

## Lab2. Serverless Hello World

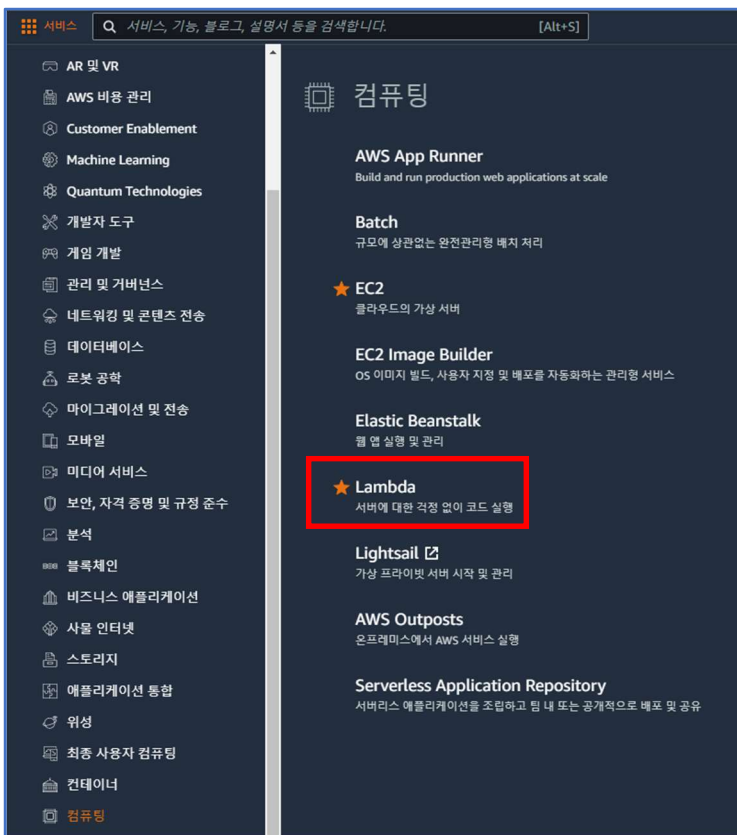
### 목적

서버리스 "Hello, World!" 실행

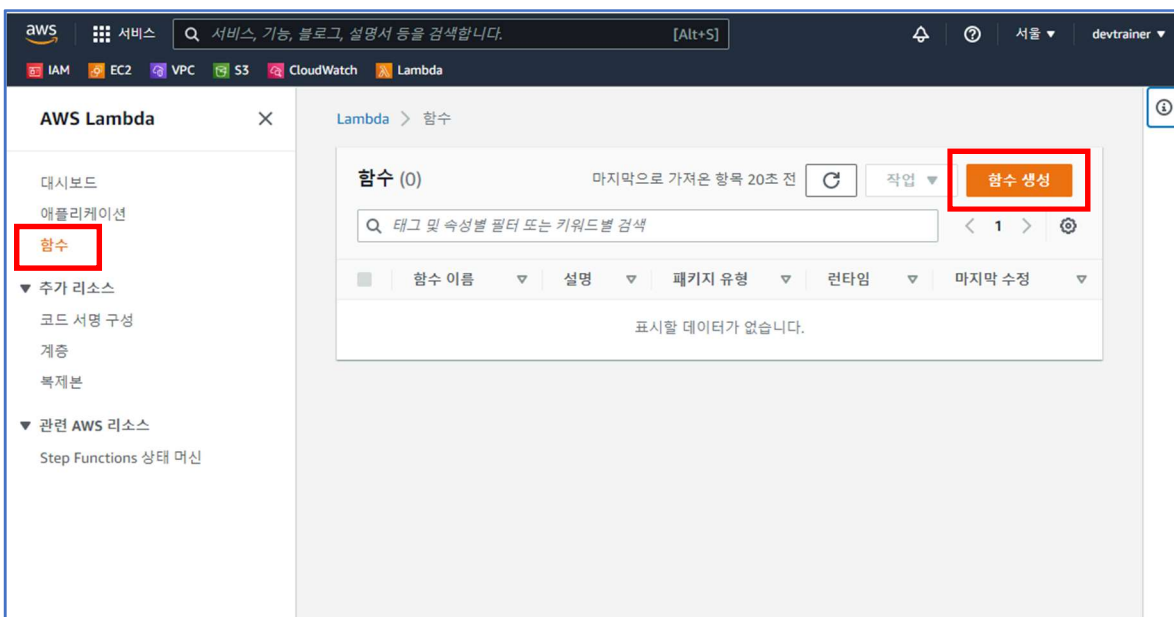
### 사전 준비물

AWS Free-Tier 계정

1. 로그인 후, [서비스] > [컴퓨팅] > [Lambda]를 클릭하여 **AWS Lambda** 페이지로 이동한다.



