Alibaba Cloud Container Service

Product Introduction

Issue: 20181016

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.
5. By law, all the content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade

secrets. No part of the Alibaba Cloud website, product programs, or content shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion , or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos , marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates).

6. Please contact Alibaba Cloud directly if you discover any errors in this document.

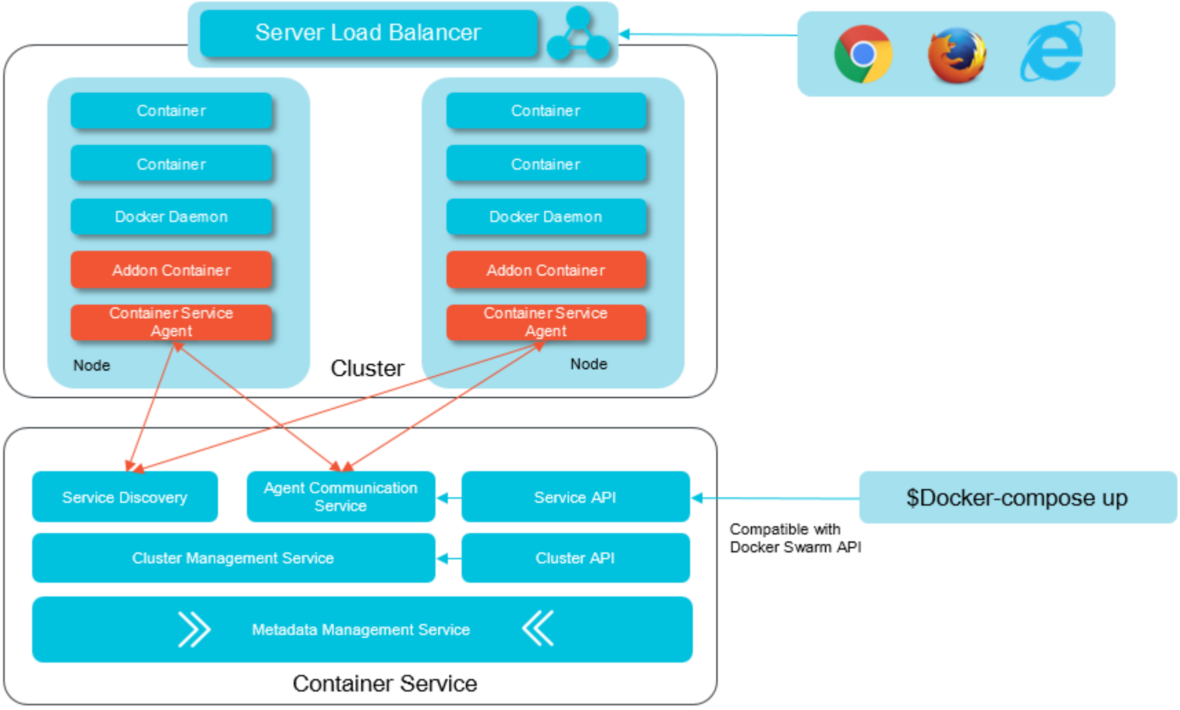
Generic conventions

Table -1: Style conventions

Style	Description	Example
⛔	This warning information indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	⛔ **Danger:** Resetting will result in the loss of user configuration data.
⚠	This warning information indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	⚠ **Warning:** Restarting will cause business interruption. About 10 minutes are required to restore business.
📋	This indicates warning information, supplementary instructions, and other content that the user must understand.	📋 **Note:** Take the necessary precautions to save exported data containing sensitive information.
	This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user.	📋 **Note:** You can use **Ctrl** + **A** to select all files.
>	Multi-level menu cascade.	**Settings** > **Network** > **Set network type**
Bold	It is used for buttons, menus, page names, and other UI elements.	Click **OK**.
`Courier font`	It is used for commands.	Run the `cd /d C:/windows` command to enter the Windows system folder.
`Italics`	It is used for parameters and variables.	`bae log list --instanceid` *`Instance_ID`*
[] or [a|b]	It indicates that it is a optional value, and only one item can be selected.	`ipconfig` *`[-all|-t]`*
{} or {a|b}	It indicates that it is a required value, and only one item can be selected.	`swich` *`{stand | slave}`*

Contents

1 Architecture



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 APIs of Alibaba Cloud are provided.
- **Service API:** APIs that are compatible with Docker Swarm APIs are provided.

