Tencent Kubernetes Engine

Product Introduction

Product Documentation

Contents

Product Introduction
Overview
Tencent Kubernetes Engine

Last updated : 2020-11-13 15:13:59

Overview

Tencent Kubernetes Engine (TKE) is a container management service with high scalability and performance that enables you to easily run applications in a managed CVM instance cluster. This service frees you from installation, OPS, and expansion of the cluster management infrastructure. In addition, it allows you to launch and terminate Docker applications, query the status of the cluster, and use various Tencent Cloud services through simple API calls. You can arrange containers in your cluster based on resource and availability requirements to meet your business or application-specific needs.

Based on native Kubernetes, TKE provides a container-oriented solution that solves operating environment issues during development, testing, and OPS and helps reduce costs and improve efficiency. It is fully compatible with the native Kubernetes APIs and extends Kubernetes plug-ins such as CBS and CLB on the Tencent Cloud. In addition, TKE provides network solutions with high reliability and performance based on Tencent Cloud VPC.

Glossary

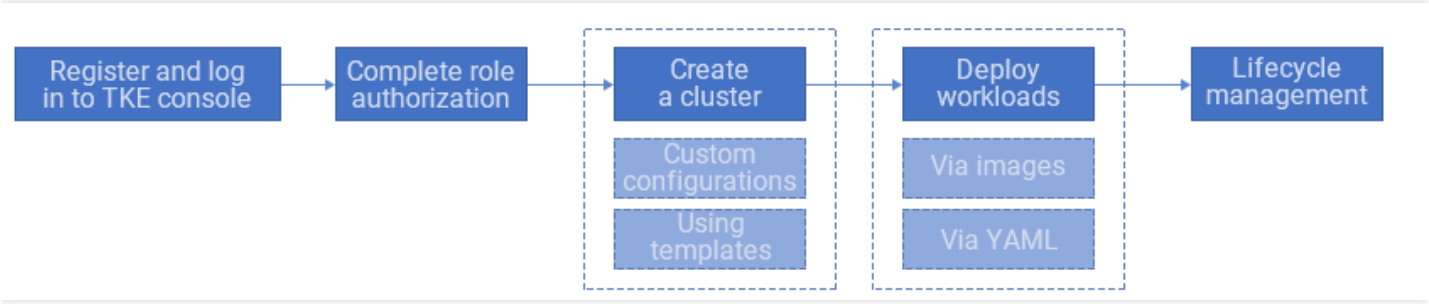
The following describes the key terms related to TKE:

- **Cluster**: the collection of cloud resources required to run containers, including several Tencent Cloud resources such as CVM instances and CLBs.
- **Pod**: a group of one or more associated containers that share the same storage and network space.
- **Workload**: a Kubernetes resource object that is used to manage the creation, scheduling, and the automatic control of Pod replicas throughout the entire lifecycle.
- **Service**: a group of microservices consisting of multiple Pods with the same configuration and the rules for accessing these Pods.
- **Ingress**: a collection of rules for routing external HTTP(S) traffic to a service.

- **Application**: features related to Helm 3.0 integrated in TKE, which provide you with various product and service capabilities including Helm Chart, Tencent Container Registry (TCR), and software services.
- **Image repository**: stores Docker images that are used to deploy TKE.

Procedure

The following figure shows the TKE usage flowchart:



1. Authorize roles.
   Sign up and log in to the TKE console and then grant permissions for your role to perform operations on relevant resources so that you can get started with TKE.
2. Create a cluster.
   You can customize a cluster or create a cluster from a template.
3. Deploy workloads.
   You can deploy workloads by deploying images or orchestrating the YAML file. For more information, see Workloads.
4. Manage the lifecycle of Pods through operations such as monitoring, upgrade, and scaling.

Pricing

TKE is now free of charge, but you need to pay for the usage of relevant Tencent Cloud resources. For more information on billing and pricing, see Pricing Description.

Relevant Services

- You can purchase several CVM instances to form a TKE cluster. The containers will run on the CVMs. For more information, see the CVM Documentation.

