Lightweight GaussDB
24.1.30

Installation Guide

Issue 01

Date 2024-04-30

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.

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: https://www.huaweicloud.com/intl/en-us/

About This Document

Introduction

Lightweight GaussDB includes the GaussDB kernel, Data Replication Software (DRS), and GaussDB Management Platform (TPOPS).

Intended Audience

This document applies to the installation of GaussDB Management Platform (TPOPS). You need to be familiar with the following knowledge:

- Computer principles
- Linux operating systems
- Windows operating systems
- Network communications

Symbol Conventions

The symbols that may be found in this document are as follows.

Symbol	Description
DANGER	Indicates an imminently hazardous situation which, if not avoided, will result in serious injury or death.
WARNING	Indicates a potentially hazardous situation which, if not avoided, could result in serious injury or death.
CAUTION	Indicates a potentially hazardous situation which, if not avoided, could result in minor or moderate injury.
NOTICE	Indicates a potentially hazardous situation which, if not avoided, could result in equipment damage, data loss, performance deterioration, or other unanticipated results. NOTICE is used to address practices not related to personal injury.

Symbol	Description
NOTE	Calls attention to important information, best practices, and tips. NOTE is used to address information not related to personal injury, equipment damage, and environment deterioration.

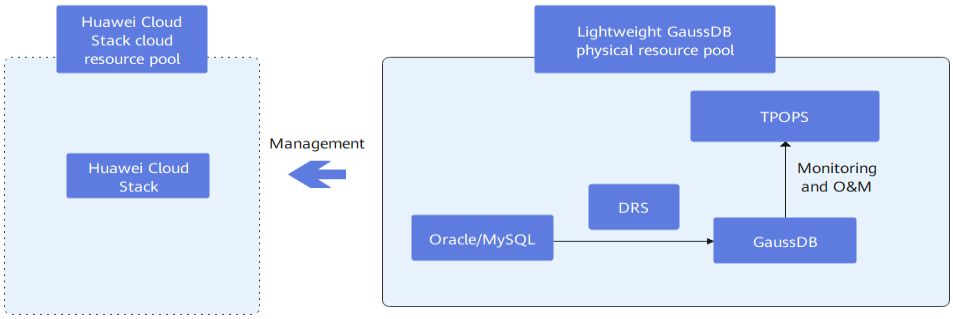
Contents

1 What Is Lightweight GaussDB?

Lightweight GaussDB is based on the Huawei Cloud Stack standard full-stack solution. It includes the GaussDB database kernel, Data Replication Software (DRS), and GaussDB Management Platform (TPOPS).

- GaussDB is an enterprise-grade, distributed relational database. It is designed based on the shared-nothing architecture and supports both x86 and Arm CPU architectures. It features the following: high-throughput, strong-consistency transaction processing capabilities, financial-grade HA, high scalability with a distributed architecture, and high-performance big data query capabilities. GaussDB is suitable for core systems used in a wide range of sectors, such as finance, telecom, and government.
- DRS is a stable, efficient, and easy-to-use cloud service for real-time database migration and synchronization. It helps you easily migrate local data to GaussDB.
- GaussDB Management Platform (TPOPS) is a database O&M management platform based on Huawei Cloud Stack Database Service (DBS). It is stable, reliable, and easy to use. With GaussDB Management Platform (TPOPS), you can obtain the consistent user experience as that on Huawei Cloud Stack without relying on Huawei Cloud Stack.

Figure 1-1 GaussDB lightweight deployment



2 Deployment Solutions

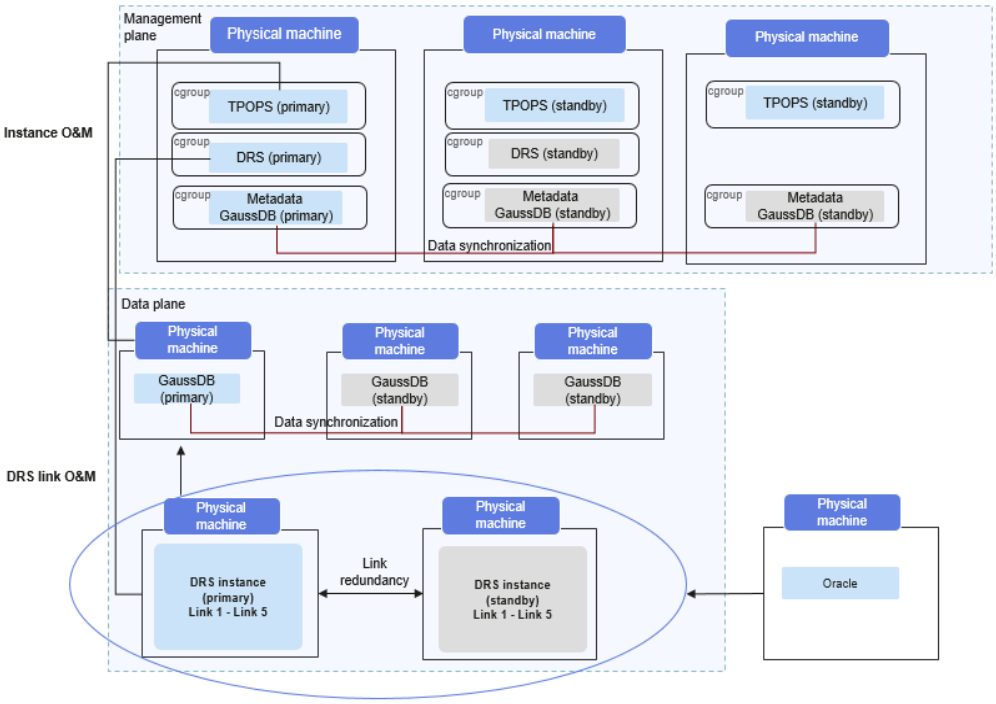
2.1 Management Plane Deployment Solution

2.1.1 Single-DC Deployment

- In commercial scenarios, GaussDB Management Platform (TPOPS) must be deployed on three servers.
- Up to 500 physical machines can be managed in standard specifications.
- GaussDB Management Platform (TPOPS) can be deployed independently or together with the DRS management plane.
- DB instances can be deployed on physical machines.

Figure 2-1 shows the deployment architecture of lightweight GaussDB management plane.

Figure 2-1 Single-DC deployment architecture

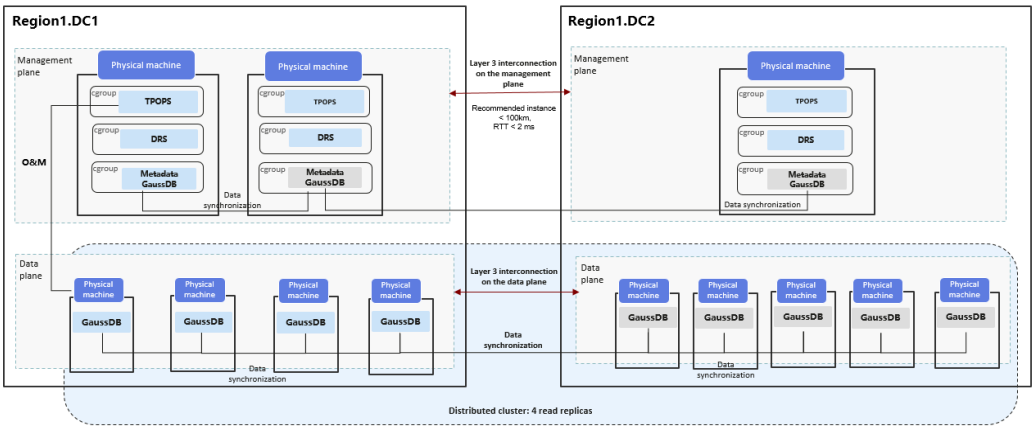


2.1.2 Intra-City Dual-DC Deployment

- At least three management nodes are required (two in the primary DC and one in the standby DC).
- If two management nodes are faulty, the GaussDB Management Platform (TPOPS) will be unavailable.
- An automatic switchover is triggered upon single-node faults in the primary DC.
- If a standby DC is faulty, services are not affected.
- GaussDB Management Platform (TPOPS) can be deployed independently or together with the DRS management plane.
- On GaussDB Management Platform (TPOPS), the SFTP service is deployed on two nodes. When the primary SFTP node is faulty, you cannot upload packages using the installation package management function. The query and download functions are not affected.

Figure 2-2 shows the deployment architecture of lightweight GaussDB management plane.

Figure 2-2 Intra-city dual-DC deployment architecture



NOTICE

In intra-city dual-DC HA scenarios, the majority of management plane nodes need to be deployed in the DC where the minority of nodes reside. When a fault occurs in the DC where the majority of nodes reside, the management plane delivers the request for forcibly starting the AZ where the minority of instances reside.

2.1.3 Optimal Configuration Requirements for GaussDB Management Platform (TPOPS)

The GaussDB management platform and DRS can be deployed separately or together. The configuration requirements for these scenarios are as follows:

- Independent deployment scenario: For details about the GaussDB management platform deployment requirements, see **Hardware Requirements for TPOPS Separately Deployed**. For details about the DRS deployment requirements, see the section about the OS configuration in installation preparations in **Installation Guide**.
- Integrated deployment scenario: For details about the GaussDB management platform deployment requirements, see **Hardware Requirements for TPOPS Deployed with DRS**.

Supported Display Specifications

GaussDB Management Platform (TPOPS) supports 1920 x 1080, 1920 x 1200, and higher resolutions (100% scaling).

2.2 GaussDB Distributed Deployment

2.2.1 Deployment Model for GaussDB Distributed Instances

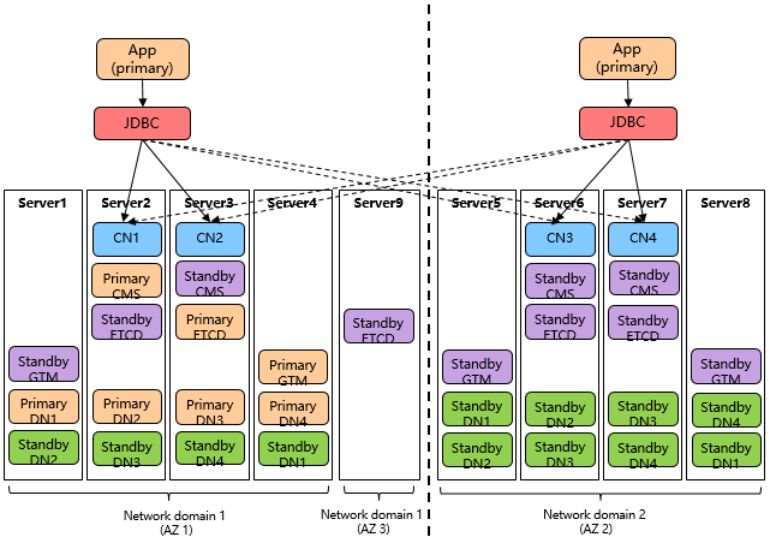
Deployment Model	Node	Shard	AZ	Description
Intra-city HA	9	4	2 + 1 (arbitration AZ)	Four nodes per AZ are symmetrically deployed, plus one quorum AZ (4+4+1).
	3	3	3	Each AZ deploys the primary DN replica on one shard, and standby DN replicas on the other two shards.
	3	3	1	Three nodes are deployed in one AZ.
	5	4	2 + 1 (arbitration AZ)	Two nodes per AZ are symmetrically deployed, plus one quorum AZ (2+2+1).
Intra-city HA + remote DR	4	4	2	Intra-city single-AZ deployment with three nodes (3 CNs + 3 DNs); remote single-AZ deployment with one node
	4	4	4	Intra-city 3-AZ deployment with three nodes for each AZ (3 CNs + 3 DNs); remote single-AZ deployment with one node
	6	6	2	Intra-city single-AZ deployment with three nodes (3 CNs + 3 DNs); remote single-AZ deployment with three nodes (3 CNs + 3 DNs)
	6	6	4	Intra-city 3-AZ deployment with three nodes for each AZ (3 CNs + 3 DNs); remote single-AZ deployment with three nodes (3 CNs + 3 DNs)
	9	8	3+1 (quorum AZ)	Intra-city 3-AZ deployment with five nodes for each AZ (4 CNs + 4 DNs); remote single-AZ deployment with four nodes (2 CNs + 4 DNs)
	13	8	3+1 (quorum AZ)	Intra-city 3-AZ deployment with nine nodes for each AZ (4 CNs + 4 DNs); remote single-AZ deployment with four nodes (2 CNs + 4 DNs)

2.2.2 Intra-City HA Deployment Scenarios

2.2.2.1 Intra-City, Dual-AZ, 9-Node Deployment (4 CNs + 4 DNs + 1 Quorum Node)

A complete intra-city active-active deployment solution consists of two service AZs and one quorum AZ. Two service AZs are deployed in peer-to-peer mode, and all data centers in the AZs can process services. The quorum AZ is responsible for auxiliary quorum, but cannot access services. This can avoid single points of failure (SPOFs) of any DB instance, AZ faults, and network faults between DCs. GaussDB also supports 4-replica (one primary and three standby DNs) and 1-quorum node deployment solution.

- AZ1 and AZ2 have complete data, and AZ3 functions as the quorum node.
- AZ1 and AZ2 can access services at the same time to support dual-AZ active-active DR.
- AZ3 serves as the quorum AZ. If one AZ is faulty, the majority of ETCD nodes can survive and database clusters can perform arbitration.
- In quorum-based replication between primary and standby DNs, there must be synchronous backup DNs across AZs and data will not be lost.
- If a standby DN is faulty, services are not interrupted. If the primary DN is faulty, a primary/standby switchover is automatically performed.
- This solution provides high availability for AZ faults. If AZ1 or AZ2 is faulty, all services in the faulty AZ are automatically switched to the other AZ. After the failover is complete, services can continue running.
- If any of AZ1 or AZ2 and the quorum AZ are faulty, you need to manually start the faulty AZs.

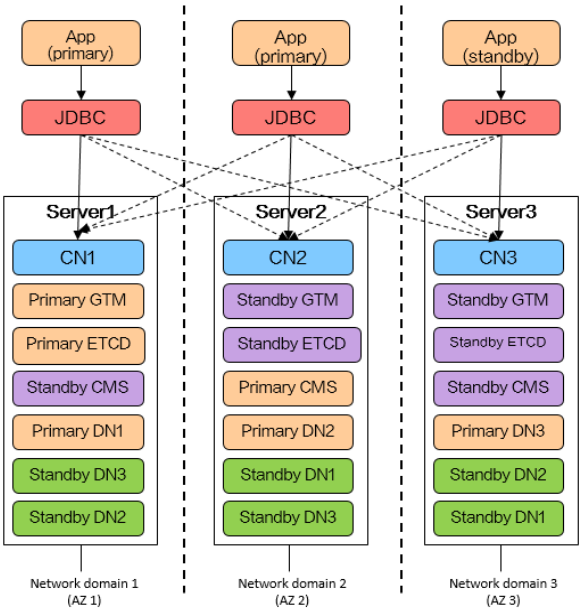


2.2.2.2 Intra-City, 3-AZ, 3-Node Deployment (3CNs + 3DNs)

The intra-city, 3-AZ deployment is supported. Three AZs are deployed in peer-to-peer mode and can access services. The deployment solution can achieve zero RPO and avoid network disconnections between data centers.

1. AZ1, AZ2, and AZ3 can access services at the same time.
2. In quorum-based replication between primary and standby DNs, there must be synchronous backup DNs across AZs and data will not be lost.

3. If a standby DN is faulty, services are not interrupted. If the primary DN is faulty, a primary/standby switchover is automatically performed.
4. This solution provides high availability for AZ faults. If AZ1, AZ2 or AZ3 is faulty, all services in the faulty AZ are automatically switched to the other AZ. After the failover is complete, services become normal.

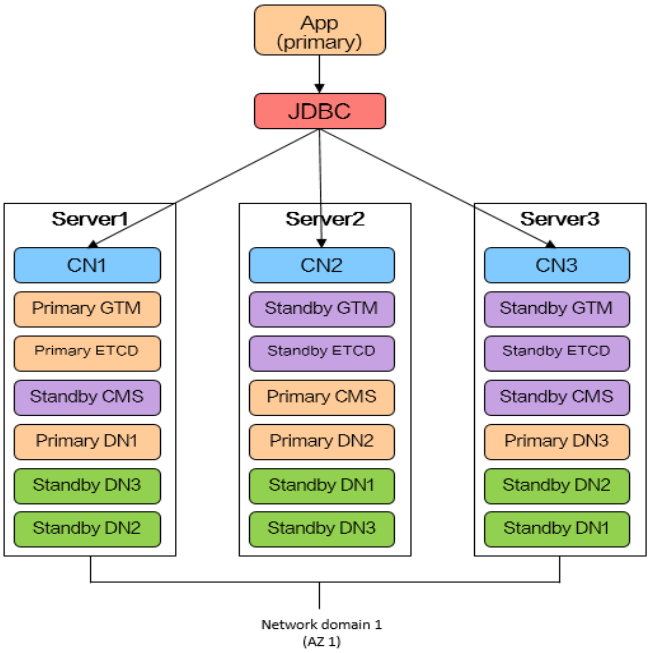


2.2.2.3 Single-AZ, 3-Node Deployment (3CNs + 3DNs)

The single-AZ three-replica deployment helps protect against node-level faults.

It is applicable to scenarios where database component and physical server faults need to be prevented, and data center DR is not required.

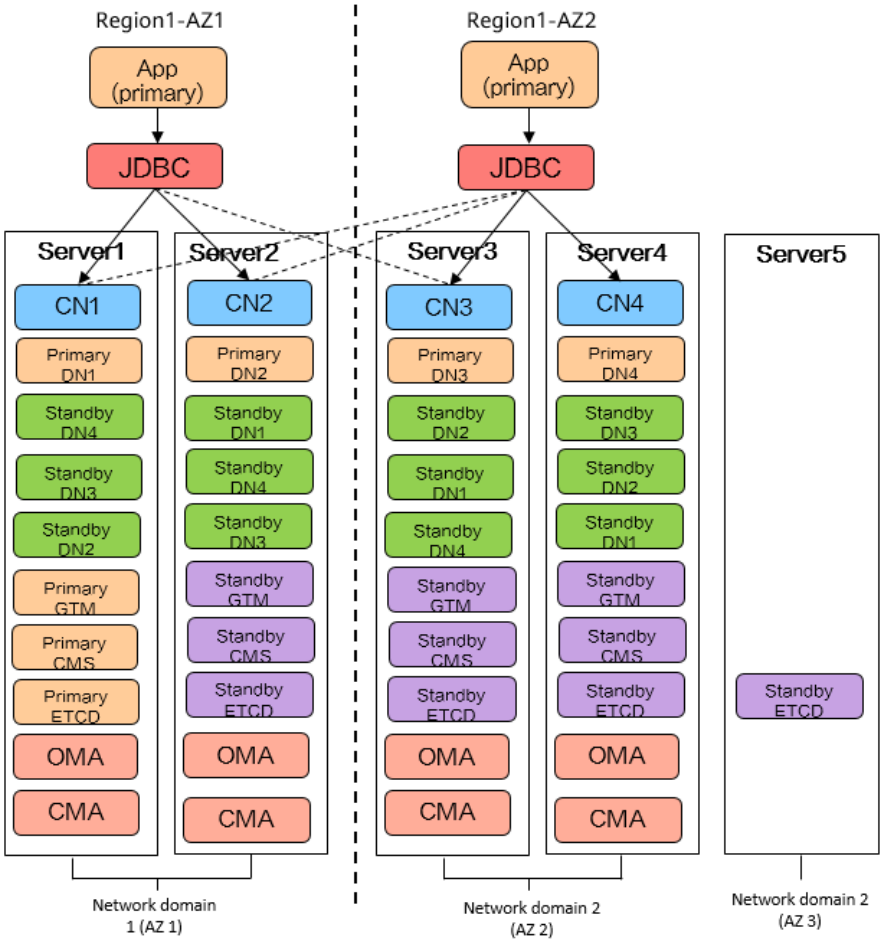
- In primary/standby DN quorum replication, data is synchronized to at least one standby to ensure zero RPO.
- If a standby DN is faulty, services are not interrupted. If the primary DN is faulty, a primary/standby switchover is automatically performed.
- There are three copies of data. If any node is faulty, the system still has two copies of data available.
- The primary and standby DNs of a shard cannot be deployed on the same server.



2.2.2.4 Intra-City, Dual-AZ, 5-Node Deployment (4 CNs + 4 DNs + 1 Quorum Node)

This deployment applies to a scenario where two active-active data centers are deployed in the same city to process core services from financial institutions and enterprises. It is suitable for customers requiring high-performance, highly reliable, and cost-insensitive systems where resources are physically isolated.

- At least five nodes are required, and the expansion increment is 4 nodes.
- This deployment solution consists of two service AZs and one quorum AZ. Both data centers have primary nodes. Two service AZs are deployed in peer-to-peer mode, and all data centers in the AZs can process services. The quorum AZ is responsible for auxiliary quorum, but cannot access services.
- An intra-city deployment can achieve high availability at the node, AZ, and data center levels. If AZ1, AZ2 or AZ3 is faulty, all services in the faulty AZ are automatically switched to the other AZ. When AZ1 or AZ2 is faulty, services can continue running after the failover is complete. When AZ3 is faulty, services are not interrupted. If any of AZ1 or AZ2 and the quorum AZ are faulty, users need to manually start the faulty AZs.
- If a standby DN is faulty, services are not interrupted. If the primary DN is faulty, a primary/standby switchover is automatically performed.
- When a fault occurs in a node, an AZ, or a data center, an RPO of zero and an RTO of less than 60 seconds (automatic switchover) can be achieved to ensure data consistency.



2.2.3 Intra-City HA and Remote DR Deployment

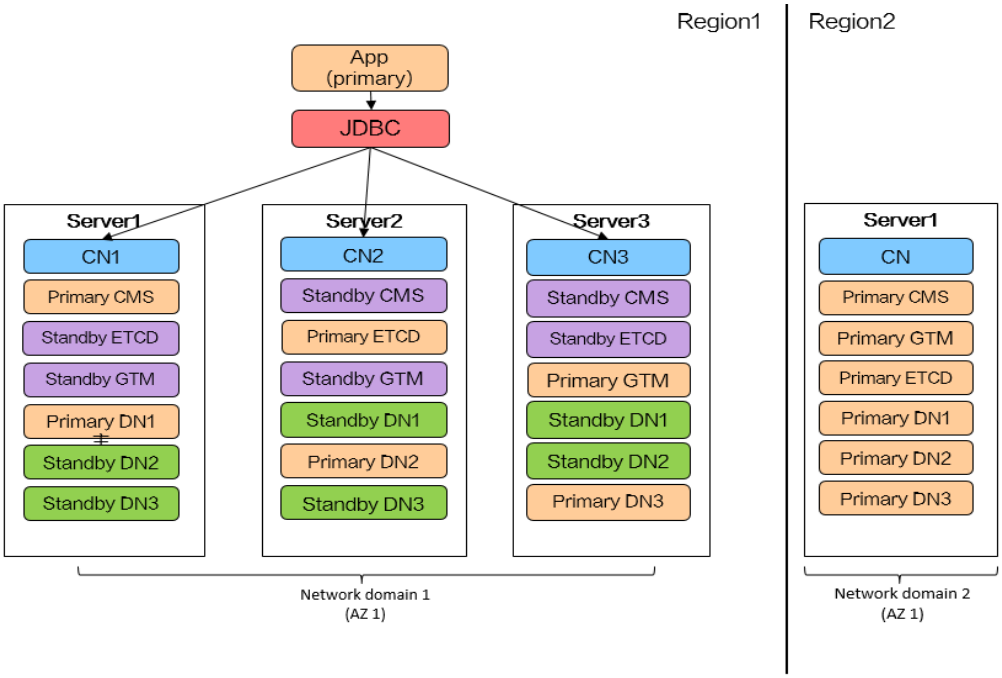
2.2.3.1 Intra-City, Single-AZ, 3-Node Deployment + Remote, Single-AZ, Single-Node Deployment

Two data centers are located in different cities, with one replica deployed separately. This solution provides component-level DR in the same city and cross-region DR.

Data reliability of three-replica or single-AZ deployment is 99.99%. Therefore, system reliability will not be improved in single-AZ deployment even if the number of replicas exceeds three.

- A complete database cluster is deployed in both the local and remote data centers.
- In primary/standby DN quorum replication, data is synchronized to at least one standby to ensure zero RPO.
- If a standby DN is faulty, services are not interrupted. If the primary DN is faulty, a primary/standby switchover is automatically performed.
- There are three copies of data. If any node is faulty, the system still has two copies of data available. In addition, any standby DN can be promoted to primary.

- If a region is faulty, you need to manually switch services to the normal region.

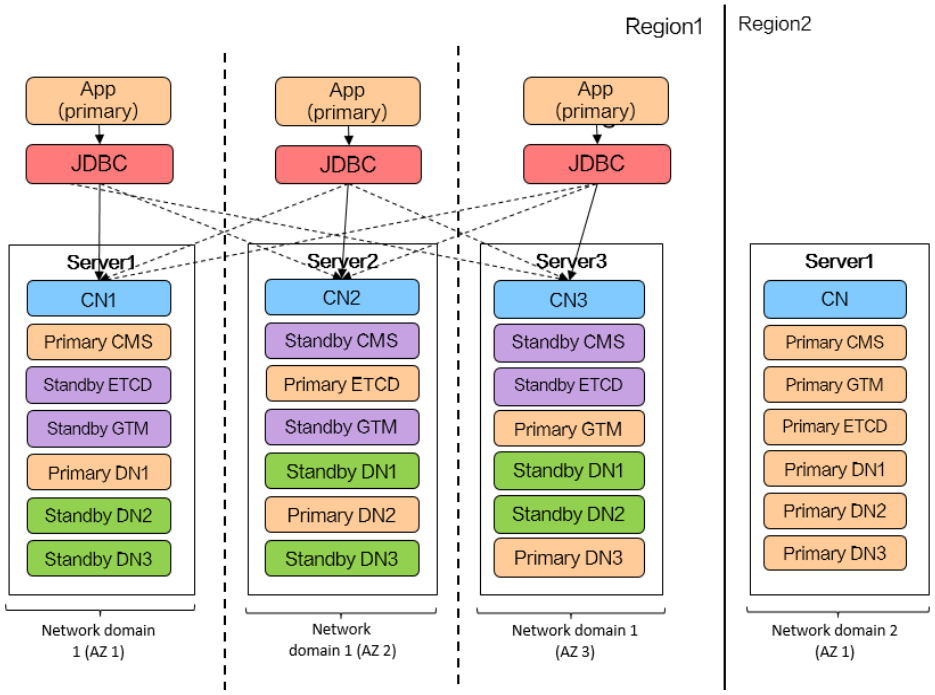


2.2.3.2 Intra-City, 3-AZ, 3-Node Deployment + Remote, Single-AZ, Single-Node Deployment

Two data centers are located in different cities, with one replica deployed separately. This solution provides component-level DR in the same city and cross-region DR.

Data reliability of three-replica or single-AZ deployment is 99.99%. Therefore, system reliability will not be improved in single-AZ deployment even if the number of replicas exceeds three.

- A complete database cluster is deployed in both the local and remote data centers.
- In primary/standby DN quorum replication, data is synchronized to at least one standby to ensure zero RPO.
- If a standby DN is faulty, services are not interrupted. If the primary DN is faulty, a primary/standby switchover is automatically performed.
- There are three copies of data. If any node is faulty, the system still has two copies of data available. In addition, any standby DN can be promoted to primary.
- If a region is faulty, you need to manually switch services to the normal region.

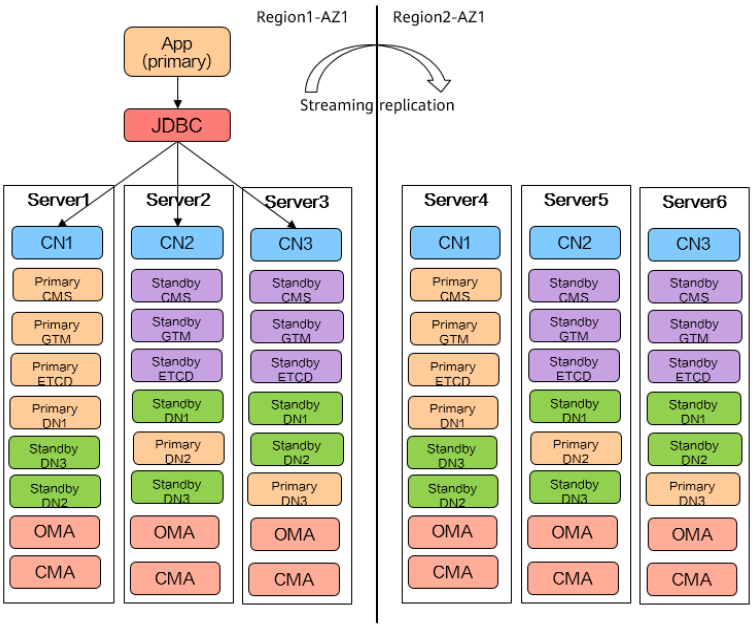


2.2.3.3 Intra-City, Single-AZ, 3-Node Deployment + Remote, Single-AZ, 3-Node Deployment

Two data centers are located in different cities, with three replicas deployed separately. This solution provides component-level DR in the same city and cross-region DR.

Data reliability of three-replica or single-AZ deployment is 99.99%. Therefore, system reliability will not be improved in single-AZ deployment even if the number of replicas exceeds three.

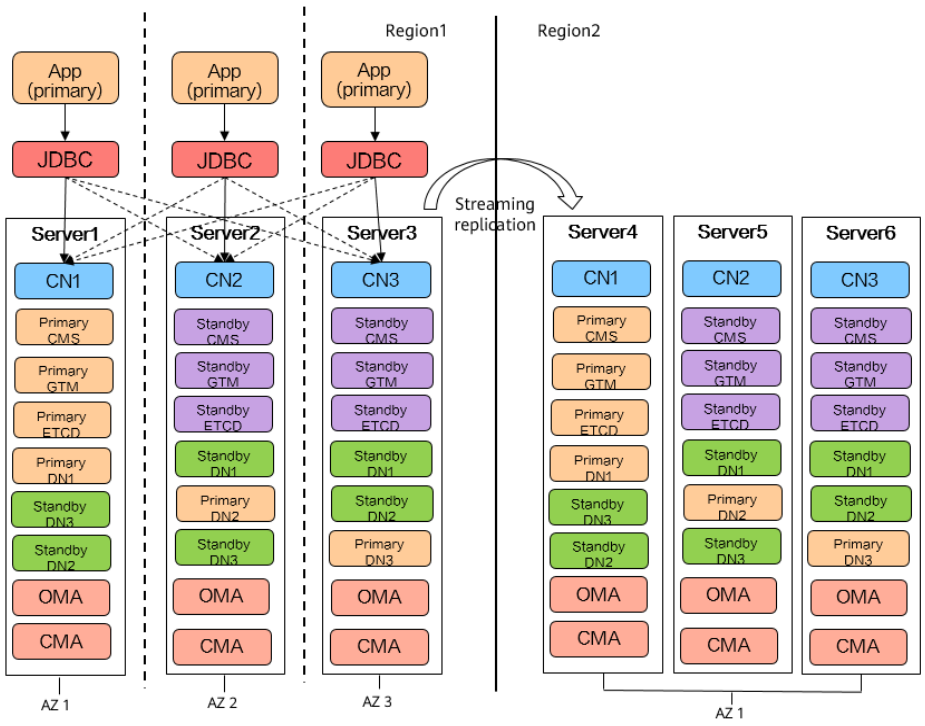
- A complete database cluster is deployed in both the local and remote data centers.
- In primary/standby DN quorum replication, data is synchronized to at least one standby to ensure zero RPO.
- If a standby DN is faulty, services are not interrupted. If the primary DN is faulty, a primary/standby switchover is automatically performed.
- There are three copies of data. If any node is faulty, the system still has two copies of data available. In addition, any standby DN can be promoted to primary.
- Cross-region DR requires manual switchover.



2.2.3.4 Intra-City, 3-AZ, 3-Node Deployment + Remote, Single-AZ, 3-Node Deployment

Among four data centers, three data centers are deployed in a city and one data center is deployed in another city. Three replicas (one primary and two DNs) are supported. In this deployment, the intra-city DCs can prevent component and AZ faults and the cross-city DC can prevent cross-region faults.

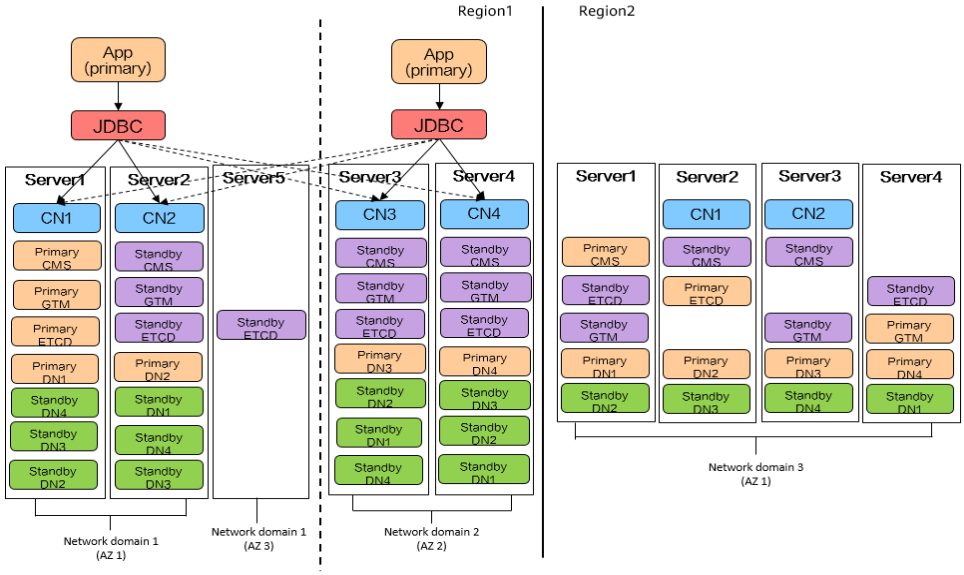
- A complete database cluster is deployed in both the local and remote data centers.
- In primary/standby DN quorum replication, data is synchronized to at least one standby to ensure zero RPO.
- If a standby DN is faulty, services are not interrupted. If the primary DN is faulty, a primary/standby switchover is automatically performed.
- There are three copies of data. If any node is faulty, the system still has two copies of data available. In addition, any standby DN can be promoted to primary.
- The intra-city DR provides high availability for AZ faults. If AZ1, AZ2 or AZ3 is faulty, all services in the faulty AZ are automatically switched to the other AZ. After the failover is complete, services become normal.
- Cross-region DR requires manual switchover.



2.2.3.5 Intra-City, 3-AZ, 5-Node Deployment + Remote, Single-AZ, 4-Node Deployment

Among four data centers, three data centers are deployed in a city and one data center is deployed in another city. Four replicas (one primary and three DNs) are supported. In this deployment, the intra-city DCs can prevent component and AZ faults and the cross-city DC can prevent cross-region faults.

- A complete database cluster is deployed in both the local and remote data centers.
- Streaming replication is used to synchronize data between the primary and standby DNs. Data is synchronized to at least one standby DN to ensure zero RPO.
- If a standby DN is faulty, services are not interrupted. If the primary DN is faulty, a primary/standby switchover is automatically performed.
- There are four copies of data. If any node is faulty, the system still has three copies of data available. In addition, any standby DN can be promoted to primary.
- The intra-city DR provides high availability for data center faults. If AZ1, AZ2 or AZ3 is faulty, all services in the faulty AZ are automatically switched to the other AZ. After the failover is complete, services become normal.
- Cross-region DR requires manual switchover.

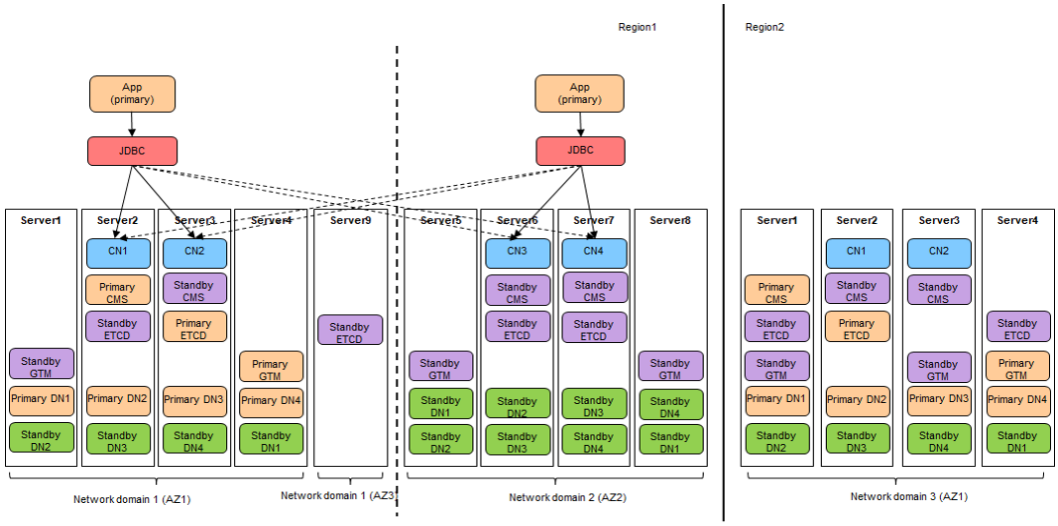


2.2.3.6 Intra-City, 3-AZ, 9-Node Deployment + Remote, Single-AZ, 4-Node Deployment

Two data centers are deployed in the same city and one data center in another city. There are the same shards in city 1 and city 2, but city 1 supports four replicas and city 2 supports two replicas. A complete intra-city active-active deployment solution consists of two service AZs and one quorum AZ. Two service AZs are deployed in peer-to-peer mode, and every data center accesses services. The quorum AZ is responsible for auxiliary quorum to avoid SPOFs. It cannot access services. The deployment solution can achieve zero RPO and withstand network disconnections between data centers. GaussDB also supports 2-AZ, 4-replica (one primary and three standby DNs), and 1-quorum AZ deployment solution. Remote DR provides region-level DR.

