Cloud Container Engine

Service Overview

Issue 01

Date 2024-03-04

HUAWEI

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

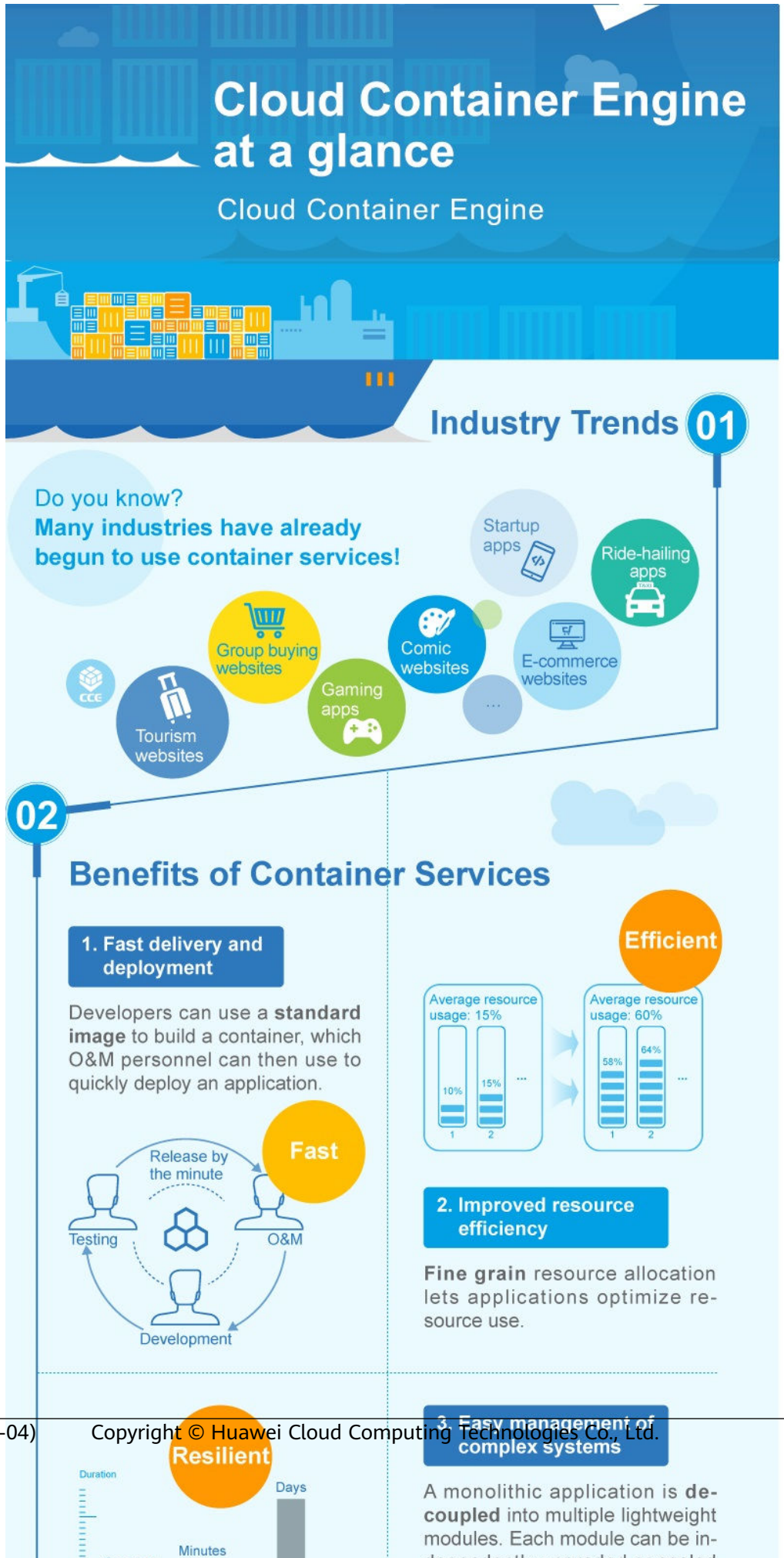
Notice

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



2 What Is CCE?

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 (ECS), networking (VPC, EIP, and ELB), storage (EVS, OBS, and SFS), and many other services. Multi-AZ, multi-region disaster recovery ensures high availability 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 ecosystem. 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

