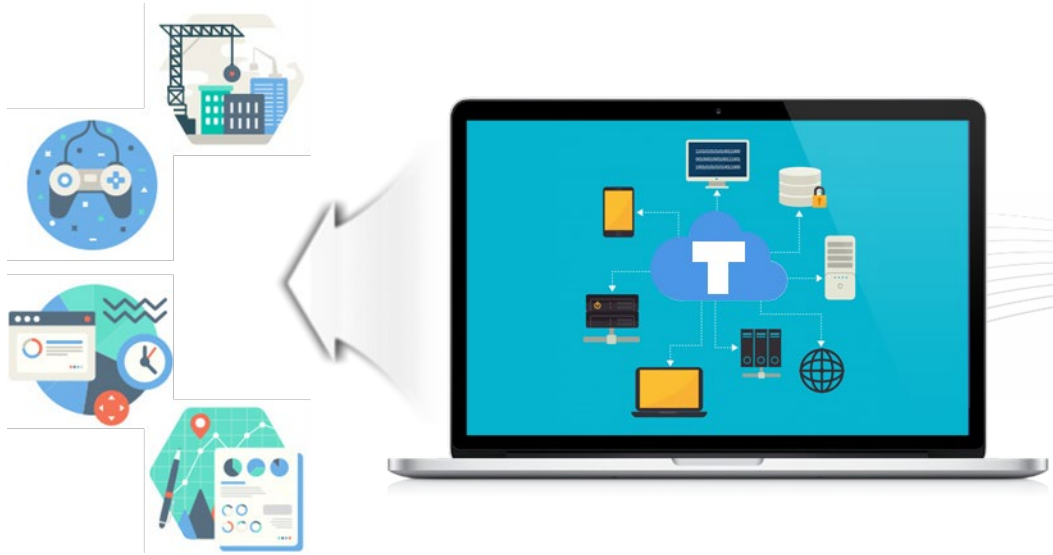


(이론) GCP-6차수

Ⅰ 서버리스 개념

Ⅱ Cloud Run Functions

Ⅲ Cloud Run 심화

서버리스 아키텍처

- 개발자가 기본 아키텍처를 관리하지 않고도 애플리케이션을 빌드하고 관리할 수 있는 소프트웨어 설계 방식
- 서버가 없음 X / 서버를 관리할 필요가 없음 O
- 개발자의 관점에서 보면 프로비저닝, 하드웨어 유지보수, 소프트웨어 및 보안 업데이트와 기타 서버 관리 태스크 등의 걱정 없이 코드를 작성하고 실행
- 서버리스 아키텍처는 트래픽에 따라 자동으로 확장되거나 축소
- 클라우드 인프라에 의존적

사용 예시

트리거 기반 작업 또는 예약된 태스크 실행(예: 일일 보고서, 백업, 비즈니스 로직 등)

웹 및 모바일 애플리케이션의 RESTful API 빌드

비동기 처리(예: 동영상 트랜스코딩)

액세스 권한 자동 삭제, 규정 준수 보안 검사 시작, 승인 전송과 같은 IT 프로세스 자동화

지속적 통합 및 지속적 배포(CI/CD) 파이프라인 자동화(예: 빌드를 트리거하는 코드 커밋)

타사 서비스 및 API와 통합

예약된 태스크 실행(예: 일일 보고서, 백업, 비즈니스 로직 등)

정형 데이터와 비정형 데이터의 실시간 데이터 처리

서버리스의 도입시

- 서버리스는 기존의 VM 방식이 가진 정적인 한계와 시스템 유지보수의 이슈를 해결하는 혁신적인 서비스 설계 방식
- 이벤트 기반으로 실행되어 비용 효율적이고 인프라 관리의 부담이 감소하고 비즈니스 로직에만 집중할 수 있다는 장점이 있음
- 하지만 기존 방식과 달라짐에 따라 주의해야하는 점들이 있음

