제미나이Gemini와 구글 클라우드Google Cloud로 무료로 뉴스레터 발송 자동화 기능을 만드는 여정!

데이오프

2026/08/25 01:34 http://blog.naver.com/spfnell2579/224389125311

요즘 다들 하는 AI.

나보다 기계치인 사람도 AI로 하루 뉴스를 정리해서 매일 아침 본인 이메일로 받아보고 있다는 얘기를 듣고 충격을 먹었다.

그래서 나도 한번 뉴스레터 발송 자동화 기능을 만들어보기로 했다.

하지만 나는 가난한 상놈.

최대한 무료로 만들어보기로 했다.

그리고 나는 쫄보이기도 하기 때문에 새로운 사이트에 회원가입 하는 방법도 피해보도록 했다.

이왕 AI를 사용하는 연습이니 제미나이와 함께 하였다.

사실 내 취향은 클로드에 조금 더 기울어져 있기에 클로드로 만들어 보려고 시도했으나

메일 자동 발송 기능 등은 클로드 유료 플랜을 구독해야만 제공이 되는 것이여서 포기했다.

난 가난하다고 말했잖아.

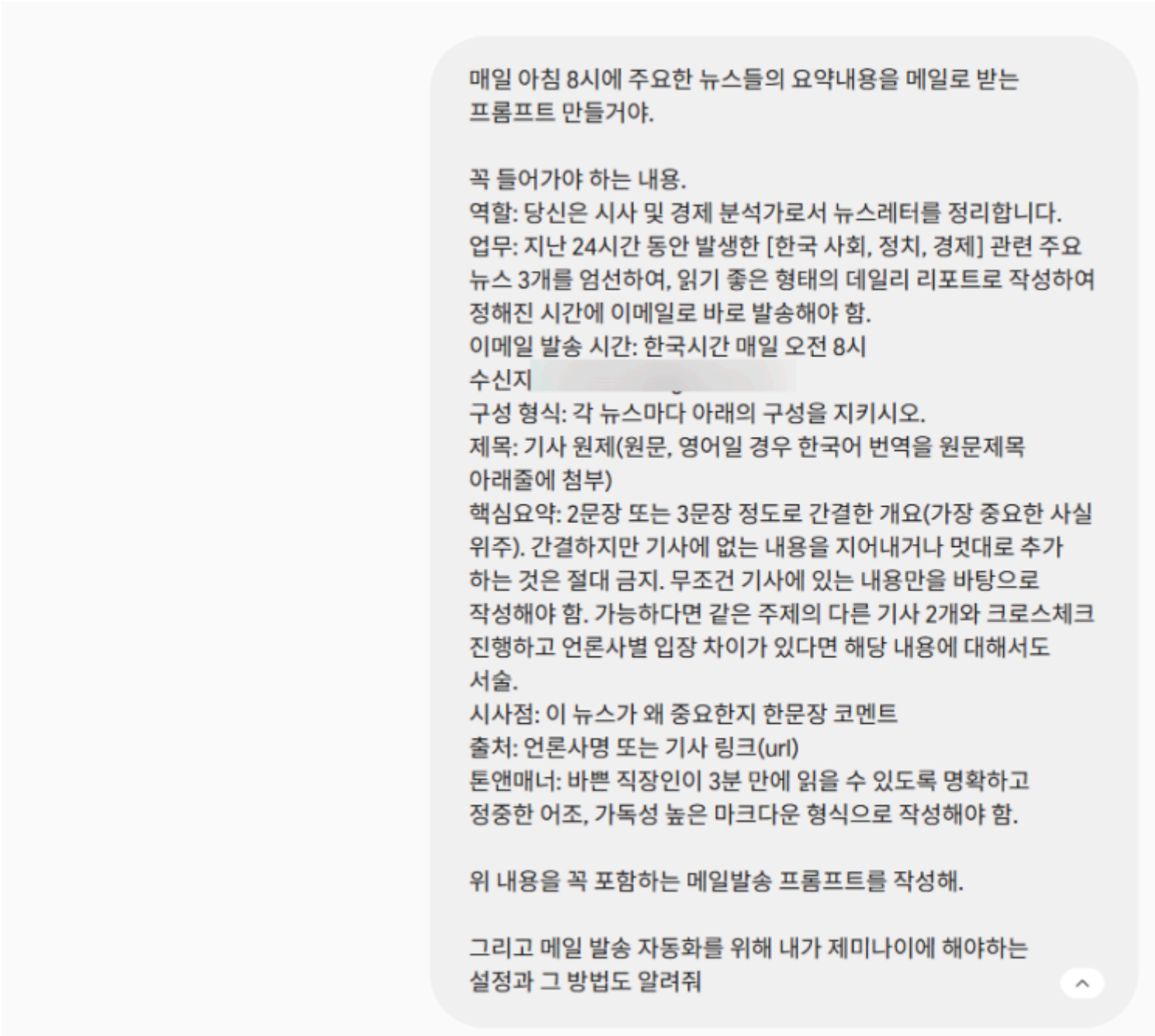
참고사항

- 예약 작업은 원격(클라우드)에서 실행되기 때문에 컴퓨터가 꺼져 있거나 Claude Desktop 앱이 닫혀 있어도 정해진 주기대로 실행돼요. Claude
- 무료 플랜에서는 Cowork 자체를 이용할 수 없어서 예약 작업 기능도 사용할 수 없어요.
- 화면 구성은 최근 변경된 지 얼마 안 된 기능이라 앱 버전이나 플랜에 따라 메뉴 명칭·위치가 조금 다를 수 있어요. 만약 위 경로에서도 안 보이시면, 현재 이용 중인 플랜(Free/Pro/Max/Team)을 확인해 보시는 게 좋아요.



너무한 내친구 클로드

그래서 플랫폼을 옮겨 제미나이에게 별도의 회원가입 없이 무료로 메일자동발송 기능을 만드는 과정을 요청했더니 제미나이와 구글 클라우드를 설정하여 만드는 방법을 알려주었다.
(사실 이 방법도 결제카드등록이 필요하긴 한데 추가결제가 일어나진 않으니 조아쓰)
(카드등록도 피하고싶다 하면 Google Apps Script를 활용하며 만드는 방법도 있다. 이 내용은 내일 하겠다)



이것이 오늘의 내 목표

필요한 툴

Cloud Run Functions: 호출을 받으면 Gemini API를 불러 뉴스를 작성하고, Gmail API로 메일 발송
Cloud Scheduler: 매일 오전 8시마다 Cloud Functions의 URL을 호출(스위치 ON)

