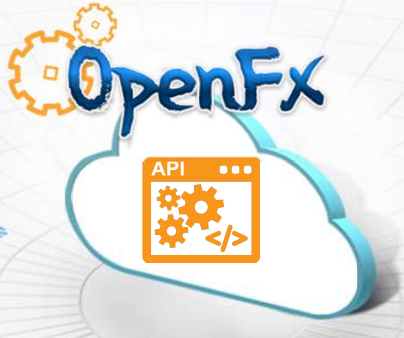
# 서버리스 컴퓨팅 기술 및 동향

## CBI Cloud, Bigdata, Iot 융합기술 세미나




2018. 11. 27 (화)

전자부품연구원
팀장 손 재 기

# 목차

Ⅰ. 서버리스 컴퓨팅 ?

Ⅱ. 서버리스 컴퓨팅 기술동향

Ⅲ. 서버리스 컴퓨팅의 향후 전망

CHAPTER

1

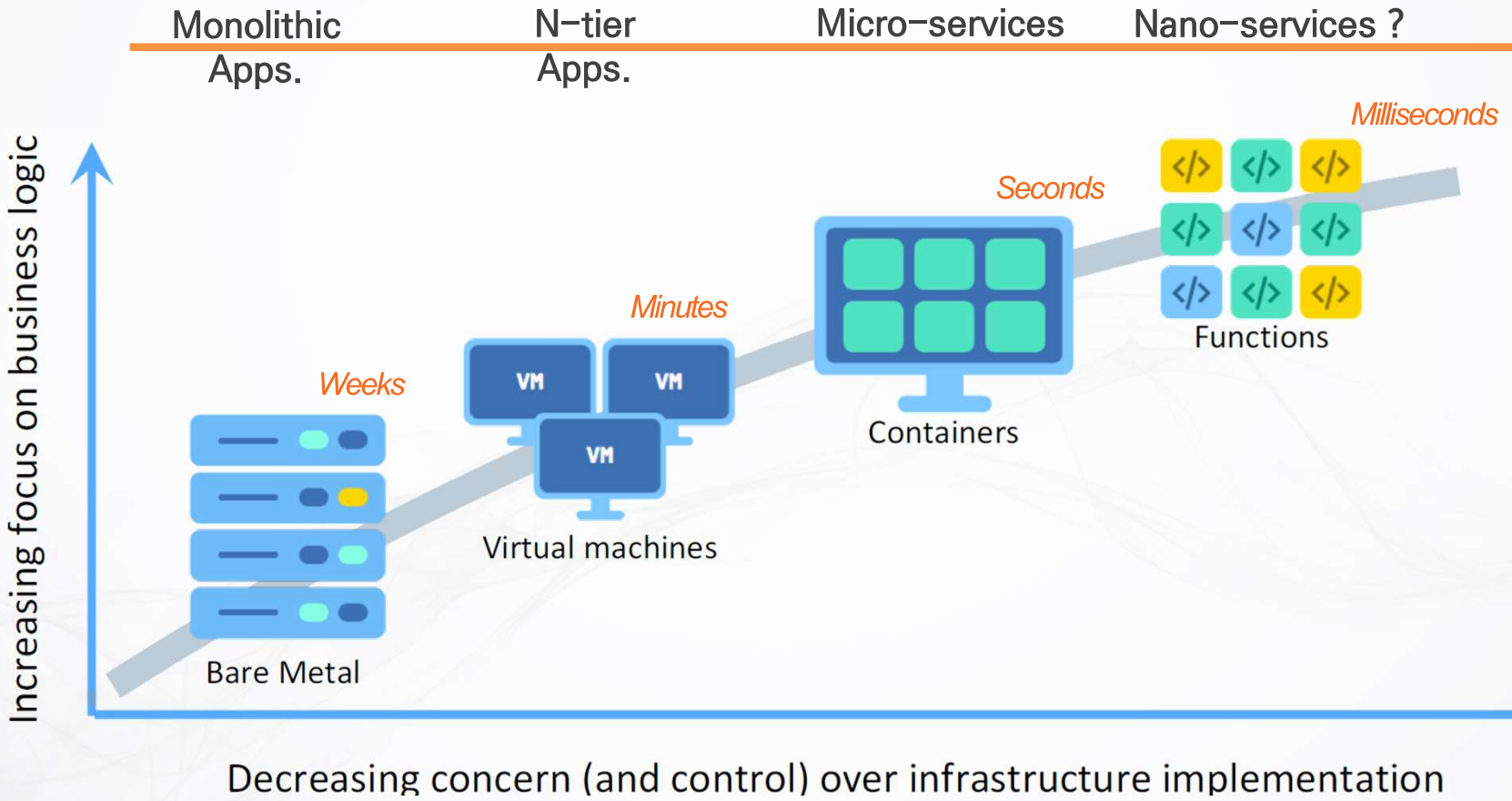
# 서버리스 컴퓨팅 ?


클라우드 참 좋은 데...



# 1. 클라우드 컴퓨팅의 진화

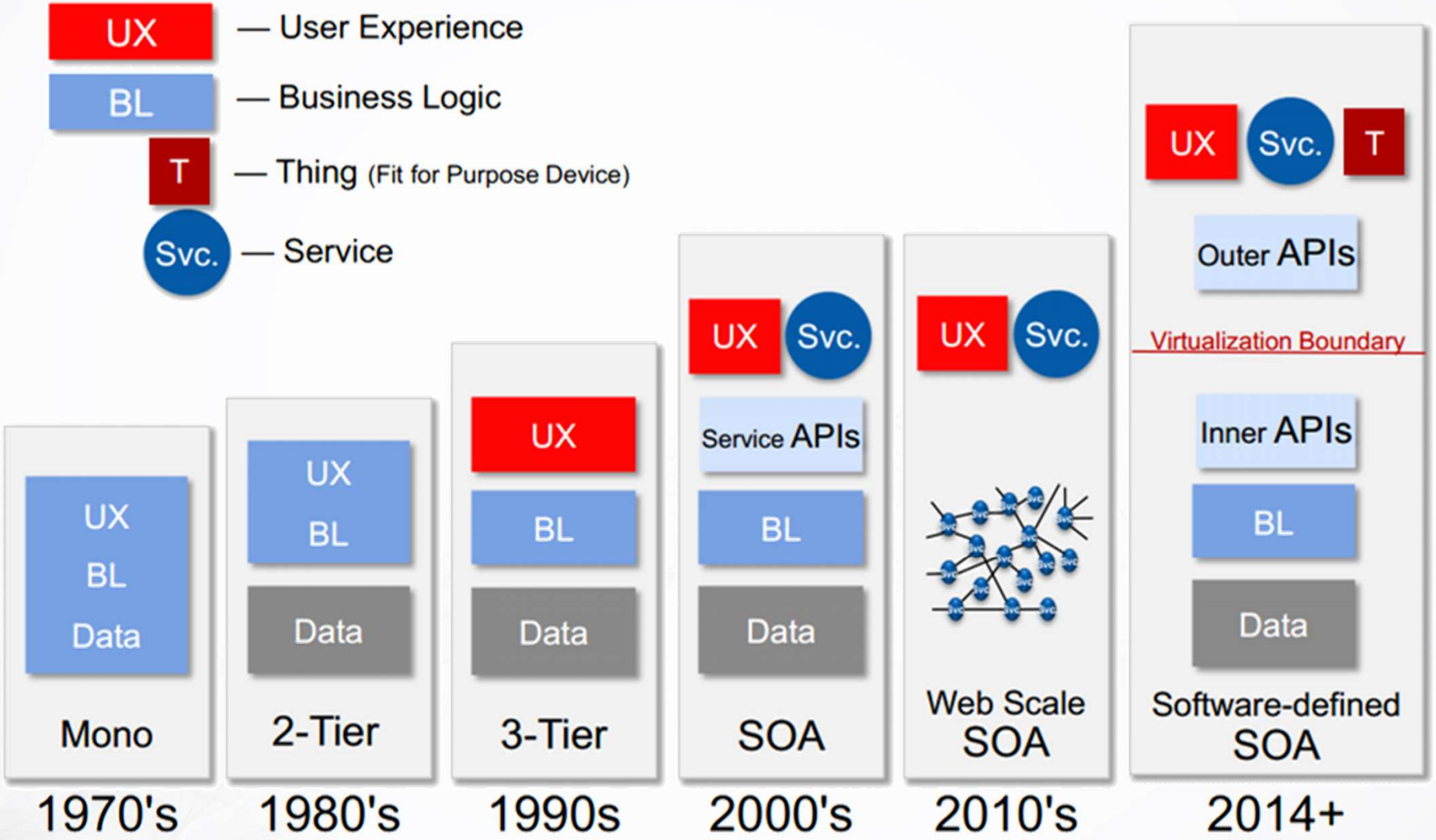
✔ 최근 10년 동안 클라우드 기술 혁신이 일어남에 따라 민첩한 비즈니스에 큰 영향을 미치는 애플리케이션 개발 및 서비스 배포에 대한 새로운 패러다임이 등장




*Source: Mark Peek et al., “State of Serverless”, KubeCon 2017.


