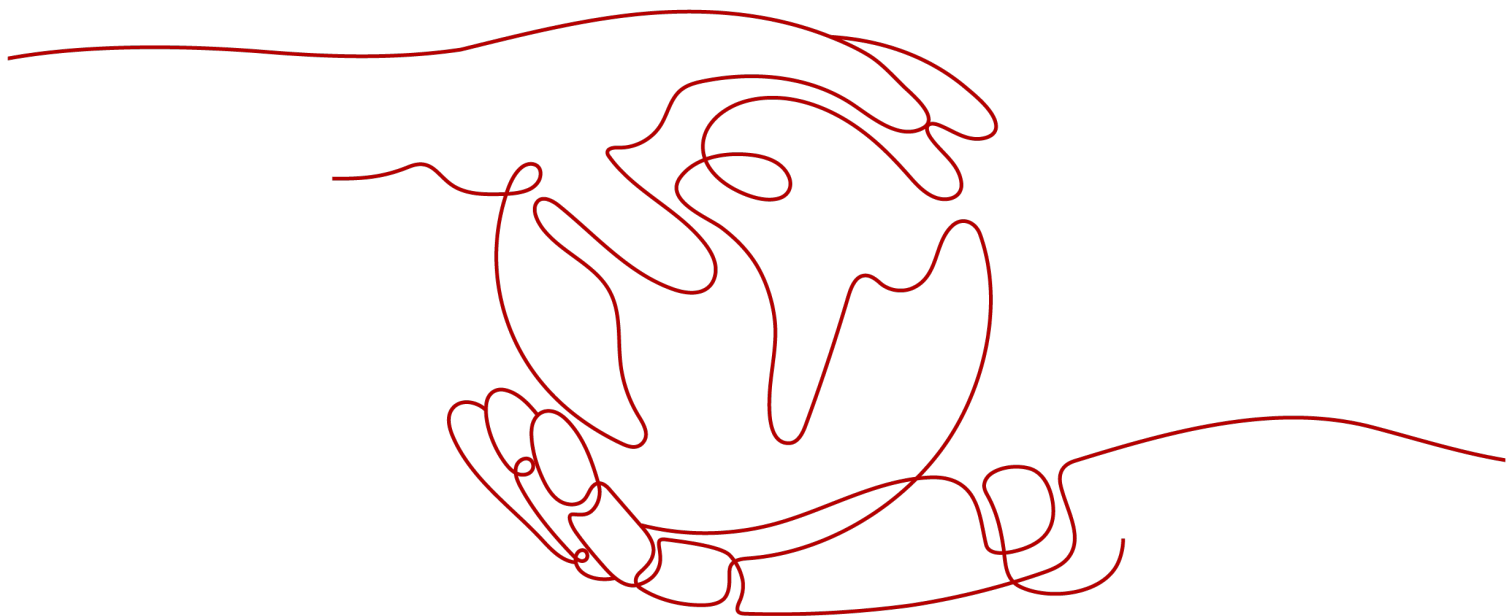


Solution Practice

Deployment Guide on DaoCloud Multi-Cloud Solution for Application Modernization

Issue 1.0

Date 2024-06-03



Copyright © Huawei Technologies Co., Ltd. 2025. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks 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 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.

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process*. For details about this process, visit the following web page:

<https://www.huawei.com/en/psirt/vul-response-process>

For vulnerability information, enterprise customers can visit the following web page:

<https://securitybulletin.huawei.com/enterprise/en/security-advisory>

Contents

1 Solution Overview	1
2 Resource and Cost Planning	11
3 Operation Process	13
4 Implementation Procedure	15
4.1 Preparing the CCE Environment	15
4.2 Deploying the Multi-Cloud Collaboration Platform	18
4.3 (Optional) Changing the Middleware	22
5 Appendixes	23
6 Change History	26

1 Solution Overview

DaoCloud Multi-Cloud Solution for Application Modernization helps modernize applications and drive digital transformation for enterprises that run distributed applications across clouds and regions. This solution provides multi-cloud cluster management, multi-cloud application orchestration, and multi-cloud service meshes.

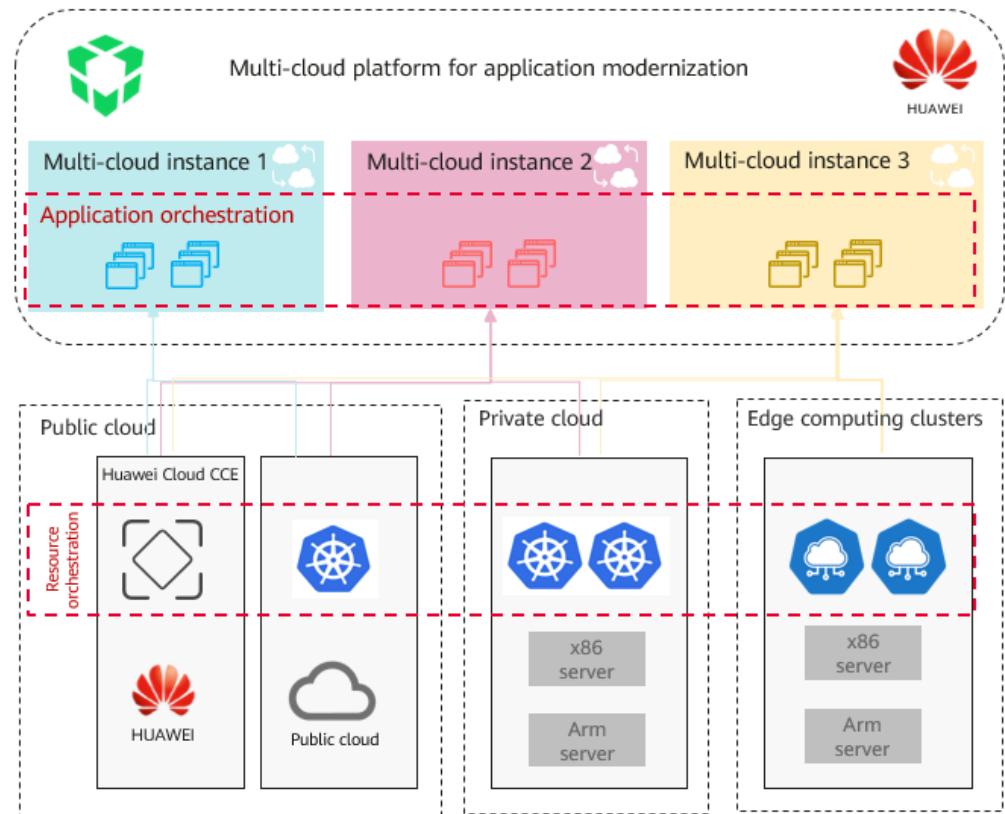
This solution enables centralized management of heterogeneous clusters (x86 and Arm) running on multiple clouds or a hybrid cloud, so you can deploy and release applications and perform O&M across clouds. This reduces cross-cloud application migration costs, simplifies O&M management, and improves application performance. This solution also supports auto scaling based on cluster resources, global load balancing, east-west communications, grayscale releases, and visualized traffic management of applications, as well as dual-mode microservice governance (microservice governance using traditional methods and service meshes). There are also various application traffic routing policies for better routing. With this solution, enterprises do not need to worry about fault recovery, so they can focus on developing efficient, lightweight, intelligent, open, elastic, and resilient modern applications.

Application Scenarios

This solution is the best choice if your enterprise needs to manage and distribute applications across clouds, import cluster information quickly, implement application failover and observability, and control permissions globally.

- **Unified Orchestration and Management of Multi-Cloud Resources and Applications**

Figure 1-1 Unified management of multi-cloud resources and applications



Pain points:

As your enterprise is running more clusters on the private cloud and heterogeneous public clouds, how to manage these clusters becomes a prominent problem.

- **Complicated cluster management:** There are a large number of clusters that need to be configured repeatedly and managed differently. Also, there is no unified entry for API calls.
- **Scattered services:** Applications have differentiated configuration in each cluster. Applications are hard to access each other across clouds, and cross-cluster migration of applications is also a problem.
- **Restricted scheduling:** Resource scheduling, application availability, and auto scaling can only be implemented within clusters.
- **Cloud vendor lock-in:** There is no neutral multi-cluster management platform. Cloud vendor lock-in happens frequently.

