Container Service

Product Introduction

Product Introduction

What is Container Service

Container Service provides the high-performance and scalable container application management service, which enables you to manage the lifecycle of containerized applications by using Docker and Kubernetes.

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 containers.

Features

Cluster management and flexible choices of regions and network environments

Container Service allows you to:

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

Multiple server hosting modes

Container Service allows you to:

- Grant 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

Container Service provides the following features:

Network

Container Service supports intercommunication between containers across hosts and high-performance VPC network drives.

Storage

Container Service supports data volume management. OSSFS, cloud disks, and Network Attached Storage (NAS) are supported.

Log

Container Service supports automatic log collection and integrating with Alibaba Cloud Log Service. You can also integrate Container Service with third-party open-source log solutions.

Monitoring

Container Service supports the monitoring at the level of containers and virtual machines (VMs). You can also integrate Container Service with third-party open-source monitoring solutions.

Scheduling

Container Service supports the policies such as cross-zone high availability and rescheduling of abnormal nodes.

Routing

Container Service supports Layer-4 and Layer-7 request forwarding and backend binding.

Permission

Container Service supports Resource Access Management (RAM) authorization and management at the level of clusters.

Support for swarm and Kubernetes

Container Service supports both swarm and Kubernetes, migrating existing systems seamlessly from offline to cloud.

Unique value-added ability and better experience of Alibaba Cloud environment

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.

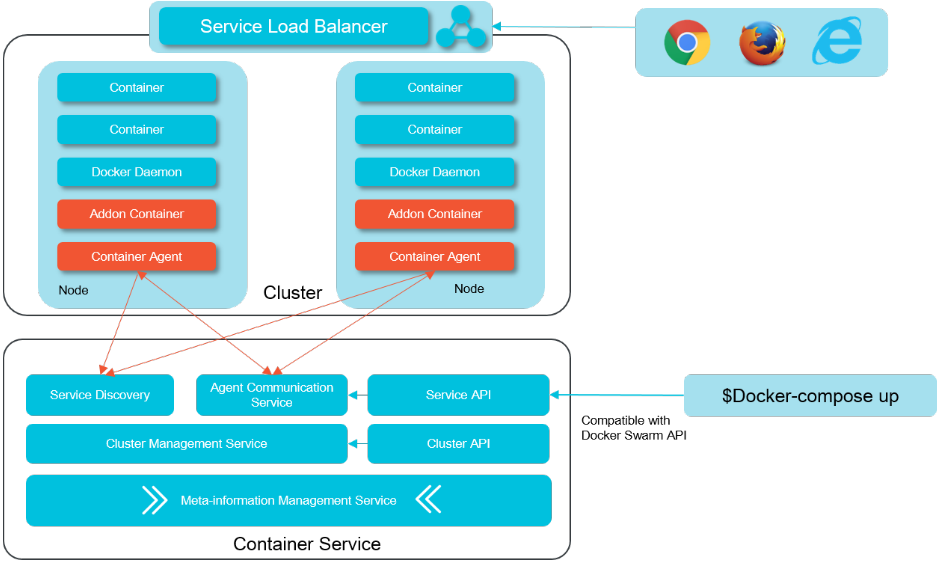
High-availability scheduling policy

Container Service can easily deal with the upstream and downstream delivery process by using the high-availability scheduling policy.

Container Service supports:

- Affinity strategy and horizontal scaling of services.
- Cross-zone high availability and disaster recovery.
- The APIs for cluster and application management to easily interconnect with the continuous integration and private deployment system.

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

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.

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 environment based on the immutable architecture.

- Continuous feedback.

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

We recommend that you use Elastic Compute Service (ECS) and Container Service together.

Microservice architecture

Implement agile development and deployment to accelerate business iteration of enterprises

In the production environment of enterprises, microservices are divided reasonably and each microservice application is stored in the Alibaba Cloud image repository. You only need 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 architecture.

Container Service supports:

- Scaling 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.

- Developing and testing 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.

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 will not be 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.

