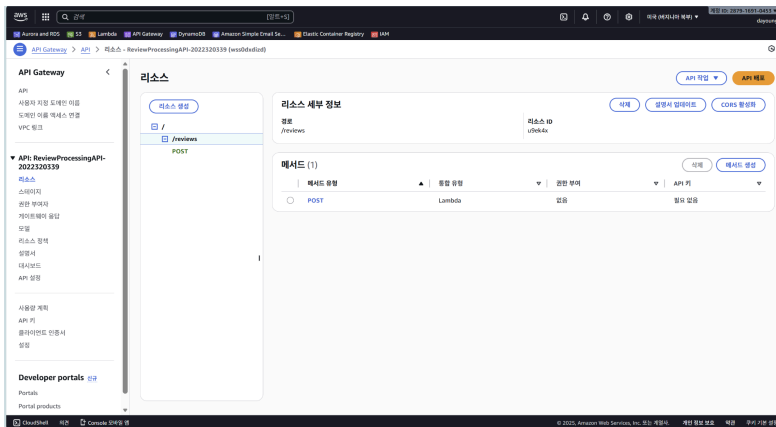


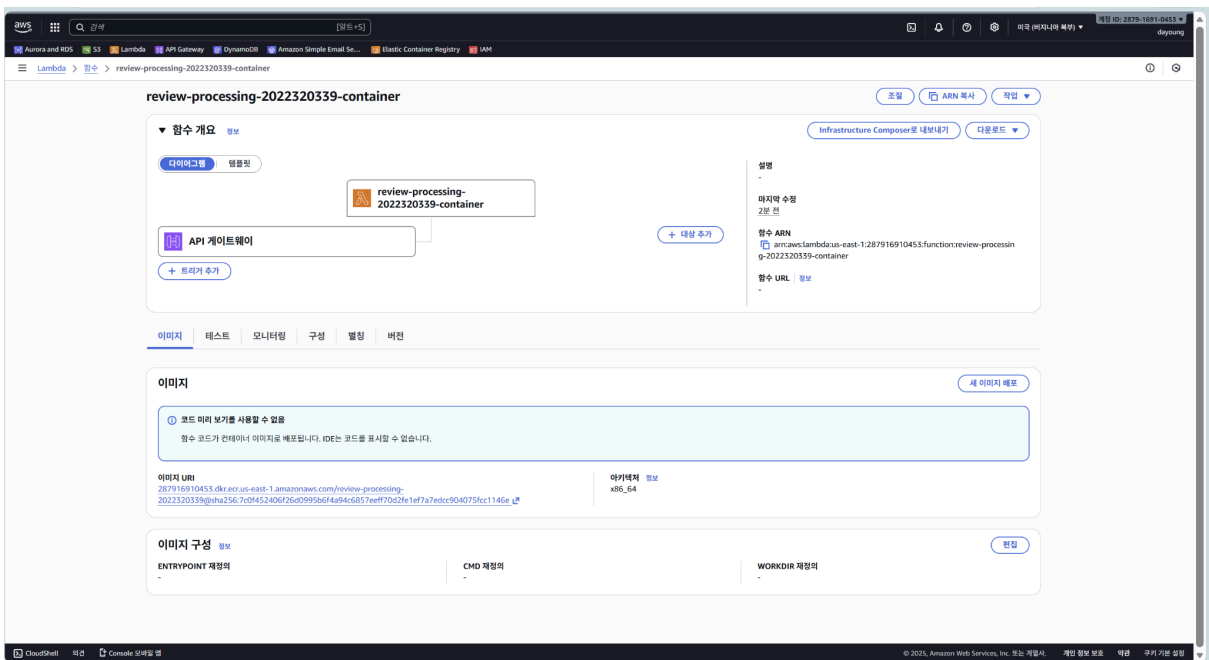
2차 실습 과제

2022320339 최다영

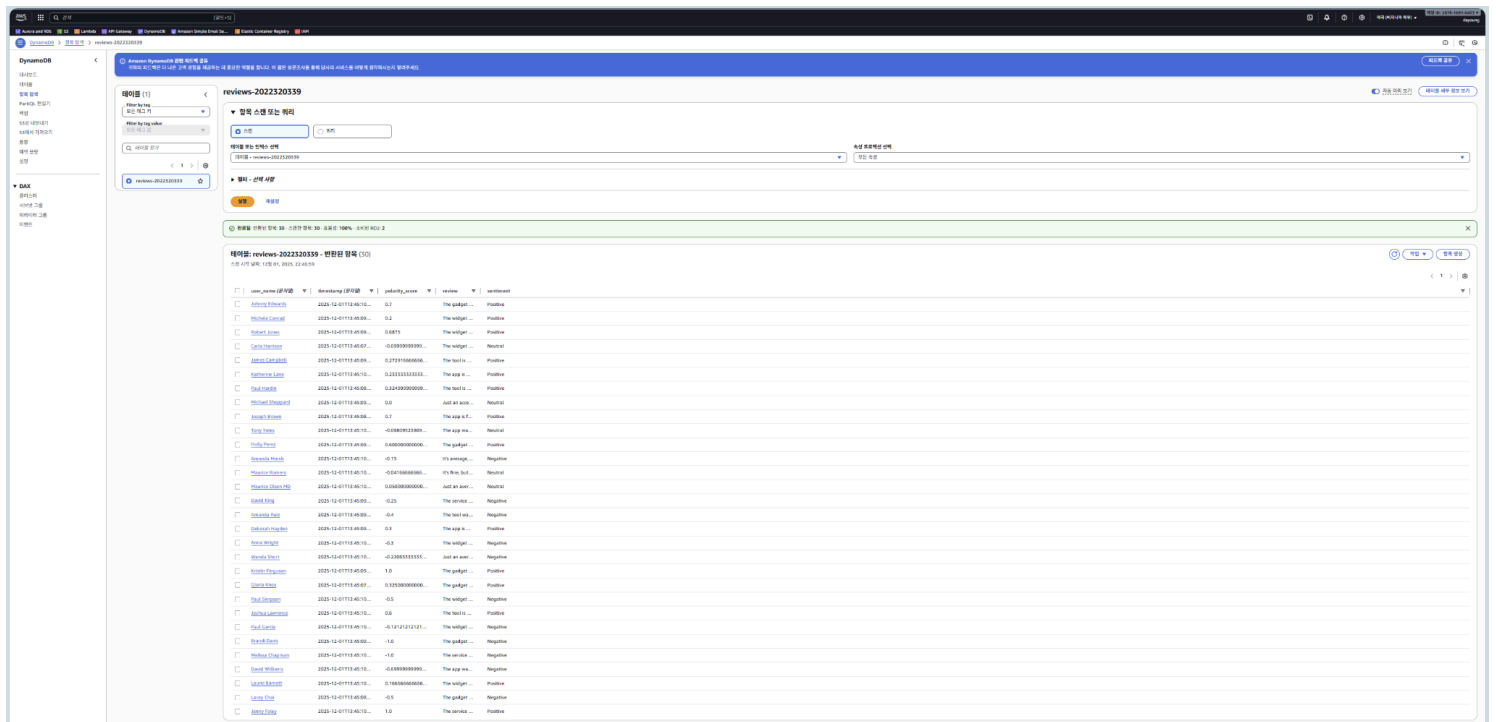
1. API gateway 리소스 스크린샷



2. 완성된 Lambda 함수 스크린샷



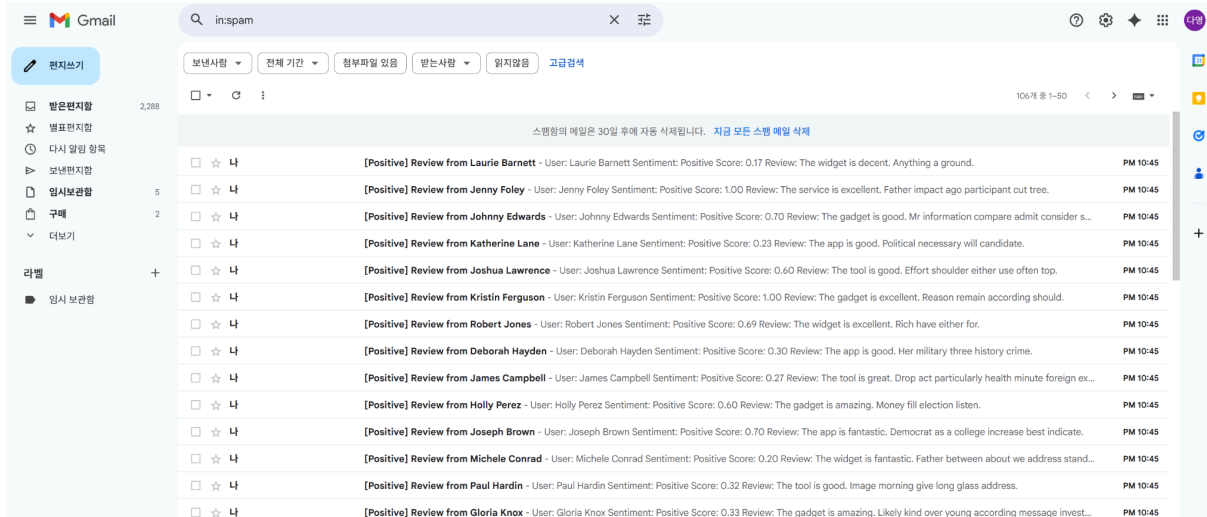
3. DynamoDB 테이블에 저장된 리뷰 데이터 스크린샷



review_id	timestamp	score	text	sentiment
review-202320339-001	2023-12-01T18:46:10...	0.7	The gadget ...	Positive
review-202320339-002	2023-12-01T18:46:10...	0.2	The widget ...	Positive
review-202320339-003	2023-12-01T18:46:10...	0.8879	The widget ...	Positive
review-202320339-004	2023-12-01T18:46:10...	-0.2889999999...	The widget ...	Neutral
review-202320339-005	2023-12-01T18:46:10...	0.278716666666...	The tool is ...	Positive
review-202320339-006	2023-12-01T18:46:10...	0.2233333333...	The app is ...	Positive
review-202320339-007	2023-12-01T18:46:10...	0.2245555555...	The tool is ...	Positive
review-202320339-008	2023-12-01T18:46:10...	0.8	Just an aver ...	Neutral
review-202320339-009	2023-12-01T18:46:10...	0.7	The app is f ...	Positive
review-202320339-010	2023-12-01T18:46:10...	-0.6889999999...	The app is ...	Neutral
review-202320339-011	2023-12-01T18:46:10...	0.6889999999...	The gadget ...	Positive
review-202320339-012	2023-12-01T18:46:10...	-0.75	The gadget ...	Negative
