

2. 서버리스 컴퓨팅이란 무엇인가?

Cloud_서비스/세일즈 툴킷

Exported on 06/19/2023

Table of Contents

1 소프트웨어 아키텍처 패러다임 변화	4
2 서버리스 컴퓨팅 정의	6
3 서버리스 컴퓨팅 현황 분석	7
4 서버리스 컴퓨팅 동작 방식	9

서버리스 컴퓨팅에 대한 소개와 도입배경, 현황을 통해 서버리스 컴퓨팅에 대한 정보를 제공한다.

1 소프트웨어 아키텍처 패러다임 변화

최근 시스템은 서버에서 가상화를 넘어 컨테이너로 진화하고 있고, 모놀리틱 아키텍처가 마이크로서비스 아키텍처로 점진 변화하고 있다. 또한 서버 구동 방식도 사용자의 요청을 응답할 때까지 대기하는 구조에서 이벤트에 의해 동작하는 이벤트 드리븐 방식으로 진화하고 있다. 이 중 **어플리케이션 개발 및 운영의 단순화**라는 측면에서 서버리스 컴퓨팅이 새로운 아키텍처 설계 패러다임으로 부각되고 있다.

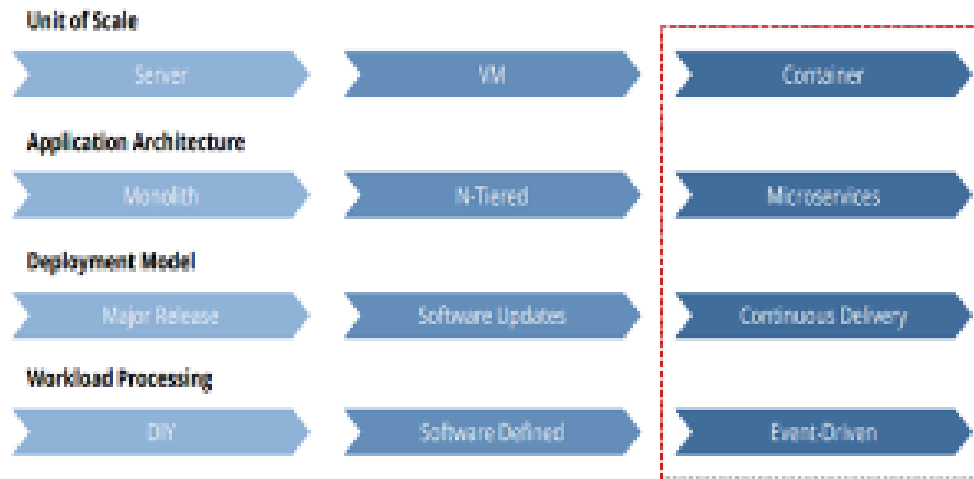


그림. 클라우드로 인한 기술 패러다임의 변화

On-premise 는 서버를 구매하고 운영체재를 설치하고, 배포환경을 구축하는데 수 개월이 소요된다. 서비스의 수요 예측에 실패하면 새로운 서버를 추가 구매하거나 서버 사양을 높여야 하는데 여기에 소요되는 시간도 수 개월이다. 그래서 이런 상황을 막고자 주로 오버스펙의 사양을 구매하게 되고 이로 인해 낭비되는 자원이 많다.

가상머신은 서버 가상화 기술로 On-premise에 비해 서비스 배포 환경을 구축하는데 소요되는 시간을 수 분내로 줄일 수 있고 인프라 제반 관리에 드는 비용도 줄일 수 있다. 또한 필요에 따라 가상서버 수나 사양을 조절하여 자원 효율성을 증대할 수 있다.

하지만 가상머신은 운영체재 설정, 보안 패치 등의 관리가 필요한데 서버가 늘어날수록 관리가 어렵고 시스템 부하에 따라 자동으로 서버를 추가 및 제거하는 오토스케일링(Auto Scaling) 기능으로는 꾸준한 트래픽이 증가하는 서비스는 해결이 되겠지만 불예측적이고 폭발적인 트래픽을 처리하기에는 역부족이다. 트래픽이 몰릴 때 서버가 자동으로 추가되어도 수분의 시간이 소요되고 트래픽을 감당할 만큼의 서버가 추가되기도 전에 트래픽이 감소할 것이기 때문이다.

