

# **Distributed Transactional Database**

## **HotDB Server -**

## **Cross-IDC Disaster Recovery**

## **Function Specification**

Version No.: V2.5.6

Shanghai Hotpu Networks Technology Co., Ltd.

Nov. 2020

**All rights reserved!** Without the written consent of the company, no unit or individual is allowed to extract and copy the contents hereof in part or in whole, and disseminate them in any form.

### Trademark statement

HotDB trademark of Hotpu is the registered trademark or trademark of our company or its affiliated company which is legally protected. Any infringement shall be investigated. Without the written permission of the company or the trademark owner, no unit and individual is allowed to use, copy, modify, disseminate, transcribe or bundling sell any part of this trademark in any form or for any reason. Any infringement of the trademark right of the company shall be investigated for legal liability.

### Cautions

The products, services or features you purchase shall be subject to the commercial contracts and terms of Shanghai Hotpu Networks Technology Co., Ltd. The final interpretation right of the products, services or features in this document shall belong to Shanghai Hotpu Networks Technology Co., Ltd.

Shanghai Hotpu Networks Technology Co., Ltd.

Address: 603-605, Block A, SIM Technology building, No. 633, Jinzhong Road, Changning District, Shanghai

Postcode: 200050

Website: [www.hotdb.com](http://www.hotdb.com)

Service email: [service\\_hotdb@hotdb.cm](mailto:service_hotdb@hotdb.cm)

Service Tel: 021-5218 0789

## Contents

1. Basic Information.....	6
1.1. Background .....	6
1.2. Disaster Recovery Target .....	6
1.3. Glossary .....	7
2. Fundamentals and architecture of DR mode.....	9
2.1. Fundamentals .....	9
2.2. Architecture of data transfer.....	9
3. Installation and deployment.....	11
3.1. Environment requirements.....	11
3.2. Function premise.....	11
3.3. Deployment architecture.....	12
3.4. Automatic deployment.....	14
3.4.1. Deploy a new set of disaster recovery environment .....	14
3.4.2. Disaster recovery environment deployment based on a running cluster ...	25
3.5. Manual deployment .....	26
3.5.1. Deploy a new set of disaster recovery environment .....	26
3.5.1.1. Parameter configuration modification .....	26
3.5.1.2. Set up ConfigDB replication relations .....	27
3.5.1.3. Set up data source replication relations .....	31
3.5.1.4. Add compute node clusters to management platform.....	32
3.5.1.5. Add data sources for IDCs .....	36
3.5.2. Disaster recovery environment deployment based on a running cluster ...	36
3.5.2.1. Parameter configuration modification .....	36
3.5.2.2. Set up ConfigDB replication relations .....	36
3.5.2.3. Set up data source replication relations .....	40
3.5.2.4. Add compute node clusters to management platform.....	40
3.5.2.5. Add data source for DR centers .....	41
4. Description of functions related to management platform .....	42
4.1. Compute node cluster selection .....	42
4.2. Deployment environment examination.....	44
4.3. Configuration .....	46
4.3.1. Node management .....	46
4.3.1.1. Add node .....	48
4.3.1.2. Master-slave setup .....	51
4.3.1.3. Switching rules.....	53

4.3.2. Configuration checking.....	55
4.3.3. Compute node parameter configuration.....	56
4.4. Monitoring .....	63
4.4.1. Logic topological graph .....	63
4.4.2. Physical topological graph.....	73
4.4.3. Other monitoring items .....	73
4.5. Management.....	77
4.5.1. Data backup .....	77
4.5.2. Data source migration.....	78
4.6. Detect .....	80
4.6.1. Master/slave data consistency detection .....	80
4.7. Event .....	82
4.7.1. Notification strategy.....	82
4.7.2. Others .....	85
4.8. Others .....	86
5. Descriptions of compute-node-related functions .....	90
5.1. Master center.....	90
5.2. DR center .....	90
6. Operation and maintenance management .....	93
6.1. Premise explanation .....	93
6.2. Failure only inside the master center .....	96
6.2.1. Compute node .....	96
6.2.1.1. Manual switching.....	96
6.2.1.2. Failover .....	96
6.2.2. ConfigDB .....	97
6.2.2.1. Failover .....	97
6.2.3. Data source.....	97
6.2.3.1. Manual switching.....	97
6.2.3.2. Failover .....	98
6.3. Failure only inside the DR center .....	99
6.3.1. Compute node .....	99
6.3.2. ConfigDB .....	100
6.3.2.1. Failover .....	100
6.3.3. Data source.....	100
6.3.3.1. Failover .....	100
6.4. IDC Failure .....	101

6.4.1. Master center fails and DR center is switched as the active IDC .....	101
6.4.1.1. Switch Pre-detection .....	101
6.4.1.2. Switch Process .....	102
6.4.2. IDC failure repair and failover.....	103
6.4.2.1. Scenario One: Abandon the original master center.....	103
6.4.2.2. Scenario Two: Make the original master center as the standby IDC	107
6.4.2.3. Scenario Three: Make the original master center as the active IDC.	114
6.4.2.4. Scenario Four: Make the original master center as the active IDC without deleting cluster.....	116
6.4.2.5. Special Instructions.....	123
7. Note.....	126
7.1.1. Periodic Plans.....	126
7.1.2. Email Alert.....	126
7.1.3. Environment Examination .....	127
7.1.4. Data Backup.....	128
7.1.5. Service Restarted .....	128
7.1.6. SSH of Servers.....	128
7.1.7. Master/Slave Data Consistency Detection.....	129
7.1.8. Service Access .....	129
7.1.9. Setup Replication.....	129
7.1.10. Important but temporarily unsupported functions .....	130
8. Reference Instructions .....	131
8.1. Compute node logs .....	131
8.2. Recommended steps for replication setup .....	134
8.2.1. Prerequisites .....	134
8.2.2. Set up replication relations when GTID is enabled .....	134
8.2.3. Set up replication relations when GTID is disabled .....	137
8.3. Under what circumstances data in the original IDC must be abandoned .....	138

# 1. Basic Information

Based on the Distributed Transactional Database Product HotDB Server - v2.5.6, this manual masterly describes the basic function usage and operation process of the cross-IDC disaster recovery scheme of HotDB Server based on the MySQL native replication function, for the user's reference and learning.

This document focuses more on the functions related to compute nodes and the management platform. If you need to know how to use the compute nodes and the management platform, please refer to the [Distributed Transactional Database HotDB Server - V2.5.3 \[Management Platform\] Function Manual](#) and the [Distributed Transactional Database HotDB Server – V2.5.3 \[Standard\]Function Manual](#).

## 1.1. Background

Data as the most important means of production, the loss and non-restorable of data is often fatal to enterprises. The distributed transactional database product has a complete set of cross-IDC disaster recovery scheme which can provide higher availability and security for database services. Therefore, HotDB Server provides a solution which can solve the HotDB Server cross-IDC disaster recovery difficulty based on MySQL native replication functions to realize the cross-IDC data synchronization and solve the problem of distributed transactional database service disaster recovery across IDCs.

## 1.2. Disaster Recovery Target

It supports synchronous data transmission across IDCs, and ensures that when the distributed transactional database service in the active center fails, it can be switched to the standby center; when the data source MySQL starts the semi-synchronous replication, the RPO can be guaranteed to be 0, and the TPS of a single IUD SQL (take the standard performance test SQL as standard) can be up to more than 100,000.

## 1.3. Glossary

Words	Descriptions
IDC Type	<p>The IDC type includes master center and DR (disaster recovery) center and is only used to identify and distinguish the two IDCs, and does not change with the service status of IDCs. In the process of using HotDB Server products, you are able to distinguish between the master center and the DR center on the compute node cluster management page.</p>
IDC Status	<p>The IDC status consists of the current active center and the current standby center, which is determined according to whether the current active compute node in the IDC provides services (3323 service port by default). The IDC in which the current master compute node provides services is the current active center; the standby IDC that provides high-availability service switching at the IDC level with the current active center is the current standby center.</p>
Disaster Recovery Mode (DR Mode)	<p>The components required for the operation of a compute node cluster with a disaster recovery relation are coordinated and deployed in the two IDCs. This cluster is called a cluster with DR mode enabled.</p>
Single-IDC Mode	<p>A compute node cluster without DR mode enabled is a single-IDC-mode cluster.</p>
Disaster Recovery	<p>Replication relation of master data source/ConfigDB between master center and DR center.</p>

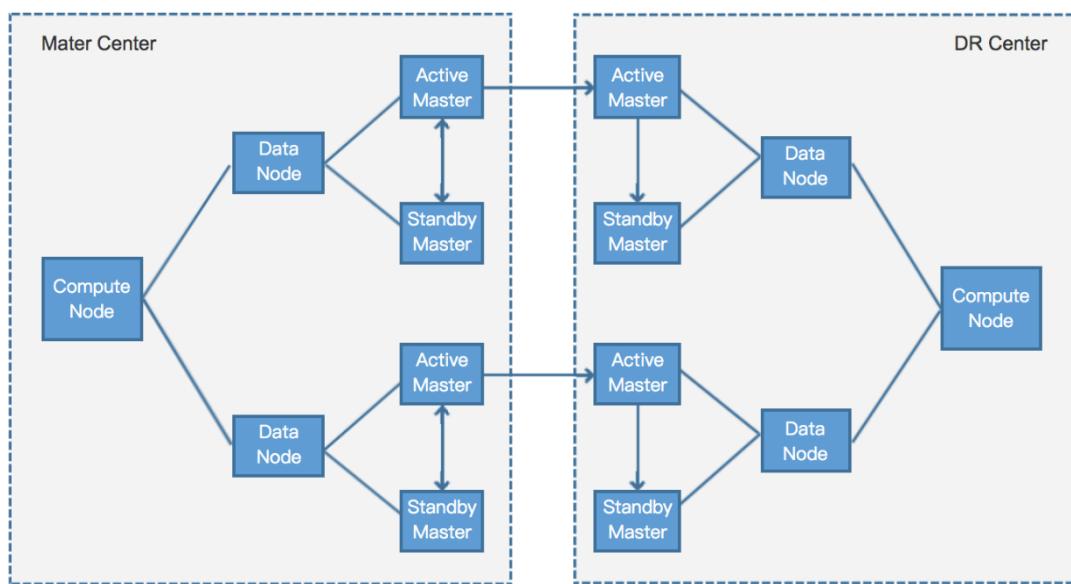
Data Replication	
Disaster Recovery  Data Replication  Status	Replication status of master data source/ConfigDB between master center and DR center.
Disaster Recovery  Data Replication  Latency	Replication latency of master data source/ConfigDB between master center and DR center.
Compute Node	I.e., the distributed transactional database service HotDB Server.
Data Source	Data source is a MySQL database service that stores data. A MySQL database can be used as a data source; one or more data sources with MySQL replication relations form a data node.
ConfigDB	ConfigDB is a MySQL database where the compute node configuration data are stored.
RPO	Recovery Point Objectives
RTO	Recovery Time Objectives

## 2. Fundamentals and architecture of DR mode

### 2.1. Fundamentals

With the data source instances under the data nodes as the unit, the DR mode will set up the master/slave replication relations of master data sources between the master center and the DR center, as well as synchronize business data of the data source of the master center with that of the corresponding data source of the DR center, based on the master/slave replication relations fundamentals of MySQL, as is the case with the ConfigDB. In the case of IDC-level failure in the master center, after manual instruction to switch the IDC, the compute node will automatically set up the replication relations of the new IDC according to the current data source, the replication status of the ConfigDB and the role configuration, and ensures the compute node service can continue.

### 2.2. Architecture of data transfer



- The user performs business SQL operation through the service port of the master compute node in the master center; the compute node parses the SQL statements and determines and distributes them to the master data sources corresponding to each data node for execution based on the sharding rules.

- At this time, through the MySQL master/slave replication function, users can synchronize data of each master data source in the master center to the corresponding master data source in the DR center.
- If there are other slave data sources in the DR center, the data will be synchronized to other slave data sources in the DR center based on the MySQL master-slave replication relations in the DR center, so as to complete the cross-IDC data synchronization.

## 3. Installation and deployment

### 3.1. Environment requirements

**Server:** physical machine is recommended for production environment; virtual machine can be used for test environment. If performance test is needed, physical machine must be used.

**Operating system:** 64 bit CentOS 6.x, 7.x or RHEL 6.x, 7.x are recommended; other operating systems are not supported.

**Software:** the running of HotDB Server (distributed transactional database) and HotDB Management (distributed transactional database platform) requires Java environment. JDK V1.7 is required when compute node version is lower than 2.5.6, and JDK V1.8 is required when compute node version is 2.5.6 and above.

**Network environment:** at least Gigabit-NIC is required in a single IDC, and the total network latency between servers across IDCs is less than 10ms. The bandwidth between IDCs is fully sufficient, with no bandwidth bottleneck and no packet loss.

#### Recommended configuration:

For the hardware environment configuration recommendation, please refer to the [Distributed Transactional Database HotDB Server - Hardware Configuration Recommendation](#).

For configuration requirements and recommendations of cluster operation environment, refer to the [Distributed Transactional Database HotDB Server - \[Cluster Configuration Reference\]](#).

### 3.2. Function premise

- The DR mode of the compute node cluster is enabled; the MGR mode of data sources or ConfigDBs are not supported temporarily.
- When the DR center is the current standby center, even if the ConfigDB and the data source replication mode is configured as master-maste, the actual replication relations can only be master-slave. When the DR center is switched to the current

active center, the compute node will build master-master replication relations for the ConfigDB and data source of the master-master mode.

- The performance throughput index is based on the internal standard performance test environment. In the case of asynchronous replication, the performance can exceed 100,000 TPS; for semi-synchronous replication, the expected performance loss is 1/3.
- It is recommended to use the automatic cluster deployment function to deploy the disaster recovery environment, or at least use the one click deployment installation script for single-component deployment before adding configuration on the management platform. Manual deployment is not recommended.
- If the compute node enables XA mode, the data source must enable semi-synchronous replication, otherwise data consistency cannot be guaranteed.

### 3.3. Deployment architecture

This section will take the following deployment architecture in master/slave mode as an example to explain the deployment related functions:



The actual connection information corresponding to each component is listed as follows:

IDC Type	Component Type	Role	Code Name	Connection Information
----------	----------------	------	-----------	------------------------

Master Center	Compute Node	Master	HotDB-01	192.168.220.186_3323_3325
		Slave	HotDB-02	192.168.220.187_3323_3325
	Data Source	Master	ds01	192.168.220.186_3307
		Standby Master	ds02	192.168.220.187_3307
	ConfigDB	Master	hc01	192.168.220.186_3306
		Standby Master	hc02	192.168.220.187_3306
DR Center	Compute Node	Master	HotDB-03	192.168.220.188_3323_3325
		Slave	HotDB-04	192.168.220.189_3323_3325
	Data Source	Master	ds03	192.168.220.188_3307
		Standby Master	ds04	192.168.220.189_3307 (Configured as master-master, however the actual replication relation is set up as master-slave)
	ConfigDB	Master	hc03	192.168.220.188_3306
		Standby Master	hc04	192.168.220.189_3306 (Configured as master-master, however the actual replication relation is set up as master-slave)

Note:

- When deploying in a real scenario, it is not recommended to deploy the compute nodes and data sources on the same server. This time is only for the convenience of subsequent explanations.
- In the real deployment environment, the number of data nodes and data sources should be planned according to the actual needs. For the convenience of explanation, only single data node deployment is made here.

## 3.4. Automatic deployment

Automatic deployment can be achieved through the cluster deployment function provided by the management platform. The cluster deployment function in the DR mode is performed orderly in the master center and the DR center of compute node clusters as. One deployment at least requires: compute node, ConfigDB, keepalived (required in master/slave node mode), and other components: data source, NTPD should be planned and installed according to the actual needs.

This section only describes in detail what should be paid attention to in the automatic deployment when the DR mode, not single-IDC mode, is enabled. For other automatic deployment related contents, please refer to the [Distributed Transactional Database HotDB Server - V2.5.3 \[Installation & Deployment\] Function Manual](#).

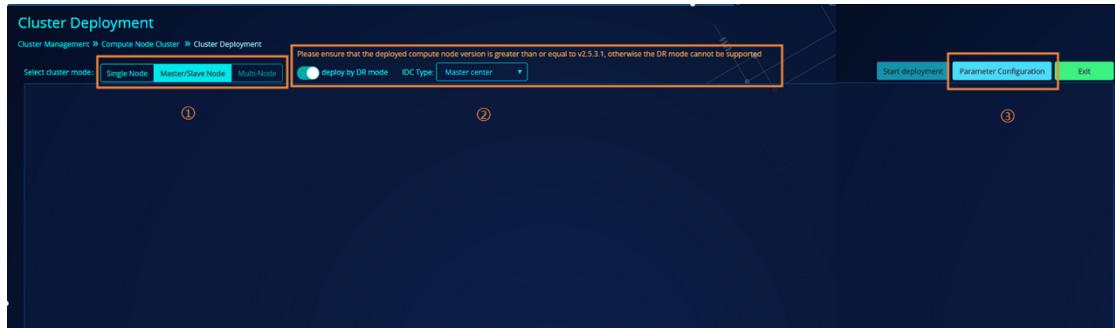
### 3.4.1. Deploy a new set of disaster recovery environment

#### 3.4.1.1. Cluster deployment – master center deployment

##### (1) Select the deployment mode:

1. Log in to the management platform as the management user, enter "Cluster Management - > Compute Node Cluster", and click the [Cluster Deployment] button to enter the cluster deployment function page.
2. Choose the compute node mode, click to enable "deploy by DR mode", choose "master center" in "IDC type" and click [Parameter Configuration] to enter the cluster deployment parameter configuration page.

In the DR mode, the compute node cluster mode is not supported at present, so it is not possible to choose multi-node as the cluster mode when "Deploy by DR mode" is enabled.



## (2) Parameter configuration:

The parameter configuration of the master center is basically the same as the single-IDC mode. The differences are listed as follows:

- The compute node v2.4 is not supported; the data source and ConfigDB MySQL v5.6 is not supported either;
- MGR is not supported by the data source and ConfigDB;
- Over “More” button of the data source and ConfigDB, GTID and semi-synchronous replication are enabled by default, and GTID is not allowed to be disabled;
- A new time zone setting module is added to configure the time zone of the operating system to ensure that the deployed component time zone is consistent with the operating system time zone. You can fill in the valid time zone under system directory /usr/share/zoneinfo/. The default time zone is Asia/Shanghai.



Therefore, the parameter configuration of the master center is as follows ([deployment architecture](#)):

**Cluster Deployment**

Cluster Management > Compute Node Cluster > Cluster Deployment

Compute Node: Master/Slave Mode Version: 2.5 Cluster Name: Group\_6534 Install NDB SQL Services  Connection Test

Master/Slave Type	*Compute Node Name	*Login Method	*Hostname	*Port	*SSH User	*SSH Password	*G1 garbage collector	*Heap Memory (G)	*Direct Memory (G)	*Service Port	*Management Port
Master	Compute Node-01	Password	192.168.240.155	22	root	.....	Off	4	24	3323	3325
Slave	Compute Node-02	Password	192.168.240.156	22	root	.....	Off	4	24	3323	3325

ConfigID: Master-M Versions: 5.7 Create High privilege Users  Connection Test

Master/Slave Type	*Instance port	*Login Method	*Hostname	*Port	*SSH Username	*SSH Password	Disk device name	Data Directory	Operation
Master	3306	Password	192.168.240.155	22	root	.....	Disk device name of data d	/root as default	
Standby Master	3306	Password	192.168.240.156	22	root	.....	Disk device name of data d	/data as default	

Keepalived

	*virtual_ipaddress	*virtual_router_id	*keepalived_net_interface_name
Master	192.168.240.106/24	106	eth0.1
Slave	192.168.240.106/24	106	eth0.1

Data Source Instance: 5.7 Number of Nodes: 1 Node Type: Master-M Generate Create High privilege Users  Import Connection Test

*Node Name	*Data Source Type	*Instance port	*Login Method	*Hostname	*Port	*SSH User	*SSH Password	*Install backup program	*Associated management platform IP	*Operation
dn_01	Active Master	3310	Password	192.168.240.155	22	root	.....	On	192.168.240.145	
dn_01	Standby M	3310	Password	192.168.240.156	22	root	.....	On	192.168.240.145	

Time Synchronization

Hostname	Install NTP Server	*Time synchronization Address
Compute Node (Master) Automatically obtain IP address of master NTP server or the first NTP server	<input checked="" type="checkbox"/>	182.92.12.11
Other Servers Enter your the servers except master Compute Node or the first Compute Node in cluster		

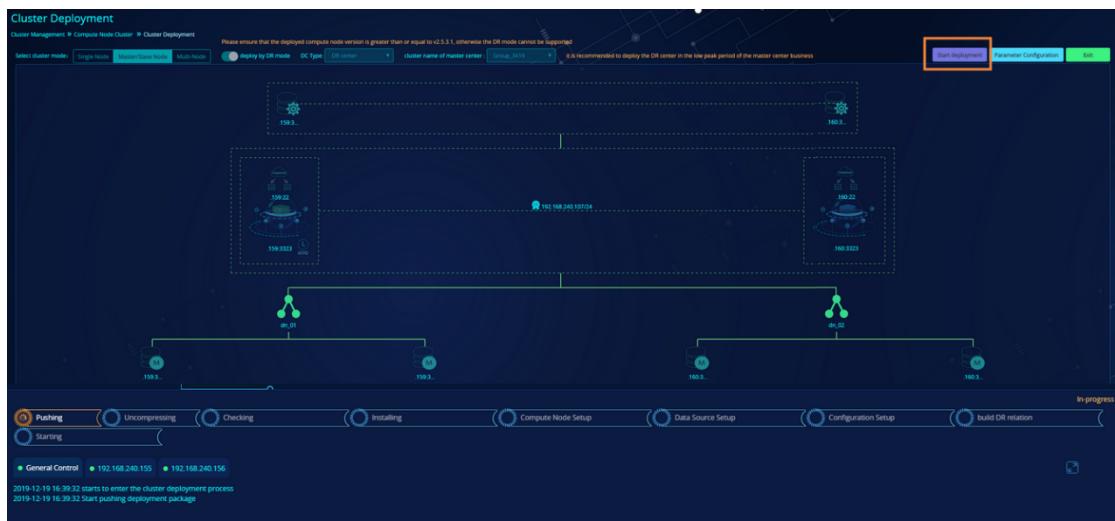
Time zone settings

Operating system time zone
Asia/Shanghai

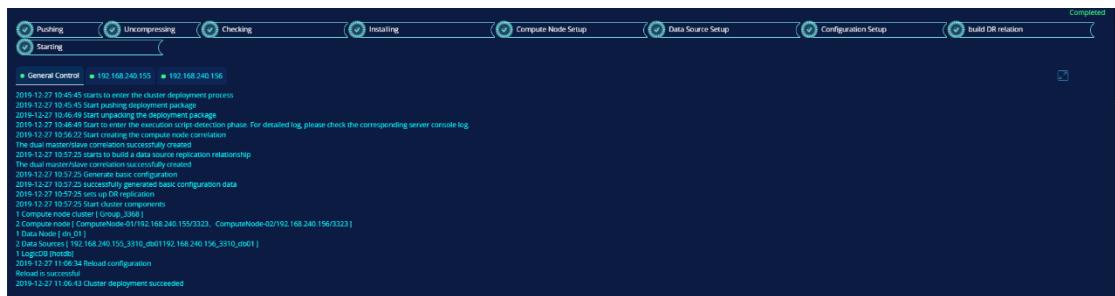
Only save Check and save Cancel

### (3) Start deployment:

- Click [Check and save] to verify the validity and integrity of the configuration parameters, and send a detection scripts to the target server to verify whether it meets the hardware requirements of the cluster deployment. If it does not meet the requirements, a pop-up prompt will appear. Before deployment, all clusters need to pass [Check and save] before installation.
- After check and accept of the management platform, click [Start deployment] to enter the installation process.



While a step "set up disaster recovery replication" is added in the installation and deployment process of management platform, there are no additional operations in this step when deploying the master center. Therefore, the installation process of the master center deployment is the same as that of a cluster deployment in single-IDC mode.



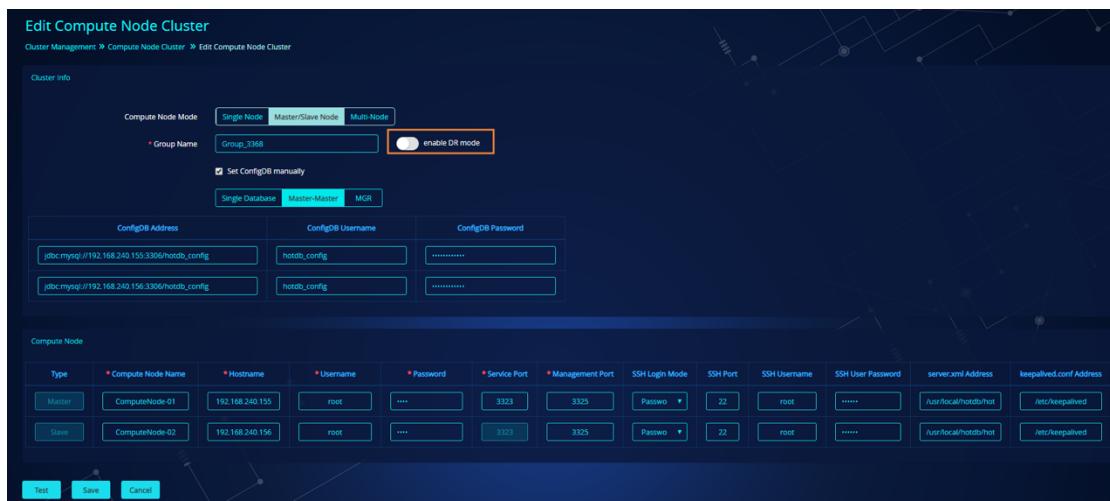
#### (4) Cluster management:

When the monitoring of cluster is enabled, a deployed master center with no DR center deployed or added can be regarded as a compute node cluster in single-IDC mode, and can be managed by single-IDC mode.

Compute Node Cluster														
Cluster Management > Compute Node Cluster														
Filter by Compute Node Cluster Name														
ID	Group Name	Cluster Mode	IDC Type	Component Name	IP Address	Service Port	Management Port	Type	Compute Node	High Available Components	ConfigDB	NTPD Time Service	Data Source Instance	Deployment Environment Grade
4	Group_1568	Master/Slave Node v.2.5.3.1	Master	ComputeNode-01	192.168.240.155	3323	3325	Master	Active	Active	Active	Active	Total number of instances: 2 Active: 2	Not Examination
				ComputeNode-02	192.168.240.156	3323	3325	Slave	Active	Active	Active	Active		Deployment Topology
1	Group_2685	Master/Slave Node v.2.5.3.1	Master center	ComputeNode-01	192.168.240.141	3323	3325	Master	Configured	Configured	Configured	Configured	Total number of instances: 4 Active: 4	Not Examination
				ComputeNode-02	192.168.240.142	3323	3325	Slave	Configured	Configured	Configured	Configured		Deployment Topology
2	Group_2685	Master/Slave Node v.2.5.3.1	DR center	ComputeNode-01	192.168.240.143	3323	3325	Master	Active	Active	Active	Active	Total number of instances: 4 Active: 4	Not Examination
				ComputeNode-02	192.168.240.144	3323	3325	Slave	Active	Active	Active	Active		Deployment Topology



That is, the DR mode of the cluster will be OFF at this time after the master center is deployed, which can be seen on the compute node details page of the IDC via cluster management page.



At this time, you can either deploy the corresponding DR center through the cluster deployment function, or after the DR center is deployed manually, click "Enable DR mode" on the Edit Compute Node Cluster page to add the DR center information to the cluster.

### 3.4.1.2. Cluster deployment – DR center deployment

#### (1) select the deployment mode:

1. After the master center is deployed, click the [Cluster Deployment] button again.
2. Similar to the master center deployment, after choosing the compute node mode, click to enable "Deploy by DR mode", choose "DR center" in "IDC type" and the corresponding "Cluster name of the master center", i.e., the master center cluster that is newly deployed.

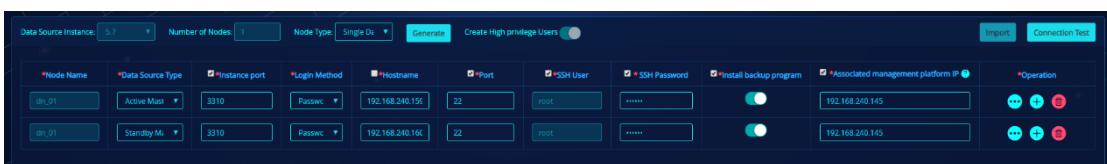
Note: it is recommended to deploy the DR center at the low peak of the master center business, otherwise it may affect the data migration time when setting up the disaster recovery replication.



## (2) Parameter configuration:

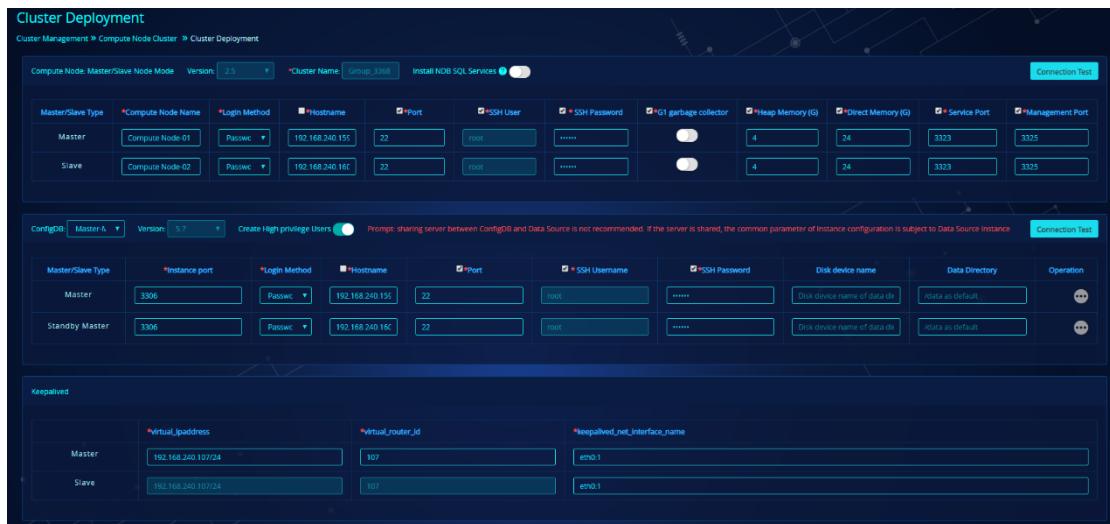
While the parameter configuration of the DR center is roughly the same as that of the master center, it needs attention that some parameter settings will be forced to be consistent with that of the master center, that is, the management platform automatically obtains the relevant parameter values of the master center and fills in the DR center with no modification allowed, which are listed as follows:

- The version of compute node, ConfigDB and data source must be consistent with that of the master center;
- "--character-set-server", "--collation-server", "--innodb-buffer-pool-size-mb" in "More Parameters" of ConfigDB and data source must be consistent with the master center;
- The DR center will automatically generate all data nodes that have existed in the master center by default, and the data node type is the same as that in the master center (in this example, a master-master data node is automatically generated). You can choose other data source types and click Generate to regenerate, but the number and name of data nodes must be consistent with the master center;



- The time synchronization address and operating system time zone must be consistent with the master center.

The parameter configuration of the DR center reference is as follows ([Deployment architecture](#)):



**Compute Node Master/Slave Node Mode**

Master/Slave Type	*Compute Node Name	*Login Method	*Hostname	*Port	*SSH User	*SSH Password	*KG1 garbage collector	*Heap Memory (G)	*Direct Memory (G)	*Service Port	*Management Port
Master	Compute Node-01	Passwd	192.168.240.155	22	root	.....	On	4	24	3323	3325
Slave	Compute Node-02	Passwd	192.168.240.165	22	root	.....	On	4	24	3323	3325

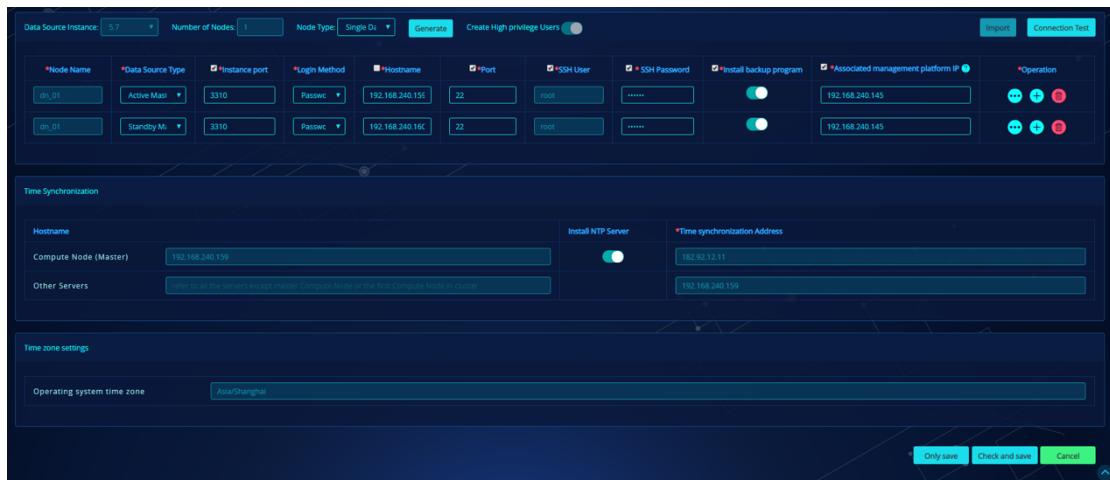
**ConfigDB** Master A, Version 5.7, Create High privilege Users On, Prompt: sharing server between ConfigDB and Data Source is not recommended. If the server is shared, the common parameter of Instance configuration is subject to Data Source instance.

Master/Slave Type	*Instance port	*Login Method	*Hostname	*Port	*SSH Username	*SSH Password	Disk device name	Data Directory	Operation
Master	3306	Passwd	192.168.240.155	22	root	.....	Disk device name of data disk	/data as default	...
Standby Master	3306	Passwd	192.168.240.155	22	root	.....	Disk device name of data disk	/data as default	...

**Keepalived**

Master	*virtual_ipaddress	*virtual_router_id	*keepalived_net_interface_name
Master	192.168.240.107/24	107	eth0.1
Slave	192.168.240.107/24	107	eth0.1



**Data Source Instance** 5.7, Number of Nodes: 1, Node Type: Single Dc, Generate, Create High privilege Users On.

*Node Name	*Data Source Type	*Instance port	*Login Method	*Hostname	*Port	*SSH User	*SSH Password	*Install backup program	*Associated management platform IP	*Operation
dn_01	Active Mst	3310	Passwd	192.168.240.155	22	root	.....	On	192.168.240.145	...
dn_01	Standby Mst	3310	Passwd	192.168.240.165	22	root	.....	On	192.168.240.145	...

**Time Synchronization**

Hostname	Install NTP Server	*Time synchronization Address
Compute Node (Master) 192.168.240.155	On	192.92.12.11
Other Servers Other could use servers which install Compute Node or other first Compute Node in cluster		192.168.240.159

**Time zone settings**

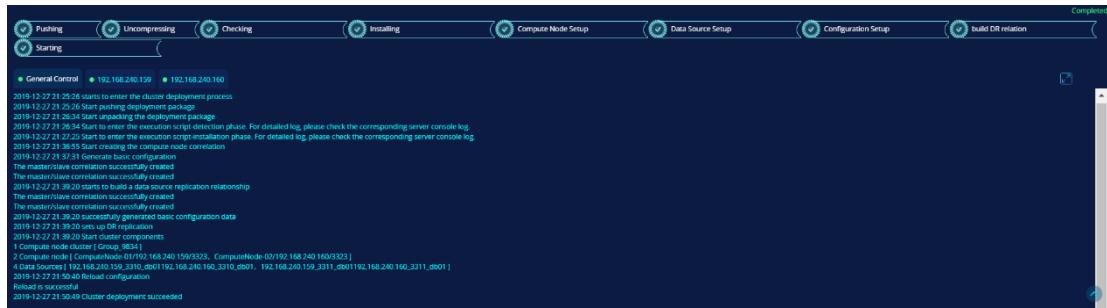
Operating system time zone
Asia/Shanghai

### (3) Start deployment:

- Click [check and save], the management platform will check whether there is error. When it is all correct, click [Start deployment] to enter the installation process.
- In step "Generate basic configuration", configuration of the DR center will be updated to ConfigDB and server.xml of the master center synchronously and take effect after dynamic loading.
- In step "Set up disaster recovery replication", data of the master center ConfigDB

and data source will be migrated to the DR center ConfigDB and data source, and the disaster recovery replication will be set up between the ConfigDB and data source of the two IDCs.

Note: due to the data import, you may wait a long time if the master center data source has a large data amount.



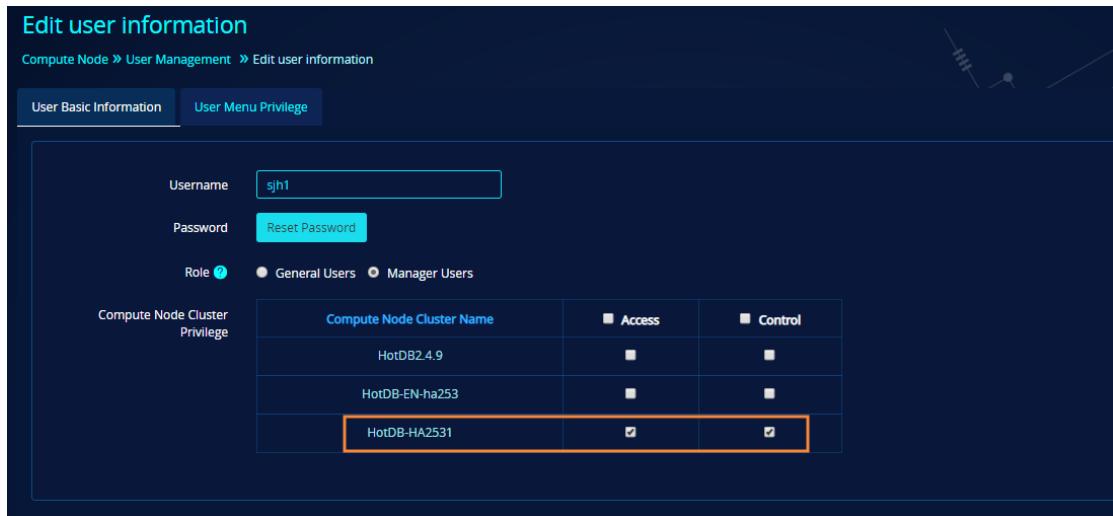
### 3.4.1.3. Cluster management

After the successful installation and deployment of the DR center, you can see a compute node cluster with two sets of IDCs and with the DR mode enabled on the cluster management page.

