Tencent Kubernetes Engine

Product Introduction

Service Notice

This document provides an overview of the as-is details of Tencent Cloud's products and services in their entirety or part. The descriptions of certain products and services may be subject to adjustments from time to time.

The commercial contract concluded by you and Tencent Cloud will provide the specific types of Tencent Cloud products and services you purchase and the service standards. Unless otherwise agreed upon by both parties, Tencent Cloud does not make any explicit or implied commitments or warranties regarding the content of this document.

Contact Us

We are committed to providing personalized pre-sales consultation and technical after-sale support. Don't hesitate to contact us at 4009100100 or 95716 for any inquiries or concerns.

Contents

Product Introduction
Overview

Last updated: 2023-09-26 17:34:22

TKE Overview

TKE(Tencent Kubernetes Engine) offers a container-centered, highly scalable, and high-performance container management service based on native Kubernetes. It is closely connected to Tencent Cloud IaaS products to help you quickly implement business containerization.
The following video introduces TKE, its core concepts, and the usage process:
Watch video

Service Description

Tencent Cloud provides multiple container services for you to deploy, manage, and expand containers:

- TKE: It offers a container-centered, highly scalable, and high-performance container management service based on native Kubernetes.
- TKE Serverless Cluster: A service model offered by Tencent Cloud Container Service that allows you to deploy workloads without purchasing nodes.
- TKE Edge: It is a container system launched by TKE that manages edge cloud resources from the central cloud. It provides edge autonomy and distributed health checks.
- TCR: It provides secure and efficient container image management and distribution services. It works with TKE to ensure a smooth experience in migrating containers to the cloud.
- Cloud-native services:
  - Tencent Cloud Service for etcd: It is an etcd management solution based on open-source etcd and optimized for cloud-native scenarios. It is provided by the TKE team and fully compatible with open-source etcd distributed storage capabilities, providing a highly stable, observable, Ops-free cloud-native etcd service.
  - Cloud-native asset management: TKE has introduced the capability of cloud-native asset management, visualizing all resource objects for users. The platform features rich filtering queries, type aggregation, and status display capabilities, assisting users in quickly locating target objects.
  - TMP: It is a monitoring and alarming solution specially optimized for cloud-native service scenarios. It has the full monitoring capabilities of open-source Prometheus and provides lightweight, stable, and highly available cloud-native monitoring services.

- Cloud-native AI: Tencent Cloud Container Service, based on its technical expertise in the cloud-native field, offers a modular, low-coupling, and highly scalable cloud-native AI service.

Notes

TKE allows you to manipulate clusters and services in the TKE console or through TencentCloud APIs.

Via the console

Cluster Type	Description	Use Cases	Documentation
Serverless cluster	It allows you to add and use super nodes, and no cluster management fees will be incurred. You can quickly configure high-specced super nodes to deploy massive security sandbox containers. You can elastically use near-infinite container resources and only need to pay for actual running Pod resources. This easily sustains highly stable online businesses and batch computing businesses, safeguarding stability while slashing costs.	It is suitable for highly stable and secure resident businesses and temporary computing tasks. Security sandbox containers and business containers are strictly isolated, without mutual interference. Tens of thousands of Pods can be started in seconds and will be billed by actual duration. Security sandbox containers can be started very quickly with super nodes.	Create a Serverless Cluster
General cluster	It is the default cluster type and is fully compatible with the standard features of open-source Kubernetes clusters. It enhances node management, cluster network, and container scheduling capabilities. In a single cluster, super nodes, native nodes, general CVM nodes, and IDC nodes can be added and managed at the same time, which means they can be combined for	Suitable for all scenarios, fully compatible with the standard capabilities of open-source Kubernetes clusters. It supports super nodes, native nodes, registered nodes, and CVMs. Resource visualization and optimization analysis are available, making it easy to improve	Creating a Standard Cluster