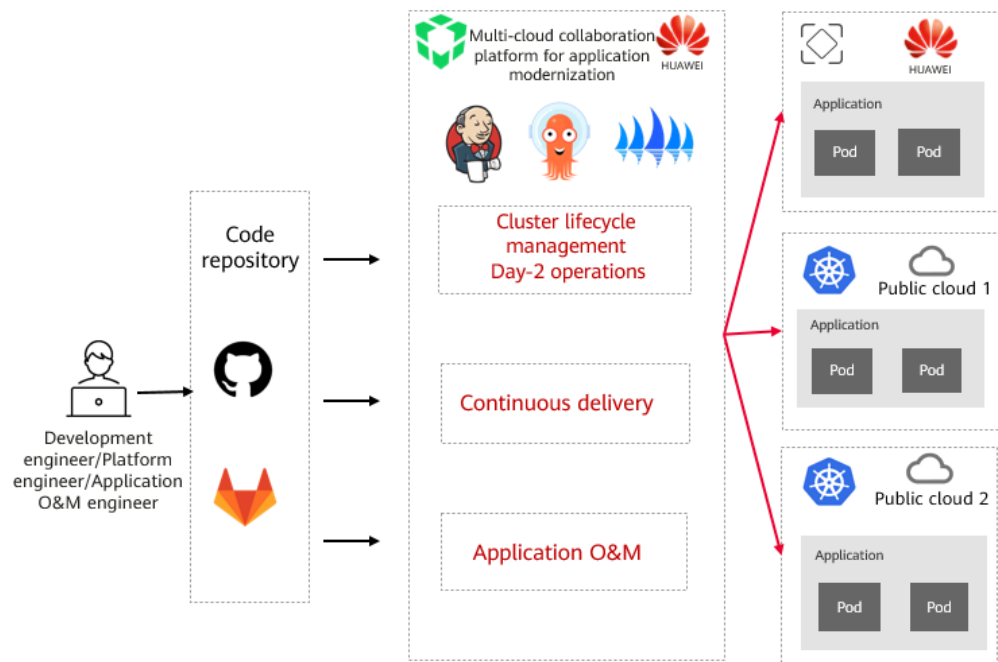
Countermeasures:

This solution enables unified orchestration and management of multi-cloud resources and applications, distributed resource collaboration and scheduling, and cross-cloud auto scaling of applications. With this solution, your enterprise can deploy applications across regions and clouds, reducing cloud costs.

- **Unified management of multi-cloud resources:** Resources on multiple clouds can be connected with just a few clicks, and the latest cluster information can be synchronized in real time. This helps you easily check

- resource changes and manage cloud resources provided by different vendors on a unified interface.
- **Multi-cloud application orchestration:** Cluster resources are abstracted into multiple cloud instances that are isolated from each other. All cloud applications can be released and maintained in a unified manner. Multi-dimensional scheduling policies are supported for transparent application access and auto scaling across clouds.
- **One-click application upgrade:** Single-cloud applications can be upgraded to multi-cloud applications with just one click, at no costs.
- **No vendor lock-in:** Multiple public cloud vendors and private cloud solutions can be selected to avoid relying on a single cloud vendor and to reduce cloud costs.
- **Integration of Multi-Cloud Resource Orchestration and Application Delivery and O&M**

Figure 1-2 Integration of multi-cloud resource orchestration and application delivery and O&M



Pain points:

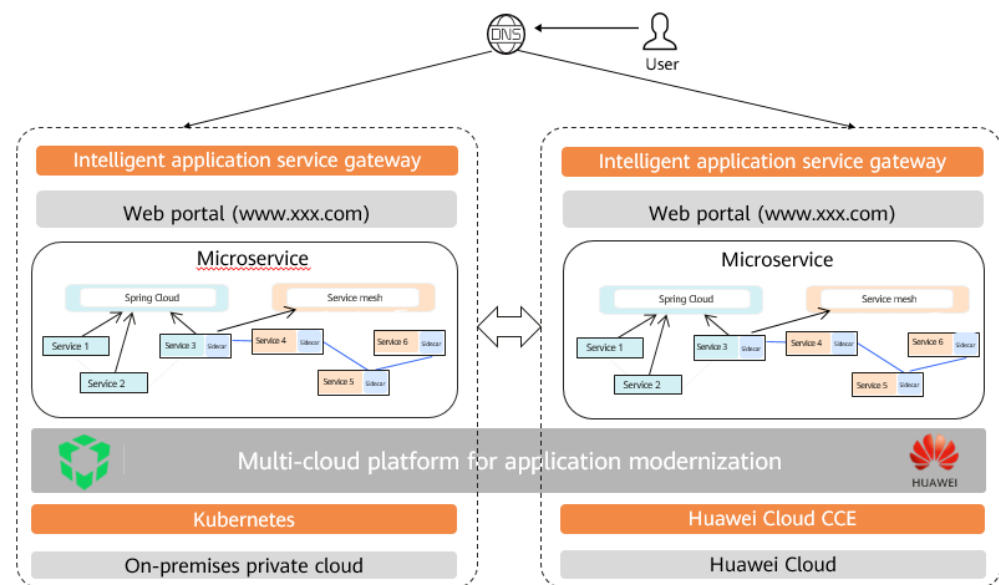
- **Complex management:** Cluster configuration and management differences among cloud platforms are challenging.
- **Difficult cluster lifecycle management:** Clusters must be managed throughout their lifecycles, from deployment and running to upgrade, capacity expansion, monitoring, and deletion. It is time-consuming and error-prone to manually manage multiple clusters at the same time.
- **Complicated and diverse service requirements:** As services expand and diversify, applications need to be deployed on different cloud platforms to meet requirements for performance, compliance, and geographical locations, which pose great challenges to continuous delivery across cloud environments.

Countermeasures:

This solution provides GitOps for integration and automation of multiple cloud platforms throughout the O&M lifecycle.

- **Declarative cloud resource orchestration:** Cloud platform engineers can use declarative code to define cloud infrastructure resources and resource orchestration rules, and use GitOps to drive the execution of orchestration tasks. In this way, cluster lifecycle can be managed, and Day-2 operations can be completed.
 - **GitOps for application delivery:** By defining a more specific application release and delivery process using code, engineers can use Kubernetes and an automatic delivery pipeline to apply changes to any cluster, which ensures consistency user experience across clouds.
 - **Declarative application O&M:** Application resources and statuses are defined using code, and the orchestration engine drives automatic execution of application O&M.
- **Multi-Cloud Application Service Governance**

Figure 1-3 Multi-cloud application service governance



Pain Points:

Applications are deployed on heterogeneous clouds in distributed mode and consist of a large number of services. However, when services call each other, the paths are complex. Also, there is no efficient service governance.

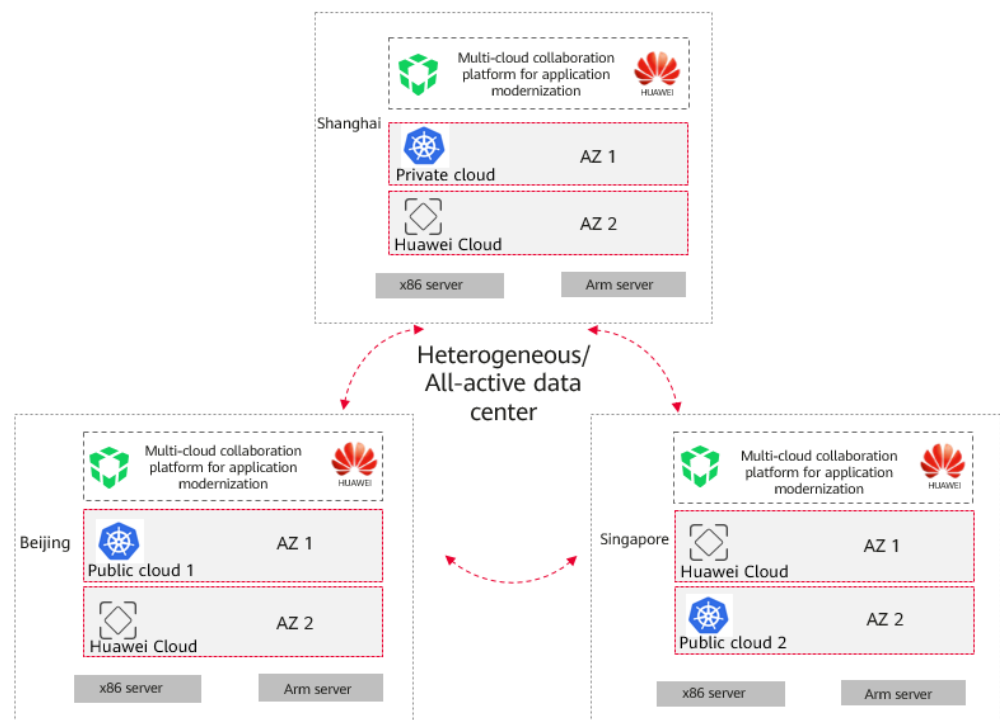
Countermeasures:

This solution enables unified governance of application services and provides highly compatible transparent communications, traffic management, dual-mode microservice governance, and end-to-end observation for applications.