Compute Node Cluster														
Cluster Management > Compute Node Cluster														
ID	Group Name	Cluster Mode	IDC Type	Cluster Component Configuration						Cluster Deployment Info				
				Component Name	IP Address	Service Port	Management Port	Type	Compute Node	High Available Components	ConfigDB	NTPD Time Service	Data Source Instance	Deployment Environment Grade
9	Group_8788	Master/Slave Node (v.2.5.3.1)	Master center	ComputeNode-01	192.168.240.155	3323	3325	Master	Active	Active	Active	Active	Total number of instances: 4 Active: 4	Not Examination
				ComputeNode-02	192.168.240.156	3323	3325	Slave	Active	Active	Active	Active	Total number of instances: 4 Active: 4	Not Examination
	Group_2685 (Deploy Monitoring)	Master/Slave Node (v.2.5.3.1)	DR center	ComputeNode-01	192.168.240.159	3323	3325	Master	Active	Active	Active	Active	Total number of instances: 4 Active: 4	Not Examination
				ComputeNode-02	192.168.240.160	3323	3325	Slave	Active	Active	Active	Active	Total number of instances: 4 Active: 4	Not Examination
1	Group_8788	Master/Slave Node (v.2.5.3.1)	Master center	ComputeNode-01	192.168.240.141	3323	3325	Master	Configured	Configured	Configured	Configured	Configured	Not Examination
				ComputeNode-02	192.168.240.142	3323	3325	Slave	Configured	Configured	Configured	Configured	Configured	Not Examination
	Group_2685 (Deploy Monitoring)	Master/Slave Node (v.2.5.3.1)	DR center	ComputeNode-01	192.168.240.143	3323	3325	Master	Active	Active	Active	Active	Total number of instances: 4 Active: 4	Not Examination
				ComputeNode-02	192.168.240.144	3323	3325	Slave	Active	Active	Active	Active	Total number of instances: 4 Active: 4	Not Examination

- The IDC types include the master center and the DR center. According to the IDC status, the icon  next to the IDC type indicates that type of the IDC is current active center;
- The two IDCs have their own component configuration and deployment, among which the current standby center does not provide the access to [Switch] and [Rebuild], that is, the current standby center does not allow to manually switch

- compute nodes;
- The two IDCs in a cluster temporarily do not support separate monitoring pause or start, and the user privilege also regards the two IDCs as a cluster and manages uniformly;



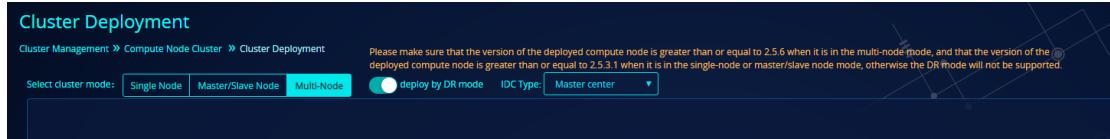
Compute Node Cluster Name	Access	Control
HotDB2.4.9	<input type="checkbox"/>	<input type="checkbox"/>
HotDB-EN-ha253	<input type="checkbox"/>	<input type="checkbox"/>
HotDB-HA2531	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

- If the cluster with DR mode enabled is deleted, the master center and the DR center will be deleted at the same time;
- On the compute node cluster editing page, if the "Enable DR mode" is ON, it is not allowed to be OFF; if it is OFF, it is allowed to be ON. The existing cluster information is used as the master center configuration, and the DR center configuration can be added to it.

### 3.4.1.4. Disaster recovery environment deployment in multi-node mode

The disaster recovery deployment of the multi-node cluster mode is basically the same as the steps of the master/slave compute node mode described in the previous chapter. The differences are as follows:

1. For disaster recovery deployment of the multi-node cluster mode, the version of compute node must be 2.5.6 and above, and "multi node" must be selected at the entrance of cluster deployment. On this basis, the master center and DR center of disaster recovery mode can be deployed.



2. For the multi-node cluster, the master center should be deployed before the DR center.
3. The configuration of the multi-node cluster in the master center and DR center is consistent with the deployment that needs to be filled in for the common multi-node cluster mode deployment.
4. In the process of automatic deployment through the management platform, the program will configure server.xml according to the information of the master center and DR center by default. The associated configuration items include:

#### **Modification items for cluster mode in master center:**

haMode is set to 4, that is, the cluster mode master center

idcId is set to 1, that is, the master center

idcNodeHost is set to the compute node information of the DR center: host name (IP) + service port, separated by English commas, for example: 192.168.220.112:3325,192.168.220.113:3325,192.168.220.114:3325

Note: the clusterName in the cluster of the same IDC must be consistent, but the master center and the DR center cannot be consistent. Therefore, the master center can be configured as HotDB-Cluster-idc1-groupID by default (groupID is the platform cluster group ID, in order to distinguish the deployment of multiple clusters on the same server).

#### **Modification items for cluster mode of DR center:**

haMode is set to 5, that is, cluster mode DR center

idcId is set to 2, that is, the DR center

idcNodeHost is set to the compute node information of the master center: host name (IP) + service port, separated by English commas, for example: 192.168.210.86:3325,192.168.210.87:3325,192.168.210.88:3325

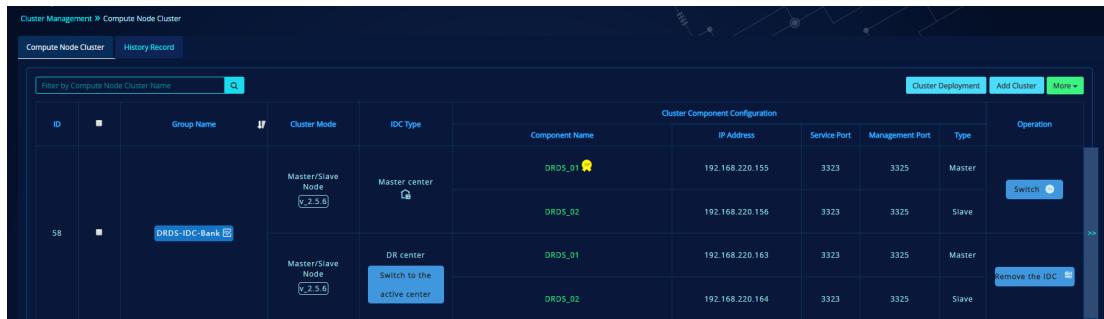
Note: the clusterName in the cluster of the same IDC must be consistent, but the master center and the DR center cannot be consistent. Through platform deployment, the default configuration is: HotDB-Cluster-idc1/idc2-

groupID (groupID is the platform cluster group ID, in order to distinguish the deployment of multiple clusters on the same server)

1. In addition to the special modification items described above, for the deployment of multi-node cluster in DR mode you should also pay attention to:

- ① The serverID in the same IDC must be continuous. In the master center and DR center, it is according to the number of nodes and increases from 1 by default.
- ② When all compute nodes of the cluster in the same IDC are deployed on one server or deployed across network segments (that is, when all compute nodes in the cluster are not on the same network segment or cluster ports are not the same), haNodeHost in the master center and DR center should be configured as the cluster information of all members in the IDC: host name (IP) + communication port, for example: 192.168.210.86:3326,192.168.210.86:3327,192.168.210.86:3328. When all compute nodes of the cluster are on the same network segment and the cluster ports are the same, configuration of this parameter is not required.
- ③ ConfigMGR (whether ConfigDB uses MGR) must be set to false, and ConfigDB MGR is not supported in multi-node cluster mode.
- ④ Other cluster related modification items, such as clusterSize, clusterNetwork, clusterHost and clusterPort, are consistent with the configured mode in the ordinary single IDC.

2. It allows the master center to be multi-node cluster mode, DR center to be the master/slave compute node mode or vice versa. Different architectures are compatible.
3. The corresponding host names of all components in the DR center cannot be consistent with the master. Otherwise, it will not pass the verification.
4. The information of multi-node cluster after normal deployment is shown in the list as follows:



The screenshot shows a table of cluster component configurations. The columns include ID, Group Name, Cluster Mode, IDC Type, Component Name, IP Address, Service Port, Management Port, Type, and Operation. There are two main rows for the DRDS-IDC-Bank cluster (ID 58).

ID	Group Name	Cluster Mode	IDC Type	Cluster Component Configuration					Operation
				Component Name	IP Address	Service Port	Management Port	Type	
58	DRDS-IDC-Bank	Master/Slave Node v 2.5.0	Master center	DRDS_01	192.168.220.155	3323	3325	Master	<button>Switch</button>
		Master/Slave Node v 2.5.0	DR center	DRDS_02	192.168.220.156	3323	3325	Slave	<button>Switch</button>
		Master/Slave Node v 2.5.0	DR center	DRDS_01	192.168.220.163	3323	3325	Master	<button>Switch to the active center</button>
				DRDS_02	192.168.220.164	3323	3325	Slave	<button>Remove the IDC</button>

### 3.4.2. Disaster recovery environment deployment based on a running cluster

The cluster deployment function of management platform is able to deploy DR center for the running cluster. Requirements for the cluster are as follows.

- If compute node mode in the cluster is master/slave compute node mode, and the version is higher than 2.5.3.1; if it is multi-node mode, and the version needs to be higher than 2.5.6 and above
- The mode of ConfigDB and data source in the cluster cannot be MGR;
- SSH information must be added to management platform in the cluster. If not, you can try again after adding it on the compute node cluster editing page;
- ConfigDB or management port of the cluster must be connected normally;
- ConfigDB or data source in the cluster must have GTID enabled.
- Try to ensure that all components of the cluster are in normal operating status, including but not limited to: no ConfigDBs or data sources marked as unavailable, master/slave status in which the high availability switch can be performed and the required configuration is met, server resource usage lower than the alert threshold, cluster throughput lower than the alert threshold, etc.

#### Deployment steps are as follows:

1. After clicking the [Cluster Deployment] button, click to enable "Deploy by DR mode", and choose "DR center" in "IDC type". Then, two cluster names, Compute node mode cluster with DR mode disabled and DR mode cluster with no corresponding DR center, will appear in the drop-down box of cluster name of master center.
2. After choosing a target cluster name and clicking [Parameter Configuration], if master center is deployed through management platform, the management platform will also automatically obtain the relevant parameter values of master center and automatically fill in DR center with no modification allowed. Please refer to the [parameter configuration and deployment process of DR center](#). If the former master

center is manually deployed offline, it is required to manually ensure that the parameters filled in are consistent with the master center configuration.

## 3.5. Manual deployment

### 3.5.1. Deploy a new set of disaster recovery environment

When deploying the disaster recovery environment, the deployment process of components in master center and DR center are the same as deployment process in single-IDC mode. Please refer to the [Distributed Transactional Database HotDB Server - V2.5.3 \[Installation & Deployment\] Function Manual](#). It needs extra attention that the number of data nodes in both IDCs should be the same, and the names should correspond one by one. The content below is masterly about the required configuration modification after deployment of the basic components of master center and DR center.

#### 3.5.1.1. Parameter configuration modification

##### 3.5.1.1.1. Master/slave mode

When deploying the compute nodes of master center or DR center, the [new parameters](#) of the DR mode need to be modified accordingly. According to the [deployment architecture](#), if the compute node mode is master/slave mode, the compute node parameters of the two IDCs should be modified as follows:

Master center:

```
<property name="url">jdbc:mysql://192.168.220.186:3306/hotdb_config</property><!-- Master  
ConfigDB address, which needs to be the real IP address of the ConfigDB service -->  
  
<property name="username">hotdb_config</property><!-- Master ConfigDB user name -->  
  
<property name="password">hotdb_config</property><!-- Master ConfigDB password -->  
  
<property name="bakUrl">jdbc:mysql://192.168.220.187:3306/hotdb_config</property><!-- Slave  
ConfigDB address, which needs to be the real IP address of the ConfigDB service -->  
  
<property name="bakUsername">hotdb_config</property><!-- Slave ConfigDB user name -->  
  
<property name="bakPassword">hotdb_config</property><!-- Slave ConfigDB password -->
```

```

<property name="drUrl">jdbc:mysql://192.168.220.188:3306/hotdb_config</property><!--
ConfigDB address in DR center -->

<property name="drUsername">hotdb_config</property><!-- ConfigDB user name in DR center -->

<property name="drPassword">hotdb_config</property><!-- ConfigDB password in DR center -->

<property name="drBakUrl">jdbc:mysql://192.168.220.189:3306/hotdb_config</property><!--
Slave ConfigDB address in DR center -->

<property name="drBakUsername">hotdb_config</property><!-- Slave ConfigDB user name in DR
center -->

<property name="drBakPassword">hotdb_config</property><!-- Slave ConfigDB password in DR
center -->

<property name="haMode">2</property><!-- High-availability mode, 0:HA, 1:Cluster, 2:HA in
master center, 3:HA in DR center) -->

<property name="idcId">1</property><!-- ID of IDC, 1:master center, 2:DR center -->

<property name="idcNodeHost">192.168.220.188:3325,192.168.220.189:3325</property><!--
Computer node info in the other IDC -->

```

### DR center:

ConfigDB-related configuration is consistent with that of the master center, unnecessary details omitted.

```

<property name="haMode">3</property><!-- High-availability mode, 0:HA, 1:Cluster, 2:HA in
master center, 3:HA in DR center) -->

<property name="idcId">2</property><!-- ID of IDC, 1:master center, 2:DR center -->

<property name="idcNodeHost">192.168.220.186:3325,192.168.220.187:3325</property><!--
Computer node info in the other IDC -->

```

### 3.5.1.1.2. Multi-node mode

Please refer to the relevant parameter adjustment instructions in the chapter [Disaster Recovery Environment Deployment in Multi-node Mode](#), and adjust it as needed (focus

on points 4 and 5)

### 3.5.1.2. Set up ConfigDB replication relations

This section will introduce the setup of ConfigDB replication relations. The deployment architecture is as follows:

IDC Type	Component Type	Role	Code Name	Connection Information
Master Center	ConfigDB	Master	hc01	192.168.220.186_3306
		Standby Master	hc02	192.168.220.187_3306
DR Center	ConfigDB	Master	hc03	192.168.220.188_3306
		Standby Master (Actually master-slave replication relation)	hc04	192.168.220.189_3306

The ConfigDB is essentially a standard MySQL instance, so under this deployment architecture, the order of setting up the ConfigDB should be:

- ① Set up a master-master replication relation between ConfigDBs in the master center;
- ② Set up a master-slave replication relation between the master ConfigDB in the master center and the master ConfigDB in the DR center;
- ③ Set up a master-slave replication relation between ConfigDBs in the DR center;
- ④ Import the ConfigDB data into the master ConfigDB in the master center.

The detailed steps are as follows:

1. By default, MySQL instances are installed on these four servers and GTID is enabled

in the configuration parameters.

Note: for more requirements and more detailed master-slave replication setup steps, please refer to the relevant chapters in the "[Recommended steps for replication setup](#)". This section describes the basic operations performed without considering the influence of other factors.

2. Set up a master-master replication relation between the MySQL instances of servers 220.186 and 220.187.

Execute the code on Instance 192.168.220.186\_3306:

```
mysql> change master to  
master_host='192.168.220.187',master_user='repl',master_password='repl',Master_Port=3306,mas  
ter_auto_position=1;  
  
mysql> start slave;
```

Execute the code on Instance 192.168.220.187\_3306:

```
mysql> change master to  
master_host='192.168.220.186',master_user='repl',master_password='repl',Master_Port=3306,mas  
ter_auto_position=1;  
  
mysql> start slave;
```

At this time, by executing show slave status on 220.186 and 220.187 respectively, you can see that the relation between Master and Host is master-master, and Slave\_IO\_Running and Slave\_SQL\_Running are in normal status. For example:

Execute the code on Instance 192.168.220.186\_3306:

```
mysql> show slave status\G  
*****  
Slave_IO_State: Waiting for master to send event  
  
Master_Host: 192.168.220.187  
  
Master_User: repl
```

```
Master_Port: 3306  
  
Connect_Retry: 60  
  
Master_Log_File: mysql-bin.000001  
  
Read_Master_Log_Pos: 57030  
  
Relay_Log_File: mysql-relay-bin.000002  
  
Relay_Log_Pos: 4785  
  
Relay_Master_Log_File: mysql-bin.000001  
  
Slave_IO_Running: Yes  
  
Slave_SQL_Running: Yes
```

### 3. Set up a master-slave replication relation (DR replication) between MySQL instances on servers 220.186 and 220.188.

Execute the code on Instance 192.168.220.188\_3306:

```
mysql> change master to  
master_host='192.168.220.186',master_user='repl',master_password='repl',Master_Port=3306,mas  
ter_auto_position=1;  
  
mysql> start slave;
```

### 4. Set up a master-slave replication relation between MySQL instances on servers 220.188 and 220.189.

Execute the code on Instance 192.168.220.189\_3306:

```
mysql> change master to  
master_host='192.168.220.188',master_user='repl',master_password='repl',Master_Port=3306,mas  
ter_auto_position=1;  
  
mysql> start slave;
```

### 5. Import the ConfigDB data to the master ConfigDB 192.168.220.186\_3306 in the master center.

```
Execute the code on Server 192.168.220.186:
```

```
root> mysql -uroot -S /data/mysql/mysqldata3306.sock/mysql.sock < /usr/local/hotdb/hotdb-server/conf/hotdb_config.sql
```

6. Wait for the replication latency to catch up, then log in to Instance 3306 on four servers to find the imported ConfigDB data, which means the ConfigDB deployment is successful.

```
mysql> use hotdb_config;  
  
mysql> show tables;  
  
+-----+  
| Tables_in_hotdb_config_replication |  
+-----+  
| hotdb_auto_failover |  
| hotdb_config_info |  
| hotdb_config_info_running |  
| hotdb_datanode |  
| hotdb_datanode_running |  
| hotdb_datasource |  
...More details omitted ...
```

### 3.5.1.3. Set up data source replication relations

Under a new disaster recovery environment deployed from zero, ConfigDB and the data source are essentially standard new MySQL instances, among which the difference is that the ConfigDB data needs to be imported after the ConfigDB replication relation is set up, while the data source replication relation has no such requirement. Therefore, for the process of setting up the data source disaster recovery relation, please refer to the [Set up ConfigDB replication relations](#) and skip the Importing ConfigDB Data step.

### 3.5.1.4. Add compute node clusters to management platform

If the clusters have been successfully deployed offline, you can add the compute node clusters to management platform by filling in the configuration through cluster adding function of management platform.

#### (1) Compute node:

Click "Enable DR mode", and fill in two sets of compute node information of the master center and the DR center. The numbers of compute nodes in two IDCs are selected respectively by two "compute node modes". Select according to different compute node modes. If it is master/slave mode, select master/slave node. If it is multi-node mode, select multi node.



Notes:

- ① In the multi-node mode, you need to add the configuration of cluster communication ports. If the compute nodes are distributed on different servers and the communication ports are consistent, you can fill in only one port (3326 by default). If the compute nodes are deployed on the same or multiple servers with different ports, you can use English commas to separate the communication ports, for example, 3326,3327,3328, and the order of the communication ports should correspond to the order of adding the compute nodes. Otherwise, when saving the verification, it will refuse to save and remind "the compute nodes are on the same server, the communication ports must match the compute nodes in numbers and cannot be repeated.".
- ② If multiple compute nodes in the cluster are deployed on the same server, the communication port+ service port+ management port should be unique to each other, and there can't be the same port combination.



The screenshot shows the 'Add Compute Node Cluster' page. In the 'Cluster Info' section, the 'Compute node mode of master center' dropdown is set to 'Multi-Node'. Below it, the 'Group Name' field contains 'Enter Compute Node Cluster Name' and the 'Cluster Network Segment' field contains 'Format Example: 192.168.200.0/24'. The 'Communication Port' field is set to '3326'. A note states: 'Please make sure the version of the added compute node is greater than or equal to 2.5.6 when it is in the DR mode. Please make sure the version of the added compute node is greater than or equal to 2.3.3.1 when it is in the single-node or master/slave node mode, otherwise the DR mode will not be supported.' At the bottom left is a 'Set ConfigDB manually' link.

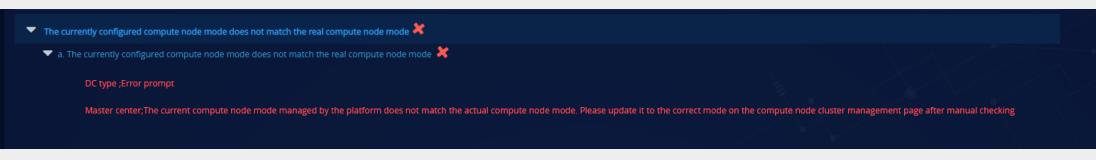
**Master center-compute node**

Type	* Compute Node Name	* Hostname	* Username	* Password	* Service Port	* Management Port	SSH Login Mode	SSH Port	SSH Username	SSH User Password	server.xml Address	keepalived.conf Address	Operation
Chist	HotDB_01	Enter Node	root	Enter Passo	3323	3325	Passwo ▾	22	Enter SSH Use	Enter SSH Passo	/usr/local/HotDB/	/etc/keepalived	+ ⚙
Chist	HotDB_02	Enter Node	root	Enter Passo	3323	3325	Passwo ▾	22	Enter SSH Use	Enter SSH Passo	/usr/local/HotDB/	/etc/keepalived	+ ⚙
Chist	HotDB_03	Enter Node	root	Enter Passo	3323	3325	Passwo ▾	22	Enter SSH Use	Enter SSH Passo	/usr/local/HotDB/	/etc/keepalived	+ ⚙

**DR center-compute node**

Type	* Compute Node Name	* Hostname	* Username	* Password	* Service Port	* Management Port	SSH Login Mode	SSH Port	SSH Username	SSH User Password	server.xml Address	keepalived.conf Address	Operation
Chist	HotDB_01	Enter Node	root	Enter Passo	3323	3325	Passwo ▾	22	Enter SSH Use	Enter SSH Passo	/usr/local/HotDB/	/etc/keepalived	+ ⚙
Chist	HotDB_02	Enter Node	root	Enter Passo	3323	3325	Passwo ▾	22	Enter SSH Use	Enter SSH Passo	/usr/local/HotDB/	/etc/keepalived	+ ⚙

③ When the cluster mode of the added compute node does not match the real compute node mode, the configuration verification will synchronously verify the current status and give an error reminder.



The error message details: 'DC type :Error prompt' and 'Master center: The current compute node mode managed by the platform does not match the actual compute node mode. Please update it to the correct mode on the compute node cluster management page after manual checking.'

## (2) ConfigDB:

If the management port of the compute node is connectable, the management platform will automatically obtain the ConfigDB information without the need of manually filling in the ConfigDB; if the management port of the compute node is not connectable, the management platform cannot obtain the ConfigDB information and needs to manually add the ConfigDB information. Check "Manually setting ConfigDB" and fill in the ConfigDB information of two IDCs respectively.

第 33 页 /共 138 页

**Cluster Info**

Compute node mode of master center

<input checked="" type="radio"/> Single Node	<input type="radio"/> Master/Slave Node	<input type="radio"/> Multi-Node
--	---	----------------------------------

\* Group Name:

Set ConfigDB manually

Please ensure that the added compute node version is greater than or equal to 2.5.3.1, otherwise the DR mode cannot be supported

ConfigDB mode of Master center

<input type="radio"/> Single Database	<input checked="" type="radio"/> Master-Master	<input type="radio"/> MGR
---------------------------------------	--	---------------------------

IDC Type	ConfigDB Types	ConfigDB Address	ConfigDB Username	ConfigDB Password
Master center	Active Master	<input type="text" value="jdbc:mysql://192.168.220.214:3316/hotdb_config"/>	<input type="text" value="hotdb_config"/>	<input type="text" value="....."/>
Master center	Standby Master	<input type="text" value="jdbc:mysql://192.168.220.215:3316/hotdb_config"/>	<input type="text" value="hotdb_config"/>	<input type="text" value="....."/>

ConfigDB mode of DR center

<input type="radio"/> Single Database	<input checked="" type="radio"/> Master-Master
---------------------------------------	--

IDC Type	ConfigDB Types	ConfigDB Address	ConfigDB Username	ConfigDB Password
DR center	Active Master	<input type="text" value="jdbc:mysql://192.168.240.76:3316/hotdb_config"/>	<input type="text" value="hotdb_config"/>	<input type="text" value="....."/>
DR center	Standby Master	<input type="text" value="jdbc:mysql://192.168.240.77:3316/hotdb_config"/>	<input type="text" value="hotdb_config"/>	<input type="text" value="....."/>

### (3) Connection detection:

Click Detect to detect whether the connection of compute nodes is normal.

Current Status	
<b>Management port connection status</b>	
Master center	<input checked="" type="checkbox"/>
DR center	<input checked="" type="checkbox"/>
<b>Obtain the SERVER.XML status</b>	
Master center	<input checked="" type="checkbox"/>
DR center	<input checked="" type="checkbox"/>
<b>ConfigDB connection status</b>	
Master center	<input checked="" type="checkbox"/>
DR center	<input checked="" type="checkbox"/>
<b>Server SSH configuration connection status</b>	
Master center	<input checked="" type="checkbox"/>
DR center	<input checked="" type="checkbox"/>
<b>ConfigDB DR status.</b>	
DR status is normal.	<input checked="" type="checkbox"/>

Example of normal connection

Current Status	
Management port connection status	
Master center	✗
DR center	✗
Obtain the SERVERXML status	
Master center	✗
DR center	✗
ConfigDB connection status	
Master center	✗
DR center	✗
Server SSH configuration connection status	
Master center	✗
DR center	✗
ConfigDB DR status.	
DR status is normal.	✗
I got it	

Example of abnormal connection

- IDC identifications are added to four detection items in the single-IDC mode, which can clearly indicate which IDC has abnormal connections;
- Disaster recovery status detection of ConfigDB is added, including two items:
  - ConfigDB replication status: detect whether the master-master (master-slave) replication relation between ConfigDBs in the two IDCs is normal and whether the disaster recovery relation between master ConfigDBs in the two IDCs is normal. If the disaster recovery relation between master ConfigDBs in the two IDCs is abnormal, the replication status of ConfigDB in the IDC will not be detected.
  - metadata consistency: whether the data of all ConfigDBs in the two IDCs are consistent with the current active ConfigDB in the master center.

Current Status			
ConfigDB DR status.			
Consistent metadata			
Metadata in master center and DR center is inconsistent,detail:			
ConfigDB Tables	ConfigDB Address	Result	Details
HOTDB_USER_RUNNING	192.168.240.156:3306/hotdb_config	Inconsistent	表: HOTDB_USER_RUNNING在节点: 192.168.240.156:3306,hotdb_config存在数据不一致,列: USERNAME, 分布区间为: r%, 并且不一致唯一键为: (USERNAME,HOST) : (root,%)(Data inconsistency in table:HOTDB_USER_RUNNING in datanode:192.168.240.156:3306,hotdb_config, inconsistent data is:USERNAME distribution in :r%; and the unique key of inconsistent rows is (USERNAME,HOST) : (root,%))
HOTDB_USER	192.168.240.156:3306/hotdb_config	Inconsistent	表: HOTDB_USER在节点: 192.168.240.156:3306,hotdb_config存在数据不一致,列: USERNAME, 分布区间为: r%, 并且不一致唯一键为: (USERNAME,HOST) : (root,%)(Data inconsistency in table:HOTDB_USER in datanode:192.168.240.156:3306,hotdb_config, inconsistent data is:USERNAME distribution in :r%; and the unique key of inconsistent rows is (USERNAME,HOST) : (root,%))

- When the DR center is switched to the current active IDC after the failure of the master center, the [Detect] here will detect based on the IDC status, that is, detect the port connection status of the current active center; detect the data consistency

of all other ConfigDBs with the current active ConfigDB in the current active center as a standard; detect the replication status internal the IDC or between IDCs, etc.

### 3.5.1.5. Add data sources for IDCs

With the compute node cluster added successfully, you can switch to the general user role, select the added cluster on the [compute node cluster selection](#) page, and click [Add Node](#) on the node management page to add data nodes and data sources for both the master center and the DR center at the same time. For details, please refer to the relevant linked section.

## 3.5.2. Disaster recovery environment deployment based on a running cluster

### 3.5.2.1. Parameter configuration modification

To deploy the DR center on the basis of the running cluster, it is necessary to modify the [new parameters](#) related to the DR mode of the master center. For the modification, please refer to the [compute node parameter configuration modification](#) under the chapter of deploying a new set of disaster recovery environment.

### 3.5.2.2. Set up ConfigDB replication relations

When deploying a DR center on the basis of a running cluster, the steps of setting up ConfigDB replication relations are basically the same as that of deploying a new set of disaster recovery environment. The difference is that when deploying a DR center on the basis of a running cluster, due to the existing configuration information in the ConfigDB of master center, the master ConfigDB data of master center should be migrated to the master ConfigDB of DR center before setting up when setting up a disaster recovery relation. Therefore, the setting up sequence of ConfigDB should be:

- ① The master-master replication relation between ConfigDBs in the master center is set up by default;

- ② Import the data of the master ConfigDB of the master center into the master ConfigDB of the DR center;
- ③ Set up a master-slave replication relation between the master ConfigDB of the master center and the master ConfigDB of the DR center;
- ④ Import the data of the master ConfigDB of the DR center into the slave ConfigDB of the DR center;
- ⑤ Set up a master-slave replication relation between ConfigDBs in the DR center;

#### **Deployment architecture for reference is as follows:**

IDC Type	Component Type	Role	Code Name	Connection Information
Master Center	ConfigDB B	Master	hc01	192.168.220.186_3306
		Standby master	hc02	192.168.220.187_3306
DR Center	ConfigDB B	Master	hc03	192.168.220.188_3306
		Standby master (Actually master-slave replication relation)	hc04	192.168.220.189_3306

#### **The detailed steps are as follows:**

1. By default, the master-master replication status of the ConfigDB in the master center is normal and the data is consistent at this time, and MySQL instance has been installed on the server in the DR center with GTID enabled in the configuration parameters.

Note: for more requirements and more detailed master-slave replication setup steps, please refer to the relevant chapters in the "[Recommended steps for replication setup](#)". This section describes the basic operations performed without considering the influence of other factors.

## 2. Import the data from MySQL Instance 220.186\_3306 to MySQL Instance 220.188\_3306.

Execute the code on Server 192.168.220.186:

```
root> mysqldump --no-defaults -uroot --port=3306 --no-tablespaces --default-character-set=utf8mb4 --all-databases --set-gtid-purged --single-transaction --events --routines --triggers --hex-blob >/usr/local/config_data.sql ;echo $?
```

0 //If the returned result is 0, the data is imported successfully:

Transmit the exported sql file to 192.168.220.188 via scp command:

```
root> scp /usr/local/config_data.sql root@192.168.220.188:/usr/local/
```

Execute the code on Server 192.168.220.188:

```
root> mysql --no-defaults --default-character-set=utf8mb4 --binary-mode --disable-reconnect --host=192.168.220.188 --port=3306 --uroot -proot < config_data.sql;echo $?
```

0 //If the returned result is 0, the data has been successfully imported:

At this time, you can see that the master ConfigDB data of the DR center has been imported successfully:

```
mysql> show tables;
```

+-----+
Tables_in_hotdb_config_replication
+-----+
hotdb_auto_failover
+-----+
hotdb_config_info
+-----+
hotdb_config_info_running
+-----+
hotdb_datanode
+-----+
hotdb_datanode_running
+-----+
hotdb_datasource

...More details omitted...

### 3. Set up the master-slave replication relation (disaster recovery relation) between MySQL instances of servers 220.186 and 220.188.

Execute the code on Instance 192.168.220.188\_3306:

```
mysql> change master to  
master_host='192.168.220.186',master_user='repl',master_password='repl',master_Port=3306,mas  
ter_auto_position=1;  
  
mysql> start slave;
```

### 4. Import data of MySQL instance 220.188\_3306 to MySQL instance 220.189\_3306.

Execute the code on Server 192.168.220.188:

```
root> mysqldump --no-defaults -uroot --port=3306 --no-tablespaces --default-character-  
set=utf8mb4 --all-databases --set-gtid-purged --single-transaction --events --routines --  
triggers --hex-blob >/usr/local/config_data.sql ;echo $?
```

0 //If the returned result is 0, the data is imported successfully:

Transmit the exported sql file to 192.168.220.189 via scp command:

```
root> scp /usr/local/config_data.sql root@192.168.220.189:/usr/local/
```

Execute the code on Server 192.168.220.189:

```
root> mysql --no-defaults --default-character-set=utf8mb4 --binary-mode --disable-reconnect  
--host=192.168.220.189 --port=3306 --uroot -proot < config_data.sql;echo $?
```

0 //If the returned result is 0, the data has been successfully imported:

### 5. Set up master-slave replication relations of MySQL instances between servers 220.188 and 220.189.

Execute the code on Instance 192.168.220.189\_3306:

```
mysql> change master to
master_host='192.168.220.188',master_user='repl',master_password='repl',master_Port=3306,mas
ter_auto_position=1;

mysql> start slave;
```

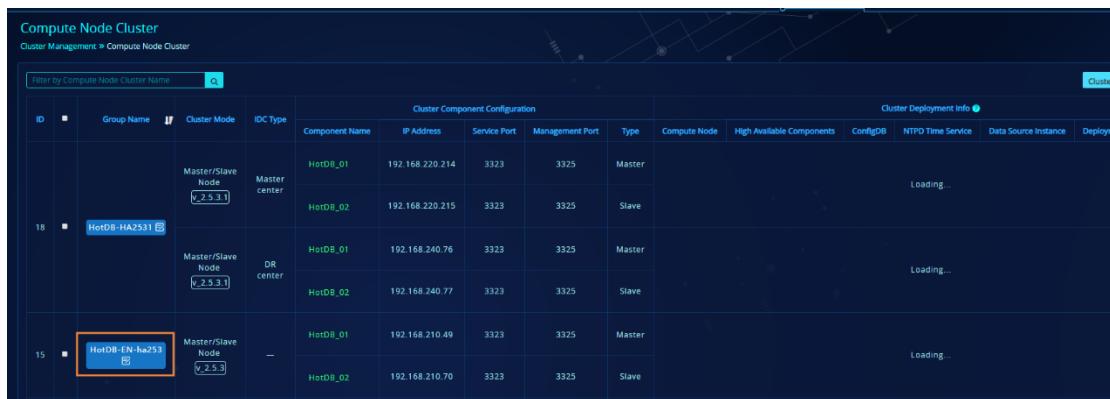
6. At this time, the data modification of master ConfigDB in master center will be synchronized to all ConfigDBs of the two IDCs, that is, the ConfigDB is successfully deployed.

### 3.5.2.3. Set up data source replication relations

ConfigDB and data source are essentially standard MySQL instances, so the principle of setting up replication relations is the same: both need to import data before setting up the replication relation. You can refer to [Set up ConfigDB replication relations](#).

### 3.5.2.4. Add compute node clusters to management platform

If clusters in the DR center have been successfully deployed offline, you can modify the DR mode status of the running cluster through management platform, fill in the configuration information of DR center, and add the DR center to the cluster.



ID	Group Name	Cluster Mode	IDC Type	Cluster Component Configuration					Cluster Deployment Info					
				Component Name	IP Address	Service Port	Management Port	Type	Compute Node	High Available Components	ConfigDB	NTPD Time Service	Data Source Instance	Deploy
18	HotDB-HA2531	Master/Slave Node v.2.5.3.1	Master center	HotDB_01	192.168.220.214	3323	3325	Master						Loading...
				HotDB_02	192.168.220.215	3323	3325	Slave						
15	HotDB-EN-ha253	Master/Slave Node v.2.5.3.1	DR center	HotDB_01	192.168.240.76	3323	3325	Master						Loading...
				HotDB_02	192.168.240.77	3323	3325	Slave						
	HotDB-EN-ha253	Master/Slave Node v.2.5.3	-	HotDB_01	192.168.210.49	3323	3325	Master						Loading...
				HotDB_02	192.168.210.70	3323	3325	Slave						

1. On the cluster management page, click the name of a running cluster to enter the compute node cluster editing page.
2. Click "Enable DR mode" to take the existing cluster information as configuration information of the master center, and add configuration information of the DR center for it. For details, please refer to [add compute node clusters to management platform](#).



### 3.5.2.5. Add data source for DR centers

On the node management page of general users, if you click Add node, management platform will judge whether there is data node missing in the newly added DR center compared with the running master center. If any, you need to [add the missing node](#) for it. For details, please refer to the relevant linked section.

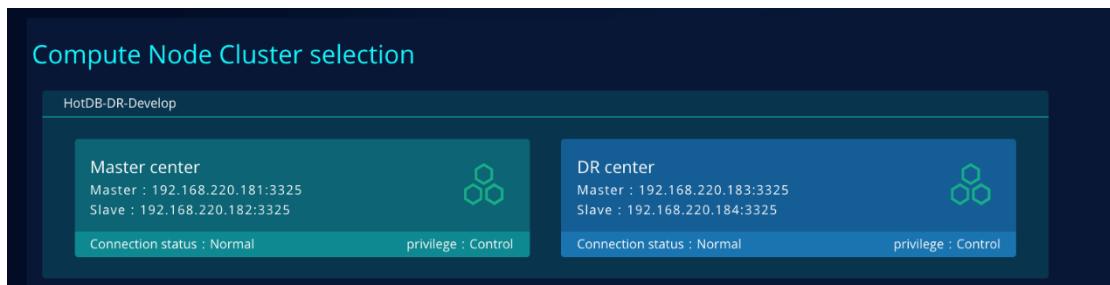
# 4. Description of functions related to management platform

This chapter needs to be read together with the [Distributed Transactional Database HotDB Server - V2.5.3 \[Management Platform\] Function Manual](#). This chapter only describes the function usage of the compute node cluster group managed by the management platform after enabling the DR mode. At the same time, the function pages not specifically mentioned display information related to the current active center and control the current active center status by default.

## 4.1. Compute node cluster selection

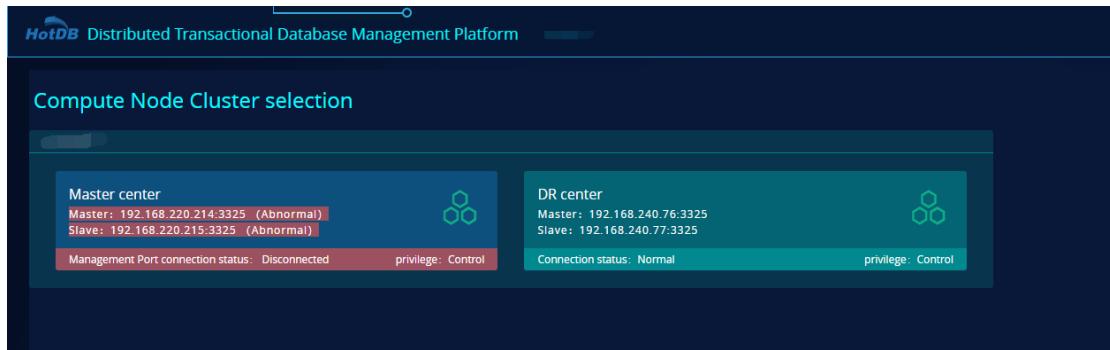
Log in to HotDB Management as a general user and enter the "compute node cluster selection" page.

- If the user has the privilege of accessing or controlling a cluster with DR mode enabled and the cluster runs normally, the current active center is displayed in green, with the current standby center in blue. When the master center is the current active center, the DR center is the current standby center, and both of them are connected normally, it is as shown in the figure below.



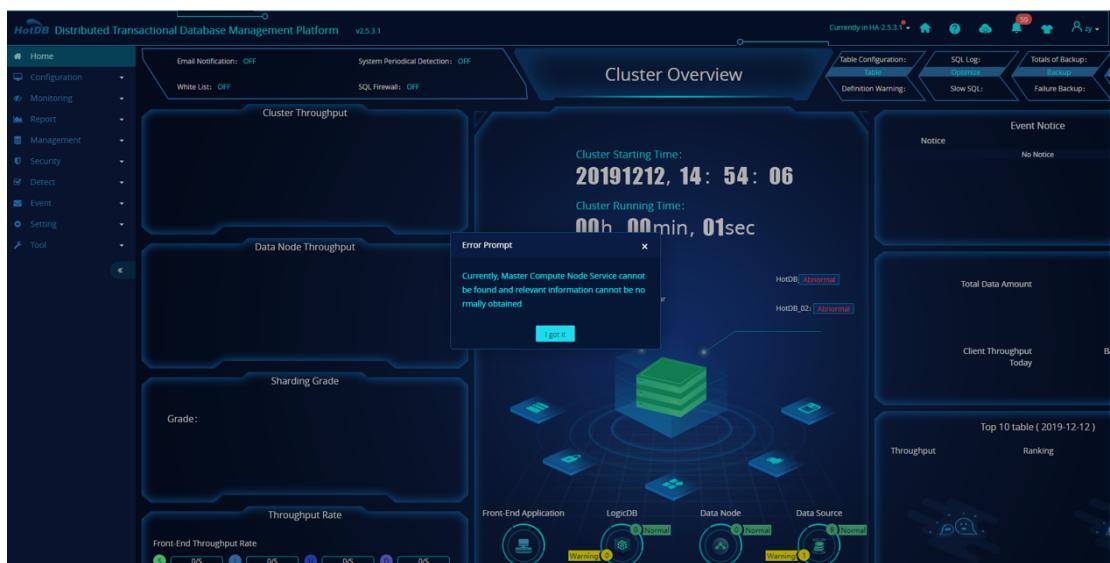
- Click either IDC module to enter the management platform. Pages not specifically mentioned in this document will display or control information of the current active center. For example, when one IDC fails to connect and switches to the other IDC, click any IDC module you will enter the management platform page that monitors the current active center, that is, the DR center after the switch.
- Each IDC module will display the connection status of its own IDC, which is displayed the same way as that of the cluster in single-IDC mode, including: when

the compute node is unable to connect, the compute node is marked red and abnormal; when the ConfigDB is unable to connect, the connection of the bottom ConfigDB is displayed abnormal; when part of the ConfigDB is unable to connect, move the cursor over the word "partially abnormal" to display the specific ConfigDB connection abnormal information, etc.



