

# 🌟 Dash-Board Image API 기능 만들기

**목적:** AWS CloudWatch api를 이용해서 Lambda function의 사용 빈도를 나타내는 이미지를 만듭니다.

작성자: 송은주

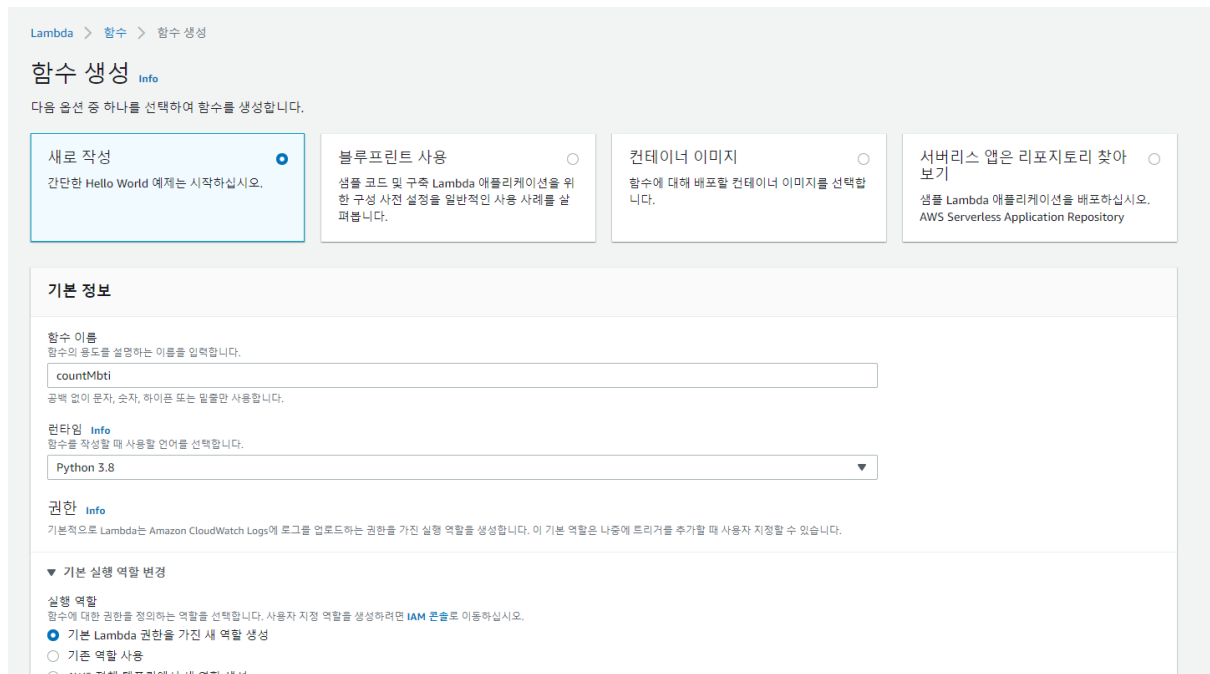
**준비물:** 만들어진 Lambda함수, (2) lambda로랜덤닉네임설정하기.pdf 파일, lambdaImageApiFunction.py 파일

기 작성된 **Lambda** 함수가 필요하기 때문에, 위의 **pdf** 파일을 이용해서 작성해주세요. 그렇지 않으면 이 **DashBoard api** 를 만드는 이유가 없습니다.

## 1. Visualizer Lambda 만들기, API gateway 연결하기

**pdf** 활용해서 **Lambda** 를 작성해 주세요.

**환경** 은 아래와 같습니다.



Lambda > 함수 > 함수 생성

### 함수 생성 Info

다음 옵션 중 하나를 선택하여 함수를 생성합니다.

**새로 작성** ☒  
간단한 Hello World 예제는 시작하십시오.

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

**컨테이너 이미지** ☐  
함수에 대해 배포할 컨테이너 이미지를 선택합니다.

**서버리스 앱은 리포지토리 찾아 보기** ☐  
샘플 Lambda 애플리케이션을 배포하십시오. AWS Serverless Application Repository

### 기본 정보

**함수 이름**  
함수의 종도를 설명하는 이름을 입력합니다.  
  
공백 없이 문자, 숫자, 하이픈 또는 밑줄만 사용합니다.

