Data Lake Insight

FAQs

Issue 01

Date 2025-01-09

HUAWEI TECHNOLOGIES CO., LTD.

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 DLI Basics

1.1 What Are the Differences Between DLI Flink and MRS Flink?

DLI Flink suits the cloud-native infrastructure. DLI Flink optimizes multiple core functions of the kernel engine. DLI provides an enterprise-level one-stop development platform with built-in development and O&M functions, freeing you from the trouble of O&M of user-built clusters. In addition to open-source connectors, DLI Flink also supports MySQL, GaussDB, MRS HBase, DMS, GaussDB(DWS) and OBS on the cloud, which allows you use DLI Flink right after service subscription. More importantly, DLI Flink jobs can automatically adapt to service requirements. Resources can be automatically and flexibly scaled to ensure service stability.

Table 1 compares the functions of DLI Flink and MRS Flink.

Table 1-1 Function comparison

Category	Features	DLI Flink	MRS Flink
Featured Capabilities	Deployment	Full hosting	Semi-hosting (cluster O&M required)
	Elastic scaling	• Container-based cluster deployment • Users can perform elastic scaling based on the service load and dynamically adjust the resources based on the job load. • Resources used by jobs can be dynamically adjusted based on job priorities.	Only YARN clusters are supported.

Category	Features	DLI Flink	MRS Flink
	Upstream and downstream data connection	● Open-source connectors and out-of-the-box connectors for data sources including databases (RDS and GaussDB), message queues (DMS), data warehouses (GuassDB DWS), and object storage (OBS). ● Higher usability and stability (in comparison with open-source connectors)	Open-source connectors only
Development and O&M	Monitoring and alarms	● Interconnection with Cloud Eye, a monitoring platform on Huawei Cloud, and SMN system on the cloud. Users can be notified of any events through emails, SMS messages, calls, and third-party office tools (webhook) ● Interconnection with the unified monitoring and alarm system (Prometheus) of companies ● Flink job rate, input and output data volume, operator backpressure, operator delay, CPU usage, and memory usage	Flink UI alarms only
	Multi-versioning	Different Flink versions can be used for jobs.	Jobs of different Flink versions are not supported by a Flink cluster.

Catego ry	Features	DLI Flink	MRS Flink
	Usability	Out-of-the-box, serverless architecture, and cross-AZ disaster recovery ● Users only need to write SQL statements and do not need to compile them. You only need to pay attention to the service code. ● You can use SQL statements to connect your jobs to various data sources, such as RDS, GaussDB(DWS), Kafka, and Elasticsearch. ● Users do not need to log in to the maintenance cluster. They can submit jobs in few clicks on the console. ● Checkpoints for Flink SQL jobs ● Flink job logs can be dumped and stored for job analysis.	Technical capabilities are required for coding, cluster setup, configuration, and O&M. ● Users need to write and compile code. ● Users need to log in to the cluster and run commands to submit the configuration. They need to maintain the cluster. ● Users need to code checkpoints to enable the function.
	Job templates	Preset general Flink SQL templates for quick start	N/A
Enterpri se security	Configuring access control	Permission control streamlined with Huawei Cloud IAM, and role-based access control	N/A
	Space isolation	Tenant-level and project-level isolation resources and code for efficient team works	N/A

1.2 What Are the Differences Between MRS Spark and DLI Spark?

Both DLI and MRS support Spark, but there are some differences in service mode, interface, application scenarios, and performance characteristics.

- The Spark component of DLI is fully managed. You can only use it through its APIs.

This mode reduces the O&M burden and allows you to focus more on data processing and analysis tasks.

For details, see **Data Lake Insight Developer Guide**.

- The Spark component of MRS is built on the VM in an MRS cluster. You can adjust and optimize Spark according to your actual needs and make API calls to use it.

  This mode provides higher freedom and customization, and is suitable for users with experience in big data processing.

  For details, see **MapReduce Service Developer Guide**.

1.3 How Do I Upgrade the Engine Version of a DLI Job?

DLI offers Spark and Flink compute engines, providing users with a one-stop serverless converged processing and analysis services for stream processing, batch processing, and interactive analysis. Currently, the recommended version for Flink compute engine is Flink 1.15, and for Spark compute engine is Spark 3.3.1.

The following walks you through on how to upgrade the engine version of a DLI job.

- **SQL job:**

  The engine version cannot be configured for SQL jobs. You need to create a queue to run SQL jobs, and the new queue will automatically use the latest version of the Spark engine.

- **Flink OpenSource SQL job:**

  a. Log in to the DLI management console.

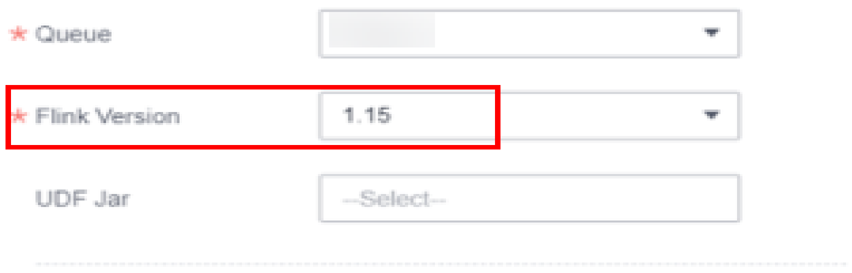
  b. In the navigation pane on the left, choose **Job Management** > **Flink Jobs**. In the job list, locate the desired Flink OpenSource SQL job.

  c. Click **Edit** in the **Operation** column.

  d. On the **Running Parameters** tab, select a new Flink version for **Flink Version**.

     When running a job using Flink 1.15 or later, you need to configure agency information on the **Runtime Configuration** tab. The key should be **flink.dli.job.agency.name** and the value should be the name of the agency. Failure to do so may affect the job's execution. For how to customize DLI agency permissions, see **Customizing DLI Agency Permissions**.

     For the syntax of Flink 1.15, see **Flink 1.15 Syntax Overview**.

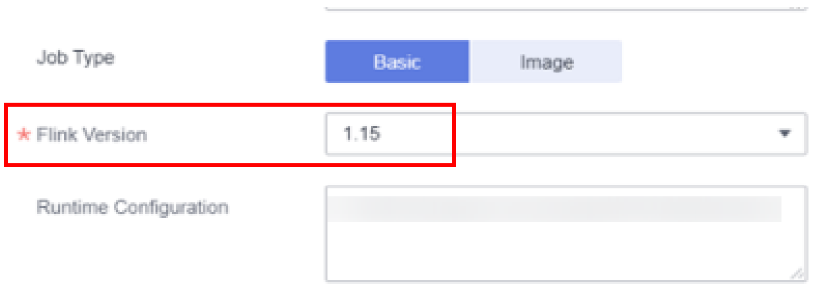


* Flink Version 1.15

- **Flink Jar job:**
  a. Log in to the DLI management console.
  b. In the navigation pane on the left, choose **Job Management** > **Flink Jobs**. In the job list, locate the desired Flink Jar job.
  c. Click **Edit** in the **Operation** column.
  d. In the parameter configuration area, select a new Flink version for **Flink Version**.

     When running a job using Flink 1.15 or later, you need to configure agency information in **Runtime Configuration**. The key should be **flink.dli.job.agency.name** and the value should be the name of the agency. Failure to do so may affect the job's execution. For how to customize DLI agency permissions, see **Customizing DLI Agency Permissions**.

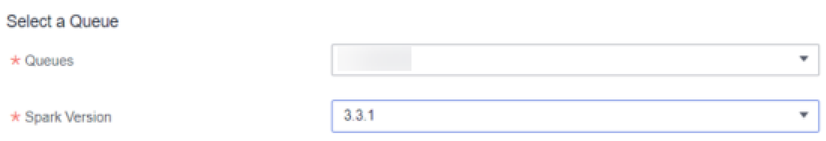
     For the syntax of Flink 1.15, see **Flink 1.15 Syntax Overview**.



Flink Version: 1.15

- **Spark Jar job:**
  a. Log in to the DLI management console.
  b. In the navigation pane on the left, choose **Job Management** > **Flink Jobs**. In the job list, locate the desired Spark Jar job.
  c. Click **Edit** in the **Operation** column.
  d. In the parameter configuration area, select a new Spark version for **Spark Version**.

     When running a job using Spark 3.3.1 or later, you need to configure a custom agency name in **Spark Arguments(--conf)**. Failure to do so may affect the job's execution. For how to customize DLI agency permissions, see **Customizing DLI Agency Permissions**.



Spark Version: 3.3.1

- **Learn more:**
  – **Flink Upgrade Guide**
  – **Flink 1.15 Release Notes**

1.4 Where Can Data Be Stored in DLI?

What Data Formats Can Be Stored in DLI?

Supported data formats:

- Parquet
- CSV
- ORC
- JSON
- Avro

Where Can Data Be Stored in DLI?

- OBS: Data used by SQL jobs, Spark jobs, and Flink jobs can be stored in OBS, reducing storage costs.
- DLI: The column-based **Parquet** format is used in DLI. That is, the data is stored in the **Parquet** format. The storage cost is relatively high.
- Datasource connection jobs can be stored in the connected services. Currently, CloudTable, CSS, DCS, DDS, GaussDB(DWS), MRS, and RDS are supported.

What Are the Differences Between DLI Tables and OBS Tables?

- DLI tables store data within the DLI service, keeping you unaware of the storage path.
- OBS tables store data in your OBS buckets, allowing you to manage source data files.

DLI tables offer advanced features like permission control and caching acceleration, providing better performance than foreign tables. However, they come with storage fees.

1.5 Can I Import OBS Bucket Data Shared by Other Tenants into DLI?

DLI supports importing data from OBS buckets shared by IAM users under the same tenant, but not from OBS buckets shared by other tenants.

This ensures data security and isolation.

For scenarios that require cross-tenant data sharing and analysis, you are advised to first mask the data and upload it to an OBS bucket before performing data analysis. After the analysis is completed, delete the temporary data in the OBS bucket in a timely manner to ensure data security.

1.6 Regions and AZs

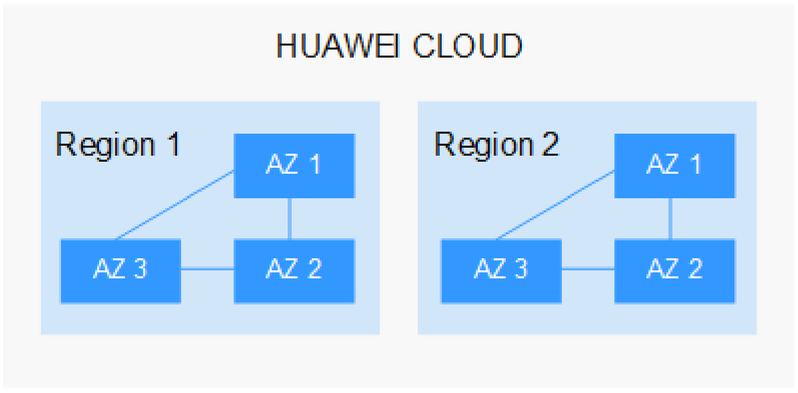
Concept

A region and availability zone (AZ) identify the location of a data center. You can create resources within a specific region and AZ.

- Regions are divided from the dimensions of geographical location and network latency. Public services, such as Elastic Cloud Server (ECS), Elastic Volume Service (EVS), Object Storage Service (OBS), Virtual Private Cloud (VPC), Elastic IP (EIP), and Image Management Service (IMS), are shared within the same region. Regions are classified as universal regions and dedicated regions. A universal region provides universal cloud services for common tenants. A dedicated region provides services of the same type only or for specific tenants.
- An AZ contains one or more physical data centers. Each AZ has independent cooling, fire extinguishing, moisture-proof, and electricity facilities. Within an AZ, computing, network, storage, and other resources are logically divided into multiple clusters. AZs within a region are interconnected using high-speed optical fibers to allow you to build cross-AZ high-availability systems.

Figure 1-1 shows the relationship between the regions and AZs.

Figure 1-1 Regions and AZs



Huawei Cloud provides services in many regions around the world. You can select a region and AZ as needed. For more information, see **Huawei Cloud Global Regions**.

Region Selection

When selecting a region, consider the following factors:

- Location

  You are advised to select a region close to you or your target users. This reduces network latency and improves access rate. However, Chinese mainland regions provide basically the same infrastructure, BGP network quality, as well as operations and configurations on resources. Therefore, if you or your target users are in the Chinese mainland, you do not need to consider the network latency differences when selecting a region.

The countries and regions outside the Chinese mainland, such as Bangkok and Hong Kong, provide services for users outside the Chinese mainland. If you or your target users are in the Chinese mainland, these regions are not recommended due to high access latency.

- If you or your target users are in Asia Pacific regions that are outside the Chinese mainland, select the **CN-Hong Kong**, **AP-Bangkok**, or **AP-Singapore** region.
- If you or your target users are in Africa, select the **AF-Johannesburg** region.
- If you or your target users are in Europe, select the **EU-Paris** region.

- Resource price

  Resource prices may vary in different regions. For details, see **Product Pricing Details**.

AZ Selection

When determining whether to deploy resources in the same AZ, consider your applications' requirements on disaster recovery (DR) and network latency.

- For high DR capability, deploy resources in different AZs in the same region.
- For low network latency, deploy resources in the same AZ.

Regions and Endpoints

Before using an API to call resources, specify its region and endpoint. For details about regions and endpoints, see **Regions and Endpoints**.

1.7 Can a Member Account Use Global Variables Created by Other Member Accounts?

No, a global variable can only be used by the user who created it. Global variables can be used to simplify complex parameters. For example, long and difficult variables can be replaced to improve the readability of SQL statements.

The restrictions on using global variables are as follows:

- Existing sensitive variables can only be used by their respective creators. Other common global variables are shared by users under the same account and project.
- If there are multiple global variables with the same name in the same project under an account, delete the redundant global variables to ensure that the global variables are unique in the same project. In this case, all users who have the permission to modify the global variables can change the variable values.
- If there are multiple global variables with the same name in the same project under an account, delete the global variables created by the user first. If there are only unique global variables, all users who have the delete permission can delete the global variables.

1.8 Is DLI Affected by the Apache Spark Command Injection Vulnerability (CVE-2022-33891)?

No.

The **spark.acls.enable** configuration item is not used in DLI. The Apache Spark command injection vulnerability (CVE-2022-33891) does not exist in DLI.

This vulnerability mainly affects data security by allowing the execution of commands with arbitrary usernames when ACL is enabled.

DLI was designed with data security and isolation in mind, and therefore, the relevant configuration items are not enabled, so it is not affected by this vulnerability.

1.9 How Do I Manage Jobs Running on DLI?

To manage a large number of DLI jobs, you can use the following methods:

- **Manage jobs by group.**

  Group tens of thousands of jobs by type and run each group on a queue.
- **Create IAM users.**

  Alternatively, create IAM users to execute different types of jobs.

  For how to create IAM users, see **Creating an IAM User**.

In addition, DLI also provides job management functions, including editing, starting, stopping, deleting, exporting, and importing jobs. You can use these functions to regularly maintain and manage jobs.

1.10 How Do I Change the Field Names of an Existing Table on DLI?

DLI does not support directly changing the field names of a table. However, you can solve this issue by migrating the table data using the following steps:

1. Create a table: Create a table and define new field names.
2. Migrate the data: Use the **INSERT INTO ... SELECT** statement to migrate the data from the old table to the new table.
3. Delete the old table: Once you have ensured that the new table has completely replaced the old table and the data migration is complete, you can delete the old table to avoid confusion.

2 DLI Elastic Resource Pools and Queues

2.1 How Can I Check the Actual and Used CUs for an Elastic Resource Pool as Well as the Required CUs for a Job?

In daily big data analysis work, it is important to allocate and manage compute resources properly to provide a good job execution environment.

You can allocate resources and adjust task execution order based on the job's compute needs and data scale, and schedule different elastic resource pools or queues to adapt to different workloads. To ensure normal job execution, the CUs required for the submitted job should be less than or equal to the remaining available CUs in the elastic resource pool.

This section describes how to view the usage of compute resources in an elastic resource pool and the required CUs for a job.

Checking the Actual and Used CUs for an Elastic Resource Pool

1. Log in to the DLI management console.
2. Choose **Resources** > **Resource Pool**.

   Locate the target resource pool in the list and check its **Actual CUs** and **Used CUs**.

   – **Actual CUs**: number of CUs that can be allocated in the elastic resource pool.

   – **Used CUs**: CUs that have been allocated to and used by the current elastic resource pool.

   To ensure normal job execution, the CUs required for the submitted job should be less than or equal to the remaining available CUs in the elastic resource pool.

   For details about the number of CUs required by different types of jobs, see **Checking the Required CUs for a Job**.

Checking the Required CUs for a Job

- **SQL job**:

  Use the monitoring dashboard provided by Cloud Eye to check the number of running and submitted jobs, and use the job count to determine the overall resource usage of SQL jobs.

- **Flink job**:

  a. Log in to the DLI management console.

  b. In the navigation pane on the left, choose **Job Management** > **Flink Jobs**.

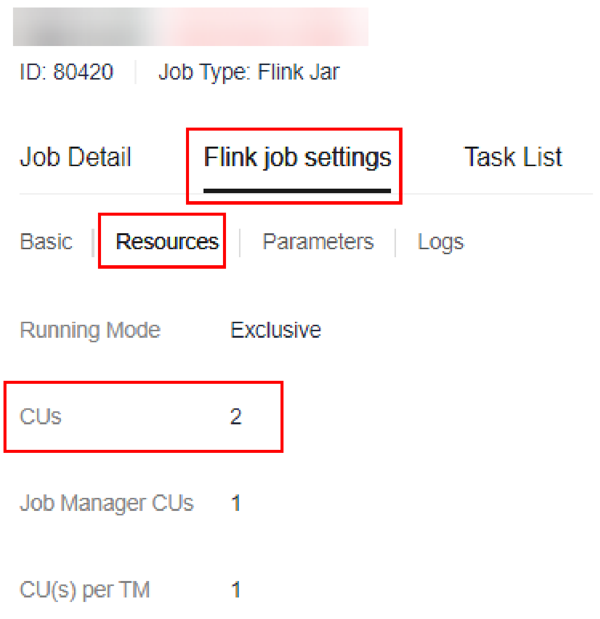
  c. In the job list, click the name of the target job.

  d. Click **Flink Job Settings** then **Resources**.

  e. Check the value of **CUs**, that is, the total number of CUs used by the job.

     You can set the number of CUs on the job editing page using the following formula: CUs = Job Manager CUs + (Parallelism/Slots per TM) x CUs per TM.

     Figure 2-1 Number of CUs required by a Flink job



ID: 80420 | Job Type: Flink Jar

Running Mode Exclusive
CUs 2
Job Manager CUs 1
CU(s) per TM 1

- **Spark job**:

  a. Log in to the DLI management console.

  b. In the navigation pane on the left, choose **Job Management** > **Spark Jobs**.

  c. Locate the target job in the list and click **Edit** in the **Operation** column.

     Check the compute resource specifications configured for the job.

     The formula is as follows:

     Number of CUs of a Spark job = Number of CUs used by executors + Number of CUs used by the driver

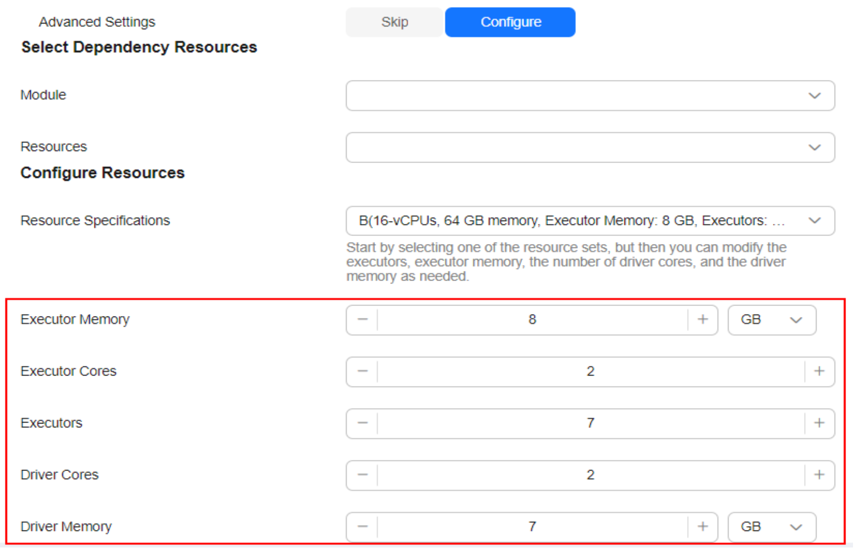
     Number of CUs used by executors = max {[(Executors x Executor Memory)/4], (Executors x Executor Cores)} x 1

Number of CUs used by the driver = max [(Driver Memory/4), Driver Cores] x 1

NOTE

- If **Advanced Settings** is set to **Skip** for a Spark job, resource specifications of type A are used by default.
- The unit of compute resource specifications for Spark jobs is CU. One CU consists of one CPU and 4 GB of memory. In the formulas above, x1 represents the conversion of CPU to CU.
- To calculate the required CUs for the executors or driver, use either the memory or the number of CPU cores. Choose the larger value between the two as the number of required CUs.

Figure 2-2 Number of CUs required by a Spark job



Advanced Settings: Skip / Configure

Select Dependency Resources

Module	
Resources	

Configure Resources

Resource Specifications	B(16-vCPUs, 64 GB memory, Executor Memory: 8 GB, Executors: ...
	Start by selecting one of the resource sets, but then you can modify the executors, executor memory, the number of driver cores, and the driver memory as needed.
Executor Memory	8 GB
Executor Cores	2
Executors	7
Driver Cores	2
Driver Memory	7 GB

2.2 How Do I Check for a Backlog of Jobs in the Current DLI Queue?

Symptom

You need to check the large number of jobs in the **Submitting** and **Running** states on the queue.

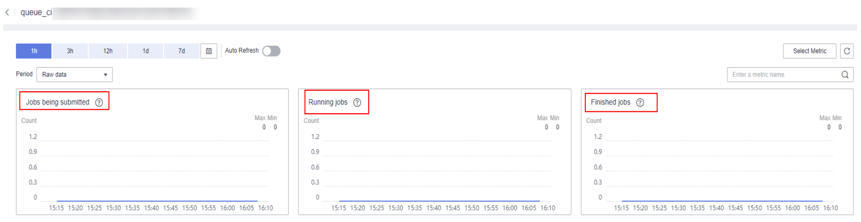
Solution

Use Cloud Eye to view jobs in different states on the queue. The procedure is as follows:

1. Log in the management console and search for Cloud Eye.
2. In the navigation pane on the left, choose **Cloud Service Monitoring** > **Data Lake Insight**.
3. On the **Cloud Service Monitoring** page, click the queue name.
4. On the monitoring page, view the following metrics to check the job status:

a. Number of jobs being submitted: Statistics of jobs in the **Submitting** state on the current queue
b. Number of running jobs: Statistics of jobs in the **Running** state on the current queue
c. Number of finished jobs: Statistics of jobs in the **Finished** state on the current queue

Figure 2-3 Viewing monitoring metrics of a queue



2.3 How Do I View the Load of a DLI Queue?

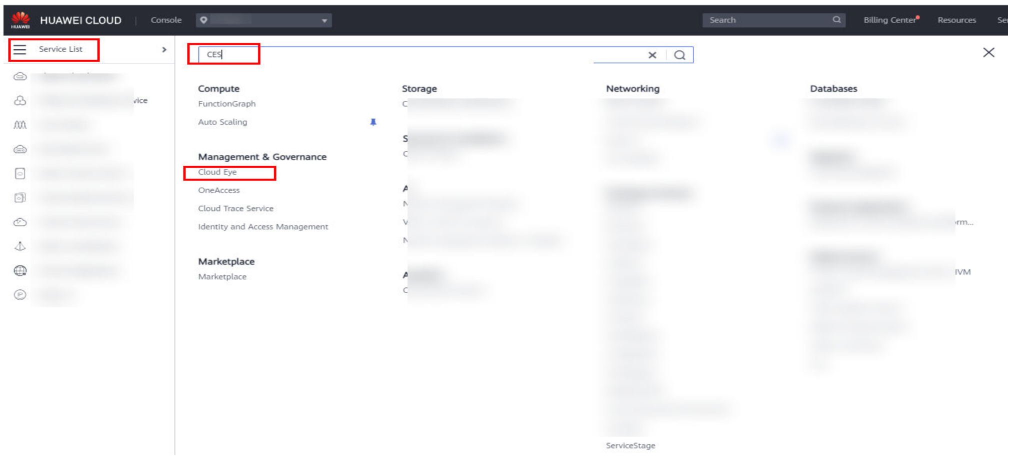
Scenario

To check the running status of the DLI queue and determine whether to run more jobs on that queue, you need to check the queue load.

Procedure

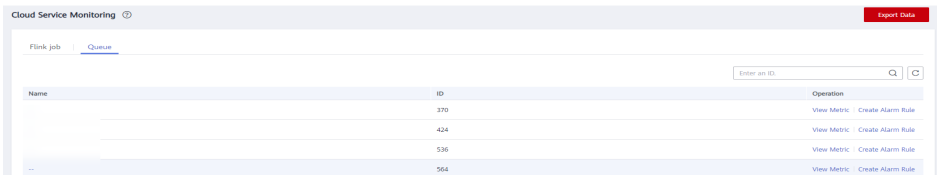
1. Search for Cloud Eye on the console.

Figure 2-4 Searching Cloud Eye



2. In the navigation pane on the left, choose **Cloud Service Monitoring** > **Data Lake Insight**.

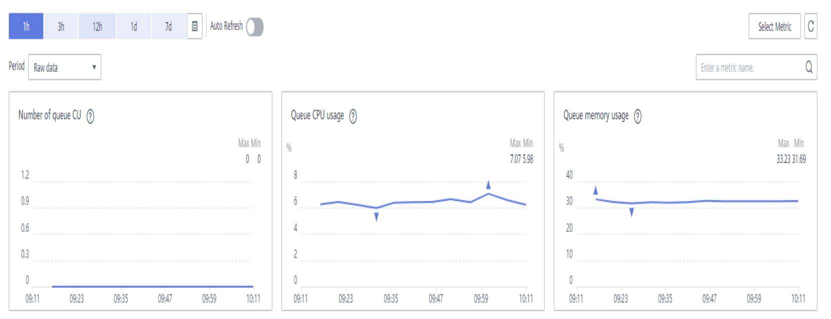
Figure 2-5 Cloud service monitoring



Name	ID	Operation
	370	View Metric | Create Alarm Rule
	424	View Metric | Create Alarm Rule
	536	View Metric | Create Alarm Rule
--	564	View Metric | Create Alarm Rule

3. Select the queue you want to view.

Figure 2-6 Viewing the queue load



2.4 How Do I Monitor Job Exceptions on a DLI Queue?

DLI allows you to subscribe to an SMN topic for failed jobs.

1. Log in to the DLI console.
2. In the navigation pane on the left, choose **Queue Management**.
3. On the **Queue Management** page, click **Create SMN Topic** in the upper left corner. For details about how to create a topic, see **Creating a Message Notification Topic** in *Data Lake Insight User Guide*.

2.5 How Do I Migrate an Old Version Spark Queue to a General-Purpose Queue?

Currently, DLI provides two types of queues, **For SQL** and **For general use**. SQL queues are used to run SQL jobs. General-use queues are compatible with Spark queues of earlier versions and are used to run Spark and Flink jobs.

You can perform the following steps to convert an old Spark queue to a general purpose queue.

1. Purchase a general purpose queue again.
2. Migrate the jobs in the old Spark queue to the new general queue. That is, specify a new queue when submitting Spark jobs.
3. Release the old Spark queue, that is, delete it or unsubscribe it from the queue.

2.6 How Do I Do If I Encounter a Timeout Exception When Executing DLI SQL Statements on the default Queue?

Symptom

After a SQL job was submitted to the default queue, the job runs abnormally. The job log reported that the execution timed out. The exception logs are as follows:

```
[ERROR] Execute DLI SQL failed. Please contact DLI service.
[ERROR] Error message:Execution Timeout
```

Possible Causes

The default queue is a public preset queue in the system for function trials. When multiple users submit jobs to this queue, traffic control might be triggered. As a result, the jobs fail to be submitted.

Solution

Buy a custom queue for your jobs.

For how to create a queue, see **Creating an Elastic Resource Pool and Adding Queues to the Pool**.

3 DLI Databases and Tables

3.1 Why Am I Unable to Query a Table on the DLI Console?

Symptom

A DLI table exists but cannot be queried on the DLI console.

Possible Causes

If a table exists but cannot be queried, there is a high probability that the current user does not have the permission to query or operate the table.

Solution

Contact the user who creates the table and obtain the required permissions. To assign permissions, perform the following steps:

1. Log in to the DLI management console as the user who creates the table. Choose **Data Management** > **Databases and Tables** form the navigation pane on the left.
2. Click the database name. The table management page is displayed. In the **Operation** column of the target table, click **Permissions**. The table permission management page is displayed.
3. Click **Set Permission**. In the displayed dialog box, set **Authorization Object** to **User**, set **Username** to the name of the user that requires the permission, and select the required permissions. For example, **Select Table** and **Insert** permissions.
4. Click **OK**.
5. Log in to the DLI console as the user that has been granted permission and check whether the table can be queried.

3.2 How Do I Do If the Compression Rate of an OBS Table Is High?

When submitting a job to import data into a DLI table, if the compression rate of the Parquet/ORC file corresponding to the OBS table is high, exceeding 5 times the compression rate, you can optimize the job performance by adjusting the configuration.

Specifically, configure **dli.sql.files.maxPartitionBytes=33554432** in the **conf** field of the submit-job request body.

The default value of this configuration item is 128 MB. Configuring it to 32 MB can reduce the amount of data read by a single task and avoid processing a large amount of data by a single task after decompression due to a high compression rate.

However, adjusting this parameter may affect the execution efficiency and resource consumption of the job. Therefore, when making adjustments, you need to choose a suitable parameter value based on the actual amount of data and compression rate.

3.3 How Do I Do If Inconsistent Character Encoding Leads to Garbled Characters?

To avoid garbled characters caused by inconsistent character encoding, you are advised to unify the encoding format of your data source when executing jobs in DLI.

DLI only supports UTF-8-encoded text, so your data needs to be encoded in UTF-8 when creating tables and importing data.

Before importing data into DLI, make sure that the source data file (such as CSV, JSON, etc.) is saved in UTF-8 encoding. If the data source is not encoded in UTF-8, convert it to UTF-8 encoding before importing.

3.4 Do I Need to to Regrant Permissions to Users and Projects After Deleting and Recreating a Table With the Same Name?

Scenario

User A created the **testTable** table in a database through a SQL job and granted user B the permission to insert and delete table data. User A deleted the **testTable** table and created a new **testTable** table. If user A wants user B to retain the insert and delete permission, user A needs to grant the permissions to user B again.

Possible Causes

After a table is deleted, the table permissions are not retained. You need to grant permissions to a user or project.

Solution

Operations to grant permissions to a user or project are as follows:

1. On the left of the management console, choose **Data Management** > **Databases and Tables**.
2. Click the database name whose table permission is to be granted. The table management page of the database is displayed.
3. Locate the row of the target table and click **Permissions** in the **Operation** column.
4. On the displayed page, click **Grant Permission** in the upper right corner.
5. In the displayed **Grant Permission** dialog box, select the required permissions.
6. Click **OK**.

3.5 How Do I Do If Files Imported Into a DLI Partitioned Table Lack Data for the Partition Columns, Causing Query Failures After the Import Is Completed?

Symptom

A CSV file is imported to a DLI partitioned table, but the imported file data does not contain the data in the partitioning column. The partitioning column needs to be specified for a partitioned table query. As a result, table data cannot be queried.

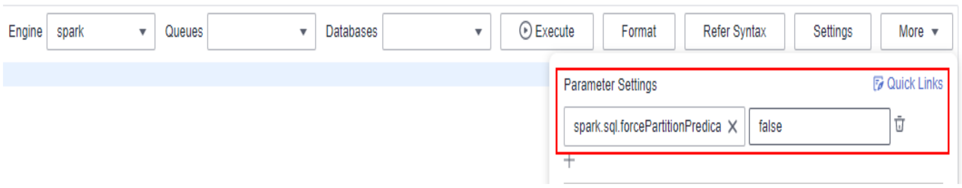
Possible Causes

When data is imported to a DLI partitionedtable, if the file data does not contain the partitioning column, the system specifies **__HIVE_DEFAULT_PARTITION__** as the column by default. If a Spark job finds that the partition is empty, **null** is returned.

Solution

1. Log in to the DLI management console. In the SQL editor, click **Settings**.
2. Add **spark.sql.forcePartitionPredicatesOnPartitionedTable.enabled** and set it to **false**.