2 Benefits

Ease to use

- Supports creating container clusters with one click.
- One-stop application lifecycle management based on containers.
- Integrates with the Alibaba Cloud abilities of virtualization, storage, network, and security.
- Supports graphical user interfaces and APIs.

Secure and controllable

- In Alibaba Cloud Container Service, your containers are running on your own Elastic Compute Service (ECS) instances and are not shared with others. Therefore, isolation issues between containers do not exist.
- In terms of network, you can use a security group to define the access policies of ECS instances and containers in a container cluster, allowing or rejecting addresses of some sources to access the containers.
- Mutual certificate verification is used by the management APIs of container clusters, avoiding the interface from being accessed by illegal users.
- Special solutions for container security, such as neuvector, can be easily integrated with Alibaba Cloud Container Service to provide higher level of security protection.

Protocol compatibility

- Supports both swarm and Kubernetes.
- The first batch to pass the conformance authentication of Kubernetes in the world.
- Supports migrating applications to cloud platforms seamlessly and managing hybrid cloud.

Efficient and reliable

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

3 Scenarios

DevOps continuous delivery

Optimal continuous delivery process

Working with Jenkins, Container Service automatically finishes the complete process of DevOps from code submitting to application deployment, makes sure that only codes passed the automated test can be delivered and deployed, and efficiently replaces the traditional method of complicated deployment and slow iteration in the industry.

Container Service can implement:

- Automation of DevOps.

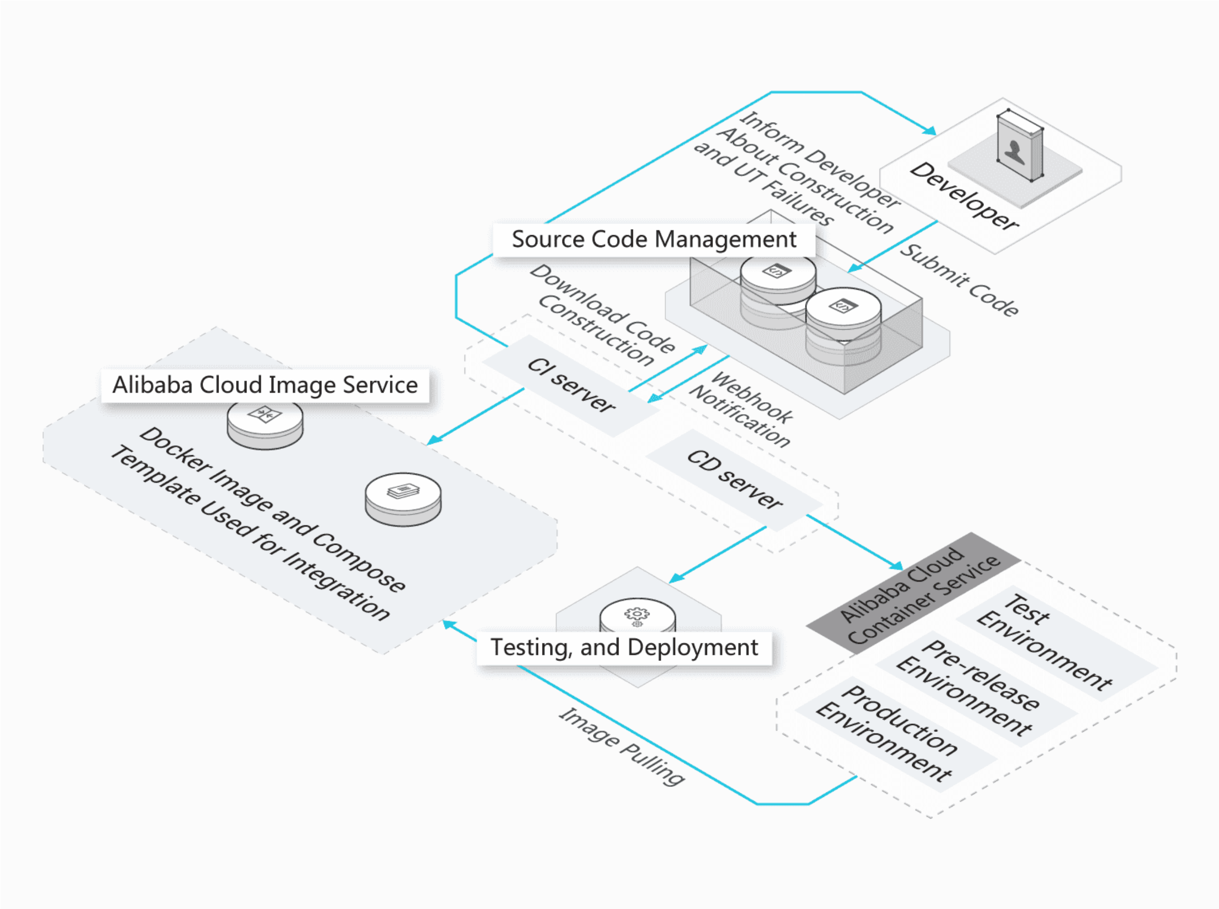
  The automation of the full process from code changes to code building, image building, and application deployment.
- Consistency of environment.