**런타임** Info  
함수를 작성할 때 사용할 언어를 선택합니다.

**권한** Info  
기본적으로 Lambda는 Amazon CloudWatch Logs에 로그를 업로드하는 권한을 가진 실행 역할을 생성합니다. 이 기본 역할은 나중에 트리거를 추가할 때 사용자 지정할 수 있습니다.

▼ 기본 실행 역할 변경

**실행 역할**  
함수에 대한 권한을 정의하는 역할을 선택합니다. 사용자 지정 역할을 생성하려면 **IAM 콘솔**로 이동하십시오.

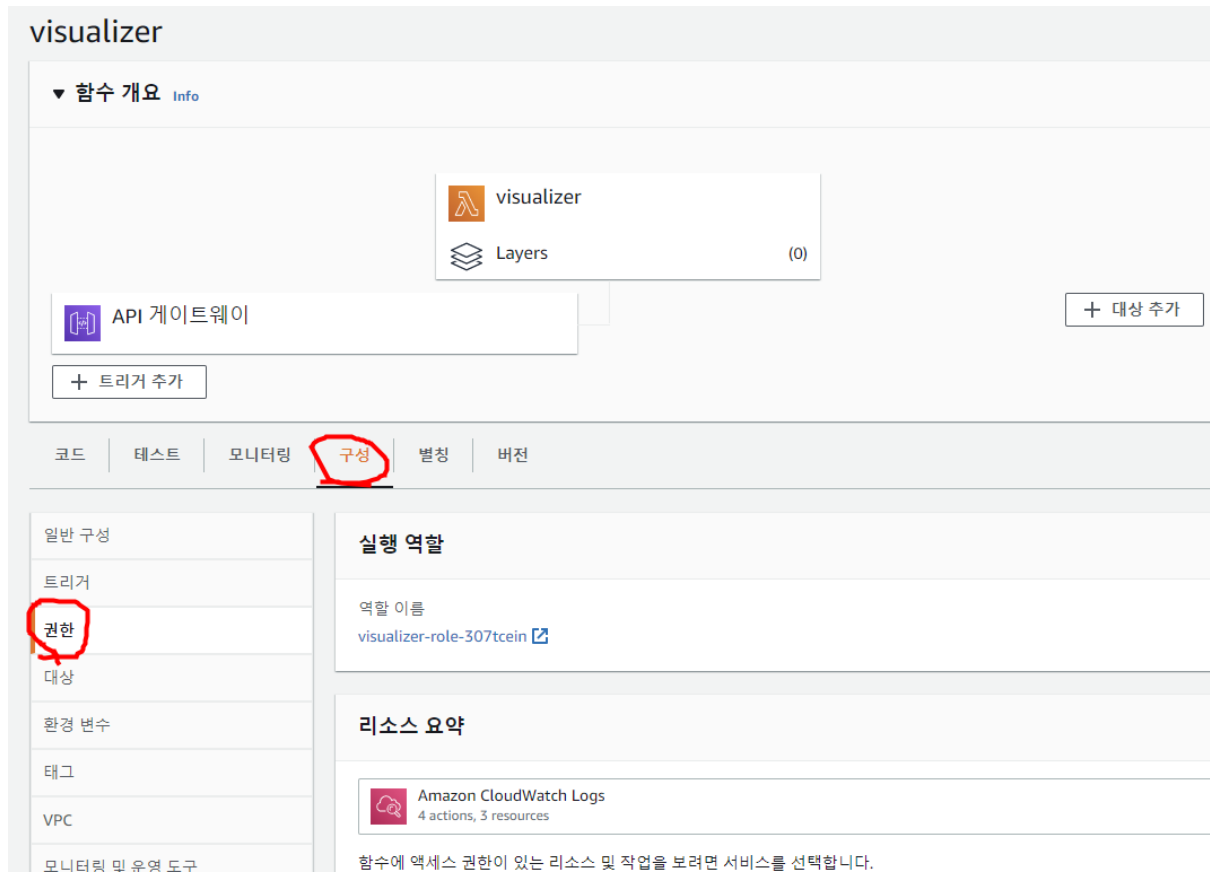
- ☒ 기본 Lambda 권한을 가진 새 역할 생성
- ☐ 기존 역할 사용
- ☐ AWS 정책 템플릿에서 새 역할 생성

함수 이름은 원하시는 대로 설정해 주세요.