컨테이너는 운영체재 가상화 기술로 물리서버나 가상머신을 더 효율적으로 사용할 수 있고 수 초내로 배포가 가능하다. 서비스(서버)가 추가될 때마다 반복적으로 작업했던 서버 초반 설정도 불필요하고 운영체재 패치 작업도 간소화되며 짧은 배포 시간과 오토스케일링 기능으로 불규칙적이고 폭발적인 트래픽도 유연하게 처리할 수 있다. 하지만 컨테이너 기술은 상대적으로 어렵고 복잡해서 컨테이너 기술을 보유한 인재가 없는 곳이라면 컨테이너 기술 도입은 부담이 될 수밖에 없다.

서버리스 컴퓨팅은 프로그래밍 언어 런타임 가상화 기술로 서버 인프라, 운영체재 설정 등의 관리 요소는 클라우드 서비스 업체(CSP)에서 담당하고 **사용자는 코드에만 집중할 수 있어 생산성이 증가**하고 시스템 관리 비용을 최소화할 수 있다. 또한 초당 수 천개의 요청까지 자동으로 확장되고 사용한 컴퓨팅 시간에 대해서만 비용이 발생하여 가상머신이나 가상머신 기반 컨테이너에 발생하는 유후 시간 비용이 없어 비용 효율적이다.

간단히 표현하자면 아래 표와 같다.

특징	On-premise	Virtual Machine	Container	Serverless Computing
배포 시간	Week	Minutes	Seconds	Milliseconds
관리	Everything	OS Middleware Source	Middleware Source	Source
확장	Order Manual	Manual	Manual	Auto
비용	one-time	Hour or Minute	Hour or Minute	Second Execute times
고가용성	이중화 필요	이중화 필요	이중화 필요	Auto
추상화	해당없음	하드웨어	OS	프로그래밍언어런타임

표. 소프트웨어 구축 환경 별 비교

2 서버리스 컴퓨팅 정의

AWS에서 2014년 Lambda 서비스를 선보이며 업계에서 서버리스 컴퓨팅 Serverless Computing 이란 용어를 사용했다. 서버리스 컴퓨팅이란 실제로 서버가 존재하지 않는다는 의미가 아니라 사용자가 서버 설정 없이 함수코드만으로 어플리케이션과 서비스를 구축하고 실행할 수 있음을 의미한다. 함수 서비스라 FaaS^{Functions as a Services} 또는 Function PaaS 로 불리기도 한다.

서버리스 컴퓨팅을 제공하는 CSP에서는 기존 서버에 필요한 서버 프로비저닝, 패치, 확장 등의 관리를 수행하여 사용자가 자신의 서비스 로직에 집중할 수 있기 때문에 생산성을 향상시킬 수 있다. 또한 코드를 실행한 만큼만 비용이 발생하여 운영 비용을 절감할 수 있다.

서버리스 컴퓨팅은 퍼블릭 CSP의 서비스를 이용하거나 오픈소스 플랫폼으로 자신의 서버에 직접 설치하여 사용할 수 있다.

대표적인 서버리스 컴퓨팅 서비스 업체들은 자신의 서비스들을 아래와 같이 소개한다.

- AWS Lambda

서버에 대한 걱정 없이 코드 실행. (생산성 향상)

사용한 컴퓨팅 시간에 대해서만 비용을 지불합니다. (사용한 만큼 비용 지불)

이벤트에 응답하여 코드를 실행하고 자동으로 기본 컴퓨팅 리소스를 관리하는 서버 없는 컴퓨팅 서비스입니다. (이벤트 드리븐, 자동 확장, 서버리스)

- Azure Functions

서버리스 아키텍처로 앱을 더 신속히 구축. (서버리스)

이벤트 중심의 서버리스 컴퓨팅 환경으로 개발을 가속화하십시오. (이벤트 드리븐, 생산성 향상)

필요에 따라 확장*하고, 사용한 만큼만 비용을 지불하면 됩니다. (사용한 만큼 비용 지불)

- Google Cloud Functions

이벤트 기반 서버리스 컴퓨팅 플랫폼 (이벤트 드리븐, 서버리스)

클라우드에서 코드를 실행하는 가장 간편한 방법 (생산성 향상)

자동확장, 우수한 가용성, 내결함성. 프로비저닝, 관리, 패치 또는 업데이트가 필요한 서버 없음 (자동 확장)

코드를 실행하는 만큼만 지불. 클라우드 서비스 연결 및 확장 (사용한 만큼 비용 지불)

