Cloud Data Migration

FAQs

Issue	01
Date	2025-02-11

Security Declaration

Vulnerability

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

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

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

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

Contents

1 General

1.1 What Are the Differences Between CDM and Other Data Migration Services?

HUAWEI CLOUD provides the following data migration services:

- **Cloud Data Migration (CDM)**
- **Object Storage Migration Service (OMS)**
- **Data Replication Service (DRS)**
- **Server Migration Service (SMS)**
- **Database and Application Migration UGO**
- **Data Express Service (DES)**

For details about the differences between the preceding services, see **Differences Between Data Migration Services**.

CDM

Cloud Data Migration (CDM) is an efficient and easy-to-use batch data integration service. Based on the big data migration to the cloud and intelligent data lake solution, CDM provides easy-to-use migration capabilities and capabilities of integrating multiple data sources to the data lake, reducing the complexity of data source migration and integration and effectively improving the data migration and integration efficiency. For details, see **What Is CDM?**.

CDM can migrate data from a database or object storage to a data lake or big data system.

Differences between CDM and DRS:

- If the destination is a big data system, CDM is recommended.
- If the destination is an OLTP database or DWS, DRS is recommended.

Differences between CDM and OMS:

- OMS is used to migrate data from the following clouds to HUAWEI CLOUD: Amazon Web Services (AWS), Alibaba Cloud, Microsoft Azure, Baidu Cloud, QingCloud, Qiniu Cloud, and Tencent Cloud.
- CDM is mainly used to migrate OBS data to a data lake or big data system for data development, cleaning, and governance. However, use OMS if you want to migrate an entire bucket.

OMS

OMS helps you migrate data from object storage on other clouds online to the OBS on HUAWEI CLOUD. For details, see **Object Storage Migration Service**.

OMS provides the following functions:

- Online data migration: It helps you easily and smoothly migrate object storage data from the public cloud of other cloud service providers to HUAWEI CLOUD.
- Cross-region replication: It enables you to replicate and back up data across regions of HUAWEI CLOUD.

Currently, OMS can migrate object storage data of the following clouds to HUAWEI CLOUD: Amazon Web Services (AWS), Alibaba Cloud, Microsoft Azure, Baidu Cloud, HUAWEI CLOUD, Kingsoft Cloud, QingCloud, Qiniu Cloud, and Tencent Cloud.

CDM can also migrate object storage data. However, CDM and OMS differ in the following way:

- OMS is used to migrate data from other clouds to HUAWEI CLOUD.
- CDM is mainly used to migrate OBS data to a data lake or big data system for data development, cleaning, and governance.

DRS

DRS is a stable, efficient, and easy-to-use cloud service for database online migration and synchronization in real time. DRS is used to migrate data from mainstream databases to other databases (including third-party databases), for example, from OLTP to OLTP or DWS. For details, see **Data Replication Service**.

Currently, the following database links are supported:

MySQL databases on HUAWEI CLOUD or other clouds to RDS for MySQL

PostgreSQL databases on HUAWEI CLOUD or other clouds to RDS for PostgreSQL

MongoDB databases on HUAWEI CLOUD or other clouds to DDS

Oracle->RDS for MySQL

……

Differences between DRS and CDM:

- DRS migrates data to a database, for example, a MySQL or a MongoDB database.
- CDM migrates data to a data lake or big data system, for example, MRS HDFS or FusionInsight HDFS.

Differences between DRS and UGO:

- DRS is used for full/incremental data migration or synchronization.
- UGO is used for the evaluation, structure migration, and syntax conversion before a heterogeneous database migration.

SMS

SMS is a P2V/V2V migration service that helps you migrate applications and data from on-premises x86 physical servers or VMs on private or public clouds to ECSs on HUAWEI CLOUD.

UGO

UGO is a professional cloud service that focuses on heterogeneous database structure migration. It automatically converts the syntax of the DDL in databases and the database SQL statements encapsulated in service programs into the SQL syntax of GaussDB or RDS on HUAWEI CLOUD. It uses pre-migration evaluation, structure migration, and automatic syntax conversion to identify possible reconstruction in advance, improve the conversion rate, and minimize the database migration cost. For details, see **Database and Application Migration UGO**.

In short, UGO is used for the evaluation, structure migration, and syntax conversion before a heterogeneous database migration.

DES

DES provides you with physical devices to make it easier to migrate terabytes, or even petabytes of data to HUAWEI CLOUD inexpensively and much faster than would be possible over a network connection. For details, see **Data Express Service**.

Differences Between Data Migration Services

Table 1-1 Differences between data migration services

Service Name	Functions	Differences with Other Services
CDM	• Migrates big data to the cloud. • Migrates multiple data sources to the data lake.	**Differences with DRS:** DRS is used to migrate databases, while CDM is used to migrate data to big data systems.

Service Name	Functions	Differences with Other Services
OMS	Object storage migration ● Migrates object storage data from other clouds to HUAWEI CLOUD. ● Migrates data between different regions of HUAWEI CLOUD.	**Differences with CDM:** OMS is used to migrate data from other clouds to HUAWEI CLOUD, while CDM is mainly used to migrate OBS data to a data lake or big data system for data development, cleaning, and governance.
DRS	Migrates data between mainstream databases and HUAWEI CLOUD. ● Online database migration ● Real-time database synchronization	● **Differences with CDM:** DRS is used to migrate databases, while CDM is used to migrate data to big data systems. ● **Differences with UGO:** DRS is used to migrate and synchronize data between homogeneous and heterogeneous databases, while UGO is used to migrate the structure and syntax of heterogeneous databases and evaluate the migration.
SMS	Server migration Migrates data from physical servers or VMs on private or public clouds to HUAWEI CLOUD	-
UGO	● Migrates database structure. ● Evaluates database migration. ● Migrates syntax.	**Differences with DRS:** DRS is used to migrate and synchronize data between homogeneous and heterogeneous databases, while UGO is used to migrate the structure and syntax of heterogeneous databases and evaluate the migration.
DES	● Migrates terabytes or even petabytes of data to HUAWEI CLOUD. ● Uses physical media for migration.	-

1.2 What Are the Advantages of CDM?

CDM is developed based on a distributed computing framework and leverages the parallel data processing technology. **Table 1-2** details the advantages of CDM.

Table 1-2 CDM advantages

Item	User-Developed Script	CDM
Ease of use	You need to prepare server resources, and install and configure software, which is time-consuming. Because the data source types are different, the program uses different access interfaces, such as JDBC and native APIs, to read and write data. In this case, various libraries and SDKs are required when you write data migration scripts, resulting in high development and management costs.	CDM provides a web-based management console for enabling services on web pages in real time. You can migrate data by configuring data sources and migration jobs on the GUI and CDM will manage and maintain the data sources and migration jobs for you. In other words, you only need to focus on the data migration logic without worrying about the environment, which greatly reduces development and maintenance costs. CDM also provides RESTful APIs to support third-party system calling and integration.
Real-time monitoring	You need to select specific versions to develop as required.	You can use Cloud Eye to automatically monitor CDM clusters in real time and manage alarms and notifications, so that you can keep track of CDM cluster performance metrics.
O&M free	You need to develop and optimize O&M functions, especially alarm and notification functions, to ensure system availability. Otherwise, manual attendance is required.	With CDM, you do not need to maintain resources such as servers and VMs. CDM has the log, monitoring, and alarm functions, which send notifications to related personnel in a timely manner to avoid 24/7 hours of manual O&M.

Item	User-Developed Script	CDM
High efficiency	During data migration, the read and write process is completed in one job. Limited by available resources, the performance is poor and generally cannot meet the requirements of scenarios where massive sets of data need to be migrated.	Based on the distributed computing framework, CDM jobs are split into independent sub-jobs and executed concurrently, which drastically improves data migration efficiency. In addition, efficient data import interfaces are provided to import data from Hive, HBase, MySQL databases, and Data Warehouse Service (DWS).
Various data sources	Different tasks must be developed for different data sources, generating a number of scripts.	Data sources such as databases, Hadoop services, NoSQL databases, data warehouses, and files are supported.
Different network environments	As the cloud computing technology develops, user data may be stored in different environments, such as public clouds, on-premises or hosted Internet data centers (IDCs), and hybrid scenarios. In heterogeneous environments, data migration is subject to various factors, for example, network connectivity, which causes inconvenience for development and maintenance.	CDM helps you easily cope with various data migration scenarios, including data migration to the cloud, data exchange on the cloud, and data migration to on-premises service systems, regardless of whether the data is stored on on-premises IDCs, cloud services, third-party clouds, or self-built databases or file systems on ECSs.

1.3 What Are the Security Protection Mechanisms of CDM?

CDM is a fully hosted service that provides the following capabilities to protect user data security:

- Instance isolation: CDM users can use only their own instances. Instances are isolated from each other and cannot access each other.

- System hardening: System hardening for security has been performed on the operating system of the CDM instance, so attackers cannot access the operating system from the Internet.

- Key encryption: Keys of various data sources entered when users create links on CDM are stored in CDM databases using high-strength encryption algorithms.

- No intermediate storage: During data migration, CDM processes only data mapping and conversion without storing any user data or data fragments.

1.4 How Do I Reduce the Cost of Using CDM?

When migrating the data on the public network, use NAT Gateway to share the EIPs with other ECSs in the subnet. In this way, data on the on-premises data center or third-party cloud can be migrated in a more economical and convenient manner.

The following details the operations:

1. Suppose that you have created a CDM cluster (no dedicated EIP needs to be bound to the CDM cluster). Record the VPC and subnet where the CDM cluster is located.

2. Create a NAT gateway. Select the same VPC and subnet as the CDM cluster.

3. After the NAT gateway is created, return to the NAT gateway console list, click the created gateway name, and then click **Add SNAT Rule**.

Figure 1-1 Adding an SNAT rule
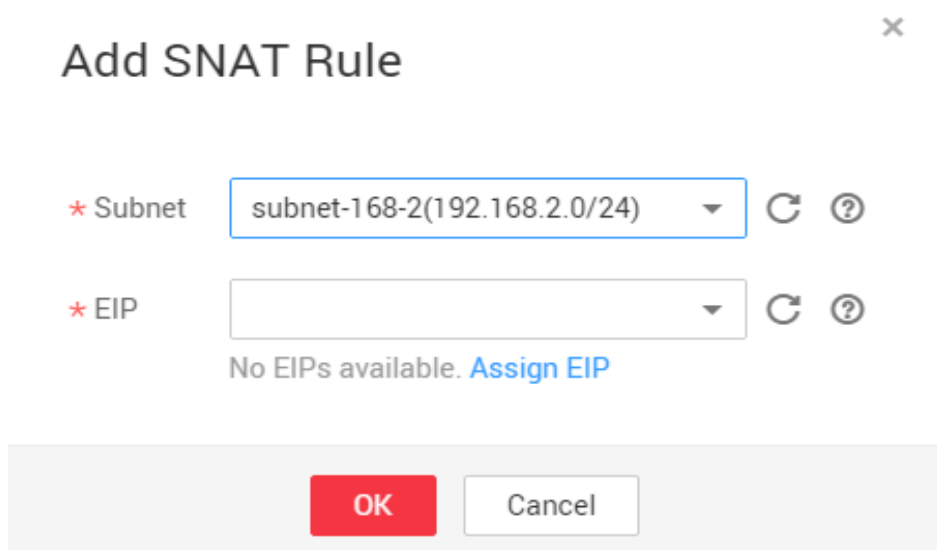
4. Select a subnet and an EIP. If no EIP is available, apply for one.

   After that, access the CDM management console and migrate data from the public network to the cloud through the Internet. For example, migrate files from the FTP server in the on-premises data center to OBS and migrate relational databases from the third-party cloud to RDS.

1.5 Will I Be Billed If My CDM Cluster Does Not Use the Data Transmission Function?

A CDM cluster is billed when it is running, even if you are not using it.

You are advised to delete the clusters that you seldom use and create clusters when needed. For details about CDM cluster billing, see **Pricing Details**.

1.6 Why Am I Billed Pay per Use When I Have Purchased a Yearly/Monthly CDM Incremental Package?

Check whether the region and specifications of the package are the same as those of the CDM cluster. If not, the package cannot be used. To view the CDM cluster specifications and region, log in to the CDM console, choose **Cluster Management** in the navigation pane, and click the cluster name in the cluster list.

If the package and the CDM cluster have the same region and specifications, pay-per-use fees are generated in the following scenarios:

If you buy a pay-per-use incremental package and then a yearly/monthly incremental package, you will be billed in pay-per-use mode first, and then in yearly/monthly mode.

1.7 How Do I Check the Remaining Validity Period of a Package?

You can check it on the official website.

Log in at the Huawei Cloud official website, select **Billing & Costs** from the username drop-down list, and choose **Orders** > **Renewals** to view the remaining validity period of the package.

1.8 Will My Data Be Retained If My Package Expires or My Pay-per-Use Resources Are in Arrears?

After a resource enters a grace period or retention period, HUAWEI CLOUD will notify you of this by email or text message. **If you still do not complete the renewal or top-up after the retention period has ended, your data stored in the DAS service will be deleted and the resources will be released.**

- Grace period: Once a monthly/yearly subscription has expired or a pay-per-use resource becomes in arrears, HUAWEI CLOUD provides a period of time during which you can renew the resource or top up your account. Within the grace period, you can still access and use your cloud service.
- Retention period: If you do not renew the yearly/monthly subscription or pay off the arrears within the grace period, the resource enters a retention period after the grace period has expired. During this period, your cloud services cannot be accessed or used, but your stored cloud data will be retained.

For details about how to set the grace period and retention period, see **Service Suspension and Resource Release**.

1.9 Can CDM Be Shared by Different Tenants?

CDM can be shared with IAM users of the same tenant through authorization.

To authorize an IAM user, perform the following steps:

1. **Create a user group and assign permissions**

   Create a user group on the IAM console, and attach the **CDM ReadOnlyAccess** policy to the group.

2. **Create an IAM user.**

   Create a user on the IAM console and add it to the user group created in **1**.

3. **Log in** and verify permissions.

   In the authorized region, perform the following operations:

   - Choose **Service List** > **Cloud Data Migration**. On the CDM console, view clusters. If no message appears indicating insufficient permissions to perform the operation, the **CDM ReadOnlyAccess** policy has already taken effect.

   - Choose any other service in **Service List**. If a message appears indicating that you have insufficient permissions to access the service, the **CDM ReadOnlyAccess** policy has already taken effect.

1.10 Can I Upgrade a CDM Cluster?

No. To use a later version cluster, you can create one.

1.11 How Is the Migration Performance of CDM?

Theoretically, a cdm.large CDM instance can migrate 1 TB to 8 TB data per day. The actual transmission rate is affected by factors such as the Internet bandwidth, cluster specifications, file read/write speed, number of concurrent jobs, and disk read/write performance. For details, see **Performance White Paper**.

1.12 What Is the Number of Concurrent Jobs for Different CDM Cluster Versions?

CDM migrates data through data migration jobs. It works in the following way:

1. When data migration jobs are submitted, CDM splits each job into multiple tasks based on the **Concurrent Extractors** parameter in the job configuration.

   📖 **NOTE**

   Jobs for different data sources may be split based on different dimensions. Some jobs may not be split based on the **Concurrent Extractors** parameter.

2. CDM submits the tasks to the running pool in sequence. Tasks (defined by **Maximum Concurrent Extractors**) run concurrently. Excess tasks are queued.

Changing Concurrent Extractors

1. The maximum number of concurrent extractors for a cluster varies depending on the CDM cluster flavor. You are advised to set the maximum number of concurrent extractors to twice the number of vCPUs of the CDM cluster.

Table 1-3 Maximum number of concurrent extractors for a CDM cluster

Flavor	vCPUs/Memory	Maximum Concurrent Extractors
cdm.large	8 vCPUs, 16 GB	16
cdm.xlarge	16 vCPUs, 32 GB	32
cdm.4xlarge	64 vCPUs, 128 GB	128

Figure 1-2 Setting Maximum Concurrent Extractors for a CDM cluster

2. Configure the number of concurrent extractors based on the following rules:

   a. When data is to be migrated to files, CDM does not support multiple concurrent tasks. In this case, set a single process to extract data.

   b. If each row of the table contains less than or equal to 1 MB data, data can be extracted concurrently. If each row contains more than 1 MB data, it is recommended that data be extracted in a single thread.

   c. Set **Concurrent Extractors** for a job based on **Maximum Concurrent Extractors** for the cluster. It is recommended that **Concurrent Extractors** is less than **Maximum Concurrent Extractors**.

   d. If the destination is DLI, you are advised to set the number of concurrent extractors to 1. Otherwise, data may fail to be written.

Figure 1-3 Setting Concurrent Extractors for a job

1.13 Can I Stop a CDM Cluster?

Clusters later than version 2.9.1.200 do not support scheduled shutdown or automatic startup and shutdown.

2 Functions

2.1 Does CDM Support Incremental Data Migration?

Symptom

Does CDM support incremental data migration?

Solution

CDM supports incremental data migration.

With scheduled jobs and macro variables of date and time, CDM provides incremental data migration in the following scenarios:

- Incremental file migration

- Incremental migration of relational databases

- HBase/CloudTable incremental migration

For details, see **Incremental Migration**.

2.2 Does CDM Support Field Conversion?

Yes. CDM supports the following field converters:

- **Anonymization**

- **Trim**

- **Reverse String**

- **Replace String**

- **Expression Conversion**

You can create a field converter on the **Map Field** page when creating a table/file migration job.

