

사용하기 전에

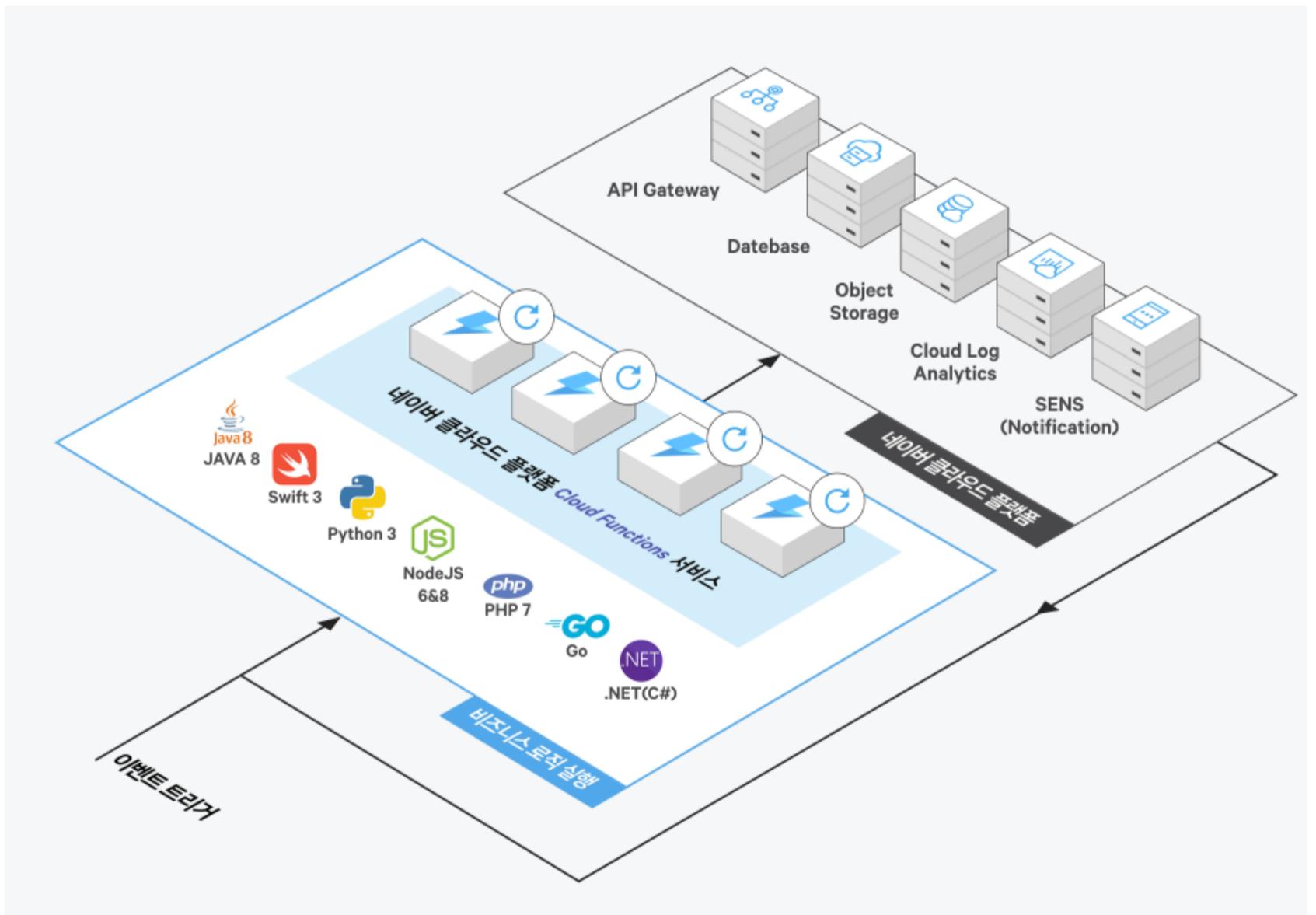
Q. Cloud Functions 를 사용하려면 어떻게 해야 하나요?