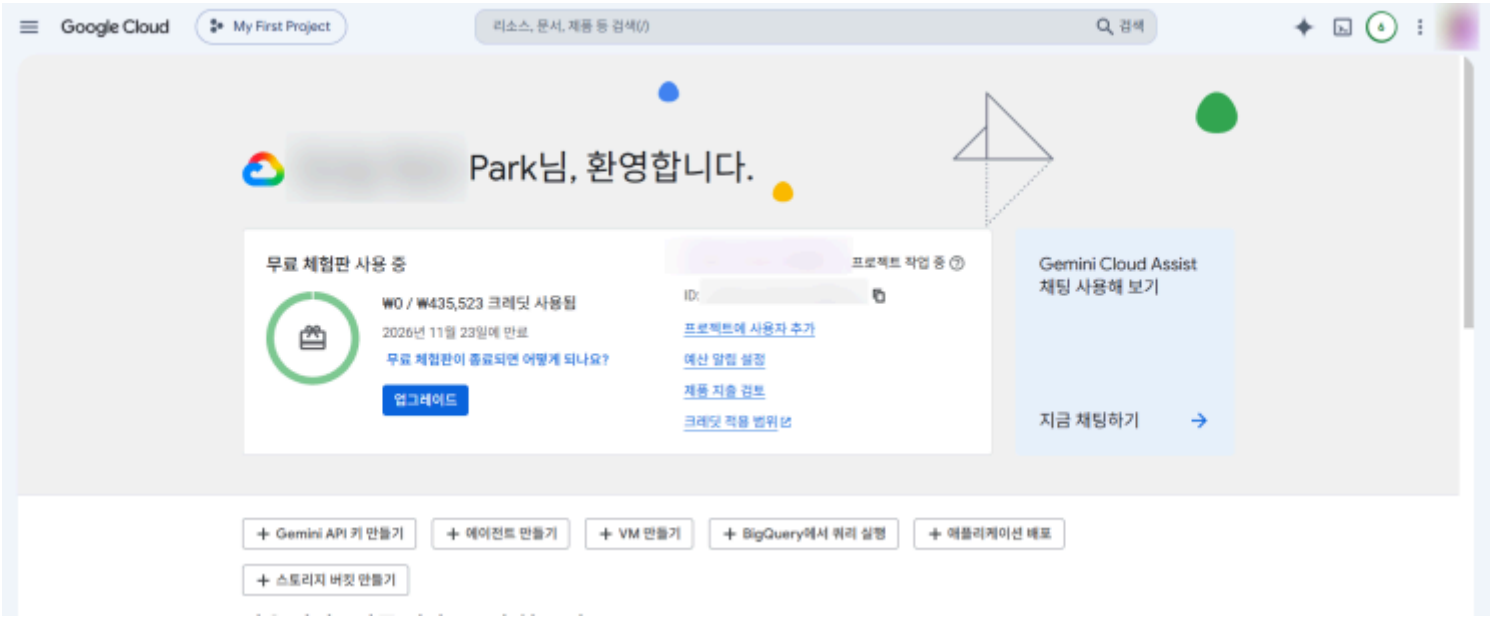
Cloud Scheduler는 스위치를 눌러주는 알람 시계' 역할만 하기 때문에, 알람을 받아 뉴스를 정리하고 메일을 발송할 Cloud Run Functions(코드 실행기)와 함께 세트로 연동이 필요했다.

작업 순서
1. Google Cloud Console 접속 및 API 활성화
2. Cloud Run Functions(메일 발송 로직) 만들기
3. Cloud Scheduler(오전 8시 알람) 설정
순으로 작업하였다

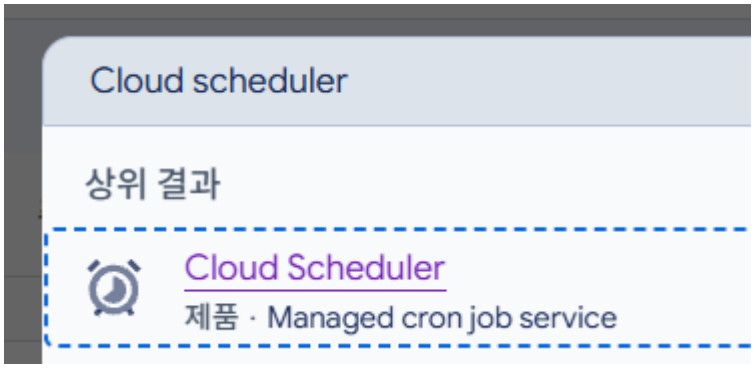
---

1. Google Cloud Console 접속 및 API 활성화

먼저 Google Cloude Console에 접속하여 나의 구글 계정을 가지고 연동하여 접속하였다.



그리고 위 검색창에서 Cloud Scheduler와 Cloud Run Functions API, 그리고 Gmail API을 각각 검색하여 '사용'을 눌러 내 계정에 추가해주었다.
'사용'버튼을 누르고 연결되기까지 약간의 시간이 걸리니 당황하지 말자.

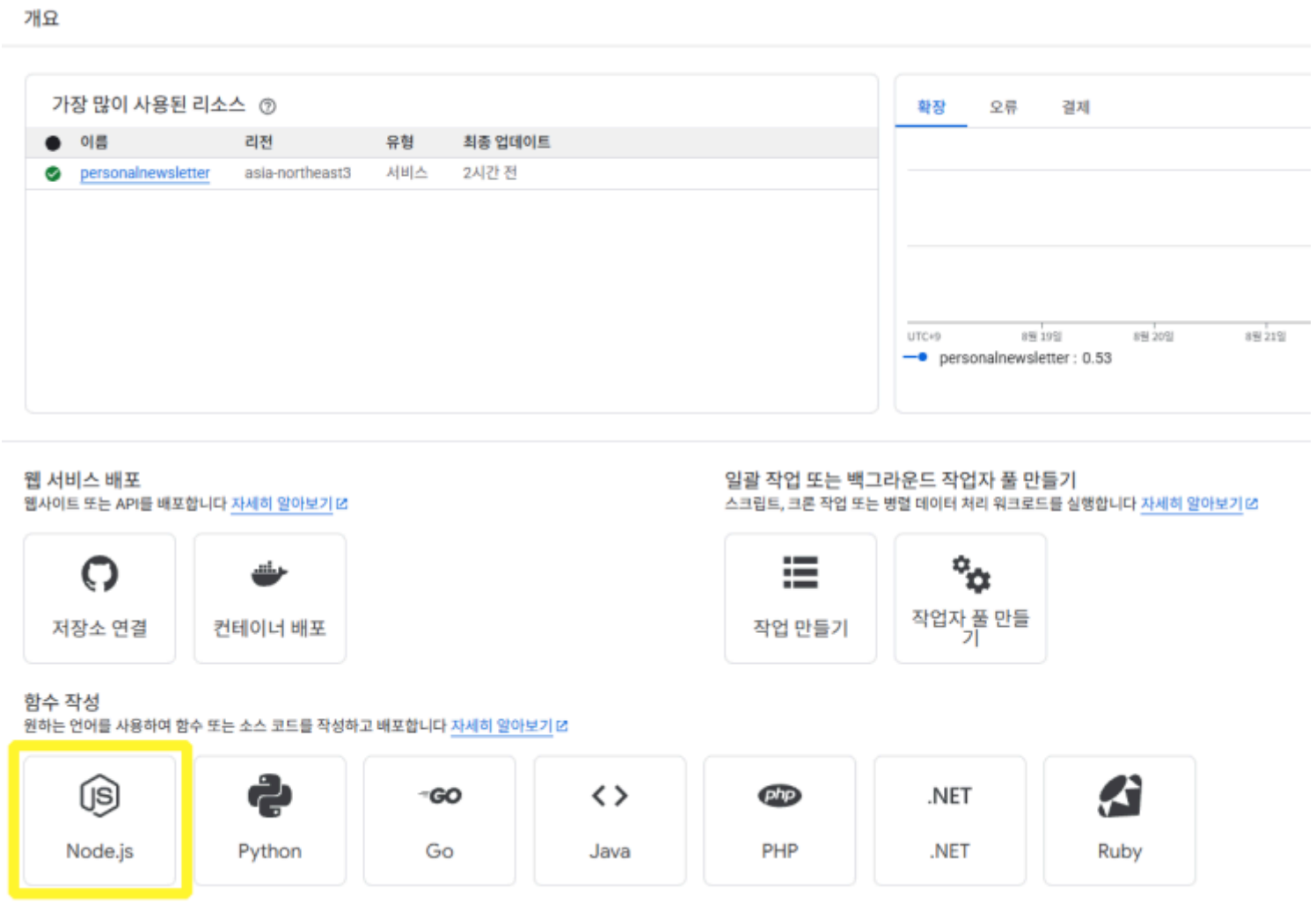


이 과정에서 구글클라우드가 결제정보를 강요한다. 이걸 안하면 사용에 제한이 생기는 듯 하다.
또한 구글클라우드는 구독제가 아닌 사용한만큼 돈을 내는 후불제 구조이다. 내가 하려는 것은 매월 제공하는 무료 용량을 넘어서지 않으니 믿고 진행했다.

