

# **Distributed Transactional Database**

## **HotDB Server -**

### **[Manual Update]**

## **Function Manual**

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@hotmail.com](mailto:service_hotdb@hotmail.com)

Service Tel: 021-5218 0789

# Contents

<b>1. Introduction</b>	<b>1</b>
<b>2. Management platform Manual Upgrade</b>	<b>2</b>
2.1. ConfigDB Upgrade	2
2.1.1. Stop Management Platform Service	2
2.1.2. Back Up ConfigDB Data	2
2.1.3. Execute ConfigDB Upgrade Script.	2
2.1.4. Handling of ConfigDB Upgrade Abnormality	3
2.2. Replace Program Package	4
2.2.1. Back Up Management Platform Directory	4
2.2.2. Upload and Unzip New Version Package	4
2.3. Update Configuration File	5
2.3.1. Modify application.properties Configuration File	5
2.4. Upgrade JDK version.	5
2.4.1. Upload and unzip OpenJDK8 installation package	6
2.4.2. Modify startup script.	6
2.5. Start Management Platform and Start Exception Handling.	7
2.5.1. Start Management Platform Services	7
2.5.2. Start Exception Handling	7
<b>3. Compute node manual upgrade</b>	<b>8</b>
3.1. Notes	8
3.1.1. Upgrade JDK version	8
3.1.2. Upgrade new license.	8
3.2. Single Node Cluster Mode Upgrade	9
3.2.1. Stop Compute Node Service.	9
3.2.2. ConfigDB Upgrade	9
3.2.3. Replace program package.	9
Back up compute node directory	9
Upload and Unzip New Version Package	10
NDB SQL Service.	10
3.2.4. Update Configuration File	11
server.xml Configuration File.	11

Compute node start script . . . . .	13
Other Configuration Files . . . . .	14
3.2.5. Start compute node and enable exception handling . . . . .	14
Start compute node service . . . . .	14
Enable exception handling . . . . .	14
3.3. Master/slave node cluster upgrade with service stopped . . . . .	15
3.3.1. ConfigDB Upgrade . . . . .	15
3.3.2. Stop compute node and keepalived service . . . . .	15
3.3.3. Upgrade master/slave compute node version . . . . .	16
3.3.4. NDB SQL Service . . . . .	16
3.3.5. Start keepalived and compute node service . . . . .	16
3.4. Master/slave node cluster upgrade without service stopped . . . . .	17
3.4.1. Upgrade conditions without service stopped . . . . .	17
3.4.2. High Availability Switch Check . . . . .	18
High Availability Switch Check Items . . . . .	18
3.4.3. ConfigDB Upgrade . . . . .	23
3.4.4. Slave Compute Node Upgrade . . . . .	23
Stop Slave Compute Node Service . . . . .	23
Back Up Slave Compute Node Directory . . . . .	23
Upload and Unzip New Version Package . . . . .	23
NDB SQL Service . . . . .	24
server.xml Configuration File . . . . .	24
Compute Node Start Script . . . . .	24
Other Configuration Files . . . . .	25
Start Slave Compute Node Service . . . . .	25
3.4.5. Master Compute Node Upgrade . . . . .	25
Stop Master Compute Node Service . . . . .	25
Check Whether High Availability Switch is Successful . . . . .	26
Back Up Master Compute Node Directory . . . . .	26
Upload and Unzip New Version Package . . . . .	26
NDB SQL Service . . . . .	26
server.xml Configuration File . . . . .	27
Compute Node Startup Script . . . . .	27
Other Configuration Files . . . . .	27
Manually Execute High Availability Environment Reconstruction . . . . .	28
Start current slave (without VIP) compute node service and keepalived . . . . .	33

Special instructions . . . . .	33
3.5. Multi-node cluster upgrade with service stopped . . . . .	33
Stop compute node service . . . . .	34
Upgrade ConfigDB . . . . .	34
Replace upgrade package and upgrade configuration file . . . . .	34
Start compute node service . . . . .	34
3.6. Multi-node cluster upgrade without service stopped . . . . .	35
3.6.1. Upgrade Conditions Without Service Stopped . . . . .	35
3.6.2. ConfigDB Upgrade . . . . .	36
3.6.3. Secondary1 Compute Node Upgrade . . . . .	36
3.6.4. Secondary2 Compute Node Upgrade . . . . .	36
3.6.5. Primary Compute Node Upgrade . . . . .	36
Stop Primary Compute Node Service . . . . .	36
3.7. Cluster upgrade in disaster recovery mode . . . . .	37
3.7.1. Master center upgrade . . . . .	37
3.7.2. DR center upgrade . . . . .	37
<b>4. HotDB Backup Manual Upgrade . . . . .</b>	<b>39</b>
4.1. Stop HotDB Backup Service . . . . .	39
4.2. Back Up Old HotDB Backup Directory . . . . .	39
4.3. Upload and Unzip New Version Package . . . . .	39
4.4. Start Service . . . . .	40
<b>5. HotDB Listener Manual Upgrade . . . . .</b>	<b>41</b>
5.1. Stop Listener Service . . . . .	41
5.2. Back Up Old HotDB Backup Directory . . . . .	41
5.3. Upload and Unzip New Version Package . . . . .	41
5.4. Modify Configuration File . . . . .	42
5.5. Start Service . . . . .	42

# 1. Introduction

To better help users to upgrade the version of management platform and compute node, the Distributed Transaction Database HotDB Server [Manual Upgrade] Function Manual is hereby edited. The manual aims to help users to manually and independently complete the basic upgrade of the management platform and compute node through the operation instructions in this document, and provides suggestions for handling the abnormality of version upgrade.

This manual is suitable for upgrade of any version of management platform or compute node. When part of version upgrade requires special handling, corresponding instructions will be given in the document, which should be paid attention to when upgrading. Finally, it is suggested that you upgrade the management platform or compute node in strict accordance with the operation instructions in the manual.

## 2. Management platform Manual Upgrade

Currently, management platform can only be manually upgraded to version below V2.5.3, and it can be semi-automatically upgraded to V2.5.3 and above (Manually replace package and update configuration files, and ConfigDB is automatically upgraded by program). When management platform upgrades to version V2.5.6 and above, additional JDK upgrade is required.

### 2.1. ConfigDB Upgrade

No ConfigDB SQL changes when some versions are upgraded (For example: update of versions with different dates in the same version number), and in such situations you do not need to pay attention to ConfigDB upgrade module.