2. [AWS Lambda] 페이지에서 왼쪽 메뉴 중 [함수]를 선택한다. 반드시 현재 실습은 **서울 Region**에서 수행한다. 페이지 오른쪽 상단의 [함수 생성]을 클릭한다.



3. [함수 생성] 페이지에서 함수를 생성하기 위해 [블루프린트 사용]을 선택한다. 그리고, 그 아래 [블루프린트]에 **hello-world-python**이라고 입력한다.

Lambda > 함수 > 함수 생성

### 함수 생성 정보

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 찾아보기  
샘플 Lambda 애플리케이션을 배포하십시오. AWS Serverless Application Repository

**블루프린트 (46) 정보**  
Q hello-world-python X

마지막으로 가져온 항목 3분 전 **내보내기**

< 1 2 3 4 5 >

4. 필터 상자에 이름=hello-world-python을 선택하고 그 아래의 **hello-world-python**을 선택한 후, [구성]을 클릭한다.

**블루프린트 (1/46) 정보** 마지막으로 가져온 항목 4분 전 **내보내기**

Q 블루프린트 필터링

일치 항목: 1개

< 1 >

**이름 = hello-world-python** X

필터 지우기

**hello-world-python**

A starter AWS Lambda function.

python3.7

취소

**구성**

5. [기본 정보]페이지에서, 다음과 같이 설정한다.

A. [함수 이름] : hello-world-python

B. [실행 역할] : AWS 정책 템플릿에서 새 역할 생성

C. [역할 이름] : lambda\_basic\_execution

Lambda > 함수 > 함수 생성 > 블루프린트 hello-world-python 구성

### 기본 정보 정보

함수 이름

hello-world-python

실행 역할

함수에 대한 권한을 정의하는 역할을 선택합니다. 사용자 지정 역할을 생성하려면 IAM 콘솔로 이동합니다.

☐ 기본 Lambda 권한을 가진 새 역할 생성

☐ 기존 역할 사용

☒ AWS 정책 템플릿에서 새 역할 생성

**이 역할을 생성하는 데 몇 분 정도 걸릴 수 있습니다. 역할을 삭제하거나 이 역할에서 신뢰 또는 권한 정책을 편집하지 마십시오.**

역할 이름

새 역할의 이름을 입력합니다.


lambda\_basic\_execution


공백 없이 문자, 숫자, 하이픈 또는 밑줄만 사용합니다.

6. 페이지를 아래로 스크롤다운하여, **[Lambda 함수 코드]**에서 **[런타임]**과 미리 템플릿에 의해 작성된 함수 코드를 확인한다. **[런타임]** 중 위에서 이미 파이썬을 선택했기 때문에, **Python 3.7**이 표시된다. 또한 **AWS Lambda**가 코드 실행을 시작할 수 있는 **핸들러**(코드의 메소드/함수)를 지정할 수 있는데, 역시 템플릿에 의해 미리 작성된 **lambda\_handler**를 확인할 수 있다. 확인후 **[함수 생성]** 버튼을 클릭한다.

### Lambda 함수 코드

코드는 선택한 블루프린트에 의해 사전 구성되어 있습니다. 함수를 생성한 후 코드를 구성할 수 있습니다. Lambda 함수 배포에 대해 [자세히 알아보세요](#).

 이 함수에 외부 라이브러리가 포함됩니다.



