Container-based CI/CD Pipeline

Overview

Container-based CI/CD pipeline provides an integrated development environment that supports simple and easy application development and effective collaboration. It improves development productivity by automating the build, test, and deployment process while improving the code quality through source history management and automatic verification.

In addition, continuous monitoring helps analyze performance changes before and after app deployment and respond to service failures ahead of time.

Architecture Diagram

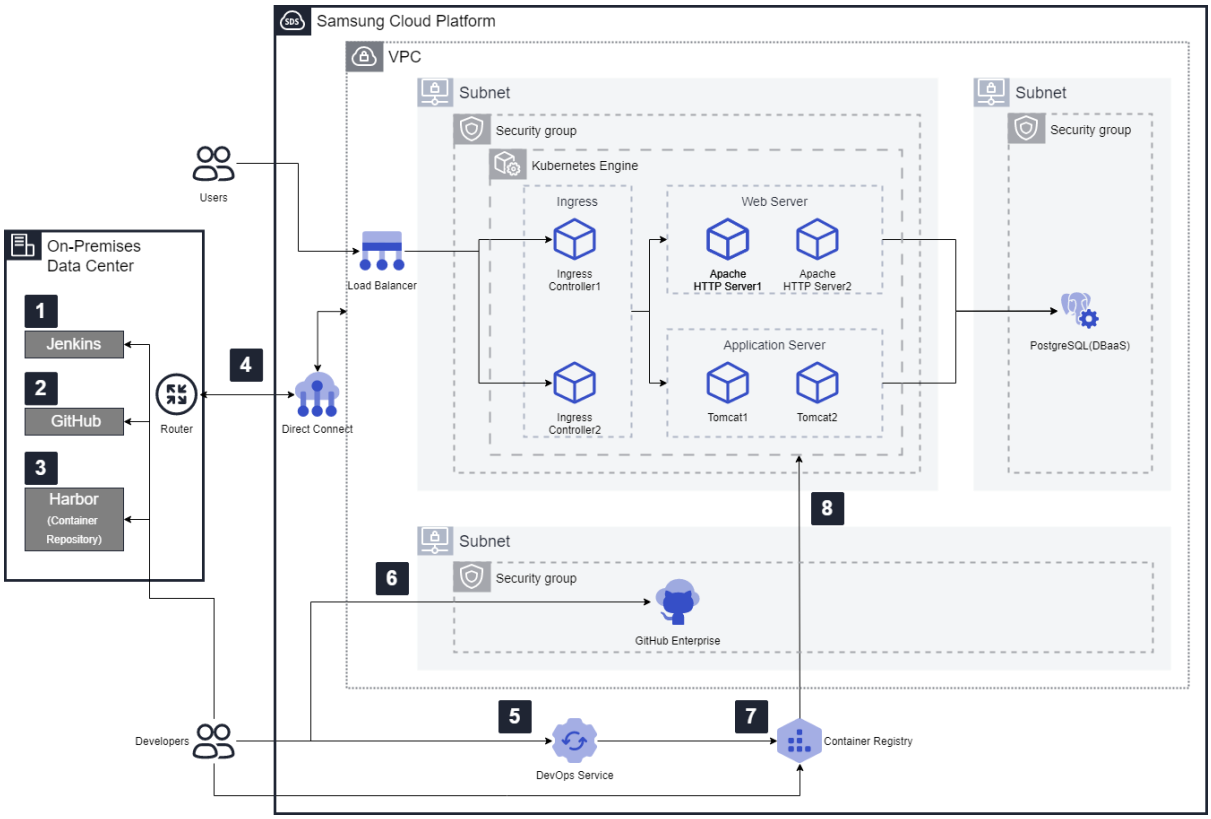

Figure 1. Container-based CI/CD pipeline

1. (When configuration within the on-premises environment is required) Establish an

environment for continuous integration (CI) and continuous deployment (CD).

2. (When configuration within the on-premises environment is required) Establish source configuration management.
3. (When configuration within the on-premises environment is required) Build a **Container Registry** that allows applications developed to be stored/managed as containers.
4. You can connect the on-premises environment with the **Kubernetes Engine** of the Samsung Cloud Platform(hereafter SCP) by applying for a **Direct Connect** service, which allows you to deploy your application in the **Kubernetes Engine**.
5. When it becomes difficult to build DevOps in a customer on-premises environment, you can apply for **DevOps** provided by Samsung Cloud Platform to configure an automated CI/CD pipeline.
6. (If configuration within the SCP is required) You can connect from DevOps to source configuration management by applying for the **GitHub Enterprise**.
7. (If configuration within the SCP is required) By applying for **Container Registry**, you can turn the built-in application on DevOps into container images and store them in the registry.
8. By continuously deploying saved container applications as **Kubernetes Engine**, you can deploy the fully developed application.

Use Cases

A. Development Environment for Defect Analysis System

By turning an automated defect identification/analysis system into a container-based Kubernetes Engine environment and utilizing DevOps's automated CI/CD pipeline, the time needed for establishing development environment can go down from two days to five minutes.

B. Development Environment for Converting Management Information System to Containers

Configuring a development environment for a management information system usually takes at least 1-2 weeks but with DevOps service, immediate configuration is possible. The service significantly improves development productivity by automating the process of image containerization and deployment of developed applications, environment settings, and execution environments.

Prerequisites

None.

Limitations

None.

Considerations

In the case of on-premises configuration, CI/CD S/W and Pipeline must be directly configured on the customer's H/W.

In order to connect an on-premises CI/CD pipeline with a **Kubernetes Engine**, you must apply for a **Direct Connect** service. The IP addresses must be specified to avoid any conflicts between the subnet IP addresses of the **VPC** and the customer's Network IP addresses. In addition, **VPC** Firewall and **Security Group** need to be registered so that the node pool of the **Kubernetes Engine** can be connected to the customer's on-premises **Container Registry**.

When using products within SCP, you can configure Pipeline by applying for **DevOps service** and **GitHub Enterprise**.

Related Products

- Kubernetes Engine
- Kubernetes Apps
- VPC
- Virtual Server
- Load Balancer
- Direct Connect
- File Storage
- DevOps Service
- Container Registry

- GitHub Enterprise

Related Documents

- Container-based web application