Figure 2-1 Creating a field converter
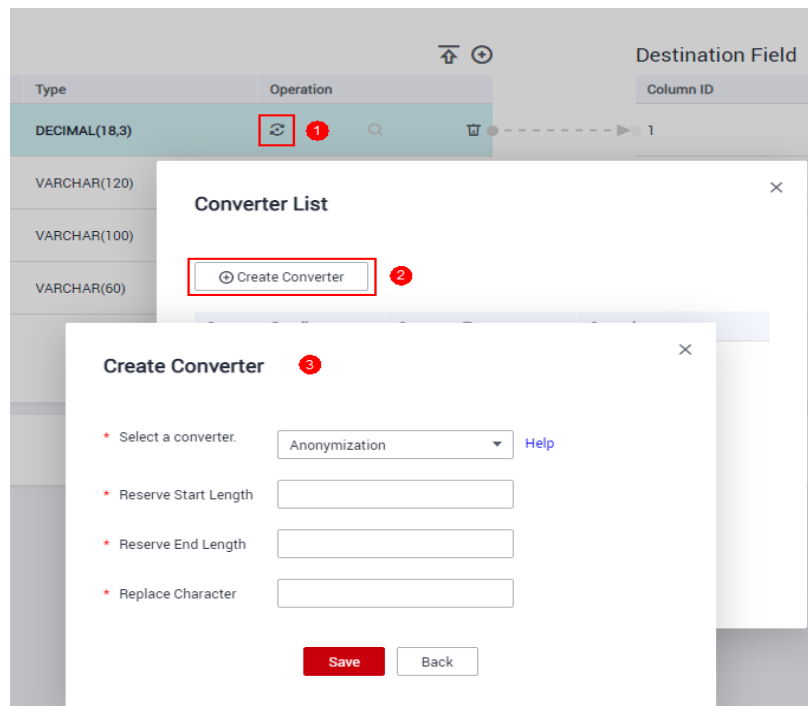
Anonymization

This converter is used to hide key information about the character string. For example, if you want to convert **12345678910** to **123****8910**, configure the parameters as follows:

- Set **Reserve Start Length** to **3**.

- Set **Reserve End Length** to **4**.

- Set **Replace Character** to ***.

Trim

This converter is used to automatically delete the spaces before and after a string. No parameters need to be configured.

Reverse String

This converter is used to automatically reverse a string. For example, reverse **ABC** into **CBA**. No parameters need to be configured.

Replace String

This converter is used to replace a character string. You need to configure the object to be replaced and the new value.

Expression Conversion

This converter uses the JSP expression language (EL) to convert the current field or a row of data. The JSP EL is used to create arithmetic and logical expressions. In

an expression, you can use integers, floating point numbers, strings, constants **true** and **false**, and **null**.

During data conversion, if the content to be replaced contains a special character, use a backslash (\) to escape the special character to a common one.

- The expression supports the following environment variables:
  - **value**: indicates the current field value.
  - **row**: indicates the current row, which is an array type.
- The expression supports the following Utils:

  a. If the field is of the string type, convert all character strings into lowercase letters, for example, convert **aBC** to **abc**.

     Expression: StringUtils.lowerCase(value)

  b. Convert all character strings of the current field to uppercase letters.

     Expression: StringUtils.upperCase(value)

  c. Convert the format of the first date field from 2018-01-05 15:15:05 to 20180105.

     Expression: DateUtils.format(DateUtils.parseDate(row[0],"yyyy-MM-dd HH:mm:ss"),"yyyyMMdd")

  d. Convert a timestamp to a date string in *yyyy-MM-dd hh:mm:ss* format, for example, convert **1701312046588** to **2023-11-30 10:40:46**.

     Expression: DateUtils.format(NumberUtils.toLong(value),"yyyy-MM-dd HH:mm:ss")

  e. Convert a date string in the yyyy-MM-dd hh:mm:ss format to a timestamp.

     Expression: DateUtils.getTime(DateUtils.parseDate(value,"yyyy-MM-dd hh:mm:ss"))

  f. If the field value is a date string in *yyyy-MM-dd* format, extract the year from the field value, for example, extract **2017** from **2017-12-01**.

     Expression: StringUtils.substringBefore(value,"-")

  g. If the field value is of the numeric type, convert the value to a new value which is two times greater than the original value:

     Expression: value*2

  h. Convert the field value **true** to **Y** and other field values to **N**.

     Expression: value=="true"?"Y":"N"

  i. If the field value is of the string type and is left empty, convert it to **Default**. Otherwise, the field value will not be converted.

     Expression: empty value? "Default":value

  j. Convert date format **2018/01/05 15:15:05** to **2018-01-05 15:15:05**:

     Expression: DateUtils.format(DateUtils.parseDate(value,"yyyy/MM/dd HH:mm:ss"),"yyyy-MM-dd HH:mm:ss")

  k. Obtain a 36-bit universally unique identifier (UUID):

     Expression: CommonUtils.randomUUID()

  l. If the field is of the string type, capitalize the first letter, for example, convert **cat** to **Cat**.

Expression: StringUtils.capitalize(value)

m. If the field is of the string type, convert the first letter to a lowercase letter, for example, convert **Cat** to **cat**.

Expression: StringUtils.uncapitalize(value)

n. If the field is of the string type, use a space to fill in the character string to the specified length and center the character string. If the length of the character string is not shorter than the specified length, do not convert the character string. For example, convert **ab** to meet the specified length 4.

Expression: StringUtils.center(value,*4*)

o. Delete a newline (including **\n**, **\r**, and **\r\n**) at the end of a character string. For example, convert **abc\r\n\r\n** to **abc\r\n**.

Expression: StringUtils.chomp(value)

p. If the string contains the specified string, **true** is returned; otherwise, **false** is returned. For example, **abc** contains **a** so that **true** is returned.

Expression: StringUtils.contains(value,"*a*")

q. If the string contains any character of the specified string, **true** is returned; otherwise, **false** is returned. For example, **zzabyycdxx** contains either **z** or **a** so that **true** is returned.

Expression: StringUtils.containsAny(value,"*za*")

r. If the string does not contain any one of the specified characters, **true** is returned. If any specified character is contained, **false** is returned. For example, **abz** contains one character of **xyz** so that **false** is returned.

Expression: StringUtils.containsNone(value,"*xyz*")

s. If the string contains only the specified characters, **true** is returned. If any other character is contained, **false** is returned. For example, **abab** contains only characters among **abc** so that **true** is returned.

Expression: StringUtils.containsOnly(value,"*abc*")

t. If the character string is empty or null, convert it to the specified character string. Otherwise, do not convert the character string. For example, convert the empty character string to null.

Expression: StringUtils.defaultIfEmpty(value,*null*)

u. If the string ends with the specified suffix (case sensitive), **true** is returned; otherwise, **false** is returned. For example, if the suffix of **abcdef** is not null, **false** is returned.

Expression: StringUtils.endsWith(value,*null*)

v. If the string is the same as the specified string (case sensitive), **true** is returned; otherwise, **false** is returned. For example, after strings **abc** and **ABC** are compared, **false** is returned.

Expression: StringUtils.equals(value,"*ABC*")

w. Obtain the first index of the specified character string in a character string. If no index is found, **-1** is returned. For example, the first index of **ab** in **aabaabaa** is 1.

Expression: StringUtils.indexOf(value,"*ab*")

x. Obtain the last index of the specified character string in a character string. If no index is found, **-1** is returned. For example, the last index of **k** in **aFkyk** is 4.

Expression: StringUtils.lastIndexOf(value,"*k*")

y. Obtain the first index of the specified character string from the position specified in the character string. If no index is found, **-1** is returned. For example, the first index of **b** obtained after the index 3 of **aabaabaa** is 5.

Expression: StringUtils.indexOf(value,"*b*",*3*)

z. Obtain the first index of any specified character in a character string. If no index is found, **-1** is returned. For example, the first index of **z** or **a** in **zzabyycdxx.** is 0.

Expression: StringUtils.indexOfAny(value,"*za*")

aa. If the string contains any Unicode character, **true** is returned; otherwise, **false** is returned. For example, **ab2c** contains only non-Unicode characters so that **false** is returned.

Expression: StringUtils.isAlpha(value)

ab. If the string contains only Unicode characters and digits, **true** is returned; otherwise, **false** is returned. For example, **ab2c** contains only Unicode characters and digits, so that **true** is returned.

Expression: StringUtils.isAlphanumeric(value)

ac. If the string contains only Unicode characters, digits, and spaces, **true** is returned; otherwise, **false** is returned. For example, **ab2c** contains only Unicode characters and digits, so that **true** is returned.

Expression: StringUtils.isAlphanumericSpace(value)

ad. If the string contains only Unicode characters and spaces, **true** is returned; otherwise, **false** is returned. For example, **ab2c** contains Unicode characters and digits so that **false** is returned.

Expression: StringUtils.isAlphaSpace(value)

ae. If the string contains only printable ASCII characters, **true** is returned; otherwise, **false** is returned. For example, for **!ab-c~**, **true** is returned.

Expression: StringUtils.isAsciiPrintable(value)

af. If the string is empty or null, **true** is returned; otherwise, **false** is returned.

Expression: StringUtils.isEmpty(value)

ag. If the string contains only Unicode digits, **true** is returned; otherwise, **false** is returned.

Expression: StringUtils.isNumeric(value)

ah. Obtain the leftmost characters of the specified length. For example, obtain the leftmost two characters **ab** from **abc**.

Expression: StringUtils.left(value,*2*)

ai. Obtain the rightmost characters of the specified length. For example, obtain the rightmost two characters **bc** from **abc**.

Expression: StringUtils.right(value,*2*)

aj. Concatenate the specified character string to the left of the current character string and specify the length of the concatenated character string. If the length of the current character string is not shorter than the specified length, the character string will not be converted. For example, if **yz** is concatenated to the left of **bat** and the length must be 8 after concatenation, the character string is **yzyzybat** after conversion.

Expression: StringUtils.leftPad(value,*8*,"*yz*")

ak. Concatenate the specified character string to the right of the current character string and specify the length of the concatenated character string. If the length of the current character string is not shorter than the specified length, the character string will not be converted. For example, if **yz** is concatenated to the right of **bat** and the length must be 8 after concatenation, the character string is **batyzyzy** after conversion.

Expression: StringUtils.rightPad(value,*8*,"*yz*")

al. If the field is of the string type, obtain the length of the current character string. If the character string is null, **0** is returned.

Expression: StringUtils.length(value)

am. If the field is of the string type, delete all the specified character strings from it. For example, delete **ue** from **queued** to obtain **qd**.

Expression: StringUtils.remove(value,"*ue*")

an. If the field is of the string type, remove the substring at the end of the field. If the specified substring is not at the end of the field, no conversion is performed. For example, remove **.com** at the end of **www.domain.com**.

Expression: StringUtils.removeEnd(value,"*.com*")

ao. If the field is of the string type, delete the substring at the beginning of the field. If the specified substring is not at the beginning of the field, no conversion is performed. For example, delete **www.** at the beginning of **www.domain.com**.

Expression: StringUtils.removeStart(value,"*www.*")

ap. If the field is of the string type, replace all the specified character strings in the field. For example, replace **a** in **aba** with **z** to obtain **zbz**.

Expression: StringUtils.replace(value,"*a*","*z*")

If the content to be replaced contains a special character, the special character must be escaped to a common character. For example, if you want to delete **\t** from a string, use the following expression: StringUtils.replace(value,"\\t",""), which means escaping the backslash (\) again.

aq. If the field is of the string type, replace multiple characters in the character string at a time. For example, replace **h** in **hello** with **j** and **o** with **y** to obtain **jelly**.

Expression: StringUtils.replaceChars(value,"*ho*","*jy*")

ar. If the string starts with the specified prefix (case sensitive), **true** is returned; otherwise, **false** is returned. For example, **abcdef** starts with **abc**, so that **true** is returned.

Expression: StringUtils.startsWith(value,"*abc*")

as. If the field is of the string type, delete all the specified characters at the beginning and end of the field. the field. For example, delete all **x**, **y**, **z**, and **b** from **abcyx** to obtain **abc**.

Expression: StringUtils.strip(value,"*xyz*b")

at. If the field is of the string type, delete all the specified characters at the end of the field, for example, delete the "abc" string at the end of the field.

Expression: StringUtils.stripEnd(value,*"abc"*)

au. If the field is of the string type, delete all the specified characters at the beginning of the field, for example, delete all spaces at the beginning of the field.

Expression: StringUtils.stripStart(value,*null*)

av. If the field is of the string type, obtain the substring after the specified position (the index starts from 0, including the character at the specified position) of the character string. If the specified position is a negative number, calculate the position in the descending order. The first digit at the end is -1. For example, obtain the character whose index is 2 from **abcde** (that is, **c**) and the string after it, that is, **cde**.

Expression: StringUtils.substring(value,*2*)

aw. If the field is of the string type, obtain the substring in a specified range (the index starts from 0, including the character at the start and excluding the character at the end). If the range is a negative number, calculate the position in the descending order. The first digit at the end is -1. For example, obtain the string between the second character (c) and fourth character (e) of **abcde**, that is, **cd**.

Expression: StringUtils.substring(value,*2*,4)

ax. If the field is of the string type, obtain the substring after the first specified character. For example, obtain the substring after the first **b** in **abcba**, that is, **cba**.

Expression: StringUtils.substringAfter(value,"*b*")

ay. If the field is of the string type, obtain the substring after the last specified character. For example, obtain the substring after the last **b** in **abcba**, that is, **a**.

Expression: StringUtils.substringAfterLast(value,"*b*")

az. If the field is of the string type, obtain the substring before the first specified character. For example, obtain the substring before the first **b** in **abcba**, that is, **a**.

Expression: StringUtils.substringBefore(value,"*b*")

ba. If the field is of the string type, obtain the substring before the last specified character. For example, obtain the substring before the last **b** in **abcba**, that is, **abc**.

Expression: StringUtils.substringBeforeLast(value,"*b*")

bb. If the field is of the string type, obtain the substring nested within the specified string. If no substring is found, **null** is returned. For example, obtain the substring between **tag** in **tagabctag**, that is, **abc**.

Expression: StringUtils.substringBetween(value,"*tag*")

bc. If the field is of the string type, delete the control characters (char≤32) at both ends of the character string, for example, delete the spaces at both ends of the character string.

Expression: StringUtils.trim(value)

bd. Convert the character string to a value of the byte type. If the conversion fails, **0** is returned.

Expression: NumberUtils.toByte(value)

be. Convert the character string to a value of the byte type. If the conversion fails, the specified value, for example, **1**, is returned.

Expression: NumberUtils.toByte(value, *1*)

bf. Convert the character string to a value of the double type. If the conversion fails, **0.0d** is returned.

Expression: NumberUtils.toDouble(value)

bg. Convert the character string to a value of the double type. If the conversion fails, the specified value, for example, **1.1d**, is returned.

Expression: NumberUtils.toDouble(value, *1.1d*)

bh. Convert the character string to a value of the float type. If the conversion fails, **0.0f** is returned.

Expression: NumberUtils.toFloat(value)

bi. Convert the character string to a value of the float type. If the conversion fails, the specified value, for example, **1.1f**, is returned.

Expression: NumberUtils.toFloat(value, *1.1f*)

bj. Convert the character string to a value of the int type. If the conversion fails, **0** is returned.

Expression: NumberUtils.toInt(value)

bk. Convert the character string to a value of the int type. If the conversion fails, the specified value, for example, **1**, is returned.

Expression: NumberUtils.toInt(value, *1*)

bl. Convert the character string to a value of the long type. If the conversion fails, **0** is returned.

Expression: NumberUtils.toLong(value)

bm. Convert the character string to a value of the long type. If the conversion fails, the specified value, for example, **1L**, is returned.

Expression: NumberUtils.toLong(value, *1L*)

bn. Convert the character string to a value of the short type. If the conversion fails, **0** is returned.

Expression: NumberUtils.toShort(value)

bo. Convert the character string to a value of the short type. If the conversion fails, the specified value, for example, **1**, is returned.

Expression: NumberUtils.toShort(value, *1*)

bp. Convert the IP string to a value of the long type, for example, convert **10.78.124.0** to **172915712**.

Expression: CommonUtils.ipToLong(value)

bq. Read an IP address and physical address mapping file from the network, and download the mapping file to the map collection. *url* indicates the address for storing the IP mapping file, for example, **http://10.114.205.45:21203/sqoop/IpList.csv**.

Expression: HttpsUtils.downloadMap("*url*")

br. Cache the IP address and physical address mappings and specify a key for retrieval, for example, **ipList**.

Expression:
CommonUtils.setCache("*ipList*",HttpsUtils.downloadMap("*url*"))

bs. Obtain the cached IP address and physical address mappings.

Expression: CommonUtils.getCache("*ipList*")

bt. Check whether the IP address and physical address mappings are cached.

Expression: CommonUtils.cacheExists("*ipList*")

bu. Based on the specified offset type (month/day/hour/minute/second) and offset (positive number indicates increase and negative number indicates decrease), convert the time in the specified format to a new time, for example, add 8 hours to **2019-05-21 12:00:00**.

Expression: DateUtils.getCurrentTimeByZone("*yyyy-MM-dd HH:mm:ss*",value, "*hour*", *8*)

bv. If the value is empty or null, "aaa" is returned. Otherwise, **value** is returned.

Expression: StringUtils.defaultIfEmpty(value,*"aaa"*)

2.3 What Component Versions Are Recommended for Migrating Hadoop Data Sources?

The recommended component versions can be used as both the source and destination.

Table 2-1 Recommended component versions

Hadoop Type	Component	Description
MRS/Apache/FusionInsight HD	Hive	2.*x* versions are not supported. The following versions are recommended: • 1.2.X • 3.1.X
	HDFS	Recommended versions: • 2.8.X • 3.1.X
	HBase	Recommended versions: • 2.1.X • 1.3.X

2.4 What Data Formats Are Supported When the Data Source Is Hive?

Symptom

Which data formats are supported when the data source is Hive?

Solution

CDM can read and write data in SequenceFile, TextFile, ORC, or Parquet format from the Hive data source.

2.5 Can I Synchronize Jobs to Other Clusters?

Symptom

Can I synchronize jobs to other CDM clusters?

Solution

CDM does not support direct job migration across clusters. However, you can use the batch job import and export function to indirectly implement cross-cluster migration as follows:

1. Export all jobs from CDM cluster 1 and save the jobs' JSON files to a local PC.

   For security purposes, no link password is exported when CDM exports jobs. All passwords are replaced by *Add password here*.

2. Edit each JSON file on the local PC by replacing *Add password here* with the actual password of the corresponding link.

3. Import the edited JSON files to CDM cluster 2 in batches to implement job migration between cluster 1 and cluster 2.

2.6 Can I Create Jobs in Batches?

Symptom

Can I create CDM jobs in batches?

Solution

CDM supports batch job creation with the help of the batch import function. You can create jobs in batches as follows:

1. Create a job manually.

2. Export the job and save the job's JSON file to a local PC.

3. Edit the JSON file and replicate more jobs in the JSON file according to the job configuration.

4. Import the JSON file to the CDM cluster to implement batch job creation.

You can also enable automatic job creation based on For Each operators. For details, see **Creating Table Migration Jobs in Batches Using CDM Nodes**.