	different business scenarios to maximize the computing resource utilization.	resource utilization. Standard K8s clusters support super nodes and native nodes.	
Edge cluster	It allows you to add and use edge nodes to quickly extend IDC Kubernetes cluster capabilities to edge regions and manage resources and application lifecycle in the cloud-native method. In addition, the innovative multi-region edge autonomy, closed loop of traffic, and distributed health checks are available.	Edge computing allows for managing edge computing resources in the cloud-native method, such as edge servers and IoT devices, which addresses poor network connections and node autonomy. Multi-region management allows for managing resources in multiple regions in the same cluster, implementing closed traffic loop in the regions. The cloud-native method of edge computing allows for closed traffic loop for multi-region management.	Creating an Edge Cluster
Registered cluster	It allows you to register Kubernetes clusters in your local infrastructure or those of other cloud vendors with TKE for unified management. You can also implement unified management of multi-cloud and multi-cluster resources in the Tencent Kubernetes Engine Distributed Cloud Center.	Multi-cloud management contributes to flexible access and management of various enterprise computing resources. High-availability disaster recovery facilitates the unified governance of multi-cluster applications, traffic, and storage. Automatic release allows for integration to the existing DevOps system to implement multi-cloud release. The	Creating a Registry Cluster

		multi-cloud management ecosystem opens up high-availability disaster recovery.	

Through TencentCloud APIs

For more information on APIs supported by TKE, see API Category.

Strengths

Last updated: 2023-09-15 14:21:16

Orchestration Advantages

Tencent Kubernetes Engine (TKE) is developed on the basis of Kubernetes, a container cluster management system made open-source by Google. Leveraging the Docker technology, Kubernetes provides containerized applications with a complete set of features ranging from deployment and execution and resource scheduling to service discovery and dynamic scaling, making it much easier to manage large-scale container clusters. Kubernetes brings the following benefits:

- Using elegant software engineering consisting of modularization and microservices, Kubernetes implements a modular design that allows you to customize network, storage, scheduling, monitoring, and log modules as needed through flexible plugins.
- The Kubernetes project community acts as an open-source platform for the implementation of container, network, and storage.

TKE vs. Customer Self-Built Container Service

Advantages	TKE	Customer Self-built Container Service
Ease of use	**Simplified cluster management** • TKE has various features such as large-scale container cluster management, resource scheduling, container arrangement, and code construction. It blocks the differences of underlying infrastructures and simplifies management and Ops of distributed applications. You no longer need to use cluster management software or design fault-tolerating cluster structures, thus eliminating all relevant management and scaling workloads. • You just need to enable a container cluster and specify the tasks you want to run, and TKE will help you complete all the cluster management work, enabling you to focus on developing Dockerized applications.	When using a self-built container management infrastructure, you usually need to go through complex management processes such as installing, operating, and scaling your own cluster management software as well as configuring management systems and monitoring solutions.

Flexible scalability	**Flexible cluster management and integration with CLB** • You can use TKE to schedule long-running applications and batch jobs flexibly. You can also use APIs to obtain the latest information about cluster status for easy integration with your customized and third-party scheduling applications. • TKE is integrated with Cloud Load Balancer (CLB), enabling you to distribute traffic among multiple containers. You just need to specify the container configuration and load balancer to be used, and the TKE management application will automatically add/delete resources for you. In addition, TKE can auto-recover faulty containers to guarantee that a sufficient number of containers are always running to sustain your applications.	You need to determine how to manually deploy container services according to the business traffic and health status, which has poor availability and scalability.
High Security and Reliability	**Secure isolation of resources and high availability of services** • TKE works inside your own Cloud Virtual Machine (CVM) instance without sharing computing resources with other customers. • Your clusters run inside Virtual Private Clouds (VPCs) where you can use your own security groups and network ACLs. These features enable a high level of isolation and help you use CVM instances to construct applications with high security and reliability. • TKE uses a distributed service structure to implement auto failover and fast migration for services. Together with distributed backend storage of stateful services, TKE also ensures high security and availability of services and data.	Due to kernel issues and imperfect namespaces of self-built container services, isolation at the tenant, device, and kernel module levels is rather poor.
High efficiency	**Fast image deployment and continuous business integration** • TKE runs inside your VPCs where quality BGP networks ensure fast upload and	The efficiency of using images to create containers cannot be guaranteed as the