- 콘솔에 접속한 다음에 아래의 절차를 통해 Cloud Functions 를 생성하고 동작을 확인하실 수 있습니다.

- 01 Action 생성
- 02 Trigger 생성
- 03 Action & Trigger 연결
- 04 실행 및 결과 확인

Q. Cloud Functions 는 어떻게 동작하나요?

- Cloud Functions 는 사용자의 인프라 관리 부담없이 원하는 로직을 분산된 클라우드 환경에서 동작하고 결과를 반환합니다.



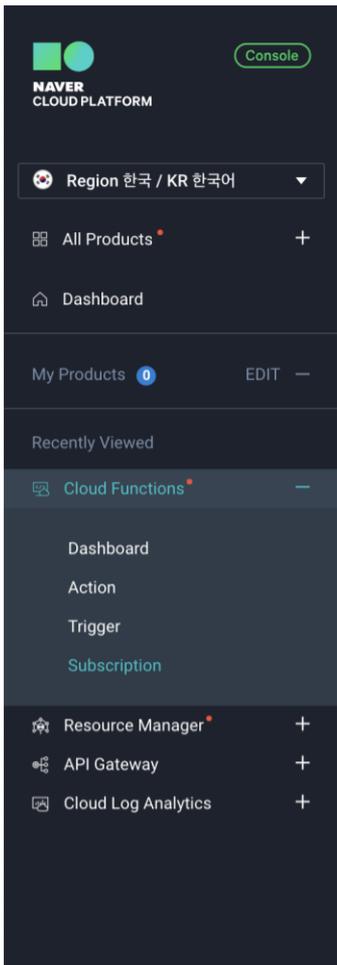
Cloud Functions 사용하기

Step 1. 서비스 사용하기

Cloud Functions 를 사용하기 위해서는 먼저 네이버 클라우드 플랫폼 콘솔에서 서비스 사용 신청을 해야 합니다.

서비스 신청

- 서비스 신청 페이지에서 이용 신청을 눌러 서비스 사용 계약을 진행합니다.



Cloud Functions

서버 관리에 대한 부담 없이 손쉽게 원하는 비즈니스 로직을 제공하는 시스템

서버를 확장성 있게 구성하거나 관리할 필요 없이 비즈니스 로직 코드 작성에 집중할 수 있습니다. 사용한 컴퓨팅 시간만큼 비용을 지불하고 효율적으로 서비스를 운영 할 수 있습니다.

