Lab1.AWS Lambda Hello World

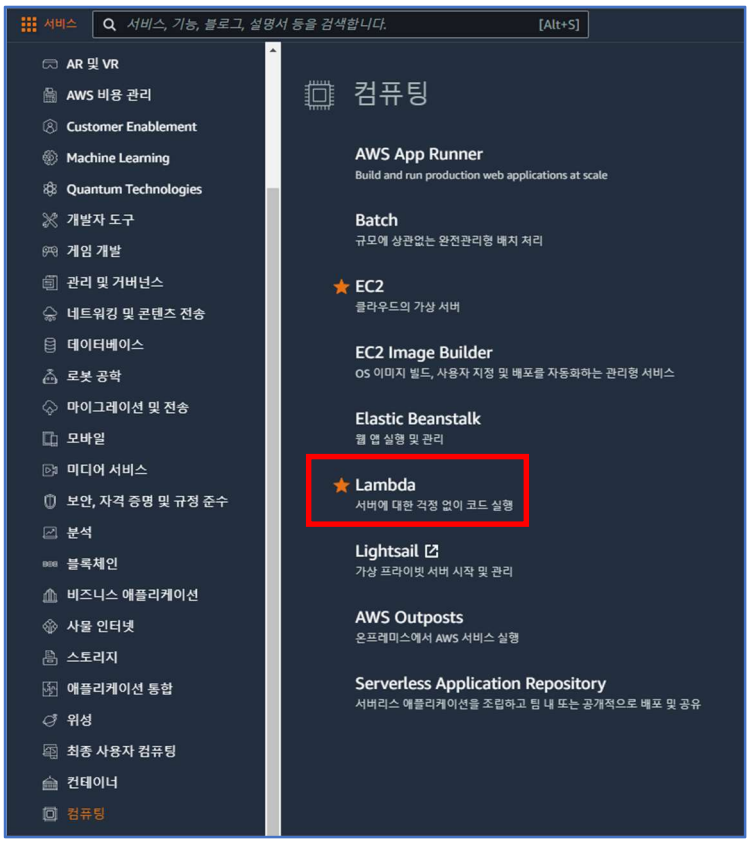
목적

AWS Lambda Function을 이용한 초 간단 실습

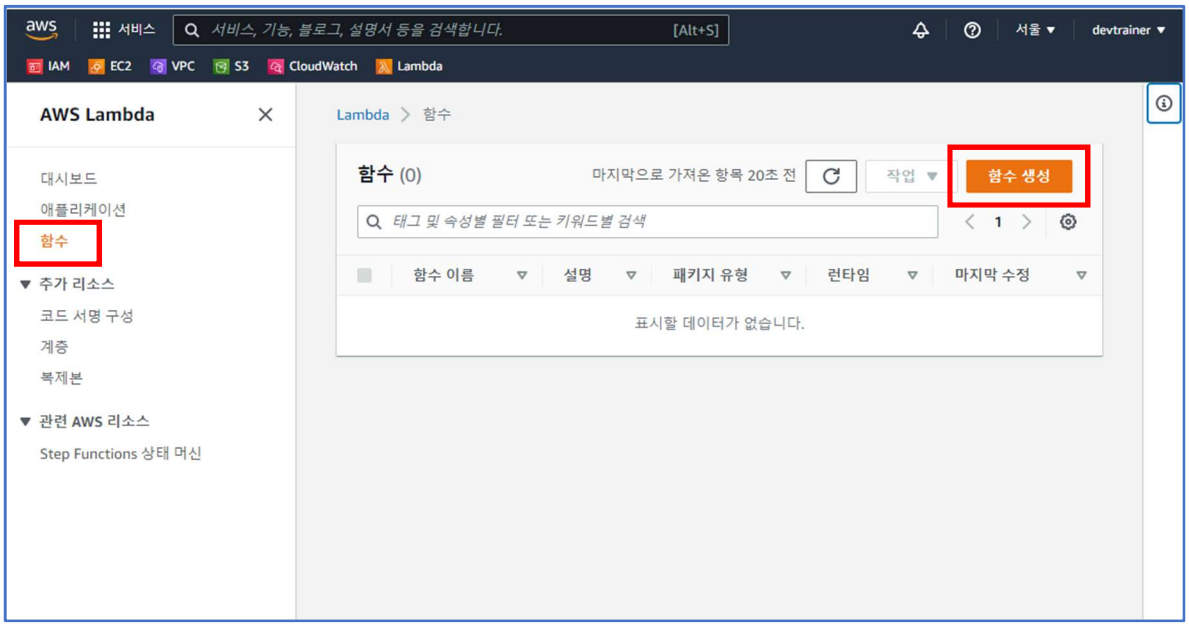
사전 준비물

AWS Free-Tier 계정

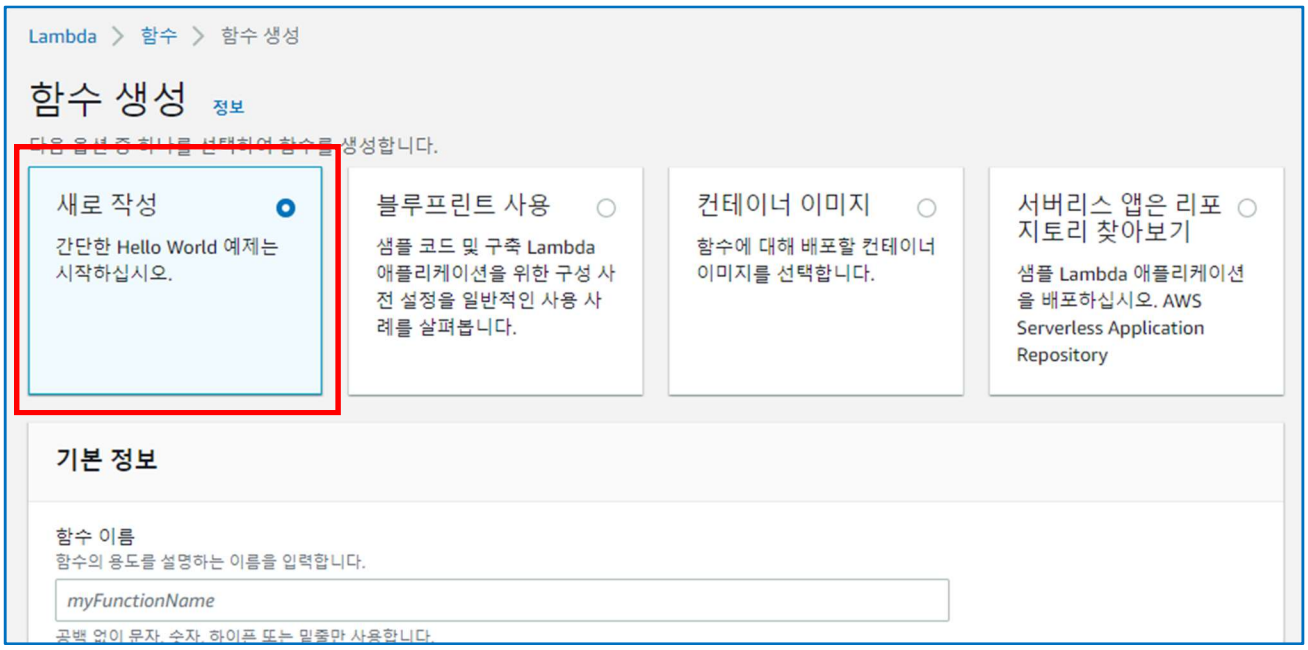
1. 로그인 후, **[서비스]** > **[컴퓨팅]** > **[Lambda]**를 클릭하여 **AWS Lambda** 페이지로 이동한다.



