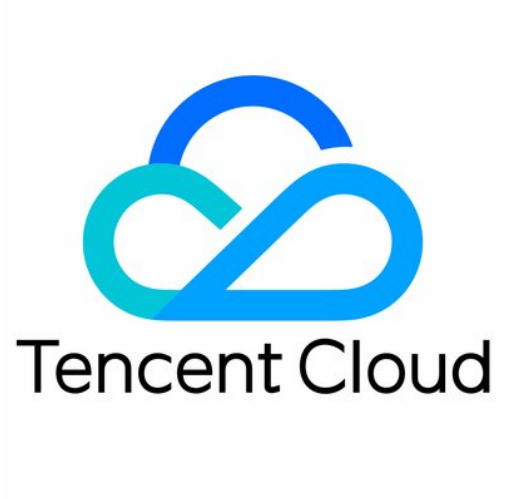
Tencent Kubernetes Engine

Product Introduction

Product Documentation



Contents

Product Introduction
Overview

Last updated：2023-07-19 15:25:23

TKE Overview

TKE 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.

Service Description

Tencent Cloud provides multiple container services for you to deploy, manage, and expand containers:
TKE general cluster: It offers a container-centered, highly scalable, and high-performance container management service based on native Kubernetes.
TKE serverless cluster: It is a service mode launched by TKE that allows you to deploy workloads without having to purchase nodes.
TKE edge cluster: 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.
Tencent Container Registry (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: It is launched by TKE to visualize all resource objects. It has rich filtering query, type aggregate, and status display capabilities, helping you quickly locate the target object.
Tencent Managed Service for Prometheus (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: It is a modular, loosely-coupled, and highly scalable service launched by TKE based on its experience in the cloud-native field.

Directions

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

Console

Cluster Type	Description	Use Cases	Reference
Serverless cluster	It allows you to add and use super nodes, and no cluster management fees will be incurred. You can quickly configure high-spec 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.	Connecting to a 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 different business scenarios to maximize the computing resource utilization.	It is suitable for all scenarios and fully compatible with the standard capabilities of open-source Kubernetes clusters. In addition, super nodes, native nodes, registered nodes, and CVM nodes are supported. Resource visualization and optimized analysis are supported, easily improving resource utilization. General Kubernetes clusters support super nodes and native nodes.	Creating a Cluster
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 lifecycles in the cloud-native method. In addition, the innovative multi-region edge autonomy, closed loop of traffic, and	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	Creating a Cluster

	distributed health checks are available.	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.	
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 multi-cloud management ecosystem opens up high-availability disaster recovery.	-

TencentCloud APIs

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

Strengths

Last updated：2023-05-06 17:36:46

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

Advantage	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 need to determine how to manually deploy container services according to the

	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.	business traffic and health status, which has poor availability and scalability.
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, self-built container services provide poor isolation for tenants, devices, and kernel modules.
High efficiency	**Fast image deployment and continuous business integration** TKE runs inside your VPCs where quality BGP networks ensure fast upload and 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.	Due to unstable network quality, self-built container services cannot guarantee the efficiency of using images for container creation.
Low cost	**High cost-effectiveness**	You need to invest a lot of money to build, install, operate, and

	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.	scale out your cluster management infrastructure.
Cloud native	**Cloud-native scenario optimization** TKE launches native nodes, a new type of nodes designed for Kubernetes environments. Leveraging Tencent Cloud's expertise in managing millions of container cores, native nodes provide users with 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** A new cloud-native asset management platform is launched. This platform offers users insights into cost distribution and resource usage from multiple perspectives, such as cost analysis, job scheduling, and refined scheduling. With this platform, users can maximize the value of every cost incurred in the cloud.	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** The super node type is a new and upgraded Tencent Cloud node type that offers availability zone-level node capabilities with custom specifications. Similar to a large-scale CVM instance, a super node can simplify resource management and scaling.	Self-building is a complex and resource-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.

