Container-based web application

Overview

Containers can offer a virtualization of application runtime environments to allow developers to deploy applications easily and continuously in any environment. Virtual machines virtualize hardware stacks, whereas containers virtualize at the operating system level. By sharing OS kernel, containers run faster and use less memory.

Kubernetes is a platform that provides resilient execution and management of distributed system environments based on automation and management of numerous container applications. Kubernetes handles application extensions and failures, provides easy and rapid deployment, and supports a successful building and operation of web application systems.

Architecture Diagram

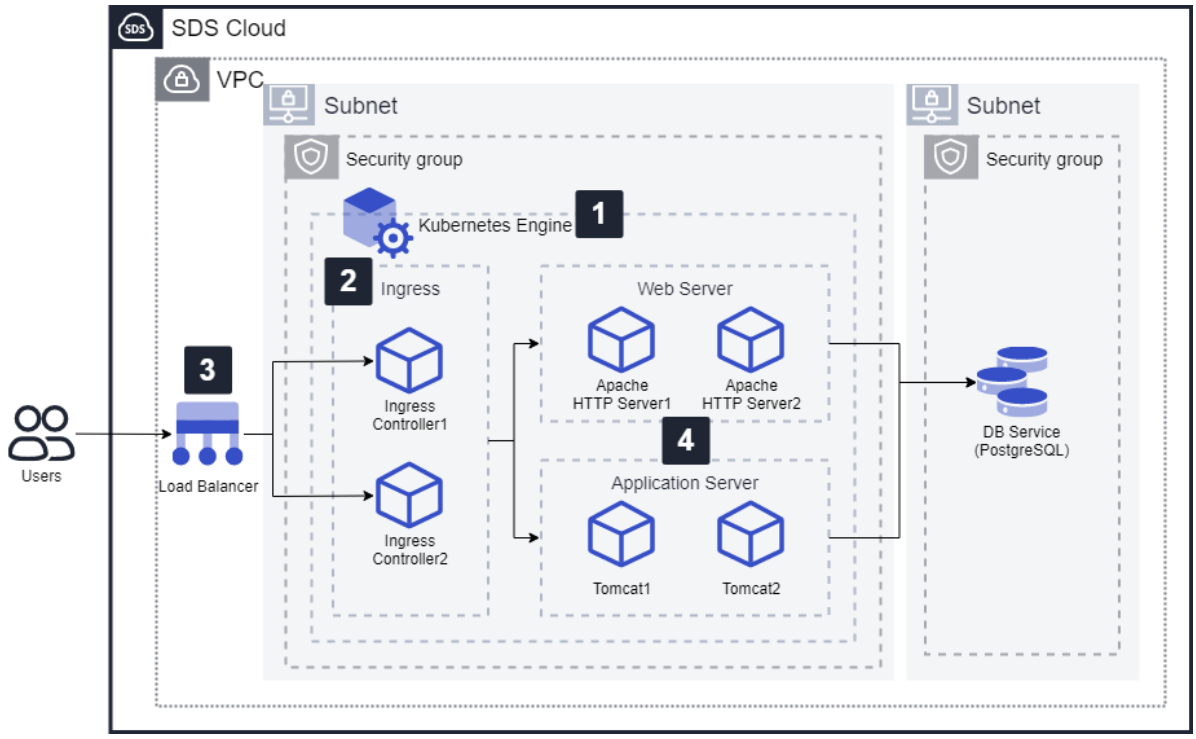

Figure 1. Container-based web application

1. Provide **Kubernetes Engine** for container runtime management and orchestration.

2. Provide Ingress Controller from the **Kubernetes Apps** to expose HTTP and HTTPS paths of the service inside the cluster.
3. Create **Load Balancer** and register worker node IP to connect to the Ingress Controller for availability.
4. Provide Apache HTTP Server, Tomcat, etc. from the **Kubernetes Apps** to configure WEB/WAS applications.

Use Cases

A. Containerization of Management Information System.

Applying container technology and standard open source software generally reduces hardware resource usage by 20% and the licensing cost by 80%.

Containerization minimizes the overlapping development and investment costs for the same function caused by different technology, software and versions for each system, thereby reducing the transition lead time.

B. Containerization of a system that needs to scale out resources due to a user spike

Scale out WEB/WAS container resources prior to the expected time of a user congestion.

In case of an unpredictable surge of users, the auto-scaling function of Kubernetes can be applied to automatically increase container resources when used above the resource threshold of CPU/memory. If it falls below the threshold, the system automatically retrieves container resources for efficient operation.

Pre-requisites

None.

Limitations

The maximum number of worker node creation in the **Kubernetes Engine** is limited to the IP range available in the subnet and the ceiling is 50.

Considerations

Container-based WEB/WAS should be optimized according to specifications of an application. Since the container image provided by **Kubernetes Apps** only has basic configuration for **Kubernetes Engine**, it is recommended to refer to the images from the community for optimization and WEB/WAS settings.

For Ingress Controller, it is recommended to use the service through **Load Balancer** to ensure load balancing and availability. In addition, domain and DNS IP registration should precede the service, as the Ingress Controller manages connections of application containers based on the domain.

WEB/WAS containers are connected to the DB system with worker node IP. If you use Security Group or DB access control, you may be able to connect to the DB system only when the worker node is registered.

Related Products

- Kubernetes Engine
- Kubernetes Apps
- VPC
- Virtual Server
- Load Balancer
- File Storage

Related Documents

- Container-based CI/CD pipeline