2.7 Can I Schedule Jobs in Batches?

Symptom

Can I schedule CDM jobs in batches?

Solution

Yes.

1. Access the DataArts Factory module of the DataArts Studio service.
2. In the navigation pane of the DataArts Factory homepage, choose **Data Development** > **Develop Job** to create a job.
3. Drag multiple CDM Job nodes to the canvas and orchestrate the jobs.

2.8 How Do I Back Up CDM Jobs?

Symptom

How do I back up CDM jobs?

Solution

You can use the batch export function of CDM to save all job scripts to a local PC. Then, you can create a cluster and import the jobs again when necessary.

2.9 What Should I Do If Only Some Nodes in a HANA Cluster Can Communicate with the CDM Cluster?

Symptom

What should I do If only some nodes in a HANA cluster can communicate with the CDM cluster?

Solution

To ensure that CDM can communicate with the HANA cluster, perform the following operations:

1. Disable Statement Routing of the HANA cluster. Note that this will increase the pressure on configuration nodes.
2. When creating a HANA link, add the advanced attribute **distribution** and set its value to **off**.

After the preceding configurations are complete, CDM can communicate with the HANA cluster.

2.10 How Do I Use Java to Invoke CDM RESTful APIs to Create Data Migration Jobs?

CDM provides RESTful APIs to implement automatic job creation or execution control by program invocation.

The following describes how to use CDM to migrate data from table **city1** in the MySQL database to table **city2** on DWS, and how to use Java to invoke CDM RESTful APIs to create, start, query, and delete a CDM job.

Prepare the following data in advance:

1. Username, account name, and project ID of the cloud account

2. Create a CDM cluster and obtain the cluster ID.

   On the **Cluster Management** page, click the CDM cluster name to view the cluster ID, for example, **c110beff-0f11-4e75-8b10-da7cd882b0ef**.

3. Create a MySQL database and a DWS database, and create tables **city1** and **city2**. The statements for creating tables are as follows:
   ```
   MySQL:
   create table city1(code varchar(10),name varchar(32));
   insert into city1 values('NY','New York');
   DWS:
   create table city2(code varchar(10),name varchar(32));
   ```

4. In the CDM cluster, create a link to MySQL, such as a link named **mysqltestlink**. Create a link to DWS, such as a link named **dwstestlink**.

5. Run the following code. You are advised to use the HttpClient package of version 4.5. Maven configuration is as follows:
   ```
   <project>
   <modelVersion>4.0.0</modelVersion>
   <groupId>cdm</groupId>
   <artifactId>cdm-client</artifactId>
   <version>1</version>
   <dependencies>
   <dependency>
   <groupId>org.apache.httpcomponents</groupId>
   <artifactId>httpclient</artifactId>
   <version>4.5</version>
   </dependency>
   </dependencies>
   </project>
   ```

Sample Code

The code for using Java to invoke CDM RESTful APIs to create, start, query, and delete a CDM job is as follows:

```
package cdmclient;
import java.io.IOException;
import org.apache.http.Header;
import org.apache.http.HttpEntity;
import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.client.config.RequestConfig;
import org.apache.http.client.methods.CloseableHttpResponse;
import org.apache.http.client.methods.HttpDelete;
import org.apache.http.client.methods.HttpGet;
```

```
import org.apache.http.client.methods.HttpPost;
import org.apache.http.client.methods.HttpPut;
import org.apache.http.entity.StringEntity;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.util.EntityUtils;
public class CdmClient {
private final static String DOMAIN_NAME=" account name";
private final static String USER_NAME=" username";
Private final static String USER_PASSWORD= "Password of the cloud user";
private final static String PROJECT_ID="Project ID";
private final static String CLUSTER_ID="CDM cluster ID";
private final static String JOB_NAME="Job name";
private final static String FROM_LINKNAME="Source link name";
private final static String TO_LINKNAME="Destination link name";
Private final static String IAM_ENDPOINT= "IAM endpoint";
Private final static String CDM_ENDPOINT= "CDM endpoint";
private CloseableHttpClient httpclient;
private String token;

public CdmClient() {
this.httpclient = createHttpClient();
this.token = login();
}

private CloseableHttpClient createHttpClient() {
CloseableHttpClient httpclient =HttpClients.createDefault();
return httpclient;
}

private String login(){
HttpPost httpPost = new HttpPost("https://"+IAM_ENDPOINT+"/v3/auth/tokens");
String json =
"{\r\n"+
"\"auth\": {\r\n"+
"\"identity\": {\r\n"+
"\"methods\": [\"password\"],\r\n"+
"\"password\": {\r\n"+
"\"user\": {\r\n"+
"\"name\": \""+USER_NAME+"\",\r\n"+
"\"password\": \""+USER_PASSWORD+"\",\r\n"+
"\"domain\": {\r\n"+
"\"name\": \""+DOMAIN_NAME+"\"\r\n"+
"}\r\n"+
"}\r\n"+
"}\r\n"+
"},\r\n"+
"\"scope\": {\r\n"+
"\"project\": {\r\n"+
"\"name\": \"PROJECT_NAME\"\r\n"+
"}\r\n"+
"}\r\n"+
"}\r\n"+
"}\r\n";
try {
StringEntity s = new StringEntity(json);
s.setContentEncoding("UTF-8");
s.setContentType("application/json");
httpPost.setEntity(s);
CloseableHttpResponse response = httpclient.execute(httpPost);
Header tokenHeader = response.getFirstHeader("X-Subject-Token");
String token = tokenHeader.getValue();
System.out.println("Login successful");
return token;
} catch (Exception e) {
throw new RuntimeException("login failed.", e);
}
}
```

```
/*Create a job.*/

public void createJob(){
HttpPost httpPost = new HttpPost("https://"+CDM_ENDPOINT+"/cdm/v1.0/"+PROJECT_ID+"/
clusters/"+CLUSTER_ID+"/cdm/job");
```

/**The JSON information here is complex. You can create a job on the job management page, click **Job JSON Definition** next to the job, copy the JSON content and convert it into a Java character string, and paste it here.
*In the JSON message body, you only need to replace the link name, data import and export table names, field list of the tables, and fields used for partitioning in the source table.**/

```
String json =
"{\r\n"+
"\"jobs\": [\r\n"+
"{\r\n"+
"\"from-connector-name\": \"generic-jdbc-connector\",\r\n"+
"\"name\": \""+JOB_NAME+"\",\r\n"+
"\"to-connector-name\": \"generic-jdbc-connector\",\r\n"+
"\"driver-config-values\": {\r\n"+
"\"configs\": [\r\n"+
"{\r\n"+
"\"inputs\": [\r\n"+
"{\r\n"+
"\"name\": \"throttlingConfig.numExtractors\",\r\n"+
"\"value\": \"1\"\r\n"+
"}\r\n"+
"],\r\n"+
"\"validators\": [],\r\n"+
"\"type\": \"JOB\",\r\n"+
"\"id\": 30,\r\n"+
"\"name\": \"throttlingConfig\"\r\n"+
"}\r\n"+
"]\r\n"+
"},\r\n"+
"\"from-link-name\": \""+FROM_LINKNAME+"\",\r\n"+
"\"from-config-values\": {\r\n"+
"\"configs\": [\r\n"+
"{\r\n"+
"\"inputs\": [\r\n"+
"{\r\n"+
"\"name\": \"fromJobConfig.schemaName\",\r\n"+
"\"value\": \"sqoop\"\r\n"+
"},\r\n"+
"{\r\n"+
"\"name\": \"fromJobConfig.tableName\",\r\n"+
"\"value\": \"city1\"\r\n"+
"},\r\n"+
"{\r\n"+
"\"name\": \"fromJobConfig.columnList\",\r\n"+
"\"value\": \"code&name\"\r\n"+
"},\r\n"+
"{\r\n"+
"\"name\": \"fromJobConfig.partitionColumn\",\r\n"+
"\"value\": \"code\"\r\n"+
"}\r\n"+
"],\r\n"+
"\"validators\": [],\r\n"+
"\"type\": \"JOB\",\r\n"+
"\"id\": 7,\r\n"+
"\"name\": \"fromJobConfig\"\r\n"+
"}\r\n"+
"]\r\n"+
"},\r\n"+
"\"to-link-name\": \""+TO_LINKNAME+"\",\r\n"+
"\"to-config-values\": {\r\n"+
"\"configs\": [\r\n"+
"{\r\n"+
"\"inputs\": [\r\n"+
```

```
"{\r\n"+
"\"name\": \"toJobConfig.schemaName\",\r\n"+
"\"value\": \"sqoop\"\r\n"+
"},\r\n"+
"{\r\n"+
"\"name\": \"toJobConfig.tableName\",\r\n"+
"\"value\": \"city2\"\r\n"+
"},\r\n"+
"{\r\n"+
"\"name\": \"toJobConfig.columnList\",\r\n"+
"\"value\": \"code&name\"\r\n"+
"}, \r\n"+
"{\r\n"+
"\"name\": \"toJobConfig.shouldClearTable\",\r\n"+
"\"value\": \"true\"\r\n"+
"}\r\n"+
"],\r\n"+
"\"validators\": [],\r\n"+
"\"type\": \"JOB\",\r\n"+
"\"id\": 9,\r\n"+
"\"name\": \"toJobConfig\"\r\n"+
"}\r\n"+
"]\r\n"+
"}\r\n"+
"}\r\n"+
"]\r\n"+
"}\r\n";
try {
StringEntity s = new StringEntity(json);
s.setContentEncoding("UTF-8");
s.setContentType("application/json");
httpPost.setEntity(s);
httpPost.addHeader("X-Auth-Token", this.token);
httpPost.addHeader("X-Language", "en-us");
CloseableHttpResponse response = httpclient.execute(httpPost);
int status = response.getStatusLine().getStatusCode();
if(status == 200){
System.out.println("Create job successful.");
}else{
System.out.println("Create job failed.");
HttpEntity entity = response.getEntity();
System.out.println(EntityUtils.toString(entity));
}
} catch (Exception e) {
e.printStackTrace();
throw new RuntimeException("Create job failed.", e);
}
}
/*Start the job.*/

public void startJob(){
HttpPut httpPut = new HttpPut("https://"+CDM_ENDPOINT+"/cdm/v1.0/"+PROJECT_ID+"/
clusters/"+CLUSTER_ID+"/cdm/job/"+JOB_NAME+"/start");
String json = "";
try {
StringEntity s = new StringEntity(json);
s.setContentEncoding("UTF-8");
s.setContentType("application/json");
httpPut.setEntity(s);
httpPut.addHeader("X-Auth-Token", this.token);
httpPut.addHeader("X-Language", "en-us");
CloseableHttpResponse response = httpclient.execute(httpPut);
int status = response.getStatusLine().getStatusCode();
if(status == 200){
System.out.println("Start job successful.");
}else{
System.out.println("Start job failed.");
HttpEntity entity = response.getEntity();
System.out.println(EntityUtils.toString(entity));
```

```
}
} catch (Exception e) {
e.printStackTrace();
throw new RuntimeException("Start job failed.", e);
}
}
/*Query the job running status cyclically until the job is complete.*/

public void getJobStatus(){
HttpGet httpGet = new HttpGet("https://"+CDM_ENDPOINT+"/cdm/v1.0/"+PROJECT_ID+"/
clusters/"+CLUSTER_ID+"/cdm/job/"+JOB_NAME+"/status");
try {
httpGet.addHeader("X-Auth-Token", this.token);
httpGet.addHeader("X-Language", "en-us");
boolean flag = true;
while(flag){
CloseableHttpResponse response = httpclient.execute(httpGet);
int status = response.getStatusLine().getStatusCode();
if(status == 200){
HttpEntity entity = response.getEntity();
String msg = EntityUtils.toString(entity);
if(msg.contains("\"status\":\"SUCCEEDED\"")){
System.out.println("Job succeeded");
break;
}else if (msg.contains("\"status\":\"FAILED\"")){
System.out.println("Job failed.");
break;
}else{
Thread.sleep(1000);
}

}else{
System.out.println("Get job status failed.");
HttpEntity entity = response.getEntity();
System.out.println(EntityUtils.toString(entity));
break;
}
}
} catch (Exception e) {
e.printStackTrace();
throw new RuntimeException("Get job status failed.", e);
}
}
/*Delete the job.*/

public void deleteJob(){
HttpDelete httpDelte = new HttpDelete("https://"+CDM_ENDPOINT+"/cdm/v1.0/"+PROJECT_ID+"/
clusters/"+CLUSTER_ID+"/cdm/job/"+JOB_NAME);
try {
httpDelte.addHeader("X-Auth-Token", this.token);
httpDelte.addHeader("X-Language", "en-us");
CloseableHttpResponse response = httpclient.execute(httpDelte);
int status = response.getStatusLine().getStatusCode();
if(status == 200){
System.out.println("Delete job successful.");
}else{
System.out.println("Delete job failed.");
HttpEntity entity = response.getEntity();
System.out.println(EntityUtils.toString(entity));
}
} catch (Exception e) {
e.printStackTrace();
throw new RuntimeException("Delete job failed.", e);
}
}
/*Close the process.*/

public void close(){
try {
```

```
httpclient.close();
} catch (IOException e) {
throw new RuntimeException("Close failed.", e);
}
}

public static void main(String[] args){
CdmClient cdmClient = new CdmClient();
cdmClient.createJob();
cdmClient.startJob();
cdmClient.getJobStatus();
cdmClient.deleteJob();
cdmClient.close();
}
}
```

2.11 How Do I Connect the On-Premises Intranet or Third-Party Private Network to CDM?

Many enterprises deploy key data sources on the intranet, such as databases and file servers. CDM runs on the cloud. To migrate the intranet data to the cloud using CDM, use any of the following methods to connect the intranet to the cloud:

- If the destination data source is an on-premises database, you need the Internet or Direct Connect. When using the Internet, ensure that an EIP has been bound to the CDM cluster, the security group of CDM allows outbound traffic from the host where the off-cloud data source is located, the host where the data source is located can access the Internet, and the connection port has been enabled in the firewall rules.
- Establish a VPN between the on-premises data center and the VPC where the service resides.
- Leverage Network Address Translation (NAT) or port forwarding to access the network in proxy mode.

The following describes how to use the port forwarding tool to access intranet data. The process is as follows:

1. Use a Windows computer as the gateway. The computer must be able to access both the Internet and the intranet.
2. Install the port mapping tool IPOP on the computer.
3. Configure port mapping using the tool.

NOTICE

If the intranet database is exposed to the public network for a long time, security risks exist. Therefore, after data migration is complete, stop port mapping.

Scenario

Suppose that the MySQL database on the intranet is migrated to DWS. **Figure 2-2** shows the network topology.

In the figure, the intranet can be either an enterprise's data center or the intranet of the virtual data center on a third-party cloud.

Figure 2-2 Network topology example
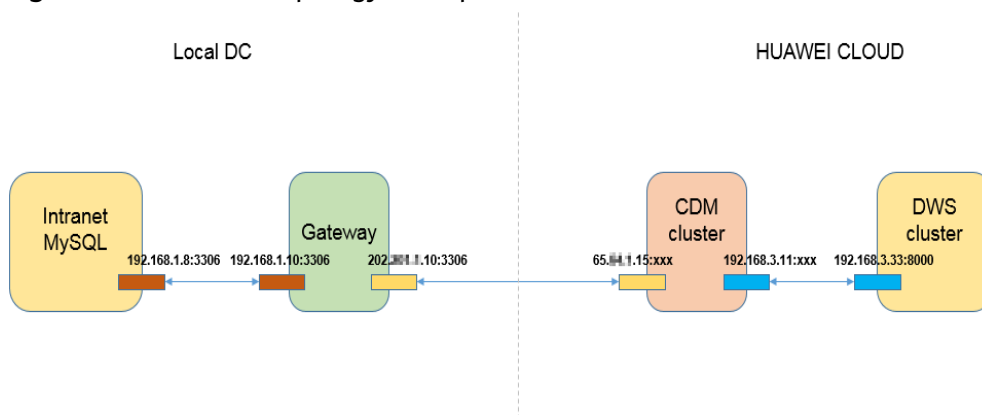
Procedure

Step 1 Use a Windows computer as the gateway. Configure both the intranet and Internet IP addresses on the computer. Conduct the following test to check whether the gateway computer can fulfill service needs.

1. Run the **ping** command on the computer to check whether the intranet address of the MySQL database is pingable. For example, run **ping 192.168.1.8**.

2. Run the **ping** command on another computer that can access the Internet to check whether the public network address of the gateway computer is pingable. For example, run **ping 202.**xx.xx**.10**.

Step 2 Download the port mapping tool IPOP and install it on the gateway computer.

Step 3 Run the port mapping tool and select **PORT Map**. See **Figure 2-3**.

- **Local IP** and **Local Port**: Configure these two parameters to the public network address and port number of the gateway computer respectively, which must be entered when creating MySQL links on CDM.

- **Mapping IP** and **Map Port**: Configure these two parameters to the IP address and port number of the MySQL database on the intranet.

Figure 2-3 Configuring port mapping
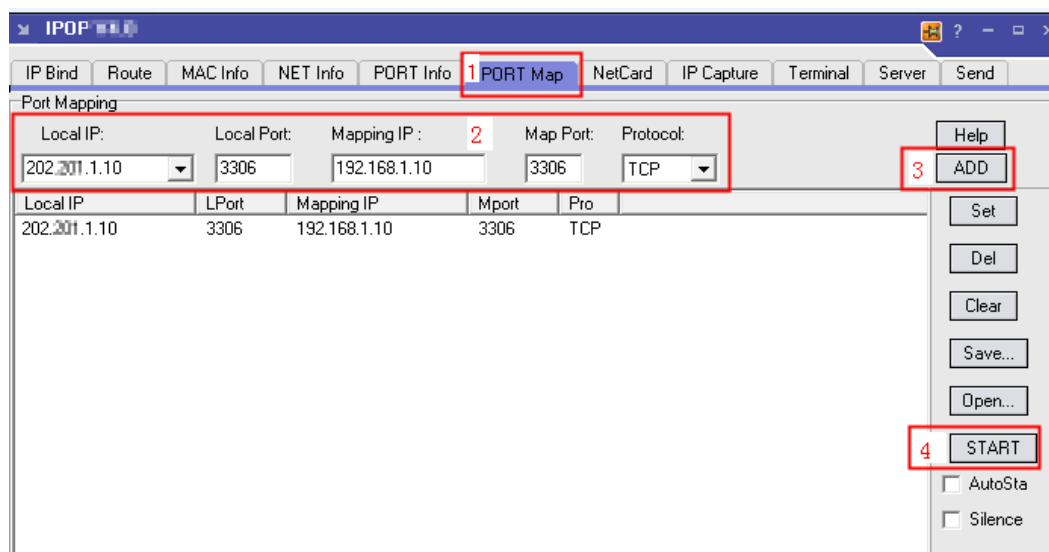
Step 4 Click **ADD** to add a port mapping relationship.