- **Transparent and smooth communications:** Service meshes are used for cross-cloud and cross-cluster traffic management, such as consistent and transparent service discovery at the application layer, request routing, health checks, timeouts, retries, and rate limiting.

- **Dual-mode microservice governance:** Microservice governance is implemented using both traditional methods and service meshes.
- **End-to-end observation:** Applications and microservices are observed comprehensively, and application data is analyzed and displayed visually.
- **Cross-cloud service authentication and access control:** Service-based authentication and authorization make access to services controllable.
- **Cross-Cloud Service Continuity**

Figure 1-4 Cross-cloud service continuity: multi-active deployment and multi-cloud DR



Pain Points:

To ensure service continuity, cross-cloud application HA is required. However, challenges in technical complexity, scalability, availability, and elasticity must be addressed.

Countermeasures:

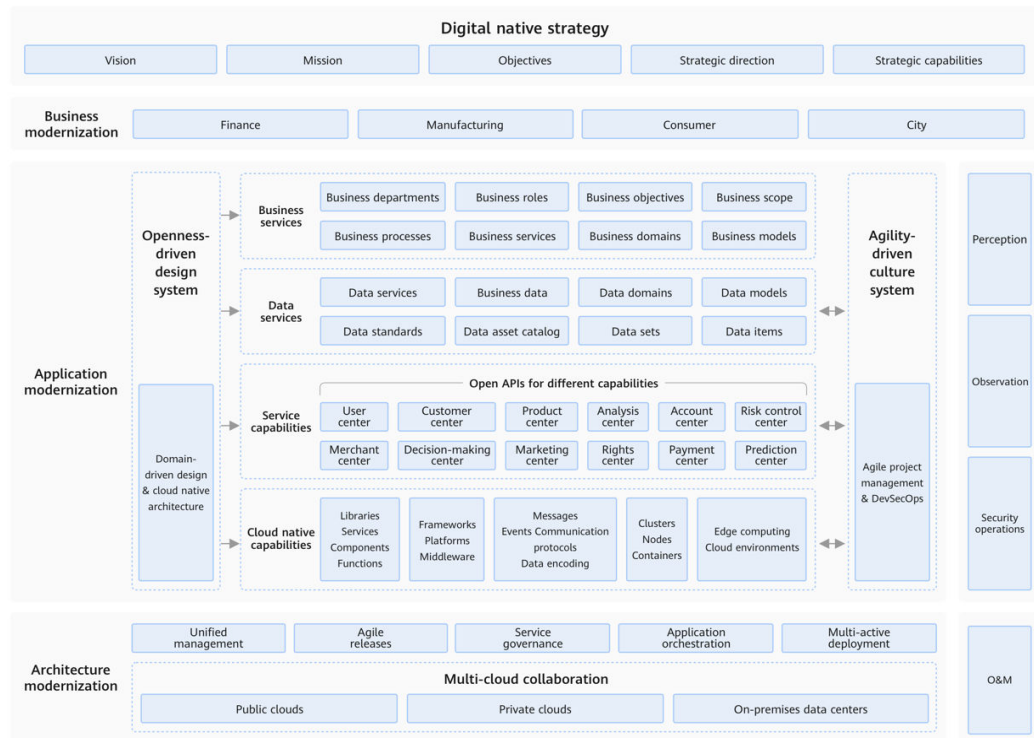
This solution allows you to deploy applications in clusters in different regions to prevent unavailability caused by faults in a single region. The resource management center of the multi-cloud collaboration platform automatically monitors the health of each cluster. If a single cloud environment becomes faulty, cross-cloud migration and traffic switchover can be quickly and automatically completed.

Solution Architecture

Service Architecture

The following figure shows the service architecture of this solution.

Figure 1-5 Service architecture

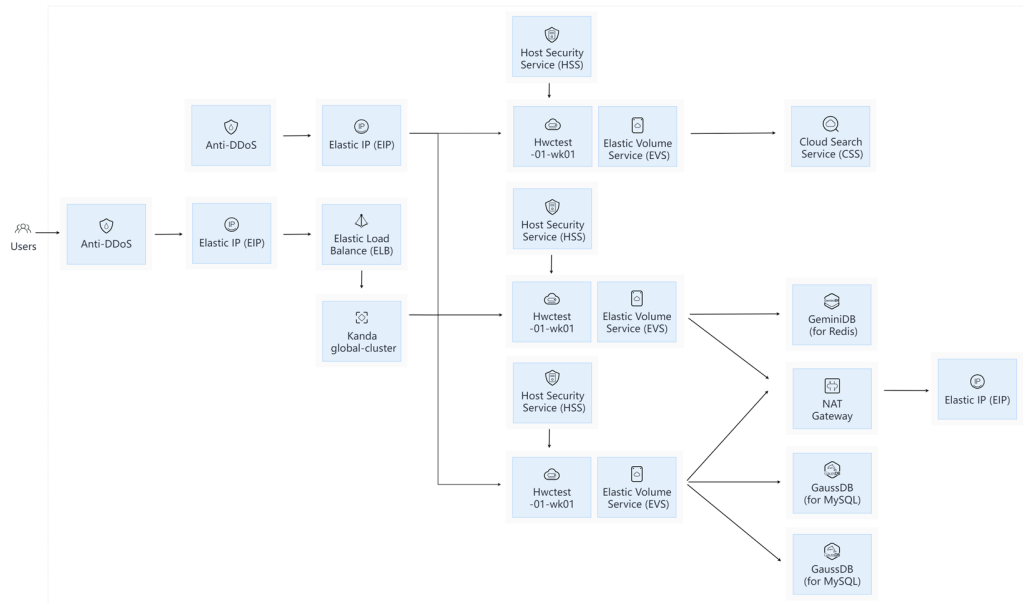


- The multi-cloud collaboration platform provided in this solution uses Huawei Cloud products, including Cloud Container Engine (CCE), SoftWare Repository for Container (SWR), Cloud Search Service (CSS), TaurusDB, GeminiDB (for Redis), Elastic Cloud Server (ECS), Host Security Service (HSS), NAT Gateway, Elastic Load Balance (ELB), Elastic IP (EIP), Scalable File Service (SFS), Elastic Volume Service (EVS), and Volume Backup Service (VBS).
- In typical scenarios, the software used in this solution includes Global Management, UI, Workbench, Container Management, Image Registry, Service Mesh, Microservice Engine, Insight (for observability), Multicloud Management (Karmada), Network, Storage, Elasticsearch middleware (CSS can be used), and MySQL middleware (TaurusDB can be used).
- These components run as containers and interact with each other through the network in the CCE cluster. Elasticsearch and MySQL provide data storage for the rest of the components. The components interact with CCE through the Network and Storage components. (The Network component automatically connects to the CCE network through Container Network Interface (CNI), and the Storage component automatically connects to the CCE storage through Container Storage Interface (CSI).)
- This solution provides security, monitoring, and HA for applications. You can also use Huawei Cloud services as required.
- Both administrators and tenants (O&M personnel, application development/release personnel, storage administrators, and network administrators) can use the multi-cloud collaboration platform.
- The clusters on public or private clouds or self-managed Kubernetes clusters are connected to the multi-cloud collaboration platform over the Internet or VPN.