- A cluster can be created in a VPC. CVM instances in the cluster can be allocated to subnets in different availability zones. For more information, see the VPC Documentation.
- You can use CLB to automatically allocate the request traffic of clients across CVM instances and then forward it to the containers running on the CVM instances. For more information, see the CLB Documentation.
- You can use Cloud Monitor to monitor the operation statistics of TKE clusters and Pods. For more information, see the Cloud Monitor Documentation.

Elastic Kubernetes Service

Last updated : 2020-09-11 10:24:28

Introduction

Elastic Kubernetes Service (EKS) is a TKE service mode that allows you to deploy workloads without purchasing any nodes. EKS is fully compatible with native Kubernetes, allowing you to purchase and manage resources natively. This service is billed based on the actual amount of resources used by containers. In addition, EKS provides extended support for Tencent Cloud products, such as storage and network products, and can ensure the secure isolation of containers. EKS is ready to use out-of-the-box.

Related Concepts

Containers and images

Containerization is a form of lightweight operating system virtualization that isolates and controls system resources so that formerly global resources are only available for processes in a container. An image is similar to a lightweight snapshot of virtual machines (VMs) and can be interpreted as a static container. An image defines all files that run in a container and the dependencies between them to ensure consistent container runtime environments.
By containerization, all applications and their dependencies are packaged into an image, and then uses the image to generate a resource-isolated environment to run the applications. This allows the applications to run independently in a consistent environment in a simple and efficient manner.

Kubernetes

Kubernetes is an open-source Container Orchestration Engine (COE) inspired by a Google project called Borg. It is one of the most important components of the Cloud Native Computing Foundation (CNCF). Kubernetes provides production-level features such as application orchestration, container scheduling, service discovery, and autoscaling. For more information, see Kubernetes Documentation.

Strengths

Native support

EKS is community-driven and supports the latest version of Kubernetes and native Kubernetes cluster management. It serves as a plug-in to provide extended support for Tencent Cloud products, such as storage, network, load balancing products. EKS is ready to use out-of-the-box.

Serverless

EKS is a fully-managed Kubernetes service, which means that you do not need to manage any computing nodes. EKS delivers computing resources by using pods. It allows you to purchase, return, and manage cloud resources as in Kubernetes.

High security and reliability

EKS can achieve 99.95% or higher availability based on the mature virtualization technology and network architecture of Tencent Cloud. Tencent Cloud ensures virtual isolation and network isolation between the EKS clusters of different users and allows users to configure network policies for a specific service by using services such as security groups and network Access Control List (ACL).

Scaling within seconds

With the lightweight virtual technology developed by Tencent Cloud, you can create or delete a TKE instance within seconds to ensure higher efficiency when creating resources. EKS allows you to configure the native Horizontal Pod Autoscaler (HPA) of Kubernetes so that services can be automatically scaled based on actual loads.

Reduced costs

The serverless architecture allows EKS to provide higher resource utilization and lower OPS costs. The flexible and efficient auto scaling capability ensures that EKS only consumes the amount of resources required by the current load.

Service integration

EKS can be highly integrated with most Tencent Cloud services, including the storage products Cloud Block Storage (CBS), Cloud File Storage (CFS), and Cloud Object Storage (COS), TencentDB product family, and virtual private cloud (VPC) product family. With this capability, EKS can provide solutions that meet the requirements of a wide range of businesses.

Use limits

- For more information, see Purchase Limits.

Pricing

EKS is a fully-managed and serverless Kubernetes service. Managed control-plane resources such as the Master and Etcd are not billed in an elastic cluster. This billing method is similar to that of a TKE-managed cluster. Workloads running in an elastic cluster are billed in pay-as-you-go mode. The fees are calculated based on the configured amount of resources and the actual use period. For more information, see Pricing.

Comparison with TKE

Feature	TKE	EKS
Kubernetes	This feature is natively supported.	This feature is natively supported. Some features are not supported due to the lack of computing nodes. For more information, see Unsupported Native Kubernetes Features.
VPC	This feature is supported.	
Computing nodes	You need to purchase and manage computing nodes such as Cloud Virtual Machine (CVM) and Bare Metal (BM) nodes on your own.	You do not need to purchase any computing nodes.
Management method	Native Kubernetes APIs and Kubectl are supported.	
Clusters	Multiple clusters can be created and managed.	
Namespaces	This feature is natively supported.	
Workloads	This feature is natively supported.	Native Kubernetes workloads, except DaemonSet, are supported.
Services	This feature is natively supported. A CLB plug-in is integrated with TKE.	
Storage	This feature is natively supported. Plug-ins such as CBS and CFS can be integrated.	