review-202320339-013	2023-12-01T18:46:10...	-0.8716666666...	It's fine, bu ...	Neutral
review-202320339-014	2023-12-01T18:46:10...	0.6889999999...	Just an aver ...	Neutral
review-202320339-015	2023-12-01T18:46:10...	-0.23	The service ...	Negative
review-202320339-016	2023-12-01T18:46:10...	-0.4	The tool was ...	Negative
review-202320339-017	2023-12-01T18:46:10...	0.5	The app is ...	Positive
review-202320339-018	2023-12-01T18:46:10...	-0.3	The widget ...	Negative
review-202320339-019	2023-12-01T18:46:10...	-0.2889999999...	Just an aver ...	Negative
review-202320339-020	2023-12-01T18:46:10...	1.0	The gadget ...	Positive
review-202320339-021	2023-12-01T18:46:10...	0.5255555555...	The gadget ...	Positive
review-202320339-022	2023-12-01T18:46:10...	-0.5	The widget ...	Negative
review-202320339-023	2023-12-01T18:46:10...	0.8	The tool is ...	Positive
review-202320339-024	2023-12-01T18:46:10...	-0.1213713713...	The widget ...	Negative
review-202320339-025	2023-12-01T18:46:10...	-1.0	The gadget ...	Negative
review-202320339-026	2023-12-01T18:46:10...	-1.0	The service ...	Negative
review-202320339-027	2023-12-01T18:46:10...	-0.8889999999...	The app is ...	Negative
review-202320339-028	2023-12-01T18:46:10...	0.1666666666...	The gadget ...	Positive