	download of images and allow a large number of containers to launch within seconds. This greatly reduces operational overheads and enables you to focus on business operations. • You can deploy your businesses on TKE. After code is submitted to GitHub or other code hosting platforms, TKE can immediately create, test, pack, and integrate code and deploy the integrated code in pre-release and production environments.	network quality of self-built container services may fluctuate significantly.
Low Costs	**High cost-effectiveness** A TKE managed cluster is more cost-effective than a self-deployed or self-built cluster. You can get a highly reliable, stable, and scalable cluster management plane at low costs and do not need to care about Ops.	You need to invest a lot of money to build, install, operate, and scale out your cluster management infrastructure.
Cloud native	**Cloud-native scenario optimization** • TKE introduces native nodes, a new type of nodes designed for Kubernetes environments. Drawing on Tencent Cloud's expertise in managing millions of container cores, native nodes offer users native, highly stable, and responsive Kubernetes node management capabilities. • Kernel optimization is made, rendering the service highly suitable for cloud-native scenarios.	Despite Kubernetes shielding the underlying infrastructure, adapting to the underlying architecture is necessary in the development process because the fundamental resources cannot be modified.
Improved efficiency	**FinOps implementation** Introducing the cloud-native asset management platform, which assists users in comprehending cost distribution and resource usage from various aspects such as cost insights, job scheduling, and fine-grained scheduling, ensuring that every expenditure in the cloud is maximized in value.	The elastic tools provided are poorly usable due to the difficulty in configuration and slow response. Furthermore, their visualization capabilities are insufficient.
Serverless	**Serverless deployment**	Self-building is a complex and resource-

	Super nodes are a brand-new node product form offered by Tencent Cloud, providing users with availability zone-level, customizable node capabilities. Using a super node is similar to using an extra-large CVM, making resource management and scaling much simpler.	intensive process. Maintaining self-built container services are challenging and such services cannot be truly serverless.

TKE Monitoring vs. Customer Self-built Container Monitoring

TKE monitoring collects and displays comprehensive statistics of around 30 metrics such as cluster, service, Pod, and container, allowing you to check cluster health and create alarms accordingly. In addition, more metrics will be available soon.

Advantages	TKE	Customer Self-built Container Service
Complete set of metrics	Approximately 30 metrics are available, including cluster, node, service, container, and Pod.	Only a few metrics are available and in-house development is required.
Low construction cost	TKE monitoring is provided when a cluster is created.	Manual construction of monitoring is required and can be expensive.
Low Ops cost	Ops is performed by the platform with guaranteed data accuracy.	Manual Ops is required.
Low storage cost	The data of each metric in the past three months is retained free of charge.	Fees are charged based on the storage size.
High scalability	TKE continues to improve and add new metrics.	Developers are required to construct new metrics.
Alarm	Yes	-
Troubleshooting	Container logs can be viewed in the console and WebShell can be used to quickly log in to containers for troubleshooting.	You need to manually log in to containers or servers for troubleshooting.