이후 **pdf** 파일 대로 **Api gateway** 도 연결해 주세요.

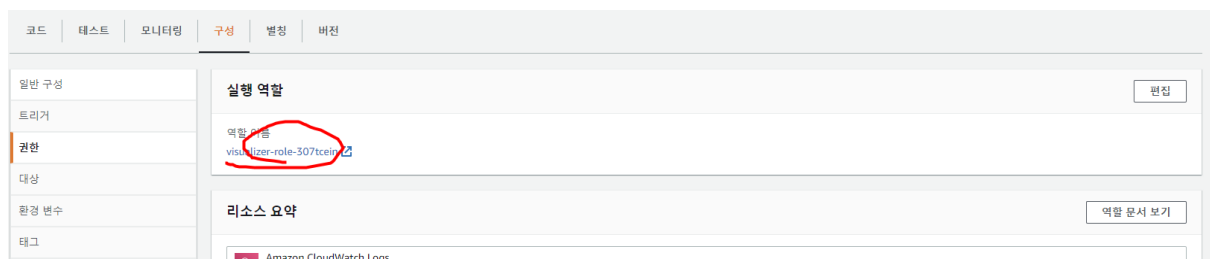
다시 **Lambda** 함수로 돌아갑시다.

## 2. Lambda에서 CloudWatch를 사용하기 위해 권한 부여하기



**Lambda** 함수의 **구성**, **권한** 으로 들어갑니다.

아래의 **실행 역할** 을 클릭해주세요.



**정책 연결** 을 클릭하여서 아래 사진과 같은 **정책** 을 추가해주세요.

이후 오른쪽의 **인라인 정책 추가** 를 클릭해주세요. **JSON** 을 클릭하세요.

권한 | 신뢰 관계 | 태그 | 액세스 관리자 | 세션 취소

▼ Permissions policies (2 정책이 적용됨)

정책 연결

정책 이름 ▼	정책 유형 ▼
▶ AWSLambdaBasicExecutionRole-0c58088e-5e6e-4197-a6a8-c1755d4489e7	관리형 정책

