Cloud Container Engine

Service Overview

Issue	01
Date	2024-11-08

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions

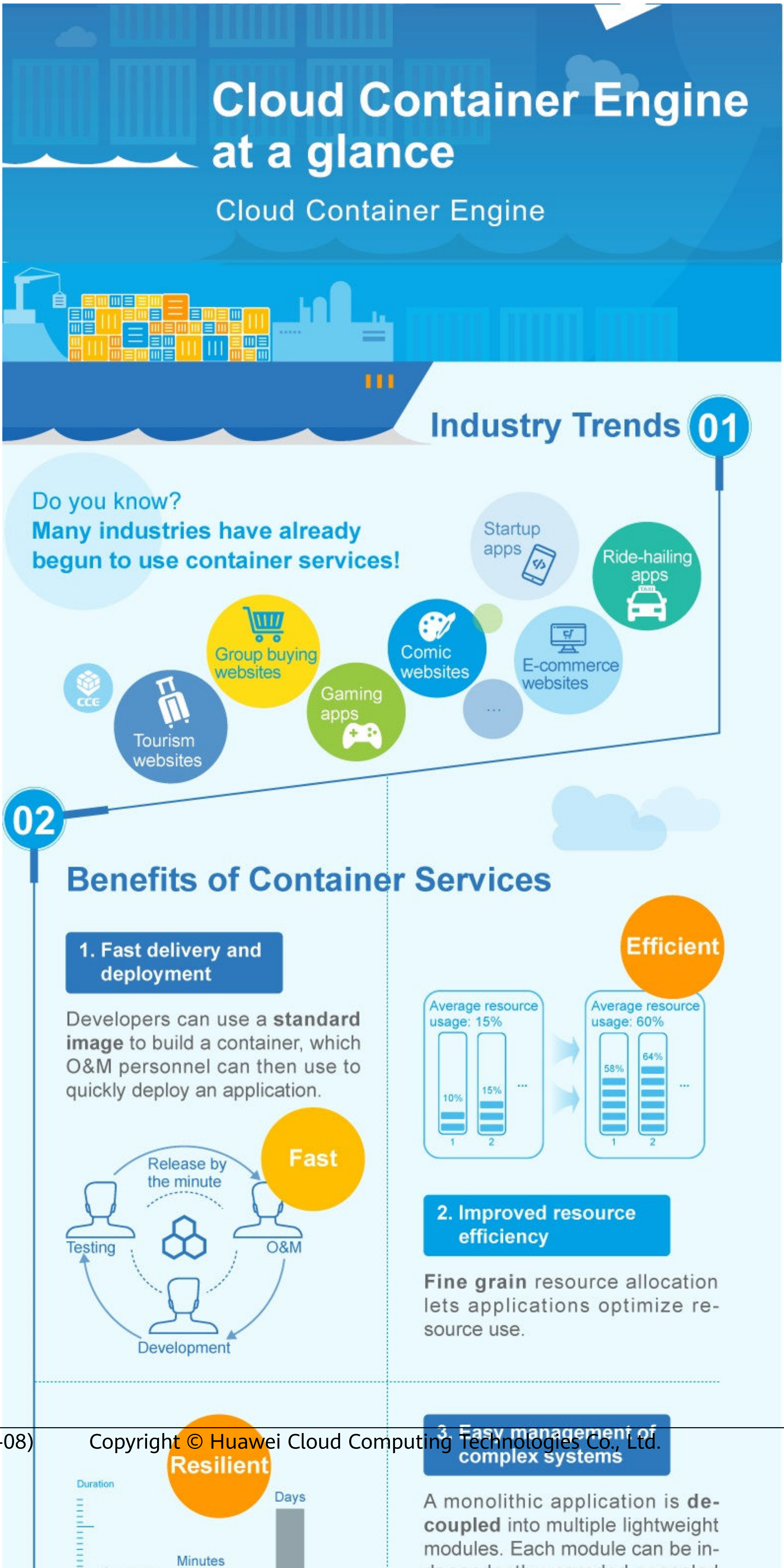
Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

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 CCE?

Cloud Container Engine (CCE) is a **Kubernetes** cluster hosting service for enterprises. It manages the enter lifecycle of containerized applications and delivers scalable, high-performance solutions for deploying and managing cloud native applications.

Why CCE?

CCE is a one-stop platform integrating compute (ECS or BMS), networking (VPC, EIP, and ELB), storage (EVS, OBS, and SFS), and many other services. It supports heterogeneous compute architectures such as GPUs, NPUs, and Arm. Multi-AZ, multi-region disaster recovery (DR) ensures high availability (HA) of **Kubernetes** clusters.

Huawei Cloud is one of world's first Kubernetes Certified Service Providers (KCSPs) and China's first participant in the Kubernetes community. It has long been contributing to open-source container communities and taking lead in the container ecosystems. Huawei Cloud 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**.

CCE Cluster Types

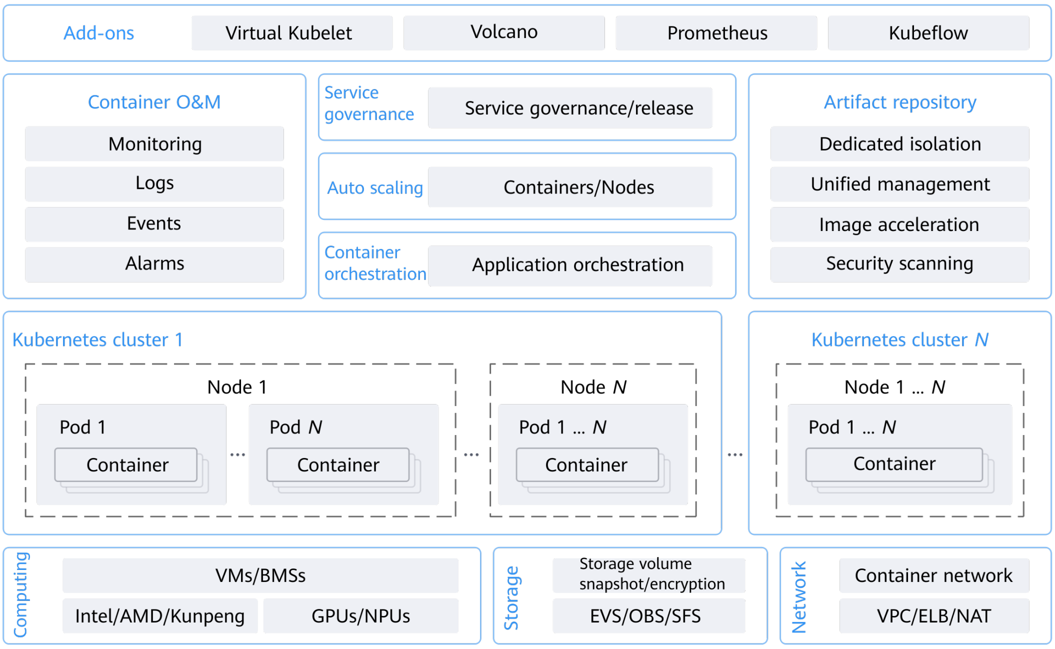
There are multiple types of CCE clusters.

Category	Subcategory	CCE Standard	CCE Turbo	CCE Autopilot
Positioning	-	Standard clusters that provide highly reliable and secure containers for commercial use	Next-generation clusters designed for Cloud Native 2.0, with accelerated compute, networking, and scheduling	Serverless clusters without user nodes and billed by actual CPU and memory usage In such clusters, no node deployment, management, or security maintenance is needed.
Application scenario	-	For users who expect to use container clusters to manage applications, obtain elastic compute resources, and enable simplified management on compute, network, and storage resources	For users who have higher requirements on performance, resource utilization, and full-scenario coverage	For users whose services suffer frequent traffic surges, such as users in the online education and e-commerce sectors
Specification difference	Network model	Cloud native 1.0 networks: for scenarios where requirements on performance are not high and there are not so many containers ● Tunnel network ● Virtual Private Cloud (VPC) network	Cloud Native 2.0 networks: for scenarios where requirements on performance are high and there are many containers Max networking scale: 2,000 nodes	Cloud Native 2.0 networks: for scenarios where requirements on performance are high and there are many containers
	Network performance	The container network is overlaid with the VPC network, causing certain performance loss.	The VPC network and container network are flattened into one for zero performance loss.	The VPC network and container network are flattened into one for zero performance loss.