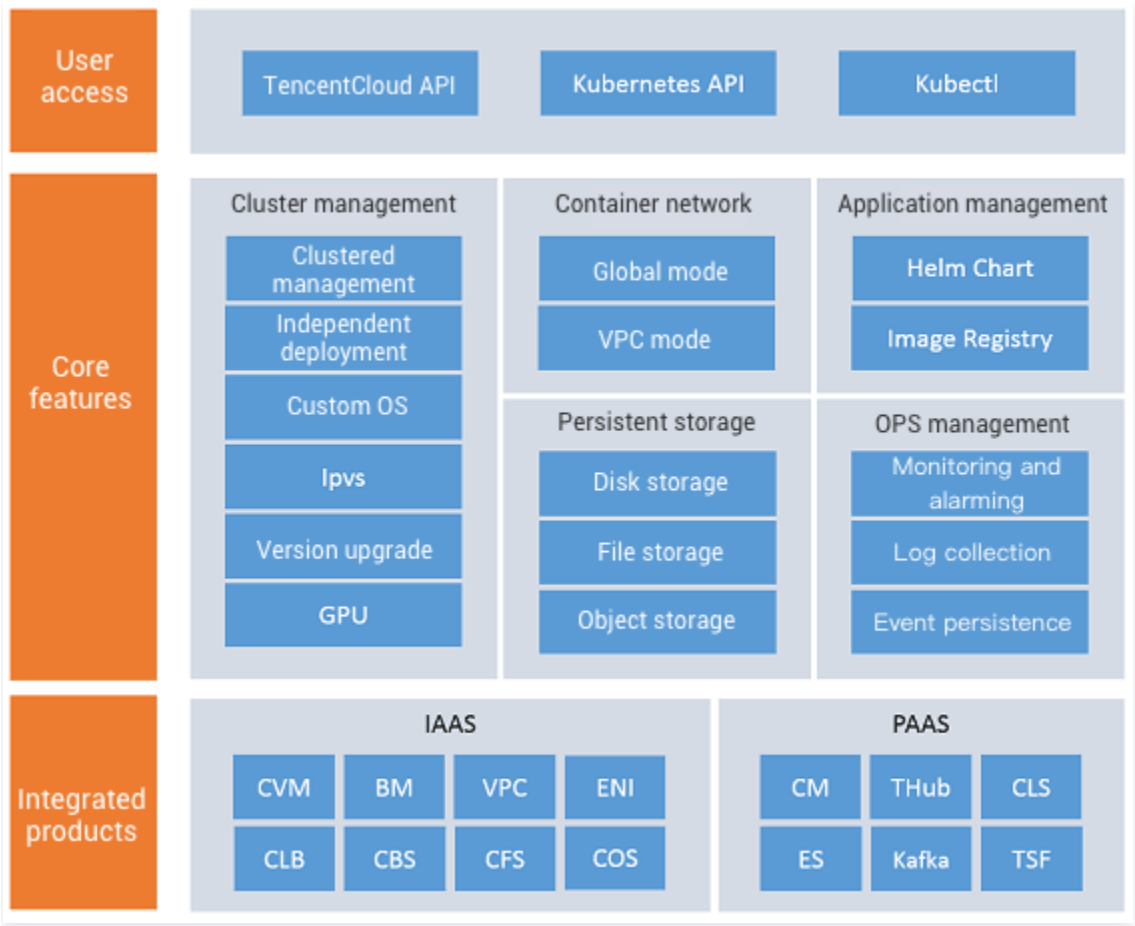
Architecture

Last updated: 2023-09-15 14:21:27

The following video introduces the architecture and related features of Tencent Cloud Container Service:
Watch video

Overall Structure

This section introduces the design and implementation of Tencent Cloud Container Service system. The product architecture is shown in the following figure:



Architecture

1. Tencent Cloud Container Service is adapted and enhanced based on native Kubernetes, supporting the capabilities of native Kubernetes.
2. Tencent Cloud's Kubernetes plugins are available to help you quickly build Kubernetes clusters in Tencent Cloud.
3. Tencent Cloud Container Service, built on top of Kubernetes, offers advanced capabilities such as cluster management, application management, and CI/CD.