Deployment Architecture

This deployment architecture uses Huawei cloud native services, data services, and basic cloud services to provide compute, storage, and networking resources. DaoCloud multi-cloud collaboration platform provides multi-cloud cluster management, multi-cloud application orchestration, and multi-cloud service meshes for application modernization.

Figure 1-6 Deployment architecture



- **Huawei Cloud products**

- The software used in this solution runs as containers on a CCE cluster consisting of ECSs.
- Some components (such as Multicloud Management and Container Management) need to communicate with the API Server of the managed clusters. These components can be scheduled to any node in the cluster. A NAT gateway is configured for these nodes to ensure external network connectivity. You can also configure security groups or other stricter access policies based on security requirements.
- Anti-DDoS monitors the traffic from the Internet to the public IP addresses of your servers in real time to detect DDoS attacks. It then scrubs attack traffic based on custom defense policies so that services run as normal. It also generates monitoring reports that provide visibility into network security.
- ELB distributes incoming traffic to multiple containers in the cluster to handle traffic spikes at different time, which expands the service capabilities of the multi-cloud collaboration platform.
- An EIP is bound to each load balancer and the NAT gateway so that the components in this solution can provide services accessible from the Internet.

- HSS is designed to protect server workloads in hybrid clouds and multi-cloud data centers. It protects servers and containers and prevents web pages from malicious modifications.
- EVS provides scalable block storage services with high reliability, high performance, and extensive specifications. EVS disks can be attached to pods for local storage.
- CSS provides ELK services for this solution.
- TaurusDB stores key service data.
- GeminiDB (for Redis) is used to cache data.
- **Software layer of the multi-cloud collaboration platform**
 - **Software:** Major components include Global Management, Container Management, Service Mesh, Insight, Multicloud Management, Image Registry, Microservice Engine, Workbench, Network, and Storage, and UI. These components run as containers. Compute, storage, and networking resources are provided by CCE. The Network component is connected to the CCE network through CNI. The Storage component interacts with CCE by using CSI, so that EVS disks can be attached to pods for local storage.
 - **O&M:** Both users and administrators can log in to the multi-cloud collaboration platform to perform operations. The platform O&M can be performed on the nodes in a CCE cluster. Insight (the component for observability) can be used to monitor application and resource usages in real time and configure alarm rules to detect exceptions in a timely manner.
 - **Security:** The container image security of each component is guaranteed by DaoCloud, and the container runtime security is guaranteed by a cloud native security module. Advanced Anti-DDoS (AAD) and Web Application Firewall (WAF) from Huawei Cloud, or third-party services can be used to provide additional security. For the underlying infrastructure, Huawei Cloud HSS or third-party services can be used to ensure security.
 - **HA deployment:** In this solution, platform components are deployed in the same AZ but have multiple replicas distributed on different nodes. TaurusDB is deployed in primary/standby mode, and the primary node and read replicas run in different AZs.
 - **Backups**
 - CSS backup:** CSS stores monitoring metrics, traces, and logs of the multi-cloud collaboration platform and platform application containers. The index data in a cluster is backed up to avoid data loss. If data loss occurs or data of a specified duration needs to be retrieved, users can use the index data to restore the data quickly. For details, see [Backup and Restoration Overview](#).
 - TaurusDB backup:** TaurusDB stores platform configuration data. Full backup and incremental backup are performed periodically. You can use a data backup to restore nodes to the state they were when the data backup was created. For details, see [Restoring Instance Data to a Specific Point in Time](#).
 - Multi-cloud collaboration platform backup:** The etcd of the CCE cluster is backed up periodically.
- **Key deployment points**

- At the access layer, Huawei Cloud ELB is used for load balancing. Users can access the multi-cloud collaboration platform over the Internet. For scenarios with high security requirements, VPN is recommended.
- There are no customer services and data in the VPC. Therefore, applications and data are not allocated to different subnets.
- In scenarios with high security requirements, a DMZ can be divided. The platform portal entrance is placed in the DMZ for O&M personnel and users to access. Users and O&M personnel can also access the platform through VPN.
- If the platform portal is exposed externally, Huawei Cloud WAF and AAD can be used for protection.
- The applications and data of the platform have multiple replicas that are distributed on different nodes in the CCE cluster for high availability and disaster recovery.
- Insight takes care of the platform O&M. In extreme cases, if the platform cannot be accessed, engineers can log in to the nodes in the CCE cluster for O&M.
- The middleware used by the platform includes Elasticsearch and MySQL, which can be replaced by CSS and TaurusDB.
- The storage component enables local volumes to be mounted to pods for data storage to meet the requirement of high-speed access.
- Huawei Cloud IoT and AI services are not involved.
- The multi-cloud collaboration platform provides identity authentication, access permissions, account management, and encryption. It also supports single sign-on (SSO) over LDAP and OIDC and allows users to use their existing account system by connecting to an identity provider.

Advantages

Solution Highlights:

- **Higher efficiency**
More than 500,000 nodes and 2 million pods are managed centrally. The cluster search performance is improved by 10 folds. The IT infrastructure investment is reduced by more than 50%.
- **Third-party cloud neutrality**
You can select cloud service providers based on your service requirements to reduce the dependence on cloud giants. There are no forbidden areas of cooperation with Huawei Cloud in the multi-cloud landscape and the independent controllable market.
- **Unique features**
Abundant features are provided, such as cross-cloud failover, dual-mode microservice governance, one-click upgrade of single-cloud applications to multi-cloud applications, cross-cloud application migration, cross-cloud application scheduling, and high-performance multi-cluster resource retrieval.
- **Technical guidance**
DaoCloud has core code maintenance capabilities and is one of the top contributors to core open-source projects. For instance, it ranked No. 2 in Karmada, No. 3 in Istio (a member of Steering Committee in 2022), and No. 3

in Kubernetes in the past year. DaoCloud also developed Clusterpedia (an open-source platform for resource search across clusters) and KWOK (a toolkit that enables setting up a cluster running thousands of nodes in seconds). Many projects have been patented.

- **Application modernization standards**

DaoCloud has released a white paper on the application modernization method system and participated in the compilation of the *Application Modernization Guide* and *Application Modernization Maturity Standards*.

Customer Benefits:

- **Consistent user experience**

Clusters of mainstream cloud vendors and open-source cloud native clusters can be managed centrally. Fine-grained permission management and multi-cloud O&M monitoring are provided to ensure a consistent user experience.

- **Distributed resource collaboration**

Collaborative scheduling and cross-cloud auto scaling of resources on more than 500,000 nodes, as well as cross-cloud failover enable special deployment of applications in different geographical locations.

- **Intelligent routing and elastic traffic management**

Cross-cloud and cross-cluster east-west communications, dual-mode microservice governance, grayscale releases, visualized traffic management, and various application traffic routing policies help manage applications more easily.

2 Resource and Cost Planning

The table below lists the resources required for deploying the multi-cloud collaboration platform.

 **NOTE**

You can adjust the planning as required.

Table 2-1 Resource and cost planning

Cloud Resource	Cloud Service	Specifications	Quantity	Price per Year (USD)
CCE cluster	CCE	CCE cluster 50 nodes HA	1	2,448.96
ECS	ECS	x86 General computing s3.4xlarge.2 16 vCPUs 32 GiB	3	8,763.84
(Optional) GaussDB database	TaurusDB	MySQL 8.0 Multi-AZ Dedicated x86 4 vCPUs and 16-GiB memory 2 nodes 100-GB storage	1	6,304.48
(Optional) GeminiDB database	GeminiDB (for Redis)	Redis 5.0 or earlier 12 GB Shared geminidb.redis.medium.4 (1 vCPU) 3 nodes	1	1,225.44
(Optional) ELK services	CSS	Elasticsearch 7.6.2 3 nodes General computing ess.spec-8u32g High I/O 500-GB storage	1	11,957.40

Cloud Resource	Cloud Service	Specifications	Quantity	Price per Year (USD)
Host security service	HSS	Enterprise edition	3	414.00
Cloud disk	EVS	General-purpose SSD 300 GB	3	1,080.00
NAT gateway	NAT Gateway	Small	1	889.87
Load balancer	ELB	Shared	2	928.56
EIP	EIP	Dedicated Dynamic BGP Bandwidth billed by fixed bandwidth 5 Mbit/s x 1 Dedicated Dynamic BGP Bandwidth billed by fixed bandwidth 1 Mbit/s x 3	4	912.00
Total Price	USD 34,924.54			

3 Operation Process

This section describes how to set up the multi-cloud collaboration platform on Huawei Cloud.