Category	Subcategory	CCE Standard	CCE Turbo	CCE Autopilot
	Network isolation	- Tunnel network model: network policies supported for communications within a cluster - VPC network model: isolation not supported	Pods can be associated with security groups for isolation. This isolation policy, based on security groups, ensures consistent security isolation both within and outside of a cluster.	Pods can be associated with security groups for isolation. This isolation policy, based on security groups, ensures consistent security isolation both within and outside of a cluster.
	Security isolation	cgroups are used to isolate common containers.	- VM-level isolation is supported for secure containers that run only on physical machines. - cgroups are used to isolate common containers.	VM-level isolation
	Edge infrastructure management	Not supported	Management of CloudPond edge sites	Not supported

CCE Cluster Architecture

Figure 2-1 CCE architecture



- Compute: VMs and BMSs can be deployed in the same CCE cluster. CCE adapts to various Huawei Cloud compute instances like Kunpeng instances, and CCE supports GPUs and Ascend compute. CCE provides GPU virtualization, shared scheduling, and resource-aware scheduling optimization.
- Networking: CCE is integrated with high-performance, secure, reliable, multi-protocol dedicated load balancers as the service traffic ingress.
- Storage: CCE is integrated with multiple storage services like EVS, SFS, and OBS and provides disk encryption, snapshot, and backup capabilities.
- Cluster service: CCE helps you manage the entire lifecycle of your clusters, including cluster buying, access, upgrade, and management.
- Container orchestration: CCE provides a console for managing Helm charts, helping you easily deploy applications using the charts and manage applications on the console.
- Artifact repository: CCE works with SoftWare Repository for Container (SWR) that supports full lifecycle management of images. SWR is an easy-to-use, secure, reliable image management system. With SWR, you can quickly deploy containerized applications on CCE.
- Auto scaling: CCE supports auto scaling of workloads and nodes, allowing you to adjust compute resources based on service requirements and policies in a cost-effective manner.
- Service governance: CCE integrates with Application Service Mesh (ASM). Grayscale release, traffic governance and monitoring, all done in a non-intrusive manner.
- Container O&M: CCE integrates with Container Intelligent Analysis (CIA) so that you can monitor applications and resources in real time, collect, manage, and analyze logs, collect metrics and events of your applications on CCE. You are allowed to enable the monitoring functions in just a few clicks.

- Add-ons: There are multiple types of add-ons available on CCE. With these add-ons, you can extend the cluster functions as needed.

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 developed on Docker and Kubernetes. It offers a wide range of features that allow you to run containers on a large scale. CCE containers are highly reliable, have high-performance, and compatible with open-source communities, making them an ideal choice for enterprise needs.

Easy to Use

- Creating CCE clusters (CCE standard, CCE Turbo, and CCE Autopilot) is a breeze with just a few clicks on the web page. You can deploy VMs and BMSs in a cluster.
- 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 seamlessly integrates with Helm charts, ASM, and add-ons to provide an out-of-the-box user experience.

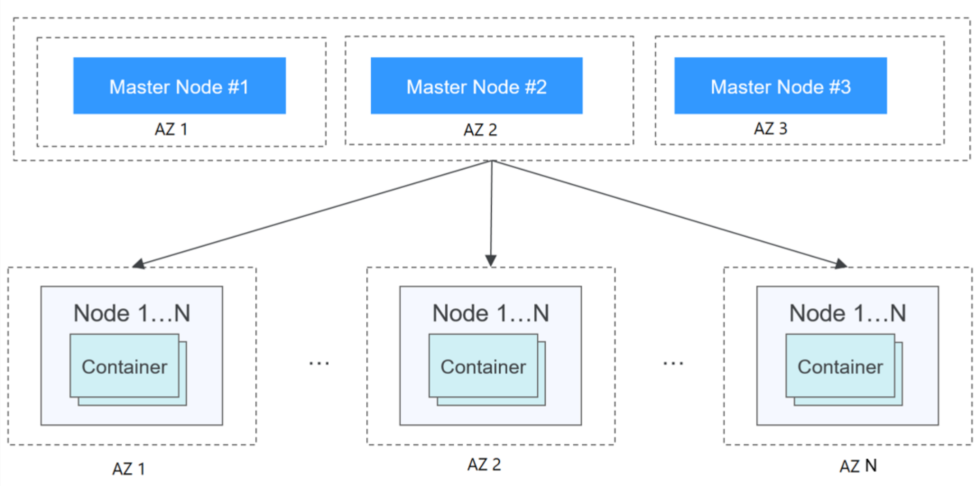
High Performance

- CCE draws on years of field experience in compute, networking, storage, and heterogeneous infrastructure and provides you high-performance cluster services. You can concurrently 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 Achieving cluster HA



- 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.

  CCE offers secure containers so that each pod runs on a separate micro-VM, has its own OS kernel, and is securely isolated at the virtualization layer.

Open and Compatible

- CCE uses Docker to simplify the management of containerized applications by providing features such as deployment, resource scheduling, networking, and auto 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 Kubernetes Cluster	CCE Cluster
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 and update a Kubernetes container cluster in a few clicks. There is 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.
Scalability	You have to assess service loads and cluster health before resizing a Kubernetes cluster.	**Managed scaling service** CCE auto scales clusters and workloads according to resource metrics and scaling policies.
Reliability	There might be security vulnerabilities or configuration errors may occur in the OS of an on-premises Kubernetes cluster, which may cause security issues such as unauthorized access and data leakage.	**Enterprise-class security and reliability** CCE offers container-optimized OS images that have undergone additional stability tests and security hardening. These images are based on native Kubernetes clusters and runtime versions, resulting in reduced management costs and risks, as well as improved reliability and security for applications.
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.

Area of Focus	On-Premises Kubernetes Cluster	CCE Cluster
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.

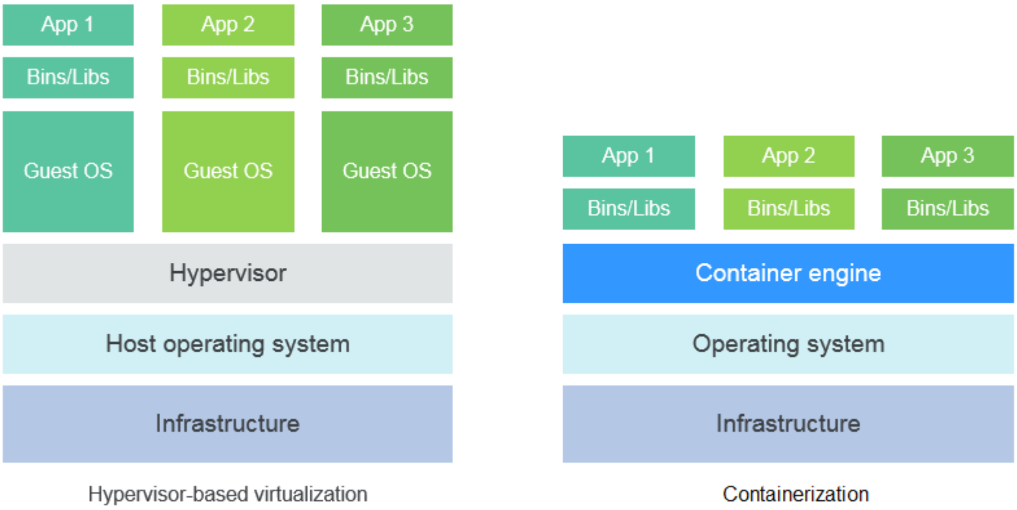
Why Containerization?

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. The isolated software processes, which are called 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 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

A Docker image is built up from a series of layers and these layers are stacked. When you create a new container, you add a container layer on top of image layers. In this way, duplicate layers are reused, which simplify application maintenance and update as well as further image extension on base images. 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

Feature	Containers	VMs
Performance	Near-native performance	Weak
Per-machine capacity	Thousands of containers	Tens of VMs

4 Application Scenarios

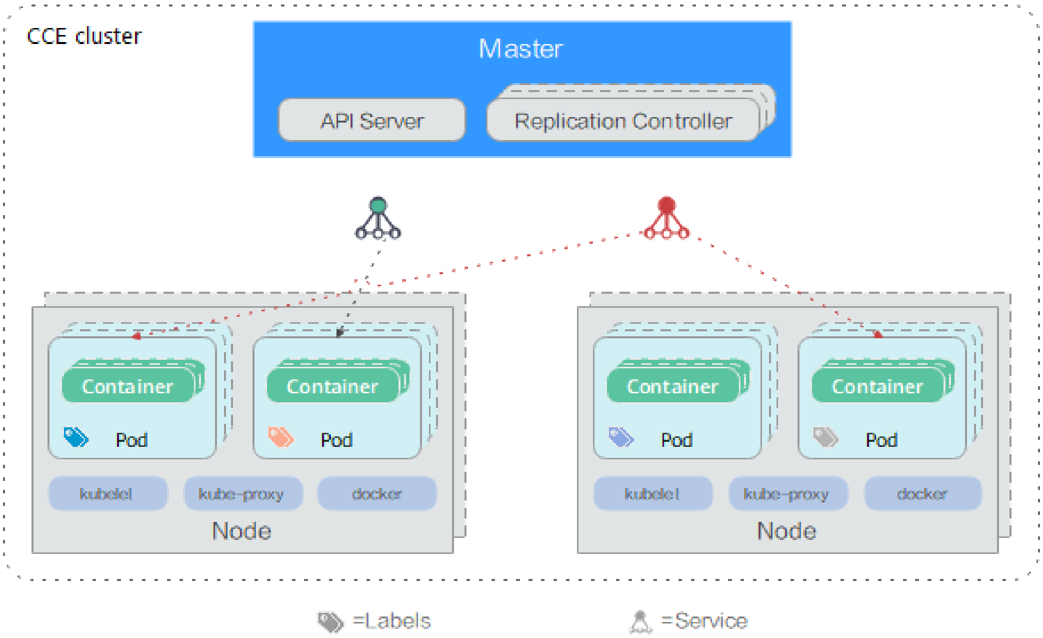
4.1 Containerized Application Management

Application Scenarios

CCE clusters enable the management of both x86 and Arm resources. With CCE, you can effortlessly create Kubernetes clusters, deploy containerized applications, and effectively manage and maintain them.