Use Cases

Microservices

Running microservices with EKS can free users from OPS of computing nodes. A service can be automatically scaled based on the actual load, and the necessary amount of resources are used to run applications, which reduces resource costs.

Offline computing

To run an offline computing task with EKS, you simply need to prepare a container image to quickly deploy workloads for the task. In addition, EKS bills only the actual amount of computing resources used during the execution of the task and stops billing when pods are automatically released after the task ends.

Online inference

EKS can run online inference services by using CPU, GPU, and vGPU resources. The abundant resource specifications and the workloads that support autoscaling improve the operating efficiency and cost-effectiveness of the online inference services.

Relevant Services

- Storage: To use a cloud disk or file storage as the persistent storage of a container, you can use CBS or CFS.
- Network:
  - To create and manage your private network, for example, to create a VPC instance and a subnet, establish a peering connection, use the NAT Gateway, configure a route table, or configure a security policy, refer to the VPC documentation.
  - To manage service access configurations for private and public network traffic, you can use CLB.
- APIs: To gain access to Tencent Cloud products and services by calling Tencent Cloud APIs, refer to the Tencent Cloud API documentation.

Tencent Kubernetes Engine for Edge

Last updated : 2020-12-14 11:17:44

Product Introduction

Edge Cloud Kubernetes Engine (ECK) is a container system for managing edge cloud resources from a centralized cloud. ECK is fully compatible with native Kubernetes. You can manage nodes in multiple data centers with one cluster and deliver applications to all edge servers with one click. ECK also comes with edge autonomy and distributed health check features.

Concepts

Containers and images

Containers are lightweight virtualization tools applied at the system level. With the ability to isolate and control system resources, containers restrict global resources access to processes in selected containers. A container image is a virtual machine snapshot and can be seen as the static form of a container. An image defines all files and dependencies required to run a container, ensuring consistency for running the container.

A container packages an application and its dependencies into an image, and then uses the image to generate a resource-isolated environment to run the application. This allows the application to run independently in a consistent environment in a simple and efficient manner.

Kubernetes

Kubernetes is an open-source Container Orchestration Engine (COE) inspired by a Google project called Borg. It is one of the most important components of the Cloud Native Computing Foundation (CNCF). Kubernetes provides production-level features such as application orchestration, container scheduling, service discovery, and autoscaling. For more information, see Kubernetes Documentation.

Benefits of ECK

Native support

ECK is an out-of-the-box service that supports the latest Kubernetes version and native Kubernetes cluster management methods.

Availability across data centers

ECK is a Kubernetes service with master components hosted in the cloud and worker nodes located anywhere you want. Users do not need to provide resources required by the master components.

Security and reliability

ECK supports private and public network certificates separation and minimal node permissions to avoid cluster access leakage. TLS encryption is used in communication between the cloud and the edge to protect system management data from leakage and tampering.

System disaster recovery

ECK provides a reliable edge autonomy capability for cloud-edge communication scenarios. It also supports distributed cluster health check to help you determine the best time to migrate pods.

Easy system OPS

ECK leveraged Tencent’s years of experience in deep tunneling technologies to enable admins to log in to containers on edge servers directly from the cloud, even if the edge devices do not have public IP addresses.

Management across clouds

ECK supports management of public cloud, private cloud, Tencent Cloud, and any other cloud computing resources.

Pricing

ECK is a Kubernetes service with master components hosted. Like TKE, ECK does not charge fees for management resources such as hosted control plane and Etcd.

Computing nodes are provided by users and therefore do not generate any additional charges.

Use Cases

Edge computing

ECK helps you manage edge computing resources, allowing you to assign and schedule resources, deploy, upgrade and terminate applications, and perform system OPS from the cloud.