런타임

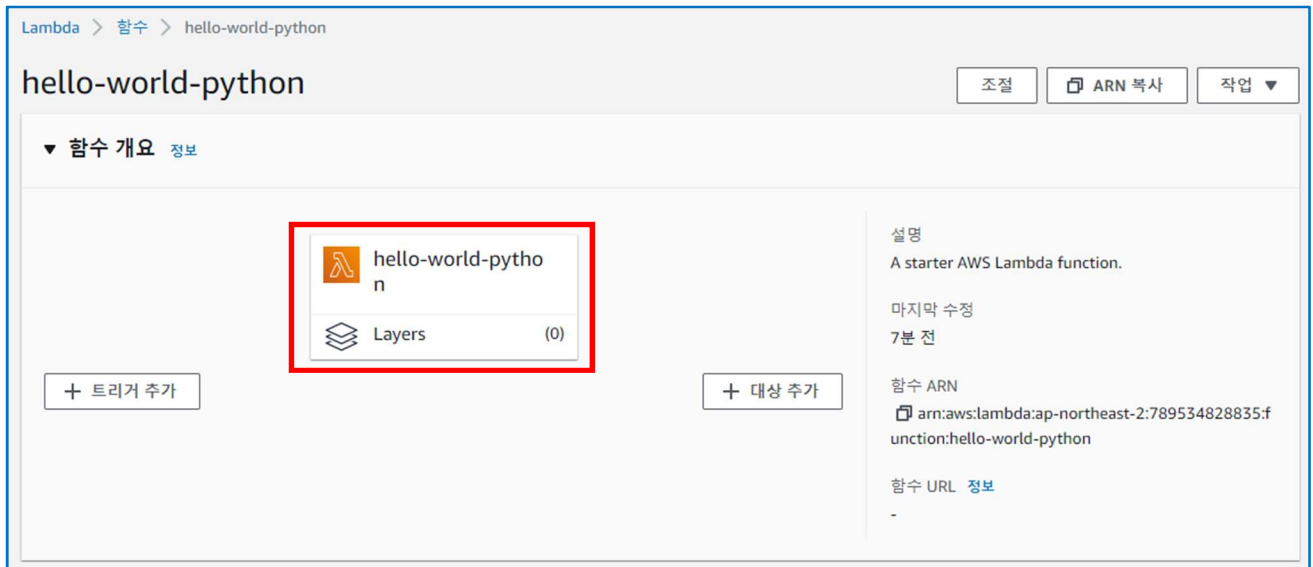
Python 3.7

```
1 import json
2
3 print('Loading function')
4
5
6 def lambda_handler(event, context):
7     #print("Received event: " + json.dumps(event, indent=2))
8     print("value1 = " + event['key1'])
9     print("value2 = " + event['key2'])
10    print("value3 = " + event['key3'])
11    return event['key1'] # Echo back the first key value
12    #raise Exception('Something went wrong')
13
```

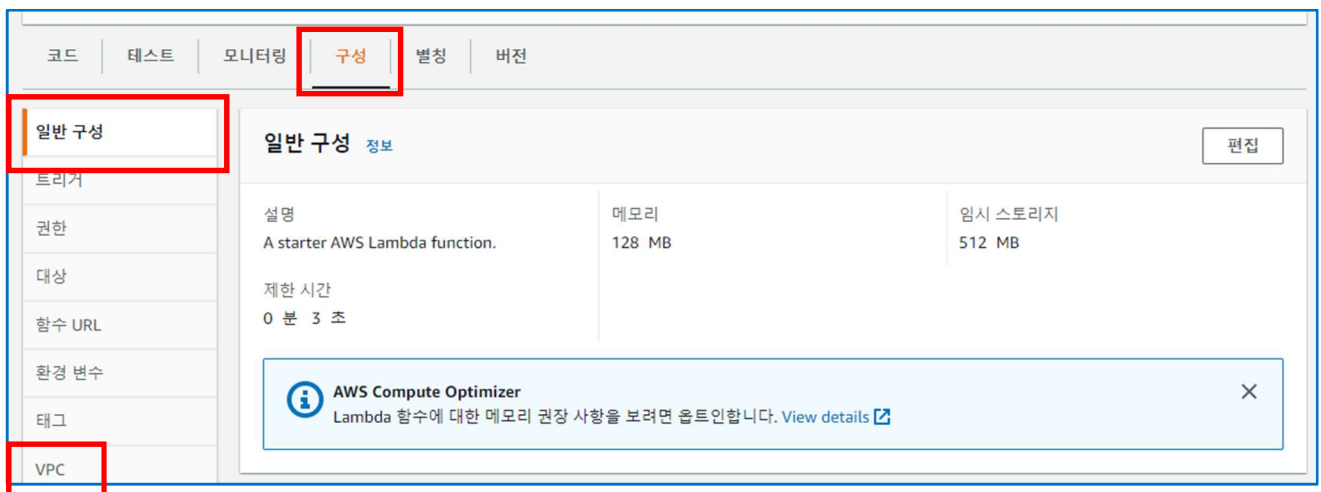
취소

함수 생성

7. 함수가 성공적으로 생성되었다.



