(2018.03)으로 업데이트됩니다.

AWS lambda를 사용하시는 분들은 아래 article을 읽어보시고 테스트를 해보세요

AWS Lambda 및 Lambda@Edge 실행 환경에 대한 향후 업데이트 계획 안내 :

<https://aws.amazon.com/.../upcoming-updates-to-the-aws-lambda.../>

참고 - Lambda Runtimes:

<https://docs.aws.amazon.com/.../latest/dg/lambda-runtimes.html>



# 에필로그



류한진님이 링크를 공유했습니다.

👤 인기 게시물 작성 멤버 · 5월 17일 · 💡 Serverless



5월21일부터 lambda의 base가 Amazon Linux(2017.03)에서 Amazon Linux(2018.03)으로 업데이트됩니다.

AWS lambda를 사용하시는 분들은 아래 article을 읽어보시고 테스트를 해보세요

AWS Lambda 및 Lambda@Edge 실행 환경에 대한 향후 업데이트 계획 안내 :

<https://aws.amazon.com/.../upcoming-updates-to-the-aws-lambda.../>

참고 - Lambda Runtimes:

<https://docs.aws.amazon.com/.../latest/dg/lambda-runtimes.html>



aws



AWS.AMAZON.COM

**AWS Lambda 및 Lambda@Edge 실행 환경에 대한 향후 업데이트 계획 안내 | Amazon Web Services**

아.....이런...

# 에필로그



류한진님이 링크를 공유했습니다.

👤 인기 게시물 작성 멤버 · 5월 17일 · 💎 Serverless



5월21일부터 lambda의 base가 Amazon Linux(2017.03)에서 Amazon Linux(2018.03)으로 업데이트됩니다.

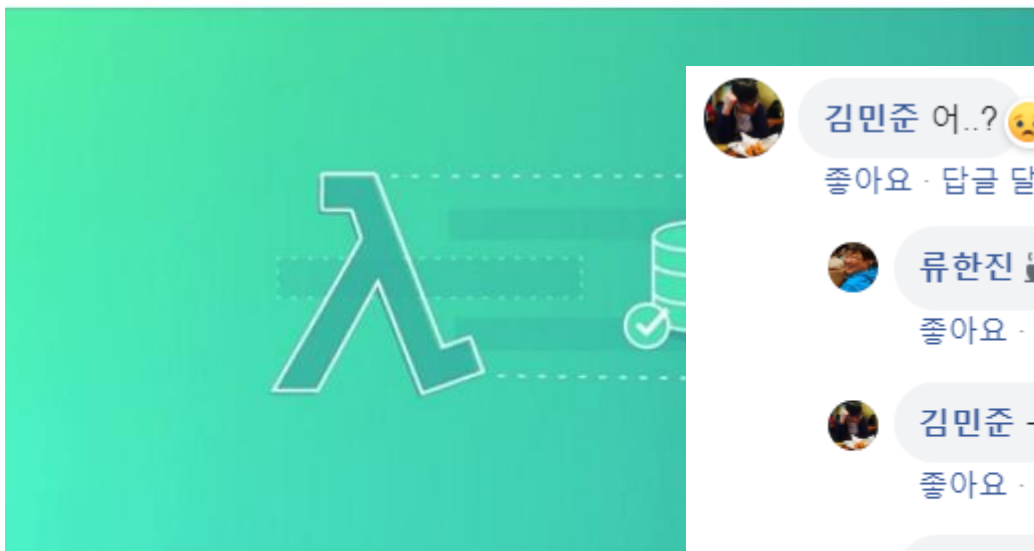
AWS lambda를 사용하시는 분들은 아래 article을 읽어보시고 테스트를 해보세요

AWS Lambda 및 Lambda@Edge 실행 환경에 대한 향후 업데이트 계획 안내 :

<https://aws.amazon.com/.../upcoming-updates-to-the-aws-lambda.../>

참고 - Lambda Runtimes:

<https://docs.aws.amazon.com/.../latest/dg/lambda-runtimes.html>



AWS.AMAZON.COM

**AWS Lambda 및 Lambda@Edge 실행 환경  
트 계획 안내 | Amazon Web Services**

## 아....이런...



김민준 어..? 😞 2

좋아요 · 답글 달기 · 7주



류한진 🍵 김민준 네 하셔야해요 ㅋㅋㅠ 메일 드릴게요

좋아요 · 답글 달기 · 7주



김민준 ———— 😞 1

좋아요 · 답글 달기 · 7주



류한진 🍵 김민준 안하셔도 되는데.. 안되고 고치는거보단 한번만 테스트를 ㅠㅠ

좋아요 · 답글 달기 · 7주

# 에필로그

결론적으로 문제점은...

