

# Microservice with Kubernetes

## 개요

Microservice Architecture(MSA)는 작은 서비스들의(Microservices) 조합을 통해 비즈니스 애플리케이션을 개발하는 접근방법을 말합니다. 이를 통해, 비즈니스 요구사항을 빠르게 개발 가능(Time to Market)하고, 변경 사항을 비교적 자유롭게 운영 환경에 즉시 배포 가능하고, 유지보수가 용이하며, 부분 장애가 전체로 미치지 않아 장애 영향 최소화 할 수 있는 장점이 있습니다.

SDS Cloud에서는 최근의 급격한 비즈니스 및 기술 변화에 대응할 수 있도록 MSA 기반 클라우드 네이티브 어플리케이션 개발을 위한 **Kubernetes Engine**, **API Gateway**, **Message Queue(Kafka, RabbitMQ)**, **Service Mesh**등의 클라우드 서비스를 제공합니다.

## 아키텍처 다이어그램

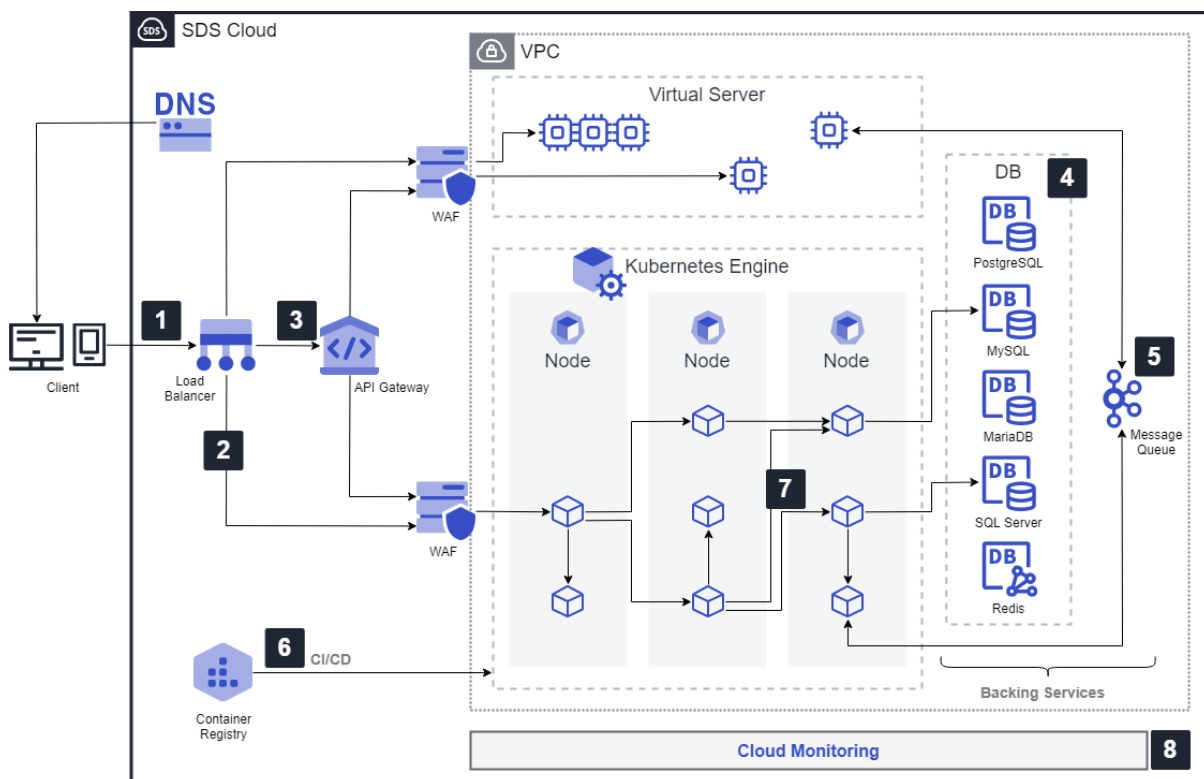


Figure 1. Kubernetes 기반 Microservice architecture

1. 클라이언트의 요청은 **Load Balancer**를 통해서 VM/Container Workload 및 **API Gateway** 서비스에 전달된다.
2. API 요청을 제외한 클라이언트의 요청은 **Kubernetes engine**에서 제공되는 ingress에 전달된다. **Kubernetes engine**은 ingress를 통해 들어온 요청을 매핑되어 있는 서비스 Pod로 라우팅 처리를 수행한다.
3. 클라이언트의 API 요청 처리를 위해 **API Gateway**는 최전방에 위치하여 외부로부터 들어오는 모든 API 호출에 대해 인증 및 서비스 라우팅 처리를 수행한다.
4. Microservice Architecture는 구조화된 서비스 단위로 분리된 DB를 구성한다.
5. Backing Service는 Microservice들이 네트워크를 통해서 사용할 수 있는 Back-end 서비스를 말하며, PostgreSQL과 같은 DBMS 서비스, Redis와 같은 in-memory cache 서비스, **Kafka**와 같은 Microservice 메시지의 비동기 처리를 위한 Message Queue 서비스 등을 제공한다.
6. DevOps 기반 CI/CD 서비스를 통해서 Microservice 어플리케이션의 개발 및 운영을 Seamless 하게 지원한다.