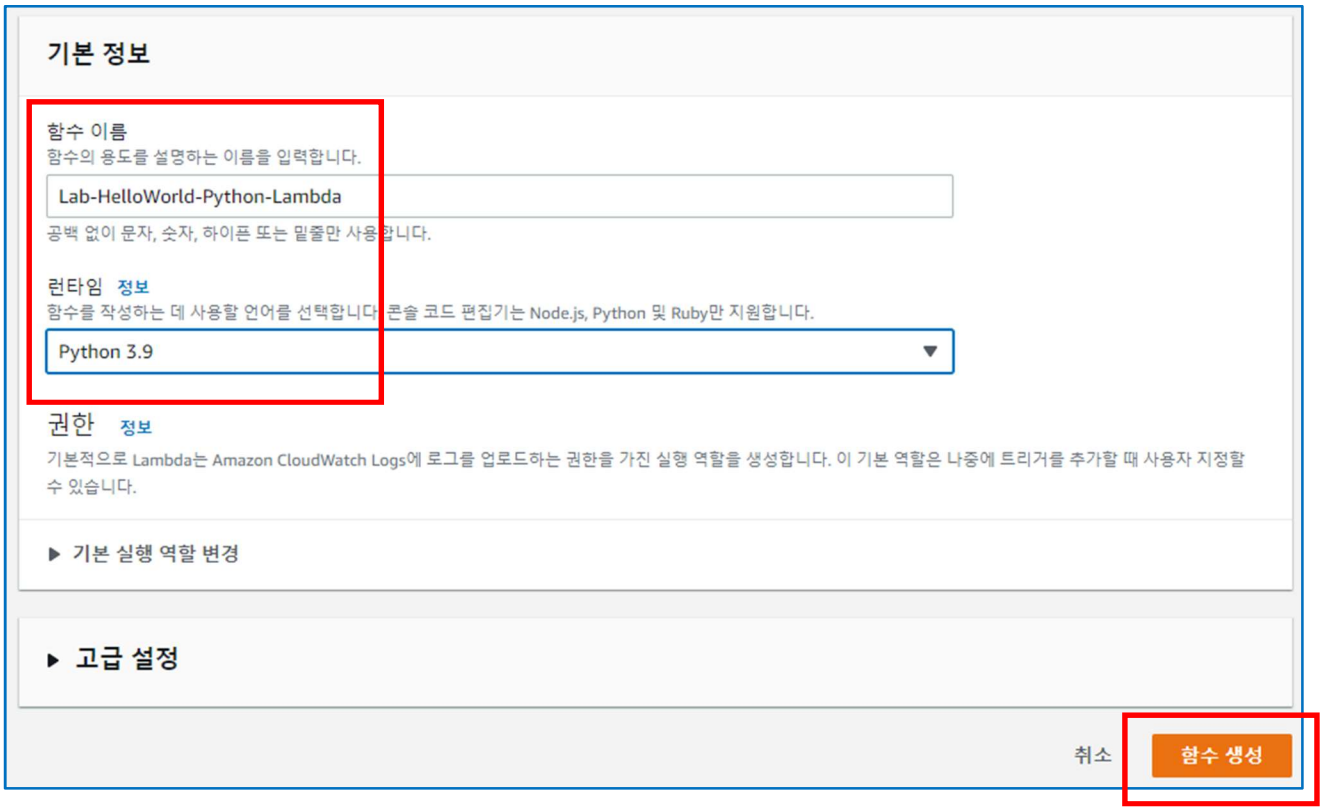
2. **[AWS Lambda]** 페이지에서 왼쪽 메뉴 중 **[함수]**를 선택한다. 반드시 현재 실습은 **서울 Region**에서 수행한다. 페이지 오른쪽 상단의 [**함수 생성]**을 클릭한다.



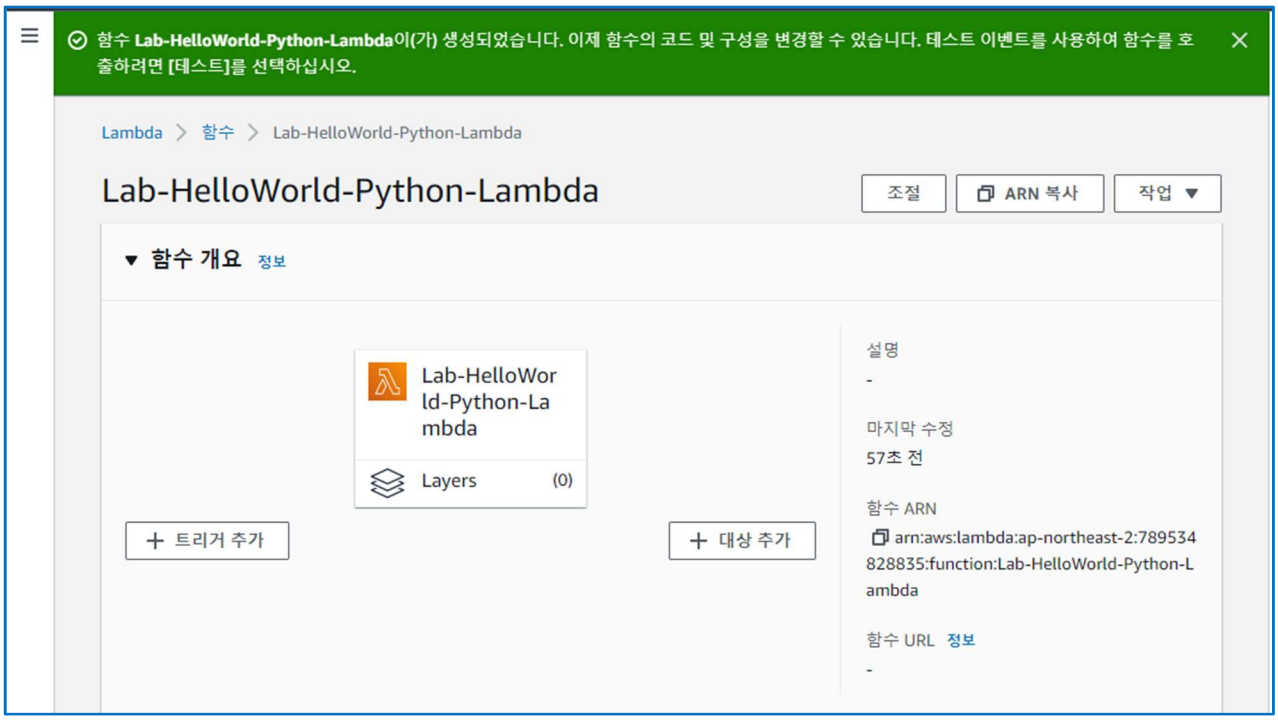
3. **[함수 생성]** 페이지에서 함수를 생성하기 위해 **[새로 작성]**을 선택한다.



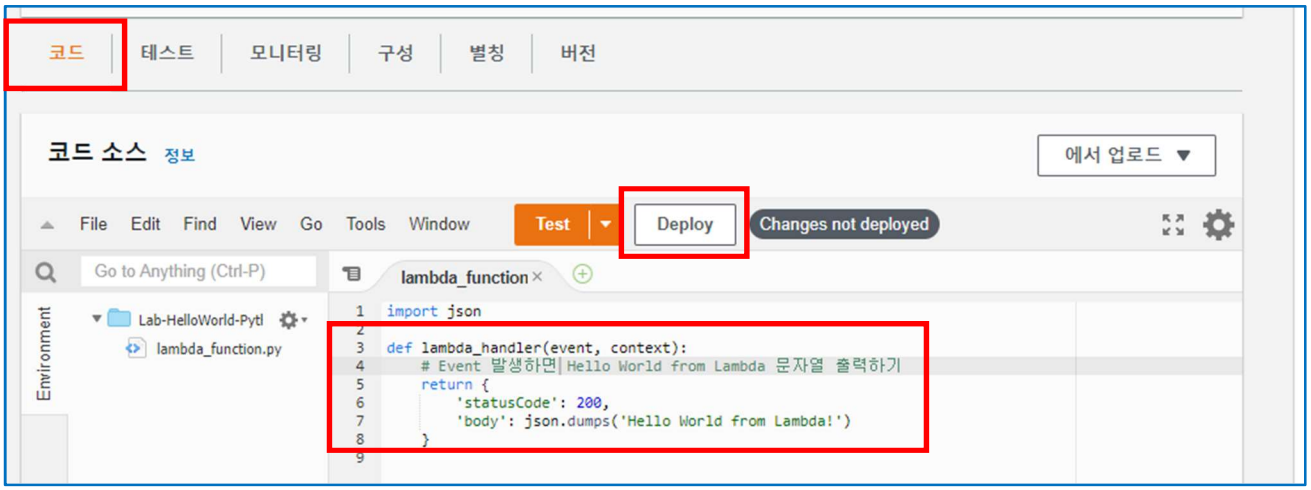
4. **[기본 정보]** 섹션에서, **[함수 이름]**은 **Lab-HelloWorld-Python-Lambda, [런타임]**은 **Python 3.9**를 선택하고, 페이지 오른쪽 하단의 **[함수 생성]**을 클릭한다. **[권한]** 같은 나머지 설정은 기본값을 사용하기로 한다.