- IBM Cloud Functions

수신 이벤트에 대한 응답으로 함수를 실행하고 사용하지 않을 때는 비용이 들지 않는 FaaS 플랫폼입니다. (이벤트 드리븐, 사용한 만큼 비용 지불)

별도의 서버 설정, 구성이나 유지보수 없이도 코드 작성과 최상의 솔루션 빌드에 집중할 수 있습니다. (서버리스, 생산성 향상)

액션 인스턴스는 요구에 정확히 맞게 스케일링된 후 사라집니다. (자동 확장)

CSP들이 공통적으로 설명하는 내용은 서버리스, 이벤트 드리븐**, 생산성 향상, 자동 확장, 사용한 만큼 비용 지불로 동일하다. 이 5가지가 서버리스의 큰 특징이라 할 수 있다.

*) Azure에서 '필요에 따라 확장하고' 라는 애매한 표현을 사용한 이유는 서버리스 컴퓨팅의 비용 모델이 사용한 만큼 과금되며 자동 확장되는 Consumption plan과 자신의 VM 상에서 지정한 만큼 확장되는App service plan 2가지이기 때문이다. 이 비용모델에 대한 내용은 5.1.3 가격 에서 자세히 작성한다.

**) 이벤트 드리븐^{Event-Driven}이란 이벤트에 반응하여 동작을 처리하는 것을 의미한다.

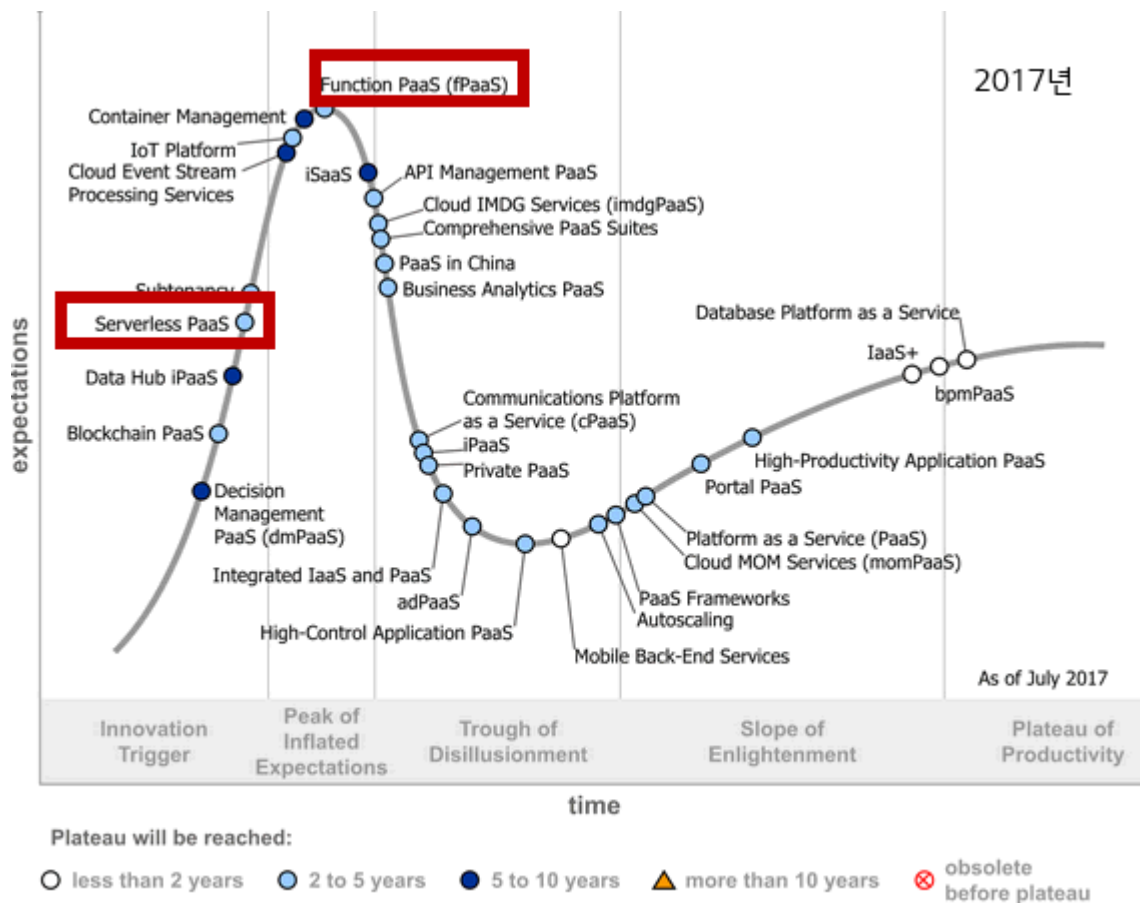
3 서버리스 컴퓨팅 현황 분석

세계 최대 IT리서치 기관인 Gartner의 작년과 올해의 10대 PaaS Hype Cycle을 비교해보면 Function PaaS (서버리스 컴퓨팅 플랫폼)이 한창 핫한 기술에서 성숙/구체화 기술로 넘어가려는 것을 확인할 수 있다.

