

## 4.서버리스 컴퓨팅 도입시 고려사항

Cloud\_서비스/세일즈 툴킷

Exported on 06/19/2023

## Table of Contents

1	공급업체 종속성.....	3
2	공급업체 마다 상이한 SLA .....	4
3	복잡한 운영 관리.....	5
4	다양한 제약 사항.....	6
4.1	최대 실행 시간.....	6
4.2	최대 동시 실행 수.....	6
4.3	콜드스타트 지연시간 .....	6

## 1 공급업체 종속성

서버리스 컴퓨팅같은 PaaS 서비스는 공급업체 종속성(Vender Lock-in)을 피할 수 없다. PaaS 서비스를 사용하다가 다른 클라우드 업체나 On-Premise로 이전하는 것은 막대한 비용이 발생한다. 또한 서버리스 컴퓨팅은 제공하는 업체별로 사용 방법이 다르고 트리거 서비스가 각 공급사 서비스별로 연계되지 않기에 멀티 클라우드 전략이 어렵다. 한 예로 AWS Kinesis 라는 데이터 스트림 서비스는 AWS Lambda 와 연결하여 사용하나 Azure Functions 나 Apache OpenWisky에는 연계할 수 없다.

또한 특정 국가에서는 시스템의 서버가 반드시 해당 국가에 위치해야 하는데 CSP의 데이터 센터는 모든 국가에 위치해 있지는 않다. 이런 상황에서는 시스템 수출 시 서버리스 컴퓨팅 뿐만 아니라 다른 PaaS 서비스들도 직접 구현해야 할 상황이 발생할 수 있다.

**공급업체 종속성을 피하려면 오픈소스 서버리스 컴퓨팅 플랫폼을 이용해야 한다.**

## 2 공급업체 마다 상이한 SLA

서버리스 컴퓨팅의 장점 중 하나는 서버 설정 등의 관리를 공급업체에서 진행하는 것이다. 하지만 공급업체의 실수로 장애가 발생할 수 있다. 한 예로 AWS의 Lambda 서비스 초기엔 잘 사용하던 함수가 몇 시간 동안 동작하지 않은 적이 있었고 AWS의 최초 서비스인 S3도 2017년 2월에 장애가 발생하여 이에 연결된 Lambda가 동작하지 않은 적도 있었다. [1] 이런 상황에 대비가 필요하고 필요에 따라서 보상을 받기 위해 SLA나 이용약관을 확인해야 한다.

AWS Lambda는 SLA가 없어 AWS의 문제로 인해 장애가 발생해도 보상 받을 수 없다. 또한 AWS의 Lambda 서비스 이용약관에는 “AWS Lambda에 업로드된 고객의 콘텐츠가 3개월 간 실행되지 않으면 30일 전에 고지하고 삭제할 수 있으며 이에 대해 어떠한 책임도 없다.” 라고 명시되어 있다.

### 42. AWS Lambda

**42.1** You are responsible for Your Content, including (a) the performance of software you use with AWS Lambda and any reference libraries we provide and (b) maintaining licenses and adhering to the license terms of any software you run.

**42.2** You are responsible for all fees incurred for Services used in connection with AWS Lambda.

**42.3** We may delete, upon 30 days' notice to you and without liability of any kind, any of Your Content uploaded to AWS Lambda if it has not been run for more than three (3) months.

그림. AWS Lambda의 서비스 이용약관 중 일부 [2]

Azure의 Functions는 사용한 만큼 비용이 발생하는 Consumption plan과 자신의 Dedicated VM에서 Functions가 동작하는 App Service plan 으로 구분하여 사용한다. 그리고 App Service plan 을 이용해야 Functions에 SLA [3]를 적용받을 수 있다. 또한 Azure Functions의 신규 버전인 2.x는 현재 Beta라 SLA에서 제외된다.

Google Cloud 의 Functions 는 SLA 를 제공한다.

IBM Cloud Functions 의 이용약관은 페이지 오류가 발생하여 현재 확인할 수 없다.