시각적 편집기 | **JSON** | 관리형 정책 가져오기

모두 확장 | 모두 축소

▼ 서비스 선택

복제 | 제거

▶ 서비스 서비스 선택

작업 작업 정의 이전 서비스 선택

리소스 리소스 적용 이전 작업 선택

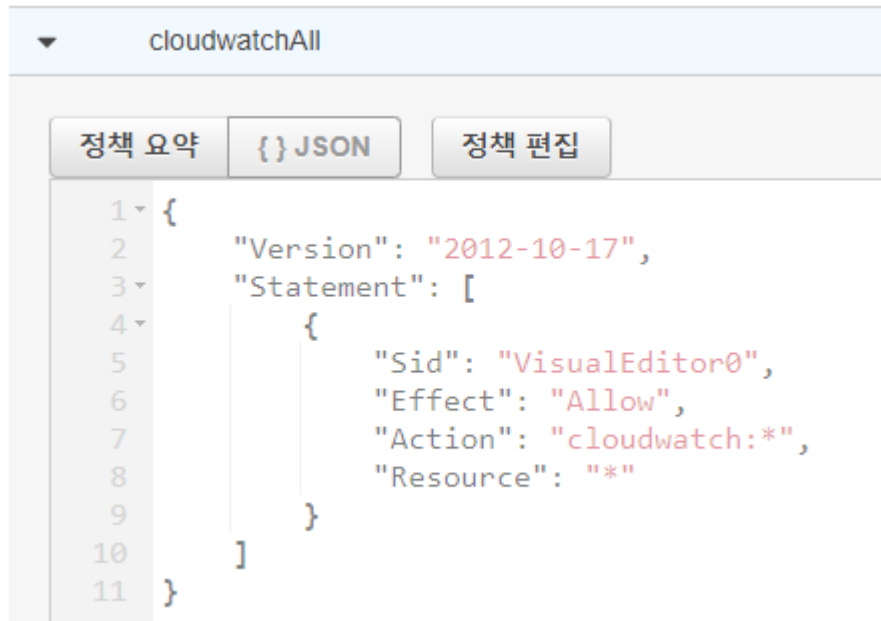
요청 조건 조건 지정 이전 작업 선택

권한 추가

아래의 `json` 을 `JSON` 에 옮겨주세요.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "cloudwatch:*",
      "Resource": "*"
    }
  ]
}
```

아래 `정책 검토` 및 `변경 내용을 저장합니다` 를 클릭하여 저장해주세요.



### 3. Lambda function 코드 수정하기

**Lambda** 함수를 보면 이런 부분이 있을거예요.

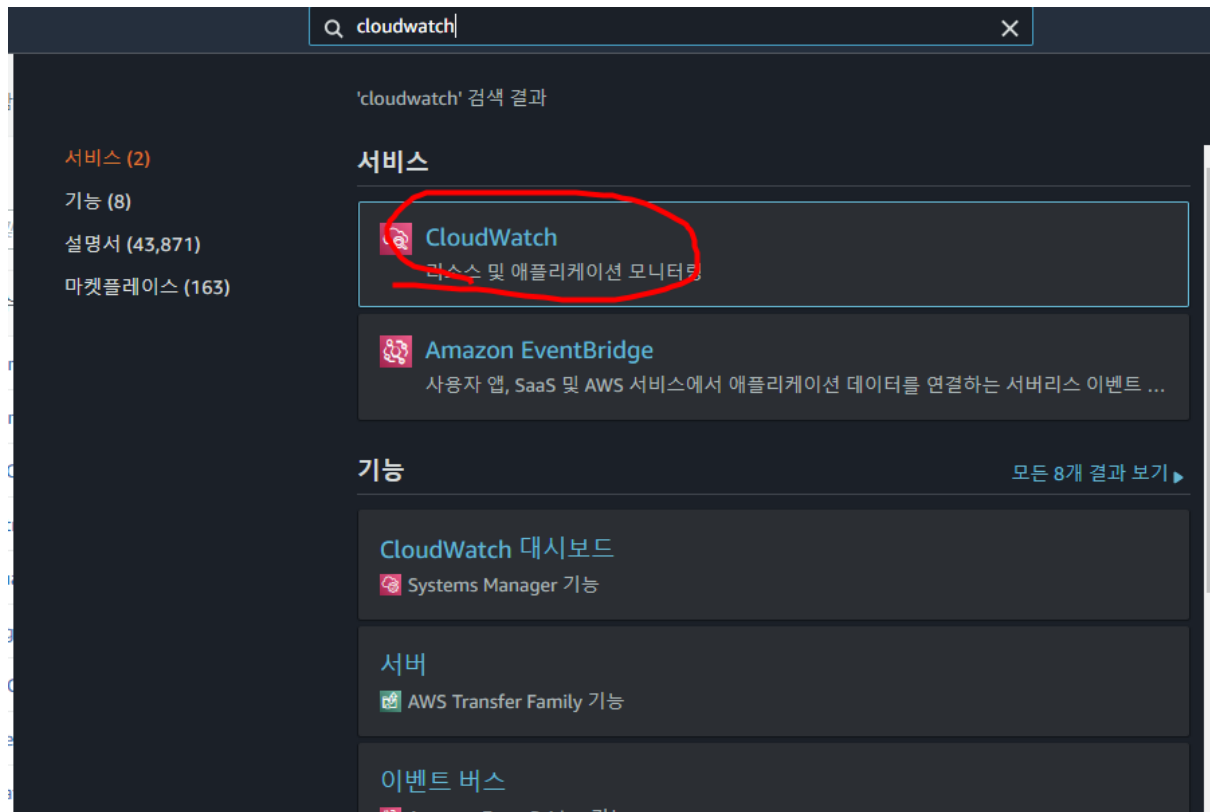
```
metric_sources = { your_matrix_source }
```

이 부분을 수정해주셔야 합니다.

탐색을 원하시는 지표로 바꿔 주세요.

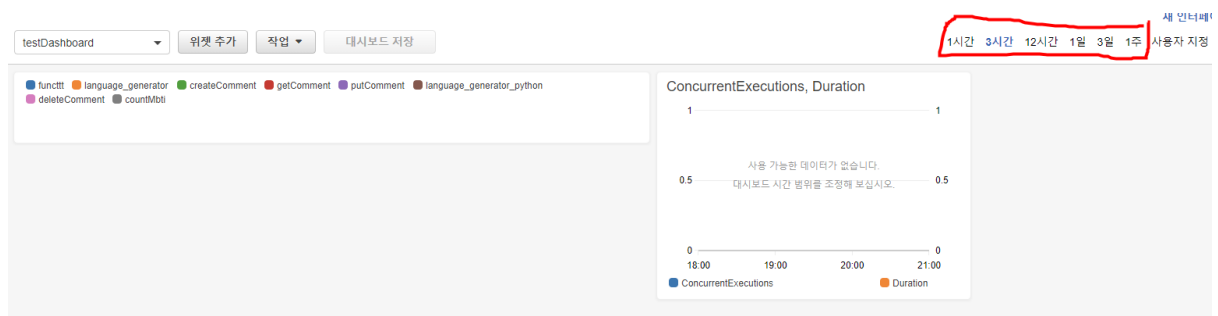