Copyright © 2018 Korea Electronics Technology Institute

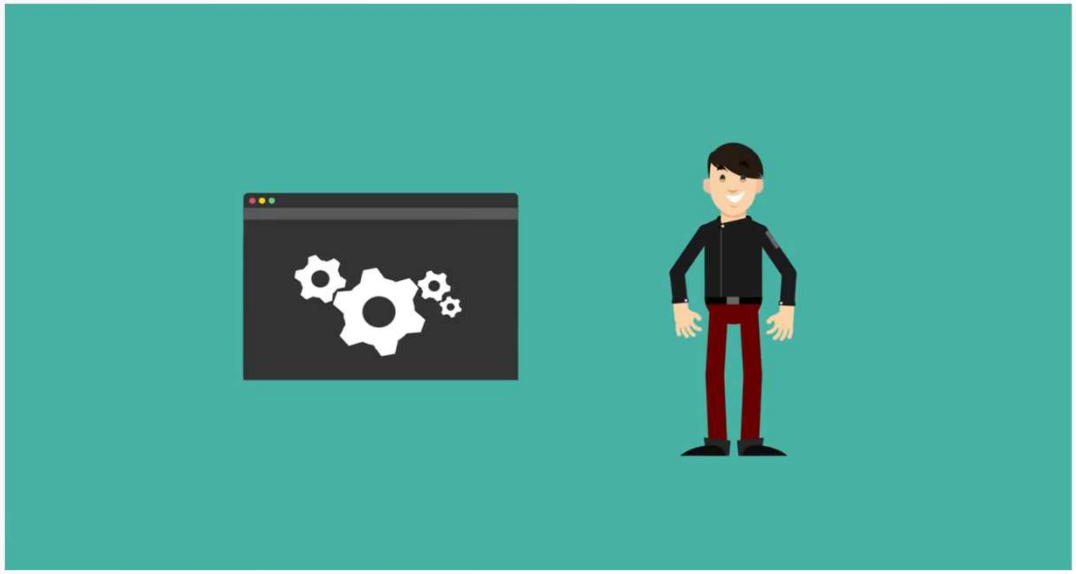
# 2. 서비스 아키텍처의 변화

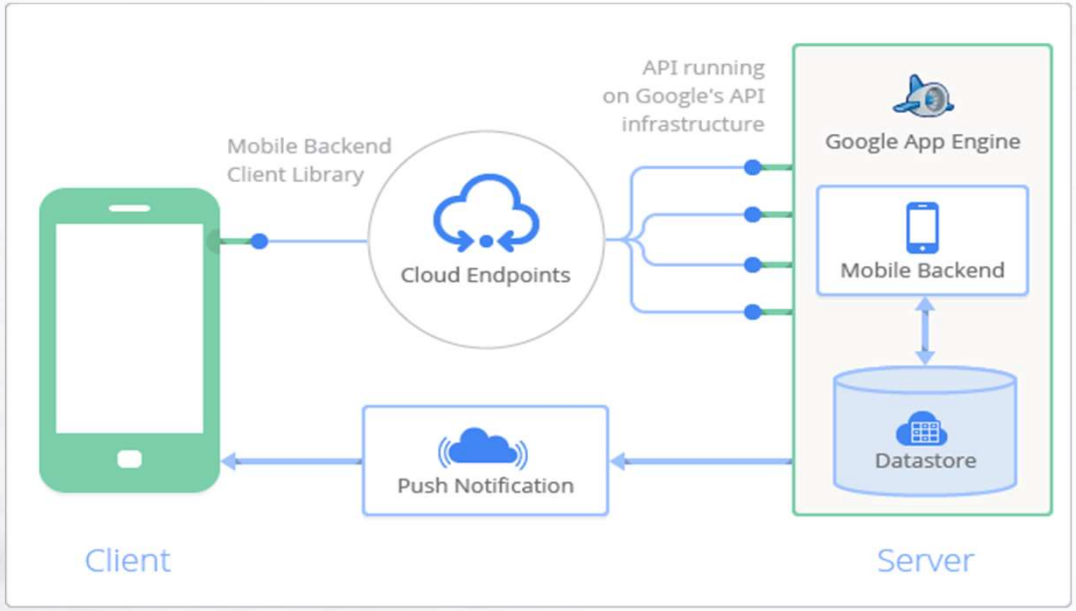



*source: Gartner, 2014


Copyright © 2018 Korea Electronics Technology Institute

# 3. 관련 기술

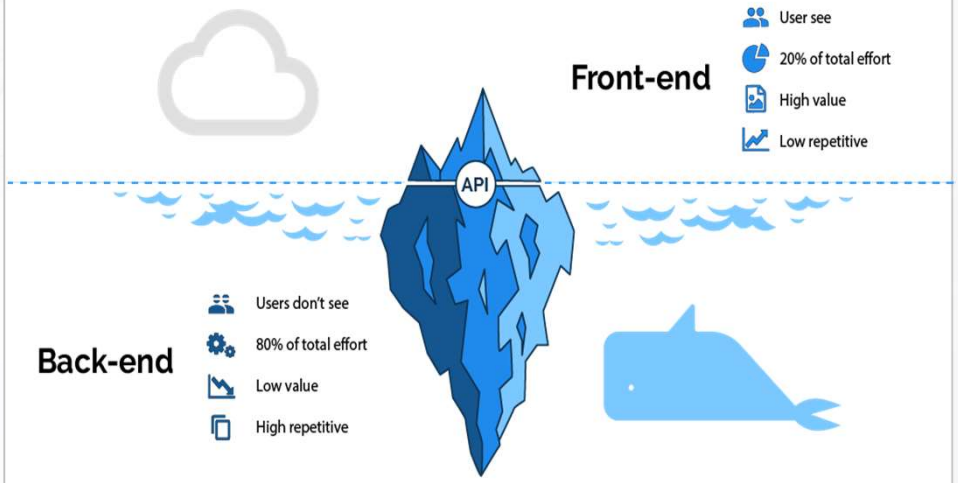





*source: https://cloud.google.com/developers/articles/mobile-backend-starter

## Backend as a Service BaaS

- 클라우드 컴퓨팅 서비스 모델 중 하나
- 모바일에 특화된 푸시 알람, GPS, 소셜 서비스 연계 기능 등을 포함하여 대부분의 필요 기능을 백엔드 클라우드에서 제공
- 통상 Mobile Backend as a Service$^{MBaaS}$라고도 함
- 대표적인 MBaaS는 Google의 Firebase



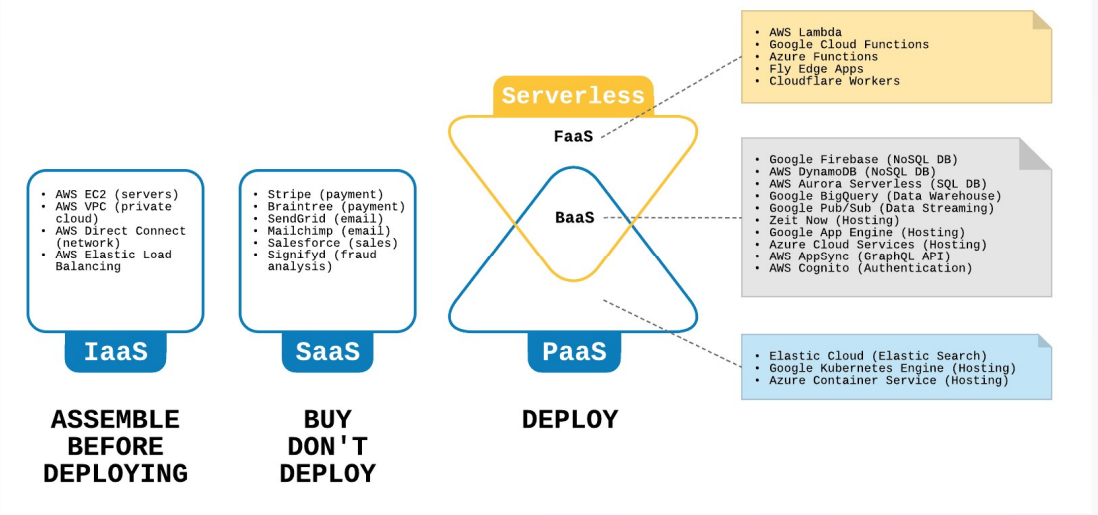


Copyright © 2018 Korea Electronics Technology Institute

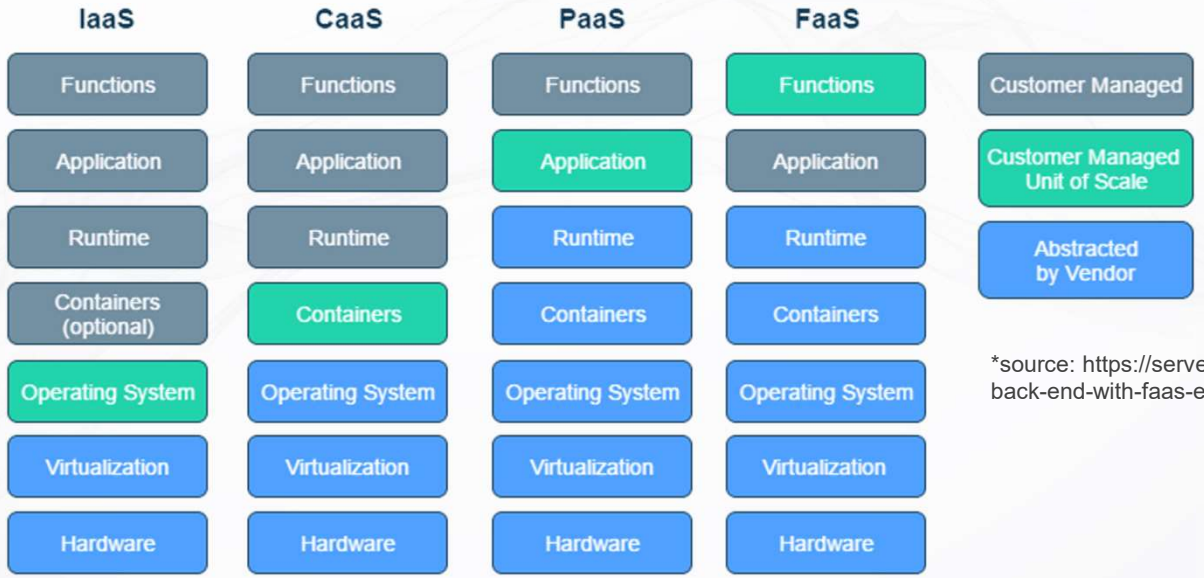
# 3. 관련 기술

## Function as a Service FaaS

- FaaS는 이벤트 중심의 컴퓨팅 환경 제공
- 개발자인 이벤트 또는 HTTP 요청에 의해 트리거되는 함수를 개발, 응용 프로그램 코드를 실행, 관리
- 작은 단위의 코드함수를 FaaS에 구현, 서버 또는 기타 기본 인프라 관리 필요없이 확장
- 특징: 상태를 갖지 않음stateless




*source: https://hackernoon.com/the-serverless-series-what-is-serverless-d651fbacf3f4