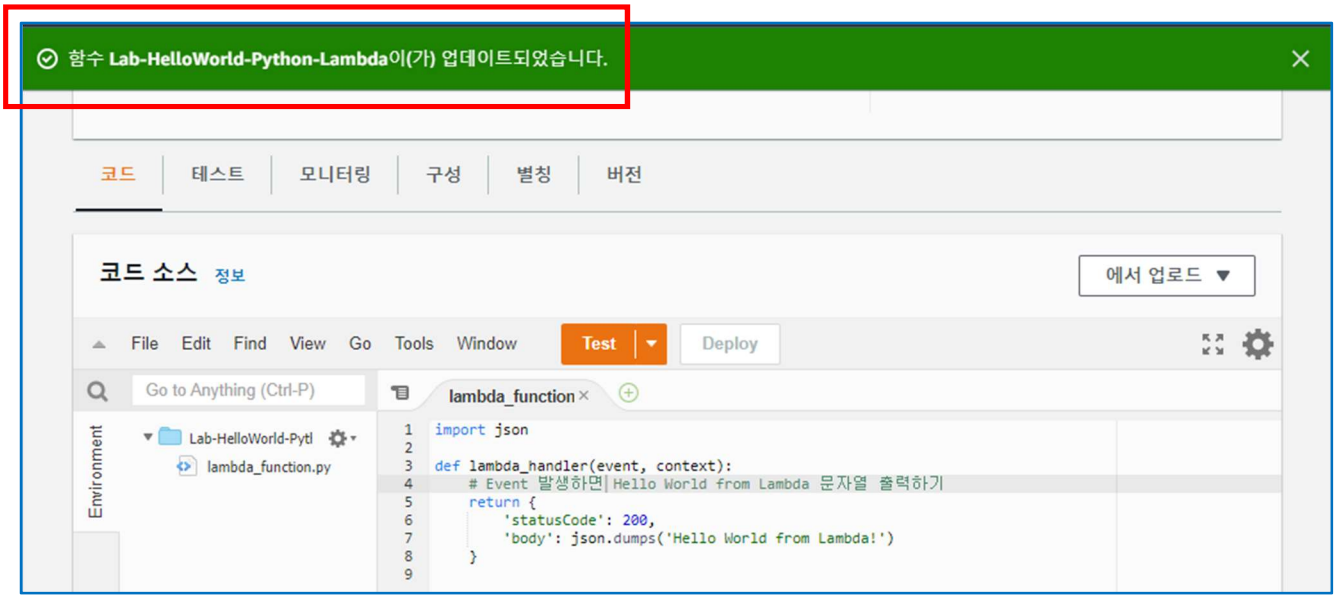
5. 정상적으로 **Lambda 함수**가 잘 생성되었다.



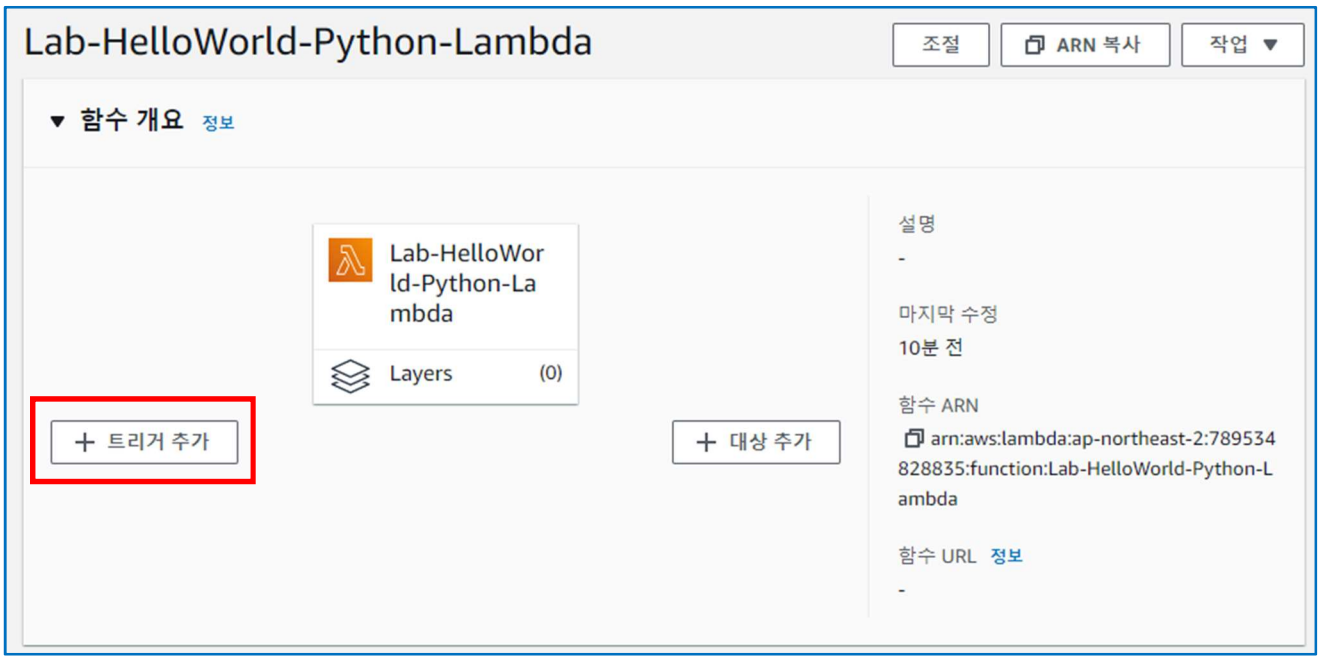
6. 페이지를 아래로 스크롤다운하여, **[코드]**탭에서 다음과 같이 코드를 간단히 수정한다. 수정이 완료되었으면 **[Deploy]** 버튼을 클릭하여 배포한다.



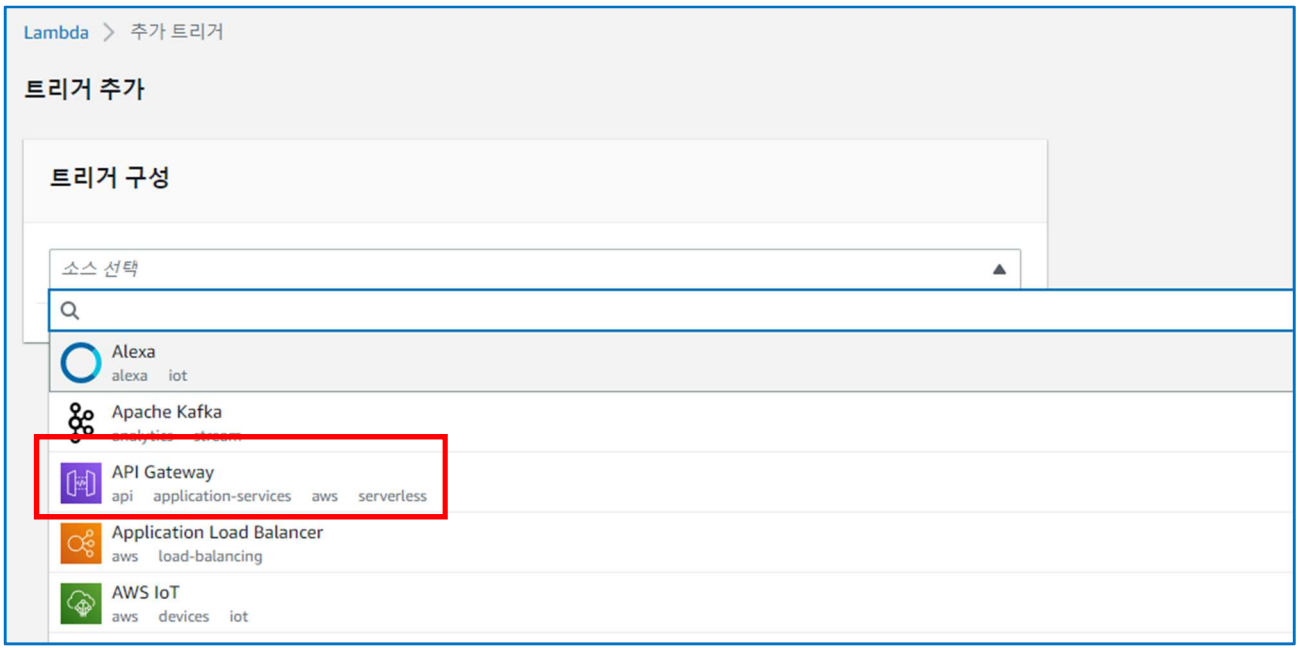
7. 생성한 코드가 잘 배포되었다.



