

Microservice with Kubernetes

Overview

The Microservice Architecture (MSA) refers to an approach of developing business applications with a combination of small services. This method encourages fast development of business requirements (Time to Market), immediate and relatively free distribution of changes to the operating environment, convenient maintenance, and minimized failure impact by preventing partial failure affecting the entire system.

SDS Cloud provides services such as **Kubernetes Engine**, **API Gateway**, Messaging (**Kafka** and **RabbitMQ**), and **Service Mesh** for MSA-based cloud-native application development to respond to the recent rapid business and technology changes.

Architecture Diagram

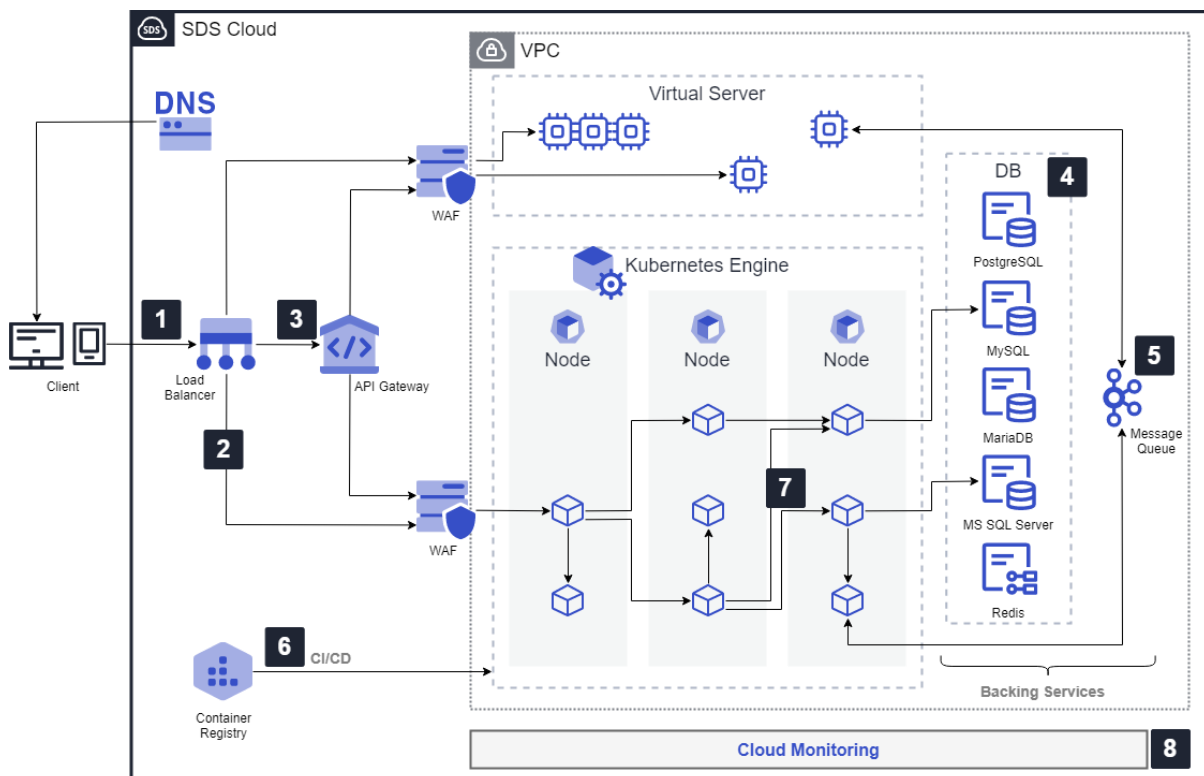


Figure 1. Microservice with Kubernetes

1. A client's request is delivered to VM/container workload and **API Gateway** service

through **Load Balancer**.

2. Requests from clients that are not API requests are delivered to ingress provided by **Kubernetes Engine**. The **Kubernetes Engine** performs route processing of the requests received through ingress to the mapped service pod.
3. For a client's API request, **API Gateway** is located at the forefront and performs authentication and service route processing for all API calls from outside.
4. Microservice Architecture includes a separate DB in a structured service unit.
5. Backing Service is a back-end service that Microservices uses through the network. DBMS service such as PostgreSQL, in-memory cache service such as Redis, and **Message Queue** service for asynchronous processing of Microservice messages are also available.
6. DevOps-based CI/CD services ensure seamless support for the development and operation of Microservices applications.
7. **Service Mesh** controls the network between microservice components. It also manages and tracks various network configuration and data flows among various applications.
8. In MSA, many microservices are operated in a distributed environment, requiring monitoring on the service status. Telemetry services such as monitoring, logging, and tracing take responsibility of configuring the environment for monitoring services and responding to issues from each service.

Use Cases

A. Microservices in logistics system.

We implemented container-based MSA when developing new applications for more flexible response architecture, lightweight applications, and availability.

The logistics system is composed of container-based sub-systems including a warehouse management system, transportation management system, and multimodal transportation system. SDS Cloud provides easy and fast deployment, an independent execution environment, scalability and resilience of Microservices in a container-based environment using **Kubernetes Engine** services.

B. Global manufacturer's development platform

The platform that supports application development of global manufacturers provides API Gateway, DevOps, and Kubernetes-based container services for Microservice-

based application development.

Through this, we support DevOps-based canary deployment for individual services and increase service reusability by cataloging individual service APIs through API Gateway.

In the future, we plan to gain flow control and visibility for Microservices through the Service mesh function.

Pre-requisites

None.

Limitations

Some functions, such as **Service Mesh** and Message Queue, are scheduled to be provided later.

Considerations

None.

Related Products

- Virtual Server
- Kubernetes Engine
- API Gateway
- DB Service
- Load Balancer
- DNS
- Cloud Monitoring
- Kafka
- Container Registry (To be available in 2022)
- DevOps (To be available in 2022)

- RabbitMQ (To be launched)
- Service Mesh (To be launched)