*source: https://serverless.zone/abstracting-the-back-end-with-faas-e5e80e837362

<FaaS의 포지셔닝>


Copyright © 2018 Korea Electronics Technology Institute

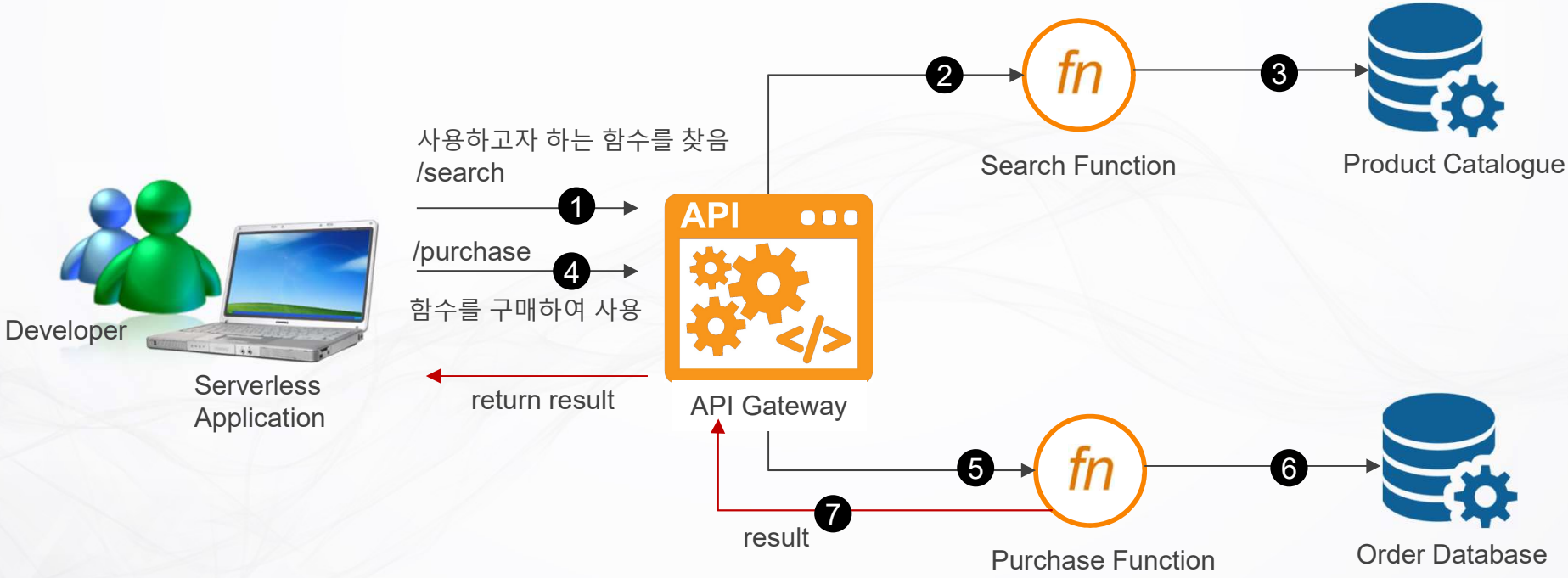

SERVERLESS
COMPUTING
Function-as-a-Service

# 4. 서버리스 컴퓨팅 ?

## Serverless != No Server

**서버리스Serverless는 특정 서버 없이 모든 규모의 이벤트에 대한 코드Function를 실행, 응답하는 클라우드 컴퓨팅 방식**

※ 요청이 있을 때만 필요한 해당 함수를 ms단위로 실행, 요청이 많은 경우 Auto-scaling → Utilization + Scalability





Copyright © 2018 Korea Electronics Technology Institute

# 5. 서버리스 컴퓨팅 정의

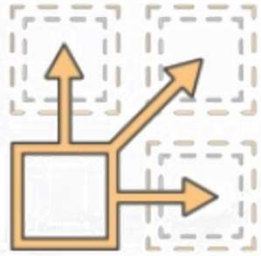
## [ Serverless* → 완전한 Server Lease ? ]

* "서버리스"라는 용어는 ?

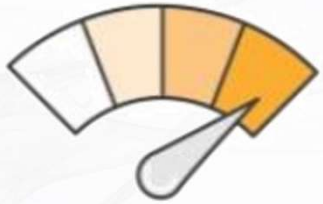
aws 람다Lambda 서비스에서 인기를 얻은 마케팅 전문 용어

### ✔ 서버리스 컴퓨팅 플랫폼은 FaaS 또는 BaaS 중 하나 또는 둘 모두를 제공*

- 아주 세분화되고, 훌륭한 Micro 단위의 PaaS 시스템
- FaaS는 서버리스 개념의 일부



Provisioning and Utilization



Operations and Management



Scaling

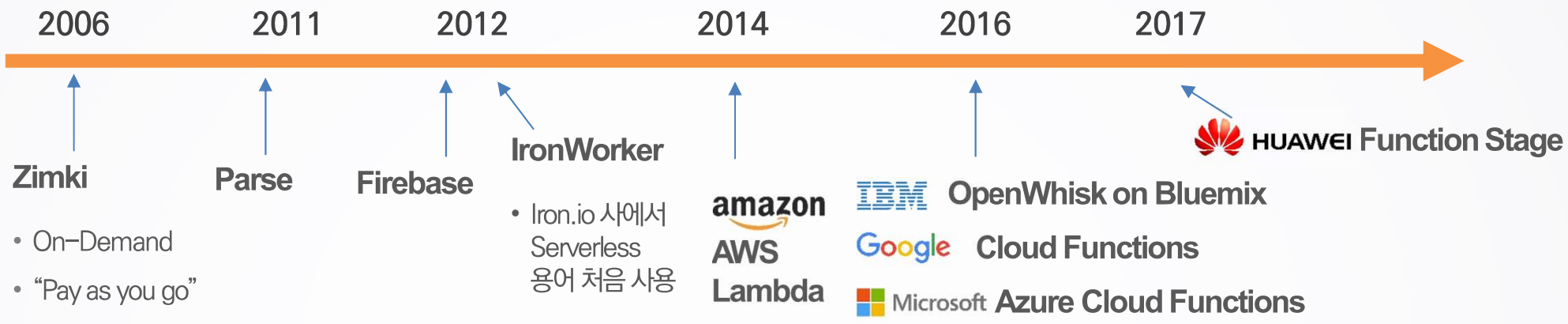


Availability and Fault Tolerance

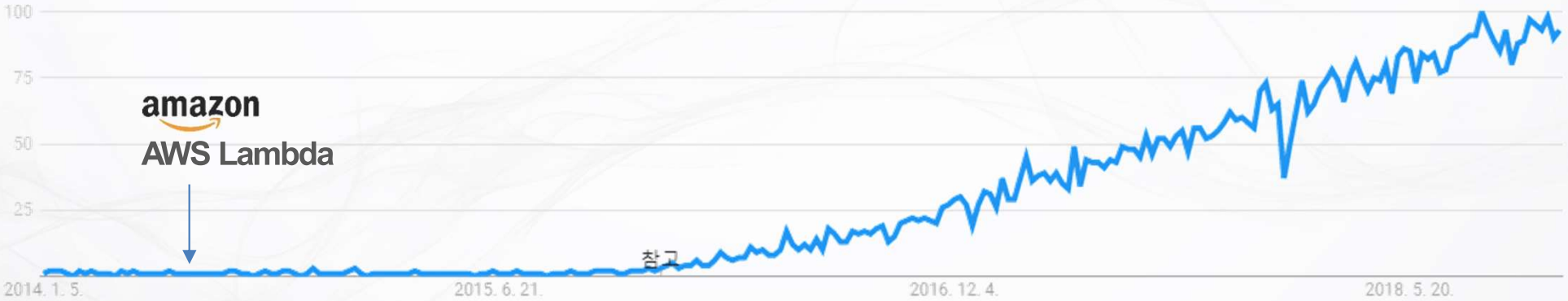
*source: CNCF, 2018


Copyright © 2018 Korea Electronics Technology Institute

# 6. 서버리스 컴퓨팅의 역사




Serverless의 Google Trends (2014. 1 ~ 2018. 11)




*source: Google Trends, 2018. 11


Copyright © 2018 Korea Electronics Technology Institute

# 7. 서버리스 컴퓨팅 기술의 응용




## Web Applications

- Static Websites
- Complex Web Apps.

RocketPunch





## Backends Systems

- Apps. & Services
- Mobile
- IoT



## Big Data Processing

- Real-time Apps.
- MapReduce
- Batch

FireEye






## Chatbots

- Powering chatbot logic

slack



## A.I Applications

- Powering voice-enabled Apps.
- A.I Skills Kit






## Autonomous IT

- Policy engines
- Extending Cloud Services
- Infrastructure Management

AMOREPACIFIC

- Needs : 글로벌 시장(뷰티케어 등) 공략 본격화 등 폭증 수요 대응하는 인프라 구축
- Benefit : 신규 글로벌 서비스 런칭(x2이상 시간향상), 비용절감(80% 이상), 해외사업 운영 안정성 확보

NEXON

서버리스 게임 아키텍처 도입, 모바일 게임시장의 특성(서비스의 짧은 수명)을 고려한 빠른 개발, 비용절감을 개선

MBC

- Needs : 실시간 모바일 시청 수요에 대한 라이브 스트리밍 지연시간 단축
- Benefit : 스트리밍 지연시간 1/3 단축(기존 30s→10s), 트랜스코딩 솔루션 및 리소스 40% 절감

*source: AWS


Copyright © 2018 Korea Electronics Technology Institute

# 8. 서버리스 컴퓨팅의 특징

| | AS-IS (기존 클라우드) | TO-BE (서버리스 클라우드) | 비고 |
|---|---|---|---|
| 서버 | 가상머신VM이나 컨테이너 필요 | 서버 준비나 관리 불필요 | 관리 포인트 × |
| 과금 | 몇달 또는 1년 단위 | 함수 실행시 100ms 단위 | 비용 ↘ |
| 확장성 | 물리적인 수동 확장 | 사용량 기반 자동 확장 | 관리 포인트 × |
| 가용성 및 장애 방지 | 기본적인 인프라에 대해 지원 | 완벽 지원 | 관리 포인트 × |
| 개발기간 | 장기 개발 사이클 | 손쉽게 개발 가능 | 개발기간 ↘ |
| 협업 | 상대적으로 복잡함 | 매우 쉬움 | 업데이트 ↗ |