Figure 3-1 Process for setting up the multi-cloud collaboration platform

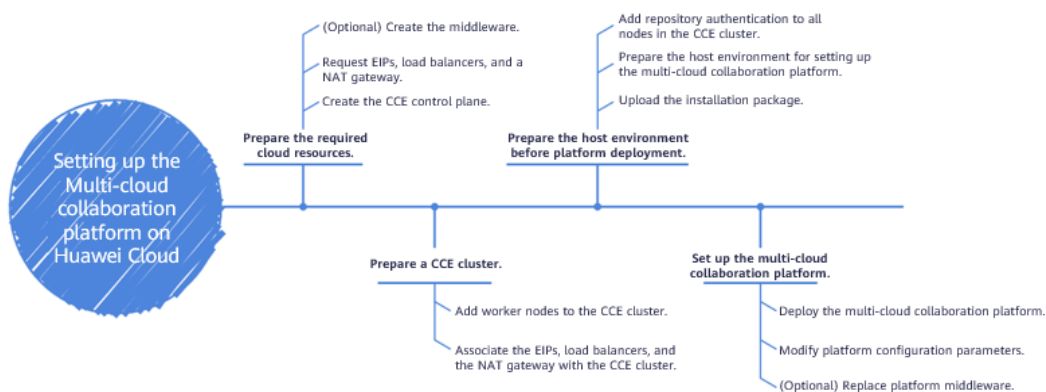


Table 3-1 Process description

No.	Step	Description
1	Prepare the required cloud resources.	Prepare the environment on Huawei Cloud: 1. Create the CCE control plane. 2. Request EIPs, load balancers, and a NAT gateway. 3. Create the middleware (MySQL, Redis, and Elasticsearch).
2	Prepare a CCE cluster.	Configure the CCE control plane: 1. Add worker nodes to the CCE cluster. 2. Bind an EIP to a worker node in the CCE cluster, an EIP to each load balancer, and an EIP to the NAT gateway. Associate the NAT gateway with the remaining worker nodes in the cluster.

No.	Step	Description
3	Prepare the host environment before platform deployment.	Prepare the environment on the CCE worker nodes: <ol style="list-style-type: none">1. Upload the multi-cloud collaboration platform installation package.2. Grant the kubectrl permission, modify the deployment configuration file, and install and deploy dependencies.3. Add a private image repository credential for all nodes in the CCE cluster.
4	Set up the multi-cloud collaboration platform.	Deploy the platform: <ol style="list-style-type: none">1. Run commands to deploy the platform.2. Modify platform parameters (such as the ELB address).3. Replace the platform middleware with Huawei Cloud products.

4 Implementation Procedure

- 4.1 Preparing the CCE Environment
- 4.2 Deploying the Multi-Cloud Collaboration Platform
- 4.3 (Optional) Changing the Middleware

4.1 Preparing the CCE Environment

Creating a CCE Cluster on Huawei Cloud

Perform the following operations to create a CCE cluster:

Step 1 Create a VPC.

On the Network Console, choose **Virtual Private Cloud > My VPCs** and click **Create VPC**.

Configure the parameters as required.

Step 2 Create a CCE cluster.

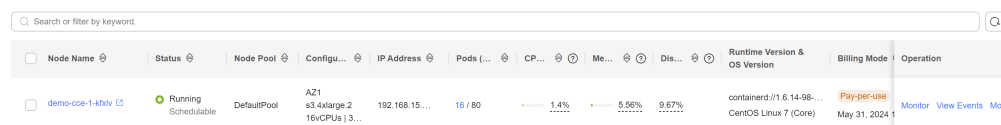
Create a CCE cluster using the VPC created in the previous step.

Confirm the specifications and complete the creation.

Step 3 Add worker nodes to the created CCE cluster.

Go to the CCE cluster details page and add nodes.

Figure 4-1 Adding worker nodes to the CCE cluster



Node Name	Status	Node Pool	Configu...	IP Address	Pods (...)	CP...	Me...	Dis...	Runtime Version & OS Version	Billing Mode	Operation
demo-cce-1-k8sv C	Running Schedulable	DefaultPool	AZ1 s3.4xlarge.2 16vCPUs 3...	192.168.15...	16 / 80	1.4%	5.56%	9.67%	containerd/v1.6.14-98... CentOS Linux 7 (Core)	Pay-per-use	Monitor View Events More

Confirm the resources.

----End

Requesting EIPs on Huawei Cloud

On the Network Console, choose **Elastic IP and Bandwidth > EIPs** and click **Buy EIP**.

Four EIPs are required.

- Two EIPs are required for setting up the multi-cloud collaboration platform:
 - One EIP is bound to a load balancer for accessing the multi-cloud collaboration platform.
 - The other EIP is bound to a load balancer for viewing the platform monitoring data.
- Two EIPs are required for the nodes in the CCE cluster to access the Internet.

Creating Load Balancers on Huawei Cloud

Perform the following operations to create two load balancers, each with an EIP bound:

Step 1 On the Network Console, choose **Elastic Load Balance > Load Balancers** and click **Buy Elastic Load Balancer**.

Step 2 Bind an EIP to each load balancer.

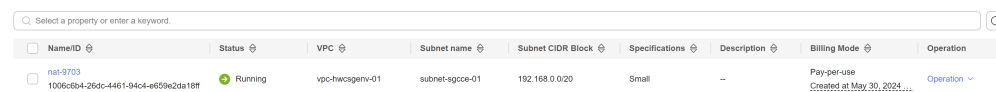
----End

Configuring NAT Gateway on Huawei Cloud

Perform the following operations to create a public NAT gateway with an EIP bound:

Step 1 Buy a public NAT gateway.

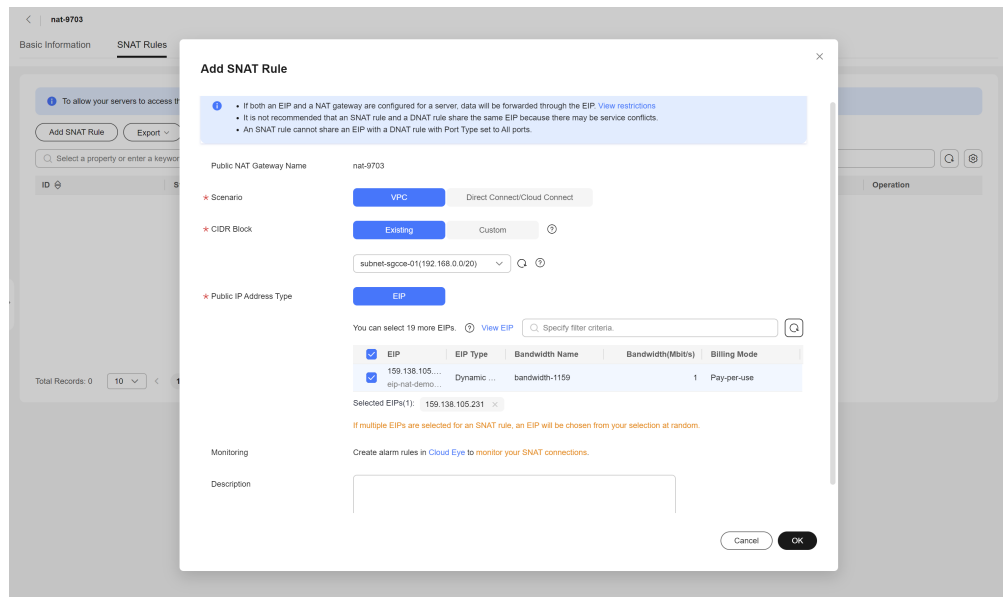
Figure 4-2 Configuring NAT gateway



NameID	Status	VPC	Subnet name	Subnet CIDR Block	Specifications	Description	Billing Mode	Operation
nat-9703 1006c804-26dc-4461-94c4-e659e2da18ff	Running	vpc-hwccserv-01	subnet-aggce-01	192.168.0.0/20	Small	--	Pay-per-use Created at May 30, 2024...	Operation

Step 2 Add an SNAT rule to the public NAT gateway to bind an EIP to this gateway so the nodes without EIPs in the CCE cluster can share the EIP to access the Internet.