Figure 3-1 Parameters



spark.sql.forcePartitionPredica false

3. Query the entire table or the partitioned table.

3.6 How Do I Fix Incorrect Data in an OBS Foreign Table Caused by Newline Characters in OBS File Fields?

Symptom

When an OBS foreign table is created, a field in the specified OBS file contains a carriage return line feed (CRLF) character. As a result, the data is incorrect.

The statement for creating an OBS foreign table is similar to the following:

```
CREATE TABLE test06 (name string, id int, no string) USING csv OPTIONS (path "obs://dli-test-001/
test.csv");
```

The file contains the following information (example):

```
Jordon,88,"aa
bb"
```

A carriage return exists between **aa** and **bb** in the last field. As a result, he data in the **test06** table is displayed as follows:

```
name    id    classno
Jordon    88    aa
bb"    null    null
```

Solution

When creating an OBS foreign table, set **multiLine** to **true** to specify that the column data contains CRLF characters. The following is an example to solve the problem:

```
CREATE TABLE test06 (name string, id int, no string) USING csv OPTIONS (path "obs://dli-test-001/
test.csv",multiLine=true);
```

3.7 How Do I Prevent a Cartesian Product Query and Resource Overload Due to Missing "ON" Conditions in Table Joins?

Symptom

The on clause was not added to the SQL statement for joining tables. As a result, the Cartesian product query occurs due to multi-table association, and the queue resources were used up. Job execution fails on the queue.

For example, the following SQL statement left-joins three tables without the on clause.

```
select
    case
      when to_char(from_unixtime(fs.special_start_time), 'yyyy-mm-dd') < '2018-10-12' and row_number()
over(partition by fg.goods_no order by fs.special_start_time asc) = 1 then 1
      when to_char(from_unixtime(fs.special_start_time), 'yyyy-mm-dd') >= '2018-10-12' and fge.is_new = 1
then 1
      else 0 end as is_new
from testdb.table1 fg
left join testdb.table2 fs
left join testdb.table3  fge
where to_char(from_unixtime(fs.special_start_time), 'yyyymmdd') = substr('20220601',1,8)
```

Solution

When you use join to perform multi-table query, you must use the on clause to reduce the data volume.

The following example uses the on clause for the table join, which greatly reduces the result set of associated query and improves the query efficiency.

```
select
     case
        when to_char(from_unixtime(fs.special_start_time), 'yyyy-mm-dd') < '2018-10-12' and row_number()
over(partition by fg.goods_no order by fs.special_start_time asc) = 1 then 1
        when to_char(from_unixtime(fs.special_start_time), 'yyyy-mm-dd') >= '2018-10-12' and fge.is_new = 1
then 1
        else 0 end as is_new
from testdb.table1 fg
left join testdb.table2 fs on fg.col1 = fs.col2
left join testdb.table3  fge on fg.col3 = fge.col4
where to_char(from_unixtime(fs.special_start_time), 'yyyymmdd') = substr('20220601',1,8)
```

3.8 How Do I Do If I Can't Query Data After Manually Adding It to the Partition Directory of an OBS Table?

Symptom

Partition data is manually uploaded to a partition of an OBS table. However, the data cannot be queried using DLI SQL editor.

Solution

After manually adding partition data, you need to update the metadata information of the OBS table. Run the following statement on desired table:

```
MSCK REPAIR TABLE table_name;
```

Query the data in the OBS partitioned table.

3.9 Why Does the "insert overwrite" Operation Affect All Data in a Partitioned Table Instead of Just the Targeted Partition?

When using the **INSERT OVERWRITE** statement to overwrite data in a partitioned table, if you find that it overwrites all data instead of the expected partitioned data, it may be because the dynamic partition overwrite feature is not enabled.

To dynamically overwrite partitioned data specified in a datasource table, you need to first set **dli.sql.dynamicPartitionOverwrite.enabled** to **true**, and then run the **INSERT OVERWRITE** statement.

The default value of **dli.sql.dynamicPartitionOverwrite.enabled** is **false**. If unset, data in the entire table is overwritten.

For details, see **Inserting Data**.

3.10 Why Does the "create_date" Field in an RDS Table (Datetime Data Type) Appear as a Timestamp in DLI Queries?

Spark does not have the datetime type and uses the TIMESTAMP type instead.

You can use a function to convert data types.

The following is an example.

select cast(create_date as string), * from table where create_date>'2221-12-01 00:00:00';

For details about the TIMESTAMP type, see **TIMESTAMP**.

3.11 How Do I Do If Renaming a Table After a SQL Job Causes Incorrect Data Size?

Changing a table name immediately after executing a SQL job may result in an incorrect data size for the table.

This is because DLI updates the metadata of the table when executing a SQL job. If the table name is changed before the job is completed, it conflicts with the metadata update process of the job, which affects the determination of the data size.

To avoid this situation, you are advised to wait for 5 minutes after the SQL job is executed before changing the table name. This ensures that the system has enough time to update the metadata of the table, avoiding inaccurate data size statistics caused by changing the table name.

3.12 How Can I Resolve Data Inconsistencies When Importing Data from DLI to OBS?

Symptom

When DLI is used to insert data into an OBS temporary table, only part of data is imported.

Possible Causes

Possible causes are as follows:

- The amount of data read during job execution is incorrect.
- The data volume is incorrectly verified.

Run a query statement to check whether the amount of imported data is correct.

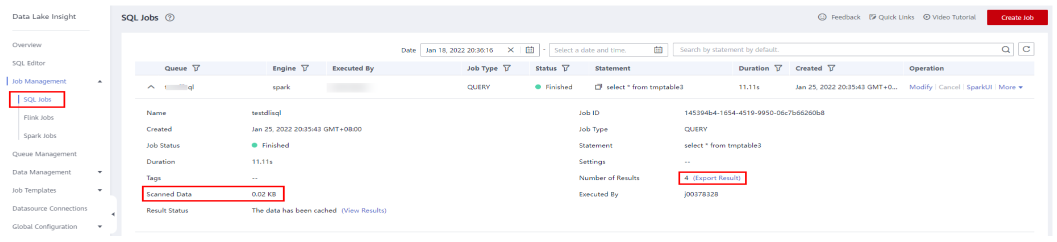
If OBS limits the number of files to be stored, add **DISTRIBUTE BY number** to the end of the insert statement.

For example, if **DISTRIBUTE BY 1** is added to the end of the insert statement, multiple files generated by multiple tasks can be inserted into one file.

Procedure

Step 1 On the DLI management console, check whether the number of results in the SQL job details is correct. The check result shows that the amount of data is correct.

Figure 3-2 Checking the amount of data



Date Jan 18, 2022 20:36:16

Queue	Engine	Executed By	Job Type	Status	Statement	Duration	Created	Operation
[redacted]ql	spark	[redacted]	QUERY	● Finished	select * from tmptable3	11.11s	Jan 25, 2022 20:35:43 GMT+0...	Modify | Cancel | SparkUI | More

Name	testdlisql	Job ID	145394b4-1654-4519-9950-06c7b66260b8
Created	Jan 25, 2022 20:35:43 GMT+08:00	Job Type	QUERY
Job Status	● Finished	Statement	select * from tmptable3
Duration	11.11s	Settings	--
Tags	--	Number of Results	4 (Export Result)
Scanned Data	0.02 KB	Executed By	j00378328
Result Status	The data has been cached (View Results)		

Step 2 Check whether the method to verify the data volume is correct. Perform the following steps to verify the data amount:

1. Download the data file from OBS.
2. Use the text editor to open the data file. The data volume is less than the expected volume.

If you used this method, you can verify that the text editor cannot read all the data.

Run the query statement to view the amount of data import into the OBS bucket. The query result indicates that all the data is imported.

This issue is caused by incorrect verification of the data volume.

----End

Related Information

For details about the SQL syntax for inserting data, see **Inserting Data** in *Data Lake Insight Spark SQL Syntax Reference*.

4 Enhanced Datasource Connections

4.1 How Do I Do If I Can't Bind an Enhanced Datasource Connection to a Queue?

Symptom

An enhanced datasource connection failed to pass the network connectivity test. Datasource connection cannot be bound to a queue. The following error information is displayed:

```
Failed to get subnet 86ddcf50-233a-449d-9811-cfef2f603213. Response code : 404, message :
{"code":"VPC.0202","message":"Query resource by id 86ddcf50-233a-449d-9811-cfef2f603213 fail.the
subnet could not be found."}
```

Cause Analysis

VPC Administrator permissions are required to use the VPC, subnet, route, VPC peering connection, and port for DLI datasource connections.

The binding fails because the user does not have the required VPC permissions.

Procedure

Step 1 Log in to the DLI management console. In the navigation pane on the left, choose **Global Configuration** > **Service Authorization**.

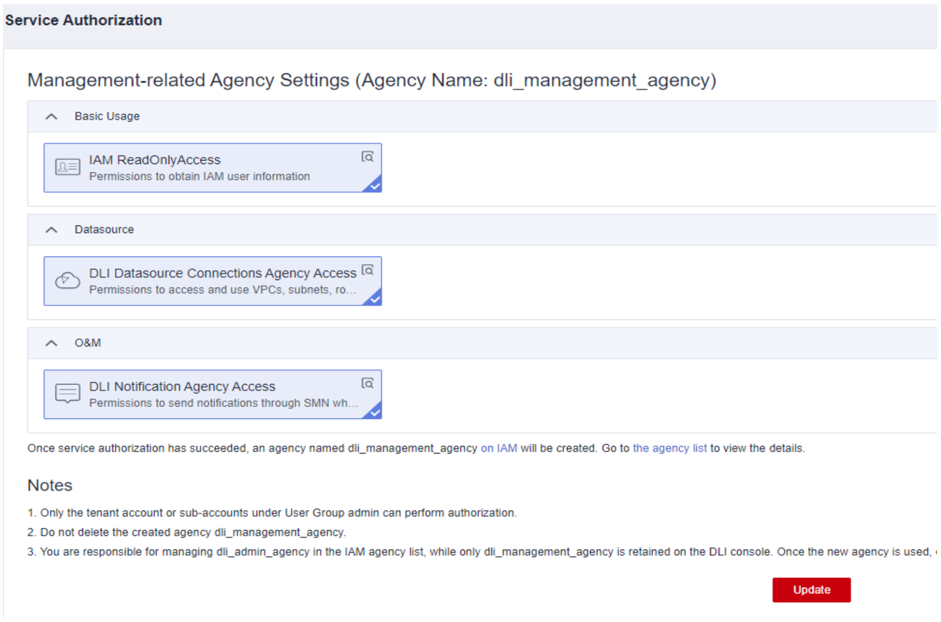
Step 2 On the agency settings page, select required permissions.

DLI Datasource Connections Agency Access is the permissions to access and use VPCs, subnets, routes, and VPC peering connections in datasource scenarios.

For more information, see **DLI Agency Permissions**.

Step 3 Select the permissions to be included in **dli_management_agency** and click **Update**.

Figure 4-1 Updating agency permissions



Service Authorization

Management-related Agency Settings (Agency Name: dli_management_agency)

Basic Usage

IAM ReadOnlyAccess
Permissions to obtain IAM user information

Datasource

DLI Datasource Connections Agency Access
Permissions to access and use VPCs, subnets, ro...

O&M

DLI Notification Agency Access
Permissions to send notifications through SMN wh...

Once service authorization has succeeded, an agency named dli_management_agency on IAM will be created. Go to the agency list to view the details.

Notes

1. Only the tenant account or sub-accounts under User Group admin can perform authorization.
2. Do not delete the created agency dli_management_agency.
3. You are responsible for managing dli_admin_agency in the IAM agency list, while only dli_management_agency is retained on the DLI console. Once the new agency is used,

Step 4 After updating the agency, recreate the datasource connection and rerun the job.

----End

4.2 How Do I Resolve a Failure in Connecting DLI to GaussDB(DWS) Through an Enhanced Datasource Connection?

Symptom

The outbound rule had been configured for the security group of the queue associated with the enhanced datasource connection. The datasource authentication used a password. The connection failed and **DLI.0999: PSQLException: The connection attempt failed** is reported.

Cause Analysis

Possible causes are as follows:

- The security group configuration is incorrect.
- The subnet configuration is incorrect.

Procedure

Step 1 Check whether the security group is accessible.

- Inbound rule: Check whether the inbound CIDR block and port in the security group have been enabled. If not, create the CIDR block and port you need.
- Outbound rule: Check whether the CIDR block and port of the outbound rule are enabled. (It is recommended that all CIDR blocks be enabled.)

Both the inbound and outbound rules of the security group are configured for the subnets of the DLI queue. Set the source IP address in the inbound direction to 0.0.0.0/0 and port 8000, indicating that any IP address can access port 8000.

Step 2 If the fault persists, check the subnet configuration. Check the network ACL associated with the GaussDB(DWS) subnet. A network ACL is an optional layer of security for your subnets. You can associate one or more subnets with a network ACL to control traffic in and out of the subnets. After the association, the network ACL denies all traffic to and from the subnet by default until you add rules to allow traffic. The check result showed that the ACL associated with the subnet where GaussDB(DWS) resides is empty.

A network ACL is associated and no inbound or outbound rules are configured. As a result, the IP address cannot be accessed.

Step 3 Perform the connectivity test. After the subnet inbound and outbound rules are configured, the datasource connection passes the connectivity test.

----End

Related Information

For details about the inbound and outbound rules, see **Why Is the Error Message "communication link failure" Displayed When I Use a Newly Activated Datasource Connection?**

4.3 How Do I Do If the Datasource Connection Is Successfully Created but the Network Connectivity Test Fails?

Symptom

A datasource connection is created and bound to a queue. The connectivity test fails and the following error information is displayed:

```
failed to connect to specified address
```

Fault Locating

The issues here are described in order of how likely they are to occur.

Troubleshoot the issue by ruling out the causes described here, one by one.

- **Check Whether a Port Number Is Added to the End of the Domain Name or IP Address**
- **Check Whether the Information of the Peer VPC and Subnet Are Correct.**
- **Check Whether the CIDR Block of the Queue Overlaps with That of the Data Source**
- **Check Whether the DLI Datasource Connections Agency Access Permission Is Granted to DLI**
- **Check Whether the Destination Security Group Allows Access from the CIDR Block of the Queue**

- **Check the Route Information of the VPC Peering Connection Corresponding to an Enhanced Datasource Connection**
- **Check Whether VPC Network ACL Rules Are Configured to Restrict Network Access**

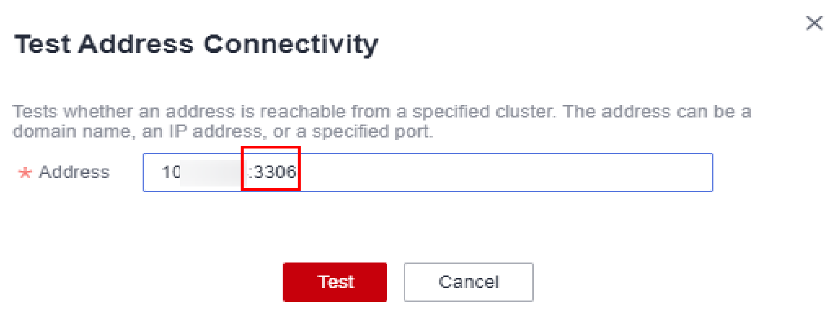
Check Whether a Port Number Is Added to the End of the Domain Name or IP Address

The port number is required for the connectivity test.

The following example tests the connectivity between a queue and a specified RDS DB instance. The RDS DB instance uses port 3306.

The following figure shows how you should specify the IP address.

Figure 4-2 Testing address connectivity



Test Address Connectivity

Tests whether an address is reachable from a specified cluster. The address can be a domain name, an IP address, or a specified port.

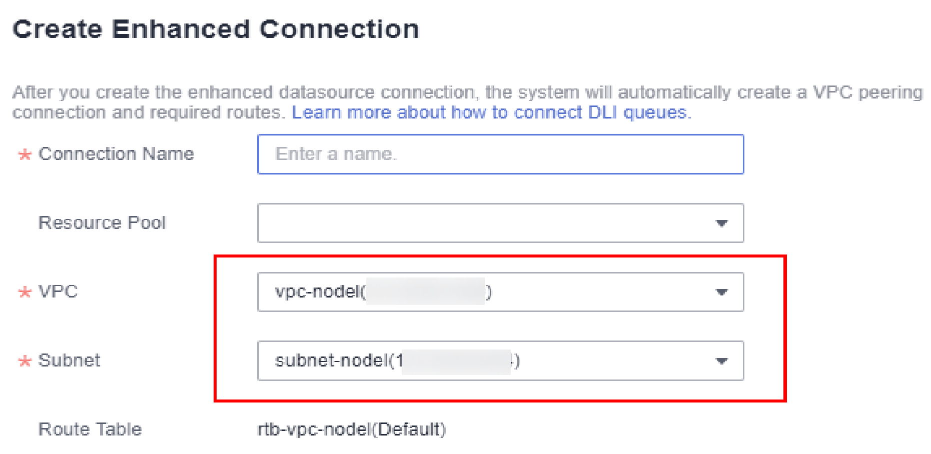
* Address	10 :3306

Check Whether the Information of the Peer VPC and Subnet Are Correct.

When you create an enhanced datasource connection, you need to specify the peer VPC and subnet.

For example, to test the connectivity between a queue and a specified RDS DB instance, you need to specify the RDS VPC and subnet information.

Figure 4-3 Creating a connection



Create Enhanced Connection

After you create the enhanced datasource connection, the system will automatically create a VPC peering connection and required routes. Learn more about how to connect DLI queues.

* Connection Name	Enter a name.
Resource Pool	
* VPC	vpc-nodel( )
* Subnet	subnet-nodel(1)
Route Table	rtb-vpc-nodel(Default)

Check Whether the CIDR Block of the Queue Overlaps with That of the Data Source

The CIDR block of the DLI queue bound with a datasource connection cannot overlap the CIDR block of the data source.

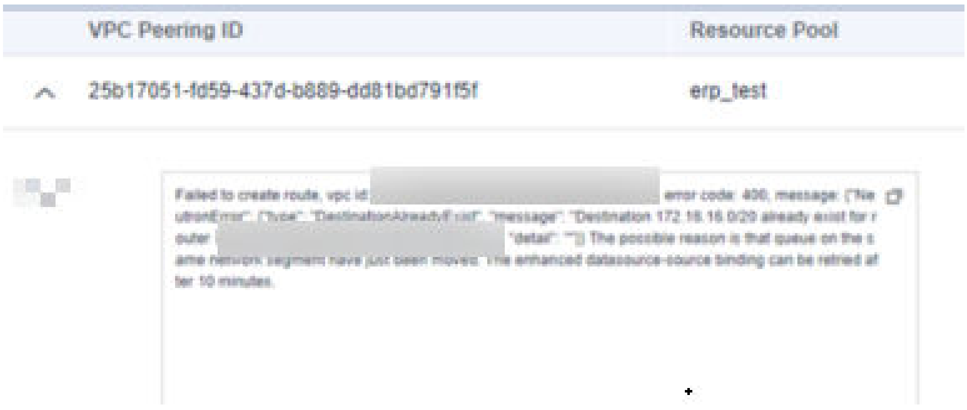
You can check whether they overlap by viewing the connection logs.

Figure 4-4 shows an example where the CIDR block conflicts of queue A and queue B. In this example, queue B is bound to an enhanced datasource connection to data source C. Therefore, a message is displayed, indicating that the network segment of queue A conflicts with that of data source C. As a result, a new enhanced datasource connection cannot be established.

Solution: Modify the CIDR block of the queue or create another queue.

Planing the CIDR blocks for your queues helps you to avoid this problem.

Figure 4-4 Viewing connection logs



VPC Peering ID	Resource Pool
25b17051-fd59-437d-b889-dd81bd791f5f	erp_test

```
Failed to create route, vpc id: [redacted] error code: 400, message: {"Ne
utronError": {"type": "DestinationAlreadyExist", "message": "Destination 172.16.16.0/20 already exist for r
outer [redacted] "detail": ""}} The possible reason is that queue on the s
ame network segment have just been moved. The enhanced datasource-source binding can be retried af
ter 10 minutes.
```

Check Whether the DLI Datasource Connections Agency Access Permission Is Granted to DLI

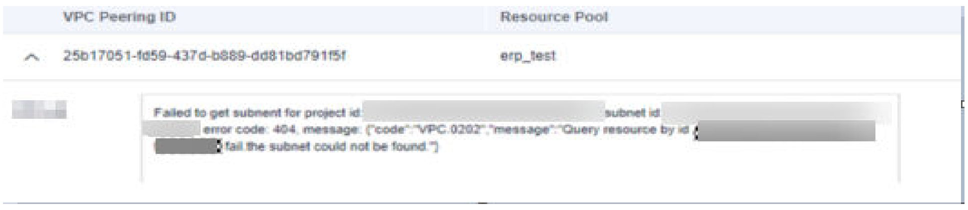
You can determine if a connection failure is due to insufficient permissions by checking the connection logs.

Figure 4-5 and Figure 4-6 show the logs when subnet ID and route ID of the destination cannot be obtained because there is no permission.

Solution: Add the **DLI Datasource Connections Agency Access** authorization on the **Global Configuration** > **Service Authorization** page.

See **Updating Agency Permissions**.

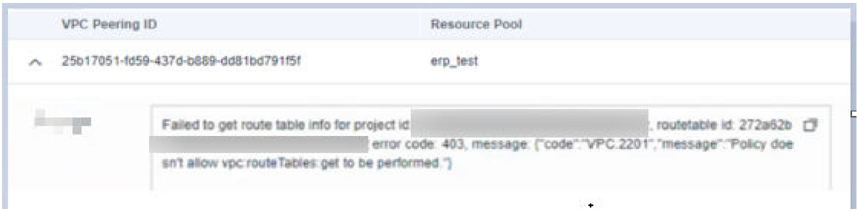
Figure 4-5 Viewing connection logs



VPC Peering ID	Resource Pool
25b17051-fd59-437d-b889-dd81bd791f5f	erp_test

```
Failed to get subnent for project id: [redacted] subnet id: [redacted]
error code: 404, message: {"code":"VPC.0202","message":"Query resource by id [redacted]
[redacted] fail.the subnet could not be found."}
```

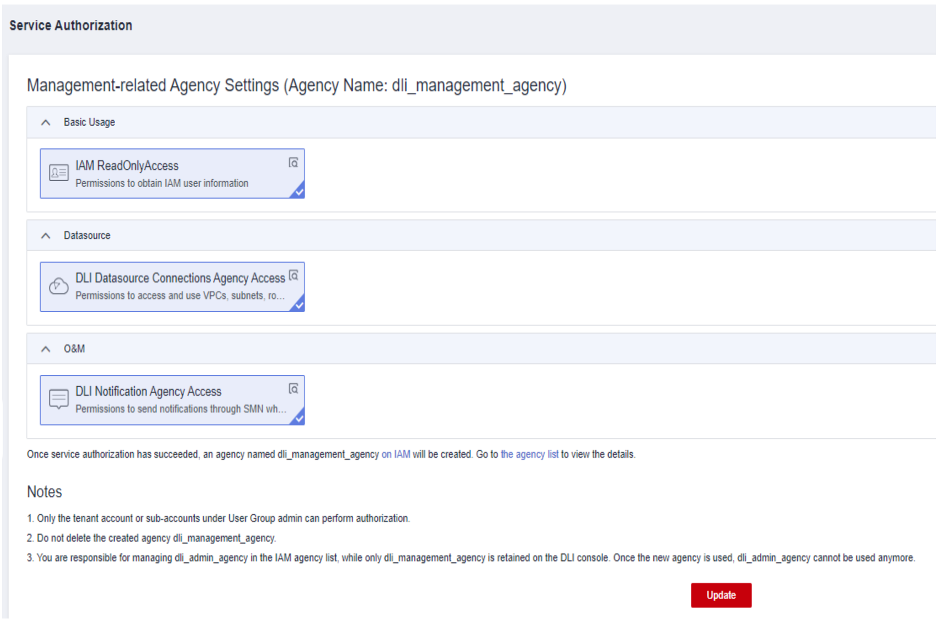
Figure 4-6 Viewing connection logs



VPC Peering ID	Resource Pool
25b17051-fd59-437d-b889-dd81bd791f5f	erp_test

```
Failed to get route table info for project id [redacted], routetable id: 272a62b
[redacted] error code: 403, message: {"code":"VPC.2201","message":"Policy doe
sn't allow vpc:routeTables:get to be performed."}
```

Figure 4-7 Selecting VPC administrator



Service Authorization

Management-related Agency Settings (Agency Name: dli_management_agency)

Basic Usage

IAM ReadOnlyAccess
Permissions to obtain IAM user information

Datasource

DLI Datasource Connections Agency Access
Permissions to access and use VPCs, subnets, ro...

O&M

DLI Notification Agency Access
Permissions to send notifications through SMN wh...

Once service authorization has succeeded, an agency named dli_management_agency on IAM will be created. Go to the agency list to view the details.

Notes

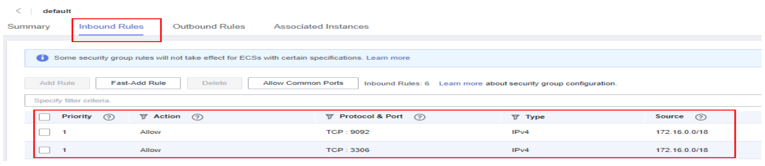
1. Only the tenant account or sub-accounts under User Group admin can perform authorization.
2. Do not delete the created agency dli_management_agency.
3. You are responsible for managing dli_admin_agency in the IAM agency list, while only dli_management_agency is retained on the DLI console. Once the new agency is used, dli_admin_agency cannot be used anymore.

Check Whether the Destination Security Group Allows Access from the CIDR Block of the Queue

To connect to Kafka, GaussDB(DWS), and RDS instances, add security group rules for the DLI CIDR block to the security group where the instances belong. For example, to connect a queue to RDS, perform the following operations:

1. Log in to the DLI console, choose **Resources** > **Queue Management** in the navigation pane on the left. On the displayed page, select the target queue, and click ⌄ to expand the row containing the target queue to view its CIDR block.
2. On the **Instance Management** page of the RDS console, click the instance name. In the **Connection Information** area, locate **Database Port** to obtain the port number of the RDS DB instance.
3. In the **Connection Information** area locate the **Security Group** and click the group name to switch to the security group management page. Select the **Inbound Rules** tab and click **Add Rule**. Set the priority to 1, protocol to TCP, port to the database port number, and source to the CIDR block of the DLI queue. Click **OK**.

Figure 4-8 VPC security group rules



Some security group rules will not take effect for ECSs with certain specifications. Learn more

Inbound Rules: 6 Learn more about security group configuration.

Specify filter criteria.

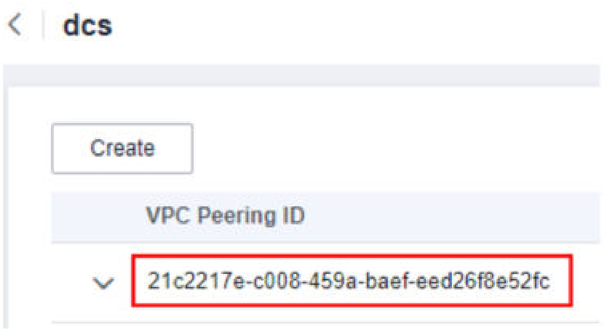
Priority	Action	Protocol & Port	Type	Source
1	Allow	TCP : 9092	IPv4	172.16.0.0/18
1	Allow	TCP : 3306	IPv4	172.16.0.0/18

Check the Route Information of the VPC Peering Connection Corresponding to an Enhanced Datasource Connection

Check the routing table of the VPC peering connection corresponding to the enhanced datasource connection. Check whether the CIDR block of the queue overlaps other CIDR blocks in the routing table. If it does, the forwarding may be incorrect.

1. Obtain the ID of the VPC peering connection created for the enhanced datasource connection.

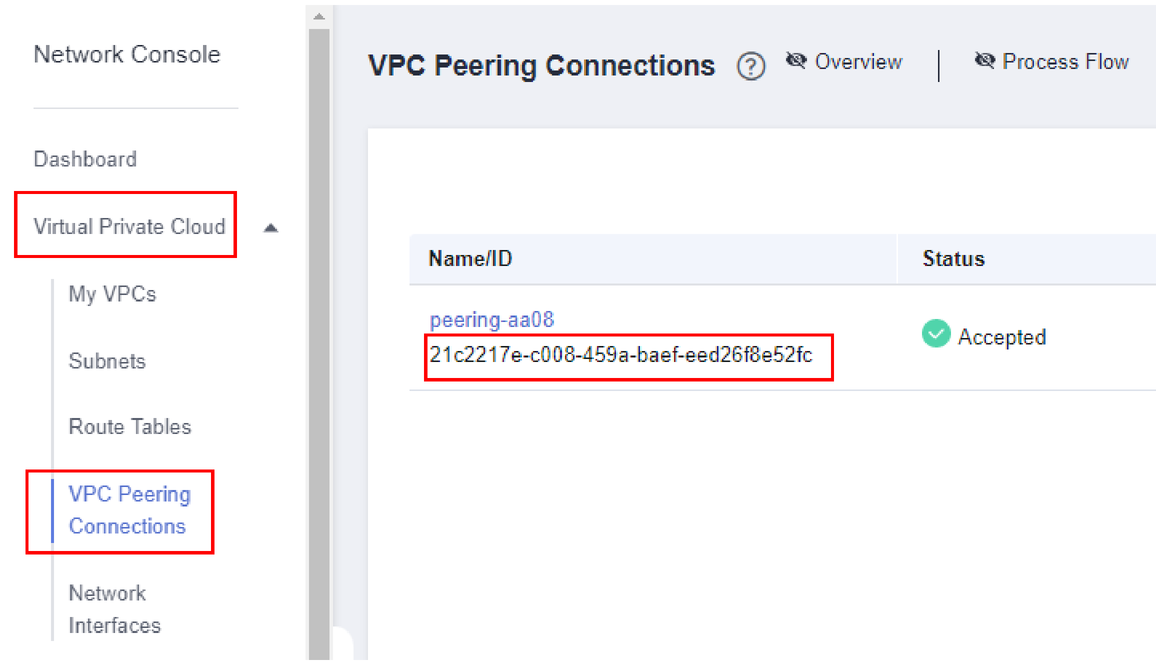
Figure 4-9 Obtaining the VPC peering connection ID



VPC Peering ID
21c2217e-c008-459a-baef-eed26f8e52fc

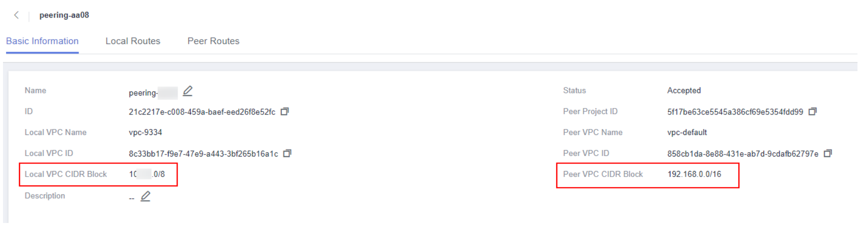
2. View the information about the VPC peering connection on the VPC console.