- A complete database cluster is deployed in both the local and remote data centers.
- In the same city, AZ1 and AZ2 have complete data. AZ3 serves as the quorum AZ. AZ1 and AZ2 can access services at the same time to implement dual-AZ active-active DR. If one AZ is faulty, the majority of ETCD nodes can survive, ensuring data consistency.
- Streaming replication is used to synchronize data between the primary and standby DNs. Data is synchronized to at least two standby DNs to ensure zero RPO.
- If a standby DN is faulty, services are not interrupted. If the primary DN is faulty, a primary/standby switchover is automatically performed.
- There are four copies of data. If any node is faulty, the system still has three copies of data available. In addition, any standby DN can be promoted to primary.
- The intra-city DR provides high availability for data center faults. If AZ1, AZ2 or AZ3 is faulty, all services in the faulty AZ are automatically switched to the other AZ. After the failover is complete, services can continue running. If any of AZ1 or AZ2 and the quorum AZ are faulty, users need to manually start the faulty AZs.

- Cross-region DR requires manual switchover.



2.3 GaussDB Primary/Standby (Centralized) Deployment

2.3.1 Deployment Model for GaussDB Primary/Standby Instances

Deployment Model	Node	Replica	AZ	Description
Intra-city HA	5	4	2 + 1 (arbitration AZ)	Two nodes per AZ are symmetrically deployed (2+2+1).
	3	3	1	Three nodes are deployed in one AZ.
	3	3	3	Each node is deployed in an AZ.
	2	2+1 (log copy)	2	One DN and one log are deployed on one node, and one DN is deployed on the other node.
Intra-city HA + remote DR	4	4	2	Intra-city single-AZ deployment with three nodes; remote single-AZ deployment with single node
	4	4	4	Intra-city three-AZ deployment with three nodes; remote single-AZ deployment with single node

Deployment Model	Node	Replica	AZ	Description
	6	5	4	Intra-city three-AZ deployment with five nodes; remote single-AZ deployment with single node
	6	6 (1 primary + 4 standby)	4	Intra-city three-AZ deployment with five nodes; remote single-AZ deployment with single node
	6	6	2	Intra-city single-AZ deployment with three nodes; remote single-AZ deployment with three nodes
	6	6	4	Intra-city 3-AZ deployment with one node for each AZ; remote single-AZ deployment with three nodes
	8	7	3+1 (quorum AZ)	Intra-city, 3-AZ deployment with five nodes for each AZ; remote single-AZ deployment with three nodes

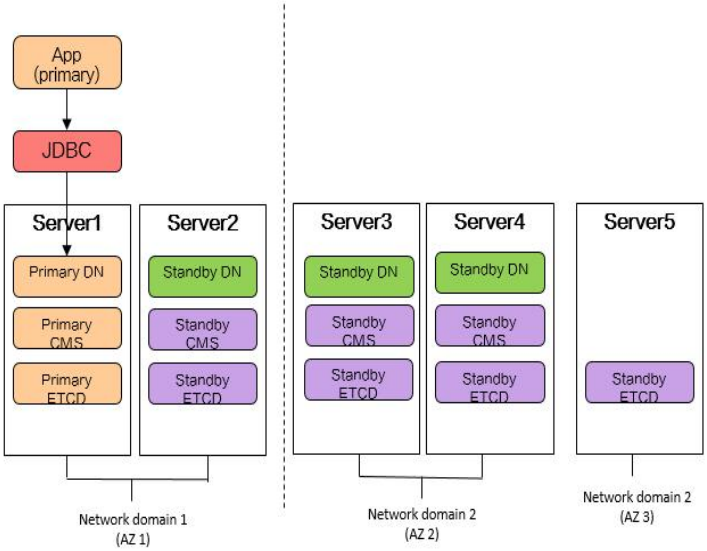
2.3.2 Intra-City HA Deployment Scenarios

2.3.2.1 Intra-City, Dual-AZ, 5-Node Deployment (4 Replicas + 1 Quorum Node)

This deployment consists of two service AZs and one quorum node. It can achieve zero RPO and avoid network disconnections between data centers. A dual-AZ 4-replica (one primary and three standby DNs) and 1-quorum node deployment solution is supported.

- AZ1 and AZ2 have complete data, and AZ3 functions as the third-party quorum node.
- AZ3 serves as the quorum AZ. If one AZ is faulty, the majority of ETCD nodes can survive, and database clusters can perform arbitration.
- Quorum and Paxos protocols are supported for replications between primary and standby DNs. There must be synchronous backup DNs across AZs and data will not be lost.
- If a standby DN is faulty, services are not interrupted. If the primary DN is faulty, a primary/standby switchover is automatically performed.
- This solution provides high availability for data center faults. If AZ1 or AZ2 is faulty, all services in the faulty AZ are automatically switched to the other AZ. After the failover is complete, services can continue running.

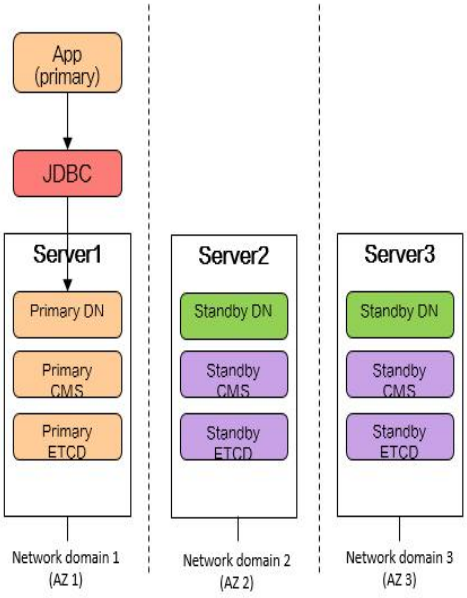
- If any of AZ1 or AZ2 and the quorum AZ are faulty, you need to manually start the faulty AZs.



2.3.2.2 Intra-City, 3-AZ, 3-Node Deployment (3 Replicas)

The intra-city, 3-AZ deployment is supported. Three AZs are deployed in peer-to-peer mode and can access services. The deployment solution can achieve zero RPO and avoid network disconnections between data centers.

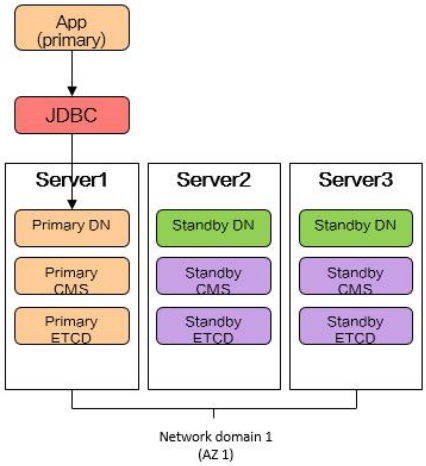
1. Quorum and Paxos protocols are supported for replications between primary and standby DNs. There must be synchronous backup DNs across AZs and data will not be lost.
2. If a standby DN is faulty, services are not interrupted. If the primary DN is faulty, a primary/standby switchover is automatically performed.
3. This solution provides high availability for data center faults. If AZ1, AZ2 or AZ3 is faulty, all services in the faulty AZ are automatically switched to the other AZ. After the failover is complete, services become normal.
4. If any of AZ1 or AZ2 and the AZ3 are faulty, you need to manually start the faulty AZs.



2.3.2.3 Single-AZ, 3-Node Deployment (3 Replicas)

The single-AZ three-replica deployment helps defend against instance-level faults. This deployment is applicable to scenarios where data center DR is not required but some server faults need to be prevented.

- Quorum and Paxos protocols are supported for replications between primary and standby DNs. Data is synchronized to at least one standby to ensure zero RPO.
- If a standby DN is faulty, services are not interrupted. If the primary DN is faulty, a primary/standby switchover is automatically performed.
- There are three copies of data. If any node is faulty, the system still has two copies of data available.

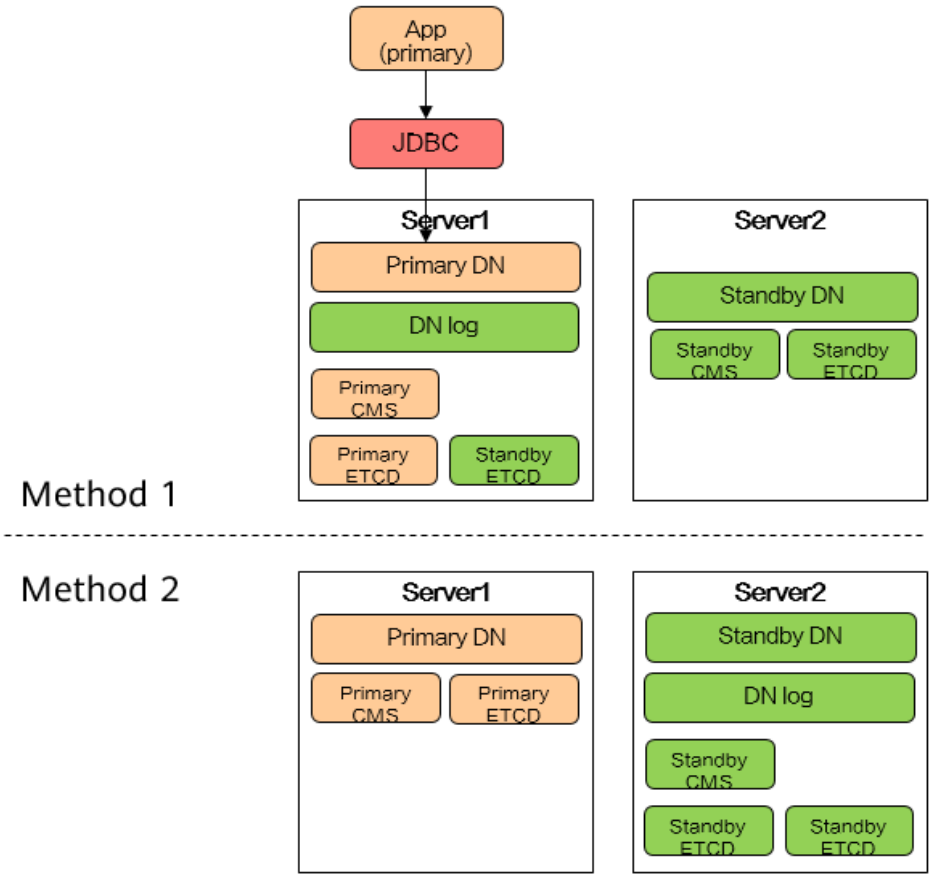


2.3.2.4 Intra-City, Single-AZ, Two-Node Deployment (1 Primary + 1 Standby + 1 Log)

- Deploy the following on two physical machines, separately:

  One primary DN, one standby DN, one log copy, three CMSs, and three ETCD nodes.
- There are two complete replicas and one log copy:
  - The log copy is used to conserve storage and compute resources.
  - The database uses the majority protocol to ensure software HA (DNs, CMSs, ETCDs).
  - If the server where the minority of DNs reside is faulty, services are not affected.
  - If the host where the majority of DNs reside is faulty, the HA cluster reduces replicas to forcibly restart the service (data may be lost). The high-availability service stops and needs to be forcibly restarted.
  - Read operations are not supported on the standby DN.
  - Two-node deployment provides less reliability. If one node is faulty, the database may become read-only.
- Overall SLA
  a. The overall SLA cannot be guaranteed because manual intervention is required to ensure strong consistency.
  b. A 99.95% SLA can ensure software HA (DNs, CMSs, ETCDs).

This deployment solution is shown in the following figure.

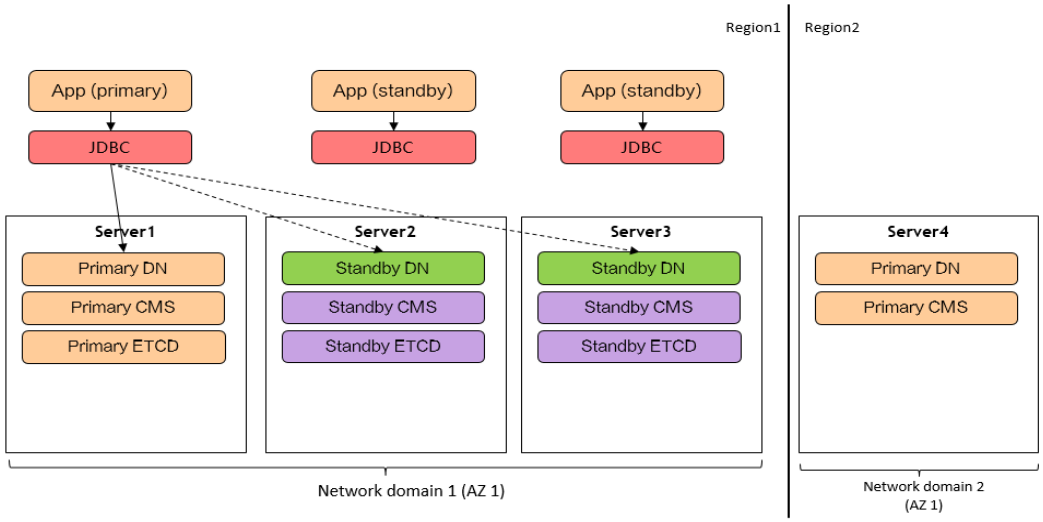


2.3.3 Intra-City HA and Remote DR Deployment

2.3.3.1 Intra-City, Single-AZ, 3-Node Deployment + Remote, Single-AZ, Single-Node Deployment

Two data centers are deployed in different cities. There are three replicas (1 primary DN and 2 standby DNs) in a data center and one replica in another data center. In this deployment, the intra-city data center can defend against component and node faults and the cross-city data center can defend against region-level faults. Intra-city single-AZ and remote single-AZ deployment are used.

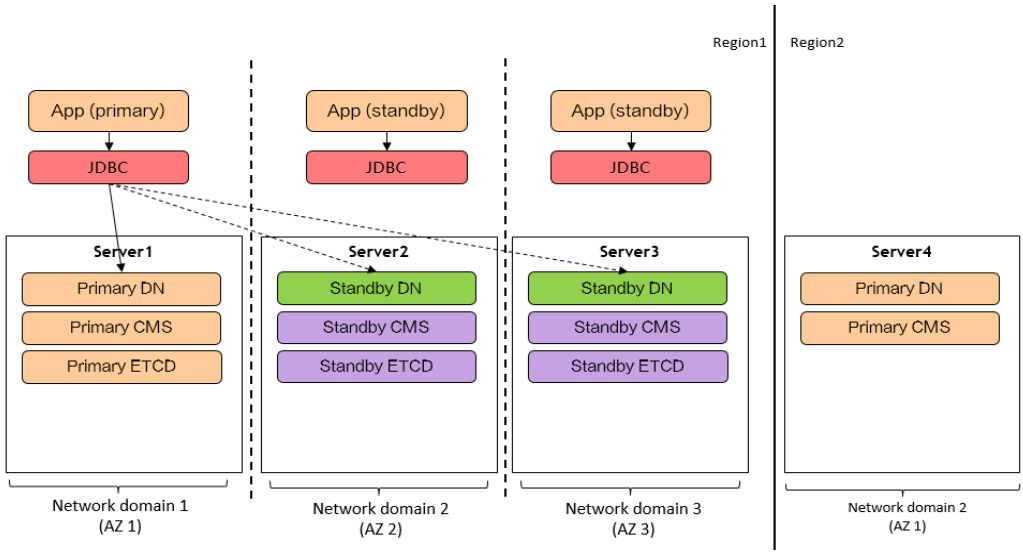
- A complete database cluster is deployed in both the local and remote data centers.
- In intra-city primary/standby DN quorum replication, data is synchronized to at least one standby DN to ensure zero RPO.
- If a standby DN is faulty, services are not interrupted. If the primary DN is faulty, a primary/standby switchover is automatically performed.
- If any of the DNs becomes faulty, the system still has two copies of data to ensure service continuity.
- A DR switchover between the primary/standby DN needs to be performed manually.



2.3.3.2 Intra-City, 3-AZ, 3-Node Deployment + Remote, Single-AZ, Single-Node Deployment

Two data centers are deployed in different cities. There are three replicas (1 primary DN and 2 standby DNs) in a data center and one replica in another data center. In this deployment, the intra-city data center can defend against component and node faults and the cross-city data center can defend against region-level faults. Intra-city three-AZ and remote single-AZ deployment are used.

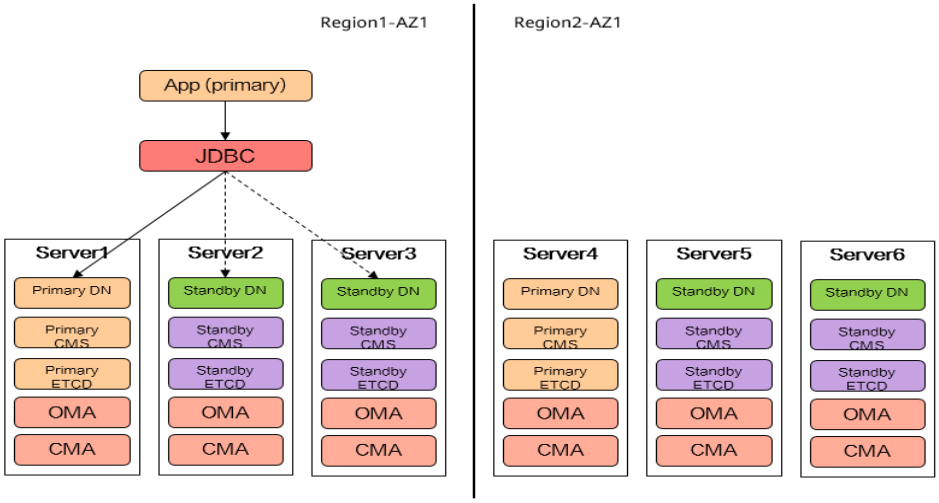
- A complete database cluster is deployed in both the local and remote data centers.
- In intra-city primary/standby DN quorum replication, data is synchronized to at least one standby DN to ensure zero RPO.
- If a standby DN is faulty, services are not interrupted. If the primary DN is faulty, a primary/standby switchover is automatically performed.
- If any of the DNs becomes faulty, the system still has two copies of data to ensure service continuity.
- A DR switchover between the primary/standby DN needs to be performed manually.



2.3.3.3 Intra-City, Single-AZ, 3-Node Deployment + Remote, Single-AZ, 3-Node Deployment

Two data centers are deployed in different cities and there are three replicas (one primary and two standby DNs) in each city. In this deployment, the intra-city data center can defend against component and node faults and the cross-city data center can defend against region-level faults.

- A complete database cluster is deployed in both the local and remote data centers.
- In intra-city primary/standby DN quorum replication, data is synchronized to at least one standby DN to ensure zero RPO.
- If a standby DN is faulty, services are not interrupted. If the primary DN is faulty, a primary/standby switchover is automatically performed.
- If any of the DNs becomes faulty, the system still has two copies of data to ensure service continuity.
- A DR switchover between the primary/standby DN needs to be performed manually.

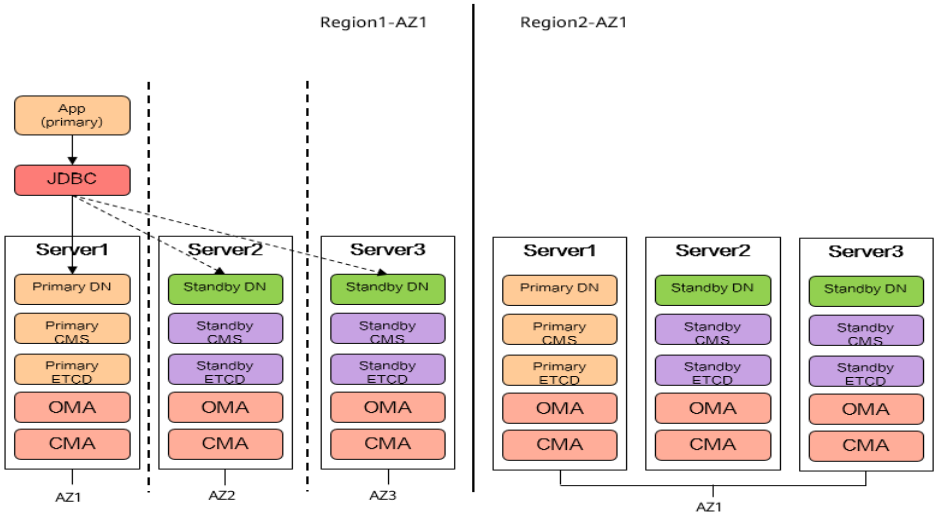


2.3.3.4 Intra-City, 3-AZ, 3-Node Deployment + Remote, Single-AZ, 3-Node Deployment

Among four data centers, three data centers are deployed in a city and a data center is deployed in another city. Three replicas (one primary and two DNs) are supported. In this deployment, the intra-city data centers can defend against instance-level and AZ-level faults and the cross-city data center can defend against region-level faults.

- A complete database cluster is deployed in both the local and remote data centers.
- In intra-city primary/standby DN quorum replication, data is synchronized to at least one standby DN to ensure zero RPO.
- If a standby DN is faulty, services are not interrupted. If the primary DN is faulty, a primary/standby switchover is automatically performed.
- There are three copies of data. If any node is faulty, the system still has two copies of data available.
- The intra-city DR provides high availability for data center faults. If AZ1, AZ2 or AZ3 is faulty, all services in the faulty AZ are automatically switched to the other AZ. After the failover is complete, services become normal.
- A DR switchover between the primary/standby DN needs to be performed manually.

Figure 2-3 Streaming replication

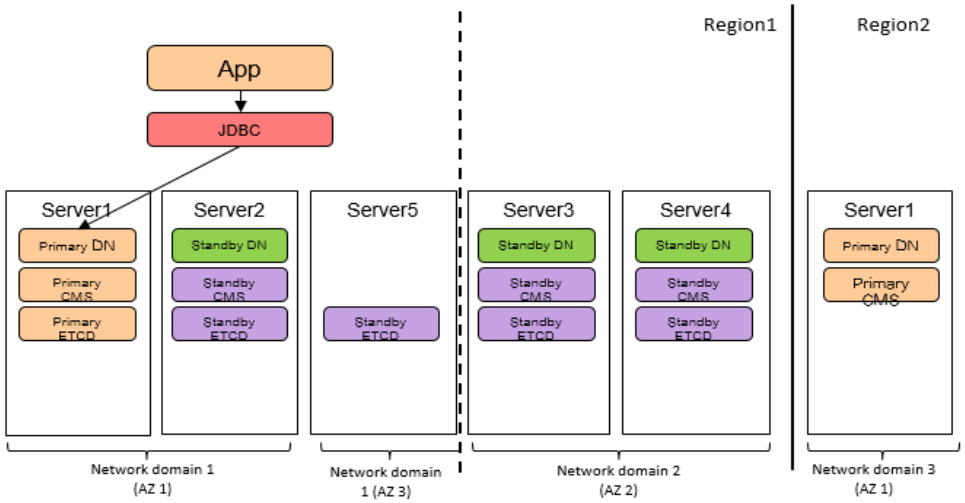


2.3.3.5 Intra-City, 3-AZ, 5-Node Deployment + Remote, Single-AZ, Single-Node Deployment

Two data centers are deployed in the same city and one data center in another city. There are four replicas in the same city and one replica in another city. A complete intra-city active-active deployment solution consists of two service AZs and one quorum AZ. Two service AZs are deployed in peer-to-peer mode, and every data center accesses services. The quorum AZ is responsible for auxiliary quorum to avoid SPOFs. It cannot access services. The deployment solution can achieve zero RPO and withstand network disconnections between data centers.

GaussDB also supports 2-AZ, 4-replica (one primary and three standby DNs), and 1-quorum AZ deployment solution. Remote DR provides region-level DR.

- A complete database cluster is deployed in both the local and remote data centers.
- In the same city, AZ1 and AZ2 have complete data. AZ3 serves as the quorum AZ. AZ1 and AZ2 can access services at the same time to implement dual-AZ active-active DR. If one AZ is faulty, the majority of ETCD nodes can survive, ensuring data consistency.
- Streaming replication is used to synchronize data between the primary and standby DNs. Data is synchronized to at least two standby DNs to ensure zero RPO.
- If a standby DN is faulty, services are not interrupted. If the primary DN is faulty, a primary/standby switchover is automatically performed.
- There are four copies of data. If any node is faulty, the system still has three copies of data available. In addition, any standby DN can be promoted to primary.
- The intra-city DR provides high availability for data center faults. If AZ1, AZ2 or AZ3 is faulty, all services in the faulty AZ are automatically switched to the other AZ. After the failover is complete, services can continue running. If any of AZ1 or AZ2 and the quorum AZ are faulty, users need to manually start the faulty AZs.
- If a region is faulty, you need to manually switch services to the normal region.

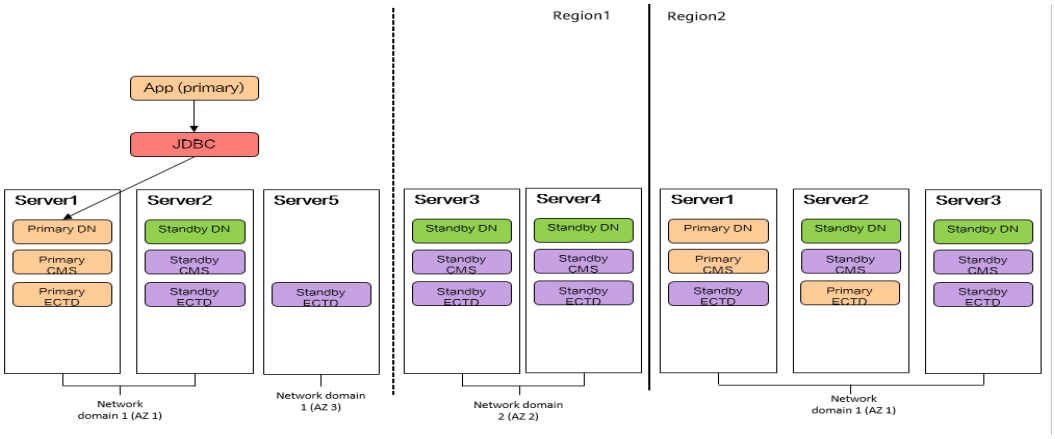


2.3.3.6 Intra-City, 3-AZ, 5-Node Deployment + Remote, Single-AZ, 3-Node Deployment

Two data centers (four replicas) are deployed in the same city, and one data center (one primary DN + two standby DNs) in another city. A complete intra-city active-active deployment solution consists of two service AZs and one quorum AZ. Two service AZs are deployed in peer-to-peer mode, and every data center accesses services. The quorum AZ is responsible for auxiliary quorum to avoid SPOFs. It cannot access services. The deployment solution can achieve zero RPO and withstand network disconnections between data centers. GaussDB also

supports 2-AZ, 4-replica (one primary and three standby DNs), and 1-quorum AZ deployment solution. Remote DR provides region-level DR.

- A complete database cluster is deployed in both the local and remote data centers.
- In the same city, AZ1 and AZ2 have complete data. AZ3 serves as the quorum AZ. AZ1 and AZ2 can access services at the same time to implement dual-AZ active-active DR. If one AZ is faulty, the majority of ETCD nodes can survive, ensuring data consistency.
- Streaming replication is used to synchronize data between the primary and standby DNs. Data is synchronized to at least two standby DNs to ensure zero RPO.
- If a standby DN is faulty, services are not interrupted. If the primary DN is faulty, a primary/standby switchover is automatically performed.
- There are four copies of data. If any node is faulty, the system still has three copies of data available. In addition, any standby DN can be promoted to primary.
- The intra-city DR provides high availability for data center faults. If AZ1, AZ2 or AZ3 is faulty, all services in the faulty AZ are automatically switched to the other AZ. After the failover is complete, services can continue running. If any of AZ1 or AZ2 and the quorum AZ are faulty, users need to manually start the faulty AZs.
- Cross-region DR requires manual switchover.

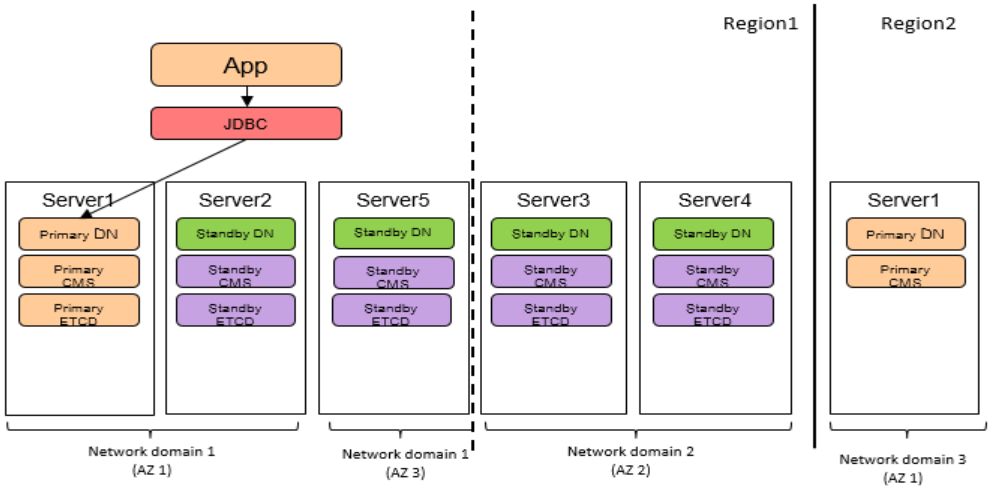


2.3.3.7 Intra-City, 3-AZ, 5-Node Deployment (1 Primary + 4 Standby) + Remote, Single-AZ, Single-Node Deployment

Two data centers are deployed in the same city and one data center in another city. There are five replicas in the same city and one replica in another city. A complete intra-city active-active deployment solution consists of three service AZs. The deployment solution can achieve zero RPO and avoid single point of failure (SPOF) and network disconnections between data centers.

- A complete database cluster is deployed in both the local and remote data centers.
- In the same city, AZ1 and AZ2 have complete data. AZ1, AZ 2, and AZ3 can access services at the same time.

- Streaming replication is used to synchronize data between the primary and standby DNs. Data is synchronized to at least two standby DNs to ensure zero RPO.
- If a standby DN is faulty, services are not interrupted. If the primary DN is faulty, a primary/standby switchover is automatically performed.
- There are five copies of data. If any node is faulty, the system still has four copies of data available. In addition, any standby DN can be promoted to primary.
- The intra-city DR provides high availability for data center faults. If AZ1, AZ2 or AZ3 is faulty, all services in the faulty AZ are automatically switched to the other AZ. After the failover is complete, services can continue running. If the majority of AZs are faulty, users need to manually start the faulty AZs.
- If a region is faulty, you need to manually switch services to the normal region.



3 Deployment Process

The following figure shows the GaussDB lightweight deployment process.

Figure 3-1 Deployment process

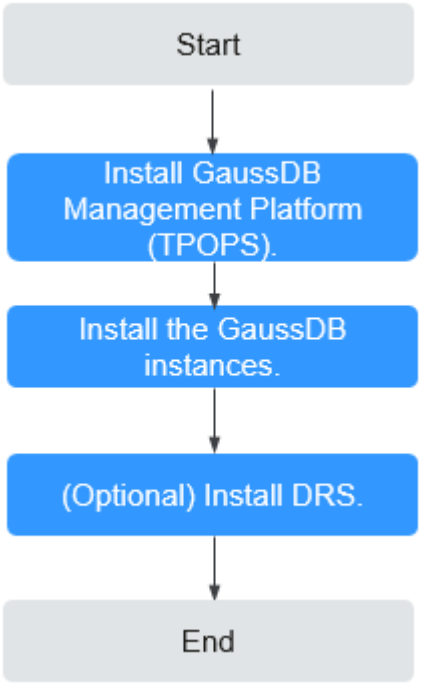


Table 3-1 GaussDB lightweight deployment process

No.	Process	Reference
1	Determining the GaussDB deployment solution	**GaussDB Distributed Deployment** or **GaussDB Primary/Standby (Centralized) Deployment**
2	Installing GaussDB Management Platform (TPOPS)	**Installing GaussDB Management Platform (TPOPS)**

No.	Process	Reference
3	Installing the GaussDB instances	**Installing Instances**
4	(Optional) Installing DRS	**(Optional) Installing DRS**

4 Installing GaussDB Management Platform (TPOPS)

4.1 Installation Overview

4.1.1 Service Overview

GaussDB Management Platform (TPOPS) is a database O&M management platform based on Huawei Cloud Stack Database Service (DBS). It is stable, reliable, and easy to use. With GaussDB Management Platform (TPOPS), you can obtain the consistent user experience as that on Huawei Cloud Stack without relying on Huawei Cloud Stack.

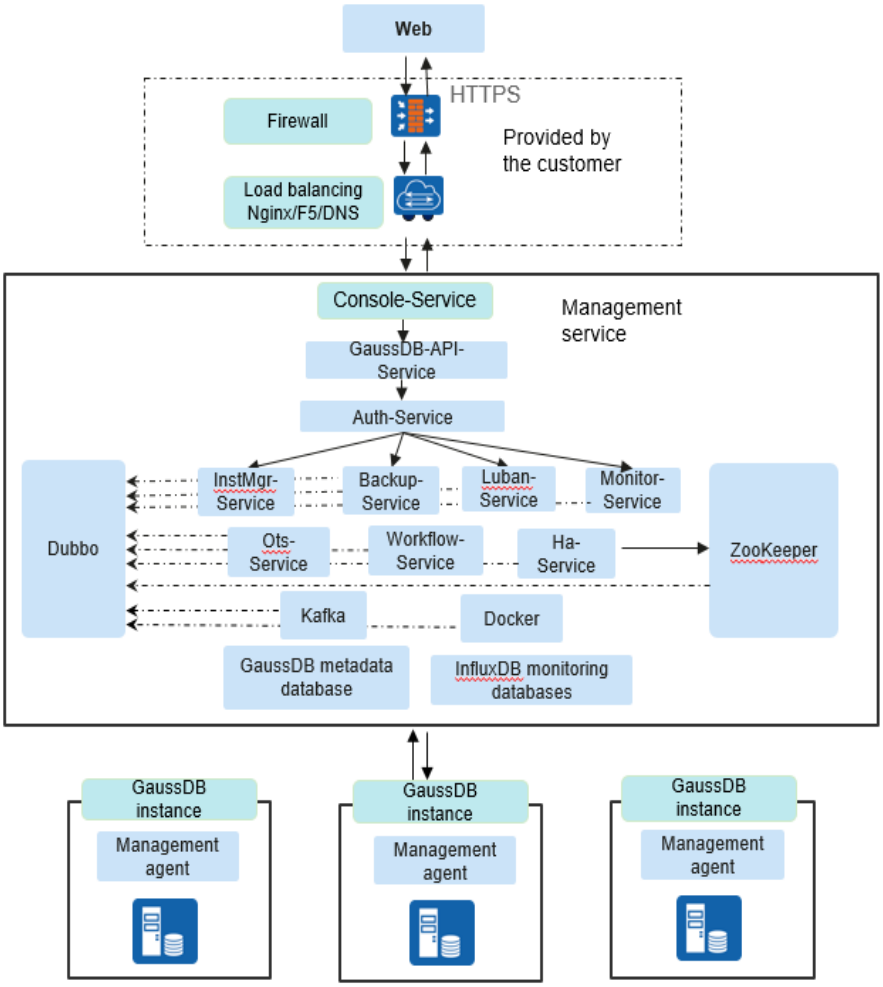
Currently, GaussDB Management Platform (TPOPS) can be deployed only in HA mode, that is, GaussDB Management Platform (TPOPS) is installed on three nodes.

The HA mode supports cross-equipment room and cross-region deployment. It effectively prevents single points of failure (SPOFs), single-node faults, and single-equipment room faults of microservices, ensuring stable service running.

4.1.2 Deployment

GaussDB Management Platform (TPOPS) is developed based on the browser/server (B/S) architecture and consists of the web, management service, and management agent. **Figure 4-1** shows the software architecture of GaussDB Management Platform (TPOPS).

Figure 4-1 Deployment architecture of each node



- Web: As a user access subsystem, it delivers operation instructions delivered by users on the web to database instances through the management service. In addition, it can transmit the data analyzed and processed by Console-Service to the web user interface through the web interaction module for display.
- Management service: The management service is the analysis and control subsystem of the management plane. It consists of microservice modules such as the web interaction module, instance service module, component service module, and data service module. It displays instance monitoring data to the web through the web interconnection module in the upstream and delivers operation instructions to the management agent through the component service module in the downstream. The information collected by the management agent is stored in the monitoring database and metabase of the management service. The instance service module and data service module of the management service analyze and process the information.
- The microservice modules are listed in the following table.