- Containerized web applications: CCE clusters interconnect with Huawei Cloud middleware such as GaussDB and Redis and support HA DR, auto scaling, public network release, and gray upgrade, helping you quickly deploy web service applications.
- Middleware deployment platform: CCE clusters can be used as middleware deployment platforms to implement stateful applications with StatefulSets and PVCs. In addition, load balancers can be used to expose middleware services.
- Jobs and cron jobs: Job and cron job applications can be containerized to reduce the dependency on the host system. Global resource scheduling secures the resource usage during task running and improves the overall resource usage in the cluster.

Figure 4-1 CCE cluster



Benefits

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

Advantages

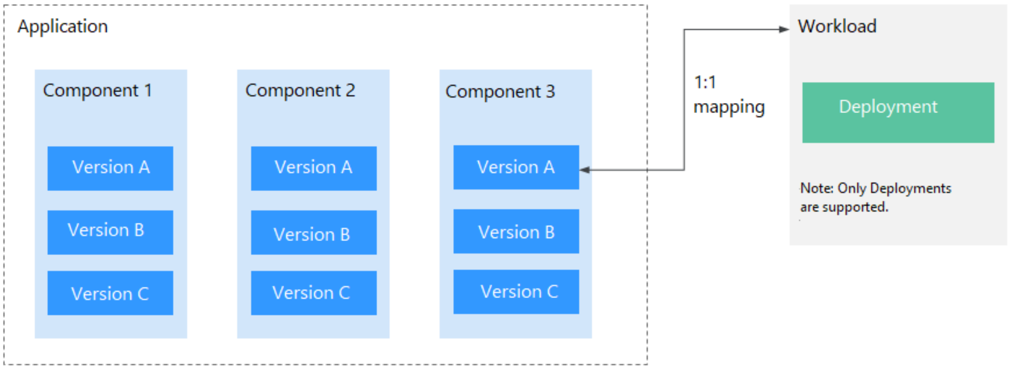
- Multiple types of workloads

  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

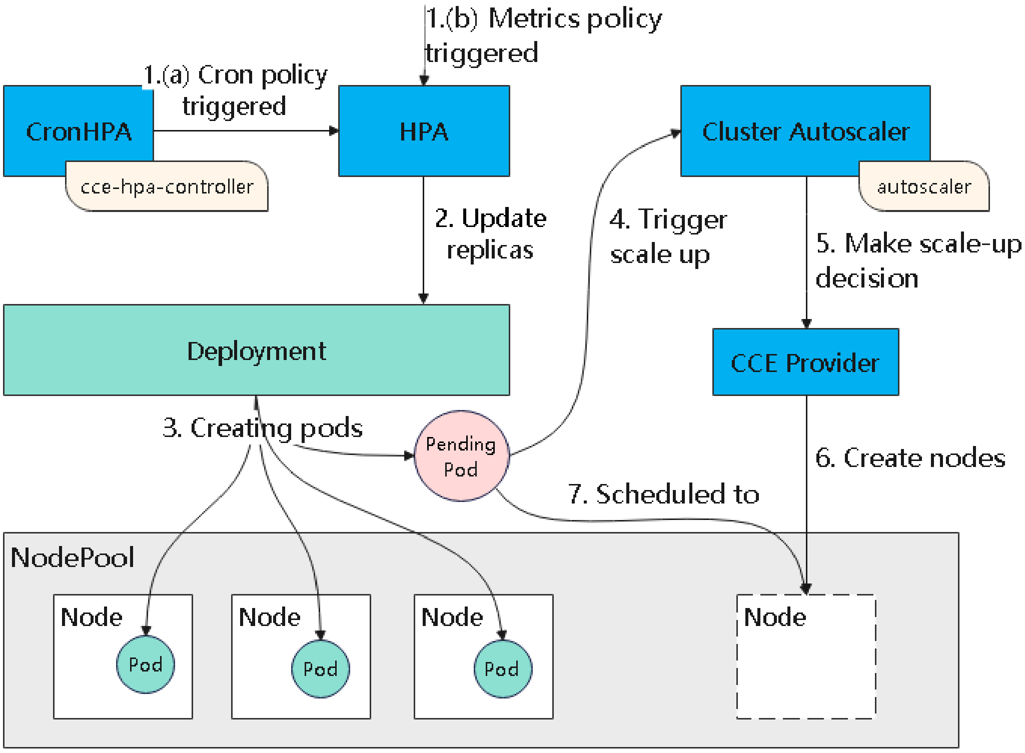
  Bills you only for the scaled cloud servers as you use.

Related Services

Add-ons: autoscaler and cce-hpa-controller

- Auto scaling for workloads: CronHPA (CronHorizontalPodAutoscaler) + HPA (Horizontal Pod Autoscaling)
- Auto scaling for clusters: CA (Cluster AutoScaling)

Figure 4-3 How auto scaling works



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.

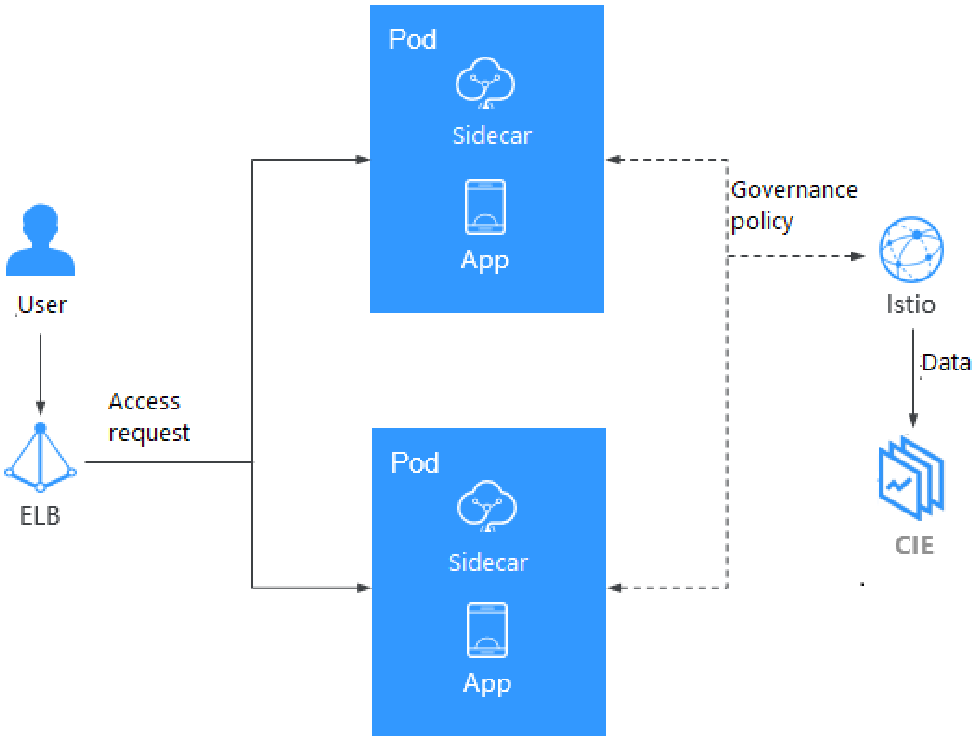
- Visibility into traffic

  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

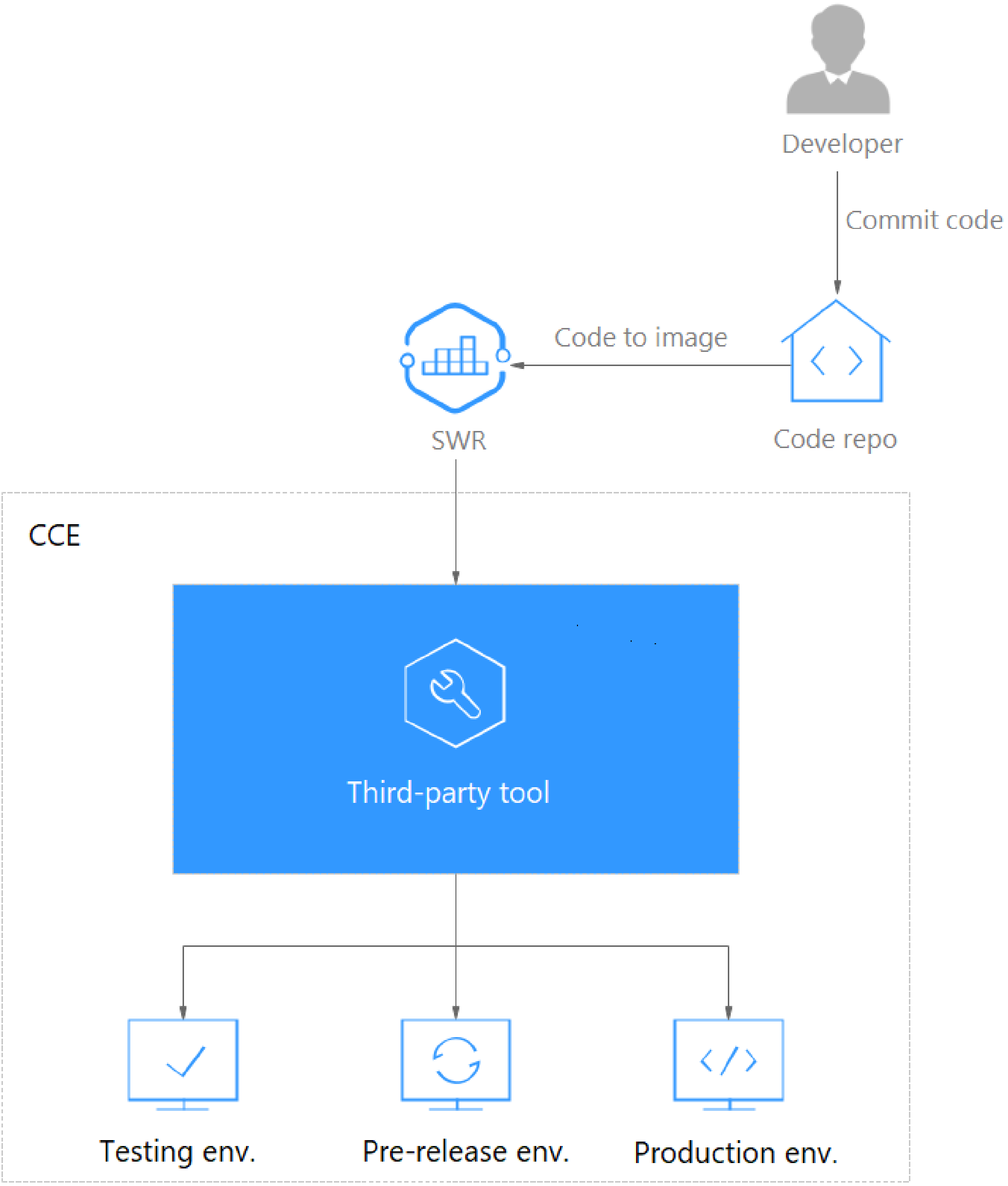
  Provides various APIs to integrate with existing CI/CD systems for in-depth customization.
- High performance