In the above screenshot, the DR center is switched to be the current active center after the master center fails.

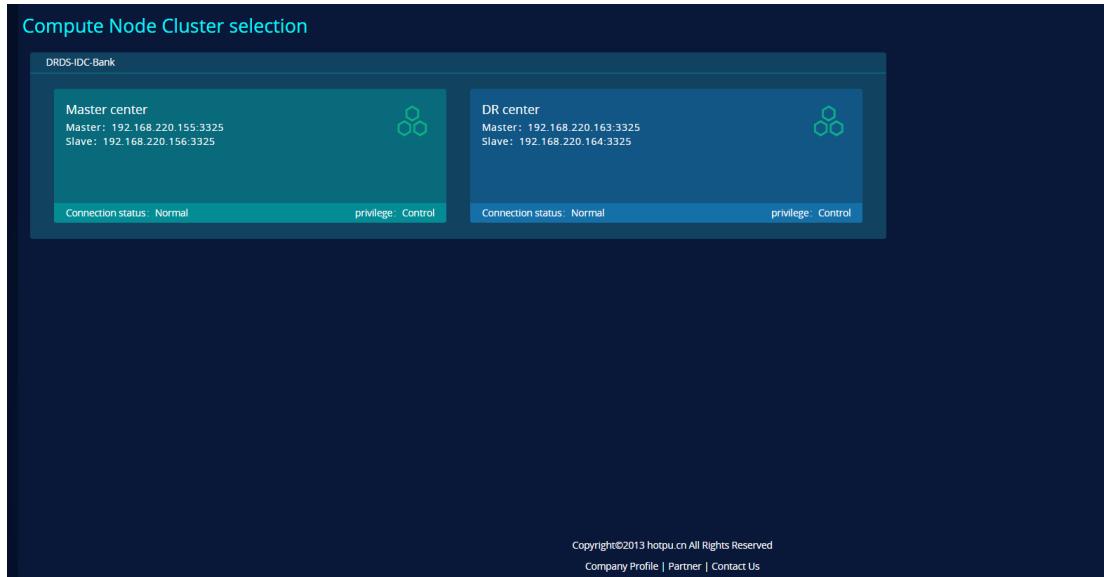
- If all compute nodes in master center cannot be connected, click either IDC panel, it will display the prompt message that the current master center cannot provide services. Please ensure that DR center will be manually enabled after all compute nodes in master center are disabled. If it is necessary to enable the DR center, please refer to the [Switch to the DR center after master center fails](#).



- After manual switching, DR center is the current active center, which is displayed in green. Click either IDC panel to enter the management platform where the DR center is identified as the current active center. If you need to restart the master

center after manual repair, please refer to [cross-IDC failure repair and switching back](#).

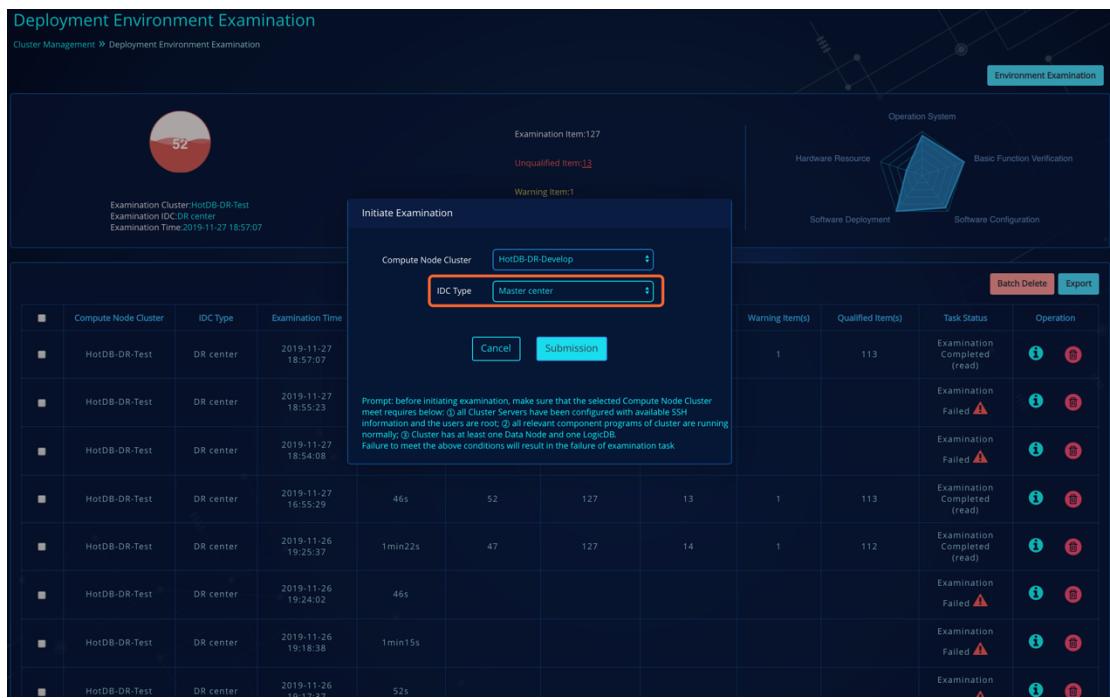
- The cluster selection page in cluster mode is shown in the following figure, and its status is similar to that of the master/slave mode.



## 4.2. Deployment environment examination

The deployment environment examination not only supports the current active center, but also the IDC that does not provide services, that is, the current standby center.

- Click [Initiate Examination], if you choose the cluster of DR mode, you can continue to choose the master center or the DR center for examination.



- Several examination items for the current standby center are skipped due to the inability to connect to the service port. The skipped items are listed as follows:

Examination Dimensions	Examination Items	Examination Details
Software Configuration	High Availability of Compute Node	The compute node service port and management port can be normally connected.
		Connection to the compute node service port via VIP is normal.
		Compute Node Mode
Basic function verification	Data Source High Availability Switch	
	Compute Node High Availability Switch	
	Backup Program	
	10s Performance Test	

## 4.3. Configuration

### 4.3.1. Node management

The node management page will manage all data nodes and all data sources of master center and DR center at the same time. For data nodes added in the master center and the DR center, it is required that the number of the data nodes is the same, and the names correspond one by one.

Data Source										
DNID	Data Node Name	Hostname	Port	Database	Type	Status	Operation	Master/Slave Status	Master/Slave Switch	IDC Type
4	4a_01	192.168.220.186	3307	db01	Active Master	Available				Master center
		192.168.220.187	3307	db01	Standby Master	Available				
4	4a_01	192.168.220.188	3307	db01	Active Master	Available				DR center
		192.168.220.189	3307	db01	Standby Master	Available				
5	4a_02	192.168.220.186	3308	db01	Active Master	Available				Master center
		192.168.220.187	3308	db01	Standby Master	Available				
5	4a_02	192.168.220.188	3308	db01	Active Master	Available				DR center
		192.168.220.189	3308	db01	Standby Master	Available				
6	4a_03	192.168.220.186	3309	db01	Active Master	Available				Master center
		192.168.220.187	3309	db01	Standby Master	Available				
6	4a_03	192.168.220.188	3309	db01	Active Master	Available				DR center

#### (1) Descriptions of the added list information:

- **IDC type:** IDC type refers to the type of IDC to which the data node belongs, including master center or DR center. Colors are displayed according to the status of IDC: available-green, standby-blue, unavailable-red. The IDC status is consistent with the IDC status displayed on the [compute node cluster selection page](#).

For example, when cluster is running normally, the master center is displayed in green, with DR center in blue; when the master center goes down, IDC types of all data nodes in the master center are displayed in red.

- **Disaster recovery status:** disaster recovery status refers to the replication status

of the current active data source between the master center and the DR center.

- If the data source of the master center is manually switched or fails over, the compute node will rebuild the replication relations of the current active data source between the master center and the DR center.
- The disaster recovery status includes Abnormal, Unknown, Normal, Setting up and Setup failed, details as follows:

**Empty:** For data nodes in the master center, the disaster recovery status is not displayed.

**Normal:** Command show slave status can be executed normally; the replication status is normal (Slave\_IO\_Running: YES, Slave\_SQL\_Running: YES) and consistent with the master-slave relation required by the disaster recovery.

**Abnormal:** replication status is abnormal (Slave\_IO\_Running: NO or Slave\_SQL\_Running: NO).

**Unknown:** the data source cannot be connected; the data source does not have sufficient privileges, detection timeout (timeout time is 1min), the non-master-slave replication relation between nodes (no replication relation is set up, or the replication relation is set up incorrectly).

**Setup failed:** it means that there is a detection failure or setup failure in the master-slave setup. Move the cursor over the word "Setup failed", it will prompt the specific failure information.

**Setting up:** it means that the master-slave setup is currently in progress. When the background operation is completed, refresh the page, and it will display Normal or Setup failed.

- If the master center fails and is switched to the DR center, the compute node will delete the replication relations of the former active data source between the master center and the DR center, and the disaster recovery status will be displayed as Unknown at this time.
- **Master/slave status:** the master/slave status function of the master center is the same as that of the single-IDC mode, which is displayed according to the configuration.
  - Note: The DR center needs to set up disaster recovery relations and multi-source replication is not allowed. When the DR center is the current standby center, if the data node type is master-master, the actual master-slave relation should be master-slave. At this time, the master-slave status only checks whether the master-slave replication is normal. When the DR center is switched

to be the current active center, if the data node type is master-master, the master-slave status will display the detection results of master-master replication status.

- If the master data source of the DR center fails, all the associated slave data sources will be set as unavailable and need manual repair.
- **Data source status:** the data source status is the same as the single-IDC mode, but it needs extra attention that when the DR center is the current standby center, if the master data source fails, the compute node will set all the related slave data sources as unavailable. When enabling the slave data source, you need to confirm that the current active data source is available, otherwise the slave data source cannot be enabled, that is, the slave data source can be enabled only after the master data source is enabled.

## (2) Descriptions of the page functions

- No [switch] button can be seen in the data source of the current standby center, that is, when the master center or the DR center is the current standby center, management platform will not provide manual switch access for the data source.
- Users can delete single or batch data sources and data nodes, while it is not allowed to separately delete the data nodes in the DR center which correspond to the master center. If the data nodes in the master center are deleted, the corresponding data nodes in the DR center will be deleted together.

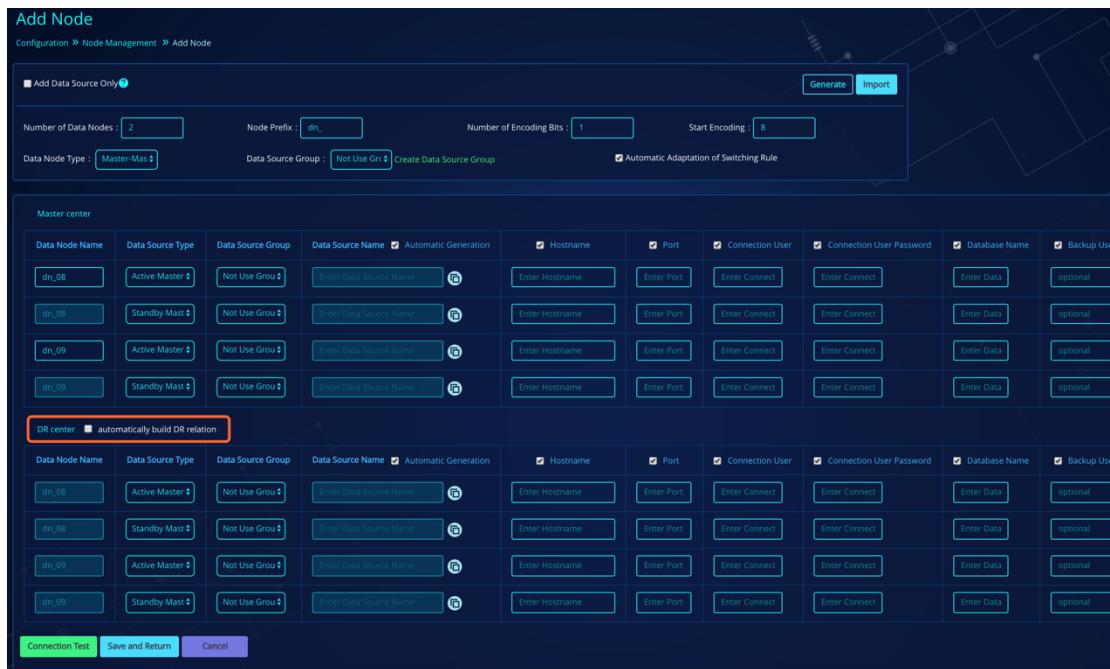
### 4.3.1.1. Add node

The function can quickly configure data nodes and data sources for two IDCs, or configure data sources for one IDC separately. Due to requirement by the DR mode that the number of data nodes added in two IDCs must be the same and the names should correspond one by one, management platform makes restrictions accordingly when adding nodes.

#### (1) Add data nodes and data sources at the same time:

- Input "Number of Data Nodes", for example, 2, which means 2 data nodes are generated under the master center and DR center modules at the same time, and the names correspond one by one.

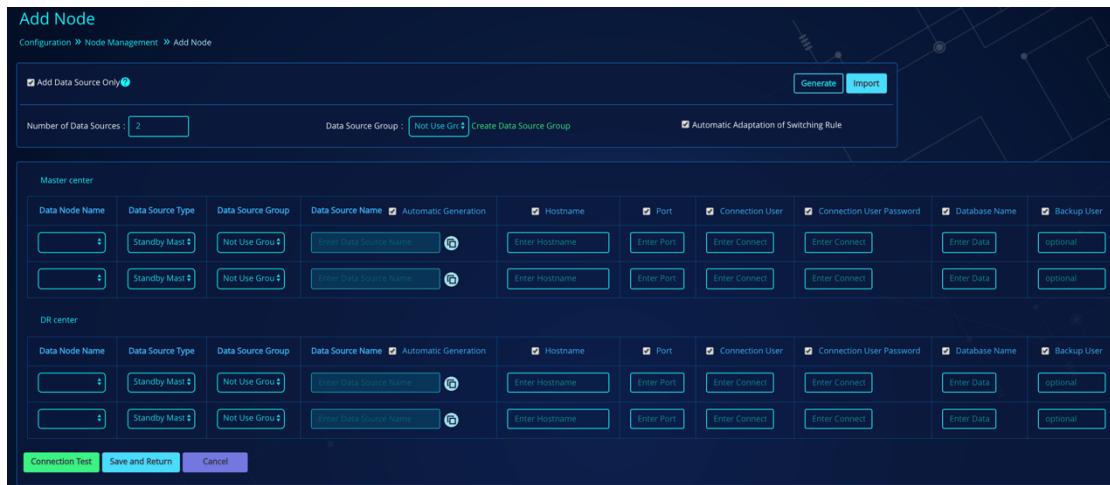
- The data node name can only be modified under the master center module, and the modification will be updated to the corresponding node in the DR center synchronously.
- It is allowed to delete data sources in any IDCs, however the deletion of data nodes must synchronize with the master center the DR center, that is, separate deletion of data nodes of DR center is not allowed.
- Check "Automatically build DR relation" to automatically set up a replication relation between the corresponding data nodes of the two added IDCs.



Data Node Name	Data Source Type	Data Source Group	Data Source Name	<input checked="" type="checkbox"/> Automatic Generation	<input checked="" type="checkbox"/> Hostname	<input checked="" type="checkbox"/> Port	<input checked="" type="checkbox"/> Connection User	<input checked="" type="checkbox"/> Connection User Password	<input checked="" type="checkbox"/> Database Name	<input checked="" type="checkbox"/> Backup User
dn_08	Active Master	Not Use Group	Enter Data Source Name	<input type="button" value="..."/>	Enter Hostname	Enter Port	Enter Connect	Enter Connect	Enter Data	optional
dn_08	Standby Mast	Not Use Group	Enter Data Source Name	<input type="button" value="..."/>	Enter Hostname	Enter Port	Enter Connect	Enter Connect	Enter Data	optional
dn_09	Active Master	Not Use Group	Enter Data Source Name	<input type="button" value="..."/>	Enter Hostname	Enter Port	Enter Connect	Enter Connect	Enter Data	optional
dn_09	Standby Mast	Not Use Group	Enter Data Source Name	<input type="button" value="..."/>	Enter Hostname	Enter Port	Enter Connect	Enter Connect	Enter Data	optional

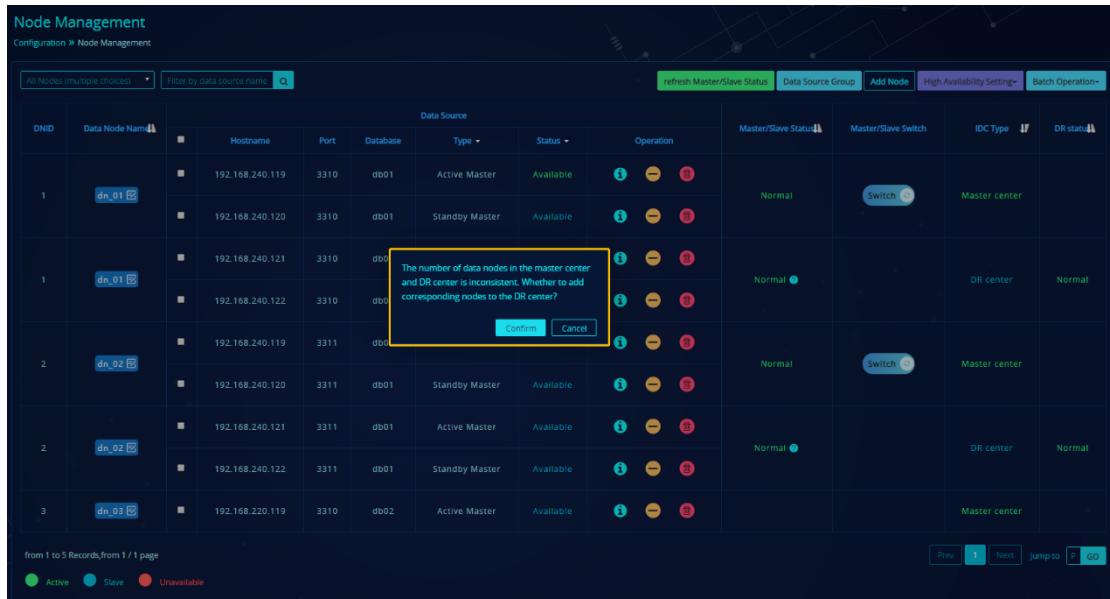
## (2) Add data source only:

- Check "Add data source only" to add data sources to data nodes that already exist in the master center or DR center.
- Addition of only data sources will not affect the logical structure of data nodes in two IDCs. Therefore, when only adding data sources, the number of data sources added under any IDC module will not be limited. For example, if you want to add data sources to a data node in the master center separately, you can delete the automatically generated data source under the DR center module.



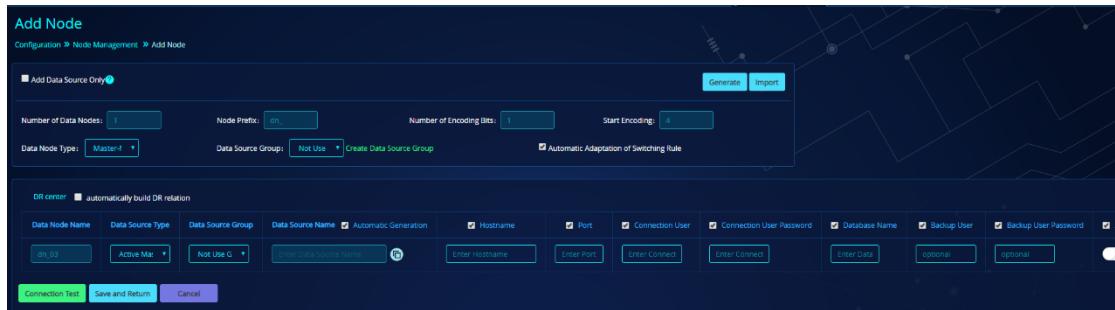
### (3) Add missing data nodes:

Click [Add Node]. When the master center is the current active center, management platform will check whether the current logical structure of the two IDCs is consistent. If it is found that there are more data nodes in the master center than in the DR center, management platform will ask users to add corresponding nodes for the DR center, otherwise the function Add Node cannot be used. This function is used to separately supplement the data nodes which correspond to the production environment for the DR center after a DR center is added to an existing production environment, which can avoid the situation where the node information is manually modified in the ConfigDB.



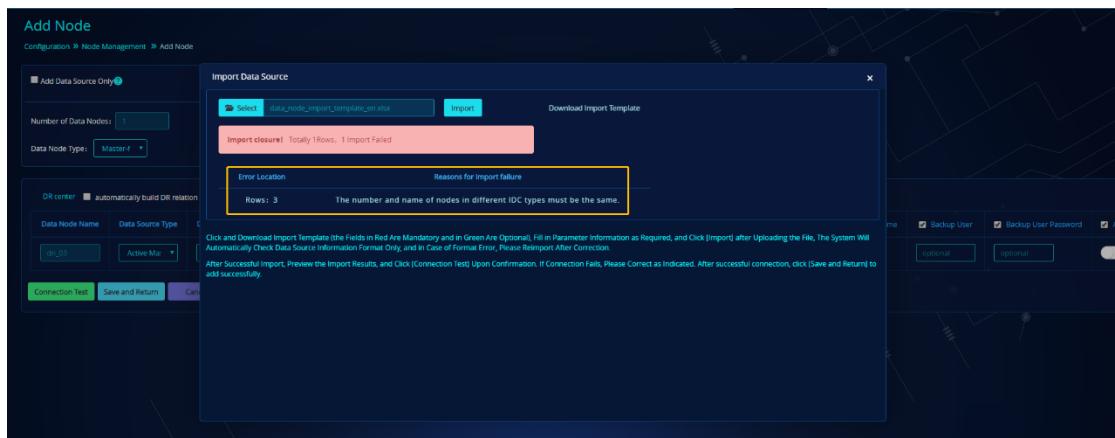
Click [Confirm], then management platform will automatically generate missing nodes in the DR center of the same node type as that of the master center. You can modify the

"Data Node Type" or "Data Source Group" and regenerate it, but the data node name and number cannot be modified which are consistent with the master center.



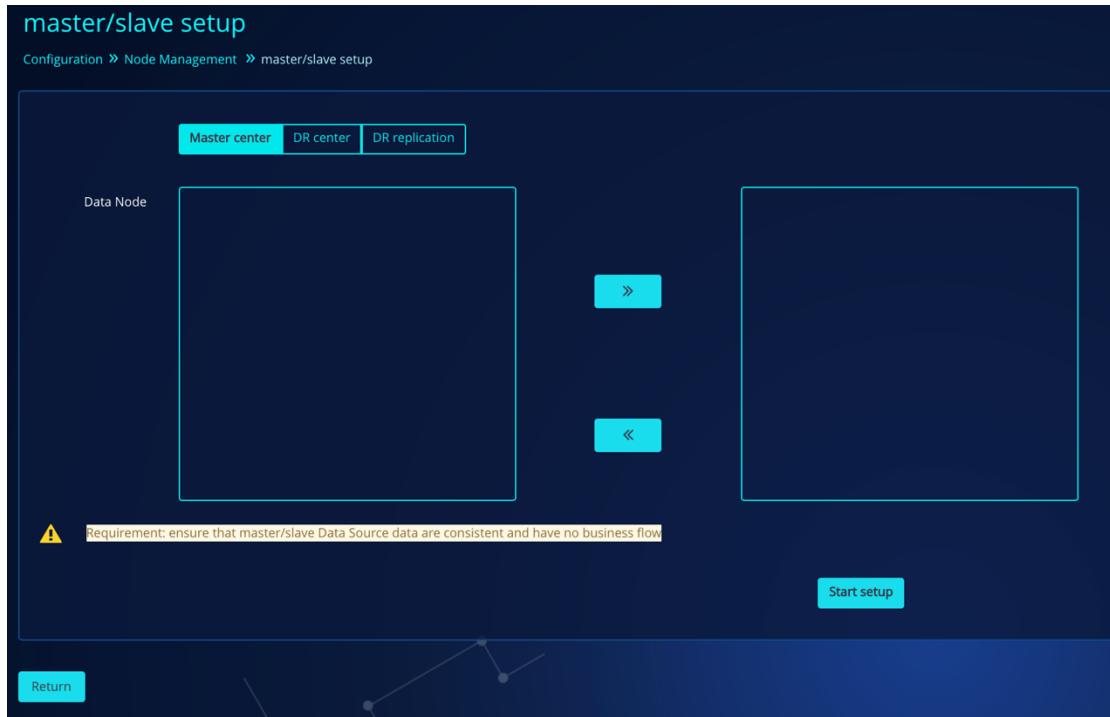
#### (4) Import function:

When using the Import function, you need to ensure that the logical structure of the imported data nodes in the master center and the DR center is the same, that is, the number of data nodes is the same, and the names correspond to each other. Otherwise, the import will fail.



##### 4.3.1.2. Master-slave setup

Master-slave setup can set up replication relations not only for the data sources in a single IDC that have not yet set up a replication relation, but also for the master data sources of the two IDCs.



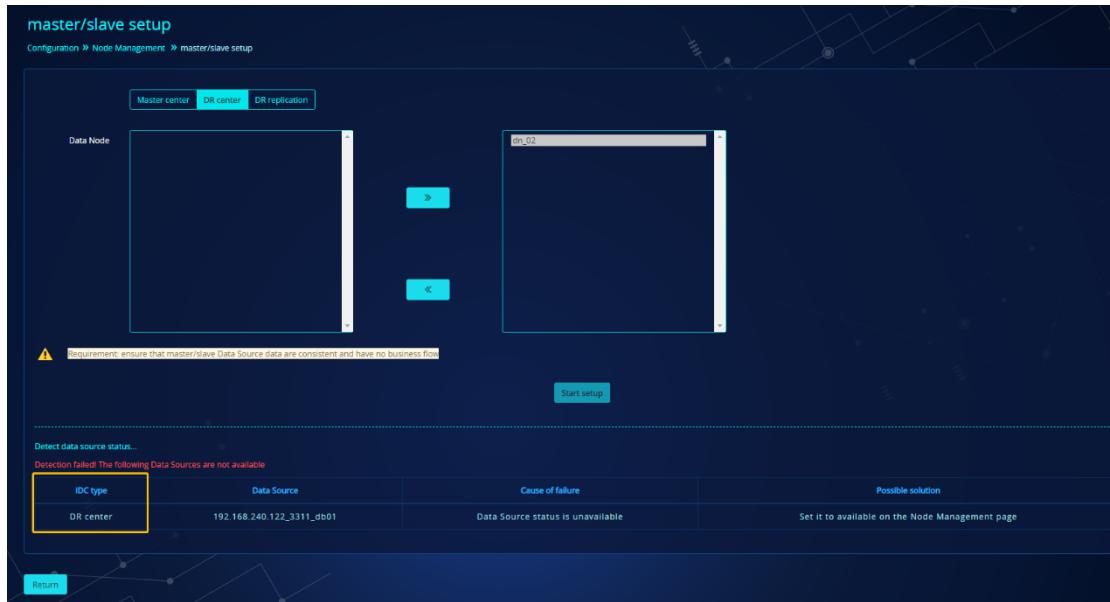
## (1) Setup instructions:

(The setup instructions only explain the differences between DR centers. For detailed setup instructions and steps, please refer to the [Distributed Transactional Database HotDB Server - V2.5.3 \[Management Platform\] Function Manual](#).)

- The data source you choose to set up cannot be configured with or have multi-master replication relations. To set up disaster recovery relations, it is required that the master data source in the DR center cannot be a slave data source in a set up replication relation; if the slave data source in the DR center has a set up replication relation, it can only be a replication relation with the master data source.
- The configuration parameters of the data source my.cnf selected for setup are correct (for those copied based on binlog and GTID, open the corresponding parameters respectively). To set up a disaster recovery relation, the data source must have GTID enabled.

## (2) Others:

- IDC type is added uniformly to the error message of setting up to identify which IDC the data source that reports errors belongs to.



- If the master center fails, the DR center will be switched to the be current active center, and the master-slave setup function will not provide the setup of disaster recovery relations.

#### 4.3.1.3. Switching rules

Switching rules configuration can configure the high availability switching priority for the data sources of the two IDCs. The compute node will detect the availability of the data source regularly. If the data source is detected to fail, the compute node will automatically switch to the standby data source according to the configured switching rules.

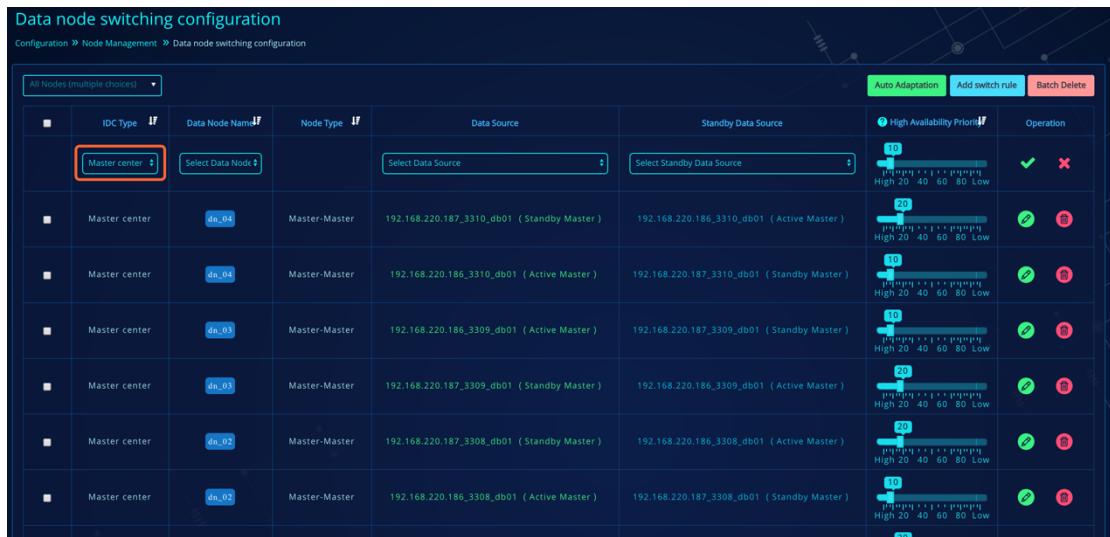
If the data source fails and no relevant switching rules are configured, automatic failover cannot be performed.

##### (1) Add switching rules:

Select the IDC type and the switching rules will be automatically adapted for the data nodes in a single IDC. The automatic adaptation rules remain unchanged.

### Data node switching configuration

Configuration > Node Management > Data node switching configuration

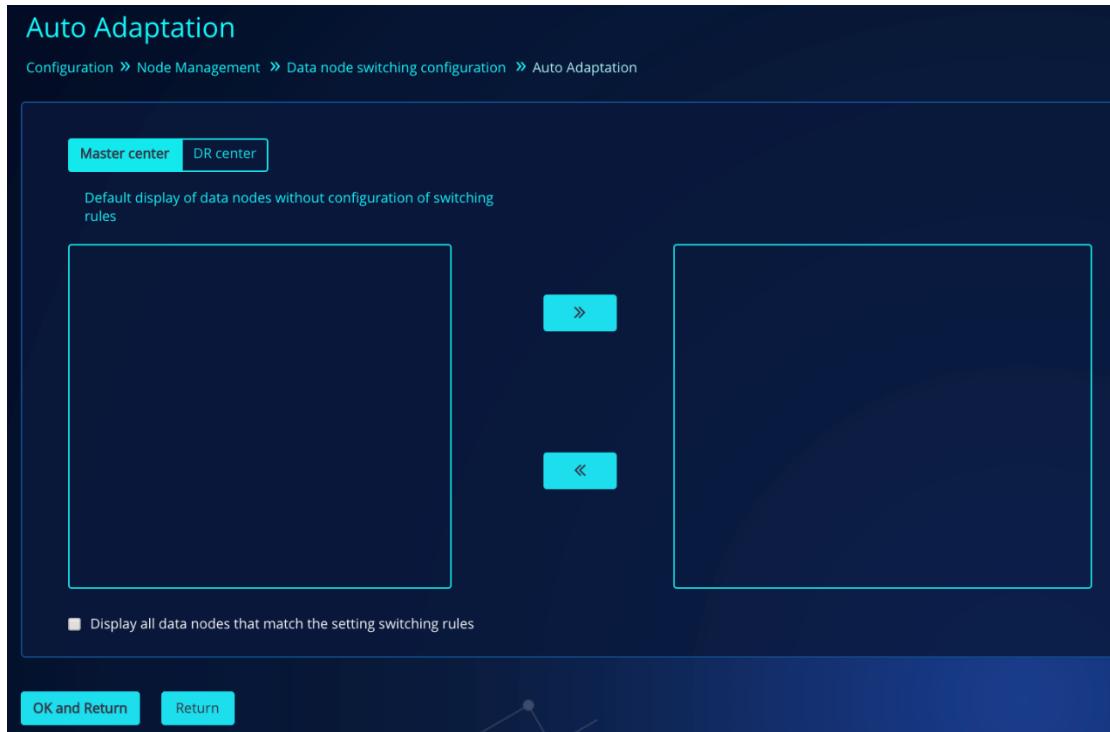


IDC Type	Data Node Name	Node Type	Data Source	Standby Data Source	High Availability Priority	Operation
Master center	dn_04	Master-Master	192.168.220.187_3310_db01 (Standby Master)	192.168.220.186_3310_db01 (Active Master)	10 20 10 10 20 20 10	✓ ✎
Master center	dn_04	Master-Master	192.168.220.186_3310_db01 (Active Master)	192.168.220.187_3310_db01 (Standby Master)	10 20 10 10 20 20 10	✎ ✎
Master center	dn_03	Master-Master	192.168.220.186_3309_db01 (Active Master)	192.168.220.187_3309_db01 (Standby Master)	10 20 10 10 20 20 10	✎ ✎
Master center	dn_03	Master-Master	192.168.220.187_3309_db01 (Standby Master)	192.168.220.186_3309_db01 (Active Master)	10 20 10 10 20 20 10	✎ ✎
Master center	dn_02	Master-Master	192.168.220.187_3308_db01 (Standby Master)	192.168.220.186_3308_db01 (Active Master)	20 20 20 20 20 20 20	✎ ✎
Master center	dn_02	Master-Master	192.168.220.186_3308_db01 (Active Master)	192.168.220.187_3308_db01 (Standby Master)	10 20 10 10 20 20 10	✎ ✎

Select the IDC type and add switching rules for the data sources in a single IDC.

The data sources in the DR center still need to be configured with intra-IDC switching rules, because when the master center fails and the DR center is switched to be the current active center, failure to configure the switching rules will result in failover failure between data sources.

### (2) Automatic adaption:



Auto Adaptation

Configuration > Node Management > Data node switching configuration > Auto Adaptation

Master center DR center

Default display of data nodes without configuration of switching rules

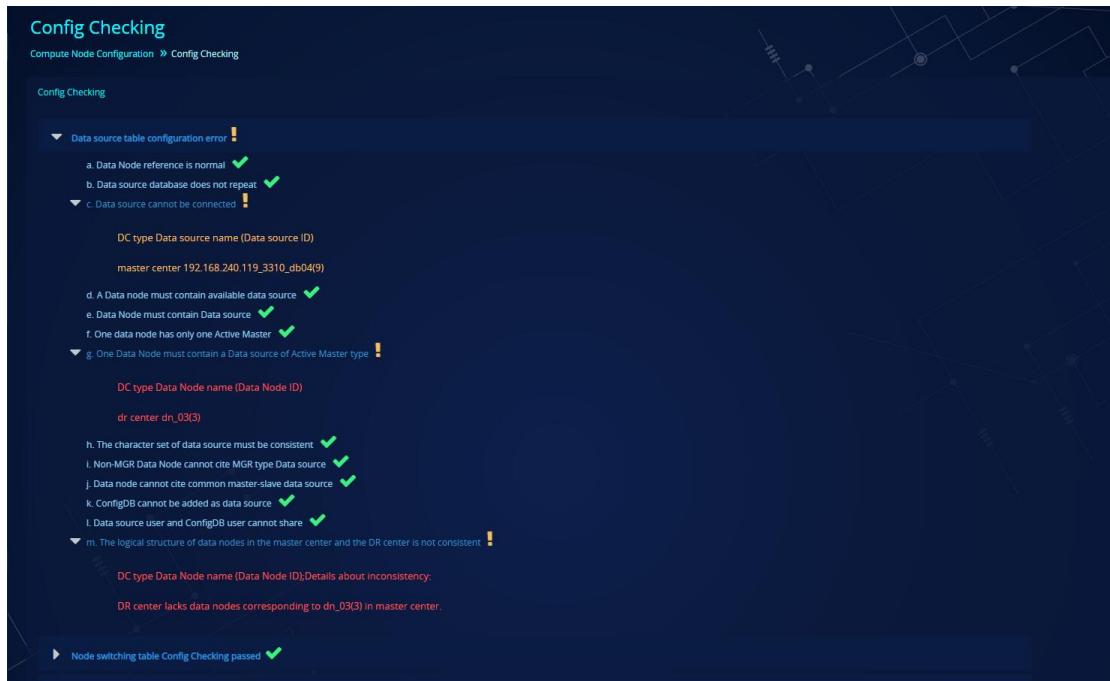
Display all data nodes that match the setting switching rules

OK and Return    Return

Select the IDC type and the switching rules will be automatically adapted for the data nodes in a single IDC. The automatic adaptation rules of the DR center remain unchanged.

### 4.3.2. Configuration checking

Configuration checking mainly provides checking function for relevant configurations of compute nodes to prevent abnormal operation caused by manual error settings or offline modification of relevant configurations of compute nodes. In the DR mode, the following adaptations are made:



- The IDC type is added to the error message related to the checking of the data source, which can clearly identify which IDC's data source fails the checking, so as to troubleshoot the problem.
- The checking item [Data source connection is normal] is changed to the checking of warning level. For example, if any data source in the master center or the DR center is connected abnormally, a warning message will appear in the configuration checking, which however will not affect the dynamic loading. When the master center or the DR center fails as the current standby center, this checking item will still check whether all data sources in the two IDCs are connected normally, which however will not affect the dynamic loading.
- If the unconnected data source causes no available nodes under the corresponding data nodes of the master center and the DR center, such as dn\_01 of the master center and dn\_01 of the DR center, an error message will appear in the checking

item [A Data Node must contain available Data Source] in the configuration checking, and the dynamic loading will not succeed. For example, if all data sources in a data node of the DR center are connected abnormally, as long as there are still available data sources in the corresponding data node of the master center, the dynamic loading will succeed, but a warning message will still appear in the configuration checking.

- An error-level checking item is added in [Data Source table configuration]: the logical structure of the data nodes in the master center and the DR center is consistent, that is, the number of data nodes in the master center and the DR center is consistent, and the names correspond one by one.
- In [ConfigDB status], [ConfigDB connection is normal] is modified as a warning-level checking item. Similar to [Data source connection is normal], check whether all ConfigDBs in the two IDCs are connected normally.
- A warning-level checking item in [ConfigDB] is added: whether the replication status between ConfigDBs is normal, that is, whether the replication status of the internal ConfigDBs of a single IDC is normal and whether the replication status of the master ConfigDB between two IDCs is normal.



### 4.3.3. Compute node parameter configuration

The compute node parameter configuration can configure the server.xml parameters of two IDCs in a visualized manner.