Step 5 Click **START** to start mapping and receive data packets.

Then, you can use the EIP to read data from the MySQL database on the intranet on CDM and import the data to DWS.

📖 **NOTE**

1. To access the on-premises data source, you must also bind an EIP to the CDM cluster.

2. Generally, DWS is accessible within the same VPC. When creating a CDM cluster, you must ensure that the VPC of the CDM cluster must be the same as that of DWS. In addition, it is recommended that CDM and DWS be in the same intranet and security group. If their security groups are different, you also need to enable data access between the security groups.

3. Port mapping can be used to migrate data between databases on the intranet or the SFTP servers.

4. For Linux computers, port mapping can also be implemented using IPTABLE.

5. When the FTP server on the intranet is mapped to the public network using port mapping, you need to check whether the PASV mode is enabled. In this case, the client and server are connected through a random port. Therefore, in addition to port 21 mapping, you also need to configure the port range mapping in PASV mode. For example, you can specify the **vsftp** port range by configuring **pasv_min_port** and **pasv_max_port**.

----End

2.12 Does CDM Support Parameters or Variables?

Symptom

Does CDM support parameters or variables?

Solution

Yes.

If a CDM job uses the **job parameters** or **variables** configured during data development, data can be indirectly migrated based on the parameters or variables during node scheduling in the DataArts Factory module.

2.13 How Do I Set the Number of Concurrent Extractors for a CDM Migration Job?

CDM migrates data through data migration jobs. It works in the following way:

1. When data migration jobs are submitted, CDM splits each job into multiple tasks based on the **Concurrent Extractors** parameter in the job configuration.

📖 **NOTE**

Jobs for different data sources may be split based on different dimensions. Some jobs may not be split based on the **Concurrent Extractors** parameter.

2. CDM submits the tasks to the running pool in sequence. Tasks (defined by **Maximum Concurrent Extractors**) run concurrently. Excess tasks are queued.
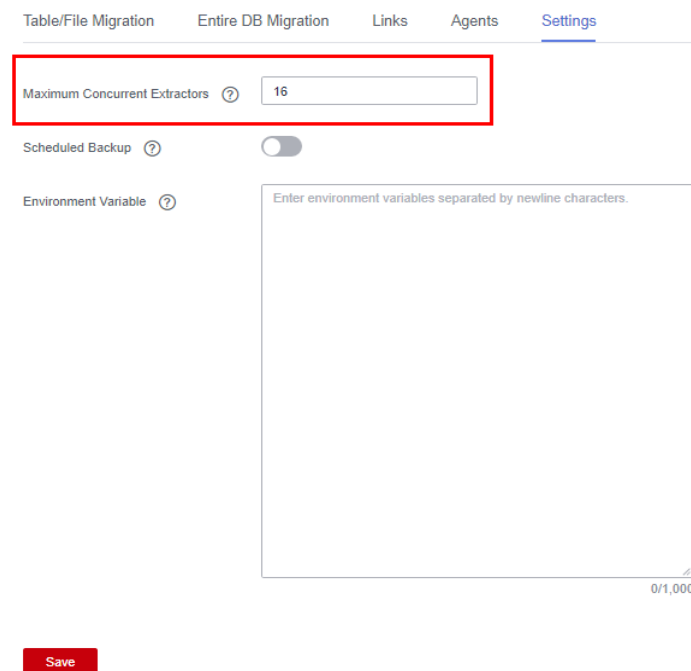
Changing Concurrent Extractors

1. The maximum number of concurrent extractors for a cluster varies depending on the CDM cluster flavor. You are advised to set the maximum number of concurrent extractors to twice the number of vCPUs of the CDM cluster.

Table 2-2 Maximum number of concurrent extractors for a CDM cluster

Flavor	vCPUs/Memory	Maximum Concurrent Extractors
cdm.large	8 vCPUs, 16 GB	16
cdm.xlarge	16 vCPUs, 32 GB	32
cdm.4xlarge	64 vCPUs, 128 GB	128

Figure 2-4 Setting Maximum Concurrent Extractors for a CDM cluster

2. Configure the number of concurrent extractors based on the following rules:

   a. When data is to be migrated to files, CDM does not support multiple concurrent tasks. In this case, set a single process to extract data.

   b. If each row of the table contains less than or equal to 1 MB data, data can be extracted concurrently. If each row contains more than 1 MB data, it is recommended that data be extracted in a single thread.

   c. Set **Concurrent Extractors** for a job based on **Maximum Concurrent Extractors** for the cluster. It is recommended that **Concurrent Extractors** is less than **Maximum Concurrent Extractors**.

d. If the destination is DLI, you are advised to set the number of concurrent extractors to 1. Otherwise, data may fail to be written.

Figure 2-5 Setting Concurrent Extractors for a job

2.14 Does CDM Support Real-Time Migration of Dynamic Data?

Symptom

Does CDM support real-time migration of dynamic data?

Solution

No.

If data is written to the source during the migration, an error may occur.

2.15 Can I Stop CDM Clusters?

Symptom

Can I stop CDM clusters?

Solution

No.

No. From April 2022 on, CDM clusters cannot be stopped. If a CDM cluster is stopped, its resources may be occupied. As a result, the cluster cannot be started again.

2.16 How Do I Obtain the Current Time Using an Expression?

Symptom

How do I obtain the current time using an expression?

Solution

You can use the **DateUtils.format(${timestamp()},"yyyy-MM-dd HH:mm:ss")** expression on the **Map Field** page to obtain the current time. For details, see **Field Conversion**.

2.17 What Is the Time Format for the Where Clause Parameters for Creating a Migration Job?

Symptom

What is the time format for the where clause parameters for creating a migration job?

Solution

For details, see the syntax for MySQL date and time types at https://dev.mysql.com/doc/refman/8.0/en/datetime.html.

2.18 Can CDM Migrate Field Comments from a Source Table to a Destination Table?

Symptom

Can CDM migrate field comments from a source table to a destination table?

Solution

It depends on the version.

Details are as follows:

This function is supported in the 2.8.6.1 and 2.9.2.1 versions, and not supported in the 2.9.1 version.

3 Troubleshooting

3.1 What Should I Do If the Log Prompts that the Date Format Fails to Be Parsed?

Symptom

When CDM is used to migrate other data sources to CSS, the job fails to be executed and the error message "Unparseable date" is displayed in the log. See **Figure 3-1**.

Figure 3-1 Log output

```
java.text.ParseException: Unparseable date: "2018/01/05 15:15:46"
        at java.text.DateFormat.parse(DateFormat.java:366) ~[na:1.8.0_112]
        at org.apache.sqoop.connector.common.DataTypeUtil.convertDateFormat
        at org.apache.sqoop.connector.elasticsearch.ElasticSearchLoader.toJ
        at org.apache.sqoop.connector.elasticsearch.ElasticSearchLoader.arr
7]
        at org.apache.sqoop.connector.elasticsearch.ElasticSearchLoader.loa
```

Possible Cause

CSS has a special processing mechanism on the time field. If the stored time data does not contain the time zone information, Kibana considers the time as the GMT.

The time displayed in the log may be different from the local time. For example, the time displayed in the log is eight hours earlier than the local time in the GMT +08:00 time zone. Therefore, when CDM migrates data to Cloud Search Service, if the index and type are automatically created by CDM (for example, if **date_test** and **test1** of the migration destination highlighted in **Figure 3-2** do not exist in Cloud Search Service, CDM automatically creates the index and type in Cloud Search Service), CDM, by default, sets the format of the time field to the standard format of *yyyy-MM-dd HH:mm:ss.SSS Z*, for example, **2018-01-08 08:08:08.666 +0800**.

Figure 3-2 Job configuration

When data is imported from another data source to CSS, if the date format in the source data is not the standard format, for example, **2018/01/05 15:15:46**, the CDM job fails to be executed, and the log shows that the date format cannot be parsed. You need to configure a field converter on CDM to convert the format of the date field to the required format of CSS.

Solution

1. Edit the job and go to the **Map Field** tab page. Click the icon for creating a converter in the row of the source field to create a converter. See **Figure 3-3**.

Figure 3-3 Creating a converter


2. Select **Expression conversion** as the converter. Currently, expression conversion supports functions of the character string and date types. The syntax is similar to the Java character string and time functions. For details about how to compile the expression, see **Expression Conversion**.

3. In this example, the source time format is *yyyy/MM/dd HH:mm:ss*. To convert the source time format to *yyyy-MM-dd HH:mm:ss.SSS Z*, perform the following operations:

a.   Add the time zone information **+0800** to the end of the original date character string. The corresponding expression is **value+" +0800"**.

b.   Use the original date format to parse the string to a date object. You can use the DateUtils.parseDate function for parsing. The syntax is **DateUtils.parseDate(String value, String format)**.

c.   Format the date object into a character string in target format by using the DateUtils.format function. The syntax is **DateUtils.format(Date date, String format)**.

In this example, the complete expression is **DateUtils.format(DateUtils.parseDate(value+" +0800","yyyy/MM/dd HH:mm:ss Z"),"yyyy-MM-dd HH:mm:ss.SSS Z")**. See **Figure 3-4**.

Figure 3-4 Configuring the expression
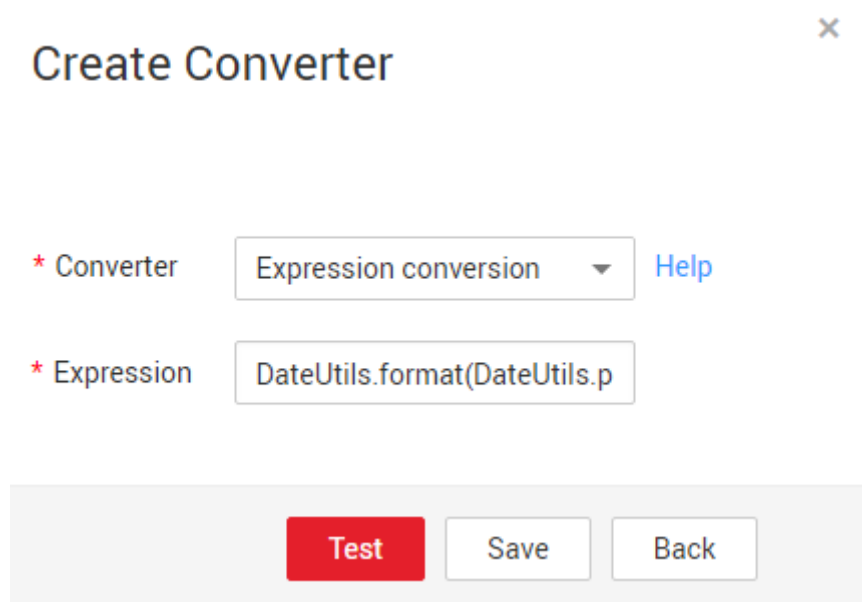
4.   Save the converter configuration and save and run the job to solve the problem that Cloud Search Service fails to parse the date format.

3.2 What Can I Do If the Map Field Tab Page Cannot Display All Columns?

Symptom

When data is exported from HBase/CloudTable using CDM, fields in the HBase/CloudTable table on the **Map Field** tab page occasionally cannot be displayed completely and cannot match the fields on the migration destination. As a result, the data imported to the migration destination is incomplete.

Possible Cause

HBase/CloudTable are schema-less, and the number of columns in each data is not fixed. On the **Map Field** page, there is a high probability that all columns

cannot be obtained by obtaining example values. In this case, the data on the migration destination is incomplete after the job is executed.

To solve this problem, perform any of the following methods:

1. Add fields on the **Map Field** tab page.

2. Edit the JSON file of the job on the **Job Management** page (modify the **fromJobConfig.columns** and **toJobConfig.columnList** parameters).

3. Export the JSON file of the job to the local PC, modify the parameters in the JSON file (the principle is the same to that in **2**), and then import the JSON file back to CDM.

You are advised to perform **1**. The following uses data migration from HBase to DWS as an example.

Solution 1: Adding Fields on the Map Field Tab Page

1. Obtain all fields in the tables to be migrated from source HBase. Use colons (:) to separate column families and columns. The following gives an example:
   ```
   rowkey:rowkey
   g:DAY_COUNT
   g:CATEGORY_ID
   g:CATEGORY_NAME
   g:FIND_TIME
   g:UPLOAD_PEOPLE
   g:ID
   g:INFOMATION_ID
   g:TITLE
   g:COORDINATE_X
   g:COORDINATE_Y
   g:COORDINATE_Z
   g:CONTENT
   g:IMAGES
   g:STATE
   ```

2. On the **Job Management** page, locate the job for exporting data from HBase to DWS, click **Edit** in the row where the job resides, and go to the **Map Field** tab page.

Figure 3-5 Field mapping 03

3. Click ⊕. In the dialog box that is displayed, select **Add a new field**.

Figure 3-6 Adding a field 04

  NOTE

- After a field is added, the example value of the new field is not displayed on the console. This does not affect the transmission of field values. CDM directly writes the field values to the migration destination.
- To add new fields, the migration source must be MongoDB, HBase, relational databases, or Redis (data in Redis must be in the Hash format).

4. After all fields are added, check whether the mapping between the migration source and destination is correct. If the mapping is incorrect, drag the fields to adjust the field mapping.

5. Click **Next** and **Save**.

Solution 2: Modifying a JSON File

1. Obtain all fields in the tables to be migrated from source HBase. Use colons (:) to separate column families and columns. The following gives an example:
   ```
   rowkey:rowkey
   g:DAY_COUNT
   g:CATEGORY_ID
   g:CATEGORY_NAME
   g:FIND_TIME
   g:UPLOAD_PEOPLE
   g:ID
   g:INFOMATION_ID
   g:TITLE
   g:COORDINATE_X
   g:COORDINATE_Y
   g:COORDINATE_Z
   g:CONTENT
   g:IMAGES
   g:STATE
   ```

2. In the DWS destination table, obtain the fields corresponding to the HBase table fields.

   If any field name corresponding to the HBase field does not exist in the DWS destination table, add it to the DWS table schema. Suppose that the fields in the DWS table are complete and are displayed as follows:

```
rowkey
day_count
category
category_name
find_time
upload_people
id
information_id
title
coordinate_x
coordinate_y
coordinate_z
content
images
state
```

3. On the **Job Management** page, locate the job for exporting data from HBase to DWS, and choose **More** > **Edit Job JSON** in the row where the job resides.

4. On the page that is displayed, edit the JSON file of the job.

   a. Modify the **fromJobConfig.columns** parameter of the migration source to the HBase fields obtained in **1**. Use & to separate column numbers and colons (:) to separate column families and columns. The following gives an example:

   ```
   "from-config-values": {
           "configs": [
               {
                   "inputs": [
                       {
                           "name": "fromJobConfig.table",
                           "value": "HBase"
                       },
                       {
                           "name": "fromJobConfig.columns",
                           "value":
   "rowkey:rowkey&g:DAY_COUNT&g:CATEGORY_ID&g:CATEGORY_NAME&g:FIND_TIME&g:UP
   LOAD_PEOPLE&g:ID&g:INFOMATION_ID&g:TITLE&g:COORDINATE_X&g:COORDINATE_Y&g:
   COORDINATE_Z&g:CONTENT&g:IMAGES&g:STATE"
                       },
                       {
                           "name": "fromJobConfig.formats",
                           "value": {
                               "2": "yyyy-MM-dd",
                               "undefined": "yyyy-MM-dd"
                           }
                       }
                   ],
                   "name": "fromJobConfig"
               }
           ]
       }
   ```

   b. Modify the **toJobConfig.columnList** parameter of the migration source to the field list of DWS obtained in **2**.

   The sequence must be the same as that of HBase to ensure correct field mapping. Use & to separate field names. The following gives an example:

   ```
   "to-config-values": {
           "configs": [
               {
                   "inputs": [
                       {
                           "name": "toJobConfig.schemaName",
                           "value": "dbadmin"
                       },
                       {
                           "name": "toJobConfig.tablePreparation",
                           "value": "DO_NOTHING"
   ```

```
        },
        {
            "name": "toJobConfig.tableName",
            "value": "DWS "
        },
        {
            "name": "toJobConfig.columnList",
            "value":
"rowkey&day_count&category&category_name&find_time&upload_people&id&information
_id&title&coordinate_x&coordinate_y&coordinate_z&content&images&state"
        },
        {
            "name": "toJobConfig.shouldClearTable",
            "value": "true"
        }
    ],
    "name": "toJobConfig"
}
]
}
```

c.   Retain the settings of other parameters, and then click **Save and Run**.

5.   After the job is completed, check whether the data in the DWS table matches the data in HBase. If the mapping is incorrect, check whether the sequences of the HBase and DWS fields in the JSON file are the same.

3.3 How Do I Select Distribution Columns When Using CDM to Migrate Data to GaussDB(DWS)?

Symptom

How do I select distribution columns when using CDM to migrate data to GaussDB(DWS)?

Solution

When using CDM to migrate data to DWS or FusionInsight LibrA and create a table on DWS, select the distribution columns on the **Map Field** tab page.

Figure 3-7 Selecting distribution columns
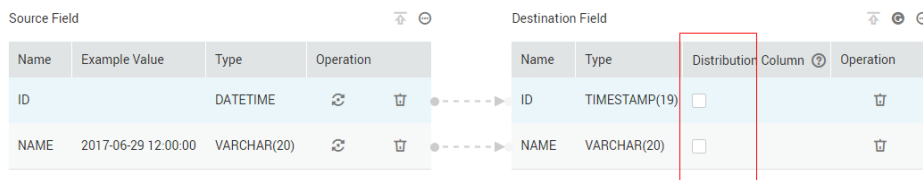
Selecting the distribution column is very important for the running of DWS/ FusionInsight LibrA. When migrating data to DWS/FusionInsight LibrA, you are advised to specify the distribution column according to the following principles:

1.   Use the primary key as the distribution column.

2.   If multiple data segments are combined as primary keys, specify all primary keys as the distribution column.

3. In the scenario where no primary key is available, if no distribution column is selected, DWS uses the first column as the distribution column by default. As a result, data skew risks exist.