  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

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 local 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

Thanks to containers' environment-independent feature, CCE manages containers that run in private and public clouds in a unified manner, and these containers can access each other. You can seamlessly migrate your apps and data on and off the cloud. This meets complex services' requirements for auto scaling, flexibility, security, and compliance. Additionally, resources in different cloud environments can be operated and maintained in a unified manner, providing easier resource scheduling and DR.

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.
- Unified architecture and auto scaling

  Unified architecture on and off the cloud can flexibly implement auto scaling, smooth migration to cope with traffic peaks.
- Decoupling and sharing

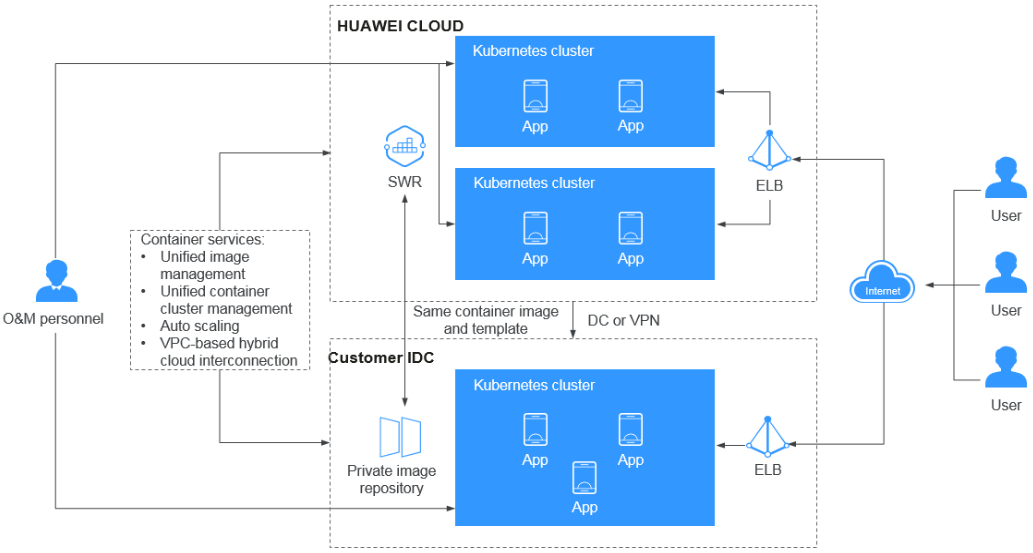
  CCE decouples data, environments, and compute capacity. Sensitive data vs general data. Development vs production. Compute-intensive services vs general services. Apps running on-premises can burst to the cloud. Your resources on and off the cloud can be better used.
- Lower costs

  Cloud resource pools can quickly be scaled out to handle traffic bursts. Manual operations are no longer needed and you can save big.

Related Services

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

Figure 4-6 How hybrid cloud works



4.6 High-Performance Scheduling

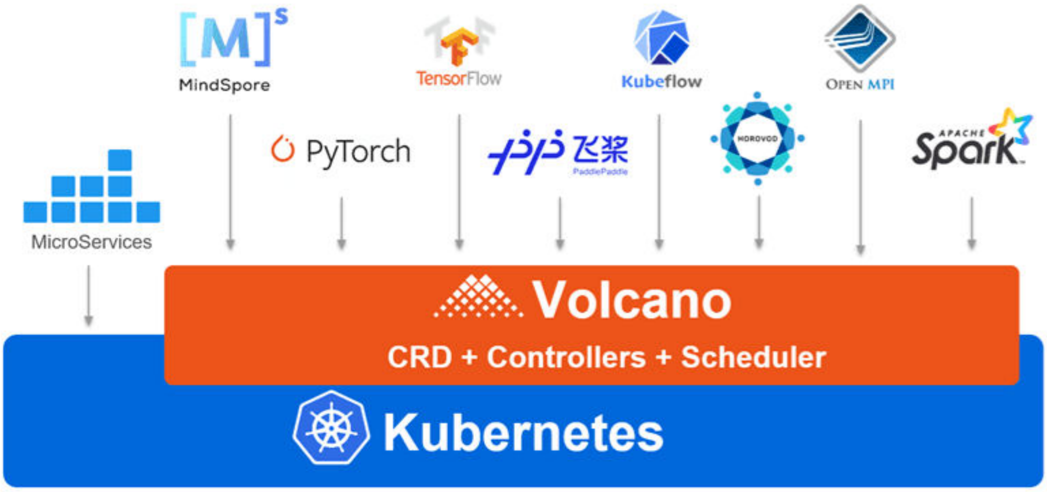
CCE integrates Volcano to support high-performance computing.

Volcano is a Kubernetes-native batch processing system. Volcano is a versatile, scalable, reliable platform for running big data and AI jobs. It supports a wide range of computing frameworks, including those for AI, big data, gene sequencing, and rendering tasks. With its advanced task scheduling and heterogeneous chip management capabilities, Volcano streamlines task running and management, resulting in greater efficiency.

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.

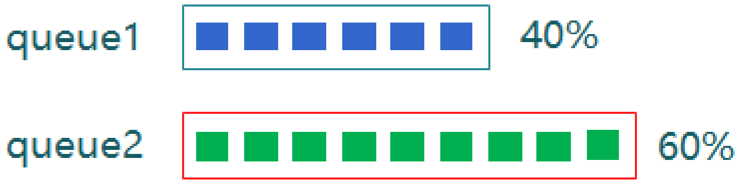


Application Scenario 2: Scheduling Optimization in Multi-Queue Scenarios

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.

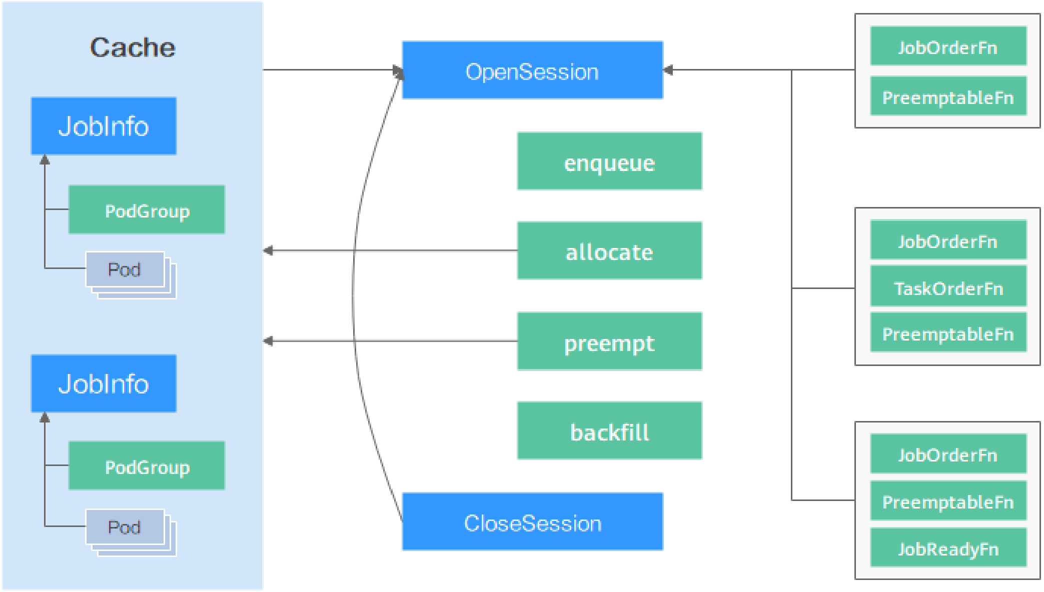
For example, in the following figure, queue 1 is allocated 40% of the cluster resources, and 60% for queue 2. In this way, two queues can be mapped to different departments or projects to use resources in the same cluster. If a queue has idle resources, they can be allocated to jobs in another queue.