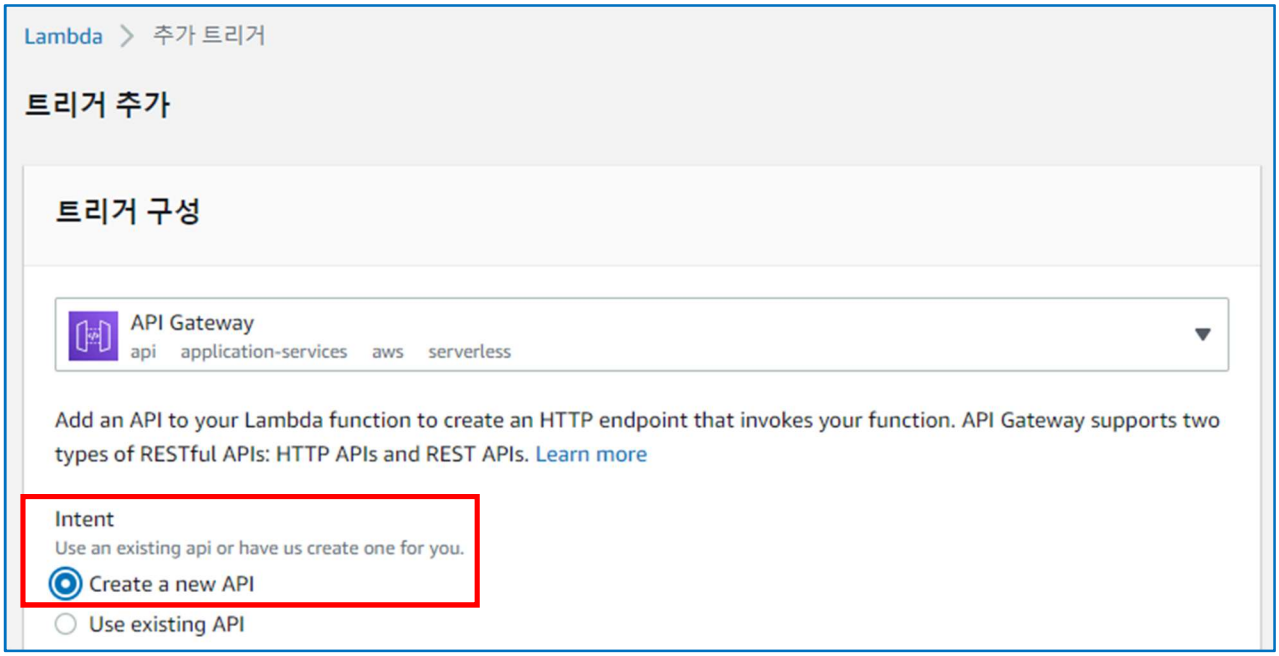
8. 배포가 완료되었으면 이 함수를 호출하게 하는 트리거를 추가해보자. 패이지를 다시 스크롤업하여 **[트리거 추가]**를 클릭한다.



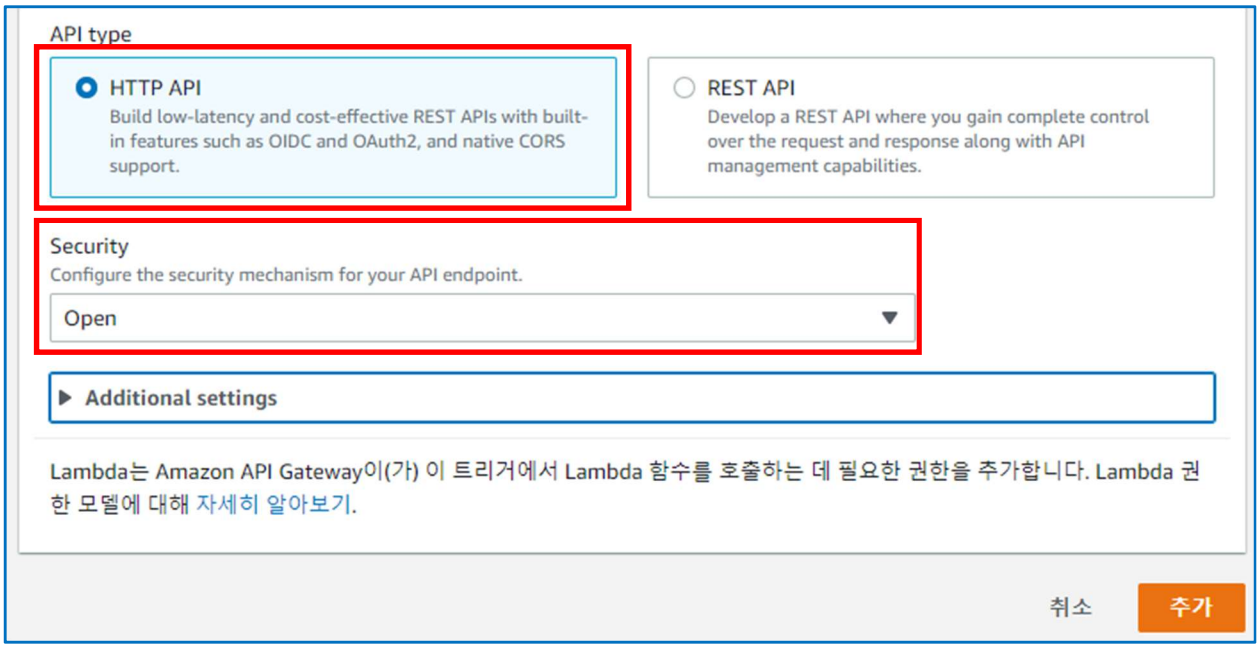
9. **[트리거 추가]** 페이지이다. **[트리거 구성]** 섹션에서, **[소스 선택]** 드롭다운 목록에서 **[API Gateway]**를 선택한다.



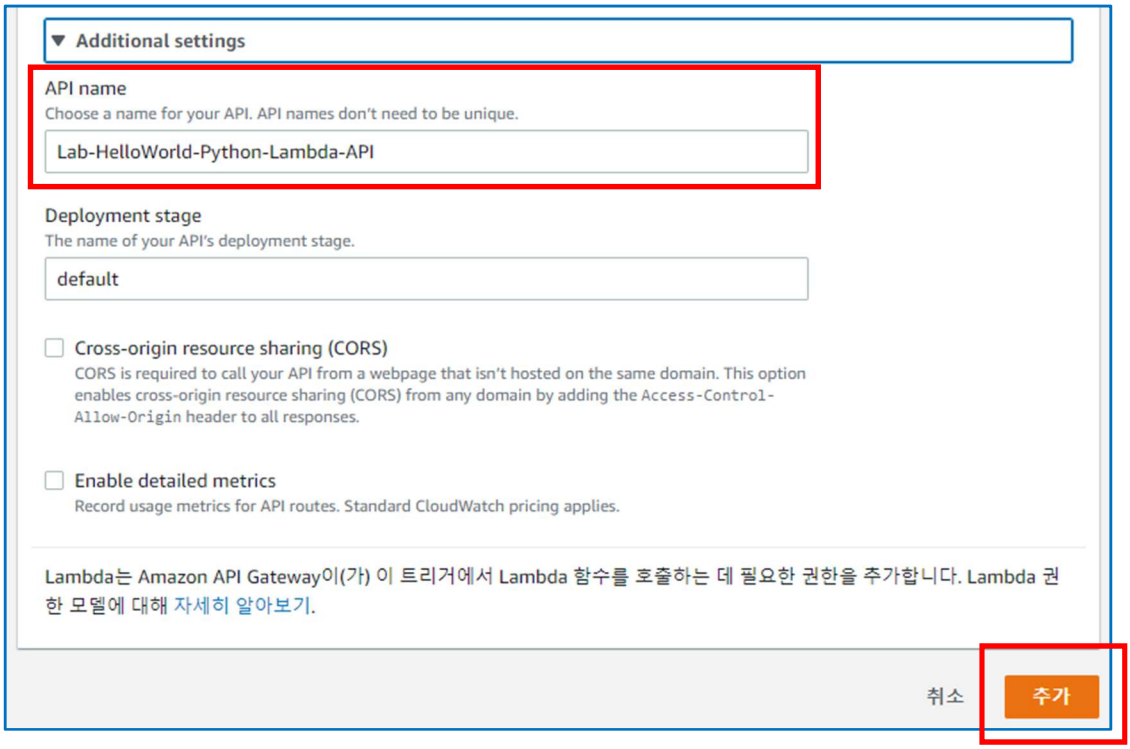
10. **[Intent]**에서 새 API를 생성하기 위해 **[Create a new API]**를 선택한다.