---

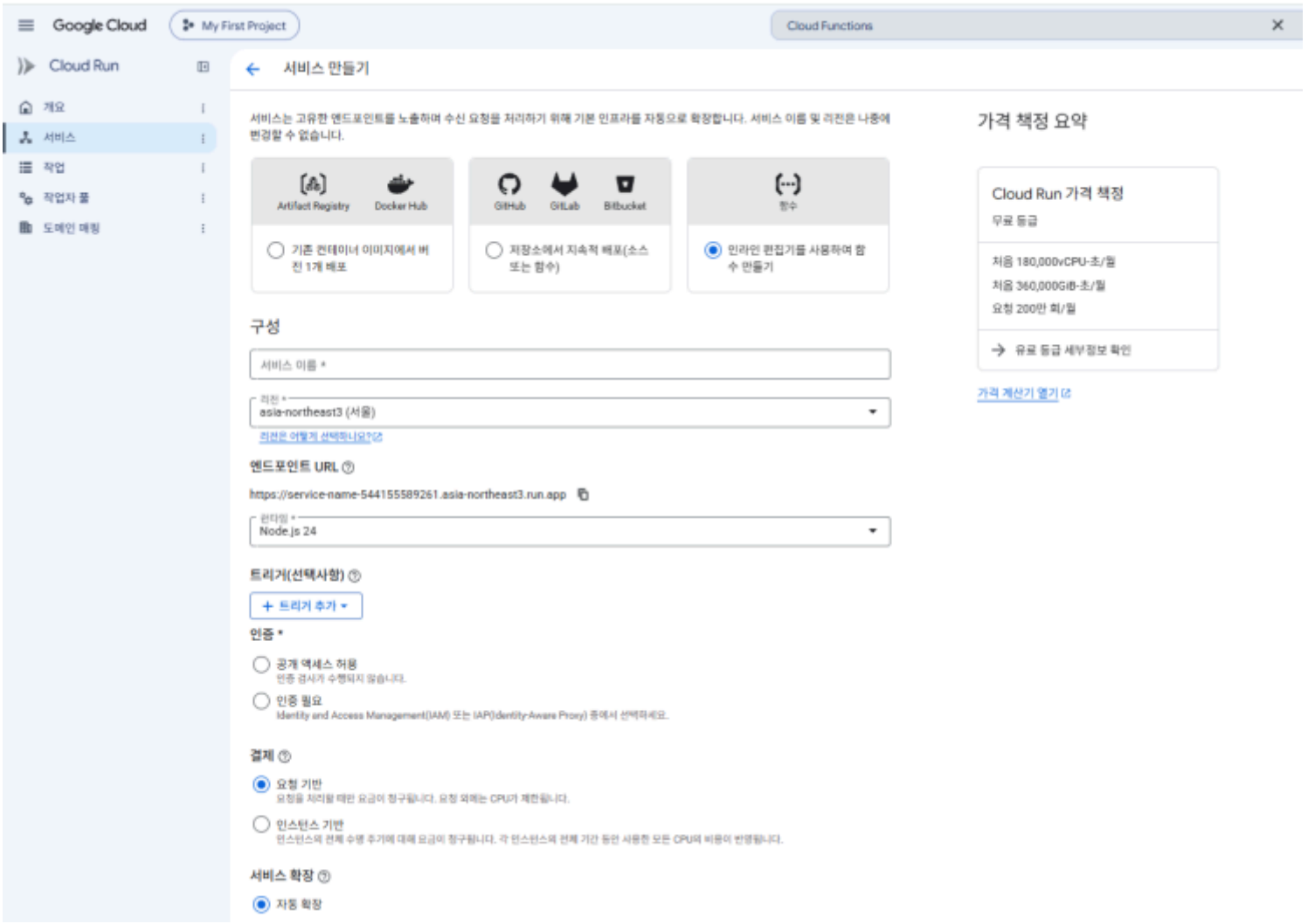
2. Cloud Run Functions(메일 발송 로직) 만들기

모두 추가가 되었으면 그 다음엔 방금 추가 한 Cloud Run Functions에 들어가 메일 발송 로직을 만들었다.



여기서 [함수 작성]메뉴의 Node.js를 언어로 선택하면 소스 작성 전 설정을 할 수 있는 페이지가 나온다.

여기서 첫번째 현기증을 느꼈다 와우.



대충 뭐...이런게 나오는데

본인이 원하는 **서비스이름**을 적고

리전region은 서울을 선택하였다. 아무래도 한국사람이니까.

리전 드롭다운에서 서울 드릅게 안보이는데 드롭다운 상단에 asia 검색하면 아시아지역 기준시간대만 필터하여 보여주니 그렇게 찾으면 쉽다.

트리거는 선택사항이라고 되어 있는데 특별히 설정하지 않으면 자동으로 HTTPS로 작동하는 듯 하다.

결제는 요청기반으로 설정해야 무료요청횟수를 초과했을때만 청구된다.

그 밑으로 있는 [컨테이너, 네트워킹, 보안] 부분이 조금 더 중요하다

여기서는 **변수**를 설정해줘야 하기 때문이다.

이거 설정 잘못하면 아무것도 안되더라.

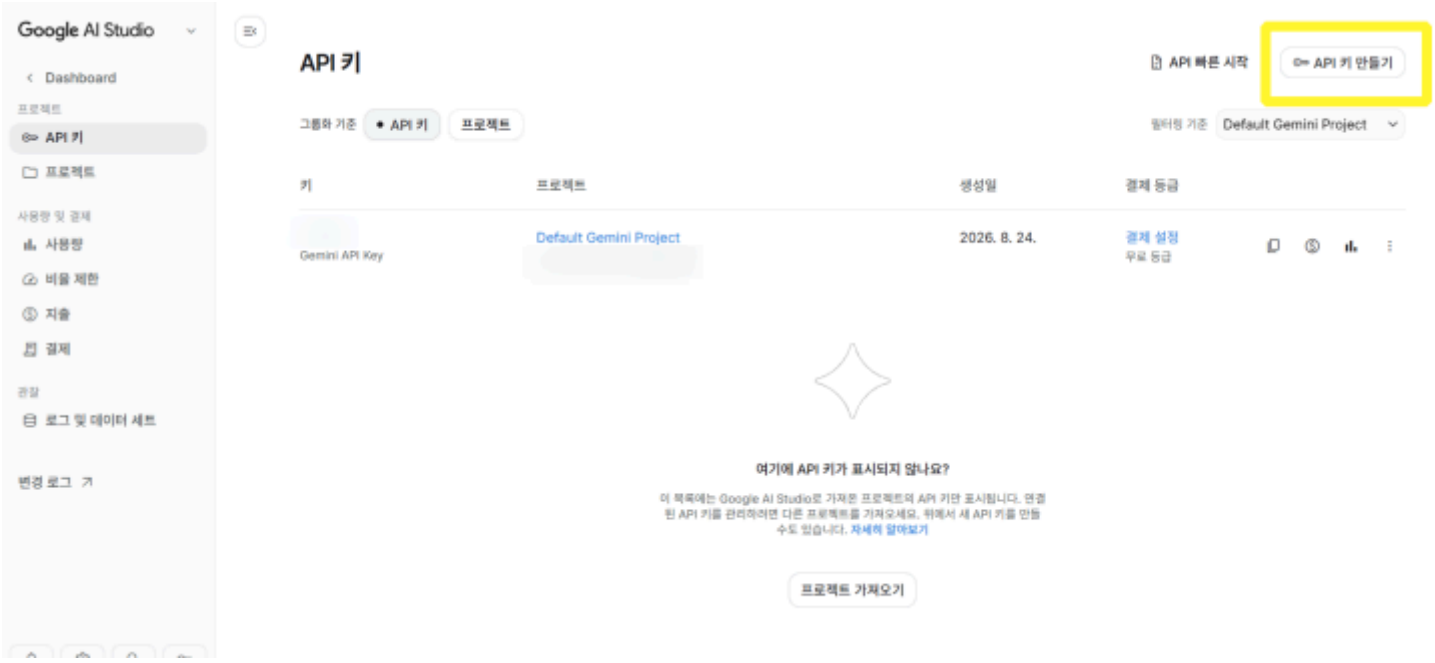


'변수 및 보안 비밀'탭으로 가서 '변수 추가'를 눌러 아래 내용을 입력 한다.