Table 4-1 Microservices

Component	Description
DBS-docker-service	Installation tool package for a stateless service
DBS-platform-data	Basic SQL statements of the GaussDB Management Platform (TPOPS) metadata database
DBS-GaussDB-feature-data	Basic SQL statements of the GaussDB Management Platform (TPOPS) metadata database
DBS-GaussDB-data	Basic SQL statements of the GaussDB Management Platform (TPOPS) metadata database
Docker	A container-based platform used to build, deploy, and run microservice applications
InfluxDB	Time series database of GaussDB Management Platform (TPOPS), which stores monitoring and alarm data
SFTP	File server
GaussDB	Metadata database of GaussDB Management Platform (TPOPS)
DBS-zookeeper	Registration center
DBS-kafka	Message middleware
DBS-monitor-service	Monitoring metrics (alarms, metrics, and top SQL statements)
DBS-rds-ha-admin	Instance monitoring service (for updating instance status and node roles)
DBS-resource-manager	Resource management
DBS-workflow	Workflow for instance creation and deletion
DBS-common-service	Public service, which is provided for instance parameter groups
DBS-auth	User authentication service
DBS-ots	Operation audit service
DBS-luban	Monitoring and O&M management
DBS-GaussDB-open-api	API entry of the GaussDB Management Platform (TPOPS) open-api microservice
DBS-GaussDB-instancemanager	Instance management service
DBS-GaussDB-backupmanager	Backup management service

Component	Description
DBS-gaussdb-console	Front-end console

- Management agent: collects running data of instances, hosts, and components and reports the data to the management service for analysis and processing. In addition, it receives instructions from the management service and performs required operations on the instances. A management agent is deployed on each node accommodating the instances. All management agents form the instance monitoring subsystem of GaussDB Management Platform (TPOPS).

4.1.3 System Requirements

4.1.3.1 General Requirements

OS Requirements

Table 4-2 Supported OSs and software packages

OS	Compatibility	Architecture
Kylin V10	SP1: 4.19.90-23.8.v2101.ky10.aarch64 SP2: 4.19.90-24.4.v2101.ky10.aarch64	Arm
	SP1: 4.19.90-23.8.v2101.ky10.x86_64 SP2: 4.19.90-24.4.v2101.ky10.x86_64	x86
UnionTech	4.19.90-2201.4.0.0135.up1.uel20.x86_64	x86
	4.19.90-2201.4.0.0135.up1.uel20.aarch64	Arm

Software Requirements

Table 4-3 GaussDB Management Platform (TPOPS) software requirements

Software	Specifications
Browser	Required versions: • Google Chrome 120, 119, or 118 • Firefox 121 or 120 • Microsoft Edge: updated with Windows 10
JRE	GaussDB Management Platform (TPOPS) supports the following JREs in Open JDK: Open JDK: 1.8.0_272 If there are version vulnerabilities, fix them in a timely manner. You are advised to use the built-in OpenJDK of the Kylin image.
Python	Python 3.7.4 or 3.7.9 (version in the OS). Run the following command to check the Python version: **python3 --version** If the queried version is not Python 3.7.4 or Python 3.7.9, install Python 3.7.4 or Python 3.7.9 by referring to **Installing Python 3**.

Specifications of Managed Nodes

Table 4-4 Instance management specifications

VM Specifications	Maximum Nodes that Can Be Managed
8 vCPUs, 64 GB	60 nodes
≥16 vCPUs and 128 GB	500 nodes

4.1.3.2 Hardware Requirements for TPOPS Separately Deployed

The following table lists the hardware requirements for the GaussDB Management Platform (TPOPS) when the GaussDB Management Platform (TPOPS) and DRS are independently deployed.

Table 4-5 Node hardware requirements

Hardware	Configuration	Description
CPU	8-core processor or more	CPU specifications of the server where the GaussDB Management Platform (TPOPS) is located
Available memory	64 GB or above	Available memory of the server where the GaussDB Management Platform (TPOPS) is located
Disk	1,150 GB or above	Available disk space of the server where the GaussDB Management Platform (TPOPS) is located SSDs (SATA SSDs or SAS SSDs) are required. **NOTE** The required disk space depends on the data storage duration.

Table 4-6 Disk space partition requirements

Hardware	Configuration	File System Format	Description
Disk space of the installation tool package Default decompression path: **/data** Total: 50 GB	50G	Ext4 is recommended.	Disk space required for uploading and installing the installation tool package and software packages of the GaussDB management platform. Default path: **/data**
Disk space of the main program Default disk mounting path: **/opt/cloud** Total: 200 GB	200 GB	Ext4 is recommended.	Disk space required for installing the GaussDB Management Platform (TPOPS) main program. Default path: **/opt/cloud** **NOTE** This path is a public path. Only the disk space required for installing the main program is provided. The space required for others such as logs and databases is not included.

Hardware	Configuration	File System Format	Description
Disk space for public logs Default disk mounting path: **/opt/cloud/logs** Total: 100 GB	100 GB (shared by microservices)	Ext4 is recommended.	Disk space required by each GaussDB Management Platform (TPOPS) service to generate logs. Default path: **/opt/cloud/logs/**
GaussDB database disk space Default disk mounting path: **/opt/gaussdb** Total: 200 GB	200 GB	Ext4 is recommended.	Disk space required for installing, deploying, and storing the primary GaussDB Management Platform (TPOPS) database. Default path: **/opt/gaussdb**
SFTP Default disk mounting path: **/opt/sftphome** Total: 100 GB	100 GB	Ext4 is recommended.	Disk space required for installing, deploying, and storing SFTP on GaussDB Management Platform (TPOPS). Default path: **/opt/sftphome**
Backup disk space Default disk mounting path: **/opt/backup** Total: 200 GB	200 GB	Ext4 is recommended.	Disk space required by the GaussDB Management Platform (TPOPS) backup storage. Default path: **/opt/backup**
Docker disk space Default disk mounting path: **/opt/docker**	200 GB	Ext4 is recommended.	Disk space required for storing Docker data of GaussDB Management Platform (TPOPS). Default path: **/opt/docker**
Disk space of the time series database InfluxDB Default path: **/opt/influxdb**	100 GB	Ext4 is recommended.	Disk space required by the storage of data such as monitoring alarms on GaussDB Management Platform (TPOPS). Default path: **/opt/influxdb**

NOTE

- The **/opt/cloud**, **/opt/cloud/logs**, **/opt/gaussdb**, **/opt/sftphome**, **/opt/backup**, **/opt/docker**, and **/opt/influxdb** directories are dedicated for installing GaussDB Management Platform (TPOPS) and do not share disk space with other directories. When GaussDB Management Platform (TPOPS) is uninstalled, all files in these directories are deleted.
- You are advised to mount each directory to a separate disk and isolate the disk space on the disk.
- You can run the **df -h** command to check the disk usage of the file system. If the disk partition requirements are not met, you are advised to mount disks by referring to **Mounting Disks**.
- The disk paths in **Table 4-6** can contain only letters, digits, and underscores (_).
- For details about the maximum number of instances that can be managed, see **Specifications of Managed Nodes**.
- You are advised to mount an independent disk to the **/data** directory with at least 50 GB of free space for storing installation packages and software packages to be installed.

4.1.3.3 Hardware Requirements for TPOPS Deployed with DRS

The following table lists the hardware requirements of the GaussDB Management Platform (TPOPS) and DRS when they are deployed together.

Table 4-7 Node hardware requirements

Hardware	Configuration	Description
CPU	96-core processor	CPU specifications of the server where the GaussDB Management Platform (TPOPS) and DRS are located
Available memory	576 GB or above	Available memory of the server where the GaussDB Management Platform (TPOPS) and DRS are located
Disk	1,150 GB + Disk space required by DRS-Service + Disk space required by the DRS-Gaussdb metabase	Available disk space of the server where the GaussDB Management Platform (TPOPS) and DRS are located For details about the disk space required by DRS-Service and that required by the DRS-Gaussdb metabase, see the "Disk Mount Points" sheet in the **LLD template**. SSDs (SATA SSDs or SAS SSDs) are required. **NOTE** The required disk space depends on the data storage duration.

Table 4-8 Disk space partition requirements

Hardware	Configuration	File System Format	Description
Disk space of the installation tool package Default decompression path: **/data** Total: 50 GB	50G	Ext4 is recommended.	Disk space required for uploading and installing the installation tool package and software packages of the GaussDB management platform. Default path: **/data**
Disk space of the main program Default disk mounting path: **/opt/cloud** Total: 200 GB	200 GB	Ext4 is recommended.	Disk space required for installing the GaussDB Management Platform (TPOPS) main program. Default path: **/opt/cloud** **NOTE** This path is a public path. Only the disk space required for installing the main program is provided. The space required for others such as logs and databases is not included.
Disk space for public logs Default disk mounting path: **/opt/cloud/logs** Total: 100 GB	100 GB (shared by microservices)	Ext4 is recommended.	Disk space required by each GaussDB Management Platform (TPOPS) service to generate logs. Default path: **/opt/cloud/logs/**
GaussDB database disk space Default disk mounting path: **/opt/gaussdb** Total: 200 GB	200 GB	Ext4 is recommended.	Disk space required for installing, deploying, and storing the primary GaussDB Management Platform (TPOPS) database. Default path: **/opt/gaussdb**
SFTP Default disk mounting path: **/opt/sftphome** Total: 100 GB	100 GB	Ext4 is recommended.	Disk space required for installing, deploying, and storing SFTP on GaussDB Management Platform (TPOPS). Default path: **/opt/sftphome**

Hardware	Configuration	File System Format	Description
Backup disk space Default disk mounting path: **/opt/backup** Total: 200 GB	200 GB	Ext4 is recommended.	Disk space required by the GaussDB Management Platform (TPOPS) backup storage. Default path: **/opt/backup**
Docker disk space Default disk mounting path: **/opt/docker**	200 GB	Ext4 is recommended.	Disk space required for storing Docker data of GaussDB Management Platform (TPOPS). Default path: **/opt/docker**
Disk space of the time series database InfluxDB Default path: **/opt/influxdb**	100 GB	Ext4 is recommended.	Disk space required by the storage of data such as monitoring alarms on GaussDB Management Platform (TPOPS). Default path: **/opt/influxdb**
DRS-Service data disk Default path: **/opt/drs**	For details about the data disk size requirements, see the "Disk Mount Points" sheet in the **LLD template**.	Ext4 is recommended.	Disk space required for installing and deploying DRS-Service and storing logs and service data. Default path: **/opt/drs**
DRS-Service local backup set disk Default path: **/opt/drs-backup**	For details about the data disk size requirements, see the "Disk Mount Points" sheet in the **LLD template**.	Ext4 is recommended.	DRS-Service backup disk space for storing the DRS-Service backup data. Default path: **/opt/drs-backup**

Hardware	Configuration	File System Format	Description
DRS-Service database disk space Default path: **/data/cluster**	For details about the data disk size requirements, see the "Disk Mount Points" sheet in the **LLD template**.	Ext4 is recommended.	Disk space required for installing, deploying, and storing the primary DRS-Service database. Default path: **/data/cluster**

NOTE

- You are advised to mount each directory to a separate disk and isolate the disk space on the disk.
- You can run the **df -h** command to check the disk usage of the file system. If the disk partition requirements are not met, you are advised to mount disks by referring to **Mounting Disks**.
- The disk paths in **Table 4-8** can contain only letters, digits, and underscores (_).
- For details about the maximum number of instances that can be managed, see **Specifications of Managed Nodes**.
- You are advised to mount an independent disk to the **/data** directory with at least 50 GB of free space for storing installation packages and software packages to be installed.

4.1.4 Account Information

This topic describes the GaussDB Management Platform (TPOPS) account information.

During the installation, the **service**, **dbadmin**, and **sftpservice** users described in **Table 4-9** need to be generated. Therefore, ensure that the preceding users do not exist in the environment before the installation.

Table 4-9 Accounts

Account	Description
root	User with the OS permission. A password is required for logging in to the management node where GaussDB Management Platform (TPOPS) is to be installed. The password of the **root** the can contain only digits, letters, and the following special characters: -_+@

Account	Description
service	User for starting microservice processes. You can specify **service_group_id** and **service_user_id** in the installation configuration file **user_edit_file.conf**. The default IDs are **1010**. If it has been occupied, change it to another value. The default path of the configuration file is **/data/docker-service/config/user_edit_file.conf**.
dbadmin	OS user used by the metadata database.
sftpservice	SFTP service user.
influxdb	OS user used by the time series database InfluxDB.

For details about accounts related to the data replication software, see ***HCS Data Replicate Software Installation Guide***.

4.1.5 Installation Process

The following figure shows the GaussDB Management Platform (TPOPS) installation process.

Figure 4-2 Installation process

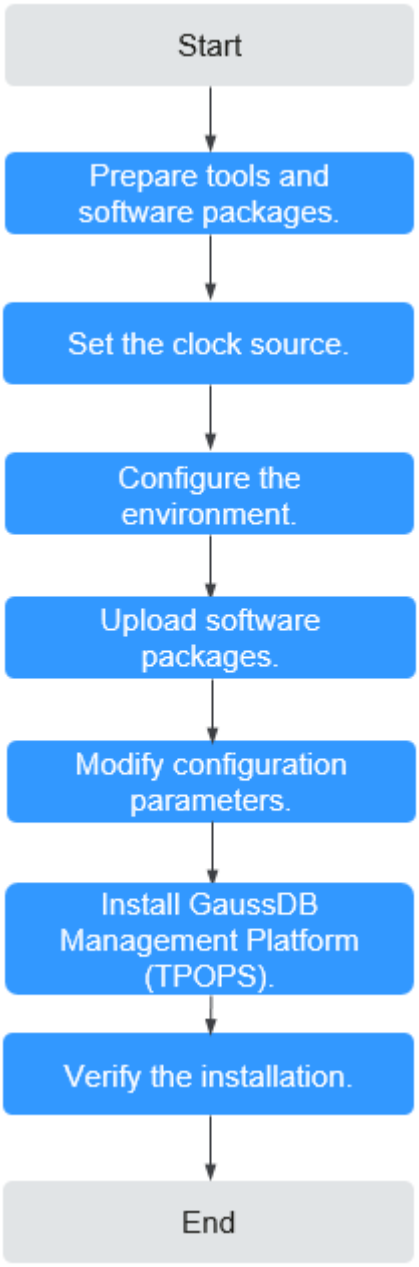


Table 4-10 GaussDB Management Platform (TPOPS) management and installation process

No.	Process	Description
1	Obtaining and verifying the software package	For details, see **Preparing Tools and Software Packages**.
2	Configuring the installation environment	For details, see **Environment Configurations**.
3	Uploading software packages	For details, see **Uploading Software Packages**.

No.	Process	Description
4	Modifying configuration parameters	For details, see **Modifying Configuration Parameters**.
5	Installing GaussDB Management Platform (TPOPS)	For details, see **Installing the GaussDB Management Platform (TPOPS)**.
6	Verifying the installation	For details, see **Post-installation Check**.

4.2 Installation Preparation

4.2.1 Preparing Tools and Software Packages

Preparing Tools

This document uses a Kylin V10 (SP2) Arm server as an example.

Prepare three Huawei TaiShan servers running Kylin V10. In addition, install the software listed in **Table 4-11** in the local Windows system.

To obtain the software, visit the official websites.

Table 4-11 Software to be obtained on the local PC running a Windows operating system

Tool	Description	How to Obtain
SSH remote login tool such as PuTTY	Cross-platform remote access tool. This tool is used to access nodes in the Windows OS during the installation, such as logging in to the node to perform pre-installation configuration and check as well as installation commands.	Download it from its official website.
WinSCP	Tool for cross-platform file transfer. This tool is used to transfer files between Windows and Linux OSs, such as uploading software packages and configuration files.	
Decompression software, for example, 7-Zip	Used to decompress ***.tar.gz** files.	

Tool	Description	How to Obtain
Browser, for example: • Chrome • Firefox • Microsoft Edge	Used to log in to GUIs. Required versions: • Google Chrome 120, 119, or 118 • Firefox 121 or 120 • Microsoft Edge: updated with Windows 10	

Preparing Software Packages

Download the GaussDB Management Platform (TPOPS) software packages and verification files based on the OS type as the **root** user.

*indicates the version number. Use the actual version number in the software packages.

For details about the open source software list of different OS images, see **Open Source Software List**. Before the installation, compare the open source software list to avoid conflicts.

Table 4-12 Version software packages

Package	Sub-Software Package Name	Description	How to Obtain
DBS-GaussDB-Server_*_.tar.gz	DBS-GaussDB-backupmanager_*_all.tar.gz	Backup management service software package	• For enterprise users, click **here**. • For carrier users, click **here**.
	DBS-GaussDB-feature-data_*_all.tar.gz	Basic SQL package of the GaussDB Management Platform (TPOPS) metadata database	
	DBS-GaussDB-data_*_all.tar.gz	Basic SQL package of the GaussDB Management Platform (TPOPS) metadata database	
	DBS-GaussDB-instancemanager_*_all.tar.gz	Software package of the instance management service	

Package	Sub-Software Package Name	Description	How to Obtain
	DBS-GaussDB-open-api_*_all.tar.gz	API entry of the GaussDB Management Platform (TPOPS) open-api microservice	
	DBS-GaussDB-agent_*_all.tar.gz	GaussDB agent package	
DBS-docker-service_*_all.tar.gz	-	Installation tool package for a stateless service	
DBS-GaussDBConsole-Server_*_.tar.gz	DBS-gaussdb-console_*_all.tar.gz	Front-end console software package	
DBS-Platform-Server_*_.tar.gz	DBS-luban_*_all.tar.gz	Monitoring and O&M management software package	
	DBS-monitor-service_*_all.tar.gz	Monitoring metric software package	
	DBS-ots_*_all.tar.gz	Operation audit software package	
	DBS-platform-data_*_all.tar.gz	Basic SQL package of the GaussDB Management Platform (TPOPS) metadata database	
	DBS-rds-ha-admin_*_all.tar.gz	Software package of the instance monitoring service	
	DBS-resource-manager_*_all.tar.gz	Resource management software package	
	DBS-workflow_*_all.tar.gz	Workflow public service software package	
	DBS-auth_*_all.tar.gz	User authentication software package	
	DBS-common-service_*_all.tar.gz	Common service software package	

Package	Sub-Software Package Name	Description	How to Obtain
GaussDB_OS_PATCH_*.zip	-	OS_PATCH package for bringing hosts online	
DBS-GaussDB-Kernel_*_.tar.gz	DBS-GaussDB-om-agent_*_.tar.gz	GaussDB om-agent package	
DBS-GaussDB-Kylin-Kernel_*_.tar.gz	-	Kylin OS kernel package for GaussDB	
DBS-GaussDB-Uniontech-Kernel_*_.tar.gz	-	UnionTech OS kernel package for GaussDB	
DBS-DBMind-Manual_*.tar.gz	-	GaussDB DBMind kernel package	
DBS-tools_*_all.tar.gz	-	Installation tool package, including certificate files	
DBS-MetaDB_Kylin_Centralized_505.1.RC1.SPC0100.B006.tar.gz	-	Kylin OS metadata database kernel package	
DBS-MetaDB_UnionTech_Centralized_505.1.RC1.SPC0100.B006.tar.gz	-	Metadata database kernel package of UnionTech OS	

4.2.2 Setting Clock Sources

4.2.2.1 Configuration Description

Before installing GaussDB Management Platform (TPOPS), ensure that the clock sources of all nodes are synchronized. The following two clock source configuration guides are provided for references. Set it based on the site requirements.

- You are advised to use Chrony Time Daemon (chrony) to automatically synchronize the system clock on each host to ensure that the clock on each GaussDB Management Platform (TPOPS) server is correct and consistent. For details, see **Configuring Time Synchronization Using Chrony**. Note that only chrony can be used to configure the clock source for UnionTech system.
- If you need to use Network Time Protocol (NTP) to automatically synchronize the system time of each server, ensure that the clock on each GaussDB

Management Platform (TPOPS) server is correct and consistent. For details, see **Configuring Time Synchronization Using NTP**.

4.2.2.2 Prerequisites

- Before configuring the clock source, you have permission on all hosts as the **root** user.
- The server configured with the clock source can communicate with the client, and the chrony or NTP service port is not blocked by the firewall.

4.2.2.3 Configuring Time Synchronization Using Chrony

Scenarios

To ensure that the time difference is within 30 seconds, you are advised to use chrony to automatically synchronize the system clock on each server to ensure that the clock on each GaussDB Management Platform (TPOPS) server is correct and consistent. For details, see **Procedure**.

Procedure

Step 1 Log in to each server where time synchronization is to be configured as the **root** user.

Step 2 Enter **chrony** and press **Tab** twice to check whether chrony is installed.

- If **chronyc** and **chronyd** are displayed, chrony has been installed. In this case, go to the next step.
- If **chronyc** and **chronyd** are not displayed, chrony is not installed. Run the following command to install it.

  yum install chrony -y

Step 3 Run the following command to modify the server configuration.

NOTE

This step is to modify the server configuration. Do not modify the client configuration.

1. Run the **vi editor** command to edit **/etc/chrony.conf**.

   vi /etc/chrony.conf

2. Add **allow all** by referring to the following figure.

   ```
   # Allow NTP client access from local network.
   #allow 192.
   allow all
   ```

3. Delete **#** and uncomment the line where **local stratum 10** is located by referring to the following figure.

   ```
   # Serve time even if not synchronized to a time source.
   local stratum 10
   ```

4. Press **Esc** and run the **:wq!** command to save the chage and exit.

5. Run the following command to restart chrony service on the server for the configuration to take effect.

   systemctl restart chronyd

Step 4 Run the following command to modify the client configuration.

NOTE

This step is to modify the client configuration. Do not modify the server configuration.

1. Run the **vi** command to edit the **/etc/chrony.conf** file on the client.

   vi /etc/chrony.conf

2. Add **#** to uncomment the original pool line in the configuration file, and add **server domain name/IP address of the time synchronization server iburst** by referring to the following figure.

3. Press **Esc** and run the **:wq!** command to save the chage and exit.
4. Run the following command to restart chrony service on the client for the configuration to take effect.

   systemctl restart chronyd

Step 5 Run the following command to check the configuration.

Run the following command on the client and server. If the time on the server is the same as that on the client, the clock source is configured successfully.

date

Information similar to the following is displayed:

```
Fri Dec 15 07:39:58 UTC 2023
```

----End

4.2.2.4 Configuring Time Synchronization Using NTP

Scenarios

To ensure that the time difference is within 30 seconds, you are advised to use Network Time Protocol (NTP) to automatically synchronize the system clock on each server to ensure that the clock on each GaussDB Management Platform (TPOPS) server is correct and consistent. For details, see **Procedure**.

Procedure

NOTE

Generally, NTP provided by the system can be used to synchronize time. If a stable and reliable NTP server is available, use it as the NTP source for all servers.

If there is no such a server, use a fixed server as the NTP source.

Step 1 Install NTP on each host using the Yum package manager.

yum install ntp ntpdate -y

Step 2 Back up the **ntp.conf** file before configuring it.

[Setting up an NTP server]

Add the following content to the end of the **/etc/ntp.conf** file on the NTP server.

Replace **192.168.0.0** in **restrict 192.168.0.0 mask 255.255.0.0** with the network segment corresponding to the IP address of the NTP server.

```
[root@* /]# vi /etc/ntp.conf
restrict default ignore
restrict 127.0.0.1
restrict 192.168.0.0 mask 255.255.0.0

driftfile /var/lib/ntp/drift
pidfile /var/run/ntpd.pid
#logfile /var/log/ntp.log

# local clock
server 127.127.1.0
fudge 127.127.1.0 stratum 10
```

[Setting up an NTP client]

Add the following content to the end of the **/etc/ntp.conf** file on the NTP client:

- Replace **192.168.0.0** in **restrict 192.168.0.0 mask 255.255.0.0** with the network segment corresponding to the IP address of the NTP client.
- Replace **192.168.*.*** in **server 192.168.*.* iburst minpoll 4 maxpoll 6** with the IP address of the NTP server.
- Replace **192.168.*.*** in **fudge 192.168.*.* stratum 10** with the IP address of the NTP server.

```
[root@* /]# vi /etc/ntp.conf
restrict default [ignore]
restrict 127.0.0.1
restrict 192.168.0.0 mask 255.255.0.0

driftfile /var/lib/ntp/drift
pidfile /var/run/ntpd.pid
#logfile /var/log/ntp.log

# local clock
server 192.168.*.* iburst minpoll 4 maxpoll 6
fudge 192.168.*.* stratum 10
```

Parameter description:

- **restrict** is used to specify a time synchronization command permission of an IP address.

  restrict [*Client IP address*] mask [*netmask_IP*] [*parameter*]

  Where **parameter** include **ignore**, **nomodify**, **noquery**, **notrap**, and **notrust**.

  - **ignore**: All types of NTP synchronization are rejected by default.
  - **nomodify:** The time on the NTP server cannot be modified but can be verified on the client.
  - **notrust**: The source of the NTP client is considered as an untrusted subnet unless the client is authenticated.
  - **noquery**: The client time cannot be queried. Users cannot run the **ntpq**, **ntpc**, or other commands to query the NTP server.

  - **notrap**: The trap remote login is not provided. The mode 6 control message trap service is not provided for matching hosts. The trap service is the subsystem of the ntpdq control message protocol, used for remote event logging programs.
- **server** is used to specify the upper-layer NTP source server.

  server [*IP or hostname*] [*prefer*]

  If the upper-layer NTP source server is unavailable, you can set this parameter to **127.127.1.0** to allow the local computer functioning as the NTP source server.

Step 3 Restart the NTP synchronization service.

systemctl restart ntpd

systemctl status ntpd

Step 4 Check the NTP synchronization status.

ntpq -p

The command output is as follows:

```
remote           refid      st  t  when  poll reach  delay  offset  jitter
==============================================================================
LOCAL(0)        .LOCL.      10  l  589   64    0     0.000 0.000  0.000
*192.168.*.*  10.10.10.1  2   u  18    64   377    1.591 0.249  0.054
```

In the preceding commands:

- **remote** indicates that the NTP server is used.
  - ***** indicates the selected NTP server.
  - **LOCAL** indicates a local host.
  - **x** indicates an NTP server is no longer used.
  - **-** indicates an NTP server is no longer used.
  - **+** indicates an NTP server is running properly and preferred.
  - **#** indicates an NTP server is running properly and idle.
- **refid** indicates a higher-level NTP server used by the remote NTP server. **INIT** indicates a file is being obtained.
- **st** indicates the stratum of a remote NTP server.
- **when** indicates the time passed since last synchronization. The default unit is seconds (**h** for hours and **d** for days).
- **poll** indicates synchronization frequency, in seconds.
- **delay** indicates the round-trip time from a local host to a remote NTP server, in milliseconds.
- **offset** indicates time offset between a local host and a remote NTP server, in milliseconds.
- **jitter** indicates average time offset between a local host and a remote NTP server, in milliseconds.

Step 5 Final check method.

Run the following command on the client and server. If the time on the server is the same as that on the client, the clock source is configured successfully.

date

Command output:

```
Thu Sep 14 02:03:11 UTC 2023
```

Step 6 (Optional) Manually synchronize time.

If the time offset is always too large, you can run the following command to manually correct the time. This method conflicts with the automatic synchronization service. You need to stop the NTP service first.

systemctl stop ntpd

ntpdate -u 192.168.*.*

Command output:

```
18 Apr 14:54:20 ntpdate[108001]: adjust time server 192.168.*.* offset -0.000180 sec
```

If this method works, configure the following information in the **crontab** file in the system.

crontab -e

Command output:

```
* * * * * root /sbin/ntpdate -u 192.168.*.* 2>&1 1>>/tmp/ntpupdate.log
```

----End

4.2.3 Configuring a Yum Repository

Step 1 Log in to the server as user **root**.

Step 2 Upload the ISO file of the supported OS to a directory, for example, **/mnt**.

Step 3 Run the following command to go to the **/mnt** directory:

cd /mnt

Step 4 Run the following command to mount the ISO file to the **/mnt** directory:

mount -o loop *<iso file name>* /mnt

Step 5 Clear the unavailable Yum repository.

rm -rf /etc/yum.repos.d/*

Step 6 Run the following command to open the **local.repo** file:

vi /etc/yum.repos.d/local.repo

Step 7 Add the following information to create the local Yum repository configuration.

```
[local]
name=local
baseurl=file:///mnt
gpgcheck=0
enabled=1
```

NOTE

The **/mnt** in **baseurl** indicates the mount path of the ISO file.

Step 8 Press **Esc** and run the following command to save the change and exit:

:wq!

Step 9 Clear the Yum cache.

yum clean all

Step 10 Cache the local Yum repository.

yum makecache

----End

4.2.4 Environment Configurations

Scenarios

This topic describes how to check and prepare for the installation of the components on which GaussDB Management Platform (TPOPS) depends and the system configuration.

NOTE

- In this document, you can log in to all GaussDB Management Platform (TPOPS) nodes as the **root** user to perform the check. You can log in to a node using a password or key. For details about how to log in to a node using a key, see **Setting the root User for Logging In to a Management Plane Node Without a Password**.
- The usernames, user directories, and software packages in this document are only used as examples.
- The monitoring and alarm services depend on the server time. Configure correct time synchronization and ensure that the time server is running properly.

Check Items

1. Run the following command to check whether the firewall is disabled. If it is not, disable the firewall.

   systemctl status firewalld

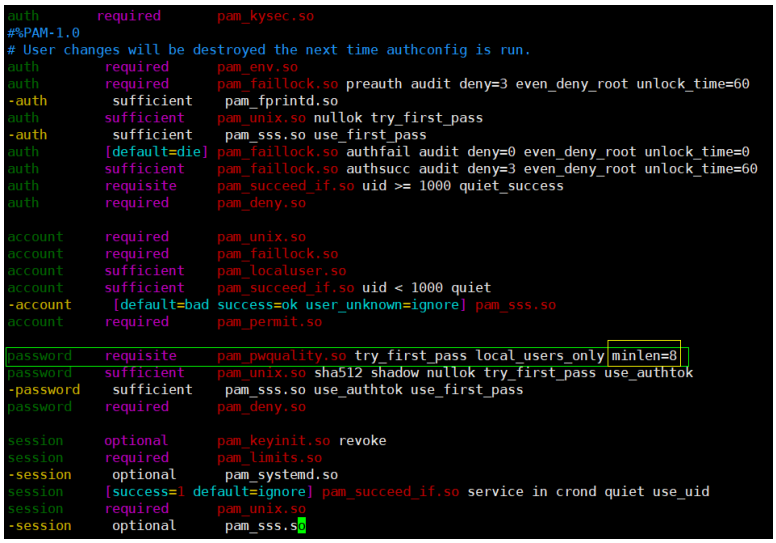
   - If the firewall status is **inactive (dead)**, the firewall is disabled.
   - If the firewall status is **active (running)**, the firewall is not disabled. Disable the firewall based on **How Do I Disable a Firewall?**.

2. Perform the following operations to set a PAM rule for the OS:

   a. Run the following command to open **/etc/pam.d/system-auth**:

   vi /etc/pam.d/system-auth

   b. Locate the line in the red box in the following figure and add **minlen=8** to the end of the line, as shown in the yellow box in the figure.



c. Run the following command to save the configuration and exit:

:wq!

3. (Optional) Check whether the audit service has memory leakage risks. For details, see **Large Memory Required Due to Memory Leaking of the Audit Service in the Kylin OS**.

4.2.5 Uploading Software Packages

Step 1 Log in to the node where GaussDB Management Platform (TPOPS) is to be installed as the **root** user.

Step 2 Run the following commands to create a directory.

Create the **/data** directory for all nodes to be installed and perform subsequent operations on any node. The following uses the **/data** directory as an example. You can create any directory.

mkdir -p /data

Step 3 Upload the GaussDB Management Platform (TPOPS) installation package to the **/data/** directory on the node where GaussDB Management Platform (TPOPS) is to be installed.

For example, upload **DBS-docker-service_*_all.tar.gz** to the **/data/** directory.

Step 4 Run the following commands to decompress the installation script package to the installation directory (**/data**):

cd /data

tar -xzvf DBS-docker-service*_all.tar.gz -C /data

Step 5 Upload the following installation packages to the **/data/docker-service/pkgs** directory:

- Metadata database kernel package (1 in total):
  - The package name corresponding to the Kylin OS is as follows:

    DBS-MetaDB_Kylin_Centralized_505.1.RC1.SPC0100.B006.tar.gz
  - The package name corresponding to the UnionTech OS is as follows:

DBS-MetaDB_UnionTech_Centralized_505.1.RC1.SPC0100.B006.tar.gz

- Microservice packages (3 in total):
  DBS-GaussDB-Server_*_.tar.gz
  DBS-Platform-Server_*_.tar.gz
  DBS-GaussDBConsole-Server_*_.tar.gz
- GaussDB instance installation packages (6 in total):
  GaussDB_OS_PATCH_*.zip
  DBS-DBMind-Manual_*.tar.gz
  DBS-tools_*_all.tar.gz
  DBS-GaussDB-Kernel_*.tar.gz
  DBS-GaussDB-Kylin-Kernel_*.tar.gz
  DBS-GaussDB-Uniontech-Kernel_*.tar.gz

  NOTE

  DBS-GaussDB-Kylin-Kernel_*.tar.gz is the installation package corresponding to the Kylin OS, and **DBS-GaussDB-Uniontech-Kernel_*.tar.gz** is the installation package corresponding to the Uniontech OS.

  When uploading the **DBS-GaussDB-Kylin-Kernel_*.tar.gz** or **DBS-GaussDB-Uniontech-Kernel_*.tar.gz** installation package, you can select the corresponding package based on the actual requirements of the data plane.

----End

NOTICE

- Ensure that the package is unique. Only one package of the same microservice or component can be uploaded.
- If some installation packages in **Step 5** are replaced, distribute the installation packages again by referring to **How Do I Distribute Installation Packages Again?**.

4.2.6 Modifying Configuration Parameters

Scenarios

This topic describes how to configure the configuration files required for the installation. For details about the parameters, see **Step 4**.

Procedure

NOTE

- All IP addresses used in this section must be those obtained by running **ifconfig**.
- The length of the configured directory cannot exceed 64 characters.

Step 1 Log in to the node where GaussDB Management Platform (TPOPS) is to be installed as the **root** user.

This node is the one you have logged in to in **Uploading Software Packages**.

Step 2 Run the following command to switch to the directory that contains the configuration file:

cd /data/docker-service/config

Step 3 Run the following command to open the configuration file:

vi user_edit_file.conf

Step 4 Modify the configuration file by referring to the following example:

CAUTION

- In the following configuration file, **node1_ip** must be set to the IP address of the node to which the command is delivered. Set **node2_ip** and **node3_ip** to the actual IP addresses of other nodes. node*_ip2 must be able to ping and communicate with the GaussDB instances over SSH. For details about the parameters, see **Configuring the Installation Configuration File**.
- If the configuration parameters are changed, distribute the packages again by performing the operations provided in section **How Do I Distribute Installation Packages Again?**.

```
[user_edit]
ssh_port = 22    # SSH port number for login between nodes
gauss_path = /opt/gaussdb    # Metadata database installation directory (cannot be in the /home directory)
node1_ip = 192.168.0.1    # Local IP address of node 1 (IP address of the execution node)
node2_ip = 192.168.0.2    # Local IP address of node 2
node3_ip = 192.168.0.3    # Local IP address of node 3
influxdb_install_ip1 = 192.168.0.1    # InfluxDB installation node 1
influxdb_install_ip2 = 192.168.0.2    # InfluxDB installation node 2
sftp_install_ip1 = 192.168.0.1    # SFTP installation node 1
sftp_install_ip2 = 192.168.0.2    # SFTP installation node 2
main_path = /opt/cloud    # Microservice running directory. The directory can be customized. Only the first-level directory can be customized. The second-level directory cloud cannot be changed.
node1_ip2 = 100.95.0.1    # IP address used by node 1 (execution node) to communicate with the GaussDB instance. (The IP address must be able to communicate with the GaussDB instance over SSH and ping the GaussDB instance.)
node2_ip2 = 100.95.0.2    # IP address used by node 2 to communicate with the GaussDB instance (The IP address must be able to communicate with the GaussDB instance over SSH and ping the GaussDB instance.)
node3_ip2 = 100.95.0.3    # IP address used by node 3 to communicate with the GaussDB instance (The IP address must be able to communicate with the GaussDB instance over SSH and ping the GaussDB instance.)
log_path = /opt/cloud/logs    # Log directory, which can be customized. Only the first-level directory can be customized. The second-level directory cloud and subsequent directories cannot be changed.
sftp_path = /opt/sftphome    # SFTP data directory, which can be customized. Only the first-level directory can be customized. The second-level directory sftphome cannot be changed.
influx_path = /opt/influxdb    # InfluxDB data directory. Ensure that the InfluxDB user has the execute permission on the parent directories of InfluxDB on the influxDB_install_ip1 and influxDB_install_ip2 nodes. For example, the user can traverse the parent directories and run the chmod a+x /directory_1/directory_2 command on the directories. Only the first-level directory can be customized. The second-level directory influxdb cannot be changed.
docker_path = /opt/docker    # Docker data directory. Only the first-level directory can be customized. The second-level directory docker cannot be changed. If Docker has been installed in a user environment, the user configuration prevails.
backup_path = /opt/backup    # Backup data directory. Only the first-level directory can be customized. The second-level directory backup cannot be changed.
service_group_id = 1010    # Service user ID that is not used
service_user_id = 1010    # Service user group ID that is not used
uninstall_all = no    # Set this parameter to yes during uninstallation. The default value is no.
use_cgroup = no    #  Indicates whether to use cgroup to restrict resources. If the management platform and DRS need to be deployed on the same server, set this parameter to yes. In other scenarios, set this parameter to no.
```


Step 5 Save the configuration and exit.

:wq!

