Cloud Container Engine

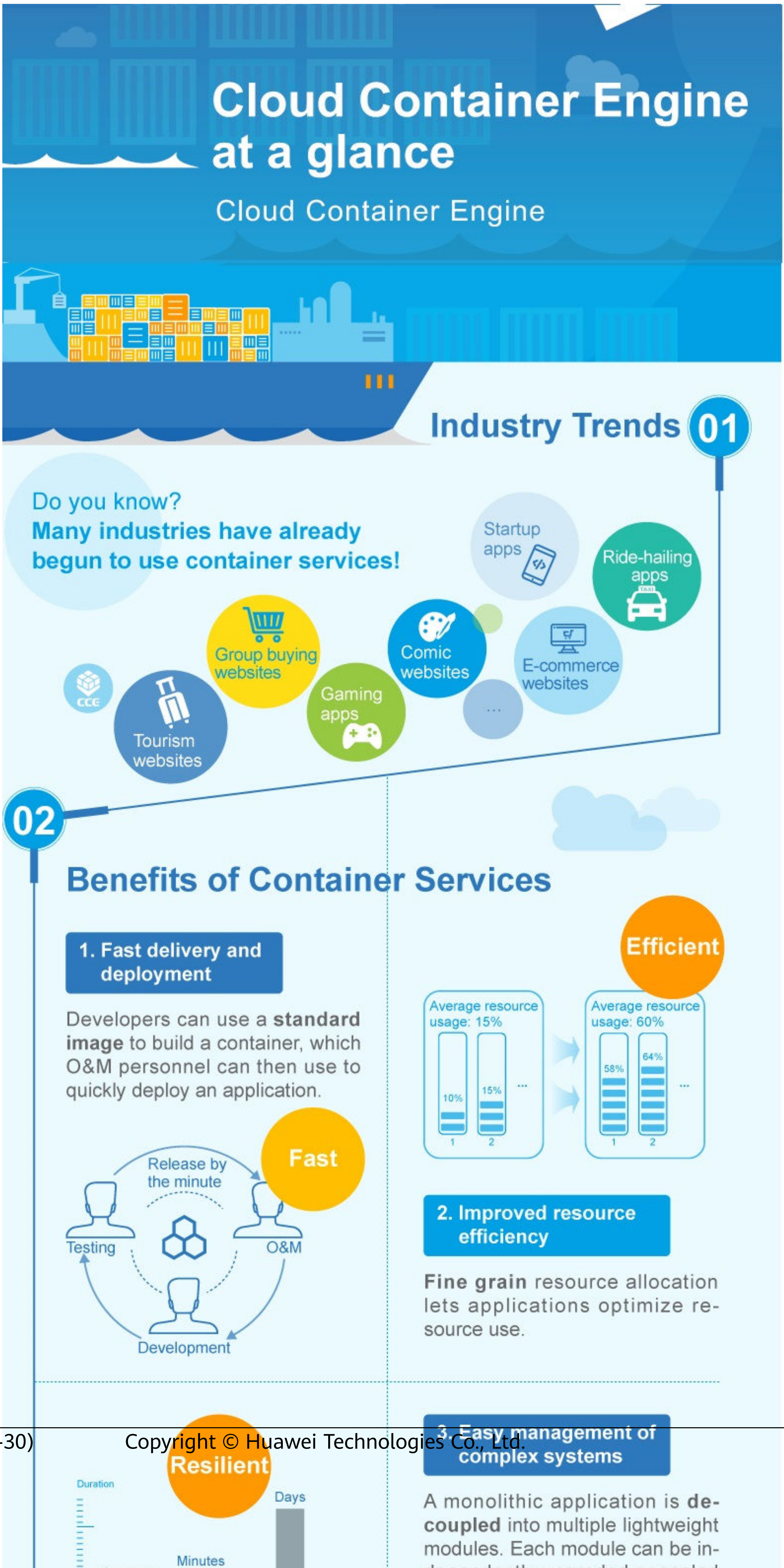
Service Overview

Issue	01
Date	2022-12-30

HUAWEI TECHNOLOGIES CO., LTD.

Contents

1 CCE Infographic



Cloud Container Engine at a glance

Cloud Container Engine

Industry Trends 01

Do you know?
Many industries have already begun to use container services!

02 Benefits of Container Services

1. Fast delivery and deployment

Developers can use a **standard image** to build a container, which O&M personnel can then use to quickly deploy an application.

2. Improved resource efficiency

Fine grain resource allocation lets applications optimize resource use.

3. Easy management of complex systems

A monolithic application is **decoupled** into multiple lightweight modules. Each module can be in-

2 What Is Cloud Container Engine?

Cloud Container Engine (CCE) is a scalable, enterprise-class hosted Kubernetes service. With CCE, you can easily deploy, manage, and scale containerized applications in the cloud.

Why CCE?

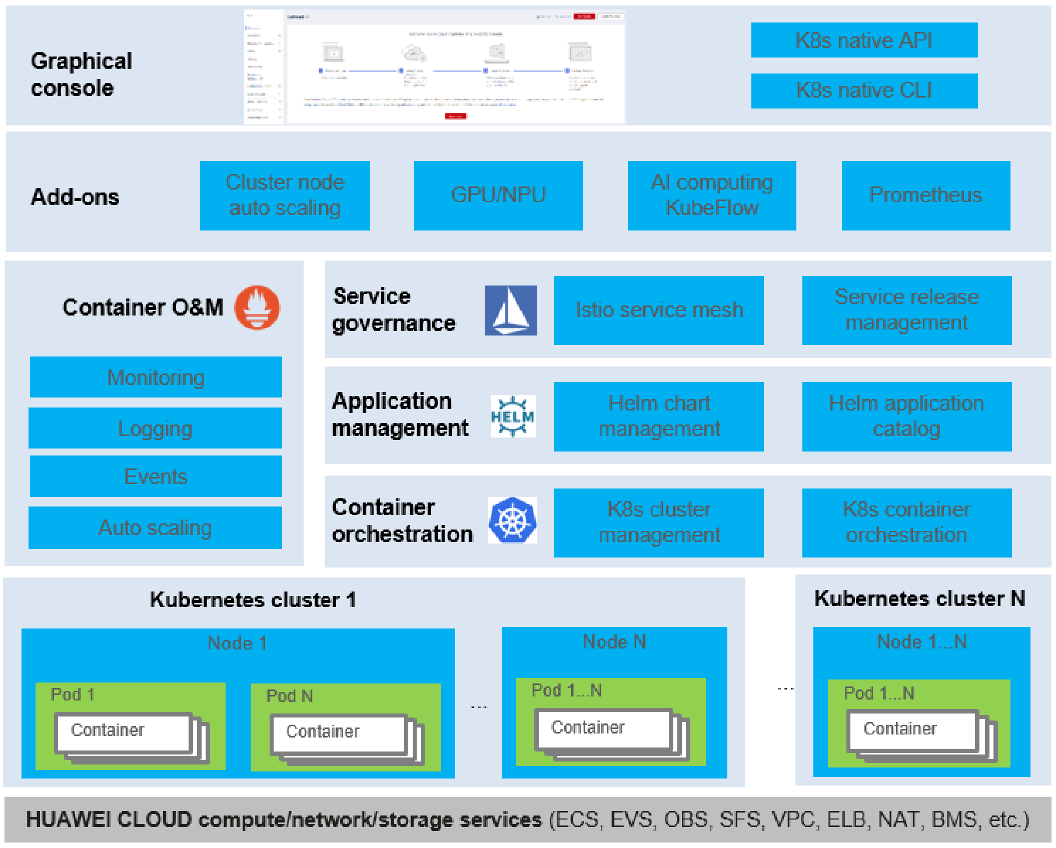
CCE is a one-stop platform integrating compute, networking, storage, and many other services. It supports heterogeneous computing architectures such as GPU, NPU, and Arm. Supporting multi-AZ and multi-region disaster recovery, CCE ensures high availability of **Kubernetes** clusters.

Huawei Cloud is one of world's first Kubernetes Certified Service Providers (KCSPs). It is the first in China to engage in the Kubernetes community. As a constant contributor, Huawei Cloud marks its presence in the container ecosystem. It is also a founder and platinum member of Cloud Native Computing Foundation (CNCF). CCE is one of the first Certified Kubernetes offerings in the world.

For more information, see **Product Advantages** and **Application Scenarios**.

Product Architecture

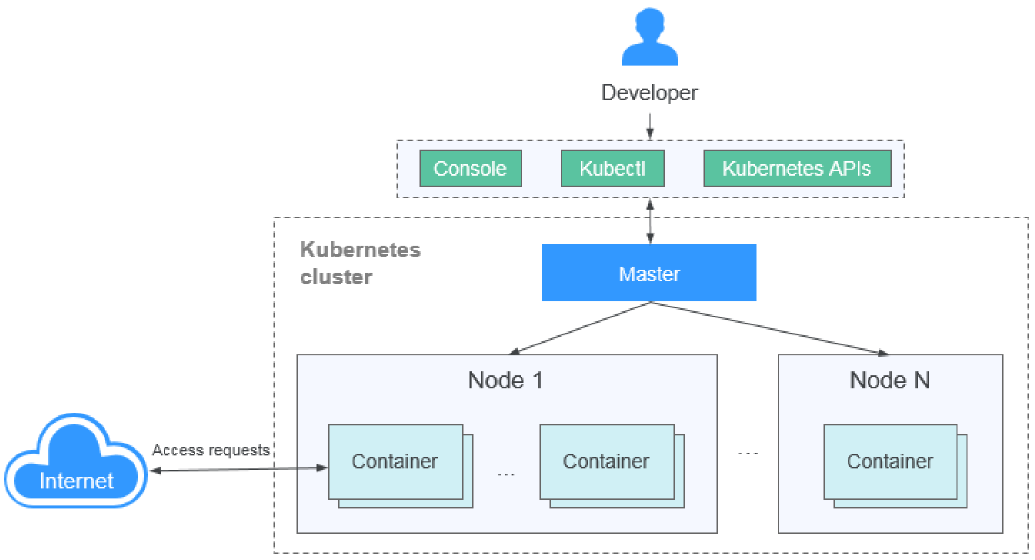
Figure 2-1 CCE architecture



Accessing CCE

You can use CCE via the CCE console, kubectl, or Kubernetes APIs. **Figure 2-2** shows the process.

Figure 2-2 Accessing CCE



CCE Learning Path

You can click **here** to learn about the fundamentals about CCE so that you can use CCE and perform O&M with ease.

3 Product Advantages

Why CCE?

CCE is a container service built on Docker and Kubernetes. A wealth of features enable you to run container clusters at scale. CCE eases containerization thanks to its reliability, performance, and open source engagement.

Easy to Use

- Creating a Kubernetes cluster is as easy as a few clicks on the web console. You can deploy and manage VMs and BMSs together.
- CCE automates deployment and O&M of containerized applications throughout their lifecycle.
- You can resize clusters and workloads by setting auto scaling policies. In-the-moment load spikes are no longer headaches.
- The console walks you through the steps to upgrade Kubernetes clusters.
- CCE supports turnkey Application Service Mesh (ASM) and Helm charts.

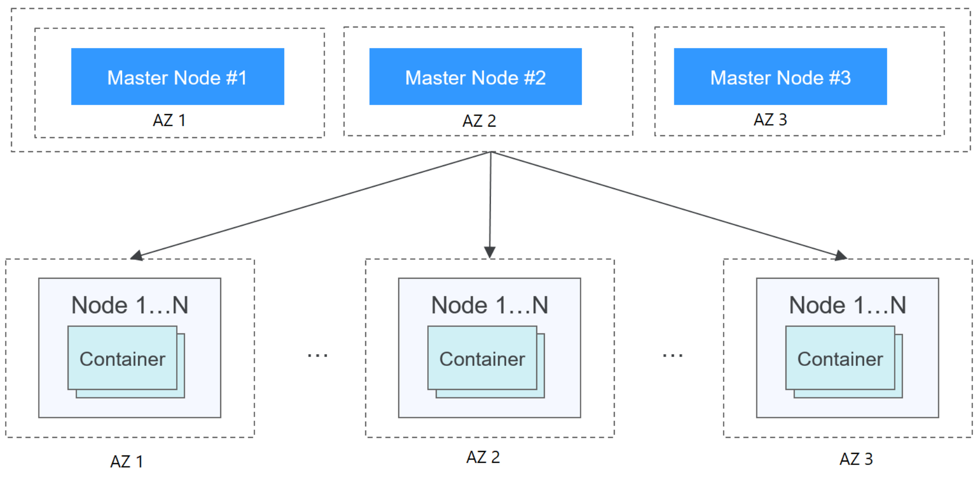
High Performance

- CCE runs on mature IaaS services and heterogeneous compute resources. You can launch containers at scale.
- AI computing is 3x to 5x better with NUMA BMSs and high-speed InfiniBand network cards.

Highly Available and Secure

- HA: Three master nodes in different AZs for your cluster control plane. Multi-active DR for your nodes and workloads. All these ensure service continuity when one of the nodes is down or an AZ gets hit by natural disasters.

Figure 3-1 High-availability setup of clusters



- Secure: Integrating IAM and Kubernetes RBAC, CCE clusters are under your full control. You can set different RBAC permissions for IAM users on the console.

Open and Compatible

- CCE runs on Docker that automates container deployment, discovery, scheduling, and scaling.
- CCE is compatible with native Kubernetes APIs and kubectl. Updates from Kubernetes and Docker communities are regularly incorporated into CCE.

Comparative Analysis of CCE and On-Premises Kubernetes Cluster Management Systems

Table 3-1 CCE clusters versus on-premises Kubernetes clusters

Area of Focus	On-Premises Cluster	CCE
Ease of use	You have to handle all the complexity in deploying and managing Kubernetes clusters. Cluster upgrades are often a heavy burden to O&M personnel.	**Easy to manage and use clusters** You can create a Kubernetes container cluster in a few clicks. No need to set up Docker or Kubernetes environments. CCE automates deployment and O&M of containerized applications throughout their lifecycle. CCE supports turnkey Helm charts. Using CCE is as simple as choosing a cluster and the workloads that you want to run in the cluster. CCE takes care of cluster management and you focus on app development.

Area of Focus	On-Premises Cluster	CCE
Scalability	You have to assess service loads and cluster health before resizing a cluster.	**Managed scaling service** CCE auto scales clusters and workloads according to resource metrics and scaling policies.
Reliability	Only one master node is available in a cluster. Once this node is down, the entire cluster is down, as well as all the applications in it.	**High availability** Enabling HA when creating a cluster will create three master nodes for the control plane. Single points of failure (SPOFs) will not shut down your cluster.
Efficiency	You have to either build an image repository or turn to a third-party one. Images are pulled in serial.	**Rapid deployment with images** CCE connects to SWR to pull images in parallel. Faster pulls, faster container build.
Cost	Heavy upfront investment in installing, managing, and scaling cluster management infrastructure	**Cost effective** You only pay for master nodes and the resources used to run and manage applications.