이름 (Name)	값 (Value)
GEMINI_API_KEY	Google AI Studio에서 발급받은 Gemini API 키
GMAIL_USER	본인의 Gmail 주소 (예:)
GMAIL_APP_PASSWORD	구글 계정 보안에서 발급받은 **16자리 앱 비밀번호**
...	

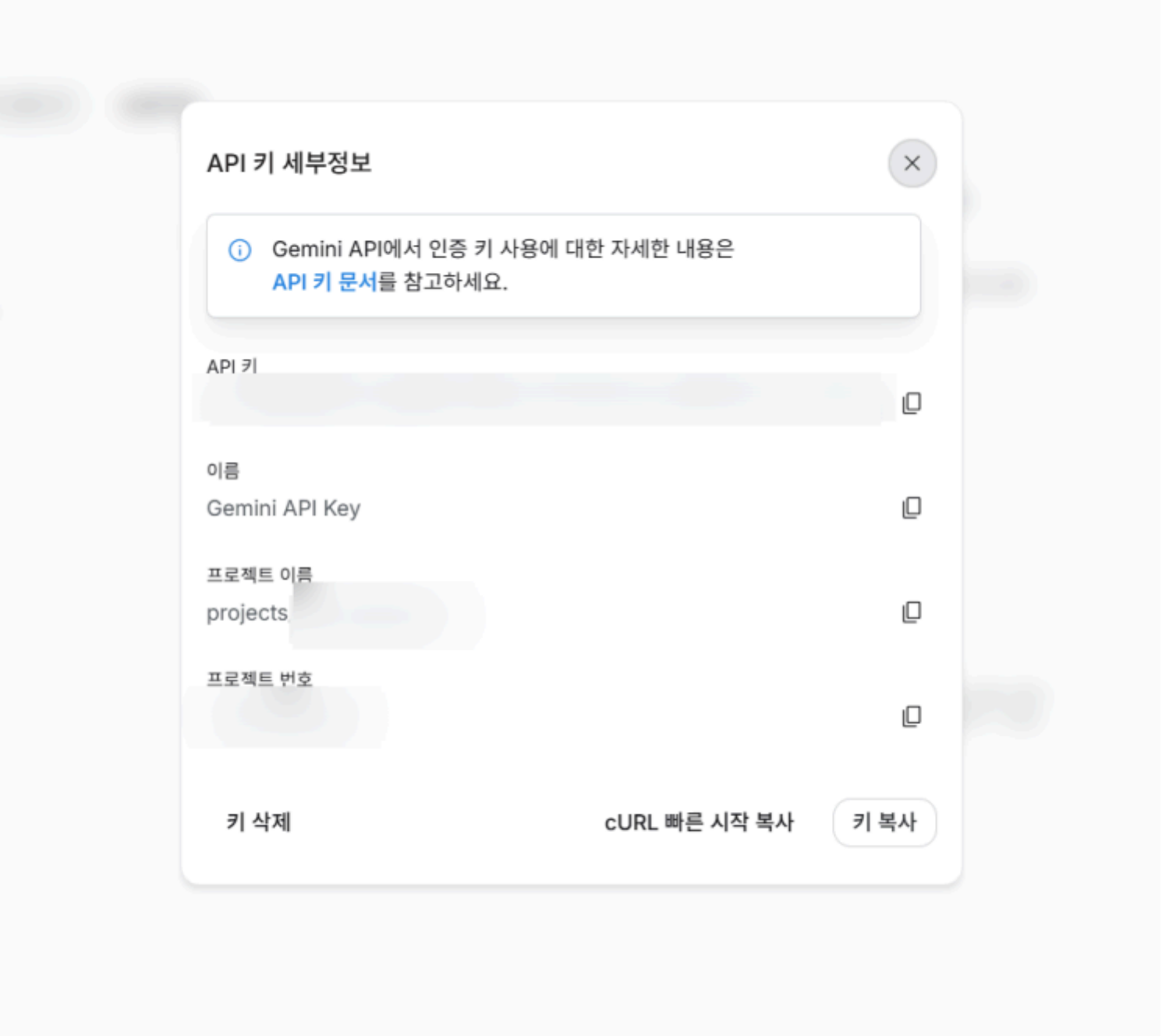
변수의 이름은 다르게 설정해도 된다. 아래에 작성할 js에서 변수를 호출할때와 스펠링만 일치하면 된다.<이것이 중요

Gemini API 키는 Google AI Studio에서 발급받아야 한다.



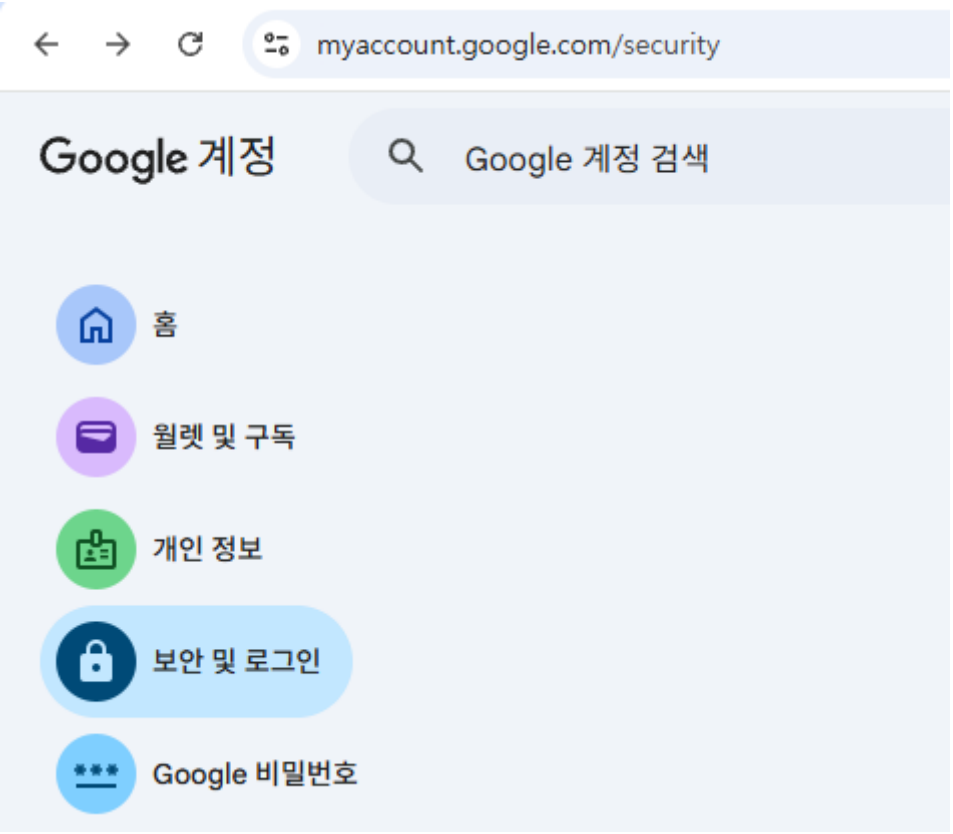
우측 상단의 'API 키 만들기'를 눌러 새 키를 만든다.