**Parameter Configuration**

Configuration > Parameter Configuration

Compute Node Cluster	DR-TEST	<input type="checkbox"/> Automatically synchronize to other compute nodes
Name	HotDB_01 (192.168.220.186 Current Master)	
DR mode	Master center	
Address of master ConfigDB of master center	jdbc:mysql://192.168.220.186:3306/hotdb_config	
User name of master ConfigDB of master center	hotdb_config	
Password of master ConfigDB of master center	hotdb_config	
Address of slave ConfigDB of master center	jdbc:mysql://192.168.220.187:3306/hotdb_config	
User name of slave ConfigDB of master center	hotdb_config	
Password of slave ConfigDB of master center	hotdb_config	
Address of master ConfigDB of DR center	jdbc:mysql://192.168.220.188:3306/hotdb_config	
User name of master ConfigDB of DR center	hotdb_config	
Password of master ConfigDB of DR center	hotdb_config	
Address of slave ConfigDB of DR center	jdbc:mysql://192.168.220.189:3306/hotdb_config	
User name of slave ConfigDB of DR center	hotdb_config	
Password of slave ConfigDB of DR center	hotdb_config	
ConfigDB is MGR or not	No	

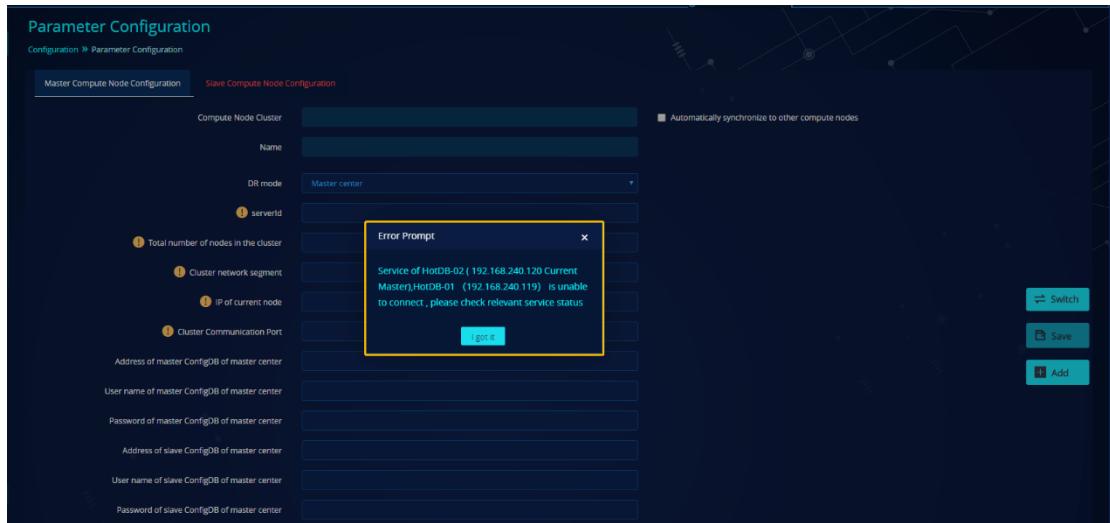
[switch to the DR center](#) [Switch](#) [Save](#) [Add](#)

- The server.xml parameter information of the master compute node of the current active center is displayed by default.
- The compute node parameters in the current active center are switched and checked through the TAB page.
- Click [switch] to switch to the parameter configuration page of the other IDC.
- If the compute node fails or the connection failure of compute nodes due to the IDC switching, the parameter configuration page of the compute node cannot be displayed.

**Parameter Configuration**

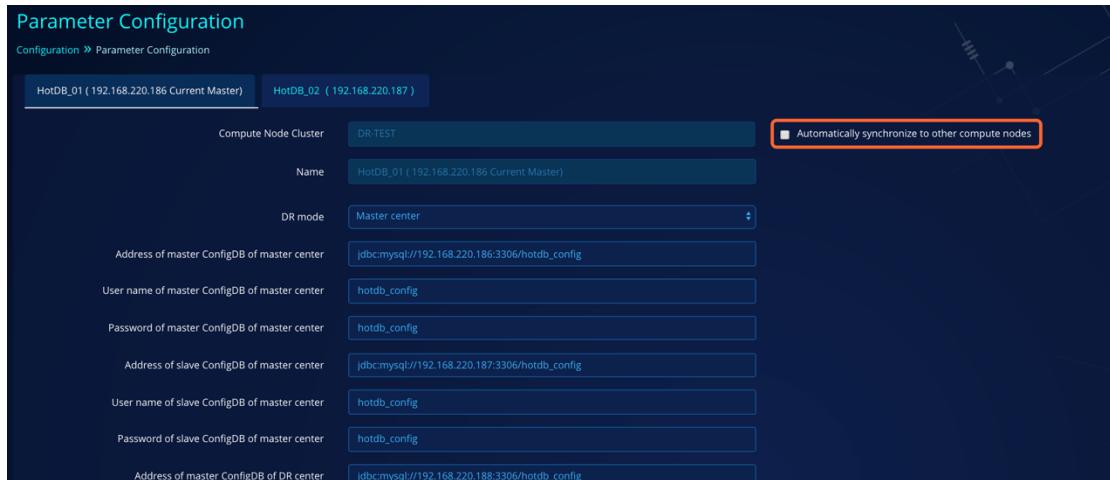
Configuration > Parameter Configuration

Compute Node Cluster	DR-TEST	<input checked="" type="checkbox"/> Automatically synchronize to other compute nodes
Name	HotDB_01 (192.168.220.186 Current Master)	
DR mode	Master center	
Address of master ConfigDB of master center	jdbc:mysql://192.168.220.186:3306/hotdb_config	
User name of master ConfigDB of master center	hotdb_config	
Password of master ConfigDB of master center	hotdb_config	
Address of slave ConfigDB of master center	jdbc:mysql://192.168.220.187:3306/hotdb_config	
User name of slave ConfigDB of master center	hotdb_config	
Password of slave ConfigDB of master center	hotdb_config	
Address of master ConfigDB of DR center	jdbc:mysql://192.168.220.188:3306/hotdb_config	



## (1) Parameter synchronization:

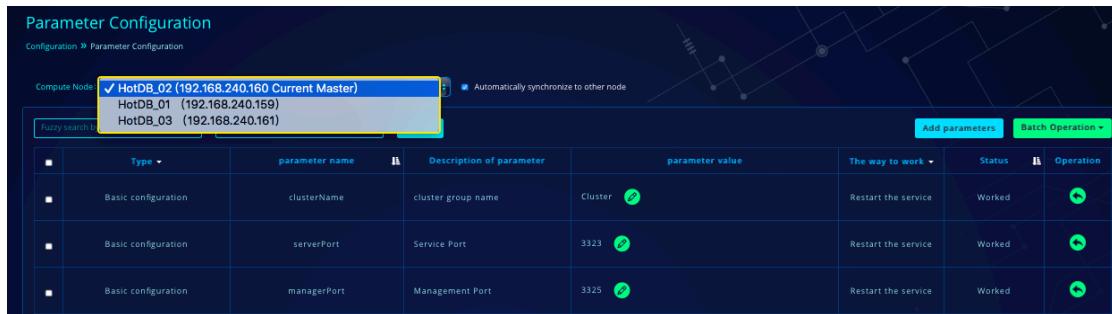
If you check "Automatically synchronize to other compute nodes", the modified or added parameters will be synchronized to other compute nodes. If the DR center is the current active center, the master compute node parameters of the DR center will be displayed and synchronized on the page by default.



Note: The seven parameters [haMode], [idcId], [idcNodeHost], [ServerId], [clusterHost], [hsaState] and [HaNodeHost] do not support the modification synchronization.

## (2) Node failure

If there is compute node failure, then the failed compute node will not be displayed in the compute node selection box.



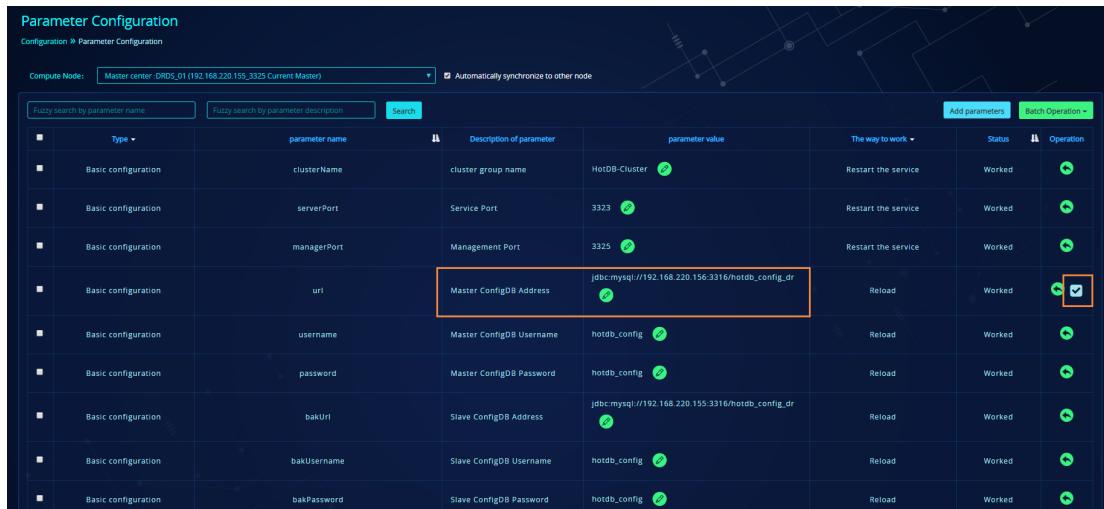
The screenshot shows the 'Parameter Configuration' page. The 'Compute Node' dropdown is set to 'HotDB\_02 (192.168.240.160 Current Master)'. Below it, there's a 'Fuzzy search by parameter name' input field. A legend indicates that a yellow box highlights the current master node. The main table lists basic configuration parameters:

Type	parameter name	Description of parameter	parameter value	The way to work	Status	Operation
Basic configuration	clusterName	cluster group name	Cluster	Restart the service	Worked	
Basic configuration	serverPort	Service Port	3323	Restart the service	Worked	
Basic configuration	managerPort	Management Port	3325	Restart the service	Worked	

### (3) Enable the ConfigDB:

When the ConfigDB mode used by the master center or DR center is master-slave or master-master, if the master ConfigDB fails, the compute node will automatically switch the connection of the ConfigDB to the slave ConfigDB or standby master ConfigDB. At this time, if the master ConfigDB failure is manually repaired, the compute node cluster will not use the master ConfigDB, you can re-enable the failed ConfigDB according to the following steps:

1. When the master ConfigDB fails, the compute node parameter configuration page is displayed as the following figure:

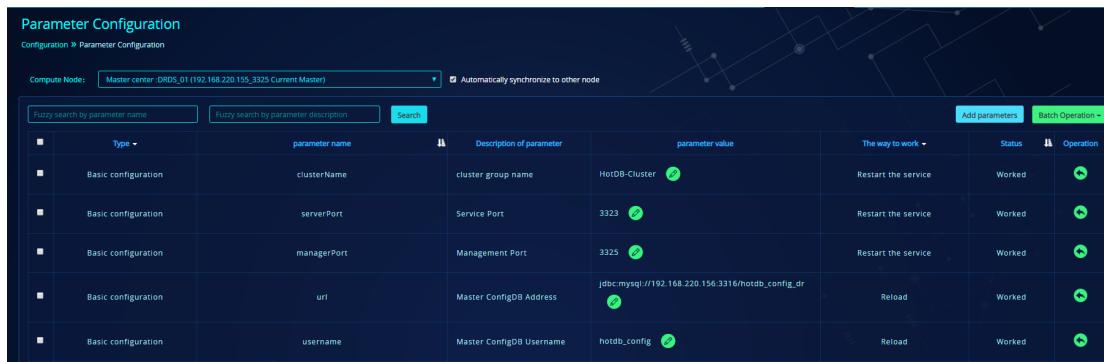


The screenshot shows the 'Parameter Configuration' page. The 'Compute Node' dropdown is set to 'Master center : DRDS\_01 (192.168.220.155\_3325 Current Master)'. The 'url' row is highlighted with a yellow box. The 'Master ConfigDB Address' field contains 'jdbc:mysql://192.168.220.156:3316/hotdb\_config\_dr'. The 'Operation' column for this row has a checked checkbox. The main table lists basic configuration parameters:

Type	parameter name	Description of parameter	parameter value	The way to work	Status	Operation
Basic configuration	clusterName	cluster group name	HotDB-Cluster	Restart the service	Worked	
Basic configuration	serverPort	Service Port	3323	Restart the service	Worked	
Basic configuration	managerPort	Management Port	3325	Restart the service	Worked	
Basic configuration	url	Master ConfigDB Address	jdbc:mysql://192.168.220.156:3316/hotdb_config_dr	Reload	Worked	
Basic configuration	username	Master ConfigDB Username	hotdb_config	Reload	Worked	
Basic configuration	password	Master ConfigDB Password	hotdb_config	Reload	Worked	
Basic configuration	bakUrl	Slave ConfigDB Address	jdbc:mysql://192.168.220.155:3316/hotdb_config_dr	Reload	Worked	
Basic configuration	bakUsername	Slave ConfigDB Username	hotdb_config	Reload	Worked	
Basic configuration	bakPassword	Slave ConfigDB Password	hotdb_config	Reload	Worked	

2. If it is determined that the master ConfigDB has returned to normal and consistent with the data of the slave ConfigDB and is up-to-date, click the enable button on the "Compute node parameter configuration" page to enable the master ConfigDB.

3. Click the enable button , and then click the [Dynamic loading] button to re-enable the master ConfigDB.

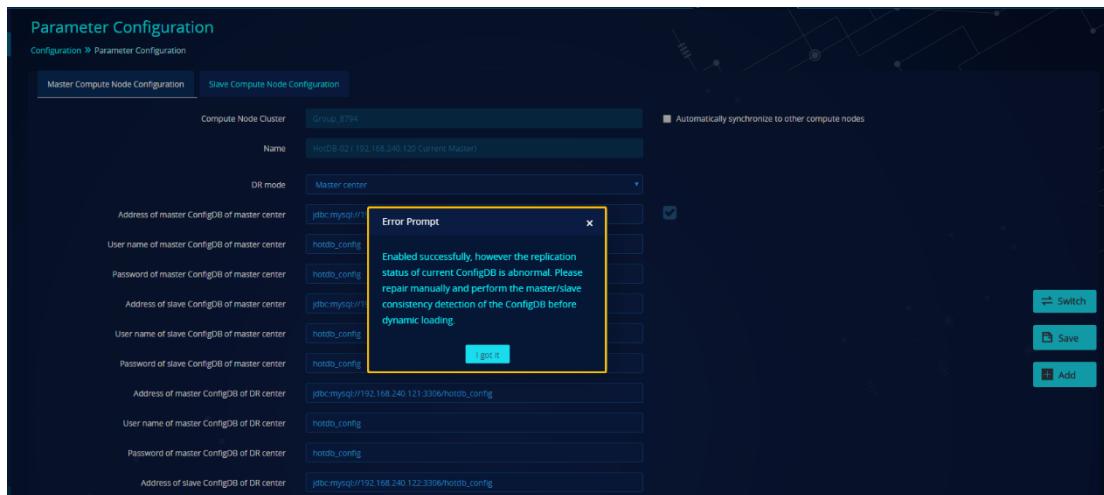


The screenshot shows a table of parameter configurations:

Type	parameter name	Description of parameter	parameter value	The way to work	Status	Operation
Basic configuration	clusterName	cluster group name	HotDB-Cluster	Restart the service	Worked	
Basic configuration	serverPort	Service Port	3323	Restart the service	Worked	
Basic configuration	managerPort	Management Port	3325	Restart the service	Worked	
Basic configuration	url	Master ConfigDB Address	jdbc:mysql://192.168.220.156:3316/hotdb_config_dr	Reload	Worked	
Basic configuration	username	Master ConfigDB Username	hotdb_config	Reload	Worked	

## Notes:

- When the DR center is the current standby center, after the master ConfigDB fails, the compute node will set both the master and slave ConfigDBs unavailable.
- When the DR center is the current standby center, when enabling the slave ConfigDB, you need to confirm that the current active ConfigDB is available; otherwise, the slave ConfigDB cannot be enabled, that is, you can enable the slave ConfigDB only after the master ConfigDB is enabled.
- If you want to enable either ConfigDB in the two IDCs, you need to ensure that the current replication status inside the IDC and between the IDCs is normal. If any of the replication status is abnormal, there will be a warning message though the dynamic loading may succeed.



The screenshot shows the Master Compute Node Configuration tab. A modal window titled "Error Prompt" contains the message:

Enabled successfully, however the replication status of current ConfigDB is abnormal. Please repair manually and perform the master/slave consistency detection of the ConfigDB before dynamic loading.

## (4) New parameters:

### 1. Configure the DR center ConfigDB

Address of master ConfigDB of master center	jdbc:mysql://192.168.220.186:3306/hotdb_config
User name of master ConfigDB of master center	hotdb_config
Password of master ConfigDB of master center	hotdb_config
Address of slave ConfigDB of master center	jdbc:mysql://192.168.220.187:3306/hotdb_config
User name of slave ConfigDB of master center	hotdb_config
Password of slave ConfigDB of master center	hotdb_config
Address of master ConfigDB of DR center	jdbc:mysql://192.168.220.188:3306/hotdb_config
User name of master ConfigDB of DR center	hotdb_config
Password of master ConfigDB of DR center	hotdb_config
Address of slave ConfigDB of DR center	jdbc:mysql://192.168.220.189:3306/hotdb_config
User name of slave ConfigDB of DR center	hotdb_config
Password of slave ConfigDB of DR center	hotdb_config

- Add the ConfigDB configuration of the DR center, including the address of the master-slave ConfigDB of DR center and the corresponding user name and password. The master center and the DR center refer to the IDC type, which does not change with the IDC status.
- All compute nodes in the two IDCs need to be configured with the master center and DR center ConfigDB configuration in server.xml.
- (The ConfigDB parameters that do not have the IDC type identifications represent the ConfigDB parameters of the master center)

(The ConfigDB parameters that do not have the IDC type identifications represent the ConfigDB parameters of the master center)

```
<propertyname="url">jdbc:mysql://192.168.220.213:3316/hotdb_config</property><!-- Master
ConfigDB address, which needs to be the real IP address of the ConfigDB service-->
<propertyname="username">hotdb_config</property><!--Master ConfigDB username-->
<propertyname="password">hotdb_config</property><!--Master ConfigDB password-->
<propertyname="bakUrl">jdbc:mysql://192.168.220.214:3316/hotdb_config</property><!--Slave
ConfigDB address, which needs to be the real IP address of the ConfigDB service-->
```

```

<property name="bakUsername">hotdb_config</property><!--Slave ConfigDB username-->

<property name="bakPassword">hotdb_config</property><!--Slave ConfigDB password-->

<property name="drUrl">jdbc:mysql://192.168.220.217:3316/hotdb_config</property><!--ConfigDB
address in DR center-->

<property name="drUsername">hotdb_config</property><!--ConfigDB username in DR center-->

<property name="drPassword">hotdb_config</property><!--ConfigDB password in DR center-->

<property name="drBakUrl">jdbc:mysql://192.168.220.218:3316/hotdb_config</property><!--Slave
ConfigDB address in DR center-->

<property name="drBakUsername">hotdb_config</property><!--Slave ConfigDB username in DR
center-->

<property name="drBakPassword">hotdb_config</property><!--Slave ConfigDB password in DR
center-->

```

## 2. haMode adds high availability in DR mode

In the DR mode, regardless of whether the high-availability or single-node compute node mode is used, set this parameter as 2 for the master center and 3 for the DR center. If the multi-node mode is used, set this parameter as 4 for the master center and 5 for the DR center.

Master center:

```
<property name="haMode">2</property><!--High-availability mode, 0:HA, 1:Cluster, 2:HA in
master center, 3:HA in DR center, 4: Cluster in master center, 5: Cluster in DR center -->
```

DR center:

```
<property name="haMode">3</property><!--High-availability mode, 0:HA, 1:Cluster, 2:HA in
master center, 3:HA in DR center, 4: Cluster in master center, 5: Cluster in DR center -->
```

Note: DR mode does not support compute nodes in cluster mode.

## 3. Add the ID of IDC and the compute node configuration of another IDC

- idcId and idcNodeHost should be configured in DR mode.

- ID of IDC: The user needs to set an ID value to uniquely identify the type of IDC. Currently by default 1 is for the master center and 2 for the DR center.
- Compute node configuration of another IDC: Connection information of all compute nodes in another IDC. Format: IP:PORT

For example, the master compute node connection information of the master center is 192.168.220.213:3325, and the slave compute node connection information is 192.168.220.214:3325; the master compute node connection information of the DR center is 192.168.220.217:3325, and the slave compute node connection information is 192.168.220.218:3325. In server.xml of the master center, the configuration is as follows:

Slave compute node in DR center:

```
<property name="idcId">1</property><!--ID of IDC, 1:master center, 2:DR center-->
<property name="idcNodeHost">192.168.220.213:3325,192.168.220.214:3325</property><!--
Computer node info in the other IDC -->
```

Slave compute node in DR center:

```
<property name="idcId">2</property><!--ID of IDC, 1:master center, 2:DR center-->
<property name="idcNodeHost">192.168.220.213:3325,192.168.220.214:3325</property><!--
Computer node info in the other IDC)-->
```

## 4.4. Monitoring

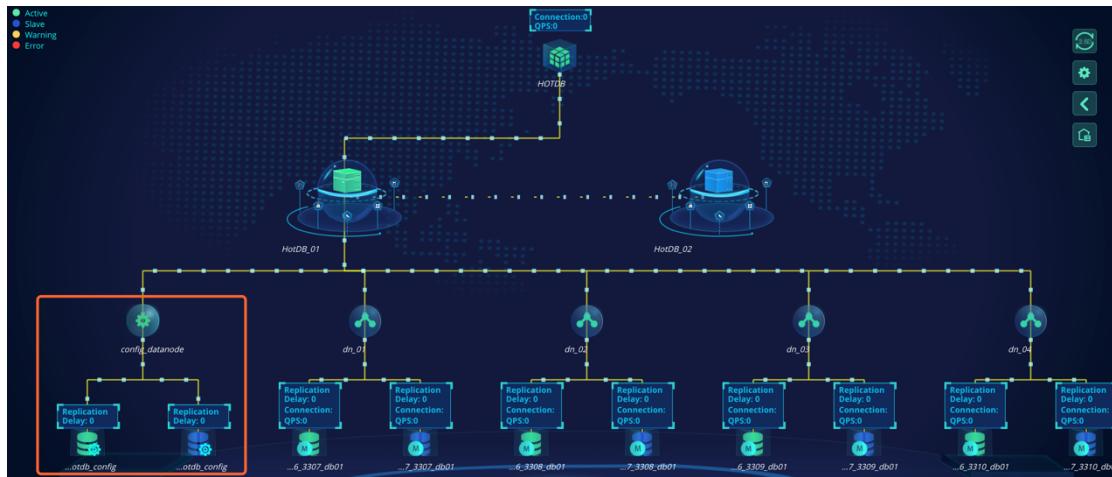
### 4.4.1. Logic topological graph

The Logic topological graph not only provides the topological graph of the current active center (called the topology of master center), but also provides the topology of DR center which monitors the component status, disaster recovery status, disaster recovery latency and other information of the two IDCs.

#### (1) Topology of master center and 2.5D topology of master center:

1. Visualized monitoring of ConfigDB is added.

- The ConfigDB components are always displayed on the far left of the topology of master center.



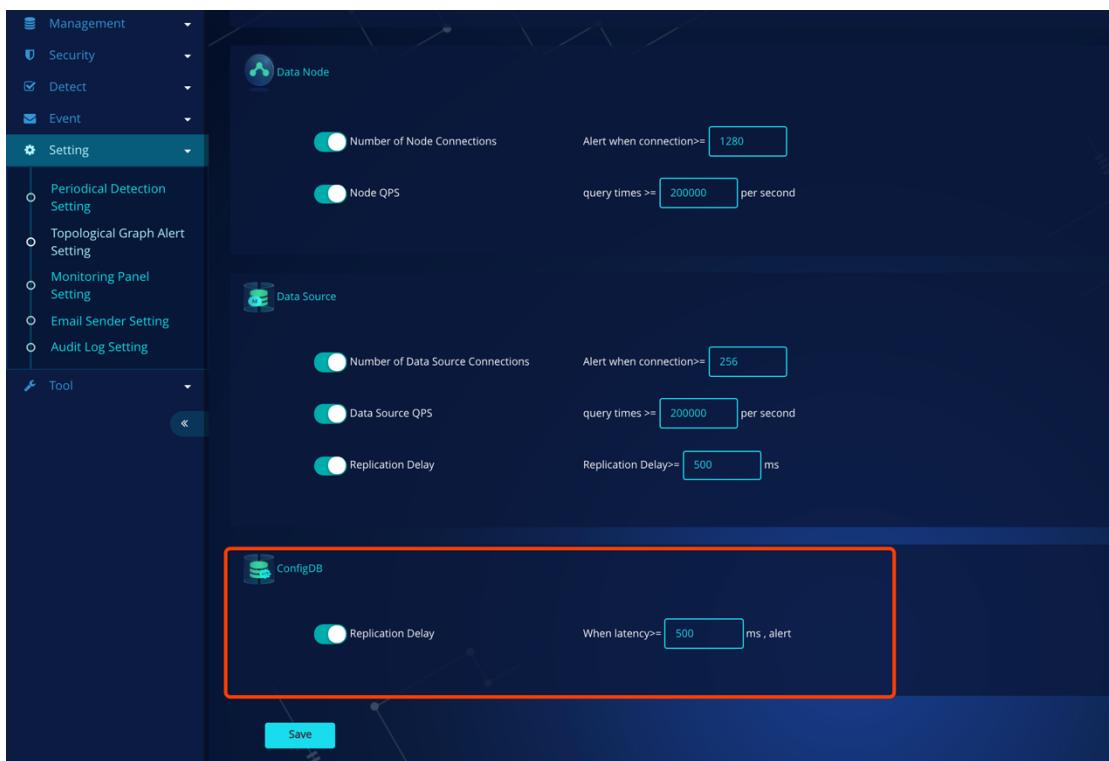
The ConfigDB components are also added in the 2.5D topology of master center, with the same functions as that of the topology of master center.



-  is a non-status icon, which connects one or two ConfigDBs , respectively representing a single node ConfigDB and a master-master ConfigDB. In single-IDC mode, if three or more ConfigDBs are connected, it represents the

ConfigDB in MGR mode.  will be in orange when all ConfigDBs are not available.

- Similar to the data source, icon of ConfigDB in red means a failure exists; moving the cursor over the red icon, the failure cause can be displayed. Orange means an abnormal replication status exists, moving the cursor over the orange icon, the abnormality cause can be displayed.
- The monitoring information of "Replication latency" is displayed in the ConfigDB. Similar to the data source, the alert threshold of replication latency can be set under ConfigDB module in "Setting → Topological Graph Alert Setting".



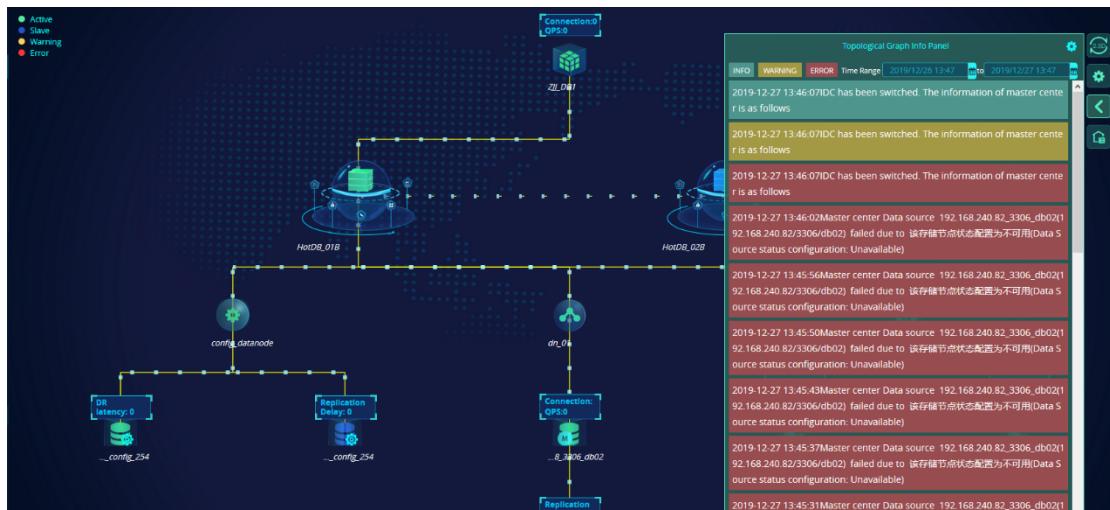
## 2. Monitoring information of ConfigDB is added to the Topological Graph Info Panel

- Warning-level monitoring information is added: The replication status of ConfigDB is abnormal and ConfigDB replication latency exceeds the threshold.

## 3. IDC switching message is added in the Topological Graph Info Panel

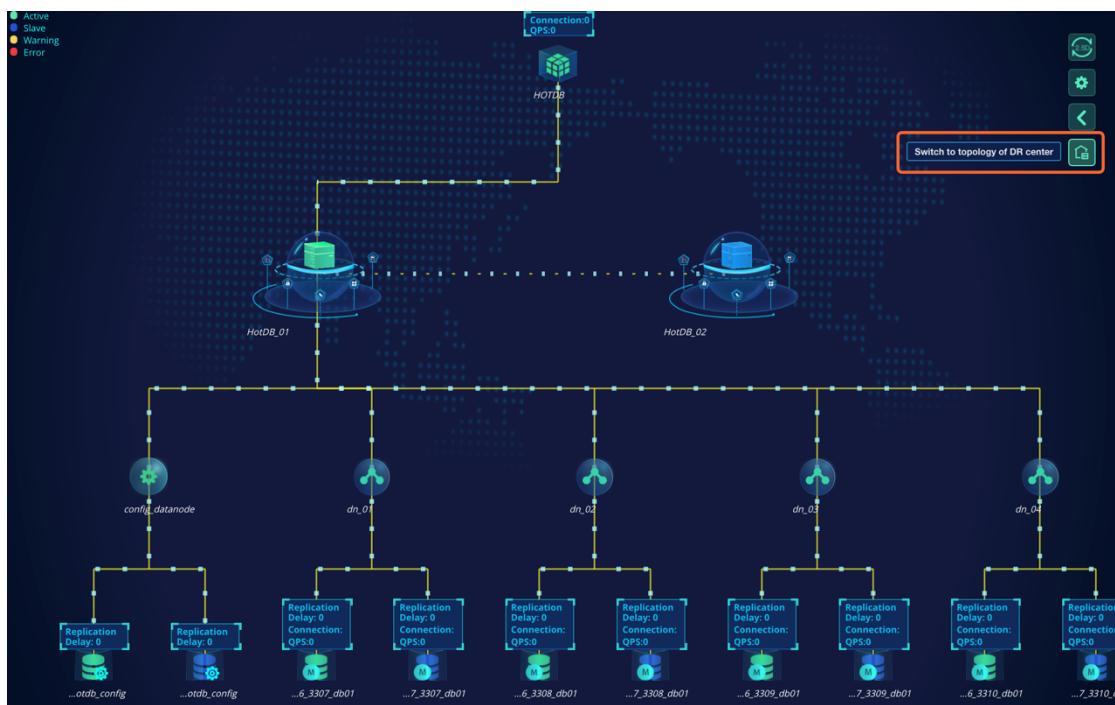
- When IDC switching occurs, a message which reports the IDC switching will appear on the Topological Graph Info Panel of the topology of master center under the three levels of info, warning and error.

- For example, when the master center is switched to the DR center, the message about the IDC switching is displayed as: At 2019-11-22 12:23:32 IDC switching occurs. The following messages are of the master center.
- The IDC switching message can be used as a dividing line of history information. In this example, the history information before the IDC switching message is of the master center, and the history information after the IDC switching message is of the DR center.



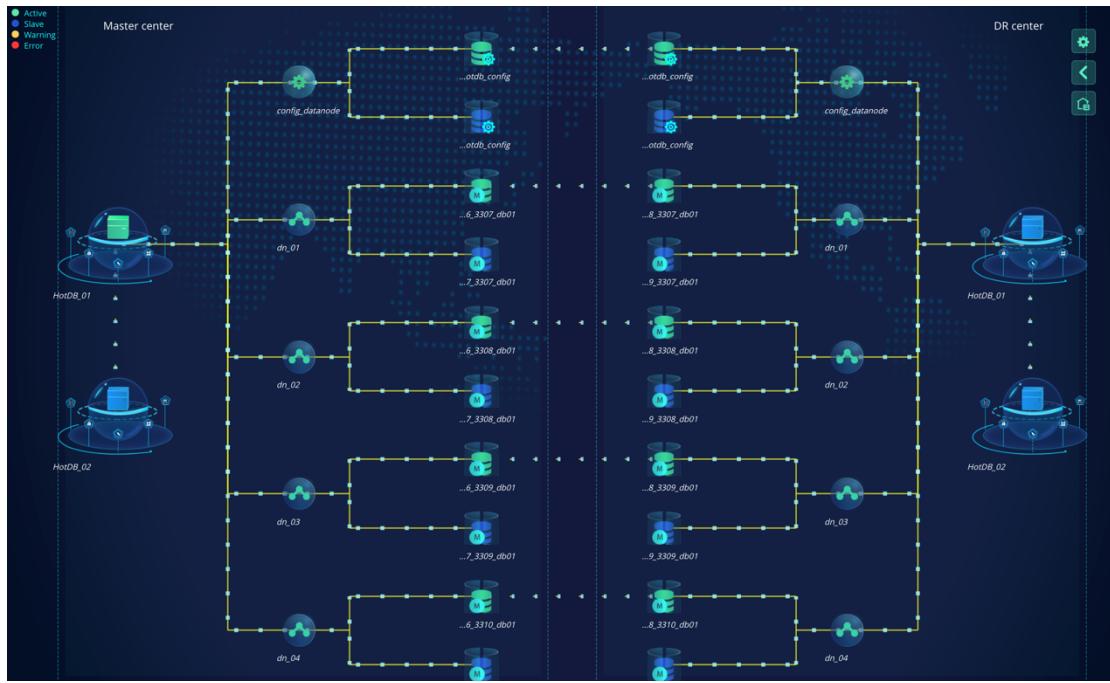
## (2) Topology of DR center:

- On the topology of master center or the 2.5D topology of master center page, click  to switch to the topology of DR center; click again to switch back to the topology of master center or 2.5D topology of master center page.
- The icon  means that in the topology of DR center, the number of components in the DR center currently in error status and not repaired is also the number of messages of errors that are still not repaired in the Topological Graph Info Panel, including the number of components that cannot be connected and the abnormal replication status of the data source or ConfigDB between the two IDCs.

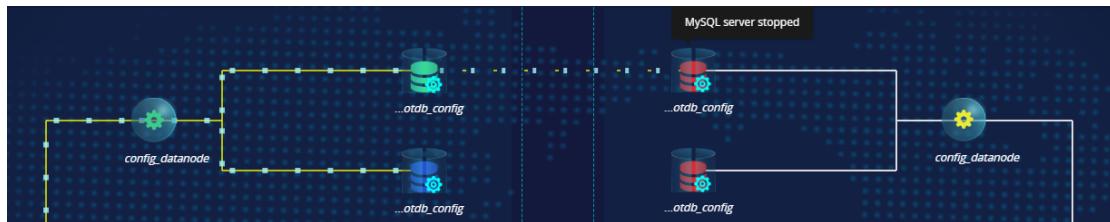


If the master center fails and the DR center is switched to be the current active center, the topology of master center will be replaced by the topology of DR center, and the icon of the topology of DR center will still display the current error number of the DR center.

- In the topology of DR center, there are six layers of components from left to right. Layers 1-3 from left to right represent the compute node, data node (ConfigDB) and data source (ConfigDB) of the master center in turn. Layers 4-6 represent the data source (ConfigDB), data node (ConfigDB) and compute node of the DR center in turn.

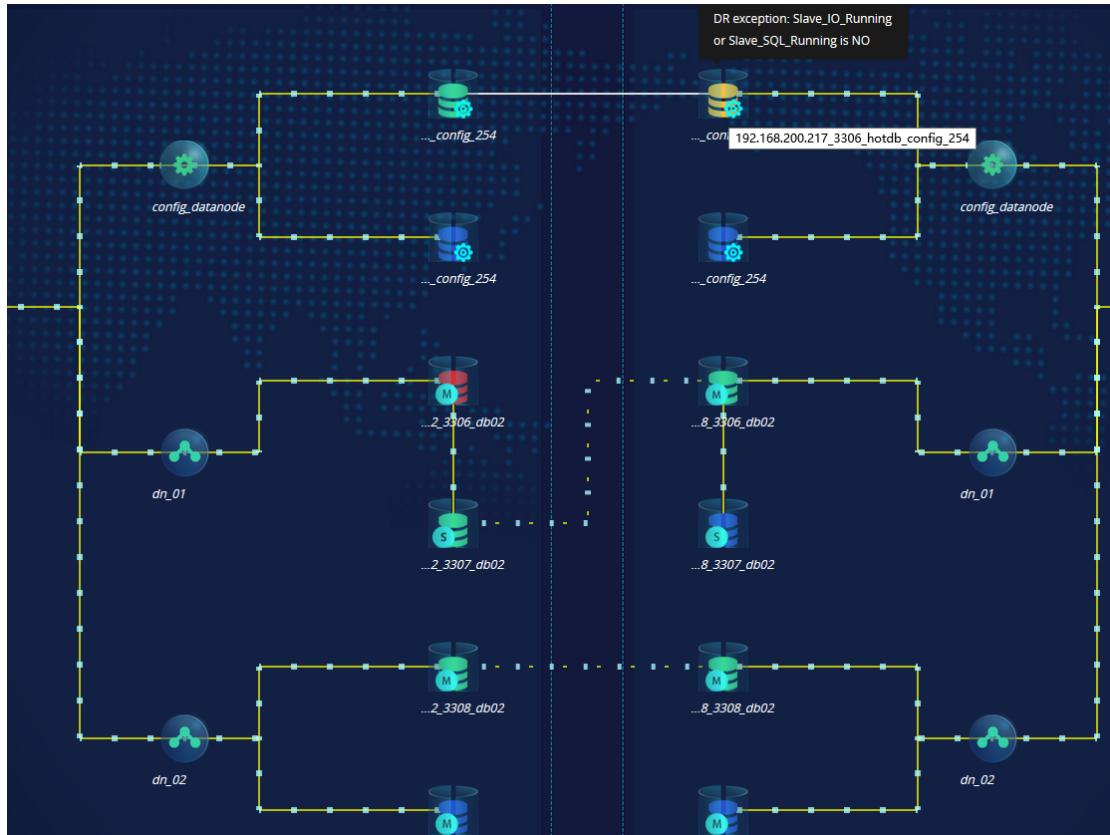


- Edges inside one IDC in the topological graph are consistent with the topology of master center. If the master ConfigDB or the master data source service of the DR center is abnormal, all data sources under the corresponding data nodes of the DR center are also set as red and unavailable. The edge is set as gray.

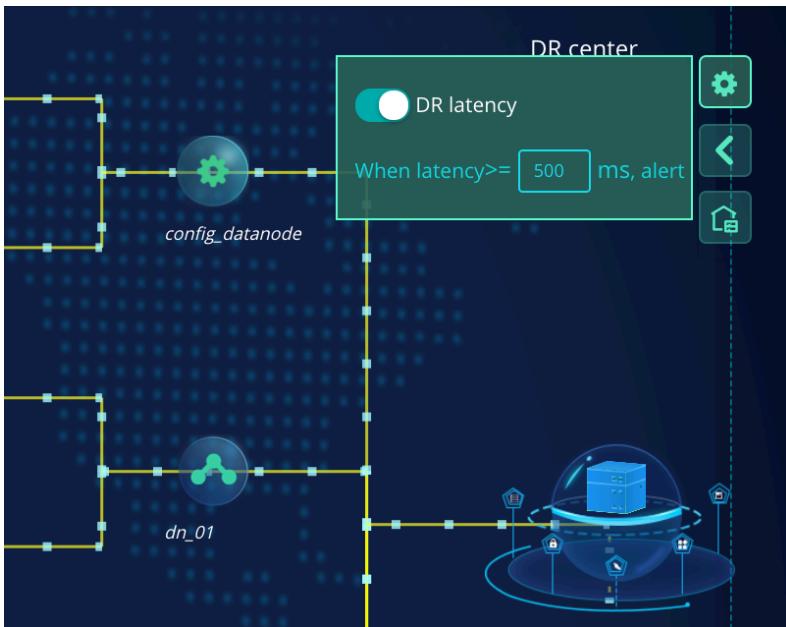


The graph shows the abnormal replication ConfigDB service in the DR center.