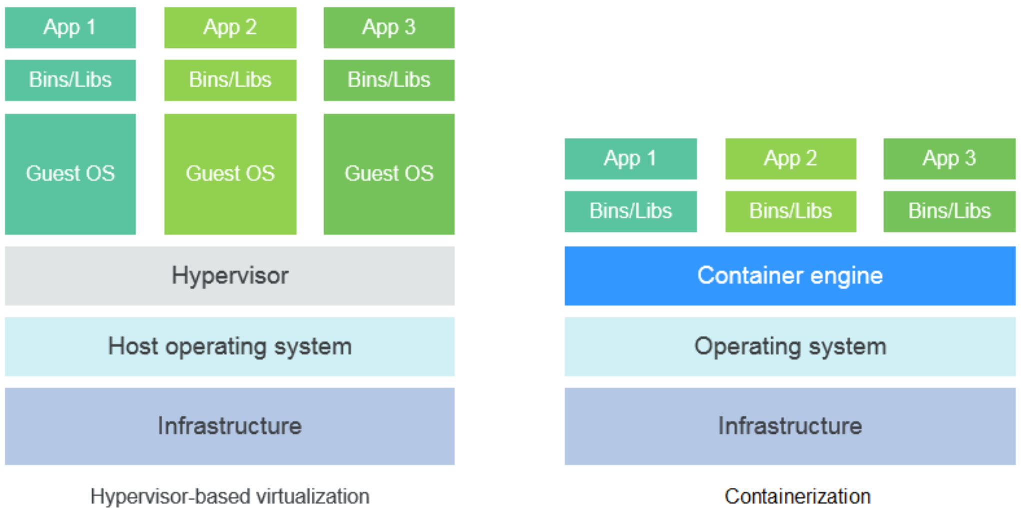
Why Containers?

Docker is written in the Go language designed by Google. It provides operating-system-level virtualization. Linux Control Groups (cgroups), namespaces, and UnionFS (for example, AUFS) isolate each software process. A Docker container packages everything needed to run a software process. Containers are independent from each other and from the host.

Docker has moved forward to enhance container isolation. Containers have their own file systems. They cannot see each other's processes or network interfaces. This simplifies container creation and management.

VMs use a hypervisor to virtualize and allocate hardware resources (such as memory, CPU, network, and disk) of a host machine. A complete operating system runs on a VM. Each VM needs to run its own system processes. On the contrary, a container does not require hardware resource virtualization. It runs an application process directly in the the host machine OS kernel. No resource overheads are incurred by running system processes. Therefore, Docker is lighter and faster than VMs.

Figure 3-2 Comparison between Docker containers and VMs



To sum up, Docker containers have many advantages over VMs.

Resource use

Containers have no overheads for virtualizing hardware and running a complete OS. They are faster than VMs in execution and file storage, while having no memory loss.

Start speed

It takes several minutes to start an application on a VM. Docker containers run on the host kernel without needing an independent OS. Apps in containers can start in seconds or even milliseconds. Development, testing, and deployment can be much faster.

Consistent environment

Different development, testing, and production environments sometimes prevent bug discovery before rollout. A Docker container image includes everything needed to run an application. You can deploy the same copy of configurations in different environments.

Continuous delivery and deployment

"Deploy once, run everywhere" would be great for DevOps personnel.

Docker supports CI/CD by allowing you to customize container images. You compile Dockerfiles to build container images and use CI systems for testing. The Ops team can deploy images into production environments and use CD systems for auto deployment.

The use of Dockerfiles makes the DevOps process visible to everyone in a DevOps team. Developers can better understand both user needs and the O&M headaches faced by the Ops team. The Ops team can also have some knowledge of the must-met conditions to run the application. The knowledge is helpful when the Ops personnel deploy container images in production.

Portability

Docker ensures environmental consistency across development, testing, and production. Portable Docker containers work the same, regardless of their running environments. Physical machines, VMs, public clouds, private clouds, or even laptops, you name it. Apps are now free to migrate and run anywhere.

Application update

Docker images consist of layers. Each layer is only stored once and different images can contain the exact same layers. When transferring such images, those same layers get transferred only once. This makes distribution efficient. Updating a containerized application is also simple. Either edit the top-most writable layer in the final image or add layers to the base image. Docker joins hands with many open source projects to maintain a variety of high-quality official images. You can directly use them in the production environment or easily build new images based on them.

Table 3-2 Containers versus traditional VMs

Feature	Containers	VMs
Start speed	In seconds	In minutes
Disk capacity	MB	GB
Performance	Near-native performance	Weak
Per-machine capacity	Thousands of containers	Tens of VMs

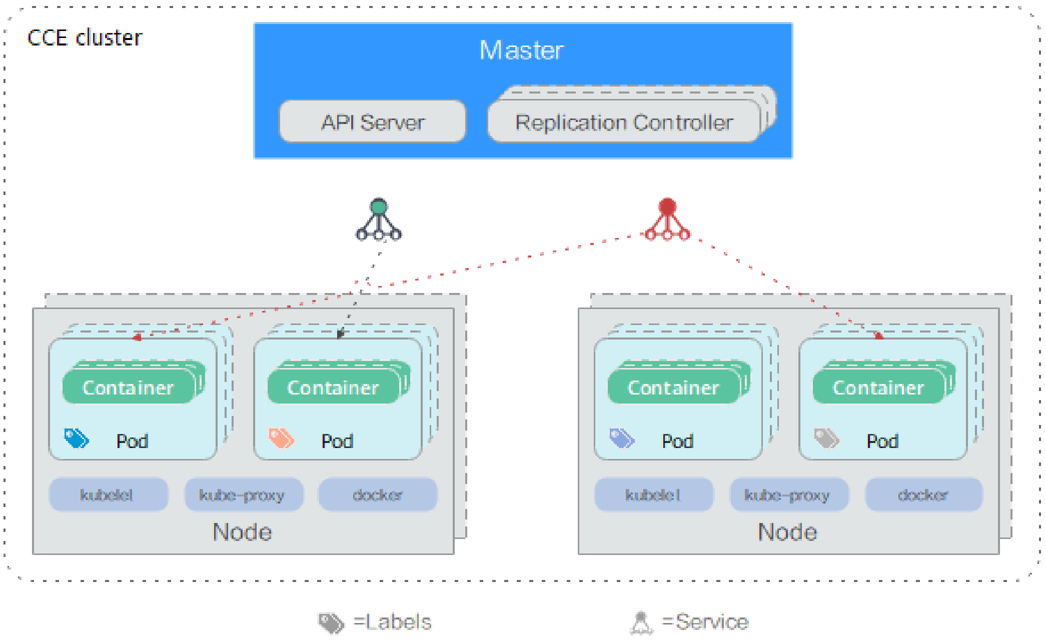
4 Application Scenarios

4.1 Infrastructure and Containerized Application Management

Application Scenario

In CCE, you can run clusters with x86 and Arm nodes. Create and manage Kubernetes clusters. Deploy containerized applications in them. All done in CCE.

Figure 4-1 CCE cluster



Benefits

Containerization requires less resources to deploy application. Services are not uninterrupted during upgrades.

Advantages

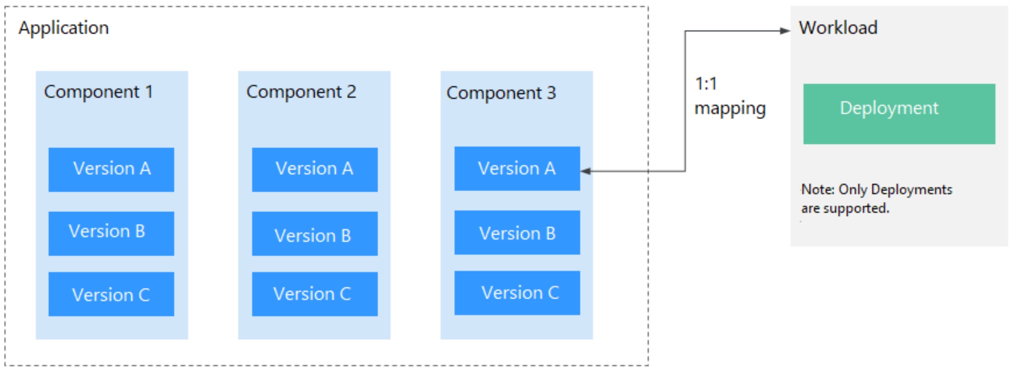
- Multiple types of workloads

  Runs Deployments, StatefulSets, DaemonSets, jobs, and cron jobs to meet different needs.
- Application upgrade

  Upgrades your apps in replace or rolling mode (by proportion or by number of pods), or rolls back the upgrades.
- Auto scaling

  Auto scales your nodes and workloads according to the policies you set.

Figure 4-2 Workload



4.2 Auto Scaling in Seconds

Application Scenarios

- Shopping apps and websites, especially during promotions and flash sales
- Live streaming, where service loads often fluctuate
- Games, where many players may go online in certain time periods

Benefits

CCE auto adjusts capacity to cope with service surges according to the policies you set. CCE adds or reduces cloud servers and containers to scale your cluster and workloads. Your applications will always have the right resources at the right time.

Advantages

- Flexible

  Allows diverse types of scaling policies and scales containers within seconds once triggered.
- Highly available

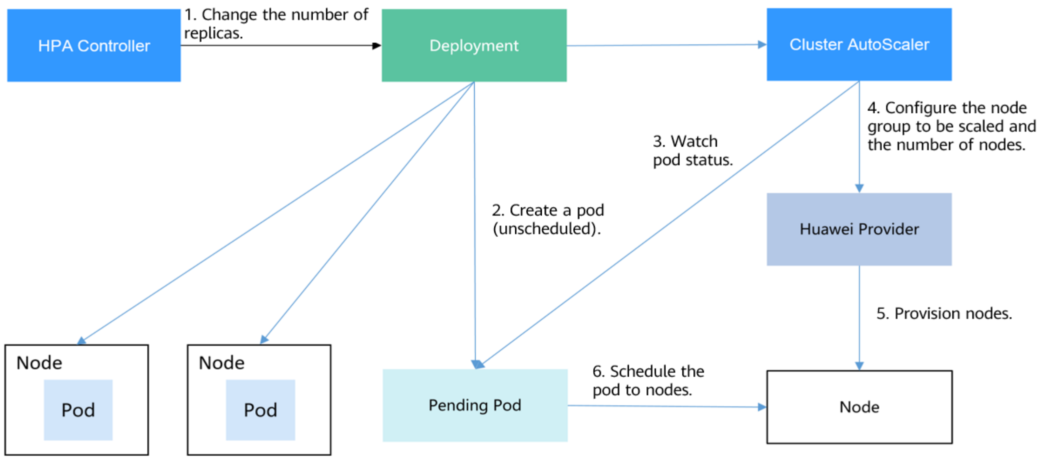
  Monitors pod running and replaces unhealthy pods with new ones.
- Lower costs

  Bills you only for the scaled cloud servers as you use.

Related Services

HPA (Horizontal Pod Autoscaling) + CA (Cluster AutoScaling)

Figure 4-3 How auto scaling works


HPA Controller
1. Change the number of replicas.
Deployment
Cluster AutoScaler
4. Configure the node group to be scaled and the number of nodes.
3. Watch pod status.
2. Create a pod (unscheduled).
Huawei Provider
5. Provision nodes.
Node
Pod
Node
Pod
Pending Pod
6. Schedule the pod to nodes.
Node

4.3 Microservice Management

Application Scenarios

Service systems are becoming more complex. Traditional architectures are failing. A popular solution is microservice, which divides an application into smaller units. Microservices are independently developed, deployed, and scaled. Microservice and container simplify app delivery, while making apps more reliable and scalable.

Distributed apps are now possible. Yet more microservices mean more complex O&M, debugging, and SecOps. These complexities demand extra coding. In this regard, CCE provides an efficient management solution.

Benefits

CCE integrates Application Service Mesh (ASM). Grayscale release, traffic governance, all done in a non-intrusive manner.

Advantages

- Out-of-the-box usability

  Once enabled, ASM serves your apps in CCE to manage traffic flows.
- Intelligent routing

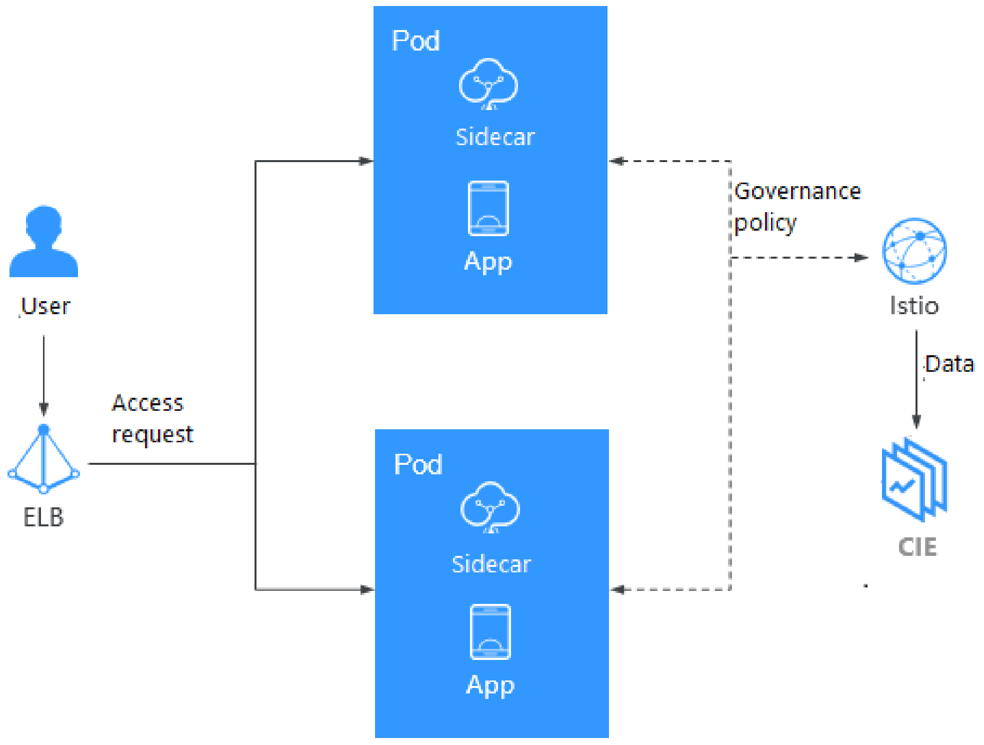
  You can add HTTP/TCP connection policies and security policies without changing your code.
- Visibility into traffic

  CCE works with ASM and APM to monitor your apps in a non-intrusive way. You can enjoy a full view of your services based on the collected data. Capabilities include real-time traffic topology, tracing, performance monitoring, and runtime diagnosis.

Related Services

Elastic Load Balance (ELB), Application Performance Management (APM), Application Operations Management (AOM)

Figure 4-4 Microservice governance



4.4 DevOps and CI/CD

Application Scenario

You may receive a lot feedback and requirements for your apps or services. You may want to boost user experience with new features. Continuous integration (CI) and delivery (CD) can help. CI/CD automates builds, tests, and merges, making app delivery faster.

Benefits

CCE works with SWR to support DevOps and CI/CD. A pipeline automates coding, image build, grayscale release, and deployment based on code sources. Existing CI/CD systems can connect to CCE to containerize legacy applications.

Advantages

- Efficient process

  Reduces scripting workload by more than 80% through streamlined processes.
- Flexible integration

Provides various APIs to integrate with existing CI/CD systems for in-depth customization.