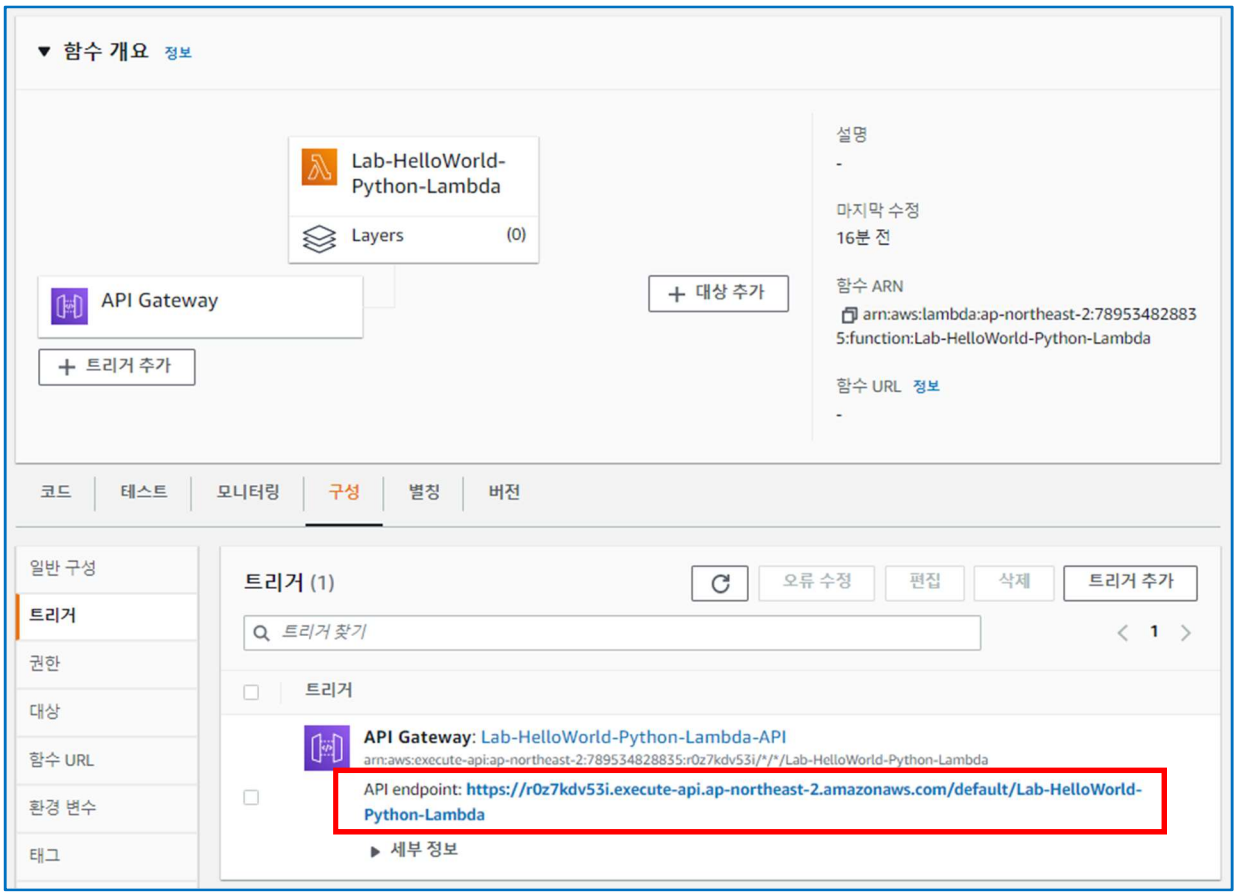
11. **[API type]**은 기본값 [**HTTP API]**를 선택하고, **[Security]**은 **[Open]**을 선택하여 인터넷 주소만 안다면 누구나 인증키 없이 접근할 수 있도록 한다. 그 아래 **[Additional settings]**를 클릭해보자.



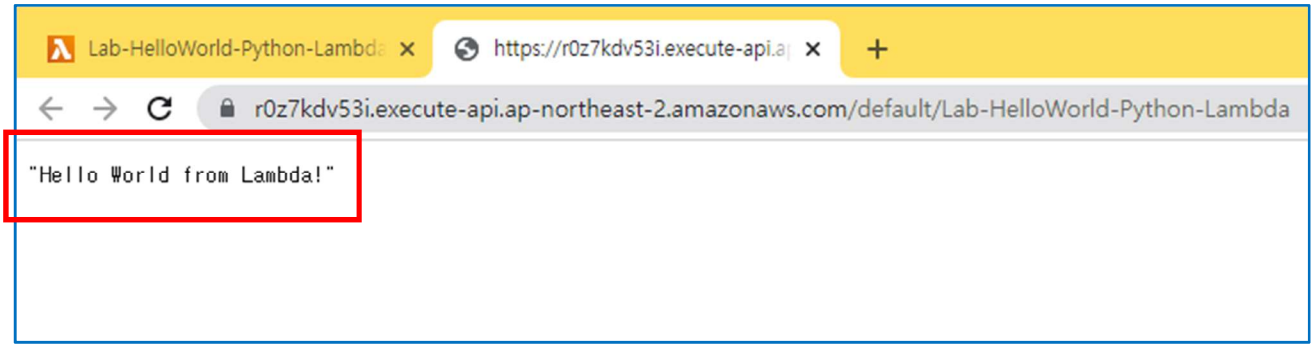
12. **[API name]**은 기본적으로 생성된 이름 그대로 사용하기로 하고, 나머지 설정 역시 기본값 그대로 사용하기로 한다. 페이지 우측 하단의 **[추가]** 버튼을 클릭한다 .



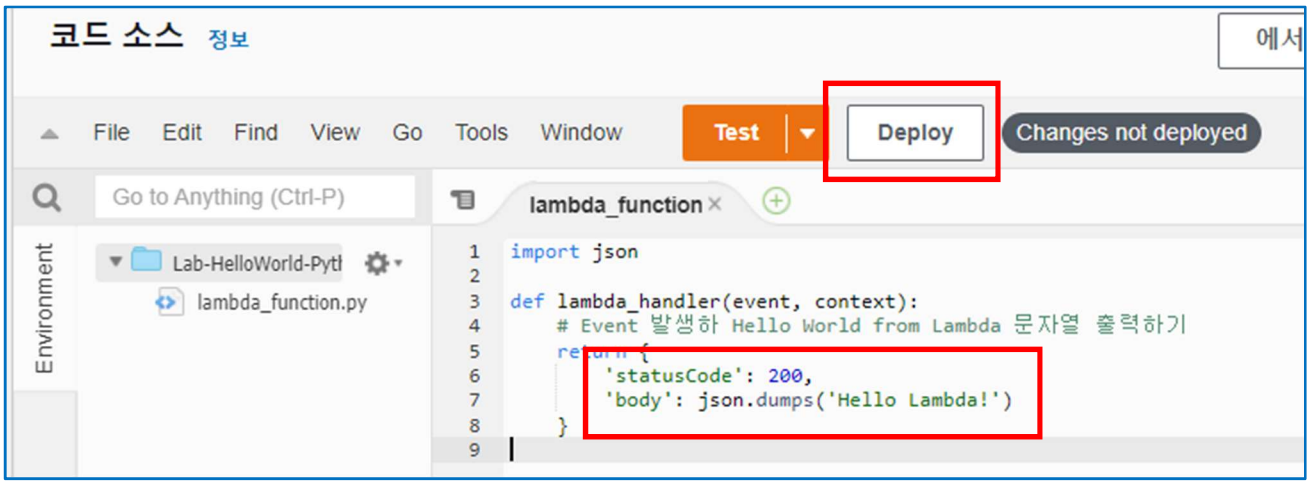
13. 이렇게 해서 **[API Gateway]**를 생성했다. 생성한 **[API Gateway]**의 **[API endpoint]** 링크를 클릭해보자.