Copyright © 2018 Korea Electronics Technology Institute

# 9. 서버리스 컴퓨팅의 단점



Vendor lock-in



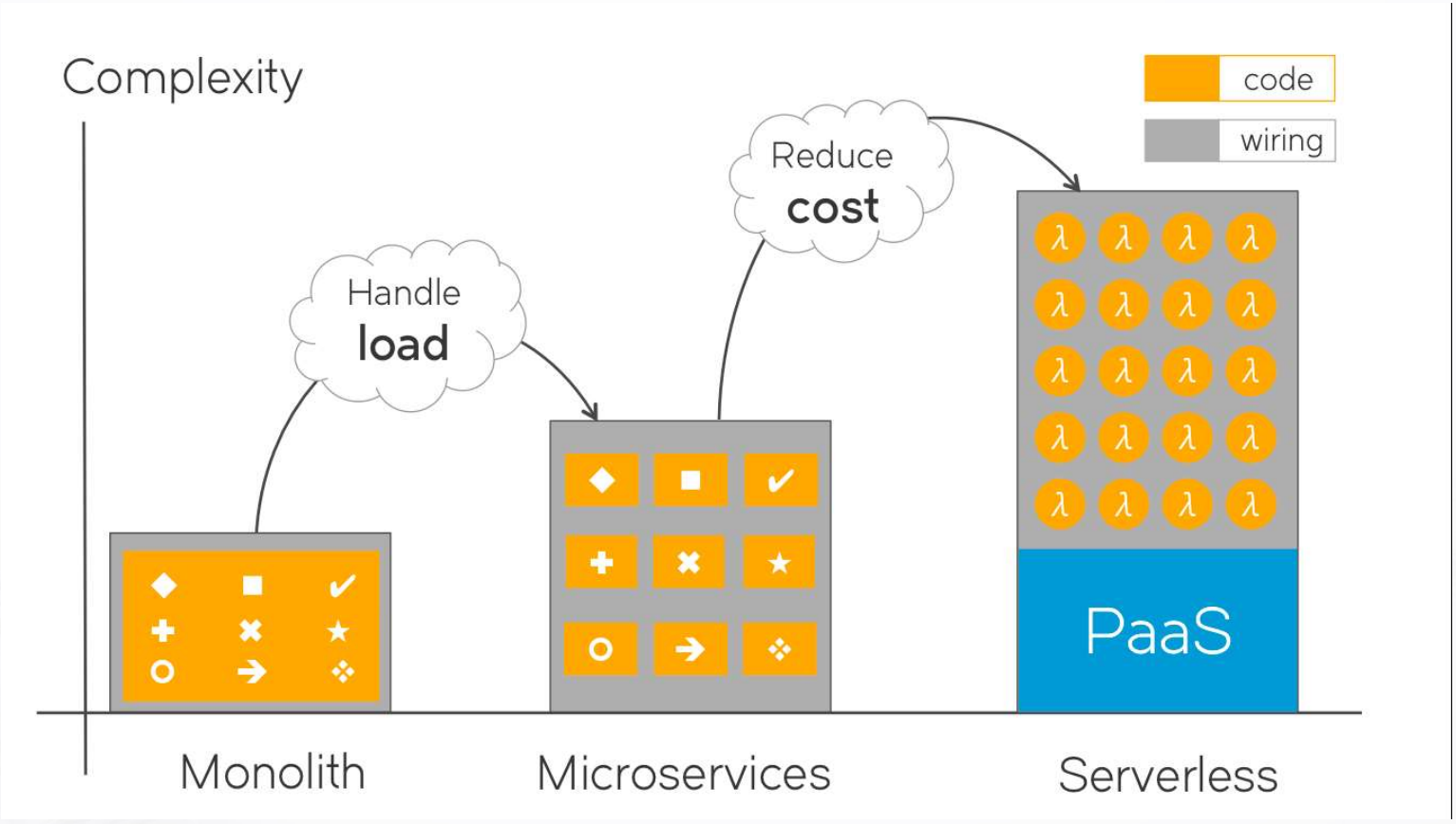

Testing



Debugging




Delayed Start-up Cold Start




*source: https://medium.freecodecamp.org/serverless-is-cheaper-not-simpler-a10c4fc30e49


Copyright © 2018 Korea Electronics Technology Institute

CHAPTER

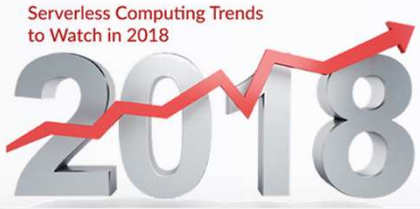
# 2

# 서버리스 컴퓨팅 기술동향

# 1. 글로벌 Overview

(주요 동향) 세계적으로 서버리스 컴퓨팅 기술은 아마존웹서비스AWS의 “Lambda”, MS의 “Azure Function”, 구글의 “Cloud Function”, IBM의 “OpenWhisk”라는 명칭으로 기술공개

- ✔ 인프라 관리자의 리소스를 최소화할 수 있는 형태의 아키텍처 설계,
- ✔ 무한한 확장성을 제공Auto Scaling하는 형태의 기술을 주요 골자로 함




01 클라우드의 사용증가에 따라 무한한 확장성을 가지면서도 리소스를 탄력적으로 사용할 수 있는 개념의 기술이 요구되고 있으며, 사용자의 입장에서 이러한 클라우드 관리 시스템의 설치, 운영, 관리에 최소한의 리소스만 사용하는 형태로 진화

02 신규 인프라의 설치/운용/확장/배포 등에 소요되는 비용 감소와 Fast-Following을 위한 서버리스Serverless 컴퓨팅 기술이 활발히 개발, 최근 기술 프로토타입 서비스의 구현과 비즈니스 모델 테스트, 모바일 개발 환경을 지원하는 기술로 활용되고 있음


Copyright © 2018 Korea Electronics Technology Institute

# 2. 기술동향 및 사례 ▶ AWS AI Ecosystem Alexa + lambda



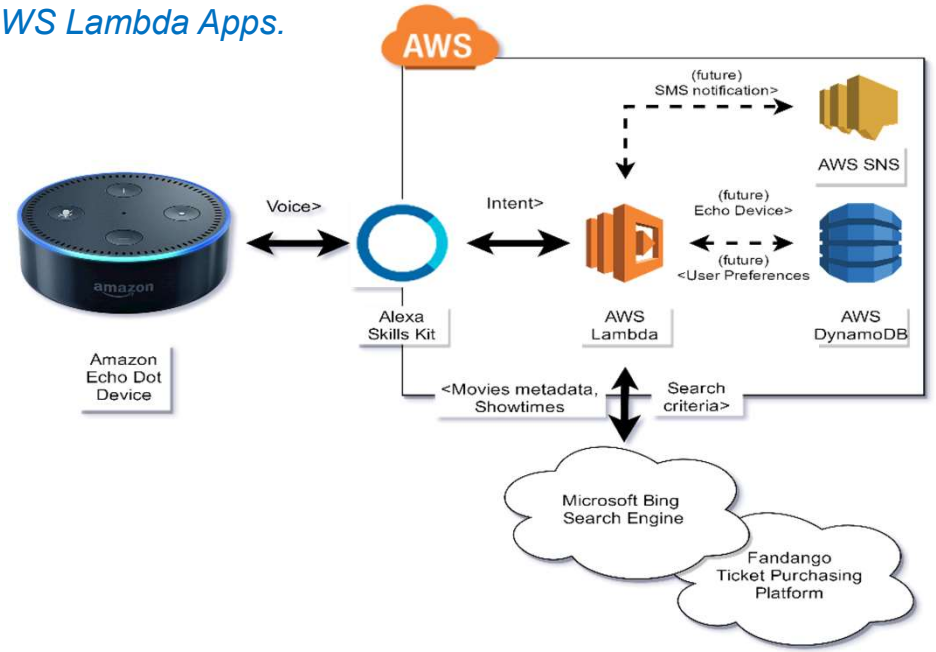








**AWS Lambda Apps.**




## 서버리스 생태계의 확산

### amazon Lambda를 통한 Alexa 서비스 생태계 구축

- amazon : '15년에 ASK(Alexa Skills Kit)라는 SDK를 공개. 개발자가 Alexa의 음성인식 기반 서비스와 특정 Apps.의 연동을 편리하게 구현 할 수 있도록 함
- Alexa 연동 앱: ASK가 공개된
  '15년: 10여 개 → '16년: 1,000개 → '17년: 10,000개 이상
- 부스 하나 없이 Alexa가 CES를 점령: Backend = Lambda