#### 2.1.1. Stop Management Platform Service

Log in to the management platform server and execute the command Stop Service:

```
#sh /usr/local/hotdb/hotdb-management/bin/hotdb_management stop
```

#### 2.1.2. Back Up ConfigDB Data

ConfigDB data must be backed up before ConfigDB is upgraded to prevent abnormal situation in upgrading process.

**mysqldump is used to back up configDB data**

```
#mysqldump -S /data/mysql/mysqldata3316/sock/mysql.sock --set-gtid-purged=off --single-transaction  
--master-data=1 --databases hotdb_cloud_config --default-character-set=utf8 -uroot >  
/usr/local/hotdb/hotdb_cloud_config_2.5.3_20190815.sql
```

#### 2.1.3. Execute ConfigDB Upgrade Script

Special instruction: if the target version of the management platform upgrade is V2.5.3 or above, ConfigDB can be upgraded by program after the management

platform replaces the new version program package and update the configuration file. Management platform ConfigDB can be semi-automatically upgraded by reference of Upgrade Center -> Management Platform ConfigDB Upgrade Function Specification in [Distributed Transaction Database HotDB Server-V2.5.x \[Management Platform\] Function Manual](#).

### Execute ConfigDB Upgrade Script

1. Log in to the management platform ConfigDB

```
#mysql -uhotdb_cloud -p -P3306 -Dhotdb_cloud_config -h127.0.0.1
```

2. Copy the upgrade content to ConfigDB for execution

Execute Change SQL Statement according to the specific upgrade script content

Special instructions:

- ConfigDB upgrade script can be obtained from product supplier.
- Note to distinguish SQL for management platform ConfigDB upgrade and SQL for compute node ConfigDB upgrade when copying upgrade script content.
- If the management platform is upgraded across versions (for example, 2.4.9->2.5.3), the ConfigDB script needs to be executed successively in version order, and disrupting the order or leaving out the upgraded content of any version is not allowed.
- All upgrade SQL statements must be executed successfully without errors when upgrading ConfigDB.
- If the upgrade is executed by copying and pasting the content into ssh terminal, please note that the character set of terminal is utf8.

## 2.1.4. Handling of ConfigDB Upgrade Abnormality

### Problems and Handling Methods

1. Statement error in ConfigDB upgrade SQL execution



- Save the SQL error message of execution, and provide ConfigDB upgrade related information including but not limited to: current version of ConfigDB and its upgrade target version, upgrade SQL statement of execution, ConfigDB backup file, etc. for product supplier for analysis and solution.
- Restore ConfigDB with the file backed up in the previous step. Please stop management platform service program during the restoring

## 2. Exception for non-SQL statement error

- Analyze the causes and solutions of the error according to error message.
- Restore ConfigDB with the file backed up in the previous step. Please stop management platform service program during the restoring.
- Re-execute the upgrade after problem is repaired and ConfigDB is restored successfully.

## 2.2. Replace Program Package

### 2.2.1. Back Up Management Platform Directory

```
#cd /usr/local/hotdb/  
  
#mv hotdb-management hotdb_management_249
```

### 2.2.2. Upload and Unzip New Version Package

#### Upload new version package

rz command or ftp file transfer tool can be used to upload new version package

#### Unzip new version package

```
#tar -xvf hotdb-management-2.4.9-ga-20190417.tar.gz -C /usr/local/hotdb
```

#### Folder authority is given to hotdb users

```
#chown -R hotdb:hotdb /usr/local/hotdb/hotdb-management
```

#### Restore directory default context

```
#restorecon -R /usr/local/hotdb/hotdb-management
```

## 2.3. Update Configuration File

### 2.3.1. Modify application.properties Configuration File

It is suggested to open the application.properties configuration file under the old management platform directory, and synchronize the changed parameter values to the configuration file under the new version directory.

```
# cd /usr/local/hotdb/hotdb-management/conf/

# vi application.properties

-----Configuration Content-----

# http port

server.port=3324

# ssl socket port

server.backup.port=3322

# hotdb management database settings

spring.datasource.url=jdbc:mysql://192.168.210.30:3307/hotdb_cloud_config_253?useUnicode=true&characterEn
coding=UTF-8&autoReconnect=true&connectTimeout=3000&socketTimeout=3600000

spring.datasource.username=hotdb_cloud

spring.datasource.password=hotdb_cloud

spring.datasource.driver-class-name=com.mysql.jdbc.Driver

.....(some parameters are omitted here, subject to the actual application.properties)

# tomcat connection pool specific

settingshotdb.config.sqlFirewall.interceptType=0:\u8BEF\u64CD\u4F5C,1:SQL\u6CE8\u5165,2:\u4E0D\u826F\u
64CD\u4F5C,3:\u8BEF\u8BBE\u7F6Ehotdb.server.log.days=14
```

## 2.4. Upgrade JDK version

If the compute node is upgraded from the lower version to v2.5.5 and below, this step

can be skipped, and the default JDK version is 1.7;

If the compute node is upgraded from the lower version to v2.5.6 or above, the bin/hotdb\_management file needs to be adjusted to adapt to OpenJDK8, detailed operations are as follows:

### 2.4.1. Upload and unzip OpenJDK8 installation package

Upload OpenJDK8 installation package

You can upload OpenJDK8U-jdk\_x64\_linux\_hotspot\_8u252b09.tar.gz installation package using rz command or ftp file transfer tool. The installation package can be obtained by contacting HotDB

Unzip the installation package

```
#mkdir -p /usr/local/jdk8
```

```
#tar -xvf OpenJDK8U-jdk_x64_linux_hotspot_8u252b09.tar.gz -C /usr/local/jdk8
```

### 2.4.2. Modify startup script

Only two adjustments are required for bin/hotdb\_management:

-----Original configuration-----

```
JAVA_BIN=$shellPath/../../jdk/bin/java
```

```
JAVA_VERSION="1.7.0_80"
```

-----New configuration-----

```
JAVA_BIN=/usr/local/jdk8/jdk8u252-b09/bin/java
```

```
JAVA_VERSION="1.8.0_252"
```

## 2.5. Start Management Platform and Start Exception Handling

### 2.5.1. Start Management Platform Services

```
#cd /usr/local/hotdb/hotdb-management/bin/  
  
#sh hotdb_management start
```

### 2.5.2. Start Exception Handling

#### Problems and Handling Methods

1. Management platform log error, and service cannot be started
  - Rename the new version directory, and rename the old directory to otldb-management.
  - Restore the ConfigDB with ConfigDB file previously backed up
  - Restart management platform service in old version
  - Analyze the start failure, and re-upgrade it with the above steps after solving it.