Figure 4-10 Viewing a VPC peering connection



Name/ID	Status
peering-aa08 21c2217e-c008-459a-baef-eed26f8e52fc	Accepted

Figure 4-11 Viewing the CIDR block of the queue

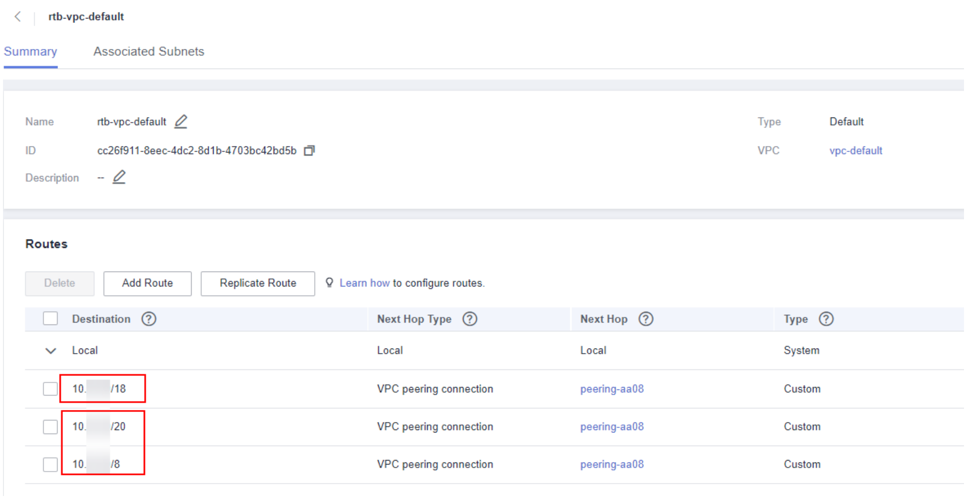


peering-aa08

Name	peering-[redacted]	Status	Accepted
ID	21c2217e-c008-459a-baef-eed26f8e52fc	Peer Project ID	5f17be63ce5545a386cf69e5354fdd99
Local VPC Name	vpc-9334	Peer VPC Name	vpc-default
Local VPC ID	8c33bb17-f9e7-47e9-a443-3bf265b16a1c	Peer VPC ID	858cb1da-8e88-431e-ab7d-9cdafb62797e
Local VPC CIDR Block	1[redacted].0/8	Peer VPC CIDR Block	192.168.0.0/16
Description	--		

3. View the route table information of the VPC corresponding to the queue.

Figure 4-12 Viewing the destination addresses in the routing table



rtb-vpc-default

Name	rtb-vpc-default	Type	Default
ID	cc26f911-8eec-4dc2-8d1b-4703bc42bd5b	VPC	vpc-default
Description	--		

Routes

Destination	Next Hop Type	Next Hop	Type
Local	Local	Local	System
10.[redacted]/18	VPC peering connection	peering-aa08	Custom
10.[redacted]/20	VPC peering connection	peering-aa08	Custom
10.[redacted]/8	VPC peering connection	peering-aa08	Custom

Check Whether VPC Network ACL Rules Are Configured to Restrict Network Access

Check whether an ACL is configured for the subnet corresponding to the datasource connection and whether the ACL rules restrict network access.

For example, if you set a CIDR block whose security group rule allows access from a queue and set a network ACL rule to deny access from that CIDR block, the security group rule does not take effect.

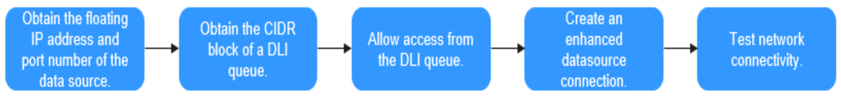
4.4 How Do I Configure Network Connectivity Between a DLI Queue and a Data Source?

- **Configuring the Connection Between a DLI Queue and a Data Source in a Private Network**

  If your DLI job needs to connect to a data source, for example, MRS, RDS, CSS, Kafka, or GaussDB(DWS), you need to enable the network between DLI and the data source.

  An enhanced datasource connection uses VPC peering to directly connect the VPC networks of the desired data sources for point-to-point data exchanges.

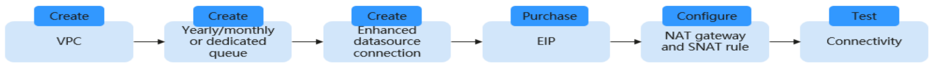
Figure 4-13 Configuration process


Obtain the floating IP address and port number of the data source.
Obtain the CIDR block of a DLI queue.
Allow access from the DLI queue.
Create an enhanced datasource connection.
Test network connectivity.

- **Configuring the Connection Between a DLI Queue and a Data Source in the Internet**

  You can configure SNAT rules and add routes to the public network to enable communications between a queue and the Internet.

Figure 4-14 Configuration process


Create
VPC
Create
Yearly/monthly or dedicated queue
Create
Enhanced datasource connection
Purchase
EIP
Configure
NAT gateway and SNAT rule
Test
Connectivity

4.5 Why Is Creating a VPC Peering Connection Necessary for Enhanced Datasource Connections in DLI?

The main reason for creating a VPC peering connection for DLI's enhanced datasource connection is to establish network connectivity between DLI and data sources in different VPCs.

When DLI needs to access external data sources located in different VPCs, it cannot directly read the data due to network isolation. By creating an enhanced datasource connection, a VPC peering connection can be used to bridge the VPC networks of DLI and the data sources, enabling data exchange and cross-source analysis.

Advantages of enhanced datasource connections:

- Network connectivity: Directly connect DLI to the target data source's VPC network for data exchange.
- Support for multiple data sources: Connect DLI to multiple data sources, such as GaussDB(DWS), RDS, CSS, and DCS.

4.6 How Do I Do If Creating a Datasource Connection in DLI Gets Stuck in the "Creating" State When Binding It to a Queue?

The possible causes and solutions are as follows:

- If you have created a queue, do not bind it to a datasource connection immediately. Wait for 5 to 10 minutes. After the cluster is started in the background, the queue can be bound to the datasource connection.
- If you have changed the network segment of a queue, do not bind it to a datasource connection immediately. Wait for 5 to 10 minutes. After the cluster is re-created in the background, the creation is successful.

4.7 How Do I Resolve the "communication link failure" Error When Using a Newly Created Datasource Connection That Appears to Be Activated?

Possible Causes

The network connectivity is abnormal. Check whether the security group is correctly selected and whether the VPC is correctly configured.

Solution

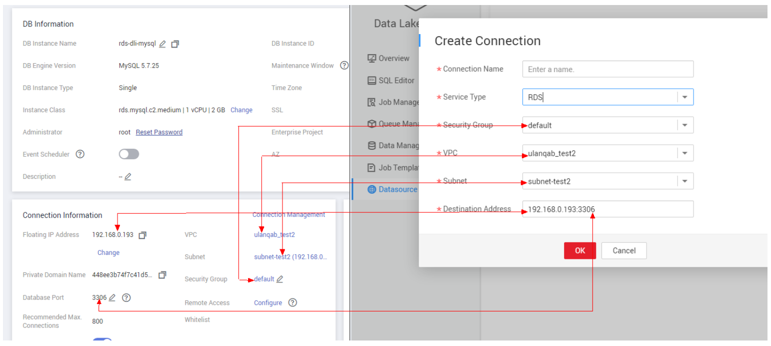
Example: When you create an RDS datasource connection, the system displays the error message **Communication link failure**.

1. Delete the original datasource connection and create a new one. When you create a connection, ensure that the selected **Security Group**, **VPC**, **Subnet**, and **Destination Address** are the same as those in RDS.

   NOTE

   Select a correct **Service Type**. In this example, select **RDS**.

   Figure 4-15 Creating a basic datasource connection-RDS

   DB Information

DB Instance Name	rds-dli-mysql	DB Instance ID	
DB Engine Version	MySQL 5.7.25	Maintenance Window	
DB Instance Type	Single	Time Zone	
Instance Class	rds.mysql.c2.medium | 1 vCPU | 2 GB Change	SSL	
Administrator	root Reset Password	Enterprise Project	
Event Scheduler		AZ	
Description	--		

   Connection Information

Floating IP Address	192.168.0.193 Change	VPC	ulanqab_test2
		Subnet	subnet-test2 (192.168.0...
Private Domain Name	448ee3b74f7c41d5...	Security Group	default
Database Port	3306	Remote Access	Configure
Recommended Max. Connections	800	Whitelist	

   Create Connection

* Connection Name	Enter a name.
* Service Type	RDS
* Security Group	default
* VPC	ulanqab_test2
* Subnet	subnet-test2
* Destination Address	192.168.0.193:3306

2. Check the configurations of VPC.

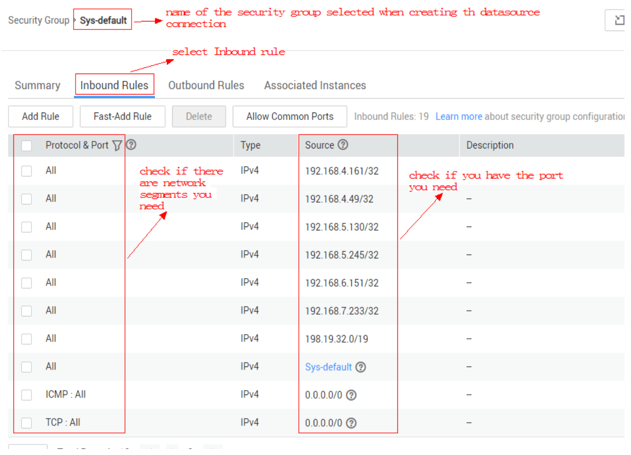
   If the error message is still displayed after you create a datasource connection according to **Step 1**, check the VPC configuration.

   – Classic datasource connections:

     ■ Inbound rule: Check whether the inbound CIDR block and port in the security group have been enabled. If not, create the CIDR block and port you need.

       Check whether the CIDR block and port are configured.

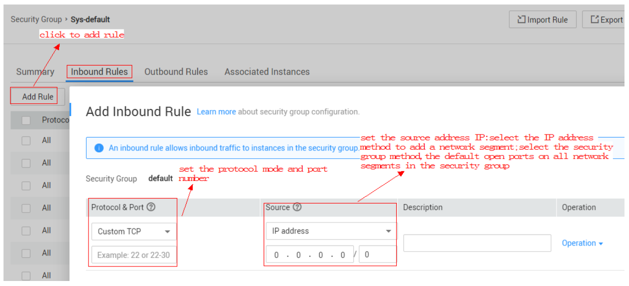
Figure 4-16 Checking whether the CIDR block and port are configured


Security Group › Sys-default
name of the security group selected when creating th datasource connection
select Inbound rule
Summary
Inbound Rules
Outbound Rules
Associated Instances
Add Rule
Fast-Add Rule
Delete
Allow Common Ports
Inbound Rules: 19 Learn more about security group configuratio
check if there are network segments you need
check if you have the port you need

Protocol & Port	Type	Source	Description
All	IPv4	192.168.4.161/32	
All	IPv4	192.168.4.49/32	–
All	IPv4	192.168.5.130/32	–
All	IPv4	192.168.5.245/32	–
All	IPv4	192.168.6.151/32	–
All	IPv4	192.168.7.233/32	–
All	IPv4	198.19.32.0/19	–
All	IPv4	Sys-default	–
ICMP : All	IPv4	0.0.0.0/0	–
TCP : All	IPv4	0.0.0.0/0	–

If no, add it.

Figure 4-17 Adding an inbound rule


Security Group › Sys-default
Import Rule
Export
click to add rule
Summary
Inbound Rules
Outbound Rules
Associated Instances
Add Rule
Protoco
All
All
All
All
All
All
All
Add Inbound Rule Learn more about security group configuration.
An inbound rule allows inbound traffic to instances in the security group.
set the source address IP:select the IP address method to add a network segment;select the security group method, the default open ports on all network segments in the security group
set the protocol mode and port number
Operation

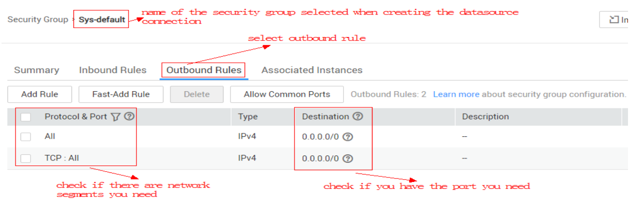
Security Group default

Protocol & Port	Source	Description	Operation
Custom TCP Example: 22 or 22-30	IP address 0 . 0 . 0 . 0 / 0		Operation

- Outbound rules: Check whether the CIDR block and port of the outbound rule are enabled. (It is recommended that all CIDR blocks be enabled.)

  Check whether the CIDR block and port are configured.

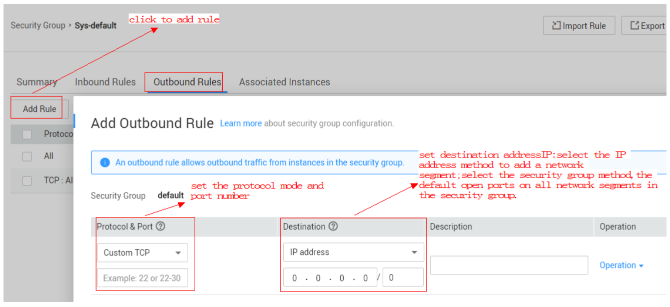
Figure 4-18 Checking whether the CIDR block and port are configured


Security Group › Sys-default
name of the security group selected when creating the datasource connection
In
select outbound rule
Summary
Inbound Rules
Outbound Rules
Associated Instances
Add Rule
Fast-Add Rule
Delete
Allow Common Ports
Outbound Rules: 2 Learn more about security group configuration.
check if there are network segments you need
check if you have the port you need

Protocol & Port	Type	Destination	Description
All	IPv4	0.0.0.0/0	–
TCP : All	IPv4	0.0.0.0/0	–

If no, add it.

Figure 4-19 Adding an outbound rule

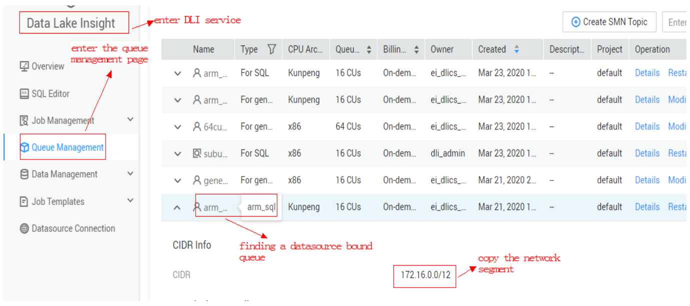

Security Group › Sys-default
click to add rule
Import Rule
Export
Summary
Inbound Rules
Outbound Rules
Associated Instances
Add Rule
Add Outbound Rule Learn more about security group configuration.
An outbound rule allows outbound traffic from instances in the security group.
set destination addressIP:select the IP address method to add a network segment;select the security group method,the default open ports on all network segments in the security group.
Security Group default
set the protocol mode and port number
Protocol & Port
Destination
Description
Operation
Custom TCP
Example: 22 or 22-30
IP address
0 . 0 . 0 . 0 / 0
Operation

– Enhanced datasource connections:

Check whether the CIDR block corresponding to the DLI queue is open. If no, add an outbound CIDR block in the VPC.

i. Find the CIDR block corresponding to the queue bound to the datasource connection in DLI.

Figure 4-20 Searching for the CIDR clock corresponding to the queue bound to the datasource connection



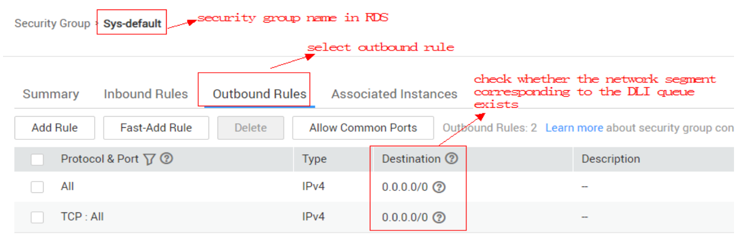
	Name	Type	CPU Arc...	Queu...	Billin...	Owner	Created	Descript...	Project	Operation
⌄	arm_...	For SQL	Kunpeng	16 CUs	On-dem...	ei_dlics_...	Mar 23, 2020 1...	--	default	Details Resta
⌄	arm_...	For gen...	Kunpeng	16 CUs	On-dem...	ei_dlics_...	Mar 23, 2020 1...	--	default	Details Modi
⌄	64cu...	For gen...	x86	64 CUs	On-dem...	ei_dlics_...	Mar 23, 2020 1...	--	default	Details Modi
⌄	subu...	For SQL	x86	16 CUs	On-dem...	dli_admin	Mar 23, 2020 1...	--	default	Details Resta
⌄	gene...	For gen...	x86	16 CUs	On-dem...	ei_dlics_...	Mar 21, 2020 2...	--	default	Details Modi
^	arm_... arm_sql		Kunpeng	16 CUs	On-dem...	ei_dlics_...	Mar 21, 2020 1...	--	default	Details Resta

CIDR Info

CIDR 172.16.0.0/12

ii. In the VPC security group, check whether the CIDR block corresponding to the DLI queue has been configured.

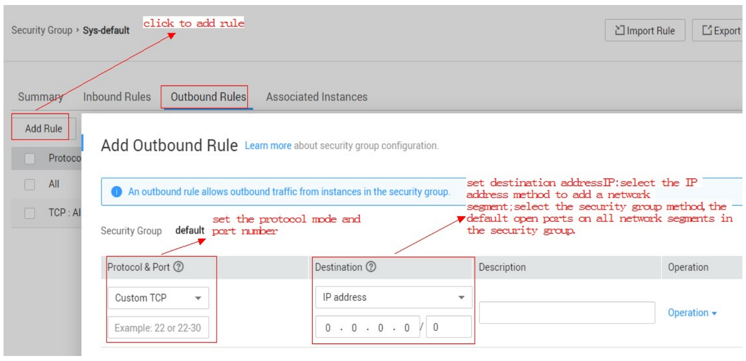
Figure 4-21 Checking the CIDR block corresponding to the DLI queue in the security group in the VPC


Security Group Sys-default
security group name in RDS
select outbound rule
Summary
Inbound Rules
Outbound Rules
Associated Instances
check whether the network segment corresponding to the DLI queue exists
Add Rule
Fast-Add Rule
Delete
Allow Common Ports
Outbound Rules: 2 Learn more about security group con

	Protocol & Port	Type	Destination	Description
☐	All	IPv4	0.0.0.0/0	--
☐	TCP : All	IPv4	0.0.0.0/0	--

If no, add it.

Figure 4-22 Adding the corresponding CIDR to the VPC


Security Group › Sys-default

click to add rule

Import Rule | Export

Summary | Inbound Rules | Outbound Rules | Associated Instances

Add Rule

Add Outbound Rule Learn more about security group configuration.

An outbound rule allows outbound traffic from instances in the security group.

set destination addressIP:select the IP address method to add a network segment;select the security group method, the default open ports on all network segments in the security group.

Security Group default

set the protocol mode and port number

Protocol & Port	Destination	Description	Operation
Custom TCP Example: 22 or 22-30	IP address 0 . 0 . 0 . 0 / 0		Operation

If the fault persists, contact Huawei technical support.

4.8 How Do I Troubleshoot a Connection Timeout Issue That Isn't Recorded in Logs When Accessing MRS HBase Through a Datasource Connection?

If you have not added the cluster host information to the datasource connection, it can lead to KRB authentication failure, resulting in a connection timeout. In this case, there will not be any error message in the logs.

You are advised to reconfigure the host information and then try to access MRS HBase again.

Log in to the DLI console and choose **Datasource Connections** > **Enhanced**. On the **Enhanced** tab, select a connection and click **Modify Host**. In the displayed dialog box, enter the host information.

The information is in the format of *Host IP address Host name/Domain name*. Each piece of information is separated by a line break.

For how to obtain MRS host information, see **Modifying Host Information**.

4.9 How Do I Fix the "Failed to get subnet" Error When Creating a Datasource Connection in DLI?

Symptom

When you create a VPC peering connection for the datasource connection, the following error information is displayed:

```
Failed to get subnet 2c2bd2ed-7296-4c64-9b60-ca25b5eee8fe. Response code : 404, message :
{"code":"VPC.0202","message":"Query resource by id 2c2bd2ed-7296-4c64-9b60-ca25b5eee8fe fail.the
subnet could not be found."}
```

Cause Analysis

The **VPC Administrator** permission is required to use the VPC, subnet, route, VPC peering connection, and port for DLI datasource connections.

The customer's failure to grant the VPC permission resulted in an error in the DLI datasource connection, which stated that the subnet could not be found.

Procedure

Step 1 Log in to the DLI management console. In the navigation pane on the left, choose **Global Configuration** > **Service Authorization**.

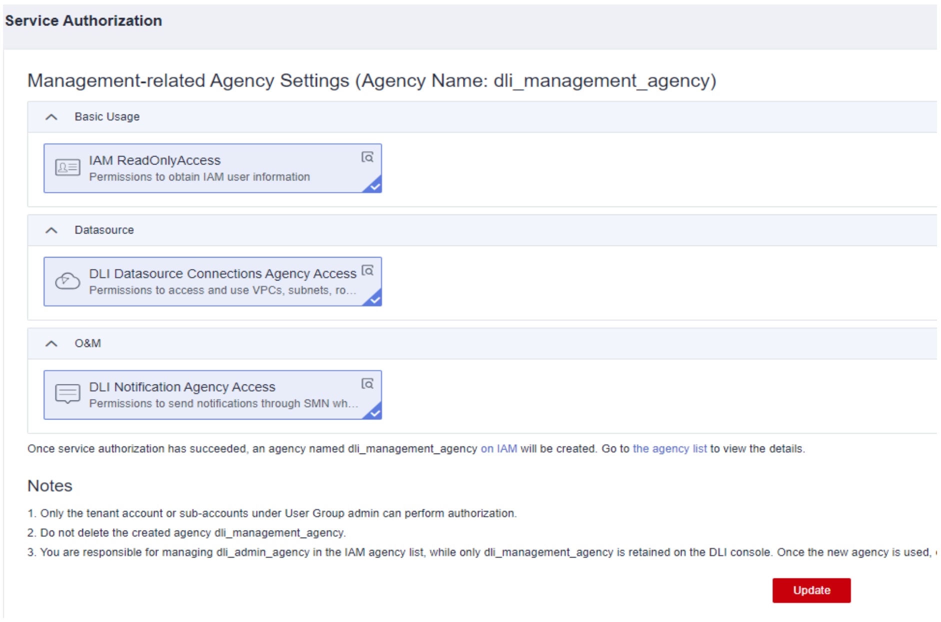
Step 2 On the agency settings page, select required permissions.

DLI Datasource Connections Agency Access is the permissions to access and use VPCs, subnets, routes, and VPC peering connections in datasource scenarios.

For more information, see **DLI Agency Permissions**.

Step 3 Select the permissions to be included in **dli_management_agency** and click **Update**.

Figure 4-23 Updating agency permissions



Service Authorization

Management-related Agency Settings (Agency Name: dli_management_agency)

Basic Usage

IAM ReadOnlyAccess
Permissions to obtain IAM user information

Datasource

DLI Datasource Connections Agency Access
Permissions to access and use VPCs, subnets, ro...

O&M

DLI Notification Agency Access
Permissions to send notifications through SMN wh...

Once service authorization has succeeded, an agency named dli_management_agency on IAM will be created. Go to the agency list to view the details.

Notes

1. Only the tenant account or sub-accounts under User Group admin can perform authorization.
2. Do not delete the created agency dli_management_agency.
3. You are responsible for managing dli_admin_agency in the IAM agency list, while only dli_management_agency is retained on the DLI console. Once the new agency is used,

Step 4 After updating the agency, recreate the datasource connection and rerun the job.

----End

4.10 How Do I Do If I Encounter the "Incorrect string value" Error When Executing insert overwrite on a Datasource RDS Table?

Symptom

A datasource RDS table was created in the DataArts Studio, and the **insert overwrite** statement was executed to write data into RDS. **DLI.0999: BatchUpdateException: Incorrect string value: '\xF0\x9F\x90\xB3' for column 'robot_name' at row 1** was reported.

Cause Analysis

The data to be written contains emojis, which are encoded in the unit of four bytes. MySQL databases use the UTF-8 format, which encodes data in the unit of three bytes by default. In this case, an error occurs when the emoji data is inserted into to the MySQL database.

Possible causes are as follows:

- A database coding error occurred.

Procedure

Change the character set to **utf8mb4**.

Step 1 Run the following SQL statement to change the database character set:

```
ALTER DATABASE DATABASE_NAME DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
```

Step 2 Run the following SQL statement to change the table character set:

```
ALTER TABLE TABLE_NAME DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
```

Step 3 Run the following SQL statement to change the character set for all fields in the table:

```
ALTER TABLE TABLE_NAME CONVERT TO CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
```

----End

Related Information

How Do I Ensure that the Character Set of an RDS MySQL Database Is Correct?

4.11 How Do I Resolve the Null Pointer Error When Creating an RDS Datasource Table?

Symptom

The system failed to create a datasource RDS table, and null pointer error was reported.

Cause Analysis

The following table creation statement was used:

```
CREATE TABLE IF NOT EXISTS dli_to_rds
 USING JDBC OPTIONS (
 'url'='jdbc:mysql://to-rds-1174405119-oLRHAGE7.datasource.com:5432/postgreDB',
 'driver'='org.postgresql.Driver',
 'dbtable'='pg_schema.test1',
 'passwdauth' = 'xxx',
 'encryption' = 'true');
```

The RDS database is in a PostGre cluster, and the protocol header in the URL is invalid.

Procedure

Change the URL to **url'='jdbc:postgresql://to-rds-1174405119-oLRHAGE7.datasource.com:5432/postgreDB** and run the creation statement. The datasource table is successfully created.

For details about the syntax for creating a datasource RDS table, see **Creating a DLI Table and Associating It with RDS**.

4.12 Error Message "org.postgresql.util.PSQLException: ERROR: tuple concurrently updated" Is Displayed When the System Executes insert overwrite on a Datasource GaussDB(DWS) Table

Symptom

The system failed to execute **insert overwrite** on the datasource GaussDB(DWS) table, and **org.postgresql.util.PSQLException: ERROR: tuple concurrently updated** was displayed.

Cause Analysis

Concurrent operations existed in the job. Two insert overwrite operations were executed on the table at the same time.

One CN was running the following statement:

```
TRUNCATE TABLE BI_MONITOR.SAA_OUTBOUND_ORDER_CUST_SUM
```

Another CN was running the following command:

```
call bi_monitor.pkg_saa_out_bound_monitor_p_saa_outbound_order_cust_sum
```

This function deletes and inserts SAA_OUTBOUND_ORDER_CUST_SUM.

Procedure

Modify job logic to prevent concurrent insert overwrite operations on the same table.

4.13 RegionTooBusyException Is Reported When Data Is Imported to a CloudTable HBase Table Through a Datasource Table

Symptom

A datasource table was used to import data to a CloudTable HBase table. This HBase table contains a column family and a rowkey for 100 million simulating data records. The data volume is 9.76 GB. The job failed after 10 million data records were imported.

Cause Analysis

1. View driver error logs.
2. View executor error logs.
3. View task error logs.

The rowkey was poorly designed causing a large amount of traffic redirected to single or very few numbers of nodes.

Procedure

1. Pre-partition the HBase.
2. Hash the rowkey.

Summary and Suggestions

Distribute data to different RegionServer. Add **distribute by rand()** to the end of the insert statement.

4.14 How Do I Do If A Null Value Is Written Into a Non-Null Field When Using a DLI Datasource Connection to Connect to a GaussDB(DWS) Table?

Symptom

A table was created on GaussDB(DWS) and then a datasource connection was created on DLI to read and write data. An error message was displayed during data writing, indicating that DLI was writing a null value to a non-null field of the table, and the job failed.

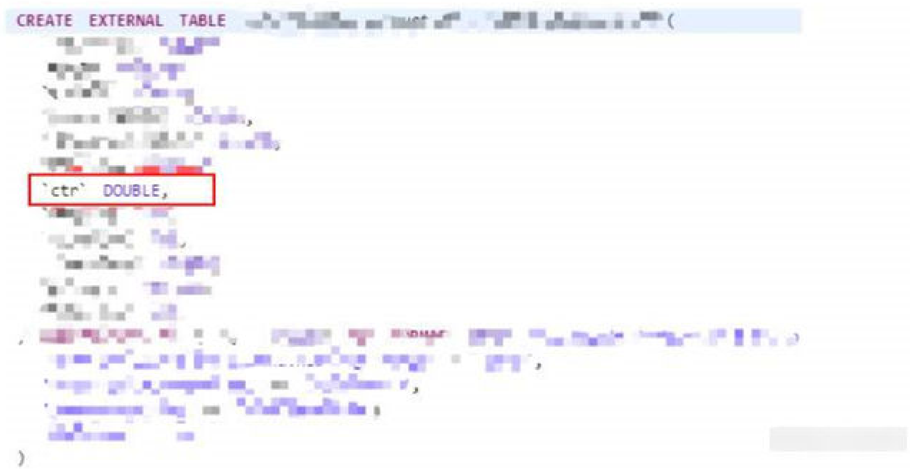
The error message is as follows:

```
DLI.0999: PSQLException: ERROR: dn_6009_6010: null value in column "ctr" violates not-null constraint
Detail: Failing row contains (400070309, 9.00, 25, null, 2020-09-22, 2020-09-23 04:30:01.741).
```

Cause Analysis

1. The CIR field in the source table is of the **DOUBLE** type.

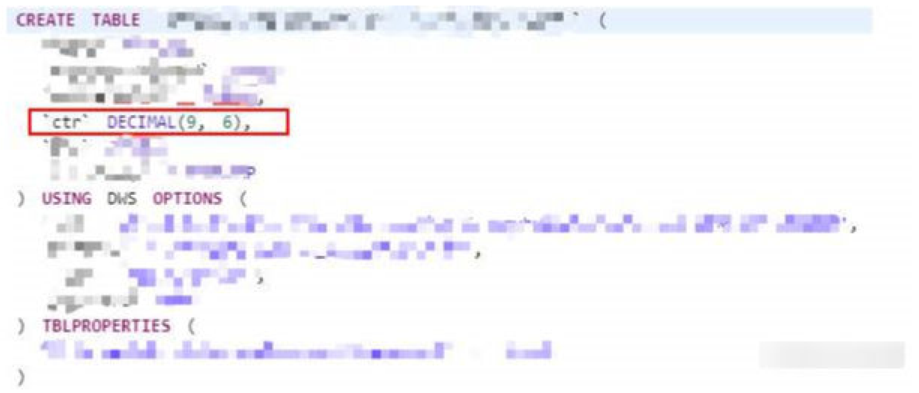
Figure 4-24 Creation statement of the source table



```
CREATE EXTERNAL TABLE [redacted] (
  [redacted]
  `ctr` DOUBLE,
  [redacted]
)
```

2. The field type in the target table is **DECIMAL(9,6)**.

Figure 4-25 Creation statement of the target table



```
CREATE TABLE [redacted] (
  [redacted]
  `ctr` DECIMAL(9, 6),
  [redacted]
) USING DWS OPTIONS (
  [redacted]
) TBLPROPERTIES (
  [redacted]
)
```

3. View the source table data. The CTR value that causes the problem is **1675**, which exceed the precision (9 – 6 = 3 digits) of the **DECIMAL(9,6)** type. A null value was generated when the double value was converted to the decimal value, and the insert operation failed.

Procedure

Change the precision of the decimal data defined in the target table.

4.15 How Do I Do If an Insert Operation Failed After the Schema of the GaussDB(DWS) Source Table Is Updated?

Symptom

A datasource GaussDB(DWS) table and the datasource connection were created in DLI, and the schema of the source table in GaussDB(DWS) were updated. During the job execution, the schema of the source table failed to be updated, and the job failed.

Cause Analysis

When the insert operation is executed on the DLI datasource table, the GaussDB(DWS) source table is deleted and recreated. If the statement for creating the datasource table is not updated on DLI, the GaussDB(DWS) source table will fail to be updated.

Procedure

Create a datasource table on DLI and add table creation configuration **truncate = true** to clear table data but not delete the table.

Summary and Suggestions

After the source table is updated, the corresponding datasource table must be updated too on DLI.

4.16 How Do I Insert Data into an RDS Table with an Auto-Increment Primary Key Using DLI?

When creating an associated RDS table in DLI, if the RDS table contains an auto-increment primary key or other auto-populated fields, you can take the following measures when inserting data in DLI:

1. Omit auto-populated fields when inserting data: When inserting data in DLI, for auto-increment primary key fields or other auto-populated fields, you can omit these fields in the insert statement. The database will automatically generate values for these fields. For example, if there is an auto-increment primary key field named **id** in the table, you can exclude this field when inserting data, and the database will assign a unique **id** value for the newly inserted row.
2. Use NULL values: If you need to explicitly specify that certain fields should be auto-populated by the database when inserting data, you can fill in NULL values in those fields. This way, the database will recognize that these fields should be automatically generated by the system, rather than specified by the user.

5 SQL Jobs

5.1 SQL Job Development

5.1.1 SQL Jobs

Can I Create Temporary Tables on DLI?

A temporary table is used to store intermediate results. When a transaction or session ends, the data in the temporary table can be automatically deleted. For example, in MySQL, you can use **create temporary table...** to create a temporary table. After a transaction or session ends, the table data is automatically deleted. Does DLI Support This Function?

Currently, **you cannot create temporary tables on DLI**. You can create a table by using SQL statements. For details about the SQL syntax supported by DLI, see **Creating a DLI Table**.

Can I Connect to DLI Locally? Is a Remote Connection Tool Supported?

Currently DLI can only be accessed through a browser. You must submit jobs on the console.

For details, see **Creating and Submitting a Spark SQL Job**.

Will a DLI SQL Job Be Killed If It Has Been Running for More Than 12 Hours?

By default, SQL jobs that have been running for more than 12 hours will be canceled to ensure stability of queues.

You can use the **dli.sql.job.timeout** parameter (unit: second) to configure the timeout interval.

Does DLI Support Local Testing of Spark Jobs?

Currently, DLI does not support local testing of Spark jobs. You can install the DLI Livy tool and use its interactive sessions to debug Spark jobs.