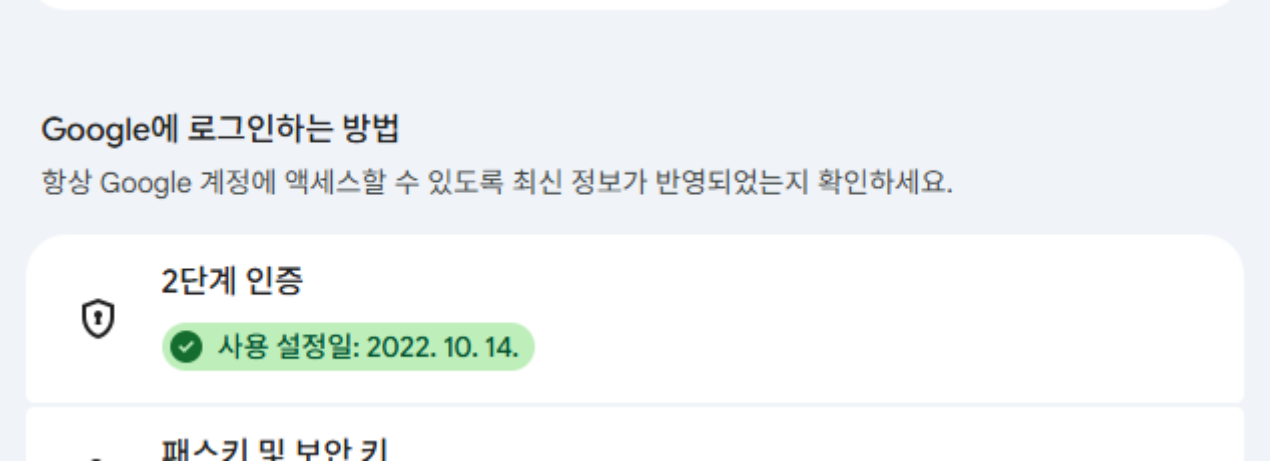

새 키 만들기
키 이름 지정
Gemini API Key 2
가져온 프로젝트 선택
Default Gemini Project
취소
키 만들기



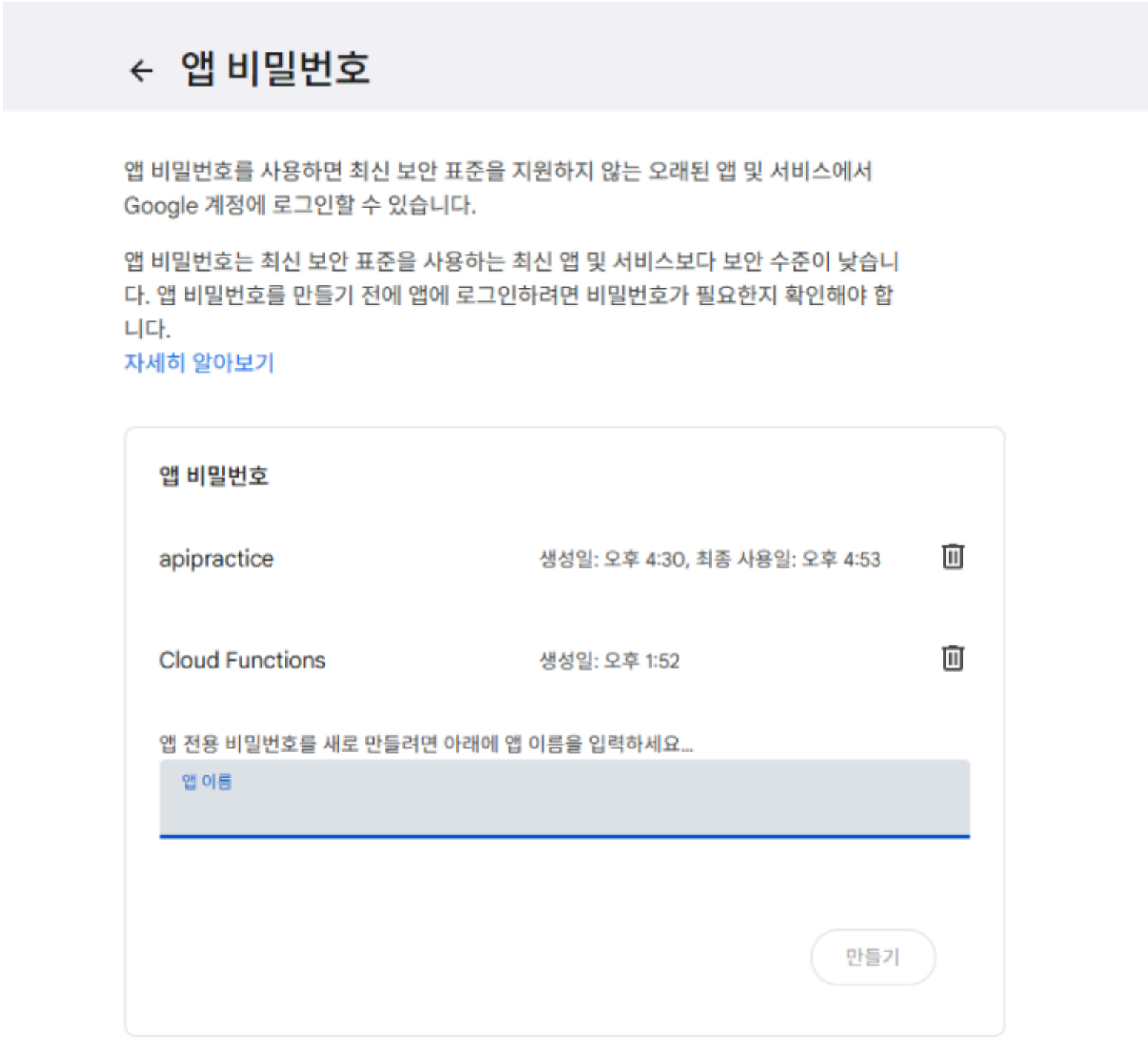
키 이름도 기본으로 정해주는게 있어서 별다른 수정 없이 그냥 '키 만들기'를 누르면 생성이 된다.
저 길고 긴 난수를 복사하여 GEMINI_API_KEY의 값으로 지정한다.

그리고 이어서 **구글 앱 비밀번호**도 발급받아야 한다.
생소한 녀석이였는데 구글 계정에서 손쉽게 무료로 발급받을 수 있다.
먼저 구글 계정에 들어가 2단계 인증이 활성화 되어있는지 확인해준다.
2단계 인증이 되어있어야만 앱 비밀번호를 받을 수 있다. 무조건 필수!





확인이 됐다면 [앱 비밀번호(myaccount.google.com/apppasswords)]로 들어간다.



회색 칸에 앱 이름(자유롭게 가능)을 입력하고 '만들기'버튼을 누른다.

그러면 16자리 코드가 생성이 되는데

이 코드는 한번 창을 닫으면 다시 확인 할 수 없으니 꼭 메모장이나 다른곳에 복사해 저장해두기로 하자.

그리고 발급받은 16자리 코드는 [변수 및 보안 비밀]탭으로 돌아가서 GMAIL_APP_PASSWORD의 값으로 지정한다.

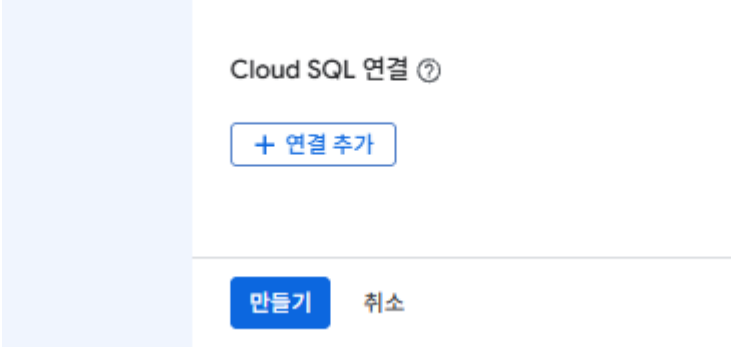
여기서 코드에 띄어쓰기가 있으면 안된다!!! 나는 이 실수를 저질러서 두번이나 수정함

GMAIL_USER에는 자동 메일을 받을 내 메일 주소를 넣어둔다.

이렇게 하면 필요한 변수의 설정이 끝난다.