## 3. Compute node manual upgrade

Compute node version can be upgraded manually or automatically. Please refer to upgrade center menu instruction of [Distributed Transaction Database HotDB Server-V2.5.x \[Management Platform\] Function Manual](#) for automatic upgrade. Manual upgrade of compute node versions of single node, master/slave node, multi-node mode cluster will be instructed below.

### 3.1. Notes

#### 3.1.1. Upgrade JDK version

If the compute node is upgraded from the lower version to v2.5.5 and below, this step can be skipped, and the default JDK version is 1.7;

If the compute node is upgraded from the lower version to v2.5.6 or above, JDK version needs to be upgraded to 1.8. For specific operations, please refer to the management platform [Upgrade JDK version](#). Besides, “JAVA\_BIN” and “JAVA\_VERSION” in bin/hotdb\_server need to be modified.

#### 3.1.2. Upgrade new license

If the compute node is upgraded from the lower version to v2.5.5 and below, this step can be skipped, and no change of license by default;

If the compute node is upgraded from the lower version to v2.5.6 or above, the license of compute node is upgraded to the official self-developed license. The license under the keys/ directory in the original package will not be used and be discarded directly; the 16-bit trial license under the keys/ directory in the new package will be stored for 90 days by default and will be activated automatically when the compute node is started. If it is found that **the number of existing nodes is greater than that of the default trial version**, you need to refer to the Distributed Transactional Database HotDB Server - [Service License] Function Manual (V2.5.6 and above) for machine code acquisition and license update before upgrading. Otherwise, the compute node service may not be enabled normally.

## 3.2. Single Node Cluster Mode Upgrade

### 3.2.1. Stop Compute Node Service

Log in to the compute node server to execute the command Stop Service:

```
#cd /usr/local/hotdb/hotdb-server/bin
```

```
#sh hotdb_server stop
```

### 3.2.2. ConfigDB Upgrade

Special instruction: ConfigDB data must be backed up before ConfigDB is upgraded to prevent abnormal situation in upgrading process. Master-master or MGR ConfigDB can only back up data of current master ConfigDB. No ConfigDB SQL changes when some versions are upgraded (For example: update of versions with different dates in the same version number), and in such situations you do not need to pay attention to ConfigDB upgrade module.

#### Upgrade operation:

The steps of Compute Node ConfigDB upgrade are the same as those of management platform ConfigDB upgrade. Please refer to the management platform [ConfigDB Upgrade Instructions](#) for operation instructions of ConfigDB backup, SQL upgrade, and upgrade abnormality handling.

### 3.2.3. Replace program package

#### Back up compute node directory

```
#cd /usr/local/hotdb/
```

```
#mv hotdb-server hotdb_server_253
```

## Upload and Unzip New Version Package

### Upload new version package

rz command or ftp file transfer tool can be used to upload new version package

### Unzip new version package

```
#tar -xvf hotdb-server-2.5.4-ga-20190812.tar.gz -C /usr/local/hotdb
```

### Folder authority is given to hotdb users

```
#chown -R hotdb:hotdb /usr/local/hotdb/hotdb-server
```

### Restore directory default context

```
#restorecon -R /usr/local/hotdb/hotdb-server
```

## NDB SQL Service

If compute node version is V2.5.3 and above before upgrade, please check whether compute node directory previously backed up contains NDB SQL service (directory starting with ndbsql). If it contains, please re-copy the NDB SQL directory in compute node directory previously backed up to the upgraded compute node directory.

### Enter the compute node directory previously backed up

```
#cd /usr/local/hotdb/hotdb_server_253
```

### Check the content under the compute node directory

```
#ll
```

```
drwxr-xr-x. 2 hotdb hotdb 4096 Aug 26 17:49 bin
```

```
drwxr-xr-x. 2 hotdb hotdb 4096 Aug 19 10:35 conf
```

```
drwxr-xr-x. 2 hotdb hotdb 4096 Aug 19 10:35 keys
```

```
drwxr-xr-x. 2 hotdb hotdb 4096 Aug 19 10:35 lib
```

```
drwxr-xr-x. 2 hotdb hotdb 4096 Jul 8 11:33 logs
```

```
lrwxrwxrwx. 1 root root 58 Aug 26 17:50 ndbsql_bin ->
```

```
/usr/local/mysql-cluster-gpl-7.5.12-linux-glibc2.12-x86_64
```

```
drwxr-xr-x. 3 hotdb hotdb 4096 Aug 26 17:53 ndbssl_data
```

```
drwxr-xr-x. 2 hotdb hotdb 4096 Aug 19 10:35 utils
```

**Copy the NDB SQL from backup directory to the upgraded compute node directory.**

```
#cp -rp /usr/local/hotdb/hotdb_server_253/ndbssl* /usr/local/hotdb/hotdb-server/
```

## 3.2.4. Update Configuration File

### server.xml Configuration File

It is suggested to open the server.xml configuration file under the compute node directory backed up, and synchronize the changed parameter values to the configuration file under the new version directory.

```
# cd /usr/local/hotdb/hotdb-server/conf/

# vi server.xml

-----Configuration Content-----

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE hotdb:server SYSTEM "server.dtd">

<hotdb:server xmlns:hotdb="http://cn.hotpu/">

    <system>

        <property name="url">jdbc:mysql://192.168.210.30:3307/hotdb_config_test253</property><!--
Master ConfigDB address, the real IP address of ConfigDB service shall be specified      -->

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

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

        <property name="bakUrl">jdbc:mysql://192.168.210.30:3307/hotdb_config_test253</property><!--
Slave ConfigDB address, the real IP address of ConfigDB service shall be specified      -->

        <property name="bakUsername">hotdb_config</property><!--
```



```

Slave ConfigDB username                                -->

    <property name="bakPassword">hotdb_config</property><!--

Slave ConfigDB password                                -->

    <property name="configMGR">>false</property><!--

Whether the ConfigDB uses MGR                          -->

    <property name="bakUrl"></property><!--

MGR ConfigDB address (if ConfigDB uses MGR, this item must be configured), the real IP address of ConfigDB
service shall be specified                              -->

    <property name="bakUsername"></property><!--

MGR ConfigDB username (if ConfigDB uses MGR, this item must be configured)                      -->

    <property name="bakPassword"></property><!--

MGR ConfigDB password (if ConfigDB uses MGR, this item must be configured)                  -->

    <property name="haMode">1</property><!--

High Availability Mode: 0: Master/Slave; 1: Cluster                                          -->

    <property name="serverId">1</property><!--

Cluster Node Number 1-N (Number of Node), unique in cluster and N<=the total number of nodes in cluster

    -->

    <property name="clusterName">HotDB-Cluster30</property><!--

Cluster Group Name                                    -->

    <property name="clusterSize">3</property><!--

Total number of nodes in cluster                      -->

    <property name="clusterNetwork">192.168.210.0/24</property><!--

Cluster Network                                        -->

    <property name="clusterHost">192.168.210.30</property><!--

The IP of this node                                    -->

    <property name="clusterPort">3326</property><!--

Cluster communication port                            -->

    <property name="haState">master</property><!--
  