  Container technology allows you to deliver not only the codes, but also the running environmen t based on the immutable architecture.
- Continuous feedback.

  Results of each integration or delivery are fed back in real time.

We recommend that you use

Elastic Compute Service (ECS) and Container Service together.


Inform Developer About Construction and UT Failures
Developer
Source Code Management
Submit Code
Download Code Construction
Webhook Notification
CI server
CD server
Alibaba Cloud Image Service
Docker Image and Compose Template Used for Integration
Alibaba Cloud Container Service
Test Environment
Pre-release Environment
Production Environment
Testing, and Deployment
Image Pulling

Microservice architecture

Implement agile development and deployment to accelerate business iteration of enterprise s

In the production environment of enterprises, microservices are divided reasonably and each microservice application is stored in the Alibaba Cloud image repository. You only have to iterate each microservice application, and Alibaba Cloud provides the capabilities of scheduling, orchestration, deployment, and gated launch.

Container Service can implement:

- Server Load Balancer and service discovery.

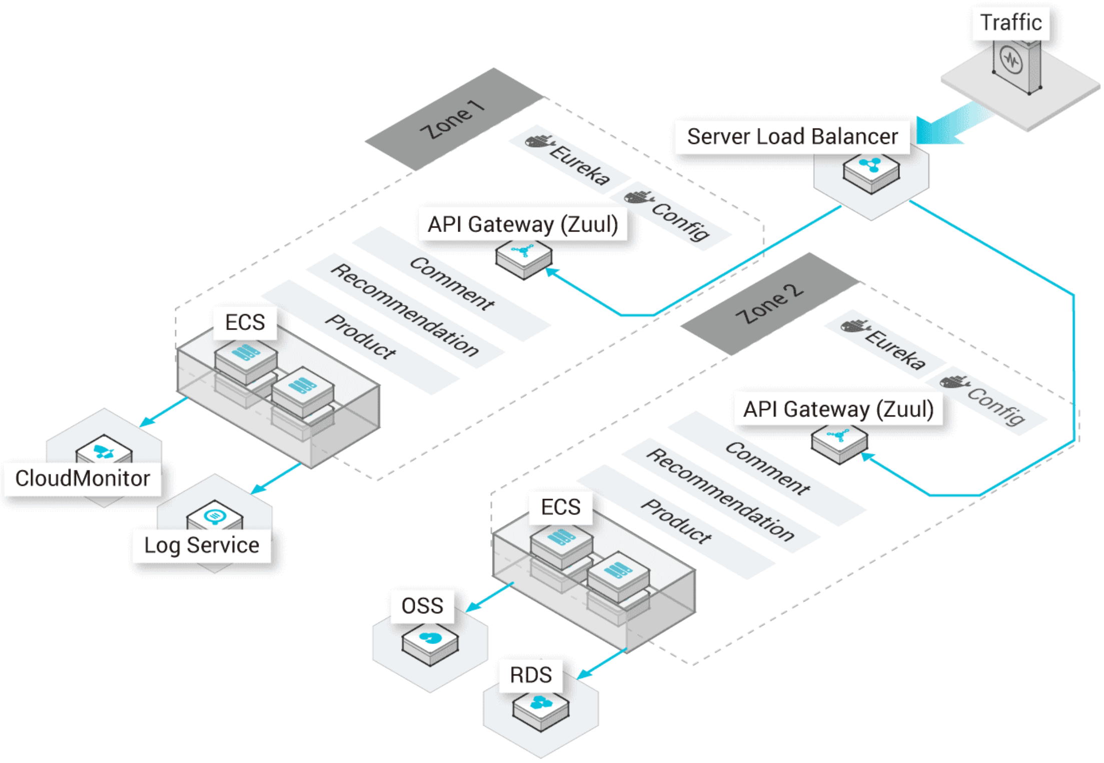
  Supports Layer-4 and Layer-7 request forwarding and backend binding.
- Many policies of scheduling and exception recovery.