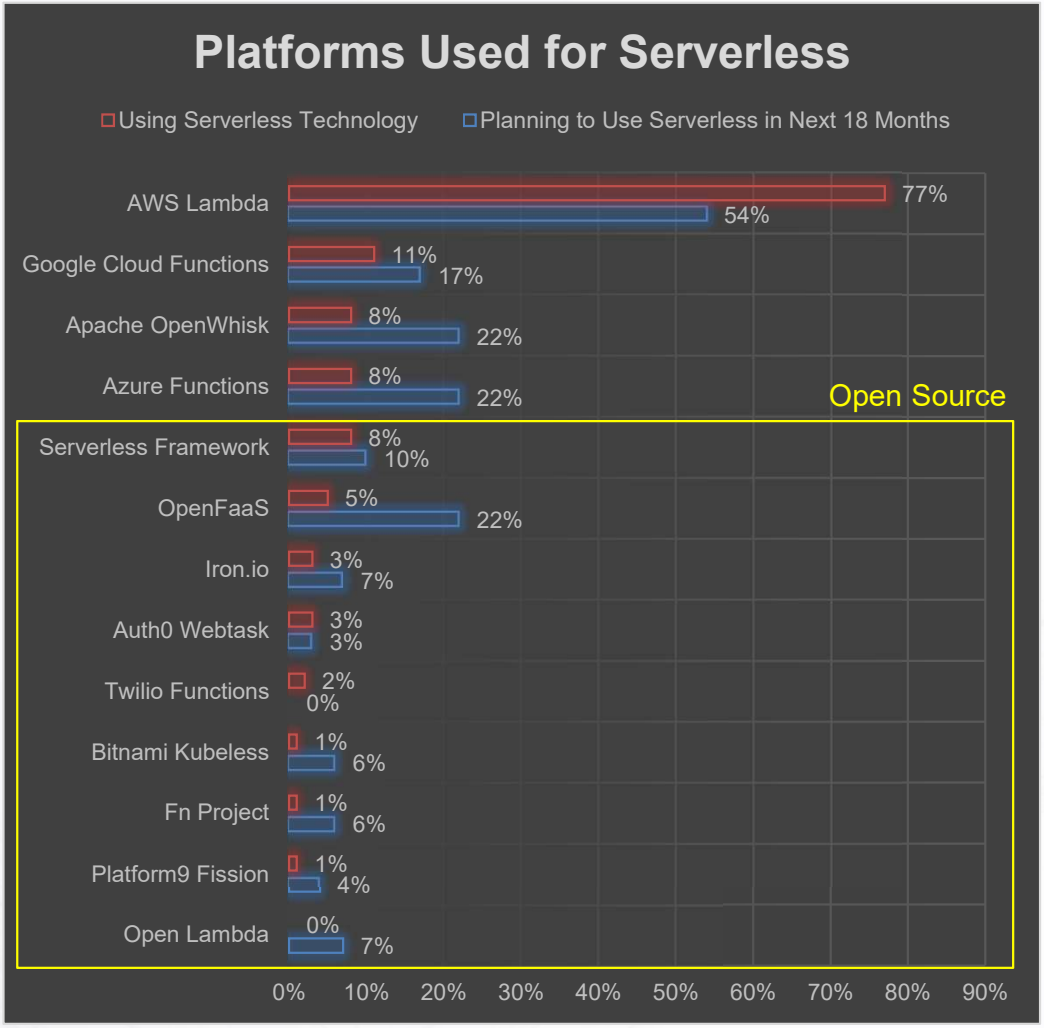
### 서버리스 생태계 선점을 위한 경쟁 심화

- amazon Lambda, Microsoft Azure Functions, Google Functions 등 다양한 도메인(AI, Mobile Web, Apps 등)에 자사의 서버리스 플랫폼을 확산하기 위해 여러 기업과 파트너쉽

**시사점** 세계 선도기업인 Amazon을 시작으로 AI, Big Data Service 등을 통해 **차기 Cloud 생태계 플랫폼 선점 전쟁**


Copyright © 2018 Korea Electronics Technology Institute

# 3. Open Source 진영의 동향




*source: The New Stack Analysis of Cloud Native Computing Foundation survey conducted in Fall 2017

## 서버리스 기술의 Open Source 가속화

amazon의 성공에 힘입어 다양한 Open Source 등장

- serverless, OPENFAAS, fn, Open Lambda 등이 공개되어 생태계를 형성 중
- 기본적인 Core Platform, CLI 등을 제공하나, 아직까지 IDE, Debugging 기능, 안정화 등의 문제점
- OPENFAAS의 경우, 타 Open Source에 비해 유연하고, 리눅스, 윈도우 컨테이너를 동시에 지원
  - Kubernetes, Docker Swarm을 위한 서버리스 프레임워크

APACHE OpenWhisk™ Open Source → Commercial

- APACHE 프로젝트로 시작하여, IBM 에서 상용화
- Open Source의 약점인 개발도구SDK, IDEIntegrated Development Environment, Debugging 도구, 안정화 등을 개선

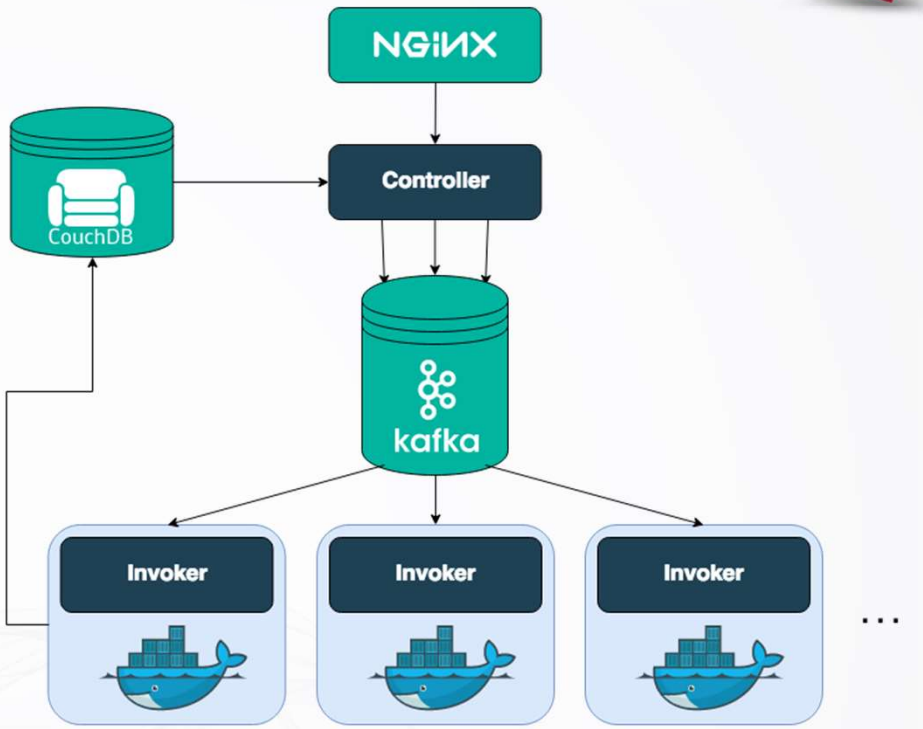
**시사점** 최근 다양한 Open Source가 공개되고 있으며 특히, OpenFaaS 진영의 비약적인 발전으로 오픈 생태계 구축 중


Copyright © 2018 Korea Electronics Technology Institute

# 4. OpenWhisk

APACHE OpenWhisk™

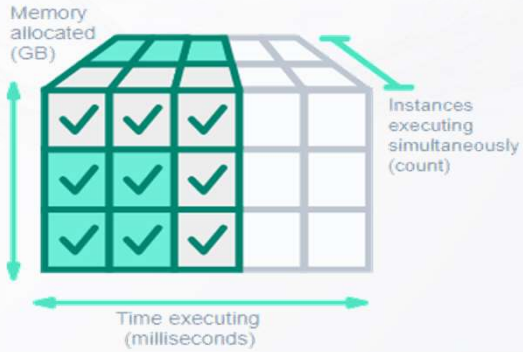
- APACHE OpenWhisk™는 이벤트에 대해 Function을 실행시키는 오픈소스 클라우드 서버리스 플랫폼으로, GitHub에 공개된 지 1년 반 만에 400여 번 포크Fork되는 등 활기찬 오픈소스 프로젝트
- IBM의 서버리스 서비스를 위한 내부 프로젝트로 초기 개발 후, 2016년 2월 오픈소스로 공개되었고, 11월부터 아파치 인큐베이션을 시작함
- 지원언어: 액션에 실행 가능한 언어로는 Node.js v6, Swift v3, 파이선Python, 자바Java, PHP가 기본 지원됨. 실제 액션이 컨테이너에서 실행되기 때문에 도커 이미지도 액션으로 지정이 가능하니, 사실상 리눅스Linux에서 실행 가능한 모든 언어 지원
- 아키텍처: 오픈소스 컴포넌트와 프레임워크로 구성됨. 기본적으로 동기Sync, 비동기Async 실행방식을 모두 지원
- IBM **Cloud**는 OpenWhisk를 FaaS 형태로 제공되는 IBM Cloud Functions 서비스 제공




<Apache OpenWhisk의 실행 아키텍처>

**과금구조**

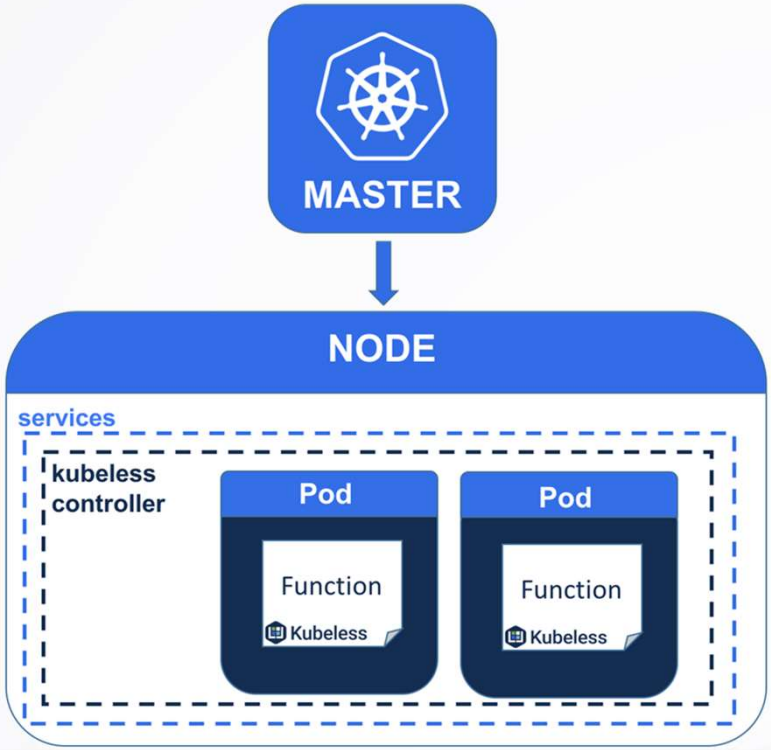
- 실행하는 시간과 사용하는 메모리를 곱한 단위인 GBs로 과금.
- 월 400,000 GBs가 매월 무료로 제공되니 간단한 API는 무료로 사용 가능. 실행에 걸리는 시간, 할당한 메모리, 실행 개수에 비례함



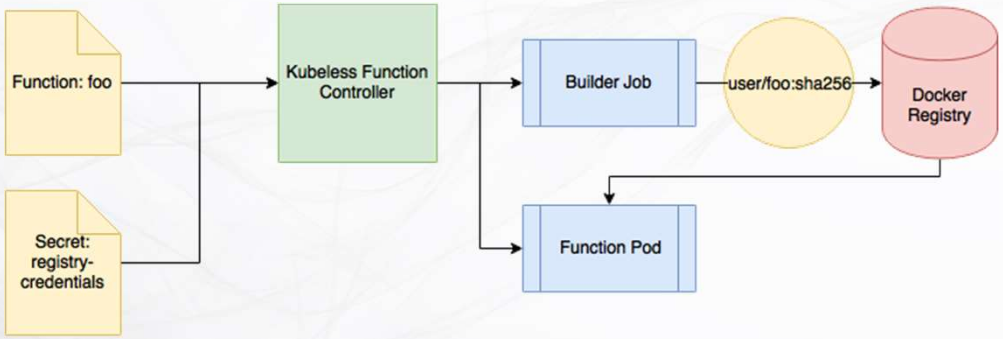

<Apache OpenWhisk의 과금 구조>


Copyright © 2018 Korea Electronics Technology Institute

# 5. Kubeless




<Kubeless의 아키텍처>




<Kubeless의 함수 빌딩 프로세스>

## Kubeless

- Kubernetes-native의 오픈 소스 서버리스 프레임워크
- Core Concepts: Functions, Triggers, Runtime
- 사용자 함수 생성: CRD Custom Resource Definitions
- Python(v2.7.x), NodeJS(v6.x 및 v8.x) 및 Ruby(v2.4.x) 지원
- 이벤트 기반 호출: Kafka, Zookeeper
- 모니터링: Prometheus
- 라이선스: Apache 2.0