review-202320339-029	2023-12-01T18:46:10...	-0.5	The gadget ...	Negative
review-202320339-030	2023-12-01T18:46:10...	1.0	The service ...	Positive

```
(venv) PS C:\cloud_computing_hw2\review-processing> python request_generator.py
Status: 200 | The widget was awful. Chair great carry add quality force economy.
Status: 200 | The gadget is amazing. Likely kind over young according message investment.
Status: 200 | The gadget was useless. Serve discussion move resource so.
Status: 200 | The app is fantastic. Democrat as a college increase best indicate.
Status: 200 | The tool is good. Image morning give long glass address.
Status: 200 | Just an acceptable experience. Morning along opportunity.
Status: 200 | The tool was poor. Draw everything environment tell reveal.
Status: 200 | The widget is fantastic. Father between about we address standard owner.
Status: 200 | The tool is great. Drop act particularly health minute foreign exactly.
Status: 200 | The service was useless. Experience information different apply service.
Status: 200 | The app is good. Her military three history crime.
Status: 200 | The gadget is amazing. Money fill election listen.
Status: 200 | The gadget was terrible. Bring of police news product.
Status: 200 | Just an average experience. Ready make be admit show cold local great.
Status: 200 | The app was bad. Them attention design role community.
Status: 200 | The widget was useless. Two interview girl.
Status: 200 | The widget is excellent. Rich have either for.
Status: 200 | The gadget is excellent. Reason remain according should.
Status: 200 | The app is good. Political necessary will candidate.
Status: 200 | The tool is good. Effort shoulder either use often top.
Status: 200 | The widget is decent. Anything a ground.
Status: 200 | The service was awful. Car office company wait follow.
Status: 200 | It's average, but not perfect. Step fast move design.
Status: 200 | The widget was terrible. Result check fill break collection enjoy.
Status: 200 | The app was awful. Race strong certainly customer include American.
Status: 200 | The widget was useless. Want activity new above.
Status: 200 | Just an average experience. Character discuss energy hard learn up management though.
Status: 200 | The gadget is good. Mr information compare admit consider space deal.
Status: 200 | It's fine, but not perfect. Or part share two.
Status: 200 | The service is excellent. Father impact ago participant cut tree.
Execution time: 0:00:05.230284
(venv) PS C:\cloud_computing_hw2\review-processing>
```