### 4. 지표 만들기 (많이 만들면 과금돼요. 주의하세요.)

**Cloudwatch** 의 **대시보드** 의 **지표** 를 먼저 보고 확인해봅시다.



여기 들어가셔서 **대시보드** 를 생성하세요. 여기서 중요한 점은 이 **DaashBoard** 이미지에 지표를 시간따라 다르게 보여 주고 싶은 경우가 있을 텐데, 빨간 네모 박스의 이 시간을 설정을 하시면 전체 시간대가 조정되는 겁니다.

예를 들면 저 시간이 **3시간** 으로 설정된 후 지표를 만들게 되면 전체 **3시간의 지표** 가 출력됩니다.



**위젯 추가** 를 눌러주세요.



원하시는 모양을 선택하시면 되는데, 여기서 주의할 점은 **번호는 안됩니다**. 다른 **api**를 사용해야 한다고 합니다.

저는 **행**으로 만들어 보겠습니다.

이 대시보드에 추가

어떤 데이터 원본에서 위젯을 생성하시겠습니까?

☐ 지표  
지표를 기반으로 위젯을 생성하고 다음 단계에서 위젯을 구성합니다.

☐ 로그  
CloudWatch Logs Insights의 쿼리 결과에 따라 위젯을 생성합니다.

**지표**를 선택하세요. 이후 **구성**을 누르세요.

제목 없는 그래프
1시간 3시간 12시간 1일 3일 1주 사용자 지정
행

CloudWatch 그래프가 비어 있습니다.  
여기에 표시할 일부 지표를 선택하십시오.

모든 지표 | 그래프로 표시된 지표 | 그래프 옵션 | 소스

