

0. OpenFaaS 소개

FaaS란?

- Function As A Service

고객들이 앱 개발 및 런칭에 관련한 하부 구조의 복잡한 빌드, 유지보수 없이 애플리케이션 기능을 개발, 실행, 관리할 수 있도록 플랫폼을 제공하는 클라우드 컴퓨팅 서비스의 한 분류이다. 이 모델을 따라 애플리케이션을 빌드하는 일은 서버리스 구조를 수행하는 한 방법이며 마이크로서비스 애플리케이션을 빌드할 때 일반적으로 사용된다. (위키백과)

- FaaS의 장점
 - 개발자 생산성 향상 및 개발 시간 단축
 - 서버 관리의 부담이 없음
 - 손쉬운 확장 및 플랫폼에서 관리하는 수평적 스케일링
 - 필요한 경우에만 리소스를 사용하거나 지불
 - 거의 모든 프로그래밍 언어로 기능 작성 가능

▶ 프로비저닝

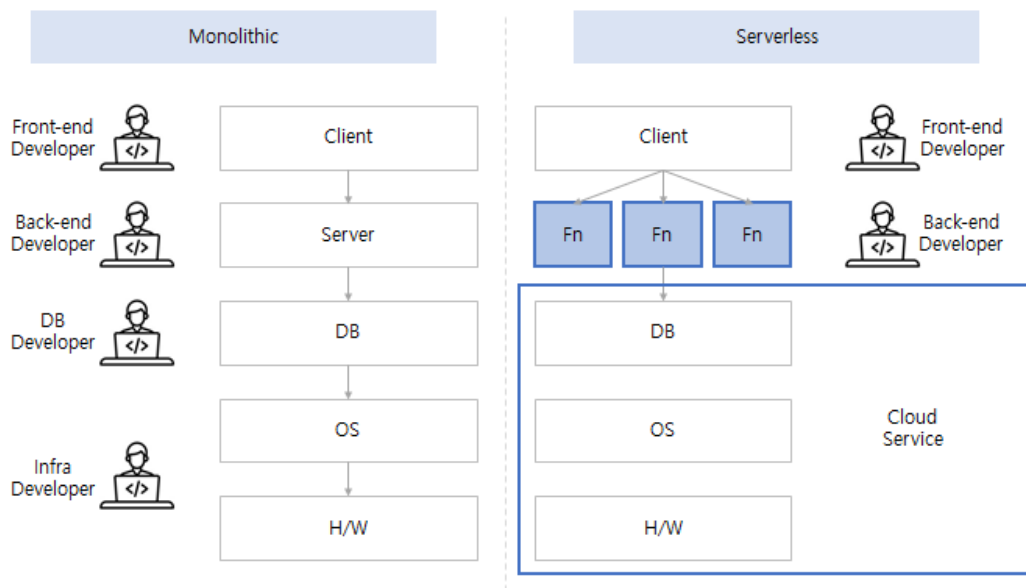
사용자 요구에 맞게 시스템 자원을 할당, 배치, 배포해 두었다가 필요 시 시스템을 즉시 사용할 수 있는 상태로 미리 준비해 두는 것 (위키백과)

- FaaS는 사용자가 컴퓨팅 리소스를 **프로비저닝** 하거나 **관리하지 않고**도 애플리케이션의 기능을 개발, 실행 및 관리할 수 있도록 함