----End

NOTE

An NTP server must be configured between GaussDB Management Platform (TPOPS) servers to ensure time consistency between GaussDB Management Platform (TPOPS) servers.

4.3 Installation Process

4.3.1 Installing the GaussDB Management Platform (TPOPS)

Scenarios

This section describes the detailed procedure for installing the GaussDB Management Platform (TPOPS), installation command output, and installation duration.

Procedure

NOTE

- Ensure that the **root** password for each GaussDB Management Platform (TPOPS) server is the same. For details about how to configure password-free installation, see **Setting Mutual Trust Between Nodes on the Management Plane**.
- The installation takes about 50 minutes, in which the installation of the metadata database GaussDB takes about 20 minutes.
- Do not manually exit during the installation.

Step 1 Go to the directory where the **appctl.sh** file is stored on the **Uploading Software Packages** node as the **root** user.

cd /data/docker-service

Step 2 Run the following command to perform a pre-installation health check. If mutual trust has not been configured between nodes, enter the password of the **root** user as prompted.

sh appctl.sh precheck

If the following information is displayed, the pre-check is successful.

```
*** CHECK BASE SETTINGS ***
 check python               | OK
 check jdk                | OK
 check net tools            | OK
 check expect             | OK
 check secc               | OK: Installed secc.
 check libcgroup            | OK: Installed libcgroup.
 check dos2unix             | OK: Installed dos2unix.
 check custom dir           | OK
 check install mode         | OK: HA mode.
 check IP format          | OK
 check net condition        | OK
```



```
start check node authentication
node not support auto authentication, will input root password
Enter the password of the root user.
check node authentication success.
Now scp precheck files...
scp precheck files done
Now doing precheck for 192.168.0.175
Now doing precheck for 192.168.0.140
Now doing precheck for 192.168.0.243
[WARNING]-[check_drs]===>[192.168.0.140: The switch of use_cgroup is no.You are not allow to install drs together on this machine.]
[WARNING]-[check_dir_total]===>[192.168.0.140: all directory are not belong to same top directory, the check of directory mem size may be incorrect.]
[WARNING]-[check_dir_mount]===>[192.168.0.140: /opt/cloud was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.140: /opt/cloud/logs was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.140: /opt/docker was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.140: /opt/gaussdb was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.140: /opt/influxdb was not mounted.]
[WARNING]-[check_users]===>[192.168.0.140: [service sftpservice] will be used, if the users already exists, we will delete the user before creating it.]
[WARNING]-[check_drs]===>[192.168.0.175: The switch of use_cgroup is no.You are not allow to install drs together on this machine.]
[WARNING]-[check_dir_total]===>[192.168.0.175: all directory are not belong to same top directory, the check of directory mem size may be incorrect.]
[WARNING]-[check_dir_mount]===>[192.168.0.175: /opt/cloud was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.175: /opt/cloud/logs was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.175: /opt/docker was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.175: /opt/gaussdb was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.175: /opt/influxdb was not mounted.]
[WARNING]-[check_users]===>[192.168.0.175: [service sftpservice] will be used, if the users already exists, we will delete the user before creating it.]
[WARNING]-[check_drs]===>[192.168.0.243: The switch of use_cgroup is no.You are not allow to install drs together on this machine.
[WARNING]-[check_dir_total]===>[192.168.0.243: all directory are not belong to same top directory, the check of directory mem size may be incorrect.]
[WARNING]-[check_dir_mount]===>[192.168.0.243: /opt/cloud was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.243: /opt/cloud/logs was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.243: /opt/docker was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.243: /opt/gaussdb was not mounted.]
[WARNING]-[check_users]===>[192.168.0.243: [service sftpservice] will be used, if the users already exists, we will delete the user before creating it.]
Precheck all completed.
```

NOTE

In the pre-check results, [**WARNING**] messages are warning messages and do not block the installation. [**ERROR**] messages are error messages. You need to resolve all error messages before installing the software.

For details about how to handle error messages, see **Pre-check Error Handling**.

Step 3 Run the following commands to install the software. If mutual trust has not been configured between nodes, enter the password of the **root** user as prompted.

sh appctl.sh install

If information similar to the following is displayed, the installation is complete.

```
*** CHECK BASE SETTINGS ***
 check python               | OK
 check jdk                  | OK
 check net tools            | OK
 check expect               | OK
 check secc                 | OK: Installed secc.
 check libcgroup            | OK: Installed libcgroup.
 check dos2unix             | OK: Installed dos2unix.
 check custom dir           | OK
 check install mode         | OK: HA mode.
 check IP format            | OK
```



```
 check net condition           | OK
start check node authentication
node not support auto authentication, will input root password
Enter the password of the root user.
start check host: 192.168.0.1 root password
check host: 192.168.0.1 root password success
start check host: 192.168.0.2 root password
check host: 192.168.0.2 root password success
start check host: 192.168.0.3 root password
check host: 192.168.0.3 root password success
*** UNTAR SERVICE ***
 untar service                | OK
*** BUILD MICRO SERVICE IMAGES ***
 check docker status          | OK
 check service origin packages |  OK
 load docker base image       | OK
 build & save zookeeper       | OK
 build & save kafka           | OK
 build & save common-service  | OK
 build & save monitor-service  |  OK
 build & save rds-ha-admin     | OK
 build & save resource-manager |  OK
 build & save workflow         | OK
 build & save auth             | OK
 build & save gaussdb-console  |  OK
 build & save luban            | OK
 build & save ots              | OK
 build & save GaussDB-open-api |  OK
 build & save GaussDB-instancemanager |  OK
 build & save GaussDB-backupmanager |  OK
 unload docker base image      | OK
*** DISTRIBUTE PACKAGES ***
 /data/docker-service left space |  OK
 check packages                | OK
 distribute gaussdb packages   | OK
 distribute service packages   | OK
 distribute data packages      | OK
 distribute sftp packages      | OK
 distribute remote ip          | OK
*** PREPARE ***
 192.168.0.1                | OK
 192.168.0.2                | OK
 192.168.0.3                | OK
Now doing precheck for 192.168.0.1
Now doing precheck for 192.168.0.2
Now doing precheck for 192.168.0.3
[WARNING]-[check_drs]===>[192.168.0.1: The switch of use_cgroup is no.You are not allow to install drs
together on this machine.]
[WARNING]-[check_dir_total]===>[192.168.0.1: all directory are not belong to same top directory, the
check of directory mem size may be incorrect.]
[WARNING]-[check_dir_mount]===>[192.168.0.1: /opt/cloud was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.1: /opt/cloud/logs was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.1: /opt/docker was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.1: /opt/gaussdb was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.1: /opt/influxdb was not mounted.]
[WARNING]-[check_users]===>[192.168.0.1: [service sftpservice] will be used, if the users already exists, we
will delete the user before creating it.]
[WARNING]-[check_drs]===>[192.168.0.2: The switch of use_cgroup is no.You are not allow to install drs
together on this machine.]
[WARNING]-[check_dir_total]===>[192.168.0.2: all directory are not belong to same top directory, the
check of directory mem size may be incorrect.]
[WARNING]-[check_dir_mount]===>[192.168.0.2: /opt/cloud was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.2: /opt/cloud/logs was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.2: /opt/docker was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.2: /opt/gaussdb was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.2: /opt/influxdb was not mounted.]
[WARNING]-[check_users]===>[192.168.0.2: [service sftpservice] will be used, if the users already exists, we
will delete the user before creating it.]
[WARNING]-[check_drs]===>[192.168.0.3: The switch of use_cgroup is no.You are not allow to install drs
```



```
together on this machine.
[WARNING]-[check_dir_total]===>[192.168.0.3: all directory are not belong to same top directory, the check of directory mem size may be incorrect.]
[WARNING]-[check_dir_mount]===>[192.168.0.3: /opt/cloud was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.3: /opt/cloud/logs was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.3: /opt/docker was not mounted.]
[WARNING]-[check_dir_mount]===>[192.168.0.3: /opt/gaussdb was not mounted.]
[WARNING]-[check_users]===>[192.168.0.3: [service sftpservice] will be used, if the users already exists, we will delete the user before creating it.]
Precheck all completed.
Start to init manifest...
init manifest successful for 192.168.0.1.
init manifest successful for 192.168.0.2.
init manifest successful for 192.168.0.3.
======== 192.168.0.1: patch ========
 patch                | complete
 base_env             | complete
======== 192.168.0.2: patch ========
 patch                | complete
 base_env             | complete
======== 192.168.0.3: patch ========
 patch                | complete
 base_env             | complete
======== 192.168.0.1: base_enviornment ========
 docker               | complete
 InfluxDB             | complete
 sftp                 | complete
 gaussdb              | complete
======== 192.168.0.2: base_enviornment ========
 docker               | complete
 InfluxDB             | complete
 sftp                 | complete
 gaussdb              | complete
======== 192.168.0.3: base_enviornment ========
 docker               | complete
 InfluxDB             | complete
 sftp                 | complete
 gaussdb              | complete
======== 192.168.0.1: CommonbaseData ========
 common-base          | complete
======== 192.168.0.2: CommonbaseData ========
 common-base          | complete
======== 192.168.0.3: CommonbaseData ========
 common-base          | complete

======== 192.168.0.1: PlatformData ========
 platform-data        | complete
======== 192.168.0.2: PlatformData ========
 platform-data        | complete
======== 192.168.0.3: PlatformData ========
 platform-data        | complete
======== 192.168.0.1: Zookeeper ========
 zookeeper            | complete
 GaussDB-feature-data | complete
 GaussDB-data         | complete
======== 192.168.0.2: Zookeeper ========
 zookeeper            | complete
 GaussDB-feature-data | complete
 GaussDB-data         | complete
======== 192.168.0.3: Zookeeper ========
 zookeeper            | complete
 GaussDB-feature-data | complete
 GaussDB-data         | complete
======== 192.168.0.1: Kafka ========
 kafka                | complete
======== 192.168.0.2: Kafka ========
 kafka                | complete
======== 192.168.0.3: Kafka ========
 kafka                | complete
```



```
======== 192.168.0.1: docker_service ========
  common-service          | complete
  monitor-service         | complete
  rds-ha-admin            | complete
  resource-manager        | complete
  workflow                | complete
======== 192.168.0.2: docker_service ========
  common-service          | complete
  monitor-service         | complete
  rds-ha-admin            | complete
  resource-manager        | complete
  workflow                | complete
======== 192.168.0.3: docker_service ========
  common-service          | complete
  monitor-service         | complete
  rds-ha-admin            | complete
  resource-manager        | complete
  workflow                | complete
======== 192.168.0.1: gaussdb_service ========
  auth                    | complete
  gaussdb-console         | complete
  luban                   | complete
  ots                     | complete
  GaussDB-open-api        | complete
  GaussDB-instancemanager | complete
  GaussDB-backupmanager   | complete
======== 192.168.0.2: gaussdb_service ========
  auth                    | complete
  gaussdb-console         | complete
  luban                   | complete
  ots                     | complete
  GaussDB-open-api        | complete
  GaussDB-instancemanager | complete
  GaussDB-backupmanager   | complete
======== 192.168.0.3: gaussdb_service ========
  auth                    | complete
  gaussdb-console         | complete
  luban                   | complete
  ots                     | complete
  GaussDB-open-api        | complete
  GaussDB-instancemanager | complete
  GaussDB-backupmanager   | complete
Installation progress [72/72] ==> 100.00%
Upload sftp packages successful for 192.168.0.1
Log in to the GaussDB management platform and choose Task Center in the navigation pane to check the status of the installation package upload task.

Welcome to TPOPS:

    https://{EIP}:8443/gaussdb/#/login
    EIP: Elastic IP address of any TPOPS node
```

----End

CAUTION

Both the upgrade and uninstallation depend on the **docker-service** directory. Exercise caution when deleting the directory.

If you need to uninstall GaussDB Management Platform (TPOPS) after the **docker-service** directory is deleted, see **Performing Uninstallation After the docker-service Directory Is Deleted** instead of **Uninstalling GaussDB Management Platform (TPOPS)**.

4.4 Post-installation Check

Step 1 Use a browser to log in to GaussDB Management Platform (TPOPS) by referring to the following URL:

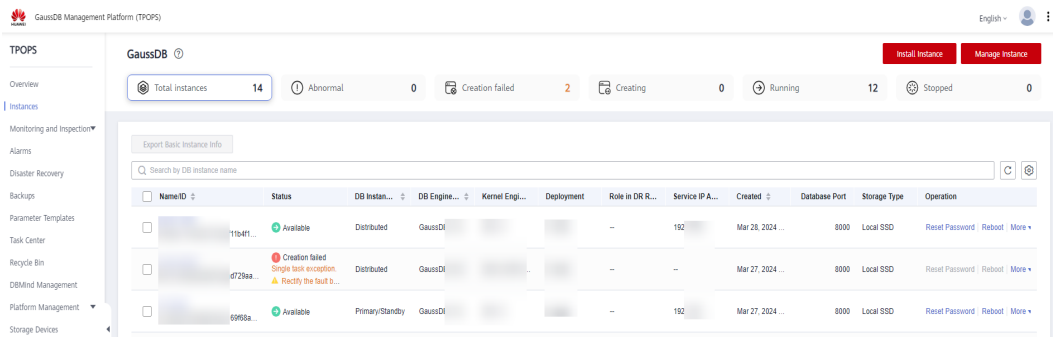
https://{*EIP*}:8443/gaussdb/#/login

Where,

- {*EIP*} indicates the EIP of any node where GaussDB Management Platform (TPOPS) is installed.
- Use the default administrator account and password for login. The default account is **admin**. For details about its default password, see ***GaussDB Management Platform (TPOPS) Account List 01***.

Step 2 Check whether the page is properly displayed. If it is, GaussDB Management Platform (TPOPS) is successfully installed.

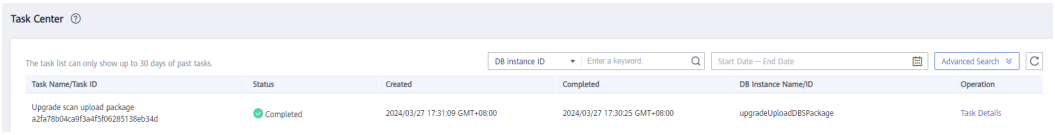
Figure 4-3 GaussDB Management Platform (TPOPS) home page



Step 3 Click **Task Center** to view the status of the GaussDB installation package upload task.

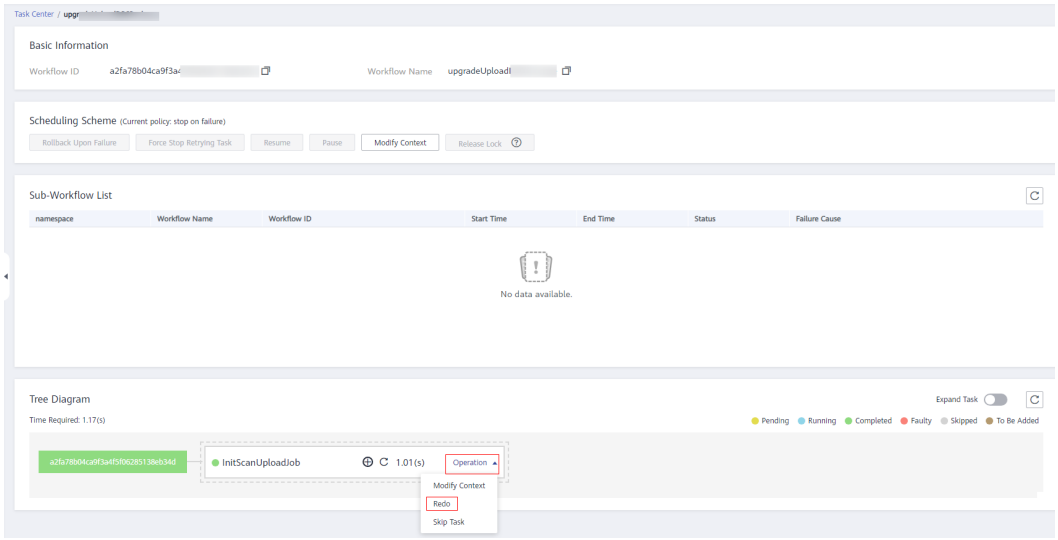
Step 4 If **Completed** is displayed, the installation package is uploaded successfully.

Figure 4-4 Installation package uploaded



Otherwise, choose **Task Center > Task Details > Operation > Redo** to re-execute the task. If the upload fails again, contact **technical support** engineers to upload the GaussDB installation package.

Figure 4-5 Task retry



----End

5 Installing Instances

5.1 System Requirements

OS Requirements

Table 5-1 Supported OSs and software packages

OS	Compatibility	Architecture
Kylin V10	SP1: 4.19.90-23.8.v2101.ky10.aarch64	Arm
	SP2: 4.19.90-24.4.v2101.ky10.aarch64	
	SP1: 4.19.90-23.8.v2101.ky10.x86_64	x86
	SP2: 4.19.90-24.4.v2101.ky10.x86_64	
UnionTech	4.19.90-2201.4.0.0135.up1.uel20.x86_64	x86
	4.19.90-2201.4.0.0135.up1.uel20.aarch64	Arm

CAUTION

Only the English operating system is supported.

Software Requirements

Table 5-2 Software requirements for GaussDB data plane nodes

Software	Specifications
Python	Python 3.7.9. Run the following command to check the Python version: **python3 --version** If the queried version is not Python 3.7.9, install Python 3.7.9 by referring to **Installing Python 3 on the Host**.

Hardware Requirements

Type	Recommended Configuration
x86	2 x 24-core 6248R CPUs, 3.0 GHz, 24 x 32 GB memory, 2 x 960 GB SATA SSDs, 24 x 960 GB SATA SSDs, SR450C-M 2G(Avago3508), 2 x 2 x 10GE/25GE
Arm	2 x 64-core Kunpeng 920 CPUs, 2.6 GHz,32 x 32 GB memory, 2 x 960 GB SATA SSDs, 24 x 960 GB SATA SSDs, 1 x Avago3508, 2 x 2 x 25GE

CAUTION

Use a host whose hardware specifications are greater than or equal to 8 vCPUs and 64 GB memory.

GaussDB instances only support seven levels of specifications: 8 vCPUs, 64 GB memory; 16 vCPUs, 128 GB memory; 32 vCPUs, 256 GB memory; 64 vCPUs, 512 GB memory; 96 vCPUs, 768 GB; 128 vCPUs, 1024 GB; and 196 vCPUs, 1569 GB memory (DBMind only). During instance installation, proper instance specifications are matched based on the minimum specifications of the selected host. If the host specifications are less than the minimum instance specifications, instances cannot be installed.

5.2 Modifying OS Configurations

5.2.1 Constraints

- Perform the operations in **Configuring OS Firewalls** to **Setting umask** as the **root** user on all instance nodes. After the operations are complete, log out of the system as the **root** user in a timely manner to prevent misoperations.

- You need to restart the devices for the configurations in **Configuring OS Firewalls** and **Disabling the Swap Memory** to take effect. You can restart the devices after the two operations are complete.
- After a DB instance is installed, use the deadline scheduling mode for SATA/SAS SSDs and the none scheduling mode for NVMe SSDs.

5.2.2 Configuring OS Firewalls

Perform the installation when the firewall is disabled. To disable the firewall, perform the following steps:

Step 1 Run the following command to check whether the firewall is disabled:

systemctl status firewalld

```
[root@host-192-168-0-36 data]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Fri 2023-11-24 15:09:25 CST; 24h ago
     Docs: man:firewalld(1)
 Main PID: 970 (code=exited, status=0/SUCCESS)
```

The red box in the preceding figure shows the firewall status.

- If the status is **active (running)**, the firewall is not disabled. Go to **Step 2**.
- If the status is **inactive (dead)**, the firewall is disabled.

Step 2 Run the following commands to disable the firewall and disable the firewall from starting upon system startup:

systemctl stop firewalld.service

systemctl disable firewalld.service

Step 3 Set the value of **SELINUX** in the **/etc/selinux/config** file to **permissive**.

NOTE

Generally, enabling SELinux improves system security but may cause program running failures. To ensure successful installation, you are advised to set this parameter to **permissive**.

1. Run the **vi** command to open the **config** file

   vi /etc/selinux/config

2. Change the value of **SELINUX** to **permissive**.

   SELINUX=permissive

3. Press **Esc** and run the **:wq!** command to save the change and exit.

Step 4 Run the following command to reboot the OS:

reboot

Step 5 Repeat **Step 1** to **Step 4** on other hosts.

> **CAUTION**
>
> After the database is installed, iptables is enabled. The database services, protocols, IP addresses, and ports are added to the firewall whitelist of each database host. You can run the **iptables -vnL** command to view the whitelist.
>
> Exercise caution when running the **iptables -F** command. Otherwise, some functions, such as data restoration, may be unavailable.

----End

5.2.3 Disabling the Swap Memory

Step 1 Log in to each host as the **root** user.

Step 2 Run the following command to temporarily disable the swap memory:

swapoff -a

Step 3 Comment out the **swap** startup item in **/etc/fstab** and restart the server to permanently disable the swap memory.

1. Run the following command to access **/etc/fstab**:

   vi /etc/fstab

2. Locate the **swap** startup item and enter **#** in front of the target command line to comment it out.

   For a Kylin OS, the command line is as follows:

   ```
   # /dev/mapper/klas-swap   none                    swap    defaults        0 0
   ```

   For a UnionTech OS, the command line is as follows:

   ```
   # /dev/mapper/uos-swap   none                    swap    defaults        0 0
   ```

3. Press **Esc** and run the following command to save the modification and exit:

   :wq!

4. Run the following command to restart the host:

   reboot

----End

5.2.4 Setting Character Set Parameters

The character set of each host must be the same.

Step 1 Run the **vi** command to open the **/etc/profile** file.

vi /etc/profile

Step 2 Add the following field:

export LANG=en_US.UTF-8

Step 3 Press **Esc** and run the **:wq!** command to save the change and exit.

Step 4 Make the **/etc/profile** file take effect.

source /etc/profile

Step 5 Run the **vi** command to open the **/etc/sysconfig/i18n** file.

vi /etc/sysconfig/i18n

Step 6 Change the value of **LANG** to **en_US.UTF-8**.

Step 7 Run the following command to ensure that the character set of the installation user takes effect:

source /etc/sysconfig/i18n

NOTE

If the **/etc/sysconfig/i18n** file does not exist, open the **/etc/locale.conf** file.

vi /etc/locale.conf

Change the value of **LANG** to **en_US.UTF-8** and run the **source /etc/locale.conf** command.

----End

5.2.5 Setting the Clock Source

Clock sources of each host need to be synchronized for the installation instance, so the time synchronization will be forced to be verified when the hosts are online. You need to set the clock sources to ensure that the time synchronization is within 1 second.

You can set the clock sources using the chrony and NTP. For details about how to set the clock sources, see **Setting Clock Sources**.

NOTICE

1. If you do not use chrony or NTP to set a clock source, skip the task **NebulaHostDetectionTask** in the host adding task flow after manually confirming the host time synchronization.
2. If NTP is configured in manual synchronization mode, the host standardization check cannot be performed. In this case, skip the task **NebulaHostDetectionTask** in the host adding task flow.

5.2.6 Setting the NIC MTU Value

NOTE

You are advised to set the MTU value of the host NIC whose architecture type is x86 to **1500** and that of the host NIC whose architecture type is Arm to **8192**.

Step 1 Run the **ifconfig** command to view the NIC that has a bound IP address, for example, **eth0**.

Step 2 Run the following command to view the MTU value displayed next to the NIC bound to the IP address:

ifconfig eth0 | grep mtu

Command output:

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
```


Step 3 If the MTU value of the command output is **1500**, check whether the MTU value meets the recommended value.

- Yes: No modification is required.
- No: Change the MTU value of the NIC by referring the following steps.

Step 4 Run the following command to open the **ifcfg-*** file:

vi /etc/sysconfig/network-scripts/ifcfg-*

*indicates the NIC queried in **Step 1**, for example, **eth0**.

Step 5 Press **i** to enter the editing mode and add the following statement to set the MTU value of the NIC. For example, set the MTU value to **1500**.

MTU=1500

Step 6 Press **Esc** and enter **:wq!** to save the change and exit.

Step 7 Run the following command to restart the network:

service network restart

----End

5.2.7 Installing OpenSSH

Scenarios

Before bringing a host online, you need to allow the host to be connected over SSH and can run the **scp** command to copy the installation package. Otherwise, the host may fail to go online.

Prerequisites

You need to configure the Yum repository for the Kylin environment. For details about how to configure the Yum repository, see **Configuring a Yum Repository**.

Procedure

Step 1 Run the following command to check whether OpenSSH is installed:

ssh -V

Information similar to the following is displayed:

```
OpenSSH_8.2p1, OpenSSL 1.1.1f  31 Mar 2020
```

Step 2 If no OpenSSH version is displayed or a message is displayed indicating that the command does not exist, run the following command to install OpenSSH:

yum -y install openssh-server

Step 3 After the installation is complete, run the following command to start the OpenSSH service:

systemctl start sshd

----End

5.2.8 Checking Expect

Scenarios

Before bringing a host online, you need to allow the **expect** command to be used to connect to the host over SSH. Otherwise, the host may fail to go online.

Prerequisites

You need to configure the Yum repository for the Kylin environment. For details about how to configure the Yum repository, see **Configuring a Yum Repository**.

Procedure

Step 1 Run the following command to check whether expect is installed on the host:

expect -v

Information similar to the following is displayed:

```
expect version 5.45.4
```

Step 2 If no expect version is displayed or a message is displayed indicating that the command does not exist, run the following command to install expect:

yum install expect -y

Step 3 Check whether the software has been installed.

expect -v

----End

5.2.9 Configuring sshd_config

Scenarios

Before bringing a host online, set **GSSAPIAuthentication** to **no** in the **sshd_config** file. Otherwise, the host may fail to go online.

Procedure

Step 1 Use vi to open the **/etc/ssh/sshd_config** file.

vi /etc/ssh/sshd_config

Step 2 Set **GSSAPIAuthentication** to **no**.

GSSAPIAuthentication no

Step 3 Press **Esc** and run the **:wq!** command to save the change and exit.

Step 4 Restart SSH.

systemctl restart sshd.service

----End

5.2.10 Setting umask

Step 1 Run the following command to view the umask value:

umask

- If the command output is less than or equal to **0022**, the umask setting is correct.
- If the command output is greater than **0022**, perform the following steps to set umask.

Step 2 Run the following command to open the **bashrc** file:

vi /etc/bashrc

Step 3 Add a line at the bottom of the **bashrc** file to ensure the value of umask is **0022**.

umask 0022

Step 4 Press **Esc** and run the **:wq!** command to save the change and exit.

Step 5 Run the following command to make the changes take effect on umask:

source /etc/bashrc

Step 6 If the Agent has been installed on the host, run the following command to restart the Agent:

touch /home/Ruby/need_shut_down.touch

Otherwise, go to step 7.

Step 7 Run the **umask** command again. If **0022** is displayed, the umask is set successfully.

----End

5.3 Preparing the Initialization Environment

5.3.1 Preparing Disks

GaussDB allows SSDs supporting SAS or SATA and the NVMe protocol (SSDs and NVMe disks for short) to be used as the primary storage device of the database.

You are advised to configure RAID 10 for data disks. RAID is not supported for NVMe drives. You can configure RAID by following instructions in the manual provided by the hardware manufacturer or using methods found on the Internet. Set **Disk Cache Policy** to **Disabled** to avoid data loss in an unexpected power-off.

The total capacity of data disks must be greater than 300 GB. (The disk capacity must be greater than 300 GB after the RAID 10 is configured.) Otherwise, no instances can be created.

Ensure that all disks (including system disks and data disks) are of the same type.

NOTICE

- During instance installation, the management system uses all disks except the system disks on the node as data disks.
- Clear the data disks in advance and keep them as raw disks (such as **sdb** and **nvme0n1** in the command output). Do not create data disks in advance.
- If there are NVMe disks in the data disks, NVMe disks are preferred. If a node has both SSD and NVMe disks, only NVMe disks are used.
- NVMe system disks are not supported.

The following shows the use of different types of disks:

```
~# lsblk
NAME             MAJ:MIN RM  SIZE   RO TYPE MOUNTPOINT
 sda               0:8    0 557.9G   0 disk
 |_sda1            8:1    0     1G   0 part /boot
 |_sda2            8:2    0 556.9G   0 part
   |_klas-root    253:0    0    70G   0 lvm  /
   |_klas-swap    253:1    0     4G   0 lvm
   |_klas-home    253:2    0 482.9G   0 lvm  /home
 sdb               8:16   0   3.6T   0 disk
 nvme0n1           259:0    0   1.5T   0 disk
```

NOTICE

The system disks can not be divided into multiple disks. For example, partitions are planned on **sdb** to mount **uos-root**.

```
[root@localhost ~]# lsblk
 NAME                MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
 sda                   8:0    0 894.3G  0 disk
 ├─sda1                8:1    0   600M  0 part /boot/efi
 ├─sda2                8:2    0     1G  0 part /boot
 └─sda3                8:3    0 892.7G  0 part
 ├─uos-root          253:0    0   1.8T  0 lvm  /
 └─uos-swap            253:1    0     4G  0 lvm
 sdb                   8:16   0 894.3G  0 disk
 └─sdb1                8:17   0 894.3G  0 part
  └─uos-root          253:0    0   1.8T  0 lvm  /
```

5.3.2 Preparing Disk Directories

Step 1 Prepare system disks and data disks.

It is recommended that two system disks form a RAID 1 group, which is mainly used by an OS.

Disks are automatically partitioned for installation instances. All disks except the system disk are used as data disks and mounted to the **/var/chroot** subdirectory. Before using the data disks, ensure that they are clean and available. You can run the following command to check the data disks: **pvcreate --test /dev/vdb**

```
~# pvcreate --test /dev/vdb
  TEST MODE: Metadata will NOT be updated and volumes will not be (de)activated.
  Physical volume "/dev/vdb" successfully created.
```

If the preceding information is displayed, the disks are cleared and available.

Step 2 Before adding a host, ensure that the **/var/chroot** directory does not exist or is empty. If the host contains some instances or has the Agent installed, you do not need to clear the sandbox directory.

Run the following command to check whether the sandbox directory is empty:

ls -Al /var/chroot

- If the following information is displayed, the **/var/chroot** directory is empty, which meets the requirement.

```
~# ll /var/chroot/
total 0
```

or

```
~# ll /var/chroot/
ls: cannot access '/var/chroot/': No such file or directory
```

- If the directory is not empty, run the following command to clear it:

rm -rf /var/chroot/{*,.*}

If "rm: cannot remove '***': Operation not permitted" is displayed, the file cannot be deleted. The possible cause is that a disk is mounted. Unmount the disk and clear the sandbox directory again.

- Run the check command again to ensure that the sandbox directory does not exist or is empty.

----End

NOTE

The management program automatically creates disks and mounts logical volumes to the data directory, log directory, and backup directory in **/var/chroot**. The directory file system format is Ext4.

5.3.3 Configuring Networks

The network planes are divided as follows:

- Management plane: used for communication between GaussDB Management Platform (TPOPS) and database nodes, and database instance management.
- Service plane: used for communication between services and database instances (distributed CNs and primary/standby DNs).
- Data plane: used for communication between primary and standby DNs, communication between CNs and DNs, backup and restoration, DBMind management, and streaming DR communication between active and standby instances.

  In lightweight deployment, management and data services can be deployed on different planes or on the same plane (that is, one to three IP addresses are supported). You can select single plane (single IP address), two planes (dual IP addresses), or three planes (three IP addresses) based on service security, network performance, and equipment room conditions. Three-plane network isolation is recommended to ensure service security and network performance.

CAUTION

1. It is recommended that the management, service, and data networks be physically isolated by dividing network planes using network switches.
2. To obtain the management IP address, the Agent process queries the default gateway and creates a socket to connect to the default gateway when it starts. Then, it obtains the IP address of the local socket as the listening address of the Agent. If the network configuration is incorrect, the Agent cannot provide services. You can run the **ip route show default** command to view the default gateway of the current server. If the required route is missing, run the **route add default gw <*ip*>** command to add the default route.
3. If the instance networks are bonded, ensure that the bond modes are consistent. If the bond modes are inconsistent, clusters may work improperly.

5.3.4 Checking Python Dependency Packages

For details about the Agent dependency versions, see **Dependent Python Library Versions**. If a Python dependency package that is not in the dependency scope has been installed in the system, the package will be automatically overwritten when **Adding a Host** is executed.

You can run the following command to view the current Python package version:

pip freeze --all

NOTE

If **-bash: pip: command not found** is returned, install the PIP tool. For details about how to install the PIP tool, contact the system provider or see the **PIP official website**.

5.3.5 (Optional) Setting the Log Directory for the Management Program

Suggestions

By default, database management program logs are displayed in **/home/Ruby/log** in **/var/log**. Create a subdirectory in **/var/log** and set the soft link **/home/Ruby/log**.

You are advised to set a separate partition for **/var/log** as follows:

```
~# lsblk
NAME            MAJ:MIN RM  SIZE  RO TYPE MOUNTPOINT
 sda              0:8    0   50G   0 disk
 |_sda1           8:1    0    1G   0 part /boot
 |_sda2           8:2    0   49G   0 part
   |_klas-root    253:0    0   20G   0 lvm  /
   |_klas-log     253:1    0    1G   0 1vm
   |_klas-home    253:2    0   18G   0 1vm  /home
   |_klas-log     253:3    0   10G   0 1vm  /var/log
```

Create partitions for **/var/log** before performing **Adding a Host**.

If you need to store the database management program logs in another directory, you can set the directory after performing **Adding a Host**. The following uses the **/gauss/agent_log** directory as an example:


```
~# lsblk
vdb                  252:16   0  300G  0 disk
├─klas2-lv_gauss 253:2     0  100G  0 lvm  /gauss
└─klas2-lv_data  253:3     0  200G  0 lvm  /data
```

Procedure

Step 1 Log in to each host as the **root** user.

Step 2 Run the following commands to change the path of the soft link **/home/Ruby/log** to **/gauss/agent_log**:

mkdir -m 700 -p /gauss/agent_log

cd /home/Ruby

cp -fr log/* /gauss/agent_log

rm -fr log

ln -s /gauss/agent_log /home/Ruby/log

chown -R Ruby:Ruby /gauss/agent_log

Step 3 Run the following command to restart the process for the modification to take effect:

ps -ef | grep 'python'| grep -v grep | awk '{print $2}' | xargs kill -9

Step 4 Run the following commands to check whether the setting is successful:

ls -l /home/Ruby/log

```
lrwxrwxrwx 1 root root 16 Jan 21 09:25 /home/Ruby/log -> /gauss/agent_log
```

ls -ld /gauss/agent_log