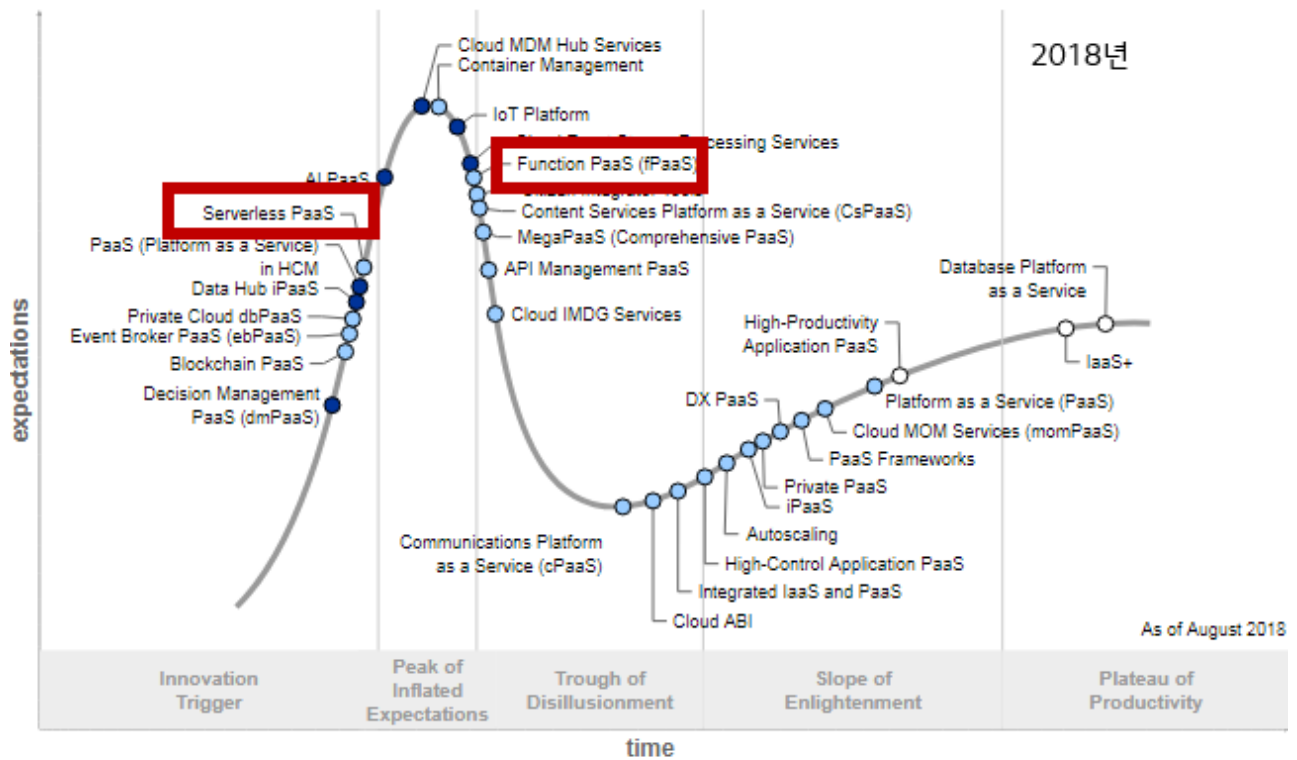
2014년에 AWS에서 서버리스 컴퓨팅 서비스인 Lambda를 처음 발표했고 2016년에 MS Azure와 Google Cloud, IBM이 서버리스 컴퓨팅 서비스를 발표하면서 서버리스 컴퓨팅의 관심이 높아졌다. 2017년에는 Oracle과 Pivotal이 오픈소스 서버리스 컴퓨팅 프로젝트를 진행했고 2018년에는 Redhat이 오픈소스 서버리스 컴퓨팅 서비스를 발표했다. 또한 서버리스 컴퓨팅을 관리할 많은 도구들이 생겨나며 계속해서 발전하고 있다.