나의 작업 환경은 Windows (Local)  
AWS Lambda 의 환경은 Amazon Linux.

라이브러리에 헤더파일이 존재하기도 하고...  
빌드 환경이 전혀 다르기 때문에 동작이 불가능한 것.

결과적으로 Github에서 Amazon Linux 상에서 빌드한 파일을 다운받아서  
파일 교체를 통해 처리 완료.

앞에서 한번 겪었던 문제점...



## 다시 작업을 진행

1. 업데이트 진행된 EC2를 접속
2. 사용했던 라이브러리 전부 설치
3. FTP를 통해 다운로드
4. Layers 신규 생성하여, 라이브러리 파일(zip) 업로드
5. 테스트
6. 정상 동작 확인 완료
7. 모든 Lambda 에 Layers를 신규 버전으로 교체

**현재 그림 잘 사용하고 있나요?**

현재 그림 잘 사용하고 있나요?

네ㅋㅋㅋ


# 정리

1. 동일한 라이브러리는 Layers 를 이용하면 여러 Lambda 재사용 가능.
2. 하지만 Amazon Linux 가 버전업이 예정되어 있다면, 안전하게 테스트 진행
3. 분산처리의 방법은 자신에게 주어진 환경에 맞는 방법을 선택  
(SNS 를 단독 사용하여도 좋고, SQS 를 섞어 사용해도 좋고...)
4. CloudWatch 를 이용하면 Lambda 를 예약작업으로 사용 가능
5. API Gateway 를 이용하면, 외부에서도 Lambda 를 실행 할 수 있음.
6. 다른 Cloud 서비스도 비슷한 것이 아주 많음. (예 Azure Function)

# 정리

1. 동일한 라이브러리는 Layers 를 이용하면 여러 Lambda 재사용 가능.
2. 하지만 Amazon Linux 가 버전업이 예정되어 있다면, 안전하게 테스트 진행
3. 분산처리의 방법은 자신에게 주어진 환경에 맞는 방법을 선택  
(SNS 를 단독 사용하여도 좋고, SQS 를 섞어 사용해도 좋고...)
4. CloudWatch 를 이용하면 Lambda 를 예약작업으로 사용 가능
5. API Gateway 를 이용하면, 외부에서도 Lambda 를 실행 할 수 있음.
6. 다른 Cloud 서비스도 비슷한 것이 아주 많음. (예 Azure Function)

사용한 서비스를 모두 자세히 설명할 수 없었습니다.  
AWS Documentation을 참고하세요!! (궁서체)



Search in AWS documentation

FeedbackEnglishSign In to the Console

[AWS](#) > AWS Documentation

# AWS Documentation

Find user guides, developer guides, API references, tutorials, and more.

## Guides and API References

### Compute

- [Amazon EC2](#)
- [AWS Batch](#)
- [Amazon ECR](#)
- [Amazon ECS](#)
- [Amazon EKS](#)
- [AWS Elastic Beanstalk](#)
- [AWS Lambda](#)
- [Amazon Lightsail](#)

### Storage

- [Amazon S3](#)
- [AWS Backup](#)
- [Amazon EBS](#)
- [Amazon EFS](#)
- [Amazon FSx](#)
- [Amazon S3 Glacier](#)
- [AWS Snowball](#)
- [AWS Storage Gateway](#)

### Database

- [Amazon Aurora](#)
- [Amazon DocumentDB](#)
- [Amazon DynamoDB](#)
- [Amazon ElastiCache](#)
- [Amazon Neptune](#)
- [Amazon RDS](#)
- [Amazon Redshift](#)

### Developer Tools

- [AWS Cloud9](#)
- [AWS CodeBuild](#)
- [AWS CodeCommit](#)
- [AWS CodeDeploy](#)
- [AWS CodePipeline](#)
- [AWS CodeStar](#)
- [AWS Tools & SDKs](#)
- [AWS X-Ray](#)

## Resources

### On this page

- [Guides and API References](#)
- [Tutorials and Projects](#)
- [SDKs and Toolkits](#)
- [General Resources](#)

Terms of UsePrivacy

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

**AWS Documentation Site**

**감사합니다.**