- ✓ 서버 관리가 불필요
- ✓ 개발 속도 향상
- ✓ On-demand Execution
- ✓ 다양한 작동 방식

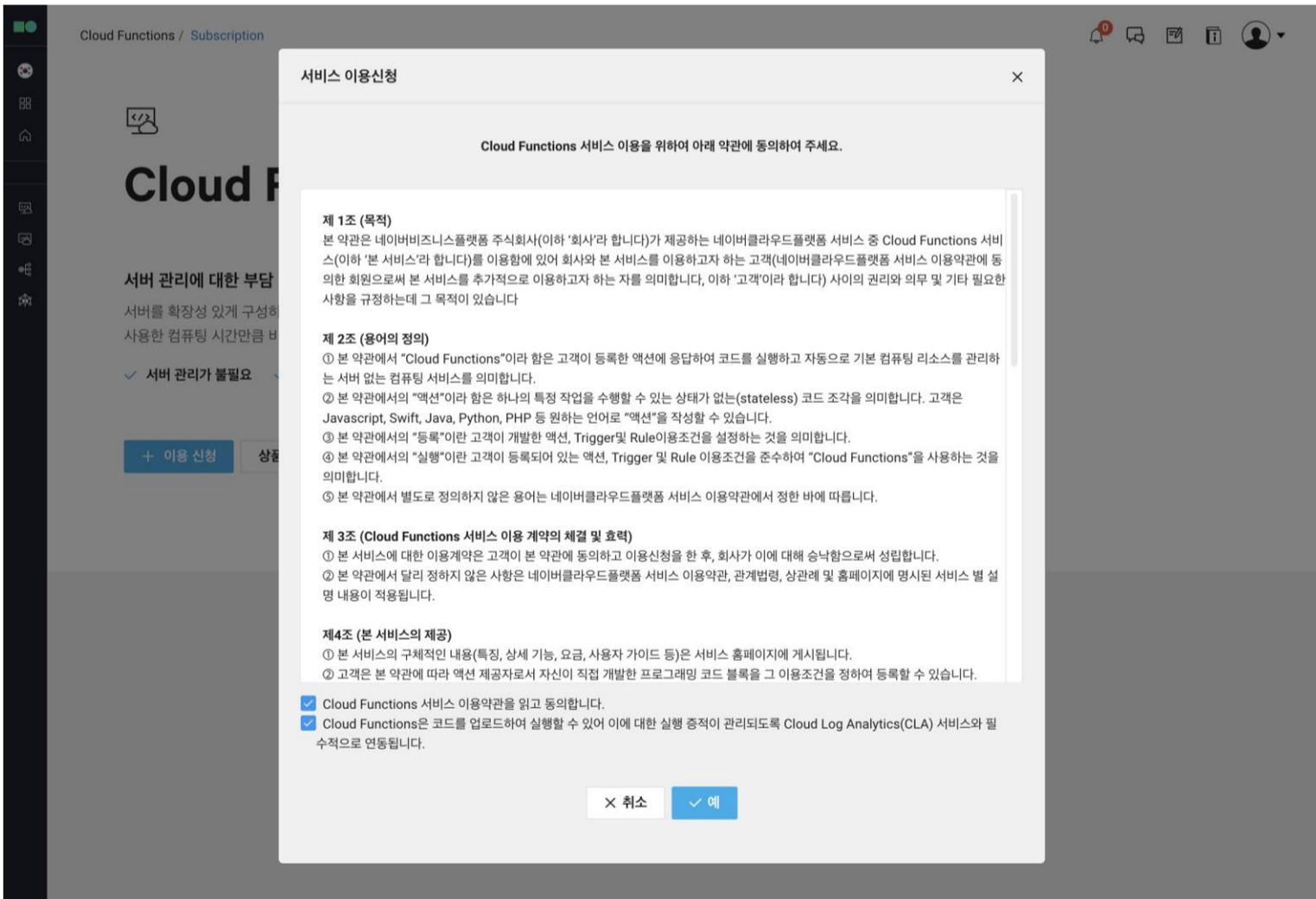
[+ 이용 신청](#)
[상품 더 알아보기](#)
[새로고침](#)



현재 이용 신청 내역이 없습니다.

[이용 신청](#) 버튼을 클릭하여 Cloud Functions을 시작하세요

- 서비스 신청 시, 관련 약관 내용에 동의해야 합니다.



Step 2. 코드 작성하기

코드를 작성합니다.

다음 코드는 파라미터인 Params 를 전달받아 Params 에 Name 과 Place 가 존재하면 해당 문자열을 반환하고, 존재하지 않으면 World 와 NCP 를 반환하는 액션(Action)의 예입니다.

- Node.js 6.x, 8.x

```
function main(params) {
```

- ```
var name = params.name || 'World';
```
- ```
var place = params.place || 'NCP';
```
- ```
return {payload: 'Hello, ' + name + ' in ' + place + '!'};
```
- }

- Python 3

- ```
def main(args):
```
- ```
 name = args.get("name", "World")
```
- ```
    place = args.get("place", "NCP")
```
- ```
 return {"payload": "Hello, " + name + " in " + place + "!"}
```

- Swift 3.1.1

- ```
func main(args: [String:Any]) -> [String:Any] {
```
- ```
 let name = args["name"] ?? "World"
```
- ```
    let place = args["place"] ?? "NCP"
```
- ```
 return ["payload" : "Hello, \(name) in \(place)"]
```
- }

- PHP 7.1

- ```
<?php
```
- ```
function main(array $args) : array
```
- {
- ```
    $name = $args["name"] ?? "World";
```
- ```
 $place = $args["place"] ?? "NCP";
```
- ```
    return ["greeting" => "Hello, $name in $place!"];
```
- }