Advantage	TKE	Customer Self-Built Container Service
Complete metrics	Approximately 30 metrics are available, such as cluster, node, service, container, and Pod (instance) metrics.	Only a few metrics are available and in-house development is required.
Low	TKE monitoring is provided when a cluster is created.	Manual construction of

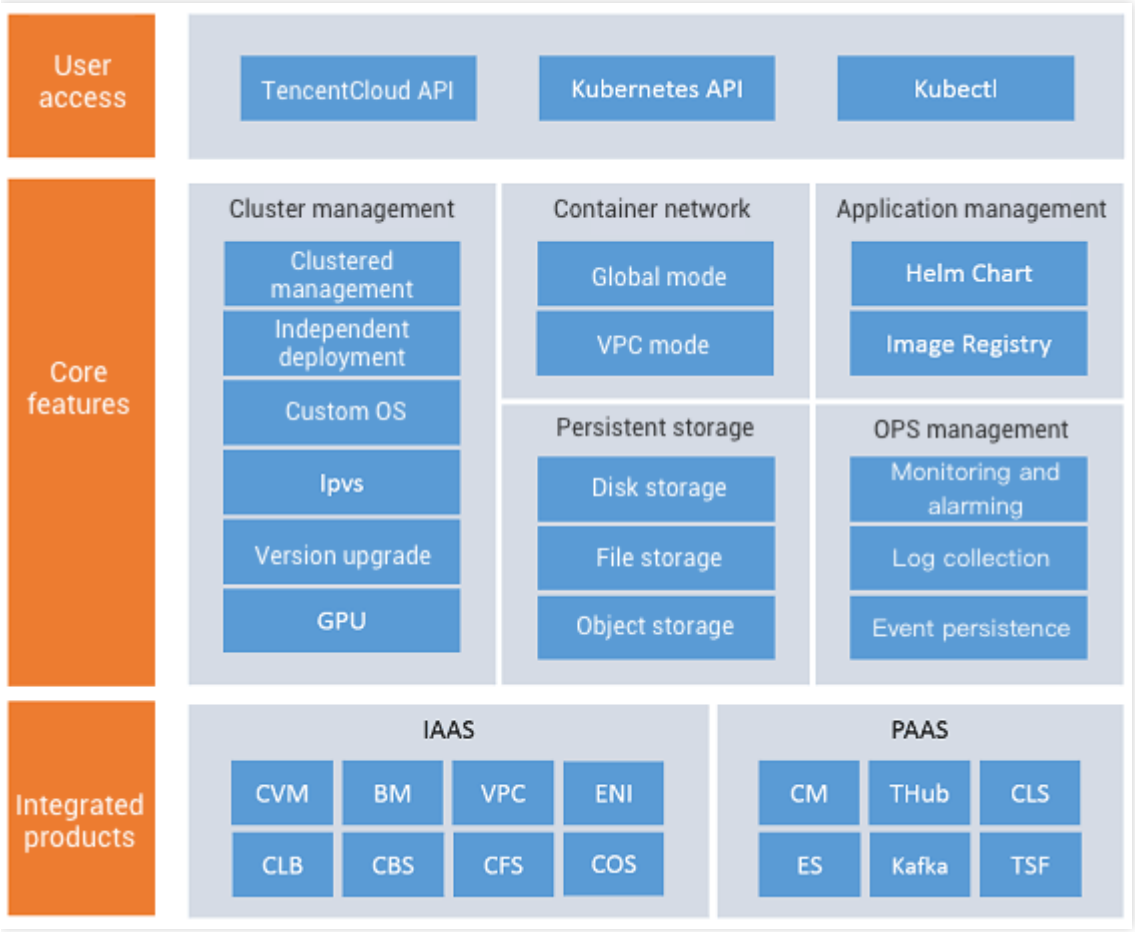
construction cost		monitoring is required and can be expensive.
Low Ops cost	Ops is performed by the platform with guaranteed data accuracy.	Manual Ops is required.
Low storage cost	The data of each metric in the past three months is retained free of charge.	Fees are charged based on the storage size.
High scalability	TKE continues to improve and add new metrics.	Developers are required to develop new metrics.
Alarming	Available	Unavailable
Troubleshooting	Container logs can be viewed in the console and web shells 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：2020-02-24 16:23:24

Overall Architecture

This section describes the design and implementation of the TKE system. Its product architecture is as shown below:



Architecture

1. TKE is adapted and expanded based on Kubernetes, so it supports native Kubernetes capabilities.
2. Tencent Cloud's Kubernetes plugins are available to help you quickly build Kubernetes clusters in Tencent Cloud.
3. TKE provides cluster management, application management, CI/CD, and other advanced capabilities on the upper layer of Kubernetes.

Module Description

1. **TKE Console and TencentCloud API**: You can manipulate clusters and services using the console, kubectl, or APIs.

