

Container Service

Product Introduction

Product Introduction

Container Service is a high-performance and scalable container application management service, which enables you to manage application lifecycles by using Docker containers.

Container Service provides multiple application release methods and the continuous delivery ability, and supports microservice architecture. By simplifying the setup of container management cluster and integrating with the Alibaba Cloud abilities of virtualization, storage, network, and security, Container Service makes an ideal running cloud environment for Docker.

Features

Cluster management and flexible choices

Container Service, as a cluster management service, allows you to:

- Select regions to create and delete clusters per your needs.
- Select environments with either classic network or Virtual Private Cloud (VPC).

Multiple server hosting modes

Container Service allows you to:

- Authorize Container Service to create Elastic Compute Service (ECS) instances and add them to a specified cluster.
- Add purchased ECS instances to a specified cluster.

One-stop container lifecycle management

As a one-stop service that manages container lifecycles, Container Service provides the following features:

- **Network:** Container Service supports intercommunication between containers across hosts by using domain names defined by container name or hostname.
- **Storage:** Container Service supports data volume management. OSSFS and Network Attached Storage (NAS) are supported.
- **Log:** Container Service supports automatic log collection and Log Service integration.
- **Monitoring:** Container Service supports monitoring containers and virtual machines (VMs).
- **Scheduling:** Container Service supports strategies such as high availability across zones and

- rescheduling of abnormal nodes.
- **Routing:** Container Service supports forwarding of Layer-4 and Layer-7 requests and their bindings to backend containers.
- **Sub-account:** Container Service supports Random Access Memory (RAM) authorization management for clusters.

Docker compatibility

Container Service is compatible with:

- Standard Docker API.
- Docker Swarm 1.2.6.
- Docker Engine CE 17.03.1.
- Docker Compose V1/V2/V3.

Unique value-added ability

Container Service has the unique value-added ability of Alibaba Cloud environment to provide you with better experience.

Container Service:

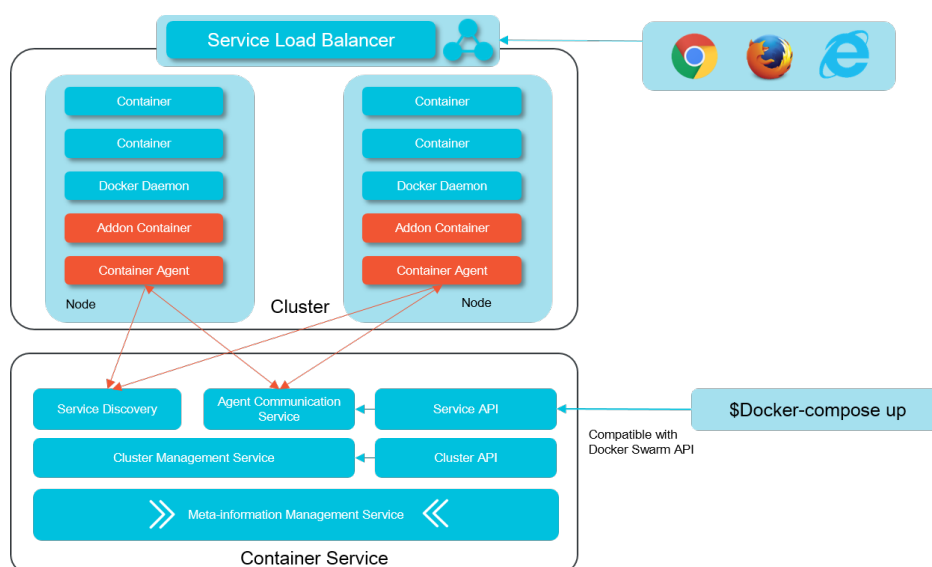
- Integrates with VPC to provide secure and high-performance deployment plans that support hybrid cloud.
- Extends Docker Compose template definition to enhance lifecycle management.
- Integrates with Server Load Balancer to provide containers with access ability.

High available scheduling strategy

Container Service can easily deal with the upstream and downstream delivery process by using the high available scheduling strategy.

Container Service supports:

- Affinity strategy and horizontal scaling of services.
- High availability across zones and disaster recovery.
- The Open APIs for cluster and application management to easily connect with the continuous integration and private deployment system.



The basic architecture of Container Service is as shown in the preceding figure, and is described as follows:

- **Cluster management service:** Docker cluster management and scheduling are supported.
- **Service discovery:** Storage of metadata (including Docker status) is supported.
- **Agent communication service:** Communication service between each host and cluster management service is supported.
- **Cluster API:** United Open APIs of Alibaba Cloud are provided.
- **Service API:** APIs that are compatible with Docker Swarm APIs are provided.

Ease to use

- Supports one-click creation of container clusters.
- Enables orchestration of applications using Docker Compose templates.
- Supports graphical user interfaces and Open APIs.

Secure and controllable

- Enables you to have your own ECS (Elastic Compute Service) instance.
- Supports customizing security groups and VPC (Virtual Private Cloud) security rules.

Protocol compatibility

- Compatible with standard Docker APIs.
- Supports migrating applications to cloud platforms seamlessly.
- Supports hybrid cloud scenarios.
- Compatible with Docker Compose template protocol.

- Interoperates with APIs for third-party scheduling delivery and system integration.

Efficient and reliable

- Supports starting massive containers in seconds.
- Supports exception recovery and automatic scaling of containers.
- Supports scheduling containers across zones.

