



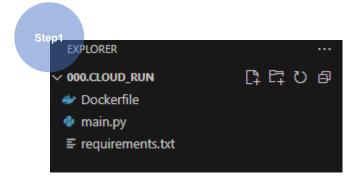
# (실습) 6차수-서버리스



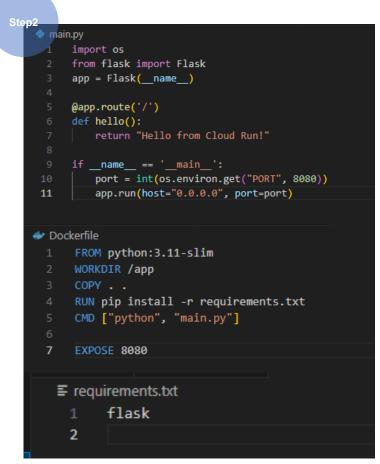
- I HTTP 트리거 함수 생성
- Ⅲ BigQuery 연동



# Cloud Run - gcloud shell



[그림 1] Cloud Run Sample 파일 구성



[그림 2] Cloud Run Sample 파일 내용



- gcloud shell

Building using Dockerfile and deploying container to Cloud Run service [cloud-run-sample-1] in project [gcp-study-463918] region [us-central1]    Building and deploying new service Uploading sources.   Building and deploying new service Building Container.   OK Building and deploying new service Done.   Building Container Logs are available at [https://console.cloud.google.com/cloud-build/builds;region=us-central1/731fd392-57df-4fe9-9869-b4db14d1b88a?project=1048797590207].   OK Creating Revision OK Routing traffic   OK Setting IAM Policy   Done.   Service [cloud-run-sample-1] revision [cloud-run-sample-1-00001-9zg] has been deployed and is serving 100 percent of traffic.   Service URL: https://cloud-run-sample-1-1048797590207.us-central1.run.app									
88	★ Bookmarks	🗀 새 폴더	oracle server	🗀 검색	Synology Web Assi.	🔇 KCU 컨			
Hello from Cloud Run!							[그림 3] Cloud Run Sample 실행 및 결과 1		
모든 행 선택	서비스 필터링								
	이름 ↑	배포유형	초당 요청 ⑦	리전	인증 ⑦	인그레스 ⑦	최근 배포 시간	배포자	추천
	cloud-run-sample-1	(**) 소스	0	us-central1	비인증 액세스 허용	전체	9분 전	jshan97@gmail.com	_

[그림 4] Cloud Run Sample 실행 및 결과 2

### I HTTP 트리거 함수 생성

#### 교육 서비스

#### **Cloud Run**

- 함수작성



[그림 5] Cloud Run Function 메뉴 – 함수 작성(클릭)



[그림 6] Cloud Run Function 메뉴 – 함수 작성(클릭)



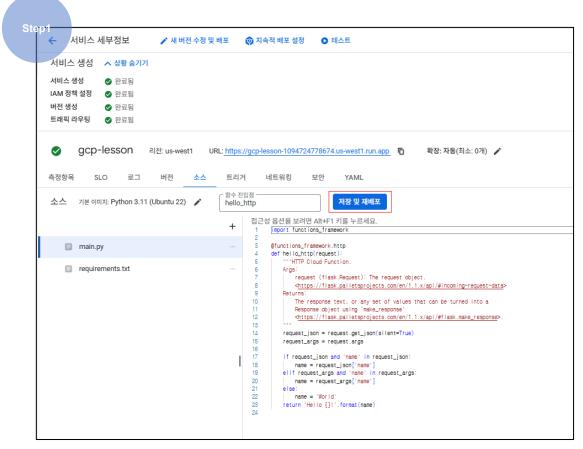
[그림 7] 함수 작성 – 구성 옵션



[그림 8] 함수 작성 – 결제 옵션



- 함수작성



[그림 9] 함수의 코드를 확인 후 저장 및 재배포(클릭)

작성된 함수의 URL 주소로 브라우저에 접속 하여 결과 확인



[그림 10] 함수 URL 로 접속한 결과



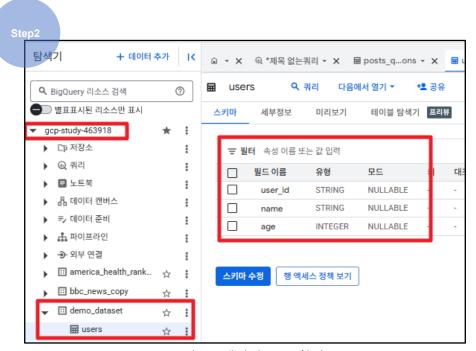
[그림 11] 함수 URL + ?name=GCP 수강생 을 추가하여 접속한 결과



BigQuery : 데이터 생성 및 확인

```
-- Dataset table 생성
CREATE TABLE demo_dataset.users (
    user_id STRING,
    name STRING,
    age INT64
);
-- 데이터 입력
INSERT INTO demo_dataset.users (user_id, name, age)
VALUES ('u001', 'Alice', 25), ('u002', 'Bob', 30), ('u003', 'Carol', 22);
-- 데이터 조회 확인
SELECT user_id, name, age
    FROM `gcp-study-463918.demo_dataset.users`
    WHERE age >= 20;
```

[그림 12] main.py



[그림 13] 생성된 table 확인



서비스 생성 소스 코드 및 설정

```
functions-framework==3.5.0
google-cloud-bigquery
pandas
db-dtypes
```

[그림 14] requirements.txt



[그림 16] 서비스 세부정보 > 소스

```
import pandas as pd
from google.cloud import bigquery
from flask import jsonify, Request
bq client = bigquery.Client()
def query_users(request: Request):
  age_param = request.args.get("age", default=20, type=int)
  query = """
    SELECT user id, name, age
    FROM 'gcp-study-463918.demo dataset.users'
    WHERE age >= @age param
  job_config = bigquery.QueryJobConfig(
    query_parameters=[
       bigquery.ScalarQueryParameter("age_param", "INT64", age_param)
  df = bq_client.query(query, job_config=job_config).to_dataframe()
  df["name"] = df["name"].str.upper()
  return jsonify(df.to dict(orient="records"))
```

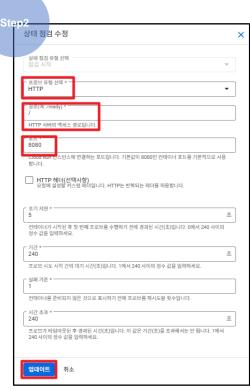
[그림 15] main.py



서비스 생성 설정 및 결과 확인



[그림 17] 컨테이너 수정



[그림 18] 상태점검 수정



[그림 19] 생성 결과



[그림 20] 실행 결과