```
drwx------ 6 Ruby Ruby 4096 Jan 21 09:29 /gauss/agent_log
```

If the preceding information is displayed, the modification is successful.

----End

5.3.6 (Optional) Preparing Floating IP Addresses

GaussDB Management Platform (TPOPS) provides the floating IP address function when you create a non-single-node instance in primary/standby deployment. To use this function, you need to prepare an available floating IP address. After the instance is installed, you can use the floating IP address to access DN node data. For example, during the primary/standby switchover, the same floating IP address is configured for the three nodes in primary/standby deployment. After the primary/standby switchover, the floating IP address can still be used to connect to the data nodes.

NOTE

- To use the floating IP address function, select the kernel installation package of 503.1.0.SPC1200, 503.1.0.SPC1300, or 503.2.0 or later.
- Ensure that the configured floating IP address is valid and unique. If the IP address is invalid, an instance cannot be accessed using the floating IP address after being installed. The floating IP address must be in the same network segment as **virtualIp**.
- After an instance is installed, the floating IP address cannot be changed.
- Ping the floating IP address on the node where the instance is to be installed. If the floating IP address cannot be pinged, the floating IP address is not in use.

5.4 Adding a Data Center

Scenarios

GaussDB Management Platform (TPOPS) provides the data center management capability so that users can add equipment rooms.

Prerequisites

You have operation rights for adding a data center.

Procedure

Step 1 **Log in to GaussDB Management Platform (TPOPS).**

Step 2 Choose **Platform Management** > **Data Centers**. The **Data Centers** tab page is displayed.

Step 3 Click **Add Data Center** in the upper right corner of the page.

Figure 5-1 Add Data Center

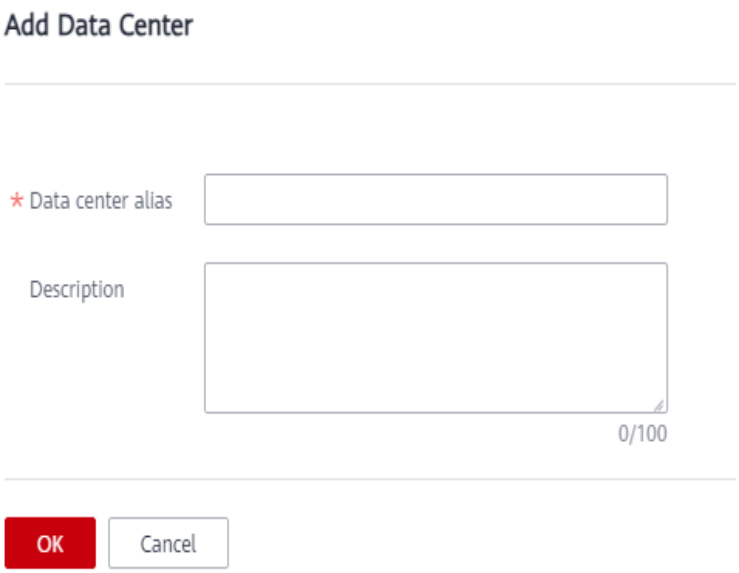


Table 5-3 Parameters for adding a data center

Parameter	Mandatory	Description
Data center alias	Yes	Specifies the equipment room alias displayed in the data center management list. A data center alias must be unique and can contain up to 20 characters. Letters, numbers, hyphens (-), and underscores (_) are supported.
Description	No	The description contains a maximum of 100 characters.

Step 4 Click **OK** to add a data center.

You can also click **Cancel** to cancel the operation.

----End

5.5 Adding a Host

5.5.1 Precautions

Scenarios

GaussDB Management Platform (TPOPS) allows you to manage hosts online. You can select and use a host to create a database in scenarios such as creating a DB instance, restoring a DB instance, and adding shards.

To bring a host online, install the OS, initialize the network, and initialize disks for the host. Then, the host can be managed as a host that can provision instances.

Prerequisites

- If an instance is already available on the host and is to be managed by GaussDB Management Platform (TPOPS), ensure that the instance is normal when the host goes online.
- You have the permission to add a host.
- Ensure that the clock sources of hosts are synchronized. For details about how to set the clock sources, see **Setting the Clock Source**.
- Flash storage is a whitelist function. The feature name in the whitelist is **gaussdbv5_feature_supportDorado**. For details about how to enable the whitelist, see **How Do I Enable or Disable the Whitelist?**.
- A host can be brought online using the **root** password or SSH private key (recommended). You must select either of them.
- If the SSH private key is used as the connection credential for bringing a host online, copy the public key content to **~/.ssh/authorized_keys**. For details, see "Platform Management" > "Data Center Management" > "Key Management" > "Downloading the Public Key" in ***GaussDB Management Platform (TPOPS) User Guide (for Lightweight GaussDB)***.

Constraints

- The **root** user is allowed to log in to the hosts.
- You can import hosts in batches. A maximum of 50 hosts can be imported at a time.
- All GaussDB Management Platform (TPOPS) nodes can connect to the host through SSH. You can run the **scp** command to copy the installation package. For details, see **Installing OpenSSH**.
- The host allows the **expect** command to process the SSH interaction. For details, see **Checking Expect**.
- Set **GSSAPIAuthentication** to **no** in the **sshd_config** file on the host. For details, see **Configuring sshd_config**.
- If no instance is available on the host, the sandbox directory of the host must be empty. That is, the **/var/chroot** directory does not exist or is empty.
- The value of **umask** is less than or equal to **0022**.
- A host cannot be added to multiple GaussDB Management Platform (TPOPS) systems repeatedly.
- If there is any instance on a host, the name of the equipment room selected when you add the host must be the same as the AZ name used during instance installation. For details about how to check the name of the AZ used for instance installation, see **Checking the Name of the AZ Used for Installing an Instance**.

5.5.2 Adding a host

Step 1 **Log in to GaussDB Management Platform (TPOPS).**

Step 2 Choose **Platform Management > Data Centers**. The **Data Centers** page is displayed.

Step 3 Click the **Data Center Alias/ID** value for the host to be queried.

Step 4 Click **Add Host**.

Figure 5-2 Adding a Host

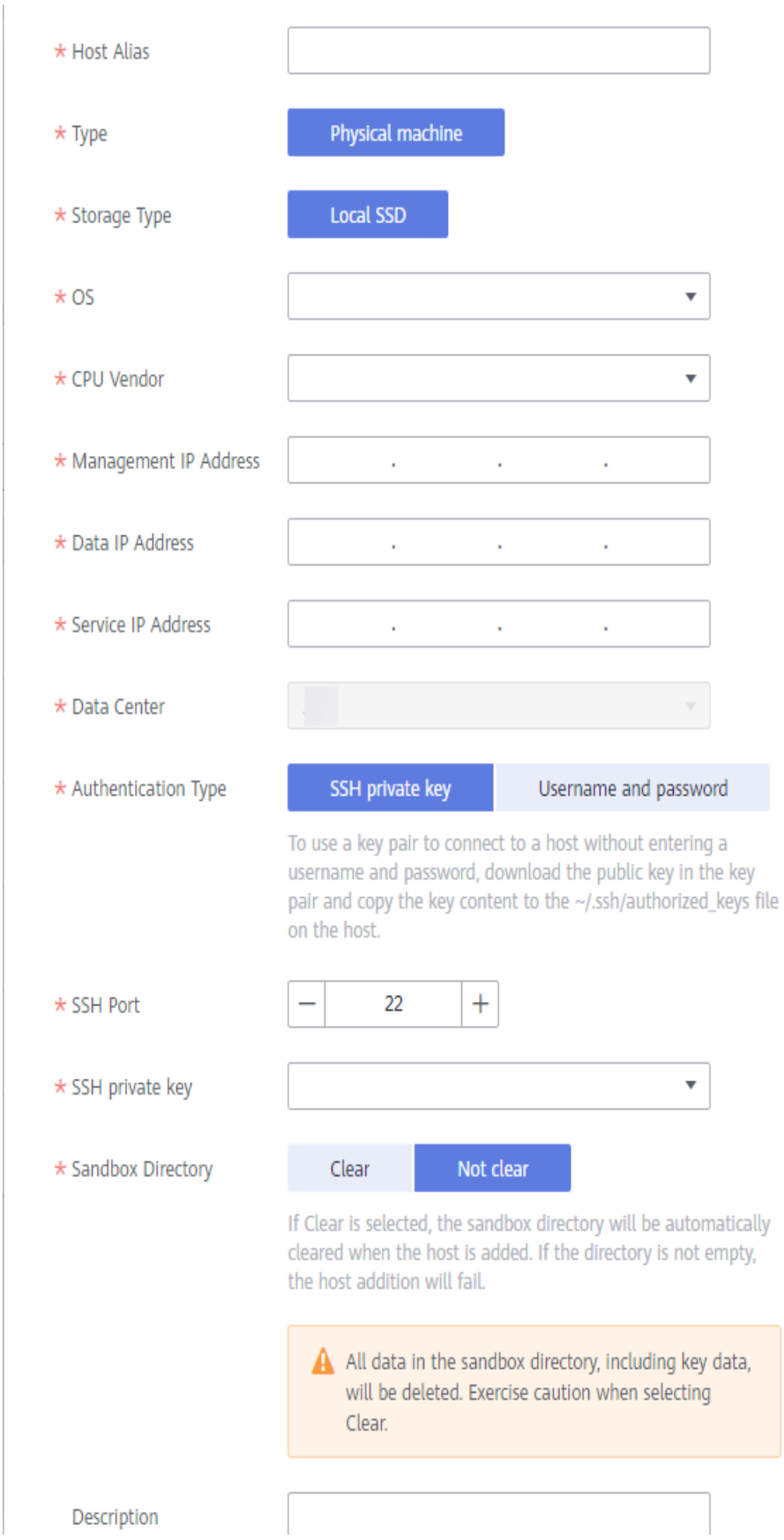


Table 5-4 Parameters for adding a host

Parameter	Description
Host Alias	Specifies the host alias displayed in the host management list. The alias can contain a maximum of 64 bytes.
Type	Select **Physical machine**.
Storage Type	Select **Flash storage** or **Local SSD**.
OS	Select **Kylin** or **UOS**.
CPU Vendor	Select **Kunpeng**, **Intel**, or **Hygon**.
Management IP Address	Enter the management IP address of the host. Three planes (three IP addresses) are recommended. That is, three NICs carry networks of three planes, respectively. If a NIC carries networks of different planes, the physical isolation between the networks of different planes may be damaged. If a single plane (single IP address) is used, there is a risk that the network is not isolated. You can use iptables to isolate the network.
Data IP Address	Enter the data IP address of the host. If a single plane (single IP address) or two planes (dual IP addresses) are used, the host data IP address is the host management IP address.
Service IP Address	Enter the service IP address of the host. If a single plane (single IP address) is used, the host service IP address is the management IP address of the host. If two planes (dual IP addresses) are used, the host service IP address is on an independent network plane, and the host data IP address and host management IP address share a network plane.
SSH Port	The port number ranges from 1 to 65535. The default value is **22**. The value cannot be **8002** or **12017**.
Data Center	The current data center is selected by default and cannot be changed.
Authorization Type	The SSH private key and username and password modes are supported. The SSH private key mode is recommended.
Description	The description can contain up to 100 characters.

Parameter	Description
Sandbox Directory	• **Clear**: The sandbox directory is automatically cleared when you add a host. All data in the sandbox directory, including key data, will be deleted. Exercise caution when performing this operation. • **Not clear**: The sandbox directory is not cleared when you add a host. A host may fail to be added if the sandbox directory is not empty.

Step 5 Click **OK** to add a host. Wait for 3 to 5 minutes until the host is added.

You can also click **Cancel** to cancel the operation.

NOTE

If a host fails to be added because it fails to pass the standardization check, you can view the failed check items in the host standardization check result, reconfigure the host, add the host again, and perform the standardization check on the host again.

For details about host standardization check items and configuration methods, see sections "Standardization Check Items for Host Management" and "Configurations for Host Standardization Check" in "Appendix" in the GaussDB Management Platform (TPOPS) user guide.

----End

5.5.3 Batch Importing Hosts

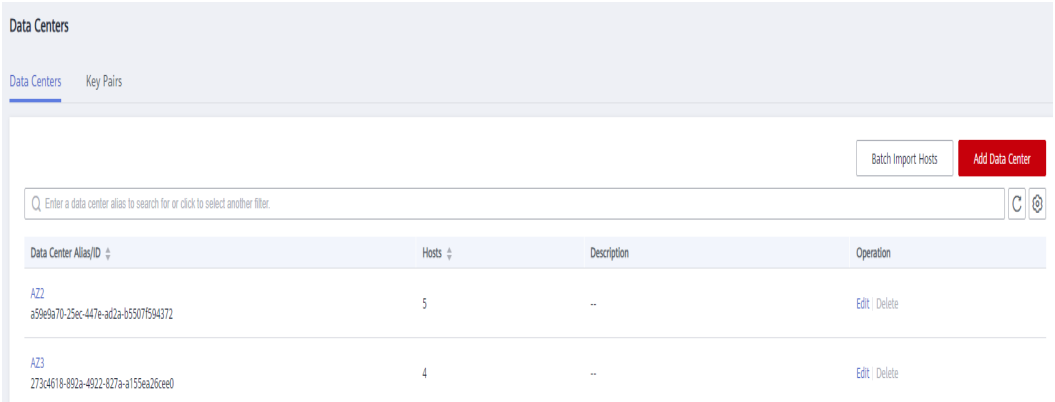
Procedure

Step 1 **Log in to GaussDB Management Platform (TPOPS).**

Step 2 Choose **Platform Management > Data Centers**.

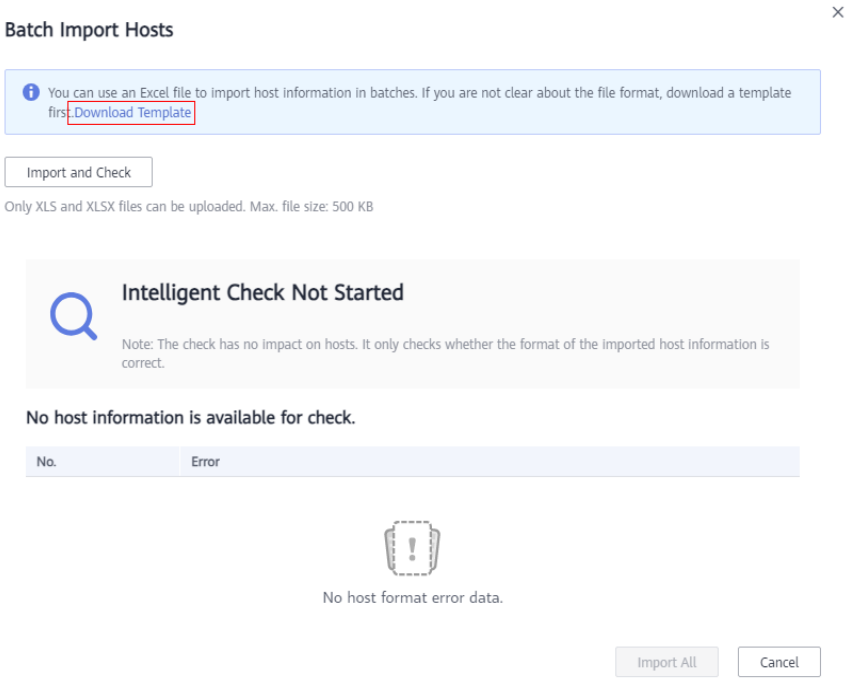
Step 3 Click **Batch Import Hosts**.

Figure 5-3 Batch importing hosts



Step 4 Click **Download Template**.

Figure 5-4 Downloading the template



Step 5 Download and fill in the host file template. The template contains the parameters as shown in **Table 5-5**.

Table 5-5 Template parameters

Parameter	Details	Description
name	Host alias	Specifies the host alias displayed in the host management list. The value can contain a maximum of 64 bytes.
type	Host type	Select **BMS**.
os_type	Operating system	Select Kylin or Uniontech.
cpu_spec	CPU type	Select **Kunpeng**, **Intel**, or **Hygon**.
storage_type	Storage type	Select **Flash storage** or **Local SSD**. **Flash storage** is displayed only when the **gaussdbv5_feature_supportDorado** whitelist is enabled.

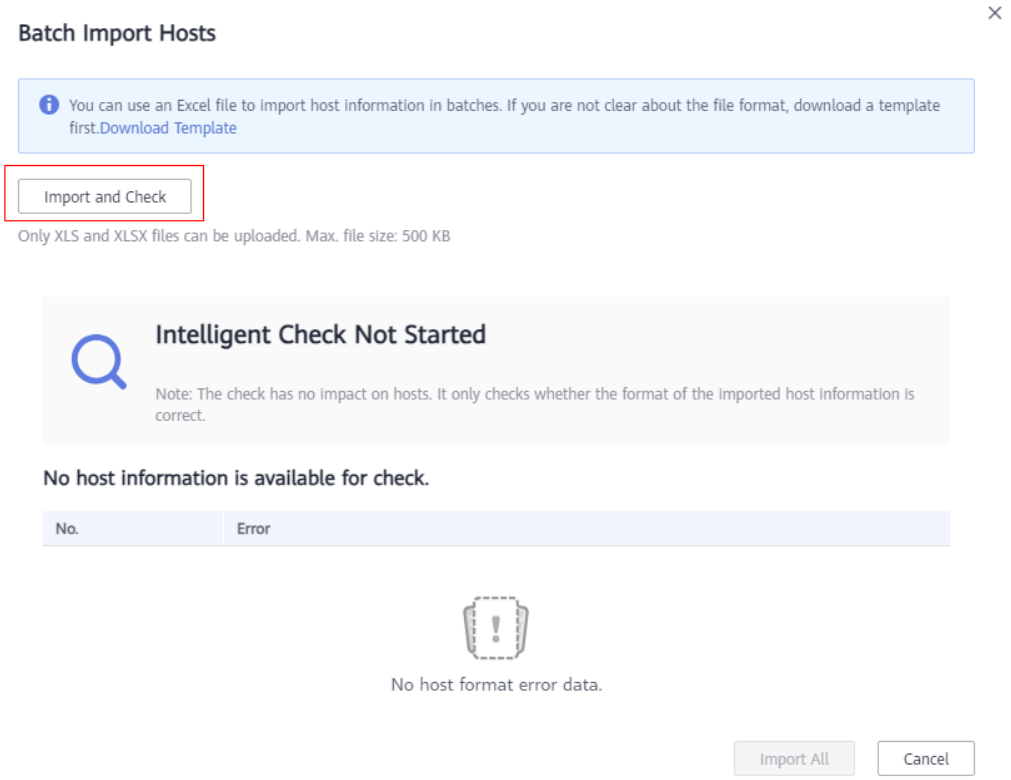
Parameter	Details	Description
manage_ip	Management IP Address	Enter the management IP address of the host. Three planes (three IP addresses) are recommended. That is, three NICs carry networks of three planes, respectively. If a NIC carries networks of different planes, the physical isolation between the networks of different planes may be damaged. If a single plane (single IP address) is used, there is a risk that the network is not isolated. You can use iptables to isolate the network.
data_ip	Data IP address	Enter the data IP address of the host. If a single plane (single IP address) or two planes (dual IP addresses) are used, the host data IP address is the host management IP address.
virtual_ip	Service IP address	Enter the service IP address of the host. If a single plane (single IP address) is used, the host service IP address is the management IP address of the host. If two planes (dual IP addresses) are used, the host service IP address is on an independent network plane, and the host data IP address and host management IP address share a network plane.
ssh_port	SSH port	The port number ranges from **1** to **65535**. The value cannot be **8002** or **12017**.
data_center_id	Data center ID	ID of the data center where the host is installed.
os_root_pwd	User password	Specifies the password of the **root** user for the host. If the password of the **root** user is used to log in to the host, this field is mandatory and the **ssh_key_id** field is left blank.
ssh_key_id	SSH private key	Specifies the SSH private key ID. If the SSH private key is used to connect to the hosts, this field is mandatory and the **os_root_pwd** field is left blank. The SSH private key mode is recommended.
description	Description	The description can contain up to 100 characters.

Parameter	Details	Description
is_force_rm_chroot_dir	Sandbox directory	The value **true** indicates that the sandbox directory is cleared, and the value **false** indicates that the sandbox directory is not cleared. If this parameter is left blank, **false** is selected by default.
is_use_ssh	SSH	**true** indicates that a SSH private key or username/password is used for adding a host.

Step 6 Click **Import and Check** to upload the completed template file.

- If the check passes, click **Import All** to import hosts in batches.
- If the check fails, view the format error, modify the file, and try again.

Figure 5-5 Import and Check



----End

5.6 (Optional) Configuring a NAS Server

CAUTION

- The following configuration method is for reference only. Configure a NAS server based on the method provided by the NAS server vendor.
- Set the user mapping of the NAS server to **all_squash**.
- Change the user and user group to which the mount directory belongs to **nobody:nobody**.
- Ensure that the minimum permission on the mount directory is 700.
- Ensure that the network between the instances and the NAS server is normal. Ensure that the iptables service is not enabled or the iptables service is configured to open the NFS service port to the network segment of the instances. The default port number is 111.
- The shared directory of the NAS file system must start with a left slash (/). The characters between the left slash (/) must be letters and digits. The directory name cannot exceed 255 characters.

Step 1 Log in to the NAS server as the **root** user.

Step 2 Run the following command to create a shared directory for the NAS file system. The shared directory of the NAS file system must start with a left slash (/). The characters between the left slash (/) must be letters and digits. The directory name cannot exceed 255 characters.

mkdir -p /data/nas

Step 3 Open the NFS configuration file **/etc/exports** and add the following content to configure the NAS server. In the information, **{nasDir}** indicates the shared directory, **{ip}** indicates the IP address of the host that can access the shared directory, and the permission parameters are listed in the brackets.

/*{nasDir}* *{ip}*(insecure,rw,sync,no_subtree_check,all_squash)

The following is a configuration example:

/data/nas *(insecure,rw,sync,no_subtree_check,all_squash)

Step 4 Run the following command to restart NFS to make the configuration take effect:

service nfs-server restart

NOTE

If the system displays a message indicating that the nfs-server service cannot be found, the NFS is not installed on the current server. Run the following command to install NFS. For details about how to configure the Yum repository, see **Configuring a Yum Repository**.

yum install nfs-utils -y

Step 5 Run the following command to change the user and user group to which the mount directory belongs to **nobody:nobody**:

chown nobody:nobody -R /data/nas

Step 6 Run the following command to make the change take effect:

exportfs -rv

Step 7 Run the following command and ensure that the minimum permission on the mount directory is 700:

chmod 700 /data/nas

Step 8 Run the following command to enable automatic startup of the NAS service:

systemctl enable nfs-server

----End

5.7 Installing GaussDB Instances

5.7.1 Installing a DB Instance on the Local Disk

Scenarios

GaussDB Management Platform (TPOPS) allows you to install managed DB instances of the metadata database in the default directory on the local disk. GaussDB instances of 8.0 or later are supported.

Prerequisites

- You need to apply for a trial license capacity when installing a DB instance for the first time. For details, see "Platform Management" > "License Management" in ***GaussDB Management Platform (TPOPS) User Guide (for Lightweight GaussDB)***.
- If the data node originally uses a non-sandbox environment, manually clear the node first. For details, see in "Appendixes" > "Manually Clearing Nodes Before Changing the Installation Mode" in ***GaussDB Management Platform (TPOPS) User Guide (for Lightweight GaussDB)***.
- Before the installation, check whether there are databases installed on the data node of other platforms. For details about how to mount a disk, see "FAQs" > "Installing a DB Instance" > "Pre-Installation Check" in ***GaussDB Management Platform (TPOPS) User Guide (for Lightweight GaussDB)***.
- If you want to install a non-single-node instance with a floating IP address in the primary/standby deployment, prepare the required floating IP address that is consistent with that of the to-be-bound network segment. Ensure that the floating IP address is valid and unique.

Constraints

- The distributed five-node cluster does not support 128 vCPUs and 1024 GB memory.
- GaussDB instances only support seven levels of specifications: 8 vCPUs, 64 GB memory; 16 vCPUs, 128 GB memory; 32 vCPUs, 256 GB memory; 64 vCPUs, 512 GB memory; 96 vCPUs, 768 GB; 128 vCPUs, 1024 GB; and 196 vCPUs,

1569 GB memory (DBMind only). Select a host whose hardware specifications are greater than or equal to 8 vCPUs and 64 GB memory.

Procedure

Step 1 **Log in to GaussDB Management Platform (TPOPS).**

Step 2 Click **Install Instance**. The **Install Instance** page is displayed.

Figure 5-6 Installing an instance

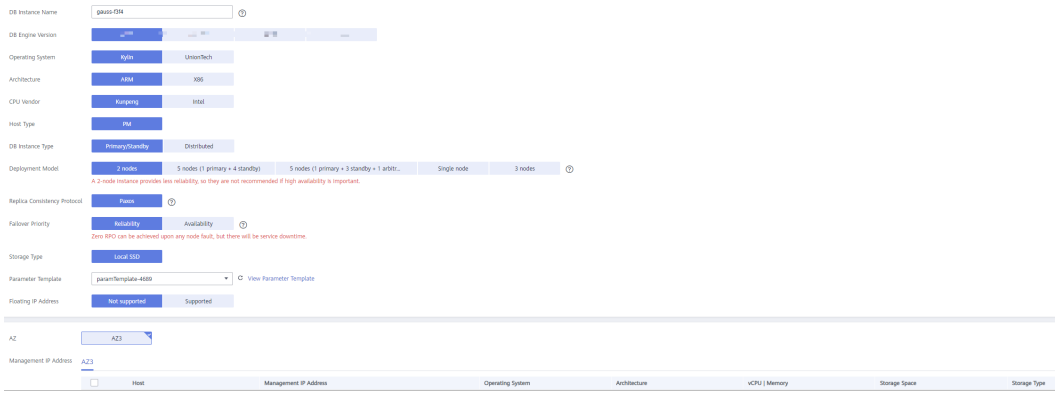


Table 5-6 Parameters for installing an instance

Parameter	Description
Instance	The instance name must start with a letter and contain 4 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
DB Engine Version	GaussDB instances of version 8.0 or later are supported.
Operating System	Displayed based on the registered host information. Currently, Kylin and UnionTech OSs are supported.
Architecture	Displayed based on the registered host information. Currently, x86 and Arm are supported.
CPU Vendor	Displayed based on the registered host information. Kunpeng, Intel, and Hygon are supported.
Host Type	Displayed based on the registered host information. Currently, only **PM** can be selected.
DB Instance Type	• **Distributed**: You can add nodes for distributed instances as needed to handle large volumes of concurrent requests. • **Primary/Standby**: Primary/Standby instances are suitable for scenarios with small and stable volumes of data, where data reliability and service availability are extremely important.

Parameter	Description
Deployment Model	• Distributed (default): – **1 node**: There are 3 shards in total. – **3 nodes**: 3-node deployment where there are three shards and each shard contains one primary DN and two standby DNs. – **4 nodes**: 4-node deployment where there are four shards. Each shard contains one primary DN and one standby DN. – **5 nodes**: 5-node deployment where there are four shards and an arbitration node. Each shard contains one primary DN and three standby DNs. – **9 nodes** (default): 9-node deployment where there are one primary DN and three standby DNs. The deployment contains four shards and an arbitration node. • Primary/Standby: – **1 node**: There is one shard. – **2 nodes (default)**: 2-node deployment where there is one shard and the log node is integrated with a data node. The shard contains one primary DN and one standby DN. Only GaussDB 3.301 and later versions are supported. – **3 nodes**: 3-node deployment where there is one shard. The shard contains one primary DN and two standby DNs. – **5 nodes (1 primary + 3 standby + 1 arbitration)**: 5-node deployment where there is a shard and an arbitration node. The shard contains one primary DN and three standby DNs. – **5 nodes (1 primary + 4 standby)**: 5-node deployment where there is a shard. The shard contains one primary DN and four standby DNs. To use the single-node deployment, enable the feature whitelist **gaussdb_feature_supportSingleMode**. For details, see **How Do I Enable or Disable the Whitelist?**. From 8.102 or later, the mode without ETCD for primary/standby instances with 2 nodes, 3 nodes, 5 nodes (1 primary + 3 standby + 1 arbitration), and 5 nodes (1 primary + 4 standby) are supported. To use mode without ETCD, you need to enable the whitelist feature **gaussdb_feature_supportDccCluster**. For details, see **How Do I Enable or Disable the Whitelist?**. Move the cursor to ⓘ to check database architectures in different deployment models.

Parameter	Description
Replica Consistency Protocol	This parameter is mandatory when **DB Instance Type** is set to **Primary/Standby**. Only GaussDB 3.200 and later versions are supported. • **Quorum**: primary/standby synchronous replication in quorum mode. After a client initiates a transaction, the primary node responds to the client only after the corresponding WAL logs are replicated to multiple replicas. The breakdown of a few nodes does not affect global availability. Quorum can ensure data consistency. • **Paxos**: It resolves log fork during log replication in quorum mode. Paxos improves the log replication throughput and enhances the DN self-arbitration capability. The two-node primary/standby deployment supports only Paxos. After the mode without etcd is enabled, two nodes, three nodes, five nodes (1 primary + 3 standby + 1 arbitration), and five nodes (1 primary + 4 standby) of the primary/standby version support only Paxos.
Transaction Consistency	This parameter is mandatory when **DB Instance Type** is set to **Distributed**. • **Strong consistency**: When an application updates data, you can query all data that has been successfully submitted, but performance is affected. • **Eventual consistency**: When an application updates data, the data you queried may not be the most current value. The most current data may take a bit of time to become available for query. However, DB instances with eventual consistency generally have higher performance. Eventual consistency cannot ensure strong read consistency of distributed transactions and consistency of transactions that depend on query results, such as INSERT INTO SELECT * FROM. Write operations that are split into multiple statements or involve in multiple nodes are not supported. DR relationships cannot be created.

Parameter	Description
Failover Priority	This parameter is displayed only when **DB Instance Type** is set to **Primary/Standby** and two-node deployment is selected. By default, **Reliability** is selected. You can modify the failover policy as required after installing a DB instance. • **Reliability**: Data consistency is given priority during a failover. This is recommended for applications with highest priority for data consistency. • **Availability**: Database availability is given priority during a failover. This is recommended for applications that require their databases to provide uninterrupted online services. **NOTE** If **Availability** is selected, exercise caution when changing the following database parameters: – **recovery_time_target**: If this parameter is incorrectly changed, the instance will undergo frequent forced failovers. To change this parameter, follow the instructions in **Contacting Technical Support**. – **audit_system_object**: If this parameter is incorrectly changed, DDL audit logs will be lost. To change this parameter, follow the instructions in **Contacting Technical Support**.
Storage Type	Displayed based on the registered host information. Currently, local SSD disks and flash storage are supported. Select **Local SSD**.
Floating IP Address	Whether the floating IP address is supported for the instance to be installed. This parameter is displayed only when **DB Instance Type** is set to **Primary/Standby** and single-node deployment is not selected. • **Not supported**: The floating IP address is not supported. • **Supported**: Enter a floating IP address. After the instance is installed, you can use the floating IP address to access DN data. For example, during the primary/standby switchover, the same floating IP address is configured for the three nodes in primary/standby deployment and is used to connect to the data nodes. After the switchover is complete, the floating IP address can still be used to connect to the data nodes. **NOTE** • To use the floating IP address function, select the kernel installation package of 503.1.0.SPC1200, 503.1.0.SPC1300, or 503.2.0 or later. • Ensure that the configured floating IP address is valid and unique. If the IP address is invalid, an instance cannot be accessed using the floating IP address after being installed. The floating IP address must be in the same network segment as **virtualIp**. • After an instance is installed, the floating IP address cannot be changed.

Parameter	Description
Parameter Template	A template of parameters for creating an instance. The template contains engine configuration values that are applied to one or more instances. You can modify the instance parameters as required after the DB instance is created. After a DB instance is created, you can change the parameter template based on service requirements.
AZ	• Distributed: – In a single-node deployment, instances can be deployed in only one AZ. – In a 3-node deployment, instances can be deployed in one or three AZs. – In a 4-node deployment, instances can be deployed in only one AZ. – In a 5-node or 9-node deployment, instances can be deployed in three AZs. • Primary/Standby: – In a single-node deployment, instances can be deployed in only one AZ. – In a two-node deployment, instances can be only deployed in two AZs. – In a 3-node deployment, instances can be deployed in one or three AZs. – In a 5-node (1 primary + 3 standby + 1 arbitration) deployment , instances can be deployed in three AZs. – In a 5-node (1 primary + 4 standby) deployment, instances can be deployed in one or three AZs. • For 5-node (1 primary + 3 standby + 1 arbitration) and 9-node instances, you can select the primary, standby, and arbitration AZs, which must be different from each other. For 3-node primary/standby instances, you can select only the primary AZ. • An AZ is a physical region where resources have independent power supplies and networks. AZs are physically isolated but interconnected through an internal network.
Management Address	Hosts that have been added. Only hosts that have not been managed are displayed.
Database Port	The port is used by applications to access the database. Value range: 1024 to 39989. Default value: **8000**. The following ports are not allowed: 2378-2380, 2400, 4999-5001, 5100, 5500, 5999-6001, 6009-6010, 6500, 8015, 8097, 8098, 8181, 9090, 9100, 9180, 9187, 9200, 12016, 12017, 20049, 20050, 21731, 21732, 32122-32126 and 39001.

Step 3 Enter the NAS configuration information.

Figure 5-7 NAS device information

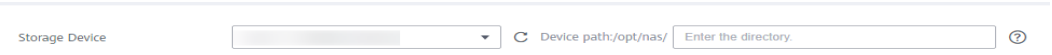


Table 5-7 Parameters for configuring NAS device information

Parameter	Description
Storage Device	You can select the target storage device for mounting. The user-defined path is the storage path of the backup file. If there is no NAS device on this platform, configure a storage device in storage device management.

Step 4 Enter the database configuration information.

Figure 5-8 DB information

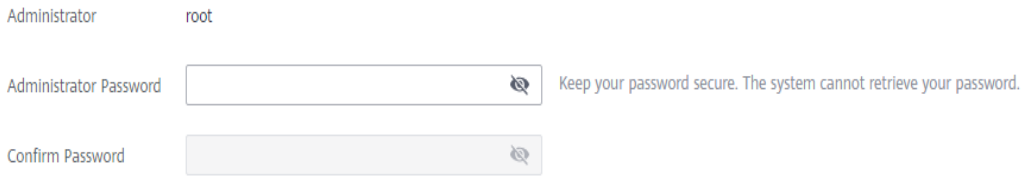
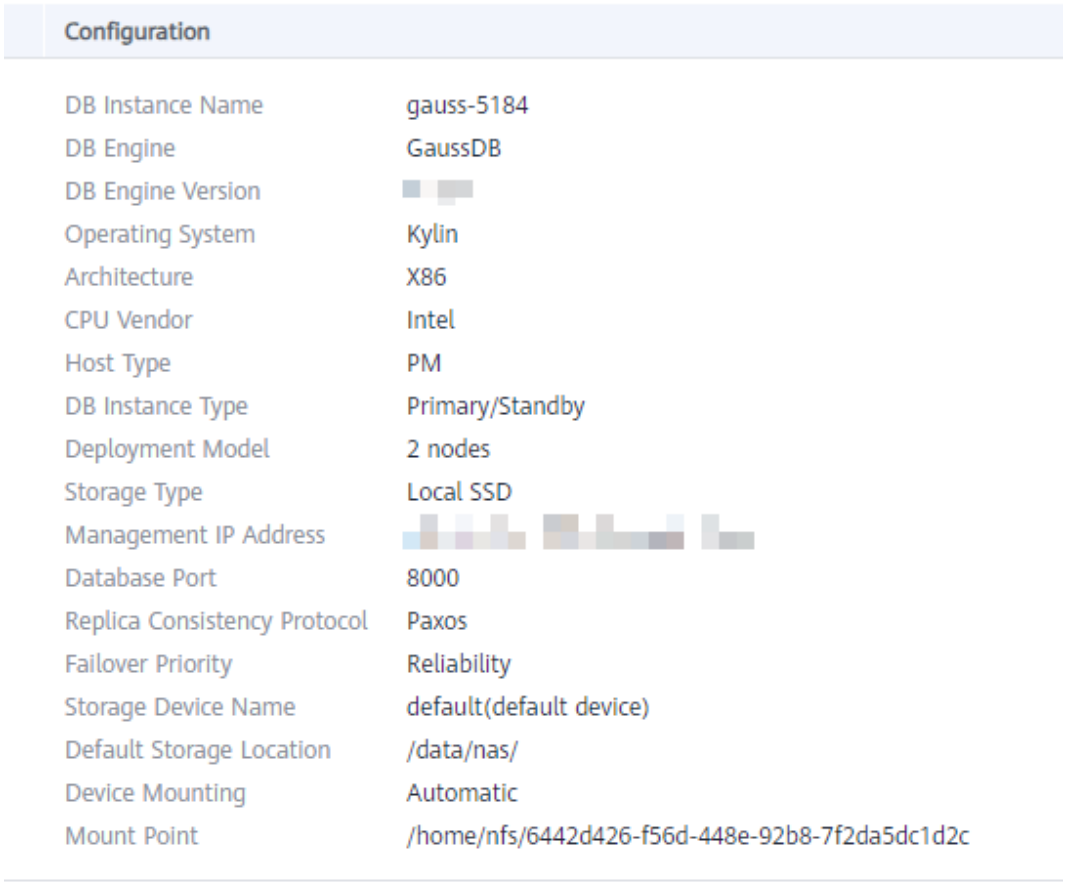


Table 5-8 Database parameters

Parameter	Description
Administrator	The default login name for the database is **root**.
Administrator Password	Enter a strong password and periodically change it for security reasons. A new password must meet the following complexity requirements: • The password consists of 8 to 32 characters. • The password must contain at least three types of the following characters: – At least one uppercase character – At least one lowercase character – At least one digit (0 to 9) – At least one of the following special characters: ~!@#%^*-_=+?, Keep your password secure. The system cannot retrieve it if it is lost.
Confirm Password	This password must be consistent with administrator password.

Step 5 Click **Apply Now**. On the displayed page, confirm the instance information.

Figure 5-9 DB instance information



- To modify the settings, click **Previous**.
- If you do not need to modify your settings, click **Submit**. The installation takes about 10 minutes.

Step 6 Go to the **Instances** page to view and manage the instance.

- You can query instances by product type, instance name, or keyword.
- You can click next to an instance name to change the name of a created instance.
- You can click under the instance name to copy the created instance ID.

----End

NOTE

- During the creation process, the GaussDB instance status is **Creating**.
- To refresh the instance list, click the refresh icon in the upper right corner of the list. When the creation process is complete, the instance status will be **Available**.
- The default database port is 8000. You can change it after a DB instance is created.
- If a NAS server has been configured, automated backup is automatically enabled after the instance is created, and a full backup is automatically generated to record the initial status of the instance.

5.7.2 Installing DB Instances in the Dorado Storage Pool

Scenarios

This section describes how to use the GaussDB management platform (TPOPS) to install GaussDB instances in the Dorado storage pool.

Only primary/standby (1 primary + 2 standby) GaussDB instances using Dorado storage can be created, and the instances can only be used in single-cluster scenarios.

This function is a whitelist function. To use this function, enable it in the whitelist. For details, see **How Do I Enable or Disable the Whitelist?**.

Prerequisites

- You need to apply for a trial license capacity when installing a DB instance for the first time.
- In the current version, primary/standby (1 primary + 2 standby) GaussDB instances using Dorado storage can be created in the sandbox environment, and can be used in the single-cluster scenarios.
- All nodes of the instance have completed the Dorado LUN partitioning. This operation is performed by the storage side.
- All nodes of the instance have mounted disks, including system disks, data disks, log disks, local backup set disks, and ectd disks. The disks need to be initialized and mounted. For details about how to mount a disk, see "FAQs" > "Installing a DB Instance" > "Requirements for Mounting Data Disks" in ***GaussDB Management Platform (TPOPS) User Guide (for Lightweight GaussDB)***.
- All nodes of the instance must be added to the host management page, and the storage type must be set to Dorado.
- Before using this function, enable the feature whitelist **gaussdbv5_feature_supportDorado**. For details, see **How Do I Enable or Disable the Whitelist?**.
- Before the installation, check whether there are databases installed on the data node of other platforms. For details about how to mount a disk, see "FAQs" > "Installing a DB Instance" > "Pre-Installation Check" in ***GaussDB Management Platform (TPOPS) User Guide (for Lightweight GaussDB)***.

Procedure

Step 1 Log in to GaussDB Management Platform (TPOPS).

Step 2 Choose **Instances** > **Install Instance**.

Step 3 For details about the parameters for installing Dorado instances, see **Table 5-9**. Other parameters are the same as those in **Installing a DB Instance on the Local Disk**.

Figure 5-10 Installing an instance

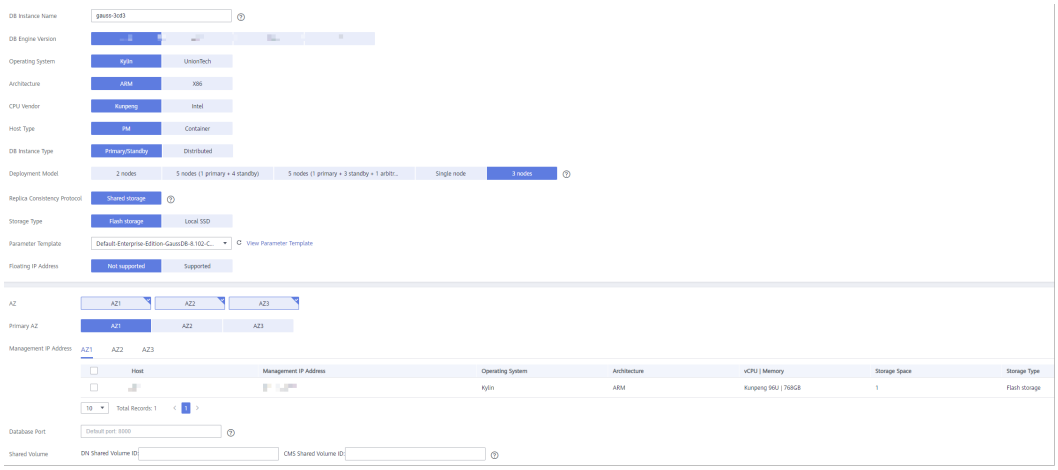


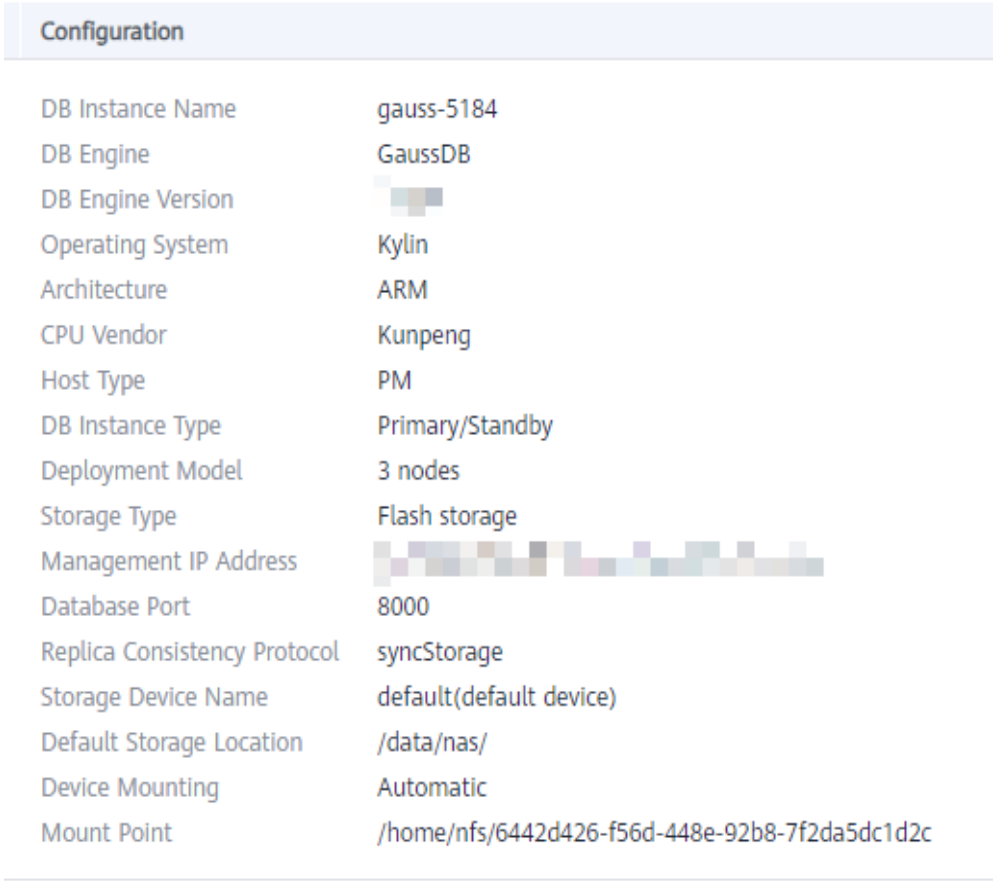
Table 5-9 Parameters for installing an instance

Parameter	Parameter Value
DB Instance Type	**Primary/Standby**
Deployment Model	**3 nodes**
Storage Type	Displayed based on the registered host. Select **Flash Storage**.
Replica Consistency Protocol	**Shared storage**: Shared storage supports flexible deployment based on decoupled compute and storage. You can plan database capacity as required and scale out storage when necessary. Log replication is uninstalled using Dorado all-flash storage, which can be used to achieve an RPO of zero in an intra-city dual-cluster DR solution.
Management IP Address	Select the host whose storage type is flash storage.

Parameter	Parameter Value
Shared Volume	The DN shared volume ID and CMS shared volume ID are provided by the storage based on the networking mode. • If NOF is used, the shared volume ID is **NGUID**. • If FC is used, the shared volume ID is **WWN**. Obtain the **DN shared volume ID** and **CMS shared volume ID** from technical support. **NOTE** • The DN shared volume is usually named **xlog**. • The CMS shared volume is usually named **om**.

Step 4 Click **Apply Now**. On the displayed page, confirm the instance information.

Figure 5-11 DB instance information



Step 5 Click **Submit**.

----End

5.8 Creating a DBMind Instance

Scenarios

GaussDB Management Platform (TPOPS) is used to create a DBMind instance.

Constraints

Currently, only Kylin V10 Arm-based DBMind instances can be created.

Procedure

Step 1 Log in to GaussDB Management Platform (TPOPS).

Step 2 Click **DBMind Management**.

Step 3 Click **Create DBMind Instance** and configure required parameters.

Figure 5-12 Creating a DBMind instance

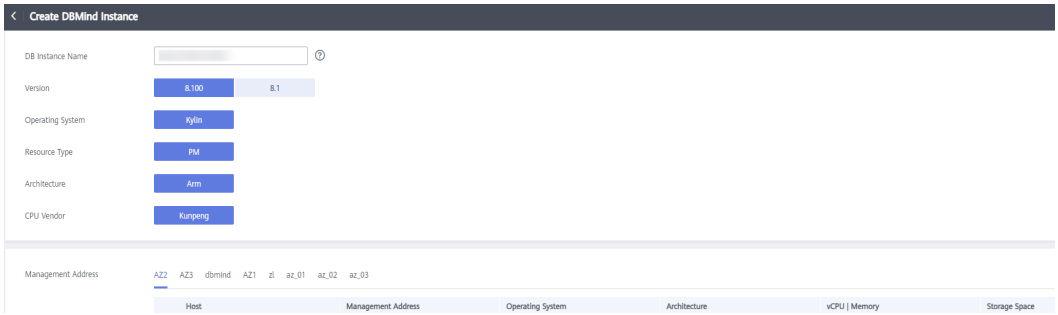


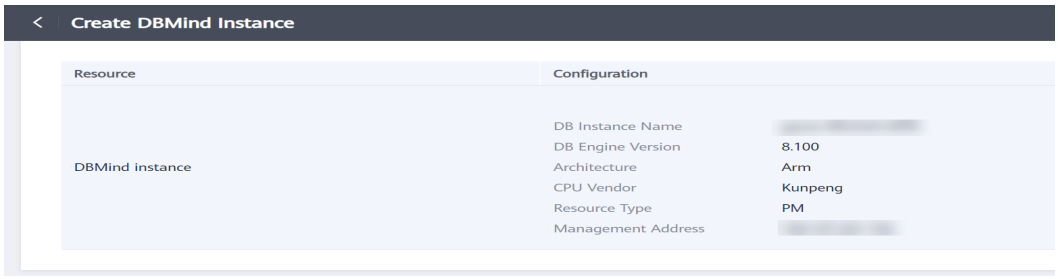
Table 5-10 Basic parameters

Parameter	Description
DB Instance Name	The instance name must start with a letter and contain 4 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
Version	DBMind 8.0 or later is supported.
Operating System	Displayed based on the operating system of the DBMind version. Currently, only **Kylin** can be selected.
Resource type	Displayed based on the registered host information. Currently, only **PM** can be selected.
Architecture	Displayed based on the architecture type of the DBMind version.
CPU Vendor	All CPU vendors of available hosts in the current region.

Parameter	Description
Management Address	Hosts that have been registered. Only available hosts are displayed.

Step 4 Click **Apply Now**. On the displayed page, confirm the instance information.

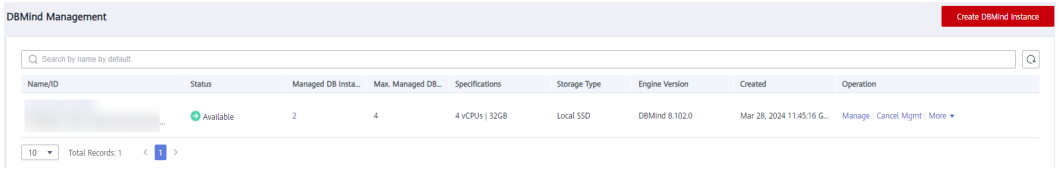
Figure 5-13 DBMind instance information



- If you need to modify your settings, click **Previous**.
- If you have confirmed the settings, click **Submit**.

Step 5 View and manage the DBMind instance on the **DBMind Management** page.

Figure 5-14 DBMind instance created successfully



- You can query the information by name, ID or keyword.
- You can click  under the instance name to copy the created instance ID.
- Click the number of instances managed by a DBMind instance. **Name/ID**, **DB Instance Type**, and **Status** that have been managed by the DBMind instance are displayed.

  Click **Name/ID** of a managed instance. The basic information about the managed instance is displayed.

Figure 5-15 Managed instances

----End

6 (Optional) Installing DRS

If you want to use DRS to migrate data online and synchronize databases in real time, download and install DRS of the required version from **here**.

7 Uninstalling GaussDB Management Platform (TPOPS)

Prerequisites

If GaussDB Management Platform (TPOPS) needs to be reinstalled after being uninstalled, bring all hosts offline first.

Constraints

If you need to uninstall GaussDB Management Platform (TPOPS) after the docker-service directory is deleted, see **Performing Uninstallation After the docker-service Directory Is Deleted**.

Procedure

Step 1 Log in to any GaussDB Management Platform (TPOPS) node as the **root** user. For details about how to configure password-free uninstallation, see **Setting Mutual Trust Between Nodes on the Management Plane**.

NOTE

- The following procedure uses the **/data/docker-service** directory as an example. Replace it with the actual directory where docker-service is located.
- The directory structure and configuration information of the execution node must be the same as those of the remote nodes (the nodes except the execution node). Otherwise, the uninstallation may fail.

Step 2 Run the following command to open the parameter configuration file:

vi /data/docker-service/config/user_edit_file.conf

Step 3 Change the value of **uninstall_all** to **yes**.

"uninstall_all":"yes"

Step 4 Press **Esc** and enter **:wq** to save the file and exit.

Step 5 Run the following commands to uninstall GaussDB Management Platform (TPOPS):

cd /data/docker-service

sh appctl.sh uninstall_all

If the following information is displayed, it is uninstalled:

```
[root@dbsnoname1 docker-service]# sh appctl.sh uninstall_all
start check node authentication
node not support auto authentication, will input root password
Enter the password of the root user.
start check host: 192.168.0.1 root password
check host: 192.168.0.1 root password success
start check host: 192.168.0.2 root password
check host: 192.168.0.2 root password success
start check host: 192.168.0.3 root password
check host: 192.168.0.3 root password success
Start to init manifest...
init manifest successful for 192.168.0.1.
init manifest successful for 192.168.0.2.
init manifest successful for 192.168.0.3.
======== 192.168.0.1: gaussdb_service ========
 auth                 | complete
 gaussdb-console      | complete
 luban                | complete
 ots                  | complete
 GaussDB-open-api     | complete
 GaussDB-instancemanager | complete
 GaussDB-backupmanager | complete
======== 192.168.0.2: gaussdb_service ========
 auth                 | complete
 gaussdb-console      | complete
 luban                | complete
 ots                  | complete
 GaussDB-open-api     | complete
 GaussDB-instancemanager | complete
 GaussDB-backupmanager | complete
======== 192.168.0.3: gaussdb_service ========
 auth                 | complete
 gaussdb-console      | complete
 luban                | complete
 ots                  | complete
 GaussDB-open-api     | complete
 GaussDB-instancemanager | complete
 GaussDB-backupmanager | complete
======== 192.168.0.1: docker_service ========
 common-service       | complete
 monitor-service      | complete
 rds-ha-admin         | complete
 resource-manager     | complete
 workflow             | complete
======== 192.168.0.2: docker_service ========
 common-service       | complete
 monitor-service      | complete
 rds-ha-admin         | complete
 resource-manager     | complete
 workflow             | complete
======== 192.168.0.3: docker_service ========
 common-service       | complete
 monitor-service      | complete
 rds-ha-admin         | complete
 resource-manager     | complete
 workflow             | complete
======== 192.168.0.1: Kafka ========
 kafka                | complete
======== 192.168.0.2: Kafka ========
 kafka                | complete
======== 192.168.0.3: Kafka ========
 kafka                | complete
======== 192.168.0.1: Zookeeper ========
 zookeeper            | complete
 GaussDB-feature-data | complete
 GaussDB-data         | complete
```



```
======== 192.168.0.2: Zookeeper ========
 zookeeper             | complete
 GaussDB-feature-data  | complete
 GaussDB-data          | complete
======== 192.168.0.3: Zookeeper ========
 zookeeper             | complete
 GaussDB-feature-data  | complete
 GaussDB-data          | complete
======== 192.168.0.1: PlatformData ========
 platform-data         | complete
======== 192.168.0.2: PlatformData ========
 platform-data         | complete
======== 192.168.0.3: PlatformData ========
 platform-data         | complete
======== 192.168.0.1: CommonbaseData ========
 common-base           | complete
======== 192.168.0.2: CommonbaseData ========
 common-base           | complete
======== 192.168.0.3: CommonbaseData ========
 common-base           | complete
======== 192.168.0.1: base_enviornment ========
 docker                | complete
 InfluxDB              | complete
 sftp                  | complete
 gaussdb               | complete
======== 192.168.0.2: base_enviornment ========
 docker                | complete
 InfluxDB              | complete
 sftp                  | complete
 gaussdb               | complete
======== 192.168.0.3: base_enviornment ========
 docker                | complete
 InfluxDB              | complete
 sftp                  | complete
 gaussdb               | complete
======== 192.168.0.1: patch ========
 patch                 | complete
 base_env              | complete
======== 192.168.0.2: patch ========
 patch                 | complete
 base_env              | complete
======== 192.168.0.3: patch ========
 patch                 | complete
 base_env              | complete

Uninstallation progress [72/72] ==> 100.00%
```

Step 6 Run the following command to clear the residual users, directories, and SFTP data:

NOTICE

The deletion command will delete all data from the SFTP server. If the SFTP data needs to be used, back up the data in advance.

Backup method: Copy all content in the **/opt/sftphome/** directory. For example: **cp -r /opt/sftphome /data/sftphome_bak**

sh appctl.sh cleanup_all

If the following information is displayed, they are deleted:

```
[root@dbsnoname1 docker-service]# sh appctl.sh cleanup_all
start check node authentication
node not support auto authentication, will input root password
Enter the password of the root user.
start check host: 192.168.0.1 root password
```



```
check host: 192.168.0.1 root password success
start check host: 192.168.0.2 root password
check host: 192.168.0.2 root password success
start check host: 192.168.0.3 root password
check host: 192.168.0.3 root password success
Start to init manifest...
init manifest successful for 192.168.0.1.
init manifest successful for 192.168.0.2.
init manifest successful for 192.168.0.3.
Cleanup 192.168.0.1 Success!
Cleanup 192.168.0.2 Success!
Cleanup 192.168.0.3 Success!
```

Step 7 Log in to all GaussDB Management Platform (TPOPS) nodes as the **root** user and run the following command to delete the docker-service directory:

rm -rf /data/docker-service

----End

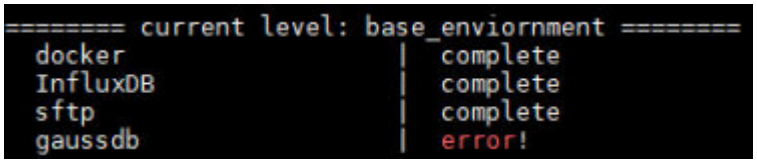
8 FAQs

8.1 What Should I Do If an Error Is Reported During the Installation of GaussDB Used by GaussDB Management Platform (TPOPS)?

8.1.1 Reinstalling the GaussDB Management Platform (TPOPS) Metadata Database

Symptom

The following error information is displayed during the installation.



Possible Causes

An error is reported during GaussDB installation.

Solution

Step 1 View the GaussDB installation logs in **/tmp/install_cluster.log** to locate the error information.

Step 2 Perform the operations in **Contacting Technical Support** to solve the GaussDB installation issue.

Step 3 Log in to the GaussDB Management Platform (TPOPS) node as the **root** user.

Step 4 Run the following command to open the configuration file:

vi /data/docker-service/config/user_edit_file.conf

Step 5 Change the value of **uninstall_all** to **yes** to enable the uninstallation function.

Step 6 Press **Esc** and run the **:wq!** command to save the change and exit.

Step 7 Run the following commands to uninstall GaussDB Management Platform (TPOPS):

cd /data/docker-service

sh appctl.sh uninstall_all

In the preceding commands, **/data/docker-service** indicates the installation directory.

Step 8 Run the following command to clear the residual installation data:

sh appctl.sh cleanup_all

Step 9 Reinstall GaussDB Management Platform (TPOPS). For details, see **Installing the GaussDB Management Platform (TPOPS)**.

----End

8.1.2 Handling the OMAgent Installation Failure

Error Message

The error log file is stored in **/tmp/install_cluster.log**.

Error information: check_omagent failed in loaclhost: ...

Cause

OMAgent fails to be installed.

Procedure

Step 1 Log in to the GaussDB Management Platform (TPOPS) node as the **root** user.

Step 2 Run the following command to obtain the Python 3 installation path:

which python3

Information similar to the following is displayed:

```
/usr/bin/python3
```

Step 3 Run the following command to check the dependencies of Python3:

ldd /usr/bin/python3

If the following information is displayed, Python3 on all nodes where GaussDB Management Platform (TPOPS) is installed must contain the dependency in bold.

```
linux-vdso.so.1 (0x00007ffc7bb2a000)
libpython3.7m.so.1.0 => /lib64/libpython3.7m.so.1.0 (0x00007fb1ad8a4000)
libcrypt.so.1 => /lib64/libcrypt.so.1 (0x00007fb1ad869000)
libpthread.so.0 => /lib64/libpthread.so.0 (0x00007fb1ad848000)
libdl.so.2 => /lib64/libdl.so.2 (0x00007fb1ad843000)
libutil.so.1 => /lib64/libutil.so.1 (0x00007fb1ad83e000)
libm.so.6 => /lib64/libm.so.6 (0x00007fb1ad6b9000)
```



```
libc.so.6 => /lib64/libc.so.6 (0x00007fb1ad4f8000)
/lib64/ld-linux-x86-64.so.2 (0x00007fb1adc40000)
```

- If there are the preceding dependencies, perform the subsequent operations.
- If the preceding dependencies are not found, use the yum source to perform the operations in **Installing Python 3** again.

Step 4 Run the following command to check the disk space:

df -h

In the following command output, the information in bold indicates the percentage of the available capacity of the **root** directory:

```
Filesystem            Size  Used Avail Use% Mounted on
devtmpfs              128G     0  128G   0% /dev
tmpfs                 128G  192K  128G   1% /dev/shm
tmpfs                 128G  4.2G  123G   4% /run
tmpfs                 128G     0  128G   0% /sys/fs/cgroup
/dev/mapper/klas-root  392G  114G  278G  30% /
tmpfs                 128G   64K  128G   1% /tmp
/dev/sda2             1014M  222M  793M  22% /boot
```

- If the disk space usage does not exceed 90%, perform **Contacting Technical Support**.
- If the disk space usage exceeds 90%, clear the root directory space.

Step 5 Run the following command to check whether there are Python packages whose permission is not 755:

ls -l /usr/local/lib/python3.7/site-packages

- If yes, run the following command to change the permission to 755:

  chmod -R 755 /usr/local/lib/python3.7/site-packages
- If no, perform **Contacting Technical Support**.

----End

8.2 How Do I Enable or Disable the Whitelist?

Scenarios

This section describes how to enable and disable the whitelist.

Procedure

Step 1 Log in to the primary node of GaussDB Management Platform (TPOPS) as user **root**.

Step 2 Run the following command to connect to the core database:

gsql -p 8635 -U core -W {*password*} -q core -h 127.0.0.1

{password} indicates the password for connecting to the core database. For details about the password, see ***GaussDB Management Platform (TPOPS) Account List 01***.

Step 3 Run the following command to enable and disable the whitelist:

UPDATE CORE.DBS_FEATURE SET STATUS = '{open/closed}'

WHERE ID = (SELECT ID FROM CORE.DBS_FEATURE WHERE NAME = '{*featureName*}' AND SITE = 'pcs-lite');

In the preceding commands:

- *{open/closed}*: **open** indicates that the whitelist is enabled. **closed** indicates that the whitelist is disabled.
- *{featureName}* indicates the name of the whitelist.

Step 4 Run the following command to check whether the whitelist is enabled or disabled:

SELECT STATUS FROM CORE.DBS_FEATURE WHERE NAME = '{*featureName*}' AND SITE = 'pcs-lite';

In the command output, **open** indicates that the function is enabled, and **closed** indicates that the function is disabled.

Step 5 Run the **\q** command to exit the core database.

----End

8.3 How Do I Disable a Firewall?

Step 1 Log in to each GaussDB Management Platform (TPOPS) node that has been installed as the **root** user.

Step 2 Run the following commands on each node to disable the firewall and disable the firewall from starting upon system startup:

systemctl stop firewalld.service

systemctl disable firewalld.service

Step 3 Use the vi editor to open the **config** file:

vi /etc/selinux/config

Step 4 Change the value of **SELINUX** to **permissive**, press **Esc**, and run the **:wq** command to save the change and exit.

SELINUX=permissive

NOTE

Generally, enabling SELinux improves system security but may cause program running failures. To ensure successful installation, you are advised to set this parameter to **permissive**.

Step 5 Run the following command to restart the OS:

reboot

----End

8.4 How Do I Distribute Installation Packages Again?

Scenarios

This topic describes how to modify the configuration or replace the package and reinstall GaussDB Management Platform (TPOPS) if GaussDB Management Platform (TPOPS) fails to be installed.

Procedure

Step 1 Log in to the GaussDB Management Platform (TPOPS) node as the **root** user.

Step 2 Run the following command to delete the identification file for distributing the installation packages:

rm -rf /data/docker-service/config/FIRST_DISTRIBUTE_ON

Step 3 Run the following command to switch to the installation script directory:

cd /data/docker-service

Step 4 Run the following command to deliver the installation instruction:

sh appctl.sh install

----End

8.5 How Do I Handle the SFTP Installation Failure?

Step 1 Locate the node where SFTP fails to be installed based on the command output.

Step 2 Log in to the node where SFTP fails to be installed as the **root** user.

Step 3 Run the following commands to go to the log directory and view the error log:

cd /opt/cloud/logs/deploy

vi install.error.log

Search for SFTP-related content in the logs. If the error message "Failed to set password for user" is displayed, the PAM rule in the installation environment does not support preset passwords. In this case, perform the following steps to modify the PAM rule.

NOTE

The **/opt/cloud/logs** directory is the value of **log_path** in the **/data/docker-service/config/user_edit_file.conf** file. Set it based on the site requirements.

Step 4 Run the following command to modify the system-auth file:

vi /etc/pam.d/system-auth

Find **pam_deepin_pw_check.so** and it to **pam_pwquality.so**.

Step 5 Press **Esc** and run the following command to save the modification and exit:

:wq!

Step 6 Run the following command to go to the installation directory

cd /data/docker-service

Step 7 Run the following command to deliver the installation script again:

sh appctl.sh install

----End

NOTE

If the error persists, **contact technical support for assistance**.

8.6 How Do I Manage Hosts?

8.6.1 Handling Host Adding Failure

Failed to Add a Host and a Message Is Displayed Indicating that the Host Network Connection Fails

Step 1 Check whether the host management IP address is correct.

- If the host management IP address is correct, go to the next step.
- If the host management IP address is incorrect, perform the following operations to delete the host and then add the host again:
  - Log in to the primary node of GaussDB Management Platform (TPOPS) as the **root** user.
  - Run the following command to connect to the core database:

    gsql -p 8635 -U core -W {*password*} -q core -h 127.0.0.1

    {password} indicates the password for connecting to the core database.
  - Run the following commands to delete the host:

    UPDATE CORE.DBS_HOST_STATIC_INFO SET STATUS='deleted' WHERE HOST_ID={*hostId*};

    UPDATE CORE.DBS_HOST_SPECIFICATION_INFO SET STATUS='deleted' WHERE ID={*hostId*};

    {hostId} indicates the host ID.
  - Run the following commands to check whether the host is deleted:

    SELECT STATUS FROM CORE.DBS_HOST_STATIC_INFO WHERE HOST_ID={*hostId*};

    SELECT STATUS FROM CORE.DBS_HOST_SPECIFICATION_INFO WHERE ID ={*hostId*};

    If **deleted** is returned in the command output, the host is deleted.
  - Run the **\q** command to exit the core database.

Step 2 Check whether the host is faulty.

- If the host is normal, check other steps.

- If the host is faulty, run the commands in **Step 1** to delete the host. After the host is restored, add the host again.

Step 3 Check whether the network configuration and network connection of the host are correct.

Step 4 Check whether the ping request is allowed to pass through the host firewall.

Step 5 Check whether the DNS settings and NIC settings of the host are correct.

----End

Failed to Add a Host and the Message "Failed to Connect to the Host Using SSH. Check the Password or sshd Configuration." Is Displayed

Step 1 Check whether the password of the **root** user and the SSH port number of the host are correct.

- If the host is deleted, the password of the **root** user or the SSH port number is incorrect. Add the host again.
- If the host fails to be deleted, see the subsequent steps.

Step 2 Check whether the host network route is correctly configured on GaussDB Management Platform (TPOPS).

Step 3 Check whether the SSH service is started on the host.

Step 4 Check whether the sshd configuration on the host is correct. Common sshd configurations are as follows:

PermitRootLogin yes

PasswordAuthentication yes

GSSAPIAuthentication no

----End

Failed to Add a Host and a Message Is Displayed Indicating that the Host Has Been Brought Online on Another Platform

Step 1 **Log in to GaussDB Management Platform (TPOPS).**

Step 2 Choose **Platform Management > Data Centers**.

Step 3 Check whether the host has been added to another GaussDB Management Platform (TPOPS).

If the host has been added to another GaussDB Management Platform (TPOPS), delete the host from another GaussDB Management Platform (TPOPS) and then add the host.

----End

Failed to Add a Host and a Message Is Displayed Indicating that the OS Patch Installation Package Fails to Be Downloaded

Step 1 Log in to GaussDB Management Platform (TPOPS).

Step 2 Choose **Platform Management** > **Installation Packages**.

Step 3 Check whether the OS patch installation package is uploaded. On the **Task Center** page, view the context details of NebulaInitAgentJob to determine the name of the OS patch installation package to be uploaded.

- If the OS patch installation package is uploaded, perform the subsequent steps.
- If no OS patch installation package is uploaded, upload one or upload it to the **/dbs/osPatch** directory on the host.

Step 4 Check whether the SFTP server is normal.

Step 5 Check whether the username and password for logging in to the SFTP server are correct.

----End

Failed to Add a Host and a Message Is Displayed Indicating that the Agent Fails to Be Installed on the Host

Check the **build.log** file in the **/dbs/osPatch/os_patch_all/** directory on the host and modify the file based on the log information.

Failed to Add a Host and the Message Indicating NIC Mapping Failure Is Displayed

Check whether the NIC configuration of the host is correct.

Failed to Add a Host and a Message Is Displayed Indicating that the Host Fails to Pass the Standardization Check

Check the host standardization check report. If a host fails the standardization check, reconfigure the host, and retry the check step in the workflow.

- If the EXPECT or SFTP check item fails, reconfigure the host and retry NebulaHostDetectionPrepareTask.
- If other check items fail, reconfigure the host and retry NebulaHostDetectionTask.

Failed to Add a Host and the Message About Checking Whether azName of the Installed Instance Is the Same As the Equipment Room Name Is Displayed

Step 1 Check the task failure cause and determine the name of the AZ where the instance is installed.

Step 2 Check whether there is an equipment room with the same name as the AZ. If yes, go to **Step 3**.

If no, create an equipment room with the same name as the AZ.

Step 3 Change the equipment room where the host is located by modifying the host static information.

Step 4 Retry the workflow.

----End

8.6.2 Failure to Delete a Host

Failed to Delete a Host and the Message "Failed to Connect to the Host Using SSH. Check the Password or SSHD Configuration." Is Displayed

Step 1 Delete the host again, and check whether the password of the **root** user and the SSH port number of the host are correct.

- If the host is deleted, the password of the **root** user or the SSH port number is incorrect. Add the host again.
- If the host fails to be deleted, see the subsequent steps.

Step 2 Check whether the host network route is correctly configured on the GaussDB Management Platform (TPOPS).

Step 3 Check whether the SSH service is started on the host.

Step 4 Check whether the SSHD configuration on the host is correct. Common SSHD configurations are as follows:

PermitRootLogin yes

PasswordAuthentication yes

GSSAPIAuthentication no

----End

8.6.3 Adding Data Disks on a Host

Scenarios

If data disks are added after a host is added, perform the operations in this section to update the data disk size of the host.

For details to add a local SSD disk, see **Scaling Out Local SSD Disks**.

Constraints

- The host is initiated, to be managed, or in use.
- The I/O type of the new data disks must be the same as that of the original data disks on the host.
- If there are sandbox instances on the host, add the data disks to the **/var/chroot/var/lib/engine/data*** directory. If there are non-sandbox instances on the host, add the data disks to the current data directory.

  Run the following command to check the type of instances on the host:

  cat /dbs/om-agent/agent_*/common/public_cloud.conf | grep dataDir

If the value of **dataDir** in the command output starts with **/var/chroot**, the sandbox type is used. Otherwise, the sandbox type is not used.

- You need to add data disks to all nodes in an instance at the same time, and the added capacity of data disks to each host must be the same. Otherwise, the instance and related functions may be affected.

Procedure

Step 1 **Log in to GaussDB Management Platform (TPOPS).**

Step 2 Query the host list, view the host whose data disk size needs to be updated, make a note of the host ID {*hostId*}, and make a note of the host status and storage type. If the host is In use, record the node ID as {*nodeId*}.

Step 3 Log in to the primary node of GaussDB Management Platform (TPOPS) as the **root** user.

Step 4 Run the following command to connect to the core database:

gsql -p 8635 -U core -W {*password*} -q core -h 127.0.0.1

{password} indicates the password for connecting to the core database.

Step 5 Run the following command to update the data disk size of the host:

UPDATE CORE.DBS_HOST_STATIC_INFO SET DATA_DISK={*dataDisk*} WHERE HOST_ID={*hostId*};

{dataDisk} indicates the data disk size of the host, in GB. In the following example, the data disk **vdc** is added. The data disk size is 300 GB before scale-out and 400 GB after scale-out.