- Java 8

Java 는 웹에서 코드 형태로 작성할 수 없고, JAR 형태로 컴파일하여 사용 가능합니다.

```
import com.google.gson.JsonObject;

public class Hello {
    public static JsonObject main(JsonObject args) {
        String name = "World";
        String place = "Naver";
        if (args.has("name"))
            name = args.getAsJsonPrimitive("name").getString();
        if (args.has("place"))
            place = args.getAsJsonPrimitive("place").getString();

        JsonObject response = new JsonObject();
        response.addProperty("payload", "Hello, " + name + " in " + place + "!");
        return response;
    }
}
```

생성할 액션의 이름, 입력 파라미터의 설명과 액션의 설명을 입력합니다.

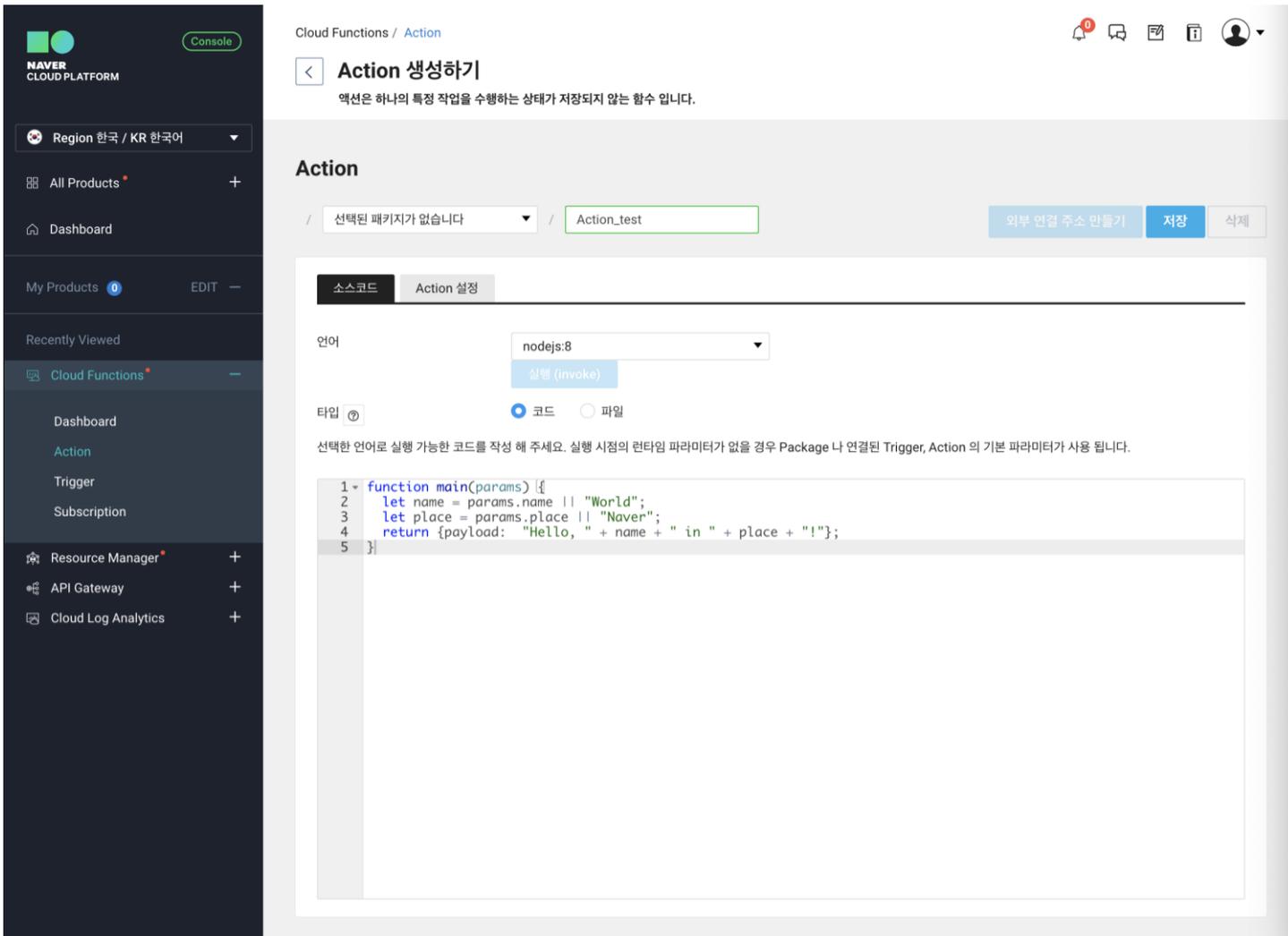
작성된 코드를 입력하고, 디폴트 파라미터로 JSON 형식의 Sample input 을 입력합니다.