For details, see **Submitting a Spark Jar Job Using Livy**.

Can I Delete a Row of Data from an OBS Table or DLI Table?

Deleting a row of data from an OBS table or DLI table is not allowed.

5.1.2 How Do I Merge Small Files?

If a large number of small files are generated during SQL execution, job execution and table query will take a long time. In this case, you should merge small files.

You are advised to use temporary tables for data transfer. There is a risk of data loss in self-read and self-write operations during unexpected exceptional scenarios.

Run the following SQL statements:

```
INSERT OVERWRITE TABLE tablename
select  * FROM  tablename
DISTRIBUTE BY floor(rand()*20)
```

5.1.3 How Do I Use DLI to Access Data in an OBS Bucket?

1. Create an OBS Table.

   For details about the creation syntax, see **Creating an OBS Table Using the DataSource Syntax**.
2. Add partitions.

   For details about the partitioning syntax, see **Adding a Partition**.
3. Import data from the OBS bucket to partitions of the table.

   For details about the syntax, see **Importing Data**.
4. Query data.

   For details about the syntax, see **Basic Statements**.

5.1.4 How Do I Specify an OBS Path When Creating an OBS Table?

Scenario

When creating an OBS table, you must specify a table path in the database. The path format is as follows: obs://xxx/database name/table name.

For details about the syntax for creating OBS tables, see **Creating an OBS Table Using the DataSource Syntax**.

Correct Example

```
CREATE TABLE `di_seller_task_activity_30d` (`user_id` STRING COMMENT' user ID...) SORTED as parquet
LOCATION 'obs://akc-bigdata/akdc.db/di_seller_task_activity_30d'
```

Analysis of a Typical Error Example

```
CREATE TABLE `di_seller_task_activity_30d` (`user_id` STRING COMMENT' user ID...) SORTED as parquet
LOCATION 'obs://akc-bigdata/akdc.db'
```

NOTE

If the specified path is **akdc.db**, data in this path will be cleared when the **insert overwrite** statement is executed.

5.1.5 How Do I Create a Table Using JSON Data in an OBS Bucket?

To associate JSON data nested in an OBS bucket, you can create a table in asynchronous mode.

The following is an example of a table creation statement that shows how to use JSON format options to specify the path in OBS:

```
create table tb1 using json options(path 'obs://....')
```

- **using json**: JSON format is used.
- **options**: used to set table options.
- **path**: path of the JSON file in OBS.

5.1.6 How Can I Use the count Function to Perform Aggregation?

The correct method for using the count function to perform aggregation is as follows:

```
SELECT
  http_method,
  count(http_method)
FROM
  apigateway
WHERE
  service_id = 'ecs' Group BY http_method
```

Or

```
SELECT
  http_method
FROM
  apigateway
WHERE
  service_id = 'ecs' DISTRIBUTE BY http_method
```

If an incorrect method is used, an error will be reported.

```
SELECT
  http_method,
  count(http_method)
FROM
  apigateway
WHERE
  service_id = 'ecs' DISTRIBUTE BY http_method
```

5.1.7 How Do I Synchronize DLI Table Data Across Regions?

You can use the cross-region replication function of OBS. The procedure is as follows:

1. Export the DLI table data in region 1 to the user-defined OBS bucket.

   For details, see **Exporting Data from DLI to OBS**.

2. Use the OBS cross-region replication function to replicate data to the OBS bucket in region 2.
3. Import or use the corresponding data as required.

5.1.8 How Do I Insert Table Data into Specific Fields of a Table Using a SQL Job?

If you need to insert data into a table but only want to specify certain fields, you can use the **INSERT INTO** statement combined with the **SELECT** clause.

However, DLI currently does not support inserting data into partial columns directly in the **INSERT INTO** statement. You need to ensure that the number and types of fields selected in the **SELECT** clause match the Schema information of the target table. That is, ensure that the data types and column field numbers of the source table and sink table are the same to avoid insertion failure.

If some fields in the sink table are not specified in the **SELECT** clause, these fields may also be inserted with default values or set to null values (depending on whether the field allows null values).

5.1.9 How Do I Troubleshoot Slow SQL Jobs?

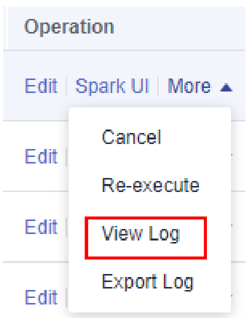
If the job runs slowly, perform the following steps to find the causes and rectify the fault:

Possible Cause 1: Full GC

Check whether the problem is caused by FullGC.

1. Log in to the DLI console. In the navigation pane, choose **Job Management** > **SQL Jobs**.
2. On the **SQL Jobs** page, locate the row that contains the target job and click **More** > **View Log** in the **Operation** column.

Figure 5-1 Viewing logs



3. Obtain the folder of the archived logs in the OBS directory. The details are as follows:
   – Spark SQL jobs:
     Locate the log folder whose name contains **driver** or **container_** *xxx* **_000001**.

Figure 5-2 Example folder name

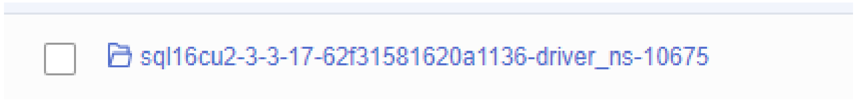
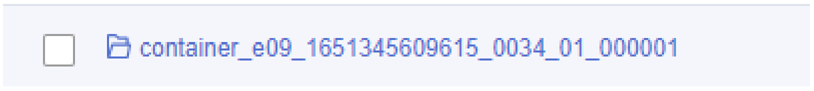

sql16cu2-3-3-17-62f31581620a1136-driver_ns-10675

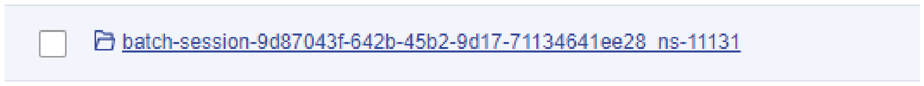
Figure 5-3 Example folder name


container_e09_1651345609615_0034_01_000001

– Spark Jar jobs:

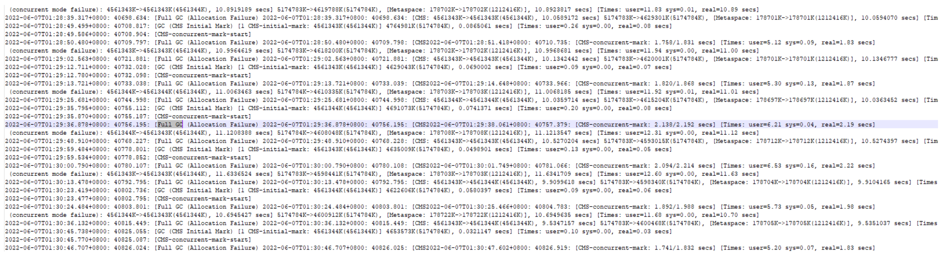
  The archive log folder of a Spark Jar job starts with **batch**.

  Figure 5-4 Example folder name


batch-session-9d87043f-642b-45b2-9d17-71134641ee28_ns-11131

4. Go to the archive log file directory and download the **gc.log.* log** file.
5. Open the downloaded **gc.log.* log** file, search for keyword **Full GC**, and check whether time records in the file are continuous and Full GC information is recorded repeatedly.

Figure 5-5 Full GC logs



```
 (concurrent mode failure): 4561343K->4561343K(4561344K), 10.8919189 secs] 5174783K->4619788K(5174784K), [Metaspace: 178702K->178702K(1212416K)], 10.8923817 secs] [Times: user=11.83 sys=0.01, real=10.89 secs]
2022-06-07T01:28:39.317+0800: 40698.634: [Full GC (Allocation Failure) 2022-06-07T01:28:39.317+0800: 40698.634: [CMS: 4561343K->4561343K(4561344K), 10.0589172 secs] 5174783K->4629301K(5174784K), [Metaspace: 178701K->178701K(1212416K)], 10.0594070 secs] [Tim
2022-06-07T01:28:49.499+0800: 40708.817: [GC (CMS Initial Mark) [1 CMS-initial-mark: 4561343K(4561344K)] 4764981K(5174784K), 0.0865061 secs] [Times: user=0.26 sys=0.00, real=0.08 secs]
2022-06-07T01:28:49.586+0800: 40708.904: [CMS-concurrent-mark-start]
2022-06-07T01:28:50.480+0800: 40709.797: [Full GC (Allocation Failure) 2022-06-07T01:28:50.480+0800: 40709.798: [CMS2022-06-07T01:28:51.418+0800: 40710.735: [CMS-concurrent-mark: 1.758/1.831 secs] [Times: user=5.12 sys=0.09, real=1.83 secs]
 (concurrent mode failure): 4561343K->4561343K(4561344K), 10.9964619 secs] 5174783K->4618200K(5174784K), [Metaspace: 178702K->178702K(1212416K)], 10.9968681 secs] [Times: user=11.94 sys=0.00, real=11.00 secs]
2022-06-07T01:29:02.563+0800: 40721.881: [Full GC (Allocation Failure) 2022-06-07T01:29:02.563+0800: 40721.881: [CMS: 4561343K->4561343K(4561344K), 10.1342442 secs] 5174783K->4620001K(5174784K), [Metaspace: 178701K->178701K(1212416K)], 10.1346777 secs] [Tim
2022-06-07T01:29:12.711+0800: 40732.028: [GC (CMS Initial Mark) [1 CMS-initial-mark: 4561343K(4561344K)] 4629043K(5174784K), 0.0690002 secs] [Times: user=0.09 sys=0.00, real=0.07 secs]
2022-06-07T01:29:12.780+0800: 40732.098: [CMS-concurrent-mark-start]
2022-06-07T01:29:13.721+0800: 40733.038: [Full GC (Allocation Failure) 2022-06-07T01:29:13.721+0800: 40733.039: [CMS2022-06-07T01:29:14.648+0800: 40733.966: [CMS-concurrent-mark: 1.820/1.868 secs] [Times: user=5.30 sys=0.13, real=1.87 secs]
 (concurrent mode failure): 4561344K->4561344K(4561344K), 11.0063463 secs] 5174784K->4610335K(5174784K), [Metaspace: 178703K->178703K(1212416K)], 11.0068185 secs] [Times: user=11.92 sys=0.01, real=11.01 secs]
2022-06-07T01:29:25.681+0800: 40744.998: [Full GC (Allocation Failure) 2022-06-07T01:29:25.681+0800: 40744.998: [CMS: 4561344K->4561344K(4561344K), 10.0359714 secs] 5174783K->4615204K(5174784K), [Metaspace: 178697K->178697K(1212416K)], 10.0363452 secs] [Tim
2022-06-07T01:29:35.795+0800: 40755.112: [GC (CMS Initial Mark) [1 CMS-initial-mark: 4561344K(4561344K)] 4691073K(5174784K), 0.0741371 secs] [Times: user=0.20 sys=0.00, real=0.08 secs]
2022-06-07T01:29:35.870+0800: 40755.187: [CMS-concurrent-mark-start]
2022-06-07T01:29:36.878+0800: 40756.195: [Full GC (Allocation Failure) 2022-06-07T01:29:36.878+0800: 40756.195: [CMS2022-06-07T01:29:38.061+0800: 40757.379: [CMS-concurrent-mark: 2.138/2.192 secs] [Times: user=6.21 sys=0.04, real=2.19 secs]
 (concurrent mode failure): 4561344K->4561343K(4561344K), 11.1208388 secs] 5174784K->4608048K(5174784K), [Metaspace: 178708K->178708K(1212416K)], 11.1213547 secs] [Times: user=12.31 sys=0.00, real=11.12 secs]
2022-06-07T01:29:48.910+0800: 40768.227: [Full GC (Allocation Failure) 2022-06-07T01:29:48.910+0800: 40768.228: [CMS: 4561343K->4561343K(4561344K), 10.5270204 secs] 5174783K->4593015K(5174784K), [Metaspace: 178712K->178712K(1212416K)], 10.5274397 secs] [Tim
2022-06-07T01:29:59.484+0800: 40778.801: [GC (CMS Initial Mark) [1 CMS-initial-mark: 4561343K(4561344K)] 4635009K(5174784K), 0.0498901 secs] [Times: user=0.13 sys=0.00, real=0.05 secs]
2022-06-07T01:29:59.534+0800: 40778.852: [CMS-concurrent-mark-start]
2022-06-07T01:30:00.790+0800: 40780.107: [Full GC (Allocation Failure) 2022-06-07T01:30:00.790+0800: 40780.108: [CMS2022-06-07T01:30:01.749+0800: 40781.066: [CMS-concurrent-mark: 2.094/2.214 secs] [Times: user=6.53 sys=0.16, real=2.22 secs]
 (concurrent mode failure): 4561343K->4561343K(4561344K), 11.6336524 secs] 5174783K->4598441K(5174784K), [Metaspace: 178703K->178703K(1212416K)], 11.6341709 secs] [Times: user=12.60 sys=0.00, real=11.63 secs]
2022-06-07T01:30:13.478+0800: 40792.795: [Full GC (Allocation Failure) 2022-06-07T01:30:13.478+0800: 40792.795: [CMS: 4561343K->4561344K(4561344K), 9.9099618 secs] 5174783K->4598340K(5174784K), [Metaspace: 178704K->178704K(1212416K)], 9.9104165 secs] [Times
2022-06-07T01:30:23.419+0800: 40802.736: [GC (CMS Initial Mark) [1 CMS-initial-mark: 4561344K(4561344K)] 4622606K(5174784K), 0.0580397 secs] [Times: user=0.09 sys=0.00, real=0.06 secs]
2022-06-07T01:30:23.477+0800: 40802.795: [CMS-concurrent-mark-start]
2022-06-07T01:30:24.484+0800: 40803.801: [Full GC (Allocation Failure) 2022-06-07T01:30:24.484+0800: 40803.801: [CMS2022-06-07T01:30:25.466+0800: 40804.783: [CMS-concurrent-mark: 1.892/1.988 secs] [Times: user=5.73 sys=0.05, real=1.98 secs]
 (concurrent mode failure): 4561344K->4561343K(4561344K), 10.6945427 secs] 5174784K->4600912K(5174784K), [Metaspace: 178722K->178722K(1212416K)], 10.6949635 secs] [Times: user=11.68 sys=0.00, real=10.70 secs]
2022-06-07T01:30:36.132+0800: 40815.449: [Full GC (Allocation Failure) 2022-06-07T01:30:36.132+0800: 40815.449: [CMS: 4561343K->4561344K(4561344K), 9.5347157 secs] 5174783K->4600468K(5174784K), [Metaspace: 178705K->178705K(1212416K)], 9.5351037 secs] [Times
2022-06-07T01:30:45.738+0800: 40825.055: [GC (CMS Initial Mark) [1 CMS-initial-mark: 4561344K(4561344K)] 4653573K(5174784K), 0.0321147 secs] [Times: user=0.10 sys=0.00, real=0.03 secs]
2022-06-07T01:30:45.770+0800: 40825.087: [CMS-concurrent-mark-start]
2022-06-07T01:30:46.707+0800: 40826.024: [Full GC (Allocation Failure) 2022-06-07T01:30:46.707+0800: 40826.025: [CMS2022-06-07T01:30:47.602+0800: 40826.919: [CMS-concurrent-mark: 1.741/1.832 secs] [Times: user=5.20 sys=0.07, real=1.83 secs]
```

Cause locating and solution

Cause 1: There are too many small files in a table.

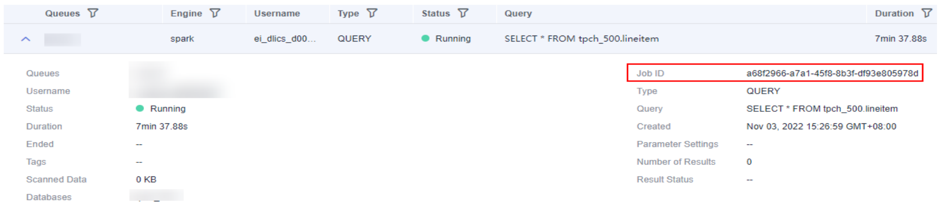
1. Log in to the DLI console and go to the SQL editor page. On the SQL Editor page, select the queue and database of the faulty job.
2. Run the following statement to check the number of files in the table and specify the *table name*.

   ```
   select count(distinct fn)  FROM
   (select input_file_name() as fn from table name) a
   ```
3. If there are too many small files, rectify the fault by referring to **How Do I Merge Small Files?**.

Cause 2: There is a broadcast table.

1. Log in to the DLI console. In the navigation pane, choose **Job Management** > **SQL Jobs**.
2. On the **SQL Jobs** page, locate the row that contains the target job and click to view the job details and obtain the job ID.

Figure 5-6 Obtaining the job ID

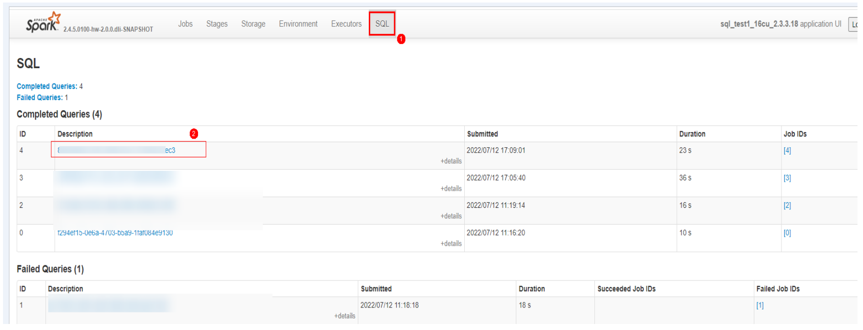


Queues	Engine	Username	Type	Status	Query	Duration
[redacted]	spark	ei_dlics_d00...	QUERY	● Running	SELECT * FROM tpch_500.lineitem	7min 37.88s

Queues	[redacted]	Job ID	a68f2966-a7a1-45f8-8b3f-df93e805978d
Username	[redacted]	Type	QUERY
Status	● Running	Query	SELECT * FROM tpch_500.lineitem
Duration	7min 37.88s	Created	Nov 03, 2022 15:26:59 GMT+08:00
Ended	--	Parameter Settings	--
Tags	--	Number of Results	0
Scanned Data	0 KB	Result Status	--
Databases	[redacted]		

3. In the **Operation** column of the job, click **Spark UI**.
4. On the displayed page, choose **SQL** from the menu bar. Click the hyperlink in the **Description** column of the row that contains the job ID.

Figure 5-7 Clicking the job link



Spark 2.4.5.0100-hw-2.0.0.dli-SNAPSHOT
Jobs
Stages
Storage
Environment
Executors
SQL
1
sql_test1_16cu_2.3.3.18 application UI
2

SQL

Completed Queries: 4
Failed Queries: 1

Completed Queries (4)

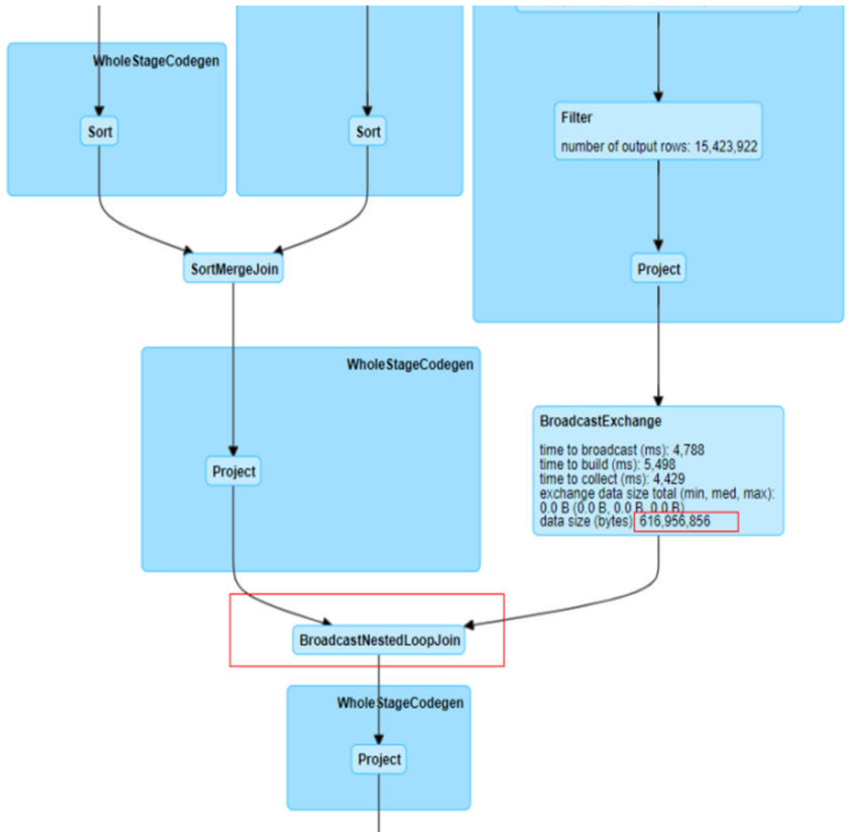
ID	Description	Submitted	Duration	Job IDs
4	[redacted]ec3 +details	2022/07/12 17:09:01	23 s	[4]
3	[redacted] +details	2022/07/12 17:05:40	36 s	[3]
2	[redacted] +details	2022/07/12 11:19:14	16 s	[2]
0	f294ef15-0e6a-4703-b5a9-1faf084e9130 +details	2022/07/12 11:16:20	10 s	[0]

Failed Queries (1)

ID	Description	Submitted	Duration	Succeeded Job IDs	Failed Job IDs
1	[redacted] +details	2022/07/12 11:18:18	18 s		[1]

5. View the DAG of the job to check whether the BroadcastNestedLoopJoin node exists.

Figure 5-8 DAG



WholeStageCodegen
Sort
Sort
Filter
number of output rows: 15,423,922
SortMergeJoin
Project
WholeStageCodegen
Project
BroadcastExchange
time to broadcast (ms): 4,788
time to build (ms): 5,498
time to collect (ms): 4,429
exchange data size total (min, med, max):
0.0 B (0.0 B, 0.0 B, 0.0 B)
data size (bytes): 616,956,856
BroadcastNestedLoopJoin
WholeStageCodegen
Project

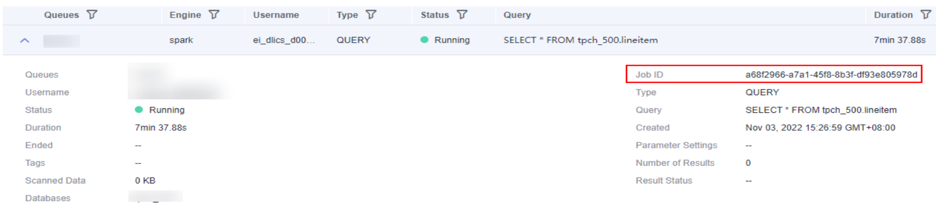
6. If the BroadcastNestedLoopJoin node exists, refer to **Why Does a SQL Job That Has Join Operations Stay in the Running State?** to rectify the fault.

Possible Cause 2: Data Skew

Check whether the problem is caused by data skew.

1. Log in to the DLI console. In the navigation pane, choose **Job Management** > **SQL Jobs**.
2. On the **SQL Jobs** page, locate the row that contains the target job and click to view the job details and obtain the job ID.

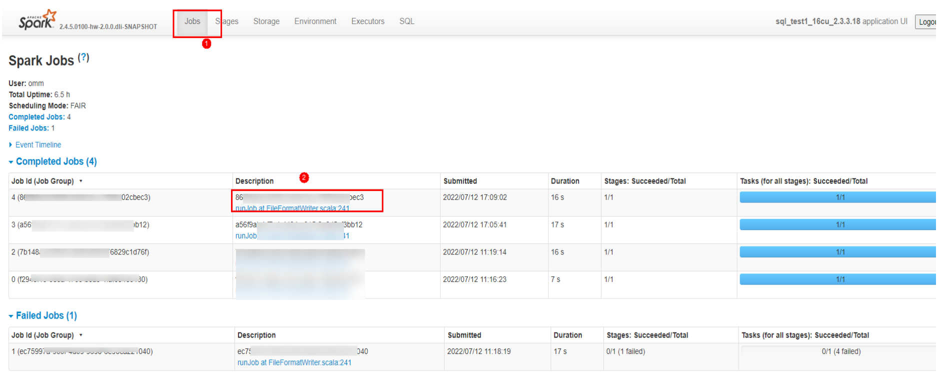
Figure 5-9 Obtaining the job ID



Queues	Engine	Username	Type	Status	Query	Duration
[redacted]	spark	ei_dlics_d00...	QUERY	● Running	SELECT * FROM tpch_500.lineitem	7min 37.88s

Queues	[redacted]	Job ID	a68f2966-a7a1-45f8-8b3f-df93e805978d
Username	[redacted]	Type	QUERY
Status	● Running	Query	SELECT * FROM tpch_500.lineitem
Duration	7min 37.88s	Created	Nov 03, 2022 15:26:59 GMT+08:00
Ended	--	Parameter Settings	--
Tags	--	Number of Results	0
Scanned Data	0 KB	Result Status	--
Databases	[redacted]		

3. In the **Operation** column of the job, click **Spark UI**.
4. On the displayed page, choose **SQL** from the menu bar. Click the hyperlink in the **Description** column of the row that contains the job ID.


Spark 2.4.5.0100-hw-2.0.0.dli-SNAPSHOT
Jobs
Stages
Storage
Environment
Executors
SQL
sql_test1_16cu_2.3.3.18 application UI
Logo
1
2

Spark Jobs (?)

User: omm
Total Uptime: 6.5 h
Scheduling Mode: FAIR
Completed Jobs: 4
Failed Jobs: 1

Event Timeline

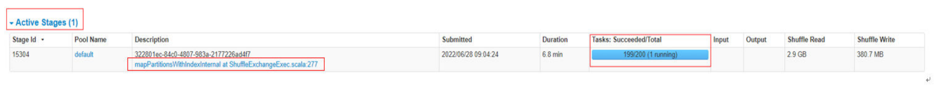
Completed Jobs (4)

Job Id (Job Group)	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
4 (8[redacted]02cbec3)	86[redacted]ec3 runJob at FileFormatWriter.scala:241	2022/07/12 17:09:02	16 s	1/1	1/1
3 (a56[redacted]b12)	a56f9a[redacted]3bb12 runJob[redacted]1	2022/07/12 17:05:41	17 s	1/1	1/1
2 (7b148[redacted]6829c1d76f)	[redacted]	2022/07/12 11:19:14	16 s	1/1	1/1
0 (f29[redacted]30)	[redacted]	2022/07/12 11:16:23	7 s	1/1	1/1

Failed Jobs (1)

Job Id (Job Group)	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
1 (ec7599[redacted]040)	ec7[redacted]040 runJob at FileFormatWriter.scala:241	2022/07/12 11:18:19	17 s	0/1 (1 failed)	0/1 (4 failed)

5. View the running status of the current stage in the Active Stage table on the displayed page. Click the hyperlink in the **Description** column.



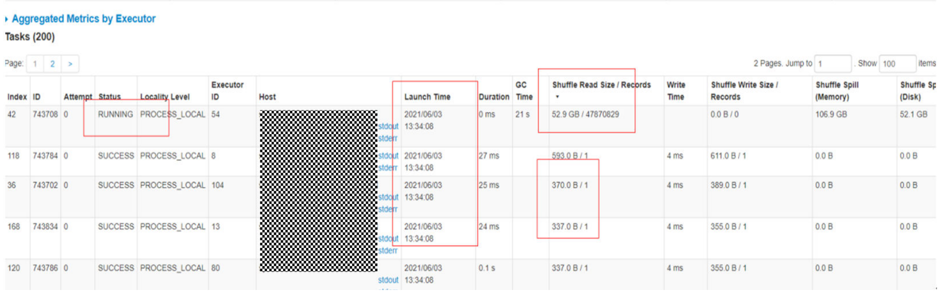
Active Stages (1)

Stage Id	Pool Name	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
15304	default	322801ec-84c0-4807-983a-2177226ad4f7 mapPartitionsWithIndexInternal at ShuffleExchangeExec.scala:277	2022/06/28 09:04:24	6.8 min	199/200 (1 running)			2.9 GB	380.7 MB

6. View the **Launch Time** and **Duration** of each task.
7. Click **Duration** to sort tasks. Check whether the overall job duration is prolonged because a task has taken a long time.

   According to **Figure 5-10**, when data skew occurs, the data volume of shuffle reads of a task is much greater than that of other tasks.

Figure 5-10 Data skew



Aggregated Metrics by Executor

Tasks (200)

Page: 1 2 >   2 Pages. Jump to 1 . Show 100 items

Index	ID	Attempt	Status	Locality Level	Executor ID	Host		Launch Time	Duration	GC Time	Shuffle Read Size / Records ▾	Write Time	Shuffle Write Size / Records	Shuffle Spill (Memory)	Shuffle Sp (Disk)
42	743708	0	RUNNING	PROCESS_LOCAL	54	[redacted]	stdout stderr	2021/06/03 13:34:08	0 ms	21 s	52.9 GB / 47870829		0.0 B / 0	106.9 GB	52.1 GB
118	743784	0	SUCCESS	PROCESS_LOCAL	8	[redacted]	stdout stderr	2021/06/03 13:34:08	27 ms		593.0 B / 1	4 ms	611.0 B / 1	0.0 B	0.0 B
36	743702	0	SUCCESS	PROCESS_LOCAL	104	[redacted]	stdout stderr	2021/06/03 13:34:08	25 ms		370.0 B / 1	4 ms	389.0 B / 1	0.0 B	0.0 B
168	743834	0	SUCCESS	PROCESS_LOCAL	13	[redacted]	stdout stderr	2021/06/03 13:34:08	24 ms		337.0 B / 1	4 ms	355.0 B / 1	0.0 B	0.0 B
120	743786	0	SUCCESS	PROCESS_LOCAL	80	[redacted]	stdout	2021/06/03 13:34:08	0.1 s		337.0 B / 1	4 ms	355.0 B / 1	0.0 B	0.0 B

Cause locating and solution

Shuffle data skew is caused by unbalanced number of key values in join.

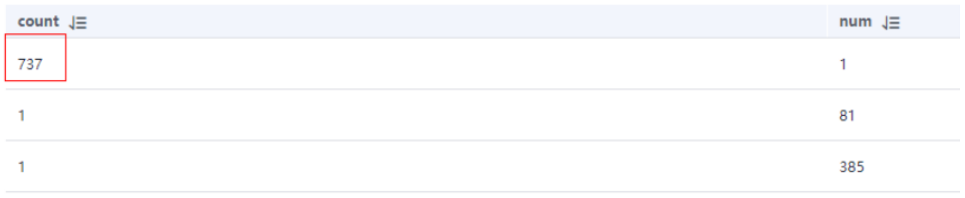
1. Perform **group by** and **count** on a join to collect statistics on the number of key values of each join. The following is an example:

   Join table **lefttbl** and table **righttbl**. **num** in the **lefttbl** table is the key value of the join. You can perform **group by** and **count** on **lefttbl.num**.

   ```
   SELECT * FROM lefttbl a LEFT join righttbl b on a.num = b.int2;
   SELECT count(1) as count,num from lefttbl  group by lefttbl.num ORDER BY count desc;
   ```

   Figure 5-11 shows the result. There are much more **num** parameters whose value are **1** than other values.

   Figure 5-11 Statistics on **num** values



count	num
737	1
1	81
1	385

2. Use **concat(cast(round(rand() * 999999999) as string)** to generate a random number for each key value.
3. If the skew is serious and random numbers cannot be generated, see **How Do I Do When Data Skew Occurs During the Execution of a SQL Job?**

5.1.10 How Do I View DLI SQL Logs?

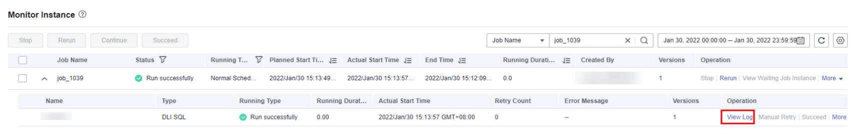
Scenario

You can view SQL job logs for routine O&M.