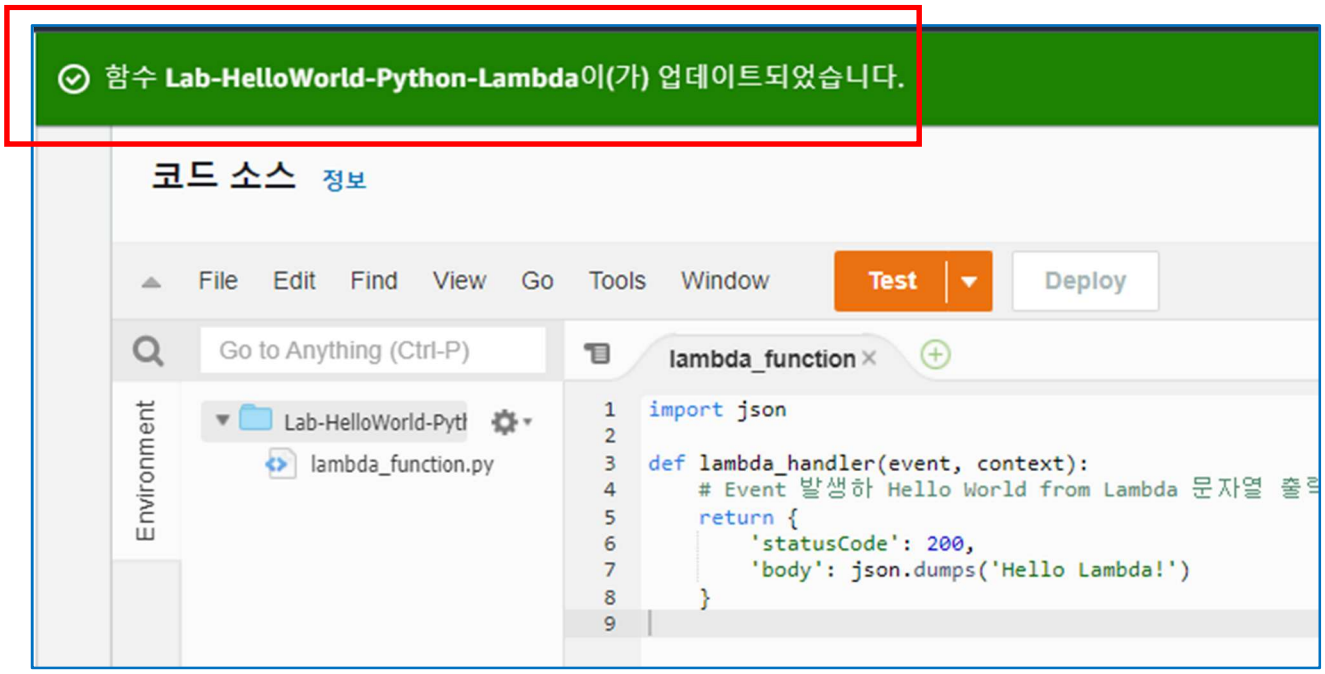
14. 앞에서 설정한 함수가 유저에게 전달하려고 했던 **"Hello World from Lambda!"**가 성공적으로 출력되는 것을 볼 수 있다.



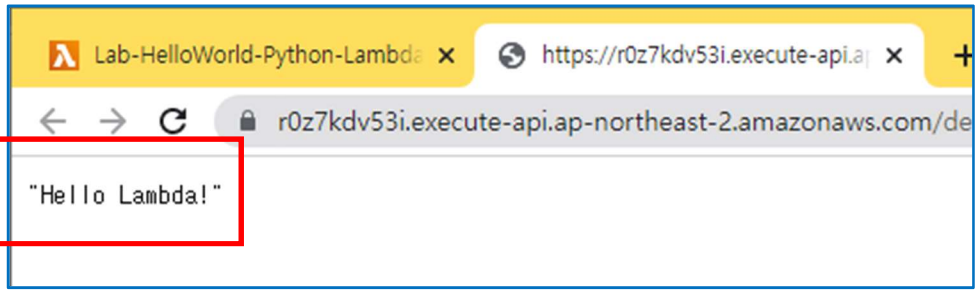
15. 다시 **[코드]**탭을 클릭하여 다음과 같이 코드를 수정했다. 그리고 **[Deploy]** 버튼을 클릭한다.



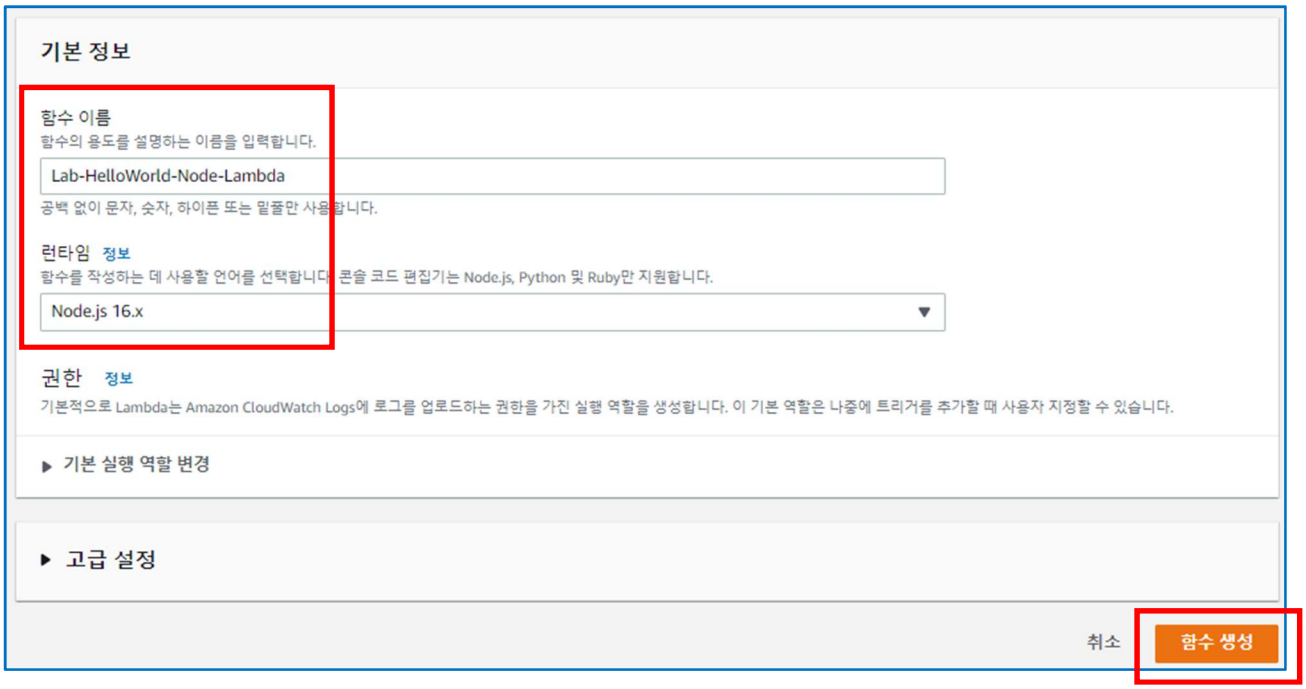
16. 수정한 코드의 업데이트가 끝나면, 다시 결과 화면에서 **F5**를 클릭하여 하면을 재시작한다.



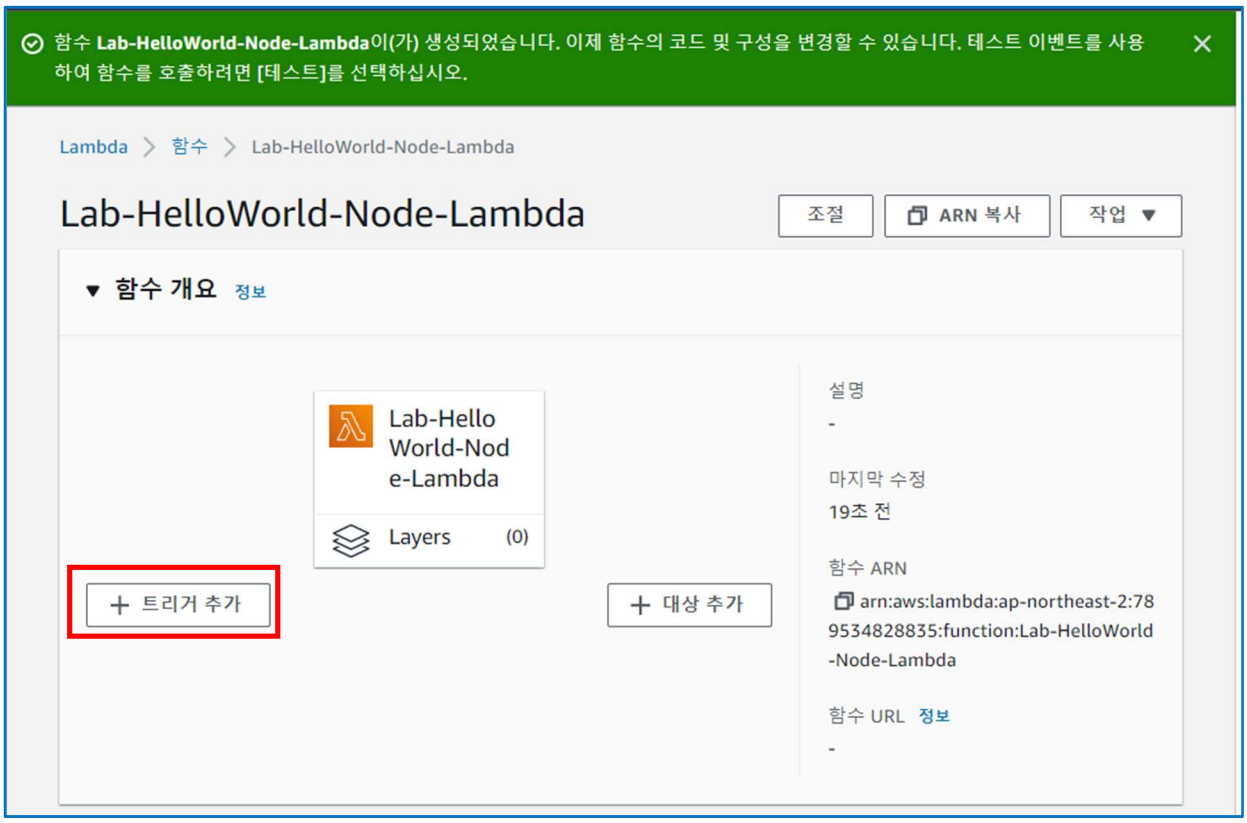
17. 변경된 문자열이 rendering 되는 것을 확인할 수 있다.