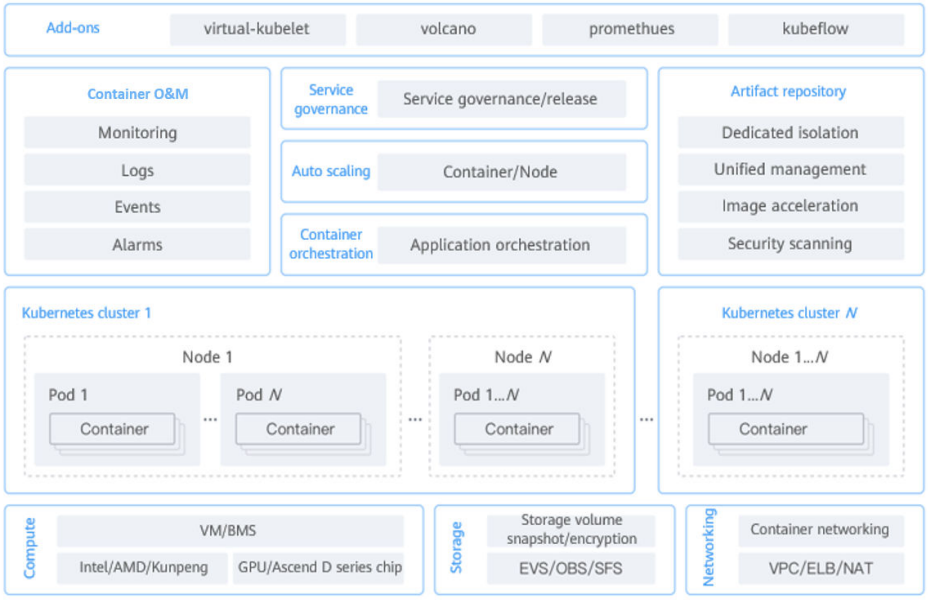
CCE provides both CCE cluster and CCE Turbo cluster.

Category	Subcategory	CCE	CCE Turbo
Positioning	-	Standard clusters that provide highly reliable and secure containers for commercial use	Next-gen container cluster designed for Cloud Native 2.0, with accelerated computing, networking, and scheduling

Category	Subcategory	CCE	CCE Turbo
Application scenario	-	For users who expect to use container clusters to manage applications, obtain elastic computing resources, and enable simplified management on computing, network, and storage resources	For users who have higher requirements on performance, resource utilization, and full-scenario coverage
Specification difference	Network model	Cloud-native network 1.0: applies to common, smaller-scale scenarios. • Tunnel network • Virtual Private Cloud (VPC) network	Cloud Native Network 2.0: applies to large-scale and high-performance scenarios. Max networking scale: 2,000 nodes
	Network performance	Overlays the VPC network with the container network, causing certain performance loss.	Flattens the VPC network and container network into one, achieving zero performance loss.
	Network isolation	• Tunnel network model: supports network policies for intra-cluster communications. • VPC network model: supports no isolation.	Associates pods with security groups. Unifies security isolation in and out the cluster via security groups' network policies.
	Security isolation	Runs common containers, isolated by cgroups.	• Physical machine: runs Kata containers, allowing VM-level isolation. • VM: runs common containers, isolated by cgroups.
	Edge infrastructure management	None	Supports management of Intelligent EdgeSite (IES).

CCE Cluster Architecture

Figure 2-1 CCE cluster architecture



- Compute: CCE supports various Huawei Cloud compute instances including both VMs and BMS servers running on Kunpeng, GPUs, or Huawei Ascend chips, allowing GPU virtualization, shared scheduling, and resource-aware scheduling optimization.
- Networking: supports interconnection with high-performance, secure, reliable, and multi-protocol dedicated load balancers as the service traffic ingress.
- Storage: provides cloud storage services like Elastic Volume Service (EVS), Scalable File Service (SFS), and Object Storage Service (OBS) and capabilities of disk encryption, snapshot, and backup.
- Kubernetes cluster service: full lifecycle cluster management including cluster buying, connecting, upgrading, and managing.
- 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: interconnects with SoftWare Repository for Container (SWR) to support full lifecycle management of images. It provides easy-to-use, secure, and reliable image management, helping you quickly deploy containerized applications.
- Auto scaling: enables resource scaling for workloads and nodes. With auto scaling, CCE allows you to economically adjust compute resources based on service requirements and policies.
- Service governance: CCE integrates Application Service Mesh (ASM). Grayscale release, traffic governance and monitoring, all done in a non-intrusive manner.
- Container O&M: CCE integrates Container Intelligent Analysis (CIA) so that CCE can monitor applications and resources in real time, collect, manage, and

analyze logs, collect metrics and events, and provide one-click monitoring function.