2. **Image service modules:** You can upload and download images through the image service modules provided by Tencent Cloud.
3. **TKE modules**: These are the core modules of TKE, and include clusters and services CRUD operations.

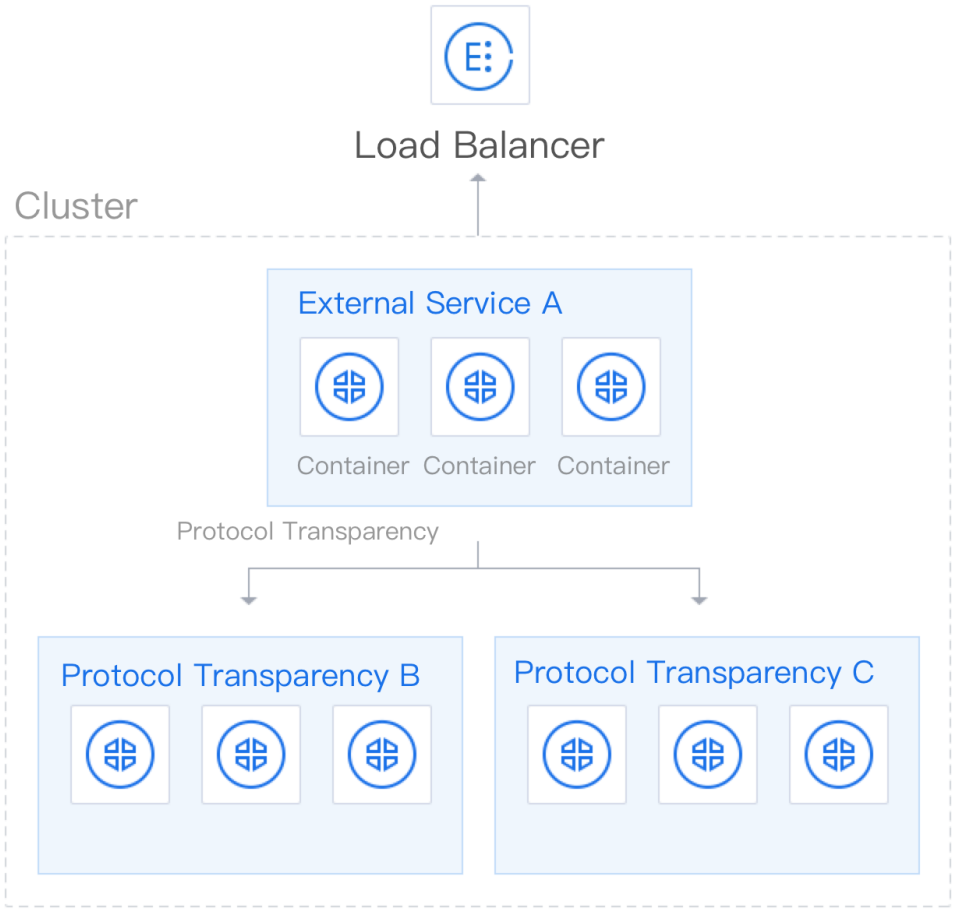
Scenarios

Last updated：2022-06-14 15:23:40

Microservice Architecture

The microservice architecture is suitable for creating complex applications. It splits your monolithic application into multiple micro-services across different dimensions, and the content of each micro-service can be managed by a Docker image.

The following figure shows the microservice architecture:



Advantages of Deploying Microservice by Using Tencent Cloud TKE

- Cluster management is simplified and cluster installation is not required.
- Seamlessly connects to Tencent Cloud's computing, network, storage, monitoring, security capabilities, and directly uses Tencent Cloud's IaaS capability.

- It is easy to use, and supports service arrangement and application management at service granularity. Resources are highly isolated while services are highly available.

Continuous Integration and Delivery

An excellent DevOps environment is provided through continuous integration and continuous delivery to greatly increase the efficiency of software release.

Continuous Integration

Allows developers to complete building and (unit) testing processes immediately after submitting new codes. According to test results, you can determine whether new codes and original codes can be properly integrated.