  Supports affinity scheduling at the level of services. Supports cross-zone high-availability and disaster recovery.
- Microservice monitoring and auto scaling.

Supports the monitoring at the level of microservices and containers. Supports auto scaling of microservices.

We recommend that you use

ECS, Relational Database Service (RDS), Object Storage Service (OSS), and Container Service together.



Hybrid cloud architecture

United Operation and Maintenance (O&M) of multiple cloud resources

Manage resources on and off the cloud at the same time in the Container Service console, without switching between multiple cloud consoles. Deploy applications on and off the cloud at the same time by using the same image and orchestration based on the characteristics unrelated to the container infrastructure.

Container Service supports:

- Scaling in and out applications on the cloud.

Expand the capacity rapidly on the cloud at the business peak period to bring some business traffic to the cloud.

- Disaster recovery on the cloud.

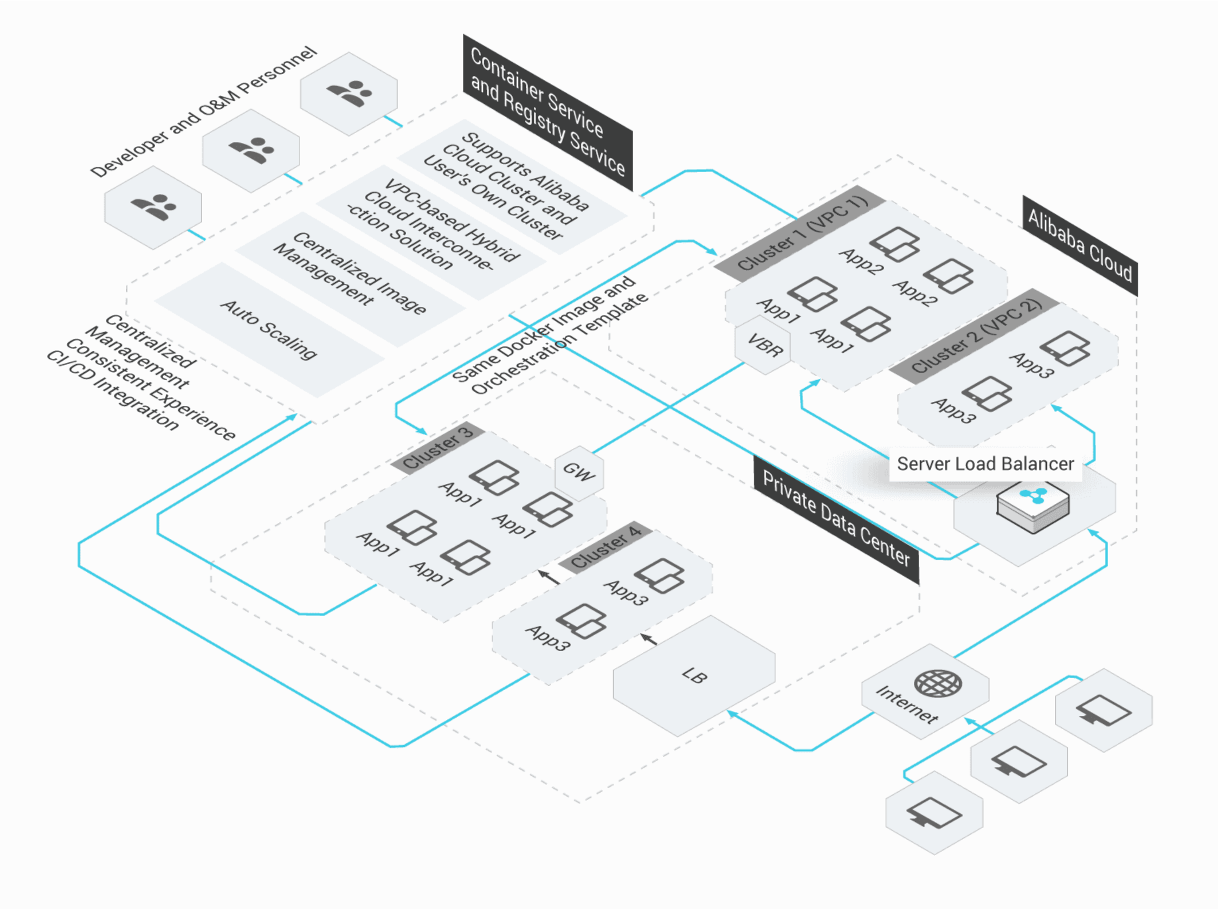
  Deploy business systems on and off the cloud at the same time to provide services off the cloud and provide disaster recovery on the cloud.