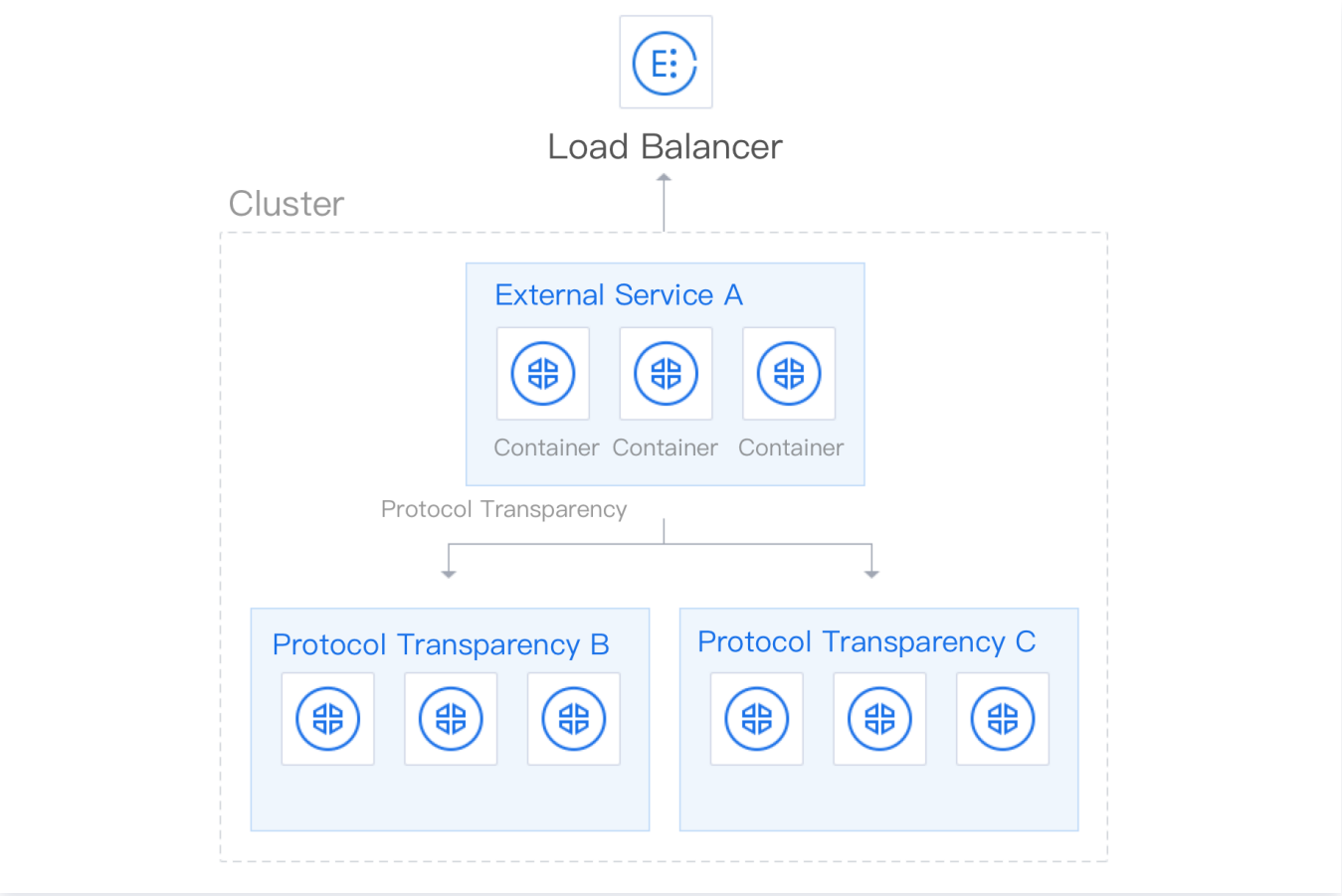
Scenarios

Last updated: 2023-10-08 15:47:05

The following video presents typical use cases of Tencent Kubernetes Engine:
Watch video

Micro-service Architecture

Micro-service architecture is suitable for building complex applications by decomposing monolithic applications into multiple microservices along various dimensions. Each microservice's content is managed using a Docker image. The micro-service architecture is illustrated in the following diagram:



Features and Advantages of Deploying Microservices with Tencent Kubernetes Engine

- Simplifies cluster management, eliminating the need for installation and administration of operational clusters.
- Seamlessly integrates Tencent Cloud's computing, networking, storage, monitoring, and security capabilities, directly utilizing Tencent Cloud's IaaS capabilities.

- Supports service orchestration, granular management of applications, easy-to-understand, secure isolation of resources, and high availability of services.

Continuous Integration and Continuous Delivery (CI/CD)

Offers an excellent DevOps environment through continuous integration and continuous delivery, significantly improving software release efficiency.

Continuous Integration

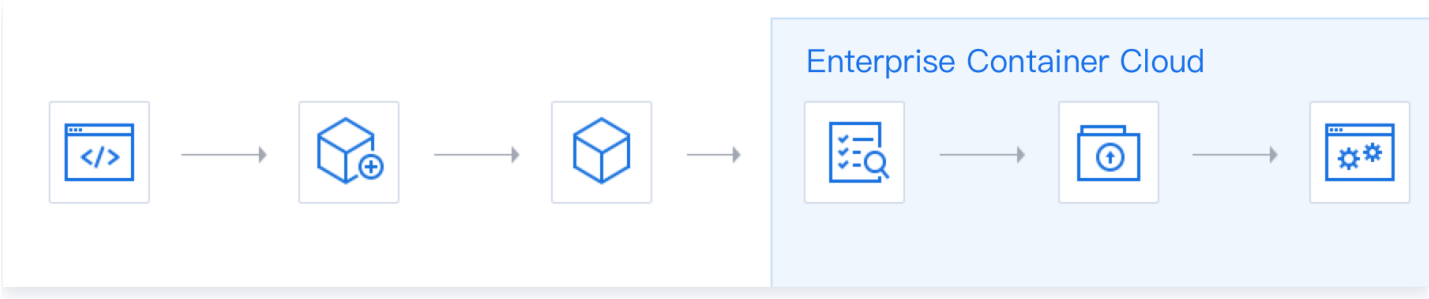
Emphasizes immediate build and (unit) testing after developers submit new code. Based on the test results, you can determine whether the new code can be correctly integrated with the existing code.