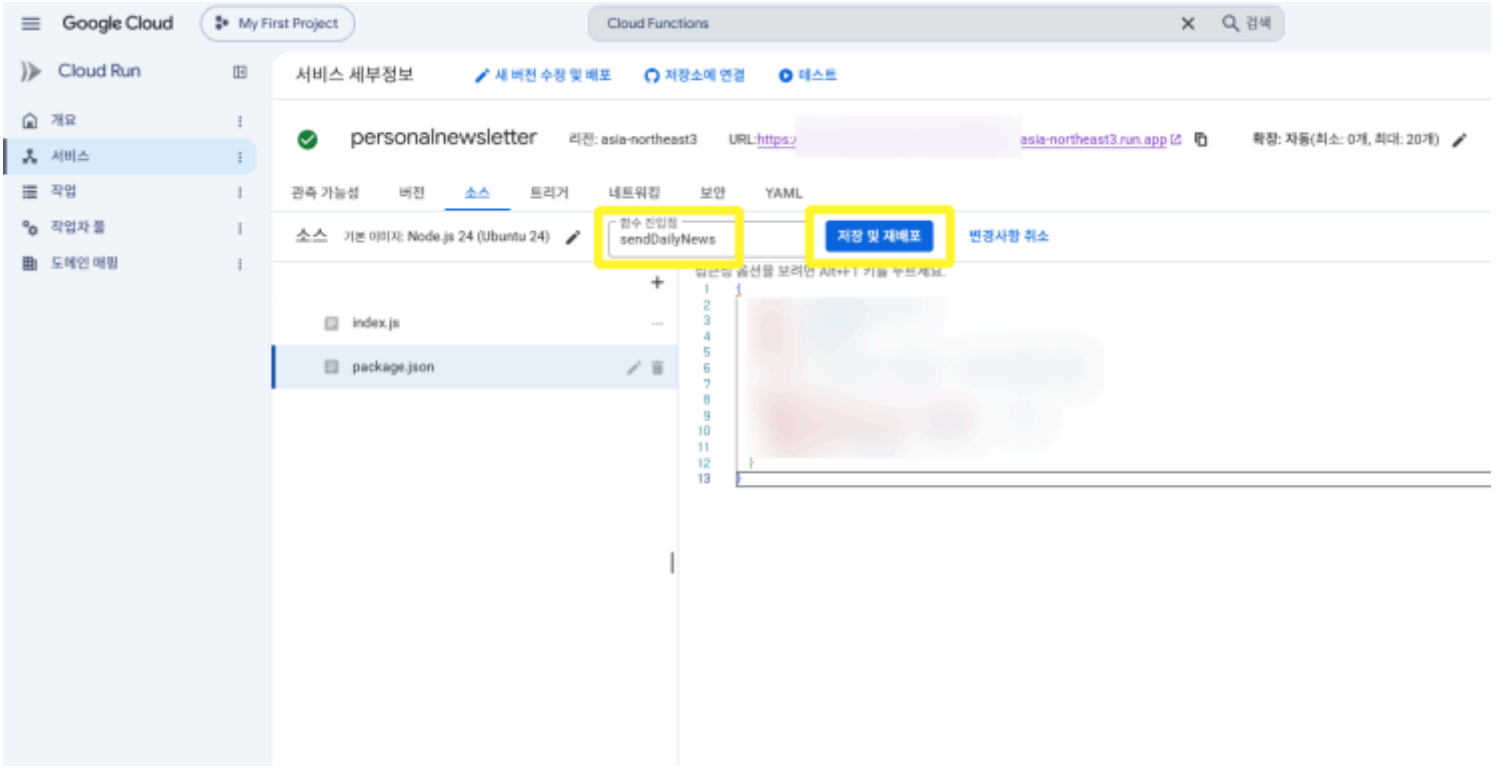
다 입력했으면 하단에 파란버튼 '만들기'를 눌러 코드를 입력하는 창으로 넘어간다.



이제 여기서 코드를 작성하면 되는데

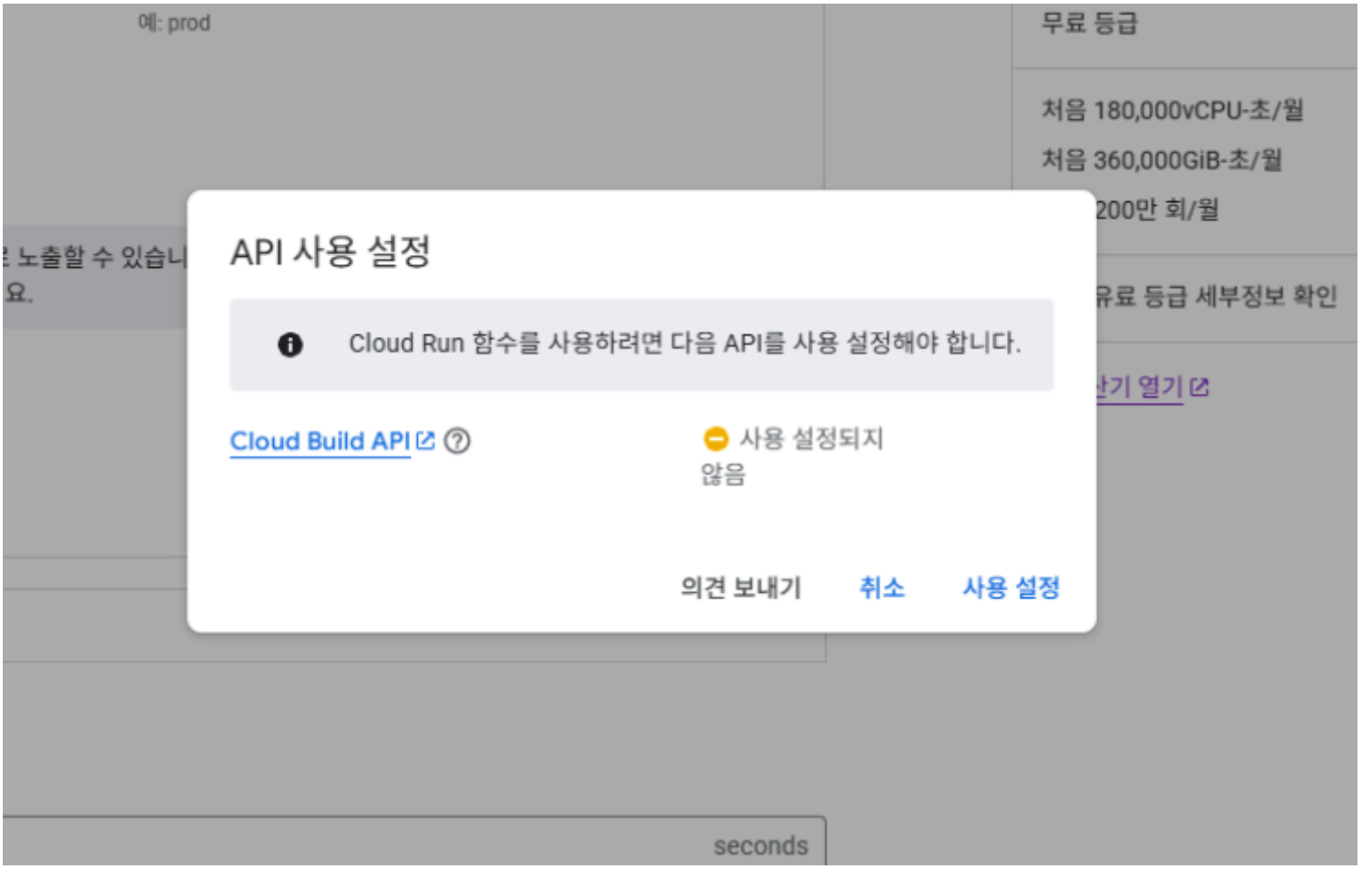
json패키지랑 스크립트는 대충 제미나이 때려서 뽑아냈다.

위의 **함수 진입점**도 꼭 수정해주는거 잊지 말아야 한다. 이것이 코드에 있는것과 일치하지 않으면 오류가 발생한다.