- Add-ons: CCE provides multiple types of add-ons for you to manage your clusters as required.

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 Helm charts.

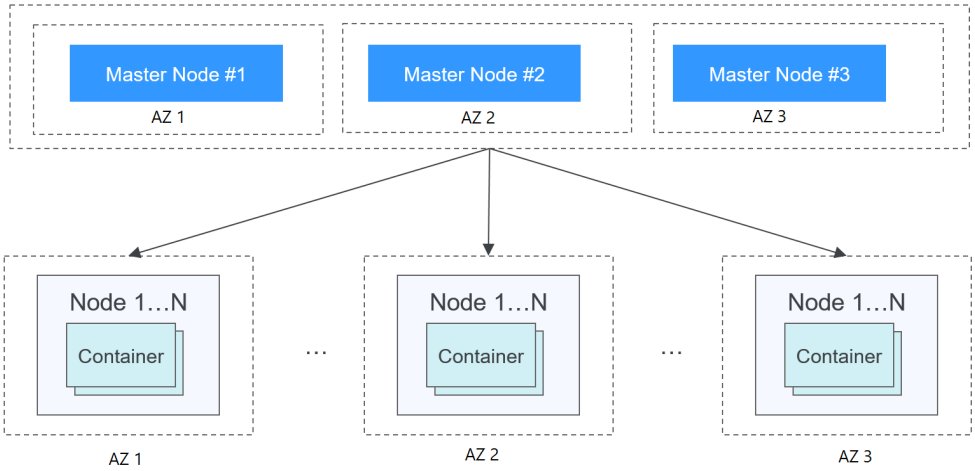
High Performance

- 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 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 Kubernetes 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 and update 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 Kubernetes Cluster	CCE
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 provides various container-optimized OS images with additional stability tests and security hardening based on native Kubernetes clusters and runtime versions, reducing management costs and risks and improving the reliability and security of 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.
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.

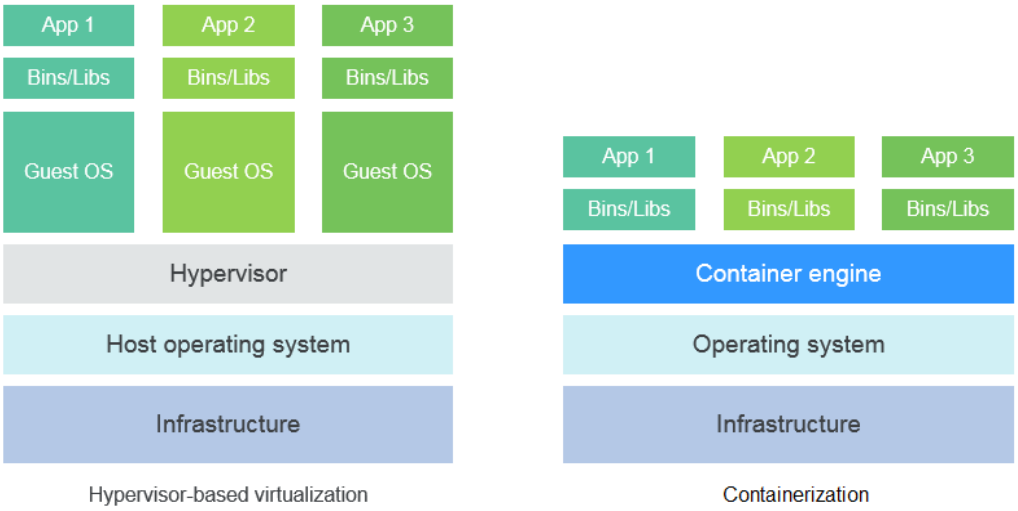
Why Containers?