주의점	상세
서드파티 종속성 관리	• 서드파티 종속성이 성능에 영향을 미치거나 보안 취약성을 초래 • 필요 최소한의 종속성만 사용하고, 정기적으로 업데이트
콜드 스타트 문제	• 일정 시간 이상 비활성 상태였던 함수를 호출될 때 지연이 발생 • 프로비저닝된 동시성이나 재시도 로직을 구현 필요
모니터링의 복잡성	• 개별 함수들이 모두 독립적으로 실행되고 있기 때문에 단순 모니터링으로는 오류를 관찰하기 어려움 • GCP Cloud Logging으로 Dashboard 구성하여 에러 취합
비용 관리 및 최적화	• 사용량 기반 과금 모델로 인해 역으로 예기치 않은 비용이 발생 위험 • Cloud Billing을 통해 실시간 비용 모니터링과 예산 설정하여 과도한 비용 발생 방지
보안 문제	• 인프라 보안은 클라우드 제공자가 담당하지만, 코드와 설정의 보안은 여전히 개발자의 책임 • 최소 권한 원칙을 적용하고 IAM 서비스를 활용하여 보안을 강화

[표 1] Serverless 도입시 주의해야 할 점

GCP 에서의 서버리스 서비스

Cloud Function

- 이벤트 기반의 서버리스 서비스형 함수
- 개발자가 인프라 관리 없이 코드를 실행할 수 있도록 함
- 함수를 배포하여 이벤트에 대한 응답으로 코드를 실행하는 서비스이기 때문에 가용한 언어가 제한

Cloud Run

- GCP의 완전 관리형 서버리스 플랫폼
- 컨테이너화된 애플리케이션을 손쉽게 배포하고 확장할 수 있도록 함
- Cloud Functions와는 다르게 가용 언어에 한계가 존재하지 않음

BigQuery

- GCP의 완전 관리형 데이터웨어하우스 서비스
- 대규모 데이터를 빠르고 효율적으로 분석할 수 있는 고성능 분석 엔진을 제공함

2017년 3월

Cloud Functions 1세대

- App Engine 인프라를 기반으로 구축되어 하나의 인스턴스는 한 개의 요청만 처리
- 실행 시간제한 / 최대 9분

2019년 11월

Cloud Run

2022년 12월

Cloud Functions 2세대

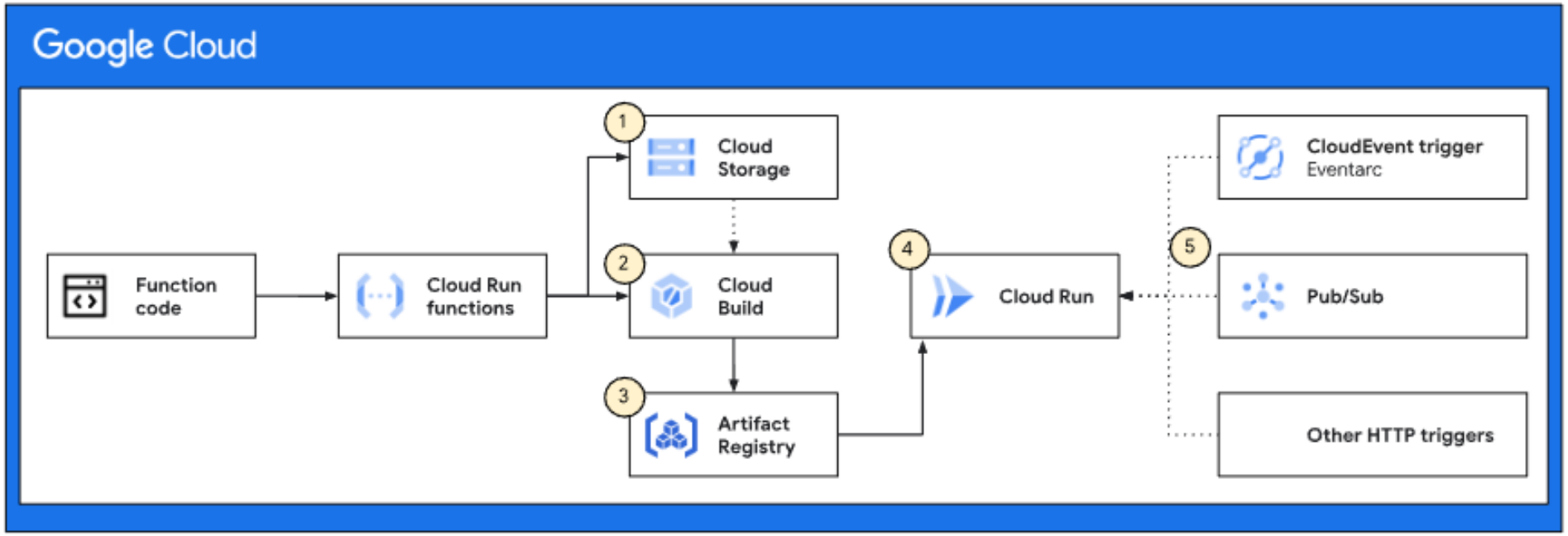
- Cloud Run의 컨테이너 기반 인프라를 바탕으로 구축
- 다중 이벤트 트리거 관리와 사용자 정의 런타임 환경 구성이 가능
- 리소스 최적화를 통해 비용을 절감
- 실행 시간제한 / 최대 60분 (HTTP 트리거-60분, 이벤트 트리거-9분)