▶ 서버리스(serverless) 컴퓨팅

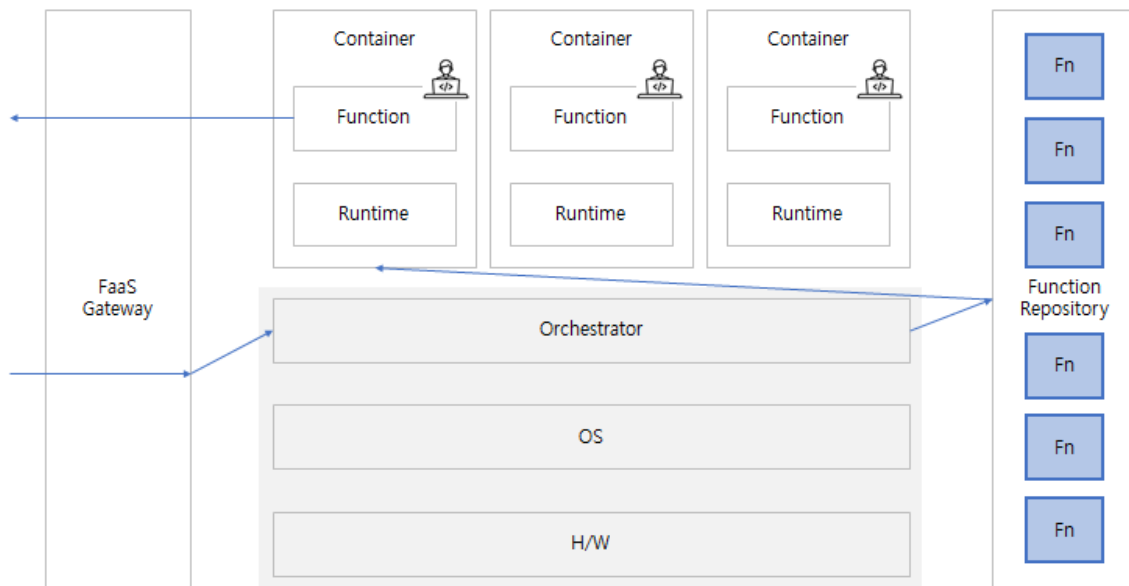
클라우드 컴퓨팅 실행 모델의 하나로, 클라우드 제공자는 동적으로 머신 자원의 할당을 관리한다. 가격은 미리 구한 용적 단위가 아닌 애플리케이션이 소비한 자원의 실제 양에 기반을 둔다. (위키백과)

- 실제 서버는 존재하나, 개발자나 운영자가 서버를 관리할 필요가 없으므로 서버리스라고 함



▶ FaaS의 동작 원리

- 개발자가 함수를 개발해 배포하면 컨테이너 단위로 필요할 때 필요한 만큼 실행



- 각 함수는 컨테이너 형태로 저장되고, 호출이 있을 때 실행되며, 일정 시간이 지난 후 소멸

FaaS 프레임워크

▶ 클라우드 서비스

- AWS Lambda
- 네이버 클라우드 플랫폼 Cloud Functions
- 구글 클라우드 펄션
- 마이크로소프트 애저 펄션
- IBM/아파치 오픈휘스크
- 오라클 클라우드 Fn