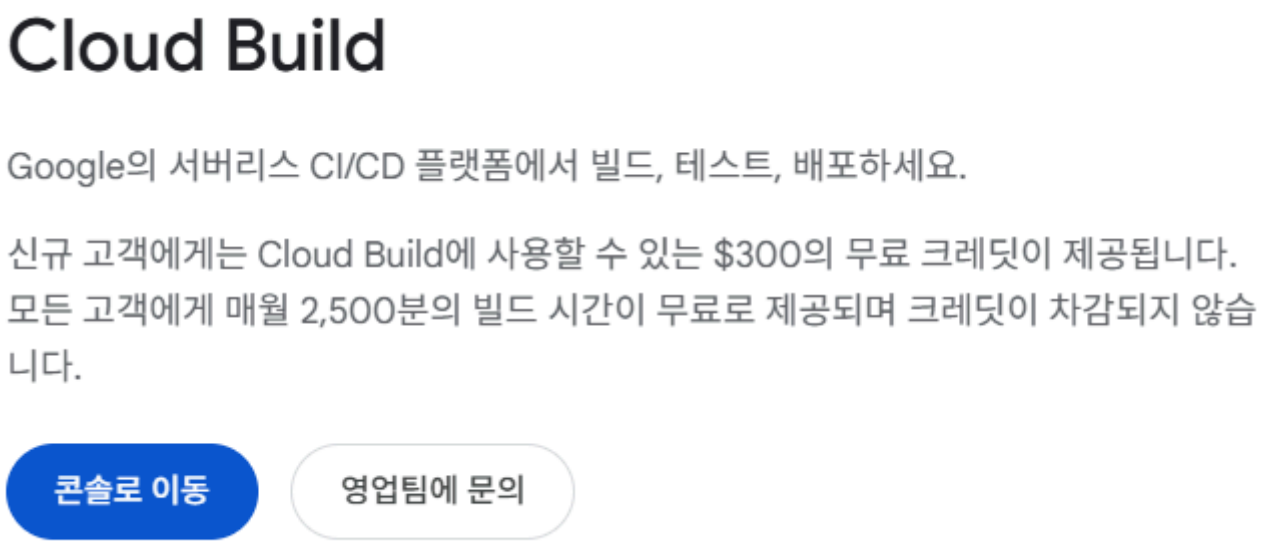


코드를 다 작성하였다면 '저장 및 재배포'를 눌러 빌드한다.

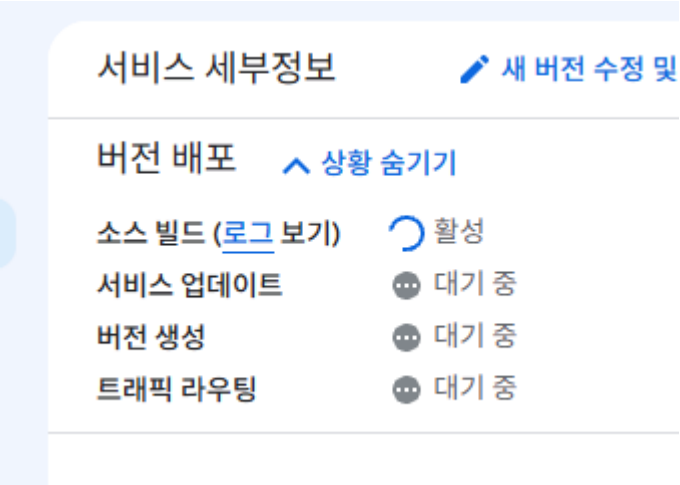
빌드가...되어야 하는데? 시작도 하기 전에 안내창이 떴다.



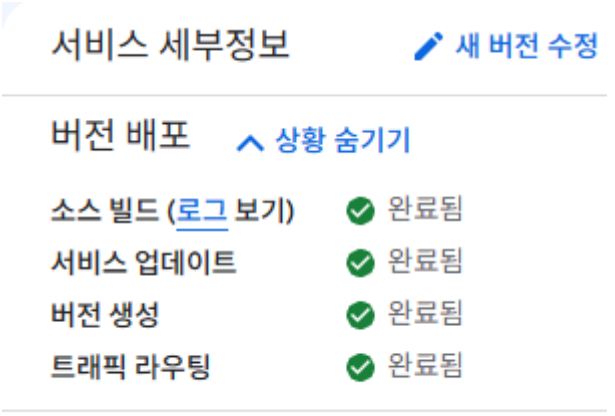
아놔 이런건 빠따빠따 알려달라고.



Cloud Build는 매월 2,500원의 무료 빌드시간을 제공한다.
이것도 이정도 코딩 가지고는 추가금이 드는것은 아니니 사용 설정을 해준다.



그러고 다시 빌드하면 로딩표시가 돌아가고는 길어도 몇분 내로 빌드가 완료된다.



완료됨 확인.

---

3. Cloud Scheduler(오전 8시 알람) 설정

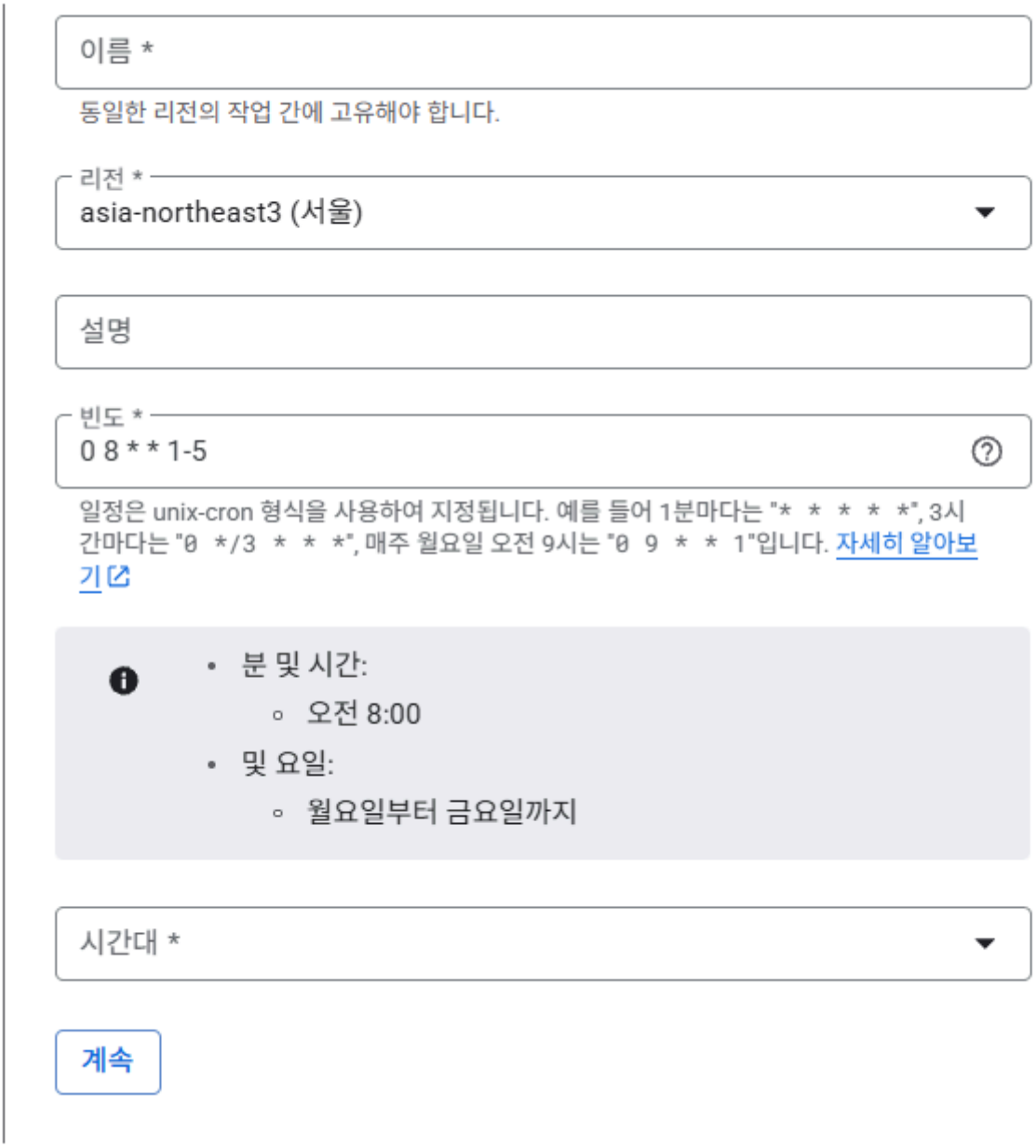
서비스 생성이 모두 완료되었으면 만들어진 URL을 가지고 Cloud Scheduler로 이동한다.