Procedure

1. Obtain the ID of the DLI job executed on the DataArts Studio console.

   Figure 5-12 Viewing logs



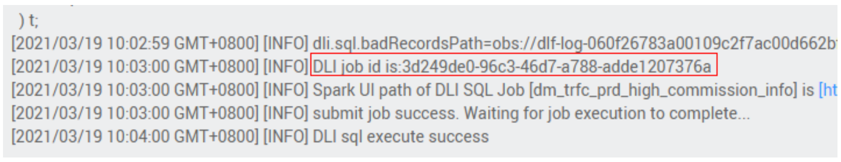
Monitor Instance

Stop Rerun Continue Succeed   Job Name job_1039   Jan 30, 2022 00:00:00 – Jan 30, 2022 23:59:59

Job Name	Status	Running T...	Planned Start Ti...	Actual Start Time	End Time	Running Durati...	Created By	Versions	Operation
job_1039	Run successfully	Normal Sched...	2022/Jan/30 15:13:49...	2022/Jan/30 15:13:57...	2022/Jan/30 15:12:09...	0.0	[redacted]	1	Stop | Rerun | View Waiting Job Instance | More

Name	Type	Running Type	Running Durat...	Actual Start Time	Retry Count	Error Message	Versions	Operation
[redacted]	DLI SQL	Run successfully	0.00	2022/Jan/30 15:13:57 GMT+08:00	0	--	1	View Log | Manual Retry | Succeed | More

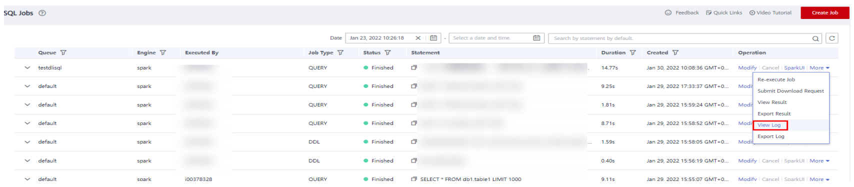
Figure 5-13 Job ID



```
  ) t;
[2021/03/19 10:02:59 GMT+0800] [INFO] dli.sql.badRecordsPath=obs://dlf-log-060f26783a00109c2f7ac00d662b
[2021/03/19 10:03:00 GMT+0800] [INFO] DLI job id is:3d249de0-96c3-46d7-a788-adde1207376a
[2021/03/19 10:03:00 GMT+0800] [INFO] Spark UI path of DLI SQL Job [dm_trfc_prd_high_commission_info] is [ht
[2021/03/19 10:03:00 GMT+0800] [INFO] submit job success. Waiting for job execution to complete...
[2021/03/19 10:04:00 GMT+0800] [INFO] DLI sql execute success
```

2. On the DLI console, choose **Job Management** > **SQL Jobs**.
3. On the **SQL Jobs** page, enter the job ID.
4. In the **Operation** column of the target job, choose **More** > **View Log** and download the logs to your PC.

Figure 5-14 Viewing logs



Queue	Engine	Executed By	Job Type	Status	Statement	Duration	Created	Operation
testdlisql	spark	[redacted]	QUERY	Finished	[redacted]	14.77s	Jan 30, 2022 10:08:36 GMT+0...	Modify | Cancel | SparkUI | More
default	spark	[redacted]	QUERY	Finished	[redacted]	9.25s	Jan 29, 2022 17:33:37 GMT+0...	Modi
default	spark	[redacted]	QUERY	Finished	[redacted]	1.81s	Jan 29, 2022 15:59:24 GMT+0...	Modi
default	spark	[redacted]	QUERY	Finished	[redacted]	8.71s	Jan 29, 2022 15:58:52 GMT+0...	Modi
default	spark	[redacted]	DDL	Finished	[redacted]	1.59s	Jan 29, 2022 15:58:05 GMT+0...	Modi
default	spark	[redacted]	DDL	Finished	[redacted]	0.40s	Jan 29, 2022 15:56:19 GMT+0...	Modify | Cancel | SparkUI | More
default	spark	i00378328	QUERY	Finished	SELECT * FROM db1.table1 LIMIT 1000	9.11s	Jan 29, 2022 15:55:07 GMT+0...	Modify | Cancel | SparkUI | More

5. Search for job ID in the downloaded logs to view the execution logs.

5.1.11 How Do I View SQL Execution Records in DLI?

Scenario

You can view the job execution records when a job is running.

Procedure

1. Log in to the DLI management console.
2. In the navigation pane on the left, choose **Job Management** > **SQL Jobs**.
3. Enter a job ID or statement to search for a job.

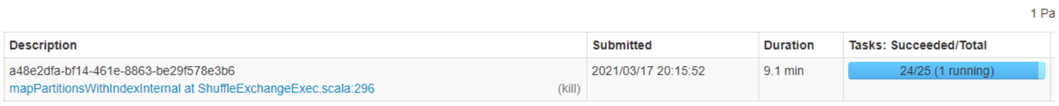
5.1.12 How Do I Do When Data Skew Occurs During the Execution of a SQL Job?

What Is Data Skew?

Data skew is a common issue during the execution of SQL jobs. When data is unevenly distributed, some compute nodes process significantly more data than others, which can impact the efficiency of the entire computation process.

For example, if you notice that a SQL query is taking a long time to execute, you can check its status in SparkUI. See **Figure 5-15**. If you see a stage that has been running for over 20 minutes with only one task remaining, it is likely due to data skew.

Figure 5-15 Data skew example



Description	Submitted	Duration	Tasks: Succeeded/Total
a48e2dfa-bf14-461e-8863-be29f578e3b6 mapPartitionsWithIndexInternal at ShuffleExchangeExec.scala:296 (kill)	2021/03/17 20:15:52	9.1 min	24/25 (1 running)

Common Data Skew Scenarios

- Group By aggregation skew

  During the execution of Group By aggregation, if some grouping keys have significantly more data than others, the larger groups will consume more compute resources and time during the aggregation process, resulting in slower processing speeds and data skew.
- JOIN operation skew

  During table JOIN operations, if the keys involved in the JOIN are unevenly distributed in one of the tables, a large amount of data will be concentrated in a few tasks while others have already completed, causing data skew.

Solution for Group By Data Skew

Select a subset of data and run **select count(*) as sum,Key from tbl group by Key order by sum desc** to identify which keys are causing data skew.

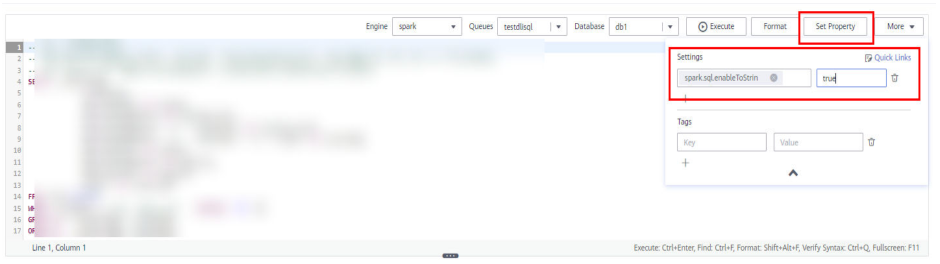
Then, for the skewed keys, you can handle them separately by adding a salt to split them into multiple tasks for individual statistics, and finally combine the results of the separate statistics.

For example, consider the following SQL query where **Key01** is identified as the skewed key causing a single task to process a large amount of data. The following steps can be taken to handle it:

```
SELECT
  a.Key,
  SUM(a.sum) AS Cnt
FROM
  (
    SELECT
      Key,
      count(*) AS sum
    FROM
      tbl
    GROUP BY
      Key,
      CASE
        WHEN KEY = 'Key01' THEN floor(random () * 200)
        ELSE 0
      END
  ) a
GROUP BY
  a.Key;
```

Solution for JOIN Data Skew

1. Log in to the DLI management console. Choose **Job Management** > **SQL Jobs** in the navigation pane. On the displayed page, locate the job you want to modify and click **Edit** in the **Operation** column to switch to the **SQL Editor** page.
2. On the **SQL editor** page, click **Set Property** and add the following Spark parameters through the **Settings** pane:



spark.sql.enableToStrin true

The string followed by the colons (:) are the configuration parameters, and the strings following the colons are the values.

```
spark.sql.enableToString:false
spark.sql.adaptive.join.enabled:true
spark.sql.adaptive.enabled:true
spark.sql.adaptive.skewedJoin.enabled:true
spark.sql.adaptive.enableToString:false
spark.sql.adaptive.skewedPartitionMaxSplits:10
```

NOTE

spark.sql.adaptive.skewedPartitionMaxSplits indicates the maximum number of tasks for processing a skewed partition. The default value is **5**, and the maximum value is **10**. This parameter is optional.

3. Click **Execute** to run the job again.

5.1.13 Why Does a SQL Job That Has Join Operations Stay in the Running State?

Symptom

A SQL job contains join operations. After the job is submitted, it is stuck in the Running state and no result is returned.

Possible Causes

When a Spark SQL job has join operations on small tables, all executors are automatically broadcast to quickly complete the operations. However, this increases the memory consumption of the executors. If the executor memory usage is too high, the job fails to be executed.

Solution

1. Check whether the **/*+ BROADCAST(u) */** falg is used to forcibly perform broadcast join in the executed SQL statement. If the flag is used, remove it.
2. Set **spark.sql.autoBroadcastJoinThreshold** to **-1**.
   a. Log in to the DLI management console and choose **Job Management** > **SQL Jobs**. In the **Operation** column of the failed job, click **Edit** to switch to the SQL editor page.
   b. Click **Settings** in the upper right corner. In the **Parameter Settings** area, add **spark.sql.autoBroadcastJoinThreshold** and set it to **-1**.
   c. Click **Execute** again to and view the job running result.

5.1.14 Why Is a SQL Job Stuck in the Submitting State?

The possible causes and solutions are as follows:

- After you purchase a DLI queue and submit a SQL job for the first time, wait for 5 to 10 minutes. After the cluster is started in the background, the submission will be successful.
- If the network segment of the queue is changed, wait for 5 to 10 minutes and then submit the SQL job immediately. After the cluster is re-created in the background, the submission is successful.
- The queue is a pay-per-use one. The queue is idle for more than one hour, and background resources have been released. You must wait for 5 to 10 minutes and then submit the SQL job. After the cluster is restarted in the background, the submission will be successful.

5.2 SQL Job O&M

5.2.1 Why Is Error "path obs://xxx already exists" Reported When Data Is Exported to OBS?

This message indicates that you are exporting data to an existing OBS path.

Solution:

- Create an OBS directory.

  You can create an OBS directory that does not exist to store the data to be exported.
- Delete the existing OBS directory.

  Deleting an existing OBS directory will also delete all data in the directory. Perform this operation with caution.
- Check permissions on the directory.

  Make sure you have the permission to access and write data to the OBS path. If you do not have the permission, contact the administrator to obtain it.

5.2.2 Why Is Error "SQL_ANALYSIS_ERROR: Reference 't.id' is ambiguous, could be: t.id, t.id.;" Displayed When Two Tables Are Joined?

This message indicates that the two tables to be joined contain the same column, but the owner of the column is not specified when the command is executed.

For example, tables tb1 and tb2 contain the **id** field.

Incorrect command:

```
select id from tb1 join tb2;
```

Correct command:

```
select tb1.id from tb1 join tb2;
```

5.2.3 Why Is Error "The current account does not have permission to perform this operation,the current account was restricted. Restricted for no budget." Reported when a SQL Statement Is Executed?

This message indicates that your operation is restricted because your account is in arrears or there is insufficient balance in your account.

Solution:

1. Check the account status.

   Check if your account is in arrears and top it up if necessary.

2. Relog in to the system.

   If the same error message persists after the top-up, log out of your account and log back in.

5.2.4 Why Is Error "There should be at least one partition pruning predicate on partitioned table XX.YYY" Reported When a Query Statement Is Executed?

When you query the partitioned table **XX.YYY**, the partition column is not specified in the search criteria.

A partitioned table can be queried only when the query condition contains at least one partition column.

Solution

Query a partitioned table by referring to the following example:

Assume that **partitionedTable** is a partitioned table and **partitionedColumn** is a partition column. The query statement is as follows:

```
SELECT * FROM partitionedTable WHERE partitionedColumn  = XXX
```

When querying each partitioned table, at least one partition condition must be included.

5.2.5 Why Is Error "IllegalArgumentException: Buffer size too small. size" Reported When Data Is Loaded to an OBS Foreign Table?

Symptom

The following error message is displayed when the LOAD DATA command is executed by a Spark SQL job to import data to a DLI table:

```
error.DLI.0001: IllegalArgumentException: Buffer size too small. size = 262144 needed = 2272881
```

In some cases ,the following error message is displayed:

```
error.DLI.0999: InvalidProtocolBufferException: EOF in compressed stream footer position: 3 length: 479
range: 0 offset: 3 limit: 479 range 0 = 0 to 479 while trying to read 143805 bytes
```

Possible Causes

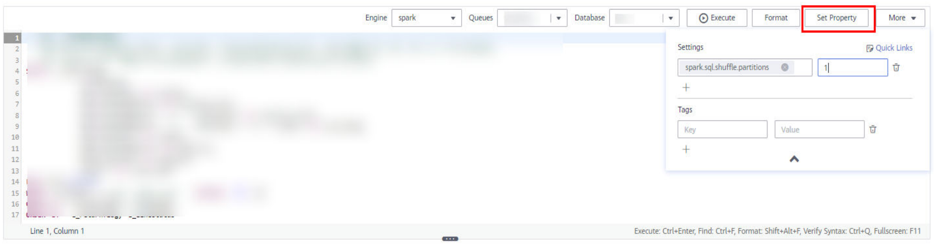
The data volume of the file to be imported is large and the value of **spark.sql.shuffle.partitions** is too large. As a result, the cache size is insufficient.

Solution

Decrease the **spark.sql.shuffle.partitions** value. To set this parameter, perform the following steps:

1. Log in to the DLI management console and choose **Job Management** > **SQL Jobs**. In the **Operation** column of the target SQL job, click **Edit** to go to the **SQL Editor** page.
2. On the displayed page, click **Set Property** and set the parameter.

Figure 5-16 Modifying the parameter


Engine
spark
Queues
Database
Execute
Format
Set Property
More
Settings
Quick Links
spark.sql.shuffle.partitions
1
Tags
Key
Value
Line 1, Column 1
Execute: Ctrl+Enter, Find: Ctrl+F, Format: Shift+Alt+F, Verify Syntax: Ctrl+Q, Fullscreen: F11

3. Execute the job again.

5.2.6 Why Is Error "DLI.0002 FileNotFoundException" Reported During SQL Job Running?

Symptom

An error is reported during SQL job execution:

```
Please contact DLI service. DLI.0002: FileNotFoundException: getFileStatus on obs://xxx: status [404]
```

Solution

Check whether there is another job that has deleted table information.

DLI does not allow multiple jobs to read and write the same table at the same time. Otherwise, job conflicts may occur and the jobs fail.

5.2.7 Why Is a Schema Parsing Error Reported When I Create a Hive Table Using CTAS?

Currently, DLI supports the creation of TEXTFILE, SEQUENCEFILE, RCFILE, ORC, AVRO, and PARQUET tables using the Hive syntax.

If you create a table using CTAS statements, specify the AVRO format, and then use a number as input for the SELECT query statement (for example, **CREATE TABLE tb_avro STORED AS AVRO AS SELECT 1**), a schema parsing exception will be reported.

If the column name is not specified, the content after SELECT is used as both the column name and inserted value. The column name of the AVRO table cannot be a digit. Otherwise, an error will be reported, indicating that the schema fails to be parsed.

Solution: You can use **CREATE TABLE tb_avro STORED AS AVRO AS SELECT 1 AS colName** to specify the column name or set the storage format to a format other than AVRO.

5.2.8 Why Is Error "org.apache.hadoop.fs.obs.OBSIOException" Reported When I Run DLI SQL Scripts on DataArts Studio?

Symptom

When you run a DLI SQL script on DataArts Studio, the log shows that the statements fail to be executed. The error information is as follows:

```
DLI.0999: RuntimeException: org.apache.hadoop.fs.obs.OBSIOException: initializing on obs://xxx.csv: status
[-1] - request id
[null] - error code [null] - error message [null] - trace :com.obs.services.exception.ObsException: OBS
servcie Error Message. Request Error:
...
Cause by: ObsException: com.obs.services.exception.ObsException: OBSs servcie Error Message. Request
Error: java.net.UnknownHostException: xxx: Name or service not known
```

Possible Causes

When you execute a DLI SQL script for the first time, you did not agree to the privacy agreement on the DLI console. As a result, the error is reported when the SQL script is executed on DataArts Studio.

Solution

1. Log in to the DLI console, click **SQL Editor** from the navigation pane. On the displayed page, enter an SQL statement in the editing window, for example, **select 1**.
2. In the displayed **Privacy Agreement** dialog box, agree to the terms.

   NOTE

   You only need to agree to the privacy agreement when it is your first time to execute the statements.
3. Run the DLI SQL script on DataArts Studio again. The script will run properly.

5.2.9 Why Is Error "UQUERY_CONNECTOR_0001:Invoke DLI service api failed" Reported in the Job Log When I Use CDM to Migrate Data to DLI?

Symptom

After the migration job is submitted, the following error information is displayed in the log:

```
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.UQueryWriter.close(UQueryWriter.java:42)
at org.apache.sqoop.connector.uquery.processor.Dataconsumer.run(Dataconsumer.java:217)
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)
```

Possible Causes

When you create a migration job to DLI on the CDM console, you set **Resource Queue** to a DLI queue for general purpose. It should be a queue for SQL.

Solution

1. On the DLI management console and click **Queue Management** in the navigation pane on the left. On the **Queue Management** page, check whether there are SQL queues.
   - If there are, go to **3**.
   - If there are no SQL queues, go to **2** to buy an SQL queue.
2. Choose **Resources** > **Resource Pool**. On the displayed page, locate the purchased elastic resource pool, and click **Add Queue** in the **Operation** column. Set **Type** to **For SQL**, set other parameters, and submit the creation request.
3. Go back to the CDM console and create a data migration job. Set **Resource Queue** to the created DLI SQL queue.
4. Submit the migration job and view the job execution logs.

5.2.10 Why Is Error "File not Found" Reported When I Access a SQL Job?

Symptom

Error message "File not Found" is displayed when a SQL job is accessed.

Possible Causes

- The system may not be able to locate the specified file path or file due to an incorrect file path or the file not existing.
- The file is in use.

Solution

- Check the file path and file name.

  Check whether the file path is correct, including the directory name and file name.
- Check if any jobs are overwriting the corresponding data.

  If the file cannot be found due to it being in use, it is usually caused by a read-write conflict. You are advised to check if any jobs are overwriting the corresponding data when querying the SQL error table.

5.2.11 Why Is Error "DLI.0003: AccessControlException XXX" Reported When I Access a SQL Job?

Symptom

Error message "DLI.0003: AccessControlException XXX" is reported when a SQL job is accessed.

Solution

Check the permissions of the OBS bucket to ensure that the account has access to the OBS bucket mentioned in the error message.

If not, contact the OBS bucket administrator to add access permissions of the bucket.

5.2.12 Why Is Error "DLI.0001: org.apache.hadoop.security.AccessControlException: verifyBucketExists on {{bucket name}}: status [403]" Reported When I Access a SQL Job?

Symptom

Error message "DLI.0001: org.apache.hadoop.security.AccessControlException: verifyBucketExists on {{bucket name}}: status [403]" is reported when a SQL job is Accessed.

Solution

Check the permissions of the OBS bucket to ensure that the account has access to the OBS bucket mentioned in the error message.

If not, contact the OBS bucket administrator to add access permissions of the bucket.

5.2.13 Why Am I Seeing the Error Message "The current account does not have permission to perform this operation,the current account was restricted. Restricted for no budget" When Executing a SQL Statement?

Symptom

Error message "The current account does not have permission to perform this operation,the current account was restricted." is reported during SQL statement execution.

Solution

1. Check the account status.

Check if your account is in arrears and top it up if necessary.

2. Relog in to the system.

   If the same error message persists after the top-up, log out of your account and log back in.

6 Flink Jobs

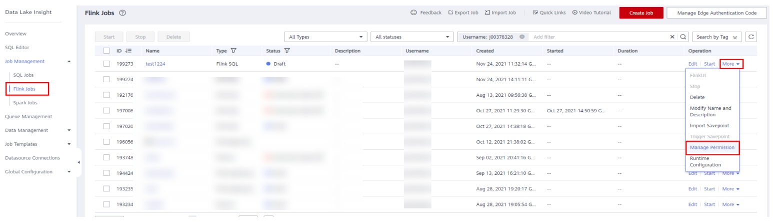
6.1 Flink Job Consulting

6.1.1 How Do I Authorize a Subuser to View Flink Jobs?

A sub-user can view queues but cannot view Flink jobs. You can authorize the sub-user using DLI or IAM.

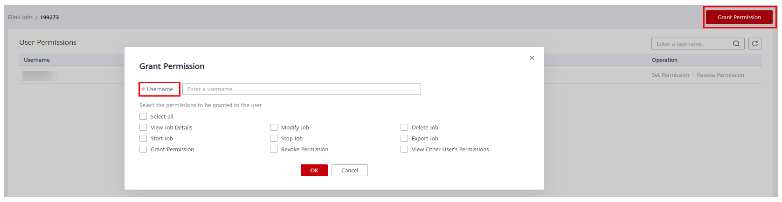
- Authorization on DLI
  a. Log in to the DLI console using a tenant account, a job owner account, or an account with the **DLI Service Administrator** permission.
  b. Choose **Job Management** > **Flink Jobs**. On the displayed page, locate the target job.
  c. In the **Operation** column of the target job, choose **More** > **Permissions**.

Figure 6-1 Managing Flink job permissions



  d. On the displayed page, click **Grant Permission**. Enter the name of the user to be authorized and select the required permissions. Click **OK**. The authorized user can view the job and perform related operations.

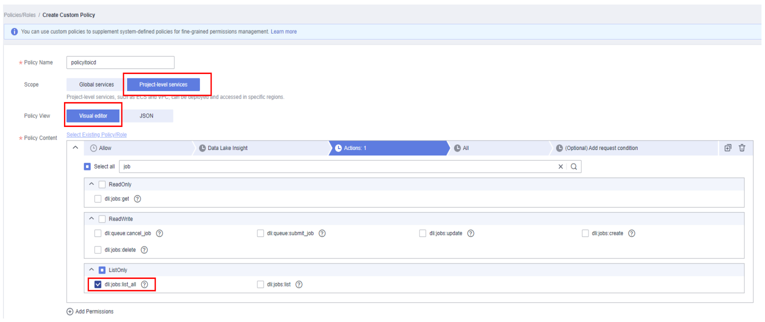
Figure 6-2 Getting permissions



- Authorization on IAM
  a. Log in to the IAM console. In the navigation pane, choose **Permissions** > **Policies/Roles**. On the displayed page, click **Create Custom Policy**.
  b. Create a permission policy for the subuser to view DLI Flink jobs.
     - **Policy Name**: Use the default name or customize a name.
     - **Scope**: Select **Project-level services**.
     - **Policy View**: Select **Visual editor**.
     - **Policy Content**: Select **Allow**, **Data Lake Insight**, and **dli:jobs:list_all** in sequence.

     Click **OK** to create the policy.

     Figure 6-3 Creating a policy

     Policies/Roles / Create Custom Policy

     You can use custom policies to supplement system-defined policies for fine-grained permissions management. Learn more

     * Policy Name: policyitoicd
     Scope: Global services | Project-level services
     Project-level services, such as ECS and VPC, can be deployed and accessed in specific regions.
     Policy View: Visual editor | JSON
     * Policy Content: Select Existing Policy/Role
     Allow | Data Lake Insight | Actions: 1 | All | (Optional) Add request condition
     Select all: job
     ReadOnly: dli:jobs:get
     ReadWrite: dli:queue:cancel_job, dli:queue:submit_job, dli:jobs:update, dli:jobs:create, dli:jobs:delete
     ListOnly: [✓ dli:jobs:list_all], dli:jobs:list
     Add Permissions

  c. Go to the **User Group** page, locate the user group to which the subuser to be authorized belongs and click the user group name. On the displayed page, click **Assign**.
  d. Grant permissions to the user group.
     - Select **Region-specific projects** for **Scope**.
     - Select the permission policy created in **b** for **Permissions**.

       You can also select **DLI Service Administrator** (with all DLI permissions) for the subuser to view Flink jobs.

6.1.2 How Do I Configure Auto Restart upon Exception for a Flink Job?

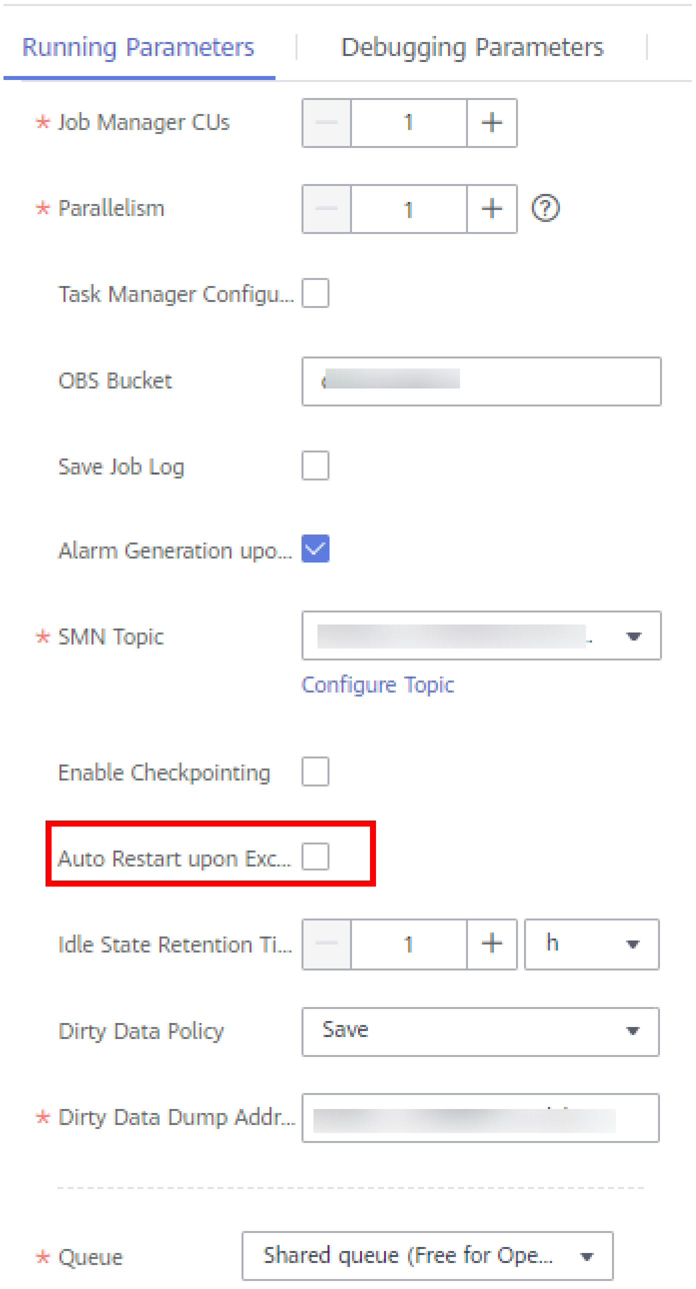
Scenario

DLI Flink jobs are highly available. You can enable the automatic restart function to automatically restart your jobs after short-time faults of peripheral services are rectified.

Procedure

1. Log in to the DLI console. In the navigation pane, choose **Job Management** > **Flink Jobs**.
2. Click a job name and select **Auto-restart upon exception** on the job editing page.

Figure 6-4 Editing a Flink SQL job



6.1.3 How Do I Save Logs for Flink Jobs?

When you create a Flink SQL job or Flink Jar job, you can select **Save Job Log** on the job editing page to save job running logs to OBS.

To set the OBS bucket for storing the job logs, specify a bucket for **OBS Bucket**. If the selected OBS bucket is not authorized, click **Authorize**.

The logs are saved in the following path: *Bucket name***/jobs/logs/***Directory starting with the job ID*. You can customize the bucket name in the path. **/jobs/logs/***Directory starting with the job ID* is a fixed format.

In the job list, click the job name. In the **Run Log** tab, click the provided OBS link to go to the path.

For how to create a Flink SQL job or Flink Jar job, see **Data Lake Insight User Guide**.

6.1.4 Why Is Error "No such user. userName:xxxx." Reported on the Flink Job Management Page When I Grant Permission to a User?

Symptom

Choose **Job Management** > **Flink Jobs**. In the **Operation** column of the target job, choose **More** > **Permissions**. When a new user is authorized, **No such user. userName:xxxx.** is displayed.

Solution

The issue is caused by the system's failure to identify new user information.

Perform the following steps:

1. Check whether the username exists.
2. If the username exists, relog in to the management console to grant permissions.

6.1.5 How Do I Restore a Flink Job from a Specific Checkpoint After Manually Stopping the Job?

Symptom

Checkpoint was enabled when a Flink job is created, and the OBS bucket for storing checkpoints was specified. I am not sure how to restore a Flink job from a specific checkpoint after manually stopping the job.

Solution

Since the Flink checkpoint and savepoint generation mechanisms and formats are consistent, you can restore the Flink job from the latest successful checkpoint in OBS. Specifically, in the Flink job list, locate the desired Flink job, click **More** in the **Operation** column, and select **Import Savepoint** to import the checkpoint.

1. Log in to the DLI console. In the navigation pane, choose **Job Management** > **Flink Jobs**.
2. Locate the row that contains the target Flink job, and click **Import Savepoint** in the **Operation** column.
3. In the displayed dialog box, select the OBS bucket path storing the checkpoint. The checkpoint save path is *Bucket name***/jobs/checkpoint/***directory starting with the job ID*. Click **OK**.
4. Start the Flink job again. The job will be restored fom the imported savepoint.

6.1.6 Why Is a Message Displayed Indicating That the SMN Topic Does Not Exist When I Use the SMN Topic in DLI?

When you set running parameters of a DLI Flink job, you can enable **Alarm Generation upon Job Exception** to receive alarms when the job runs abnormally or is in arrears.

If a message is displayed indicating that the SMN topic does not exist, perform the following steps:

1. Check whether the SMN topic has been created.

   If not, create a topic on the SMN console.

   For details about how to customize an SMN topic, see **Creating a Topic**.

2. Check IAM permissions.

   If the SMN topic already exists but the system still prompts that it does not exist, go to the IAM service, select the user group where the corresponding IAM user is, and ensure that the SMN policy for the corresponding region has been added to the user group.

3. Confirm the topic name and region.

   Make sure that the SMN topic name and region configured in DLI match the actual SMN topic created. If the SMN topic name is not consistent, the system will prompt that the SMN topic does not exist.

6.2 Flink SQL Jobs

6.2.1 How Do I Map an OBS Table to a DLI Partitioned Table?

Scenario

When using a Flink SQL job, you need to create an OBS partition table for subsequent batch processing.

Procedure

In the following example, the **day** field is used as the partition field with the parquet encoding format to dump **car_info** data to OBS. For details, see **Flink SQL Syntax Reference**.

```
create sink stream car_infos (
  carId string,
  carOwner string,
  average_speed double,
  day string
  ) partitioned by (day)
  with (
    type = "filesystem",
    file.path = "obs://obs-sink/car_infos",
    encode = "parquet",
    ak = "{{myAk}}",
    sk = "{{mySk}}"
);
```

Structure of the data storage directory in OBS: **obs://obs-sink/car_infos/day=xx/part-x-x**.

After the data is generated, the OBS partition table can be established for subsequent batch processing through the following SQL statements:

1. Create an OBS partitioned table.
```
create table car_infos (
  carId string,
  carOwner string,
  average_speed double
)
  partitioned by (day string)
  stored as parquet
  location 'obs://obs-sink/car-infos';
```
2. Restore partition information from the associated OBS path.
```
alter table car_infos recover partitions;
```

6.2.2 How Do I Change the Number of Kafka Partitions in a Flink SQL Job Without Stopping It?

Symptom

You used Flink 1.10 to run a Flink Opensource SQL job. You set the number of Kafka partitions for the job a small value at the beginning and need to increase the number now.

Solution

Add the following parameters to the SQL statement:

```
connector.properties.flink.partition-discovery.interval-millis="3000"
```

This statement allows you to increase or decrease the number of Kafka partitions without stopping the Flink job.

6.2.3 How Do I Fix the DLI.0005 Error When Using EL Expressions to Create a Table in a Flink SQL Job?

Symptom

When I run the creation statement with an EL expression in the table name in a Flink SQL job, the following error message is displayed:
```
DLI.0005: AnalysisException: t_user_message_input_#{date_format(date_sub(current_date(), 1),
'yyyymmddhhmmss')} is not a valid name for tables/databases. Valid names only contain alphabet
characters, numbers and _.
```

Solution

Replace the number sign (#) in the table name to the dollar sign ($). The format of the EL expression used in DLI should be **${*expr*}**.

Before modification:

```
t_user_message_input_#{date_format(date_sub(current_date(), 1), 'yyyymmddhhmmss')}
```

After modification:

```
t_user_message_input_${date_format(date_sub(current_date(), 1), 'yyyymmddhhmmss')}
```

After the modification, the Flink SQL job can correctly parse table names and dynamically generate table names based on EL expressions.