- Edges between two IDCs, that is, the edges of the current active data sources between the master center and the DR center, indicates the disaster recovery status. If the DR center status is abnormal, edges between the two IDCs are set as gray.

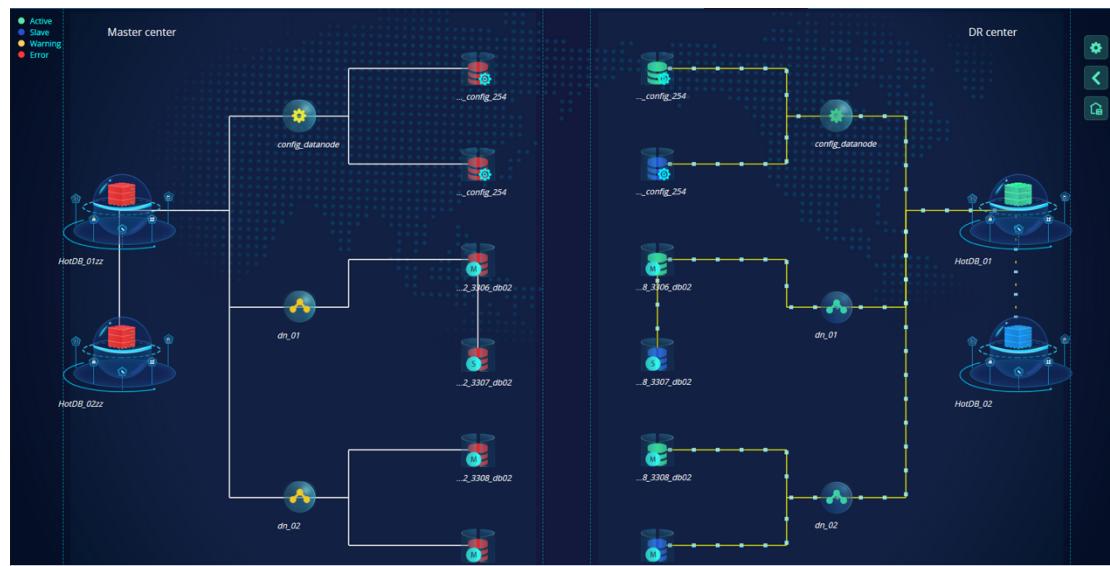


- The data sources with replication latency in the master center and DR center will display monitoring information of the replication latency. If the replication latency is 0, it will not be displayed. Same as that of the topology of master center, the alert threshold for the replication latency is set under the ConfigDB or data source module in "Setting —> Topological Graph Alert Setting".
- The current active compute node in the DR center will display monitoring information of the disaster recovery latency. The alert threshold for the disaster recovery latency of the ConfigDB and the data source is set uniformly in  on the topology of DR center page.

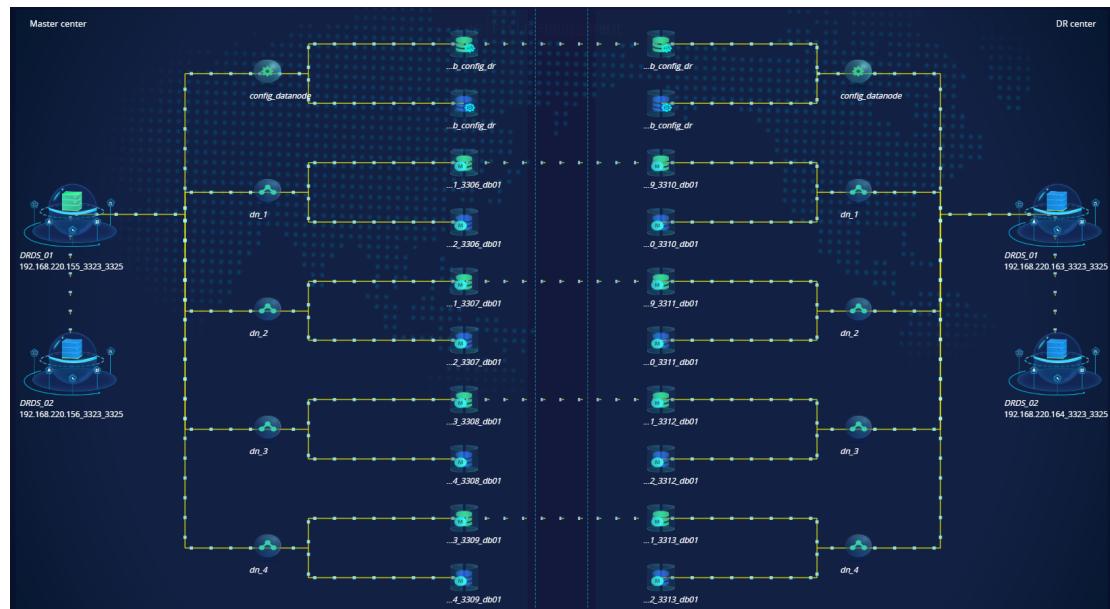


- The display order of the ConfigDB and the data source from top to bottom is: the ConfigDB at the top, the data source failure (in red), the data source warning (in yellow), the data source with the replication latency warning or the disaster recovery latency warning, and the normal node at the bottom.
- If the master center fails and the DR center is switched to be the current active center, the components (compute node, ConfigDB and data source) of the master center are in red and in unavailable status and the edge is gray on the Topology of DR center; the components of the DR center are displayed based on the actual operation status.

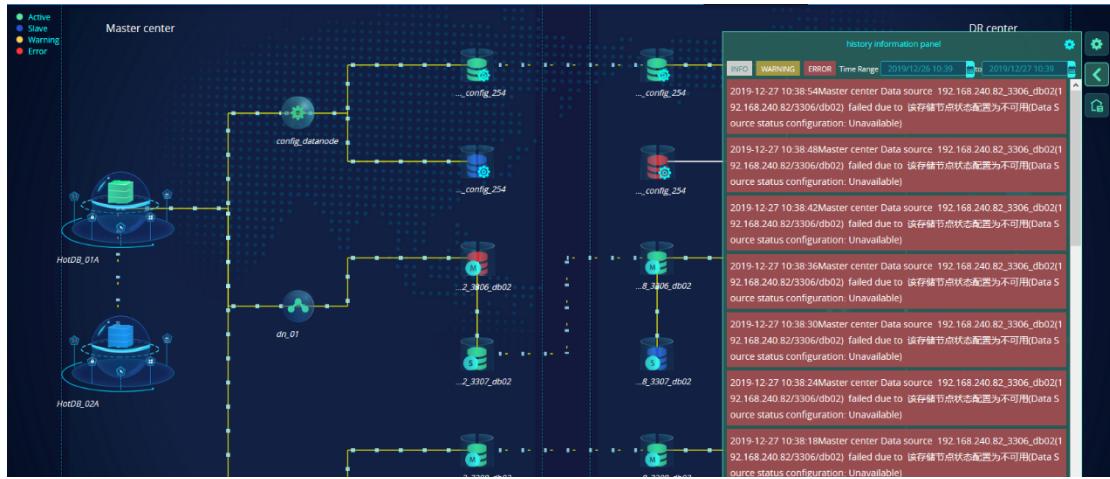
At this time, the disaster recovery status of the master data sources between the master center and the DR center is no longer detected, so the edges of the disaster recovery status are not displayed.



- The disaster recovery architecture in the multi-node mode is shown in the following figure, and its function is similar to that of the master/slave mode.



### (3) Topological Graph Info Panel of topology of DR center:



No matter whether the IDC fails or not, the Topological Graph Info Panel will monitor the status of the two IDCs. If the master center fails and the DR center is switched to be the current active center, other status will still be monitored except the disaster recovery relation between the two IDCs.

## 1. Info

- Add data sources and data nodes for both two IDCs;
- Add data sources only in one single IDC;
- Delete data sources only in one single IDC;
- Delete data sources and data nodes in the two IDCs at the same time.

## 2. Warning

- The replication status of data sources or ConfigDBs in a single IDC is abnormal;
- The replication latency of data sources or ConfigDBs in a single IDC exceeds the threshold;
- The disaster recovery latency of data sources and ConfigDBs between two IDCs exceeds the threshold.

## 3. Error

- Failure of data sources, ConfigDBs, or compute nodes in a single IDC;
- Switching of Data sources, ConfigDBs, or compute nodes occur in a single IDC;
- The last available data source in the current active center is abnormal;

- The replication status of data sources or ConfigDBs between the two IDCs is abnormal;
- The IDC switching occurs.

#### 4.4.2. Physical topological graph

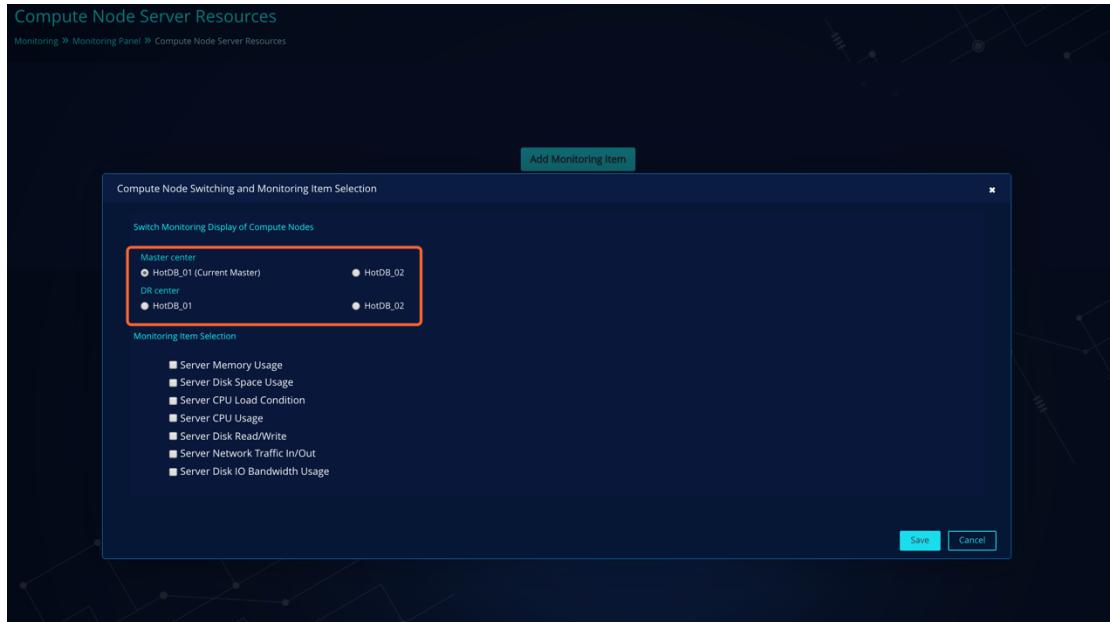
The physical topological graph adds a toggle button for the IDC switching, through which you can view the relations between the components of the master center or the DR center and the server, the use of server resources, and the running status of the services of each component.



#### 4.4.3. Other monitoring items

##### (1) Compute node server resources:

You can select to monitor all compute nodes that can connect to the management port whether in the master center or in the DR center on the compute node server resources function page. In [Add Monitoring Item], you can select the compute nodes of the master center or DR center you want to monitor. while the master compute node of the current active center is selected by default.



If the management port of compute nodes cannot be connected, for example, if the compute node fails or the compute node of DR center does not enable the management port, you cannot check and monitor the compute node.

## (2) Other server resources:

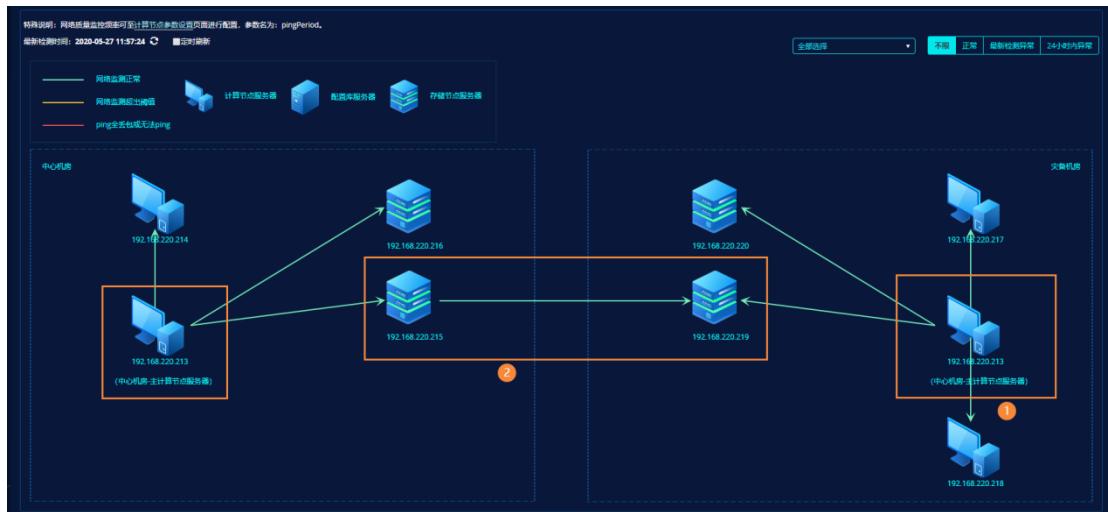
For other server resources, you can view detailed information about all server resources configured with SSH information in the two IDCs, including: CPU usage, memory usage, disk usage, and network traffic in and out.

(It is a prerequisite to configure the server's SSH information in "Configuration->Server")



### (3) Network quality:

You can refer to the cross-IDC network quality monitoring information. There are some differences between the cross-IDC network quality topology and that of the single IDC.



#### Page Description:

- The master compute node server in the master center needs to ping all servers in the master center except for itself and all servers in the DR center, so the master compute node server in the master center is placed in the master center and the DR center in the topology area above (as shown in mark 1 above);
- The cross-IDC network quality topology only displays: the master center compute node server (including the master/slave compute nodes), the DR center compute node server (including the master/slave compute nodes), the master center ConfigDB server, the master center data source server, the DR center ConfigDB server, the DR center data source server;
- If the server is shared by the service programs, the server role is divided according to the priority described in the single IDC network;
- The cross-IDC network replication relation requires the network quality link connection according to the replication relation between the master/slave data sources of the master IDC and the DR center (as shown in Note 2 above);

#### IDC switching instructions:



If the IDC switching happens, that is, the compute node of the current DR center provides service, only the network quality monitoring status of the DR center will be displayed; all components in the master center will be displayed gray with no monitoring. There will be no network connection from the DR center to the master center, and the network connection relation of the DR center degenerates to be consistent with that of the single IDC.

#### (4) Monitoring management:

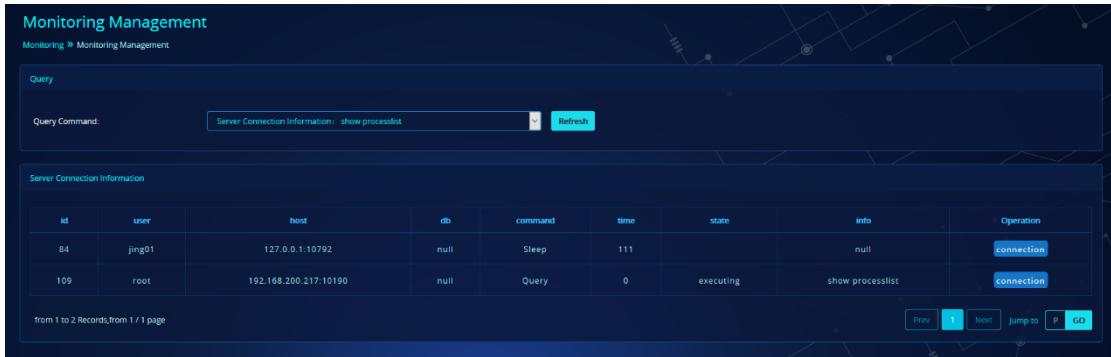
Monitoring management page can be used to check the relations between the front-end and back-end connections of compute nodes in the two IDCs and other effective management through the compute node management port.



The screenshot shows the 'Monitoring Management' page with a 'Query' section. It includes a 'Compute Node Selection:' dropdown set to '[Master center], [DR center]' and a 'Record' section. Below these are two input fields: 'Query Command:' containing 'Server Connection Information: show processlist' and a 'Record' field. To the right of the 'Query Command:' field is a checkbox list with the following items checked: '[Select all]', 'Master center', 'HotDB\_01', 'HotDB\_02', 'DR center', 'HotDB\_01', and 'HotDB\_02'. A 'Refresh' button is located next to the checkbox list.

For example, if you select "Server Connection Information: show processlist" in "Query Command", the server connection information of the master compute node server in the

current active center is displayed by default:



ID	User	Host	DB	Command	Time	State	Info	Operation
84	jing01	127.0.0.1:10792	null	Sleep	111		null	<button>connection</button>
109	root	192.168.200.217:10190	null	Query	0	executing	show processlist	<button>connection</button>

## 4.5. Management

### 4.5.1. Data backup

The management platform does not provide the data backup function for the DR center. If you want to restore the data backup of the current active center to another IDC, you can check the item "Backup to Remote Path simultaneously" in "Start Backup" to transfer all the backup files of each data source in the current IDC to a remote path through scp command before restoring the backup files to the corresponding data sources one by one according to the correspondence of the data sources offline.

### Start Backup

Management » Data Backup » Backup Task

LogicDB: Please select LogicDB

Backup Type: Full Backup

Backup ConfigDB Simultaneously: Yes (selected)

Compute the File MD5 Value or not

**Full Backup Setting**

Backup the binlog from last Full Backup to now simultaneously

Local Backup Root Directory: /data/mysql/dbbackup/

Compression Method: No Compression

Encryption Method: No Encryption

Backup File Format: SQL

Backup Statement Option: INSERT

Backup to Remote Path simultaneously

Remote Replication Mode: SCP (selected), RSYNC

Remote Host User: [Input Field]

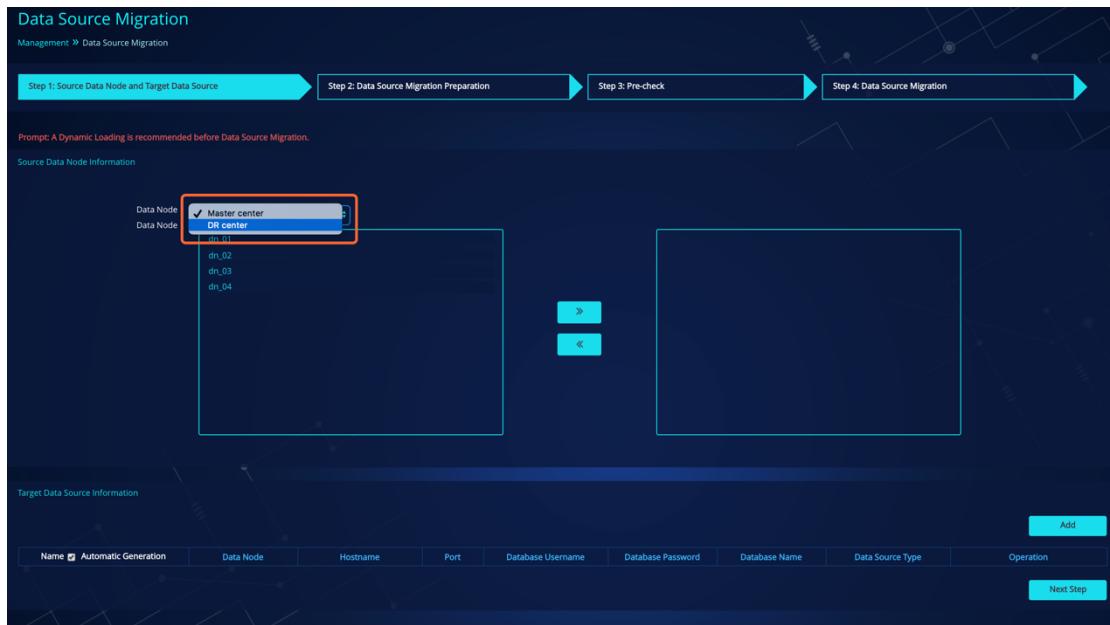
Remote Host Password: [Input Field]  Sign In without password

Remote Hostname: [Input Field]

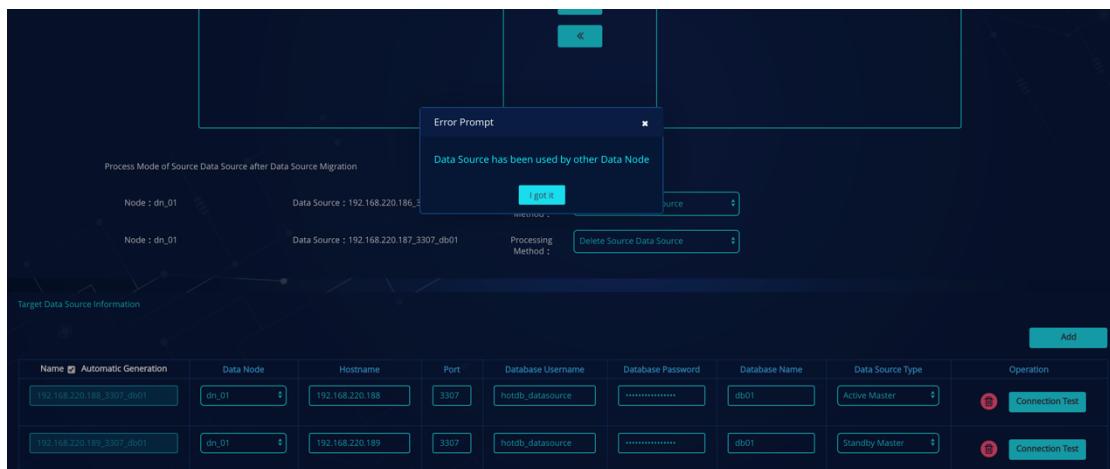
Remote Storage Directory: [Input Field]

#### 4.5.2. Data source migration

The management platform supports the migration of data sources in the master center or DR center to new data sources, and the re-setup of disaster recovery relations.



- You can choose whether to migrate the data source of the master center or the data source of the DR center by selecting the IDC type, however, you still need to meet the prerequisites of using the migration function, for example, you need to manually import the data of the old data source to the new data source and set up the replication relations from the old data source to the new data source. It needs attention that the configured target new database instance cannot coincide with the existing instance in the DR center, that is, the instance filled in by the detection cannot coincide with all the existing data source instances in the ConfigDB.



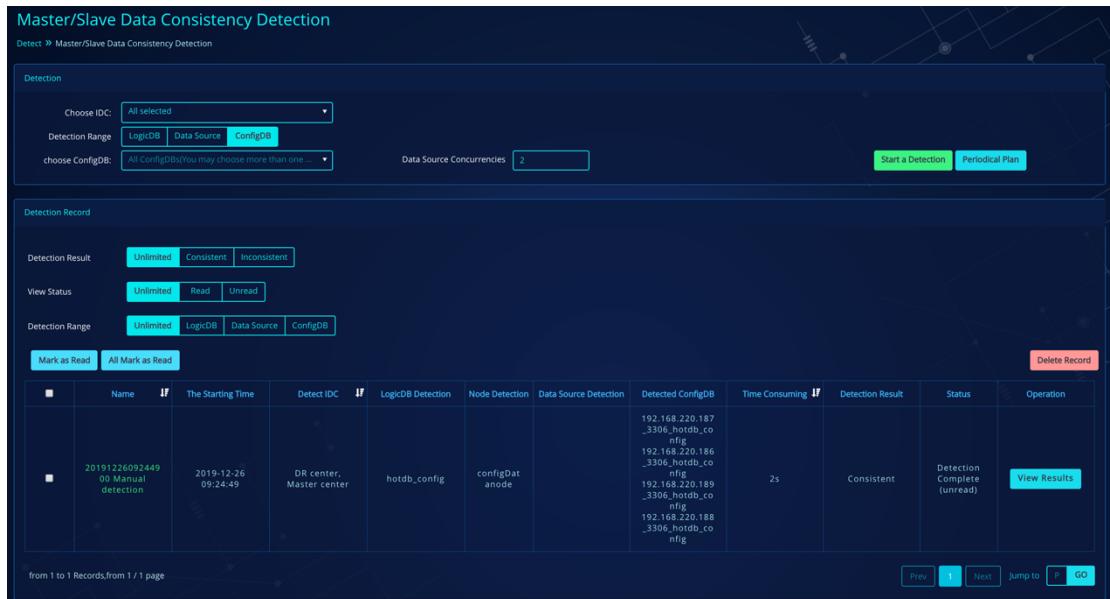
- In [Step 4: Data Source Migration], a replication relation will be set up of the current active data sources between the master center and the DR center. For example, if to migrate the data nodes of the master center, the management platform will set up a replication relation between the target master data source of the master center and

the current active data source of the DR center.

## 4.6. Detect

### 4.6.1. Master/slave data consistency detection

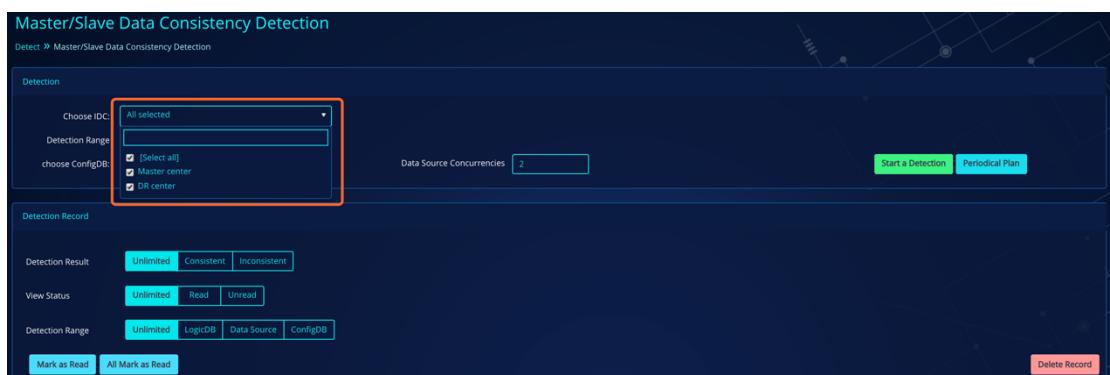
The IDC type is a multiple choice drop-down box, that is, you can select a central IDC or a disaster recovery IDC or two IDCs at the same time.



The screenshot shows the 'Master/Slave Data Consistency Detection' interface. In the 'Detection' section, the 'Choose IDC' dropdown is set to 'All selected'. Below it, the 'Detection Range' dropdown is set to 'LogicDB'. The 'choose ConfigDB' dropdown shows 'All ConfigDBs (You may choose more than one)' with a value of '2'. On the right, there are 'Start a Detection' and 'Periodical Plan' buttons. In the 'Detection Record' section, there is a table with columns: Name, If, The Starting Time, Detect IDC, If, LogicDB Detection, Node Detection, Data Source Detection, Detected ConfigDB, Time Consuming, If, Detection Result, Status, and Operation. One row is shown, indicating a manual detection on 2019-12-26 at 09:24:49, detected by 'DR center, Master center' using 'hotdb\_config' and 'configData\_anode'. The 'Detected ConfigDB' column lists several IP addresses and port numbers. The 'Detection Result' is 'Consistent' and the 'Status' is 'Detection Complete (unread)'. A 'View Results' button is present. At the bottom, there are navigation buttons for 'Prev', 'Next', 'Jump to', and 'GO'.

#### (1) Choose IDC:

The Choose IDC item is a drop-down box with multiple choices, that is, you can select a master center, a DR center or the two IDCs both.



This screenshot is similar to the previous one but highlights the 'Choose IDC' dropdown in the 'Detection' section. The dropdown is set to 'All selected' and contains three options: 'Select all', 'Master center', and 'DR center'. The other interface elements, including the detection record table, are identical to the first screenshot.

For detection range of LogicDB or detection range of data source:

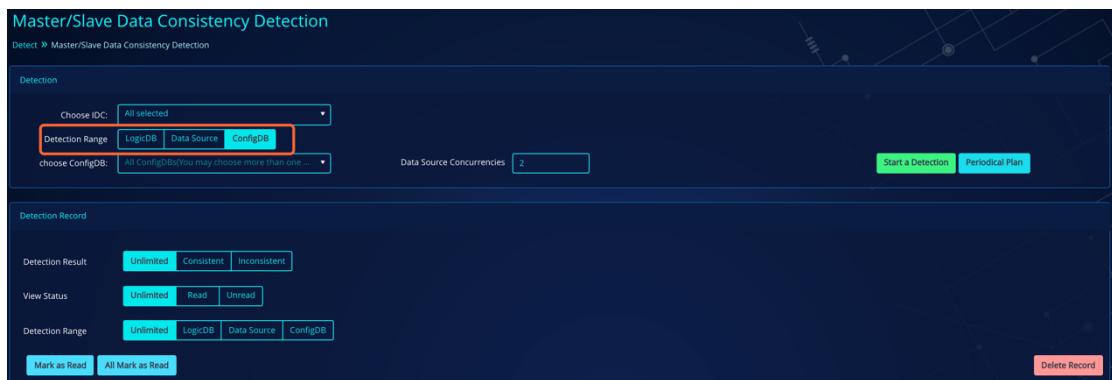
- If a single IDC is selected, the data sources under the selected data nodes in the

IDC will be compared with the master data sources under the same data nodes to see whether the data is consistent, same logic as the master/slave data consistency detection in single-IDC mode.

- If two IDCs are both selected, all the data sources under the selected data nodes will be compared with the master data sources under the corresponding data nodes of the current active center to see whether the data is consistent.

For example, in the detection range of **LogicDB**, select logicdb01, which is associated with the dn\_01 of the master center and the DR center. Click [Start a Detection], and then the current active data source (ds\_01) under the dn\_01 of master center will be compared with all the data sources (ds\_01, ds\_02, ds\_03, ds\_04) under the dn\_01 of the two IDCs respectively.

## (2) Detection range of ConfigDB:



- In detection range of ConfigDB, select the ConfigDB address you want to detect and fill in the concurrency number (the default is 2)
- Click [Start a Detection] to check the data consistency between the selected ConfigDB and the master ConfigDB of the current active center.
- Click [View Results] to view the details of detection results. If there are inconsistencies, locations and details of the inconsistencies will be displayed in the detection results.

Master/Slave Data Consistency Detection			
Detect > Master/Slave Data Consistency Detection			
Detection Result:	<input checked="" type="checkbox"/> Unlimited <input type="checkbox"/> Consistent <input type="checkbox"/> Inconsistent		
choose ConfigDB:	<input type="checkbox"/> All ConfigDBs (You may choose more) <input type="checkbox"/>		
Detect IDC: DR center, Master center			
Start Time: 2019/12/26 10:20:29 Consuming Time: 14s			
Back to Consistency Detection			
ConfigDB	Table	Result	Details
192.168.240.77_3316.hotdb_config,192.168.240.76_3316.hotdb_config	HOTDB_IP_WHITELIST_RUNNING	<span style="color: red;">●</span>	<p>表: HOTDB_IP_WHITELIST_RUNNING在节点: 192.168.240.77_3316.hotdb_config,192.168.240.76_3316.hotdb_config 存在数据不一致。列: ID, 分布区间: [-26,24-25], 且段ID: (24),(25) (Data inconsistency in table HOTDB_IP_WHITELIST_RUNNING in node: 192.168.240.77_3316.hotdb_config,192.168.240.76_3316.hotdb_config, inconsistent data is: ID distribution in [-26,24-25], and the unique key of inconsistent rows is (ID)-(24),(25)) Please Check Table Definition</p>
192.168.240.77_3316.hotdb_config,192.168.240.76_3316.hotdb_config	HOTDB_FAILOVER	<span style="color: red;">●</span>	<p>表: HOTDB_FAILOVER在节点: 192.168.240.77_3316.hotdb_config,192.168.240.76_3316.hotdb_config 存在数据不一致。列: ID, 分布区间: [-27,24-29], 且段ID: (24),(25),(26),(27) (Data inconsistency in table HOTDB_FAILOVER in node: 192.168.240.77_3316.hotdb_config,192.168.240.76_3316.hotdb_config, inconsistent data is: ID distribution in [-27,24-29], and the unique key of inconsistent rows is (ID)-(24),(25),(26),(27)) Please Check Table Definition</p>
192.168.240.77_3316.hotdb_config,192.168.240.76_3316.hotdb_config	HOTDB_SERVER_STATUS	<span style="color: red;">●</span>	<p>表: HOTDB_SERVER_STATUS在节点: 192.168.240.77_3316.hotdb_config,192.168.240.76_3316.hotdb_config 存在数据不一致。列: ID, 分布区间: [-1,1], 且段ID: (1) (Data inconsistency in table HOTDB_SERVER_STATUS in node: 192.168.240.77_3316.hotdb_config,192.168.240.76_3316.hotdb_config, inconsistent data is: ID distribution in [-1,1], and the unique key of inconsistent rows is (ID)-(1)) Please Check Table Definition</p>

## 4.7. Event

#### 4.7.1. Notification strategy

After adding notification strategy, it can alert you by email to the failure or abnormalities of the master center or DR center during cluster operation. The monitoring items including [Failure Real-time Monitoring], [Compute Node Server Resource Monitoring], [Compute Node Service Status], [Data Source Information Monitoring], and [ConfigDB Monitoring] are added with the IDC identifications to alert the failure or abnormalities related to the current active center; [Other Server Resource Monitoring] and [License Authorization Monitoring] are added with IDC identifications to identify the IDC where the server exceeds the alert threshold and the IDC to which the compute node with abnormal license belongs.

**(1) Email information:**

Added email information: whether to enable the DR mode, type of the current active center, and compute node information of two IDCs.

Email information:  
Compute Node Cluster name: Group\_7843  
DR mode: Open  
Current master center: Master center  
Master center Compute Node information: 计算节点-01(192.168.220.221/3325)  
DR center Compute Node information: 计算节点-01(192.168.220.223/3325)

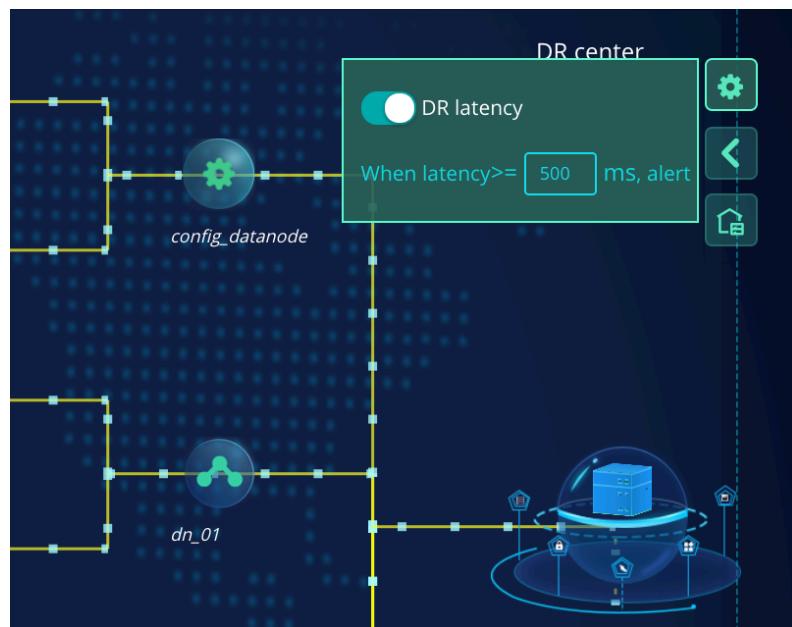
## (2) Data source information monitoring:



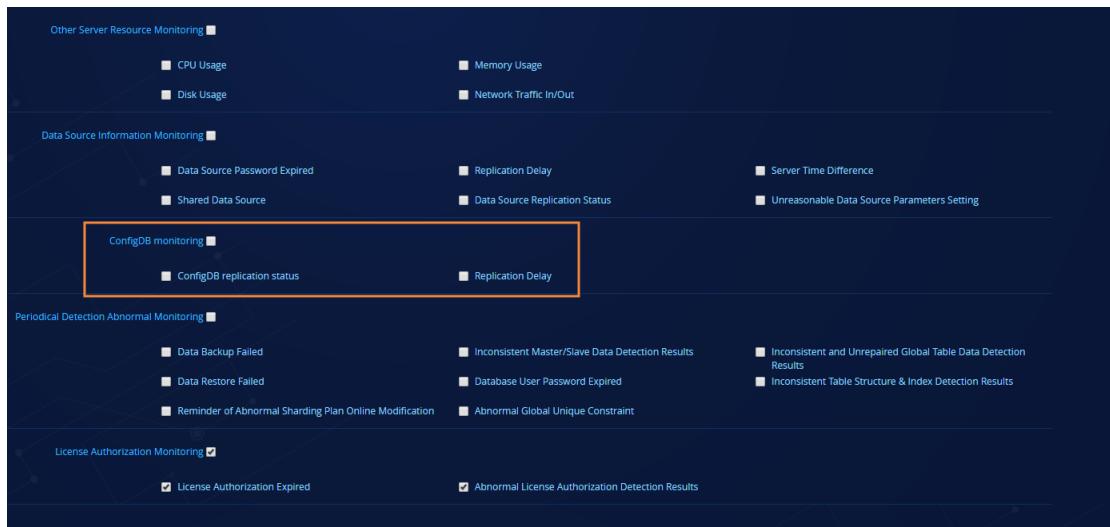
In the data source information monitoring, for [Replication Delay] and [Data Source Replication Status], in addition to alerting the abnormality of the internal data source in the current active center, it also alerts the disaster recovery latency and the abnormal disaster recovery status between the current active data source in the master center and the current active data source in the DR center.

### Threshold setting:

- ¬ The replication latency still corresponds to the replication latency under the data source module in "Setting -> Topological Graph Alert Setting"
- ¬ The disaster recovery latency corresponds to the disaster recovery latency setting of the topology in DR center in "Monitoring-> [Logical Topological Graph](#)"



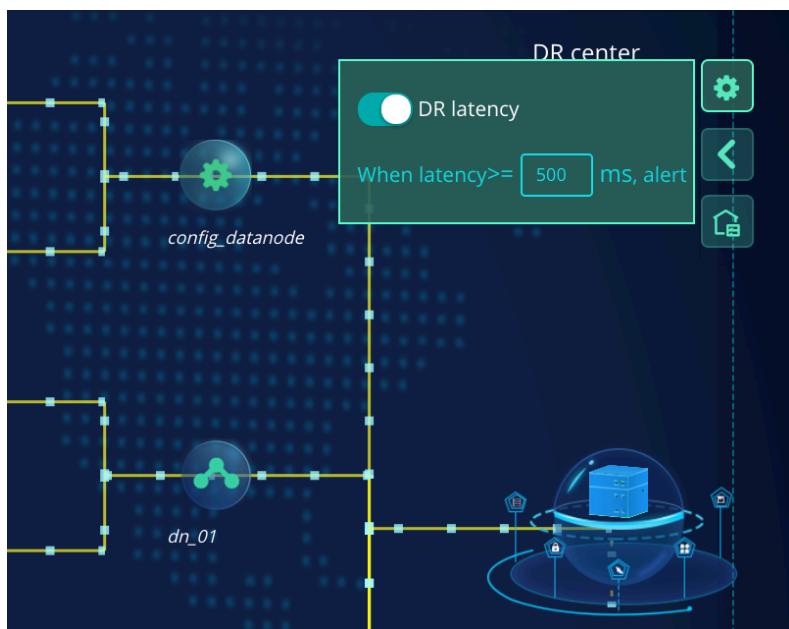
## (3) ConfigDB information monitoring:



In ConfigDB information monitoring, alert is added of the replication latency and abnormal replication status between ConfigDBs in the current active center and between the master ConfigDBs in the two IDCs.