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

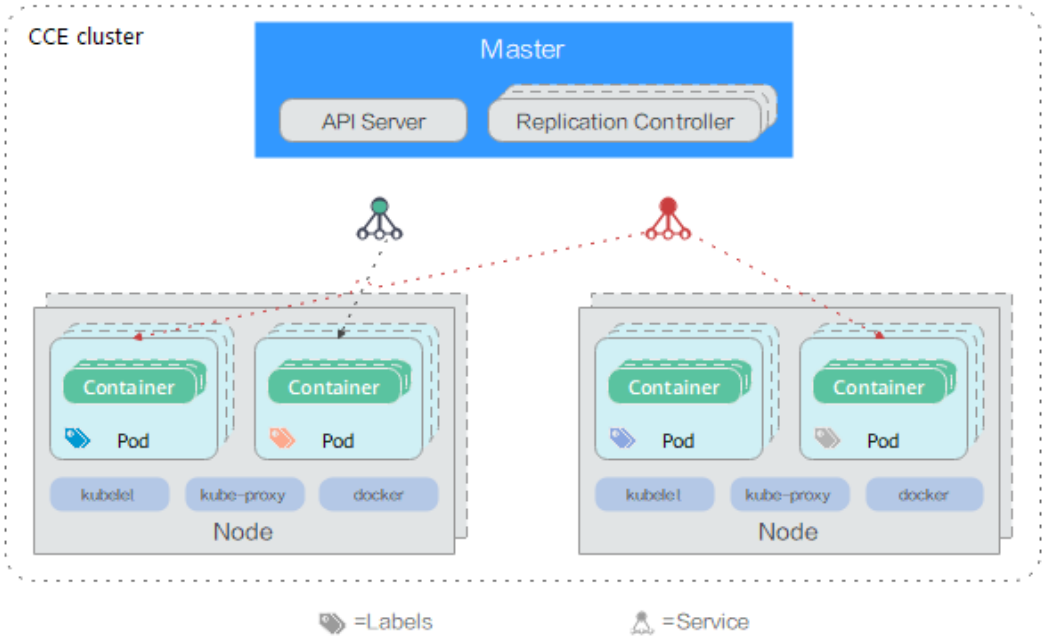
4.1 Containerized Application Management

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.