6.2.4 Why Is No Data Queried in the DLI Table Created Using the OBS File Path When Data Is Written to OBS by a Flink Job Output Stream?

Symptom

After data is written to OBS through the Flink job output stream, data cannot be queried from the DLI table created in the OBS file path.

For example, use the following Flink result table to write data to the **obs://obs-sink/car_infos** path in OBS.

```
create sink stream car_infos_sink (
  carId string,
  carOwner string,
  average_speed double,
  buyday string
  ) partitioned by (buyday)
  with (
    type = "filesystem",
    file.path = "obs://obs-sink/car_infos",
    encode = "parquet",
    ak = "{{myAk}}",
    sk = "{{mySk}}"
);
```

Use the following statements to create a DLI partition table with data retrieved from the OBS file path. No data is found when you query the **car_infos** table on DLI.

```
create table car_infos (
  carId string,
  carOwner string,
  average_speed double
)
  partitioned by (buyday string)
  stored as parquet
  location 'obs://obs-sink/car_infos';
```

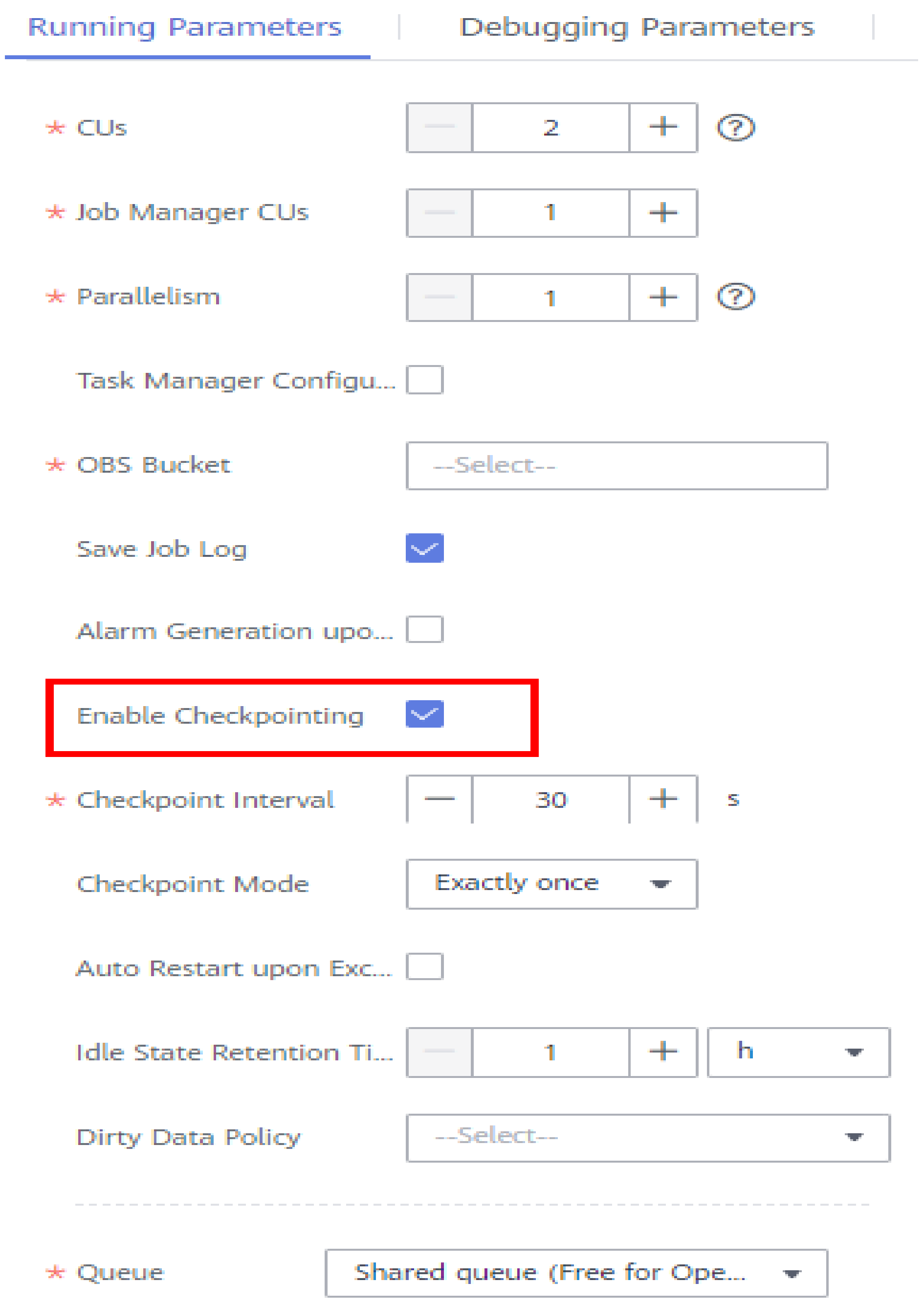
Solution

1. Check whether checkpointing is enabled for the Flink result table (**car_infos_sink** in the preceding example) when you create the job on DLI. If checkpointing is disabled, enable it and run the job again to generate OBS data files.

   To enable checkpointing, perform the following steps:

   a. Log in to the DLI management console. Choose **Job Management** > **Flink Jobs** in the left navigation pane. Locate the row that contains the target Flink job and click **Edit** in the **Operation** column.
   b. In the **Running Parameters** area, check whether **Enable Checkpointing** is enabled.

Figure 6-5 Enable checkpointing



* CUs	2
* Job Manager CUs	1
* Parallelism	1
Task Manager Configu...	☐
* OBS Bucket	--Select--
Save Job Log	✓
Alarm Generation upo...	☐
Enable Checkpointing	✓
* Checkpoint Interval	30 s
Checkpoint Mode	Exactly once
Auto Restart upon Exc...	☐
Idle State Retention Ti...	1 h
Dirty Data Policy	--Select--
* Queue	Shared queue (Free for Ope...

2. Check whether the structure of the Flink result table is the same as that of the DLI partitioned table. For the preceding example, check whether the fields of the **car_infos_sink** table are consistent with those of the **car_infos** table.
3. Check whether the partitioning information of the the partitioned table is restored after it is created using the OBS file. The following statement restore partitions of the **car_infos** table:

```
alter table car_infos recover partitions;
```

6.2.5 Why Does a Flink SQL Job Fails to Be Executed, and Is "connect to DIS failed java.lang.IllegalArgumentException: Access key cannot be null" Displayed in the Log?

Symptom

After a Flink SQL job is submitted on DLI, the job fails to be executed. The following error information is displayed in the job log:

```
connect to DIS failed java.lang.IllegalArgumentException: Access key cannot be null
```

Possible Causes

When configuring job running parameters for the Flink SQL job, Save Job Log or Checkpointing is enabled, and an OBS bucket for saving job logs and Checkpoints is configured. However, the IAM user who runs the Flink SQL job does not have the OBS write permission.

Solution

1. Log in to the IAM console, search for the IAM user who runs the job in the upper left corner of the **Users** page.
2. Click the desired username to view the user group where the user belongs.
3. In the navigation pane on the left, choose **User Groups**, and search for the user group of the target user. Click the user group name, and view the permissions of the current user in **Permissions**.
4. Check whether the user group has the permission to write data to OBS, for example, **OBS OperateAccess**. If the user group does not have the OBS write permission, grant the permission to the user group.
5. Wait for 5 to 10 minutes for the permission to take effect. Run the Flink SQL job again and check the job running status.

6.2.6 Data Writing Fails After a Flink SQL Job Consumed Kafka and Sank Data to the Elasticsearch Cluster

Symptom

After a Flink SQL job consumed Kafka and sent data to the Elasticsearch cluster, the job was successfully executed, but no data is available.

Cause Analysis

Possible causes are as follows:

- The data format is incorrect.
- The data cannot be processed.

Procedure

Step 1 Check the task log on the Flink UI. The JSON body is contained in the error message, indicating that the data format is incorrect.

Step 2 Check the data format. The Kafka data contains nested JSON bodies, which cannot be parsed.

Step 3 Use either of the following methods to solve the problem:

- Create a JAR file with the UDF.
- Modify the configuration data.

Step 4 Change the data format and execute the job again.

----End

6.2.7 How Does Flink Opensource SQL Parse Nested JSON?

- **Kafka message**

```
{
  "id": 1234567890,
  "name": "swq",
  "date": "1997-04-25",
  "obj": {
    "time1": "12:12:12",
    "str": "test",
    "lg": 1122334455
  },
  "arr": [
    "ly",
    "zpk",
    "swq",
    "zjy"
  ],
  "rowinarr": [
    {
      "f1": "f11",
      "f2": 111
    },
    {
      "f1": "f12",
      "f2": 222
    }
  ],
  "time": "13:13:13",
  "timestamp": "1997-04-25 14:14:14",
  "map": {
    "flink": 123
  },
  "mapinmap": {
    "inner_map": {
      "key": 234
    }
  }
}
```

- **Flink Opensource SQL**

```
create table kafkaSource(
  id            BIGINT,
  name          STRING,
  `date`        DATE,
  obj           ROW<time1 TIME,str STRING,lg BIGINT>,
  arr           ARRAY<STRING>,
  rowinarr      ARRAY<ROW<f1 STRING,f2 INT>>,
  `time`        TIME,
  `timestamp`   TIMESTAMP(3),
  `map`         MAP<STRING,BIGINT>,
  mapinmap      MAP<STRING,MAP<STRING,INT>>
) with (
  'connector' = 'kafka',
  'topic' = 'topic-swq-3',
  'properties.bootstrap.servers' = '10.128.0.138:9092,10.128.0.119:9092,10.128.0.212:9092',
  'properties.group.id' = 'swq-test',
  'scan.startup.mode' = 'latest-offset',
  'format' = 'json'
);
create table printSink (
  id             BIGINT,
  name           STRING,
  `date`         DATE,
  str            STRING,
  arr            ARRAY<STRING>,
  nameinarray    STRING,
  rowinarr       ARRAY<ROW<f1 STRING,f2 INT>>,
  f2             INT,
  `time`         TIME,
  `timestamp`    TIMESTAMP(3),
  `map`          MAP<STRING,BIGINT>,
  flink          BIGINT,
  mapinmap       MAP<STRING,MAP<STRING,INT>>,
  `key`          INT
) with ('connector' = 'print');

insert into
  printSink
select
  id,
  name,
  `date`,
  obj.str,
  arr,
  arr[4],
  rowinarr,
  rowinarr[1].f2,
  `time`,
  `timestamp`,
  `map`,
  `map`['flink'],
  mapinmap,
  mapinmap['inner_map']['key']
from kafkaSource;
```

- **Result**

```
+I(1234567890,swq,1997-04-25,test,[ly, zpk, swq, zjy],zjy,[f11,111,
f12,222],111,13:13:13,1997-04-25T14:14:14,{flink=123},123,{inner_map={key=234}},234)
```

CAUTION

1. Use the following methods to obtain elements in the containers of different types.
   - map: map['key']
   - array: array[index]
   - row: row.key
2. The index of an array starts from 1. Array[1] is the first element.
3. The elements of an array must be of the same type, and the elements of a row can be of different types.

6.2.8 Why Is the Time Read by a Flink OpenSource SQL Job from the RDS Database Is Different from the RDS Database Time?

Symptom

The time read by a Flink OpenSource SQL job from the RDS database is inconsistent with the RDS database time.

Possible Causes

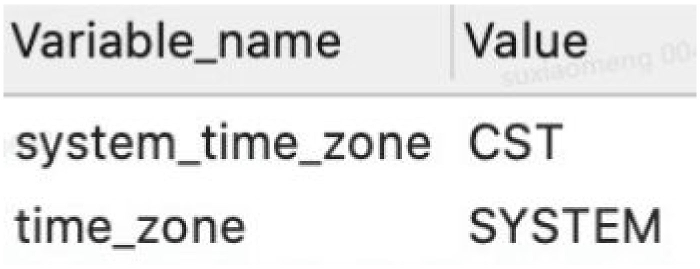
The time zone of a database is improperly set. Generally, there is a 13-hour difference.

Run the following statement in the RDS database:

```
show variables like '%time_zone%'
```

The result is as follows:

Figure 6-6 Execution result



```
Variable_name      Value
system_time_zone   CST
time_zone          SYSTEM
```

Table 6-1 Parameters

Parameter	Description
system_time_zone	Database time zone The value **SYSTEM** indicates the system time of the database server. The value of the system time zone is **CST**. Therefore, the database time zone is **CST**.
time_zone	Time zone of the server where the database is located. The server is a computer. If the default time zone of the computer where the local database is located is China Standard Time, the value of **system_time_zone** is **CST**.

Root cause: A bug can occur when **time_zone** in MySQL is set to **SYSTEM** and **system_time_zone** is set to **CST**.

In MySQL, CST refers to China Standard Time, which is UTC+08:00. However, in Java, CST stands for Central Standard Time (USA), which is UTC–05:00.

Flink TaskManager is a Java process that reads the time zone set in the JDBC driver code of the MySQL database through **TimeZone.getTimeZone(canonicalTimezone)**. The method gets time zone CST (UTC+8), but the actual time zone is CST (UTC-5).

Solution

1. Do not set the value of **time_zone** to **SYSTEM**. You can set it to, for example, **+08:00**.
2. Include the time zone when setting **jdbcUrl**.

   For example, **jdbc:mysql://localhost:3306/test?serverTimezone=Asia/Shanghai**.

6.2.9 Why Does Job Submission Fail When the failure-handler Parameter of the Elasticsearch Result Table for a Flink Opensource SQL Job Is Set to retry_rejected?

Description

Job submission fails when the **failure-handler** parameter of the Elasticsearch result table for a Flink Opensource SQL job is set to **retry_rejected**.

Possible Causes

This is a design defect in the open-source component.

Solution

Change **retry_rejected** to **retry-rejected**.

6.2.10 How Do I Configure Connection Retries for Kafka Sink If it is Disconnected?

Symptom

You used Flink 1.10 to run a Flink Opensource SQL job. The job failed after the following error was reported when Flink Sink wrote data to Kafka.

```
Caused by: org.apache.kafka.common.errors.NetworkException: The server disconnected before a response was received.
```

Possible Causes

The CPU usage is too high. As a result, the network is intermittently disconnected.

Solution

Add **connector.properties.retries=5** to the SQL statement.

```
create table kafka_sink(
    car_type string
   , car_name string
```

```
    , primary key (union_id) not enforced
) with (
    "connector.type" = "upsert-kafka",
    "connector.version" = "0.11",
    "connector.properties.bootstrap.servers" = "xxxx:9092",
    "connector.topic" = "kafka_car_topic ",
    "connector.sink.ignore-retraction" = "true",
    "connector.properties.retries" = "5",
    "format.type" = "json"
);
```

6.2.11 How Do I Write Data to Different Elasticsearch Clusters in a Flink Job?

In a Flink job, you can use **CREATE** statements to define source and sink tables, and specify their connector types and related attributes.

If you need to write data to different Elasticsearch clusters, you need to configure different connection parameters for each cluster and ensure that the Flink job can correctly route data to each cluster.

In this example, the connector type and related attributes are defined for es1 and es2.

Add the following SQL statements to the Flink job:

```
create source stream ssource(xx);
create sink stream es1(xx) with (xx);
create sink stream es2(xx) with (xx);
insert into es1 select * from ssource;
insert into es2 select * from ssource;
```

6.2.12 Why Does DIS Stream Not Exist During Job Semantic Check?

To rectify this fault, perform the following steps:

1. Log in to the DIS management console. In the navigation pane, choose **Stream Management**. View the Flink job SQL statements to check whether the DIS stream exists.
2. If the DIS stream was not created, create a DIS stream by referring to "Creating a DIS Stream" in the **Data Ingestion Service User Guide**.

   Ensure that the created DIS stream and Flink job are in the same region.
3. If a DIS stream has been created, check whether the DIS stream and the Flink job are in the same region.

6.2.13 Why Is Error "Timeout expired while fetching topic metadata" Repeatedly Reported in Flink JobManager Logs?

If the Flink JobManager prompts "Timeout expired while fetching topic metadata", it means that the Flink job timed out while trying to fetch metadata for the Kafka topic.

In this case, you need to first check the network connectivity between the Flink job and Kafka, and ensure that the queue where the Flink job is executed can access the VPC network where Kafka is located.

If the network is unreachable, configure the network connectivity first and then re-execute the job.

6.3 Flink Jar Jobs

6.3.1 Can I Upload Configuration Files for Flink Jar Jobs?

Uploading a Configuration File for a Flink Jar Job

You can upload configuration files for custom jobs (Jar).

1. Upload the configuration file to DLI through **Package Management**.
2. In the **Other Dependencies** area of the Flink Jar job, select the created DLI package.
3. Load the file through **ClassName.class.getClassLoader().getResource("userData/fileName")** in the code.
   – **ClassName** indicates the name of the class that needs to access the file.
   – **userData** is a fixed file path name, which cannot be changed or customized.
   – **fileName** indicates the name of the file to be accessed.

NOTE

The examples in this section only apply to Flink 1.12. For how to develop Flink 1.15 Jar jobs, see **Using Flink Jar to Write Data to OBS**.

Using a Configuration File

- Solution 1: Load the file content to the memory in the **main** function and broadcast the content to each taskmanager. This method is applicable to the scenario where a small number of variables need to be loaded in advance.
- Solution 2: Load the file when initializing the operator in **open**. A relative or absolute path can be used.

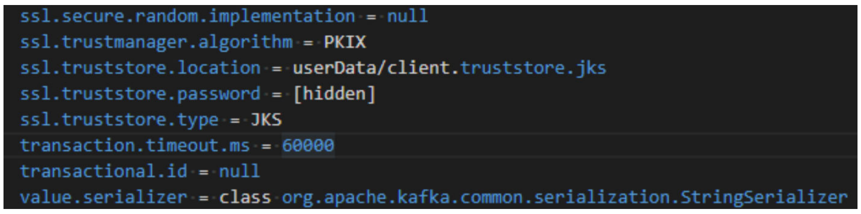
  Take Kafka sink as an example. Two files (**userData/kafka-sink.conf** and **userData/client.truststore.jks**) need to be loaded.

  – **Example of using a relative path:**

```
Relative path: confPath = userData/kafka-sink.conf
@Override
public void open(Configuration parameters) throws Exception {
    super.open(parameters);
    initConf();
    producer = new KafkaProducer<>(props);
}
private void initConf() {
    try {
        URL url = DliFlinkDemoDis2Kafka.class.getClassLoader().getResource(confPath);
        if (url != null) {
            LOGGER.info("kafka main-url: " + url.getFile());
        } else {
            LOGGER.info("kafka url error......");
        }
        InputStream inputStream = new BufferedInputStream(new FileInputStream(new
File(url.getFile()).getAbsolutePath()));
```

```
        props.load(new InputStreamReader(inputStream, "UTF-8"));
        topic = props.getProperty("topic");
        partition = Integer.parseInt(props.getProperty("partition"));
        vaildProps();
    } catch (Exception e) {
        LOGGER.info("load kafka conf failed");
        e.printStackTrace();
    }
}
```

Figure 6-7 Example relative path configuration

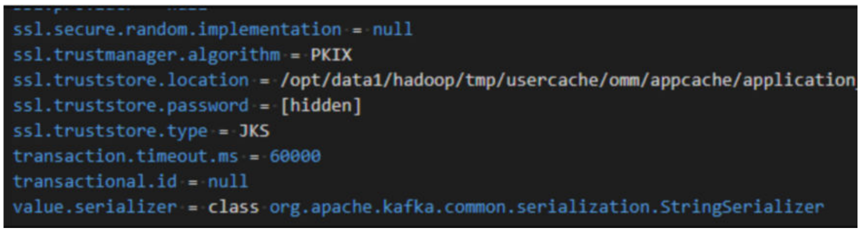


```
ssl.secure.random.implementation = null
ssl.trustmanager.algorithm = PKIX
ssl.truststore.location = userData/client.truststore.jks
ssl.truststore.password = [hidden]
ssl.truststore.type = JKS
transaction.timeout.ms = 60000
transactional.id = null
value.serializer = class org.apache.kafka.common.serialization.StringSerializer
```

– **Example of using an absolute path:**

```
Absolute path: confPath = userData/kafka-sink.conf / path = /opt/data1/hadoop/tmp/
usercache/omm/appcache/application_xxx_0015/container_xxx_0015_01_000002/userData/
client.truststore.jks
@Override
public void open(Configuration parameters) throws Exception {
    super.open(parameters);
    initConf();
    String path = DliFlinkDemoDis2Kafka.class.getClassLoader().getResource("userData/
client.truststore.jks").getPath();
    LOGGER.info("kafka abs path " + path);
    props.setProperty("ssl.truststore.location", path);
    producer = new KafkaProducer<>(props);
}
private void initConf() {
    try {
        URL url = DliFlinkDemoDis2Kafka.class.getClassLoader().getResource(confPath);
        if (url != null) {
            LOGGER.info("kafka main-url: " + url.getFile());
        } else {
            LOGGER.info("kafka url error......");
        }
        InputStream inputStream = new BufferedInputStream(new FileInputStream(new
File(url.getFile()).getAbsolutePath()));
        props.load(new InputStreamReader(inputStream, "UTF-8"));
        topic = props.getProperty("topic");
        partition = Integer.parseInt(props.getProperty("partition"));
        vaildProps();
    } catch (Exception e) {
        LOGGER.info("load kafka conf failed");
        e.printStackTrace();
    }
}
```

Figure 6-8 Example absolute path configuration



```
ssl.secure.random.implementation = null
ssl.trustmanager.algorithm = PKIX
ssl.truststore.location = /opt/data1/hadoop/tmp/usercache/omm/appcache/application
ssl.truststore.password = [hidden]
ssl.truststore.type = JKS
transaction.timeout.ms = 60000
transactional.id = null
value.serializer = class org.apache.kafka.common.serialization.StringSerializer
```

6.3.2 Why Does a Flink Jar Package Conflict Result in Job Submission Failure?

Symptom

The dependency of your Flink job conflicts with a built-in dependency of the DLI Flink platform. As a result, the job submission fails.

Solution

Check whether there are conflicting JAR files.

DLI Flink provides a series of pre-installed dependency packages in the DLI service to support various data processing and analysis tasks.

If the JAR file you upload contains a package that already exists in the DLI Flink runtime platform, it will prompt a Flink Jar package conflict, resulting in job submission failure.

Delete the repetitive package by referring to the dependency package information provided in the *Data Lake Insight User Guide* and then upload the package.

For details about DLI built-in dependencies, see **Built-in Dependencies**.

6.3.3 Why Does a Flink Jar Job Fail to Access GaussDB(DWS) and a Message Is Displayed Indicating Too Many Client Connections?

Symptom

When a Flink Jar job is submitted to access GaussDB(DWS), an error message is displayed indicating that the job fails to be started. The job log contains the following error information:

```
FATAL:  Already too many clients, active/non-active/reserved: 5/508/3
```

Cause Analysis

The number of GaussDB(DWS) database connections exceeds the upper limit. In the error information, the value of **non-active** indicates the number of idle connections. For example, if the value of **non-active** is 508, there are 508 idle connections.

Solution

Perform the following steps to solve the problem:

1. Log in to the GaussDB(DWS) command window and run the following SQL statement to release all idle (non-active) connections temporarily:
```
SELECT PG_TERMINATE_BACKEND(pid) from pg_stat_activity WHERE state='idle';
```
2. Check whether the application actively releases the connections. If the application does not, optimize the code to release the connections.

3. On the GaussDB (DWS) management console, configure parameter **session_timeout**, which controls the timeout period of idle sessions. After an idle session's timeout period exceeds the specified value, the server automatically closes the connection.

   The default value of this parameter is **600** seconds. The value **0** indicates that the timeout limit is disabled. Do not set **session_timeout** to **0**.

   The procedure for setting parameter **session_timeout** is as follows:

   a. Log in to the GaussDB(DWS) management console.
   b. In the navigation pane on the left, click **Clusters**.
   c. In the cluster list, find the target cluster and click its name. The **Basic Information** page is displayed.
   d. Click the **Parameter Modifications** tab and modify the value of parameter **session_timeout**. Then click **Save**.
   e. In the **Modification Preview** dialog box, confirm the modification and click **Save**.

For more information, see **An Error Indicating Too Many Client Connections**.

6.3.4 Why Is Error Message "Authentication failed" Displayed During Flink Jar Job Running?

Symptom

An exception occurred when a Flink Jar job is running. The following error information is displayed in the job log:

```
org.apache.flink.shaded.curator.org.apache.curator.ConnectionState - Authentication failed
```

Possible Causes

Service authorization is not configured for the account on the **Global Configuration** page. When the account is used to create a datasource connection to access external data, the access fails.

Solution

Step 1 Log in to the DLI management console. In the navigation pane on the left, choose **Global Configuration** > **Service Authorization**.

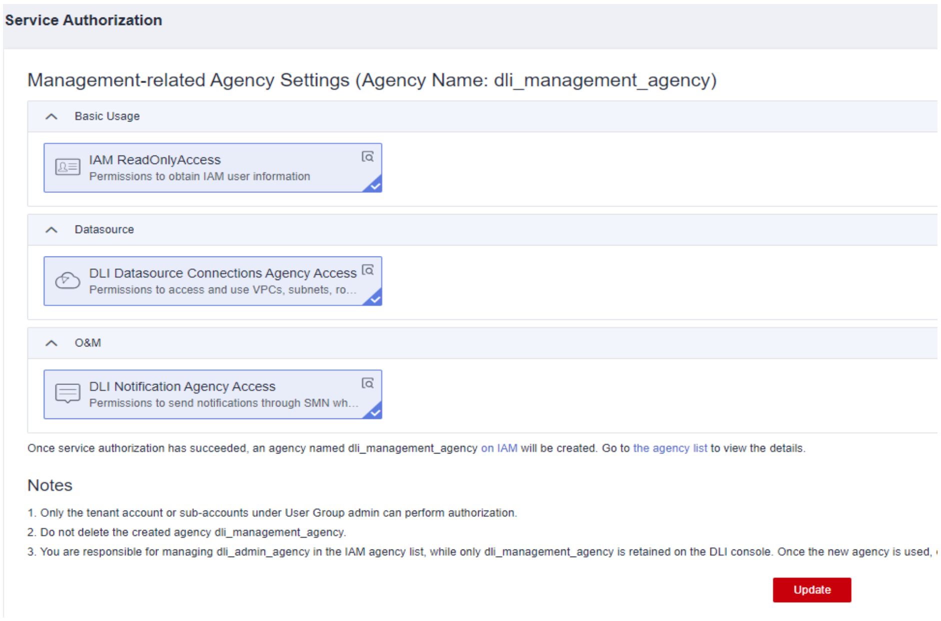
Step 2 On the agency settings page, select required permissions.

DLI Datasource Connections Agency Access is the permissions to access and use VPCs, subnets, routes, and VPC peering connections in datasource scenarios.

For more information, see **DLI Agency Permissions**.

Step 3 Select the permissions to be included in **dli_management_agency** and click **Update**.

Figure 6-9 Updating agency permissions



Service Authorization

Management-related Agency Settings (Agency Name: dli_management_agency)

Basic Usage

IAM ReadOnlyAccess
Permissions to obtain IAM user information

Datasource

DLI Datasource Connections Agency Access
Permissions to access and use VPCs, subnets, ro...

O&M

DLI Notification Agency Access
Permissions to send notifications through SMN wh...

Once service authorization has succeeded, an agency named dli_management_agency on IAM will be created. Go to the agency list to view the details.

Notes

1. Only the tenant account or sub-accounts under User Group admin can perform authorization.
2. Do not delete the created agency dli_management_agency.
3. You are responsible for managing dli_admin_agency in the IAM agency list, while only dli_management_agency is retained on the DLI console. Once the new agency is used,

Step 4 After updating the agency, recreate the datasource connection and rerun the job.

----End

6.3.5 Why Is Error Invalid OBS Bucket Name Reported After a Flink Job Submission Failed?

Symptom

The storage path of the Flink Jar job checkpoints was set to an OBS bucket. The job failed to be submitted, and an error message indicating an invalid OBS bucket name was displayed.

Possible Causes

1. Check that the OBS bucket name is correct.
2. Check that the AK/SK has the required permission.
3. Set the dependency to **provided** to prevent JAR file conflicts.
4. Check that the **esdk-obs-java-3.1.3.jar** version is used.
5. Confirm that the cluster configuration is faulty.

Procedure

Step 1 Set the dependency to **provided**.

Step 2 Restart the **clusteragent** cluster after an upgrade to make the configuration take effect.

Step 3 Remove the OBS dependency. Otherwise, the checkpoints cannot be written to OBS.

----End

6.3.6 Why Does the Flink Submission Fail Due to Hadoop JAR File Conflict?

Symptom

Flink Job submission failed. The exception information is as follows:

```
Caused by: java.lang.RuntimeException: java.lang.ClassNotFoundException: Class
org.apache.hadoop.fs.obs.metrics.OBSAMetricsProvider not found
 at org.apache.hadoop.conf.Configuration.getClass(Configuration.java:2664)
 at org.apache.hadoop.conf.Configuration.getClass(Configuration.java:2688)
 ... 31 common frames omitted
 Caused by: java.lang.ClassNotFoundException: Class org.apache.hadoop.fs.obs.metrics.OBSAMetricsProvider
not found
 at org.apache.hadoop.conf.Configuration.getClassByName(Configuration.java:2568)
 at org.apache.hadoop.conf.Configuration.getClass(Configuration.java:2662)
 ... 32 common frames omitted
```

Cause Analysis

Flink JAR files conflicted. The submitted Flink JAR file conflicted with the HDFS JAR file of the DLI cluster.

Procedure

Step 1 Configure **hadoop-hdfs** in the POM file as follows:

```
<dependency>
<groupId>org.apache.hadoop</groupId>
<artifactId>hadoop-hdfs</artifactId>
<version>${hadoop.version}</version>
<scope> provided </scope>
</dependency>
```

Alternatively, use the **exclusions** tag to exclude the association.

Step 2 To use HDFS configuration files, change **core-site.xml**, **hdfs-site.xml**, and **yarn-site.xml** to **mrs-core-site.xml**, **mrs-hdfs-site.xml** and **mrs-hbase-site.xml**, respectively.

```
conf.addResource(HBaseUtil.class.getClassLoader().getResourceAsStream("mrs-core-site.xml"), false);
conf.addResource(HBaseUtil.class.getClassLoader().getResourceAsStream("mrs-hdfs-site.xml"), false);
conf.addResource(HBaseUtil.class.getClassLoader().getResourceAsStream("mrs-hbase-site.xml"), false);
```

----End

6.3.7 How Do I Locate a Flink Job Submission Error?

1. On the Flink job management page, hover the cursor on the status of the job that fails to be submitted to view the brief information about the failure.

   The possible causes are as follows:

   – Insufficient CUs: Increase the number of CUs of the queue.

   – Failed to generate the JAR file: Check the SQL syntax and UDFs.

2. If you cannot locate the fault or the call stack is incorrect, click the job name to go to the job details page and click the **Commit Logs** tab to view the job submission logs.

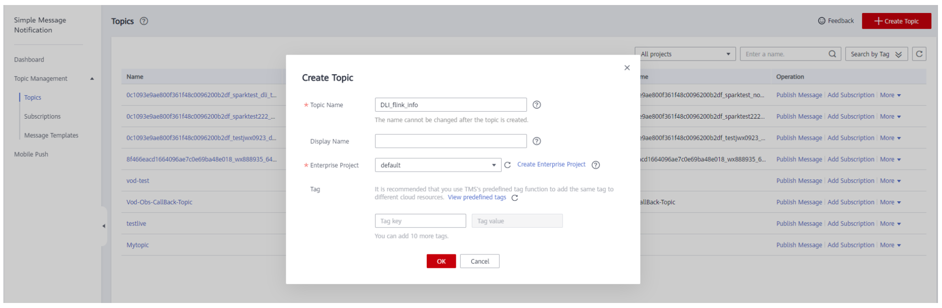
6.4 Flink Job Performance Tuning

6.4.1 What Is the Recommended Configuration for a Flink Job?

When you create a Flink job, you can perform the following operations to ensure high reliability of stream applications:

1. Create an SMN topic and add an email address or mobile number to subscribe to the topic. You will receive a subscription notification by an email or message. Click the link to confirm the subscription.