Therefore, when a single table or entire database is imported to DWS/FusionInsight LibrA, you are advised to manually select a distribution column; otherwise, CDM automatically selects one. For more information about distribution columns, see **GaussDB(DWS)**.

If the DWS primary key or table contains only one field, the field type must be a common character string, value, or date. When data is migrated from another database to DWS, if automatic table creation is selected, the primary key must be of the following types. If no primary key is set, at least one of the following fields must be set. Otherwise, the table cannot be created and the CDM job fails.

- INTEGER TYPES: TINYINT, SMALLINT, INT, BIGINT, NUMERIC/DECIMAL
- CHARACTER TYPES: CHAR, BPCHAR, VARCHAR, VARCHAR2, NVARCHAR2, TEXT
- DATA/TIME TYPES: DATE, TIME, TIMETZ, TIMESTAMP, TIMESTAMPTZ, INTERVAL, SMALLDATETIME

3.4 What Do I Do If the Error Message "value too long for type character varying" Is Displayed When I Migrate Data to DWS?

Symptom

When you use CDM to migrate data to DWS/FusionInsight LibrA, the migration fails and the error message "**value too long for type character varying**" is displayed in the log. See **Figure 3-8**.

Figure 3-8 Log output


```
Caused by: org.postgresql.util.PSQLException: ERROR: value too long for type character varying(50)
    Where: COPY fl_behavior_module, line 72, column MODULE_NAME: "████████-████████"
        at org.postgresql.core.v3.QueryExecutorImpl.receiveErrorResponse(QueryExecutorImpl.java:2477)
        at org.postgresql.core.v3.QueryExecutorImpl.processCopyResults(QueryExecutorImpl.java:1107)
        at org.postgresql.core.v3.QueryExecutorImpl.writeToCopy(QueryExecutorImpl.java:989)
        at org.postgresql.core.v3.CopyInImpl.writeToCopy(CopyInImpl.java:35)
        ... 16 common frames omitted
```

Possible Cause

The data migrated to DWS is in Chinese, and the table is automatically created at the migration destination. The length of the varchar field of DWS is calculated by byte, and a Chinese character may occupy three bytes in UTF-8 encoding. If the length of a Chinese character exceeds that of the varchar field of DWS, an error occurs and the error message "**value too long for type character varying**" is displayed.

Solution

To solve this problem, you can select **Extend Field Length** to **Yes**, so that the length of the **varchar** field is automatically increased by three times when the destination table is created.

Edit the table/file migration job on CDM. In **Destination Job Configuration**, set **Auto Table Creation** to **Auto creation**, **Extend Field Length** is displayed in **Show Advanced Attributes**. Set **Extend Field Length** to **Yes**.

Figure 3-9 Extending field length

3.5 What Can I Do If Error Message "Unable to execute the SQL statement" Is Displayed When I Import Data from OBS to SQL Server?

Symptom

When CDM is used to import data from OBS to SQL Server, the job fails to be executed and error message "Unable to execute the SQL statement. Cause: "String or binary data truncated" is displayed.

Possible Cause

The data in OBS exceeds the length limit of the SQL Server database.

Solution

When creating a table in the SQL Server database, increase the length of the database field. The length of the database field must be greater than that of the data in OBS.

3.6 What Should I Do If the Cluster List Is Empty, I Have No Access Permission, or My Operation Is Denied?

Symptom

When using CDM, you may encounter the following permission-related issues:

- The cluster list is empty on the CDM homepage.
- A message is displayed, indicating that you do not have the access permission.
- When you try to start a job or restart a cluster, an error message is displayed, indicating that your operation is denied by the current policy.

Possible Cause

The preceding issues are caused by incorrect permission configuration.

Solution

- If CDM is a module of DataArts Studio, perform the following operations:
  a. Check whether the DAYU Administrator or DAYU User role has been added. For details, see **DataArts Studio Permissions Management**.
  b. Check whether you have the permission (developer or viewer role) to access the workspace. For details, see **DataArts Studio Permissions**.
- If CDM is an independent service, perform the following operations:

a. Check whether IAM fine-grained authentication is enabled.

- If it is disabled, check whether the **CDM Administrator** role has been added to the user group.

- If it is enabled, go to **2**.

b. Check whether a custom or preset policy has been added to enable you to access CDM, such as the **CDM FullAccess** and **CDM ReadOnlyAccess** policies. For details, see **Permissions Management**.

c. Check whether an access denial policy has been added to the enterprise project.

3.7 Why Is Error ORA-01555 Reported During Migration from Oracle to DWS?

Symptom

When CDM is used to migrate Oracle data to DWS, an error is reported, as shown in **Figure 3-10**.

Figure 3-10 Symptom


```
665  2020-09-21 22:51:02,991 ERROR LocalJobRunner Map Task #3 [org.apache.sqoop.common.SqoopException:111] SqoopException
666  java.sql.SQLException: ORA-01555: snapshot too old: rollback segment number 3 with name "_SYSSMU3_2097677531$" too small
667
668      at oracle.jdbc.driver.T4CTTIoer11.processERROR(T4CTTIoer11.java:494)
669      at oracle.jdbc.driver.T4CTTIoer11.processERROR(T4CTTIoer11.java:446)
670      at oracle.jdbc.driver.T4C8Oall.processERROR(T4C8Oall.java:1054)
671      at oracle.jdbc.driver.T4CTTIfun.receive(T4CTTIfun.java:623)
672      at oracle.jdbc.driver.T4CTTIfun.doRPC(T4CTTIfun.java:252)
673      at oracle.jdbc.driver.T4C8Oall.doOALL(T4C8Oall.java:612)
674      at oracle.jdbc.driver.T4CPreparedStatement.doOall8(T4CPreparedStatement.java:226)
675      at oracle.jdbc.driver.T4CPreparedStatement.fetch(T4CPreparedStatement.java:1023)
676      at oracle.jdbc.driver.OracleStatement.fetchMoreRows(OracleStatement.java:3353)
677      at oracle.jdbc.driver.InsensitiveScrollableResultSet.fetchMoreRows(InsensitiveScrollableResultSet.java:736)
678      at oracle.jdbc.driver.InsensitiveScrollableResultSet.absoluteInternal(InsensitiveScrollableResultSet.java:692)
679      at oracle.jdbc.driver.InsensitiveScrollableResultSet.next(InsensitiveScrollableResultSet.java:406)
680      at org.apache.sqoop.connector.jdbc.sql.impl.WrapResultSet.next(WrapResultSet.java:36)
681      at org.apache.sqoop.connector.jdbc.GenericJdbcExtractor.extractObjectRecord(GenericJdbcExtractor.java:151)
682      at org.apache.sqoop.connector.jdbc.GenericJdbcExtractor.extract(GenericJdbcExtractor.java:129)
683      at org.apache.sqoop.connector.jdbc.GenericJdbcExtractor.extract(GenericJdbcExtractor.java:59)
684      at org.apache.sqoop.job.mr.SqoopMapper.runInternal(SqoopMapper.java:184)
685      at org.apache.sqoop.job.mr.SqoopMapper.run(SqoopMapper.java:81)
686      at org.apache.hadoop.mapred.MapTask.runNewMapper(MapTask.java:799)
687      at org.apache.hadoop.mapred.MapTask.run(MapTask.java)
688      at org.apache.hadoop.mapred.LocalJobRunner$Job$MapTaskRunnable.run(LocalJobRunner.java:271)
689      at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
690      at java.util.concurrent.FutureTask.run(FutureTask.java:266)
691      at org.apache.sqoop.submission.mapreduce.MapperExecutorGroup$1.lambda$execute$0(MapperExecutorGroup.java:222)
692      at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
693      at java.util.concurrent.FutureTask.run(FutureTask.java:266)
694      at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
695      at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
696      at java.lang.Thread.run(Thread.java:748)
697  Caused by: oracle.jdbc.OracleDatabaseException: ORA-01555: snapshot too old: rollback segment number 3 with name "_SYSSMU3_2097677531$" too small
698
699      at oracle.jdbc.driver.T4CTTIoer11.processERROR(T4CTTIoer11.java:498)
700      ... 28 common frames omitted
```

Cause Analysis

1. During data migration, if the entire table is queried and the table contains a large amount of data, the query takes a long time.

2. During the query, other users frequently perform the **commit** operation.

3. The RBS (the tablespace used for rollback) of Oracle is small. As a result, the migration task is not complete, the source database has been updated, and the rollback times out.

Summary and Suggestions

1. Reduce the data volume queried each time.
2. Modify the database configurations to increase the RBS of the Oracle database.

3.8 What Should I Do If the MongoDB Connection Migration Fails?

Symptom

What should I do If the MongoDB connection migration fails?

Solution

By default, the **userAdmin** role has only the permissions to manage roles and users and does not have the read and write permissions on a database.

If the MongoDB connection fails to be migrated, you need to view the user permission information in the MongoDB connection to ensure that the user has the read and write permissions on the specified database.

3.9 What Should I Do If a Hive Migration Job Is Suspended for a Long Period of Time?

Symptom

What should I do If a Hive migration job is suspended for a long period of time?

Solution

Manually stop the Hive migration job and add the following attribute settings to the Hive data connection:

- **Attribute Name**: **hive.server2.idle.operation.timeout**
- **Value**: **10m**

In the figure on the left:

3.10 What Should I Do If an Error Is Reported Because the Field Type Mapping Does Not Match During Data Migration Using CDM?

Symptom

When you use CDM to migrate data to DWS, the migration job fails and the error message "value too long for type character varying" is displayed in the execution log.

Possible Cause

The possible cause is that the type of the source table does not match that of the target table. For example, the **dli** field of the source is of the string type, and the **dws** field of the destination is of the varchar(50) type. As a result, the precision is default and the error message "value too long for type character varying" is reported. This issue also occurs for conversion from string to bigint and from bigint to int.

Solution

- Locate the field that is incorrectly mapped based on the error information and contact the DBA to modify the table structure.
- If this issue occurs only for a small amount of data, you can configure the dirty data policy to solve the issue.

3.11 What Should I Do If a JDBC Connection Timeout Error Is Reported During MySQL Migration?

Symptom

The following error message is displayed during MySQL migration: "Unable to connect to the database server. Cause: connect timed out."

Possible Cause

The table has a large data volume, and the source end uses the where statement to filter data. However, the column is not an index column or the column values are not discrete. As a result, the entire table is scanned during the query, causing a JDBC connection timeout. The **c_date** field is not an index column, as shown in **Figure 3-11**.

Figure 3-11 Non-index column

Solution

1. Contact the DBA to modify the table structure, set the columns to be filtered as index columns, and try again.

   If the failure persists because the data is not discrete, perform **2** to **4** and increase the JDBC timeout duration.

2. Locate the MySQL link name based on the job and obtain the link information.

   Figure 3-12 Link information

3. Click the **Links** tab and click **Edit** to edit the link.

   Figure 3-13 Editing the link

4. Click **Show Advanced Attributes**, add parameters **connectTimeout** and **socketTimeout** and their values in **Link Attributes** , and click **Save**.

Figure 3-14 Editing advanced attributes

3.12 What Should I Do If a CDM Migration Job Fails After a Link from Hive to GaussDB(DWS) Is Created?

Symptom

A CDM migration job fails after a link from Hive to GaussDB(DWS) is created.

Solution

You are advised to clear historical data and try again. In addition, when creating a migration job, you are advised to enable the system to clear historical data. This greatly reduces the probability of failures.

3.13 How Do I Use CDM to Export MySQL Data to an SQL File and Upload the File to an OBS Bucket?

Symptom

How do I use CDM to export MySQL data to an SQL file and upload the file to an OBS bucket?

Solution

CDM does not support this operation. You are advised to manually export a MySQL data file, enable the SFTP service on the server, and create a CDM job with SFTP as the source and OBS as the destination. Then you can execute the created job to transfer the file.

3.14 What Should I Do If CDM Fails to Migrate Data from OBS to DLI?

Symptom

CDM fails to migrate data from OBS to DLI.

Solution

Dirty data writing is configured, but no dirty data exists. You need to decrease the number of concurrent tasks to avoid this issue.

3.15 What Should I Do If Error "Configuration Item [linkConfig.createBackendLinks] Does Not Exist" or "Configuration Item [throttlingConfig.concurrentSubJobs] Does Not Exist" Is Reported?

Symptom

Error message "Configuration Item [linkConfig.createBackendLinks] Does Not Exist" or "Configuration Item [throttlingConfig.concurrentSubJobs] Does Not Exist" is displayed during job creation.

Possible Cause

If you create a link or save a job in a CDM cluster of an earlier version, and then access a CDM cluster of a later version, this error occurs occasionally.

Solution

Manually clear the browser cache to avoid this error.

3.16 What Should I Do If Message "CORE_0031:Connect time out. (Cdm.0523)" Is Displayed During the Creation of an MRS Hive Link?

Symptom

Message "CORE_0031:Connect time out. (Cdm.0523)" is displayed during the creation of an MRS Hive link.

Solution

If a message is displayed indicating that the configuration file cannot be downloaded during the creation of an MRS Hive link, insufficient user permissions occur. You are advised to create another service user, grant required permissions to the user, and try again.

To create a data connection for an MRS security cluster, do not use user **admin**. The **admin** user is the default management page user and cannot be used as the authentication user of the security cluster. You can create an MRS user and set **Username** and **Password** to the username and password of the created MRS user when creating an MRS data connection.

> 📖 **NOTE**
>
> - If the CDM cluster version is 2.9.0 or later and the MRS cluster version is 3.1.0 or later, the created user must have the permissions of the **Manager_viewer** role to create links on CDM. To perform operations on databases, tables, and columns of an MRS component, you also need to add the database, table, and column permissions of the MRS component to the user by following the instructions in the MRS documentation.
>
> - If the CDM cluster version is earlier than 2.9.0 or the MRS cluster version is earlier than 3.1.0, the created user must have the permissions of **Manager_administrator** or **System_administrator** to create links on CDM.
>
> - A user with only the **Manager_tenant** or **Manager_auditor** permission cannot create connections.

3.17 What Should I Do If Message "CDM Does Not Support Auto Creation of an Empty Table with No Column" Is Displayed When I Enable Auto Table Creation?

Symptom

Message "CDM Does Not Support Auto Creation of an Empty Table with No Column" is displayed when I enable auto table creation.

Possible Cause

The cause is that the database table name contains special characters, resulting in incorrect syntax. You can resolve this issue by renaming the database table according to the naming rules for database objects.

For example, the name of a data table in the DWS data warehouse can contain a maximum of 63 characters and support letters, digits, underscores (_), dollar signs ($), and number signs (#), and must start with a letter or underscore (_).

Solution

Rename the table.

3.18 What Should I Do If I Cannot Obtain the Schema Name When Creating an Oracle Relational Database Migration Job?

Symptom

I cannot obtain the schema name when creating an Oracle relational database migration job.

Possible Cause

The latest ORACLE_8 driver, which is not supported, may have been uploaded, for example, the Oracle Database 21c (21.3) driver.

Solution

You are advised to use the ojdbc8.jar driver in Oracle Database 12c. You can download the driver from **https://www.oracle.com/database/technologies/jdbc-ucp-122-downloads.html**.

3.19 What Should I Do If invalid input syntax for integer: "true" Is Displayed During MySQL Database Migration?

Symptom

The MySQL database stores values **0** and **1**, rather than **true** and **false**. However, **true** or **false** is read during MySQL database migration, and the following error information is displayed: Unable to execute the SQL statement. Cause: ERROR: invalid input syntax for integer: "true" Where: COPY sd_mask_ext, line 1, column mask_type.

Possible Cause

By default, **tinyInt1isBit** is set to **true** for MySQL databases. As a result, **TINYINT(1)** is processed as **BIT** (that is, **Types.BOOLEAN**), and **1** or **0** is read as **true** or **false**.

Solution

In the advanced attributes of the MySQL link, add either of the following parameters so that tables can be properly created at the destination:

- Parameter **tinyInt1isBit**, with its value set to **false**
- Parameter **mysql.bool.type.transform**, with its value set to **false**

Figure 3-15 Adding link attributes
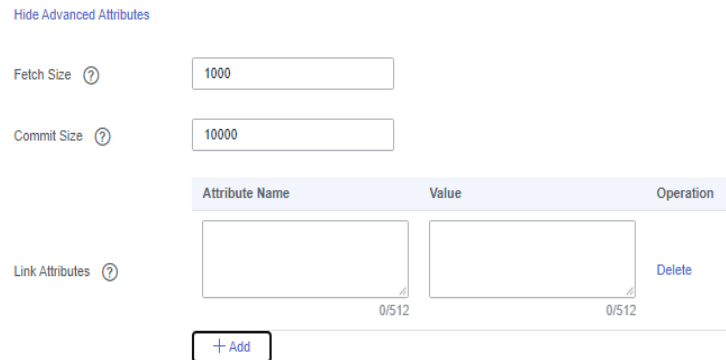
3.20 What Should I Do If the Migration Source Is Oracle and Error Message "snapshot too old" Is Displayed?

This error is caused by Oracle constraints. During the migration, no data can be updated, deleted, or added in the source table. You can increase the value of **UNDO_RETETION** and adjust the size of the UNDO tablespace.

3.21 What Should I Do If Error "Identifier name is too long" Is Reported During Entire DB Migration to Hive?

Symptom

An error message is displayed, indicating that the table name is too long, which is not true.

Possible Cause

A staging table whose name consists of the actual table name and a staging table suffix is created before data is imported. As a result, the job is abnormal.

Solution

- In the advanced attributes of job configuration, set **Import to Staging Table** to **No** so that data will not be imported to the staging table first.
- Shorten the actual table name.

3.22 What Should I Do If Data Is Lost During Migration to DLI?

Symptom

After a job for migrating data to DLI is successfully executed, the number of migrated data records is inconsistent with that in the DLI table, that is, some data is lost.

