

✨ Dash-Board Lambda Log API 기능 만들기

목적: AWS CloudWatch api를 이용해서 Lambda function의 호출 Log를 불러오는 api를 만듭니다.

작성자: 송은주

준비물: 만들어진 Lambda함수, (2) lambda로랜덤닉네임설정하기.pdf 파일, lambdaLogApiFunction.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.

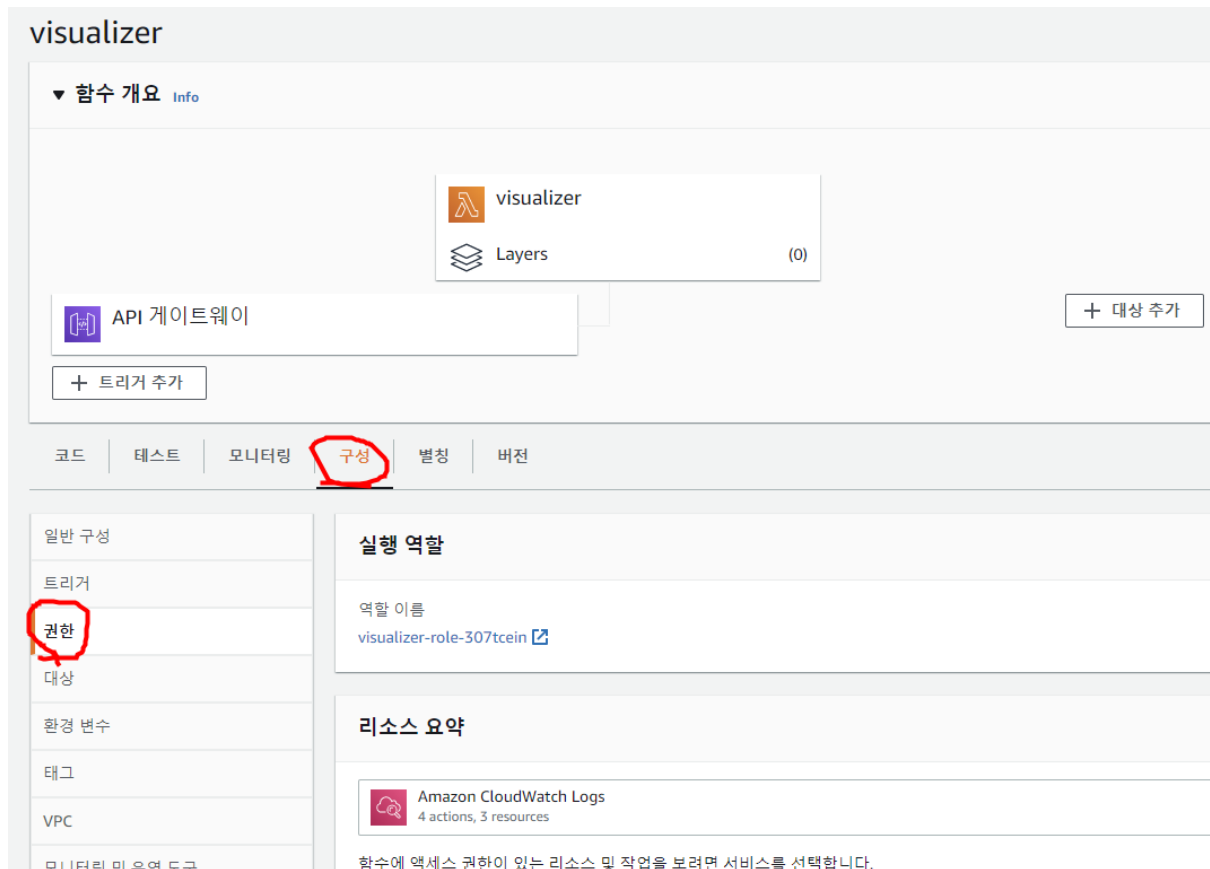
- ☒ 기본 Lambda 권한을 가진 새 역할 생성
- ☐ 기존 역할 사용
- ☐ AWS 정책 템플릿에서 새 역할 생성

함수 이름은 원하시는 대로 설정해 주세요.