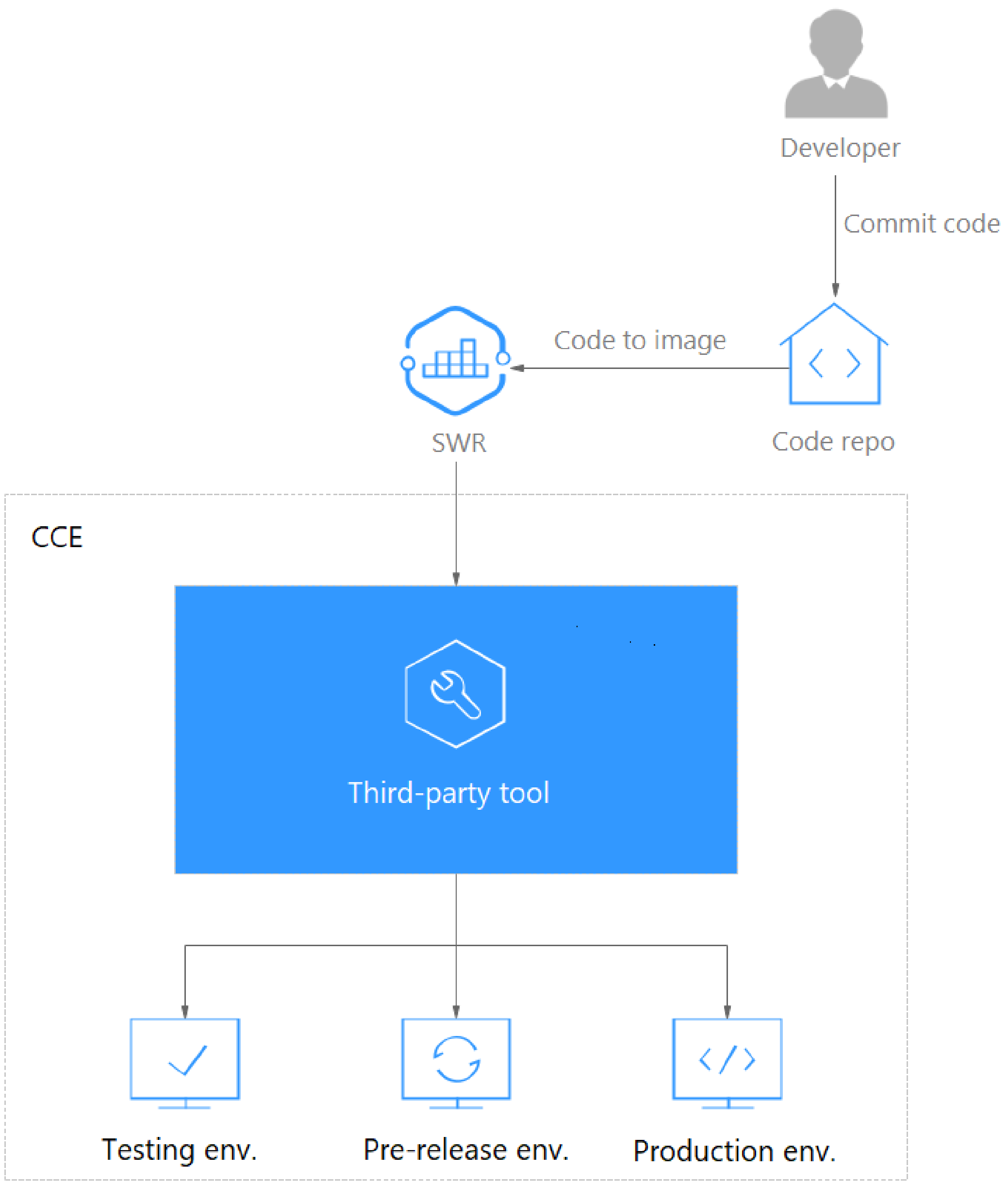
- High performance

  Enables flexible scheduling with a containerized architecture.

Related Services

Software Repository for Container (SWR), Object Storage Service (OBS), Virtual Private Network (VPN)

Figure 4-5 How DevOps works



4.5 Hybrid Cloud Architecture

Application Scenarios

- Multi-cloud deployment and disaster recovery

  Running apps in containers on different clouds can ensure high availability. When a cloud is down, other clouds respond and serve.

- Traffic distribution and auto scaling

  Large organizations often span cloud facilities in different regions. They need to communicate and auto scale — start small and then scale as system load grows. CCE takes care of these for you, cutting the costs of maintaining facilities.

- Migration to the cloud and database hosting

  Industries like finance and security have a top concern on data protection. They want to run critical systems in local IDCs while moving others to the cloud. They also expect one unified dashboard to manage all systems.

- Environment decoupling

  To ensure IP security, you can decouple development from production. Set up one on the public cloud and the other in the local IDC.

Benefits

Your apps and data can flow free on and off the cloud. Resource scheduling and DR are much easier, thanks to environment-independent containers. CCE connects private and public clouds for you to run containers on them.

Advantages

- On-cloud DR

  Multicloud prevents systems from outages. When a cloud is faulty, CCE auto diverts traffic to other clouds to ensure service continuity.

- Automatic traffic distribution

  CCE reduces access latency by processing user requests on the nodes nearest to the users. Overloaded apps in local IDCs can be burst to the cloud backed by auto scaling.

- Decoupling and sharing

  CCE decouples data, environments, and compute capacity. Sensitive data vs general data. Development vs production. Compute-intensive services vs general services. Apps running on-premises can burst to the cloud. Your resources on and off the cloud can be better used.

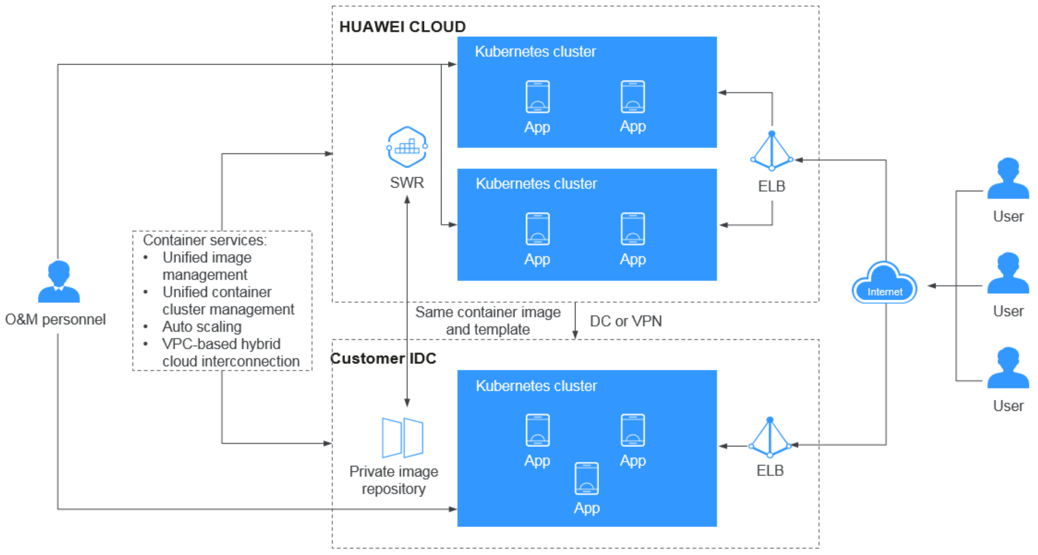
- Lower costs

  Public cloud resource pools, backed by auto scaling, can respond to load spikes in time. Manual operations are no longer needed and you can save big.

Related Services

Elastic Cloud Server (ECS), Direct Connect (DC), Virtual Private Network (VPN), SoftWare Repository for Container (SWR)

Figure 4-6 How hybrid cloud works



4.6 High-Performance Scheduling

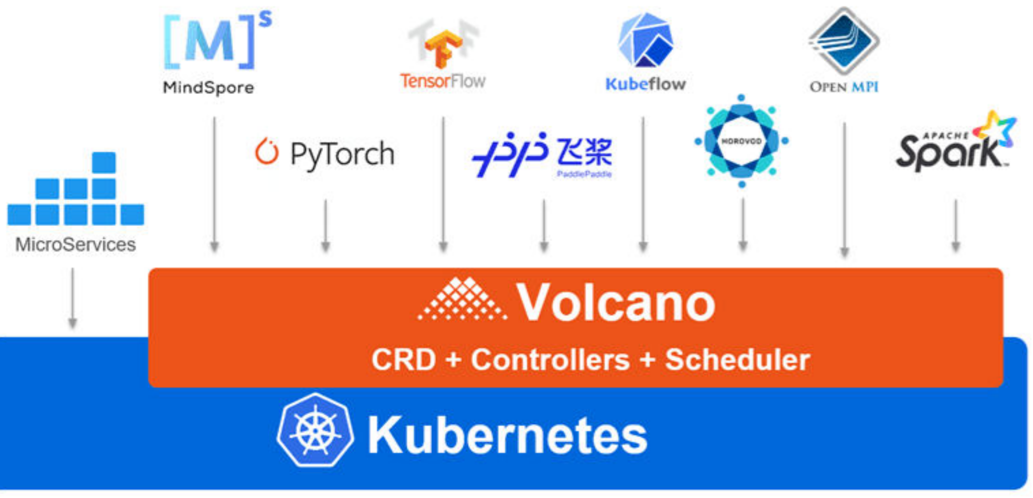
CCE integrates Volcano to support high-performance computing.

Volcano is a Kubernetes-native batch processing system. Volcano provides a universal, scalable, and stable platform to run big data and AI jobs. It is compatible with general computing frameworks for AI, big data, gene sequencing, and rendering tasks. Volcano's excellence in task scheduling and heterogeneous chip management makes task running and management more efficient.

Application Scenario 1: Hybrid Deployment of Multiple Types of Jobs

Multiple types of domain frameworks are developed to support business in different industries. These frameworks, such as Spark, TensorFlow, and Flink, function irreplaceably in their service domains. They are not working alone, as services and businesses are becoming increasingly complex. However, resource scheduling becomes a headache as clusters in these frameworks grow larger and a single service may have fluctuating loads. Therefore, a unified scheduling system is in great demand.

Volcano abstracts a common basic layer for batch computing based on Kubernetes. It supplements Kubernetes in scheduling and provides flexible and universal job abstractions for computing frameworks. These abstractions (Volcano Jobs) are implemented through multi-task templates to describe multiple types of jobs (such as TensorFlow, Spark, MPI, and PyTorch). Different types of jobs can be run together, and Volcano uses its unified scheduling system to realize cluster resource sharing.

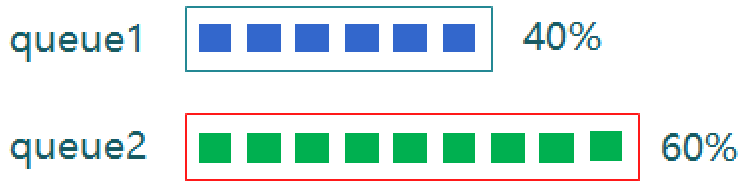


Application Scenario 2: Scheduling Optimization in Multi-Queue Scenarios

Resource isolation and sharing are often required when you use a Kubernetes cluster. However, Kubernetes does not support queues. It cannot share resources when multiple users or departments share a machine. Without queue-based resource sharing, HPC and big data jobs cannot run.

Volcano supports multiple resource sharing mechanisms with queues. You can set the **weight** for a queue. The cluster allocates resources to the queue by calculating the ratio of the weight of the queue to the total weight of all queues. You can also set the resource **capability** for a queue to determine the upper limit of resources that can be used by the queue.

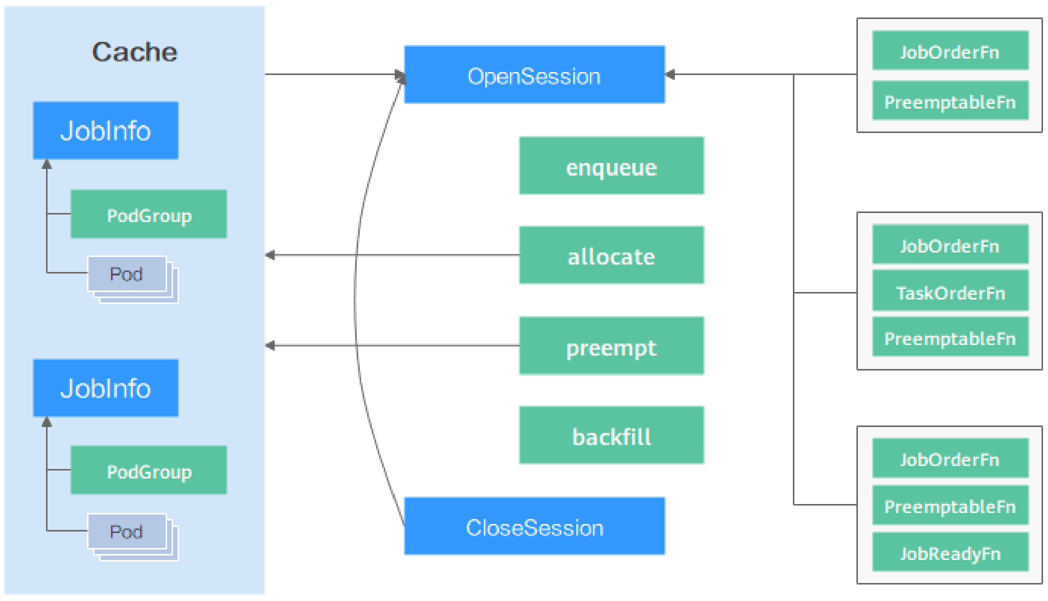
For example, in the following figure, queue 1 is allocated 40% of the cluster resources, and 60% for queue 2. In this way, two queues can be mapped to different departments or projects to use resources in the same cluster. If a queue has idle resources, they can be allocated to jobs in another queue.



Application Scenario 3: Multiple Advanced Scheduling Policies

Containers are scheduled to nodes that satisfy their requirements on compute resources, such as CPU, memory, and GPU. Normally, there will be more than one qualified node. Each could have different volume of available resources left for new workloads. Volcano automatically analyzes the resource utilization of each scheduling plan and help you achieve the optimal deployment results in great ease.

The following figure shows how the Volcano scheduler schedules resources. First, the scheduler loads the pod and PodGroup information in the API server to the scheduler cache. In a scheduler session, Volcano goes through three phases: OpenSession, action calling, and CloseSession. In OpenSession, the scheduling policy you configured in the scheduler plugin is loaded. In action calling, the configured actions are called one by one and the loaded scheduling policy is used. In CloseSession, final operations are performed to complete scheduling.



Volcano scheduler provides plugins to support multiple scheduling actions (such as enqueue, allocate, preempt, reclaim and backfill) and scheduling policies (such as gang, priority, drf, proportion and binpack). You can configure them as required. The APIs provided by the scheduler can also be used for customized development.

Application Scenario 4: High-Precision Resource Scheduling

Volcano provides high-precision resource scheduling policies for AI and big data jobs to improve compute efficiency. Take TensorFlow as an example. Configure affinity between ps and worker and anti-affinity between ps and ps, so that ps and worker to the same node. This improves the networking and data interaction performance between ps and worker, thereby improving the compute efficiency. However, when scheduling pods, the default Kubernetes scheduler only checks whether the affinity and anti-affinity configurations of these pods conflict with those of all running pods in the cluster, and does not consider subsequent pods that may also need scheduling.

The task-topology algorithm provided by Volcano calculates the task and node priorities based on the affinity and anti-affinity configurations between tasks in a job. The task affinity and anti-affinity policies in a job and the task-topology algorithm ensure that the tasks with affinity configurations are preferentially scheduled to the same node, and pods with anti-affinity configurations are scheduled to different nodes. The difference between the task-topology algorithm and the default Kubernetes scheduler is that the task-topology algorithm considers the pods to be scheduled as a whole. When pods are scheduled in batches, the affinity and anti-affinity settings between unscheduled pods are considered and applied to the scheduling processes of pods based on priorities.

Benefits

Running containers on high-performance GPU-accelerated cloud servers significantly improves AI computing performance by three to five folds. GPUs can cost a lot and sharing a GPU among containers greatly reduces AI computing costs. In addition to performance and cost advantages, CCE also offers fully managed clusters that will hide all the complexity in deploying and managing your AI applications so you can focus on high-value development.