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, 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 custom 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.

Application Scenario 5: Hybrid Deployment of Online and Offline Jobs

Many services see surges in traffic. To ensure performance and stability, resources are often requested at the maximum needed. However, the surges may ebb very shortly and resources, if not released, are wasted in non-peak hours. Especially for online jobs that request a large quantity of resources to ensure SLA, resource utilization can be as low as it gets. Resource oversubscription is the process of making use of idle requested resources. Oversubscribed resources are suitable for deploying offline jobs, which focus on throughput but have low SLA requirements and can tolerate certain failures. Hybrid deployment of online and offline jobs in a cluster can better utilize cluster resources.

The default scheduler of Kubernetes schedules resources by pod, regardless of the type of services running in pods. This mechanism fails in hybrid deployment. Here is where Volcano comes in. Volcano provides intelligent algorithms to schedule jobs by type based on how many resources are required and available. Idle cluster resources can be further used with mechanisms such as resource preemption and time division multiplexing.

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

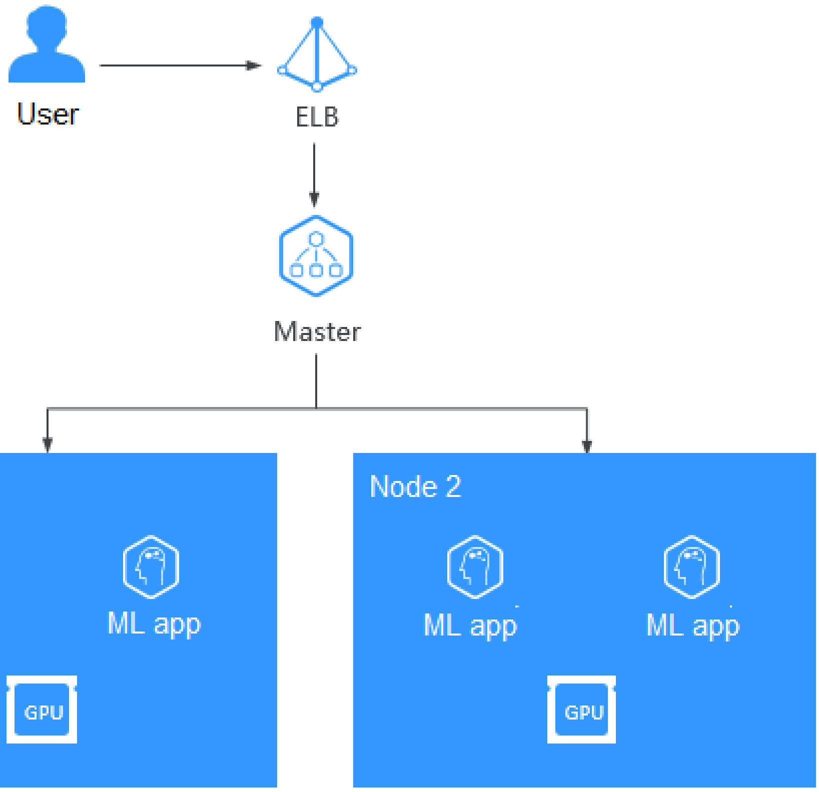
By integrating Volcano, CCE has the following advantages in running high-performance computing, big data, and AI jobs:

- **Hybrid deployment** of HPC, big data, and AI jobs
- **Optimized multi-queue scheduling**: Queues can be effectively managed for scheduling jobs. Complex job scheduling capabilities such as queue priority and multi-level queues are supported.
- **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.
- **Hybrid deployment of online and offline jobs**: Hybrid deployment is supported, and CPU and memory resources can be oversubscribed to better utilize cluster 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

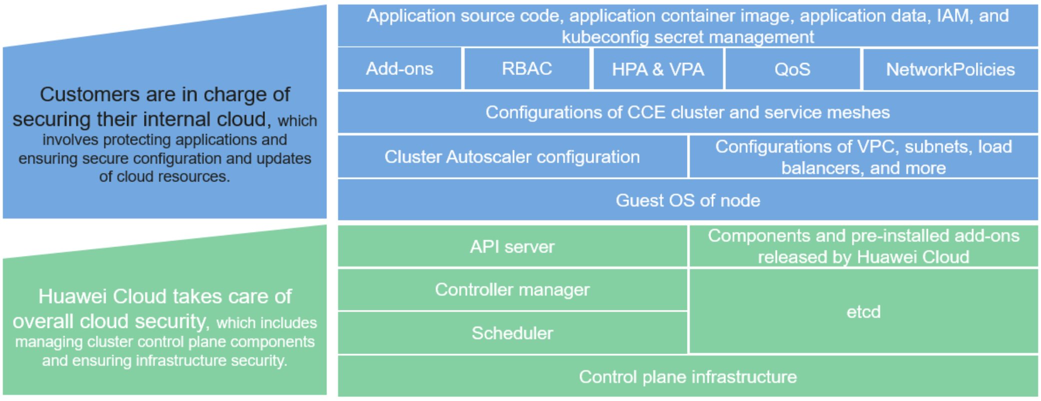
Huawei guarantees that its commitment to cyber security will never be outweighed by the consideration of commercial interests. Huawei Cloud has developed a comprehensive cloud service security assurance system to address the growing challenges and threats of cloud security. This system is tailored to different regions and industries and leverages Huawei's unique software and hardware advantages, as well as compliance with laws, regulations, industry standards, and security ecosystem.

Figure 5-1 illustrates the responsibilities shared by Huawei Cloud and CCE 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 and 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 outlines the strategies and measures used to establish a secure cloud. It covers a range of topics, including the cloud security strategies, the shared responsibility model, compliance and privacy, security personnel and organizations, infrastructure security, tenant service and security, engineering security, O&M security, and ecosystem security.

Figure 5-1 Shared security responsibility model



Customers are in charge of securing their internal cloud, which involves protecting applications and ensuring secure configuration and updates of cloud resources.

Huawei Cloud takes care of overall cloud security, which includes managing cluster control plane components and ensuring infrastructure security.

Application source code, application container image, application data, IAM, and kubeconfig secret management
Add-ons
RBAC
HPA & VPA
QoS
NetworkPolicies
Configurations of CCE cluster and service meshes
Cluster Autoscaler configuration
Configurations of VPC, subnets, load balancers, and more
Guest OS of node
API server
Components and pre-installed add-ons released by Huawei Cloud
Controller manager
Scheduler
etcd
Control plane infrastructure

5.2 Data Protection

CCE takes different measures to keep data secure and reliable.

Table 5-1 CCE data protection methods 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**
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**

Measure	Description	Documentation
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**

5.3 Audit and Logging

Audit

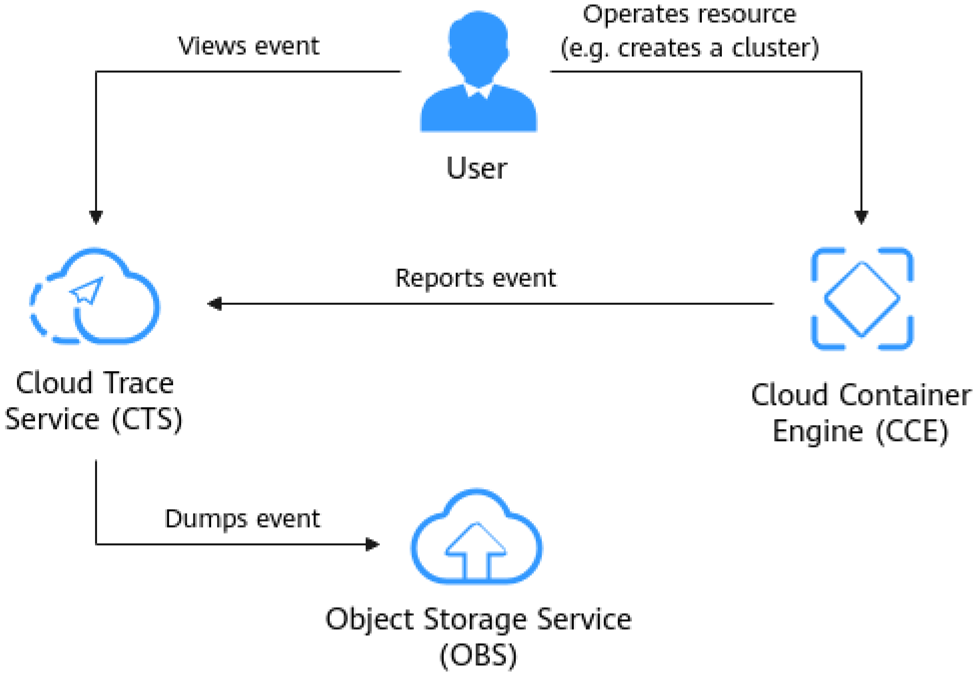
Cloud Trace Service (CTS) records operations on the cloud resources in your account. You can use the logs generated by CTS to perform security analysis, track resource changes, audit compliance, and locate faults.

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



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**.