- 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

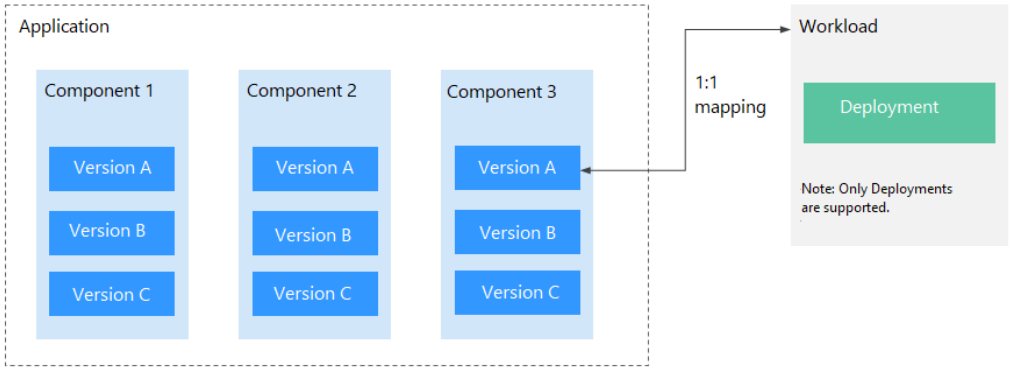
- Multiple types of workloads

  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

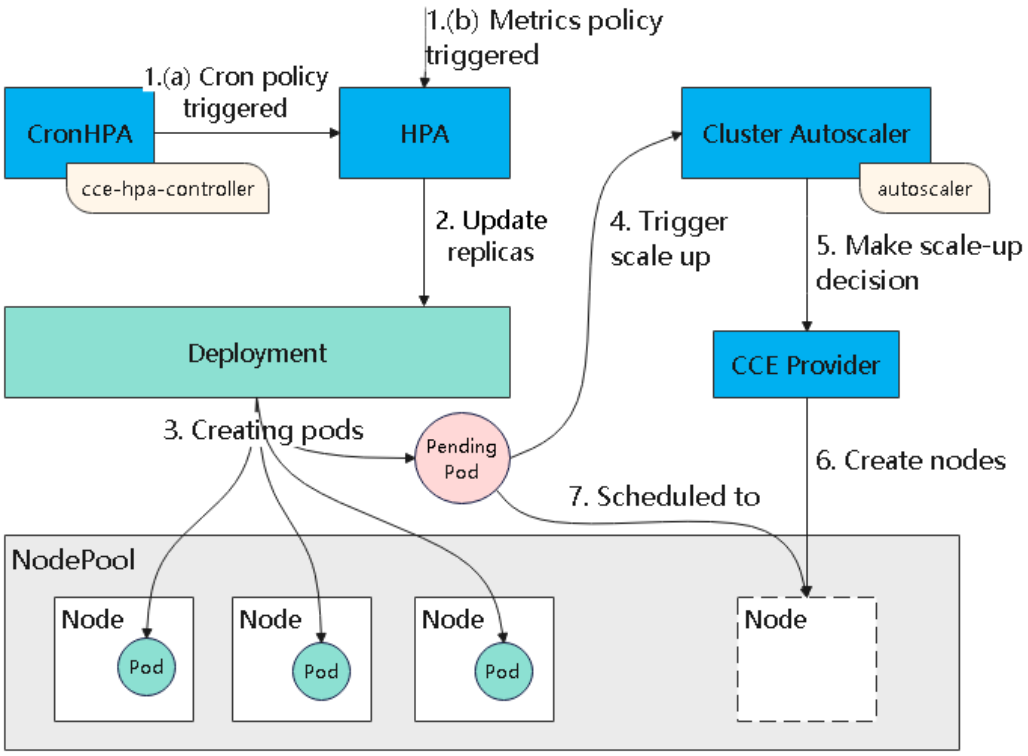
  Bills you only for the scaled cloud servers as you use.

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 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

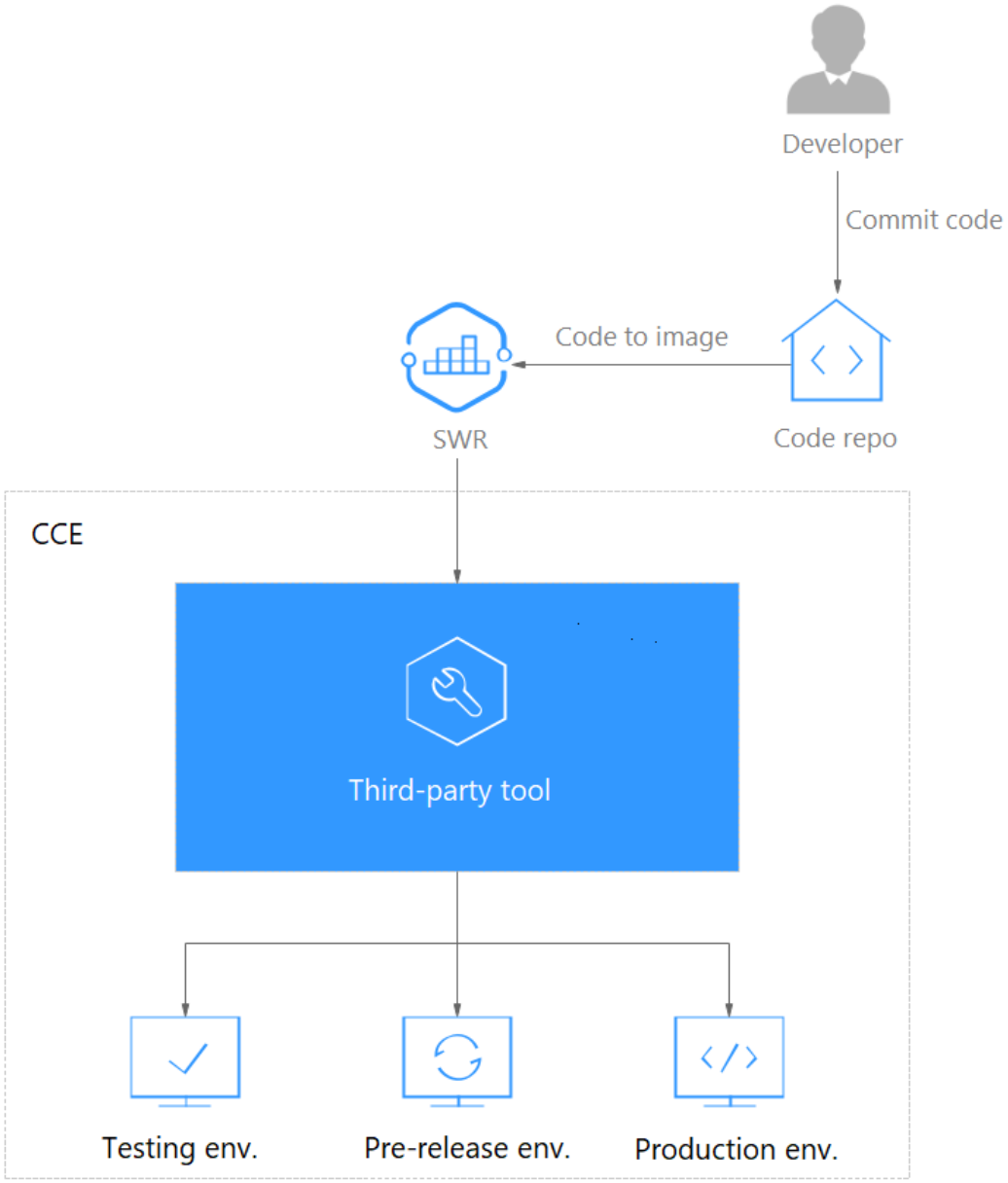
  Provides various APIs to integrate with existing CI/CD systems for in-depth customization.
- High performance