- 디폴트 파라미터의 예(일반적인 경우 JSON 필수)

- {
- ```
"name": "NCP",
```
- ```
"place": "Cloud Function"
```
- }

Step 3. 액션(Action) 생성하기

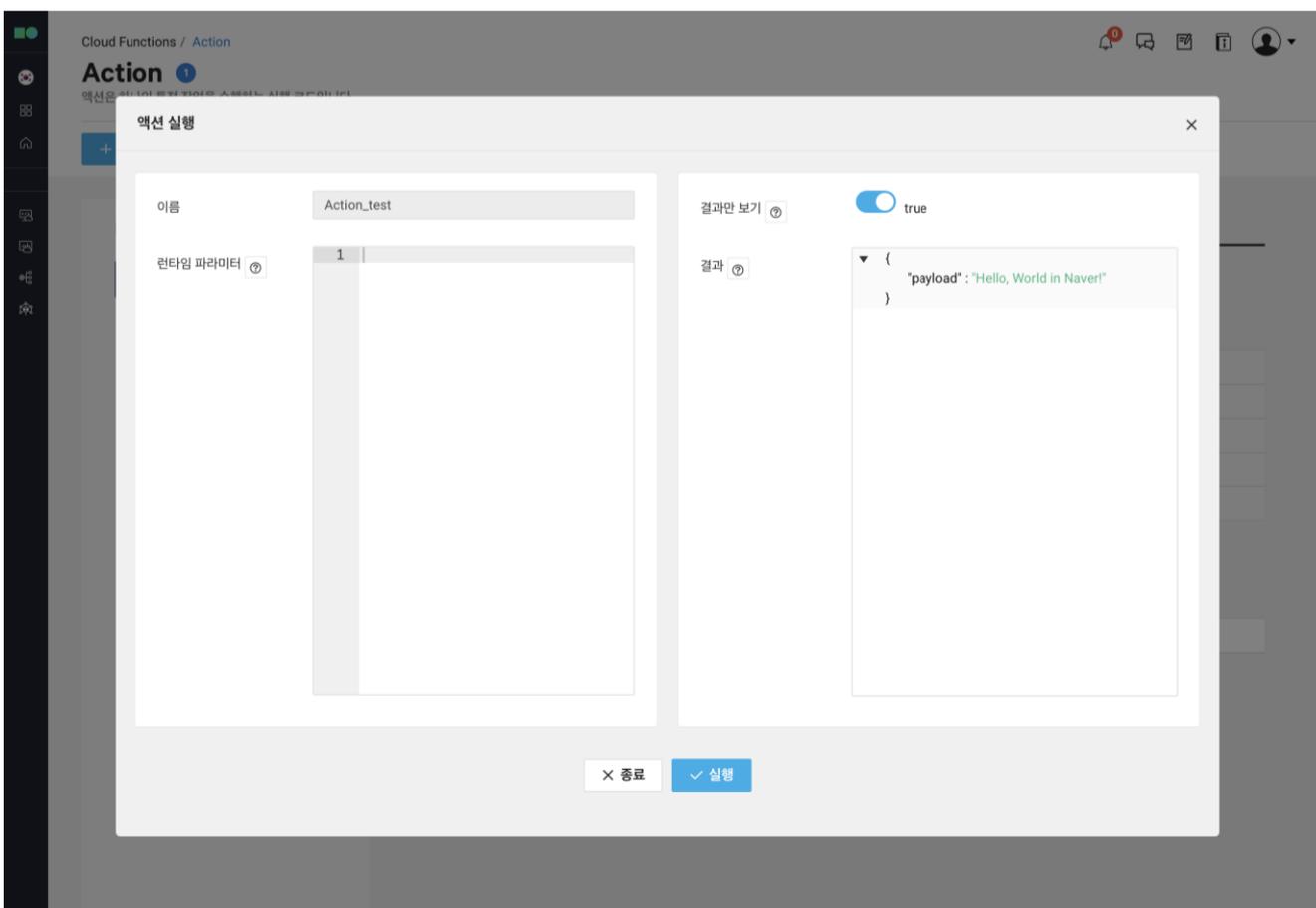
UI 콘솔을 이용한 코드 작성의 예



- 액션, 트리거는 공통적으로 Entity 라는 명칭으로 불립니다. Entity 이름은 중복될 수 없기 때문에 고유한 이름을 작성해야 합니다.
- 패키지를 선택하지 않으면 기본 패키지가 없는 상태로 생성됩니다. 패키지를 만들어 액션을 패키지에 포함시키는 것을 권장합니다.
- 코드는 웹에서 바로 작성하거나 ZIP(Java 는 JAR) 형식의 파일을 업로드하여 등록할 수 있습니다.
- ZIP(JAR) 파일로 등록하는 코드는 작성 방법이 정해져 있으니 반드시 작성 가이드를 참고하여 작성해야 합니다.
- 액션의 옵션에 대한 자세한 내용은 "[Package, Action 사용하기](#)"에서 설명합니다.

Step 4. 액션 실행하기

생성한 액션을 선택한 후 실행하기 버튼을 클릭하여 액션을 실행합니다.



URL 을 통한 액션 실행에서 **Cloud Functions** 의 액션은 기본적으로는 비동기 방식으로 실행되지만 `--blocking` 옵션을 사용하여 동기 방식으로 동작하게 할 수 있습니다.

`--result` 는 액션의 최종 실행 결과만을 확인하는 옵션입니다.

액션의 옵션에 대한 자세한 내용은 "[Package, Action 사용하기](#)"에서 설명합니다.