```

HA role, master node: master, slave node: backup (this item is invalid in cluster mode)

```
-->
```

```
<property name="haNodeHost"></property><!--
```

HA role, other nodes IP: PORT (used under master/slave mode, PORT means manager port, for example:

192.168.200.2:3325)

```
<property name="serverPort">3323</property><!--
```

Server Port

```
-->
```

```
<property name="managerPort">3325</property><!--
```

Manager Port

```
-->
```

```
<property name="processors">16</property><!--
```

Number of Processors

```
-->
```

.....(some parameters are omitted here, subject to the actual server.xml)

```
</hotdb:server>
```

## Compute node start script

It is suggested to open the hotdb\_server script under the compute node bin/directory backed up, and synchronize the changed parameter values to the script file under the new version directory.

```
#!/bin/sh

PID_DIR="$HOTDB_HOME"/run

PID_FILE="$PID_DIR"/hotdb-server.pid

HA_STARTUP="$HOTDB_HOME"/bin/keepalived

DRIVER_DIR="$HOTDB_HOME"/utils

HOTDB_LOGS="$HOTDB_HOME"/logs/hotdb.log
```

```
HOTDB_CONSOLE_LOG="$HOTDB_HOME"/logs/console.log

DRIVER_PACKAGE="$DRIVER_DIR"/aksusbd-7*.tar.gz

TMPFILE_DIR="$HOTDB_HOME"/HotDB-TEMP

JAVA_BIN="$HOTDB_HOME"/../jdk/bin/java

JAVA_VERSION="1.7.0_80"


#with CMS Garbage Collection

JAVA_OPTS="-server -Xms4G -Xmx4G -XX:MaxDirectMemorySize=24G"

#with G1 Garbage Collection

#JAVA_OPTS="-server -Xms16G -Xmx16G -XX:MaxDirectMemorySize=24G"

.....(some parameters are omitted here, subject to the actual hotdb_server script)

HOTDB_CLASSPATH="$HOTDB_HOME/conf:$HOTDB_HOME/lib/classes"
```

## Other Configuration Files

In addition to the values of compute node configuration in old version needing to be synchronized with server.xml configuration file and compute node start script, you should also check whether relevant parameter setting of log4j2.xml under conf/directory is changed.

### 3.2.5. Start compute node and enable exception handling

#### Start compute node service

```
#cd /usr/local/hotdb/hotdb-server/bin

#sh hotdb_server start
```

#### Enable exception handling

#### Problems and handling methods

Compute node log error, and service cannot be started

- Rename the new version directory, and rename the old directory to hotdb-server.
- Restore the ConfigDB with ConfigDB file previously backed up
- Restart compute node service in old version directory
- Analyze the start failure, and re-upgrade it with the above steps after solving it.

### 3.3. Master/slave node cluster upgrade with service stopped

Version upgrade of master/slave compute node in master/slave mode cluster with service stopped will be instructed below. The compute node will not provide any service in upgrading. Please note that the upgrade is executed when no operation is in progress.

#### 3.3.1. ConfigDB Upgrade

The steps of Compute Node ConfigDB upgrade are the same as those of management platform ConfigDB upgrade, but the upgrade SQL content executed is different from the ConfigDB address connected. Please refer to the management platform [ConfigDB Upgrade Instructions](#) for specific steps.

#### 3.3.2. Stop compute node and keepalived service

**Stop slave keepalived service**

```
#service keepalived stop
```

**Stop slave compute node service**

```
#sh /usr/local/hotdb/hotdb-server/bin/hotdb_server stop
```

**Stop master keepalived service**

```
#service keepalived stop
```

**Stop master compute node service**

```
#sh /usr/local/hotdb/hotdb-server/bin/hotdb_server stop
```

### 3.3.3. Upgrade master/slave compute node version

Please refer to [Compute Node Version Upgrade](#) operation instructions in single node cluster mode for replacing upgrade package, updating configuration file, etc.

### 3.3.4. NDB SQL Service

If compute node version is V2.5.3 and above before upgrade, please check whether compute node directory previously backed up contains NDB SQL service (directory starting with ndbsql). If it contains, please re-copy the NDB SQL directory in compute node directory previously backed up to the upgraded compute node directory.

Please refer to [Single Node Cluster Compute Node Upgrade NDB SQL Service](#) for specific operations.

### 3.3.5. Start keepalived and compute node service

Start master compute node service first; after the compute node service is completely started (server port and manager port can be connected normally), start the master compute node keepalived service.

**Start master computer node service**

```
#cd /usr/local/hotdb/hotdb-server/bin
```

```
#sh hotdb_server start
```

**Start master keepalived service**

```
#service keepalived start
```

After VIP address of master keepalived is connected through ping, start slave compute node service. After slave compute node service is completely started (server port is closed, and manager port can be connected normally), start slave keepalived service.

**Start slave computer node service**

```
#cd /usr/local/hotdb/hotdb-server/bin
```

```
#sh hotdb_server start
```

**Start slave keepalived service**

```
#service keepalived start
```

## 3.4. Master/slave node cluster upgrade without service stopped

Version upgrade of master/slave compute node in master/slave node cluster mode without service stopped will be instructed below. A high availability switch of compute node will occur during the upgrade, which will temporarily disconnect the client application during the switch. It is suggested to upgrade in production environment during the low peak of operation.