5.4 Security Risk Monitoring

CCE collaborates with AOM to monitor Kubernetes native containers. This allows you to monitor your applications and resources in real-time, collecting metrics and events to analyze application health. You can easily view multi-dimensional data in a comprehensive and visually clear manner. Moreover, you can collect and monitor custom metrics of workloads based on your requirements, enabling you to implement personalized monitoring policies.

- System metrics

  Basic monitoring data includes CPU, memory, and disk metrics. You can monitor the health and loads of a cluster. For details, see **Overview**. You can view these metrics of clusters, nodes, and workloads on the CCE or AOM console.

- Custom metrics

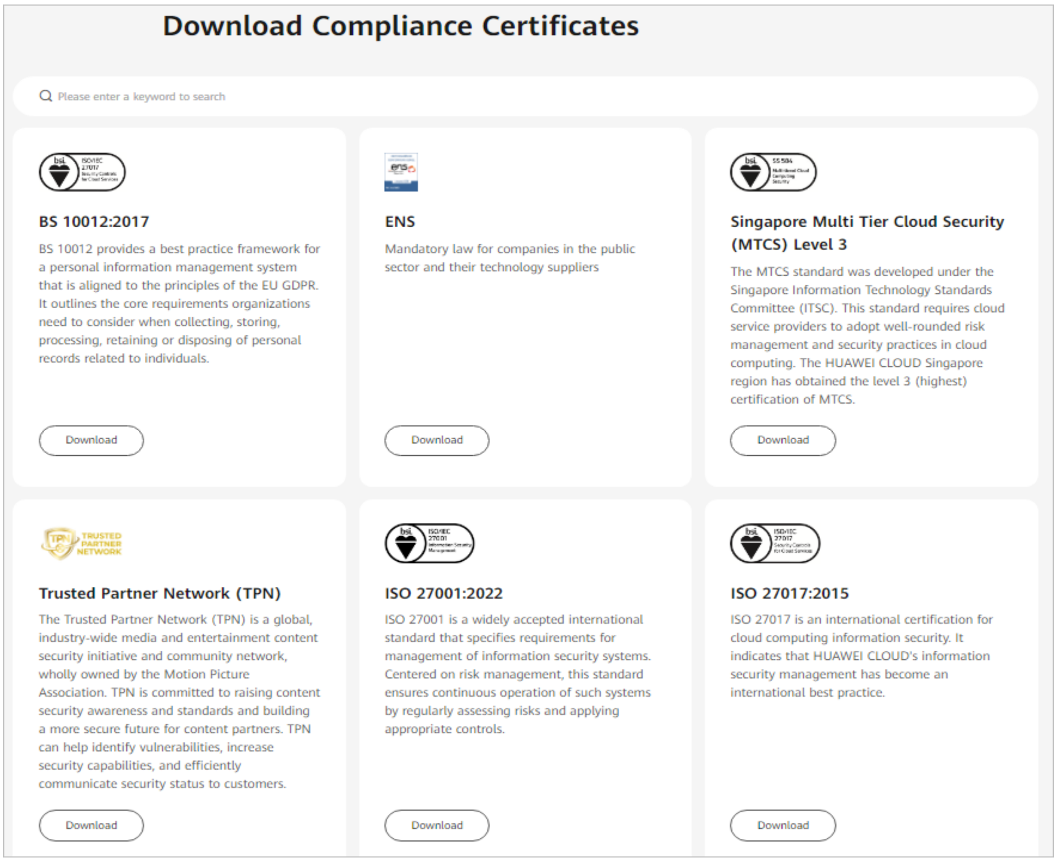
  CCE gathers custom metrics from applications and uploads them to AOM, providing you with tailored monitoring services. You can customize monitoring metrics to meet specific service requirements. For details, see **Monitoring Custom Metrics Using Cloud Native Cluster Monitoring**.

5.5 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



Download Compliance Certificates

Please enter a keyword to search

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.

Download

ENS

Mandatory law for companies in the public sector and their technology suppliers

Download

Singapore Multi Tier Cloud Security (MTCS) Level 3

The MTCS standard was developed under the Singapore Information Technology Standards Committee (ITSC). This standard requires cloud service providers to adopt well-rounded risk management and security practices in cloud computing. The HUAWEI CLOUD Singapore region has obtained the level 3 (highest) certification of MTCS.

Download

Trusted Partner Network (TPN)

The Trusted Partner Network (TPN) is a global, industry-wide media and entertainment content security initiative and community network, wholly owned by the Motion Picture Association. TPN is committed to raising content security awareness and standards and building a more secure future for content partners. TPN can help identify vulnerabilities, increase security capabilities, and efficiently communicate security status to customers.

Download

ISO 27001:2022

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.

Download

ISO 27017:2015

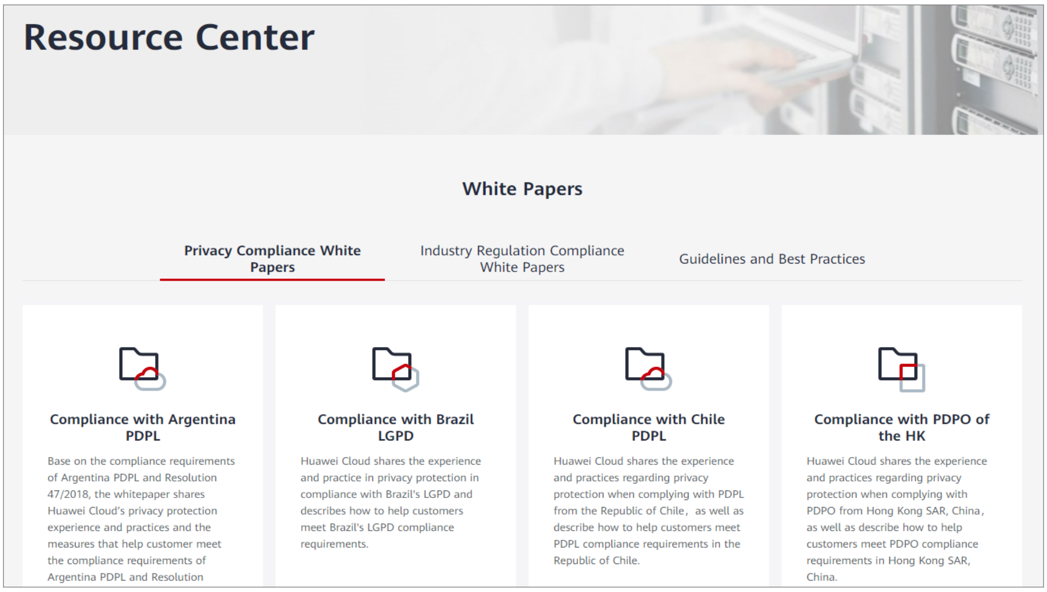
ISO 27017 is an international certification for cloud computing information security. It indicates that HUAWEI CLOUD's information security management has become an international best practice.

Download

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 Permissions

CCE permissions management allows you to assign permissions to IAM users and user groups under your tenant accounts. CCE combines the advantages of IAM and 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. A user group is simply a group of users. By granting cluster permissions to specific user groups, you can enable those users to perform various operations on clusters, including creating or deleting clusters, nodes, node pools, charts, and add-ons. In the meantime, you can restrict other user groups to only view clusters.

  Cluster-level permissions involve non-Kubernetes native 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, CCE automatically assigns the cluster-admin permission to you, which means you have full control on all resources in all namespaces in the cluster.
- When viewing CCE resources on the console, the resources displayed depend on the namespace permissions. If no namespace permissions are granted, the console will not show you the resources.

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

New IAM users do not have any permissions assigned by default. You need to first add them to one or more groups and then attach policies or roles to these groups. The IAM users then inherit permissions from the groups and can perform specified operations on cloud services based on the permissions they have been assigned.

CCE is a project-level service deployed for specific 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 the authorized region.

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, 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 6-1 lists all the system-defined permissions for CCE.

Table 6-1 System-defined permissions for 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	System-defined roles	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** ● If you are assigned with both **CCE Administrator** and **NAT Gateway Administrator** permissions, you can use NAT Gateway functions for clusters. ● If an IAM user is required to grant cluster namespace permissions to other users or user groups, the user must have the IAM read-only permission.
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

Role/ Policy Name	Description	Type	Dependencies
CCE ReadOnly Access	Permissions to view CCE cluster resources, excluding the namespace-level permissions of the clusters (with Kubernetes RBAC enabled)	Policy	None

Table 6-2 Common operations supported by system-defined permissions

Operation	CCE ReadOnlyAccess	CCE FullAccess	CCE Administrator
Creating a cluster	Not supported	Supported	Supported
Deleting a cluster	Not supported	Supported	Supported
Updating a cluster, for example, updating cluster node scheduling parameters and providing RBAC support to clusters	Not supported	Supported	Supported
Upgrading a cluster	Not supported	Supported	Supported
Waking up a cluster	Not supported	Supported	Supported
Hibernating a cluster	Not supported	Supported	Supported
Listing all clusters	Supported	Supported	Supported
Querying cluster details	Supported	Supported	Supported
Adding a node	Not supported	Supported	Supported
Deleting one or more nodes	Not supported	Supported	Supported
Updating a node. For example, changing the node name.	Not supported	Supported	Supported
Querying node details	Supported	Supported	Supported
Listing all nodes	Supported	Supported	Supported
Listing all jobs	Supported	Supported	Supported