8. 페이지를 스크롤다운하여 [구성]탭의 [일반 구성]을 확인해보자. 여기서 [메모리], [제한 시간]을 구성할 수 있으며, 그 아래의 [VPC]등도 설정할 수 있다. 이번 실습에서는 기본값 그대로 사용하기로 한다.



9. 이제 테스트를 설정할 차례이다. [테스트] 탭을 선택한다. [테스트 이벤트] 섹션에서 [이벤트 작업 테스트]를 [새 이벤트 생성]을 선택하고 [이벤트 이름]은 HelloWorldEvent를 입력한다.

코드 | **테스트** | 모니터링 | 구성 | 별칭 | 버전

### 테스트 이벤트

이벤트를 저장하지 않고 함수를 호출하려면 JSON 이벤트를 구성한 다음 테스트를 선택합니다.

이벤트 작업 테스트

- ☒ 새 이벤트 생성
- ☐ 저장된 이벤트 편집

이벤트 이름

HelloWorldEvent

문자, 숫자, 점, 하이픈 및 밑줄을 사용하여 최대 25자로 구성합니다.

10. [이벤트 공유 설정]은 기본값으로, [템플릿]은 hello-world를 목록에서 선택한다. 그 아래 [이벤트 JSON]에서 "key1"의 값을 "hello, world!"로 변경한다. 그리고 [테스트] 버튼을 클릭하여 테스트를 수행한다.

### 테스트 이벤트

이벤트를 저장하지 않고 함수를 호출하려면 JSON 이벤트를 구성한 다음 테스트를 선택합니다.

이벤트 작업 테스트

- ☒ 새 이벤트 생성
- ☐ 저장된 이벤트 편집

이벤트 이름

HelloWorldEvent

문자, 숫자, 점, 하이픈 및 밑줄을 사용하여 최대 25자로 구성합니다.

이벤트 공유 설정

- ☒ 프라이빗
- ☐ 공유 가능

이 이벤트는 공유 가능한 이벤트에 액세스하고 이를 사용할 수 있는 권한이 있는 동일한 계정 내 IAM 사용자가 사용할 수 있습니다. 자세히 알아보기

템플릿 - 선택 사항

hello-world

### 이벤트 JSON

JSON 형식 지정

```
1 {  
2   "key1": "hello, world!",  
3   "key2": "value2",  
4   "key3": "value3"  
5 }
```

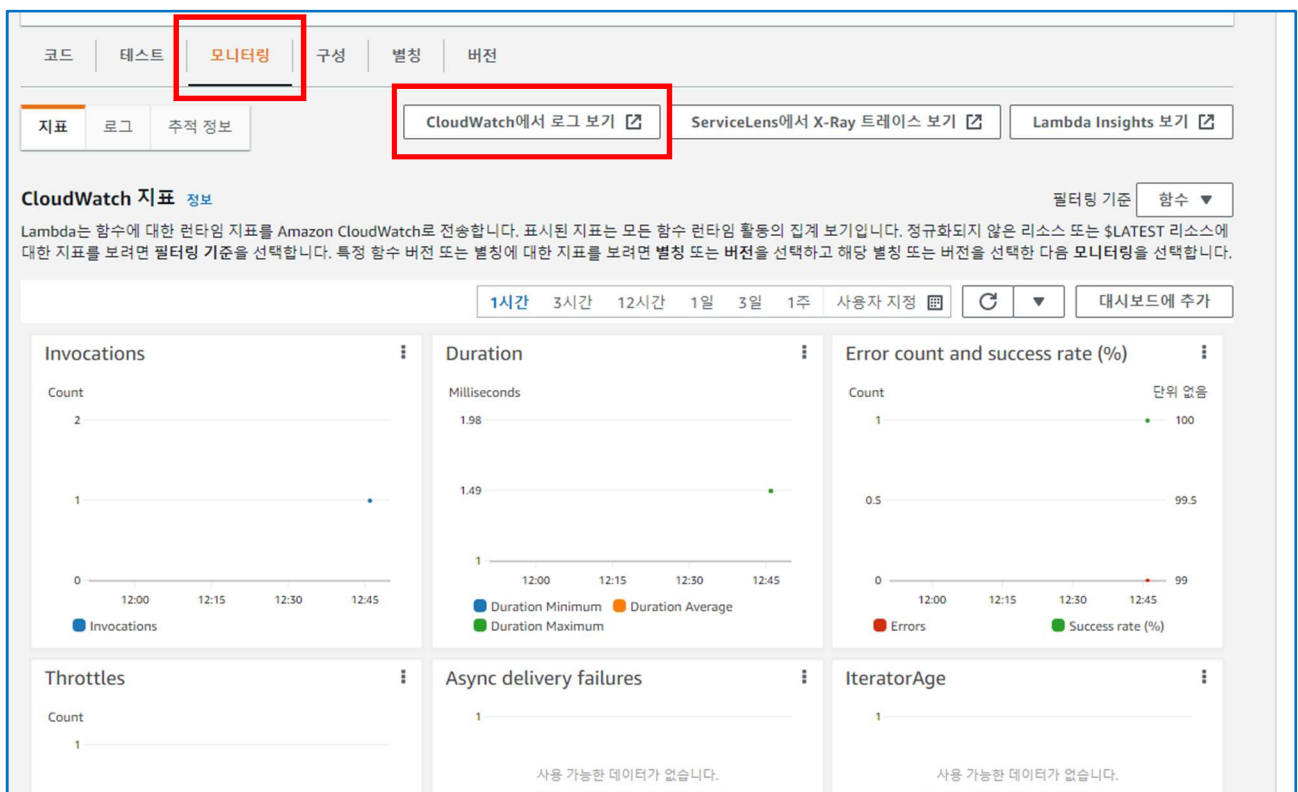
11. 함수가 성공적으로 실행되면, 다음과 같은 결과를 확인할 수 있다. [실행 결과:성공, 그리고 출력 예상 값 "hello, world!", 그리고 [요약]과 [로그 출력]을 확인할 수 있다.