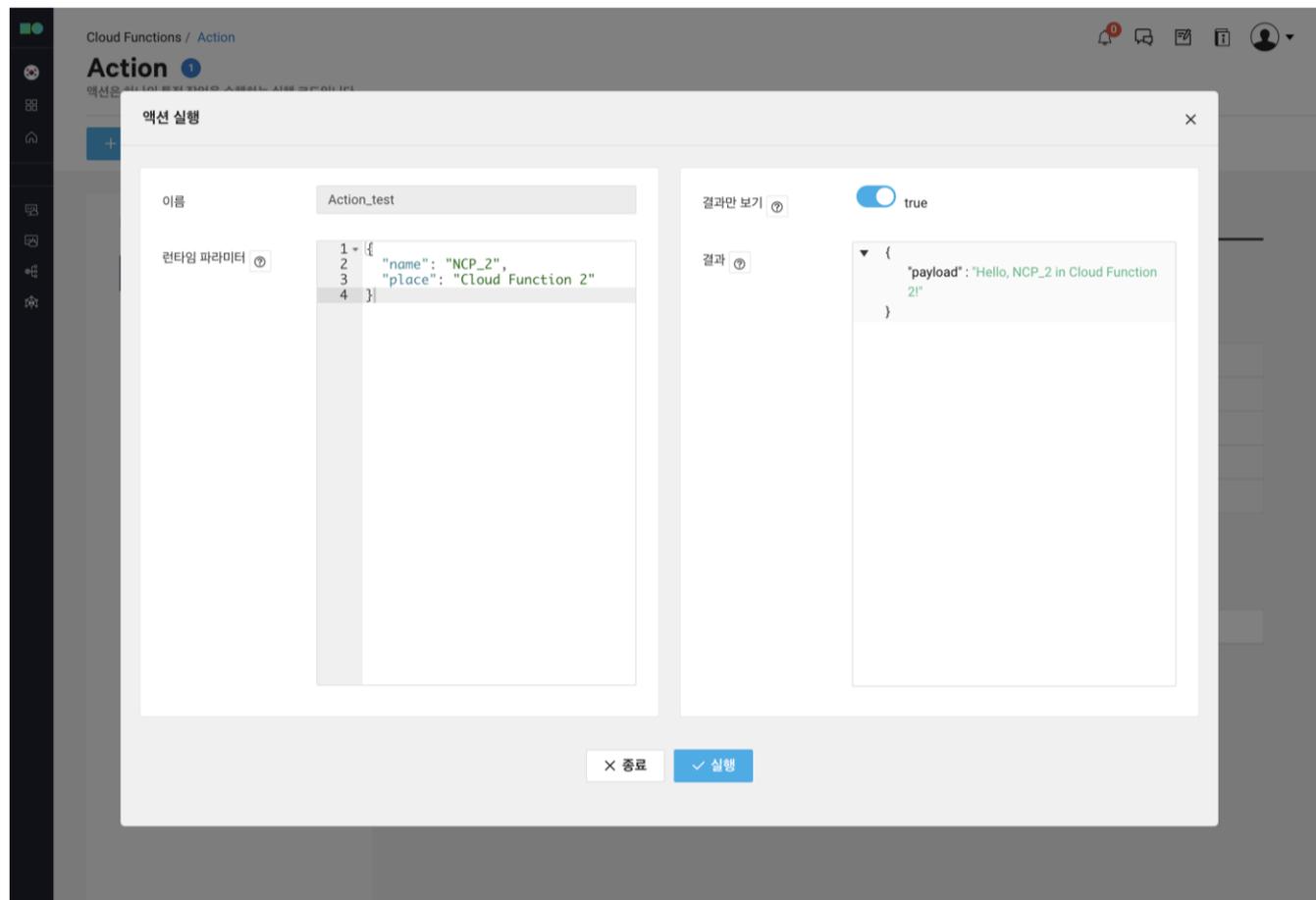
Step 5. 파라미터를 전달하여 액션 실행하기

실행 시점에 파라미터를 전달하여 액션을 실행할 수 있습니다.

디폴트 파라미터가 아닌 파라미터를 전달하여 액션을 실행합니다.

```
{
  "name": "NCP_2",
  "place": "Cloud Function 2"
}
```

UI 콘솔을 이용해 파라미터를 전달해서 액션을 실행하는 예



Step 6. 상세 실행 결과 확인

실행된 결과는 각 Action 이나 Trigger 의 모니터링 탭에서 확인 가능합니다. 실행의 성공, 실패와 걸린시간, 그리고 Log 등의 정보를 볼 수 있습니다.

- 실행 결과는 결과 내 문자열이나 ActivationId 필드 값 등을 통해 조회 가능하며, 이에대한 자세한 내용은 [Action 실행하기](#)에서 추가로 다루도록 합니다.

UI 콘솔을 이용한 상세 결과 확인

No.	ID	Status	Start	Duration	자세히 보기
1	j7CKbFTPBDWP/93aa3a54b3074d2f21	success	2019-05-02 17:50:05 (UTC+09:00)	44 ms	보기
2	j7CKbFTPBDWP/9953683123304b7f93683123301b7f12	success	2019-05-02 17:49:59 (UTC+09:00)	31 ms	보기
3	j7CKbFTPBDWP/9bd33fc49d68405d933fc49d68205dc6	success	2019-05-02 17:48:08 (UTC+09:00)	33 ms	보기

연관 정보 바로가기

아래 가이드에서 연관 정보를 확인할 수 있습니다.

- [Package, Action 사용 가이드](#)
- [Node.js 6, 8 Action 생성 가이드](#)
- [Python 3 Action 생성 가이드](#)
- [Java 8 Action 생성 가이드](#)
- [Swift 3 Action 생성 가이드](#)
- [PHP 7 Action 생성 가이드](#)
- [.Net Action 생성 가이드](#)
- [Go Action 생성 가이드](#)
- [Trigger 사용 가이드](#)
- [Action 실행하기](#)