- Development and test off the cloud.

  Release the applications seamlessly on the cloud after the development and test off the cloud.

We recommend that you use

ECS, Virtual Private Cloud (VPC), and Express Connect together.


Developer and O&M Personnel
Container Service and Registry Service
Supports Alibaba Cloud Cluster and User's Own Cluster
VPC-based Hybrid Cloud Interconne--ction Solution
Centralized Image Management
Auto Scaling
Centralized Management
Consistent Experience
CI/CD Integration
Same Docker Image and Orchestration Template
Cluster 1 (VPC 1)
App2
App2
App1
App1
VBR
Alibaba Cloud
Cluster 2 (VPC 2)
App3
App3
Server Load Balancer
Cluster 3
App1
App1
App1
App1
GW
Private Data Center
Cluster 4
App3
App3
LB
Internet

Auto scaling architecture

Automatic expansion/contraction for the business according to the business traffic

Container Service can automatically expand or contract the business according to the business traffic, without manual intervention. In this way, the system is not down because of traffic surge and not timely expansion, and the waste due to a large number of idle resources is avoided.

Container Service can implement:

- Rapid response.

  Trigger the container expansion in seconds when the business traffic reaches the expansion indicator.

- Full automation.

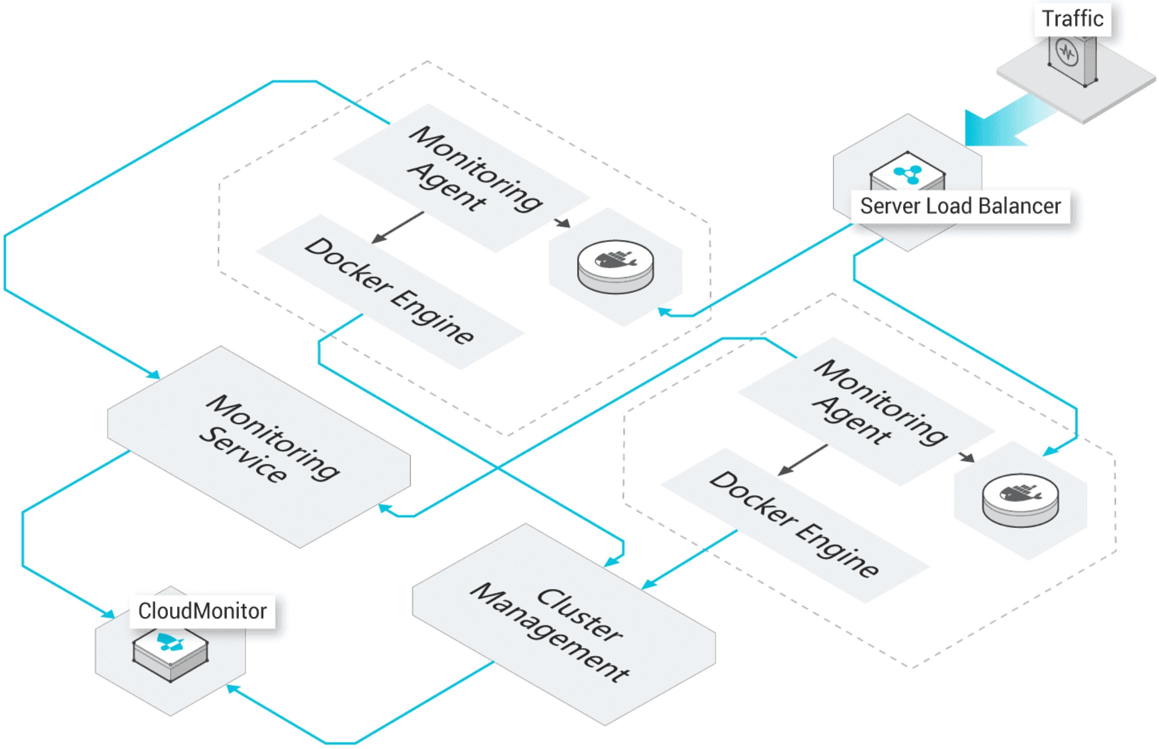
  The expansion or contraction process is fully automated, without manual intervention.