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

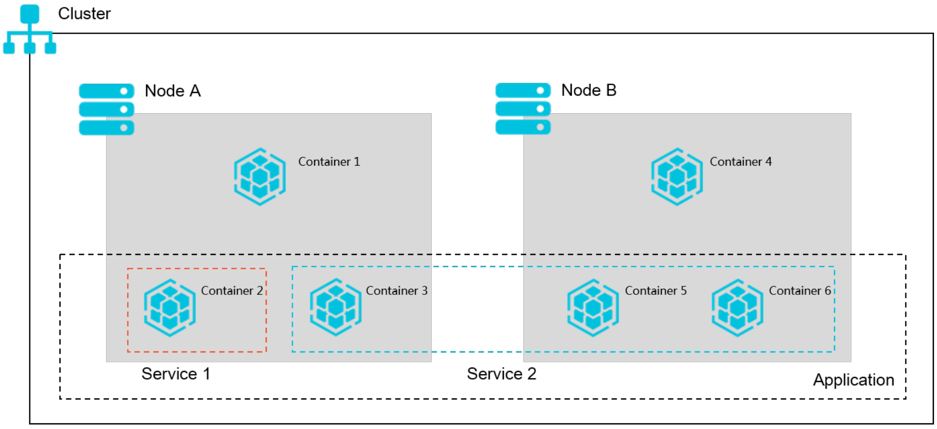
Application software that can be created by using an image or an orchestration template. Each application can contain one or more services.

Service

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

Associations

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



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

Limits

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 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 node

- 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.

Limits for Kubernetes clusters

The limits for Alibaba Cloud Container Service Kubernetes 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.

Currently, Kubernetes clusters only support Linux containers. The support for Windows containers is in the works.

Currently, Kubernetes clusters only support Virtual Private Cloud (VPC). You can select to create a VPC or use an existing VPC when creating a Kubernetes cluster.

Currently, all the created cluster resources, such as Elastic Compute Service (ECS) instances and Server Load Balancer instances, are Pay-As-You-Go resources. You can change the Pay-As-You-Go instances to monthly or yearly subscription instances in the console. Monthly or yearly subscription resources will be supported in the future.

ECS instances

Only support the CentOS operating system.

Only support creating Pay-As-You-Go ECS instances.

Limits for adding an existing node:

> The ECS instance to be added must be in the same region and VPC as the cluster.
>
> When adding an existing ECS instance, make sure that your ECS instance has a public IP if the network type is classic or an Elastic IP (EIP) if the network type is VPC, or the corresponding VPC has configured the NAT gateway. In short, make sure the corresponding node can access public network normally. Otherwise, the ECS instance fails to be added.
>
> The ECS instance to be added must be under the same account as the cluster.

Cluster expansion and contraction

The number of worker nodes must be from 1 to 20.

Currently, you can only expand or contract a cluster manually. Auto scaling is not supported.

Currently, master nodes in a Kubernetes cluster cannot be expanded automatically.

According to the Resource Orchestration Service (ROS) rules, elastic contraction will not contract the nodes created when the cluster is created or the nodes manually added to the cluster, but will only contract the nodes manually expanded by you. The contraction rule is based on the creation time. The later the node is added by expansion, the earlier the node is recycled.

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

Release notes

2017-06

[New function] Swarm mode clusters are open to all regions.

- Supports swarm mode, routing mesh, and service-level commands.
- To create your swarm mode cluster, see Create a cluster.

2017-05

[New product] Deep learning solution goes online.

- Build a deep learning lab on the cloud with one click in Container Service.
- Supports development, training, and prediction of models. See Deep learning overview.

[New function] Supports managing networks and secret keys.

- Supports managing networks and secret keys.
- Swarm mode clusters support independent network management. Applications and secret keys are managed separately to enhance the application security.

[New region] Adds the regions Hong Kong and China North 3.

2017-04

[New function] Swarm mode clusters go online.

- Use native IPVS to improve the routing performance between services. Supports service-level

commands. Currently swarm mode clusters are open to regions US West and China East 2.
- For more information about swarm mode clusters, see Swarm mode cluster overview.

[New region] Adds the Singapore region.

2017-03

[New function] Supports upgrading to Docker 17.03.1.

[New function] Supports hybrid cloud Calico 2.0.

2017-02

[New function] Supports upgrading to Docker Registry 2.6.0.

- Container Service supports the latest Docker Registry version 2.6.0, which provides optimized stability and performance.
- We recommend that you upgrade Docker to 1.12.6. For how to upgrade Docker, see Upgrade Docker daemon.

[New function] Supports the Operation Logs function.

- You can view operation history in the operation logs to troubleshoot and analyze problems.
- Log on to the Container Service console. Click **Operation Logs** in the left-side navigation pane.

[New region] Adds the Japan region.

- The data center of Japan region is in Tokyo.

2017-01

[New function] Supports upgrading to Docker 1.12.5.

- Container Service supports the latest Docker version 1.12.5. The latest version provides better stability and optimized network performance.
- For how to upgrade your Docker, see Upgrade Docker daemon.

[New region] Adds the Hong Kong region.

- You can open a ticket to apply for whitelist access.