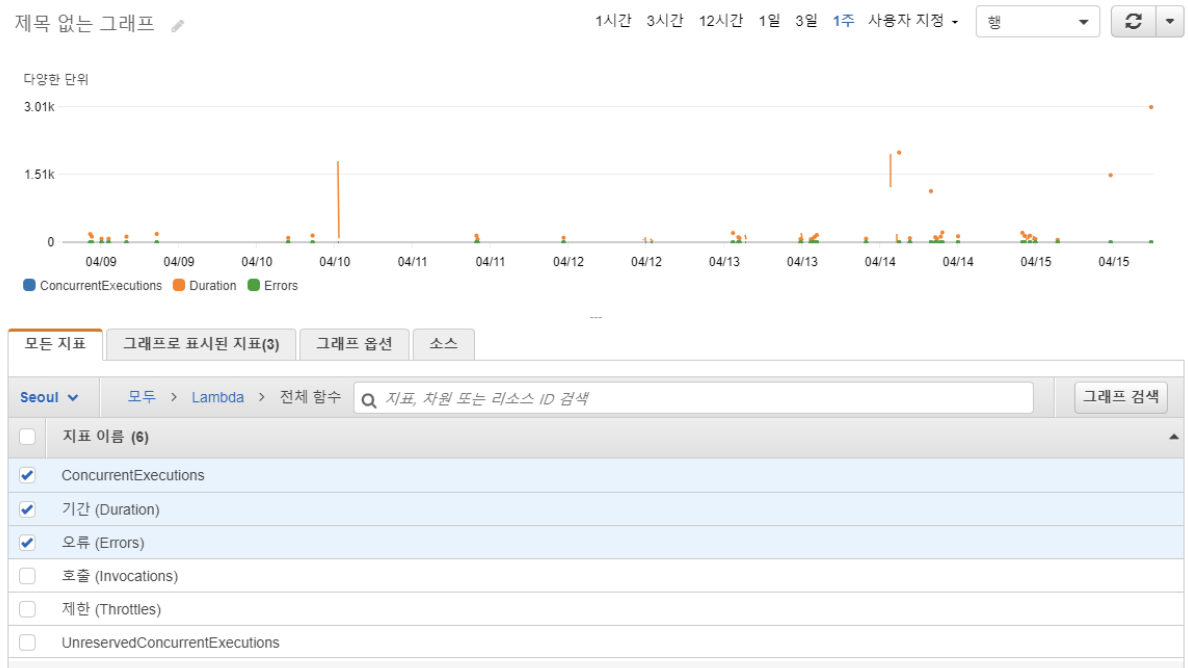
Seoul
지표, 차원 또는 리소스 ID 검색
그래프 검색

305지표

<b>ApiGateway</b> 38지표	<b>DynamoDB</b> 12지표	<b>EBS</b> 18지표
<b>EC2</b> 34지표	<b>Lambda</b> 86지표	<b>로그</b> 18지표
<b>RDS</b> 36지표	<b>S3</b> 10지표	<b>상태</b> 1지표

이제부터 원하는 **지표** 를 만들어 봅시다. 아래 **모든 지표** 에서 **Lambda** 를 누르시고.. 원하시는 대로 꾸려서 눌러보시면 됩니다.

예를 들면 저는 이렇게 꾸려보겠습니다.



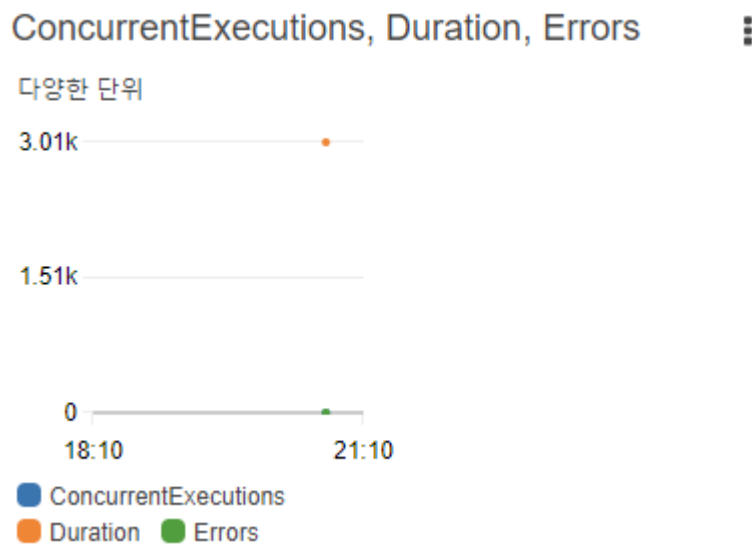
옆에 **소스** 버튼이 있는데 이것 눌러 봅시다.



여기에 있는 이 **JSON** 이 위의 **Matrix** 부분입니다. 원하시는 **보기** 를 누르셔서 **JSON** 을 fix시키시고, 이를 붙여넣으시면 됩니다.

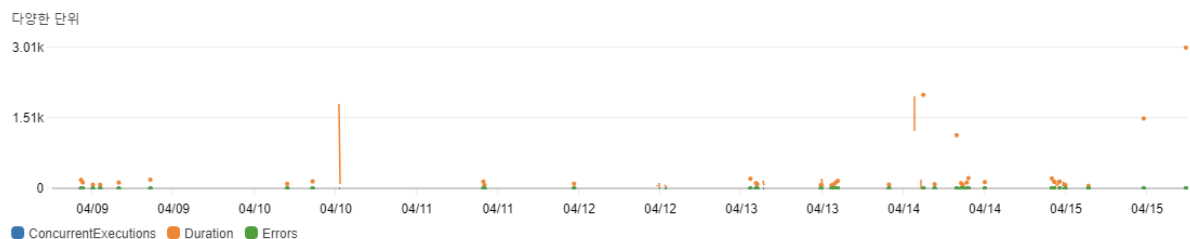
**보기** 는 뭘까요?

- **대시보드**: 조그마한 이미지로 나옵니다.



(대충 이런 모양)

- **이미지 API**: 지금 위에 예시로 보이는 것 처럼 길게 나옵니다.