Advantages

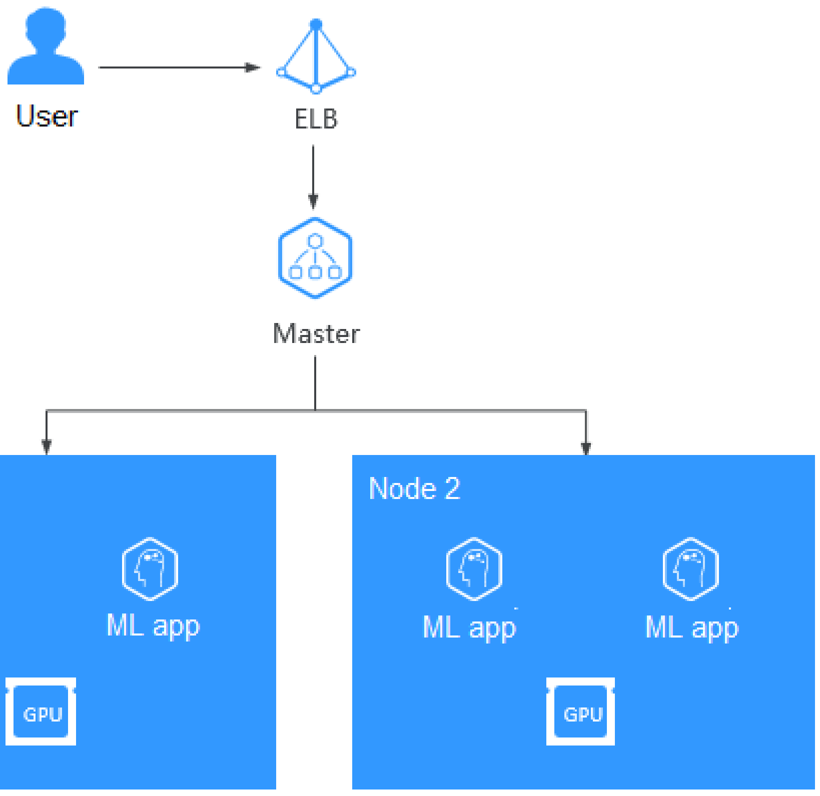
By integrating Volcano, CCE has the following advantages in running high-performance computing, big data, and AI jobs:

- **Hybrid deployment** of HPC, big data, and AI jobs
- **Optimized multi-queue scheduling**: Multiple queues can be used for multi-tenant resource sharing and group planning based on priorities and time periods.
- **Advanced scheduling policies**: gang scheduling, fair scheduling, resource preemption, and GPU topology
- **Multi-task template**: You can use a template to define multiple tasks in a single Volcano Job, beyond the limit of Kubernetes native resources. Volcano Jobs can describe multiple job types, such as TensorFlow, MPI, and PyTorch.
- **Job extension plugins**: The Volcano Controller allows you to configure plugins to customize environment preparation and cleanup in stages such as job submission and pod creation. For example, before submitting a common MPI job, you can configure the SSH plugin to provide the SSH information of pod resources.

Related Services

GPU-accelerated Cloud Server (GACS), Elastic Load Balance (ELB), and Object Storage Service (OBS)

Figure 4-7 How AI computing works



5 Security

5.1 Shared Responsibilities

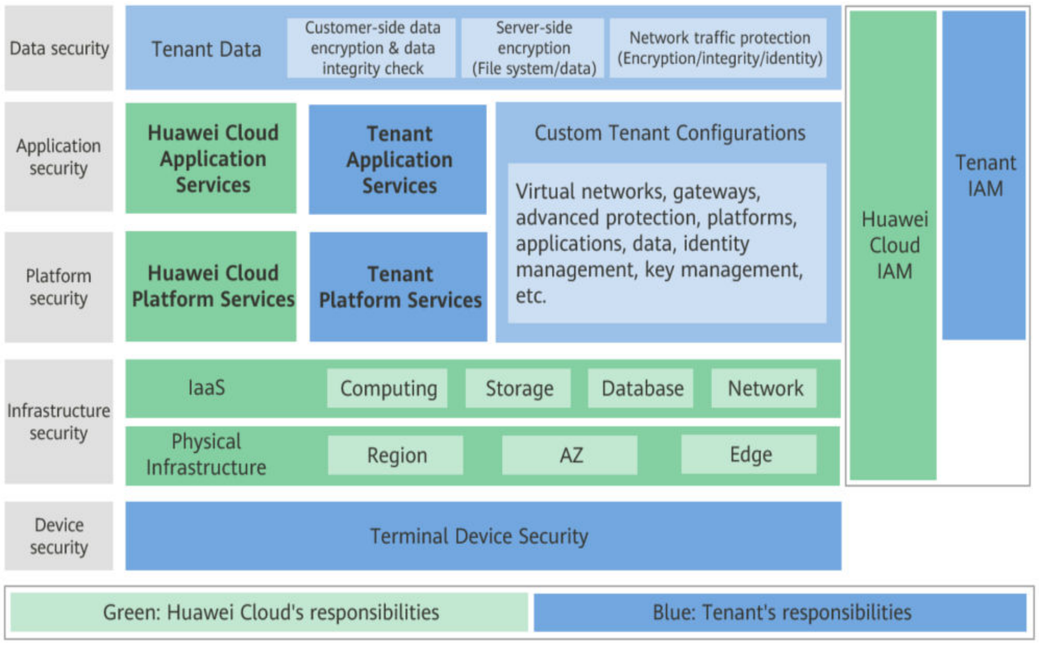
Huawei guarantees that its commitment to cyber security will never be outweighed by the consideration of commercial interests. To cope with emerging cloud security challenges and pervasive cloud security threats and attacks, Huawei Cloud builds a comprehensive cloud service security assurance system for different regions and industries based on Huawei's unique software and hardware advantages, laws, regulations, industry standards, and security ecosystem.

Figure 5-1 illustrates the responsibilities shared by Huawei Cloud and users.

- **Huawei Cloud**: Ensure the security of cloud services and provide secure clouds. Huawei Cloud's security responsibilities include ensuring the security of our IaaS, PaaS, and SaaS services, as well as the physical environments of the Huawei Cloud data centers where our IaaS, PaaS, and SaaS services operate. Huawei Cloud is responsible for not only the security functions and performance of our infrastructure, cloud services, and technologies, but also for the overall cloud O&M security and, in the broader sense, the security compliance of our infrastructure and services.
- **Tenant**: Use the cloud securely. Tenants of Huawei Cloud are responsible for the secure and effective management of the tenant-customized configurations of cloud services including IaaS, PaaS, and SaaS. This includes but is not limited to virtual networks, the OS of virtual machine hosts and guests, virtual firewalls, API Gateway, advanced security services, all types of cloud services, tenant data, identity accounts, and key management.

Huawei Cloud Security White Paper elaborates on the ideas and measures for building Huawei Cloud security, including cloud security strategies, the shared responsibility model, compliance and privacy, security organizations and personnel, infrastructure security, tenant service and security, engineering security, O&M security, and ecosystem security.

Figure 5-1 Huawei Cloud shared security responsibility model


Data security
Tenant Data
Customer-side data encryption & data integrity check
Server-side encryption (File system/data)
Network traffic protection (Encryption/integrity/identity)
Application security
Huawei Cloud Application Services
Tenant Application Services
Custom Tenant Configurations
Virtual networks, gateways, advanced protection, platforms, applications, data, identity management, key management, etc.
Platform security
Huawei Cloud Platform Services
Tenant Platform Services
Infrastructure security
IaaS
Computing
Storage
Database
Network
Physical Infrastructure
Region
AZ
Edge
Huawei Cloud IAM
Tenant IAM
Device security
Terminal Device Security
Green: Huawei Cloud's responsibilities
Blue: Tenant's responsibilities

5.2 Identity Authentication and Access Control

Identity Authentication

You can use CCE via the CCE console, APIs, and SDKs, but all your requests must be authenticated.

CCE provides identity authentication for cloud services and clusters.

For cloud services, APIs are opened through API Gateway. You can operate cloud infrastructure resources (for example, creating nodes) and cluster resources (for example, creating workloads). Two authentication modes are available. Use either of them. For details, see **Authentication**.

- Token: Requests are authenticated using tokens. For details about tokens, see **Obtaining a User Token Through Password Authentication**.
- AK/SK: Requests are encrypted using AK/SK pairs. This mode is more secure. For details about access keys, see **Access Keys**.

For clusters, CCE allows you to operate cluster resources (for example, creating workloads) through the Kubernetes native API server, but not cloud infrastructure resources (for example, creating nodes). You need to access the cluster using a kubeconfig file. For details, see **Connecting to a Cluster Using kubectl**. You can obtain a kubeconfig file in the following ways:

- Console: **Obtaining a Cluster Certificate**
- API: **Obtaining a Cluster Certificate**

Access Control

CCE combines IAM and Kubernetes RBAC for you to manage cluster and namespace permissions. You can assign different permissions for IAM users and user groups under your account. For details, see **Permissions Management**.

Table 5-1 CCE access control

Permissions	Description	Documentation
Cluster	Cluster permissions management evolves out of IAM system policies. IAM users in the same user group have the same permissions. On IAM, you can configure system policies to describe which IAM user groups can perform which operations on cluster resources. For example, you can grant user group A permissions to create and delete cluster X, add nodes, or install add-ons, while granting user group B permissions to view information about cluster X.	**Cluster Permissions (IAM-based)**
Namespace	You can regulate users' or user groups' access to Kubernetes resources in a single namespace based on their Kubernetes RBAC roles. CCE has been enhanced based on open source capabilities. You can assign RBAC roles to IAM users or user groups on the CCE console or calling APIs using IAM tokens.	**Namespace Permissions (Kubernetes RBAC-based)**

5.3 Data Protection

CCE takes different measures to keep data secure and reliable.

Table 5-2 CCE data protection measures and features

Measure	Description	Documentation
Certificate for service discovery	Applications in CCE clusters can use HTTPS for secure data transmission. You can create Services (layer 4) and ingresses (layer 7) to connect to a load balancer as required.	**Configuring HTTPS Certificates** **Enabling HTTP for Services**

Measure	Description	Documentation
HA deployment	HA solutions in CCE: ● Deploy three master nodes for a cluster. ● Distribute worker nodes in different AZs ● Create a workload and distribute it to different AZs or nodes.	**Implementing High Availability for Containers in CCE**
Disk encryption	CCE supports multiple types of storage resources, as well as HA and encryption measures to secure your data.	**Storage Overview**
Cluster secret	A secret is a cluster resource that holds sensitive data, such as authentication and key information. Its contents are user-defined. After creating secrets, you can use them as files or environment variables in a containerized workload.	**Creating a Secret**
Protection for critical operations	With this function enabled, the system authenticates user's identity when they perform any risky operation like deleting a cluster.	**Critical Operation Protection**

5.4 Audit and Logging

Audit

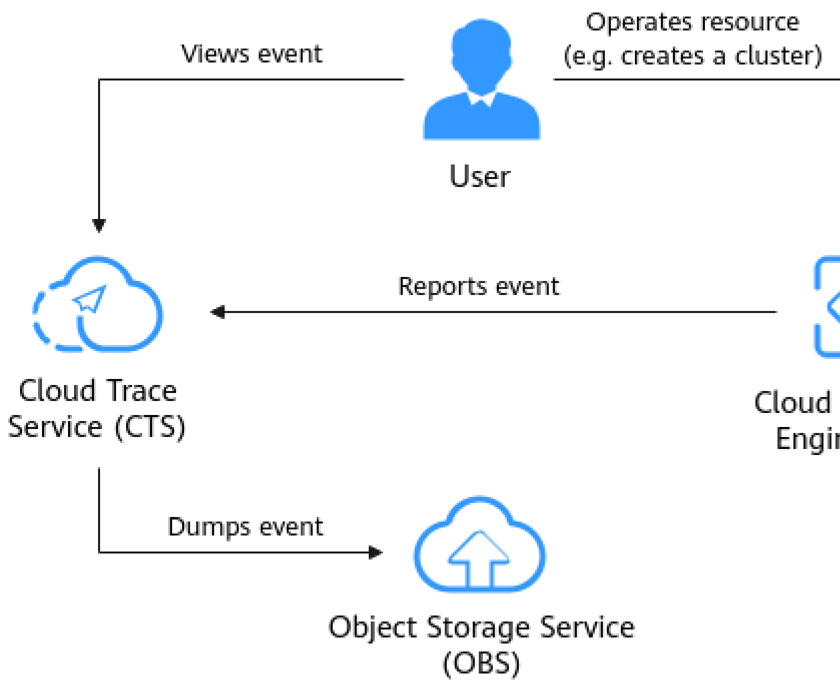
Cloud Trace Service (CTS) records operations on the cloud resources in your account. You can use the logs generated by CTS to perform security analysis, track resource changes, audit compliance, and locate faults.

After you enable CTS, it starts recording operations on CCE resources and stores the operation records of the last seven days. For details about CCE operations that can be recorded by CTS, see **CCE Operations Supported by CTS**.

For details about how to enable and configure CTS, see **Enabling CTS**.

For details about how to view CTS logs, see **Querying CTS Logs**.

Figure 5-2 CTS


Views event
Operates resource (e.g. creates a cluster)
User
Reports event
Cloud Trace Service (CTS)
Cloud Container Engine (CCE)
Dumps event
Object Storage Service (OBS)

Logs

CCE allows you to configure policies for collecting, managing, and analyzing workload logs periodically to prevent logs from being over-sized.

CCE works with AOM to collect workload logs. When a node is created, the ICAgent (the DaemonSet named **icagent** in the kube-system namespace of the cluster) of AOM is installed by default. After the ICAgent collects workload logs (*.log, *.trace, and *.out formats) and reports them to AOM, you can view them on the CCE or AOM console.

For details about workload logging, see **Container Logs**.

5.5 Security Risk Monitoring

CCE works with AOM to monitor your clusters. CCE monitors and collects system metrics from underlying cluster resources and workloads running in the clusters. You can also define custom monitoring metrics for your workloads.

- System metrics

  These metrics cover CPU, memory, and disks. For details, see **Monitoring Overview**. You can view these metrics of clusters, nodes, and workloads on the CCE or AOM console.
- Custom metrics

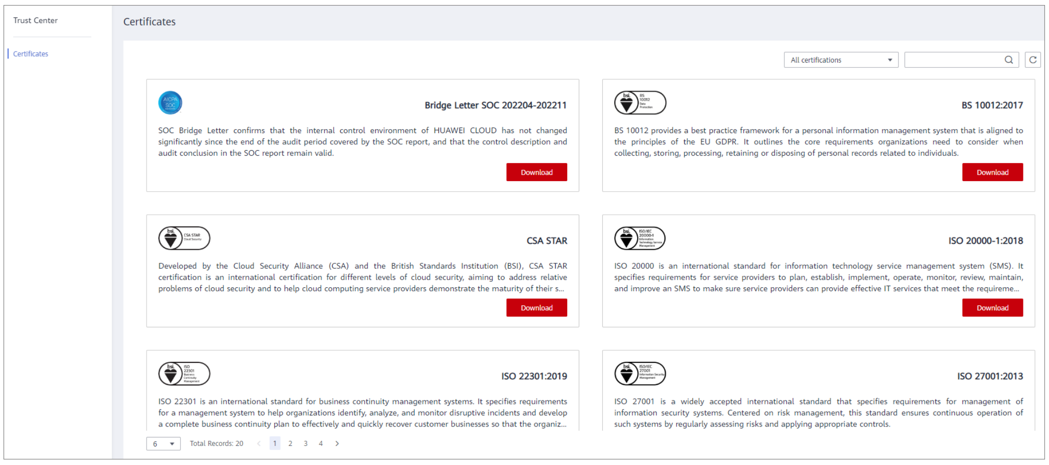
  CCE can collect custom metrics and report them to AOM. For details, see **Custom Monitoring**.

5.6 Certificates

Compliance Certificates

Huawei Cloud services and platforms have obtained various security and compliance certifications from authoritative organizations, such as International Organization for Standardization (ISO). You can **download** them from the console.

Figure 5-3 Downloading compliance certificates



Bridge Letter SOC 202204-202211

SOC Bridge Letter confirms that the internal control environment of HUAWEI CLOUD has not changed significantly since the end of the audit period covered by the SOC report, and that the control description and audit conclusion in the SOC report remain valid.