Possible Cause

1. Check the job configuration. The number of concurrent extractors is 3.

2. According to the DLI documentation, you are not advised to concurrently insert data into a table.

Solution

Set the number of concurrent extractors to 1 and run the job again.

3.23 What Should I Do If the Oracle Link Connectivity Test Is Successful on the Link Creation Page but Fails on the Links Page?

Symptom

On the page for creating an Oracle link, the link connectivity test is successful.

On the **Links** page, the Oracle link connectivity test fails. The following information is displayed.

Failed to connect to the server. Check whether the IP address, hostname, and port number are correct, and whether the network security group and firewall are correctly configured. For details, see the ORA-01005 null password given. logon denied(Cdm 0941) message returned by the database.

Solution

Check whether the IP address, hostname, and port number are correct, and whether the security group and firewall are correctly configured. Locate the fault by referring to the message returned by the database. It is found that the password of the Oracle database contains less than eight characters. Create another Oracle link with the password of the Oracle database set correctly, and the issue is rectified.

3.24 What Should I Do If Destination Fields Are Not Displayed When Auto Table Creation Is Enabled in the Destination Job Configuration?

Symptom

In a job for migrating data from SQL Server to DWS, auto table creation is enabled in the destination job configuration, but destination fields are not displayed on the **Map Field** page.
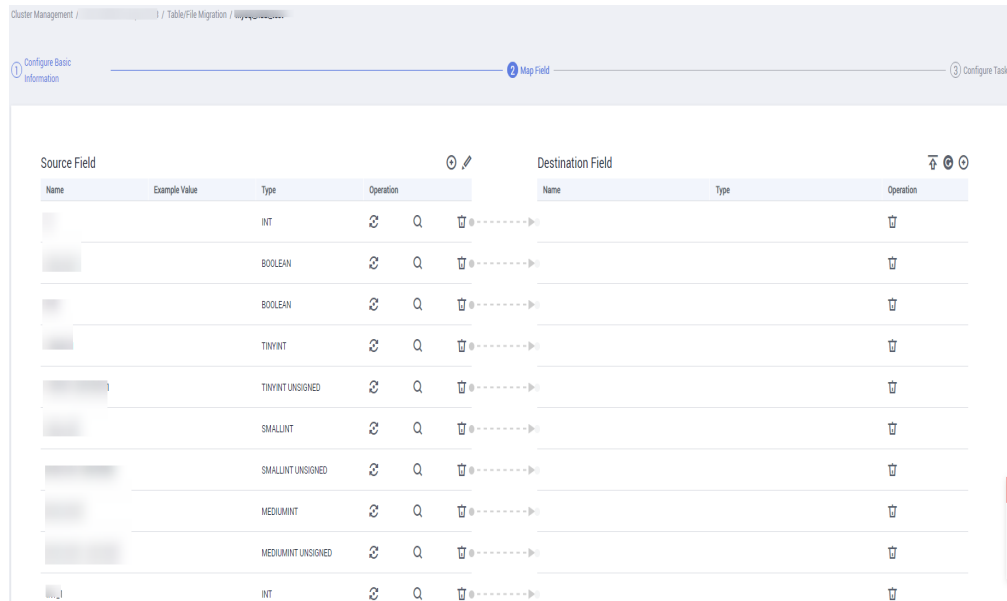
Possible Cause

1. The following error message is displayed in the backend log: org.postgresql.util.PSQLException: ERROR: relation "table name" does not exist.

2. The error may be caused by the sandbox enabled for the CDM cluster, but the error persists after the sandbox is disabled and the CDM cluster is restarted.

3. The name of the table in the source database contain only lowercase letters, but the name of the table to be migrated contains uppercase letters. Ensure that the name of the table to be migrated is consist with that in the database, and this issue is rectified.
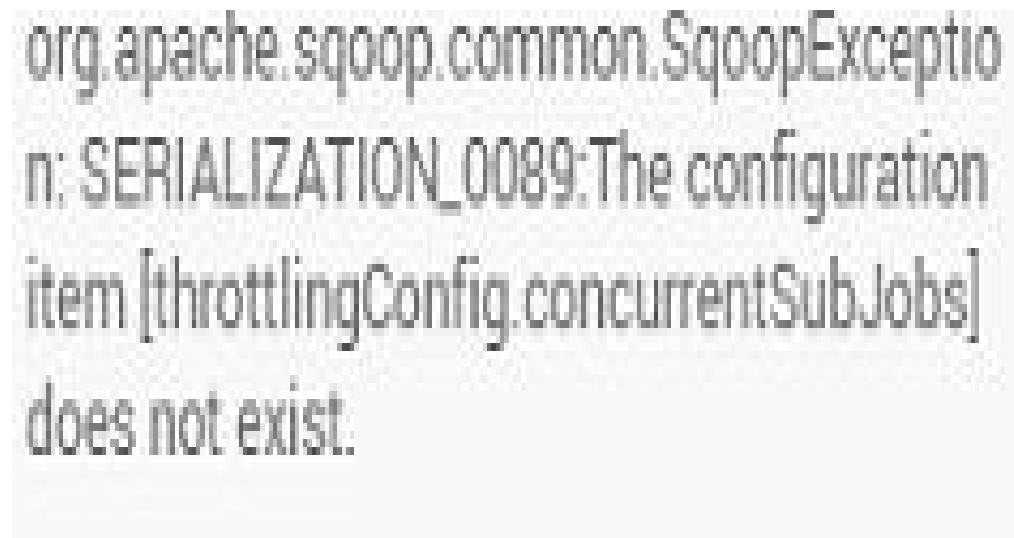
Solution

In the source job configuration, enter the name of the table to be migrated based on its real name in the database or search for and select the table name.

3.25 What Should I Do If a Job Exported from a Cluster Fails to Be Imported to Another Cluster?

Symptom
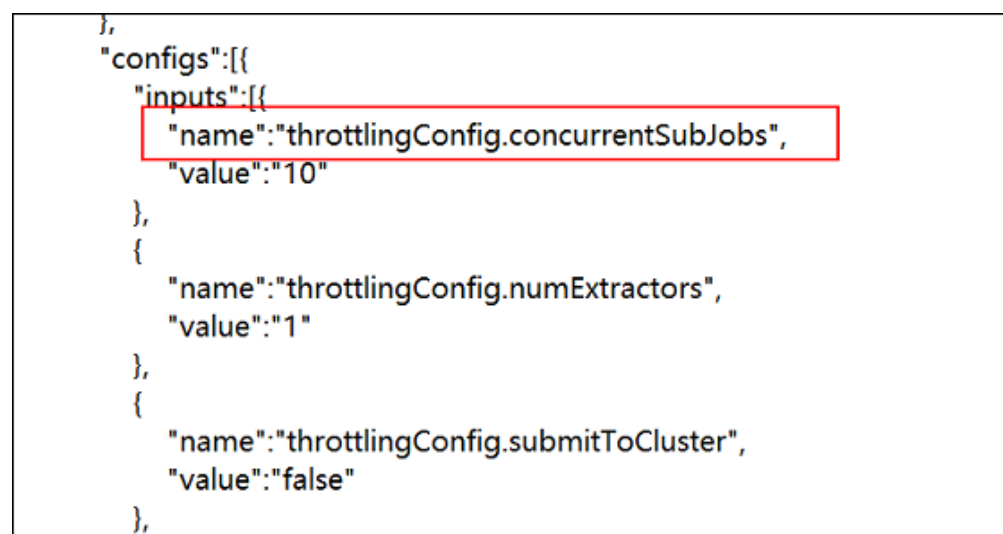
When a job is exported from a CDM cluster of version 2.6.0 to a CDM cluster of version 2.8.6.1, an error message shown in the following figure is displayed.

Possible Cause

1. Some parameters of the two clusters are incompatible. Check the **throttlingConfig.concurrentSubJobs** parameter (number of concurrent subjobs, which is unavailable in the new cluster) in the JSON file exported from the old cluster.

2. Delete the following configuration item from the exported JSON file and import the job again to the new cluster again. The import is successful.

```
{
"name":"throttlingConfig.concurrentSubJobs",
"value":"10"
},
```

Solution

Delete the **"name":"throttlingConfig.concurrentSubJobs"** configuration item from the exported JSON file and import the JSON file again.

3.26 What Should I Do If an Error Message Is Displayed Indicating that the Block Is Missing During HDFS File Migration?

Symptom

1. A connector is created to link the on-premises HDFS and OBS, and the link connectivity is normal.

2. The following error message is displayed when the migration is started:

Error: java.io.IOException: org.apache.hadoop.hdfs.BlockMissingException: Could not obtain block: BP-787476470-192.168.152.10-1573351961380:blk_1073742171_1347 file=/user/hive/warehouse/db_hive.db/emp/emp.txt (state=,code=0)

Possible Cause

1. Files can be obtained using the HDFS client, which means that the file block is not lost.

2. Check whether all DataNode instances of the HDFS service are started. If DataNode instances are stopped, the block cannot be obtained. Check the CDM cluster can communicate with the DataNode instances properly.

   Note: DataNode instances in the HDFS use port 9866 to transmit data.

Solution

The firewall is enabled for DataNode instances. As a result, CDM fails to connect to the DataNode instances and obtain the blocks. Disable the firewall for DataNode instances, and this issue is rectified.

3.27 What Should I Do If the CDM Job Management Page Cannot Be Accessed and a Message Is Displayed Indicating that the Network or Server Cannot Be Accessed?

Symptom

The **Job Management** page cannot be accessed, and a message is displayed, indicating that the network or server is abnormal and asking you to try again.

Solution

1. Press **F12** and check the returned values of interfaces.

2. Check the metrics of the CDM cluster, such as the disk, memory, and CPU.

3. If the preceding metrics of the CDM cluster are normal, clear the browser cache and click **Job Management** again.

3.28 What Should I Do If a CDM Cluster of Version 2.8.6 Fails to Migrate Data from OBS to DLI?

Symptom

Two CDM clusters (version 2.6.0 and 2.8.6, respectively) are used to migrate data from OBS to DLI. Configurations of the two migration jobs are the same. The CDM cluster of version 2.6.0 successfully migrates the data, but the CDM cluster of version 2.8.6 fails to migrate the data. The following figure shows log with the error message.

Possible Cause

1. The error may occur because the field type conversion between the source and destination is abnormal.

2. Change the field type for the destination table from biginit to string and rerun the job. The job still fails, and the same error is reported.

3. Enable dirty data and rerun the job. The job still fails, but three pieces of data are successfully migrated to destination table.

4. After a comparison of the data records that fail to be migrated and those that are successfully migrated, it is suspected that the error is caused by fields of the Boolean type.

Solution

Change the type of the destination fields from Boolean to string and rerun the job. The job is successful. Some Boolean fields at the source have a null value, which results in a migration failure.

In CDM 2.8.6, the cluster verification is stricter. If the data type of the destination is Boolean, the system checks the data type during data insertion. If the data type is not Boolean, an error is reported.

3.29 What Should I Do If Error Message "Read timed out" Is Displayed During DWS Data Migration?

Symptom

When CDM is used to migrate data from a GaussDB(DWS) cluster to another, the following error is reported after some data is written:

```
org.postgresql.util.PSQLException: Database connection failed when ending copy.......caused by:
java.net.SocketTimeoutExcepetion:Read timed out;
```

Fault Analysis

In both the source and destination job configurations, the where clause rather than an index column is used to filter data. The query scans the entire table that contains hundreds of millions of rows. As a result, the JDBC database connection fails and data reading times out, causing a sqoop exception and job failure.

In a migration job, CDM functions as a client to extract some data from the source and write the data to the destination. Then, CDM repeats this process until all data is written to the destination. You can add advanced attribute **socketTimeout** and set an interval between each data extraction and writing process so that the CDM session is always normal.

Solution

Increase the JDBC connection timeout interval and run the migration job again.

Step 1 Go to the **Links** page and locate the connector between the source and destination.

Step 2 Click **Edit** in the **Operation** column. On the displayed page, click **Show Advanced Attributes** and then **Add** to add attribute **socketTimeout** and set its value to **36000**. Then test and save the link.

**Step 3** Restart the job and wait until it is successfully executed.

----End

3.30 What Should I Do If No Database or Table Information Can Be Obtained Through a Hive Link?

Symptom

The database and table content cannot be obtained through a Hive link in the CDM cluster. After the database and table are manually configured, fields can be displayed, but an error message is displayed, indicating that the Hive client fails to be initialized.

get_table_req
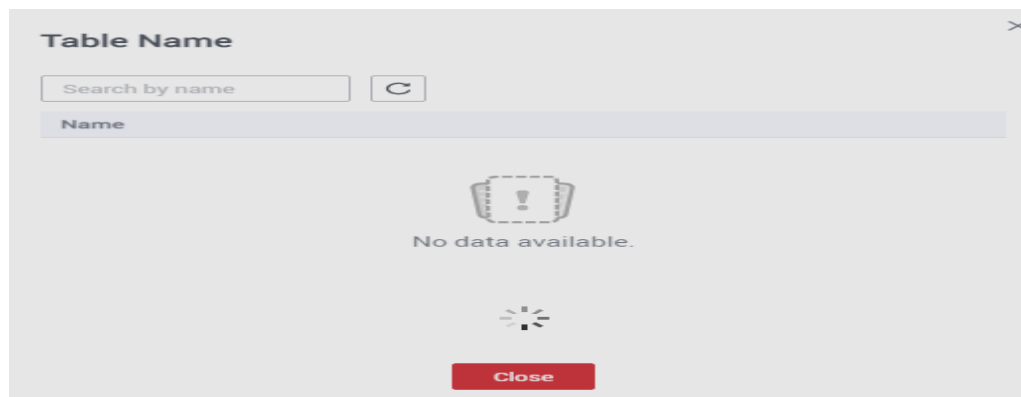
Solution

1. The MRS cluster version is 1.8.1, and the CDM cluster version is 2.6.0.
2. According to the error message, the Hive SDK encapsulated in CDM cannot identify the Hive data source, but the Hive link connectivity is normal. Therefore, check the parameters for the Hive connector.

   The user, user group, and role of MRS Manager in the security cluster are correctly configured. The Hive version is HIVE_3_*X*.

3. As the Hive version of the MRS cluster version 1.8.1 is 1.2.1, the Hive version hive_1_*X* should be selected for the Hive connector. Change the Hive version and re-create a job. The job can be created successfully, and the issue is rectified.

3.31 What Should I Do If Error "get filesystem" Is Reported During the Creation of a FusionInsight HDFS Link?

Symptom

During the connectivity test for a FusionInsight HDFS link, a message is displayed indicating that the file system fails to be obtained.
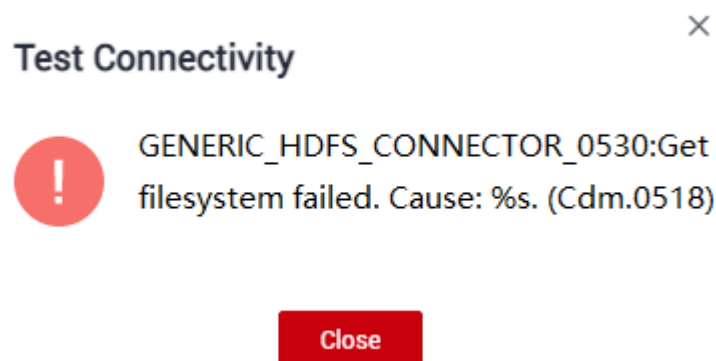
Solution

The management IP address is incorrect. Use a floating IP address of the cluster and the port of the HDFS WebUI, and the issue is rectified.

```
[root@node-master1IClx HDFS]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 1            l netmask 255.255.255.0  broadcast 1
        ether f                a txqueuelen 1000  (Ethernet)
        RX packets 299702871  bytes 44983338695 (41.8 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 251978522  bytes 183960299702 (171.3 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0:wsom: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet            l netmask 255.255.255.0  broadcast
        ether          :3d.      txqueuelen 1000  (Ethernet)

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 1        1  netmask 255.0.0.0
        loop  txqueuelen 0  (Local Loopback)
        RX packets 1041006791  bytes 359182733000 (334.5 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1041006791  bytes 359182733000 (334.5 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[root@node-master1IClx HDFS]#
```


| Hadoop | Overview | Datanodes | Datanode Volume Failures | Snapshot | Startup Progress | Utilities |

Overview 'node-master1IClx:9820' (active)

Namespace:	hacluster
Namenode ID:	19
Started:	Wed Dec 02 16:26:44 +0800 2020
Version:	3.1.1-mrs-2.0, 35f9l f
Compiled:	1kins from (no branch)
Cluster ID:	myh er
Block Pool ID:	

Summary

3.32 What Should I Do If Error "Invoke DLI service api failed" Is Reported When the Data Migration from MySQL to DLI Is About to Complete?

Symptom

When the migration of 40 million data records from MySQL to DLI is about to complete, the following error is displayed.

Fault Analysis

Step 1 According to the error message, the error may be caused by the write operation on DLI. However, as some log screenshots are lacking, go to the CDM cluster and view the job log.

```
2022-02-28 16:55:05.650|ERROR|LocalJobRunner Map Task #1 #loader|||o.a.s.connector.uquery.UQueryLoader An error occurred when l
oading data to DLI.
org.apache.sqoop.common.SqoopException: UQUERY_CONNECTOR_0001:Invoke DLI service api failed, failed reason is %s.
        at org.apache.sqoop.connector.uquery.intf.impl.UQueryUploadJob.beginCommit(UQueryUploadJob.java:30)
        at org.apache.sqoop.connector.uquery.UQueryLoader.waitForUQueryCompletion(UQueryLoader.java:207)
        at org.apache.sqoop.connector.uquery.UQueryLoader.load(UQueryLoader.java:103)
        at org.apache.sqoop.connector.uquery.UQueryLoader.load(UQueryLoader.java:32)
        at org.apache.sqoop.job.mr.SqoopOutputFormatLoadExecutor$ConsumerThread.runInterval(SqoopOutputFormatLoadExecutor.java:
747)
        at org.apache.sqoop.job.mr.SqoopOutputFormatLoadExecutor$ConsumerThread.run(SqoopOutputFormatLoadExecutor.java:629)
        at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
        at java.util.concurrent.FutureTask.run(FutureTask.java:266)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        at java.lang.Thread.run(Thread.java:748)
Caused by: com.huawei.dli.sdk.exception.DLIException: HTTP Status:400 Bad Request; DLI.0001 queue type must be sql
        at com.huawei.dli.sdk.SQLJob.submitSqlNoRetry(SQLJob.java:311)
        at com.huawei.dli.sdk.SQLJob.asyncSubmit(SQLJob.java:127)
        at com.huawei.dli.sdk.UploadJob.beginCommit(UploadJob.java:220)
        at org.apache.sqoop.connector.uquery.intf.impl.UQueryUploadJob.beginCommit(UQueryUploadJob.java:27)
        ... 10 common frames omitted
```

Step 2 According to the error analysis, the queue type may be incorrect (the correct type is SQL queue). Data is first written to an OBS file and then imported to a DLI table through foreign table mapping. The data has been basically imported, but an error is reported during the final mapping because an incorrect queue is used (the DLI SQL queue should be used).