Figure 4-3 Adding an SNAT rule



Step 3 Use any node that has no EIP bound to access the Internet to test network connectivity.

----End

(Optional) Deploying Middleware on Huawei Cloud

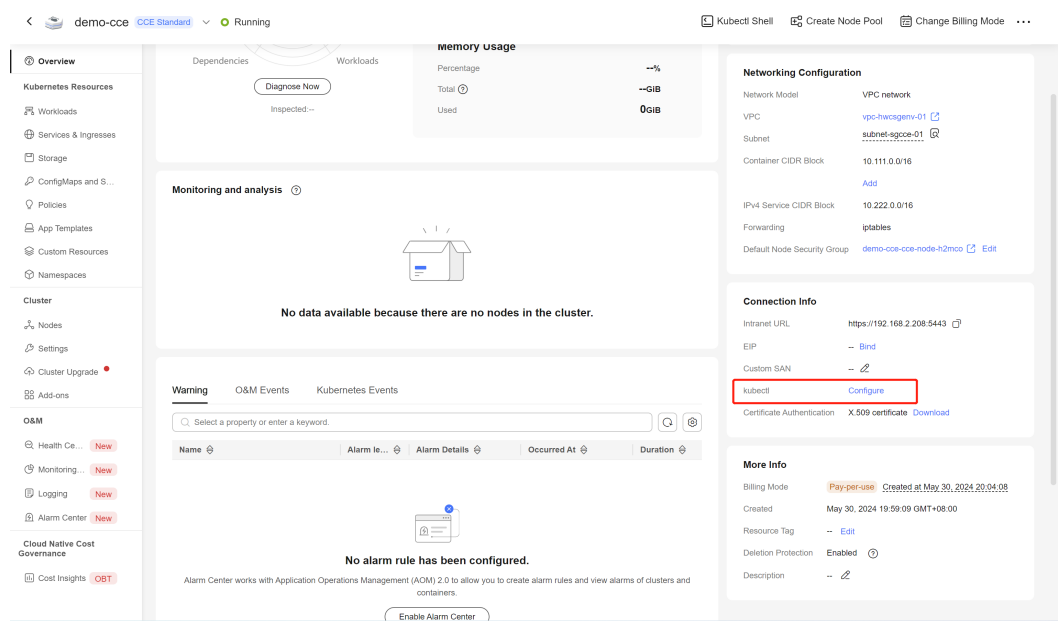
After the multi-cloud collaboration platform is set up, containerized middleware is automatically deployed. In the production environment, Huawei Cloud products are recommended. The middleware is independent of the platform to improve platform stability.

- Deploy TaurusDB.
On the Huawei Cloud management console, select TaurusDB and create a DB instance.
- Deploy GeminiDB (for Redis).
On the Huawei Cloud management console, select GeminiDB and create a DB instance with **Compatible API** set to **Redis**.
- Deploy CSS.
On the Huawei Cloud management console, select CSS and create an Elasticsearch cluster.

Granting kubectl Permissions on the Worker Nodes in the CCE Cluster

Grant the kubectl operation permissions on the worker nodes in the CCE cluster by following the instructions on the CCE console.

Figure 4-4 Granting kubectl permissions on worker nodes



4.2 Deploying the Multi-Cloud Collaboration Platform

Installing the Multi-Cloud Collaboration Platform

Install the multi-cloud collaboration platform using the CCE cluster.

Prerequisites

- The CCE version is v1.22.x or later.
- A private image repository whose cluster network is reachable and storage is greater than 50 GB is available.

Procedure

Step 1 Log in to a node in the CCE cluster.

Step 2 Download the installation package of the latest version from the [Download Center](#).

CPU architecture	Version	Download URL
x86-64 (also known as AMD64)	v0.10.0	/
Arm64	v0.10.0	/

Step 3 Decompress the installation package.

```
##Command for decompressing the installation package of the AMD64 architecture:
tar -xvf offline-v0.10.0-amd64.tar
```

Step 4 (Optional) Install nerdctl. (If containerd is used as the container runtime, install nerdctl on each worker node in the CCE cluster.)

```
wget https://github.com/containerd/nerdctl/releases/download/v1.4.0/nerdctl-1.4.0-linux-amd64.tar.gz
tar -zxvf nerdctl-1.4.0-linux-amd64.tar.gz
cp -a nerdctl /usr/local/bin/
```

Step 5 Obtain the cluster configuration file **clusterConfig.yaml** from the **offline/sample** directory and modify the file as required.

Example configuration:

```
apiVersion: provision.daocloud.io/v1alpha3
kind: ClusterConfig
metadata:
  creationTimestamp: null
spec:
  loadBalancer:
    type: cloudLB # Huawei Cloud ELB is recommended.
  istioGatewayVip: 10.5.14.XXX/32 (an EIP)
  insightVip: 10.5.14.XXX/32 (another EIP)
  fullPackagePath: /home/offline # Directory where the installation package is stored
  imagesAndCharts:
    type: external
  externalImageRepo: http://release.daocloud.io # Private image repository address
```

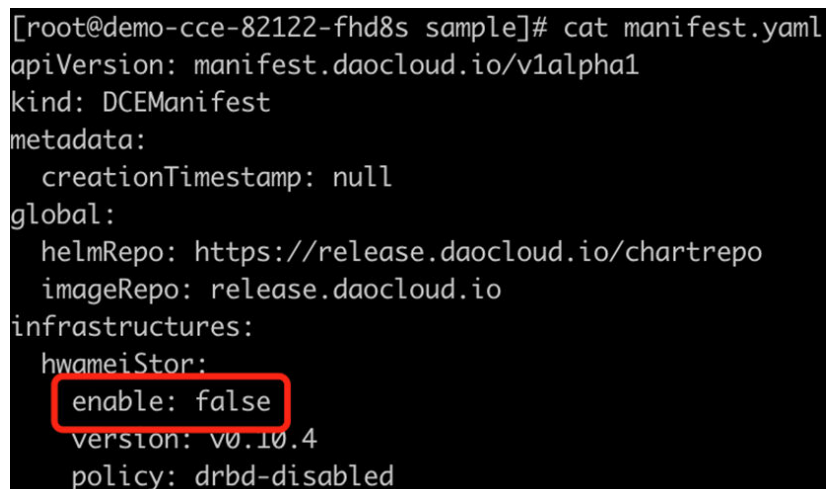
Step 6 (Optional) Configure the manifest file. You can obtain the file from the **offline/sample** directory and modify the file as required.

To enable HwameiStor, ensure that no default StorageClass exists in the cluster. If a default StorageClass exists, delete the configuration of the default StorageClass.

If HwameiStor is not enabled, you can use the StorageClass automatically created by CCE. In this case, perform the following steps:

1. Change the automatically created csi-disk to the default StorageClass.
kubectyl patch storageclass csi-disk -p '{"metadata": {"annotations": {"storageclass.kubernetes.io/is-default-class": "true"}}}'
2. Modify the **/offline/sample/manifest.yaml** file to disable HwameiStor.

Figure 4-5 Modifying the /offline/sample/manifest.yaml file



```
[root@demo-cce-82122-fhd8s sample]# cat manifest.yaml
apiVersion: manifest.daocloud.io/v1alpha1
kind: DCManifest
metadata:
  creationTimestamp: null
global:
  helmRepo: https://release.daocloud.io/chartrepo
  imageRepo: release.daocloud.io
infrastructures:
  hwameiStor:
    enable: false
    version: v0.10.4
    policy: drbd-disabled
```

Step 7 Perform the private image repository authentication on all worker nodes.

Docker: Add authentication to the **daemon.json** file.

containerd: Add authentication to the containerd configuration file.

Step 8 Install the multi-cloud collaboration platform.