BS 10012:2017

BS 10012 provides a best practice framework for a personal information management system that is aligned to the principles of the EU GDPR. It outlines the core requirements organizations need to consider when collecting, storing, processing, retaining or disposing of personal records related to individuals.

CSA STAR

Developed by the Cloud Security Alliance (CSA) and the British Standards Institution (BSI), CSA STAR certification is an international certification for different levels of cloud security, aiming to address relative problems of cloud security and to help cloud computing service providers demonstrate the maturity of their s...

ISO 20000-1:2018

ISO 20000 is an international standard for information technology service management system (SMS). It specifies requirements for service providers to plan, establish, implement, operate, monitor, review, maintain, and improve an SMS to make sure service providers can provide effective IT services that meet the requireme...

ISO 22301:2019

ISO 22301 is an international standard for business continuity management systems. It specifies requirements for a management system to help organizations identify, analyze, and monitor disruptive incidents and develop a complete business continuity plan to effectively and quickly recover customer businesses so that the organiz...

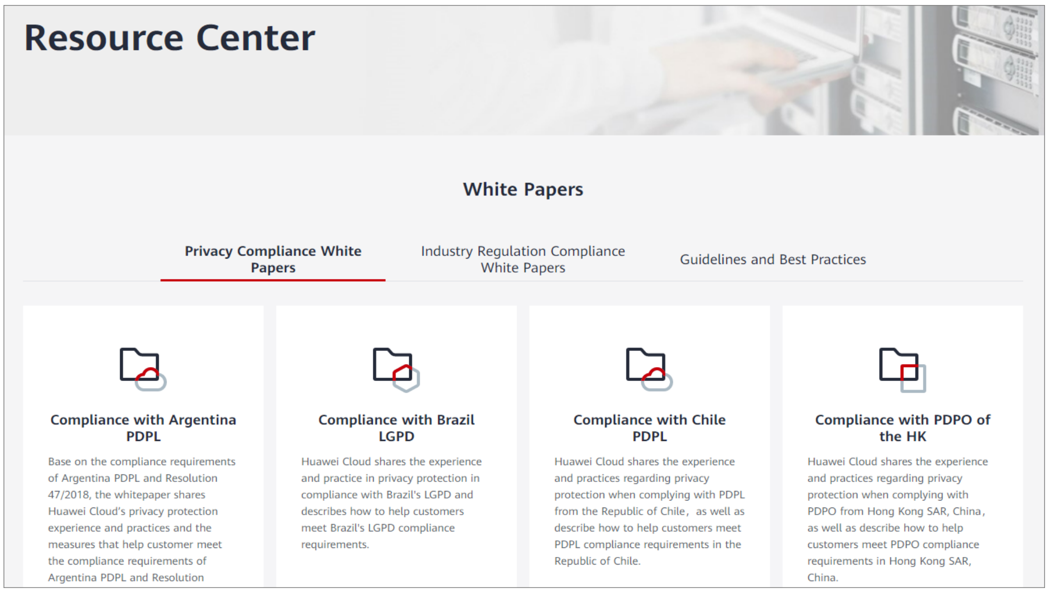
ISO 27001:2013

ISO 27001 is a widely accepted international standard that specifies requirements for management of information security systems. Centered on risk management, this standard ensures continuous operation of such systems by regularly assessing risks and applying appropriate controls.

Resource Center

Huawei Cloud also provides the following resources to help users meet compliance requirements. For details, see **Resource Center**.

Figure 5-4 Resource center


Resource Center
White Papers
Privacy Compliance White Papers
Industry Regulation Compliance White Papers
Guidelines and Best Practices

Compliance with Argentina PDPL

Base on the compliance requirements of Argentina PDPL and Resolution 47/2018, the whitepaper shares Huawei Cloud's privacy protection experience and practices and the measures that help customer meet the compliance requirements of Argentina PDPL and Resolution

Compliance with Brazil LGPD

Huawei Cloud shares the experience and practice in privacy protection in compliance with Brazil's LGPD and describes how to help customers meet Brazil's LGPD compliance requirements.

Compliance with Chile PDPL

Huawei Cloud shares the experience and practices regarding privacy protection when complying with PDPL from the Republic of Chile, as well as describe how to help customers meet PDPL compliance requirements in the Republic of Chile.

Compliance with PDPO of the HK

Huawei Cloud shares the experience and practices regarding privacy protection when complying with PDPO from Hong Kong SAR, China, as well as describe how to help customers meet PDPO compliance requirements in Hong Kong SAR, China.

6 Notes and Constraints

This section describes the notes and constraints on using CCE.

Clusters and Nodes

- After a cluster is created, the following items cannot be changed:
  – Cluster type. For example, change a **Kunpeng cluster** to a **CCE cluster**.
  – Number of master nodes. For example, you cannot change a non-HA cluster (with one master node) to an HA cluster (with three master nodes).
  – AZ of a master node.
  – Network configuration of the cluster, such as the VPC, subnet, container CIDR block, Service CIDR block, IPv6 settings, and kube-proxy (forwarding) settings.
  – Network model. For example, change the **tunnel network** to the **VPC network**.
- Applications cannot be migrated between different namespaces.
- Currently, the created ECS instances (nodes) support the **pay-per-use** and **yearly/monthly** billing modes. Other resources (such as load balancers) support the pay-per-use billing mode. You can change the billing mode from pay-per-use to yearly/monthly on the management console for the created ECS instances.
- Nodes created during cluster creation support **pay-per-use** and **yearly/monthly** billing modes, but with the following constraints:
  – If the cluster to be created is pay-per-use, the nodes created in the cluster must also be pay-per-use.
  – If the cluster to be created is billed on a yearly/monthly basis, nodes in the cluster are either pay-per-use or billed on a yearly/monthly basis.
  – If nodes added after cluster creation are billed on a yearly/monthly basis, they need to be renewed separately from the cluster.

  Note: If you purchase a node after a cluster is created, the billing mode of the node is not restricted by that of the cluster.
- Underlying resources, such as ECSs (nodes), are limited by quotas and their inventory. Therefore, only some nodes may be successfully created during cluster creation, cluster scaling, or auto scaling.

- The ECS (node) specifications must be higher than 2 cores and 4 GB memory.
- To access a CCE cluster through a VPN, ensure that the VPN CIDR block does not conflict with the VPC CIDR block where the cluster resides and the container CIDR block.

Networking

- By default, a NodePort Service is accessed within a VPC. If you need to use an EIP to access a NodePort Service through public networks, bind an EIP to the node in the cluster in advance.
- LoadBalancer Services allow workloads to be accessed from public networks through **ELB**. This access mode has the following restrictions:
  - It is recommended that automatically created load balancers not be used by other resources. Otherwise, these load balancers cannot be completely deleted, causing residual resources.
  - Do not change the listener name for the load balancer in clusters of v1.15 and earlier. Otherwise, the load balancer cannot be accessed.
- Constraints on network policies:
  - Only clusters that use the tunnel network model support network policies.
  - Egresses are not supported for network policies.
  - Network isolation is not supported for IPv6 addresses.

Volumes

- Constraints on EVS volumes:
  - By default, CCE creates EVS disks billed in **pay-per-use** mode. To use EVS disks billed in **yearly/monthly** mode, see **Yearly/Monthly-Billed EVS Disks**.
  - EVS disks cannot be attached across AZs and cannot be used by multiple workloads, multiple pods of the same workload, or multiple jobs.
  - Data in a shared disk cannot be shared between nodes in a CCE cluster. If the same EVS disk is attached to multiple nodes, read and write conflicts and data cache conflicts may occur. When creating a Deployment, you are advised to create only one pod if you want to use EVS disks.
  - For clusters earlier than v1.19.10, if an HPA policy is used to scale out a workload with EVS volumes mounted, the existing pods cannot be read or written when a new pod is scheduled to another node.

    For clusters of v1.19.10 and later, if an HPA policy is used to scale out a workload with EVS volume mounted, a new pod cannot be started because EVS disks cannot be attached.
  - When you create a StatefulSet and add a cloud storage volume, existing EVS volumes cannot be used.
  - EVS disks that have partitions or have non-ext4 file systems cannot be imported.
  - Container storage in CCE clusters of Kubernetes 1.13 or later version supports encryption. Currently, E2E encryption is supported only in certain regions.

  - EVS volumes cannot be created in specified enterprise projects. Only the default enterprise project is supported.
- Constraints on SFS volumes:
  - Container storage in CCE clusters of Kubernetes 1.13 or later version supports encryption. Currently, E2E encryption is supported only in certain regions.
  - Volumes cannot be created in specified enterprise projects. Only the default enterprise project is supported.
- Constraints on OBS volumes:
  - CCE clusters of v1.7.3-r8 and earlier do not support OBS volumes. You need to upgrade these clusters or create clusters of a later version that supports OBS.
  - Kunpeng clusters do not support obsfs. Therefore, parallel file systems cannot be mounted.
  - Volumes cannot be created in specified enterprise projects. Only the default enterprise project is supported.
- Constraints on snapshots and backups:
  - The snapshot function is available **only for clusters of v1.15 or later** and requires the CSI-based everest add-on.
  - The subtype (common I/O, high I/O, or ultra-high I/O), disk mode (SCSI or VBD), data encryption, sharing status, and capacity of an EVS disk created from a snapshot must be the same as those of the disk associated with the snapshot. These attributes cannot be modified after being queried or set.

Services

A Service is a Kubernetes resource object that defines a logical set of pods and a policy by which to access them.

A maximum of 6,000 Services can be created in each namespace.

CCE Cluster Resources

There are resource quotas for your CCE clusters in each region.

Item	Constraints on Common Users
Total number of clusters in a region	50
Number of nodes in a cluster (cluster management scale)	You can select 50, 200, 1,000, or 2,000 nodes. A maximum of 5,000 nodes are supported.
Maximum number of container pods created on each worker node	This number can be set on the console when you are creating a cluster. In the VPC network model, a maximum of 256 pods can be created.

Dependent Underlying Cloud Resources

Category	Item	Constraints on Common Users
Compute	Pods	1,000
	Cores	8,000
	RAM capacity (MB)	16384000
Networking	VPCs per account	5
	Subnets per account	100
	Security groups per account	100
	Security group rules per account	5000
	Routes per route table	100
	Routes per VPC	100
	VPC peering connections per region	50
	Network ACLs per account	200
	Layer 2 connection gateways per account	5
Load balancing	Elastic load balancers	50
	Load balancer listeners	100
	Load balancer certificates	120
	Load balancer forwarding policies	500
	Load balancer backend host group	500
	Load balancer backend server	500

7 Pricing Details

Billing Items

CCE billing comprises cluster costs and IaaS resource costs.

1. **Cluster costs** depend on the cluster type, scale (max. nodes in the cluster), and availability (HA mode).

   NOTE

   The management scale indicates the number of ECSs or BMSs in a cluster.

   For more details, see **CCE Pricing Details**.

2. **IaaS resource costs** are incurred from the IaaS resources you use to run worker nodes in your cluster. These resources, created either manually or automatically, include ECSs, EVS disks, EIPs, bandwidth, and load balancers.

   For more pricing details, see **Product Pricing Details**.

Billing Modes

CCE is billed on a pay-per-use or yearly/monthly basis.

- Pay-per-use: It is a pay-after-use mode. Billing starts when a resource is provisioned and stops when the resource is deleted. You can use cloud resources as required and stop paying for them when you no longer need them. There is no upfront payment for excess capacity.

  NOTE

  The following are pricing principles in the case of CCE cluster hibernation or node shutdown. Note that there are many types of cluster nodes and ECS is used as an example.

  - **Cluster hibernation**: For a hibernated cluster, you will not be billed for master nodes, but for the resources used by or bound to the cluster, such as the EVS disks, EIPs, and bandwidth (billed in their own ways, yearly/monthly or pay-per-use).
  - **Node shutdown**: Hibernating a cluster will not stop worker nodes in the cluster. You can select **Stop all nodes in the cluster** to do so. You can also stop a node on the ECS console after hibernating a cluster. For details, see **Stopping a Node**.

    Most nodes are no longer billed after they are stopped, excluding certain types of ECSs (ones with local disks attached, such as disk-intensive and ultra-high I/O ECSs). For details, see **ECS Billing**.

- Yearly/monthly: It is a pay-before-use mode. Yearly/monthly billing provides a more significant discount than pay-per-use and is recommended for long-term use of cloud services. When you purchase a yearly/monthly package, the system will deduct the package cost from your cloud account based on the chosen specifications.
- Billing mode change: The billing mode cannot be changed within the billing cycle.

NOTICE

- Clusters follow a tiered pricing plan. Pricing for each tier varies with cluster size and type.
- Once a monthly/yearly subscription has expired or a pay-per-use resource becomes in arrears, Huawei Cloud provides a period of time during which you can renew the resource or top up your account. Within the grace period, you can still access and use your cloud service. For details, see **What Is a Grace Period? How Long Is It?What Is a Retention Period? How Long Is It?**

Configuration Changes

From pay-per-use to yearly/monthly billing: You can change the cluster billing mode from pay-per-use to yearly/monthly billing. After the change, master nodes, worker nodes, and cloud resources (such as EVS disks and EIPs) used by your cluster will all be billed on a yearly/monthly basis and a new order will be generated. The nodes and cloud resources will be ready for use immediately after you pay for the new order.

From yearly/monthly billing to pay-per-use: Clusters billed on a yearly/monthly basis cannot change to pay-per-use within the billing cycle. Note that pay-per-use clusters can be directly deleted, but clusters billed on a yearly/monthly basis cannot be deleted. To stop using the clusters billed on a yearly/monthly basis, go to the Billing Center and **unsubscribe from them**.

Notes

- Cash coupons will not be returned after you downgrade specifications of the cloud servers that are purchased using cash coupons.
- You will need to pay the price difference between the original and new specifications after upgrading cloud server specifications.
- Downgrading cloud server specifications (the amount of CPU or memory resources) will impair cloud server performance.
- If you downgrade cloud server specifications and then upgrade it to the original specifications, you will still need to pay the price difference incurred by the upgrade.

8 Permissions

CCE allows you to assign permissions to IAM users and user groups under your tenant accounts. CCE combines the advantages of Identity and Access Management (IAM) and Kubernetes Role-based Access Control (RBAC) to provide a variety of authorization methods, including IAM fine-grained/token authorization and cluster-/namespace-scoped authorization.

CCE permissions are described as follows:

- **Cluster-level permissions**: Cluster-level permissions management evolves out of the system policy authorization feature of IAM. IAM users in the same user group have the same permissions. On IAM, you can configure system policies to describe which IAM user groups can perform which operations on cluster resources. For example, you can grant user group A to create and delete cluster X, add a node, or install an add-on, while granting user group B to view information about cluster X.

  Cluster-level permissions involve CCE non-Kubernetes APIs and support fine-grained IAM policies and enterprise project management capabilities.

- **Namespace-level permissions**: You can regulate users' or user groups' access to **Kubernetes resources**, such as workloads, jobs, and Services, in a single namespace based on their Kubernetes RBAC roles. CCE has also been enhanced based on open-source capabilities. It supports RBAC authorization based on IAM user or user group, and RBAC authentication on access to APIs using IAM tokens.

  Namespace-level permissions involve CCE Kubernetes APIs and are enhanced based on the Kubernetes RBAC capabilities. Namespace-level permissions can be granted to IAM users or user groups for authentication and authorization, but are independent of fine-grained IAM policies. For details, see **Using RBAC Authorization**.

CAUTION

- **Cluster-level permissions** are configured only for cluster-related resources (such as clusters and nodes). You must also configure **namespace permissions** to operate Kubernetes resources (such as workloads, jobs, and Services).
- After you create a cluster of v1.11.7-r2 or later, CCE automatically assigns the cluster-admin permissions of all namespaces in the cluster to you, which means you have full control on the cluster and all resources in all namespaces.