2024년 8월

Cloud Functions 2세대를 Cloud Run Functions로 리브랜딩

Cloud Run Functions을 만들었는데 Cloud Run을 실행한다..?

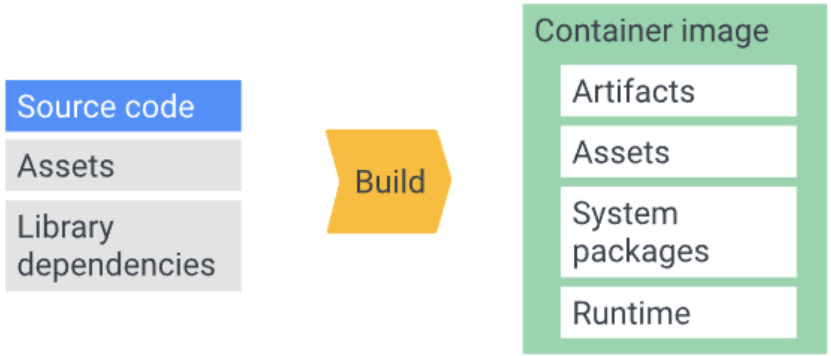
- Cloud Function은 실행 시 서비스가 바로 실행되는 것이 아니라 Cloud Run을 기반으로 실행됨
  1. 함수 소스 코드를 Cloud Storage 버킷에 저장
  2. Cloud Build로 코드를 컨테이너 이미지에 자동으로 빌드
  3. 함수 이미지를 Artifact Registry 이미지 레지스트리로 내보냄
  4. Cloud Run에서 함수 이미지에 액세스하여 컨테이너 실행



[그림 1] Cloud Run Functions 실행 흐름

서비스 또는 작업은 컨테이너 이미지에 패키징 되어야 함

- 서비스 또는 작업을 Cloud Run에 배포하려면 이를 컨테이너 이미지에 패키징 해야 함
- 컨테이너 이미지는 서비스 실행에 필요한 모든 것이 포함된 패키지 (빌드 아티팩트, 애셋, 시스템 패키지, 런타임(선택사항)이 포함)



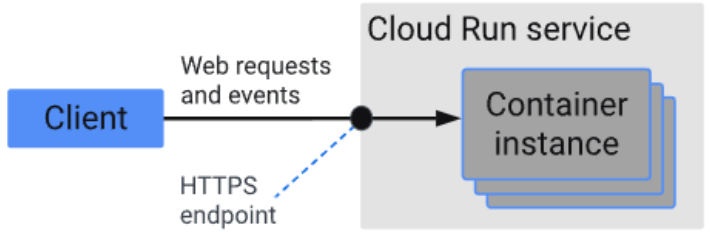
[그림 2] Cloud Run 컨테이너 빌드

Cloud Run 사용 시 고려할 점

- HTTP, HTTP/2, WebSockets, gRPC를 통해 제공되는 요청, 스트림, 이벤트를 지원하거나 실행하여 완료되는 작업
- 로컬 영구 파일 시스템이 필요하지 않지만 로컬 임시 파일 시스템 또는 네트워크 파일 시스템이 필요
- 동시에 실행되는 여러 앱 인스턴스를 처리하도록 작성되어야 함
- 요구 수준이 인스턴스당 CPU 8개와 32GiB 메모리를 초과하지 않아야 함