이후 pdf 파일 대로 Api gateway 도 연결해 주세요.

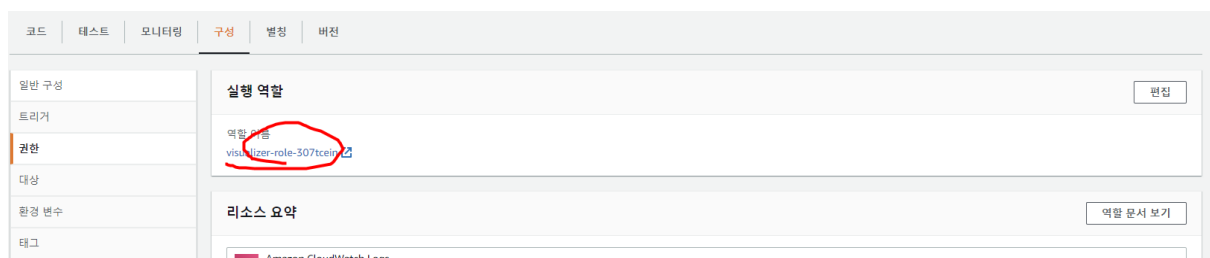
다시 Lambda 함수로 돌아갑시다.

2. Lambda에서 CloudWatch를 사용하기 위해 권한 부여하기



Lambda 함수의 구성, 권한으로 들어갑니다.

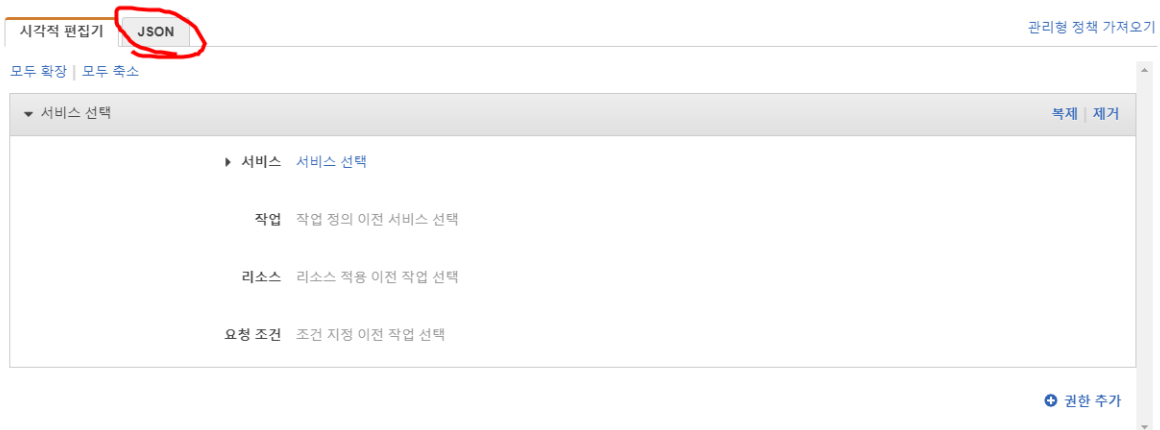
아래의 실행 역할을 클릭해주세요.





정책 연결 을 클릭하여서 사진과 같은 정책을 추가해주세요.

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



아래의 json 을 JSON 에 옮겨주세요.

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

정책은 사용자, 그룹, 또는 역할에 할당할 수 있는 AWS 권한을 정의합니다. 시각적 편집기에서 JSON을 사용하여 정책을 생성하고 편집할 수 있습니다. [자세히 알아보기](#)

시각적 편집기

JSON

관리형 정책 가져오기

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "VisualEditor0",
6       "Effect": "Allow",
7       "Action": "logs:*",
8       "Resource": "*"
9     }
10  ]
11 }
```

보안: 0 오류: 0 경고: 0 추천: 0

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

3. Lambda function 코드 수정하기

Lambda 함수를 보면 이런 부분이 있을거예요.