<Kubeless의 UI>

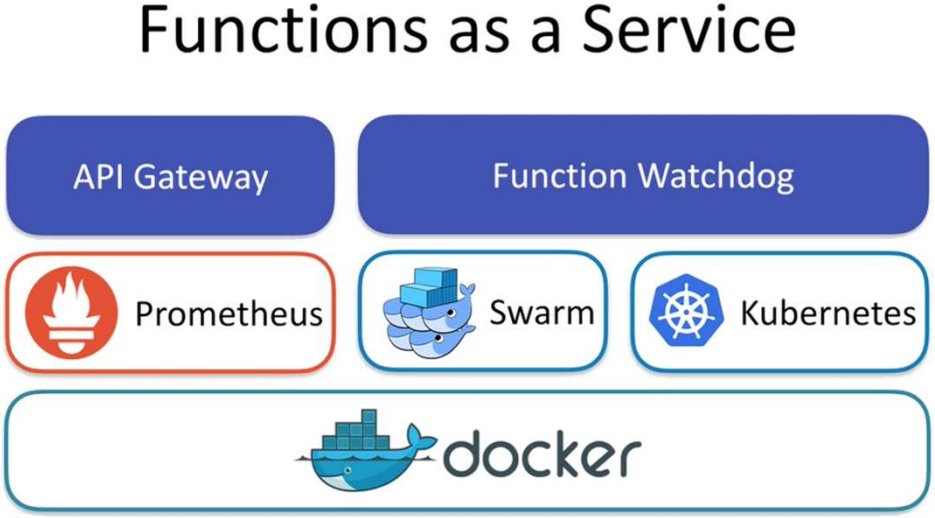
*source: https://blog.cloudboost.io/kubeless-is-more-9f20fb443b5a


Copyright © 2018 Korea Electronics Technology Institute

# 6. OpenFaaS

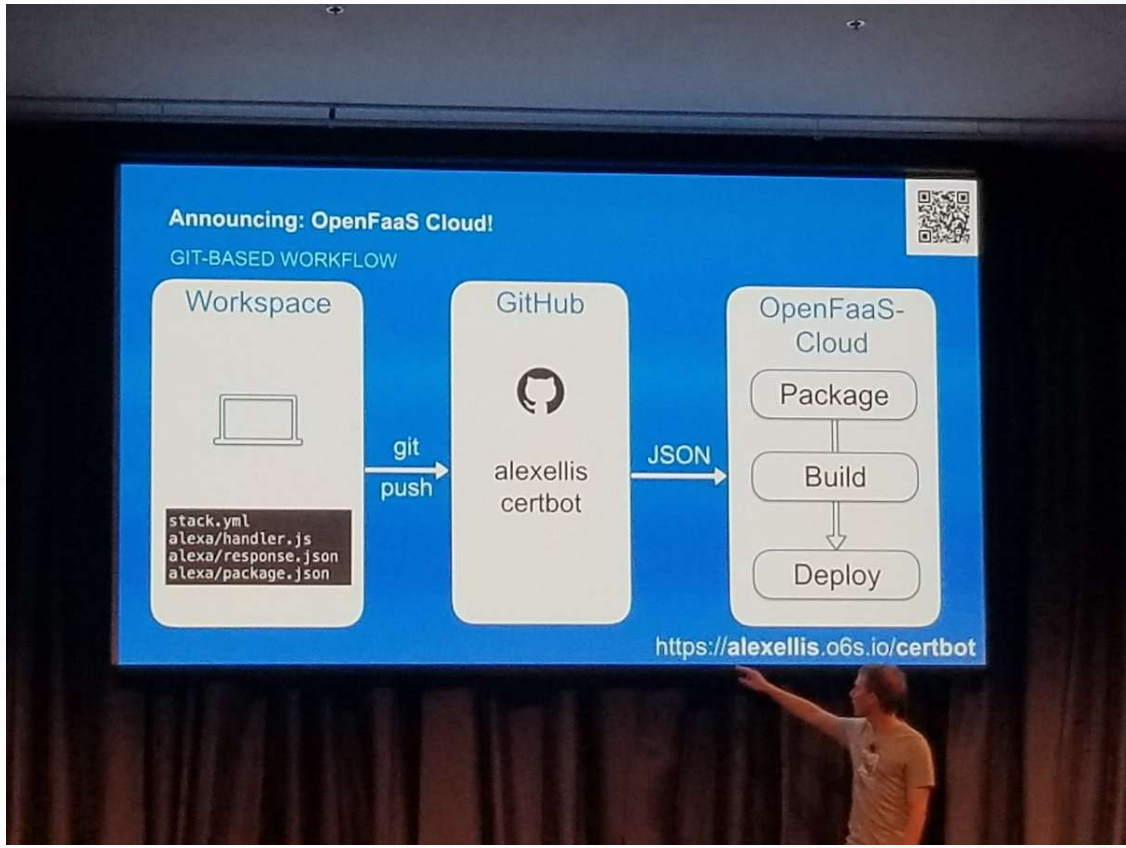
## OPENFAAS

- Docker-Swarm or Kubernetes 기반
- 다른 오케스트레이터를 지원을 위한 Plug-In 기능
- 최근 VMware의 지원
- 현재 Kafka와의 통합을 위한 개발 진행
- 가장 활발한 Community 보유
- 라이선스: MIT




<OpenFaaS 시스템의 아키텍처>

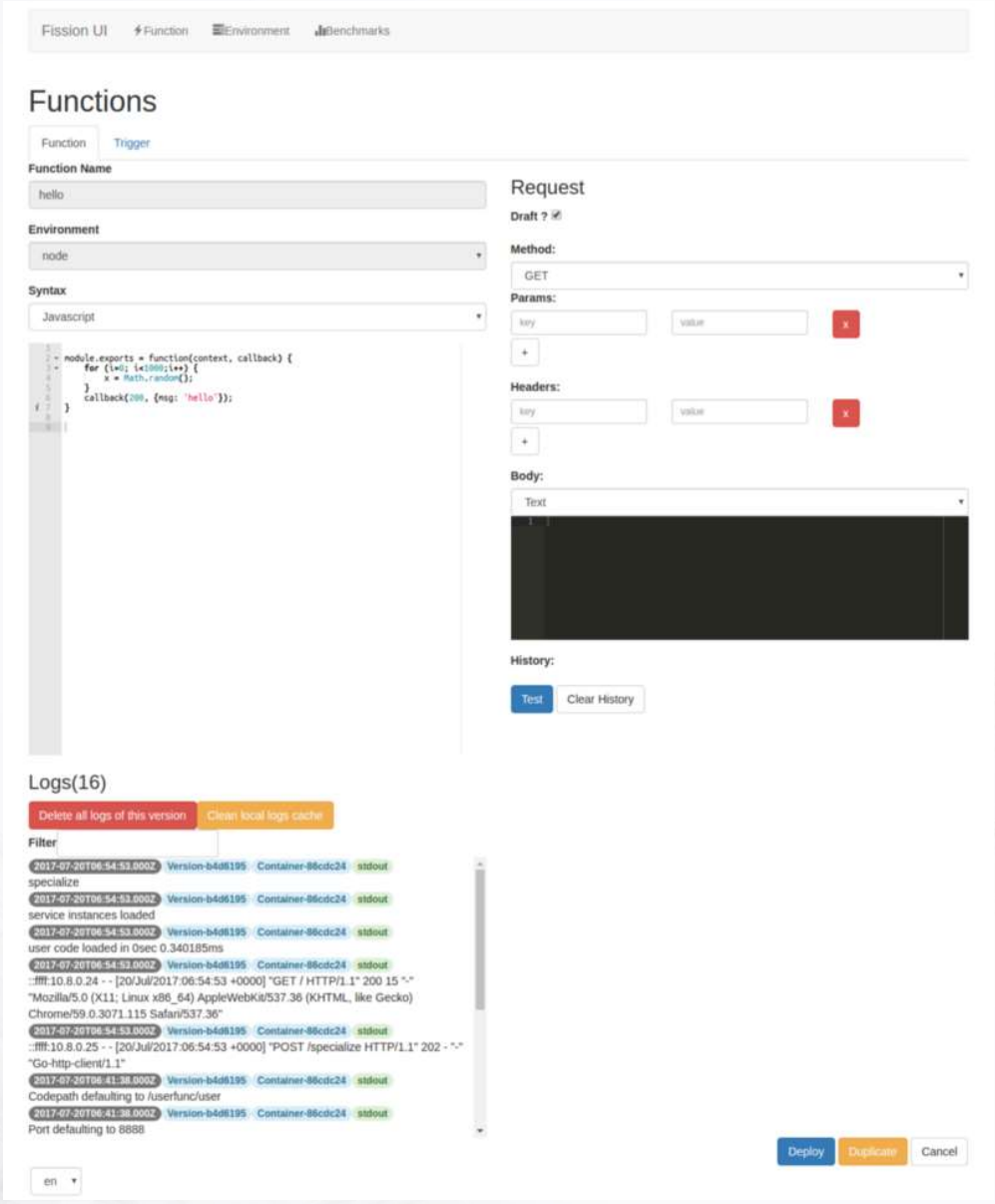
OpenFaaS cloud : 네이티브 GitHub 통합 기능을 위한 GitOps