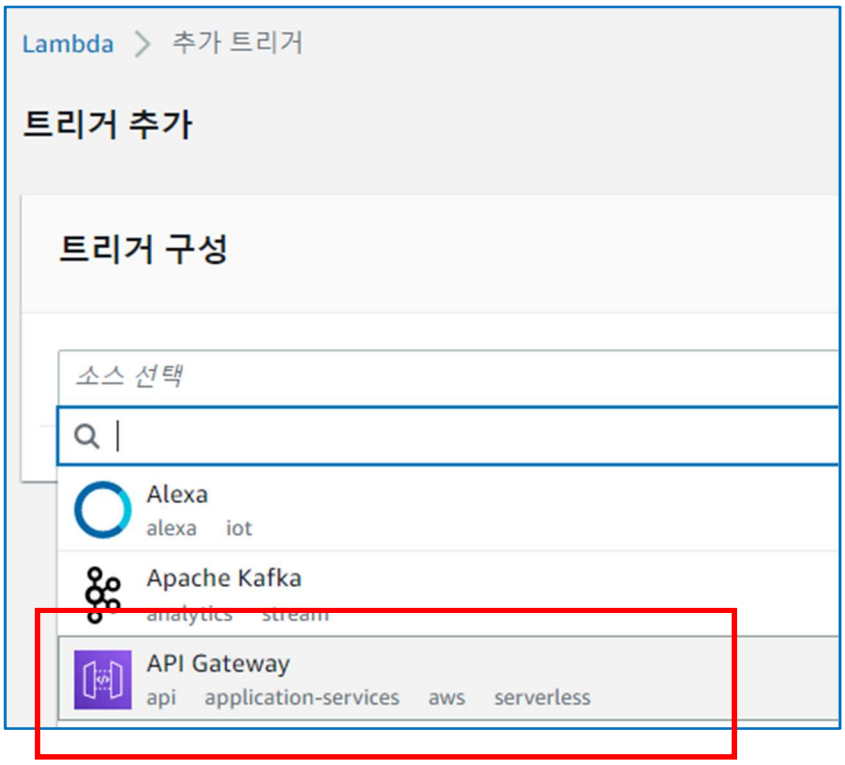
18. 이번에는 새 함수를 생성하겠다. **[함수 이름]**은 **"Lab-HelloWorld-Node-Lambda"**로, **[런타임]**은 **Node.js 16.x**로 설정한 후, **[함수 생성]**을 클릭한다.



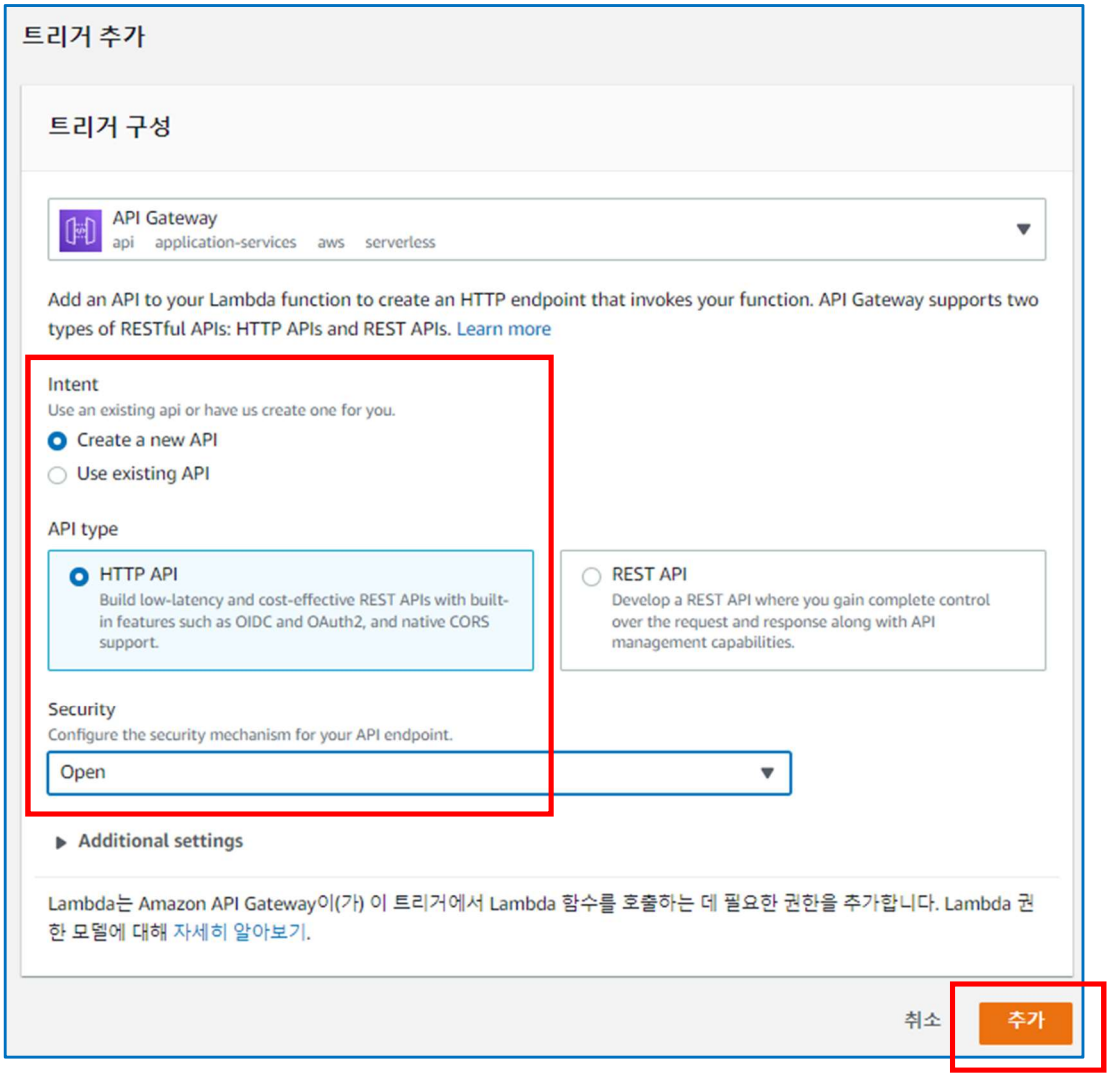
19. 새로운 함수가 잘 생성되었다. 이제 **[트리거 추가]**를 클릭한다.