### Threshold setting:

- The replication latency corresponds to the replication latency under the ConfigDB module in "Setting -> Topological graph alert setting"
- The disaster recovery latency corresponds to the disaster recovery latency setting of the topology in the DR center in "Monitoring -> [Logical Topological Graph](#)"



## 4.7.2. Others

### (1) History Events:

The history events will display the history events that occurred in the current active center recorded by the management platform. Contents include: notification of completed task, notification of abnormal periodical detection, and early warnings triggered by the platform, etc. The IDC type is displayed as the IDC type when the history event occurs.

History Events						
Event >> History Events						
Time Range	2019-12-19 17:27:56		to	2019-12-26 17:27:56		
Alert Type	All Alert Type					
Alert Level	All Alert Level					
Data Node	All Nodes					
						<a href="#">Empty History Events</a> <a href="#">Export</a>
Time	IDC Type	Alert Type	Level	Data Node	Alert Contents	
2019-12-26 03:00:00	Master center	Master/Slave consistency detection	INFO		Master/slave consistency detection is completed and detection results are consistent	
2019-12-26 03:00:00	Master center	Master/Slave consistency detection	INFO		Master/slave consistency detection is completed and detection results are consistent	
2019-12-25 02:47:52	Master center	Master/Slave consistency detection	INFO		Master/slave consistency detection is completed and detection results are consistent	
2019-12-25 02:51:52	Master center	Master/Slave consistency detection	INFO		Master/slave consistency detection is completed and detection results are consistent	
2019-12-25 02:51:52	Master center	Master/Slave consistency detection	INFO		Master/slave consistency detection is completed and detection results are consistent	
2019-12-25 02:51:52	Master center	Master/Slave consistency detection	INFO		Master/slave consistency detection is completed and detection results are consistent	
2019-12-25 02:51:52	Master center	Master/Slave consistency detection	INFO		Master/slave consistency detection is completed and detection results are consistent	
2019-12-25 02:51:52	Master center	Master/Slave consistency detection	INFO		Master/slave consistency detection is completed and detection results are consistent	
2019-12-24 03:32:52	Master center	Abnormal Table Structure & Index detection	WARNING		2019-12-24 03:32:52 Table structure and table index detection finds tables are found abnormal in the table index definition or table has no index	
2019-12-24 01:46:14	Master center	Abnormal Table Structure & Index detection	WARNING		2019-12-24 01:46:14表结构与表索引检测中发现1张表存在表索引定义异常或表无索引	
2019-12-23 03:00:00	Master center	Abnormal Table Structure & Index detection	WARNING		2019-12-23 03:00:00表结构与表索引检测中发现1张表存在表索引定义异常或表无索引	

### (2) Compute Node Logs:

The compute node logs record the log information generated by the compute nodes in the master center and the DR center during operation. By default, the log information of the master compute node in the current active center is displayed. You can select or view other compute node log information.

**Compute Node Logs**

Event > Compute Node Logs

Compute Node: [Master center: HotDB\_01] Time Range: 2019-12-26 17:28:20 to 2019-12-26 17:28:20 Prompt: Display Logs of Recent One Week by Default

Log Type: [Select all] Master Center, HotDB\_01, HotDB\_02, DR Center, HotDB\_03, HotDB\_04

Log Level: View Status: Log Contents:

Mark as Read All Mark as Read Highlight WARN and ERROR Refresh Log Log Setting Delete Log Download Log File

IDC Type	Compute Node	View Status	Time	Level	Type	Log Contents	Operation
Master center	HotDB_01	Unread	2019-12-26 12:08:39	WARN	AUTHORITY	Compute Node Trial expires in 1967 hours and 18 minutes and 32 seconds.	<a href="#">View Details</a>
Master center	HotDB_01	Unread	2019-12-25 17:51:32	WARN	INIT	using master config in start up:[id:-1,nodeId:-1 192.168.220.181:3306/Compute Node_config sta...	<a href="#">View Details</a>
Master center	HotDB_01	Unread	2019-12-25 17:46:43	WARN	INIT	using master config in start up:[id:-1,nodeId:-1 192.168.220.181:3306/Compute Node_config sta...	<a href="#">View Details</a>
Master center	HotDB_01	Unread	2019-12-25 17:31:41	WARN	INIT	using master config in start up:[id:-1,nodeId:-1 192.168.220.181:3306/Compute Node_config sta...	<a href="#">View Details</a>
Master center	HotDB_01	Unread	2019-12-25 12:08:07	WARN	AUTHORITY	Compute Node Trial expires in 1991 hours and 19 minutes and 4 seconds.	<a href="#">View Details</a>
Master center	HotDB_01	Unread	2019-12-25 12:27:54	WARN	INIT	using master config in start up:[id:-1,nodeId:-1 192.168.220.181:3306/Compute Node_config sta...	<a href="#">View Details</a>
Master center	HotDB_01	Unread	2019-12-24 12:07:21	WARN	AUTHORITY	Compute Node Trial expires in 2015 hours and 19 minutes and 49 seconds.	<a href="#">View Details</a>
Master center	HotDB_01	Unread	2019-12-24 12:07:18	WARN	INIT	The datasources' time_zones are not consistent.	<a href="#">View Details</a>
Master center	HotDB_01	Unread	2019-12-24 12:07:16	WARN	INIT	using master config in start up:[id:-1,nodeId:-1 192.168.220.181:3306/Compute Node_config sta...	<a href="#">View Details</a>

## 4.8. Others

### (1) server:

The server is added with the IDC type, which manages the server SSH information of all components in the cluster. When configuring or adding server SSH information, you need to select the IDC type corresponding to the server.

**Server**

Configuration > Server

Hostname (multiple choices): SSH Connection Status (multiple choices): Add Server Batch Operation

IDC Type	Hostname	Service Program Association	SSH Connection Status	Username	Operation
DR center	192.168.220.184	Compute Node HotDB_04(192.168.220.184:3325) ConfigDB 192.168.220.184:3306/hotdb_config Data Source 192.168.220.184:3308/db2531 192.168.220.184:3309/db2531 192.168.220.184:3310/db2531 keepalive vip:192.168.220.242	Normal	root	<a href="#">Configuration SSH</a> <a href="#">Stop Monitoring</a> <a href="#">Remove</a>
DR center	192.168.220.184	Compute Node HotDB_04(192.168.220.184:3325) ConfigDB 192.168.220.184:3306/hotdb_config Data Source 192.168.220.184:3308/db2531 192.168.220.184:3309/db2531 192.168.220.184:3310/db2531 keepalive vip:192.168.220.242	Normal	root	<a href="#">Configuration SSH</a> <a href="#">Stop Monitoring</a> <a href="#">Remove</a>
DR center	192.168.220.183	Compute Node HotDB_03(192.168.220.183:3325) ConfigDB 192.168.220.183:3306/hotdb_config Data Source 192.168.220.183:3308/db2531 192.168.220.183:3309/db2531	Normal	root	<a href="#">Configuration SSH</a> <a href="#">Stop Monitoring</a> <a href="#">Remove</a>

## (2) Data source password:

The data source password is also added with the IDC type, and displays the IDC type corresponding to all data sources.

Data Source User Password						
Security > Password Safety > Data Source User Password						
IDC Type		Data Source Instance	Username	Expiration Countdown	Data Source Name	Operation
Master center		192.168.210.53:3308	hotdb_datasource	87Day(s)	192.168.210.53_3308_db01	
Master center		192.168.220.216:3308	hotdb_datasource	87Day(s)	192.168.220.216_3308_db01	
Master center		192.168.210.53:3309	hotdb_datasource	87Day(s)	192.168.210.53_3309_db01	
Master center		192.168.220.216:3309	hotdb_datasource	87Day(s)	192.168.220.216_3309_db01	
Master center		192.168.220.216:3309	hotdb_datasource	87Day(s)	192.168.220.216_3309_db04	
DR center		192.168.240.79:3309	hotdb_datasource	87Day(s)	192.168.240.79_3309_db01	
Master center		192.168.220.216:3306	hotdb_datasource	87Day(s)	192.168.220.216_3306_db01	
DR center		192.168.240.79:3308	hotdb_datasource	87Day(s)	192.168.240.79_3308_db01	
Master center		192.168.210.53:3307	hotdb_datasource	87Day(s)	192.168.210.53_3307_db01	
DR center		192.168.240.79:3307	hotdb_datasource	87Day(s)	192.168.240.79_3307_db01	
Master center		192.168.220.216:3307	hotdb_datasource	87Day(s)	192.168.220.216_3307_db01	

from 1 to 10 Records, from 1 / 3 page  
Prompt: Expiration Date and Advance Reminder Time of Password could be modified in [Setting->Periodical Detection Setting]

Batch Edit One-click Setting History Record

Prev 1 2 3 Next Jump to 1 GO

## (3) Information collection:

The information collection tool of management platform supports the collection of logs and configuration files for abnormality analysis of all components of the two IDCs.

Information Collection

Tool > Information Collection

Collection Record

Collect by Scene Cluster Running Status Performance Test One-key Collection

Collection List

Management Platform	Compute Node	Data Source
<ul style="list-style-type: none"> <li>hotdb-management.log</li> <li>message</li> <li>Management Platform and Current Server Related Information</li> </ul>	<ul style="list-style-type: none"> <li>hotdb.log</li> <li>console.log</li> <li>message</li> <li>server.xml</li> <li>Start Script</li> <li>Compute Node Related Information</li> <li>Compute Node Server Related Information</li> </ul>	<ul style="list-style-type: none"> <li>error.log</li> <li>hotdb_backup.log</li> <li>message</li> <li>my.cnf</li> <li>Data Source and Current Server Related Information</li> </ul>
ConfigDB	<ul style="list-style-type: none"> <li>error.log</li> <li>message</li> <li>my.cnf</li> <li>ConfigDB Table Configuration</li> <li>ConfigDB and Current Server Related Information</li> </ul>	

Collection Setting  Compute Node Server Exports Whole JVM Memory Information

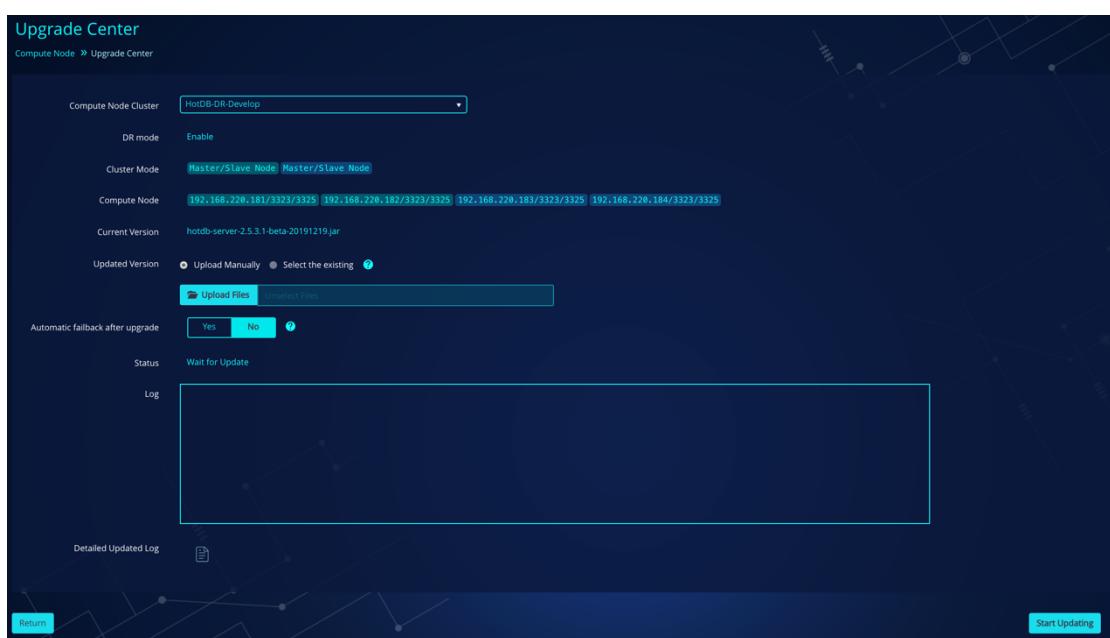
If there is a server in the two IDCs that cannot be connected, you can choose to skip the collection of server information or cancel the collection task. If there are compute node

services that cannot be connected in the two IDCs, you can manually enter the compute node installation directory and continue the collection task. That is, neither the failure of the IDC or the unopened service port of the compute node of the DR center will affect the information collection task.

#### (4) Upgrade Center:

The upgrade center also supports cross-version or minor-version upgrade, that is, users can upgrade the lower version of the compute node to version 2.5.3.1, so that the already running low-version compute node cluster can enable the DR mode after the upgrade is completed.

Select the compute node cluster that needs to be upgraded. If the cluster has the DR mode enabled, the color green will be used to mark the master center, and blue to mark the DR center for the items Cluster Mode and the Compute Node. After uploading the upgrade package, click [Start Update] to enter the upgrade process.



After the upgrade is complete, you can view the detailed update logs:

**Upgrade Center**

Compute Node » Upgrade Center

Compute Node Cluster	HotDB-DR-Develop
DR mode	Enable
Cluster Mode	Master/Slave Node Master/Slave Node
Compute Node	[192.168.220.181/3323/3325] [192.168.220.182/3323/3325] [192.168.220.183/3323/3325] [192.168.220.184/3323/3325]
Pre-Upgrade Version	hotdb-server-2.5.3.1-beta-20191213.jar
Upgrade Version	hotdb-server-2.5.3.1-beta-20191217.tar.gz
Status	Upgraded Successfully
Log	<pre> 2019-12-18 20:05:50 Enter cluster HotDB-DR-Develop for version update 2019-12-18 20:05:54 High availability switching detection: HotDB_01 (192.168.220.181 / 3323) HotDB_02 (192.168.220.182 / 3323) passed 2019-12-18 20:06:28 Close service: HotDB_02 (192.168.220.182 / 3323) service is closed 2019-12-18 20:06:29 Upload /usr/local/hotdb/hotdb-management/bin/upgrade/hotdb-server-2.5.3.1-beta-20191217.tar.gz to /usr/local/hotdb/update successfully 2019-12-18 20:06:30 Backup file [Version patch]: HotDB_02 (192.168.220.182 / 3323), lib directory file, backup file HotDB-DR-Develop_HotDB_02_20191218200629.tar.gz storage directory /usr/local/hotdb/hotdb-server /backup 2019-12-18 20:06:30 Replace files [Version patch]: HotDB_02 (192.168.220.182 / 3323) replaced lib directory files 2019-12-18 20:06:41 Enable service: HotDB_02 (192.168.220.182 / 3323) service is enabled 2019-12-18 20:06:46 Starting service: HotDB_02 (192.168.220.182 / 3323) Starting succeeded 2019-12-18 20:06:46 0 backup tasks running 2019-12-18 20:06:46 0 restore tasks running 2019-12-18 20:06:46 Detection before stopping service: HotDB_01 (192.168.220.181 / 3323) detection passed 2019-12-18 20:07:19 Close service: HotDB_01 (192.168.220.181 / 3323) service is closed 2019-12-18 20:07:20 Upload /usr/local/hotdb/hotdb-management/bin/upgrade/hotdb-server-2.5.3.1-beta-20191217.tar.gz to /usr/local/hotdb/update successfully 2019-12-18 20:07:21 Backup file [Version patch]: HotDB_01 (192.168.220.181 / 3323), lib directory file, backup file HotDB-DR-Develop_HotDB_01_20191218200720.tar.gz storage directory /usr/local/hotdb/hotdb-server /backup 2019-12-18 20:07:38 Enable service: HotDB_01 (192.168.220.181 / 3323) service is enabled 2019-12-18 20:07:38 High availability environment creation: HotDB_01 (192.168.220.181 / 3323), HotDB_02 (192.168.220.182 / 3323) start 2019-12-18 20:07:39 High availability environment reconstruction execution: high availability environment reconstruction succeeded </pre>
Detailed Updated Log	

### Note:

- If the currently configured master center is not the current active center and the current master center management port cannot be connected, the upgrade cannot be performed.
- If the compute node in the master center is in the master/slave mode, the page will display "whether to switch back after the upgrade". Select "Yes" to switch back to the former master compute node after the upgrade (the master/slave high availability switch will be performed during the upgrade).
- If the compute node in the DR center is in master/slave mode, no matter whether the upgrade is successful or not, only the master compute node management port will be enabled after the upgrade, that is, the slave compute node management port or keepalived components will not be enabled.
- The upgrade process will first upgrade the master center and then the DR center. The cluster that failed to upgrade will still perform the rollback process. The rollback process will also first roll back the master center and then the DR center.

## 5. Descriptions of compute-node-related functions

### 5.1. Master center

In the DR mode, the compute node service in a master center is nearly the same as that in a single IDC. Under normal circumstances, the service ports are provided by master compute nodes of the master center, while the slave compute nodes are standby.

### 5.2. DR center

The master-slave compute node services in DR center are in standby status before the IDC-level switching of service occurs, and only the management port (3325 by default) provides services.

The online command is disabled for the master-slave compute nodes of DR center before the IDC-level switching of service occurs, mainly to differentiate from the general high availability switching operation. When the online command is executed, the following prompt will be given:

```
root@192.168.220.183:(none) 8.0.15-HotDB-2.5.3.1 07:58:26> online;  
ERROR 10192 (HY000): access denied. online is not allowed in a DR HotDB Server.
```

The master compute node of DR center is not involved in HA high availability switching and master selection of master center. In addition to some show commands, only manual commands are accepted: online\_dr switch the IDC.

```
root@192.168.220.183:(none) 8.0.15-HotDB-2.5.3.1 08:12:31> online_dr;  
Query OK, 1 row affected (5 min 4.35 sec)
```

Once the master compute node of DR center executes the online\_dr command, an IDC-level switching of the compute node services will occur, and the master compute node of DR center will provide services. For other situations, please refer to the detailed description of the [IDC failure](#) in related chapters. At the same time, the compute node

logs output of DR center are as follows, which means that the service of the DR center is successfully enabled:

```
2019-12-12 19:50:47.257 [INFO] [MANAGER] [$NIOExecutor-1-0]
cn.hotpu.hotdb.manager.ManagerQueryHandler(178) - online_dr by [thread=$NIOExecutor-1-
0, id=8514, user=root, host=192.168.220.183, port=3325, localport=13838, schema=null]

2019-12-12 19:50:47.258 [INFO] [MANAGER] [Labor-2] cn.hotpu.hotdb.HotdbServer(2111) - DR
online start

.....
2019-12-12 19:50:50.587 [INFO] [MANAGER] [Labor-2] cn.hotpu.hotdb.HotdbServer(2142) - DR
online end
```

The slave compute node of DR center is not involved in the HA high availability switching and master selection of master center. When the IDC-level switching of the compute node occurs, that is, when the master compute node of the DR center provides services, if the master compute node of the DR center fails at this time, it can execute enable\_online; command before executing online or online\_dr command to enable the slave compute node of the DR center. After the command is executed, the slave compute node in the DR center can automatically enable the service port (3323 by default) to continue the service by enabling the process judging.

```
root@192.168.220.184: (none) 8.0.15-HotDB-2.5.3.1 08:10:31> enable_online;
Query OK, 1 row affected (11 min 5.39 sec)

root@192.168.220.184: (none) 8.0.15-HotDB-2.5.3.1 08:22:27> online_dr;
Query OK, 1 row affected (0.01 sec)
```

Note: after the IDC-level failure and the online\_dr command is manually executed in the master compute node of DR center, the enable\_online command needs to be executed in the slave compute node of the DR center at the same time to ensure that the master compute node of the DR center can automatically switch to the slave compute node of the DR center after failure. Otherwise, the master compute node of the DR center may not be able to switch to the slave compute node of the DR center after failure.

If it is multi-node cluster mode, it needs to execute the `online_dr` command in the primary compute node of the DR center, to ensure that the successful switching in the compute node clusters in the DR center.

If you execute the `online_dr` command directly on the slave compute node of DR center, the IDC switching will also occur, and the service port (3323) will be enabled on the party currently executing the disaster recovery service startup command. At this time, you need to pay attention to whether the VIP and service port of the keepalived service are on the same server.

The master compute node of DR center needs `online_dr` to enable the service normally every time it starts the process; the slave compute node of DR center in the master/slave mode needs `enable_online` to accept online commands and provide subsequent high availability service every time it starts the process. In multi-node mode, it only needs to execute the `online_dr` command to the primary compute node.

# 6. Operation and maintenance management

This chapter masterly describes the judgment logic of the compute nodes and how to correctly repair the failure in the case of core component service failure and IDC-level failure during the operation of the compute node disaster recovery cluster.

## 6.1. Premise explanation

For the convenience of subsequent explanation, except for special remark, the following basic information description shall be taken as an example in the operation and maintenance management chapter.

Prepare a set of normally running DR mode compute node cluster (including master/slave compute nodes, master-master data sources, master-master ConfigDB). For the convenience of subsequent explanation, all components are represented by simple code names, including:

The master compute node of the master center is HotDB-01; and the slave compute node is HotDB-02;

The master compute node of the DR center is HotDB-03; and the slave compute node is HotDB-04;

The corresponding master data source of data node dn01 in the master center is ds01; and the standby master data source is ds02;

The corresponding master data source of data node dn01 in the DR center is ds03; and the standby master data source is ds04;

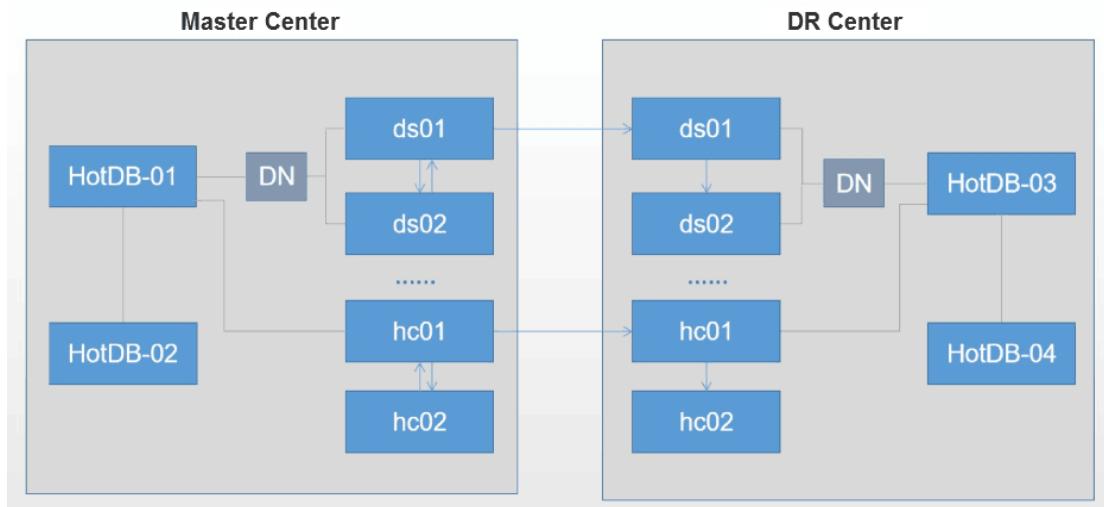
The master ConfigDB of the master center is hc01, and the standby master ConfigDB is hc02;

The master ConfigDB of the DR center is hc03, and standby master ConfigDB is hc04;

A master-master replication relation is set up between ds01 and ds02; a master-slave relation is set up between ds01 and ds03, ds03 as the slave; a master-slave relation is set up between ds03 and ds04, ds04 as the slave;

A master-master replication relation is set up between hc01 and hc02; a master-slave relation is set up between hc01 and hc03, ds03 as the slave; a master-slave relation is set up between hc03 and hc04, hc04 as the slave.

Graph representation (edge arrow indicates data flow direction):



All MySQL instance versions are 5.7.25, GTID enabled, and semi-synchronous replication is enabled by default for all replication relations. Other parameter configurations and system parameter settings are the same as the default installation and deployment conditions.

The actual connection information corresponding to each component is listed as follows:

IDC Type	Component Type	Role	Code Name	Connection Information
Master Center	Compute Node	Master	HotDB-01	192.168.220.181_3323_3325
		Slave	HotDB-02	192.168.220.182_3323_3325
	Data Source	Master	ds01	192.168.220.181_3307
		Master-master	ds02	192.168.220.182_3307
	ConfigDB	Master	hc01	192.168.220.181_3306

		Master-master	hc02	192.168.220.182_3306
DR Center	Compute Node	Master	HotDB-03	192.168.220.183_3323_3325
		Master-master	HotDB-04	192.168.220.184_3323_3325
	Data Source	Master	ds03	192.168.220.183_3307
		Master-master	ds04	192.168.220.184_3307
	ConfigDB	Master	hc03	192.168.220.183_3306
		Master-master	hc04	192.168.220.184_3306

Note:

- The data node names of the same group of data sources with disaster recovery relations are the same;
- By default, the data sources and ConfigDBs of the DR center are only set up with the replication relation from the master to the standby master, and do not set up a return circuit, which is different from the role of the actual standby master;
- When deploying in a real scenario, it is not recommended to deploy compute nodes and data sources on the same server. This time is only for the convenience of subsequent explanation.
- All data sources and ConfigDBs with master-slave relations need to set the value of `rpl_semi_sync_master_wait_for_slave_count` as in correct status in advance.
- There is little difference between the multi-node cluster mode and the master/slave mode, so the master/slave mode shall prevail. In the multi-node mode, when the DR center is enabled, it should be noted to execute `online_dr` command in the primary compute node and replace haMode according to the role.

## 6.2. Failure only inside the master center

This section mainly describes the function usage when high-availability switching (including manual switching or failover) occurs in only the core service components in the master center in a cluster environment where services are normally provided.

### 6.2.1. Compute node

#### 6.2.1.1. Manual switching

Compute Node Cluster														
Cluster Management > Compute Node Cluster														
Filter by Compute Node Cluster Name: <input type="text"/> <input type="button" value="Q"/>														
ID	Group Name	Cluster Mode	IDC Type	Cluster Component Configuration						Cluster Deployment Info				
3	Node 3	Single Node	Master Center	HotDB_01	192.168.220.211	3323	3325	Master						
	Single Node	DR Center		HotDB_01	192.168.210.80	3323	3325	Master						
2	Group 2M	Master/Slave Node	Master Center	HotDB_01	192.168.240.119	3323	3325	Master	Active	Active	Active	Total number of instances: 4	25	
		Master/Slave Node	DR center	HotDB_02	192.168.240.120	3323	3325	Slave	Active	Active	Active	Total number of instances: 4 Active: 4	65	
	Group 2M	Master/Slave Node	Master Center	HotDB_03	192.168.240.121	3323	3325	Master	ACTIVE	Active	ACTIVE	Total number of instances: 4 Active: 4	52	
		Master/Slave Node	DR center	HotDB_04	192.168.240.122	3323	3325	Slave	Active	Active	Active	Total number of instances: 4 Active: 4	62	
from 1 to 2 Records, from 1 / 1 page														
<input type="button" value="Prev"/> <input type="button" value="1"/> <input type="button" value="Next"/> <input type="button" value="Jump to"/> <input type="text" value="60"/>														

Enter the "Cluster Management"->"Compute Node Cluster" page. When you manually click the [Switch] button corresponding to the compute node group with the master-slave relation to perform manual compute node switching operations, switching will only occur in the master center, which has no effect on the compute node services of the DR center. The logic of high-availability switching and re-setup operations in all master center compute node services is the same as that of the ordinary single IDC mode.

#### 6.2.1.2. Failover

When the master compute node of the current master center fails, it will be automatically switched to the slave compute node of the master center according to the configuration of keepalived, the high-availability scheduling component. The compute node services in the equipment center of the DR center are not affected. The compute node high-availability service is only associated with the compute nodes in the IDC, and the judgment is made inside the current IDC. The failure re-setup method is the

same as the ordinary single IDC mode.

## 6.2.2. ConfigDB

### 6.2.2.1. Failover

When the master ConfigDB of the master center fails, same principle as that of the [data source](#) judgement, the standby master ConfigDB will automatically start service after catching up with replication. The master ConfigDB of the DR center will automatically reset the replication relation with the former master ConfigDB of the master center, and set up a replication relation with the standby master ConfigDB of the master center (the master ConfigDB of the DR center is as a slave), that is, the master-slave replication relation between hc01 and hc03 is cleared, and hc03 will set up a replication relation with hc02 as a slave.

When the failed master ConfigDB of the master center is restored, if the master-master replication relation of the ConfigDB in the master center is normal, you can enable the master ConfigDB through the parameter configuration page and perform the dynamic loading operation to switch the connection back to the failed former master ConfigDB. In the process, the master compute node will automatically reset the ConfigDB replication relations, and replication relation between the master ConfigDB newly enabled by the slave ConfigDB of the former master center and the master ConfigDB of the DR center will be reset up, that is, the replication relation between hc02 and hc03 is cleared, and hc03 will reset up a replication relation with hc02 as a slave.

## 6.2.3. Data source

### 6.2.3.1. Manual switching

Click [Switch] button corresponding to the master center on the "Node Management" page of the management platform, the compute node will automatically detect whether the replication relations between the master nodes of the current master center and the standby master nodes or the slave nodes are normal and whether the replication latency is within 10s. The switching can be successful when the conditions are met. At the same time, the master data sources of the DR center will be automatically set up with replication relations with the standby master nodes or slave nodes in the switched

master center, without manual intervention. That is, after switching from ds01 to ds02, the replication relation between ds01 and ds03 will be cleared, and ds03 as a slave will reset up a replication relation with ds02.

Data Source								Master/Slave Status	Master/Slave Switch	IDC Type	IP	DR status
DNID	Data Node Name	Hostname	Port	Database	Type	Status	Operation					
3	dn_01	192.168.240.82	3306	db01	Active Master	Available		Normal		Master center		
		192.168.240.82	3307	db01	Standby Slave	Available						
3	dn_01	192.168.210.98	3306	db01	Active Master	Available		Normal		DR center		Normal
		192.168.210.98	3307	db01	Standby Slave	Available						
15	dn_04	192.168.240.82	3308	db01	Active Master	Available		Normal		Master center		
		192.168.240.82	3309	db01	Standby Master	Available						
15	dn_04	192.168.210.98	3308	db01	Active Master	Available		Normal		DR center		Normal
		192.168.210.98	3309	db01	Standby Master	Available						

Note: If the switched data source of the master center is with a master-slave replication relation, the master data source of the former master center will be set unavailable after manual switching. It is required to set up a replication relation in the master center between the data source and the master data source, and to ensure the catching up with replication, thus the dynamic loading can be switched back to the former master data source in the master center, and the disaster recovery relation will be reset automatically without manual intervention. That is, when ds01 and ds02 are in a master-slave relation instead of a master-master relation, ds01 is manually switched to ds02 (slave only), and ds01 is set as unavailable. When ds01 is enabled, only if ds01 has a replication relation with ds02, ds01 is the slave of ds02, and there is no replication latency and other effects can the dynamic loading be successful and ds01 be enabled.

### 6.2.3.2. Failover

When the master data source fails, the service will be automatically started after the standby master data source catches up with replication (whether to wait for the replication catching up is controlled by the parameter `waitForSlaveInFailover`, waiting is enabled by default), and the master data source of the DR center will automatically reset the replication relation with the master data source of the former master center and set up a replication relation with the standby master ConfigDB of the master center (the master data source in the DR center is the slave).

When the replication relation between the failed data sources in the master center is master-slave instead of master-master, if you want to promote the former failed data source to be the current active data source, the operation mode is consistent with that of the ConfigDB master/slave cases. You need to set up the needed replication return circuit in the master center first, and ensure that there are no replication exceptions and the former master data source can be return to after dynamic loading.

## 6.3. Failure only inside the DR center

This section mainly describes the function usage when high-availability switching (including manual switching or failover) occurs in the core service components in the DR center in a cluster environment where services are normally provided.

### 6.3.1. Compute node

As the DR center does not provide real database services, it is recommended to disable keepalived, the master/slave high availability components of the DR center, during day-to-day operation and maintenance to avoid unnecessary recovery operations caused by VIP drift in the DR center. If keepalived do not provide services, when the compute node in the DR center fails, regardless of the master/slave roles, you only need to restart the service in an available operation environment.

If the keepalived service in the DR center needs to be retained due to special requirements, when the master compute node in the DR center fails, the VIP will drift to the server where the slave compute node is located after the keepalived detection timeout of the compute node service availability. At this time, only when VIP is guaranteed to go back to the master compute node in the former DR center can the master compute node back up normally. That is, disable the keepalived service in the slave compute node, ensure that the master/slave compute nodes are enabled, and then enable in turn the keepalived service in master compute node and the keepalived service in slave compute node.

## 6.3.2. ConfigDB

### 6.3.2.1. Failover

After the master ConfigDB of the DR center fails, the standby master ConfigDB of the DR center will be automatically set as unavailable. When the failure is restored, you need to click the [Enable] button on the "[Parameter Configuration](#)" page of management platform before the dynamic loading takes effect. When the standby master ConfigDB in the DR center is enabled, its master ConfigDB must be enabled; otherwise, the operation is invalid. If only the slave ConfigDB of the DR center fails, only the slave ConfigDB will be marked as unavailable and will not affect other services.

## 6.3.3. Data source

### 6.3.3.1. Failover

Same logic as that of ConfigDB. After the master data source in the DR center fails, other slave data sources in the IDC related to the failed master data source are also remarked as unavailable. That is, when ds03 fails and is marked as unavailable, ds04 is also marked as unavailable. After the service is restored, when the failed data source in the master center is enabled, the master data source must be enabled to enable the slave data source.

The master data sources of the DR center provide services normally. However, if the standby data sources of the DR center fail, only the standby data sources and their slaves will be marked as unavailable and the current used data source is not affected. That is, when ds04 fails and is marked as unavailable, ds03 is not affected; but if ds04 has other slaves such as ds05, then ds05 will also be marked as unavailable. When the data source marked as unavailable is enabled, ensure that the former master-slave replication synchronization status has no exceptions.

When the failed data source is enabled, it takes effect after the dynamic loading. The data sources in the DR center do not support manual switching.

After the manual switching and failover of all data sources and ConfigDBs, the compute nodes automatically adjust the value of rpl\_semi\_sync\_master\_wait\_for\_slave\_count as correct.

## 6.4. IDC Failure

This section mainly describes precautions and instructions when master center fails in the IDC level, DR center is switched as the active IDC and provides services during a compute node cluster is providing services normally, taken the compute node cluster in the master/slave mode as an example. This description mainly describes the steps of manual switching and repair of IDC. If automatic switching, repair and removal of the IDC through the management platform is wanted, on the basis of ensuring that the management platform version is 2.5.6 and above, you can refer to the [Distributed Transactional Database HotDB Server - V2.5.6 \[Visualization of Cross-IDC DR Switching\] Function Manual](#).

### 6.4.1. Master center fails and DR center is switched as the active IDC

When the compute node services of the master center fail irreparably, you can manually determine whether to switch to the DR center. When you decide to switch to the DR center, you need to refer to the description in the following sections for manual pre-judgment and inspection to ensure that the switch can be successful.

#### 6.4.1.1. Switch Pre-detection

- Check that all the services and processes of master and slave compute nodes in the master center have stopped and exited.
- Check that there is no abnormality in the service status of the master data source of each data node and master ConfigDB in the DR center, and there is no abnormality in the replication status between master center and DR center.(The compute node in the DR center will also automatically check when it is started,

but manual inspection in advance can increase the probability of successful failover.)

- If the keepalived service in the DR center has been started before, you need to check whether the VIP of the keepalived in the DR center is on the current active compute node. (if there is no high availability of compute nodes, you can skip this step)

#### 6.4.1.2. Switch Process

After confirming the switch pre-detection, you can execute the online\_dr command on the management port of compute nodes in the DR center. After receiving the command, the compute node will make the following judgments in order:

1. Send the offline command to the master compute node in the master center again to try to avoid multiple compute node services running simultaneously;
  2. Wait for all the master data sources and master ConfigDB of the DR center to catch up with the replication;
  3. Automatically clear all the data sources and ConfigDBs replication relation between the master center and the DR center;
  4. Mark all data sources and ConfigDBs in the master center as unavailable;
  5. After passing other verifications that need to be done when start service port, start the service port of master computer node in the DR center to provide services;
  6. At the same time, the replication relation between data sources in a single data node is automatically determined by the compute node. The compute node will set up or reset the replication relation in the DR center according to the configuration roles of data sources in the DR center.
- If the data source is configured as a “standby master”, compute node will compare the GTID position: When the master data source has more or equal GTID than the standby master data source, compute node will automatically set up a replication between these two data sources (the standby master data source is the host) so that there is a master-master replication for this data node; when the master data source has fewer GTID than the standby master data source(generally, this situation does

- not occur, and it is more possible when human operation involved), compute node will perform reset slave all on the standby master data source and set the standby master data source unavailable;
- If other data sources in the DR center are configured as “slave”, then only the GTID replication position will be detected. If the master data source has less GTID than the slave data source, compute node will perform reset slave all on the slave data source to clean up replication relations and set the slave data source as unavailable;
  - Other replication relations (such as master-master with slave(s)) will be determined same as above whether to set up or reset replication relation based on specific role configurations.
7. If currently there is high-availability compute node services in the DR center, you need to continue to execute the enable\_online command on the management port of the standby compute node in the DR center to ensure that when the current master compute node fails, you can perform compute node service switchover in the DR center according to high-availability rules of keepalived component.
8. Finally, check whether the keepalived status of the DR center is normal. The VIP is on the current master compute node, and there is no abnormality in the service status and role status of the master and standby keepalived, master and standby compute nodes.
9. If necessary, perform SQL operations on the compute nodes that are currently providing services to confirm that the services are normal

### 6.4.2. IDC failure repair and failover

At the current stage, the product only supports intelligent process of switching from the master center to DR center. The switch back of service in IDC level, cross-IDC high availability reconstruction and role of IDC switching will be launched in the future. Such scenarios can only be handled manually.