The screenshot displays the AWS Lambda console interface for a specific function execution. It is divided into three main sections, each highlighted with a red box:

- 실행 결과: 성공 (로그)**: This section shows the execution status as 'Success' and provides a link to view the logs. Below this, there is a '세부 정보' (Details) section with the text: '아래 영역은 실행 로그의 마지막 4KB를 보여줍니다.' (The area below shows the last 4KB of the execution log.) and the output: '"hello, world!"'.
- 요약**: This section provides a summary of the execution metrics in a two-column table:

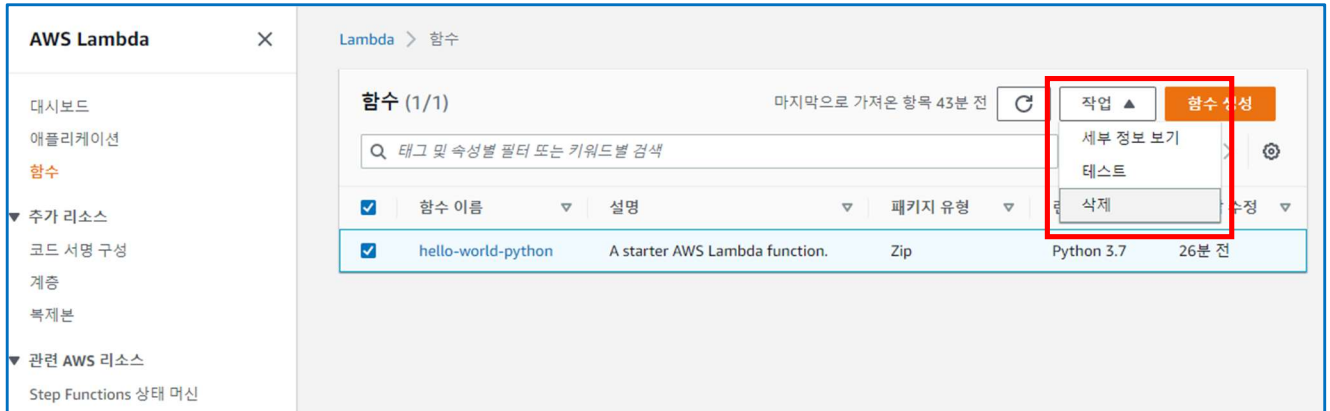
요청 ID	요청 ID
코드 SHA-256	e5d6d611-eace-486b-b52b-d46b1361c9d7
시작 시간	기간
122.82 ms	1.49 ms
청구 시간	리소스 구성
2 ms	128 MB
사용된 최대 메모리	
36 MB	
- 로그 출력**: This section shows the raw log output from the function. It includes the request ID, function name, and the output 'value1 = hello, world!'. It also provides a summary of the execution: 'START RequestId: e5d6d611-eace-486b-b52b-d46b1361c9d7 Version: \$LATEST', 'Loading function', 'value1 = hello, world!', 'value2 = value2', 'value3 = value3', 'END RequestId: e5d6d611-eace-486b-b52b-d46b1361c9d7', and 'REPORT RequestId: e5d6d611-eace-486b-b52b-d46b1361c9d7 Duration: 1.49 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 36 MB Init Duration: 122.82 ms'.