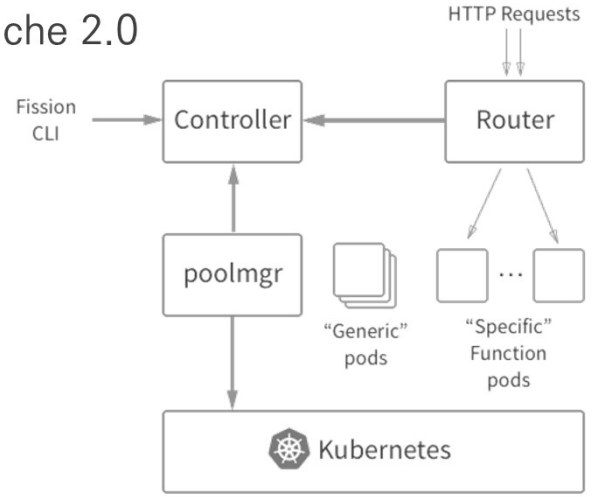
Copyright © 2018 Korea Electronics Technology Institute

# 7. Fission




fission

- "Fast Serverless Functions for Kubernetes"
- 구글의 GoLang으로 개발
- Core Concepts: *Controller, Environment, Executer, Router, Kubewatcher, Logger*
- 매우 빠른 Cold-Start (~100ms)
- 지원언어: Go, NodeJS, PHP7, Python3, .Net
- Fission Workflows
- 라이선스: Apache 2.0




<Fission의 빠른 응답성을 위한 아키텍처>

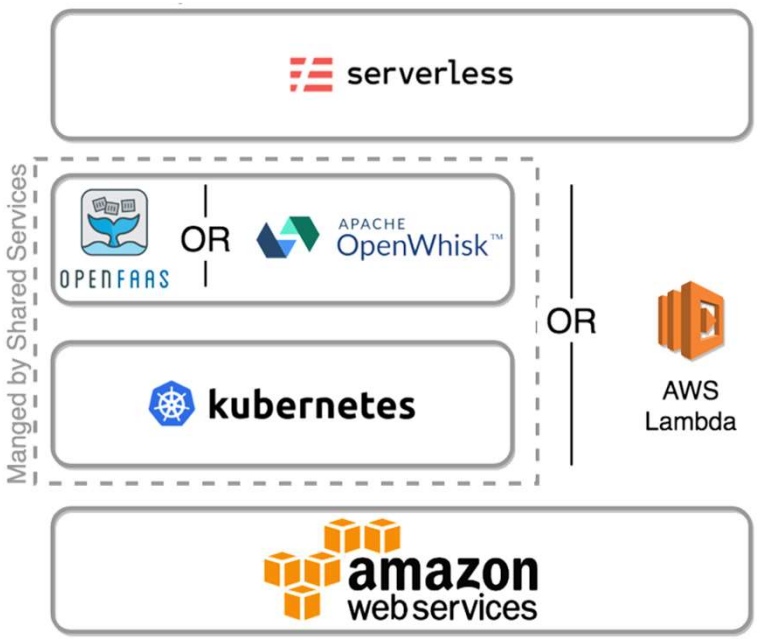
*source: https://kubernetes.io/blog/2017/01/fission-serverless-functions-as-service-for-kubernetes


Copyright © 2018 Korea Electronics Technology Institute

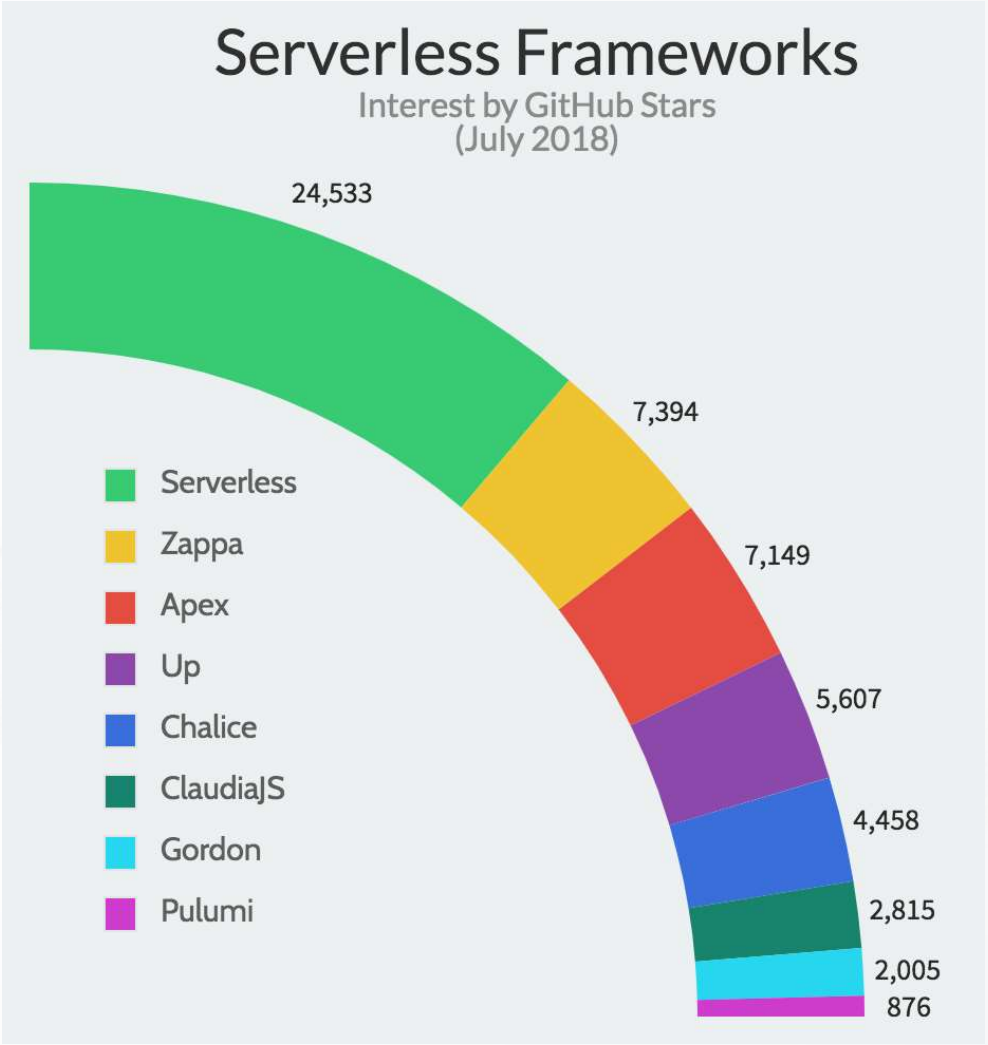
# 8. Serverless Framework

SERVERLESS

- 다양한 클라우드 제공자에 배포하기 위한 도구
- 지원 클라우드:
  - AWS Lambda, Azure Functions, Google Cloud Functions
  - OpenWhisk, Kubeless, SpotInst…




*source: https://winderresearch.com/a-comparison-of-serverless-frameworks-for-kubernetes-openfaas-openwhisk-fission-kubeless-and-more/




*source: https://serifandsemaphore.io/the-serverless-framework-landscape-ef58365b3f17


Copyright © 2018 Korea Electronics Technology Institute

# 9. Kubernetes Landscape



| Project | Open Source | Community | Backed by | Started |
|---|---|---|---|---|
| Apache OpenWhisk | Yes | Large | ASF, IBM, Adobe, Red Hat | 2015 |
| Azure Functions Runtime | Yes | Tiny | Microsoft | Late 2017 |
| Fission | Yes | Small | Platform9 | 2016 |
| Fn | Yes | Small | Oracle | Late 2017 |
| *Funktion* | *Yes* | *Tiny* | *Red Hat/Fabric8* | *2017* |
| Kubeless | Yes | Small | Bitnami | 2017 |
| Nuclio | Yes | Tiny | iguazio | Mid 2017 |
| OpenFaaS | Yes | Large | VMware | Late 2016 |
| Project Riff | Yes | Tiny | Pivotal | Mid 2017 |

*source: https://blog.openshift.com/wp-content/uploads/2018-04-12-Function-as-a-Service-on-Kubernetes-OCB.pdf