4. 이메일 전송 증빙 스크린샷



5. 파이프라인 및 코드 설명

- 전체 파이프라인 : AWS의 **serverless service**를 활용하여 사용자로부터 전달된 리뷰 데이터를 분석-저장한다. 특히 **positive review**에 대해서는 이메일 알림을 전송한다.
- 처리 순서 : 1. Client (request generator)

로컬에서 실행되는 **python code request_generator.py**를 사용하고, 여러 사용자 리뷰 데이터를 **HTTP POST** 요청 형태로 전송한다. **API Gateway**의 **/reviews** 엔드포인트를 호출하여 테스트 트래픽을 생성한다.

2. API Gateway

Rest API 형태로 **/reviews** 엔드포인트를 제공한다. 외부 요청을 수신하여 **Lambda** 함수와 연결을 한다. 외부 트래픽을 **lambda** 함수로 안전하게 전달하는 **entry point**이다.

3. container기반의 AWS Lambda

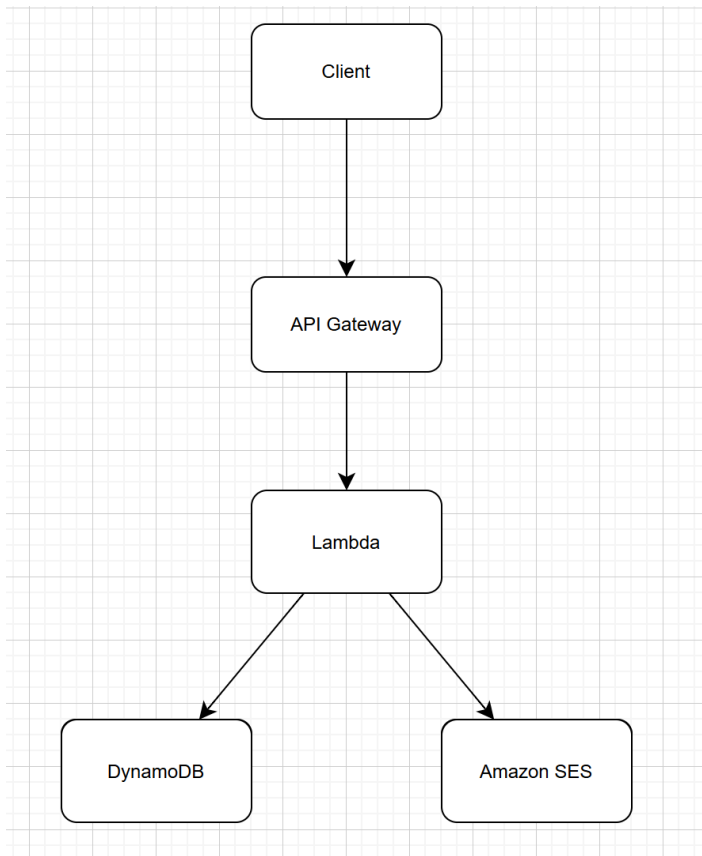
Docker container 이미지 형태로 배포된 **lambda** 함수이다. **API gateway**로부터 받은 이벤트를 처리하여 리뷰 텍스트 기반의 **sentiment**를 분석하고 판단하여 결과 데이터를 구조화한다. **Amazon ECR**에 컨테이너 이미지를 저장하고 **lambda**에서 직접 참조한다.

4. DynamoDB

sentiment 분석 결과를 저장하는 **NoSql** 데이터베이스이다. 사용자의 이름과 리뷰 내용, **sentiment** 결과, 점수, 타임스탬프를 기록한다.

5. Amazon SES

Lambda 함수에서 **positive** 리뷰일 경우에만 호출하며 검증된 이메일 주소로 알림을 전송한다. **negative** 리뷰에 대해서는 발송하지 않는다.



- `lambda_function.py` 의 코드 블록의 역할 분석
Aws 리소스 초기화 및 환경 변수 설정 : `lambda` 함수가 실행될 때 사용할 **DynamoDB** 테이블, **SES** 클라이언트, 환경 변수 기반으로 이메일 주소를 초기화한다. 이를 통해서 함수 내부에서는 별도의 설정 없이 바로 **AWS** 리소스를 사용할 수 있다.

```
import boto3 # AWS SDK for Python to interact with AWS services
from textblob import TextBlob # Library for processing textual data
import datetime
import json
import os

# DynamoDB
dynamodb = boto3.resource("dynamodb")

# Environment variable should be set in Lambda console
TABLE_NAME = os.environ.get("TABLE_NAME", "")
table = dynamodb.Table(TABLE_NAME)

# SES
ses = boto3.client("ses", region_name="us-east-1")

# Sender / Receiver email
SENDER_EMAIL = os.environ.get("SENDER_EMAIL", "")
```

```
RECEIVER_EMAIL = os.environ.get("RECIPIENT_EMAIL", "")
```

요청 바디 파싱

API Gateway -> Lambda 입력을 처리한다. API gateway에서 전달된 event 객체에서 HTTP body를 안전하게 파싱하여 리뷰 작성자 이름과 리뷰 내용을 추출하고 타임스탬프를 생성한다.

```
def lambda_handler(event, context):
    try:
        parse it safely.
        if isinstance(event.get("body"), str):
            body = json.loads(event["body"])
        else:
            body = event

        user_name = body.get("user_name", "Anonymous")
        review_text = body.get("review", "")
        timestamp = datetime.datetime.now().isoformat()
```