```
[root@host-192-168-1-106 ~]# lsblk
NAME                  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
vda                   252:0    0   40G  0 disk
├─vda1                252:1    0    1G  0 part /boot
└─vda2                252:2    0   39G  0 part
  ├─klas-root         253:0    0   35G  0 lvm  /
  └─klas-swap         253:1    0    4G  0 lvm  [SWAP]
vdb                   252:16   0  300G  0 disk  300 GB before capacity expansion
├─gaussdbvg-mydata1   253:2    0  200G  0 lvm  /var/chroot/var/lib/engine/data1
├─gaussdbvg-backupdata 253:3   0   20G  0 lvm  /var/chroot/var/lib/log/backup
└─gaussdbvg-etcddata  253:4    0   64G  0 lvm  After capacity expansion, the
vdc                   252:32   0  100G  0 disk capacity is 300 GB + 100 GB = 400 GB
```

Step 6 If the host status is in use and **Storage Type** is **Local SSD**, calculate and update the disk size of the node.

- Run the following command to calculate the disk size of the node and record it as {*data*}.

  SELECT ROUND(({*dataDisk*}*0.95-SUM(v1.SIZE_IN_BYTES)/1000000000)/(SELECT COUNT(*) FROM DBS_VOLUME v2 JOIN DBS_PARENTSHIP p2 ON v2.ID=p2.ENTITY_ID WHERE p2.PARENT_ENTITY_ID = {*nodeId*} AND p2.ENTITY_TYPE_TAG = 'vo' AND v2.PURPOSE = 'DATA')/40, 0)*40*1000000000 AS RESULT FROM DBS_PARENTSHIP p1 LEFT JOIN DBS_VOLUME v1 ON p1.ENTITY_ID = v1.ID AND v1.PURPOSE in ('ETCD', 'BACKUP') WHERE p1.PARENT_ENTITY_ID = {*nodeId*} AND p1.ENTITY_TYPE_TAG = 'vo';

- Run the following command to update the disk size of the node:

  UPDATE DBS_VOLUME SET UPDATED_AT=NOW(), SIZE_IN_BYTES={*data*} WHERE ID in (SELECT ID FROM DBS_VOLUME WHERE ID IN (SELECT ENTITY_ID FROM DBS_PARENTSHIP WHERE PARENT_ENTITY_ID = {*nodeId*} AND ENTITY_TYPE_TAG = 'vo') AND PURPOSE = 'DATA');

Step 7 If the host status is in use and **Storage Type** is **Flash storage**, run the following command to update the disk size of the node:

UPDATE DBS_VOLUME SET UPDATED_AT=NOW(), SIZE_IN_BYTES= ROUND({*dataDisk*}*0.95*1024*1024, 0) WHERE ID in (SELECT ID FROM DBS_VOLUME WHERE ID IN (SELECT ENTITY_ID FROM DBS_PARENTSHIP WHERE PARENT_ENTITY_ID = {*nodeId*} AND ENTITY_TYPE_TAG = 'vo') AND PURPOSE = 'DATA');

Step 8 Run the **\q** command to exit the core database.

Step 9 **Log in to GaussDB Management Platform (TPOPS).**

Step 10 Query the host list and check whether the data disk size of the host is updated. If it is not, check whether the operation in **Step 5** is successfully executed.

Step 11 If the host status is in use, check whether the storage space of the instance on the host is updated. If it is not, check whether the operations in **Step 6** or **Step 7** has been executed.

----End

8.6.4 Checking the Name of the AZ Used for Installing an Instance

Scenarios

This section describes how to check the name of AZ used for installing an instance on a host that already has instances installed.

Procedure

Step 1 Log in to the host as user **root**.

Step 2 Run the following commands to switch to the database user and update environment variables (user **Ruby** as an example):

su - Ruby

source ~/gauss_env_file

Step 3 Run the following command to check the name of the AZ used for instance installation:

cm_ctl query -CvzALL


```
[Ruby@host-192-168-1-90 ~]$ cm_ctl query -CvzALL
[  CMServer State   ]

node                  instance state
--------------------------------------
test 1  192.168.1.184 1        Primary
test 2  192.168.1.75  2        Standby
test 3  192.168.1.90  3        Standby

[    ETCD State     ]

node                  instance state
-------------------------------------------
test 1  192.168.1.184 7001     StateFollower
test 2  192.168.1.75  7002     StateFollower
test 3  192.168.1.90  7003     StateLeader

[   Cluster State   ]

cluster_state   : Normal
redistributing  : No
balanced        : Yes
current_az      : AZ_ALL

[  Datanode State   ]

node                  instance state           | node                  instance state           | node                  instance state
-----------------------------------------------------------------------------------------------------------------------------------------
test 1  192.168.1.184 6001     P Primary Normal | test 2  192.168.1.75  6002     S Standby Normal | test 3  192.168.1.90  6003     S Standby Normal
```

As shown in the preceding figure, the name of the AZ used for instance installation is **test**.

----End

8.7 Large Memory Required Due to Memory Leaking of the Audit Service in the Kylin OS

Symptom

This section describes the memory leakage risk of a specific audit version. If the memory leaks, the audit service occupies a large amount of memory. You can upgrade the audit service as required.

Procedure

Step 1 Log in to a server on the management plane as the **root** user.

Step 2 Run the following command to check the audit service version:

rpm -qa audit

```
[root@dbsnoname2 test]# rpm -qa audit
audit-3.0-5.se.06.ky10.aarch64
```

- If the version is audit-3.0-5.se.06, upgrade it.
- If the version is not audit-3.0-5.se.06, skip **Step 3** to **Step 5**.

Step 3 Obtain the audit-3.0-5.se.08.ky10 version package (**x86 system download address** for x86-based OSs and **Arm system download address** for Arm-based OSs) by referring to the **recovery guide**. You need to download the following RPM packages:

python3-audit-3.0-5.se.08.ky10.*.rpm

audit-libs-3.0-5.se.08.ky10.*.rpm

audit-3.0-5.se.08.ky10.*.rpm

Step 4 Upload the RPM packages downloaded in **Step 3** to any temporary directory, go to the directory, and run the following command to upgrade the component:

rpm -Uvh *.rpm

Step 5 Run the following commands to restart the service:

systemctl daemon-reload

systemctl start auditd.service

Step 6 Run the following command to check whether the service is normal.

systemctl status auditd.service

```
[root@localhost BUILD]# systemctl status auditd.service
● auditd.service - Security Auditing Service
   Loaded: loaded (/usr/lib/systemd/system/auditd.service; enabled; vendor pres>
   Active: active (running) since Thu 2022-01-06 13:41:14 CST; 15h ago
     Docs: man:auditd(8)
           https://github.com/linux-audit/audit-documentation
  Process: 930 ExecStart=/sbin/auditd (code=exited, status=0/SUCCESS)
  Process: 943 ExecStartPost=/sbin/augenrules --load (code=exited, status=0/SUC>
 Main PID: 937 (auditd)
    Tasks: 5
   Memory: 4.9M
   CGroup: /system.slice/auditd.service
           ├─937 /sbin/auditd
           ├─940 /usr/sbin/sedispatch
           └─941 /bin/audisp-secaudit
```

If **active (running)** is displayed, the service is normal.

Step 7 Run the following command to check the audit service version:

rpm -qa audit

```
[root@dbsnoname2 ~]# rpm -qa audit
audit-3.0-5.se.08.ky10.aarch64
```

If **audit-3.0-5.se.08.ky10.*** is displayed, the service is upgraded.

----End

A Appendixes

> **CAUTION**
>
> Perform the following operations as the **root** user. After the operations are complete, log out of the system as the **root** user in a timely manner to prevent misoperations.

A.1 Configuring the Installation Configuration File

This topic describes the parameters in the GaussDB Management Platform (TPOPS) installation configuration file.

Step 1 Run the following command as the **root** user to open the directory where the software package is stored:

cd /data/docker-service/config

Step 2 Open the **user_edit_file.conf** file and set parameters.

vi user_edit_file.conf

The following table provides parameter details.

Table A-1 Parameters

Parameter	Description
ssh_port	Port number for SSH communication between nodes. The default value is **22**.
influxDB_install_ip1	InfluxDB installation node 1.
influxDB_install_ip2	InfluxDB installation node 2.
sftp_install_ip1	SFTP installation node 1.
sftp_install_ip2	SFTP installation node 2.

Parameter	Description
node1_ip	IP address 1 of the GaussDB Management Platform (TPOPS) metadata database.
node2_ip	IP address 2 of the GaussDB Management Platform (TPOPS) metadata database.
node3_ip	IP address 3 of the GaussDB Management Platform (TPOPS) metadata database.
main_path	Microservice running directory.
log_path	Log directory.
sftp_path	SFTP installation directory.
gauss_path	Database installation directory.
influx_path	InfluxDB installation directory.
docker_path	Docker installation directory.
backup_path	Backup directory.
node1_ip2	The GaussDB Management Platform (TPOPS) metadata database IP address 1 that can communicate with the GaussDB instances over SSH and be pinged.
node2_ip2	The GaussDB Management Platform (TPOPS) metadata database IP address 2 that can communicate with the GaussDB instances over SSH and be pinged.
node3_ip2	The GaussDB Management Platform (TPOPS) metadata database IP address 3 that can communicate with the GaussDB instances over SSH and be pinged.
service_group_id	Group ID of the **service** user.
service_user_id	ID of the **service** user.
uninstall_all	Whether to uninstall all nodes on the management platform.
use_cgroup	Whether to use cgroup to isolate resources when DRS is deployed together with the management platform.

NOTE

- Chinese paths are not allowed in the configuration file.
- The IP address and port number of the service cannot be changed during the installation.

----End

A.2 Pre-check Error Handling

NOTE

To handle the error information in the pre-check, perform the following steps:

1. Run the following command on all GaussDB Management Platform (TPOPS) nodes to check whether JRE is installed:

   java -version

   Information similar to the following is displayed:

   ```
   openjdk version "1.8.*"
   OpenJDK Runtime Environment Bisheng (build 1.8.*)
   OpenJDK 64-Bit Server VM Bisheng (build 25.*, mixed mode)
   ```

   If the preceding information is not displayed or a message is displayed indicating that the command does not exist, install and configure JRE by referring to **Installing JRE**.

2. Run the following command on all GaussDB Management Platform (TPOPS) nodes to check whether expect is installed:

   expect -v

   Information similar to the following is displayed:

   ```
   expect version 5.45.4
   ```

   If no expect version is displayed or a message is displayed indicating that the command does not exist, install expect by referring to **Installing the Expect**.

3. Run the following command on all GaussDB Management Platform (TPOPS) nodes to check whether the OpenSSL tool is installed:

   openssl version

   ```
   OpenSSL 1.1.1f  31 Mar 2020
   ```

   If no OpenSSL version is displayed or a message is displayed indicating that the command does not exist, run the following command to install OpenSSL. For details about how to configure the Yum source, see **Configuring a Yum Repository**.

   yum -y install openssl

4. On a management-plane node, run the following command to check whether the dos2unix tool is installed:

   dos2unix -V

   If information similar to the following is displayed, the tool has been installed:

   ```
   dos2unix 7.4.1 (2019-09-24)
   With Unicode UTF-16 support.
   With native language support.
   With support to preserve the user and group ownership of files.
   LOCALEDIR: /usr/share/locale
   http://waterlan.home.xs4all.nl/dos2unix.html
   ```


If no dos2unix version is displayed or a message is displayed indicating that the command does not exist, run the following command to install dos2unix:

yum -y install dos2unix

5. The installation user needs to run the **locale** command to check whether the character set of the operating system is **en_US.UTF-8**. If it is not, run the following command to modify or add the configuration:

   vi /etc/sysconfig/i18n

   Change the value of **LANG** to **en_US.UTF-8** and run the **source /etc/sysconfig/i18n** command.

   NOTE

   Some hosts do not have the **/etc/sysconfig/i18n** directory. Run the following command:

   vi /etc/locale.conf

   Change the value of **LANG** to **en_US.UTF-8** and run the **source /etc/locale.conf** command.

6. Log in to all GaussDB Management Platform (TPOPS) management nodes as the **root** user and run the following command to check whether net-tools is installed:

   ifconfig

   If the NIC information is displayed in the command output, the tool has been installed. If a message is displayed indicating that the command does not exist, see **Installing net-tools**.

7. Log in to all GaussDB Management Platform (TPOPS) management nodes as the **root** user and run the following command to check whether libcgroup is installed:

   rpm -qa | grep libcgroup

   - If the version and architecture of libcgroup are displayed in the command output, libcgroup has been installed.
     ```
     libcgroup-0.41-23.ky10.aarch64
     ```
   - If no command output is displayed, libcgroup has not been installed. Run the following command to install libcgroup:

     yum -y install libcgroup

8. Log in to all nodes where GaussDB Management Platform (TPOPS) is to be installed as the **root** user and run the following command to check the Python version:

   python3 --version

   If the following information is displayed, the Python3 version is used:

   ```
   Python 3.7.9
   ```

   Python 3.7 or later must be installed on all nodes where GaussDB Management Platform (TPOPS) is installed. If the version is not Python 3.7 or later, re-install Python by referring to **Installing Python 3**.

   NOTE

   Python 3 must be contained in system images in GaussDB Management Platform (TPOPS). The installation environment is Python 3.7.9.

A.3 Open Source Software List

NOTE

The file path of the open source software is **GaussDB_OS_PATCH_*******.zip/os_patch_all**.

x86-based Kylin V10 SP1

Table A-2 kylin_v10_x86_sp1

Software	File Path	File Name
libbasicobjects.so.0	os_patch_all/os/kylin_v10_x86_sp1/rpm	ding-libs-0.6.1-42.ky10.x86_64.rpm
libbasicobjects.so.0.1.0	os_patch_all/os/kylin_v10_x86_sp1/rpm	ding-libs-0.6.1-42.ky10.x86_64.rpm
libcollection.so.4	os_patch_all/os/kylin_v10_x86_sp1/rpm	ding-libs-0.6.1-42.ky10.x86_64.rpm
libcollection.so.4.1.1	os_patch_all/os/kylin_v10_x86_sp1/rpm	ding-libs-0.6.1-42.ky10.x86_64.rpm
libdhash.so.1	os_patch_all/os/kylin_v10_x86_sp1/rpm	ding-libs-0.6.1-42.ky10.x86_64.rpm
libdhash.so.1.1.0	os_patch_all/os/kylin_v10_x86_sp1/rpm	ding-libs-0.6.1-42.ky10.x86_64.rpm
libini_config.so.5	os_patch_all/os/kylin_v10_x86_sp1/rpm	ding-libs-0.6.1-42.ky10.x86_64.rpm
libini_config.so.5.2.1	os_patch_all/os/kylin_v10_x86_sp1/rpm	ding-libs-0.6.1-42.ky10.x86_64.rpm
libpath_utils.so.1	os_patch_all/os/kylin_v10_x86_sp1/rpm	ding-libs-0.6.1-42.ky10.x86_64.rpm
libpath_utils.so.1.0.1	os_patch_all/os/kylin_v10_x86_sp1/rpm	ding-libs-0.6.1-42.ky10.x86_64.rpm

Software	File Path	File Name
libref_array.so.1	os_patch_all/os/kylin_v10_x86_sp1/rpm	ding-libs-0.6.1-42.ky10.x86_64.rpm
libref_array.so.1.2.1	os_patch_all/os/kylin_v10_x86_sp1/rpm	ding-libs-0.6.1-42.ky10.x86_64.rpm
libcom_err.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	e2fsprogs-devel-1.45.6-0.p01.ky10.x86_64.rpm
libe2p.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	e2fsprogs-devel-1.45.6-0.p01.ky10.x86_64.rpm
libext2fs.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	e2fsprogs-devel-1.45.6-0.p01.ky10.x86_64.rpm
libss.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	e2fsprogs-devel-1.45.6-0.p01.ky10.x86_64.rpm
libexpect.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	expect-5.45.4-3.ky10.x86_64.rpm
libexpect5.45.4.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	expect-5.45.4-3.ky10.x86_64.rpm
proxymech.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	gssproxy-0.8.0-11.ky10.x86_64.rpm
libkeyutils.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	keyutils-libs-devel-1.5.10-11.ky10.x86_64.rpm
libkadm5clnt_mit.so.11	os_patch_all/os/kylin_v10_x86_sp1/rpm	krb5-1.17-9.ky10.x86_64.rpm
libkadm5clnt_mit.so.11.0	os_patch_all/os/kylin_v10_x86_sp1/rpm	krb5-1.17-9.ky10.x86_64.rpm
libkadm5srv_mit.so.11	os_patch_all/os/kylin_v10_x86_sp1/rpm	krb5-1.17-9.ky10.x86_64.rpm
libkadm5srv_mit.so.11.0	os_patch_all/os/kylin_v10_x86_sp1/rpm	krb5-1.17-9.ky10.x86_64.rpm
libgssapi_krb5.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	krb5-devel-1.17-9.ky10.x86_64.rpm

Software	File Path	File Name
libgssrpc.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	krb5-devel-1.17-9.ky10.x86_64.rpm
libk5crypto.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	krb5-devel-1.17-9.ky10.x86_64.rpm
libkadm5clnt.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	krb5-devel-1.17-9.ky10.x86_64.rpm
libkadm5clnt_mit.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	krb5-devel-1.17-9.ky10.x86_64.rpm
libkadm5srv.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	krb5-devel-1.17-9.ky10.x86_64.rpm
libkadm5srv_mit.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	krb5-devel-1.17-9.ky10.x86_64.rpm
libkdb5.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	krb5-devel-1.17-9.ky10.x86_64.rpm
libkrad.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	krb5-devel-1.17-9.ky10.x86_64.rpm
libkrb5.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	krb5-devel-1.17-9.ky10.x86_64.rpm
libkrb5support.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	krb5-devel-1.17-9.ky10.x86_64.rpm
libcgroup.so.1	os_patch_all/os/kylin_v10_x86_sp1/rpm	libcgroup-0.41-23.ky10.x86_64.rpm
libcgroup.so.1.0.41	os_patch_all/os/kylin_v10_x86_sp1/rpm	libcgroup-0.41-23.ky10.x86_64.rpm
pam_cgroup.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	libcgroup-0.41-23.ky10.x86_64.rpm
libffi.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	libffi-devel-3.3-7.ky10.x86_64.rpm
libselinux.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	libselinux-devel-2.9-se.05.ky10.x86_64.rpm

Software	File Path	File Name
libsepol.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	libsepol-devel-2.9-1.ky10.x86_64.rpm
libstdc++.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	libstdc++-devel-7.3.0-20190804.h30.ky10.x86_64.rpm
libverto-glib.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	libverto-devel-0.3.1-2.ky10.x86_64.rpm
libverto-libev.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	libverto-devel-0.3.1-2.ky10.x86_64.rpm
libverto-libevent.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	libverto-devel-0.3.1-2.ky10.x86_64.rpm
libverto.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	libverto-devel-0.3.1-2.ky10.x86_64.rpm
libnfsidmap.so.1	os_patch_all/os/kylin_v10_x86_sp1/rpm	nfs-utils-2.4.2-2.ky10.x86_64.rpm
libnfsidmap.so.1.0.0	os_patch_all/os/kylin_v10_x86_sp1/rpm	nfs-utils-2.4.2-2.ky10.x86_64.rpm
nsswitch.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	nfs-utils-2.4.2-2.ky10.x86_64.rpm
static.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	nfs-utils-2.4.2-2.ky10.x86_64.rpm
umich_ldap.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	nfs-utils-2.4.2-2.ky10.x86_64.rpm
libcrypto.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	openssl-devel-1.1.1f-2.ky10.x86_64.rpm
libssl.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	openssl-devel-1.1.1f-2.ky10.x86_64.rpm
libpcre2-16.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	pcre2-devel-10.33-2.ky10.x86_64.rpm

Software	File Path	File Name
libpcre2-32.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	pcre2-devel-10.33-2.ky10.x86_64.rpm
libpcre2-8.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	pcre2-devel-10.33-2.ky10.x86_64.rpm
libpcre2-posix.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	pcre2-devel-10.33-2.ky10.x86_64.rpm
libpython3.7m.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	python3-devel-3.7.9-6.ky10.x86_64.rpm
_ctypes_test.cpython-37m-x86_64-linux-gnu.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	python3-devel-3.7.9-6.ky10.x86_64.rpm
_testbuffer.cpython-37m-x86_64-linux-gnu.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	python3-devel-3.7.9-6.ky10.x86_64.rpm
_testcapi.cpython-37m-x86_64-linux-gnu.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	python3-devel-3.7.9-6.ky10.x86_64.rpm
_testimportmultiple.cpython-37m-x86_64-linux-gnu.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	python3-devel-3.7.9-6.ky10.x86_64.rpm
_tkinter.cpython-37m-x86_64-linux-gnu.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	python3-devel-3.7.9-6.ky10.x86_64.rpm
_xxtestfuzz.cpython-37m-x86_64-linux-gnu.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	python3-devel-3.7.9-6.ky10.x86_64.rpm
libz.so	os_patch_all/os/kylin_v10_x86_sp1/rpm	zlib-devel-1.2.11-17.1.ky10.x86_64.rpm

x86-based Kylin V10 SP2

Table A-3 kylin_v10_x86_sp2

Software	File Path	File Name
libbasicobjects.so.0	os_patch_all/os/kylin_v10_x86_sp2/rpm	ding-libs-0.6.1-42.ky10.x86_64.rpm
libbasicobjects.so.0.1.0	os_patch_all/os/kylin_v10_x86_sp2/rpm	ding-libs-0.6.1-42.ky10.x86_64.rpm
libcollection.so.4	os_patch_all/os/kylin_v10_x86_sp2/rpm	ding-libs-0.6.1-42.ky10.x86_64.rpm
libcollection.so.4.1.1	os_patch_all/os/kylin_v10_x86_sp2/rpm	ding-libs-0.6.1-42.ky10.x86_64.rpm
libdhash.so.1	os_patch_all/os/kylin_v10_x86_sp2/rpm	ding-libs-0.6.1-42.ky10.x86_64.rpm
libdhash.so.1.1.0	os_patch_all/os/kylin_v10_x86_sp2/rpm	ding-libs-0.6.1-42.ky10.x86_64.rpm
libini_config.so.5	os_patch_all/os/kylin_v10_x86_sp2/rpm	ding-libs-0.6.1-42.ky10.x86_64.rpm
libini_config.so.5.2.1	os_patch_all/os/kylin_v10_x86_sp2/rpm	ding-libs-0.6.1-42.ky10.x86_64.rpm
libpath_utils.so.1	os_patch_all/os/kylin_v10_x86_sp2/rpm	ding-libs-0.6.1-42.ky10.x86_64.rpm
libpath_utils.so.1.0.1	os_patch_all/os/kylin_v10_x86_sp2/rpm	ding-libs-0.6.1-42.ky10.x86_64.rpm
libref_array.so.1	os_patch_all/os/kylin_v10_x86_sp2/rpm	ding-libs-0.6.1-42.ky10.x86_64.rpm
libref_array.so.1.2.1	os_patch_all/os/kylin_v10_x86_sp2/rpm	ding-libs-0.6.1-42.ky10.x86_64.rpm

Software	File Path	File Name
libcom_err.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	e2fsprogs-devel-1.45.6-0.p01.ky10.x86_64.rpm
libe2p.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	e2fsprogs-devel-1.45.6-0.p01.ky10.x86_64.rpm
libext2fs.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	e2fsprogs-devel-1.45.6-0.p01.ky10.x86_64.rpm
libss.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	e2fsprogs-devel-1.45.6-0.p01.ky10.x86_64.rpm
libexpect.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	expect-5.45.4-3.ky10.x86_64.rpm
libexpect5.45.4.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	expect-5.45.4-3.ky10.x86_64.rpm
proxymech.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	gssproxy-0.8.3-1.ky10.x86_64.rpm
libkeyutils.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	keyutils-libs-devel-1.6.3-1.ky10.x86_64.rpm
libkadm5clnt_mit.so.12	os_patch_all/os/kylin_v10_x86_sp2/rpm	krb5-1.18.2-1.ky10.x86_64.rpm
libkadm5clnt_mit.so.12.0	os_patch_all/os/kylin_v10_x86_sp2/rpm	krb5-1.18.2-1.ky10.x86_64.rpm
libkadm5srv_mit.so.12	os_patch_all/os/kylin_v10_x86_sp2/rpm	krb5-1.18.2-1.ky10.x86_64.rpm
libkadm5srv_mit.so.12.0	os_patch_all/os/kylin_v10_x86_sp2/rpm	krb5-1.18.2-1.ky10.x86_64.rpm
libgssapi_krb5.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	krb5-devel-1.18.2-1.ky10.x86_64.rpm
libgssrpc.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	krb5-devel-1.18.2-1.ky10.x86_64.rpm
libk5crypto.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	krb5-devel-1.18.2-1.ky10.x86_64.rpm

Software	File Path	File Name
libkadm5clnt.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	krb5-devel-1.18.2-1.ky10.x86_64.rpm
libkadm5clnt_mit.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	krb5-devel-1.18.2-1.ky10.x86_64.rpm
libkadm5srv.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	krb5-devel-1.18.2-1.ky10.x86_64.rpm
libkadm5srv_mit.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	krb5-devel-1.18.2-1.ky10.x86_64.rpm
libkdb5.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	krb5-devel-1.18.2-1.ky10.x86_64.rpm
libkrad.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	krb5-devel-1.18.2-1.ky10.x86_64.rpm
libkrb5.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	krb5-devel-1.18.2-1.ky10.x86_64.rpm
libkrb5support.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	krb5-devel-1.18.2-1.ky10.x86_64.rpm
libcgroup.so.1	os_patch_all/os/kylin_v10_x86_sp2/rpm	libcgroup-0.42.2-1.ky10.x86_64.rpm
libcgroup.so.1.0.42	os_patch_all/os/kylin_v10_x86_sp2/rpm	libcgroup-0.42.2-1.ky10.x86_64.rpm
pam_cgroup.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	libcgroup-0.42.2-1.ky10.x86_64.rpm
libstdc++.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	libstdc++-devel-7.3.0-20190804.35.p02.ky10.x86_64.rpm
libverto-glib.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	libverto-devel-0.3.1-2.ky10.x86_64.rpm
libverto-libev.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	libverto-devel-0.3.1-2.ky10.x86_64.rpm

Software	File Path	File Name
libverto-libevent.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	libverto-devel-0.3.1-2.ky10.x86_64.rpm
libverto.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	libverto-devel-0.3.1-2.ky10.x86_64.rpm
libnfsidmap.so.1	os_patch_all/os/kylin_v10_x86_sp2/rpm	nfs-utils-2.5.1-3.ky10.x86_64.rpm
libnfsidmap.so.1.0.0	os_patch_all/os/kylin_v10_x86_sp2/rpm	nfs-utils-2.5.1-3.ky10.x86_64.rpm
nsswitch.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	nfs-utils-2.5.1-3.ky10.x86_64.rpm
regex.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	nfs-utils-2.5.1-3.ky10.x86_64.rpm
static.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	nfs-utils-2.5.1-3.ky10.x86_64.rpm
umich_ldap.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	nfs-utils-2.5.1-3.ky10.x86_64.rpm
libcrypto.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	openssl-devel-1.1.1f-4.p01.ky10.x86_64.rpm
libssl.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	openssl-devel-1.1.1f-4.p01.ky10.x86_64.rpm
_ctypes_test.cpython-37m-x86_64-linux-gnu.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	python3-devel-3.7.9-6.ky10.x86_64.rpm
_testbuffer.cpython-37m-x86_64-linux-gnu.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	python3-devel-3.7.9-6.ky10.x86_64.rpm
_testcapi.cpython-37m-x86_64-linux-gnu.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	python3-devel-3.7.9-6.ky10.x86_64.rpm

Software	File Path	File Name
_testimportmulti-ple.cpython-37m-x86_64-linux-gnu.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	python3-devel-3.7.9-6.ky10.x86_64.rpm
_tkinter.cpython-37m-x86_64-linux-gnu.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	python3-devel-3.7.9-6.ky10.x86_64.rpm
_xxtestfuzz.cpython-37m-x86_64-linux-gnu.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	python3-devel-3.7.9-6.ky10.x86_64.rpm
libpython3.7m.so	os_patch_all/os/kylin_v10_x86_sp2/rpm	python3-devel-3.7.9-6.ky10.x86_64.rpm

Arm-based Kylin V10 SP1

Table A-4 kylin_v10_arm_sp1

Software	File Path	File Name
libbasicobjects.so.0	os_patch_all/os/kylin_v10_arm_sp1/rpm	ding-libs-0.6.1-42.ky10.aarch64.rpm
libbasicobjects.so.0.1.0	os_patch_all/os/kylin_v10_arm_sp1/rpm	ding-libs-0.6.1-42.ky10.aarch64.rpm
libcollection.so.4	os_patch_all/os/kylin_v10_arm_sp1/rpm	ding-libs-0.6.1-42.ky10.aarch64.rpm
libcollection.so.4.1.1	os_patch_all/os/kylin_v10_arm_sp1/rpm	ding-libs-0.6.1-42.ky10.aarch64.rpm
libdhash.so.1	os_patch_all/os/kylin_v10_arm_sp1/rpm	ding-libs-0.6.1-42.ky10.aarch64.rpm
libdhash.so.1.1.0	os_patch_all/os/kylin_v10_arm_sp1/rpm	ding-libs-0.6.1-42.ky10.aarch64.rpm
libini_config.so.5	os_patch_all/os/kylin_v10_arm_sp1/rpm	ding-libs-0.6.1-42.ky10.aarch64.rpm

Software	File Path	File Name
libini_config.so.5.2.1	os_patch_all/os/kylin_v10_arm_sp1/rpm	ding-libs-0.6.1-42.ky10.aarch64.rpm
libpath_utils.so.1	os_patch_all/os/kylin_v10_arm_sp1/rpm	ding-libs-0.6.1-42.ky10.aarch64.rpm
libpath_utils.so.1.0.1	os_patch_all/os/kylin_v10_arm_sp1/rpm	ding-libs-0.6.1-42.ky10.aarch64.rpm
libref_array.so.1	os_patch_all/os/kylin_v10_arm_sp1/rpm	ding-libs-0.6.1-42.ky10.aarch64.rpm
libref_array.so.1.2.1	os_patch_all/os/kylin_v10_arm_sp1/rpm	ding-libs-0.6.1-42.ky10.aarch64.rpm
libcom_err.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	e2fsprogs-devel-1.45.6-0.p01.ky10.aarch64.rpm
libe2p.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	e2fsprogs-devel-1.45.6-0.p01.ky10.aarch64.rpm
libext2fs.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	e2fsprogs-devel-1.45.6-0.p01.ky10.aarch64.rpm
libss.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	e2fsprogs-devel-1.45.6-0.p01.ky10.aarch64.rpm
libexpect.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	expect-5.45.4-3.ky10.aarch64.rpm
libexpect5.45.4.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	expect-5.45.4-3.ky10.aarch64.rpm
proxymech.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	gssproxy-0.8.0-11.ky10.aarch64.rpm
libkeyutils.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	keyutils-libs-devel-1.5.10-11.ky10.aarch64.rpm
libkadm5clnt_mit.so.11	os_patch_all/os/kylin_v10_arm_sp1/rpm	krb5-1.17-9.ky10.aarch64.rpm
libkadm5clnt_mit.so.11.0	os_patch_all/os/kylin_v10_arm_sp1/rpm	krb5-1.17-9.ky10.aarch64.rpm

Software	File Path	File Name
libkadm5srv_mit.so.11	os_patch_all/os/kylin_v10_arm_sp1/rpm	krb5-1.17-9.ky10.aarch64.rpm
libkadm5srv_mit.so.11.0	os_patch_all/os/kylin_v10_arm_sp1/rpm	krb5-1.17-9.ky10.aarch64.rpm
libgssapi_krb5.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	krb5-devel-1.17-9.ky10.aarch64.rpm
libgssrpc.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	krb5-devel-1.17-9.ky10.aarch64.rpm
libk5crypto.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	krb5-devel-1.17-9.ky10.aarch64.rpm
libkadm5clnt.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	krb5-devel-1.17-9.ky10.aarch64.rpm
libkadm5clnt_mit.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	krb5-devel-1.17-9.ky10.aarch64.rpm
libkadm5srv.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	krb5-devel-1.17-9.ky10.aarch64.rpm
libkadm5srv_mit.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	krb5-devel-1.17-9.ky10.aarch64.rpm
libkdb5.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	krb5-devel-1.17-9.ky10.aarch64.rpm
libkrad.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	krb5-devel-1.17-9.ky10.aarch64.rpm
libkrb5.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	krb5-devel-1.17-9.ky10.aarch64.rpm
libkrb5support.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	krb5-devel-1.17-9.ky10.aarch64.rpm
libcgroup.so.1	os_patch_all/os/kylin_v10_arm_sp1/rpm	libcgroup-0.41-23.ky10.aarch64.rpm
libcgroup.so.1.0.41	os_patch_all/os/kylin_v10_arm_sp1/rpm	libcgroup-0.41-23.ky10.aarch64.rpm

Software	File Path	File Name
pam_cgroup.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	libcgroup-0.41-23.ky10.aarch64.rpm
libffi.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	libffi-devel-3.3-7.ky10.aarch64.rpm
libselinux.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	libselinux-devel-2.9-se.05.ky10.aarch64.rpm
libsepol.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	libsepol-devel-2.9-1.ky10.aarch64.rpm
libstdc++.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	libstdc++-devel-7.3.0-20190804.h30.ky10.aarch64.rpm
libverto-glib.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	libverto-devel-0.3.1-2.ky10.aarch64.rpm
libverto-libev.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	libverto-devel-0.3.1-2.ky10.aarch64.rpm
libverto-libevent.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	libverto-devel-0.3.1-2.ky10.aarch64.rpm
libverto.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	libverto-devel-0.3.1-2.ky10.aarch64.rpm
libnfsidmap.so.1	os_patch_all/os/kylin_v10_arm_sp1/rpm	nfs-utils-2.4.2-2.ky10.aarch64.rpm
libnfsidmap.so.1.0.0	os_patch_all/os/kylin_v10_arm_sp1/rpm	nfs-utils-2.4.2-2.ky10.aarch64.rpm
nsswitch.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	nfs-utils-2.4.2-2.ky10.aarch64.rpm
static.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	nfs-utils-2.4.2-2.ky10.aarch64.rpm
umich_ldap.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	nfs-utils-2.4.2-2.ky10.aarch64.rpm

Software	File Path	File Name
libcrypto.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	openssl-devel-1.1.1f-2.ky10.aarch64.rpm
libssl.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	openssl-devel-1.1.1f-2.ky10.aarch64.rpm
libpcre2-16.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	pcre2-devel-10.33-2.ky10.aarch64.rpm
libpcre2-32.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	pcre2-devel-10.33-2.ky10.aarch64.rpm
libpcre2-8.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	pcre2-devel-10.33-2.ky10.aarch64.rpm
libpcre2-posix.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	pcre2-devel-10.33-2.ky10.aarch64.rpm
libpython3.7m.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	python3-devel-3.7.9-6.ky10.aarch64.rpm
_ctypes_test.cpython-37m-aarch64-linux-gnu.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	python3-devel-3.7.9-6.ky10.aarch64.rpm
_testbuffer.cpython-37m-aarch64-linux-gnu.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	python3-devel-3.7.9-6.ky10.aarch64.rpm
_testcapi.cpython-37m-aarch64-linux-gnu.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	python3-devel-3.7.9-6.ky10.aarch64.rpm
_testimportmultiple.cpython-37m-aarch64-linux-gnu.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	python3-devel-3.7.9-6.ky10.aarch64.rpm
_tkinter.cpython-37m-aarch64-linux-gnu.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	python3-devel-3.7.9-6.ky10.aarch64.rpm
_xxtestfuzz.cpython-37m-aarch64-linux-gnu.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	python3-devel-3.7.9-6.ky10.aarch64.rpm

Software	File Path	File Name
libz.so	os_patch_all/os/kylin_v10_arm_sp1/rpm	zlib-devel-1.2.11-17.1.ky10.aarch64.rpm

Arm-based Kylin V10 SP2

Table A-5 kylin_v10_arm_sp2

Software	File Path	File Name
libbasicobjects.so.0	os_patch_all/os/kylin_v10_arm_sp2/rpm	ding-libs-0.6.1-42.ky10.aarch64.rpm
libbasicobjects.so.0.1.0	os_patch_all/os/kylin_v10_arm_sp2/rpm	ding-libs-0.6.1-42.ky10.aarch64.rpm
libcollection.so.4	os_patch_all/os/kylin_v10_arm_sp2/rpm	ding-libs-0.6.1-42.ky10.aarch64.rpm
libcollection.so.4.1.1	os_patch_all/os/kylin_v10_arm_sp2/rpm	ding-libs-0.6.1-42.ky10.aarch64.rpm
libdhash.so.1	os_patch_all/os/kylin_v10_arm_sp2/rpm	ding-libs-0.6.1-42.ky10.aarch64.rpm
libdhash.so.1.1.0	os_patch_all/os/kylin_v10_arm_sp2/rpm	ding-libs-0.6.1-42.ky10.aarch64.rpm
libini_config.so.5	os_patch_all/os/kylin_v10_arm_sp2/rpm	ding-libs-0.6.1-42.ky10.aarch64.rpm
libini_config.so.5.2.1	os_patch_all/os/kylin_v10_arm_sp2/rpm	ding-libs-0.6.1-42.ky10.aarch64.rpm
libpath_utils.so.1	os_patch_all/os/kylin_v10_arm_sp2/rpm	ding-libs-0.6.1-42.ky10.aarch64.rpm
libpath_utils.so.1.0.1	os_patch_all/os/kylin_v10_arm_sp2/rpm	ding-libs-0.6.1-42.ky10.aarch64.rpm

Software	File Path	File Name
libref_array.so.1	os_patch_all/os/kylin_v10_arm_sp2/rpm	ding-libs-0.6.1-42.ky10.aarch64.rpm
libref_array.so.1.2.1	os_patch_all/os/kylin_v10_arm_sp2/rpm	ding-libs-0.6.1-42.ky10.aarch64.rpm
libcom_err.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	e2fsprogs-devel-1.45.6-1.ky10.aarch64.rpm
libe2p.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	e2fsprogs-devel-1.45.6-1.ky10.aarch64.rpm
libext2fs.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	e2fsprogs-devel-1.45.6-1.ky10.aarch64.rpm
libss.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	e2fsprogs-devel-1.45.6-1.ky10.aarch64.rpm
libexpect.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	expect-5.45.4-3.ky10.aarch64.rpm
libexpect5.45.4.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	expect-5.45.4-3.ky10.aarch64.rpm
proxymech.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	gssproxy-0.8.3-1.ky10.aarch64.rpm
libkeyutils.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	keyutils-libs-devel-1.6.3-1.ky10.aarch64.rpm
libkadm5clnt_mit.so.12	os_patch_all/os/kylin_v10_arm_sp2/rpm	krb5-1.18.2-1.ky10.aarch64.rpm
libkadm5clnt_mit.so.12.0	os_patch_all/os/kylin_v10_arm_sp2/rpm	krb5-1.18.2-1.ky10.aarch64.rpm
libkadm5srv_mit.so.12	os_patch_all/os/kylin_v10_arm_sp2/rpm	krb5-1.18.2-1.ky10.aarch64.rpm
libkadm5srv_mit.so.12.0	os_patch_all/os/kylin_v10_arm_sp2/rpm	krb5-1.18.2-1.ky10.aarch64.rpm
libgssapi_krb5.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	krb5-devel-1.18.2-1.ky10.aarch64.rpm

Software	File Path	File Name
libgssrpc.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	krb5-devel-1.18.2-1.ky10.aarch64.rpm
libk5crypto.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	krb5-devel-1.18.2-1.ky10.aarch64.rpm
libkadm5clnt.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	krb5-devel-1.18.2-1.ky10.aarch64.rpm
libkadm5clnt_mit.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	krb5-devel-1.18.2-1.ky10.aarch64.rpm
libkadm5srv.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	krb5-devel-1.18.2-1.ky10.aarch64.rpm
libkadm5srv_mit.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	krb5-devel-1.18.2-1.ky10.aarch64.rpm
libkdb5.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	krb5-devel-1.18.2-1.ky10.aarch64.rpm
libkrad.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	krb5-devel-1.18.2-1.ky10.aarch64.rpm
libkrb5.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	krb5-devel-1.18.2-1.ky10.aarch64.rpm
libkrb5support.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	krb5-devel-1.18.2-1.ky10.aarch64.rpm
libcgroup.so.1	os_patch_all/os/kylin_v10_arm_sp2/rpm	libcgroup-0.42.2-1.ky10.aarch64.rpm
libcgroup.so.1.0.42	os_patch_all/os/kylin_v10_arm_sp2/rpm	libcgroup-0.42.2-1.ky10.aarch64.rpm
pam_cgroup.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	libcgroup-0.42.2-1.ky10.aarch64.rpm
libstdc++.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	libstdc++-devel-7.3.0-20190804.35.p02.ky10.aarch64.rpm

Software	File Path	File Name
libverto-glib.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	libverto-devel-0.3.1-2.ky10.aarch64.rpm
libverto-libev.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	libverto-devel-0.3.1-2.ky10.aarch64.rpm
libverto-libevent.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	libverto-devel-0.3.1-2.ky10.aarch64.rpm
libverto.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	libverto-devel-0.3.1-2.ky10.aarch64.rpm
libnfsidmap.so.1	os_patch_all/os/kylin_v10_arm_sp2/rpm	nfs-utils-2.5.1-3.ky10.aarch64.rpm
libnfsidmap.so.1.0.0	os_patch_all/os/kylin_v10_arm_sp2/rpm	nfs-utils-2.5.1-3.ky10.aarch64.rpm
nsswitch.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	nfs-utils-2.5.1-3.ky10.aarch64.rpm
regex.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	nfs-utils-2.5.1-3.ky10.aarch64.rpm
static.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	nfs-utils-2.5.1-3.ky10.aarch64.rpm
umich_ldap.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	nfs-utils-2.5.1-3.ky10.aarch64.rpm
libcrypto.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	openssl-devel-1.1.1f-4.p01.ky10.aarch64.rpm
libssl.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	openssl-devel-1.1.1f-4.p01.ky10.aarch64.rpm
libpython3.7m.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	python3-devel-3.7.9-6.ky10.aarch64.rpm

Software	File Path	File Name
_ctypes_test.cpython-37m-aarch64-linux-gnu.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	python3-devel-3.7.9-6.ky10.aarch64.rpm
_testbuffer.cpython-37m-aarch64-linux-gnu.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	python3-devel-3.7.9-6.ky10.aarch64.rpm
_testcapi.cpython-37m-aarch64-linux-gnu.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	python3-devel-3.7.9-6.ky10.aarch64.rpm
_testimportmultiple.cpython-37m-aarch64-linux-gnu.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	python3-devel-3.7.9-6.ky10.aarch64.rpm
_tkinter.cpython-37m-aarch64-linux-gnu.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	python3-devel-3.7.9-6.ky10.aarch64.rpm
_xxtestfuzz.cpython-37m-aarch64-linux-gnu.so	os_patch_all/os/kylin_v10_arm_sp2/rpm	python3-devel-3.7.9-6.ky10.aarch64.rpm

x86-based UnionTech OSs

Table A-6 uniontech_x86

Software	File Path	File Name
libbasicobjects.so.0	os_patch_all/os/uniontech_x86/rpm	ding-libs-0.6.1-42.uel20.x86_64.rpm
libbasicobjects.so.0.1.0	os_patch_all/os/uniontech_x86/rpm	ding-libs-0.6.1-42.uel20.x86_64.rpm
libcollection.so.4	os_patch_all/os/uniontech_x86/rpm	ding-libs-0.6.1-42.uel20.x86_64.rpm
libcollection.so.4.1.1	os_patch_all/os/uniontech_x86/rpm	ding-libs-0.6.1-42.uel20.x86_64.rpm
libdhash.so.1	os_patch_all/os/uniontech_x86/rpm	ding-libs-0.6.1-42.uel20.x86_64.rpm

Software	File Path	File Name
libdhash.so.1.1.0	os_patch_all/os/uniontech_x86/rpm	ding-libs-0.6.1-42.uel20.x86_64.rpm
libini_config.so.5	os_patch_all/os/uniontech_x86/rpm	ding-libs-0.6.1-42.uel20.x86_64.rpm
libini_config.so.5.2.1	os_patch_all/os/uniontech_x86/rpm	ding-libs-0.6.1-42.uel20.x86_64.rpm
libpath_utils.so.1	os_patch_all/os/uniontech_x86/rpm	ding-libs-0.6.1-42.uel20.x86_64.rpm
libpath_utils.so.1.0.1	os_patch_all/os/uniontech_x86/rpm	ding-libs-0.6.1-42.uel20.x86_64.rpm
libref_array.so.1	os_patch_all/os/uniontech_x86/rpm	ding-libs-0.6.1-42.uel20.x86_64.rpm
libref_array.so.1.2.1	os_patch_all/os/uniontech_x86/rpm	ding-libs-0.6.1-42.uel20.x86_64.rpm
libcom_err.so	os_patch_all/os/uniontech_x86/rpm	e2fsprogs-devel-1.45.6-7.uel20.x86_64.rpm
libe2p.so	os_patch_all/os/uniontech_x86/rpm	e2fsprogs-devel-1.45.6-7.uel20.x86_64.rpm
libext2fs.so	os_patch_all/os/uniontech_x86/rpm	e2fsprogs-devel-1.45.6-7.uel20.x86_64.rpm
libss.so	os_patch_all/os/uniontech_x86/rpm	e2fsprogs-devel-1.45.6-7.uel20.x86_64.rpm
libexpect.so	os_patch_all/os/uniontech_x86/rpm	expect-5.45.4-5.uel20.x86_64.rpm
libexpect5.45.4.so	os_patch_all/os/uniontech_x86/rpm	expect-5.45.4-5.uel20.x86_64.rpm
proxymech.so	os_patch_all/os/uniontech_x86/rpm	gssproxy-0.8.3-1.uel20.x86_64.rpm

Software	File Path	File Name
libkeyutils.so	os_patch_all/os/uniontech_x86/rpm	keyutils-libs-devel-1.6.3-1.uel20.x86_64.rpm
libkadm5clnt_mit.so.12	os_patch_all/os/uniontech_x86/rpm	krb5-1.18.2-5.uel20.x86_64.rpm
libkadm5clnt_mit.so.12.0	os_patch_all/os/uniontech_x86/rpm	krb5-1.18.2-5.uel20.x86_64.rpm
libkadm5srv_mit.so.12	os_patch_all/os/uniontech_x86/rpm	krb5-1.18.2-5.uel20.x86_64.rpm
libkadm5srv_mit.so.12.0	os_patch_all/os/uniontech_x86/rpm	krb5-1.18.2-5.uel20.x86_64.rpm
libgssapi_krb5.so	os_patch_all/os/uniontech_x86/rpm	krb5-devel-1.18.2-5.uel20.x86_64.rpm
libgssrpc.so	os_patch_all/os/uniontech_x86/rpm	krb5-devel-1.18.2-5.uel20.x86_64.rpm
libk5crypto.so	os_patch_all/os/uniontech_x86/rpm	krb5-devel-1.18.2-5.uel20.x86_64.rpm
libkadm5clnt.so	os_patch_all/os/uniontech_x86/rpm	krb5-devel-1.18.2-5.uel20.x86_64.rpm
libkadm5clnt_mit.so	os_patch_all/os/uniontech_x86/rpm	krb5-devel-1.18.2-5.uel20.x86_64.rpm
libkadm5srv.so	os_patch_all/os/uniontech_x86/rpm	krb5-devel-1.18.2-5.uel20.x86_64.rpm
libkadm5srv_mit.so	os_patch_all/os/uniontech_x86/rpm	krb5-devel-1.18.2-5.uel20.x86_64.rpm
libkdb5.so	os_patch_all/os/uniontech_x86/rpm	krb5-devel-1.18.2-5.uel20.x86_64.rpm
libkrad.so	os_patch_all/os/uniontech_x86/rpm	krb5-devel-1.18.2-5.uel20.x86_64.rpm
libkrb5.so	os_patch_all/os/uniontech_x86/rpm	krb5-devel-1.18.2-5.uel20.x86_64.rpm

Software	File Path	File Name
libkrb5support.so	os_patch_all/os/uniontech_x86/rpm	krb5-devel-1.18.2-5.uel20.x86_64.rpm
libcgroup.so.1	os_patch_all/os/uniontech_x86/rpm	libcgroup-0.42.2-1.uel20.x86_64.rpm
libcgroup.so.1.0.42	os_patch_all/os/uniontech_x86/rpm	libcgroup-0.42.2-1.uel20.x86_64.rpm
pam_cgroup.so	os_patch_all/os/uniontech_x86/rpm	libcgroup-0.42.2-1.uel20.x86_64.rpm
libffi.so	os_patch_all/os/uniontech_x86/rpm	libffi-devel-3.3-8.uel20.x86_64.rpm
libstdc++.so	os_patch_all/os/uniontech_x86/rpm	libstdc++-devel-7.3.0-20211123.43.uel20.x86_64.rpm
libverto-glib.so	os_patch_all/os/uniontech_x86/rpm	libverto-devel-0.3.1-2.uel20.x86_64.rpm
libverto-libev.so	os_patch_all/os/uniontech_x86/rpm	libverto-devel-0.3.1-2.uel20.x86_64.rpm
libverto-libevent.so	os_patch_all/os/uniontech_x86/rpm	libverto-devel-0.3.1-2.uel20.x86_64.rpm
libverto.so	os_patch_all/os/uniontech_x86/rpm	libverto-devel-0.3.1-2.uel20.x86_64.rpm
libnfsidmap.so.1	os_patch_all/os/uniontech_x86/rpm	nfs-utils-2.5.1-4.uel20.x86_64.rpm
libnfsidmap.so.1.0.0	os_patch_all/os/uniontech_x86/rpm	nfs-utils-2.5.1-4.uel20.x86_64.rpm
nsswitch.so	os_patch_all/os/uniontech_x86/rpm	nfs-utils-2.5.1-4.uel20.x86_64.rpm
regex.so	os_patch_all/os/uniontech_x86/rpm	nfs-utils-2.5.1-4.uel20.x86_64.rpm

Software	File Path	File Name
static.so	os_patch_all/os/uniontech_x86/rpm	nfs-utils-2.5.1-4.uel20.x86_64.rpm
umich_ldap.so	os_patch_all/os/uniontech_x86/rpm	nfs-utils-2.5.1-4.uel20.x86_64.rpm
libcrypto.so	os_patch_all/os/uniontech_x86/rpm	openssl-devel-1.1.1f-13.uel20.x86_64.rpm
libssl.so	os_patch_all/os/uniontech_x86/rpm	openssl-devel-1.1.1f-13.uel20.x86_64.rpm
libpython2.7.so	os_patch_all/os/uniontech_x86/rpm	python2-devel-2.7.18-1.uel20.x86_64.rpm
_ctypes_test.so	os_patch_all/os/uniontech_x86/rpm	python2-devel-2.7.18-1.uel20.x86_64.rpm
libpython3.7m.so	os_patch_all/os/uniontech_x86/rpm	python3-devel-3.7.9-18.uel20.x86_64.rpm
_ctypes_test.cpython-37m-x86_64-linux-gnu.so	os_patch_all/os/uniontech_x86/rpm	python3-devel-3.7.9-18.uel20.x86_64.rpm
_testbuffer.cpython-37m-x86_64-linux-gnu.so	os_patch_all/os/uniontech_x86/rpm	python3-devel-3.7.9-18.uel20.x86_64.rpm
_testcapi.cpython-37m-x86_64-linux-gnu.so	os_patch_all/os/uniontech_x86/rpm	python3-devel-3.7.9-18.uel20.x86_64.rpm
_testimportmultiple.cpython-37m-x86_64-linux-gnu.so	os_patch_all/os/uniontech_x86/rpm	python3-devel-3.7.9-18.uel20.x86_64.rpm
_tkinter.cpython-37m-x86_64-linux-gnu.so	os_patch_all/os/uniontech_x86/rpm	python3-devel-3.7.9-18.uel20.x86_64.rpm
_xxtestfuzz.cpython-37m-x86_64-linux-gnu.so	os_patch_all/os/uniontech_x86/rpm	python3-devel-3.7.9-18.uel20.x86_64.rpm

Arm-based UnionTech OSs

No open source software is included.

A.4 Installing JRE

The following uses Kylin JRE as an example. For details about how to configure YUM, see **Configuring a Yum Repository**.

Step 1 Run the following command as the **root** user to install OpenJdk:

yum install java-1.8.*-openjdk -y

Step 2 Run the following command to check whether Java is correctly configured:

java -version

If the following information is displayed and the correct version number is displayed, the installation is complete.