**이런 내용을 참고하여 크리티컬한 업무에는 서버리스 컴퓨팅을 사용하지 말거나 오류가 발생할 경우를 대비하여 호출 단에서 예외처리를 해야 한다.**

[1] <https://aws.amazon.com/ko/message/41926/>

[2] [https://aws.amazon.com/service-terms/?nc1=h\\_ls](https://aws.amazon.com/service-terms/?nc1=h_ls)

[3] [https://azure.microsoft.com/en-us/support/legal/sla/functions/v1\\_0/](https://azure.microsoft.com/en-us/support/legal/sla/functions/v1_0/)

### 3 복잡한 운영 관리

CSP의 서버리스 컴퓨팅 서비스를 이용하면 별도의 서비스에 로그가 쌓이게 되고 보관 비용이 발생한다. 수많은 Functions가 있는 경우, 이슈 발생 시 원인파악이 쉽지 않기에 이를 위해 배포 시 단위 테스트와 연계된 서비스간 통합 테스트도 필요하고 문제 발생 시 롤백할 수 있는 방안도 마련해야 하며, 문제 발생 시 이전 버전으로 롤백할 수 있는 방안도 마련해두어야 한다.

또한 서버리스 컴퓨팅은 단독으로 실행할 수 없고 트리거와 연계 서비스가 필요한데, 이를 위해서 서로간에 권한을 부여해야 한다. 일반적인 프로젝트 환경에서 개발 편의를 위해 모든 권한을 부여하는 경우가 있는데, 이는 보안 사고로 이루어질 수 있기 때문에 **필요한 권한만 사용하는 것이 중요하다**. 부족한 프로젝트 일정때문에 많은 권한을 부여하고 차차 권한을 회수하는 방식을 사용하는 경우가 있는데, 이 작업은 번거롭고 쉽지 않기 때문에 권장하지 않는다.

## 4 다양한 제약 사항

서버리스 컴퓨팅은 물리서버의 자원을 효율적으로 사용되게 설계되었기에 함수를 최대로 실행할 수 있는 시간이나 최대 로 동작하는 함수 수에 제한이 있고 사용하지 않는 컨테이너를 반환하고 호출 시 다시 할당하는 과정에서 발생하는 지연 시간이 존재하여 이를 염두해두고 서버리스 컴퓨팅을 도입해야 한다.

### 4.1 최대 실행 시간

서버리스 컴퓨팅은 실행 제한시간이 존재하고 각 함수 별로 설정할 수 있어 잘못된 함수 로직으로 인한 과다 비용 발생을 방지할 수 있다. 하지만 제한 시간이 초과될 경우 함수가 바로 종료되니 이런 사항을 고려하여 함수를 작성해야 한다. 또한 서버리스 컴퓨팅 플랫폼별로 9 ~ 15분의 최대 실행 시간의 제한이 존재한다. **오래 동작하는 서비스인 경우 함수를 더 잘게 쪼개 사용하거나 가상머신 같은 ServerFull한 서비스를 이용해야 한다.** **최대 실행시간이 초과된 함수의 재시도는 호출 단에서 처리하는 것이 안전하다.**

### 4.2 최대 동시 실행 수

서버리스 컴퓨팅 플랫폼별로 함수를 동시에 실행할 수 있는 실행 수에 제한이 있으며 기준은 각각 다르다. 자세한 내용은 [5.1 퍼블릭 클라우드 플랫폼]을 참고하기 바란다.

만약을 대비하여 함수 호출서비스에 **동시 실행 수 오류가 발생한 경우의 로직을 추가해야 안정적인 서비스를 유지할 수 있다.**

### 4.3 콜드스타트 지연시간

서버리스 컴퓨팅이 이벤트 호출에 의해 시작할 때 함수를 수행하기 위해 컨테이너를 할당받는 준비과정이 수행되는데 이를 콜드스타트 Cold Start라 한다. 수행이 종료된 후 할당받은 컨테이너를 바로 반납하지 않고 일정기간 대기하고 있다 재 호출이 일어나면 할당받았던 컨테이너를 재사용하는데 이를 웜스타트 Warm Start라 한다. 콜드 스타트는 웜스타트에 비해 대략 100ms ~ 10초의 지연시간이 발생기 때문에 지연시간을 고려해서 비기능설계를 해야 하며, 콜드스타트로 인한 지연시간을 무시할 수 없는 서비스는 서버리스 컴퓨팅 외의 방안을 이용해야 한다.

- 공급업체 종속성(see page 3)
- 공급업체 마다 상이한 SLA(see page 4)
- 복잡한 운영 관리(see page 5)
- 다양한 제약 사항(see page 6)
  - 최대 실행 시간(see page 6)
  - 최대 동시 실행 수(see page 6)
  - 콜드스타트 지연시간(see page 6)