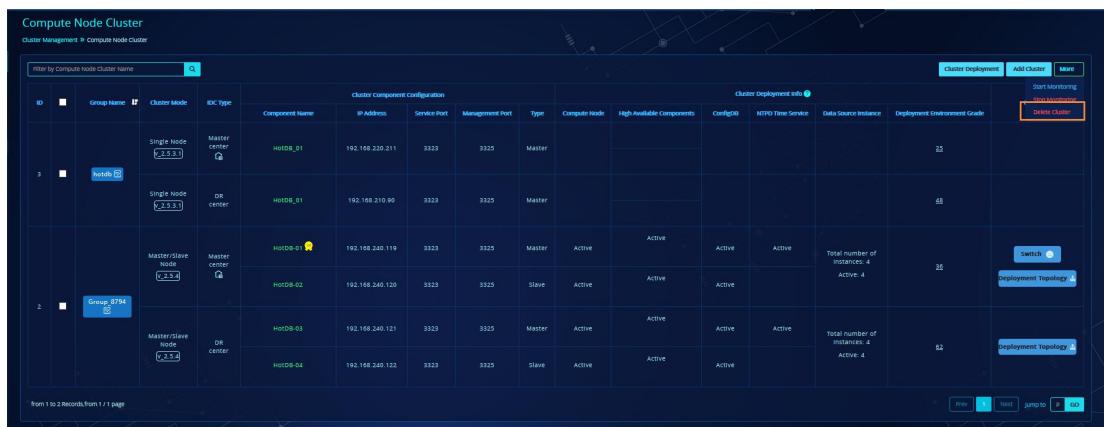
#### 6.4.2.1. Scenario One: Abandon the original master center

After the master center fails and the DR center is successfully switched, if the data in

the original master center cannot be recovered, you have to abandon the monitoring information of the master center and DR center, then re-add the active IDC as a new compute node cluster to ensure that the management platform can monitor it normally. Otherwise, some garbage data will be left to affect the current monitoring status of the active IDC. The necessary steps are as follows:

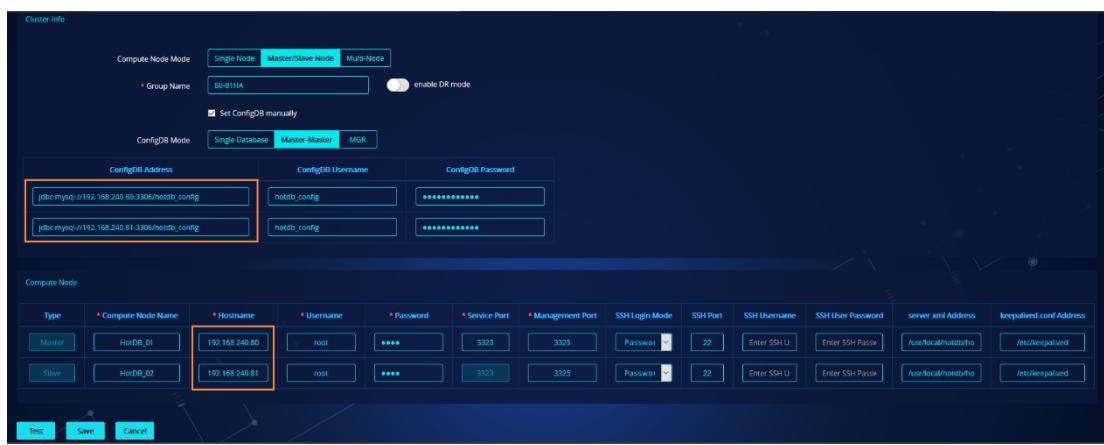
Step 1: Make a data backup of the ConfigDB of the management platform that is currently providing the service, and the ConfigDB of compute nodes, so that they can be restored after misuse;

Step 2: Delete the original cluster in the DR mode through the "Cluster Management" -> "Computer Node Cluster" page of the administrator interface in the management platform;



Cluster Component Configuration												Cluster deployment info				Start Monitoring		
ID	Group Name	Cluster Mode	IOC Type	Component Name	IP Address	Service Port	Management Port	Type	Compute Node	High Available Components	ConfigDB	HTDB Time Service	Data Source Instance	Deployment Environment Grade	Action			
3	hotdb_01	Single Node [1,2,3,4]	Master center [1]	HotDB_01	192.168.220.211	3323	3325	Master								2.0		
				HotDB_02	192.168.210.90	3323	3325	Master										2.0
2	Group: 8794	Master/Slave Node [2,3,4]	Master center [1]	HotDB_03	192.168.240.119	3323	3325	Master	Active	Active	Active			Total number of instances: 4		2.0		
				HotDB_04	192.168.240.120	3323	3325	Slave	Active	Active	Active	Active	Active	Active			Active: 4	2.0
				Master/Slave Node [2,3,4]	DR center	HotDB_05	192.168.240.121	3323	3325	Master	Active	Active	Active			Total number of instances: 4		2.0
						HotDB_06	192.168.240.122	3323	3325	Slave	Active	Active	Active	Active	Active	Active		

Step 3: Still through the "Cluster Management" -> "Compute Node Cluster" page, click [Add Cluster] to add the active IDC as a new compute node cluster and add the ConfigDB information after the disaster recovery switchover (as long as the compute node management port is currently connectable, no manual configuration is required here by default).

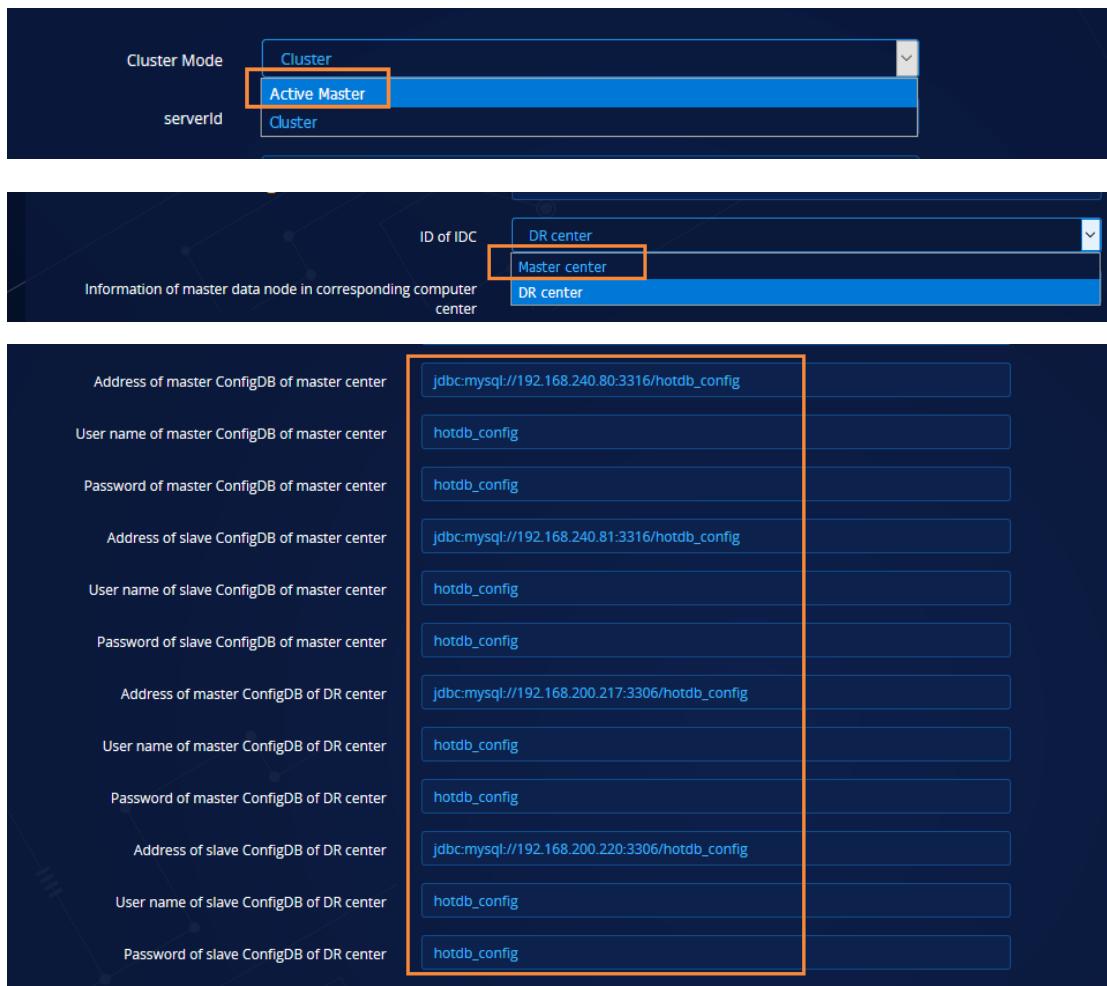


Cluster Info											
Compute Node Mode			ConfigDB Mode			Compute Node					
Single Node			Master-Slave Node			Multi-Node					
Group Name: 8794			IP Address: 192.168.1.104			enable DR mode					
<input checked="" type="checkbox"/> Set ConfigDB manually			<input type="checkbox"/> Single Database			<input type="checkbox"/> Master-Master					
ConfigDB Address:			ConfigDB Username:			ConfigDB Password:					
jdbc:mysql://192.168.240.80:3306/hotdb_config			hotdb_config			*****					
jdbc:mysql://192.168.240.81:3306/hotdb_config			hotdb_config			*****					
Compute Node											
Type	Compute Node Name	Hostname	Username	Password	Service Port	Management Port	SSH Login Mode	SSH Port	SSH Username	SSH User Password	Server XML Address
Master	HotDB_01	192.168.240.80	root	*****	3323	3325	Passwor	22	Enter SSH U	Enter SSH Pass	/usr/local/hotdb/bin
Slave	HotDB_02	192.168.240.81	root	*****	3323	3325	Passwor	22	Enter SSH U	Enter SSH Pass	/usr/local/hotdb/bin
<input type="button" value="Test"/> <input type="button" value="Save"/> <input type="button" value="Cancel"/>											

Step 4: Add user privileges to the new cluster for general users

Step 5: Log in to the management platform as a general user, select the newly-added cluster

Step 6: Go to the "Configuration"->"Parameter Configuration" page and modify the related configuration parameters as shown in red below: select HA as cluster mode and select master center as IDC ID (here is the preparation for [scenario two](#)). The ConfigDB connection information should be filled and saved according to the actual current ConfigDB information;



Address of master ConfigDB of master center	jdbc:mysql://192.168.240.80:3316/hotdb_config
User name of master ConfigDB of master center	hotdb_config
Password of master ConfigDB of master center	hotdb_config
Address of slave ConfigDB of master center	jdbc:mysql://192.168.240.81:3316/hotdb_config
User name of slave ConfigDB of master center	hotdb_config
Password of slave ConfigDB of master center	hotdb_config
Address of master ConfigDB of DR center	jdbc:mysql://192.168.200.217:3306/hotdb_config
User name of master ConfigDB of DR center	hotdb_config
Password of master ConfigDB of DR center	hotdb_config
Address of slave ConfigDB of DR center	jdbc:mysql://192.168.200.220:3306/hotdb_config
User name of slave ConfigDB of DR center	hotdb_config
Password of slave ConfigDB of DR center	hotdb_config

Step 7: Go to the "Configuration"->"Node Management" page to delete the data stored for the original master center.(You can choose to search by IP, or you can choose to search by unavailable status, but you need to be careful not to delete the data used by the current active IDC by mistake)

**Node Management**

Configuration > Node Management

All Nodes (multiple choices)		85	Search							refresh Master/Slave Status	Data Source Group	Add Node	High Availability Settings	Batch Operation	
DNIID	Data Node Name			Hostname	Port	Database	Type	Status	Operation	Master/Slave Status	Master/Slave Switch	IDC Type	Batch Edit	Batch Delete	Batch Enable
3	dn_01	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	192.168.240.82	3306	db01	Active Master	Unavailable	<span>Info</span> <span>Checkmark</span> <span>Warning</span>	Normal		Master center			
		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	192.168.240.82	3307	db01	Standby Slave	Unavailable	<span>Info</span> <span>Checkmark</span> <span>Warning</span>	Normal		Master center			
15	dn_04	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	192.168.240.82	3308	db01	Active Master	Unavailable	<span>Info</span> <span>Checkmark</span> <span>Warning</span>	Normal		Master center			
		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	192.168.240.82	3309	db01	Standby Master	Unavailable	<span>Info</span> <span>Checkmark</span> <span>Warning</span>	Normal		Master center			

from 1 to 2 Records, from 1 / 1 page

● Active ● Slave ● Unavailable

Prev 1 Next Jump to P GO

**Node Management**

Configuration > Node Management

All Nodes (multiple choices)		Filter by data source name	Search							refresh Master/Slave Status	Data Source Group	Add Node	High Availability Settings	Batch Operation	
DNIID	Data Node Name			Hostname	Port	Database	Type	Status	Operation	Master/Slave Status	Master/Slave Switch	IDC Type	Batch Edit	Batch Delete	Batch Enable
3	dn_01	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	192.168.210.98	3306	db01	Active Master	Unavailable	<span>Info</span> <span>Checkmark</span> <span>Warning</span>	Normal		DR center	Unknown		
		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	192.168.210.98	3307	db01	Standby Slave	Unavailable	<span>Info</span> <span>Checkmark</span> <span>Warning</span>	Normal		DR center	Unknown		
15	dn_04	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	192.168.210.98	3308	db01	Active Master	Unavailable	<span>Info</span> <span>Checkmark</span> <span>Warning</span>	Normal		DR center	Unknown		
		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	192.168.210.98	3309	db01	Standby Master	Unavailable	<span>Info</span> <span>Checkmark</span> <span>Warning</span>	Normal		DR center	Unknown		

from 1 to 2 Records, from 1 / 1 page

● Active ● Slave ● Unavailable

Prev 1 Next Jump to P GO

Step 8: Enter the master ConfigDB of the current active IDC and replace all the IDC ID related in the hotdb\_datasource table with 1 (1 represents the master center, which has no practical meaning without disaster recovery)

```
update hotdb_datasource set idc_id=1;
```

Step 9: Enter the master ConfigDB of the current active IDC, then check the result of the hotdb\_config\_info table and make sure that v value of the row whose k column is hotdb\_master\_config\_status is 1. (The data of this row represents whether the current master ConfigDB is available. It will not be updated after IDC switchover, just check it and make sure that the current ConfigDB information can be read normally.) If it is not 1, update the row of data to 1.

```
root@localhost:hotdb_config 5.7.25-log 07:47:19> select * From hotdb_config_info where k='hotdb_master_config_status' ;
```

k	v	description

```
+-----+-----+
| hotdb_master_config_status | 1 | Master ConfigDB status |
+-----+-----+
1 row in set (0.00 sec)
```

Step 10: Execute reload;

Step 11: Check whether the service status of current compute nodes, ConfigDB and data sources conforms to the current actual situation.

#### **6.4.2.2. Scenario Two: Make the original master center as the standby IDC**

After the original DR center is switched as the current active IDC, when the original components in the original master center can be restored, you can consider to make the original master as the current standby IDC. In the case, you must manually perform the operations described in this section based on the completion of operations described in scenario one. Subsequent configuration modification operations, the operation steps are as follows (note that it must be based on the description of scenario one):

Step 1: Make a data backup of the ConfigDB of the management platform that is currently providing the service, and the ConfigDB of compute nodes, so that they can be restored after misuse;(try not to conflict with the previous backup file).

Step 2: Check whether the compute node service process or keepalived process of the original master center exists. If it exists, you can directly close them all without affecting the compute node services that are currently providing services (in the case of multi node cluster mode, there is no need to pay attention to keepalived, and LVS control is available.); it is mainly to avoid unnecessary operation and maintenance costs for mis-operation of this compute node.

Step 3: Establish replication relation of the ConfigDBs and data sources between the current active IDC and the original master center and swap the replication hosts. The reference operation steps in the swap process are as follows:

- After the disaster recovery switchover, when the service of original DR center

starts and become the current active IDC, under normal circumstances, compute node will automatically establish a master-master replication between data sources according to the data source configuration in the current active IDC and establish a master-master replication between ConfigDBs by default. No special operation is required. that is, ds03 and ds04, hc03 and hc04 have master-master replication;

- It is necessary to manually clean up the replication relation between the data sources and ConfigDBs of the original master center, that is, execute reset slave all. At the same time, if the data of the original master center can be abandoned or must be abandoned, clear the data and perform the reset master operation;
- Export the current data of the data sources in the current active IDC and import it to the target master data source and the master ConfigDB that are about to be set up replication in the original master center. (for data export and replication setup, please refer to the master-slave replication setup recommended steps);
- Establish a master-slave replication relation between the master data source/ConfigDB in the active IDC and the master data source/ConfigDB in the original master center main. The data sources/ConfigDB in the original master center are slaves. That is, a master-slave relationship is established between ds01 and ds03 while ds01 is a slave; a master-slave relationship is established between hc03 and hc01 while hc01 is a slave.
- Establish a replication relation between the data source/ConfigDB in the original master center, that is, a master-slave relationship is established between ds01 and ds02 while ds02 is the slave; a master-slave relationship is established between hc01 and hc02 while hc02 is the slave. Note: For the sake of insurance, before setting up a replication relation, it is recommended that export the source data from the master data source in the original master center ds01 and the ConfigDB hc01 and then import into ds02 and hc02, finally set up a replication relation.
- If semi-synchronous replication is enabled, you need to manually modify the value of rpl\_semi\_sync\_master\_wait\_for\_slave\_count of the master data source/ConfigDB in the current active IDC to a correct number as required to match the number of its real slaves. Refer to the following statement:

```
set global rpl_semi_sync_master_wait_for_slave_count=XXX;
```

Step 4: Modify the configuration file server.xml of the original master center and the current active IDC to the configuration of the new state of DR mode, refer to the following:

**Master compute node in the current active IDC:**

```
<property name="drUrl">jdbc:mysql://192.168.220.181:3306/hotdb_config</property><!--  
Configuration database address in DR center -->  
  
<property name="drBakUrl">jdbc:mysql://192.168.220.182:3306/hotdb_config</property><!--  
Slave ConfigDB address in DR IDC -->  
  
<property name="haMode">2</property><!-- High-availability mode, 0:HA, 1:Cluster, 2:HA in  
master center, 3:HA in DR center -->  
  
<property name="idcId">1</property><!-- ID of IDC, 1:master center, 2:DR center -->
```

**Standby compute node in the current active IDC:**

```
<property name="drUrl">jdbc:mysql://192.168.220.181:3306/hotdb_config</property><!--  
Configuration database address in DR center -->  
  
<property name="drBakUrl">jdbc:mysql://192.168.220.182:3306/hotdb_config</property><!--  
Slave ConfigDB address in DR center -->  
  
<property name="haMode">2</property><!-- High-availability mode, 0:HA, 1:Cluster, 2:HA in  
master center, 3:HA in DR center -->  
  
<property name="idcId">1</property><!-- ID of IDC, 1:master center, 2:DR center -->
```

**Master compute node in the original master center:**

```
<property name="url">jdbc:mysql://192.168.220.183:3306/hotdb_config</property><!-- ConfigDB  
address-->  
  
<property name="bakUrl">jdbc:mysql://192.168.220.184:3306/hotdb_config</property><!-- Slave  
ConfigDB address -->
```

```

<property name="drUrl">jdbc:mysql://192.168.220.181:3306/hotdb_config</property><!--
Configuration database address in DR center -->

<property name="drBakUrl">jdbc:mysql://192.168.220.182:3306/hotdb_config</property><!!--
Slave ConfigDB address in DRIDC -->

<property name="haMode">3</property><!!-- High-availability mode, 0:HA, 1:Cluster, 2:HA in
master center, 3:HA in DR center -->

<property name="idcId">2</property><!!-- ID of IDC, 1:master center, 2:DR center -->

Standby compute node in the original master center:

<property name="url">jdbc:mysql://192.168.220.183:3306/hotdb_config</property><!!-- ConfigDB
address-->

<property name="bakUrl">jdbc:mysql://192.168.220.184:3306/hotdb_config</property><!!-- Slave
ConfigDB address -->

<property name="drUrl">jdbc:mysql://192.168.220.181:3306/hotdb_config</property><!!--
Configuration database address in DR center -->

<property name="drBakUrl">jdbc:mysql://192.168.220.182:3306/hotdb_config</property><!!--
Slave ConfigDB address in DRIDC -->

<property name="haMode">3</property><!!-- High-availability mode, 0:HA, 1:Cluster, 2:HA in
master center, 3:HA in DR center -->

<property name="idcId">2</property><!!-- ID of IDC, 1:master center, 2:DR center -->

```

Step 5: Log in to the master ConfigDB in the current active IDC, and set the availability of the ConfigDB of original master center to 1. (after disaster recovery switchover in IDC level, the compute node will automatically mark the status of ConfigDB in master center as unavailable. It needs to be enabled here to ensure that after reloading, the compute node can read the correct ConfigDB information according to the current latest configuration.) Also, check whether the status of other ConfigDB in this table is in accordance with the actual availability status:

**Update the availability of the ConfigDB of master center:**

```
update hotdb_config_info set v='1' where k='hotdb_biz_idc_config_ok' or  
k='hotdb_master_config_status';
```

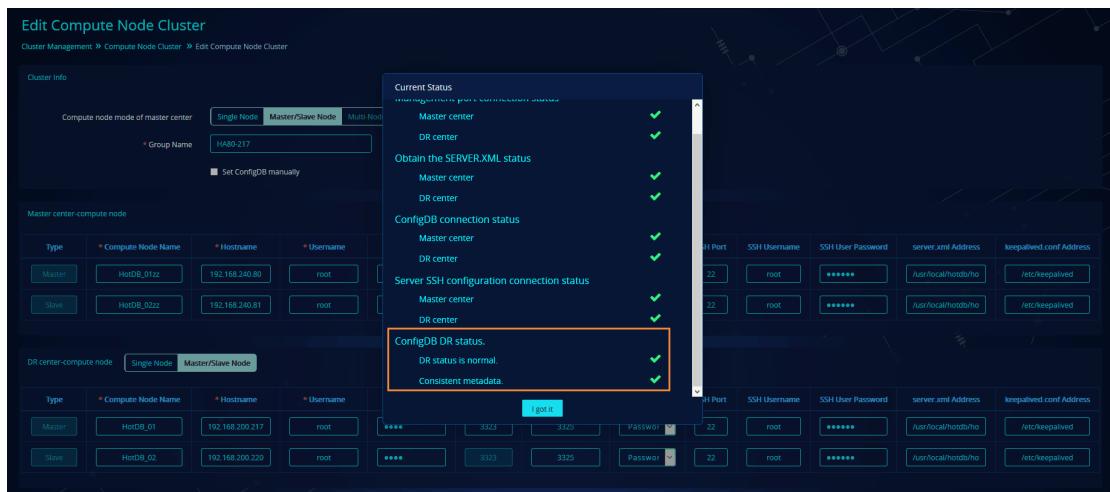
Check the availability of other ConfigDB:

```
select * From hotdb_config_info limit 1,5;
```

k	v	description
hotdb_biz_idc_config_ok	1	ConfigDB in master center ok or not
hotdb_dr_idc_config_ok	1	ConfigDB in DR center ok or not
hotdb_dr_idc_master_config_status	1	Master ConfigDB status in DR center
hotdb_dr_idc_slave_config_ok	1	standby ConfigDB in DR center ok or not
hotdb_master_config_status	1	Master ConfigDB status

Step 6: Start the services of the active and standby compute nodes of the original master center. (Keepalived don't need to be started temporarily. If a failover at IDC level occurs afterwards, you can start keepalived after restarting services.)

Step 7: Log in to the management platform as a manager user and go to the "Cluster Management"->"Compute Node Cluster" page to turn on the DR mode for this cluster, and configure the original master center as the current standby IDC. Save after passing test.

Step 8: Log in to the management platform as a general user account and select the newly added cluster in DR mode.

Step 9. Go to the "Configuration"-> "Node Management" page to add data nodes and data sources in the DR center.

**Node Management**

Configuration > Node Management

		Data Source						Master/Slave Status	Master/Slave Switch	IDC Type	DR status
DNID	Data Node Name	Hostname	Port	Database	Type	Status	Operation				
1	dn_01	192.168.240.119	3310	db01	Active Master	Available				Master center	
		192.168.240.120	3310	db01	Standby Master	Available					
1	dn_01	192.168.240.121	3310	db01	Active Master	Available				DR center	Normal
		192.168.240.122	3310	db01	Standby Master	Available					
2	dn_02	192.168.240.119	3311	db01	Active Master	Available				Master center	
		192.168.240.120	3311	db01	Standby Master	Available					
2	dn_02	192.168.240.121	3311	db01	Active Master	Available				DR center	Normal
		192.168.240.122	3311	db01	Standby Master	Available					
3	dn_03	192.168.220.119	3310	db02	Active Master	Available					Master center

from 1 to 5 Records, from 1 / 1 page

Prev Next Jump to GO

● Active ● Slave ● Unavailable

**Add Node**

Configuration > Node Management > Add Node

<input checked="" type="checkbox"/> Add Data Source Only																																																											
Number of Data Nodes:	4	Node Prefix:	db_	Number of Encoding Bits:	1																																																						
Start Encoding:	8																																																										
Data Node Type:	Master	Data Source Group:	Not Use	Create Data Source Group	<input checked="" type="checkbox"/> Automatic Adaptation of Switching Rule																																																						
<input checked="" type="checkbox"/> DR center <input type="checkbox"/> automatically build DR relation																																																											
<table border="1"> <thead> <tr> <th>Data Node Name</th> <th>Data Source Type</th> <th>Data Source Group</th> <th>Data Source Name</th> <th><input checked="" type="checkbox"/> Automatic Generation</th> <th></th> </tr> </thead> <tbody> <tr> <td>dn_03</td> <td>Active Master</td> <td>Not Use G</td> <td>192.168.240.111_3310_db03</td> <td></td> <td><input checked="" type="checkbox"/> Hostname</td> </tr> <tr> <td>dn_03</td> <td>Standby Master</td> <td>Not Use G</td> <td>192.168.240.112_3310_db03</td> <td></td> <td><input checked="" type="checkbox"/> Port</td> </tr> <tr> <td>dn_04</td> <td>Active Master</td> <td>Not Use G</td> <td>192.168.240.112_3310_db04</td> <td></td> <td><input checked="" type="checkbox"/> Connection User</td> </tr> <tr> <td>dn_04</td> <td>Standby Master</td> <td>Not Use G</td> <td>192.168.240.112_3310_db04</td> <td></td> <td><input checked="" type="checkbox"/> Connection User Password</td> </tr> <tr> <td>dn_05</td> <td>Active Master</td> <td>Not Use G</td> <td>192.168.240.112_3310_db05</td> <td></td> <td><input checked="" type="checkbox"/> Database Name</td> </tr> <tr> <td>dn_05</td> <td>Standby Master</td> <td>Not Use G</td> <td>192.168.240.112_3310_db05</td> <td></td> <td><input checked="" type="checkbox"/> Backup User</td> </tr> <tr> <td>dn_07</td> <td>Active Master</td> <td>Not Use G</td> <td>192.168.240.121_3310_db06</td> <td></td> <td><input checked="" type="checkbox"/> Backup User Password</td> </tr> <tr> <td>dn_07</td> <td>Standby Master</td> <td>Not Use G</td> <td>192.168.240.122_3310_db06</td> <td></td> <td></td> </tr> </tbody> </table>						Data Node Name	Data Source Type	Data Source Group	Data Source Name	<input checked="" type="checkbox"/> Automatic Generation		dn_03	Active Master	Not Use G	192.168.240.111_3310_db03		<input checked="" type="checkbox"/> Hostname	dn_03	Standby Master	Not Use G	192.168.240.112_3310_db03		<input checked="" type="checkbox"/> Port	dn_04	Active Master	Not Use G	192.168.240.112_3310_db04		<input checked="" type="checkbox"/> Connection User	dn_04	Standby Master	Not Use G	192.168.240.112_3310_db04		<input checked="" type="checkbox"/> Connection User Password	dn_05	Active Master	Not Use G	192.168.240.112_3310_db05		<input checked="" type="checkbox"/> Database Name	dn_05	Standby Master	Not Use G	192.168.240.112_3310_db05		<input checked="" type="checkbox"/> Backup User	dn_07	Active Master	Not Use G	192.168.240.121_3310_db06		<input checked="" type="checkbox"/> Backup User Password	dn_07	Standby Master	Not Use G	192.168.240.122_3310_db06		
Data Node Name	Data Source Type	Data Source Group	Data Source Name	<input checked="" type="checkbox"/> Automatic Generation																																																							
dn_03	Active Master	Not Use G	192.168.240.111_3310_db03		<input checked="" type="checkbox"/> Hostname																																																						
dn_03	Standby Master	Not Use G	192.168.240.112_3310_db03		<input checked="" type="checkbox"/> Port																																																						
dn_04	Active Master	Not Use G	192.168.240.112_3310_db04		<input checked="" type="checkbox"/> Connection User																																																						
dn_04	Standby Master	Not Use G	192.168.240.112_3310_db04		<input checked="" type="checkbox"/> Connection User Password																																																						
dn_05	Active Master	Not Use G	192.168.240.112_3310_db05		<input checked="" type="checkbox"/> Database Name																																																						
dn_05	Standby Master	Not Use G	192.168.240.112_3310_db05		<input checked="" type="checkbox"/> Backup User																																																						
dn_07	Active Master	Not Use G	192.168.240.121_3310_db06		<input checked="" type="checkbox"/> Backup User Password																																																						
dn_07	Standby Master	Not Use G	192.168.240.122_3310_db06																																																								
<input type="button" value="Connection Test"/> <input type="button" value="Save and Return"/> <input type="button" value="Cancel"/>																																																											

Step 10: Check whether the replication status or DR replication status in the current active IDC is abnormal. You can check whether the replication status of the data sources or ConfigDBs are normal through the "Node Management" and "Configuration Verification" pages.

Step 11: Dynamic loading.

Step 12. Check whether the topologic graph and various monitoring indicators are normal, and check whether all the heartbeat is monitored by executing the show @@ heartbeat command through management port of compute node in the current active IDC.

### 6.4.2.3. Scenario Three: Make the original master center as the active IDC

When a failover at IDC level occurs and the original DR center is switched as the current active IDC, if the service status of each component in the original master center can be restored and it needs to be switched back to the original master center, the actual roles of all the IDC remain the same with configuration, then you need to manually do the modifications as follows:

#### ➤ When compute nodes can be stopped

Step 1: Stop the management platform that is currently providing services, all compute node services, and keepalived service. (Note that no matter which IDC type is, the keepalived services corresponding to the standby compute node and the standby compute node service should be stopped first, and then the master keepalived service and master compute node service should be stopped).

Step 2: Wait the data sources in the current active IDC for catching up replication and clean up all replication relation between data sources or ConfigDB in the original master center or DR center.

Step 3: Export data from the master data source in the current active IDC. Based on this data, follow normal steps and requirements of setting up replication in the DR mode to rebuild the replication in both IDC and DR replication between master data source of these two IDC. That is, after the master center fails and the DR center is switched as the current active IDC, the replication relation should be: replication between ds01 and ds02 should be master-master, replication between ds03 and ds04 should be master-master while ds03 is the current active master. After the original master center is restored and its roles are made sure not to be changed, it is necessary to restore the replication relation based on the data of ds03 according to the standard steps of replication setup: replication between ds01 and ds02 should be master-master, replication between ds01 and ds03 should be master-slave(ds03 is a slave), replication between ds03 and ds04 should be master-slave(ds04 is a slave). All the replication

relation between data sources need to be rebuilt, as are the replication relation between ConfigDB.

Step 4: After confirming that there is no abnormality in the replication relation between the data sources or ConfigDBs, execute the following statement in the master ConfigDB (should be ds01 at this time) to set the availability of the ConfigDB of master center to 1. Also, check whether the status of other ConfigDB in this table is in accordance with the actual availability status:

**Update the availability of the ConfigDB of master center:**

```
update hotdb_config_info set v='1' where k='hotdb_biz_idc_config_ok' or
k='hotdb_master_config_status';
```

**Check the availability of other ConfigDB:**

```
select * From hotdb_config_info limit 1,5;
```

k	v	description
hotdb_biz_idc_config_ok	1	ConfigDB in master center ok or not
hotdb_dr_idc_config_ok	1	ConfigDB in DR center ok or not)
hotdb_dr_idc_master_config_status	1	Master ConfigDB status in DR center
hotdb_dr_idc_slave_config_ok	1	standby ConfigDB in DR center ok or not
hotdb_master_config_status	1	Master ConfigDB status

Step 5: After confirming that the first 4 steps are correct, start the compute nodes and keepalived service of the original master center and DR center according to the normal process; At this time, for the front-end application, the master data has been officially switched back, and the IDC type has not changed;

Step 6. Start the management platform and check whether the replication relation and

status in topologic graph are normal.

#### ➤ When compute nodes cannot be stopped

Referring to the steps described in Scenario one and Scenario two, you need to make the current active IDC as the master center of a new cluster and make the original master center as the DR center of this cluster. Then make another switchover at the IDC level and rebuild the DR relation. That is, HotDB-01 and HotDB-03 are the master compute node of the master center and DR center respectively. When a failover at the IDC level occurs at the first time, HotDB-03 is switched as the master compute node of the current active IDC. If you want to switch back to HotDB-01 without stopping services, you need to modify the IDC type of HotDB-01 as a DR center and HotDB-03 as a master center; then execute online\_dr command to switch HotDB-01 as the current active IDC and modify the IDC type of HotDB-01 as a master center and HotDB-03 as a DR center. It is equivalent to perform the steps described in Scenario one and Scenario two twice, and then rebuilding to make HotDB-01 as a master center and HotDB-03 as a DR center.

#### 6.4.2.4. Scenario Four: Make the original master center as the active IDC without deleting cluster

When using the operation methods described in the previous sections, once failover at the IDC level occurs, the historical monitoring data of the failed IDC will be deleted. In actual use, as long as the data in the master data can be recovered, users may not want to discard the existing cluster group information, but they may want the IDC switch guaranteed. At this time, the data of ConfigDB and the configuration file must be manually modified, and replication relation between data sources must be rebuilt to complete the IDC type swap. The reference steps are as follows:

Step 1: Under the environment of [premise explanation](#), a failover at the IDC level occurs, HotDB-03 and HotDB-04 are current master and slave compute nodes in the current active IDC and data of ds01, ds02, hc01, hc02 in the original master center can be recovered. At this time, make sure that the active and standby compute node services of the original master center have been shut down, that is, the services of HotDB-01 and HotDB-02 have been shut down;

Step 2: Make a data backup of the ConfigDB of the management platform that is currently providing the service, and the ConfigDB of compute nodes, so that they can be restored after misuse;(try not to conflict with the previous backup file).

Step 3: Check whether the compute node service process or keepalived process of the original master center exists. If it exists, you can directly close them all without affecting the compute node services that are currently providing services; it is mainly to avoid unnecessary operation and maintenance costs for mis-operation of this compute node.

Step 4: Establish replication relation of the ConfigDBs and data sources between the current active IDC and the original master center and swap the replication hosts. The reference operation steps in the swap process are as follows:

- After the disaster recovery switchover, when the service of original DR center starts and become the current active IDC, under normal circumstances, compute node will automatically setup a master-master replication between data sources according to the data source configuration in the current active IDC and also setup a master-master replication between ConfigDBs by default. No special operation is required. that is, ds03 and ds04, hc03 and hc04 have master-master replication;
- It is necessary to manually clean up the replication relation between the data sources and ConfigDBs of the original master center, that is, execute reset slave all. At the same time, if the data of the original master center can be abandoned or must be abandoned, clear the data and perform the reset master operation; (Note: In the previous step, if GTIDs in the original master center are qual or less than the original DR center, and the missing GTIDs are not in the gtid\_purged set of the master data sources in the original DR center. You can also try to set up replication without executing reset master. But after replication catchup, make sure to perform a master-slave data consistency detection to ensure that the data is truly consistent.)
- Export data of each master data source and each master ConfigDB in the current active IDC and import it to the target master data source and the master ConfigDB that are about to be set up replication in the original master center. (for data export and replication setup, please refer to the master-slave replication setup recommended steps);

- Setup a master-slave replication relation between the master data source/ConfigDB in the active IDC and the master data source/ConfigDB in the original master center main. The data sources/ConfigDB in the original master center are slaves. That is, a master-slave relationship is established between ds01 and ds03 while ds01 is a slave; a master-slave relationship is established between hc03 and hc01 while hc01 is a slave.
- Setup a replication relation between the data source/ConfigDB in the original master center, that is, a master-slave relationship is established between ds01 and ds02 while ds02 is the slave; a master-slave relationship is established between hc01 and hc02 while hc02 is the slave. Note: For the sake of insurance, when setting up a replication relation, it is recommended that export the source data from the master data source in the original master center ds01 and the ConfigDB hc01 and then import into ds02 and hc02, finally set up a replication relation.
- If semi-synchronous replication is enabled, you need to manually modify the value of rpl\_semi\_sync\_master\_wait\_for\_slave\_count of the master data source/ConfigDB in the current active IDC to a correct number as required to match the number of its real slaves. Refer to the following statement:

```
set global rpl_semi_sync_master_wait_for_slave_count=XXX;
```

Step 5: Modify the configuration file server.xml of the original master center and the current active IDC to the configuration of the new state of DR mode, refer to the following:

**Master compute node in the current active IDC:**

```
<property name="url">jdbc:mysql://192.168.220.183:3306/hotdb_config</property><!-- ConfigDB address -->

<property name="bakUrl">jdbc:mysql://192.168.220.184:3306/hotdb_config</property><!-- Slave ConfigDB address -->

<property name="drUrl">jdbc:mysql://192.168.220.181:3306/hotdb_config</property><!-- Configuration database address in DR center -->

<property name="drBakUrl">jdbc:mysql://192.168.220.182:3306/hotdb_config</property><!-- Slave ConfigDB address in DR center -->
```

```
<property name="haMode">2</property><!-- High-availability mode, 0:HA, 1:Cluster, 2:HA in master center, 3:HA in DR center -->
```

```
<property name="idcId">1</property><!-- ID of IDC, 1:master center, 2:DR center -->
```

**Standby compute node in the current active IDC:**

```
<property name="url">jdbc:mysql://192.168.220.183:3306/hotdb_config</property><!-- ConfigDB address-->
```

```
<property name="bakUrl">jdbc:mysql://192.168.220.184:3306/hotdb_config</property><!-- Slave ConfigDB address -->
```

```
<property name="drUrl">jdbc:mysql://192.168.220.181:3306/hotdb_config</property><!-- Configuration database address in DR center -->
```

```
<property name="drBakUrl">jdbc:mysql://192.168.220.182:3306/hotdb_config</property><!-- Slave ConfigDB address in DR center -->
```

```
<property name="haMode">2</property><!-- High-availability mode, 0:HA, 1:Cluster, 2:HA in master center, 3:HA in DR center -->
```

```
<property name="idcId">1</property><!-- ID of IDC, 1:master center, 2:DR center -->
```

**Master compute node in the original master center:**

```
<property name="url">jdbc:mysql://192.168.220.183:3306/hotdb_config</property><!-- ConfigDB address-->
```

```
<property name="bakUrl">jdbc:mysql://192.168.220.184:3306/hotdb_config</property><!-- Slave ConfigDB address -->
```

```
<property name="drUrl">jdbc:mysql://192.168.220.181:3306/hotdb_config</property><!-- Configuration database address in DR center -->
```

```
<property name="drBakUrl">jdbc:mysql://192.168.220.182:3306/hotdb_config</property><!-- Slave ConfigDB address in DRIDC -->
```

```
<property name="haMode">3</property><!-- High-availability mode, 0:HA, 1:Cluster, 2:HA in master center, 3:HA in DR center -->
```

```
<property name="idcId">2</property><!-- ID of IDC, 1:master center, 2:DR center -->
```

**Standby compute node in the original master center:**

```
<property name="url">jdbc:mysql://192.168.220.183:3306/hotdb_config</property><!-- ConfigDB address-->
```

```
<property name="bakUrl">jdbc:mysql://192.168.220.184:3306/hotdb_config</property><!-- Slave ConfigDB address -->
```

```
<property name="drUrl">jdbc:mysql://192.168.220.181:3306/hotdb_config</property><!-- Configuration database address in DR center -->
```

```
<property name="drBakUrl">jdbc:mysql://192.168.220.182:3306/hotdb_config</property><!-- Slave ConfigDB address in DRIDC -->
```

```
<property name="haMode">3</property><!-- High-availability mode, 0:HA, 1:Cluster, 2:HA in master center, 3:HA in DR center -->
```

```
<property name="idcId">2</property><!-- ID of IDC, 1:master center, 2:DR center -->
```

**Step 6: Log in to the master ConfigDB in the current active IDC, and set the availability of the ConfigDB of original master center to 1 and swap IDC ID:**

```
update hotdb_config_info set v='1' where k='hotdb_biz_idc_config_ok' or k='hotdb_master_config_status';

update hotdb_datasource t1 join hotdb_datasource t2 on(t1.idc_id=1 and t2.idc_id=2) set t1.idc_id=t2.idc_id, t2.idc_id=t1.idc_id;
```

**Step 7: Log in to ConfigDB of the management platform and execute the following SQL statement to swap IDC ID (make sure to replace the group\_id, which is consistent with the cluster ID at the cluster management page of management platform):**

```
UPDATE hotdb_info t1 JOIN hotdb_info t2 ON (t1.group_id = ? and t2.group_id = ? and t1.room_type=1 and t2.room_type=2) SET t1.room_type =
t2.room_type, t2.room_type=t1.room_type;
```

```

UPDATE host_info t1 JOIN host_info t2 ON (t1.group_id = ? and t2.group_id = ? and
t1.room_type=1 and t2.room_type=2) SET t1.room_type =
t2.room_type, t2.room_type=t1.room_type;

UPDATE host_soft_info t1 JOIN host_soft_info t2 ON (t1.group_id = ? and t2.group_id = ? and
t1.room_type=1 and t2.room_type=2) SET t1.room_type =
t2.room_type, t2.room_type=t1.room_type;

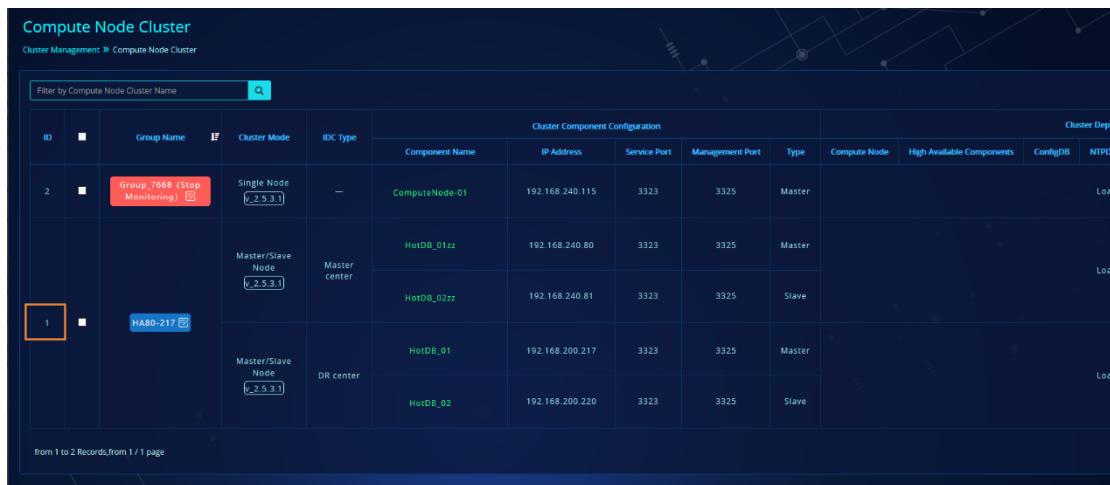
UPDATE hotdb_config t1 JOIN hotdb_config t2 ON (t1.group_id = ? and t2.group_id = ? and
t1.room_type=1 and t2.room_type=2) SET t1.room_type =
t2.room_type, t2.room_type=t1.room_type;

UPDATE env_check t1 JOIN env_check t2 ON (t1.group_id = ? and t2.group_id = ? and
t1.room_type=1 and t2.room_type=2) SET t1.room_type =
t2.room_type, t2.room_type=t1.room_type;

UPDATE mslog t1 JOIN mslog t2 ON (t1.group_id = ? and t2.group_id = ? and t1.room_type=1 and
t2.room_type=2) SET t1.room_type = t2.room_type, t2.room_type=t1.room_type;

```

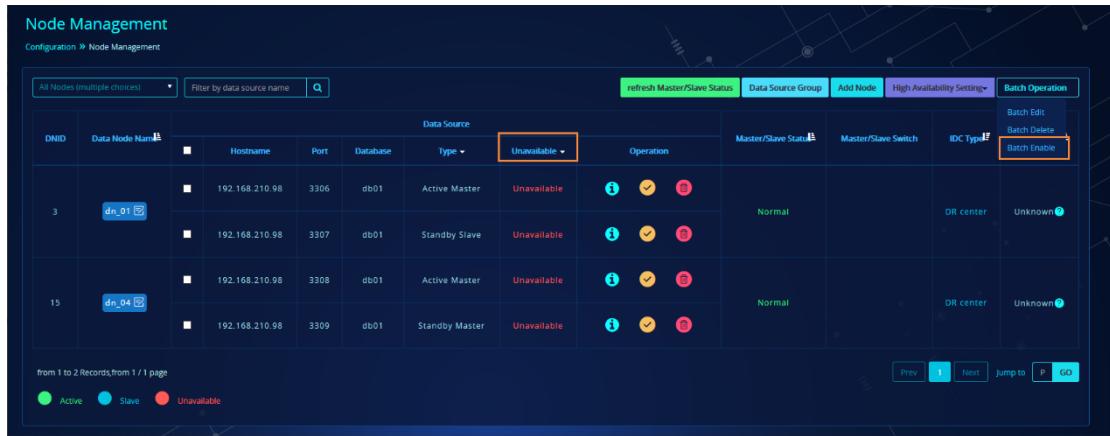
Example of page display after replacement:



ID	Group Name	IP	Cluster Mode	IDC type	Cluster Component Configuration						Cluster Dept		
					Component Name	IP Address	Service Port	Management Port	Type	Compute Node		High Available Components	ConfigDB
2	Group_7668 (Stop Monitoring)		Single Node v_2.5.3.1	-	ComputeNode-01	192.168.240.115	3323	3325	Master				Los
1	HAB0-217		Master/Slave Node v_2.5.3.1	Master center	HotDB_01zz	192.168.240.80	3323	3325	Master				Los
					HotDB_02zz	192.168.240.81	3323	3325	Slave				
			Master/Slave Node v_2.5.3.1	DR center	HotDB_01	192.168.200.217	3323	3325	Master				Los
					HotDB_02	192.168.200.220	3323	3325	Slave				

from 1 to 2 Records, from 1 / 1 page

Step 8: Log in to the management platform and enable the data sources in the original master center (current DR center) that is automatically disabled:



Step 9: Check whether the replication status and DR replication status in the current active IDC is abnormal. You can check the replication status of data sources and ConfigDBs through the "Node Management" and "Configuration Checking" pages.