7. Microservice 구성 요소간의 네트워크를 제어하기 위해 **Service Mesh** 기술을 사용할 수 있다. **Service Mesh** 기술을 통해 분할된 어플리케이션간의 다양한 네트워크 구성과 데이터 흐름에 대한 관리 및 추적이 가능하다.
8. MSA에서는 다수의 Microservice가 분산환경에서 운영되기 때문에 서비스 상태에 대한 모니터링이 필요하다. Monitoring, Logging, Tracing 과 같은 Telemetry서비스는 서비스들을 모니터링하고, 서비스 별로 발생하는 이슈들에 대응할 수 있도록 환경을 구성하는 역할을 한다.

## 사용 사례

### A. 물류 시스템 Microservice 적용

물류 시스템의 Application 경량화 및 가용성 확보 및 비즈니스 확대 시 유연한 대응이 가능한 아키텍처를 위해 신규 Application 개발 시 컨테이너 기반의 MSA를 적용하여 구축을 수행 했습니다.

물류 시스템을 구성하는 창고관리 시스템, 수배송시스템, 복합운송시스템 등 세부 시스템은 컨테이너로 구성되고 SDS Cloud는 표준적인 Kubernetes Engine 서비스를 이용하여 컨테이너 기반의 환경에서 Microservice의 손쉽고 빠른 배포, 독립적인 실행 환경, 확장성 및 복원성을 제공합니다.

## B. 글로벌 제조사의 개발 플랫폼 사례

글로벌 제조사의 Application 개발을 지원하는 플랫폼은 Microservice 기반의 Application 개발을 위한 API Gateway, DevOps, Kubernetes 기반 컨테이너 서비스를 제공합니다.

이를 통해서 개별 구축 서비스에 대한 DevOps 기반 Canary 배포를 지원하고 API Gateway를 통해서 개별 서비스의 API를 카탈로그화 하여 서비스의 재사용성을 높이고 있습니다.

향후 Service mesh 기능을 통해 Microservice에 대한 흐름 제어 및 가시성을 확보할 예정입니다.

## 선결 사항

없음

## 제약 사항

Service Mesh, Message Queue등의 일부 기능은 추후 제공 예정입니다.

## 고려 사항

없음

## 관련 상품

- Virtual Server
- Kubernetes Engine
- API Gateway
- DB Service
- Load Balancer
- DNS

- Cloud Monitoring
- Kafka
- Container Registry ('21년 출시 예정)
- DevOps Tools ('21년 출시 예정)
- RabbitMQ (추후 제공 예정)
- Service Mesh (추후 제공 예정)