Cluster-level Permissions (Assigned by Using IAM System Policies)

By default, new IAM users do not have permissions assigned. You need to add a user to one or more groups, and attach permissions policies or roles to these groups. Users inherit permissions from the groups to which they are added and can perform specified operations on cloud services based on the permissions.

CCE is a project-level service deployed and accessed in specific physical regions. To assign CCE permissions to a user group, specify the scope as region-specific projects and select projects for the permissions to take effect. If **All projects** is selected, the permissions will take effect for the user group in all region-specific projects. When accessing CCE, the users need to switch to a region where they have been authorized to use the CCE service.

You can grant users permissions by using roles and policies.

- Roles: A type of coarse-grained authorization mechanism that defines permissions related to user responsibilities. This mechanism provides only a limited number of service-level roles for authorization. When using roles to assign permissions, you need to also assign other roles on which the permissions depend to take effect. However, roles are not an ideal choice for fine-grained authorization and secure access control.
- Policies: A type of fine-grained authorization mechanism that defines permissions required to perform operations on specific cloud resources under certain conditions. This mechanism allows for more flexible policy-based authorization, meeting requirements for secure access control. For example, you can assign users only the permissions for managing a certain type of clusters and nodes. Most policies define permissions based on APIs. For the API actions supported by CCE, see **Permissions Policies and Supported Actions**.

Table 8-1 lists all the system permissions supported by CCE.

Table 8-1 System permissions supported by CCE

Role/ Policy Name	Description	Type	Dependencies
CCE Administrator	Read and write permissions for CCE clusters and all resources (including workloads, nodes, jobs, and Services) in the clusters	Role	Users granted permissions of this policy must also be granted permissions of the following policies: **Global service project**: OBS Buckets Viewer and OBS Administrator **Region-specific projects**: Tenant Guest, Server Administrator, ELB Administrator, SFS Administrator, SWR Admin, and APM FullAccess **NOTE** Users with both **CCE Administrator** and **NAT Gateway Administrator** policies can use NAT Gateway functions for clusters.
CCE FullAccess	Common operation permissions on CCE cluster resources, excluding the namespace-level permissions for the clusters (with Kubernetes RBAC enabled) and the privileged administrator operations, such as agency configuration and cluster certificate generation	Policy	None.
CCE ReadOnly Access	Permissions to view CCE cluster resources, excluding the namespace-level permissions of the clusters (with Kubernetes RBAC enabled)	Policy	None.

Table 8-2 Common operations supported by CCE system policies

Operation	CCE ReadOnlyAccess	CCE FullAccess	CCE Administrator
Creating a cluster	x	√	√
Deleting a cluster	x	√	√
Updating a cluster, for example, updating cluster node scheduling parameters and providing RBAC support to clusters	x	√	√
Upgrading a cluster	x	√	√
Waking up a cluster	x	√	√
Hibernating a cluster	x	√	√
Listing all clusters	√	√	√
Querying cluster details	√	√	√
Adding a node	x	√	√
Deleting one or more nodes	x	√	√
Updating a cluster node, for example, updating the node name	x	√	√
Querying node details	√	√	√
Listing all nodes	√	√	√
Listing all jobs	√	√	√
Deleting one or more cluster jobs	x	√	√
Querying job details	√	√	√
Creating a storage volume	x	√	√
Deleting a storage volume	x	√	√
Performing operations on all Kubernetes resources	√ (Kubernetes RBAC required)	√ (Kubernetes RBAC required)	√

Operation	CCE ReadOnlyAccess	CCE FullAccess	CCE Administrator
Performing all operations on an Elastic Cloud Server (ECS)	x	√	√
Performing all operations on Elastic Volume Service (EVS) disks EVS disks can be attached to cloud servers and scaled to a higher capacity whenever needed.	x	√	√
Performing all operations on VPC A cluster must run in a VPC. When creating a namespace, you need to create or associate a VPC for the namespace so that all containers in the namespace will run in the VPC.	x	√	√
Viewing details of all resources on an ECS In CCE, a node is an ECS with multiple EVS disks.	√	√	√
Listing all resources on an ECS	√	√	√
Viewing details about all EVS disk resources EVS disks can be attached to cloud servers and scaled to a higher capacity whenever needed.	√	√	√
Listing all EVS resources	√	√	√

Operation	CCE ReadOnlyAccess	CCE FullAccess	CCE Administrator
Viewing details about all VPC resources A cluster must run in a VPC. When creating a namespace, you need to create or associate a VPC for the namespace so that all containers in the namespace will run in the VPC.	√	√	√
Listing all VPC resources	√	√	√
Viewing details about all Elastic Load Balance (ELB) resources	x	x	√
Listing all ELB resources	x	x	√
Viewing Scalable File Service (SFS) resource details	√	√	√
Listing all SFS resources	√	√	√
Viewing Application Operations Management (AOM) resource details	√	√	√
Listing AOM resources	√	√	√
Performing all operations on AOM auto scaling rules	√	√	√

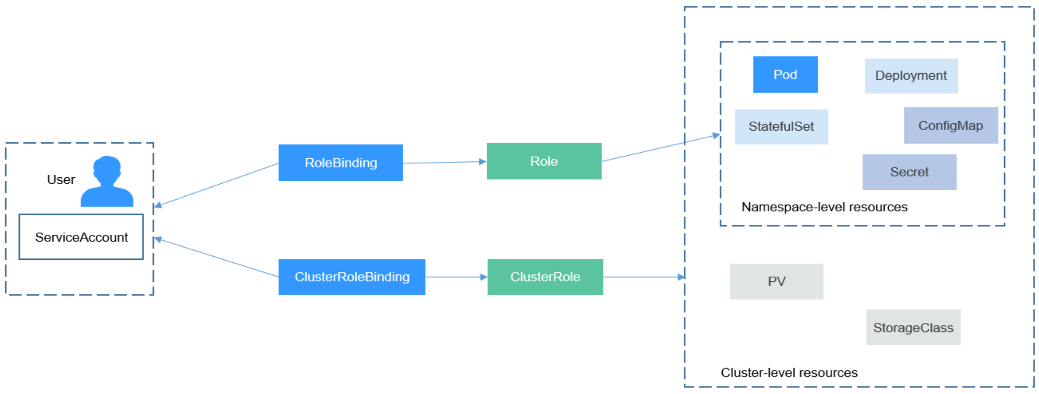
Namespace-level Permissions (Assigned by Using Kubernetes RBAC)

You can regulate users' or user groups' access to Kubernetes resources in a single namespace based on their Kubernetes RBAC roles. The RBAC API declares four kinds of Kubernetes objects: Role, ClusterRole, RoleBinding, and ClusterRoleBinding, which are described as follows:

- Role: defines a set of rules for accessing Kubernetes resources in a namespace.
- RoleBinding: defines the relationship between users and roles.
- ClusterRole: defines a set of rules for accessing Kubernetes resources in a cluster (including all namespaces).
- ClusterRoleBinding: defines the relationship between users and cluster roles.

Role and ClusterRole specify actions that can be performed on specific resources. RoleBinding and ClusterRoleBinding bind roles to specific users, user groups, or ServiceAccounts. See the following figure.

Figure 8-1 Role binding



On the CCE console, you can assign permissions to a user or user group to access resources in one or multiple namespaces. By default, the CCE console provides the following five ClusterRoles:

- view: has the permission to view namespace resources.
- edit: has the permission to modify namespace resources.
- admin: has all permissions on the namespace.
- cluster-admin: has all permissions on the cluster.
- psp-global: controls sensitive security aspects of the pod specification.

In addition to cluster-admin, admin, edit, and view, you can define Roles and RoleBindings to configure the permissions to add, delete, modify, and query resources, such as pods, Deployments, and Services, in the namespace.

Helpful Links

- **IAM Service Overview**
- **Granting Cluster-level Permissions**
- **Permissions Policies and Supported Actions**

9 Basic Concepts

9.1 Basic Concepts

CCE provides highly scalable, high-performance, enterprise-class Kubernetes clusters and supports Docker containers. With CCE, you can easily deploy, manage, and scale containerized applications in the cloud.

The graphical CCE console enables E2E user experiences. In addition, CCE supports native Kubernetes APIs and kubectl. Before using CCE, you are advised to understand related basic concepts.

Cluster

A cluster is a group of one or more cloud servers (also known as nodes) in the same subnet. It has all the cloud resources (including VPCs and compute resources) required for running containers.

Node

A node is a cloud server (virtual or physical machine) running an instance of the Docker Engine. Containers are deployed, run, and managed on nodes. The node agent (kubelet) runs on each node to manage container instances on the node. The number of nodes in a cluster can be scaled.

Node Pool

A node pool contains one node or a group of nodes with identical configuration in a cluster.

Virtual Private Cloud (VPC)

A VPC is a logically isolated virtual network that facilitates secure internal network management and configurations. Resources in the same VPC can communicate with each other, but those in different VPCs cannot communicate with each other by default. VPCs provide the same network functions as physical networks and also advanced network services, such as elastic IP addresses and security groups.

Security Group

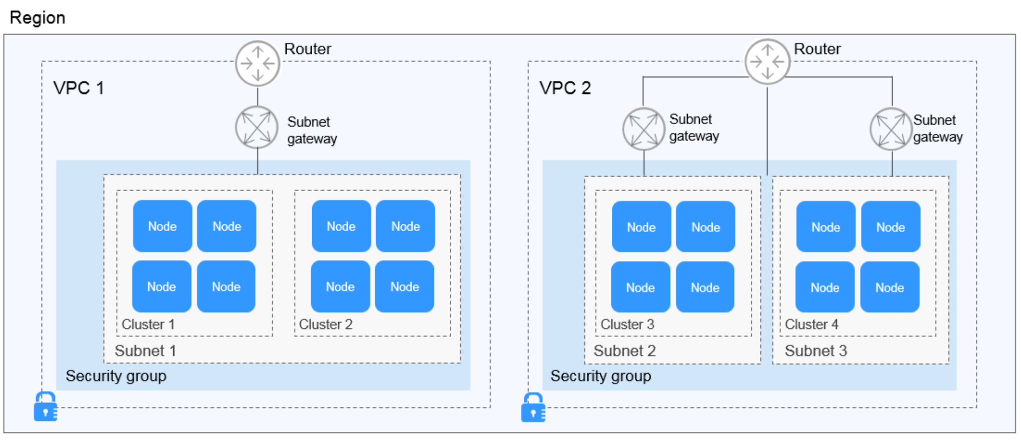
A security group is a collection of access control rules for ECSs that have the same security protection requirements and are mutually trusted in a VPC. After a security group is created, you can create different access rules for the security group to protect the ECSs that are added to this security group.

Relationship Between Clusters, VPCs, Security Groups, and Nodes

As shown in **Figure 9-1**, a region may comprise multiple VPCs. A VPC consists of one or more subnets. The subnets communicate with each other through a subnet gateway. A cluster is created in a subnet. There are three scenarios:

- Different clusters are created in different VPCs.
- Different clusters are created in the same subnet.
- Different clusters are created in different subnets.

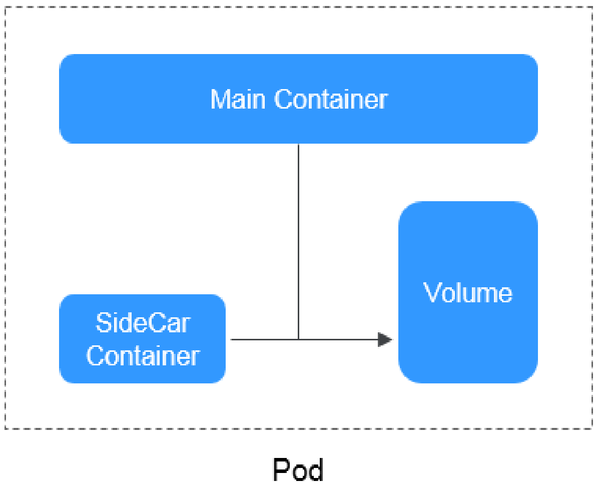
Figure 9-1 Relationship between clusters, VPCs, security groups, and nodes



Pod

A pod is the smallest and simplest unit in the Kubernetes object model that you create or deploy. A pod encapsulates an application container (or, in some cases, multiple containers), storage resources, a unique network IP address, and options that govern how the containers should run.

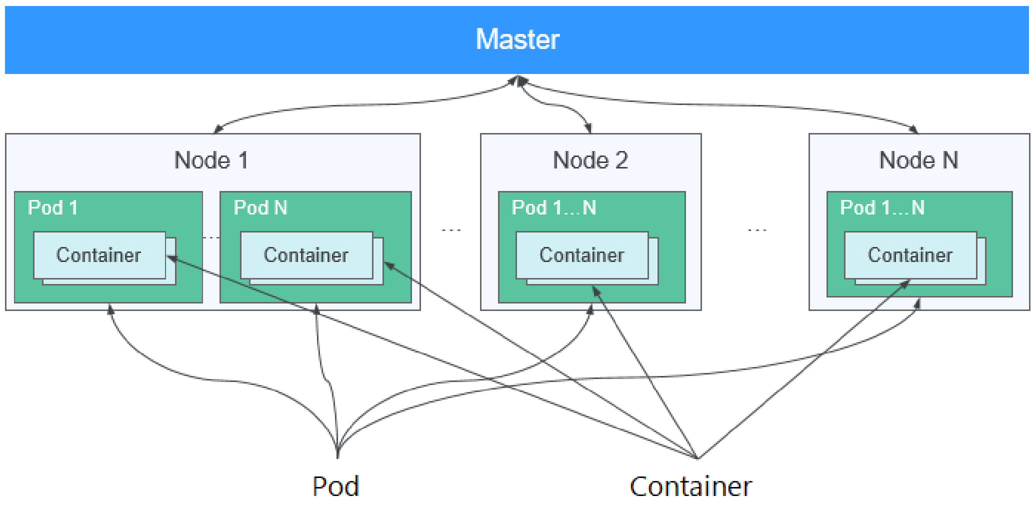
Figure 9-2 Pod



Container

A container is a running instance of a Docker image. Multiple containers can run on one node. Containers are actually software processes. Unlike traditional software processes, containers have separate namespace and do not run directly on a host.

Figure 9-3 Relationships between pods, containers, and nodes

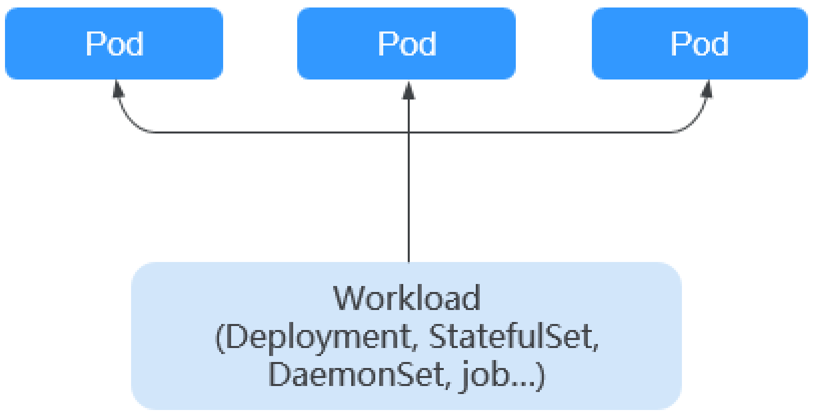


Workload

A workload is an application running on Kubernetes. No matter how many components are there in your workload, you can run it in a group of Kubernetes pods. A workload is an abstract model of a group of pods in Kubernetes. Workloads classified in Kubernetes include Deployments, StatefulSets, DaemonSets, jobs, and cron jobs.