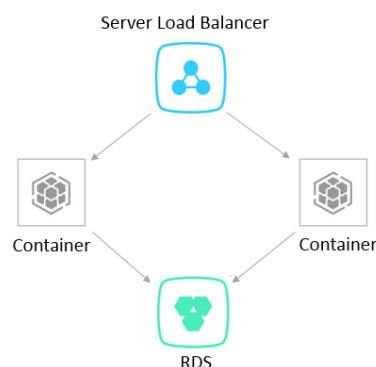
High traffic websites

For websites which have a high volume of traffic, or experience sudden spikes for a very short time, Alibaba Cloud Container Service provides automatic scaling of Docker applications. Used in integration with Server Load Balancer, the Container Service can efficiently manage traffic peaks and maintain a consistent user experience.

Solution architecture:

ECS instances + Server Load Balancer instance to the container cluster + ApsaraDB for data storage

You can use WordPress or other container images to deploy a web application in one single click.



Creation of a continuously integrated system

When a new application is launched in the market, a lot of steps are involved in deploying the application from your codebase to the production site, resulting in high overhead costs.

In such scenarios, Alibaba Cloud Container Service implements a CI/CD pipeline which reduces cost as well as time to market. This helps companies to accelerate application development, automate the deployment steps, and offer a more stable product.

Steps to implement CI/CD pipeline:

1. Create a Docker repository of the automatic build in Alibaba Cloud Container Registry service, and associate the source control management systems to GitHub or Alibaba Cloud Code.

2. Once code is committed, the Container Registry service is triggered and builds the Docker image automatically.
3. After creating an image, call back the Open API of the Container Service to update the container application.



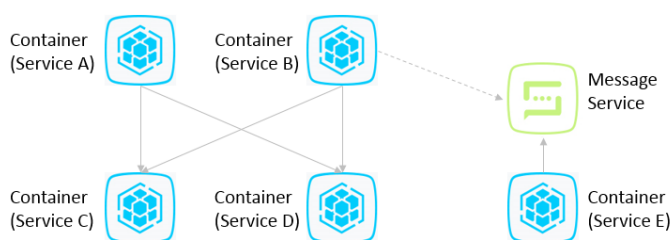
Microservice architecture

Monolithic applications are extremely complex, and can be difficult to maintain, upgrade, and update with new features. In order to update a small feature, you generally need to redeploy the entire application. Implementing microservice architecture resolves this issue by creating a single application as a suite of small, independent services that run in their own processes and are developed and deployed independently.

Alibaba Cloud Container Service packages such microservices and deploys them with Docker compose templates.

Solution architecture:

1. Split a monolithic application into several microservices, and package microservices with Docker images.
2. Use the Docker Compose template to describe the configurations and dependencies of services.
3. Deploy the application with the selected Compose template.



Cluster

A cluster is a collection of cloud resources that are required to run containers. It associates with several server nodes, Server Load Balancer instances, Virtual Private Cloud (VPC), and other cloud resources.

Node

A node is a server (either a virtual machine (VM) instance or a physical server) that is installed with a Docker Engine and is used to deploy and manage containers. The Agent program of Container Service is installed in a node and registered to a cluster. The quantity of nodes in a cluster is scalable.

Container

A container is a runtime instance created using a Docker image. A single node can run multiple containers.

Image

A Docker image is a standard packaging format of a container application. You can specify an image to deploy container applications. The image can be from the Docker Hub, Alibaba Cloud Container Hub, or your private registry. An image ID is uniquely identified by the URI of the image repository and the image tag name (the latest tag name is used by default).

Orchestration template

An orchestration template contains definitions of a group of container services and their interconnection relationships, and can be used to deploy and manage multiple container applications. Container Service supports and extends the Docker Compose template specifications.

Application

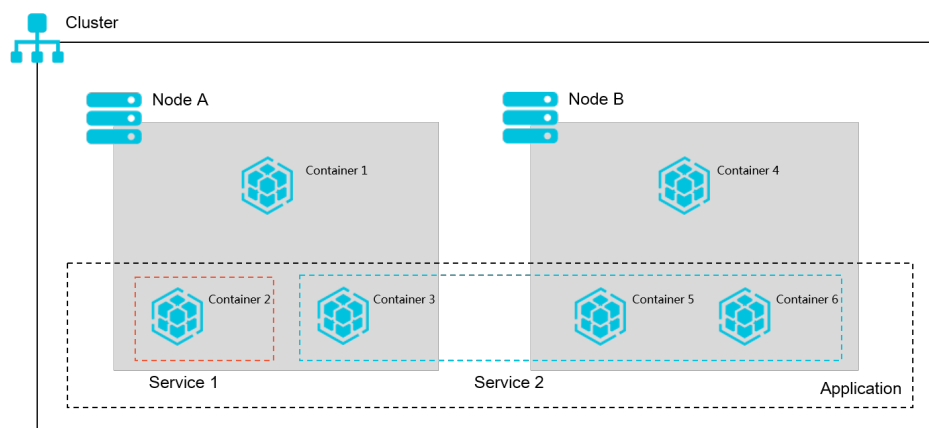
An application can be created by using an image or an orchestration template. Each application can contain one or more services.

Service

A service is a group of containers defined based on the same image and configurations. It is used as a scalable microservice.

Associations

See the following figure for the associations among cluster, node, container, application, and service.



Reference

For more container related glossaries, see [Docker glossary](#).