### 3.4.1. Upgrade conditions without service stopped

Upgrade without service stopped of master/slave node cluster mode shall meet the following conditions, or [compute node version can only be upgraded with service stopped](#).

- SQL for Compute node ConfigDB upgrade does not contain any statement for “alter table” modifying existing column(except for modifying the length or range of the added column).
- SQL for Compute node ConfigDB upgrade does not contain any statement of “drop table”.
- SQL for Compute node ConfigDB upgrade does not contain any statement of

“update\delete” existing data.

### 3.4.2. High Availability Switch Check

High availability switch is required when compute node is upgraded. To ensure the smooth completion of switch, it is necessary to manually check whether the current high availability environment conforms to switch conditions. All the following check items should be met, or the high availability switch may fail.

#### High Availability Switch Check Items

##### 1. Master/slave compute node service is normal

Requirement: Current master compute node server port (3323 by default) is open normally; manager port (3325 by default) is open normally; current slave compute node server port is closed, and manager port is open normally.

##### 2. Master/slave compute node configuration file server.xml configuration is correct

###### ➤ Current master compute node server.xml configuration

```
<property name="haState">master</property>< HA role, master node: master, slave node: backup><property name="haNodeHost"></property><HA role, other nodes IP:PORT>
```

###### ➤ Current slave compute node server.xml configuration

```
<property name="haState">backup</property>< HA role, master node: master, slave node: backup>
```

```
<property name="haNodeHost">192.168.200.190:3325</property><HA role, other nodes IP:PORT>
```

**Note:** the above IP address shall be the IP address of server of the current master compute node, and the port number is the manager port of current master compute node

##### 3. Master/slave keepalived configuration file keepalived.conf configuration is correct

###### ➤ Current master compute node keepalived.conf configuration

```
! Configuration File for keepalived
```

```
global_defs {
```

```
router_id HotDB Server-ha

}

vrrp_script check_HotDB Server_process {

script "/bin/bash /usr/local/hotdb/hotdb-server/bin/check_hotdb_p

rocess.sh process"

interval 5

fall 2

rise 1

weight -10

}

vrrp_script check_HotDB Server_connect_state {

state

code

script "/bin/bash /usr/local/hotdb/hotdb-server/bin/check_hotdb_p

rocess.sh connect_master"

interval 5

fall 3

rise 1

timeout 5

weight -10

}

vrrp_instance VI_1 {

state BACKUP

interface eth1

virtual_router_id 89

nopreempt

priority 100
```



```
advert_int 1

authentication {

auth_type PASS

auth_pass 1111

}

track_script {

check_HotDB Server_process

check_HotDB Server_connect_state

}

#be careful in red hat

track_interface {

eth1

}

virtual_ipaddress {

192.168.200.140/24 dev eth1 label eth1:1

}

notify_master "/bin/bash /usr/local/hotdb/hotdb-server/bin/chec

k_hotdb_process.sh master_notify_master"

notify_backup "/bin/bash /usr/local/hotdb/hotdb-server/bin/chec

k_hotdb_process.sh master_notify_backup"

notify_fault "/bin/bash /usr/local/hotdb/hotdb-server/bin/check_hotdb_process.sh master_notify_backup"

}

➤ Current slave compute node keepalived.conf configuration

! Configuration File for keepalived

global_defs {

router_id HotDB Server-ha
```

```
}

vrp_script check_HotDB_Server_process {

script "/bin/bash /usr/local/hotdb/hotdb-server/bin/check_hotdb_p

rocess.sh process"

interval 5

fall 2

rise 1

weight -10

}

vrp_script check_HotDB_Server_connect_state {

state

code

script "/bin/bash /usr/local/hotdb/hotdb-server/bin/check_hotdb_p

rocess.sh connect_backup"

interval 5

fall 3

rise 1

timeout 5

weight -10

}

vrp_instance VI_1 {

state BACKUP

interface eth0

virtual_router_id 89

priority 95

advert_int 1

authentication {
```

```

auth_type PASS

auth_pass 1111

}

track_script {

check_HotDB Server_process

check_HotDB Server_connect_state

}

#be careful in red hat

track_interface {

eth0

}

virtual_ipaddress {

192.168.200.140/24 dev eth0 label eth0:1

}

notify_master "/bin/bash /usr/local/hotdb/hotdb-server/bin/chec

k_hotdb_process.sh backup_notify_master"

notify_backup "/bin/bash /usr/local/hotdb/hotdb-server/bin/chec

k_hotdb_process.sh backup_notify_backup"

notify_fault "/bin/bash /usr/local/hotdb/hotdb-server/bin/check_hotdb_process.sh backup_notify_backup"

}

```

#### 4. Configuration check is normal

- Configuration check is normal. You can check whether ConfigDB configuration is correct in the menu “Configuration->Configuration Check” in management platform
- Compute node memory information is consistent with ConfigDB. You can execute the command reload @@config through “dynamic loading” function of management platform or logging in to manager port (3325 by default) to ensure the consistency of the information

#### 5. Keepalived is running normally

- Master/slave keepalived is running normally. You can inquire it in master/slave compute node server through command “service keepalived status”

## 6. The VIP of Keepalived is on current master compute node

- Execute the “ip addr” on current master compute node server, and the display contains the virtual IP address of keepalived configuration)

### 3.4.3. ConfigDB Upgrade

The steps of Compute Node ConfigDB upgrade are the same as those of management platform ConfigDB upgrade, but the upgrade SQL content executed is different from the ConfigDB address connected. Please refer to the management platform [ConfigDB Upgrade Instructions](#) for specific steps.

### 3.4.4. Slave Compute Node Upgrade

#### Stop Slave Compute Node Service

Log in to slave compute node server and execute the command Stop Service:

```
#sh /usr/local/hotdb/hotdb-server/bin/hotdb_server stop
```

#### Back Up Slave Compute Node Directory

```
#cd /usr/local/hotdb/  
  
#mv hotdb-server hotdb_server_249
```

#### Upload and Unzip New Version Package

##### Upload new version package

rz command or ftp file transfer tool can be used to upload new version package

##### Unzip new version package

```
#tar -xvf hotdb-server-2.4.9-ga-20190812.tar.gz -C /usr/local/hotdb
```

**Folder authority is given to hotdb users**

```
#chown -R hotdb:hotdb /usr/local/hotdb/hotdb-server
```

**Restore directory default context**

```
#restorecon -R /usr/local/hotdb/hotdb-server
```

## NDB SQL Service

If compute node version is V2.5.3 and above before upgrade, please check whether compute node directory previously backed up contains NDB SQL service (directory starting with ndbsql). If it contains, please re-copy the NDB SQL directory in compute node directory previously backed up to the upgraded compute node directory.