1. Install dependencies.

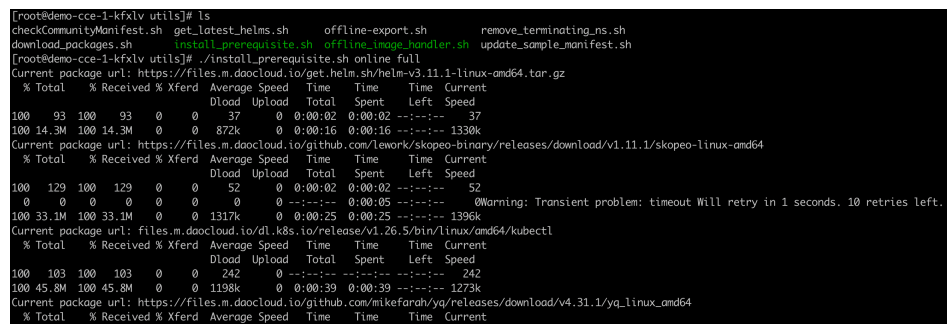
Go to the **offline** directory and install the dependencies.

```
cd /xxx/offline/utis
./install_prerequisite.sh online full
```

Wait until the installation is complete.

2. Install the multi-cloud collaboration platform.

Figure 4-6 Installing the multi-cloud collaboration platform



```
/offline/dce5-installer install-app -m ./offline/sample/manifest.yaml -c ./offline/sample/clusterConfig.yaml
```

NOTE

Obtain parameter details through **./dce5-installer --help**.

- -z indicates the minimum installation.
- -c specifies the cluster configuration file. You do not need to specify -c when NodePort is used for communications.
- -d enables the debug mode.
- -m specifies the manifest file used for installation.
- --serial indicates that after this parameter is specified, all installation tasks are executed in serial mode.

Step 9 When a message indicating that the installation is successful, use the URL displayed on the screen and the default username (**admin**) and password (**changeme**) to log in to the multi-cloud collaboration platform.

Figure 4-7 Default username (admin) and password (changeme)

```
S:18PM: TEST SUITE: None
S:18PM: NOTES:
S:18PM:
S:18PM:
S:18PM:
S:18PM:
S:18PM:
S:18PM:
S:18PM:
S:18PM:
S:18PM:
S:18PM: 1. Get the Insight UI URL, by running these commands:
S:18PM: export POD_NAME=$(kubectl get pods --namespace insight-system -l "app.kubernetes.io/name=ui,app.kubernetes.io/instance=insight" -o jsonpath="{.items[0].metadata.name}")
S:18PM: export CONTAINER_PORT=$(kubectl get pod --namespace insight-system $POD_NAME -o jsonpath="{.spec.containers[0].ports[0].containerPort}")
S:18PM: kubectl --namespace insight-system port-forward $POD_NAME 8080:$CONTAINER_PORT
S:18PM: echo "Visit http://127.0.0.1:8080 to use your application"
S:18PM: Install IPava Dashboard v0.4.1
S:18PM: "ipava" already exists with the same configuration, skipping
S:18PM: Hang tight while we grab the latest from your chart repositories...
S:18PM: ...Successfully got an update from the "ipava" chart repository
S:18PM: Update Complete. #Happy Helming!
S:18PM: Release "ipava" does not exist. Installing it now.
S:18PM: NAME: ipava
S:18PM: LAST DEPLOYED: Mon Sep 19 17:18:37 2022
S:18PM: NAMESPACE: ipava-system
S:18PM: STATUS: deployed
S:18PM: REVISION: 1
S:18PM: NOTES:
S:18PM: 1. Get the application URL by running these commands:
S:18PM: Install Insight Agent v0.9.4
S:18PM: "insight" already exists with the same configuration, skipping
S:18PM: Hang tight while we grab the latest from your chart repositories...
S:18PM: ...Successfully got an update from the "insight" chart repository
S:18PM: Update Complete. #Happy Helming!
S:18PM: Release "insight-agent" does not exist. Installing it now.
S:18PM: NAME: insight-agent
S:18PM: LAST DEPLOYED: Mon Sep 19 17:19:01 2022
S:18PM: NAMESPACE: insight-system
S:18PM: STATUS: deployed
S:18PM: REVISION: 1
S:18PM: -----
S:18PM: [!] Use your Web Browser to access DaoCloud Enterprise 5th Portal at : http://172.30.47.104:31227
S:18PM: -----
S:18PM: All set. Enjoy your journey to cloud native with DaoCloud Enterprise 5th !
S:18PM: -----
[root@localhost ~]#
```

Step 10 Use the obtained license to activate the multi-cloud collaboration platform.

----End

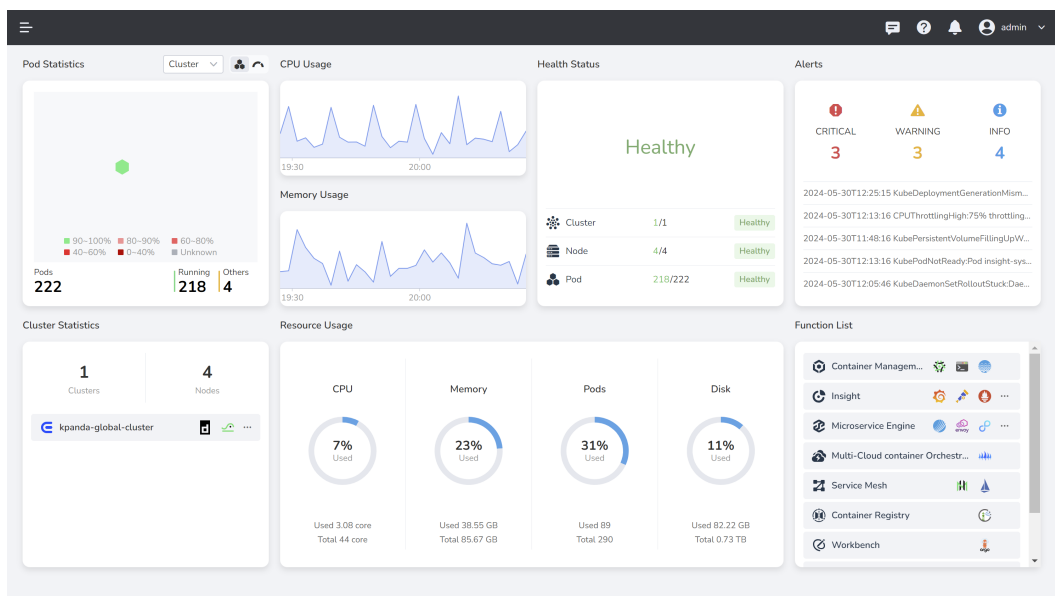
Verifying the Multi-Cloud Collaboration Platform

Verify that the multi-cloud collaboration platform is ready after it is deployed.

Log in to the platform as prompted and complete the activation.

If the GUI functions are normal and no error is reported, the platform is ready.

Figure 4-8 Verifying the multi-cloud collaboration platform



4.3 (Optional) Changing the Middleware

Change the middleware of the multi-cloud collaboration platform to Huawei Cloud products.

Prerequisites

- MySQL, Redis, and Elasticsearch provided by Huawei Cloud are available.
- The middleware and the CCE cluster can access each other.

NOTE

After the middleware is changed, data is automatically initialized and stored in the new middleware.

- Modify Ipavo configuration:
 - a. Create the ipavo database and the login user **ipavo** in the MySQL database.
 - b. Change the middleware address (MySQL) in the Ipavo configuration file and restart the Ipavo service.
- Modify Ghippo configuration:
 - a. Create the ghippo, audit, and keycloak databases in the MySQL database, and the login users **ghippo**, **audit**, and **keycloak**.
 - b. Change the middleware address (MySQL) in the Ghippo configuration file and restart the Ghippo service.
- Modify Kpanda configuration:
 - a. Create the kpanda database and the login user **kpanda** in the MySQL database.
 - b. Change the middleware addresses (MySQL and Redis) in the Kpanda configuration file and restart the Kpanda service.
- Modify Insight configuration:
 - a. Create the insight database and the login user **insight** in the MySQL database.
 - b. Change the middleware addresses (MySQL and Elasticsearch) in the Insight configuration file and restart the Insight service.
- Modify Skoala configuration:
 - a. Create the skoala database and the login user **skoala** in the MySQL database.
 - b. Change the middleware address (MySQL) in the Skoala configuration file and restart the Skoala service.
- Modify Amamba configuration:
 - a. Create the amamba database and the login user **amamba** in the MySQL database.
 - b. Change the middleware address (MySQL) in the Amamba configuration file and restart the Amamba service.