Continuous Delivery

Upon continuous integration, deploy the integrated code to the runtime environment.

Advantages

By deploying services on Tencent Kubernetes Engine, developers can immediately build, test, and package integration after submitting new code to Git or other code platforms. Based on continuous integration, the integrated code is deployed to pre-release and production environments.
The Continuous Integration and Continuous Delivery process is illustrated in the following diagram:



Rapid business migration to the cloud

Individuals or enterprises migrating their businesses to Tencent Cloud can choose Tencent Kubernetes Engine to simplify cloud configuration, cluster management, and accelerate service delivery rates. Tencent Kubernetes Engine allow you to create services with a single click, quickly achieving application containerization deployment while also providing elastic scaling, on-demand deployment, high availability, easy expansion, developer-friendly features, and reduced labor costs. As illustrated in the following diagram:

Features

Last updated: 2023-09-15 14:21:49

Cluster Management

TKE allows you to manage your container clusters easily and efficiently while ensuring security and reliability, enabling seamless integration with Tencent Cloud's computing, storage, and networking capabilities.

Module	Feature
Cluster composition	• Introducing native nodes and super nodes based on the FinOps theory, empowering users to reduce costs and improve efficiency. • Supports most CVM models, allowing you to add new and existing hosts. For a complete list of supported models, please refer here. • Support for node registration allows users to host non-Tencent Cloud hosts in TKE clusters. • Cross-availability zone deployment of hosts in a cluster • Exclusive cluster ownership and secure VPC isolation • Cluster network customization and flexible pod network configuration
Cluster Management	• Dynamic cluster scaling, node upgrade/downgrade, and cluster upgrade • Support for cluster registration and unified management of and access to different types of clusters, including clusters on Tencent Cloud, clusters hosted by third-party cloud service providers, customers' self-managed clusters, and edge clusters • Diverse monitoring metrics and custom alarm policies • Offers kernel-optimized images specifically designed for container scenarios. • Custom Kubernetes Component Launch Parameters
Kubernetes management	• Support for multiple Kubernetes versions and version upgrade • Kubernetes certificate management and cluster operations using kubectl • Dozens of auto scaling metrics with powerful elastic capabilities • Resource object browser for viewing all resource objects in the cluster including those with custom resource definition (CRD) • Various Layer-7 access capabilities • Over a dozen of add-ons for enhancing cluster performance

Application Management

The application management feature of TKE can help you quickly create multiple services and deploy applications in different operating environments.

Module	Feature
Application composition	• Multiple TKE service types • Multiple types of resources such as Kubernetes Deployment and DaemonSet
Application Management	• Fast application creation based on custom templates or ready-to-use templates on the template market • Real-time comparison for application update • Quick deployment/stop of services in applications
Template management	• Custom templates and ready-to-use templates on the template market • Quick template copying

Service management

Service management provides an efficient container management solution including various features such as quick creation of service, quick scaling, load balancing, service discovery, service monitoring, and health check, making it easier for you to manage your containers.

Module	Feature
Service Deployment	• Service deployment with a single instance and multiple containers • Multiple service access modes • Cross-availability zone deployment of instances in a service • Configuration of affinity and anti-affinity scheduling
Service management	• Rolling update and quick update • Dynamic scaling • Remote login to service containers
Service operations	• Viewing of detailed monitoring metrics • Viewing of stdout and stderr logs of service containers • Configuration of service alarm policies • Two health check modes: liveness probe and readiness probe • Automatic restoration after container exceptions

ConfigMap Management

ConfigMaps are used to specify the read-in settings of some programs when they are started. You can use different ConfigMaps for different objects.