Please refer to [Single Node Cluster Compute Node Upgrade NDB SQL Service](#) for specific operations.

## server.xml Configuration File

It is suggested to open the server.xml configuration file under the old compute node directory, and synchronize the changed parameter values to the configuration file under the new version directory.

```
# cd /usr/local/hotdb/hotdb-server/conf/
```

```
# vi server.xml
```

-----Configuration Content-----

Note: The reference modification is not repeated here, please view the [Update server.xml Configuration File](#) in single node cluster mode upgrade.

## Compute Node Start Script

It is suggested to open the hotdb\_server script under the compute node bin/directory backed up, and synchronize the changed parameter values to the script file under the

new version directory.

```
# cd /usr/local/hotdb/hotdb-server/bin/
```

```
# vi hotdb_server
```

-----Configuration Content-----

Note: The reference modification is not repeated here, please view the Update [Compute Node Start Script](#) in single node cluster node upgrade.

## Other Configuration Files

In addition to the values of compute node configuration in old version needing to be synchronized with server.xml configuration file and compute node start script, you should also check whether relevant parameter setting of log4j2.xml under conf/directory is changed.

## Start Slave Compute Node Service

```
#cd /usr/local/hotdb/hotdb-server/bin
```

```
#sh hotdb_server start
```

Note: When server port (3323 by default) is closed and manager port (3325 by default) is open after slave compute node service is started, it is normal.

### 3.4.5. Master Compute Node Upgrade

## Stop Master Compute Node Service

**Log in to master compute node server and execute the command Stop Service:**

```
#cd /usr/local/hotdb/hotdb-server/bin
```

```
#sh hotdb_server stop
```

## Check Whether High Availability Switch is Successful

When master compute node service is closed, high availability switch of the program will occur, and then the **server port (3323 by default)** of the slave compute node **will be started**,

And when executing the command “ip addr” on slave compute node server, you can check that the **virtual IP address of keepalived has been drifted**.

Note: If any of the above requirement fails to be met, then the high availability switch fails, and you need to change the upgrade without service stopped to the upgrade with service stopped.

## Back Up Master Compute Node Directory

```
#cd /usr/local/hotdb/  
  
#mv hotdb-server hotdb-server_249
```

## Upload and Unzip New Version Package

### Upload new version package

rz command or ftp file transfer tool can be used to upload new version package

### Unzip new version package

```
#tar -xvf hotdb-server-2.4.9-ga-20190812.tar.gz -C /usr/local/hotdb
```

### Folder authority is given to hotdb users

```
#chown -R hotdb:hotdb /usr/local/hotdb/hotdb-server
```

### Restore directory default context

```
#restorecon -R /usr/local/hotdb/hotdb-server
```

## NDB SQL Service

If compute node version is V2.5.3 and above before upgrade, please check whether compute node directory previously backed up contains NDB SQL service (directory starting with ndbsql). If it contains, please re-copy the NDB SQL directory in

compute node directory previously backed up to the upgraded compute node directory.

Please refer to [Single Node Cluster Compute Node Upgrade NDB SQL Service](#) for specific operations.

## server.xml Configuration File

It is suggested to open the server.xml configuration file under the old compute node directory, and synchronize the changed parameter values to the configuration file under the new version directory.

```
# cd /usr/local/hotdb/hotdb-server/conf/
```

```
# vi server.xml
```

-----Configuration Content-----

Note: The reference modification is not repeated for details, please view the [Update server.xml Configuration File](#) in single node cluster node upgrade.

## Compute Node Startup Script

It is suggested to open the hotdb\_server script under the compute node bin/directory backed up, and synchronize the changed parameter values to the script file under the new version directory.

```
# cd /usr/local/hotdb/hotdb-server/bin/
```

```
# vi hotdb_server
```

-----Configuration Content-----

Note: The reference modification is not repeated for details. Please view the [Update Compute Node Start Script](#) in single node cluster node upgrade.

## Other Configuration Files

In addition to the values of compute node configuration in old version needing to be synchronized with server.xml configuration file and compute node start script, you



should also check whether relevant parameter setting of log4j2.xml under conf/directory is changed.

## Manually Execute High Availability Environment

### Reconstruction

To ensure the normal start of current slave compute node, high availability environment reconstruction needs to be manually executed. If the version of the management platform used is V2.4.8 and above, the “high availability reconstruction” can be used to replace the following operations.

Note: “current slave” in the following operation instruction is the compute node without VIP (keepalived virtual IP) on compute node server, “current master” is the compute node where VIP is located. The command “ip addr” can be executed on master/slave compute node servers to check the drift location of current VIP, so as to confirm the master/slave status of current compute node.

#### 1. Stop current slave (without VIP) keepalived service

```
#service keepalived stop
```

#### 2. Current master (with VIP) compute node server.xml and keepalived.conf modification

##### Current master (with VIP) compute node server.xml configuration modification

```
<property name="haState">master</property>< HA role, master mode: master, slave node: backup><property name="haNodeHost"></property><HA role, other nodes IP:PORT>
```

##### Current slave (without VIP) compute node keepalived.conf configuration modification

```
! Configuration File for keepalived
```

```
global_defs {
```

```
router_id HotDB Server-ha
```

```
}
```

```
vrrp_script check_HotDB Server_process {
```

```
script "/bin/bash /usr/local/hotdb/hotdb-server/bin/check_hotdb_p
```

```
rocess.sh process"

interval 5

fall 2

rise 1

weight -10

}

vrrp_script check_HotDB_Server_connect_state {

state

code

script "/bin/bash /usr/local/hotdb/hotdb-server/bin/check_hotdb_p

rocess.sh connect_master"

interval 5

fall 3

rise 1

timeout 5

weight -10

}

vrrp_instance VI_1 {

state BACKUP

interface eth1

virtual_router_id 89

nopreempt

priority 100

advert_int 1

authentication {

auth_type PASS

auth_pass 1111
```

```

}

track_script {

check_HotDB Server_process

check_HotDB Server_connect_state

}

#be careful in red hat

track_interface {

eth1

}

virtual_ipaddress {

192.168.200.140/24 dev eth1 label eth1:1

}

notify_master "/bin/bash /usr/local/hotdb/hotdb-server/bin/chec

k_hotdb_process.sh master_notify_master"

notify_backup "/bin/bash /usr/local/hotdb/hotdb-server/bin/chec

k_hotdb_process.sh master_notify_backup"

notify_fault "/bin/bash /usr/local/hotdb/hotdb-server/bin/check_hotdb_process.sh master_notify_backup"

}