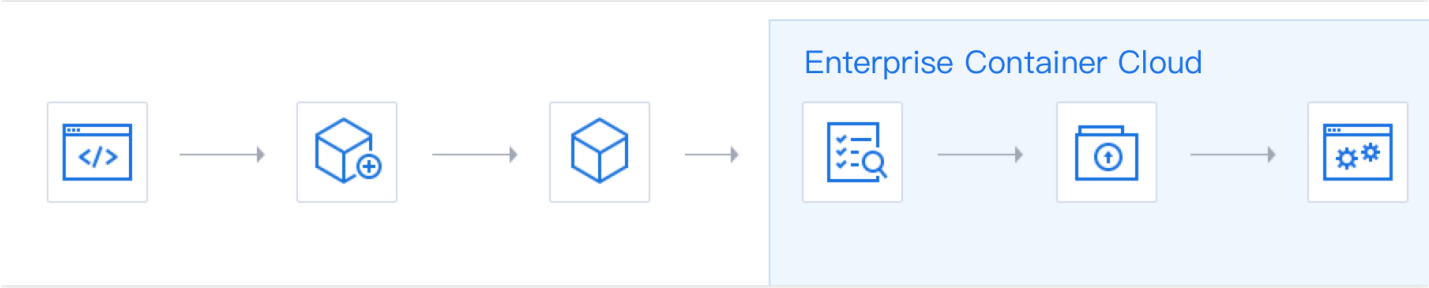
Continuous Delivery

Based on continuous integration, the integrated codes will be deployed in the operating environment.

Advantages

By deploying services on Tencent Cloud TKE, developers can perform such operations as building, testing, packaging, and integration immediately after submitting new codes. Then, they deploy the integrated codes into the pre-release environment and live environment through continuous integration.

The following figure shows the process of continuous integration and continuous delivery.



Migrating Business Quickly to Tencent Cloud

To migrate individual or enterprise business to Tencent Cloud, you can use Tencent Cloud TKE to simplify cloud configuration and cluster management to improve the efficiency of service delivery.
Tencent Cloud TKE allows you to quickly create services, realize containerized deployment of applications, and also achieve auto-scaling, on-demand deployment, high availability, easy capacity expansion, friendly development, and

labor cost reduction. See the figure below.

Features

Last updated：2023-07-19 15:35:38

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	FinOps-based native nodes and super nodes for cost reduction and performance enhancement Support for most CVM models and host creation/adding Support for node registration and hosting of non-Tencent Cloud CVMs 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 Kernel optimized images dedicated 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：2022-11-08 15:31:38

Basic Concepts

Cluster

A cluster is a collection of cloud resources required to run a container, including several CVMs and CLBs. You can run your applications in a cluster.

Managed cluster

TKE provides the Kubernetes cluster management service for managing Master and Etcd nodes. In this mode, the Tencent Cloud technical team centrally manages and maintains the Master and Etcd nodes of a Kubernetes cluster. You only need to purchase worker nodes for the cluster in order to run workloads and do not need to care about cluster management.

Independently deployed cluster

TKE provides an independent Master deployment mode in which you have full control over your cluster. In this mode, the Master and Etcd nodes of the Kubernetes cluster are deployed in your CVM instances, and you have full management and operation permissions for the Kubernetes cluster.

Serverless cluster

An elastic cluster is a serverless Kubernetes cluster that allows you to deploy workloads without purchasing nodes. Managed control-plane resources, such as the Master and Etcd nodes, are not billed in an elastic cluster, just as in a TKE managed cluster.

Edge cluster

In an edge cluster, you can manage nodes deployed in multiple data centers (DCs), deliver an application to all edge nodes with one-click, and enjoy edge autonomy and distributed health check capabilities.

Node

A node is a basic element of a container cluster. It can be either a virtual machine or a physical machine, depending on the service. Each node contains the basic components required to run a pod, including Kubelet and Kube-proxy.

Container

Docker containers allow you to run applications independently in an independent environment. Multiple containers can run in a node.

Image

Docker images are used to deploy TKE. Each image has a unique ID (image repository address + image name + image tag). Currently, Docker Hub official images and users' private images are supported.

Advanced Concepts

Node pool

Nodes in a node pool have the same model, label, and taint attributes and can be dynamically scaled in or out. With these features, you can conveniently and quickly create, manage, and terminate nodes and dynamically scale nodes in and out.

Application

Kubernetes applications can run in a cluster. TKE allows you to create applications from the application market, third-party applications, and private applications.

Image repository

An image repository is used to store Docker images, which are used to deploy TKE.

Application market

The TKE application market provides Kubernetes community applications suitable for various scenarios. You can use application packages from the application market to create and run applications in clusters.

Security group

A security group is a virtual firewall that can filter stateful data packets. It is used to configure network access control for one or more CVMs. It is an important network security isolation method provided by Tencent Cloud. For more information, see Security Groups.