5 Appendixes

Context

Table 5-1 Common Terms

Acronym/ Abbreviation	Term	Description	Component
CR	container runtime	A container runtime is a component that is responsible for managing the execution and lifecycle of containers within the environment.	Docker containerd
SC	StorageClass	A StorageClass provides a way for administrators to describe the classes of storage they offer.	NFS hostPath Disk
CNI	Container Network Interface	Container Network Interface (CNI) plugins comply with Advanced Program-to Program Communications (APPC) or CNI and are crucial for cluster networking.	Calico Cilium Macvlan
VPC	Virtual Private Cloud	A VPC isolates a private virtual network for cloud resources, such as cloud servers, containers, and databases.	N/A
Registry	image repository	A container image repository is a vault (or repository collection) used to store Kubernetes container images, container images developed using DevOps, and container images created from containerized applications.	Private repository Public repository

Troubleshooting

- **The multi-cloud collaboration platform cannot be accessed using an EIP (bound to the virtual IP address of istio-ingressgateway)**

When you are accessing the multi-cloud collaboration platform using an EIP, the page is not displayed, and you are redirected to a private IP address of the VPC.

If this happens, you need to replace and modify the address information in the Ghippo module.

```
helm -n ghippo-system get values ghippo > /tmp/ghippo
helm -n ghippo-system upgrade ghippo ghippo/ghippo --values=/tmp/ghippo --set
global.reverseProxy=https://119.x.x.x (This is the EIP bound to the virtual IP address of istio-
ingressgateway.)
```

- **istio-ingressgateway cannot be started when the cluster (or any worker node) is restarted**

Figure 5-1 Error message

```
2022-09-06T10:08:21.729574Z    warning envoy config    gRPC config for type.googleapis.com/envoy.config.listener.v3.Listener rejected: Error adding/updat
ing listener(s) 0.0.0.0:8080: Provider 'origins-0' in jwt_authn config has invalid local jwks: Jwks RSA [n] or [e] field is missing or has a parse error
2022-09-06T10:08:22.691699Z    warn    Envoy proxy is NOT ready: config received from XDS server, but was rejected: cds updates: 1 successful, 0 rejected
; lds updates: 0 successful, 1 rejected
2022-09-06T10:08:24.684404Z    warn    Envoy proxy is NOT ready: config received from XDS server, but was rejected: cds updates: 1 successful, 0 rejected
; lds updates: 0 successful, 1 rejected
2022-09-06T10:08:26.684665Z    warn    Envoy proxy is NOT ready: config received from XDS server, but was rejected: cds updates: 1 successful, 0 rejected
; lds updates: 0 successful, 1 rejected
2022-09-06T10:08:28.684081Z    warn    Envoy proxy is NOT ready: config received from XDS server, but was rejected: cds updates: 1 successful, 0 rejected
; lds updates: 0 successful, 1 rejected
2022-09-06T10:08:30.684783Z    warn    Envoy proxy is NOT ready: config received from XDS server, but was rejected: cds updates: 1 successful, 0 rejected
```

Possible cause: The jwtUri address of the RequestAuthentication ghippo CR cannot be accessed. As a result, Istiod cannot deliver configurations to the istio-ingressgateway.

Solution:

- Back up RequestAuthentication ghippo CR.
`kubectl get RequestAuthentication ghippo -n istio-system -o yaml > ghippo-ra.yaml`
 - Delete RequestAuthentication ghippo CR.
`kubectl delete RequestAuthentication ghippo -n istio-system`
 - Restart Istio.
`kubectl rollout restart deploy/istiod -n istio-system`
`kubectl rollout restart deploy/istio-ingressgateway -n istio-system`
 - Apply RequestAuthentication ghippo CR again.
`kubectl apply -f ghippo-ra.yaml`
 ## Before applying RequestAuthentication ghippo CR, ensure that ghippo-apiserver and ghippo-keycloak have been started.
- **MySQL becomes faulty.**

An error is reported when the CR fails to create a database.

The database is running normally. An error is reported when the CR is used to create the database. The cause is that the password of the MySQL root user contains special characters.

Figure 5-2 Error reported when the CR fails to create a database

```
10216 03:10:58.392145    1 orchestrator_reconcile.go:548] orchestrator-reconciler "msg"=skip set read-only/writable "key"=["Namespace":"ncamel-system", "Name":"ncamel-common-mysql-cluster"] "instance"
="["Namespace":"ncamel-common-mysql-cluster-mysql-1.mysql.ncamel-system", "IsUpToDate":false, "MasterHostname":""]
10216 03:11:19.075301    1 orchestrator_reconcile.go:548] orchestrator-reconciler "msg"=skip set read-only/writable "key"=["Namespace":"ncamel-system", "Name":"ncamel-common-mysql-cluster"] "instance"
="["Namespace":"ncamel-common-mysql-cluster-mysql-1.mysql.ncamel-system", "IsUpToDate":false, "MasterHostname":""]
10216 03:11:21.209760    1 deleg.go:130] controller_mysql-database "msg"=creating MySQL database "database"=insight "name"=insight-database
10216 03:11:21.245071    1 controller.go:304] controller_mysql-database "msg"=Reconciler error "error"=failed to create database, err: Error 1045: Access denied for user 'root'@'10.129.2.164' (using
password: 'HES3') "name"=insight-database "namespace"=ncamel-system
10216 03:11:38.372228    1 orchestrator_reconcile.go:548] orchestrator-reconciler "msg"=skip set read-only/writable "key"=["Namespace":"ncamel-system", "Name":"ncamel-common-mysql-cluster"] "instance"
="["Namespace":"ncamel-common-mysql-cluster-mysql-0.mysql.ncamel-system", "IsUpToDate":false, "MasterHostname":"ncamel-common-mysql-cluster-mysql-1.mysql.ncamel-system"]
10216 03:12:13.670017    1 orchestrator_reconcile.go:548] orchestrator-reconciler "msg"=skip set read-only/writable "key"=["Namespace":"ncamel-system", "Name":"ncamel-common-mysql-cluster"] "instance"
="["Namespace":"ncamel-common-mysql-cluster-mysql-0.mysql.ncamel-system", "IsUpToDate":false, "MasterHostname":"ncamel-common-mysql-cluster-mysql-1.mysql.ncamel-system"]
10216 03:12:33.383554    1 orchestrator_reconcile.go:548] orchestrator-reconciler "msg"=skip set read-only/writable "key"=["Namespace":"ncamel-system", "Name":"ncamel-common-mysql-cluster"] "instance"
="["Namespace":"ncamel-common-mysql-cluster-mysql-0.mysql.ncamel-system", "IsUpToDate":false, "MasterHostname":"ncamel-common-mysql-cluster-mysql-1.mysql.ncamel-system"]
10216 03:12:33.383677    1 orchestrator_reconcile.go:548] orchestrator-reconciler "msg"=skip set read-only/writable "key"=["Namespace":"ncamel-system", "Name":"ncamel-common-mysql-cluster"] "instance"
```

Solution:

a. View the password.

```
[root@master-01 ~]$ kubectl get secret -n mcamel-system mcamel-common-mysql-cluster-secret -o=jsonpath='{.data.ROOT_PASSWORD}' | base64 -d
```

If the password contains hyphens (-), the following error message is displayed when you enter the password in the MySQL shell:

```
bash-4.4# mysql -uroot -p
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
```

b. Perform the clean-up and rebuilding.

- Method 1: Clear the data directory, delete the pod, and wait until the sidecar is running. Then delete the data directory and the pod again.

```
[root@master-01 ~]# kubectl exec -it mcamel-common-mysql-cluster-mysql-1 -n mcamel-system -c sidecar -- /bin/sh
sh-4.4# cd /var/lib/mysql
sh-4.4# ls | xargs rm -rf
```

- Method 2: Delete the PVC and then delete the pod.

```
kubectl delete pvc data-mcamel-common-mysql-cluster-mysql-1 -n mcamel-system
kubectl delete pod mcamel-common-mysql-cluster-mysql-1 -n mcamel-system
```

6 Change History

Table 6-1 Change History

Release On	Description
2024-06-03	This issue is the first official release.