----End

Solution

Contact the user to check that the selected queue is not an SQL queue and that no SQL queue is available. Let the user purchase a DLI SQL queue to perform the migration.

3.33 What Should I Do If a MongoDB Field Fails to Be Migrated to the Destination?

Symptom

In a CDM job, a field is added for the source MongoDB and mapped to the destination MongoDB. After the job is executed, the field is not migrated to the destination.

Fault Analysis

Step 1 According to the documentation, CDM cannot obtain all columns by obtaining sample values.

Step 2 MongoDB is a document database and does not have schemes. When a field is added for the source MongoDB, the JSON key of the first record is obtained in field mapping. CDM supports the combine() function, which can encapsulate non-common columns into one column.

----End

Solution

Step 1 When the MongoDB Reader plug-in is used to read data, the combine() function can combine multiple fields in the MongoDB document into a JSON string which is migrated to the destination as a field.

Step 2 The destination database decomposes the received data using SQL statements.

```
1    insert overwrite table bi_temp.enterprise_ranking_test
2    select
3    enterprise_name,
4    external_id,
5    type,
6    rank_name,
7    rank_year,
8    province_code,
9    get_json_object(combine,'$.extra') as extra,
10   is_valid,
11   get_json_object(combine,'$.owner_enterprise') as owner_enterprise,
12   release_time,
13   enterprise_id,
14   create_time,
15   update_time,
16   get_json_object(combine,'$.commend_party') as commend_party,
17   combine
18   from
19   bi_temp.enterprise_ranking_test
20
```

> **□ NOTE**
>
> This does not affect mapping of the existing fields. combine() returns a JSON string that contains all new fields. Data can be obtained using SQL statements at the destination.

----End

3.34 What Should I Do If a Field Is Escaped and Contains a Backslash (\) During the Migration of a DLI Foreign Table (OBS File) to DWS?

Symptom

After the migration of a DLI foreign table to GaussDB(DWS), a field has an extra pair of quotation marks which contain an extra escape character. Other fields are normal.

Source

ndard_ia_count	glacier_size	glacier_count	total_size	total_count	tenant_level	illegal_tag	secure_score	secure_score_fact...	other_indicators	mainland_recommend_class_na...	update_time
	0	0	7504	8	V5	null	100	{"base_line_deducti...	{"inbound_atta...	g	1640815349
	0	0	27472480	2747248	V5	null	80	{"base_line_deducti...	{"inbound_atta...	g	1640815349
	0	0	1249273...	4	V5	null	80	{"base_line_deducti...	{"inbound_atta...	g	1640815349
	0	0	5753260	188	V0	null	80	{"base_line_deducti...	{"inbound_atta...	s	1640815349
	0	0	294128	92	V4	null	80	{"base_line_deducti...	{"inbound_atta...	s	1640815349
	0	0	2420344...	2448	V4	null	80	{"base_line_deducti...	{"inbound_atta...	s	1640815349
	0	0	1779301...	776	V0	null	60	{"base_line_deducti...	{"inbound_atta...	default	1640815349
	0	0	5966242...	12538588	V5	null	58	{"base_line_deducti...	{"inbound_atta...	g	1640815349
	0	0	3064382...	24	V5	null	98	{"base_line_deducti...	{"inbound_atta...	g	1640815349
	0	0	1325010...	16	V5	null	98	{"base_line_deducti...	{"inbound_atta...	g	1640815349

Fault Analysis

1. As shown in the preceding figure, the sample value at the source contains special characters such as brackets and quotation marks. The JDBC driver escapes the field, and the escape character is displayed at the destination.

2. For DLI foreign table and OBS bucket storage, and file-to-table migration, you can add a quote character (") to the source job configuration.

Solution

In the source job configuration, set **Use Quote Char** to **Yes** and use a double quotation mark (").

3.35 What Should I Do If "Error occurs during loader run" Is Reported During Migration from PostgreSQL to Hive?

Symptom

During data migration from PostgreSQL to Hive, "Error occurs during loader run" is reported.

Fault Analysis

1. Check the errors in the CDM log of the previous day: 2021-09-29 10:35:32,638 ERROR LocalJobRunner Map Task #13 #loader [org.apache.sqoop.connector.hive.hiveWriter.HiveOrcWriter:83] Create file system error.

   java.nio.file.AccessDeniedException: obs-itotshujuruhu-bingxing-fangcongyang: doesBucketExist on obs-itotshujuruhu-bingxing-fangcongyang: com.obs.services.exception.ObsException: Error message:Request Error.OBS servcie Error Message. -- ResponseCode: 403

2. According to the errors "hiveWriter.HiveOrcWriter:83] Create file system error" and "Error.OBS servcie Error Message. -- ResponseCode: 403", an error occurs during synchronization of tables from Hive to the OBS directory. **OBS Support** may be set to **No** during the connector creation.

3. Check the connector configuration. It is found that **OBS Support** is set to **No**. This parameter specifies whether OBS storage is supported. If Hive table data is stored in OBS, the value must be set to **Yes**.

Solution

Set **OBS Support** to **Yes** and enter the password.

3.36 What Should I Do If Error Message "Lost connection to MySQL server during query" Is Displayed During Migration from MySQL to DWS?

Symptom

Error message "GENERIC_JDBC_CONNECTOR_0904:ERROR occurs while retrieving data from result. Cause : closed connection:stream closed con:192.168.XX.XX." is displayed during data migration from MySQL to DWS.

Fault Analysis

1. The error may occur due to a large amount of user data or a connection interruption. Check whether the timeout duration is configured in the advanced attributes.

2. The timeout duration is 6,000,000 milliseconds, which is large enough. Check whether the MySQL database of the customer is normal. Run the SQL statement for querying the source in the log and run the SQL statement on the MySQL client to check whether an error is reported.

```
SELECT F_MARKET_ID, F_MARKET_NAME, F_MARKET_TYPE_ID, F_LOWEST_FEE, F_EFFECT_DATE, F_EXPIRE_DATE, F_IS_SMS, F_IS_NOTICE,
F_SMS_CONTENT, F_NOTICE_CONTENT, F_STATE, F_OPERATER, F_OPERATER_NAME, F_OPERATE_TIME, F_MODIFIER, F_MODIFIER_NAME,
F_MODIFED_TIME, F_EXECUTE_TIME, F_EXECUTE_DAY, F_EXECUTE_STATE, F_NUMBEROFPEOPLE, F_EVERPEOPLENUMBER, F_REWARD_TYPE, F_IS_WECHAT,
F_WECHAT_CONTENT, F_SHOP_RULE, F_USE_TERMINAL, F_REDPAPER_WISHING, F_REDPAPER_ACTIVITY_NAME, F_REDPAPER_REMARK,
F_REDPAPER_SEND_NAME, F_IS_ANAMARKET, F_COMPAREDAY, F_SHOP_USER_DIVIDE, F_Change_Exp_Day, F_Exp_Date, F_Exp_Multiple,
F_NUMBEROFPEOPLE_CURRENT, F_CONSUMPTION_CODITION_RULE, F_EXCLUSIVE_PRIVILEGE, F_USE_TYPE, F_MARKET_TREE_CODE,
F_TICKET_RULE_OF_MARKET, F_BIRTH_EXPIRE_DATE, F_BIRTH_EFFECT_DATE, F_BIRTH_DATE, F_BIRTH_TYPE, F_BIRTH_CONDIN, F_MER,
F_REDPAPER_WISHING_ACCOUNT, F_REDPAPER_ACTIVITY_NAME_ACCOUNT, F_REDPAPER_REMARK_ACCOUNT, F_REDPAPER_SEND_NAME_ACCOUNT,
F_CUS_SUPER, F_CUS_GRADE, F_DATE_TYPE, F_OPERATE_STATE, F_DIRECT_TYPE, F_MAX_AWARD_VALUE FROM activity.t_market_plan_main_info
WHERE 1=1
> Lost connection to MySQL server during query
```

3. When the query statement is executed to query the entire table, error message "Lost connection to MySQL server during query" is displayed. When the **count** statement is executed again to query data, the query is successful.

4. The value of **max_allowed_packet** for the MySQL database is too small. However, the value is 1 GB and cannot be increased.

5. Check the connector configuration. It is found that **Fetch Size** is set to **3000**. As a result, the volume of data to be queried in a batch may exceed 1 GB, and an error is reported.

Solution

1. Set **Fetch Size** to **1000**.

2. Use the where clause to periodically migrate some data.

3.37 What Should I Do If Error Message "For input string: "false" Is Displayed During Field Type Conversion During MySQL-to-DLI Migration?

Symptom

The following error is reported during MySQL-to-DLI migration: java.lang.NumberFormatException: For input string: "false".

```
                at java.lang.Thread.run(Thread.java:748)
2021-07-28 16:20:46,854 ERROR uquery-loader-1 [org.apache.sqoop.connector.uquery.processor.Dataconsumer:174] failed to convert the record.
java.lang.NumberFormatException: For input string: "false"
                at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)
                at java.lang.Integer.parseInt(Integer.java:580)
                at java.lang.Integer.parseInt(Integer.java:615)
                at org.apache.sqoop.connector.uquery.processor.Dataconsumer.convert(Dataconsumer.java:271)
                at org.apache.sqoop.connector.uquery.processor.Dataconsumer.run(Dataconsumer.java:172)
                at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
                at java.util.concurrent.FutureTask.run(FutureTask.java:266)
                at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
                at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
                at java.lang.Thread.run(Thread.java:748)
2021-07-28 16:20:46,854 ERROR uquery-loader-0 [org.apache.sqoop.connector.uquery.processor.Dataconsumer:174] failed to convert the record.
java.lang.NumberFormatException: For input string: "false"
                at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)
                at java.lang.Integer.parseInt(Integer.java:580)
                at java.lang.Integer.parseInt(Integer.java:615)
                at org.apache.sqoop.connector.uquery.processor.Dataconsumer.convert(Dataconsumer.java:271)
                at org.apache.sqoop.connector.uquery.processor.Dataconsumer.run(Dataconsumer.java:172)
                at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
                at java.util.concurrent.FutureTask.run(FutureTask.java:266)
                at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
                at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
                at java.lang.Thread.run(Thread.java:748)
2021-07-28 16:20:46,854 ERROR uquery-loader-1 [org.apache.sqoop.connector.uquery.processor.Dataconsumer:174] failed to convert the record.
java.lang.NumberFormatException: For input string: "false"
                at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)
                at java.lang.Integer.parseInt(Integer.java:580)
                at java.lang.Integer.parseInt(Integer.java:615)
                at org.apache.sqoop.connector.uquery.processor.Dataconsumer.convert(Dataconsumer.java:271)
                at org.apache.sqoop.connector.uquery.processor.Dataconsumer.run(Dataconsumer.java:172)
                at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
                at java.util.concurrent.FutureTask.run(FutureTask.java:266)
                at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
                at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
                at java.lang.Thread.run(Thread.java:748)
2021-07-28 16:20:46,854 ERROR uquery-loader-0 [org.apache.sqoop.connector.uquery.processor.Dataconsumer:174] failed to convert the record.
java.lang.NumberFormatException: For input string: "false"
                at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)
                at java.lang.Integer.parseInt(Integer.java:580)
                at java.lang.Integer.parseInt(Integer.java:615)
                at org.apache.sqoop.connector.uquery.processor.Dataconsumer.convert(Dataconsumer.java:271)
                at org.apache.sqoop.connector.uquery.processor.Dataconsumer.run(Dataconsumer.java:172)
                at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
                at java.util.concurrent.FutureTask.run(FutureTask.java:266)
                at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
```

Fault Analysis

1. An error may occur when the boolean type whose value is "false" is converted to the int type. Check the mapping on the **Map Field** page.

udid	9C4D2F84CB4A6A...	VARCHAR(100)	⟳	udid	string
os	Android10	VARCHAR(100)	⟳	os	string
device	HUAWEI ALP-AL00	VARCHAR(100)	⟳	device	string
nation	CN	VARCHAR(100)	⟳	nation	string
login_tag	1.2.1.3	VARCHAR(100)	⟳	login_tag	string
support_gpu_insta...	0	TINYINT	⟳	support_gpu_instancing	int
language	zh-CN	VARCHAR(100)	⟳	language	string
runm	5.94 GB	VARCHAR(100)	⟳	runm	string
netcode	46002	VARCHAR(100)	⟳	netcode	string
nettype	WIFI	VARCHAR(100)	⟳	nettype	string
devres	Point(1080, 1772)	VARCHAR(100)	⟳	devres	string
cpunum	8	VARCHAR(100)	⟳	cpunum	string
cpumaxf	1844000	VARCHAR(100)	⟳	cpumaxf	string
cpuminf	509000	VARCHAR(100)	⟳	cpuminf	string
sdkId	1627314838169-7...	VARCHAR(100)	⟳	sdkid	string
step_id	60001001	INT	⟳	step_id	int

2. According to the field mapping analysis in the previous step, the source field **support_gpu_instancing** is of the TINYINT type, and its value is **0** or **1**.

However, the actual value should be **false** or **true**. An error is reported when fields of the int type are migrated to the destination, indicating that the type conversion is incorrect. This is because MySQL automatically identifies and converts **0** or **1** to **false** or **true**.

Solution

To resolve this issue, use either of the following methods:

- Modify the table creation statement at the destination and change int type to boolean type.
- Change the value of **mysql.bool.type.transform** to **false** for the source MySQL database.

For the second method:

You can add parameter **mysql.bool.type.transform** and set its value to **false** in the advanced attributes of the MySQL connector.

3.38 What Should I Do If an Error Occurs During the Migration of TINYINT Data from MySQL to DWS?

Symptom

During data migration from MySQL to DWS, error "ERROR:value '-1'is out of range for 8 b-bit integer" is reported.

vmall_mms_rt.tbl_activity_baseinfo

org.apache.sqoop.common.SqoopException: GENERIC_JDBC_CONNECTOR_0902:Unable to execute the SQL statement. Cause : ERROR: value "-1" is out of range for 8-bit integer
Where: COPY tbl_activity_baseinfo, line 1, column create_source_type: "-1".
at org.apache.sqoop.utils.ErrorResourceUtil.throwSqoopException(ErrorResourceUtil.java:31)
at org.apache.sqoop.connector.jdbc.Writer.DwsCsvWriter.throwException(DwsCsvWriter.java:444)
at org.apache.sqoop.connector.jdbc.Writer.DwsCsvWriter.endCopy(DwsCsvWriter.java:363)
at org.apache.sqoop.connector.jdbc.Writer.DwsCsvWriter.onClose(DwsCsvWriter.java:308)
at org.apache.sqoop.connector.jdbc.Writer.DwsCsvWriter.close(DwsCsvWriter.java:294)
at org.apache.sqoop.connector.jdbc.GenericJdbcLoader.load(GenericJdbcLoader.java:129)
at org.apache.sqoop.connector.jdbc.GenericJdbcLoader.load(GenericJdbcLoader.java:44)
at
org.apache.sqoop.job.mr.SqoopOutputFormatLoadExecutor$ConsumerThread.runInterval(SqoopOutputFormatLoadExecutor.java:728)
at org.apache.sqoop.job.mr.SqoopOutpu

Fault Analysis

1. Value **-1** cannot be inserted into the destination. Check the field mapping at the destination.

2. Check the table creation statement.

```
 1 SET search_path = vmall_mms_rt;
 2 CREATE  TABLE tbl_activity_baseinfo (
 3     id bigint NOT NULL,
 4     code character varying(100),
 5     type integer,
 6     name character varying(1000),
 7     start_time timestamp without time zone,
 8     end_time timestamp without time zone,
 9     status integer,
10     description character varying(4000),
11     rule_desc character varying(1024),
12     create_by character varying(256),
13     create_time timestamp without time zone,
14     update_by character varying(256),
15     update_time timestamp without time zone,
16     submit_by character varying(256),
17     multi_lang character varying(6000),
18     be_code character varying(100),
19     version integer,
20     extend_type integer,
21     extend character varying(1024),
22     batch_id integer,
23     carrier_code character varying(100),
24     create_source_type tinyint
25 )
```

3. According to the preceding figure, INT1 is the alias of the DWS field type TINYINT, so field mapping is correct. Check whether DWS TINYINT supports the range and why the error message indicates that **-1** is not supported. According to the DWS field type description, DWS TINYINT supports the range [0,255] and does not support negative numbers. The TINYINT type of MySQL supports range [-128,127].

4. SMALLINT supports negative numbers. You are advised to use SMALLINT when creating tables at the destination.

Solution

1. The mapping field INT1 is the alias of the TINYINT type of DWS. The mapping is correct.

2. For DWS, the value of TINYINT ranges from 0 to 255. If the migration source is MySQL contains negative numbers such as –1, you are advised to use SMALLINT (whose value range is –32,768 to +32,767).