Management across clouds

With ECK, you can easily manage computing resources stored in various locations, form different cloud providers to your on-premises data centers, enjoying the convenience of centralized cloud

management.

Additional Services

For more information on how to call TencentCloud APIs to access Tencent Cloud products and services, see TencentCloud APIs.

Strengths

Last updated : 2020-02-24 16:26:17

Orchestration Advantages

Kubernetes-based Service

TKE is developed on the basis of Kubernetes (K8s), 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.

Benefits of Kubernetes

- 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	Tencent Cloud TKE	Customer Self-built Container Service

Advantage	Tencent Cloud 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 cluster status for easy integration with your customized and third-party scheduling applications. • TKE is integrated with 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 is 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.

Advantage	Tencent Cloud TKE	Customer Self-built Container Service
Security and stability	**High isolation of resources and high availability of services** • TKE works inside your own CVM instance without sharing computing resources with other customers. • Your clusters run inside 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 while ensuring high security and availability of services and data together with distributed backend storage of stateful services.	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 download of images and make high numbers of containers able to launch within seconds, greatly reducing operational overheads and enabling you to focus more 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 services and deploy the integrated code in pre-release and production environments.	The efficiency of using images to create containers cannot be guaranteed as the network quality of self-built container services may fluctuate significantly.
Low cost	**Free of charge TKE services** There are no extra fees for using TKE, and APIs can be called in containers for free to create cluster management applications. You only need to pay for the Tencent Cloud service resources created for storage and running applications, such as CVM and CBS.	You need to spend big money creating, installing, operating, maintaining, and scaling your own cluster management infrastructures.

TKE Monitoring vs. 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.

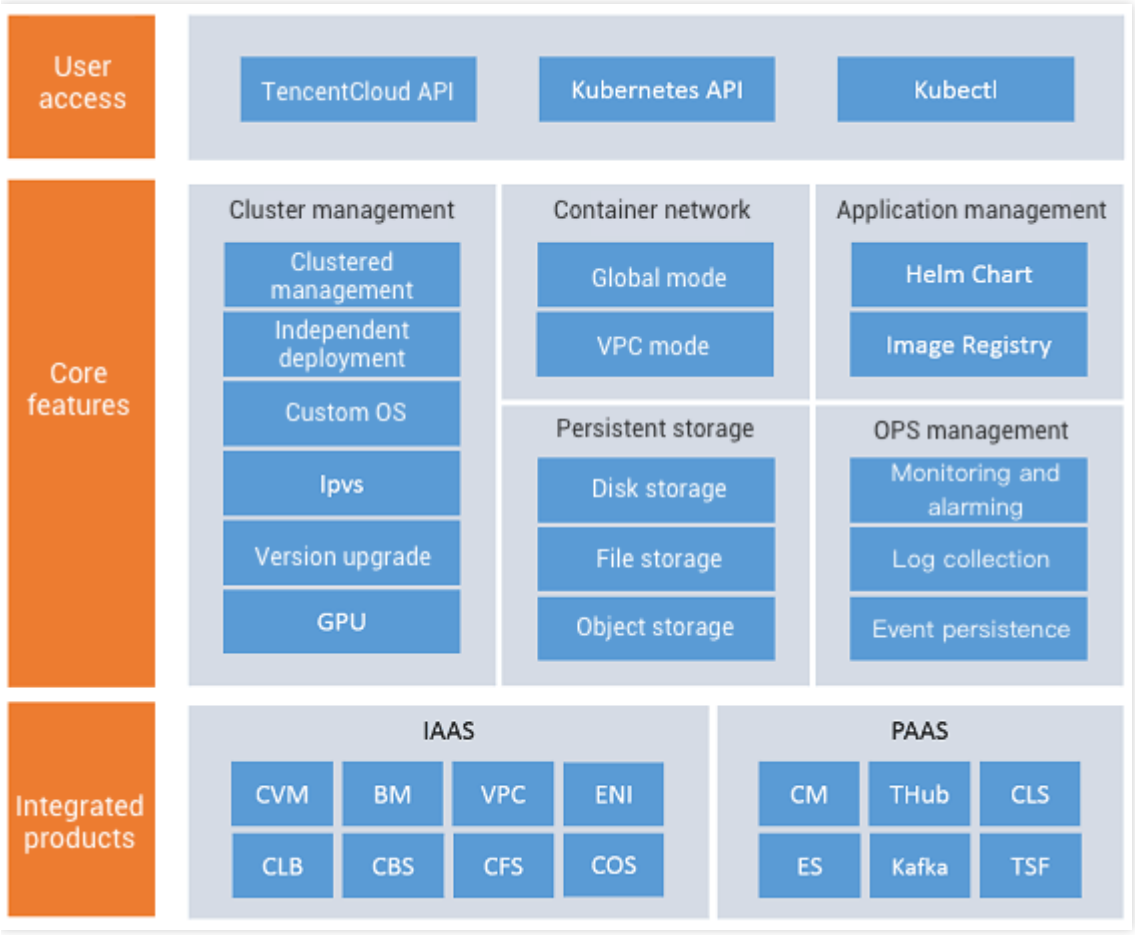
Advantage	TKE	Self-built Container Service
Complete set of metrics	About 30 metrics are available, including cluster, service, container, and pod.	Only a few metrics are available and in-house development is required.
Low construction cost	TKE monitoring is provided when a cluster is created.	Monitoring needs to be constructed manually and costs a lot.
Low OPS cost	Metric OPS is performed by the platform with guaranteed data accuracy.	Metrics need to be maintained manually.
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 has been optimizing and adding new metrics.	Developers are required to construct new metrics.
Alarming	Available	Unavailable
Troubleshooting	Container logs can be viewed in the console and WebShell can be used to quickly log in to containers for troubleshooting.	Containers or servers need to be logged in to manually 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 : 2019-10-25 16:34:49