- Low cost.

  Contract the capacity automatically when the traffic is reduced to avoid the waste of resources.

We recommend that you use

ECS and CloudMonitor together.



4 Limits for swarm clusters

The limits for Alibaba Cloud Container Service swarm clusters are as follows.

Cluster

- By default, you can create at most five clusters in all regions and add at most 20 worker nodes in each cluster. To create more clusters or add more nodes to a cluster, open a ticket.
- The Elastic Compute Service (ECS) instances and Server Load Balancer instances created with clusters only support the Pay-As-You-Go billing method.

Expand a cluster

The nodes added by expanding a cluster are Pay-As-You-Go nodes.

Add an existing ECS instance

- The ECS instance to be added must be in the same region and use the same network type (Virtual Private Cloud (VPC)) as the cluster.
- When adding an existing ECS instance, make sure that your ECS instance has an Elastic IP (EIP) for the network type VPC. Otherwise, the ECS instance fails to be added.
- The ECS instance to be added must be under the same account as the cluster.

Bind a Server Load Balancer instance

- You can only bind a Server Load Balancer instance to a cluster of the same region.
- You can only bind a Server Load Balancer instance to a cluster created by the same account.
- A VPC cluster can bind an Internet Server Load Balancer instance or an intranet Server Load Balancer instance.
- One cluster can only bind one Server Load Balancer instance.
- Two clusters cannot share one Server Load Balancer instance.

5 Terms

Basic terms

Cluster

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 server (either an Elastic Compute Service (ECS) instance or a physical server) that has a Docker Engine installed 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 number of nodes in a cluster is scalable.

Container

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

Image

A standard packaging format of a container application in Docker. You can specify an image to deploy containerized 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 (latest by default).

Orchestration template

A template type that contains definitions of a group of container services and their interconnecting relationships, which can be used to deploy and manage multiple container applications. Container Service supports and extends the Docker Compose template specifications.