12. AWS Lambda는 함수를 자동으로 모니터링하고 [Amazon CloudWatch]를 통해 지표를 보고한다. 그 내용을 [모니터링] 탭을 클릭하여 확인할 수 있다.

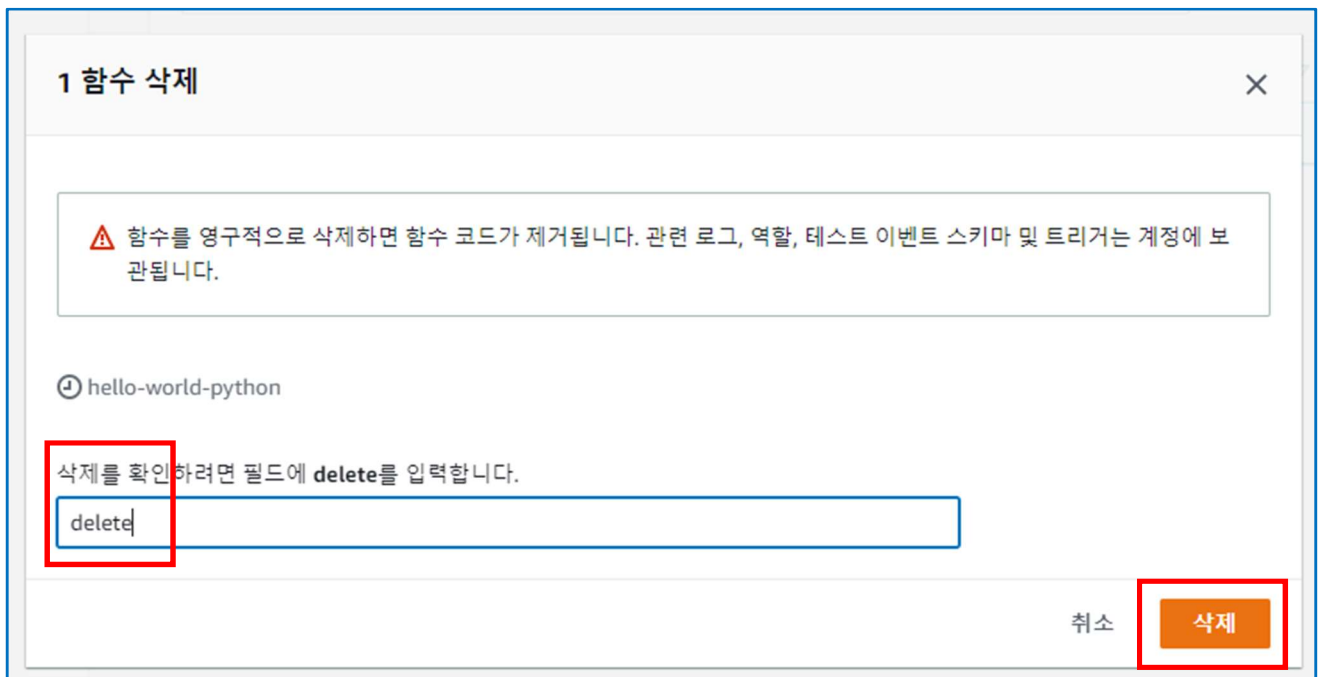




13. **AWS Lambda** 함수를 콘솔에서 삭제할 수 있다. 왼쪽 메뉴에서 **[함수]**를 선택하고 **[함수]**페이지에서 해당 함수를 선택한 후, **[작업]**에서 **[삭제]**를 선택하면 된다.



14. **[함수 삭제]** 창에서, 삭제를 위해 **delete**를 입력하고 **[삭제]**를 클릭한다.



15. [닫기]를 클릭하여 실습을 마친다.

1 함수 삭제

⚠ 함수를 영구적으로 삭제하면 함수 코드가 제거됩니다. 관련 로그, 역할, 테스트 이벤트 스키마 및 트리거는 계정에 보관됩니다.

✔ hello-world-python

삭제를 확인하려면 필드에 **delete**를 입력합니다.

delete

취소

닫기