  Enables flexible scheduling with a containerized architecture.

Related Services

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

Figure 4-4 How DevOps works



4.4 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

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.

- 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.

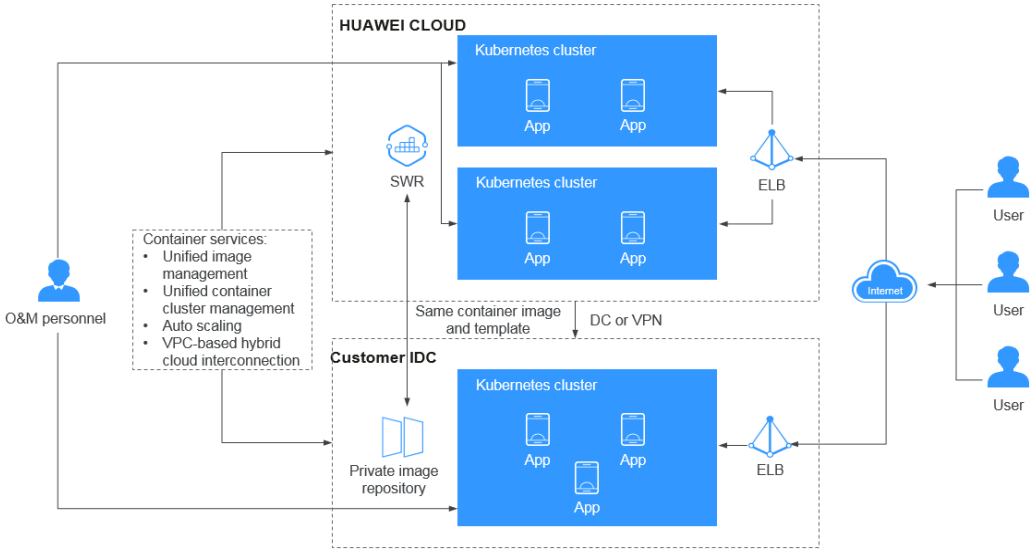
- Lower costs

  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 4-5 How hybrid cloud works



5 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:
  - 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.
- 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:

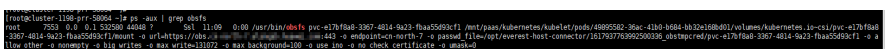
  - Only clusters that use the tunnel network model support network policies. Network policies are classified into the following types:
    - Ingress: All versions support this type.
  - Network isolation is not supported for IPv6 addresses.

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 attacked to multiple nodes, I/O conflicts and data cache conflicts may occur. Therefore, create 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 disks attached, 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 disks attached, 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 all PVCs/PVs that use the same underlying SFS or SFS Turbo file system to a pod. This leads to a pod startup failure because not all PVCs can be mounted to the pod due to the same **volumeHandle** values of these PVs.
    - The **persistentVolumeReclaimPolicy** parameter in the PVs is suggested to 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.
- Constraints on OBS volumes:
  - If OBS volumes are used, the owner group and permission of the mount point cannot be modified.
  - CCE allows parallel file systems to be mounted using OBS SDKs or PVCs. If PVC mounting is used, the obsfs tool provided by OBS must be used. An obsfs resident process is generated each time an object storage volume generated from the parallel file system is mounted to a node, as shown in the following figure.