▶ 오픈소스

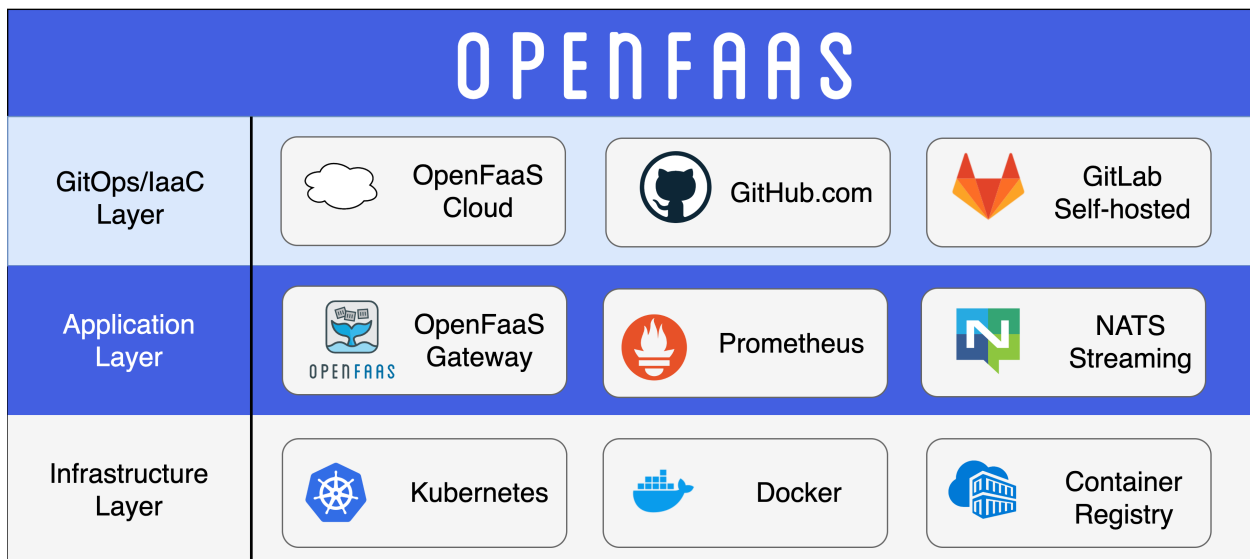
- OpenFaas
 - 다양한 언어를 지원

- 가장 사용성이 좋고, 바로 사용해볼 수 있어 가장 인기가 많음
- Kubeless
 - Kubernetes core 기능들을 이용해 개발되어, Kubernetes kernel에 이미 존재하는 코드를 로직에 포함할 필요가 없어 코드를 경량화할 수 있음
- Fission
 - autoscaling으로 많은 부분에서 리소스 관리를 신경쓰지 않을 수 있음

OpenFaas

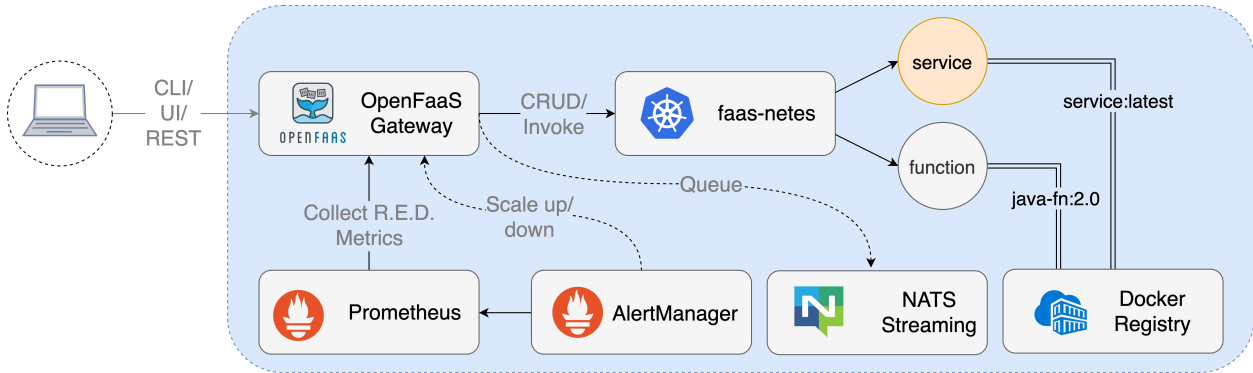
| <https://docs.openfaas.com/>

▶ Stack



- OpenFaas를 배포하는 플랫폼으로 Kubernetes를 권장
- NATS : 비동기 실행과 queue 제공
- Prometheus : 메트릭을 제공하고 AlertManager를 통해 auto-scaling이 가능하도록 함
- Container Registry : API를 통해 OpenFaas에 배포할 수 있도록 각 artifact를 보유

▶ Conceptual Workflow



- REST API, CLI, UI를 통해 지정한 endpoint 또는 기본으로 노출된 루트로 Gateway에 접속
- Prometheus가 auto-scaling에 사용되는 메트릭 수집
- /function/Name에서 /async-function/NAME으로 변경해 NATS Streaming 대기열에서 호출
- faas-netes를 통해 docker registry의 service 또는 function을 호출
 - ※ faas-provider를 통해 CRUD, invoke, scaling을 수행
 - faas-netes는 Kubernetes용 공식 openfaas provider임
 - 이외에 faasd, Docker Swarn, faas-memory 등 다른 provider도 있음