```
log_group = 'your lambda path'
```

이 부분을 수정해주셔야 합니다.

탐색을 원하시는 함수 이름으로 바꿔 주세요.

예를 들면 저는 **AWS Lambda** 에 함수가 이런 식으로 구성되어 있습니다.

	함수 이름
<input type="radio"/>	countMbti
<input type="radio"/>	countAll
<input type="radio"/>	getComment
<input type="radio"/>	metricvisualizer
<input type="radio"/>	visualizer
<input type="radio"/>	language_generator_python
<input type="radio"/>	putComment
<input type="radio"/>	deleteComment
<input type="radio"/>	createComment

이 중 `language_generator_python` 을 탐색하고 싶으면, 이렇게 수정해주세요.

`log_group = '/aws/lambda/language_generator_python'` 로 수정해주세요.

이후 `Deploy` 를 눌러 배포합니다.

4. 결과 보기

예를 들면 제가 만든 함수는 이런 결과를 내놓고 있습니다.

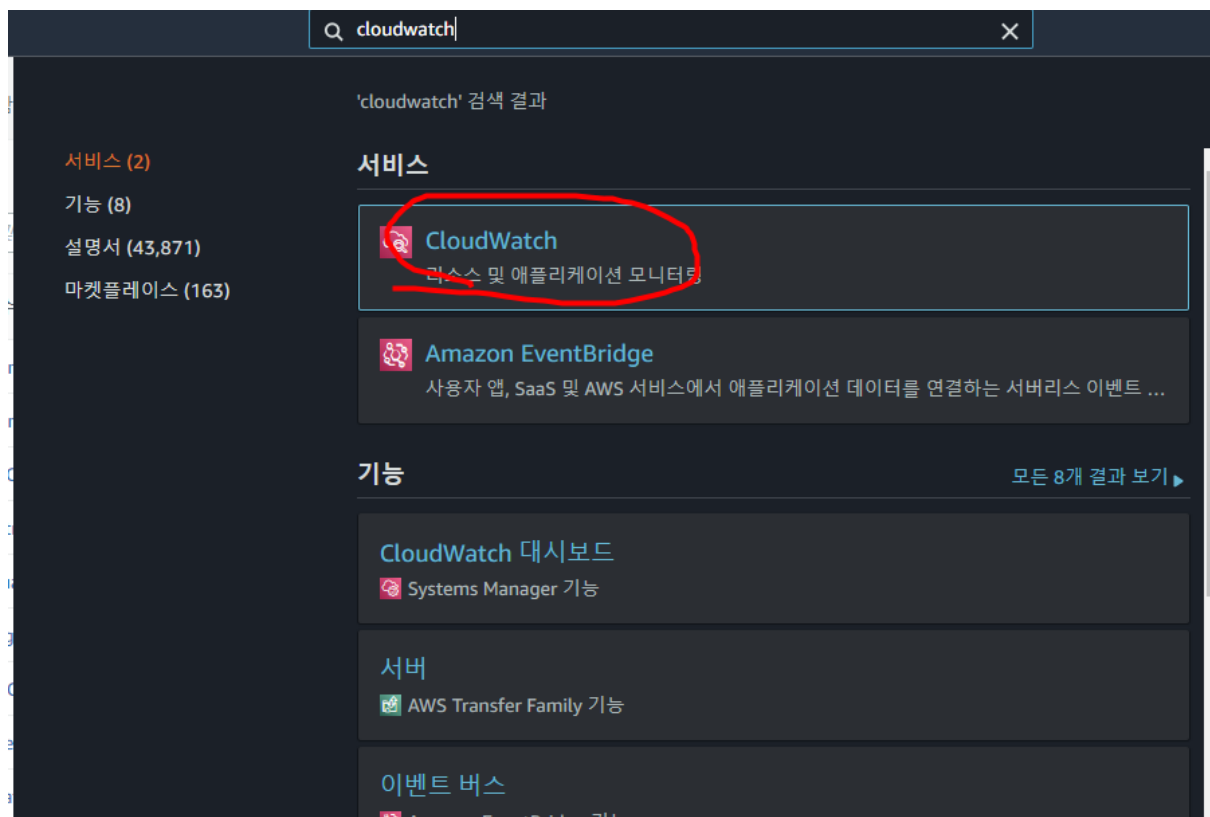
```
{
  "results":[
  ],
  "statistics":{
    "recordsMatched":0.0,
    "recordsScanned":0.0,
    "bytesScanned":0.0
  },
  "status":"Complete",
  "ResponseMetadata":{
    "RequestId":"08a1d05f-24ee-44c4-8a13-cb655270da33",
    "HTTPStatusCode":200,
    "HTTPHeaders":{
```