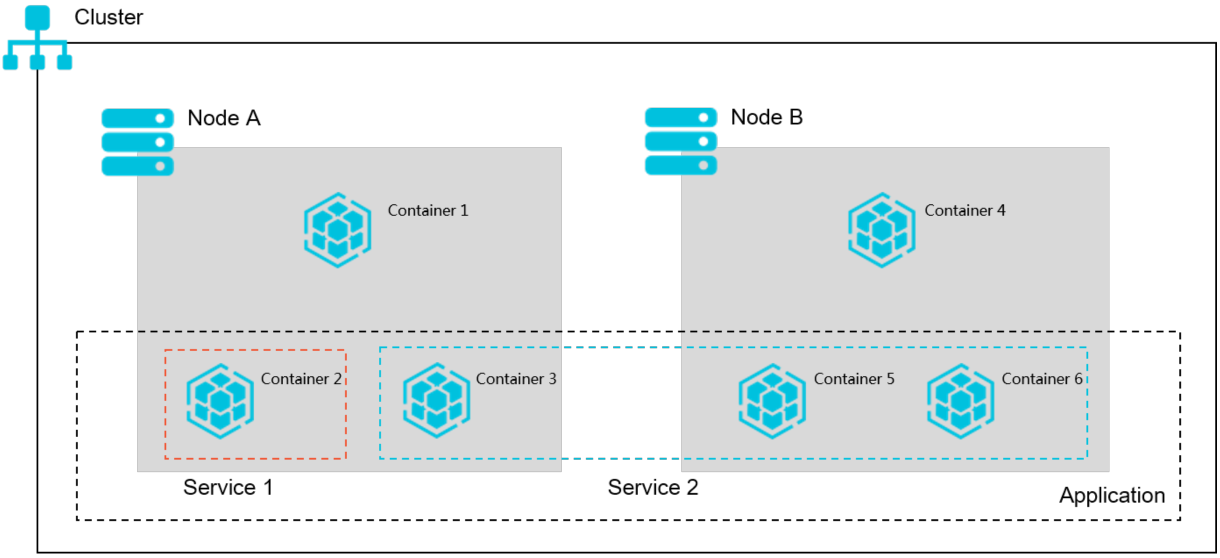
Application

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

Service

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

Associations



Kubernetes terms

Kubernetes, an open-source and large-scale container orchestration and scheduling system of Google, is used to automatically deploy, expand, and manage containerized applications and has the characteristics such as portability, scalability, and automatic scheduling.

Node

Nodes in Kubernetes clusters provide the computing power and are the active servers where all the pods are running. The active servers can be physical machines or virtual machines and containers that are running on the kubelet management nodes must run on the active servers.

Namespace

The namespace provides the virtual isolation for Kubernetes clusters. By default, Kubernetes clusters have three namespaces: the default namespace default, and the system namespaces kube-system and kube-public. The administrator can also create new namespaces to meet the requirements.

Pod

The minimum basic unit of Kubernetes that is used to deploy applications or services. A pod can encapsulate one or more containers, storage resources, an independent network IP address, and policy options of managing and controlling the running method of containers.

Replication Controller

Replication Controller (RC) makes sure that a specified number of pod replicas are running in a Kubernetes cluster at any time by monitoring the running pod. One or more pod replicas can be specified. If the number of pod replicas is less than the specified number, RC starts to run new pod replicas. If the number of pod replicas exceeds the specified number, RC starts to stop the redundant pod replicas.

Replica Set

The upgraded version of RC. The only difference between Replica Set (RS) and RC is the support for selector. RS supports more types of matching modes. Generally, the RS objects are not used independently, but are used as the deployment parameters in the ideal status.

Deployment

The deployment indicates an update operation for a Kubernetes cluster by users, has a wider application range than RS, and can create a service, update a service, and perform a rolling update of a service. Performing a rolling update of a service actually creates a new RS, gradually adds the number of replicas in the new RS to the ideal status, and reduces the number of replicas in the old RS to zero. Such a compound operation cannot be described well by an RS, but can be described by a more common deployment. We do not recommend that you manually mange and use the RS created by the deployment.

Service

The basic operation unit of Kubernetes. As the abstract of real application service, each service has many containers to provide the support. The port of Kube-Proxy and service selector determine the service request to pass to the backend container, and a single access interface is displayed externally. The external is not required to know how the backend works, which is good for expanding or maintaining the backend.

Labels

Essentially a collection of key-value pairs that are attached to the resource objects. Labels are used to specify the attributes of objects that are meaningful for users, but do not have any direct significance for kernel systems. You can add a label directly when creating an object or modify the label at any time. Each object can have more labels, but the key value must be unique.

Volume

The volumes in Kubernetes clusters are similar to the Docker volumes. The only difference is that the range of Docker volumes is a container, while the lifecycle and range of Kubernetes volumes are a pod. The volumes declared in each pod are shared by all the containers in the pod. You can use the Persistent Volume Claim (PVC) logical storage, and ignore the actual storage technology in the backend. The specific configurations about Persistent Volume (PV) are completed by the storage administrator.

PV and PVC

PV and PVC allow the Kubernetes clusters to have the abstract logical capabilities of storage, and you can ignore the configurations of actual backend storage technology in the pod configuration logic, leaving the configurations to the PV configurator. The relationship between PV and PVC of storage is similar to that between node and pod of computing. PV and node are the resource provider, changed according to the cluster infrastructure, and configured by the Kubernetes cluster administrator. PVC and pod are the resource user, changed according to the business and service requirements, and configured by the Kubernetes cluster user, namely, the service administrator.

Ingress

A collection of rules that authorize the inbound access to the cluster. You can provide the externally accessible URL, Server Load Balancer, SSL, and name-based virtual host by using the Ingress configurations. You can request the Ingress by posting Ingress resources to API servers . The Ingress Controller is used to implement Ingress by using Server Load Balancer generally, and can configure the edge router and other frontends, which helps you handle the traffic in the HA method.

Related documents

- *Docker glossary*
- *Kubernetes concepts*