```

### 3. Current slave (without VIP) compute node server.xml and keepalived.conf modification

#### Current slave compute node server.xml configuration

```

<property name="haState">backup</property>< HA role, master node: master, slave node: backup>

<property name="haNodeHost">192.168.200.190:3325</property><HA role, other nodes IP:PORT>

```

Note: the above IP address shall be the IP address of server of the current master compute node, and the port number is the manager port of current master compute node

#### Current slave (without VIP) compute node keepalived.conf configuration

```
! Configuration File for keepalived

global_defs {

router_id HotDB Server-ha

}

vrrp_script check_HotDB Server_process {

script "/bin/bash /usr/local/hotdb/hotdb-server/bin/check_hotdb_p

rocess.sh process"

interval 5

fall 2

rise 1

weight -10

}

vrrp_script check_HotDB Server_connect_state {

state

code

script "/bin/bash /usr/local/hotdb/hotdb-server/bin/check_hotdb_p

rocess.sh connect_backup"

interval 5

fall 3

rise 1

timeout 5

weight -10

}

vrrp_instance VI_1 {

state BACKUP

interface eth0

virtual_router_id 89
```

```
priority 95

advert_int 1

authentication {

auth_type PASS

auth_pass 1111

}

track_script {

check_HotDB Server_process

check_HotDB Server_connect_state

}

#be careful in red hat

track_interface {

eth0

}

virtual_ipaddress {

192.168.200.140/24 dev eth0 label eth0:1

}

notify_master "/bin/bash /usr/local/hotdb/hotdb-server/bin/chec

k_hotdb_process.sh backup_notify_master"

notify_backup "/bin/bash /usr/local/hotdb/hotdb-server/bin/chec

k_hotdb_process.sh backup_notify_backup"

notify_fault "/bin/bash /usr/local/hotdb/hotdb-server/bin/check_hotdb_process.sh backup_notify_backup"

}
```

#### 4. Current master (with VIP) keepalived service execute configuration reloading

```
#service keepalived reload
```

## Start current slave (without VIP) compute node service and keepalived

### Start compute node service

```
#cd /usr/local/hotdb/hotdb-server/bin
```

```
#sh hotdb_server start
```

### Start keepalived service after successfully starting compute node service

```
#service keepalived start
```

## Special instructions

- After the completion of upgrade without service stopped, the virtual IP of keepalived will be drifted. The master/slave compute node roles in the cluster will be swapped.
- If master/slave compute node roles need to be restored to their state before upgrade, you can swap compute node roles again through “high availability switch” function provided by management platform or through manual switch (directly stop the simulated failover of current primary service).
- A high availability reconstruction must be executed after completion of switch, through “high availability reconstruction” function provided by management platform or by reference to “[manually execute high availability environment reconstruction](#)”.

## 3.5. Multi-node cluster upgrade with service stopped

Upgrade of version of three compute nodes in multi-node cluster mode will be instructed below. Multi-node cluster with more than three compute nodes can also be upgraded by reference to the following operation instructions.

## Stop compute node service

### Stop secondary1 compute node service

```
#sh /usr/local/hotdb/hotdb-server/bin/hotdb_server stop
```

### Stop secondary2 compute node service

```
#sh /usr/local/hotdb/hotdb-server/bin/hotdb_server stop
```

### Stop primary compute node service

```
#sh /usr/local/hotdb/hotdb-server/bin/hotdb_server stop
```

## Upgrade ConfigDB

The steps of Compute Node ConfigDB upgrade are the same as those of management platform ConfigDB upgrade, but the upgrade SQL content executed is different from the ConfigDB address connected. Please refer to the management platform [ConfigDB Upgrade Instructions](#) for specific steps.

## Replace upgrade package and upgrade configuration file

Version package is replaced and corresponding configuration file content is upgraded successively according to [Compute Node Version Upgrade](#) operation instruction in single node cluster mode.

## Start compute node service

Compute node service program is started in no particular order, and can be started successively and then log in to any compute node manager port (3325 by default) and execute the command “show @@cluster” to inquire the running status and role information of compute node in current cluster.

## 3.6. Multi-node cluster upgrade without service stopped

Upgrade of version of three compute nodes in multi-node cluster mode will be instructed below. Multi-node cluster with more than three compute nodes can also be upgraded by reference to the following operation instructions.

### 3.6.1. Upgrade Conditions Without Service Stopped

Upgrade without service stopped of multi-node cluster mode shall meet the following conditions, or [compute node version can only be upgraded with service stopped](#).

- The upgrade SQL for Compute node ConfigDB does not contain any statement of “alter table” modifying existing column (except for statements of modifying the length or range of the added column).
- The upgrade SQL of Compute node ConfigDB does not contain any statement “drop table”.
- The upgrade SQL for Compute node ConfigDB does not contain any statement of “update\delete” existing data.
- The compute node version before upgrade shall be V2.5.1 and above.
- If the compute node version before upgrade is V2.5.1 or V2.5.3, the date in the version must be greater than or equal to 20190821 (August 21, 2019).

#### Upgrade Instructions:

Upgrading secondary compute node in cluster will cause interruption of client connection on the secondary compute node; upgrading primary compute node will cause interruption not only of client connection, but also of some transactions in progress. It is suggested to upgrade in production environment during the low peak of operation.

**Special Instructions:** primary and secondary are the roles of current compute node displayed when executing command “show @@cluster” in manager port (3325 by default). The secondary1 and secondary2 below represent the compute nodes with



secondary attribute in three compute node clusters, whose numbers at the end are in no particular order and can represent any secondary compute node.

### 3.6.2. ConfigDB Upgrade

The steps of Compute Node ConfigDB upgrade are the same as those of management platform ConfigDB upgrade, but the upgrade SQL content executed is different from the ConfigDB address connected. Please refer to the management platform [ConfigDB Upgrade Instructions](#) for specific steps.