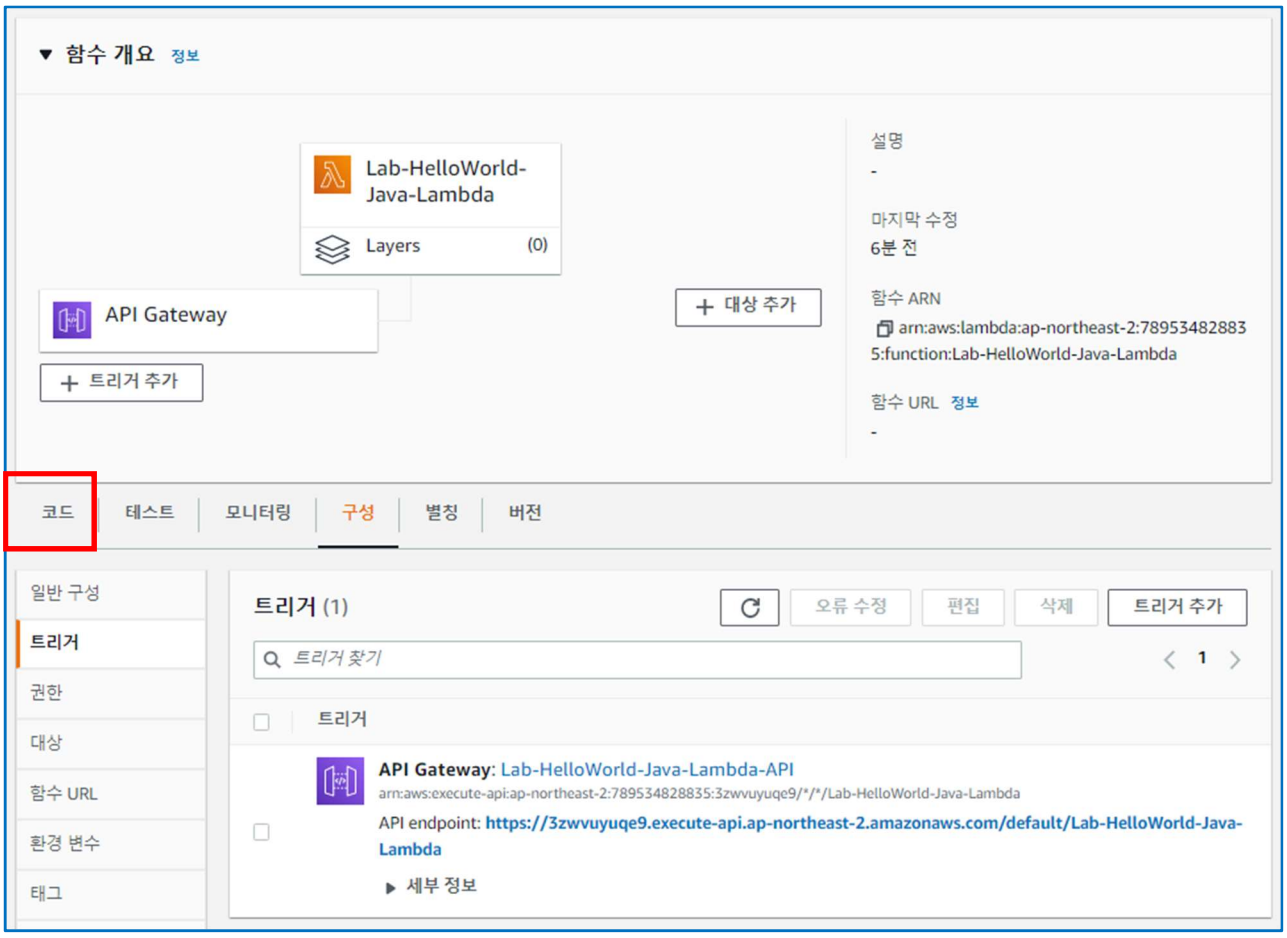
20. **[트리거 추가]**페이지에서, **[트리거 구성]** 섹션에서 **[소스 선택]**을 **[API Gateway]**로 선택했다.



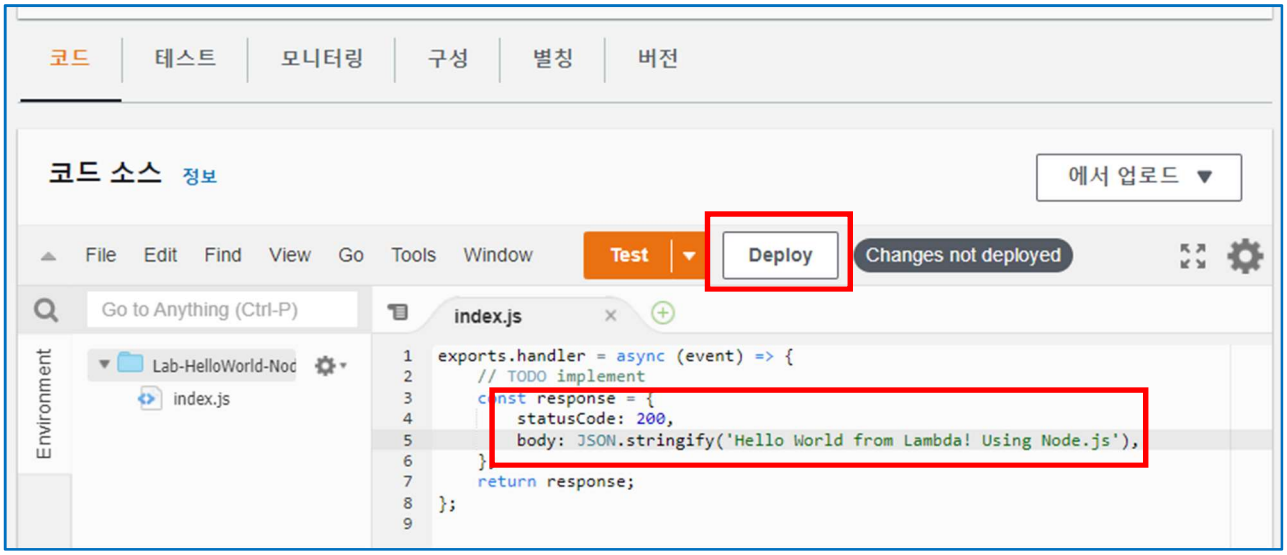
21. 이전에 생성했던 **트리거**와 같이 기본값을 설정하고 **[추가]** 버튼을 클릭한다.



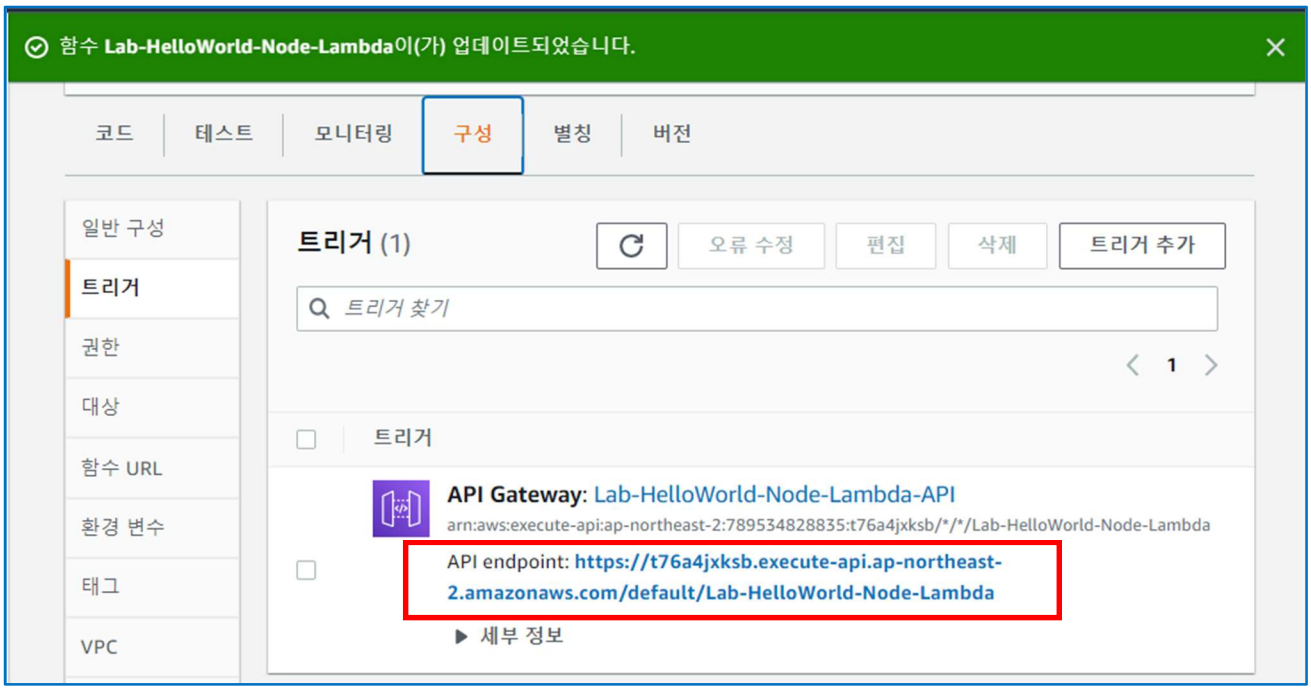
22. **트리거**가 성공적으로 추가되었으면 **[코드]**을 클릭하여 함수의 소스를 확인해 보자.



23. 코드를 다음과 같이 수정하고 **[Deploy]** 버튼을 클릭한다.



24. 함수 코드의 수정이 끝나면 **[API endpoint]**의 링크를 클릭한다.



25. 우리는 불과 수 분만에 간단한 코드 편집만으로 웹 서버를 생성할 수 있었다.