- **Deployment**: Pods are completely independent of each other and functionally identical. They feature auto scaling and rolling upgrade. Typical examples include Nginx and WordPress.
- **StatefulSet**: Pods are not completely independent of each other. They have stable persistent storage, and feature orderly deployment and deletion. Typical examples include MySQL-HA and etcd.
- **DaemonSet**: A DaemonSet ensures that all or some nodes run a pod. It is applicable to pods running on every node. Typical examples include Ceph, Fluentd, and Prometheus Node Exporter.
- **Job**: It is a one-time task that runs to completion. It can be executed immediately after being created. Before creating a workload, you can execute a job to upload an image to the image repository.
- **Cron job**: It runs a job periodically on a given schedule. You can perform time synchronization for all active nodes at a fixed time point.

Figure 9-4 Relationship between workloads and pods

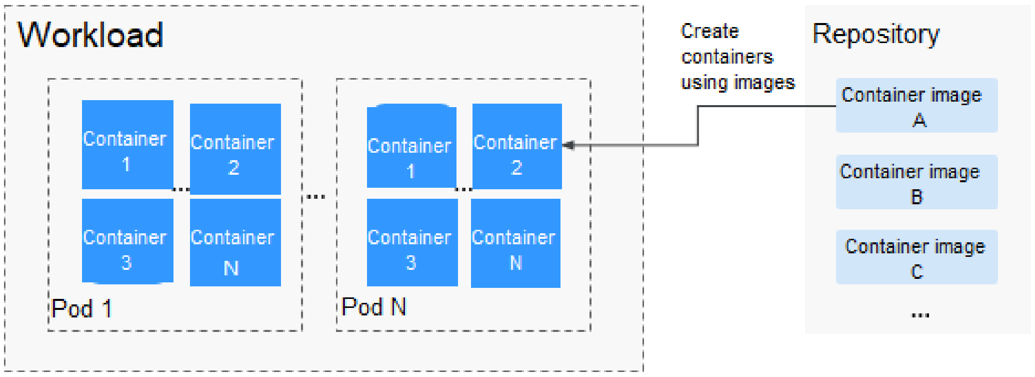


Image

Docker creates an industry standard for packaging containerized applications. Docker images are like templates that include everything needed to run containers, and are used to create Docker containers. In other words, Docker image is a special file system that includes everything needed to run containers: programs, libraries, resources, and configuration files. It also contains configuration parameters (such as anonymous volumes, environment variables, and users) required within a container runtime. An image does not contain any dynamic data. Its content remains unchanged after being built. When deploying containerized applications, you can use images from Docker Hub, SoftWare Repository for Container (SWR), and your private image registries. For example, a Docker image can contain a complete Ubuntu operating system, in which only the required programs and dependencies are installed.

Images become containers at runtime, that is, containers are created from images. Containers can be created, started, stopped, deleted, and suspended.

Figure 9-5 Relationship between images, containers, and workloads



Namespace

A namespace is an abstract collection of resources and objects. It enables resources to be organized into non-overlapping groups. Multiple namespaces can be created inside a cluster and isolated from each other. This enables namespaces to share the same cluster services without affecting each other. Examples:

- You can deploy workloads in a development environment into one namespace, and deploy workloads in a test environment into another namespace.
- Pods, Services, ReplicationControllers, and Deployments belong to a namespace (named **default**, by default), whereas nodes and PersistentVolumes do not belong to any namespace.

Service

A Service is an abstract method that exposes a group of applications running on a pod as network services.

Kubernetes provides you with a service discovery mechanism without modifying applications. In this mechanism, Kubernetes provides pods with their own IP addresses and a single DNS for a group of pods, and balances load between them.

Kubernetes allows you to specify a Service of a required type. The values and actions of different types of Services are as follows:

- **ClusterIP**: ClusterIP Service, as the default Service type, is exposed through the internal IP address of the cluster. If this mode is selected, Services can be accessed only within the cluster.
- **NodePort**: NodePort Services are exposed through the IP address and static port of each node. A ClusterIP Service, to which a NodePort Service will route, is automatically created. By sending a request to <NodeIP>:<NodePort>, you can access a NodePort Service from outside of a cluster.
- **LoadBalancer (ELB)**: LoadBalancer (ELB) Services are exposed by using load balancers of the cloud provider. External load balancers can route to NodePort and ClusterIP Services.
- **DNAT**: A DNAT gateway translates addresses for cluster nodes and allows multiple cluster nodes to share an EIP. DNAT Services provide higher reliability than EIP-based NodePort Services in which the EIP is bound to a single node

and once the node is down, all inbound requests to the workload will be distributed.

Layer-7 Load Balancing (Ingress)

An ingress is a set of routing rules for requests entering a cluster. It provides Services with URLs, load balancing, SSL termination, and HTTP routing for external access to the cluster.

Network Policy

Network policies provide policy-based network control to isolate applications and reduce the attack surface. A network policy uses label selectors to simulate traditional segmented networks and controls traffic between them and traffic from outside.

ConfigMap

A ConfigMap is used to store configuration data or configuration files as key-value pairs. ConfigMaps are similar to secrets, but provide a means of working with strings that do not contain sensitive information.

Secret

Secrets resolve the configuration problem of sensitive data such as passwords, tokens, and keys, and will not expose the sensitive data in images or pod specs. A secret can be used as a volume or an environment variable.

Label

A label is a key-value pair and is associated with an object, for example, a pod. Labels are used to identify special features of objects and are meaningful to users. However, labels have no direct meaning to the kernel system.

Label Selector

Label selector is the core grouping mechanism of Kubernetes. It identifies a group of resource objects with the same characteristics or attributes through the label selector client or user.

Annotation

Annotations are defined in key-value pairs as labels are.

Labels have strict naming rules. They define the metadata of Kubernetes objects and are used by label selectors.

Annotations are additional user-defined information for external tools to search for a resource object.

PersistentVolume

A PersistentVolume (PV) is a network storage in a cluster. Similar to a node, it is also a cluster resource.

PersistentVolumeClaim

A PV is a storage resource, and a PersistentVolumeClaim (PVC) is a request for a PV. PVC is similar to pod. Pods consume node resources, and PVCs consume PV resources. Pods request CPU and memory resources, and PVCs request data volumes of a specific size and access mode.

Auto Scaling - HPA

Horizontal Pod Autoscaling (HPA) is a function that implements horizontal scaling of pods in Kubernetes. The scaling mechanism of ReplicationController can be used to scale your Kubernetes clusters.

Affinity and Anti-Affinity

If an application is not containerized, multiple components of the application may run on the same virtual machine and processes communicate with each other. However, in the case of containerization, software processes are packed into different containers and each container has its own lifecycle. For example, the transaction process is packed into a container while the monitoring/logging process and local storage process are packed into other containers. If closely related container processes run on distant nodes, routing between them will be costly and slow.

- Affinity: Containers are scheduled onto the nearest node. For example, if application A and application B frequently interact with each other, it is necessary to use the affinity feature to keep the two applications as close as possible or even let them run on the same node. In this way, no performance loss will occur due to slow routing.
- Anti-affinity: Instances of the same application spread across different nodes to achieve higher availability. Once a node is down, instances on other nodes are not affected. For example, if an application has multiple replicas, it is necessary to use the anti-affinity feature to deploy the replicas on different nodes. In this way, no single point of failure will occur.

Node Affinity

By selecting labels, you can schedule pods to specific nodes.

Node Anti-Affinity

By selecting labels, you can prevent pods from being scheduled to specific nodes.

Pod Affinity

You can deploy pods onto the same node to reduce consumption of network resources.

Pod Anti-Affinity

You can deploy pods onto different nodes to reduce the impact of system breakdowns. Anti-affinity deployment is also recommended for workloads that may interfere with each other.

Resource Quota

Resource quotas are used to limit the resource usage of users.

Resource Limit (LimitRange)

By default, all containers in Kubernetes have no CPU or memory limit. LimitRange (**limits** for short) is used to add a resource limit to a namespace, including the minimum, maximum, and default amounts of resources. When a pod is created, resources are allocated according to the **limits** parameters.

Environment Variable

An environment variable is a variable whose value can affect the way a running container will behave. A maximum of 30 environment variables can be defined at container creation time. You can modify environment variables even after workloads are deployed, increasing flexibility in workload configuration.

The function of setting environment variables on CCE is the same as that of specifying ENV in a Dockerfile.

Chart

For your Kubernetes clusters, you can use **Helm** to manage software packages, which are called charts. Helm is to Kubernetes what the apt command is to Ubuntu or what the yum command is to CentOS. Helm can quickly search for, download, and install charts.

Charts are a Helm packaging format. It describes only a group of related cluster resource definitions, not a real container image package. A Helm chart contains only a series of YAML files used to deploy Kubernetes applications. You can customize some parameter settings in a Helm chart. When installing a chart, Helm deploys resources in the cluster based on the YAML files defined in the chart. Related container images are not included in the chart but are pulled from the image repository defined in the YAML files.

Application developers need to push container image packages to the image repository, use Helm charts to package dependencies, and preset some key parameters to simplify application deployment.

Helm directly installs applications and their dependencies in the cluster based on the YAML files in a chart. Application users can search for, install, upgrade, roll back, and uninstall applications without defining complex deployment files.

9.2 Cloud Native 2.0 and Huawei Cloud

Huawei is one of the earliest adopters of container technologies. The company started to containerize its products in 2013 and used Kubernetes in 2014. Our experience and full-stack container services can help you migrate and grow in cloud.

Cloud Native 2.0 runs on a standardized, cloud native infrastructure. This is what Huawei Cloud provides for you to run your applications.

Cloud Native 2.0

Cloud Native Development

Cloud native technologies, such as container, microservice, and dynamic orchestration, are booming. They propel service innovations in industries like finance, manufacturing, and Internet. Use cases in more scenarios are on the go, and the industry ecosystem is expanding.

From "On Cloud" to "In Cloud"

Many new applications are now cloud native. Applications, data, and AI collaborate in the cloud throughout their lifecycle. Legacy applications co-exist with new ones.

New Cloud Native Enterprises

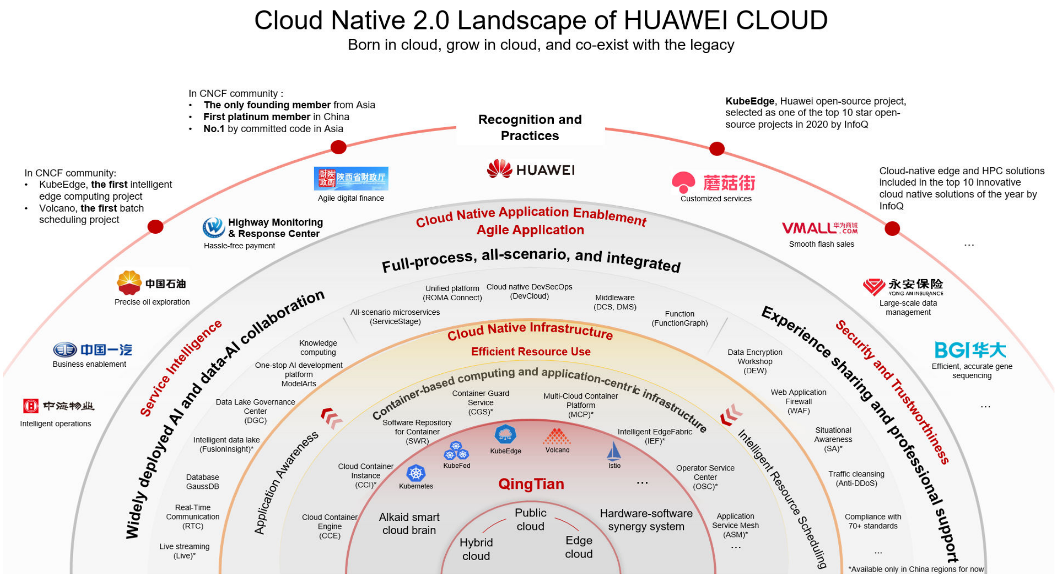
Cloud Native 2.0 is a new phase for intelligent upgrade of enterprises. On Huawei Cloud, you can find the resources you need for this upgrade. Efficient, agile, intelligent, secure, and trustworthy are what they stand for.

Cloud Native Landscape of Huawei Cloud

Huawei Cloud is deploying cloud native in infrastructure services, making them application-centric.

These infrastructure services include Cloud Container Engine (CCE), SoftWare Repository for Container (SWR), Intelligent EdgeFabric (IEF), and Application Orchestration Service (AOS). On top of them run four cloud native solutions. Scenarios covered include bare metal, HPC, hybrid cloud, and edge computing. You can run distributed services on this high-performance infrastructure, backed by a robust cloud native ecosystem.

Figure 9-6 Huawei Cloud offerings for Cloud Native 2.0


Cloud Native 2.0 Landscape of HUAWEI CLOUD
Born in cloud, grow in cloud, and co-exist with the legacy
In CNCF community :
The only founding member from Asia
First platinum member in China
No.1 by committed code in Asia
Recognition and Practices
KubeEdge, Huawei open-source project, selected as one of the top 10 star open-source projects in 2020 by InfoQ
In CNCF community:
KubeEdge, the first intelligent edge computing project
Volcano, the first batch scheduling project
Cloud-native edge and HPC solutions included in the top 10 innovative cloud native solutions of the year by InfoQ
Agile digital finance
HUAWEI
Customized services
Highway Monitoring & Response Center
Hassle-free payment
Smooth flash sales
Precise oil exploration
Large-scale data management
Business enablement
Efficient, accurate gene sequencing
Intelligent operations
...
Cloud Native Application Enablement
Agile Application
Full-process, all-scenario, and integrated
Unified platform (ROMA Connect)
Cloud native DevSecOps (DevCloud)
All-scenario microservices (ServiceStage)
Middleware (DCS, DMS)
Function (FunctionGraph)
Widely deployed AI and data-AI collaboration
Service Intelligence
Knowledge computing
One-stop AI development platform ModelArts
Data Lake Governance Center (DGC)
Intelligent data lake (FusionInsight)*
Database GaussDB
Real-Time Communication (RTC)
Live streaming (Live)*
Experience sharing and professional support
Security and Trustworthiness
Data Encryption Workshop (DEW)
Web Application Firewall (WAF)
Situational Awareness (SA)*
Traffic cleansing (Anti-DDoS)
Compliance with 70+ standards
Cloud Native Infrastructure
Efficient Resource Use
Container-based computing and application-centric infrastructure
Application Awareness
Intelligent Resource Scheduling
Container Guard Service (CGS)*
Multi-Cloud Container Platform (MCP)*
Software Repository for Container (SWR)
Intelligent EdgeFabric (IEF)*
Cloud Container Instance (CCI)*
Operator Service Center (OSC)*
Cloud Container Engine (CCE)
Application Service Mesh (ASM)*
KubeFed
KubeEdge
Volcano
Istio
Kubernetes
QingTian
Alkaid smart cloud brain
Hardware-software synergy system
Public cloud
Hybrid cloud
Edge cloud
*Available only in China regions for now

Cloud Native Infrastructure

Huawei Cloud provides customers with cloud native infrastructure services to help customers redefine their infrastructure, enable ubiquitous applications, and refactor application architecture. With these services, applications can run on a shared base and collaborate with each other across clouds and between clouds and edges to accelerate service innovation.

- **Cloud Container Engine (CCE)** allows you to create highly scalable, high-performance, enterprise-class Kubernetes clusters to run containers. CCE provides full-stack container services, including cluster and application lifecycle management, service mesh, Helm charts, add-ons, and scheduling. With CCE, you can easily deploy, manage, and scale containerized applications on Huawei Cloud.
- **Software Repository for Container (SWR)** hosts container images that can be used to quickly deploy containerized applications. It provides easy, secure, and reliable management over container images throughout their lifecycle.
- **Container Guard Service (CGS)** scans vulnerabilities and configurations in images, helping enterprises detect the container environment, which cannot be found by the traditional security software. CGS also delivers functions such as process whitelist configuration, read-only file protection, and container escape detection to minimize the security risks for a running container.
- **Intelligent EdgeFabric (IEF)** extends cloud applications to edge nodes and associates edge and cloud data, meeting customer requirements for remote control, data processing, analysis, decision-making, and intelligence of edge computing resources. IEF also provides unified on-cloud O&M capabilities, such as device/application monitoring and log collection, to achieve edge-cloud synergy.
- **Multi-Cloud Container Platform (MCP)** is developed on Huawei Cloud container technologies and community-advanced cluster federation. It can centrally manage multiple clusters across clouds and allows uniform deployment and traffic distribution of multi-cluster application. In addition to multi-cloud disaster recovery, MCP can also separate services and data, development and production, and computing and services.