    Figure 5-1 obsfs resident process



Reserve 1 GiB of memory for each obsfs process. For example, for a node with 4 vCPUs and 8 GiB of memory, an obsfs parallel file system should be mounted to **no more than** eight pods.

NOTE

- An obsfs resident process runs on a node. If the consumed memory exceeds the upper limit of the node, the node malfunctions. On a node with 4 vCPUs and 8 GiB of memory, if more than 100 pods are mounted to a parallel file system, the node will be unavailable. Control the number of pods mounted to a parallel file system on a single node.

  - Kata containers do not support OBS volumes.

- Restrictions on using 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.
- Restrictions on using 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 (SCSI or VBD), 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.
  - 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.

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.
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.

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

Category	Item	Constraints on Common Users
	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

NOTE

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

6 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, make sure you have a top-up account with a 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 **Billing Items**.

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

7 Permissions

CCE 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. 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, 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)

By default, new IAM users do not have permissions assigned. 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, 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.

Table 7-1 lists all the system permissions supported by CCE.

Table 7-1 System permissions supported by CCE

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

Table 7-2 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	√	√	√

Operation	CCE ReadOnlyAccess	CCE FullAccess	CCE Administrator
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)	√
Viewing all CIA resources	√	√	√
Performing operations on all CIA resources	x	√	√
Performing all operations on ECSs	x	√	√
Performing all operations on 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, 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	√	√	√

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.	√	√	√
Listing all EVS resources	√	√	√
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.	√	√	√
Listing all VPC resources	√	√	√
Viewing details about all ELB resources	x	x	√
Listing all ELB resources	x	x	√
Viewing details about all SFS resources	√	√	√
Listing all SFS resources	√	√	√
Viewing details about all AOM resources	√	√	√
Listing AOM resources	√	√	√
Performing all operations on AOM auto scaling rules	√	√	√

Namespace-level Permissions (Assigned by Using Kubernetes RBAC)

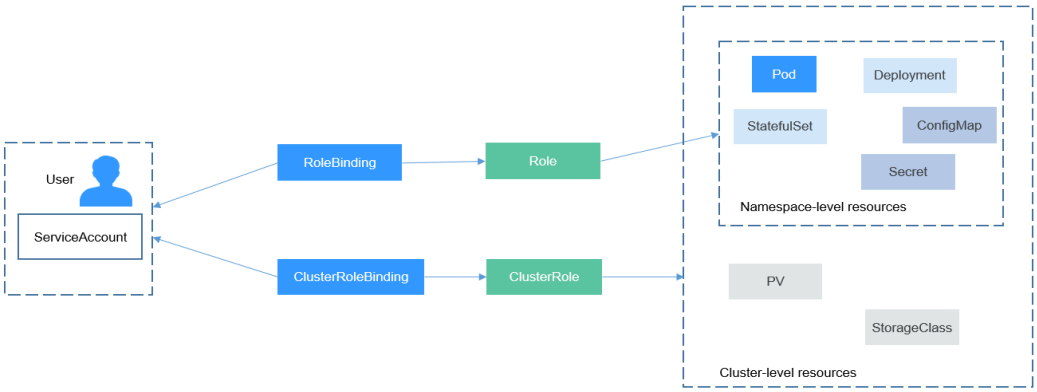
You can regulate users' or user groups' access to Kubernetes resources in a single namespace based on their Kubernetes RBAC roles. The RBAC API declares four kinds of Kubernetes objects: Role, ClusterRole, RoleBinding, and ClusterRoleBinding, which are described as follows:

- 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 7-1 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:

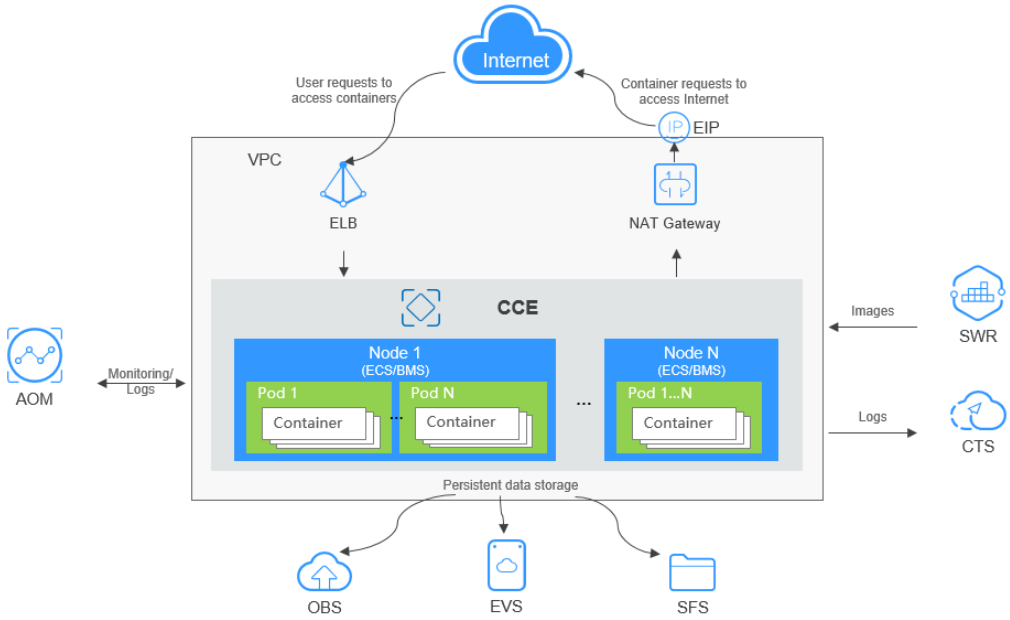
- view (read-only): read-only permission on most resources in all or selected namespaces.
- edit (development): read and write permissions on most resources in all or selected namespaces. If this ClusterRole is configured for all namespaces, its capability is the same as the O&M permission.
- admin (O&M): 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.

In addition to the preceding ClusterRoles, you can define Roles and RoleBindings to configure the permissions to add, delete, modify, and obtain resources such as pods, Deployments, and Services in the namespace.

8 Related Services

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

Figure 8-1 Relationships between CCE and other services



Relationships Between CCE and Other Services

Table 8-1 Relationships between CCE and other services

Service	Relationship
ECS	An ECS with multiple EVS disks is a node in CCE. You can choose ECS specifications during node creation.

Service	Relationship
VPC	For security reasons, all clusters created by CCE must run in VPCs. When creating a namespace, create a VPC or bind an existing VPC to the namespace so all containers in the namespace will run in this VPC.
ELB	CCE works with ELB to load balance a workload's access requests across multiple pods.
NAT Gateway	The NAT Gateway service provides source network address translation (SNAT), which translates private IP addresses to a public IP address by binding an elastic IP address (EIP) to the gateway.
SWR	An image repository is used to store and manage Docker images.
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.
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.
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.
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.
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.

9 Regions and AZs

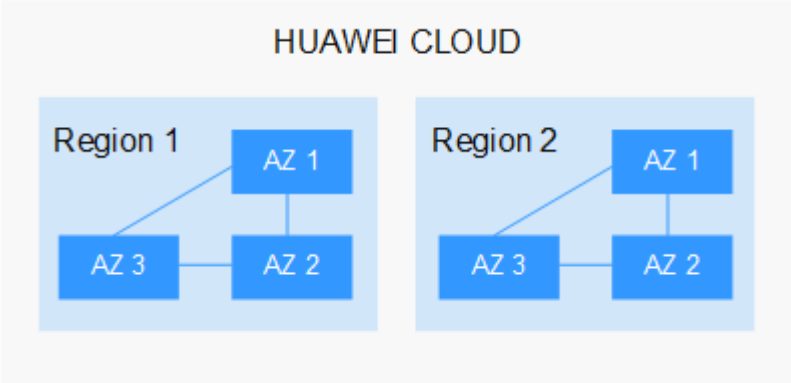
Definition

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.

shows the relationship between the region and AZ.

Figure 9-1 Regions and AZs



Huawei Cloud provides services in many regions around the world. You can select a region and AZ as needed.

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.

- 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.