[New function] Supports Compose V2 file format.

- Supports visualization resolution and verification.

[New function] Upgrades logging function.

- Supports distinguishing hostname to help locate the problem.
- Upgrade ilogtail to 0.11.6. For how to upgrade system services, see Upgrade system services.

[New function] Supports selecting whether or not to disable system services when you create a cluster.

- The system services include simple routing service, volume service, log service, and monitor service.

2016-12

[New function] Supports configuration management.

[Function optimization] Supports Compose 1.9 and Registry 2.5.1.

2016-11

[New function] Supports upgrading to Docker 1.12.

- Container Service supports the latest Docker version 1.12.3. The latest version provides better stability and optimized network performance.
- For how to upgrade your Docker, see Upgrade Docker daemon.

[New function] The international site goes online.

- Overseas users can use services in Container Service of six regions to assist the microservice architecture.
  To use Container Service, overseas users must clear the cookies and visit https://www.alibabacloud.com/product/container-service.

2016-10

[New region] Adds the Shanghai and US West regions.

- US West, the first overseas node of Container Service, goes online. Using together with the overseas building service has a better performance.

[New function] Supports node auto scaling.

- For scenarios such as seckill or big sale, you must expand the container clusters elastically. In earlier versions, Container Service supports container auto scaling. In this version, Container Service supports node auto scaling.

- For more information, see Node auto scaling.

[New function] Supports timing tasks.

- Container Service supports timing task definition by using Docker Compose.
- For more information, see Timing tasks.

[New function] Integrates with CloudMonitor.

- Besides the Container Service console, you can also use the CloudMonitor console to view the monitoring information of the clusters, nodes, and containers.
- For more information, see Container monitoring service.

[New function] Supports selecting to automatically migrate containers when removing or resetting nodes.

- When taking offline or resetting an ECS instance, you can first migrate the containers of this instance to another instance to guarantee the smooth online business.
- For more information, see Node management.

2016-08

[New region] The Hangzhou finance cloud goes online.

[New function] Supports application scheduling of offline computing.

[New function] Supports container auto scaling.

[New function] Supports virtual network operators.

2016-07

[New region] Adds the Qingdao region.

[New function] Supports custom routing.

[New function] Supports blue-green release for applications.

2016-06

[New function] Supports Docker 1.11.

[Function optimization] Simplifies creation procedure and optimizes creation procedure of zero-node clusters.

[New function] Supports binding a Server Load Balancer instance and setting cluster root domain name.

[New function] Supports APIs for image management.

[New function] Supports triggers for resource scaling.

[New function] Supports dynamically adjusting the resource limit for services and containers.

2016-05-05 Container Service is commercially available.

[New function] Supports Docker 1.10.

[Function optimization] Supports upgrading system services.

[New function] Unifies classic and VPC network models and supports mutual access with domain names defined by container name or hostname.

[New function] Supports automatically collecting logs.

[New function] Supports the monitoring at the level of containers and virtual machines.

[New function] Supports the rescheduling policies of cross-zone high availability and abnormal nodes.

[New function] Supports Layer-4 and Layer-7 request forwarding and backend binding.

[New function] Supports authorizing and managing cluster-level sub-accounts.

2016-03

[New function] Supports upgrading Ubuntu kernel of special versions.

[New function] Supports resetting and removing nodes.

[New function] Supports deleting containers.

[New function] Supports creating and redeploying trigger APIs.

[New function] Adds data volume management and supports OSSFS.

[New function] Adds custom Server Load Balancer.

[New function] Supports updating RAM authorization information.

2016-02

[New function] Supports adding existing ECS instances to clusters by using scripts.

[New function] Supports editing cluster names.

[New function] Supports aggregating cluster status and automatically refreshing list data.

[New function] Adds links to applications and services in cluster list.

[New function] Adds two dimensions for container monitoring.

[New function] Provides the force stop command for applications and services.

[New function] Supports activating and stopping the system application routing.

[New function] Supports prompting you to activate RAM sub-account when a cluster is created.

2016-01

[New region] Adds the Shenzhen region.

[New function] Supports CentOS operating system.

[New function] Supports displaying Docker labels of machine nodes.

[New function] Supports customizing system disk type and data disk type.

[Function optimization] Supports optional EIP.

[New function] Supports adding existing ECS instances to clusters by resetting images.

[New function] Supports pulling the latest image when deploying images of the same tag.

[New function] Supports configuring the container memory limit.

[New function] Supports parameterized deployment for Compose template.

[Function optimization] Supports deleting scrolls when deleting applications and services.

2015-12

The beta product version goes online.