   ◨ **NOTE**

   The value range of the TINYINT type of Hive and MySQL is [–128, 127]. The value range of the TINYINT type of DWS is [0,255].
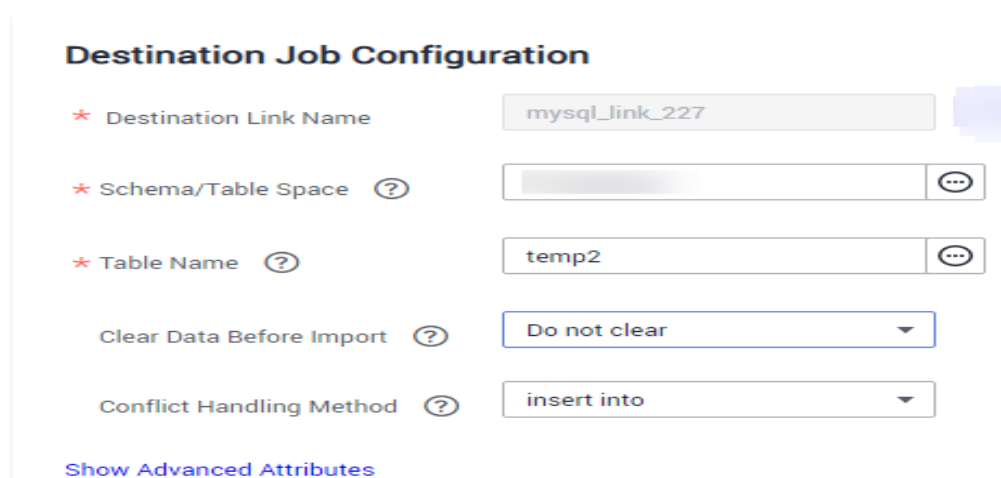
3.39 What Should I Do If the Data Volume Is Inconsistent Before and After Data Migration?

Symptom

After CDM is used to migrate data, the number of data records in the destination database is dozens or thousands greater than that in the source database.

Fault Analysis

In the destination job configuration, **Clear Data Before Import** is set to **No**. As a result, multiple operations may be performed and duplicate data exists.
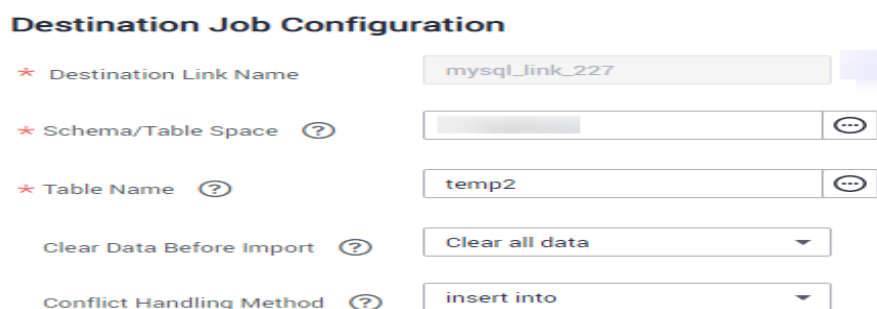
Destination Job Configuration

* Destination Link Name	mysql_link_227
* Schema/Table Space ?	
* Table Name ?	temp2
Clear Data Before Import ?	Do not clear
Conflict Handling Method ?	insert into

Show Advanced Attributes

Solution

Set **Clear Data Before Import** to **Yes**. Rerun the migration job and check that the number of records is the same before and after the migration.

Destination Job Configuration

* Destination Link Name	mysql_link_227
* Schema/Table Space ?	
* Table Name ?	temp2
Clear Data Before Import ?	Clear all data
Conflict Handling Method ?	insert into

3.40 What Should I Do If an Error Is Reported Indicating an Incorrect Username or Password During Link Creation but They Are Correct Indeed?

Symptom

During the creation of a MySQL link, an error is reported indicating an incorrect username or password, but they are correct indeed. With the same configuration, a data link can be successfully created on ROMA.

Fault Analysis

The MySQL database has a whitelist. Test the link from a MySQL client on another ECS that can communicate with the current ECS through the intranet.

```
2020-11-26 17:16:43,571 ERROR http-nio-▨▨▨▨▨▨▨▨▨-exec-6 [org.apache.sqoop.server.SqoopProtocolServlet:78] Exception in POST http://▨▨▨▨▨▨▨▨/sqoop/v1/link
org.apache.sqoop.common.SqoopException: CORE_0028:User name or password is incorrect. Detail message : Access denied for user '▨▨▨▨▨▨▨▨▨▨▨'@'▨▨▨▨▨▨▨▨' (using password: YES)
        at org.apache.sqoop.utils.ErrorResourceUtil.throwSqoopException(ErrorResourceUtil.java:31)
        at org.apache.sqoop.connector.jdbc.GenericJdbcExecutor.throwException(GenericJdbcExecutor.java:1246)
        at org.apache.sqoop.connector.jdbc.GenericJdbcExecutor.getConnection(GenericJdbcExecutor.java:1203)
        at org.apache.sqoop.connector.jdbc.configuration.LinkConfig$ConfigValidator.validate(LinkConfig.java:269)
        at org.apache.sqoop.connector.jdbc.configuration.LinkConfig$ConfigValidator.validate(LinkConfig.java:204)
        at org.apache.sqoop.validation.ConfigValidationRunner.executeValidator(ConfigValidationRunner.java:182)
        at org.apache.sqoop.validation.ConfigValidationRunner.validateArray(ConfigValidationRunner.java:148)
        at org.apache.sqoop.validation.ConfigValidationRunner.validateConfig(ConfigValidationRunner.java:129)
        at org.apache.sqoop.validation.ConfigValidationRunner.validate(ConfigValidationRunner.java:90)
        at org.apache.sqoop.model.ConfigUtils.validateConfigs(ConfigUtils.java:298)
```

Solution

The MySQL account is not authorized to use the CDM cluster. The MySQL database needs to authorize the IP address of the CDM cluster.

3.41 What Should I Do If a Message Is Displayed Indicating that the Fields in the Lower Camel Case Do Not Exist During Data Migration from a Database to OBS?

Symptom

When data is written to OBS from a database, a message is displayed indicating that fields in the lower camel case do not exist.

Fault Analysis

The fields are named in lower camel case, but the PG database is case sensitive. Therefore, the fields cannot be found.

Solution

Set a quote character in the advanced attributes of the link configuration. The issue is resolved.

3.42 What Should I Do If Error "invalid utf-8 character string" Is Reported When CSV Data Is Inserted into the MySQL Database?

Symptom

The migration job fails to be executed, and message "invalid utf-8 character string" is displayed.

Fault Analysis

The error may be caused by the data format. Check the log to confirm the issue.

Solution

1. Check the sqoop logs in the background. The error may be caused by the incorrect source data type and format.
2. Analyze the source data type. It is found that dirty data exists and the source data type is incorrect.
3. Configure dirty data dump on the CDM console and rerun the job. The job is successfully executed, and the dirty data type in the OBS bucket is incorrect.

3.43 What Should I Do If a Scheduled Job Fails and Link Connectivity Is Abnormal?

Symptom

The connection to the source database is abnormal. The following message is displayed:

Check whether the IP address, host name, and port number are correct, and whether the network security group and firewall are correctly configured. Locate the cause based on the error message.

Fault Analysis

Query the cluster information, obtain the public IP address, and curl the IP address of the source database from the CDM cluster.

According to the result, the CDM cluster is faulty. The EIP is not bound to the CDM cluster.

You are advised to unbind and then bind the EIP to the CDM cluster.

Solution

After releasing the EIP, bind another EIP to the CDM cluster or assign a VPC agency to the CDM cluster, and check whether the EIP is normal.

3.44 What Should I Do If CSV Data Fails to Be Inserted into a MySQL Database?

Symptom

The job fails to be executed, and message "invalid utf-8 character string" is displayed.

Fault Analysis

According to the error information, dirty data exists in the user data and the types of some fields do not match.

Solution

1. Check the sqoop logs in the background. The error may be caused by the incorrect data type or format of the source. (You can obtain the job logs from the customer or export all logs on the console.)

2. Analyze the source data type. It is found that dirty data exists and the source data type is incorrect.

3. Configure dirty data dump on the CDM console and rerun the job. The job is successfully executed, and the dirty data type in the OBS bucket is incorrect.

3.45 What Should I Do If Error "timeout waiting for connection from pool" Is Reported During ES Writing?

Symptom

Error "timeout waiting for connection from pool" is reported during ES writing, and the log contains multiple **es_rejected_execution_exception** fields.

Fault Analysis

On the cerebro page, the index has only one shard. However, when an index with three shards is created, **es_rejected_execution_exception** is also reported.
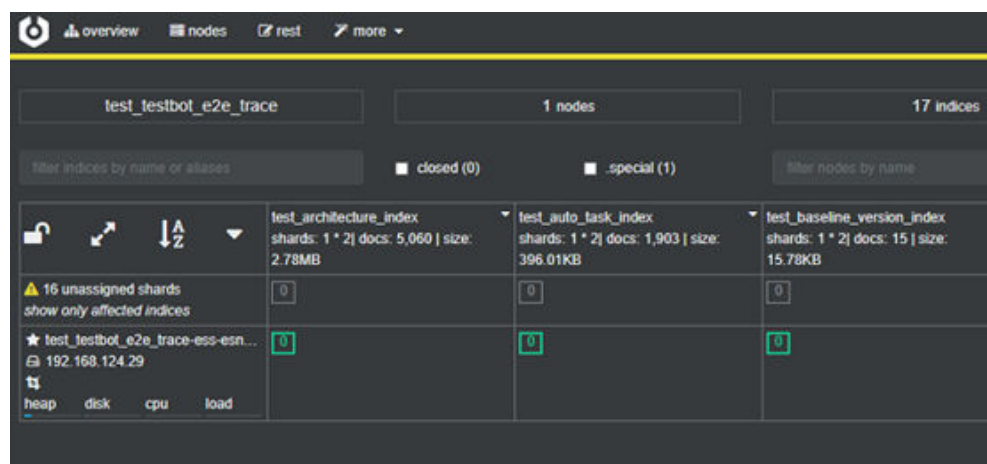
It is found that almost all records are written to a shard. The root cause is that hot spots are generated.

Solution

During migration, the user selected a primary key, which replaces **_id**. The calculated shards belong to the same index.

1. Users are not advised to select the primary key so that ES automatically generates **_id**. In this way, the obtained hash values are scattered.

2. If the user's application must replace **_id** with its own primary key, the user is advised to use an ES cluster with better performance.

3.46 Why Is Error ORA-01555 Reported During Migration from Oracle to DWS?

Symptom

Error ORA-01555 is reported during data migration from Oracle to DWS.

```
665  2020-09-21 22:51:02,991 ERROR LocalJobRunner Map Task #3 [org.apache.sqoop.common.SqoopException:111] SqoopException
666  java.sql.SQLException: ORA-01555: snapshot too old: rollback segment number 3 with name "_SYSSMU3_2097677531$" too small
667
668      at oracle.jdbc.driver.T4CTTIoer11.processERROR(T4CTTIoer11.java:494)
669      at oracle.jdbc.driver.T4CTTIoer11.processERROR(T4CTTIoer11.java:446)
670      at oracle.jdbc.driver.T4C8Oall.processERROR(T4C8Oall.java:1054)
671      at oracle.jdbc.driver.T4CTTIfun.receive(T4CTTIfun.java:623)
672      at oracle.jdbc.driver.T4CTTIfun.doRPC(T4CTTIfun.java:252)
673      at oracle.jdbc.driver.T4C8Oall.doOALL(T4C8Oall.java:612)
674      at oracle.jdbc.driver.T4CPreparedStatement.doOall8(T4CPreparedStatement.java:226)
675      at oracle.jdbc.driver.T4CPreparedStatement.fetch(T4CPreparedStatement.java:1023)
676      at oracle.jdbc.driver.OracleStatement.fetchMoreRows(OracleStatement.java:3353)
677      at oracle.jdbc.driver.InsensitiveScrollableResultSet.fetchMoreRows(InsensitiveScrollableResultSet.java:736)
678      at oracle.jdbc.driver.InsensitiveScrollableResultSet.absoluteInternal(InsensitiveScrollableResultSet.java:692)
679      at oracle.jdbc.driver.InsensitiveScrollableResultSet.next(InsensitiveScrollableResultSet.java:406)
680      at org.apache.sqoop.connector.jdbc.sql.impl.WrapResultSet.next(WrapResultSet.java:36)
681      at org.apache.sqoop.connector.jdbc.GenericJdbcExtractor.extractObjectRecord(GenericJdbcExtractor.java:151)
682      at org.apache.sqoop.connector.jdbc.GenericJdbcExtractor.extract(GenericJdbcExtractor.java:129)
683      at org.apache.sqoop.connector.jdbc.GenericJdbcExtractor.extract(GenericJdbcExtractor.java:59)
684      at org.apache.sqoop.job.mr.SqoopMapper.runInternal(SqoopMapper.java:184)
685      at org.apache.sqoop.job.mr.SqoopMapper.run(SqoopMapper.java:81)
686      at org.apache.hadoop.mapred.MapTask.runNewMapper(MapTask.java:799)
687      at org.apache.hadoop.mapred.MapTask.run(MapTask.java)
688      at org.apache.hadoop.mapred.LocalJobRunner$Job$MapTaskRunnable.run(LocalJobRunner.java:271)
689      at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
690      at java.util.concurrent.FutureTask.run(FutureTask.java:266)
691      at org.apache.sqoop.submission.mapreduce.MapperExecutorGroup$1.lambda$execute$0(MapperExecutorGroup.java:222)
692      at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
693      at java.util.concurrent.FutureTask.run(FutureTask.java:266)
694      at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
695      at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
696      at java.lang.Thread.run(Thread.java:748)
697  Caused by: oracle.jdbc.OracleDatabaseException: ORA-01555: snapshot too old: rollback segment number 3 with name "_SYSSMU3_2097677531$" too small
698
699      at oracle.jdbc.driver.T4CTTIoer11.processERROR(T4CTTIoer11.java:498)
700      ... 28 common frames omitted
```

Fault Analysis

1. Each table contains an average of 200 to 500 million data records. Source data is updated every 10 minutes.

2. CDM does not support real-time migration but supports scheduled migration. As the source data is updated every 10 minutes, the source database has been updated before the migration job is complete, resulting in a rollback timeout.

3. If the entire table is queried during data migration and the query takes a long time, other users frequently perform the commit operation during the query, and the RBS of the Oracle database is small.

Solution

There are three solutions:

- Reduce the data volume queried each time.

- Increase the RBS of the Oracle data.

- Reduce frequent commit operations. This requires adjustment of the production service logic, which is almost impossible.

3.47 What Should I Do If the FTP Connectivity Test Fails and an Internal Server Error Is Reported?

Symptom

The root and FTPadmin users can log in to the ECS through FTP. Ports 21 and 20 of the security group are enabled. However, an internal server error is reported during the creation of an FTP link.

Solution

The background log shows that the error is a connection timeout, which may be caused by security group restrictions. Verify that the connectivity test is successful when the security group allows all traffic to pass through.

```
        at com.cmw.tomcat.common.RequestHeaderNameValve.invoke(RequestHeaderNameValve.java:60)
        at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:343)
        at org.apache.coyote.http11.Http11Processor.service(Http11Processor.java:616)
        at org.apache.coyote.AbstractProcessorLight.process(AbstractProcessorLight.java:65)
        at org.apache.coyote.AbstractProtocol$ConnectionHandler.process(AbstractProtocol.java:831)
        at org.apache.tomcat.util.net.NioEndpoint$SocketProcessor.doRun(NioEndpoint.java:1634)
        at org.apache.tomcat.util.net.SocketProcessorBase.run(SocketProcessorBase.java:49)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        at org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61)
        at java.lang.Thread.run(Thread.java:748)
2022-07-25 22:50:05.713|ERROR|https-jsse-nio2-172.22.61.85-21203-exec-6|[o.a.s.server.SqoopProtocolServlet:66]|Exception in POST https://100.78.233.110:21203/sqoop/v1/link
org.apache.sqoop.common.SqoopException: FTP_CONNECTOR_0600:Can't connect to the FTP server. Cause: Connection timed out (Connection timed out)
        at org.apache.sqoop.utils.ErrorResourceUtil.throwSqoopException(ErrorResourceUtil.java:31)
        at org.apache.sqoop.connector.ftp.ftpclient.FtpConnectorClient.connect(FtpConnectorClient.java:172)
        at org.apache.sqoop.connector.ftp.configuration.LinkConfig$ConfigValidator.validate(LinkConfig.java:55)
        at org.apache.sqoop.connector.ftp.configuration.LinkConfig$ConfigValidator.validate(LinkConfig.java:50)
        at org.apache.sqoop.validation.ConfigValidationRunner.executeValidator(ConfigValidationRunner.java:183)
        at org.apache.sqoop.validation.ConfigValidationRunner.validateArray(ConfigValidationRunner.java:149)
        at org.apache.sqoop.validation.ConfigValidationRunner.validateConfig(ConfigValidationRunner.java:130)
        at org.apache.sqoop.validation.ConfigValidationRunner.validate(ConfigValidationRunner.java:91)
        at org.apache.sqoop.model.ConfigUtils.validateConfigs(ConfigUtils.java:310)
        at org.apache.sqoop.handler.LinkRequestHandler.createUpdateLink(LinkRequestHandler.java:308)
        at org.apache.sqoop.handler.LinkRequestHandler.handleEvent(LinkRequestHandler.java:104)
        at org.apache.sqoop.server.v1.LinkServlet.handlePostRequest(LinkServlet.java:71)
        at org.apache.sqoop.server.SqoopProtocolServlet.doPost(SqoopProtocolServlet.java:61)
```

The firewall of the FTP server must allow the following communications so that the active FTP mode is available:

1. Any port greater than 1024 to port 21 of the FTP server (initial connection of the client).

2. Port 21 of the FTP server to a port greater than 1024 (control port for the server to respond to the client)

3. Port 20 of the FTP server to a port greater than 1024 (port for the communication between the initialized data link and the data port of the client)

4. Port greater than 1024 to port 20 of the FTP server (port through which the client sends ACK responses to the server)

3.48 What Should I Do If All Users Except User root Cannot Access RDS for MySQL?

Symptom

All users except user **root** cannot access RDS for MySQL.

Solution

1. Log in to the server, run the **mysql -u root –p** command to access the database, and enter the password.

2. Run the **mysql>use mysql** command.

3. Authorize users to access the database:

   For example, if you want to enable user **root** to access the MySQL server from any host using password 123456, run the following command:

   mysql>GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY '123456' WITH GRANT OPTION;

   If you want to enable user **abc** to access the MySQL server from a host whose IP address is 10.10.50.127 using password 654321, run the following command:

   mysql>GRANT ALL PRIVILEGES ON *.* TO 'abc'@'10.10.50.127' IDENTIFIED BY '654321' WITH GRANT OPTION;

4. Run the **mysql>FLUSH PRIVILEGES** command to refresh permissions.