Module	Feature
ConfigMap Management	• Multiple ConfigMap versions • Visualized configuration mode and YAML-based configuration mode
ConfigMap use	• ConfigMaps mounted to container directories as volumes • ConfigMaps imported as environment variables • ConfigMaps replacing application template variables

Image Management

Tencent Cloud Image Registry contains official Docker Hub images and private images. Image management enables you to quickly create images and deploy services.

Module	Feature
Image Management	• Creation of private image repositories • Viewing and use of DockerHub image repositories • Management of multiple image namespaces
Image use	• Image creation through high-speed private network channels • Image upload and download over public networks
CI/CD	• Configuration of automatic construction of private images • Configuration of image triggers

Concepts

Last updated: 2023-09-15 14:22:08

Fundamental Concepts

Clusters

A cluster refers to the collection of cloud resources required for running containers, including several Tencent Cloud resources such as CVM instances and load balancers. You can run your applications within the cluster.

Managed cluster

Tencent Cloud Container Service provides a fully managed Kubernetes cluster management service, where the Master and Etcd of the Kubernetes cluster are centrally managed and maintained by the Tencent Cloud technical team. You only need to purchase the worker nodes required for running the cluster workloads, without worrying about the management aspect of the cluster. Managed clusters can be further categorized into: Standard clusters, Serverless clusters, Edge clusters, and Registered clusters.

- **Standard Cluster**

  It is the default cluster type and is fully compatible with the standard features of open-source Kubernetes clusters. It enhances node management, cluster network, and container scheduling capabilities. In a single cluster, super nodes, native nodes, general CVM nodes, and IDC nodes can be added and managed at the same time, which means they can be combined for different business scenarios to maximize the computing resource utilization.
- **Serverless Cluster**

  You can deploy workloads in a serverless Kubernetes cluster without purchasing nodes. Similar to the container service managed cluster, there are no fees for the managed Master and Etcd control plane resources.
- **Edge Cluster**

  It supports managing nodes located in multiple data centers in the same cluster, and delivering applications to all edge nodes with one click. Also, it has edge autonomy and distributed health check capabilities.
- **Registered Cluster**

  Users can register their local infrastructure's Kubernetes clusters or other cloud vendors' Kubernetes clusters with the container service for unified management. The cloud-native distributed cloud center product can be used to achieve unified management across multiple clouds and clusters.

Self-deployed cluster

TKE provides an independent master node deployment mode, in which you have full control over your cluster. In this mode, the master nodes and etcd nodes of the Kubernetes cluster are deployed on the CVM you purchased, and you have all management and operation permissions for the Kubernetes cluster.

Node

Nodes are the fundamental elements that make up a container cluster. Depending on the business requirements, nodes can be virtual machines or physical machines. Each node contains the essential components required to run Pods, including Kubelet, Kube-proxy, and others.

TKE

Docker containers are designed to allow users to run independent programs in relatively isolated environments, with multiple containers running on a single node.

Video flipping

Docker images are used to deploy container services, and each image has a unique identifier (consisting of the image's Registry address, image name, and image tag). Currently, supported images include Docker Hub's official images and user private images.

Advanced Concepts

Node pool

A node pool provides features such as unified model, unified tag, Taint and dynamic scaling. You can quickly create, manage and terminate nodes as well as realize dynamic scaling of nodes with these features.

Apply

Kubernetes applications that can run within a cluster are supported by the container service, including the creation of marketplace applications, third-party applications, and private applications.

Image Registry

Image repositories are used to store Docker images, which can be deployed for container services.

Marketplace

Currently, it supports Kubernetes community applications suitable for various scenarios and allows the use of application packages from the application marketplace to create and run applications in specific clusters.

Security Group

A security group is a virtual firewall with stateful data packet filtering capabilities. It is used to configure network access control for single or multiple CVM instances and serves as an important means of network security isolation provided by Tencent Cloud. For more information on security groups, please refer to Security Groups.

Add-on

Tencent Cloud offers components such as Event Persistence, Log Collection, GPU Management, Object Storage, and File Storage. Users can extend the cluster's functionality by installing these add-on components.

Namespace

A Kubernetes cluster supports the configuration of multiple namespaces, with each namespace acting as a relatively independent virtual space where resources are isolated from each other. Clusters can manage resources through partitioning using namespaces.