```
openjdk version "1.8.0_*"
OpenJDK Runtime Environment Bisheng (build 1.8.*)
OpenJDK 64-Bit Server VM Bisheng (build 25.*, mixed mode)
```

----End

A.5 Installing Python 3

Prerequisites

You have configured the Yum repository. For details about how to configure the Yum repository, see **Configuring a Yum Repository**.

Procedure

Step 1 Log in to the node where the instance is deployed as the **root** user.

Step 2 Run the following command to install Python3:

yum -y install python3

----End

A.6 Installing Python 3 on the Host

Prerequisites

You have configured the Yum repository. For details about how to configure the Yum repository, see **Configuring a Yum Repository**.

Install Python 3 before adding a host. If the Python 3 version does not pass the host standardization check, delete the host, install Python 3, and then add the host.

Procedure

Step 1 Run the following command to install the dependency environment:

yum -y install make rng-tools perl libffi-devel sqlite-devel openssl-devel python3-devel gcc-c++ libcgroup libcgroup-tools

systemctl restart rngd

systemctl restart cgconfig

Step 2 Install Python 3 of the required version. You can use the Yum repository or Python 3 installation package to install Python 3. The Python3.7.9 installation is used as an example.

- Installation using the Yum repository: Check whether the version of the Python 3 installation package provided in the Yum repository is Python 3.7.9. If yes, run the **yum -y install python3.7** command to install.
- Installation using the Python 3 installation package: Run the following command (The **Python3.7.9.tgz** installation package is used as an example):
  a. Run the **mkdir -p /usr/local/python3** command to create the python 3 folder. The **/usr/local/python3** folder is used as an example.
  b. Run the **tar -zxvf Python-3.7.9.tgz** command to decompress the **Python-3.7.9.tgz** installation package.
  c. Run the **cd Python-3.7.9** command to switch to the directory containing the decompressed files.
  d. Run the following command to perform compilation and installation:

     ./configure --prefix=/usr/local/python3 --enable-optimizations --enable-shared

     make -sj && make install

Step 3 Run the following command to modify the soft link (The Python 3 installation path **/usr/local/python3** is used as an example):

ln -sf /usr/local/python3/lib/libpython3.7m.so.1.0 /usr/lib64/libpython3.7m.so.1.0

ln -sf /usr/lib64/libpython3.7m.so.1.0 /usr/lib64/libpython3.7m.so

ln -sf /usr/local/python3/lib/libpython3.so /usr/lib64/libpython3.so

ln -s /usr/lib64/libffi.so.6 /usr/lib64/libffi.so.7

rm -f /usr/bin/pip /usr/bin/pip3 /usr/bin/python /usr/bin/python3

ln -s /usr/local/python3/bin/pip /usr/local/bin/pip

ln -s /usr/local/python3/bin/pip3 /usr/local/bin/pip3

ln -s /usr/local/python3/bin/pip3.7 /usr/local/bin/pip3.7

ln -s /usr/local/python3/bin/python3 /usr/bin/python3

ln -s /usr/bin/python3 /usr/bin/python

chmod -R 755 /usr/local/python3

chmod 755 -R /usr/local/lib64/python3.7

chmod 755 -R /usr/local/lib/python3.7

NOTICE

If the error message "ln: failed to create symbolic link 'xxx': File exists" is displayed, skip this step.

Step 4 Run the following command to check whether Python 3 is installed:

python --version

If **Python 3.7.9** is displayed in the command output, the installation is successful.

----End

A.7 Installing the Expect

Prerequisites

You need to configure the Yum repository for the Kylin environment. For details about how to configure the Yum repository, see **Configuring a Yum Repository**.

Procedure

Step 1 Run the following command to install Expect:

yum install expect -y

Step 2 Check whether the software is installed.

expect -v

----End

A.8 Installing net-tools

Prerequisites

You have configured the YUM source in the Kylin environment before installing the software. For details, see **Configuring a Yum Repository**.

Procedure

Step 1 Log in to a node as the **root** user.

Step 2 Run the following command to install the net-tools software:

yum install net-tools -y

Step 3 Check whether the software is installed.

ifconfig

----End

A.9 Mounting Disks

Scenarios

This section describes how to mount disks during GaussDB Management Platform (TPOPS) installation. You need to mount disks to all nodes to be installed.

Prerequisites

You have planned seven independent disks. For details about the size of each disk, see **Table 4-6** or **Table 4-8**.

This section uses **/opt/cloud** as an example to describe how to mount a disk. The procedure for mounting disks to other directories is similar.

Procedure

Step 1 Run the following command to query the disk information:

lsblk

Information similar to the following is displayed:

```
NAME             MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                8:0    0 557.9G  0 disk
├─sda1             8:1    0   600M  0 part /boot/efi
├─sda2             8:2    0     1G  0 part /boot
└─sda3             8:3    0 556.3G  0 part
  ├─euleros-root 253:0    0    70G  0 lvm  /
  ├─euleros-swap 253:1    0     4G  0 lvm
  └─euleros-home 253:12   0 482.3G  0 lvm  /home
sdb                8:16   0     7T  0 disk
```

The following uses **/dev/sdb** as an example to describe how to mount a disk.

Step 2 Run the following command to format the disk:

mkfs.ext4 /dev/sdb

Step 3 Run the following command to query the UUID of the disk partition to be mounted:

blkid /dev/sdb

Information similar to the following is displayed:

```
/dev/sdb: UUID="*" BLOCK_SIZE="4096" TYPE="Y"
```

Step 4 Run the following commands to open the **fstab** file and configure mounting parameters (*indicates the obtained UUID). The **/opt/cloud** directory is used as an example.

vi /etc/fstab

UUID=* /opt/cloud Y defaults 1 2

NOTE

In the preceding command, the first parameter indicates the obtained UUID, the second parameter indicates the path to be mounted, and the third parameter indicates the **type** value obtained by running the **blkid** command. **defaults** indicates the mounting option, and the last two digits indicate the backup and detection options.

Step 5 Press **Esc** and run the following command to save the modification and exit:

:wq!

Step 6 Run the following command to make the configuration take effect:

mount -a

----End

A.10 Dependent Python Library Versions

setuptools-57.0.0

greenlet-1.1.0

setuptools-scm-3.3.3

typing-extensions-3.10.0.0

zipp-3.4.1

importlib-metadata-4.6.0

SQLAlchemy-1.4.20

PyYAML-5.4.1

pyrsistent-0.18.0

attrs-21.2.0

jsonschema-3.2.0

kafka-python-2.0.2

dnspython-1.16.0

eventlet-0.31.0

protobuf-3.17.3

esdk-obs-python-3.20.11

portalocker-2.3.0

concurrent-log-handler-0.9.19

psutil-5.8.0

pexpect-4.8.0

pycparser-2.20

cffi-1.14.5

bcrypt-3.2.0

six-1.16.0

ptyprocess-0.7.0

PyNaCl-1.3.0

cryptography-3.3.1

paramiko-2.7.2

future-0.18.2

pkgconfig-1.5.1

lz4-3.1.3

A.11 Setting the root User for Logging In to a Management Plane Node Without a Password

Scenarios

This section describes how to use a key to log in to a node on the management plane as the **root** user.

Prerequisites

Customers can use a password to log in to a node on the management plane as the **root** user and agree to generate and deliver a private key.

Procedure

Step 1 Log in to any node on the management plane as the **root** user. If there are multiple nodes, perform the following operations on each node:

Step 2 Run the following command to open the specified file:

vi /etc/ssh/sshd_config

Step 3 Search for the following two lines:

```
PubkeyAuthentication yes
AuthorizedKeysFile    .ssh/authorized_keys
```

- If the configurations are not found or inconsistent with the preceding information, add or modify the configurations, press **Esc**, and run the **:wq!** command to save the modification and exit.

  Run the following command to restart the SSHD service:

  systemctl restart sshd
- If the configurations are found and consistent with the preceding information, go to the next step.

Step 4 Run the following command to go to the specified directory:

cd /root/.ssh

Step 5 Run the following command without entering any value and press **Enter** to generate a key pair:

ssh-keygen

```
[root@localhost .ssh]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:Xm8R8AOKXtzsAS9n27pIEu2USfetvEhSYTT+T+5HM1s root@localhost.localdomain
The key's randomart image is:
+---[RSA 3072]----+
|        . =      |
|       o O =     |
|      . * % +    |
|     . + X B +   |
|      o S = = o  |
|       = o + * oE|
|      . = o = o.=|
|       o + + o ..|
|        . o . .. |
+----[SHA256]-----+
```

Step 6 Run the following commands in the current directory to generate an authorized public key file:

touch authorized_keys

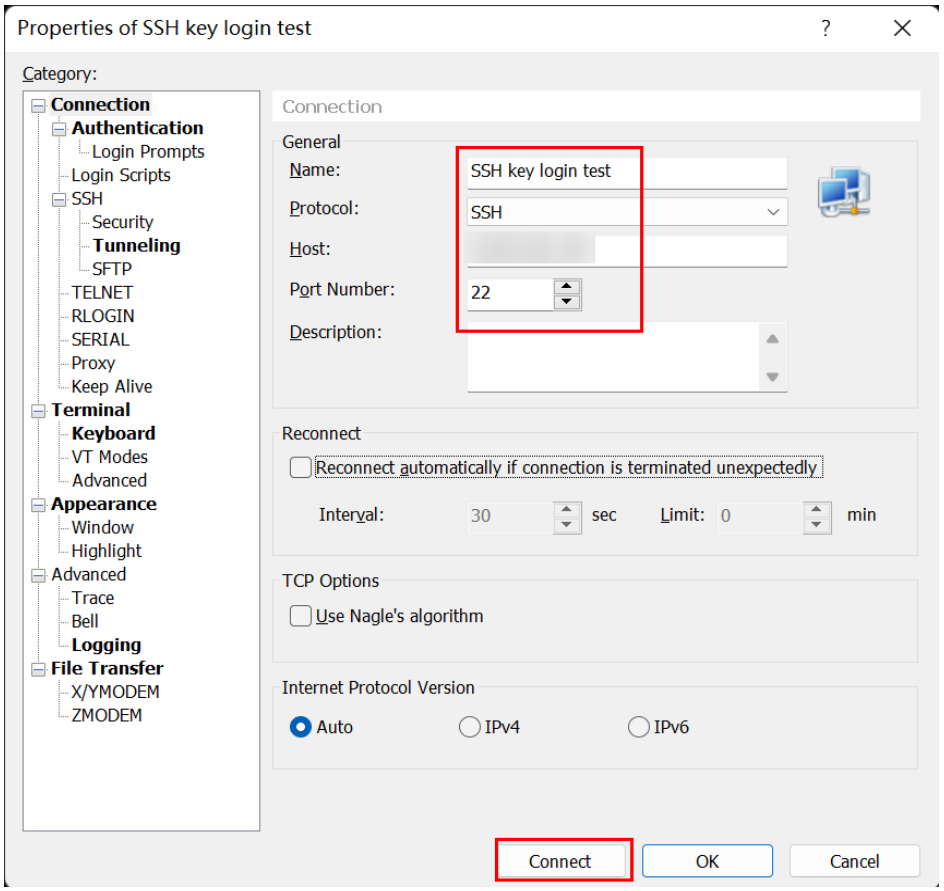
chmod 600 authorized_keys

cat id_rsa.pub >> authorized_keys

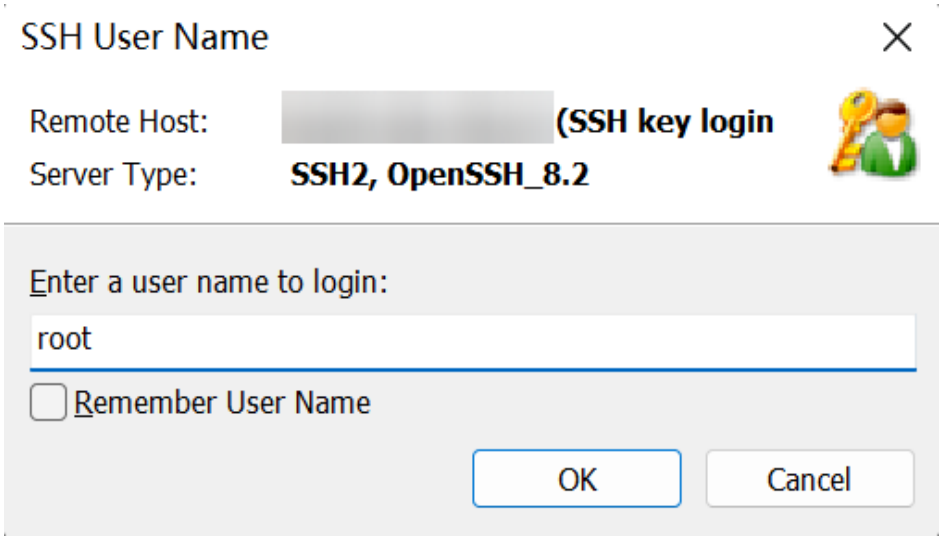
Step 7 Download the SSH private key file **id_rsa** generated in the current directory to a local PC.

Step 8 The following uses Xshell as an example to describe how to configure a key for logging in to a node on the management plane.

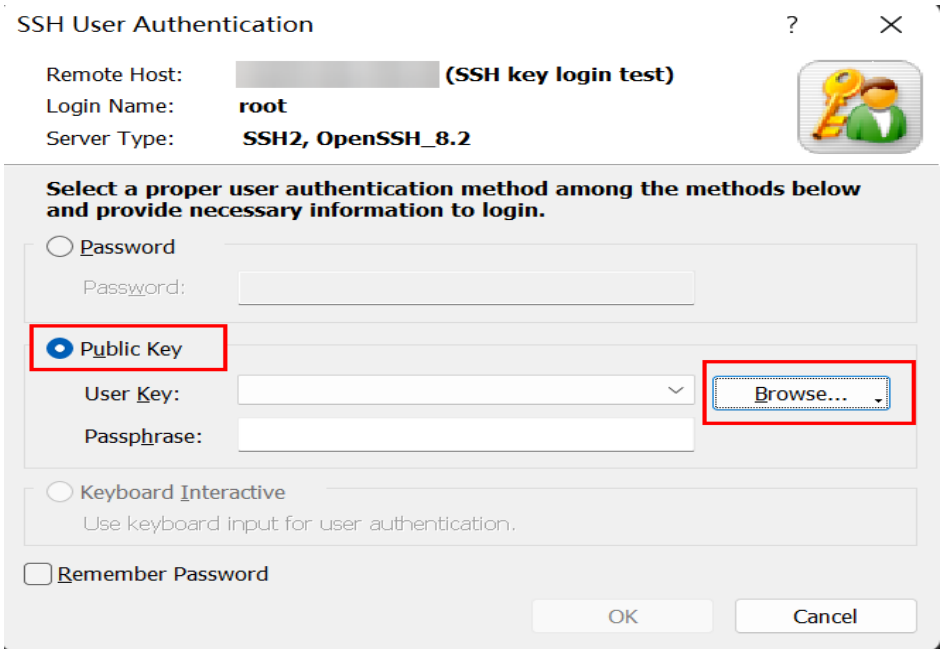
1. Create a session, enter the name, host, and port number, and click **Connect**.

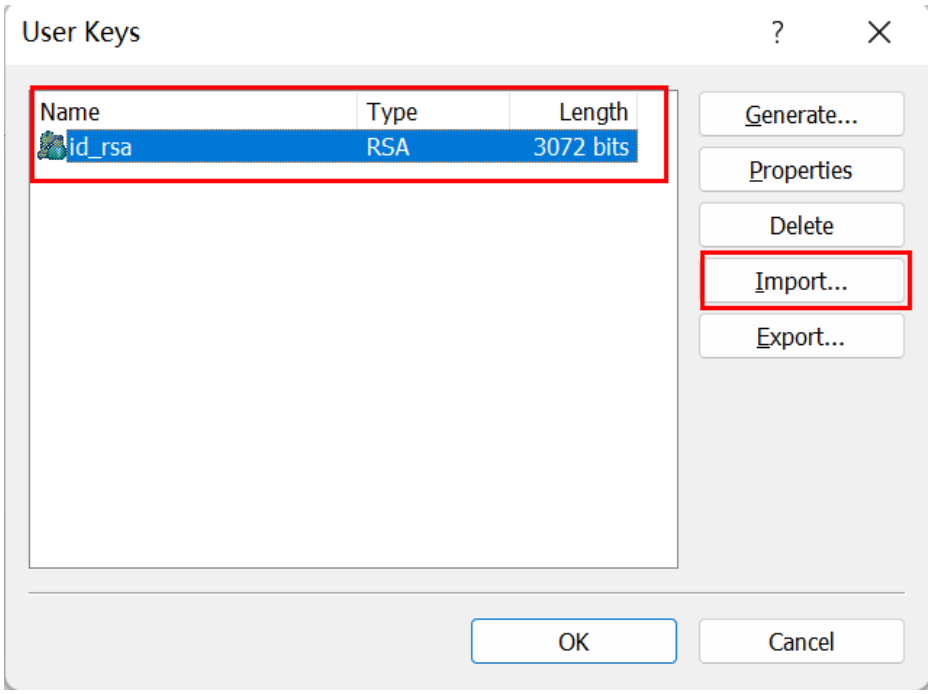


2. Enter the username **root** and click **OK**.



3. Select **Public Key**, click **Browse** > **Import**, and import the **id_rsa private** key file downloaded to the local PC.





4. Click **OK**. to log in to the node as the **root** user.

----End

A.12 Setting Mutual Trust Between Nodes on the Management Plane

Scenarios

Configure mutual trust between **root** accounts of the management plane nodes.

Prerequisites

Customers can log in to the management plane nodes as the **root** user and agree to configure mutual trust between nodes.

Procedure

Step 1 Log in to any node on the management plane as the **root** user. If there are multiple nodes, perform the following operations on each node:

Step 2 Run the following command to open the specified file:

vi /etc/ssh/sshd_config

Step 3 Search for the following two lines:

```
PubkeyAuthentication yes
AuthorizedKeysFile    .ssh/authorized_keys
```

- If the configurations are not found or inconsistent with the preceding information, add or modify the configurations, press **Esc**, and run the **:wq!** command to save the modification and exit.

  Run the following command to restart the SSHD service:

  systemctl restart sshd

- If the configurations are found and consistent with the preceding information, go to the next step.

Step 4 Run the following commands to check whether the SSH private key **id_rsa** and public key file **id_rsa.pub** are generated in the **/root/.ssh** directory:

cd /root/.ssh

ls id_rsa

ls id_rsa.pub

- If they are not generated, run the following command without entering any value and press **Enter** repeatedly to generate **id_rsa** and **id_rsa.pub**:

  ssh-keygen

```
[root@localhost .ssh]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:Xm8R8AOKXtzsAS9n27pIEu2USfetvEhSYTT+T+5HM1s root@localhost.localdomain
The key's randomart image is:
+---[RSA 3072]----+
|         . =     |
|       o O =     |
|      . * % +    |
|     . + X B +   |
|      o S = = o  |
|       = o + * oE|
|      . = o = o.=|
|       o + + o ..|
|        . o . .. |
+----[SHA256]-----+
```

- If they are generated, go to the next step.

Step 5 Run the following commands in the directory to generate an authorized public key file:

touch authorized_keys

chmod 600 authorized_keys

Step 6 Perform the following operations to add the public key of the current node to the authorized public key file of the current node:

1. Run the following command to display the public key information:

   cat id_rsa.pub

2. Copy the public key information.
3. Run the following command to write the copied public key information to the **authorized_keys** file and save the file:

   vi authorized_keys

Step 7 Log in to the other two nodes on the management plane and run the following commands for them:

cd /root/.ssh

touch authorized_keys

chmod 600 authorized_keys

vi authorized_keys

Write the public key information copied in **Step 6** to the **authorized_keys** file on the nodes and save the file.

Step 8 Repeat **Step 1** to **Step 7** to add the public key of each node to the authorized public key files of the three nodes.

----End

A.13 Scaling Out Local SSD Disks

Scenarios

This section describes scale-out methods by adding a new data disk to a host with instances.

Procedure

Step 1 Log in to the host as user **root**.

Step 2 Run the **lsblk** command to ensure that the local SSD disk to be scaled out is identified by the host.

Assume that a sandbox instance exists on the host and the data disk VDC is added for scale-out. In the following command output, the data disk scale-out path is **/var/chroot/var/lib/engine/data1**.


```
[root@host-192-168-1-106 ~]# lsblk
NAME                    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
vda                     252:0    0   40G  0 disk
├─vda1                  252:1    0    1G  0 part /boot
└─vda2                  252:2    0   39G  0 part
  ├─klas-root           253:0    0   35G  0 lvm  /
  └─klas-swap           253:1    0    4G  0 lvm  [SWAP]
vdb                     252:16   0  300G  0 disk
├─gaussdbvg-mydata1     253:2    0  200G  0 lvm  /var/chroot/var/lib/engine/data1
├─gaussdbvg-backupdata  253:3    0   20G  0 lvm  /var/chroot/var/lib/log/backup
└─gaussdbvg-etcddata    253:4    0   64G  0 lvm  /var/chroot/usr/local/etcd
vdc                     252:32   0  100G  0 disk
```

Step 3 Run the following command to create the **extend.sh** script and open it.

vi extend.sh

Step 4 Copy the following code to the **extend.sh** script.

```
#! /bin/bash
new_disks=$1
target=$2
extend_size=$3
# new_disk make sure the disk is valid. It can be a list(when split by ',').
# target eg:/var/chroot/var/lib/engine/data1

# show lsblk info, check the disk is managed by lvm or lvm2.
read vg_name disk_type mount_point <<< $(echo $(lsblk -l|grep $target|awk {'print($ 1, $6,$7)'}))
echo "lsblk info: $vg_name, $disk_type, $mount_point"

if [[ $disk_type == lvm* ]]
then
 echo "Type: $disk_type,  ..."
else
 echo "It's not a lvm filesystem"
 exit 1
fi

# split the new_disks
disk_list=(`echo $new_disks |tr ',' ' '`)

# ready to create a pv one by one.
for new_disk in ${disk_list[@]}
do
  # cmd : pvcreate /dev/sdc
  pvcreate $new_disk
done

# pv without vg_group
echo "----------------------"
echo "$(pvs)"
echo "----------------------"

# vg extend volume one by one
for new_disk in ${disk_list[@]}
do
  # cmd : vgextend gaussdbvg /dev/sdc
  vgextend gaussdbvg $new_disk
done

# figure out the target mount_point's FileSystem.
read file_system <<< $(df -h |grep $target |awk {'print($1)'})
echo "The $target 's mapper is: $file_system ."

# cmd : 100% vg free; lvextend -l  +100%FREE /dev/mapper/gaussdbvg-mydata1
#   lvextend -l +100%FREE $file_system , should be sum(new_disks)
for new_disk in ${disk_list[@]}
do
   # cmd : vgextend gaussdbvg /dev/sdc
```



```
    read size <<< $(echo $(fdisk -l|grep -w $new_disk |awk {'print($3)'}))
    lvextend -L +$extend_size""G $file_system
done
resize2fs $file_system
```

Step 5 Press **Esc** and run the following command to save the change and exit the **extend.sh** script.

:wq!

Step 6 Run the following command to execute the **extend.sh** script to scale out disk VDC by 100 GB.

chmod +x extend.sh; ./extend.sh /dev/vdc /var/chroot/var/lib/engine/data1 100;

The script supports the scale-out of multiple disks. For example, run the following command to scale out disks VDC and VDD.

chmod +x extend.sh; ./extend.sh /dev/vdc,/dev/vdd /var/chroot/var/lib/engine/data1 100;

If there are multiple data disk scale-out paths, allocate the scale-out evenly.

Step 7 Run the **lsblk** command to check the scale-out results.

```
[root@host-192-168-1-106 ~]# lsblk
NAME                   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
vda                    252:0    0   40G  0 disk
├─vda1                 252:1    0    1G  0 part /boot
└─vda2                 252:2    0   39G  0 part
  ├─klas-root          253:0    0   35G  0 lvm  /
  └─klas-swap          253:1    0    4G  0 lvm  [SWAP]
vdb                    252:16   0  300G  0 disk
├─gaussdbvg-mydata1    253:2    0  300G  0 lvm  /var/chroot/var/lib/engine/data1
├─gaussdbvg-backupdata 253:3    0   20G  0 lvm  /var/chroot/var/lib/log/backup
└─gaussdbvg-etcddata   253:4    0   64G  0 lvm  /var/chroot/usr/local/etcd
vdc                    252:32   0  100G  0 disk
└─gaussdbvg-mydata1    253:2    0  300G  0 lvm  /var/chroot/var/lib/engine/data1
```

----End

A.14 Performing Uninstallation After the docker-service Directory Is Deleted

Scenarios

If the **docker-service** directory is deleted, perform the following operations to uninstall GaussDB Management Platform (TPOPS):

Procedure

Step 1 Log in to any GaussDB Management Platform (TPOPS) node as the **root** user.

Step 2 Upload **docker-service** corresponding to the version to be uninstalled to the **/data** directory.

Step 3 Run the following command to decompress the software package:

cd /data

tar -xvf DBS-docker-service_*_all.tar.gz -C /data

Step 4 Run the following command to modify the configuration file:

vi /data/docker-service/config/user_edit_file.conf

The file must be the same as that configured during installation. For details about how to modify the configurations during installation, see **Step 3**.

NOTE

- You can run the **ps -ef | grep sftpd | grep -v grep** command on the three nodes to query the SFTP installation node. If any command output is displayed on a node, the node is the SFTP installation node.
- You can run the **ps -ef | grep influxd | grep -v grep** command on the three nodes to query the InfluxDB installation node. If any command output is displayed on a node, the node is the InfluxDB installation node.

Step 5 Run the following command to render parameters:

sh /data/docker-service/action/optionAction/render_args_new.sh install

Step 6 Run the following command to copy **docker-service** from the target node to the other two nodes:

scp -r /data/docker-service root@{*ip*}:/data

{*ip*} indicates the IP addresses of the other two management plane nodes.

Step 7 Log in to the other two nodes as the **root** user and run the following commands to modify the **render_args.yml** configuration file:

sed -i -e '/local_ip:/d' -e '/local_100:/d' -e '/static_route_list:/d' -e '/openapi_float_url_for_open_gauss:/d' /data/docker-service/package/cloud/common/render/render_args.yml

echo "local_ip: *node2_ip*" >> render_args.yml

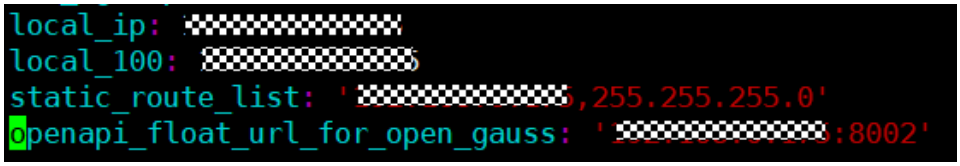
echo "local_100: *node2_ip2*" >> render_args.yml

echo "static_route_list: '*node2_ip*,255.255.255.0'" >> render_args.yml

echo "openapi_float_url_for_open_gauss: '*node2_ip*:8002'" >> render_args.yml

Assume that **node1_ip** is the execution node, and **node2_ip** and **node2_ip2** are used as an example. **node2_ip** and **node2_ip2** are the IP addresses of the remote nodes, which correspond to **node2_ip** and **node2_ip2** configured in **user_edit_file.conf**. The same applies to **node3_ip**.

The following figure shows the **render_args.yml** file after modification.



The hidden part is the IP addresses of the related nodes.

Step 8 Log in to the node where the uninstallation command is to be executed (the node where the **docker-service** directory is decompressed) as the **root** user and run the following commands to uninstall the lightweight management plane:

cd /data/docker-service

sh appctl.sh uninstall_all

sh appctl.sh cleanup_all

----End

A.15 Contacting Technical Support

You can provide your feedback to us in either of the following ways:

- Call the hotline of the support website.
- Go to the **Products Support** page of the support website.