Cloud Native Application Enablement

Huawei Cloud enables customers with full-stack cloud native capabilities to support agile applications, business intelligence, secure and trustworthy services, and continuous evolution.

Agile Application

- **CodeArts** is a one-stop, cloud-based DevOps platform built with Huawei's practices of nearly three decades in R&D, together with its cutting-edge R&D ideas, and state-of-the-art R&D tools. These out-of-the-box cloud services enable you to manage projects, host code, run pipelines, check code, and build, deploy, test, and release your applications in the cloud anytime, anywhere.
- **ServiceStage** is an application and microservice management platform that facilitates application deployment, monitoring, O&M, and governance. ServiceStage provides a full-stack solution for enterprises to develop microservice, mobile, and web applications. This solution helps enterprises easily migrate various applications onto the cloud, allowing enterprises to focus on service innovation for digital transformation.
- **ROMA Connect** is a full-stack application and data integration platform designed for diverse service scenarios. ROMA Connect provides lightweight

message, data, API, device, and model integration for cloud and on-premises applications across regions to simplify enterprise cloudification, helping enterprises achieve digital transformation.

- **Distributed Message Service (DMS) for Kafka** is a message queuing service based on the open-source Apache Kafka. It provides Kafka premium instances with isolated computing, storage, and bandwidth resources. DMS for Kafka allows you to apply resources and configure topics, partitions, and replicas based on service requirements. It can be used out of the box and frees you from deployment and O&M so that you can focus on the agile development of your applications.
- **FunctionGraph** hosts and computes event-driven functions. All you need to do is write your code and set conditions.

Service Intelligence

- **ModelArts** is a one-stop AI development platform. For machine learning and deep learning, it supports data preprocessing, semi-automated data labeling, distributed training, automated model building, and on-demand deployment of device-edge-cloud models. ModelArts can help AI developers build models quickly and manage the lifecycles of AI workflows.
- **GaussDB(for MySQL)** is a next-generation, enterprise-class distributed database service that is fully compatible with MySQL. It uses a decoupled compute-storage architecture and data functions virtualization (DFV) storage that auto-scales up to 128 TB per DB instance. There is no need to deal with sharding and there is virtually no risk of data loss. It combines the high availability and performance of commercial databases with the cost-effectiveness of open-source databases.

Security and Trustworthiness

- **Data Security Center (DSC)** is a next-generation cloud data protection platform that protects your assets with functions such as risk classification, sensitive data identification, watermark source tracing, and static data masking. DSC monitors data security and gives you a comprehensive view of your data security in the cloud.
- **Host Security Service (HSS)** helps you identify and manage the assets on your servers, eliminate risks, and defend against intrusions and web page tampering. There are also advanced protection and security operations functions available to help you easily detect and handle threats.
- **Situation Awareness (SA)** is a security management and situation analysis platform of Huawei Cloud. It can detect over 20 types of risks, including DDoS attacks, brute-force attacks, web attacks, Trojans, zombies, abnormal behavior, vulnerability exploits, and command and control (C&C). SA gives you a comprehensive overview of your global security situation by leveraging the big data analysis technologies, making it easier for you to analyze attack events, threat alarms, and attack sources.
- **Anti-DDoS Service (ADS)** provides multiple security solutions to defend against DDoS attacks. ADS includes CNAD Basic (Anti-DDoS), CNAD Pro, and AAD.

9.3 Mappings Between CCE and Kubernetes Terms

Kubernetes (K8s) is an open-source system for automating deployment, scaling, and management of container clusters. It is a container orchestration tool and a leading solution based on the distributed architecture of the container technology. Kubernetes is built on the open-source Docker technology that automates deployment, resource scheduling, service discovery, and dynamic scaling of containerized applications.

This topic describes the mappings between CCE and Kubernetes terms.

Table 9-1 Mappings between CCE and Kubernetes terms

CCE	Kubernetes
Cluster	Cluster
Node	Node
Node pool	NodePool
Container	Container
Image	Image
Namespace	Namespace
Deployment	Deployment
StatefulSet	StatefulSet
DaemonSet	DaemonSet
Job	Job
Cron job	CronJob
Pod	Pod
Service	Service
ClusterIP	Cluster IP
NodePort	NodePort
LoadBalancer	LoadBalancer
Layer-7 load balancing (ingress)	Ingress
Network policy	NetworkPolicy
Chart	Template
ConfigMap	ConfigMap
Secret	Secret
Label	Label

CCE	Kubernetes
Label selector	LabelSelector
Annotation	Annotation
Volume	PersistentVolume
PersistentVolumeClaim	PersistentVolumeClaim
Auto scaling	HPA
Node affinity	NodeAffinity
Node anti-affinity	NodeAntiAffinity
Pod affinity	PodAffinity
Pod anti-affinity	PodAntiAffinity
Webhook	Webhook
Endpoint	Endpoint
Quota	Resource Quota
Resource limit	Limit Range

9.4 CCE Turbo Cluster

CCE launches its next-gen container cluster, CCE Turbo, for container deployments at scale. Performance, scaling, and scheduling all upgrade to the next level.

CCE Turbo clusters feature accelerated computing, networking, and scheduling. These features further free you from managing service running, and enable easier innovations.

Cluster Advantages

- **Accelerated computing**

  On a software-hardware collaborative infrastructure, computing consumes fewer but performs better.
- **Accelerated networking**

  The Cloud Native Network 2.0 model flattens two network layers into one. Such lossless networking allows applications to better communicate.
- **Accelerated scheduling**

  Application and resource scheduling becomes more intelligent and faster.
- **Security isolation**

  The Kata container engine provides VM-level security isolation for applications.

Comparison Between CCE Turbo Clusters and CCE Clusters

The following table lists the differences between CCE Turbo clusters and CCE clusters:

Table 9-2 Cluster types

Dimension	Sub-dimension	CCE Turbo Cluster	CCE Cluster
Cluster	Positioning	Next-gen container cluster, with accelerated computing, networking, and scheduling. Designed for Cloud Native 2.0	Standard cluster for common commercial use
	Node type	Hybrid deployment of VMs and bare-metal servers	Hybrid deployment of VMs and bare-metal servers
Networking	Model	**Cloud Native Network 2.0**: applies to large-scale and high-performance scenarios. Max networking scale: 2,000 nodes	**Cloud-native network 1.0**: applies to common, smaller-scale scenarios. ● Tunnel network model ● VPC network model
	Performance	Flattens the VPC network and container network into one. No performance loss.	Overlays the VPC network with the container network, causing certain performance loss.
	Container network isolation	Associates pods with security groups. Unifies security isolation in and out the cluster via security groups' network policies.	● Tunnel network model: supports network policies for intra-cluster communications. ● VPC network model: supports no isolation.
Security	Isolation	● Physical machine: runs Kata containers, allowing VM-level isolation. ● VM: runs common containers.	Runs common containers, isolated by cgroups.

How to Buy

Learn how to **buy and use a CCE Turbo cluster**.

9.5 Regions and AZs

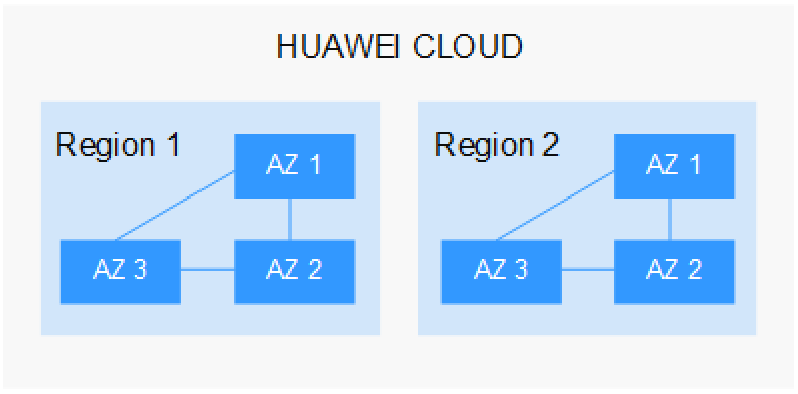
Definition

A region and availability zone (AZ) identify the location of a data center. You can create resources in a specific region and AZ.

- Regions are divided based on geographical location and network latency. Public services, such as Elastic Cloud Server (ECS), Elastic Volume Service (EVS), Object Storage Service (OBS), Virtual Private Cloud (VPC), Elastic IP (EIP), and Image Management Service (IMS), are shared within the same region. Regions are classified as universal regions and dedicated regions. A universal region provides universal cloud services for common domains. A dedicated region provides services of the same type only or for specific domains.
- An AZ contains one or more physical data centers. Each AZ has independent cooling, fire extinguishing, moisture-proof, and electricity facilities. Within an AZ, computing, network, storage, and other resources are logically divided into multiple clusters. AZs in a region are interconnected through high-speed optic fibers. This is helpful if you will deploy systems across AZs to achieve higher availability.

Figure 9-7 shows the relationship between the region and AZ.

Figure 9-7 Regions and AZs



Huawei Cloud provides services in many regions around the world. You can select a region and AZ as needed. For more information, see **Global Products and Services**.

How to Select a Region?

When selecting a region, consider the following factors:

- Location

  Select a region close to you or your target users to reduce network latency and improve access rate. Chinese mainland regions provide basically the same infrastructure, BGP network quality, as well as operations and configurations on resources. If you or your target users are in the Chinese mainland, you do not need to consider the network latency differences when selecting a region.

- If you or your target users are in the Asia Pacific region, except the Chinese mainland, select the **CN-Hong Kong**, **AP-Bangkok**, or **AP-Singapore** region.
- If you or your target users are in South Africa, select the **AF-Johannesburg** region.
- If you or your target users are in Europe, select the **EU-Paris** region.
- If you or your target users are in Latin America, select the **LA-Santiago** region.

NOTE

The **LA-Santiago** region is located in Chile.

- Resource price

  Resource prices may vary in different regions. For details, see **Product Pricing Details**.

Selecting an AZ

When deploying resources, consider your applications' requirements on disaster recovery (DR) and network latency.

- For high DR capability, deploy resources in different AZs within the same region.
- For lower network latency, deploy resources in the same AZ.

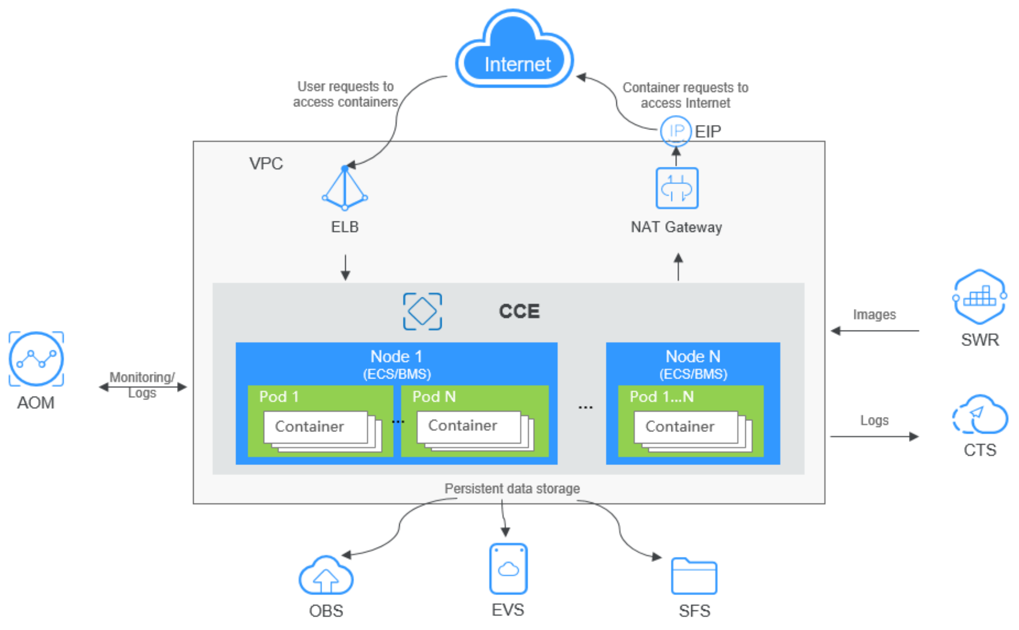
Regions and Endpoints

When using an API to access resources, you must specify a region and endpoint. For details, see **Regions and Endpoints**.

10 Related Services

CCE works with the following cloud services and requires permissions to access them.

Figure 10-1 Relationships between CCE and other services



Relationships Between CCE and Other Services

Table 10-1 Relationships between CCE and other services

Service	Relationship	Related Features
Elastic Cloud Server (ECS)	An ECS with multiple EVS disks is a node in CCE. You can choose ECS specifications during node creation.	• **Buying a Node** • **Accepting Existing Nodes into a Cluster**

Service	Relationship	Related Features
Virtual Private Cloud (VPC)	For security reasons, all clusters created by CCE must run in **VPCs**. When creating a namespace, you need to create a VPC or bind an existing VPC to the namespace so all containers in the namespace will run in this VPC.	**Buying a CCE Cluster**
Elastic Load Balance (ELB)	CCE works with ELB to load balance a workload's access requests across multiple pods. When **ELB** is used, the load balancer's address, instead of the workload address, is exposed to users. User requests first arrive at ELB via a public network and then routed by ELB to different pods of the workload.	● **Creating a Deployment** ● **Creating a StatefulSet** ● **LoadBalancer**
NAT Gateway	The NAT Gateway service provides source network address translation (SNAT) for container instances in a VPC. The SNAT feature translates private IP addresses of these container instances to the same EIP, which is a public IP address reachable on Internet. You can define SNAT rules on the **NAT gateway** to let containers access the Internet.	● **Creating a Deployment** ● **Creating a StatefulSet** ● **DNAT**
Software Repository for Container (SWR)	An image repository is used to store and manage Docker images. You can create workloads from images in **SWR**.	● **Creating a Deployment** ● **Creating a StatefulSet**
Elastic Volume Service (EVS)	EVS disks can be attached to cloud servers and scaled to a higher capacity whenever needed. An ECS with multiple EVS disks is a node in CCE. You can choose ECS specifications during node creation.	**Using EVS Volumes**

Service	Relationship	Related Features
Object Storage Service (OBS)	OBS provides stable, secure, cost-efficient, and object-based cloud storage for data of any size. With OBS, you can create, modify, and delete buckets, as well as uploading, downloading, and deleting objects. CCE allows you to create an OBS volume and attach it to a path inside a container.	**Using OBS Volumes**
Scalable File Service (SFS)	SFS is a shared, fully managed file storage service. Compatible with the Network File System protocol, SFS file systems can elastically scale up to petabytes, thereby ensuring top performance of data-intensive and bandwidth-intensive applications. You can use SFS file systems as persistent storage for containers and attach the file systems to containers when creating a workload.	**Using SFS Volumes**
Application Operations Management (AOM)	AOM collects container log files in formats like .log from CCE and dumps them to AOM. On the AOM console, you can easily query and view log files. In addition, AOM monitors CCE resource usage. You can define metric thresholds for CCE resource usage to trigger auto scaling.	**Collecting Standard Output Logs of Containers**
Cloud Trace Service (CTS)	CTS records operations on your cloud resources, allowing you to query, audit, and backtrack resource operation requests initiated from the management console or open APIs as well as responses to these requests.	**CCE Operations Supported by CTS**