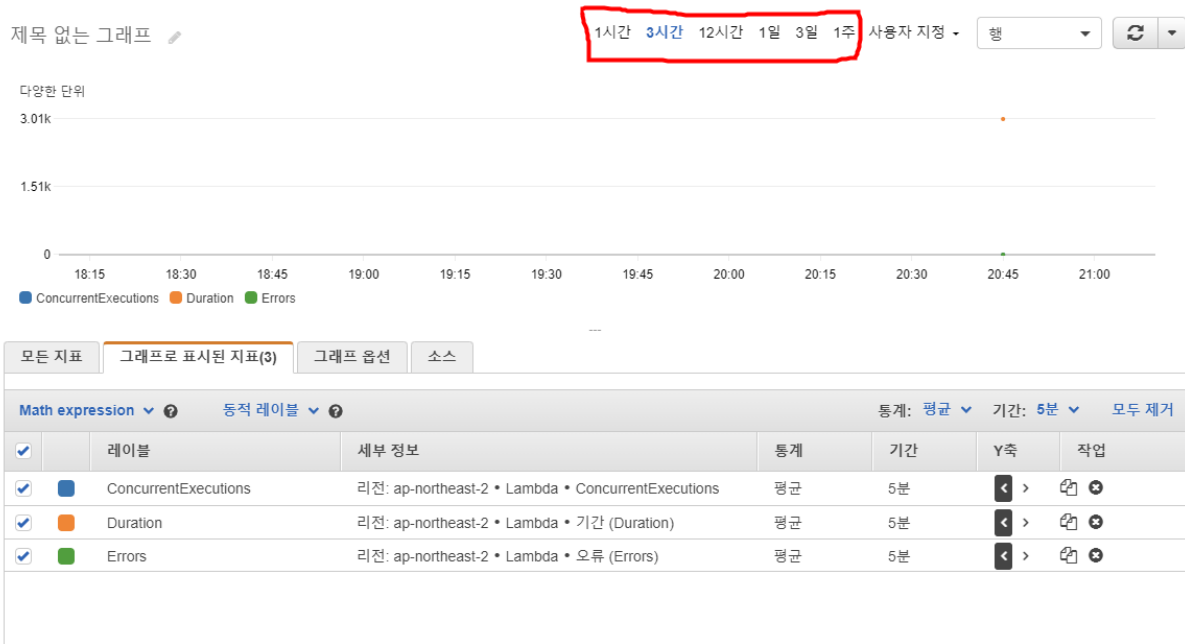
(대충 이런 모양)

## 주의사항

이 **위젯 생성** 에서 저기 빨간 박스 친 부분의 시간을 설정한다고 해도, 설정 안됩니다.

꼭 위에서 말씀드렸던 것처럼 **전체 대시보드의 시간을 바꾸셔야 합니다.**





이제 저는 이런 식으로 `python` 코드를 수정하겠죠!

```
metric_sources = {
    "view": "timeSeries",
    "stacked": false,
    "metrics": [
        [ "AWS/Lambda", "ConcurrentExecutions" ],
        [ ".", "Duration" ],
        [ ".", "Errors" ]
    ],
    "region": "ap-northeast-2",
    "period": 300
}
```

이후 `Lambda` 를 `Deploy` 하고 누르면 이미지가 도착하게 됩니다.

## 어디에쓰나요?

우리 `SSBTI` 에는 굳이 `AWS 콘솔` 에 들어오지 않아도 홈페이지에서 편하게 `log` 를 보는 `대시보드` 를 꾸리고자 했습니다. 이 때 가져다가 씁니다.

즉 `대시보드 콘솔` 확인하는 API의 주소(보통 `.env.local` )를 원하시는 주소로 바꿔주시면 되겠습니다!