Figure 6-10 Creating a topic

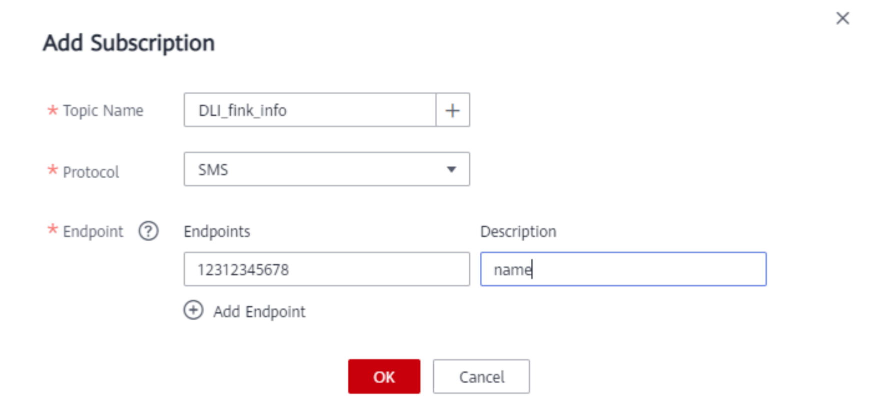


0c1093e9ae800f361f48c0096200b2df_sparktest_dli_t...
0c1093e9ae800f361f48c0096200b2df_sparktest222_...
0c1093e9ae800f361f48c0096200b2df_testjwx0923_d...
8f466eacd1664096ae7c0e69ba48e018_wx888935_64...
vod-test
Vod-Obs-CallBack-Topic
testlive
Mytopic

Create Topic

* Topic Name: DLI_flink_info
The name cannot be changed after the topic is created.
Display Name:
* Enterprise Project: default   Create Enterprise Project
Tag: It is recommended that you use TMS's predefined tag function to add the same tag to different cloud resources. View predefined tags
Tag key   Tag value
You can add 10 more tags.
OK   Cancel

Figure 6-11 Adding a subscription



Add Subscription

* Topic Name: DLI_fink_info
* Protocol: SMS
* Endpoint: Endpoints 12312345678   Description name
Add Endpoint
OK   Cancel

2. Log in to the DLI console, create a Flink SQL job, write SQL statements for the job, and configure running parameters.

NOTE

The reliability configuration of a Flink Jar job is the same as that of a SQL job.

a. Set **CUs**, Job **Manager CUs**, and **Max Concurrent Jobs** based on the following formulas:

Total number of CUs = Number of manager CUs + (Total number of concurrent operators / Number of slots of a TaskManager) x Number of TaskManager CUs

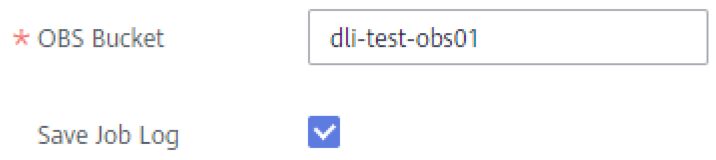
For example, if the total number of CUs is 9, the number of manager CUs is 1, and the maximum number of concurrent jobs is 16, the number of compute-specific CUs is 8.

If you do not configure TaskManager specification, a TaskManager occupies 1 CU by default and has no slot. To ensure a high reliability, set the number of slots of the TaskManager to 2, according to the preceding formula.

Set the maximum number of concurrent jobs twice the number of CUs.

b. Select **Save Job Log** and select an OBS bucket. If the bucket is not authorized, click **Authorize**. This allows job logs be saved to your OBS bucket after a job fails for fault locating.

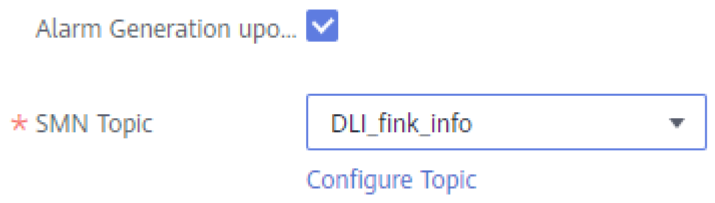
Figure 6-12 Specifying a bucket



* OBS Bucket	dli-test-obs01
Save Job Log	✓

c. Select **Alarm Generation upon Job Exception** and select the SMN topic created in **1**. This allows DLI to send notifications to your email box or phone when a job exception occurs, so you can be aware of any exceptions in time.

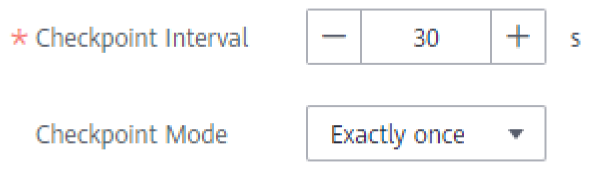
Figure 6-13 Alarm generation upon job exception



Alarm Generation upo...	✓
* SMN Topic	DLI_fink_info

d. Select **Enable Checkpointing** and set the checkpoint interval and mode as needed. This function ensures that a failed Flink task can be restored from the latest checkpoint.

Figure 6-14 Checkpoint parameters



* Checkpoint Interval	30 s
Checkpoint Mode	Exactly once

NOTE

- Checkpoint interval refers to the time interval between two consecutive checkpoint triggers. The checkpoint mechanism has an impact on real-time computing performance. When configuring the interval, consider the impact on business performance and recovery duration. You are advised to set the interval to be **greater than the checkpoint completion duration**, preferably 5 minutes.
- The **Exactly once** mode ensures that each piece of data is consumed only once, and the **At least once** mode ensures that each piece of data is consumed at least once. Select a mode as you need.

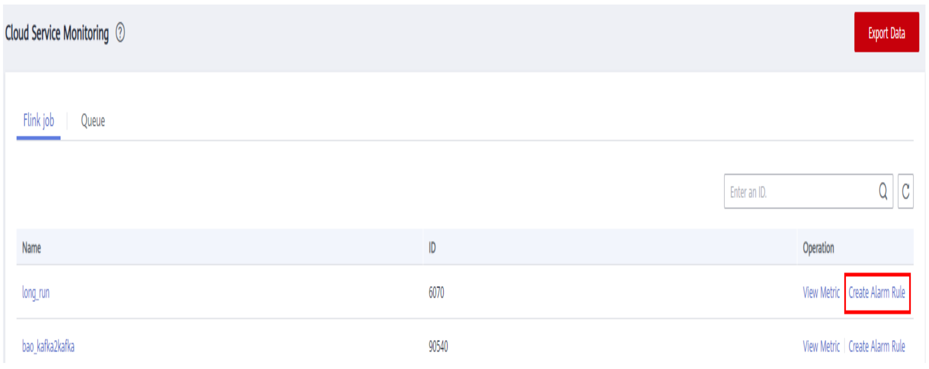
e. Select **Auto Restart upon Exception** and **Restore Job from Checkpoint**, and set the number of retry attempts as needed.

f. Configure **Dirty Data Policy**. You can select **Ignore**, **Trigger a job exception**, or **Save** based on your service requirements.

g. Select a queue and submit and run the job.

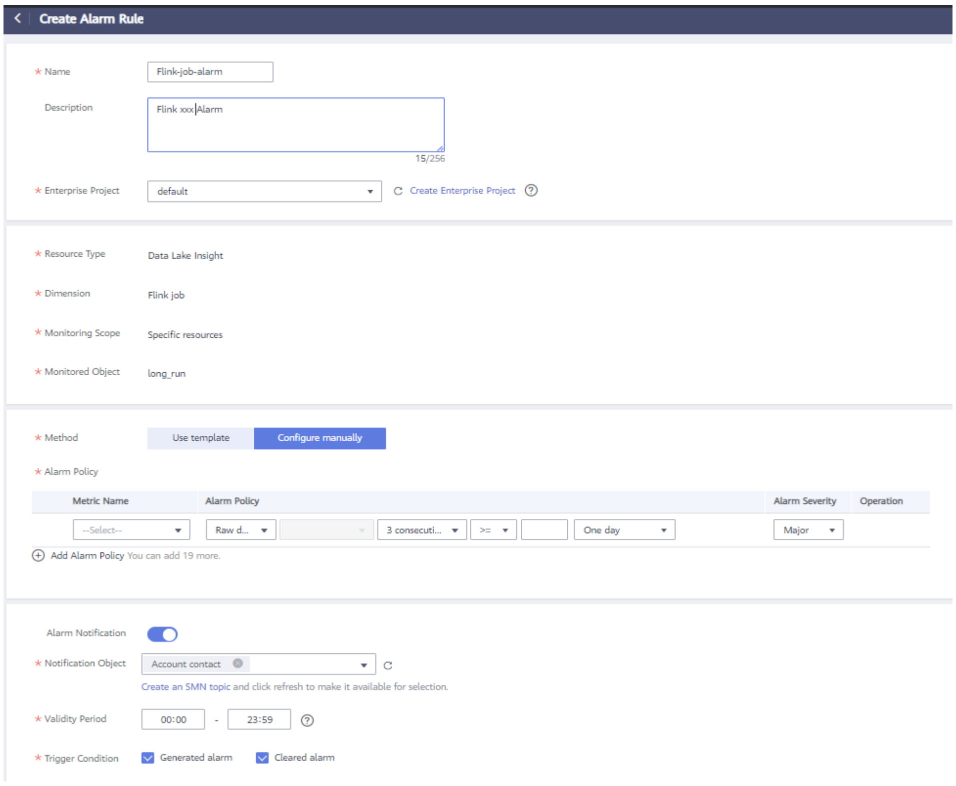
3. Log in to the Cloud Eye console. In the navigation pane on the left, choose **Cloud Service Monitoring** > **Data Lake Insight**. Locate the target Flink job and click **Create Alarm Rule**.

Figure 6-15 Cloud service monitoring



Name	ID	Operation
long_run	6070	View Metric | Create Alarm Rule
bao_kafka2kafka	90540	View Metric | Create Alarm Rule

Figure 6-16 Creating an alarm rule



Create Alarm Rule

* Name	Flink-job-alarm
Description	Flink xxx Alarm (15/256)
* Enterprise Project	default
* Resource Type	Data Lake Insight
* Dimension	Flink job
* Monitoring Scope	Specific resources
* Monitored Object	long_run
* Method	Use template [✓ Configure manually]
* Alarm Policy	

Metric Name	Alarm Policy						Alarm Severity	Operation
--Select--	Raw d...		3 consecuti...	>=		One day	Major	

Add Alarm Policy You can add 19 more.

Alarm Notification	On
* Notification Object	Account contact
	Create an SMN topic and click refresh to make it available for selection.
* Validity Period	00:00 - 23:59
* Trigger Condition	✓ Generated alarm ✓ Cleared alarm

DLI provides various monitoring metrics for Flink jobs. You can define alarm rules as required using different monitoring metrics for fine-grained job monitoring.

For details about the monitoring metrics, see **DLI Monitoring Metrics** in *Data Lake Insight User Guide*.

6.4.2 Flink Job Performance Tuning

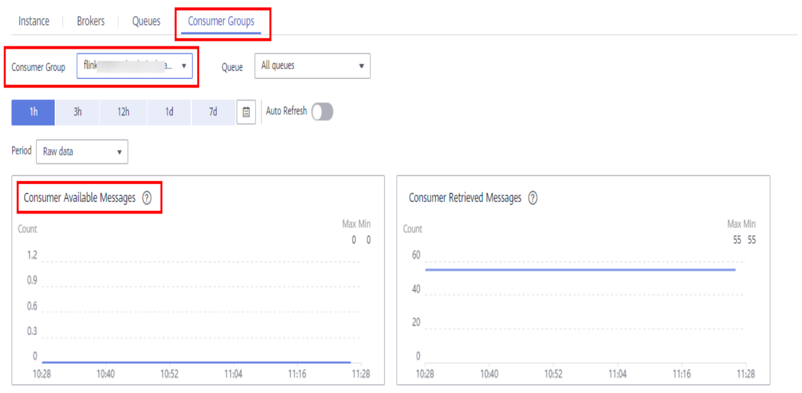
Basic Concepts of Performance Tuning

- Data Stacking in a Consumer Group

  The accumulated data of a consumer group can be calculated by the following formula: Total amount of data to be consumed by the consumer group = Offset of the latest data – Offset of the data submitted to the consumer group

  If your Flink job is connected to the Kafka premium edition, you can view the customer group on the Cloud Eye console. To view consumer available messages, choose **Cloud Service Monitoring** > **Distributed Message Service** form the navigation pane. On the displayed page, select **Kafka Premium** and click the **Consumer Groups** tab. Click the Kafka instance name and select the target consumer group.

Figure 6-17 Consumer group



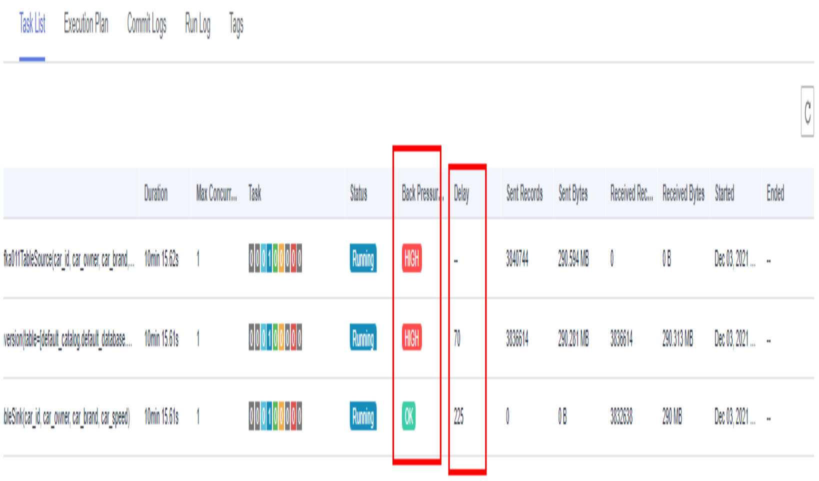
- Back Pressure Status

  Back pressure status is working load status of an operator. The back pressure is determined by the ratio of threads blocked in the output buffer to the total taskManager threads. This ratio is calculated by periodically sampling of the taskManager thread stack. By default, if the ratio is less than 0.1, the back pressure status is OK. If the ratio ranges from 0.1 to 0.5, the backpressure status is LOW. If the ratio exceeds 0.5, the backpressure status is HIGH.

- Delay

  Delay indicates the duration from the time when source data starts being processed to the time when data reaches the current operator. The data source periodically sends a LatencyMarker (current timestamp). After receiving the LatencyMarker, the downstream operator subtracts the timestamp from the current time to calculate the duration. You can view the back pressure status and delay of an operator on the Flink UI or in the task list of a job. Generally, high back pressure and delay occur in pairs.

Figure 6-18 Back pressure status and delay

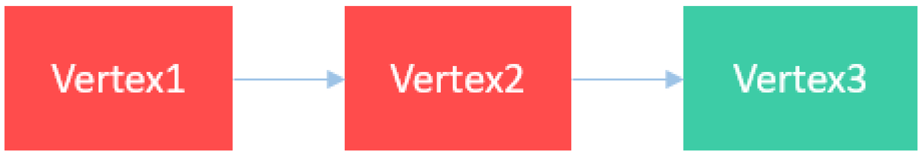


	Duration	Max Concurr...	Task	Status	Back Pressur...	Delay	Sent Records	Sent Bytes	Received Rec...	Received Bytes	Started	Ended
fka011TableSource(car_id, car_owner, car_brand,...	10min 15.62s	1	0 0 0 1 0 0 0 0 0	Running	HIGH	--	3840744	290.594 MB	0	0 B	Dec 03, 2021 ...	--
version(table=[default_catalog.default_database...	10min 15.61s	1	0 0 0 1 0 0 0 0 0	Running	HIGH	70	3836614	290.281 MB	3836614	290.313 MB	Dec 03, 2021 ...	--
bleSink(car_id, car_owner, car_brand, car_speed)	10min 15.61s	1	0 0 0 1 0 0 0 0 0	Running	OK	225	0	0 B	3832638	290 MB	Dec 03, 2021 ...	--

Performance Analysis

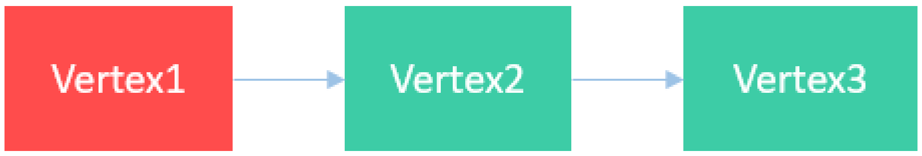
Due to Flink back pressure, the data source consumption rate can be lower than the production rate when performance of a Flink job is low. As a result, data is stacked in a Kafka consumer group. In this case, you can use back pressure and delay of the operator to find its performance bottleneck.

- The following figure shows that the back pressure of the last operator (sink) of the job is normal (green), and the back pressure of the previous two operators is high (red).

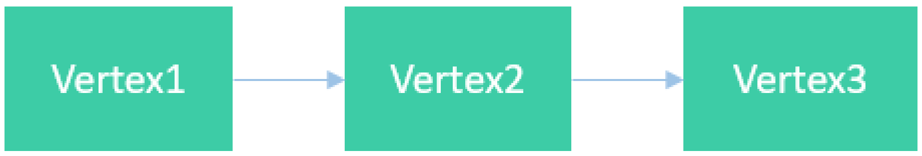
  In this scenario, the performance bottleneck is the sink and the optimization is specific to the data source. For example, for the JDBC data source, you can adjust the write batch using **connector.write.flush.max-rows** and JDBC rewriting parameter **rewriteBatchedStatements=true** to optimize the performance.

- The following figure shows a scenario where the back pressure of the last second operator is normal.

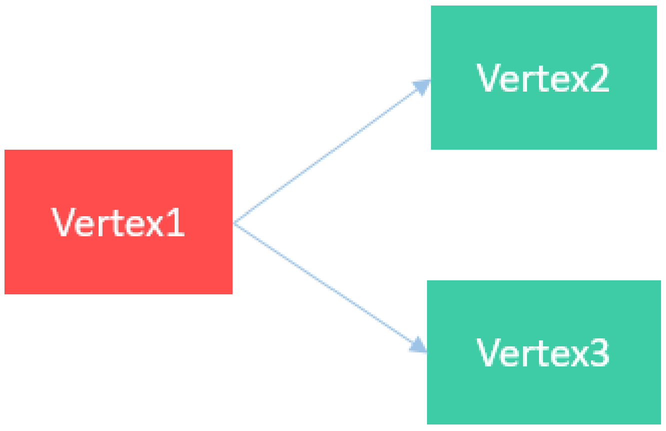
  In this scenario, the performance bottleneck is the Vertex2 operator. You can view the description about the function of the operator for further optimization.

- The back pressure of all operators is normal, but data is stacked.

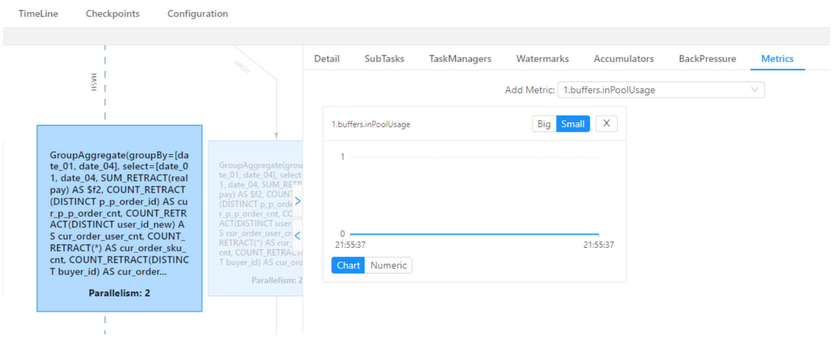
  In this scenario, the performance bottleneck is the source, and the performance is mainly affected by the data read speed. In this case, you can increase the number of Kafka partitions and the number of concurrent sources to solve the problem.

- The following figure shows that the back pressure of an operator is high, and its subsequent concurrent operators do not have back pressure.



In this scenario, the performance bottleneck is Vertex2 or Vertex3. To find out the specific bottleneck operator, enable inPoolUsage monitoring on the Flink UI page. If the inPoolUsage for operator concurrency is 100% for a long time, the corresponding operator is likely to be the performance bottleneck. In this case, you check the operator for further optimization.

Figure 6-19 inPoolUsage monitoring



Add Metric: 1.buffers.inPoolUsage

1.buffers.inPoolUsage

GroupAggregate(groupBy=[date_01, date_04], select=[date_01, date_04, SUM_RETRACT(realpay) AS $f2, COUNT_RETRACT(DISTINCT p_p_order_id) AS cur_p_p_order_cnt, COUNT_RETRACT(DISTINCT user_id_new) AS cur_order_user_cnt, COUNT_RETRACT(*) AS cur_order_sku_cnt, COUNT_RETRACT(DISTINCT buyer_id) AS cur_order...

Parallelism: 2

Performance Tuning

- Rocksdb state tuning

  Top N sorting, window aggregate calculation, and stream-stream join involve a large number of status operations. You can optimize the performance of state operations to improve the overall performance. You can try any of the following optimization methods:

  – Increase the state operation memory and reduce the disk I/O.

    ■ Increase the number of CU resources in a single slot.

    ■ Set optimization parameters:

      ○ taskmanager.memory.managed.fraction=xx

      ○ state.backend.rocksdb.block.cache-size=xx

      ○ state.backend.rocksdb.writebuffer.size=xx

  – Enable the micro-batch mode to avoid frequent state operations.

    Set the following parameters:

        - table.exec.mini-batch.enabled=true
        - table.exec.mini-batch.allow-latency=xx
        - table.exec.mini-batch.size=xx
    - Use ultra-high I/O local disks to accelerate disk operations.
- Group aggregation tuning

  The data skew problem is solved by Local-Global that divides a group aggregation into two stages: doing local aggregation in upstream first, and then global aggregation in downstream. To enable Local-global aggregation, set optimization parameter: **table.optimizer.aggphase-strategy=TWO_PHASE**
- Tuning count distinct
    - If the associated keys of count distinct are sparse, using Local-Globa cannot solve the problem of SPOF. In this case, you can configure the following parameters to optimize bucket splitting.
        - table.optimizer.distinct-agg.split.enabled=true
        - table.optimizer.distinct-agg.split.bucket-num=xx
    - Replace CASE WHEN with FILTER:

      For example:
      ```
      COUNT(DISTINCT CASE WHEN flag IN ('android', 'iphone')THEN user_id ELSE NULL END) AS
      app_uv
      ```
      Can be changed to:
      ```
      COUNT(DISTINCT user_id) FILTER(WHERE flag IN ('android', 'iphone')) AS app_uv
      ```
- Optimizing dimension table join

  The dimension table in joined with the key of each record in the left table. The matched in the cache is performed first. If no match is found, the remotely obtained data is used for matching. The optimization is as follows:
    - Increase the JVM memory and the number of cached records.
    - Set indexes for the dimension table to speed up query.

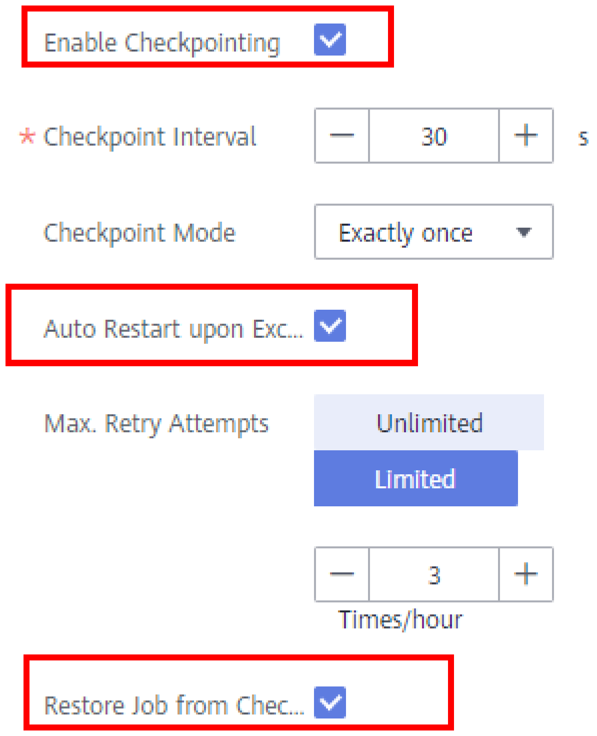
6.4.3 How Do I Prevent Data Loss After Flink Job Restart?

The DLI Flink checkpoint/savepoint mechanism is complete and reliable. You can use this mechanism to prevent data loss when a job is manually restarted or restarted due to an exception.

- To prevent data loss caused by job restart due to system faults, perform the following operations:
    - For Flink SQL jobs, select **Enable Checkpointing** and set a proper checkpoint interval that allows for the impact on service performance and the exception recovery duration. Select **Auto Restart upon Exception** and **Restore Job from Checkpoint**. After the configuration, if a job is restarted abnormally, the internal state and consumption position will be restored from the latest checkpoint file to ensure no data loss and accurate and consistent semantics of the internal state such as aggregation operators. In addition, to ensure that data is not duplicated, use a database or file system with a primary key as the data source.

Otherwise, add deduplication logic (data generated from the latest successful checkpoint to the time exception occurred will be repeatedly consumed) for downstream processes.

Figure 6-20 Flink job configuration parameters



Enable Checkpointing ✓

* Checkpoint Interval: 30 s

Checkpoint Mode: Exactly once

Auto Restart upon Exc... ✓

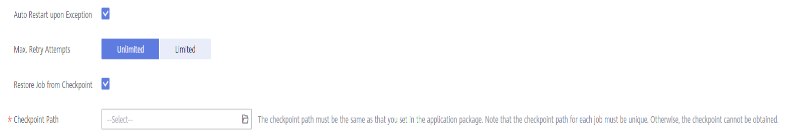
Max. Retry Attempts: Unlimited / [✓ Limited]

3 Times/hour

Restore Job from Chec... ✓

- For Flink Jar jobs, you need to enable checkpointing in the code. Additionally, if you have a custom state to save, you need to implement the ListCheckpointed interface and set a unique ID for each operator. In the job configuration, select **Restore Job from Checkpoint** and configure the checkpoint path.

Figure 6-21 Enable checkpointing



Auto Restart upon Exception ✓

Max. Retry Attempts: [✓ Unlimited] / Limited

Restore Job from Checkpoint ✓

* Checkpoint Path: --Select-- The checkpoint path must be the same as that you set in the application package. Note that the checkpoint path for each job must be unique. Otherwise, the checkpoint cannot be obtained.

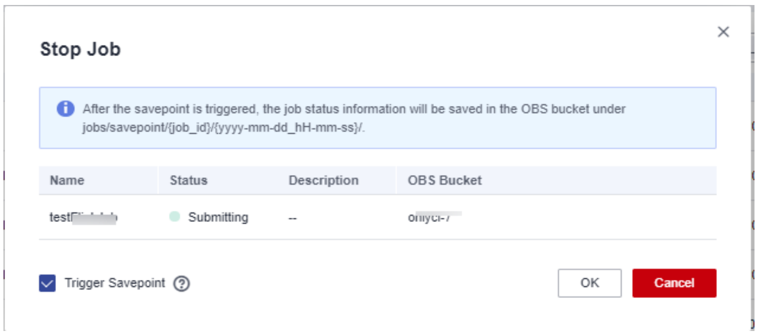
NOTE

Flink checkpointing ensures that the internal state data is accurate and consistent. However, for custom Source/Sink or stateful operators, you need to implement the ListCheckpointed API to ensure the reliability of service data.

- To prevent data loss after a job is manually restarted due to service modification, perform the following operations:
  - For jobs without internal states, you can set the start time or consumption position of the Kafka data source to a time before the job stops.

– For jobs with internal states, you can select **Trigger Savepoint** when stopping the job. Enable **Restore Savepoint** when you start the job again. The job will restore the consumption position and state from the selected savepoint file. The generation mechanism and format of Flink checkpoints are the same as those of savepoints. You can go to the Flink job list and choose **More** > **Import Savepoint** in the **Operation** column of a Flink job to import the latest checkpoint in OBS and restore the job from it.

Figure 6-22 Stopping a Job



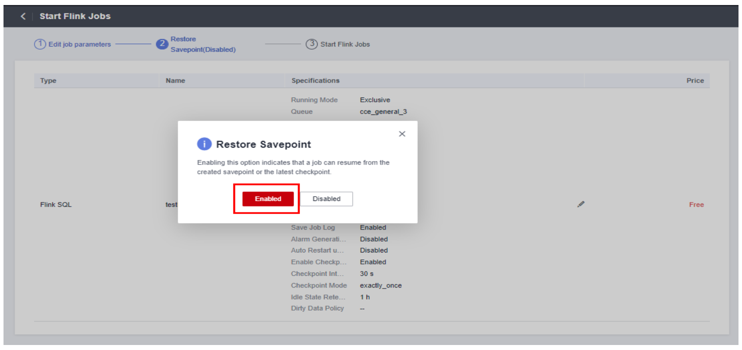
Stop Job

After the savepoint is triggered, the job status information will be saved in the OBS bucket under jobs/savepoint/{job_id}/{yyyy-mm-dd_hH-mm-ss}/.

Name	Status	Description	OBS Bucket
testF[redacted]	Submitting	--	onlyci-[redacted]

☑ Trigger Savepoint

Figure 6-23 Restoring the job from a savepoint or checkpoint



Type	Name	Specifications		Price
Flink SQL	test	Running Mode	Exclusive	Free
		Queue	cce_general_3	
		Save Job Log	Enabled	
		Alarm Generati...	Disabled	
		Auto Restart u...	Disabled	
		Enable Checkp...	Enabled	
		Checkpoint Int...	30 s	
		Checkpoint Mode	exactly_once	
		Idle State Rete...	1 h	
		Dirty Data Policy	--	

Restore Savepoint

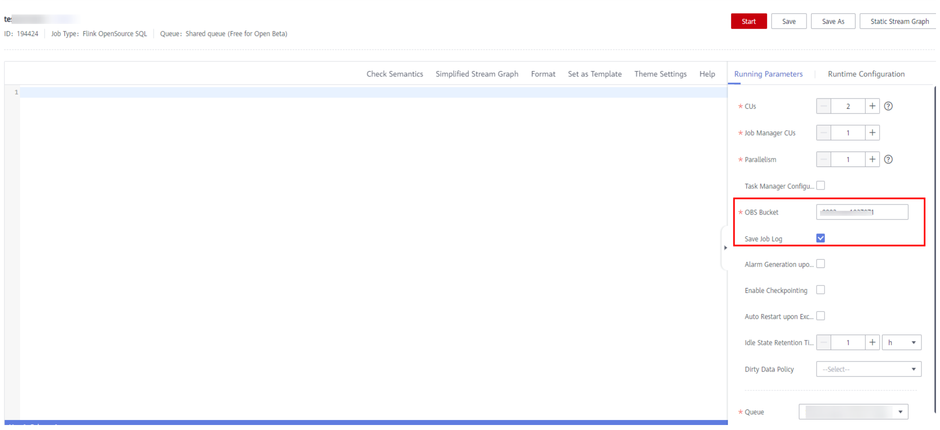
Enabling this option indicates that a job can resume from the created savepoint or the latest checkpoint.

[✓ Enabled] Disabled

6.4.4 How Do I Locate a Flink Job Running Error?

1. On the Flink job management, click **Edit** in the **Operation** column of the target job. On the displayed page, check whether **Save Job Log** in the **Running Parameters** tab is enabled.

Figure 6-24 Enabling **Save Job Logs**



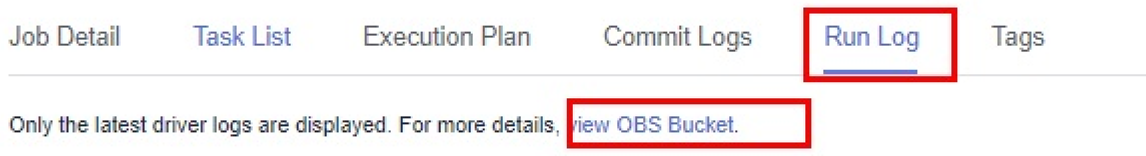
te[redacted]
ID: 194424 | Job Type: Flink OpenSource SQL | Queue: Shared queue (Free for Open Beta)

- CUs: 2
- Job Manager CUs: 1
- Parallelism: 1
- Task Manager Configu...: ☐
- OBS Bucket: [redacted]
- Save Job Log: ✓
- Alarm Generation upo...: ☐
- Enable Checkpointing: ☐
- Auto Restart upon Exc...: ☐
- Idle State Retention Ti...: 1 h
- Dirty Data Policy: --Select--
- Queue: [redacted]

- – If the function is enabled, go to 3.
- – If the function is disabled, running logs will not be dumped to an OBS bucket. In this case, perform 2 to save job logs.

2. On the job running page, select **Save Job Log** and specify an OBS bucket for storing the logs. Click **Start** to run the job again. After the executed is complete, perform 3 and subsequent steps.
3. In the Flink job list, click the job name. On the displayed job details page, click the **Run Log** tab.
4. Click **view OBS Bucket** to obtain the complete run logs of the job.