Add-on

Add-ons include event persistence components, log collection components, GPU management components, and COS and file storage components. You can install these components to extend related features for clusters.

Namespace

You can set multiple namespaces in a Kubernetes cluster. Each namespace is an independent virtual space. Resources in different namespaces are isolated. A cluster can use namespaces to manage resources by partition.

Workloads

Type	Description

Deployment	A Deployment workload declares a pod template and a policy for controlling how the pod runs. It is used to deploy stateless applications. You can declare the number of replicas, scheduling policy, and update policy for a pod that runs in the Deployment workload as required.
StatefulSet	A StatefulSet workload is used to manage stateful applications. It creates a persistent identifier for each pod based on the specifications. The identifier will be retained after the pod is migrated, terminated, or restarted. When using persistent storage, you can map storage volumes to identifiers. If your application does not require a persistent identifier, we recommend that you use a Deployment workload to deploy the application.
StatefulSet with fixed pod IP addresses	TKE provides StatefulSet workloads with fixed pod IP addresses. Pods created by StatefulSet workloads of this type will be assigned private network IP addresses through the ENI. The TKE VPC-CNI plugin is responsible for IP address assignment. When a pod is restarted or migrated, its IP address can remain unchanged.
DaemonSet	A DaemonSet workload is used to deploy resident backend programs in a cluster., such as node log collection. DaemonSet ensures that specified pods are running on all or certain nodes. When you add new nodes to a cluster, pods are deployed automatically. When nodes are removed from the cluster, pods are retrieved automatically.
Job	A Job creates one or multiple pods and ensure that these pods run according to the rules until they are terminated.
CronJob	A CronJob workload periodically runs a Job workload based on a preset plan.

Service

You can deploy various containers in a Kubernetes cluster. Some containers use HTTP or HTTPS to provide external Layer-7 network services, and the others use TCP or UDP to provide Layer-4 network services. Service resources defined by Kubernetes are used to manage Layer-4 network service access in a cluster. Based on the Layer-4 network, a service exposes TKE in a cluster.

Ingress

An ingress exposes HTTP and HTTPS services in a Layer-7 network and provides common Layer-7 network capabilities. An ingress is a collection of rules that allow access to services of a cluster. You can configure different forwarding rules to allow different URLs to access different services.

ConfigMap

- **ConfigMap**: a key-value pair. ConfigMap allows you to decouple the configuration from the runtime image to make the application more portable.
- **Secret**: a key-value pair. A secret stores sensitive information, such as passwords, tokens, and keys to reduce the risk of information leakage.

Volume

A volume is a directory that may contain some data. Containers in a pod can access the directory. The volume lifecycle is the same as that of a pod and longer than that of containers running in the pod. Typically, it saves data when a container is restarted.

PersistentVolume (PV)

PVs are used to store resources in a cluster, such as nodes. A PV is independent of the lifecycle of the pod, and different types of PVs can be created based on different StorageClass types.

PersistentVolumeClaim (PVC)

PersistentVolumeClaim (PVC) is a storage request in a cluster. Pods consume node resources, and the PVC consumes PV resources. If PV resources are insufficient, the PVC can dynamically create PVs.

StorageClass

StorageClass describes the storage type. A cluster admin can define different storage types for a cluster.

References

- Official Docker documentation
- Official Kubernetes documentation

Native Kubernetes Terms

Last updated：2020-08-27 10:57:00

This document describes the mapping between terms in TKE and native Kubernetes.

TKE	Native Kubernetes
Cluster	Cluster
Node	Node
Node pool	Node pool
Container	Container
Image	Image
Instance	Pod
Namespace	Namespace
Stateless Workload	Deployment
Stateful Workload	StatefulSet
Task	Job
Scheduled Task	CronJob
Service	Service
Route	Ingress
Label	Label
Configuration Item	Configmap
Secret Key	Secret
Volume	Volume
PV	PersistentVolume
PVC	PersistentVolumeClaim
Auto Scaling	HPA

TKE	Native Kubernetes
Cluster IP	Cluster IP
Node Port	NodePort
Load Balancer	LoadBalancer
Node Affinity	NodeAffinity
Application Affinity	PodAffinity
Application Anti-affinity	PodAntiAffinity
Selector	LabelSelector
Annotation	Annotation
Trigger	Webhook
Endpoint	Endpoint
Resource Quota	Resource Quota
Resource Limit	Limit Range
Template	Template