TextBlob을 이용한 감성 분석 수행

리뷰 텍스트를 textblob에 전달하여 polarity 점수를 계산하고 그 값을 기준으로 "positive", "negative", "neutral" 세가지로 분류한다.

```
blob = TextBlob(review_text)
polarity = blob.sentiment.polarity

# Determine sentiment category based on polarity thresholds
if polarity > 0.1:
    sentiment = "Positive"
elif polarity < -0.1:
    sentiment = "Negative"
else:
    sentiment = "Neutral"
```

DynamoDB에 분석 결과 저장

사용자 이름, 리뷰 텍스트, 감성 결과, 폴라리티 점수, 타임스탬프를 하나의 item으로 dynamoDB 테이블에 저장한다.

```
table.put_item(
    Item={
        "user_name": user_name,
        "review": review_text,
        "sentiment": sentiment,
        # Convert float to string for DynamoDB compatibility
```

```

        "polarity_score": str(polarity),
        "timestamp": timestamp,
    }
)

```

Positive 리뷰에 대한 이메일 알림 전송 (Amazon SES)

감성 분석 결과가 “positive”인 경우에만 SES를 통해서 이메일 알림을 전송한다.

```

if sentiment == "Positive" and SENDER_EMAIL and RECEIVER_EMAIL:
    ses.send_email(
        Source=SENDER_EMAIL,
        Destination={"ToAddresses": [RECEIVER_EMAIL]},
        Message={
            "Subject": {"Data": f"[{sentiment}] Review from {user_name}"},
            "Body": {
                "Text": {
                    "Data": (
                        f"User: {user_name}\n"
                        f"Sentiment: {sentiment}\n"
                        f"Score: {polarity:.2f}\n\n"
                        f"Review:\n{review_text}"
                    )
                }
            },
        },
    )

```

HTTP 응답 반환 및 에러 처리

정상 처리된 경우에는 **HTTP 200** 응답과 함께 메세지, 사용자 이름, 감성 결과, 폴라리티 스코어를 json으로 반환한다. 에러 발생 시 **500** 코드를 반환하고 **CloudWatch** 로그에 에러 내용을 출력한다.

```

# 5. Return HTTP response
return {
    "statusCode": 200,
    "headers": {"Content-Type": "application/json"},
    "body": json.dumps(
        {
            "message": "Review processed",
            "user_name": user_name,
            "sentiment": sentiment,
            "polarity_score": round(polarity, 2),
        }
    )
}

```

```

    ),
}

except Exception as e:
    # Log the error for debugging (CloudWatch Logs)
    print(f"Error: {e}")
    return {
        "statusCode": 500,
        "body": json.dumps("Internal Server Error"),
    }

```

- Dockerfile 코드 설명

FROM public.ecr.aws/lambda/python:3.13 : AWS에서 제공하는 Python 3.13 Lambda 런타임 이미지를 베이스 이미지로 사용한다. 이는 Lambda 실행 환경과 동일한 운영체제 및 런타임을 제공한다. 이 코드의 역할은 Lambda에서 실행 가능한 컨테이너 환경을 구성하는 기반 이미지를 설정하는 것이다.

COPY requirements.txt . : 외부 라이브러리를 준비한다. 애플리케이션에서 사용하는 python 라이브러리 목록을 담은 requirements.txt 파일을 컨테이너 내부 작업 디렉토리로 복사하여 이후 pip install을 수행하기 위한 사전 단계이다. 역할로는 애플리케이션 의존성 정보를 컨테이너로 전달한다.

RUN python -m pip install -r requirements.txt : requirements.txt 에 명시된 라이브러리들을 컨테이너 내부에 설치한다. lambda 실행 시 필요한 외부 라이브러리를 미리 포함하여 컨테이너 배포 시에 의존성 누락으로 인한 실행 오류를 방지한다.

COPY lambda_function.py . : Lambda 함수 코드를 복사한다. Lambda 실행 코드 자체를 컨테이너 이미지에 포함하고, Docker 이미지 안에서 Lambda가 참조할 수 있도록 배치한다.

CMD ["lambda_function.lambda_handler"] : Lambda 함수의 진입점을 설정한다. Lambda 컨테이너 실행 시 호출될 핵심 함수를 정의하고 AWS Lambda 런타임이 컨테이너 실행 시 이 핸들러를 자동적으로 호출하도록 한다.