'작업만들기'를 눌러서 작업 예약을 시작한다.

먼저 일정을 정해줘야 한다.

- 일정 정의



리전은 동일하게 서울로 선택하고

저 '빈도'가 새로운 부분이였다.

'크론'이라고 하는 표현식인데 각 자리는 순서대로 분, 시, 일, 월, 요일을 나타낸다.

자세한 내용은 공식에서 제공하는 설명서를 참고하는 것이 더 정확하고 빠른 이해에 도움이 될것이라 생각한다.

[구글 클라우드가 제공하는 크론 표현식 가이드]



크론 작업 형식 및 시간대 | Cloud Scheduler | Google Cloud Documentation

Google은 AI 기술을 사용하여 콘텐츠를 사용자의 기본 언어로 번역합니다. AI 번역에는 오류가 있을 수 있습니다. Home chevron_right Documentation chevron_right Application development chevron_right Cloud Scheduler chevron_right 가이드 크론 작업 형식 및 시간대 크론 작업은 반복되는 주기로 예약되며 unix-cron open_in_new 기반 형식으로 지정됩니다. 작업이 하루에 여러 번 실행되거나 특정 날짜와 특정 달에 실행되도록 일정을...

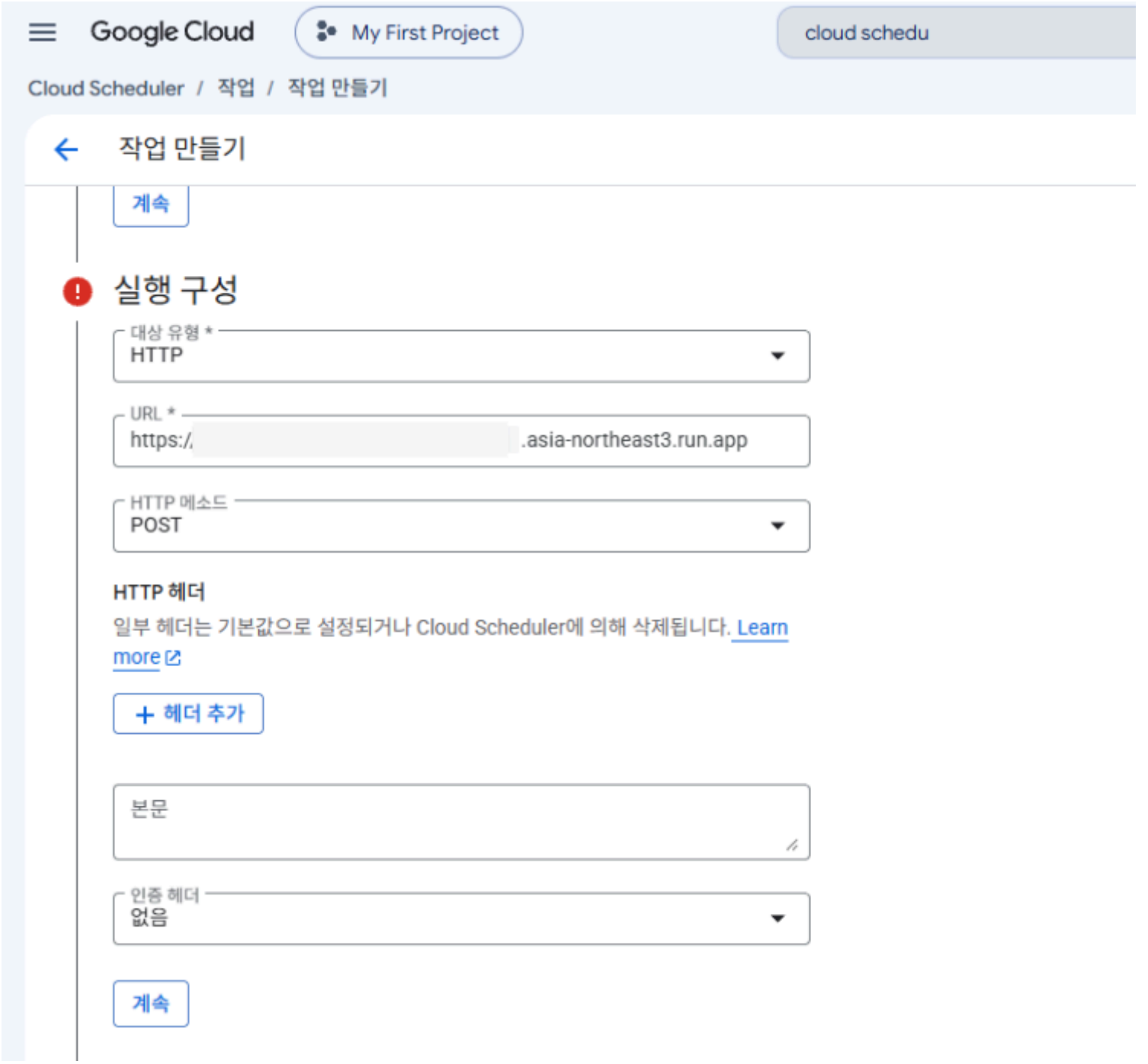
cloud.google.com

나는 매일 오전 8시00분 월요일부터 금요일에 받아보고 싶었기 때문에

0 8 * * 1-5 라고 설정하였다.

각 값 사이에 띄어쓰기는 필수이다.

여기는 이정도만 설정하면 충분하니 '계속'을 눌러 실행 구성 파트로 넘어간다.



개인이 선호하는 설정이 있는 것이 아닌 이상 기본설정에서 바꿀것은 없고

아까 Cloud Run Functions에서 스크립트를 빌드하고 나온 URL을 여기에 붙여넣어준다.

지정한 시간에 호출해야 하는 소스를 알려주는 것이다.

계속을 눌러 '설정 구성'으로 넘어간다.

여기도 개인이 선호하는 설정이 있는 것이 아니면 바꾸지 않고 '만들기'를 눌러 일정을 생성한다.

- 설정 구성(선택사항)

만들기 취소



작업이 다 만들어진 모습

이렇게 되면 메일 발송 자동화를 위한 준비는 모두 되었다. 별 다른 문제가 없다면 예정된 시간이 되었을때 구글이 알아서 내 메일로 메일이 발송될 것이다.

근데 그럼 그렇지
내가 이런걸 한번에 성공할리가 없지.

예정된 시간이 되었는데 오류가 발생했다는 로그가 떴다.



스크립트에서 호출하는 제미나이 버전에 문제가 있다고 한다.
제미나이가 더이상 지원하지 않는 오래된 낮은 버전의 제미나이 모델 버전을 껴서 코드를 작성해준 것이였다.

제미나이한테 제미나이를 호출하는 코드를 짜달랬는데 더이상 안쓰는 버전을 던져주다니
이런 무능한 전자돌쇠놈 같으니.
괘씸하지만 어쩔 수 없으니 그냥 얌전히 버전을 고쳐주었다.
수정 전: const model = genAI.getGenerativeModel({ model: 'gemini-2.5-flash' });
수정 후: const model = genAI.getGenerativeModel({ model: 'gemini-3.6-flash' });

제미나이 모델 버전을 수정하면서 다른 부분도 조금 더 보완을 하였다.

다시 약속된 시간을 기다리고...

드디어!



이런 식으로 메일이 들어왔다!

성공이다!

연습은 뉴스레터로 했지만 뉴스레터는 지루하고 현학적이고 재미없으니
내 흥미를 이끌만한 웹 검색 내용으로 내용을 좀 바꾸어 수정해보려 한다.
예를 들어 매일 한개씩 집밥 레시피를 추천해서 받는다던가.

이런 분위기의 블로그 글을 써보는건 또 처음인데
이거 한다고 아침부터 밤까지 컴퓨터를 붙잡고 있었지만
배운 것도 많고 즐거웠었다. 오랜만에 언어 배우던때로 돌아간 느낌이고ㅎㅎ


굿이노라