Cloud Run : 서비스

- 웹 요청, 이벤트, 함수에 응답하는 코드를 실행하는 데 사용
- 안정적인 HTTPS 엔드포인트를 실행하는 데 필요한 인프라를 제공



[그림 3] Cloud Run 서비스 흐름도

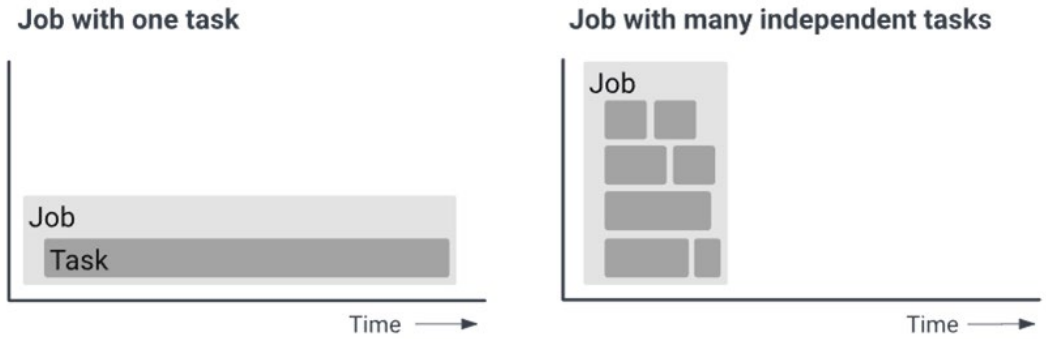
활용 사례	상세
웹사이트 및 웹 애플리케이션	• 원하는 스택을 사용해서 웹앱을 빌드하고 SQL 데이터베이스에 액세스하고, 동적 HTML 페이지를 렌더링
API 및 마이크로서비스	• REST API나 GraphQL API 또는 HTTP 또는 gRPC로 통신하는 비공개 마이크로 서비스를 빌드
스트리밍 데이터 처리	• Cloud Run 서비스는 Pub/Sub 푸시 구독에서 메시지를 수신하고 Eventarc에서 이벤트를 수신
비동기 워크로드	• Cloud Run Functions는 Pub/Sub 주제의 메시지, Cloud Storage 버킷의 변경사항, Firebase 이벤트와 같은 비동기 이벤트에 응답
AI 추론	• GPU 구성 여부와 관계없이 Cloud Run 서비스는 추론 모델 및 모델 학습과 같은 AI 워크로드를 호스팅

[표 2] Cloud Run 서비스 활용 예시

Cloud Run : 작업

- 작업을 수행하고, 작업이 완료되면 종료하는 코드를 실행하기 위해 사용
- 반복 작업을 예약하거나, 워크플로의 일부로 실행
- 배열 작업을 사용하여 여러 개의 동일하고 독립적인 인스턴스를 병렬 실행 ㄱ

  예) Cloud Storage에서 1000개의 이미지 크기 조정 작업을 병렬 처리



[그림 4] Cloud Run 작업 순차 처리 vs 병렬 처리

활용 사례	상세
스크립트 또는 도구	• 스크립트를 실행하여 데이터베이스 마이그레이션 또는 기타 운영 작업을 수행
배열 작업	• Cloud Storage 버킷에 있는 모든 파일의 처리를 대량으로 동시에 수행
예약된 작업	• 정기적인 간격으로 인보이스를 만들고 전송하거나 데이터베이스 쿼리 결과를 XML로 저장하고 파일을 몇 시간 간격으로 업로드

[표 3] Cloud Run 작업 활용 예시

Cloud Run : 함수 트리거

- 함수가 호출되거나 트리거 되는 방식은 코드를 작성할 때 지정하는 함수 유형에 따라 다름

구분	상세
CloudEvent 트리거	• Eventarc에서 트리거한 Pub/Sub 이벤트
	• Eventarc에서 트리거한 Cloud Storage 이벤트
	• Eventarc에서 트리거한 Firestore 이벤트
HTTP 트리거	• HTTP 요청으로 호출
	• Workflows를 사용하여 워크플로의 일부로 서비스 호출
	• Cloud Scheduler를 사용하여 일정에 따라 서비스 호출
	• Cloud Tasks를 사용하여 비동기 태스크 실행
	• Pub/Sub 푸시(push) 구독에서 서비스 트리거

[표 4] 함수 트리거 종류