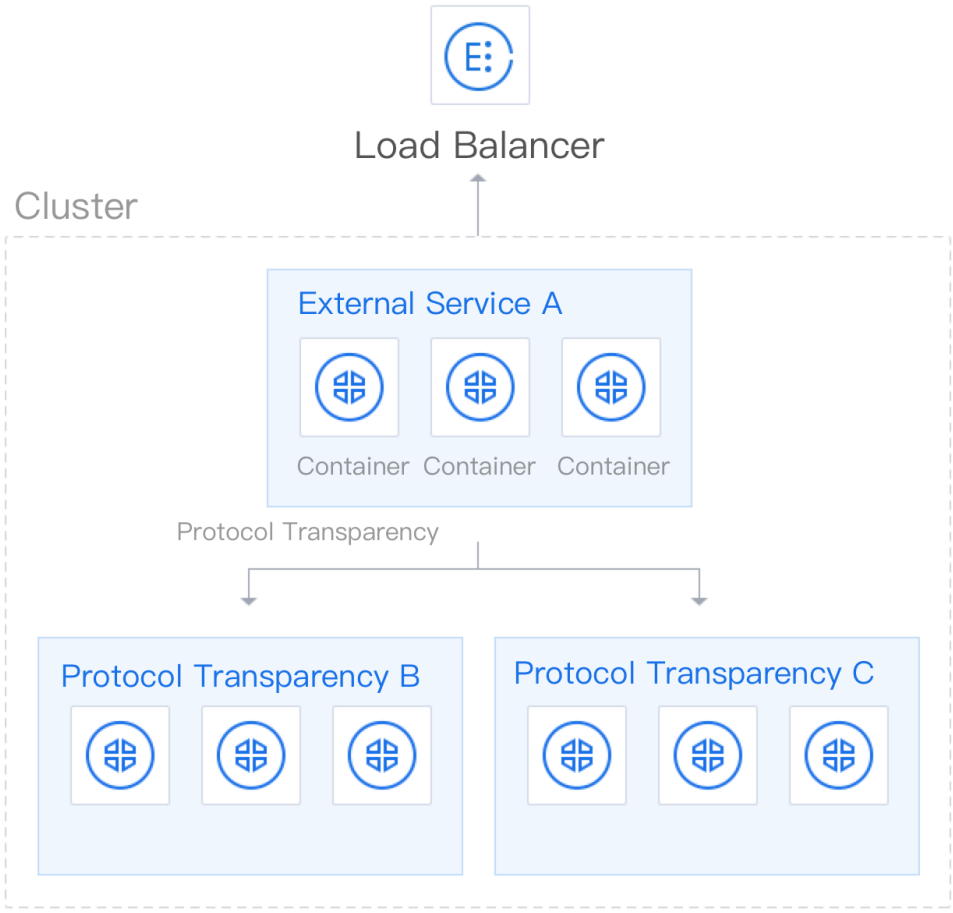
Microservice Architecture

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.