```
"x-amzn-requestid": "08a1d05f-24ee-44c4-8a13-cb655270da33",
"content-type": "application/x-amz-json-1.1",
"content-length": "110",
"date": "Thu, 15 Apr 2021 20:45:21 GMT"
},
"RetryAttempts": 0
}
}
```

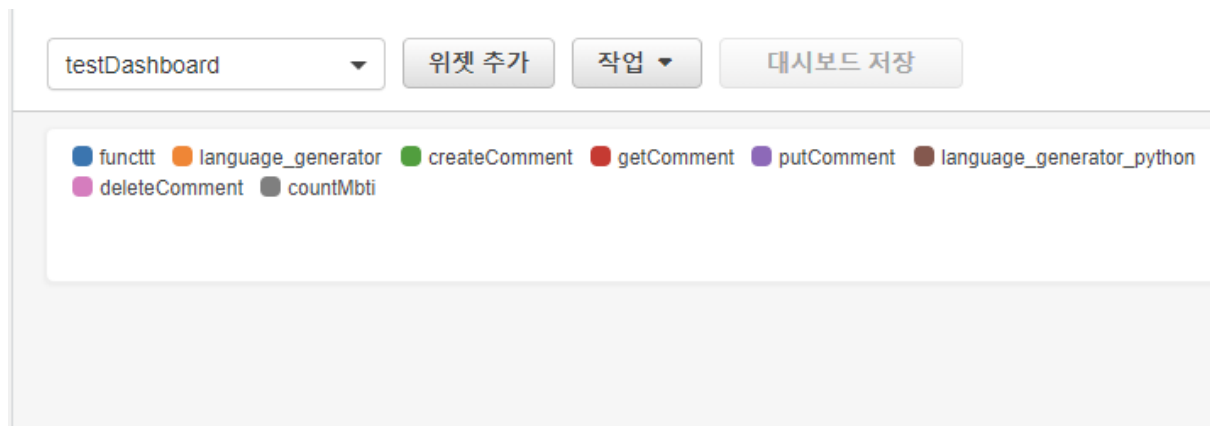
5. 이게 뭐예용? (Optional, 안해도 됨. 이 결과가 무엇인지 확인하고 싶은 사람만)

여기서부터는 대시보드에 위젯을 많이 만들면 **과금** 이 될 수 있으니 주의하세요!!!!(제가 과금한 경험으로는 최대 약 1,000원 정도였습니다.)

CloudWatch 의 대시보드 에서 log 를 받아오는 거예용.



여기 들어가서 **대시보드** 를 생성하시고 **위젯 추가** 를 눌러주세요.



로그 테이블 을 눌러주세요.

구성할 위젯 유형 선택



누르시면 아래와 같은 창이 뜹니다.

testDashboard 대시보드에 대해 새 위젯을 생성하는 중입니다.

로그 그룹 선택

```

1 fields.@timestamp, @message
2 | sort @timestamp desc
3 | limit 20

```

쿼리 실행

저장

기록

로그

시각화

여기에 있는 저 쿼리 가 Lambda 에 있는 query 와 같습니다. 로그 그룹 을 누르면 이렇게 원하는 함수의 실행 로그를 볼 수 있는 거예요.

로그 그룹 선택

검색하려면 입력

☐ /aws/lambda/countMbti

☐ /aws/lambda/createComment

☐ /aws/lambda/deleteComment

이걸 왜 쓰나요?

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

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