Workload

Local Disk Types	Description
Deployment	A Deployment defines a Pod template and controls the runtime policies for stateless applications. You can declare the number of replicas, scheduling policies, update policies, and more for the Pods running in a Deployment, based on your business requirements.
StatefulSet	Primarily used for managing stateful applications, the created Pods possess persistent identifiers according to the specifications. These identifiers are retained even after Pod migration or termination and restart. When persistent storage is required, you can map storage volumes to identifiers one-to-one. If your application does not require a persistent identifier, it is recommended to use Deployment for deploying the application.
Static Pod IP Type	Tencent Cloud Container Service offers the capability to extend StatefulSet with fixed IPs. Pods created by this type of StatefulSet will be assigned real private network IP addresses through Elastic Network Interfaces (ENIs). The Tencent Cloud Container Service VPC-CNI plugin is responsible for IP

StatefulSet	allocation, ensuring that the IP address remains unchanged when a Pod is restarted or migrated.
DaemonSet	Primarily used for deploying background processes that reside within the cluster, such as node log collection, DaemonSet ensures that the specified Pod runs on all or a subset of nodes. When new nodes are added to the cluster, Pods are automatically deployed. Pods will be automatically reclaimed when nodes are removed from the cluster.
Job	The Job controller creates one or more Pods, which run according to the specified rules until they complete their execution.
CronJob	Run a Job periodically according to the specified schedule.

Sevice

Users can deploy various containers in Kubernetes, some of which provide Layer 7 network services through HTTP and HTTPS protocols, while others provide Layer 4 network services through TCP and UDP protocols. The Service resource defined by Kubernetes is used to manage Layer 4 network service access within the cluster. Based on Layer 4 networking, Service exposes the capabilities of container services within the cluster.

Ingress

Ingress provides exposure for HTTP and HTTPS protocol services under the seventh layer of the network, as well as various common capabilities within this layer. An Ingress is a collection of rules that allow access to services within a cluster. You can configure different forwarding rules to allow different URLs to access different services within the cluster.

Configuration items

- **ConfigMap**: A key-value pair of the key-value type. ConfigMap allows you to decouple configurations from running images, enhancing the portability of your applications.
- **Secret**: A key-value pair used for storing sensitive information such as passwords, tokens, and keys, reducing the risk of direct exposure to external parties.

Storage volume

At its core, a storage volume (Volume) is simply a directory that may contain some data, which can be accessed by containers within a Pod. The lifecycle of a storage volume is the same as that of a Pod and typically lasts longer than the containers running within the Pod, preserving data during container restarts.

PersistentVolume（PV）

Storage resources within the cluster, such as nodes, are considered cluster resources. PVs are independent of the Pod's lifecycle and are created with different types based on the StorageClass type.

PersistentVolumeClaim（PVC）

Storage requests within the cluster, such as PersistentVolumes (PVs) used by Pods for node resources, and PersistentVolumeClaims (PVCs) declaring the use of PV resources. When PV resources are insufficient, PVCs can also dynamically create PVs.

StorageClass

Describing storage types, cluster administrators can define different storage classes for the cluster.

Reference

- Official Docker Documentation
- Official Kubernetes Documentation

Native Kubernetes Terms

Last updated: 2023-09-15 14:22:19

This document primarily compares the terminology used in Tencent Kubernetes Engine (TKE) with that of native Kubernetes, as shown in the table below:

TKE	Native Kubernetes
Clusters	Cluster
Node	Node
Node pool	NodePool
TKE	Container
Video flipping	Image
Instance	Pod
Namespace	Namespace
Stateless Workloads	Deployment
Stateful Workloads	StatefulSet
Task scheduling	Job
Scheduled action	CronJob
Sevice	Service
Route	Ingress
Tag	Label
Configuration items	Configmap
Key	Secret
Volume	Volume
PV	PersistentVolume
PVC	PersistentVolumeClaim
HPA	HPA

Virtual Cluster IP	Cluster IP
NodePort	NodePort
Cloud Load Balancer	LoadBalancer
Node affinity	NodeAffinity
Application Affinity	PodAffinity
Anti-affinity	PodAntiAffinity
Selector	LabelSelector
Annotation	Annotation
Trigger	Webhook
Endpoint	Endpoint
Resource Quotas	Resource Quota
Resource limits	Limit Range
Template	Template