Operation	CCE ReadOnlyAccess	CCE FullAccess	CCE Administrator
Deleting one or more cluster jobs	Not supported	Supported	Supported
Querying job details	Supported	Supported	Supported
Creating a storage volume	Not supported	Supported	Supported
Deleting a storage volume	Not supported	Supported	Supported
Performing operations on all Kubernetes resources	Supported (Kubernetes RBAC required)	Supported (Kubernetes RBAC required)	Supported
Viewing all CIA resources	Supported	Supported	Supported
Performing operations on all CIA resources	Not supported	Supported	Supported
Performing all operations on ECSs	Not supported	Supported	Supported
Performing all operations on EVS disks EVS disks can be attached to cloud servers and scaled to a higher capacity whenever needed.	Not supported	Supported	Supported
Performing all operations on VPC A cluster must run in a VPC. When creating a namespace, create or associate a VPC for the namespace so that all containers in the namespace will run in the VPC.	Not supported	Supported	Supported
Viewing details of all resources on an ECS In CCE, a node is an ECS with multiple EVS disks.	Supported	Supported	Supported
Listing all resources on an ECS	Supported	Supported	Supported

Operation	CCE ReadOnlyAccess	CCE FullAccess	CCE Administrator
Viewing details about all EVS disk resources EVS disks can be attached to cloud servers and scaled to a higher capacity whenever needed.	Supported	Supported	Supported
Listing all EVS resources	Supported	Supported	Supported
Viewing details about all VPC resources A cluster must run in a VPC. When creating a namespace, create or associate a VPC for the namespace so that all containers in the namespace will run in the VPC.	Supported	Supported	Supported
Listing all VPC resources	Supported	Supported	Supported
Viewing details about all ELB resources	Not supported	Not supported	Supported
Listing all ELB resources	Not supported	Not supported	Supported
Viewing details about all SFS resources	Supported	Supported	Supported
Listing all SFS resources	Supported	Supported	Supported
Viewing details about all AOM resources	Supported	Supported	Supported
Listing all AOM resources	Supported	Supported	Supported
Performing all operations on AOM auto scaling rules	Supported	Supported	Supported

Namespace-level Permissions (Assigned by Using Kubernetes RBAC)

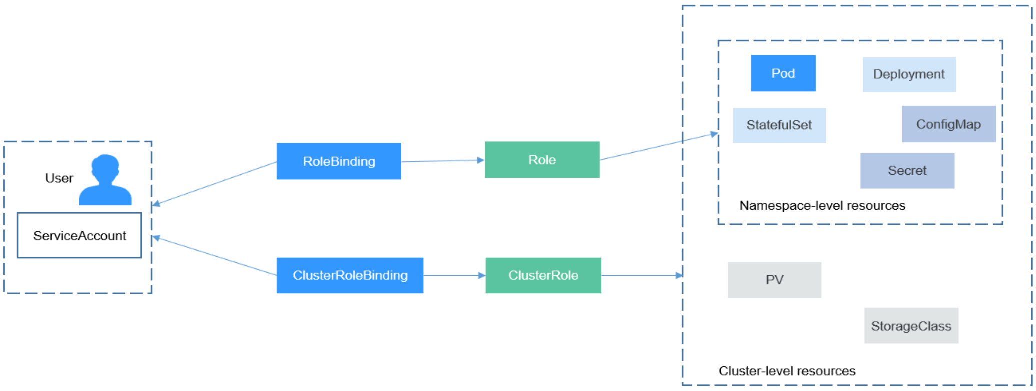
You can regulate users' or user groups' access to Kubernetes resources in a single namespace based on their Kubernetes RBAC roles. The RBAC API declares four kinds of Kubernetes objects: Role, ClusterRole, RoleBinding, and ClusterRoleBinding, which are described as follows:

- 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 6-1 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 ClusterRoles:

- view (read-only): read-only permission on most resources in all or selected namespaces.
- edit (development): read and write permissions on most resources in all or selected namespaces. If this ClusterRole is configured for all namespaces, its capability is the same as the O&M permission.
- admin (O&M): read and write permissions on most resources in all namespaces, and read-only permission on nodes, storage volumes, namespaces, and quota management.
- cluster-admin (administrator): read and write permissions on all resources in all namespaces.
- drainage-editor: drain a node.
- drainage-viewer: view the nodal drainage status but cannot drain a node.
- geip-editor-role: use of global EIPs in the cluster, but only read and write permissions on GEIPs.
- icagent-clusterRole: report Kubernetes event logs to LTS.

In addition to the preceding typical ClusterRoles, you can define Role and RoleBinding to grant permissions to add, delete, modify, and obtain resources (such as nodes, PVs, and CustomResourceDefinitions) and different resources (such as pods, Deployments, and Services) within specific namespaces. This allows for more precise permission control.

Helpful Links

- **IAM Service Overview**
- **Cluster Permissions (IAM-based)**
- **Permissions Policies and Supported Actions**

7 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, a CCE standard cluster cannot be changed to a CCE Turbo cluster.
  - Number of master nodes: For example, a non-HA cluster (with one master node) cannot be changed to an HA cluster (with three master nodes).
  - AZ where a master node is deployed
  - 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, a container tunnel network cannot be changed to a VPC network.
- The ECS nodes created on CCE can be billed on a pay-per-use or yearly/monthly basis. Other resources such as load balancers are billed on a pay-per-use basis. If an involved service allows you to change the billing mode from pay-per-use to yearly/monthly, the billing mode can be changed on the service console.
- If yearly/monthly ECS nodes are managed in a cluster, you are not allowed to renew them on the CCE console. Instead, renew the ECSs on the ECS console.
- CCE underlying resources such as ECS nodes are limited by quota and their inventory. It is possible that only some nodes are created during cluster creation, cluster scaling, or auto scaling.
- ECS node specifications: CPU ≥ 2 cores, memory ≥ 4 GiB
- 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.

Networks

- By default, a NodePort Service is accessed within a VPC. To access a NodePort Service through the Internet, bind an EIP to the node in the cluster beforehand.
- LoadBalancer Services allow workloads to be accessed from public networks through ELB. This access mode has the following restrictions:
  - Automatically created load balancers should not be used by other resources. Otherwise, these load balancers cannot be completely deleted.
  - 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. Network policies are classified into the following types:
    - Ingress: All versions support this type.
    - Egress: The cluster must be of v1.23 or later.
  - Network isolation is not supported for IPv6 addresses.
  - If you upgrade a CCE cluster to a version that supports egress rules in in-place mode, the rules will not work because the node OS is not upgraded. In this case, reset the node.

Storage Volumes

- Constraints on EVS volumes:
  - EVS disks cannot be attached across AZs and cannot be used by multiple workloads, multiple pods of the same workload, or multiple tasks. Data sharing of a shared disk is not supported between nodes in a CCE cluster. If an EVS disk is attached to multiple nodes, I/O conflicts and data cache conflicts may occur. Therefore, select only one pod when creating a Deployment that uses 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 volumes mounted, a new pod cannot be started because EVS disks cannot be attached.
- Constraints on SFS volumes:
  - Multiple PVs can use the same SFS or SFS Turbo file system with the following restrictions:
    - Do not mount the PVCs/PVs that use the same underlying SFS or SFS Turbo volume to one pod. This will lead to a pod startup failure because not all PVCs can be mounted to the pod due to the same **volumeHandle** value.
    - The **persistentVolumeReclaimPolicy** parameter in the PVs must be set to **Retain**. Otherwise, when a PV is deleted, the associated underlying volume may be deleted. In this case, other PVs associated with the underlying volume malfunction.

        - When the underlying volume is repeatedly used, enable isolation and protection for ReadWriteMany at the application layer to prevent data overwriting and loss.
    - If SFS 3.0 is used, the Everest add-on of v2.0.9 or later must be installed in the cluster.
    - If SFS 3.0 is used, the owner group and permission of the mount point cannot be modified. The default owner of the mount point is user **root**.
    - If SFS 3.0 is used, there may be a latency during the creation or deletion of PVCs and PVs. The billing duration is determined by the time when the SFS system is created or deleted on the SFS console.
    - If the reclamation policy of SFS 3.0 is set to **Delete**, PVs and PVCs can be properly deleted only after all files are manually deleted from the mounted SFS system.
- Constraints on OBS volumes:
    - If OBS volumes are used, the owner group and permission of the mount point cannot be modified.
    - Every time an OBS volume is mounted to a workload through a PVC, a resident process is created in the backend. When a workload uses too many OBS volumes or reads and writes a large number of object storage files, resident processes will consume a significant amount of memory. To ensure stable running of the workload, make sure that the number of OBS volumes used does not exceed the requested memory. For example, if the workload requests for 4 GiB of memory, the number of OBS volumes should be **no more than** 4.
    - Kata containers do not support OBS volumes.
    - Hard links are not supported when common buckets are mounted.
- Constraints on local PVs:
    - Local PVs are supported only when the cluster version is v1.21.2-r0 or later and the Everest add-on version is 2.1.23 or later. Version 2.1.23 or later is recommended.
    - Removing, deleting, resetting, or scaling in a node will cause the PVC/PV data of the local PV associated with the node to be lost, which cannot be restored or used again. In these scenarios, the pod that uses the local PV is evicted from the node. A new pod will be created and stays in the pending state. This is because the PVC used by the pod has a node label, due to which the pod cannot be scheduled. After the node is reset, the pod may be scheduled to the reset node. In this case, the pod remains in the creating state because the underlying logical volume corresponding to the PVC does not exist.
    - Do not manually delete the corresponding storage pool or detach data disks from the node. Otherwise, exceptions such as data loss may occur.
    - A local PV cannot be mounted to multiple workloads or jobs at the same time.
- Constraints on local EVs:
    - Local EVs are supported only when the cluster version is v1.21.2-r0 or later and the Everest add-on version is 1.2.29 or later.
    - Do not manually delete the corresponding storage pool or detach data disks from the node. Otherwise, exceptions such as data loss may occur.