Copyright © 2018 Korea Electronics Technology Institute

CHAPTER

# 3

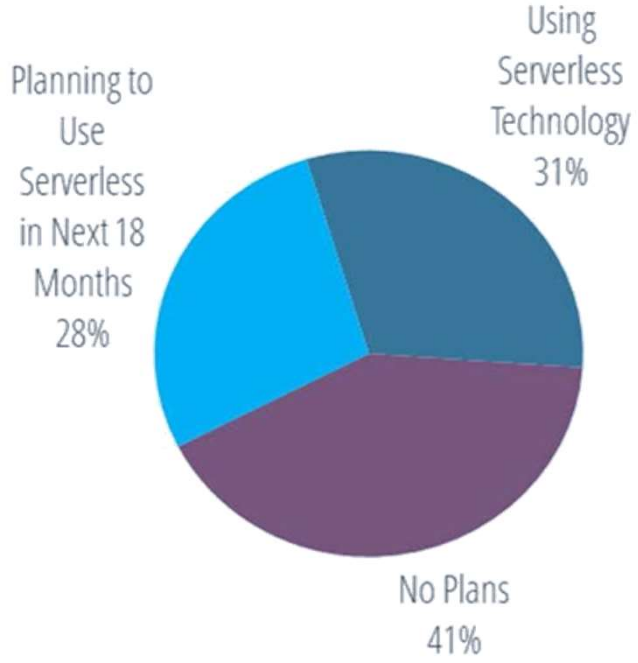
# 서버리스 컴퓨팅의 향후 전망

# 1. 서버리스 컴퓨팅 시장동향

**시사점** 향후 18개월 이내에 약 59%가 서버리스 기술을 도입, 도입예정 → 밝은 시장전망

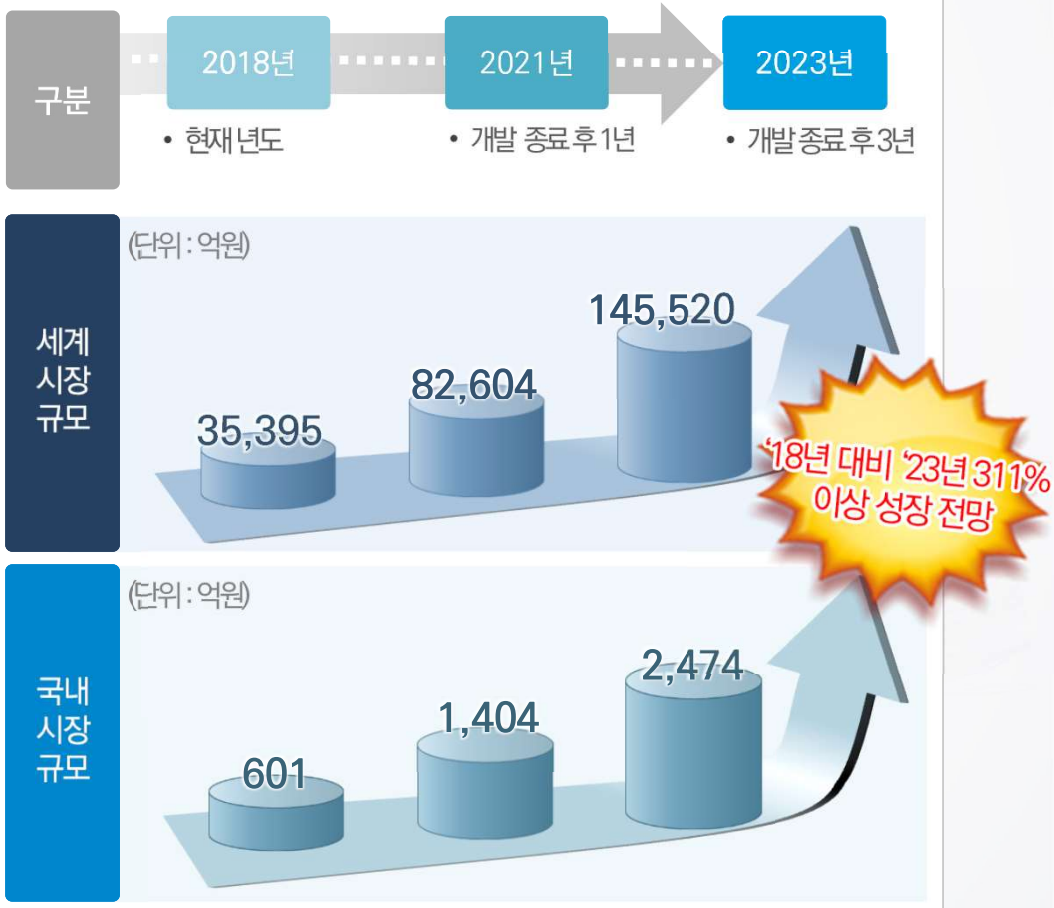
## Serverless Status

- 서버리스 클라우드 서비스 도입이 급속도로 진행될 것으로 예상되며, 빅데이터/인공지능을 포함한 다양한 응용 분야에서 수요가 예상




*source: The New Stack Analysis of Cloud Native Computing Foundation survey conducted in Fall 2017

## 서버리스 기술의 수요 확산 가속화



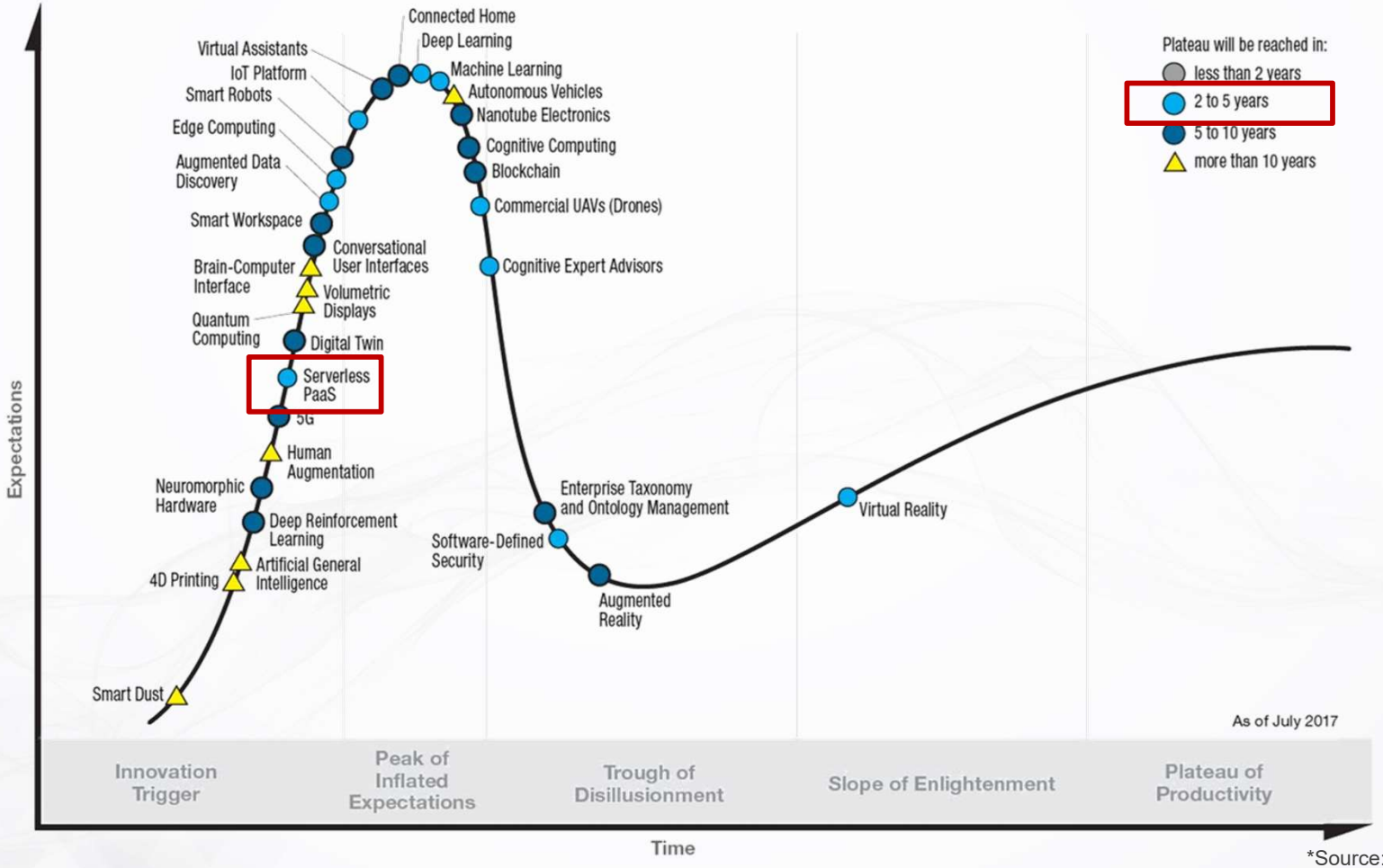

*source: Research and Markets, Global Forecast 2021


Copyright © 2018 Korea Electronics Technology Institute

# 2. 서버리스 컴퓨팅 기술전망

**시사점** 향후 2~5년 이내에 클라우드 분야에 Key Technology로 부상

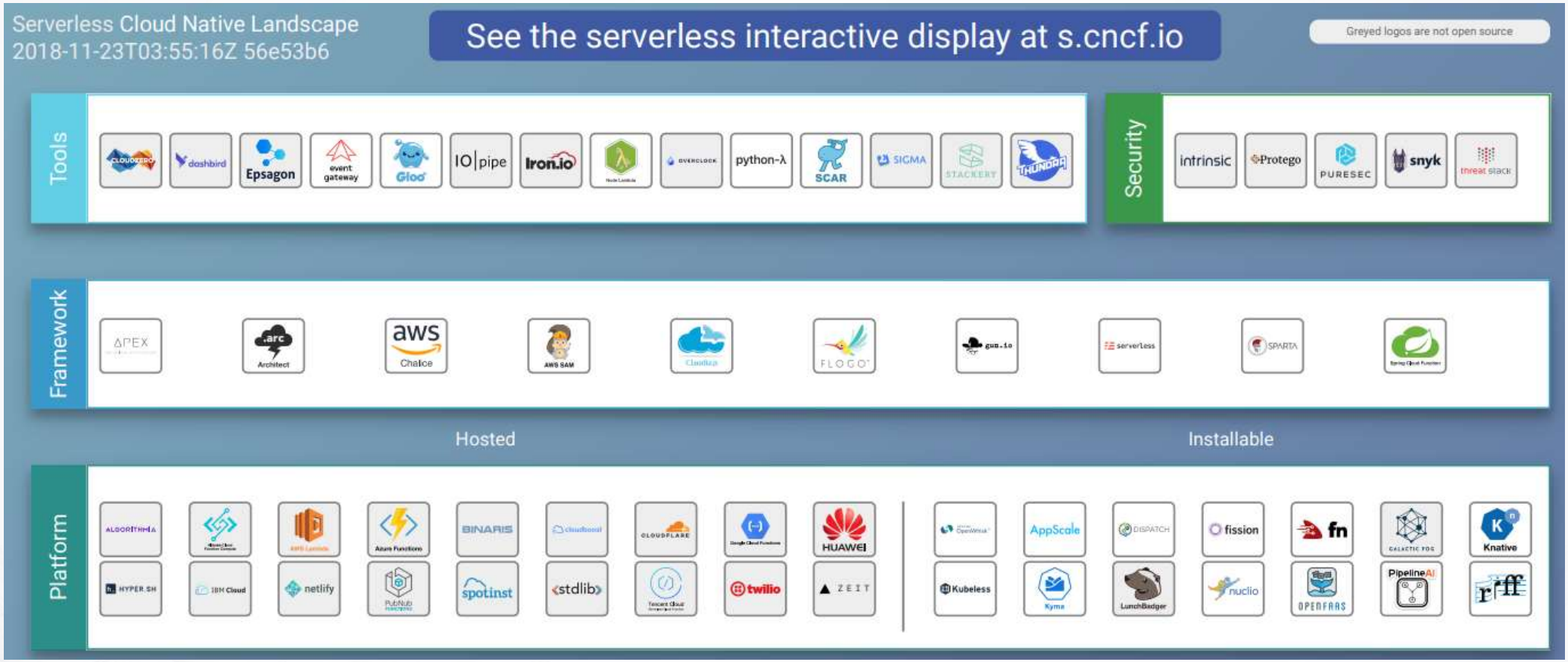
## Gartner Hype Cycle for Emerging Technologies, 2017




*Source: Gartner (2017. 7)


Copyright © 2018 Korea Electronics Technology Institute

# 3. Landscape




*Source: https://landscape.cncf.io/format=landscape&grouping=landscape&landscape=serverless


Copyright © 2018 Korea Electronics Technology Institute