서버리스 컴퓨팅은 생산성 증대, 이벤트에 의한 자동 확장, 고가용성, 비용 절감 등의 매력적인 장점이 있는 반면에 구동 시간 제한, 동시 실행 수 제한, 공급업체 종속성, 부족한 문서, 신기술이 가진 불안정함 등의 무시할 수 없는 단점들이 존재하여 아직까지 크게 확산되지는 않았지만 차츰 안정화되고 관리 툴들이 계속해서 발전하여 2021년 안에 성숙/구체화 기술이 될 것이라 예상된다.

서버리스 컴퓨팅의 발전으로 궁극적으로 소프트웨어의 구축의 패러다임이 될 수 있기에 이에 대한 대비가 필요하고 서버리스 컴퓨팅이 적합한 분야에는 적극 도입하거나 검토가 필요하다.



© 2017 Gartner, Inc.



Plateau will be reached:

○ less than 2 years ● 2 to 5 years ● 5 to 10 years ▲ more than 10 years ⊗ obsolete before plateau

© 2018 Gartner, Inc.

그림. Gartner Hype Cycle for PaaS, 2017/2018

4 서버리스 컴퓨팅 동작 방식

서버리스 컴퓨팅은 아래와 같은 방식으로 동작한다.

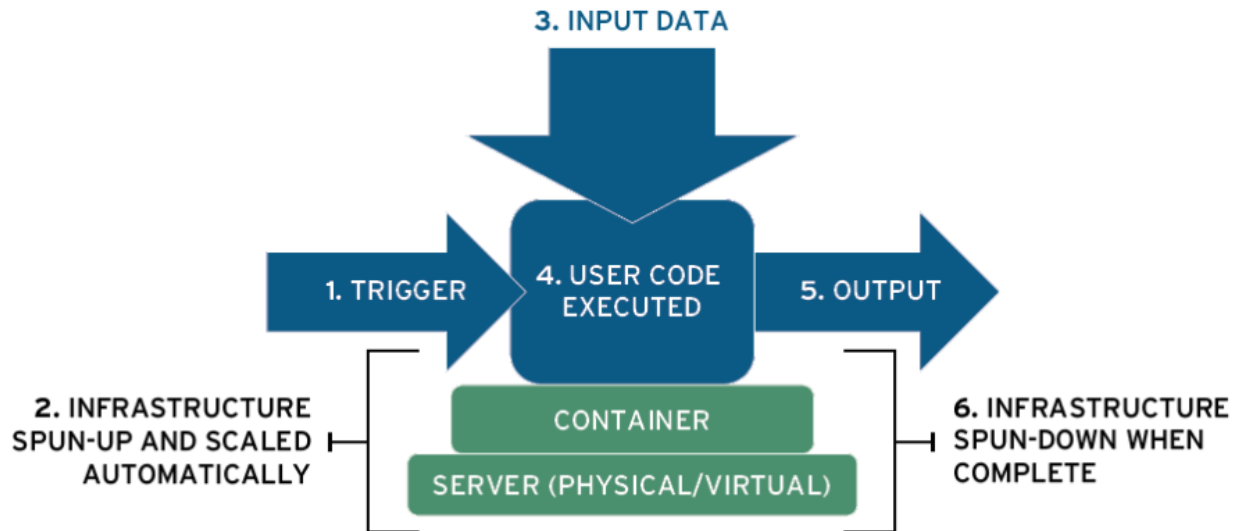


그림. 서버리스 컴퓨팅 동작원리

1. 이벤트 트리거에 의해 서버리스 컴퓨팅을 호출한다.
2. 서버리스 컴퓨팅이 설치된 서버에서 함수를 처리하기 위한 컨테이너를 할당한다.
3. 특정 스토리지에서 함수의 코드를 무결성 체크 후 할당된 컨테이너에 등록하고 이벤트 트리거에서 전달한 이벤트 소스와 함께 함수를 호출한다.
4. 함수 코드를 수행한다.
5. 결과값을 내보낸다.
6. 일정 시간이 지나면 할당된 컨테이너를 회수한다.

소프트웨어 아키텍처 패러다임 변화(see page 4)

서버리스 컴퓨팅 정의(see page 6)

서버리스 컴퓨팅 현황 분석(see page 7)

서버리스 컴퓨팅 동작 방식(see page 9)