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.

The following figure shows the microservice architecture:



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.

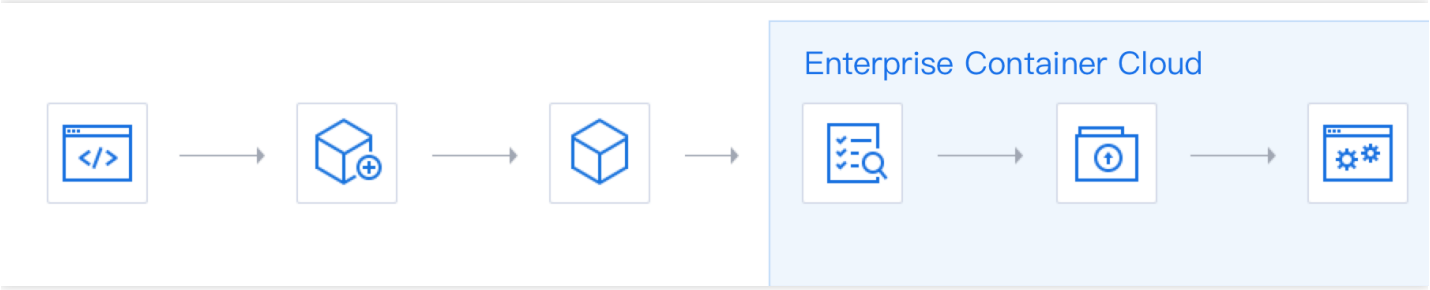
Continuous Delivery

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 : 2020-02-24 16:21:48

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	• Creates and adds exiting CVM instances of all models as nodes to clusters • Supports cross-AZ deployment of nodes in a cluster • Supports pay-as-you-go billing • User-exclusive clusters; isolates clusters through VPCs • Customizes cluster network and configures container network as you need
Cluster Management	• Supports cluster scaling, node scale-in/out • Provides rich monitoring metrics; supports custom alarm policies
Kubernetes Management	• Supports multiple Kubernetes editions and edition upgrage • Manages Kubernetes certificates; manipulates clusters directly using kubectl • Easily manages namespaces in the console

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	• Supports various types of TKE services • Supports Kubernetes Deployment, DaemonSet and other resource types

Module	Feature
Application Management	• Creates applications quickly using "My Templates" or templates in the Template Market • Supports real-time comparison of updated applications • Deploys and stops services in an application with one click
Template Management	• Supports "My Templates" and Template Market • Duplicates templates in one click

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	• Supports service deployment in single-pod multi-container manner • Supports multiple service access methods • Supports cross-AZ deployment of pods in a service • Configures affinity and anti-affinity scheduling
Service Management	• Supports rolling updates and fast updates of services • Supports dynamical scaling of services • Supports remote login for containers in services
Service OPS	• Checks detailed monitoring data of services in different metrics • Checks `stdout` and `stderr` logs of containers in services • Configures alarm policies for services • Configures health check (survival check and readiness check) • Auto-recovers containers in case of exception

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

Module	Feature
ConfigMap Management	• Multiple ConfigMap versions are supported • Two editing forms are supported, i.e., visualization and YAML
ConfigMap Use	• Mounts ConfigMaps to container directory as volumes • Imports ConfigMaps as environment variables • Uses ConfigMaps in place of 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	• Creates private image repositories • Checks and uses DockerHub image repositories • Checks and uses TencentHub image repositories • Manages multiple image namespaces
Image Use	• Creates images through hi-speed private network • Uploads and downloads images over a public network
CI/CD	• Auto-building of private images can be set • Image triggers can be set

Concepts

Last updated : 2020-11-24 17:13:49

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.

Elastic 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