Step 10: Reload

Step 11: Check whether the topologic graph and various monitoring indicators are normal, and check whether all the heartbeat is monitored by executing the show @@heartbeat command through management port of compute node in the current active IDC and check whether IDC type has been swapped by executing the show @@configurl or show @@datanode command. For example:

@@heartbeat												
dn	ds_id	ds_type	host	port	db	retry	status	period	execute_time	last_active_time	stop	
48	208	master	192.168.220.183	3310	db2531	0.0	idle	4000	0,0,0	2019-12-14 18:51:32	false	
48	209	master's slave	192.168.220.184	3310	db2531	0.0	idle	4000	0,0,0	2019-12-14 18:51:32	false	
48	201	master's slave	192.168.220.182	3310	db2531	0.0	idle	4000	0,0,0	2019-12-14 18:51:32	false	
48	200	master	192.168.220.181	3310	db2531	0.0	idle	4000	0,0,0	2019-12-14 18:51:32	false	
46	205	master's slave	192.168.220.184	3308	db2531	0.0	idle	4000	0,0,0	2019-12-14 18:51:32	false	
46	204	master	192.168.220.183	3308	db2531	0.0	idle	4000	0,0,0	2019-12-14 18:51:32	false	
46	185	master	192.168.220.181	3308	db2531	0.0	idle	4000	0,0,0	2019-12-14 18:51:32	false	
46	186	master's slave	192.168.220.182	3308	db2531	0.0	idle	4000	0,0,0	2019-12-14 18:51:32	false	
47	206	master	192.168.220.183	3309	db2531	0.0	idle	4000	0,0,0	2019-12-14 18:51:32	false	
47	207	master's slave	192.168.220.184	3309	db2531	0.0	idle	4000	0,0,0	2019-12-14 18:51:32	false	
47	196	master	192.168.220.181	3309	db2531	0.0	idle	4000	0,0,0	2019-12-14 18:51:32	false	
47	197	master's slave	192.168.220.182	3309	db2531	0.0	idle	4000	0,0,0	2019-12-14 18:51:32	false	
45	203	master's slave	192.168.220.184	3307	db2531	0.0	idle	4000	0,0,0	2019-12-14 18:51:32	false	
45	202	master	192.168.220.183	3307	db2531	0.0	idle	4000	0,0,0	2019-12-14 18:51:32	false	
45	198	master's slave	192.168.220.182	3307	db2531	0.0	idle	4000	0,0,0	2019-12-14 18:51:32	false	
45	188	master	192.168.220.181	3307	db2531	0.0	idle	4000	0,0,0	2019-12-14 18:51:32	false	
-1	-2	master's slave	192.168.220.184	3306	hotdb_config	0.0	idle	4000	0,0,0	2019-12-14 18:51:32	false	
-1	-1	master	192.168.220.183	3306	hotdb_config	0.0	idle	4000	0,0,0	2019-12-14 18:51:32	false	
-1	-5	master's slave	192.168.220.182	3306	hotdb_config	0.0	idle	4000	0,0,0	2019-12-14 18:51:32	false	
-1	-4	master	192.168.220.181	3306	hotdb_config	0.0	idle	4000	0,0,0	2019-12-14 18:51:32	false	

@@configurl						
id	url	username	password	is_current	is_available	idc_type
2	jdbc:mysql://192.168.220.184:3306/hotdb_config	hotdb_config	hotdb_config	FALSE	TRUE	Biz-IDC
1	jdbc:mysql://192.168.220.183:3306/hotdb_config	hotdb_config	hotdb_config	TRUE	TRUE	Biz-IDC
5	jdbc:mysql://192.168.220.182:3306/hotdb_config	hotdb_config	hotdb_config	FALSE	TRUE	DR-IDC
4	jdbc:mysql://192.168.220.181:3306/hotdb_config	hotdb_config	hotdb_config	TRUE	TRUE	DR-IDC

@@datasource													
dn	ds	name	type	status	host	port	schema	active	idle	size	unavailable_reason	flow_control	idc_id
48	208	192.168.220.183-3310_db2531	1	1	192.168.220.183	3310	db2531	2	30	32	NULL	0/64	1
48	209	192.168.220.182-3310_db2531	2	1	192.168.220.182	3310	db2531	1	31	32	NULL	0/64	1
48	201	192.168.220.182-3310_db2531	1	1	192.168.220.182	3310	db2531	1	32	33	NULL	0/64	2
48	200	192.168.220.181-3310_db2531	1	1	192.168.220.181	3310	db2531	1	31	32	NULL	0/64	2
46	205	192.168.220.184-3308_db2531	2	1	192.168.220.184	3308	db2531	1	31	32	NULL	0/64	1
46	204	192.168.220.183-3308_db2531	1	1	192.168.220.183	3308	db2531	2	32	34	NULL	0/64	1
46	192	192.168.220.181-3308_db2531	1	1	192.168.220.181	3308	db2531	1	31	32	NULL	0/64	2
46	193	192.168.220.182-3308_db2531	2	1	192.168.220.182	3308	db2531	1	31	32	NULL	0/64	1
47	206	192.168.220.183-3309_db2531	1	1	192.168.220.183	3309	db2531	2	30	32	NULL	0/64	1
47	205	192.168.220.182-3309_db2531	2	1	192.168.220.182	3309	db2531	1	31	32	NULL	0/64	1
47	197	192.168.220.182-3309_db2531	2	1	192.168.220.182	3309	db2531	1	31	32	NULL	0/64	2
47	196	192.168.220.181-3309_db2531	1	1	192.168.220.181	3309	db2531	1	31	32	NULL	0/64	2
45	203	192.168.220.184-3307_db2531	2	1	192.168.220.184	3307	db2531	1	31	32	NULL	0/64	1
45	202	192.168.220.183-3307_db2531	1	1	192.168.220.183	3307	db2531	2	30	32	NULL	0/64	1
45	189	192.168.220.181-3307_db2531	2	1	192.168.220.182	3307	db2531	1	31	32	NULL	0/64	2
45	188	192.168.220.181-3307_db2531	1	1	192.168.220.181	3307	db2531	1	31	32	NULL	0/64	2
-1	-2	bakconfigdatasource	2	1	192.168.220.184	3306	hotdb_config	1	8	9	NULL	N/A	1
-1	-1	configdatasource	1	1	192.168.220.183	3306	hotdb_config	2	9	11	NULL	N/A	1
-1	-5	configdatasource	2	1	192.168.220.182	3306	hotdb_config	1	9	10	NULL	N/A	2
-1	-4	drconfigdatasource	1	1	192.168.220.181	3306	hotdb_config	1	7	8	NULL	N/A	2

### 6.4.2.5. Special Instructions

- When a switchover at IDC level occurs, if the target IDC has slave data sources or standby master data sources, log file hotdb.log will output the replication position of the master data source and slave data source that are about to setup a replication relation. By copying the position, You will know the GTID execution status of the data sources after IDC switchover:

Example 1: Log output of replication position of ConfigDB

```
2019-12-14 19:07:58.317 [INFO] [INNER] [$I-NIOREACTOR-3-RW]
cn.hotpu.hotdb.mysql.nio.handler.ChangeIDCDatasourceReplByConfig(351) - master([id:-4, nodeId:-1 192.168.220.183:3306/hotdb_config status:1, charset:utf8],
Executed_Gtid_Set:46906d49-0b87-11ea-91a0-525400e2c4b6:1-2015, f629710f-0c19-11ea-bff2-525400edf2d2:1-3524, fc61739a-0c19-11ea-b887-5254007e6b1d:1-2) change master to master_standby([id:-5, nodeId:-1 192.168.220.184:3306/hotdb_config status:1, charset:utf8],
Executed_Gtid_Set:46906d49-0b87-11ea-91a0-525400e2c4b6:1-2015, f629710f-0c19-11ea-bff2-525400edf2d2:1-3524, fc61739a-0c19-11ea-b887-5254007e6b1d:1-2).
```

Example 2: Log output of replication position of data source

```
2019-12-14 19:08:00.546 [INFO] [INNER] [$NIOREACTOR-2-RW]
cn.hotpu.hotdb.mysql.nio.handler.ChangeIDCDatasourceReplByConfig(351) - master([id:192, nodeId:46 192.168.220.183:3308/db2531 status:1, charset: utf8mb4],
Executed_Gtid_Set:4a12db23-0c1a-11ea-a751-525400edf2d2:1-3524, a53b2400-0b87-11ea-b97e-525400e2c4b6:1-3) change master to master_standby([id:193, nodeId:46 192.168.220.184:3308/db2531 status:1, charset: utf8mb4], Executed_Gtid_Set:4a12db23-0c1a-11ea-a751-525400edf2d2:1-3524, a53b2400-0b87-11ea-b97e-525400e2c4b6:1-3).
```

Example 3: Slave data source has more GTID than the master

```
2019-12-14 19:08:00.546 [WARN] [INNER] [$NIOREACTOR-2-RW]
cn.hotpu.hotdb.mysql.nio.handler.ChangeIDCDatasourceReplByConfig(329) -
Slave([id:189, nodeId:45 192.168.220.184:3307/db2531 status:1, charset: utf8mb4])'s
Executed_Gtid_Set(1e81d24b-0c1a-11ea-b34f-525400edf2d2:1-3518, 70ee41d5-0b87-11ea-85cd-
```

```
525400e2c4b6:1, 7afdc35d-0b87-11ea-91a3-52540073fb81:1) is greater than
master([id:188, nodeId:45 192.168.220.183:3307/db2531 status:1, charset:utf8mb4])'s
Executed_Gtid_Set(1e81d24b-0c1a-11ea-b34f-525400edf2d2:1-3518, 70ee41d5-0b87-11ea-85cd-
525400e2c4b6:1).
```

#### Example 4: Semi-synchronous replication status monitoring

```
2019-12-14 19:07:57.820 [WARN] [INIT] [Labor-26]
cn.hotpu.hotdb.backend.BackendDatasource(806) - [id:-5, nodeId:-1
192.168.220.184:3306/hotdb_config status:1, charset:utf8], Rpl_semi_sync_master_status or
Rpl_semi_sync_slave_status must be 'ON' in DR mode
```

- All descriptions in this Maintenance Management chapter are explained with GTID and semi-synchronous replication enabled. If you consider semi-synchronization replication of data sources and ConfigDB in a single IDC and asynchronous replication between this two IDC, we recommend the following configuration:
  1. It is recommended that you set `rpl_semi_sync_master_wait_no_slave = 0` and `rpl_semi_sync_master_timeout` should not be set too long for instances (including data sources and ConfigDB) in the master center. Also set `rpl_semi_sync_slave_enabled` as disable for the master instance in the DR center.
  2. After a switchover at IDC level occurs, when you manually check the semi-synchronous replication parameters, you need to execute the stop slave; start slave command once to make it take effect.
  3. After failover of the master instances in the DR center, you need to manually disable `rpl_semi_sync_slave_enabled` for the current active master and enable `rpl_semi_sync_slave_enabled` for the original master , and execute the stop slave; start slave command once on both instances to make it effective.
- Or, if semi-synchronous replication is not considered in the DR center at all, you can directly disable `rpl_semi_sync_slave_enabled` for all the instances in DR center so that there is no need to maintain the semi-synchronization status in the DR center. If semi-synchronous replication is not enabled in the DR replication relationship, you must strictly pay attention that the RPO cannot be set to 0.



# 7. Note

When the DR mode is turned on, it is essential to monitor the service status for the entire cluster with two IDC. In this process, you must pay attention to turning on some necessary periodic plans and monitoring functions to avoid risks in advance.

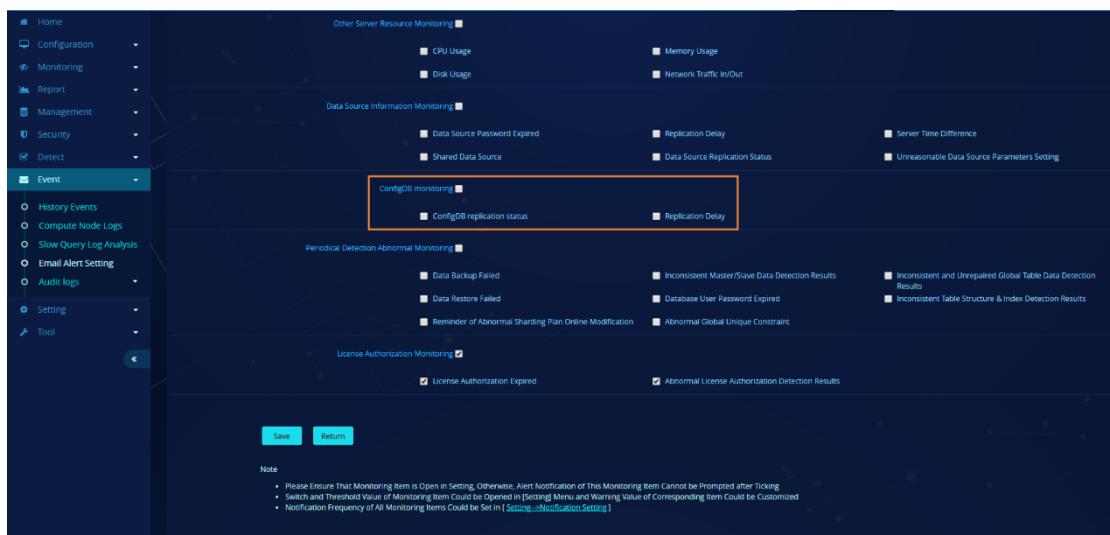
## 7.1.1. Periodic Plans

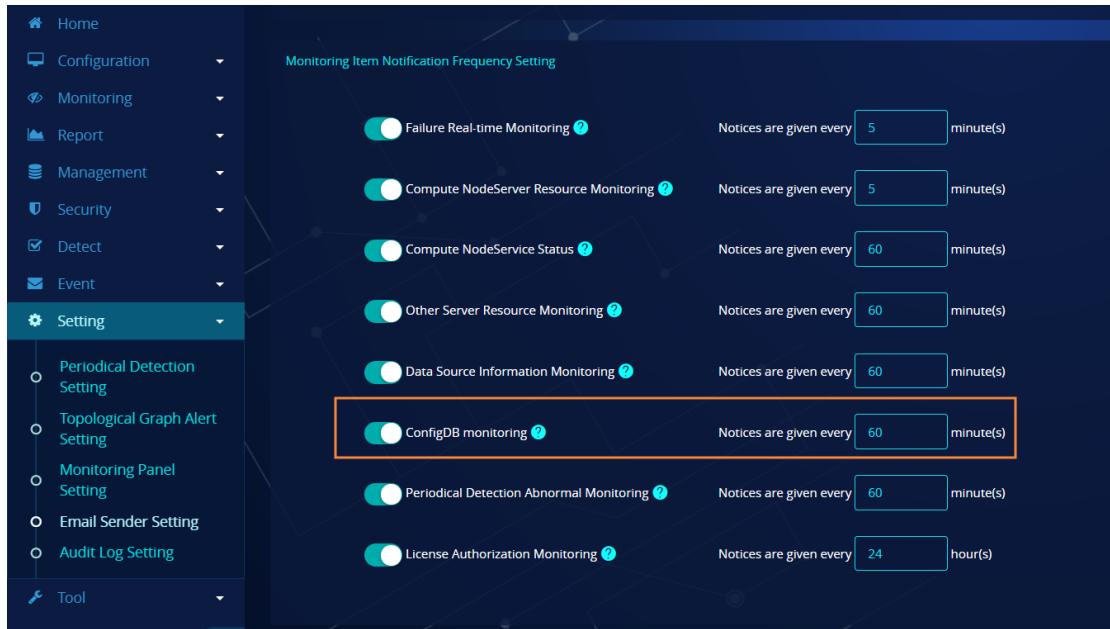
It is recommended to enable replication status [monitoring](#) of ConfigDBs and data sources;

It is recommended to set a periodic plan for [master/slave consistency detection](#) to check the data consistency of master center and DR center.

## 7.1.2. Email Alert

It is recommended that you set an email alert including monitoring the service status of all compute nodes and data sources in order to be aware of error information of each core component in time. And newly-added item, replication status of ConfigDB, can be set on the "Events"-> "Email Alert Setting"-> "Add Notification Strategy" page. For example, the following figure shows an example of setting up for monitoring replication status of ConfigDB:





The content of the email alert will explain the specific IDC type where the problem occurred:

#### Data Source related monitoring alert reminder

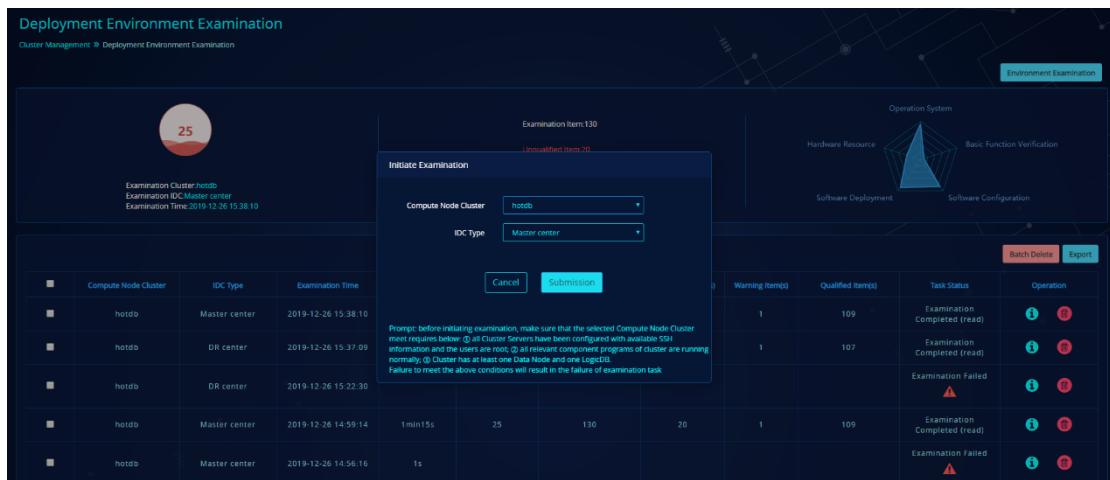
Email information:  
 Compute Node Cluster name: Group\_7843  
 DR mode: Open  
 Current master center: DR center  
 Master centerCompute Node information: 计算节点-01(192.168.220.221/3325)  
 DR centerCompute Node information: 计算节点-01(192.168.220.223/3325)

##### **[Data Source Replication Status]**

1. dr center The master-slave replication of data source under data node: dn\_01 (id: 1) is abnormal, cause of anomaly: Replication anomaly information: 191212 10:08:45 Error 'Table 'test4' already exists' on query. Default database: 'db01'. Query: 'create table test4(id int)'

### 7.1.3. Environment Examination

After cluster deployment is completed without any errors, it is recommended to use the manager account of management platform to enter the "Cluster Management" -> "Deployment Environment Examination" page to perform environment examinations for both master center and DR center, and follow the suggestions to optimize unqualified configuration.



## 7.1.4. Data Backup

Currently data backup only supports backup for the master center, but you can transmit the backup files to the remote server. However, during the transmission process, it is necessary to pay attention to the consumption of network bandwidth by large backup files.

## 7.1.5. Service Restarted

When the services of failed data sources or ConfigDB are started, you must make sure that all replication relation abnormalities have been repaired.

## 7.1.6. SSH of Servers

It is recommended to configure the server SSH information on the "Configuration"->"Server" page when you start to use it, so as to assist the management platform to complete some simple maintenance and monitoring operations.

Server						
Configuration > Server						
		Hostname (multiple choices)		SSH Connection Status (m...)		
IDCType	Hostname	Service Program Association	SSH Connection Status	Username	Operation	
DR center	192.168.240.122	Compute Node HotDB-04(192.168.240.122/3325) ConfigDB 192.168.240.122.3306/notdb_config Data Source 192.168.240.122.3310/db01 192.168.240.122.3311/db01 Backup Program backup Compute Node HotDB-03(192.168.240.121/3325) ConfigDB 192.168.240.121.3306/notdb_config Data Source 192.168.240.121.3310/db01 192.168.240.121.3311/db01 Backup Program backup	Normal	root	<a href="#">Configuration SSH</a> <a href="#">Stop Monitoring</a> <a href="#">Remove</a>	
DR center	192.168.240.121		Normal	root	<a href="#">Configuration SSH</a> <a href="#">Stop Monitoring</a> <a href="#">Remove</a>	

### 7.1.7. Master/Slave Data Consistency Detection

Whenever there is an inconsistent detection record of master and slave instances in the master center or the DR center, you should resolve as soon as possible to ensure data consistency and make data consistency detection regularly for both master center and DR center. When a failover at IDC level occurs, the historical data before the time of the last master/slave data consistency detection can be guaranteed to be correct.

### 7.1.8. Service Access

It is not allowed to open the service port in the DR center through the `online_dr` command when the master center is running, unless it is confirmed to switch IDC;

The DR center must not use the same VIP as the master center to avoid the risk of applications directly accessing across IDCs;

It is recommended to stop `keepalived`, the master and slave high availability components of the DR center during the daily operation and maintenance to avoid unnecessary recovery operations caused by VIP drift in the DR center.

### 7.1.9. Setup Replication

When manually setting up a replication relation, you must enable `auto_position` mode of GTID and cannot set any Channel names;

Multi-source replication is not allowed. There is a one-to-one match between the data nodes and instances in this two IDCs. Otherwise, multi-source replication will occur.

Before doing any failovers, please check whether the master-slave replication relation is normal.

### 7.1.10. Important but temporarily unsupported functions

Data sources currently do not support MGR replication mode.

When semi-synchronous replication is not enabled or degrades on the data sources and ConfigDBs, it is not guaranteed that the RPO will be 0 after switchover at the IDC level occurs.

## 8. Reference Instructions

After the DR mode is enabled, you must pay attention to some abnormal log alerts and follow the correct way to setup replication relation during actual use and maintenance. This section mainly describes this type of reference instructions.

### 8.1. Compute node logs

- Degraded semi-synchronous replication status of the slave data source (possibly cause that RPO is not 0):

```
2019-12-10 12:17:00.630 [WARN] [TIMER] [$NIOExecutor-5-1]
cn.hotpu.hotdb.manager.handler.LoggerHandler(25) - DR_IDC's datasource: -5([id:-5, nodeId:-1
192.168.220.184:3306/hotdb_config status:1, charset:utf8])'s RPL_SEMI_SYNC_SLAVE_STATUS is
OFF
```

- The number of acks returned by the slave that is set for the semi-synchronous replication of slave data source is inconsistent with the actual number (possibly cause that RPO is not 0):

```
root@localhost:(none) 5.7.25-log 06:03:22> show slave hosts;
+-----+-----+-----+-----+
| Server_id | Host | Port | Master_id | Slave_UUID
+-----+-----+-----+-----+
| 2201823307 |      | 3307 | 2201813307 | 7afdc35d-0b87-11ea-91a3-52540073fb81
| 2201833307 |      | 3307 | 2201813307 | 1e81d24b-0c1a-11ea-b34f-525400edf2d2
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

root@localhost:(none) 5.7.25-log 06:03:27> show variables like '%semi%';
+-----+-----+
| Variable_name | value
+-----+-----+
| rpl_semi_sync_master_enabled | ON
| rpl_semi_sync_master_timeout | 10000
| rpl_semi_sync_master_trace_level | 32
| rpl_semi_sync_master_wait_for_slave_count | 1
| rpl_semi_sync_master_wait_no_slave | ON
| rpl_semi_sync_master_wait_point | AFTER_SYNC
| rpl_semi_sync_slave_enabled | ON
| rpl_semi_sync_slave_trace_level | 32
+-----+-----+
8 rows in set (0.01 sec)
```

```
root@localhost:(none) 5.7.25-log 06:03:37> show status like '%semi%';
+-----+-----+
| variable_name | value |
+-----+-----+
| Rpl_semi_sync_master_clients | 2 |
| Rpl_semi_sync_master_net_avg_wait_time | 0 |
| Rpl_semi_sync_master_net_wait_time | 0 |
| Rpl_semi_sync_master_net_waits | 2019 |
| Rpl_semi_sync_master_no_times | 1 |
| Rpl_semi_sync_master_no_tx | 4140 |
| Rpl_semi_sync_master_status | ON |
| Rpl_semi_sync_master_timefunc_failures | 0 |
| Rpl_semi_sync_master_tx_avg_wait_time | 21187 |
| Rpl_semi_sync_master_tx_wait_time | 23221226 |
| Rpl_semi_sync_master_tx_waits | 1096 |
| Rpl_semi_sync_master_wait_pos_backtraverse | 0 |
| Rpl_semi_sync_master_wait_sessions | 0 |
| Rpl_semi_sync_master_yes_tx | 1203 |
| Rpl_semi_sync_slave_status | ON |
+-----+-----+
15 rows in set (0.00 sec)
```

```
2019-12-10 12:16:00.634 [WARN] [TIMER] [$NIOExecutor-5-1]
cn.hotpu.hotdb.manager.handler.LoggerHandler(25) - datasource: 169([id:169, nodeId:43
192.168.220.181:3307/db2531 status:1, charset:utf8mb4])'s
RPL_SEMI_SYNC_MASTER_WAIT_FOR_SLAVE_COUNT=2, real slave count=1
```

- The actual replication relation between the data source or ConfigDB is inconsistent with the configuration in ConfigDB.

```
2019-12-14 19:08:00.426 [WARN] [INNER] [$NIOREACTOR-3-RW]
cn.hotpu.hotdb.mysql.nio.handler.InitSlaveStatusHandler(99) - Datasource: 188's master
datasource: 192.168.220.183:3307 not in datanode: 45
```

- When IDC is switched or the compute node that is providing services is offline, the current back-end connection related to it will also be closed. For related log information, please refer to:

```
2019-12-14 19:08:00.978 [WARN] [CONNECTION] [$NIOREACTOR-2-RW]
cn.hotpu.hotdb.net.NIOSocketWR(181) - exception(stream closed) in reading backend
connection: [id:196, nodeId:47 192.168.220.181:3309/db2531 status:1, charset:utf8mb4] id=931

threadId=5801 isAuthenticated=true responseHandler=null isNetFull=false
bufferInFullNet=false lastUsedTime=1576321520613 tookTime=1576321520605 sendingData=false
inFlowControl=false maybeSlow=false heavyRowEof=false inFieldPacket=false
```

```

needSyncAutocommit=false syncAutocommitValue=true usingReadBuffer=false
usingWriteBuffer=false readBuffer=java.nio.DirectByteBuffer[pos=0 lim=16384 cap=16384]
writeBuffer=null writeQueue=0 host=192.168.220.181 port=3309 localPort=22248

connectionVars=[dsCharset=utf8mb4 autocommit=true savepointChecked=false txIsolation=2
charsetIndex=45 characterSetClient=utf8mb4 characterSetResults=utf8mb4
characterSetConnection=utf8mb4 collationConnection=utf8mb4_general_ci foreignK
eyChecks=true uniqueChecks=true txReadOnly=false]

connectionStat=[startupTime=1576320655994 lastReadTime=1576321520594
lastWriteTime=1576321520594 netInBytes=5533 netOutBytes=1477] state=idle pauseTime=0
usedMemory=16384 lastSQL=[null]

readData=1576321520613 lastWritten=184 inner=false toBeClosed=false isClosed=false
closeReason=null MySQLVersion=5.7.25

```

➤ Log records when service port is closed:

```

2019-12-14 19:07:56.610 [INFO] [MANAGER] [$NIOExecutor-0-2]
cn.hotpu.hotdb.manager.response.Offline(66) - received offline command
from:[thread=$NIOExecutor-0-
2, id=1056, user=root, host=192.168.220.181, port=3325, localport=16928, schema=nul
1]

2019-12-14 19:07:56.612 [INFO] [MANAGER] [Labor-10] cn.hotpu.hotdb.HotdbServer(2149) -
MANAGER offline start

```

➤ Log records when service in the DR center is started:

```

2019-12-14 19:07:56.552 [INFO] [MANAGER] [Labor-26] cn.hotpu.hotdb.HotdbServer(2115) -
DR online start

```

## 8.2. Recommended steps for replication setup

### 8.2.1. Prerequisites

- The version numbers of all MySQL instances about to setup replication are exactly the same;
- The result of executing show master status on the new slave instance does not include any GTID (GTID is disabled or GTID is empty). Also, the results of executing show slave status and show slave hosts are both empty;
- If the GTID function is enabled on the old master instance, each set of GTID value in the Executed\_Gtid\_Set by executing show master status must be continuous without any breakpoints. (That is, any GTID must start from 1 and reach the maximum. There must be no gap in the middle of any GTID.) Note: For the old master instance with parallel replication turned on, it tends to see that GTID is not continuous. In this case, it is recommended to manually decide whether to continue to setup replication;
- For those MySQL instances are deployed by single component deployment, the ConfigDB instances can only be setup replication with the ConfigDB instances, and the data source instances can only be setup replication with the data source instances. Do not mix instances;
- It should be setup in the business period with low flow. If the amount of data is large, sufficient time must be reserved for execution;
- If the MySQL instance is installed by single component deployment, it is recommended to use the dbbackup user or hotdb\_datasource user to export and import data and use the repl user when setting up replication.

### 8.2.2. Set up replication relations when GTID is enabled

If the GTID function on both the new instances and the old master instances about to set up replication is enabled, that is, gtid\_mode is on, the following method can be used to set up the master-slave replication:

Step 1: Check whether the remaining free disk space of the directory used for exporting data is sufficient. It must be ensured that the remaining space of the directory used by the data is greater than 1.5 times + 10GB of space required for the current query. For the size of the current data, refer to SQL statement: select sum (data\_length) from information\_schema.tables (unit: byte b).

Step 2: Use the following parameters to export the old master instance data:

```
mysqldump --no-defaults --all-databases --default-character-set=utf8mb4 --single-transaction --set-gtid-purged --events --routines --triggers --hex-blob --no-tablespaces --host=xxx --port=xxx --user=xxx -pxxx > file name ;echo $?
```

```
Example: mysqldump --no-defaults --all-databases --default-character-set=utf8mb4 --single-transaction --set-gtid-purged --events --routines --triggers --hex-blob --no-tablespaces --host=127.0.0.1 --port=3306 --user=dbbackup -pdbs backup > backup_data.sql ;echo $?
```

```
mysqldump: [Warning] Using a password on the command line interface can be insecure.
```

```
0
```

If the return value is 0 and the export command does not report an error, it means the data is exported successfully. (This is normal: mysqldump: [Warning] Using a password on the command line interface can be insecure.)

Note: By adopting the above method to export data, SET @@SESSION.SQL\_LOG\_BIN = 0; is added by default. Therefore, you must make sure that you do not setup DR replication first and then import data to the master in DR center when setting up the replication relation between master instances in the master center and DR center, and the replication relation in the DR center itself. (Wrongly believing that data will synchronize to the slave in DR center). You have to set up replication relations after importing data correctly into each instance.

Step 3: After the file is exported, check if the file content is available. Transmit the file through SCP command to the server where the new instance to be the slave stores. Before execute SCP command, check the data directory of the new instance to see if the remaining disk space was sufficient. It is required the remaining disk space of the

target data directory to be greater than 2.5 times the size of the exported file + 10GB.  
The command to check the new instance data directory can refer to:

```
show global variables like 'datadir'
```

Step 4: After the file transfer is successful, you can choose to delete the exported file on the original master instance as required.

Step 5: Use the following command to import the file:

```
mysql --no-defaults --default-character-set=utf8mb4 --binary-mode --disable-reconnect --host=xxx --port=xxx --user=xxx -pxxx < file name ;echo $?
```

```
Example: [root@hotdb-220-182 ~]# mysql --no-defaults --default-character-set=utf8mb4 --binary-mode --disable-reconnect --host=127.0.0.1 --port=3306 --user=dbbackup -pdbname < /data/mysql/mysqldata3306/backup_data.sql ;echo $?
```

```
mysql: [Warning] Using a password on the command line interface can be insecure.
```

```
0
```

If the return value is 0 and the export command does not report an error, it means that data is imported successfully. ( This is normal: mysqldump: [Warning] Using a password on the command line interface can be insecure.)

Step 6: Run the following command to refresh privileges. Note that after the refresh, if the instance is reconnected, the username and password of the connection are the same as the old master instance.

```
FLUSH NO_WRITE_TO_BINLOG PRIVILEGES;
```

Step 7: Consider executing the following statements for all tables:

```
ANALYZE NO_WRITE_TO_BINLOG TABLE xxx;
```

Step 8: Delete the original data backup file under data directory of the new instance as required.

Step 9: Set up a replication relationship using GTID auto-location.

```
change master to master_host = 'xxx',master_user='xxx',master_password='xxx',  
master_port=xxx,master_auto_position=1;
```

```
Example: change master to master_host =
'192.168.220.181',master_user='repl',master_password='repl',
master_port=3306, master_auto_position=1;
```

### 8.2.3. Set up replication relations when GTID is disabled

Note: If the DR mode is enabled for the compute node cluster, all data sources are required to have GTID turned on. The introduction here is only for reference. There is no direct correlation in the actual maintenance of the DR mode.

If the GTID function of the new instance and the old master instance for the replication relation to be set up is turned off or one of them is turned off, the replication relation can only be set up by using binlog position. The main process is the same as the process of GTID enabled, but the data import, export, and replication setup involved in steps 2, 5, and 9 are different:

Step 2: export data command refers to:

```
mysqldump -no-defaults -all-databases -default-character-set=utf8mb4 -single-
transaction -loose-set-gtid-purged=off -master-data=2 -events -routines -triggers
-hex-blob -no-tablespaces -host=xxx -port=xxx -user=xxx -pxxxx > file name ;echo $?
```

Step 5: import data command refers to:

```
mysql -no-defaults -default-character-set=utf8mb4 -binary-mode -disable-reconnect
-host=xxx -port=xxx -user=xxx -pxxxx < file name ;echo $?
```

Step 9: set up replication refers to:

```
change master to master_host = 'xxx',master_user='xxx',master_password='xxx',
MASTER_LOG_FILE='mysql-bin.xxx', MASTER_LOG_POS=xxx;
```

Note: *MASTER\_LOG\_FILE* = 'mysql-bin.xxx', *MASTER\_LOG\_POS* = xxx; This position is the position obtained when the data was exported in step 2, which can be obtained by: *head -c 4096 file without duplicate name | grep -I 'change master'*. If there are multiple new instances that need to set up a master-slave replication and there is a multipe

levels of slaves relation, you need to execute `show master status` on the corresponding master to obtain the binlog position after the data is imported and before performing any write operations. You cannot directly take the export position to set up a replication relation.

## 8.3. Under what circumstances data in the original IDC must be abandoned

If a failover at the IDC level occurs, you need to pay attention that data in the original master center must be discarded in the following cases:

When it is decided manually that some data is not synchronized to the original DR center (the current active IDC), since the data is based on the original DR center (the current active IDC), the data has been synchronized cannot be synchronized any more.

If you need to rebuild the original master center, you need to delete the data from the original master center and export the data from the current active IDC, and then perform subsequent IDC recovery operations, otherwise it will cause data confusion (note that the use of semi-synchronous replication may also cause data confusion, but such data is a transaction that has issued a commit but has not received ok. we cannot simply consider such a transaction as a transaction have been submitted or must have been submitted. It must be decided on the processing method by the data being actually queried).