  - Ensure that the **/var/lib/kubelet/pods/** directory is not mounted to the pod on the node. Otherwise, the pod, mounted with such volumes, may fail to be deleted.
- 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 (VBD or SCSI), 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 checked or configured.
  - Snapshots can be created only for EVS disks that are available or in use, and a maximum of seven snapshots can be created for a single EVS disk.
  - Snapshots can be created only for PVCs created using the storage class (whose name starts with csi) provided by the Everest add-on. Snapshots cannot be created for PVCs created using the FlexVolume storage class whose name is ssd, sas, or sata.
  - Snapshot data of encrypted disks is stored encrypted, and that of non-encrypted disks is stored non-encrypted.

Add-ons

CCE uses Helm charts to deploy add-ons. To modify or upgrade an add-on, perform operations on the **Add-ons** page or use open add-on management APIs. Do not directly modify add-on resources on the backend. Otherwise, add-on exceptions or other unexpected problems may occur.

CCE Cluster Resources

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

Item	Constraints on Common Users	Exception Handling
Total number of clusters in a region	50	Submit a **service ticket** to request for increasing the quota.
Number of nodes in a cluster (cluster management scale)	A maximum of 50, 200, 1000, or 2000 nodes can be selected.	Submit a **service ticket** to request for increasing the quota. A maximum of 10,000 nodes are supported.
Maximum number of pods on a node	256 **NOTE** In a CCE Turbo cluster, the maximum number of pods on a node is determined by the number of NICs that can be used by the node.	To increase the deployment density on a node, submit a **service ticket** to increase the maximum number of pods on a node. A maximum of 512 pods are supported.

Item	Constraints on Common Users	Exception Handling
Maximum number of pods managed by a cluster	100,000 pods	If the number of supported pods cannot meet your requirements, submit a **service ticket** to request for technical support to optimize your clusters based on the service model.

Cluster Capacity Limit

The capacity of a cluster is made up of various resource types, including container groups (pods), cloud storage instances (persistent volumes), and Services. Additionally, the size of these resource objects can also impact the cluster capacity.

For example:

- If there are too many pods, the maximum number of pods will decrease within a certain performance range.
- As the number of pods approaches the upper limit, the upper limits of other resource types in the cluster will also decrease accordingly.

Since clusters in actual application environments contain multiple resource types, it is possible that the number of resources for a single type may not reach its upper limit. It is important to monitor cluster resource usage regularly and plan and manage the resources effectively to ensure the best performance of all resources. If the current specifications do not meet your requirements, you can scale out the cluster to ensure stability.

Dependent Underlying Cloud Resources

Category	Item	Constraints on Common Users
Compute	Pods	1000
	Cores	8000
	RAM capacity (MB)	16,384,000
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

Category	Item	Constraints on Common Users
	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

NOTE

If the current quota cannot meet your requirements, submit a **service ticket** to request for increasing your quota.

8 Billing

Billing Modes

There are yearly/monthly and pay-per-use billing modes to meet your requirements. For details, see **Billing Modes**.

- Yearly/Monthly is a prepaid billing mode. You pay in advance for a subscription term. Before purchasing yearly/monthly resources, ensure that your account has sufficient balance.
- Pay-per-use is a postpaid billing mode. You pay as you go and just pay for what you use.

After purchasing CCE clusters or cluster resources, you can change their billing modes if the current billing mode cannot meet your service requirements. For details, see **Billing Mode Changes**.

Billing Items

You will be billed for clusters, nodes, and other cloud service resources. For details about the billing factors and formulas for each billed item, see **Billed Items**.

1. **Clusters**: the cost of resources used by master nodes. It varies with the cluster type (VMs or BMSs and the number of master nodes) and size (the number of worker nodes).

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

2. **Other cloud resources**: the cost of IaaS resources in use. Such resources, which are created either manually or automatically during cluster creation, include ECSs, EVS disks, EIPs, bandwidth, and load balancers.

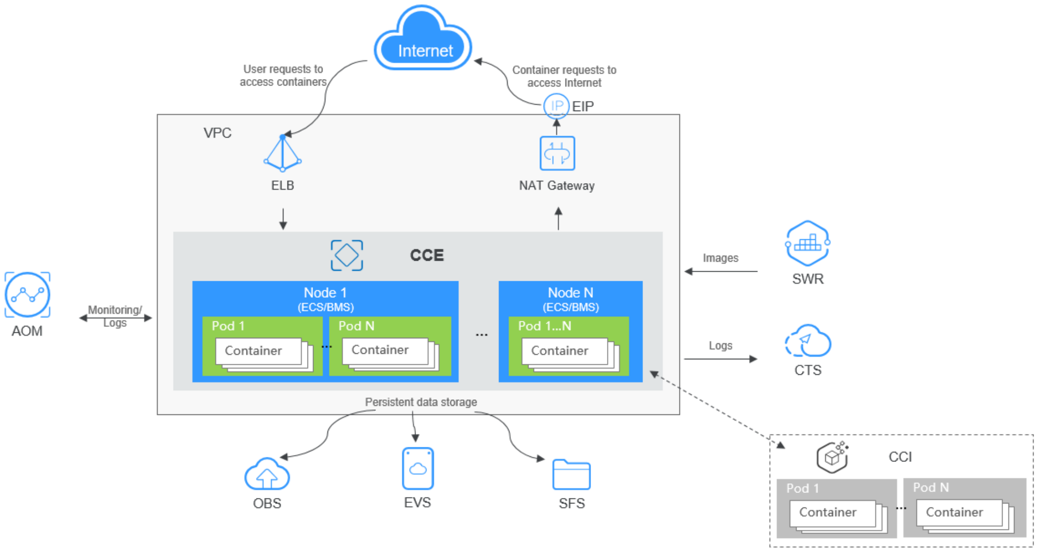
   For more pricing details, see **Product Pricing Details**.

For more information about the billing samples and the billing for each item, see **Billing Examples**.

9 Related Services

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

Figure 9-1 Relationships between CCE and other services



Relationships Between CCE and Other Services

Table 9-1 Relationships between CCE and other services

Service	Relationship	Related Feature
ECS	An ECS with multiple EVS disks is a node in CCE. You can choose ECS specifications during node creation.	• **Creating a Node** • **Adding Nodes for Management**

Service	Relationship	Related Feature
VPC	VPCs are required for CCE clusters to function. They offer segregated network environments. When creating a node pool within a cluster, nodes in the pool are assigned private IP addresses from the VPC CIDR block.	**Buying a CCE Standard/ Turbo Cluster**
ELB	Load balancers can be associated with applications created on CCE. They help distribute external access traffic to various backend containerized applications. You can use **elastic load balances** to access CCE workloads from external networks.	● **Creating a Deployment** ● **Creating a StatefulSet** ● **LoadBalancer**
NAT Gateway	The NAT Gateway service offers source network address translation (SNAT), which allows private IP addresses to be translated into public IP addresses by binding an elastic IP address (EIP) to the gateway. This enables container instances within the VPC to share EIPs and access the Internet. You can define SNAT rules on the **NAT gateway** to let containers access the Internet.	● **Creating a Deployment** ● **Creating a StatefulSet** ● **DNAT**
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**
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 an EVS Disk Through a Dynamic PV**

Service	Relationship	Related Feature
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 an OBS Bucket Through a Dynamic PV**
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 an SFS File System Through a Dynamic PV**
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.	**Overview**
Cloud Trace Service (CTS)	CTS records operations on your cloud resources, allowing you to obtain, 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 Cloud Trace Service**

10 Regions and AZs

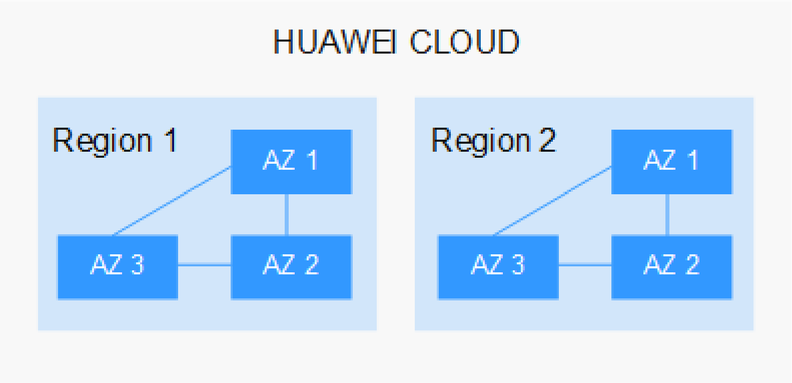
Definition

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 10-1 shows the relationship between the region and AZ.

Figure 10-1 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 more information about regions and endpoints, see **Regions and Endpoints**.