### 3.6.3. Secondary1 Compute Node Upgrade

Please refer to [Compute Node Version Upgrade](#) operation instructions in single node cluster mode for replacing upgrade package, updating configuration file, and starting compute node service.

### 3.6.4. Secondary2 Compute Node Upgrade

Please refer to [Compute Node Version Upgrade](#) operation instructions in single node cluster mode for replacing upgrade package, updating configuration file, and starting compute node service.

### 3.6.5. Primary Compute Node Upgrade

#### Stop Primary Compute Node Service

```
#cd /usr/local/hotdb/hotdb-server/bin
```

```
#sh hotdb_server stop
```

Note: Stopping primary compute node service will cause switch of master node in cluster. Please ensure the normal operation of other secondary compute node services before the primary compute node is stopped.

Please refer to [Compute Node Version Upgrade](#) operation instructions in single node cluster mode for replacing upgrade package, updating configuration file, starting

compute node service, etc.

## 3.7. Cluster upgrade in disaster recovery mode

To manually upgrade a compute node cluster with disaster recovery mode enabled, you need to upgrade with IDC as the unit.

If the IDC switching occurs, it is recommended to repair the IDC failure before upgrading. That is to say, for the cluster of disaster recovery mode, it is recommended to upgrade manually when there are no abnormal components in the two IDCs and the replication status of the IDC is normal.

For the cluster upgrade of the disaster recovery mode, the master center must be upgraded first. Upgrade the DR center after ensuring that the upgrade is correct.

### 3.7.1. Master center upgrade

The upgrade of the master center in the disaster recovery mode is roughly the same as that of the cluster in the single-IDC mode.

If the compute nodes of the master center are in the single-node mode, please refer to the [Single Node Cluster Mode Upgrade](#).

If the compute nodes of the master center are in the master/slave node mode, please refer to the [Master/slave node cluster upgrade with service stopped](#) or the [Master/slave node cluster upgrade without service stopped](#).

If the compute nodes of the master center are in the multi-node mode, please refer to the [Multi-node cluster upgrade with service stopped](#) or the [Multi-node cluster upgrade without service stopped](#).

### 3.7.2. DR center upgrade

Because the DR center does not provide real database services, in the day-to-day operation and maintenance process, the compute nodes of the DR center only open the

management port or are in the shutdown state.

If the compute nodes of the DR center are in the single-node mode, please refer to the [Single Node Cluster Mode Upgrade](#).

If the compute nodes of the DR center are in the master/slave node mode, please refer to the [Master/slave node cluster upgrade with service stopped](#) or the [Master/slave node cluster upgrade without service stopped](#).

If the compute nodes of the master center are in the multi-node mode, please refer to the [Multi-node cluster upgrade with service stopped](#) or the [Multi-node cluster upgrade without service stopped](#).

Notice: There is no need to upgrade ConfigDB during the upgrade the DR center. It will be automatically synchronized through the MySQL replication relation between ConfigDBs in the master center and DR center during the upgrade of the ConfigDB of the master center.

## 4. HotDB Backup Manual Upgrade

HotDB Backup does not necessarily need to be frequently upgraded, only when the HotDB Backup has great function optimization or bug repairs. When updating HotDB Backup, all HotDB Backups deployed on data source server shall be upgraded in the same way.

### 4.1. Stop HotDB Backup Service

Before stopping HotDB Backup service, please confirm that there is no uncompleted backup task on current management platform, or the backup task may be interrupted abnormally.

```
#cd /usr/local/hotdb/hotdb-backup/bin  
  
#sh hotdb_backup stop
```

### 4.2. Back Up Old HotDB Backup Directory

```
#cd /usr/local/hotdb/  
  
#mv hotdb-backup/ hotdb-backup_1.0
```

### 4.3. Upload and Unzip New Version Package

#### Upload new version package

rz command or ftp file transfer tool can be used to upload new version package

#### Unzip new version package

```
#tar -xvf hotdb-backup-2.0-20190109.tar.gz -C /usr/local/hotdb
```

#### Folder authority is given to hotdb users

```
#chown -R hotdb:hotdb /usr/local/hotdb/hotdb-backup
```

#### Restore directory default context

```
#restorecon -R /usr/local/hotdb/hotdb-backup
```

## 4.4. Start Service

```
#cd /usr/local/hotdb/hotdb-backup/bin
```

```
#sh hotdb_backup start -h 192.168.220.104 -p 3322
```

Note: IP address is the address of management platform server associated with HotDB Backup, and the port number is the parameter value of server.backup.port in the management platform configuration file application.properties

**Special instruction:** The HotDB Backup upgrades in other data source servers just follow the above process.

## 5. HotDB Listener Manual Upgrade

HotDB Listener generally does not need to upgrade frequently. It can only be updated when Listener has large function optimization or bug repair. Updating the Listener requires the same upgrade operation to be performed on the Listener deployed on all data source servers.

### 5.1. Stop Listener Service

Before stopping the Listener service, please confirm whether the current compute node is under SQL operation, or there may be transaction loss.

```
#cd /usr/local/hotdb/hotdb-listener/bin  
  
#sh hotdb_listener stop
```

### 5.2. Back Up Old HotDB Backup Directory

```
#cd /usr/local/hotdb/  
  
#mv hotdb-listener/ hotdb-listener_1.0
```

### 5.3. Upload and Unzip New Version Package

Upload new version package

You can use the rz command or ftp to upload a new version package

Unzip the new version package

```
#tar -zxvf hotdb-listener-0.0.1-linux.tar.gz -C /usr/local/hotdb
```

Give hotdb users folder privilege

```
#chown -R hotdb:hotdb /usr/local/hotdb/hotdb-listener
```

Restore the default content

```
#restorecon -R /usr/local/hotdb/hotdb-listener
```

## 5.4. Modify Configuration File

Modifying the management port configuration file is the same as that before the service is stopped. 3330 by default:

```
#cd /usr/local/hotdb/hotdb-listener/conf  
  
#vi config.properties  
  
host=0.0.0.0  
  
port=3330
```

## 5.5. Start Service

```
#cd /usr/local/hotdb/hotdb-listener/bin  
  
#sh hotdb_listener start
```

**Special instruction:** Listener upgrade on other data source servers only needs to follow the above process.