Figure 6-25 Viewing run logs



Only the latest driver logs are displayed. For more details, view OBS Bucket.

5. Download the latest **jobmanager.log** file, search for the keyword **RUNNING to FAILED**, and determine the failure cause based on the errors in the context.
6. If the information in the **jobmanager.log** file is insufficient for locating the fault, find the corresponding **taskmanager.log** file in the run logs and search for the keyword **RUNNING to FAILED** to confirm the failure cause.

6.4.5 How Can I Check if a Flink Job Can Be Restored From a Checkpoint After Restarting It?

What Is Restoration from a Checkpoint?

Flink's checkpointing is a fault tolerance and recovery mechanism. This mechanism ensures that real-time programs can self-recover in case of exceptions or machine issues during runtime.

Principles for Restoration from Checkpoints

- When a job fails to be executed or a resource restarts due to an exception that is not triggered by manual operations, data can be restored from a checkpoint.
- However, if the calculation logic of a job is modified, the job cannot be restored from a checkpoint.

Application Scenarios

Table 6-2 lists some common scenarios of restoring data from a checkpoint for your reference.

For more scenarios, refer to **Principles for Restoration from Checkpoints** and assess whether data can be restored from a checkpoint based on the actual situation.

Table 6-2 Common scenarios of restoring data from a checkpoint

Scenario	Restoration from a Checkpoint	Description
Adjust or increase the number of concurrent tasks.	Not supported	This operation alters the parallelism of the job, thereby changing its execution logic.
Modify Flink SQL statements and Flink Jar jobs.	Not supported	This operation modifies the algorithmic logic of the job with respect to resources. For example, if the original algorithm involves addition and subtraction, but the desired state requires multiplication, division, and modulo operations, it cannot be restored directly from the checkpoint.
Modify the static stream graph.	Not supported	This operation modifies the algorithmic logic of the job with respect to resources.
Modify the **CU(s) per TM** parameter.	Supported	The modification of compute resources does not affect the operational logic of the job's algorithm or operators.
A job runs abnormally or there is a physical power outage.	Supported	The job parameters and algorithm logic are not modified.

6.4.6 Why Are Logs Not Written to the OBS Bucket After a DLI Flink Job Fails to Be Submitted for Running?

Mode for storing generated job logs when a DLI Flink job fails to be submitted or executed. The options are as follows:

- If the submission fails, a submission log is generated only in the **submit-client** directory.
- You can view the logs generated within 1 minute when the job fails to be executed on the management console.

  Choose **Job Management** > **Flink Jobs**, click the target job name to go to the job details page, and click **Run Log** to view real-time logs.
- If the running fails and exceeds 1 minute (the log dump period is 1 minute), run logs are generated in the **application_***xx* directory.

Flink dependencies have been built in the DLI server and security hardening has been performed based on the open-source community version. To avoid dependency package compatibility issues or log output and dump issues, be careful to exclude the following files when packaging:

- Built-in dependencies (or set the package dependency scope to "provided")
- Log configuration files (example, log4j.properties/logback.xml)
- JAR file for log output implementation (example, log4j).

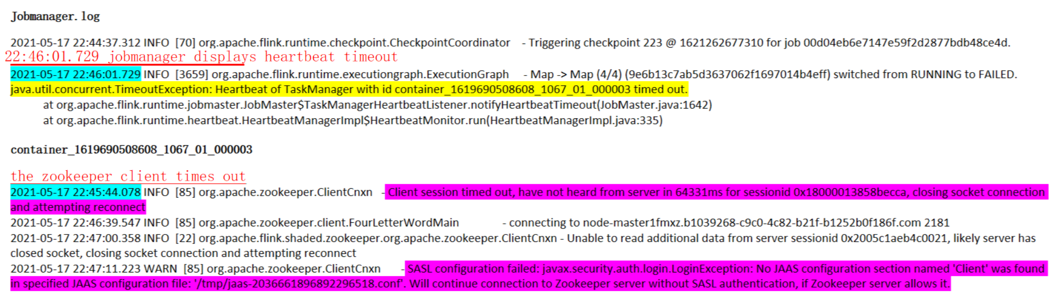
On this basis, the **taskmanager.log** file rolls as the log file size and time change.

6.4.7 Why Is the Flink Job Abnormal Due to Heartbeat Timeout Between JobManager and TaskManager?

Symptom

JobManager and TaskManager heartbeats timed out. As a result, the Flink job is abnormal.

Figure 6-26 Error information

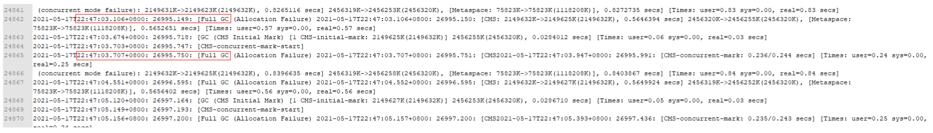


```
Jobmanager. log

2021-05-17 22:44:37.312 INFO  [70] org.apache.flink.runtime.checkpoint.CheckpointCoordinator   - Triggering checkpoint 223 @ 1621262677310 for job 00d04eb6e7147e59f2d2877bdb48ce4d.
22:46:01.729 jobmanager displays heartbeat timeout
2021-05-17 22:46:01.729 INFO  [3659] org.apache.flink.runtime.executiongraph.ExecutionGraph    - Map -> Map (4/4) (9e6b13c7ab5d3637062f1697014b4eff) switched from RUNNING to FAILED.
java.util.concurrent.TimeoutException: Heartbeat of TaskManager with id container_1619690508608_1067_01_000003 timed out.
        at org.apache.flink.runtime.jobmaster.JobMaster$TaskManagerHeartbeatListener.notifyHeartbeatTimeout(JobMaster.java:1642)
        at org.apache.flink.runtime.heartbeat.HeartbeatManagerImpl$HeartbeatMonitor.run(HeartbeatManagerImpl.java:335)

container_1619690508608_1067_01_000003

the zookeeper client times out
2021-05-17 22:45:44.078 INFO  [85] org.apache.zookeeper.ClientCnxn   - Client session timed out, have not heard from server in 64331ms for sessionid 0x1800013858becca, closing socket connection
and attempting reconnect
2021-05-17 22:46:39.547 INFO  [85] org.apache.zookeeper.client.FourLetterWordMain          - connecting to node-master1fmxz.b1039268-c9c0-4c82-b21f-b1252b0f186f.com 2181
2021-05-17 22:47:00.358 INFO  [22] org.apache.flink.shaded.zookeeper.org.apache.zookeeper.ClientCnxn - Unable to read additional data from server sessionid 0x2005c1aeb4c0021, likely server has
closed socket, closing socket connection and attempting reconnect
2021-05-17 22:47:11.223 WARN  [85] org.apache.zookeeper.ClientCnxn       - SASL configuration failed: javax.security.auth.login.LoginException: No JAAS configuration section named 'Client' was found
in specified JAAS configuration file: '/tmp/jaas-2036661896892296518.conf'. Will continue connection to Zookeeper server without SASL authentication, if Zookeeper server allows it.
```

Possible Causes

1. Check whether the network is intermittently disconnected and whether the cluster load is high.
2. If Full GC occurs frequently, check the code to determine whether memory leakage occurs.

Figure 6-27 Full GC



```
 (concurrent mode failure): 2149631K->2149623K(2149632K), 0.8265116 secs] 2456319K->2456253K(2456320K), [Metaspace: 75823K->75823K(1118208K)], 0.8272735 secs] [Times: user=0.83 sys=0.00, real=0.83 secs]
2021-05-17T22:47:03.106+0800: 26995.149: [Full GC (Allocation Failure) 2021-05-17T22:47:03.106+0800: 26995.150: [CMS: 2149632K->2149625K(2149632K), 0.5646394 secs] 2456320K->2456255K(2456320K), [Metaspace:
75823K->75823K(1118208K)], 0.5652651 secs] [Times: user=0.57 sys=0.00, real=0.57 secs]
2021-05-17T22:47:03.674+0800: 26995.718: [GC (CMS Initial Mark) [1 CMS-initial-mark: 2149625K(2149632K)] 2456255K(2456320K), 0.0284012 secs] [Times: user=0.06 sys=0.00, real=0.03 secs]
2021-05-17T22:47:03.703+0800: 26995.747: [CMS-concurrent-mark-start]
2021-05-17T22:47:03.707+0800: 26995.750: [Full GC (Allocation Failure) 2021-05-17T22:47:03.707+0800: 26995.751: [CMS2021-05-17T22:47:03.947+0800: 26995.991: [CMS-concurrent-mark: 0.236/0.244 secs] [Times: user=0.24 sys=0.00,
real=0.25 secs]
 (concurrent mode failure): 2149632K->2149625K(2149632K), 0.8396635 secs] 2456319K->2456258K(2456320K), [Metaspace: 75823K->75823K(1118208K)], 0.8403867 secs] [Times: user=0.84 sys=0.00, real=0.84 secs]
2021-05-17T22:47:04.551+0800: 26996.595: [Full GC (Allocation Failure) 2021-05-17T22:47:04.552+0800: 26996.595: [CMS: 2149632K->2149627K(2149632K), 0.5649924 secs] 2456319K->2456252K(2456320K), [Metaspace:
75823K->75823K(1118208K)], 0.5656402 secs] [Times: user=0.56 sys=0.00, real=0.56 secs]
2021-05-17T22:47:05.120+0800: 26997.164: [GC (CMS Initial Mark) [1 CMS-initial-mark: 2149627K(2149632K)] 2456253K(2456320K), 0.0286710 secs] [Times: user=0.05 sys=0.00, real=0.03 secs]
2021-05-17T22:47:05.149+0800: 26997.193: [CMS-concurrent-mark-start]
2021-05-17T22:47:05.156+0800: 26997.200: [Full GC (Allocation Failure) 2021-05-17T22:47:05.157+0800: 26997.200: [CMS2021-05-17T22:47:05.393+0800: 26997.436: [CMS-concurrent-mark: 0.235/0.243 secs] [Times: user=0.25 sys=0.00,
```

Handling Procedure

- If Full GC occurs frequently, check the code to determine whether memory leakage occurs.
- Allocate more resources for a single TaskManager.
- Contact technical support to modify the cluster heartbeat configuration.

7 Spark Jobs

7.1 Spark Job Development

7.1.1 Spark Jobs

Does DLI Spark Support Scheduled Periodic Jobs?

DLI Spark does not support job scheduling. You can use other services, such as DataArts Studio, or use APIs or SDKs to customize job schedule.

Can I Define the Primary Key When I Create a Table with a Spark SQL Statement?

The Spark SQL syntax does not support primary key definition.

Can DLI Spark Jar Jobs Access GaussDB(DWS) Datasource Tables?

Yes.

For details, see **Connecting to GaussDB(DWS)** and **Accessing SQL Database Tables**.

How Do I Check the Version of the Spark Built-in Dependency Package?

DLI built-in dependencies are provided by the platform by default. In case of conflicts, you do not need to upload them when packing JAR files of Spark or Flink Jar jobs.

For details about how to check the version, see **Built-in Dependency Package**.

Can I Download Packages on the Package Management Page?

No, the packages cannot be downloaded.

How Do I Use an API to Access DLI Through a Public Network?

To access DLI from the public network, use the domain name **dli.*{regionid}*.myhuaweicloud.com**.

- For details about DLI endpoints, see **Endpoints**.
- For details about DLI APIs, see **Data Lake Insight API Reference**.

Which Path Should the Third-Party Dependency Jar File Be Stored for the Customized Spark 3.1.1 Image?

Store the third-party dependency in the **/opt/spark/jars** directory.

7.1.2 How Do I Use Spark to Write Data into a DLI Table?

To use Spark to write data into a DLI table, configure the following parameters:

- fs.obs.access.key
- fs.obs.secret.key
- fs.obs.impl
- fs.obs.endpoint

The following is an example:

```
import logging
from operator import add
from pyspark import SparkContext

logging.basicConfig(format='%(message)s', level=logging.INFO)

#import local file
test_file_name = "D://test-data_1.txt"
out_file_name = "D://test-data_result_1"

sc = SparkContext("local","wordcount app")
sc._jsc.hadoopConfiguration().set("fs.obs.access.key", "myak")
sc._jsc.hadoopConfiguration().set("fs.obs.secret.key", "mysk")
sc._jsc.hadoopConfiguration().set("fs.obs.impl", "org.apache.hadoop.fs.obs.OBSFileSystem")
sc._jsc.hadoopConfiguration().set("fs.obs.endpoint", "myendpoint")

# red: text_file rdd object
text_file = sc.textFile(test_file_name)

# counts
counts = text_file.flatMap(lambda line: line.split(" ")).map(lambda word: (word, 1)).reduceByKey(lambda a,
b: a + b)
# write
counts.saveAsTextFile(out_file_name)
```

7.1.3 How Do I Set Up AK/SK So That a General Queue Can Access Tables Stored in OBS?

(Recommended) Solution 1: Using a Temporary AK/SK

The temporary AK/SK is recommended. For details, see **Obtaining a Temporary Access Key and Security Token** in *Identity and Access Management API Reference*.

NOTE

Hard-coded or plaintext AK and SK pose significant security risks. To ensure security, encrypt your AK and SK, store them in configuration files or environment variables, and decrypt them when needed.

Table 7-1 Obtaining access credentials

Type	Instruction	Description
Flink job	**Flink OpenSource SQL Jobs Using DEW to Manage Access Credentials**	Guideline for using DEW to manage and access credentials for Flink OpenSource SQL jobs. When writing the output data of Flink jobs to MySQL or GaussDB(DWS), set attributes such as the username and password in the connector.
	Flink Jar Jobs Using DEW to Acquire Access Credentials for Reading and Writing Data from and to OBS	Guideline for Flink Jar jobs to acquire an AK/SK to read and write data from and to OBS.
	Obtaining Temporary Credentials for Flink Job Agencies	DLI provides a common interface to obtain temporary credentials for Flink job agencies set by users during job launch. The interface encapsulates the obtained temporary credentials for the job agency in the **com.huaweicloud.sdk.core.auth.BasicCredentials** class. Guideline for obtaining a temporary credential for a Flink job agency.
Spark job	**Spark Jar Jobs Using DEW to Acquire Access Credentials for Reading and Writing Data from and to OBS**	Guideline for Spark Jar jobs to acquire an AK/SK to read and write data from and to OBS.
	Obtaining Temporary Credentials for Spark Job Agencies	Guideline for obtaining a temporary credential for a Spark Jar job agency.

Solution 2: Setting Up a Spark Jar Job to Obtain the AK/SK

- **To obtain the AK/SK, set the parameters as follows:**
  - Create a SparkContext using code.

```
val sc: SparkContext = new SparkContext()
sc.hadoopConfiguration.set("fs.obs.access.key", ak)
sc.hadoopConfiguration.set("fs.obs.secret.key", sk)
```

   – Create a SparkSession using code.

```
val sparkSession: SparkSession = SparkSession
      .builder()
      .config("spark.hadoop.fs.obs.access.key", ak)
      .config("spark.hadoop.fs.obs.secret.key", sk)
      .enableHiveSupport()
      .getOrCreate()
```

- **To obtain the AK/SK and security token and use them together for authentication, set the parameters as follows:**
   – Create a SparkContext using code.

```
val sc: SparkContext = new SparkContext()
sc.hadoopConfiguration.set("fs.obs.access.key", ak)
sc.hadoopConfiguration.set("fs.obs.secret.key", sk)
sc.hadoopConfiguration.set("fs.obs.session.token", sts)
```

   – Create a SparkSession using code.

```
val sparkSession: SparkSession = SparkSession
      .builder()
      .config("spark.hadoop.fs.obs.access.key", ak)
      .config("spark.hadoop.fs.obs.secret.key", sk)
      .config("spark.hadoop.fs.obs.session.token", sts)
      .enableHiveSupport()
      .getOrCreate()
```

7.1.4 How Do I View the Resource Usage of DLI Spark Jobs?

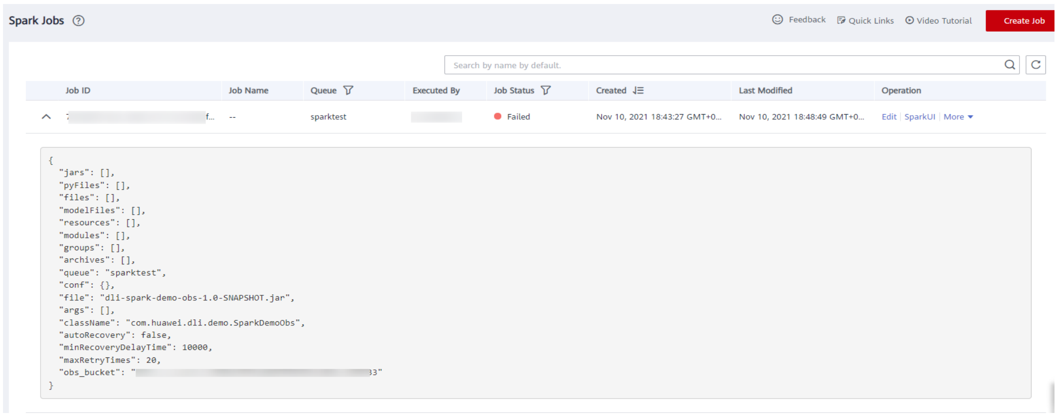
Viewing the Configuration of a Spark Job

Log in to the DLI console. In the navigation pane, choose **Job Management** > **Spark Jobs**. In the job list, locate the target job and click ⌄ next to Job ID to view the parameters of the job.

NOTE

The content is displayed only when the parameters in **Advanced Settings** are configured during Spark job creation. For details about how to create a Spark job, see **Creating a Spark Job**.

Figure 7-1 Viewing the configuration of a Spark job



Job ID	Job Name	Queue	Executed By	Job Status	Created	Last Modified	Operation
7[redacted]f...	--	sparktest	[redacted]	Failed	Nov 10, 2021 18:43:27 GMT+0...	Nov 10, 2021 18:48:49 GMT+0...	Edit | SparkUI | More

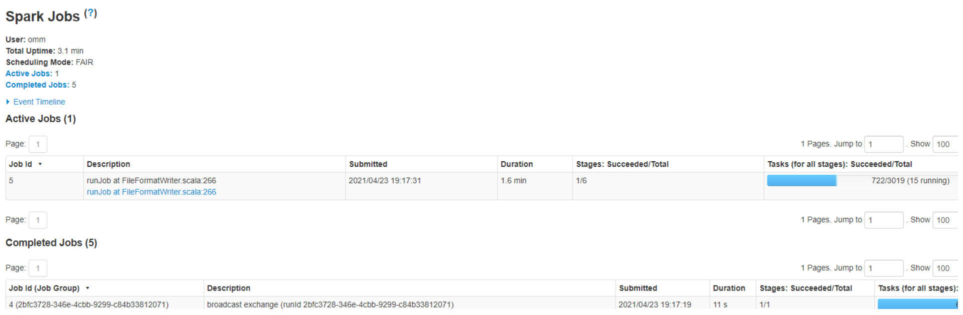
```
{
  "jars": [],
  "pyFiles": [],
  "files": [],
  "modelFiles": [],
  "resources": [],
  "modules": [],
  "groups": [],
  "archives": [],
  "queue": "sparktest",
  "conf": {},
  "file": "dli-spark-demo-obs-1.0-SNAPSHOT.jar",
  "args": [],
  "className": "com.huawei.dli.demo.SparkDemoObs",
  "autoRecovery": false,
  "minRecoveryDelayTime": 10000,
  "maxRetryTimes": 20,
  "obs_bucket": "[redacted]33"
}
```

Viewing Real-Time Resource Usage of a Spark Job

Perform the following operations to view the number of running CUs occupied by a Spark job in real time:

1. Log in to the DLI console. In the navigation pane, choose **Job Management** > **Spark Jobs**. In the job list, locate the target job and click **SparkUI** in the **Operation** column.
2. On the Spark UI page, view the real-time running resources of the Spark job.

Figure 7-2 SparkUI



Spark Jobs (?)

User: omm
Total Uptime: 3.1 min
Scheduling Mode: FAIR
Active Jobs: 1
Completed Jobs: 5

Active Jobs (1)

Page: 1
1 Pages. Jump to 1 . Show 100

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
5	runJob at FileFormatWriter.scala:266 runJob at FileFormatWriter.scala:266	2021/04/23 19:17:31	1.6 min	1/6	722/3019 (15 running)

Page: 1
1 Pages. Jump to 1 . Show 100

Completed Jobs (5)

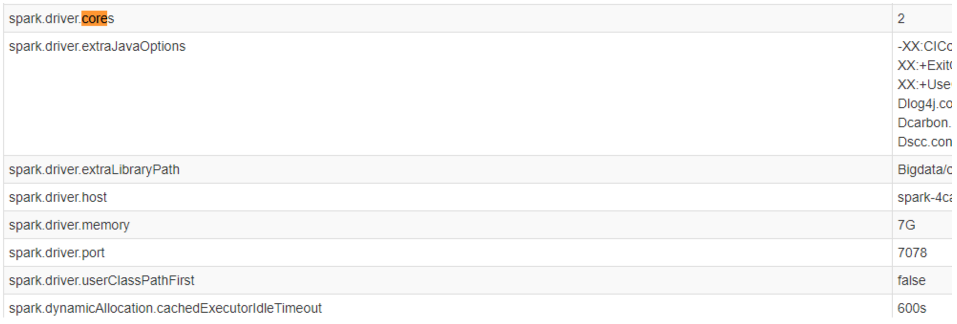
Page: 1
1 Pages. Jump to 1 . Show 100

Job Id (Job Group)	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages):
4 (2bfc3728-346e-4cbb-9299-c84b33812071)	broadcast exchange (runId 2bfc3728-346e-4cbb-9299-c84b33812071)	2021/04/23 19:17:19	11 s	1/1	

3. On the Spark UI page, view the original configuration of the Spark job (available only to new clusters).

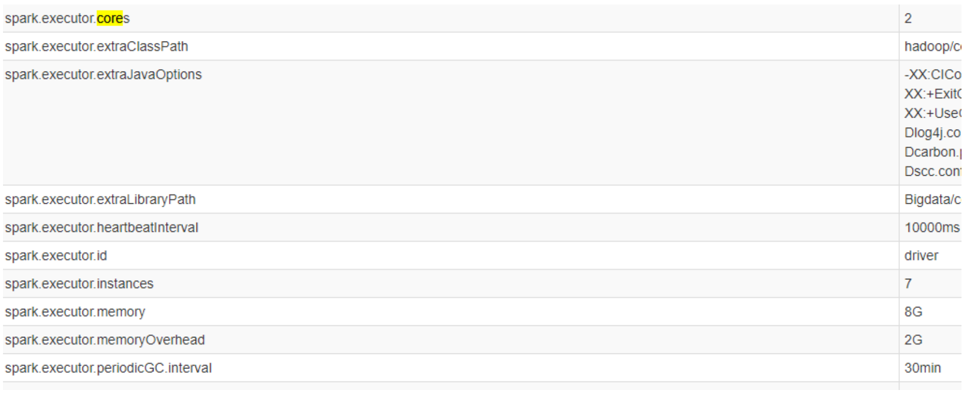
   On the Spark UI page, click **Environment** to view **Driver** and **Executor** information.

Figure 7-3 Driver information



spark.driver.cores	2
spark.driver.extraJavaOptions	-XX:CICc XX:+Exit XX:+Use Dlog4j.co Dcarbon. Dscc.con
spark.driver.extraLibraryPath	Bigdata/c
spark.driver.host	spark-4ca
spark.driver.memory	7G
spark.driver.port	7078
spark.driver.userClassPathFirst	false
spark.dynamicAllocation.cachedExecutorIdleTimeout	600s

Figure 7-4 Executor information



spark.executor.cores	2
spark.executor.extraClassPath	hadoop/c
spark.executor.extraJavaOptions	-XX:CICo XX:+Exit XX:+Use Dlog4j.co Dcarbon. Dscc.con
spark.executor.extraLibraryPath	Bigdata/c
spark.executor.heartbeatInterval	10000ms
spark.executor.id	driver
spark.executor.instances	7
spark.executor.memory	8G
spark.executor.memoryOverhead	2G
spark.executor.periodicGC.interval	30min

7.1.5 How Do I Use Python Scripts to Access the MySQL Database If the pymysql Module Is Missing from the Spark Job Results Stored in MySQL?

1. If the pymysql module is missing, check whether the corresponding EGG package exists. If the package does not exist, upload the pyFile package on the **Package Management** page. The procedure is as follows:
   a. Upload the **egg** package to the specified OBS path.
   b. Log in to the DLI management console and choose **Data Management** > **Package Management**.
   c. On the **Package Management** page, click **Create Package** in the upper right corner to create a package.
   d. In the **Create Package** dialog, set the following parameters:
      - Type: Select **PyFile**.
      - **OBS Path**: Select the OBS path where the **egg** package is stored.
      - Set **Group** and **Group Name** as you need.
   e. Click **OK**.
   f. On the Spark job editing page where the error is reported, choose the uploaded **egg** package from the **Python File Dependencies** drop-down list and run the Spark job again.
2. To interconnect PySpark jobs with MySQL, you need to create a datasource connection to enable the network between DLI and RDS.

   For how to create a datasource connection on the management console, see **Data Lake Insight User Guide**.

   For how to call an API to create a datasource connection, see **Data Lake Insight API Reference**.

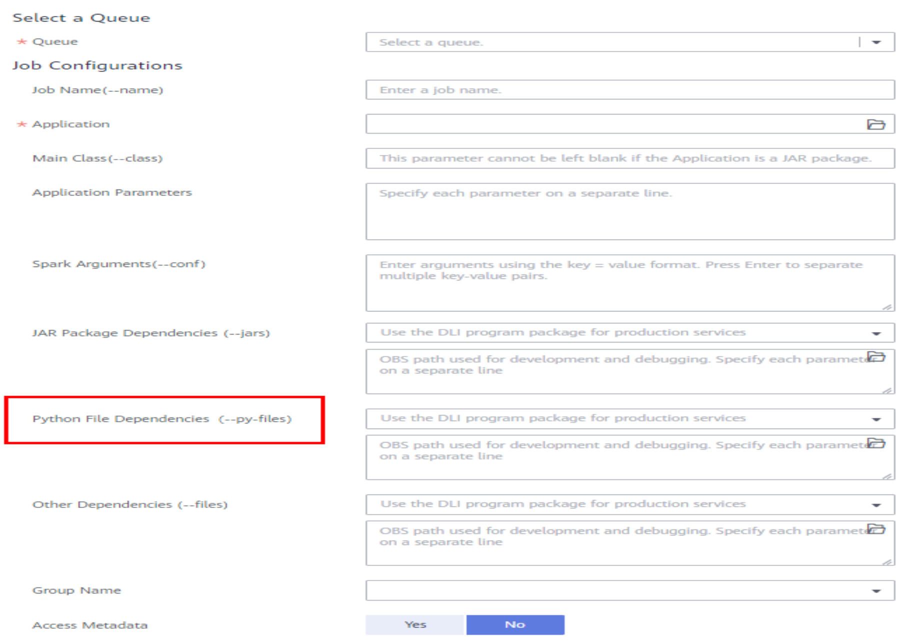
7.1.6 How Do I Run a Complex PySpark Program in DLI?

DLI natively supports PySpark.

For most cases, Python is preferred for data analysis, and PySpark is the best choice for big data analysis. Generally, JVM programs are packed into JAR files and depend on third-party JAR files. Similarly, Python programs also depend on third-party libraries, especially big data analysis programs related to PySpark-based converged machine learning. Traditionally, Python libraries are installed directly on the execution machine using pip. However, for serverless services like DLI, users do not need to and are not aware of the underlying compute resources. How can we ensure that users can run their programs more effectively?

DLI has built-in algorithm libraries for machine learning in its compute resources. These common algorithm libraries meet the requirements of most users. What if a user's PySpark program depends on a program library that is not provided by the built-in algorithm library? Actually, the dependency of PySpark is specified based on PyFiles. On the DLI Spark job page, you can directly select the Python third-party program library (such as ZIP and EGG) stored on OBS.

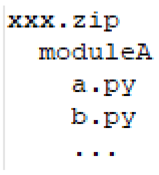
Figure 7-5 Spark job editor page



Select a Queue	
* Queue	Select a queue.
Job Configurations	
Job Name(--name)	Enter a job name.
* Application	
Main Class(--class)	This parameter cannot be left blank if the Application is a JAR package.
Application Parameters	Specify each parameter on a separate line.
Spark Arguments(--conf)	Enter arguments using the key = value format. Press Enter to separate multiple key-value pairs.
JAR Package Dependencies (--jars)	Use the DLI program package for production services OBS path used for development and debugging. Specify each parameter on a separate line
Python File Dependencies (--py-files)	Use the DLI program package for production services OBS path used for development and debugging. Specify each parameter on a separate line
Other Dependencies (--files)	Use the DLI program package for production services OBS path used for development and debugging. Specify each parameter on a separate line
Group Name	
Access Metadata	Yes [✓ No]

The compressed package of the dependent third-party Python library has structure requirements. For example, if the PySpark program depends on moduleA (import moduleA), the compressed package must meet the following structure requirement:

Figure 7-6 Compressed package structure requirement



```
xxx.zip
  moduleA
    a.py
    b.py
    ...
```

That is, the compressed package contains a folder named after a module name, and then the Python file of the corresponding class. Generally, the downloaded Python library may not meet this requirement. Therefore, you need to compress the Python library again. In addition, there is no requirement on the name of the compressed package. Therefore, it is recommended that you compress the packages of multiple modules into a compressed package. Now, a large and complex PySpark program is configured and runs normally.

7.1.7 How Do I Use JDBC to Set the spark.sql.shuffle.partitions Parameter to Improve the Task Concurrency?

Scenario

When shuffle statements, such as GROUP BY and JOIN, are executed in Spark jobs, data skew occurs, which slows down the job execution.

To solve this problem, you can configure **spark.sql.shuffle.partitions** to improve the concurrency of shuffle read tasks.

Configuring spark.sql.shuffle.partitions

You can use the **set** clause to configure the **dli.sql.shuffle.partitions** parameter in JDBC. The statement is as follows:

```
Statement st = conn.stamte()
st.execute("set spark.sql.shuffle.partitions=20")
```

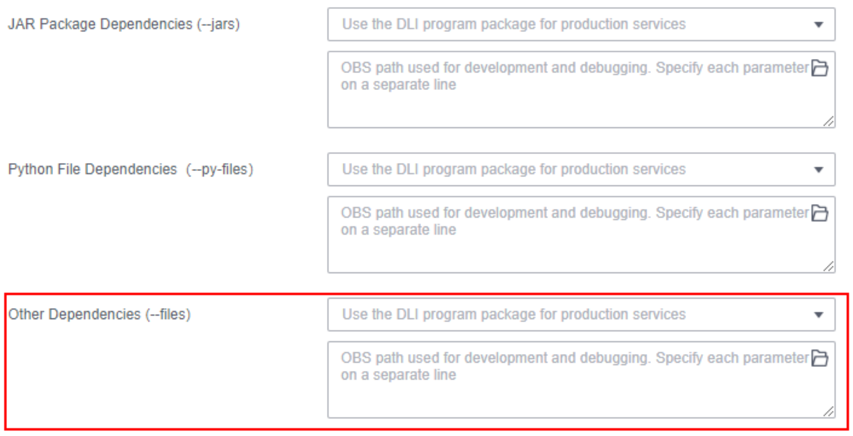
7.1.8 How Do I Read Uploaded Files for a Spark Jar Job?

You can use SparkFiles to read the file submitted using **--file** form a local path: **SparkFiles.get(** *"Name of the uploaded file"* **)**.

NOTE

- The file path in the Driver is different from that obtained by the Executor. The path obtained by the Driver cannot be passed to the Executor.
- You still need to call **SparkFiles.get(** *"filename"* **)** in Executor to obtain the file path.
- The **SparkFiles.get()** method can be called only after Spark is initialized.

Figure 7-7 Adding other dependencies


JAR Package Dependencies (--jars)
Use the DLI program package for production services
OBS path used for development and debugging. Specify each parameter on a separate line
Python File Dependencies (--py-files)
Use the DLI program package for production services
OBS path used for development and debugging. Specify each parameter on a separate line
Other Dependencies (--files)
Use the DLI program package for production services
OBS path used for development and debugging. Specify each parameter on a separate line

The java code is as follows:

```
package main.java

import org.apache.spark.SparkFiles
import org.apache.spark.sql.SparkSession

import scala.io.Source

object DliTest {
  def main(args:Array[String]): Unit = {
    val spark = SparkSession.builder
      .appName("SparkTest")
      .getOrCreate()

    // Driver: obtains the uploaded file.
    println(SparkFiles.get("test"))

    spark.sparkContext.parallelize(Array(1,2,3,4))
         // Executor: obtains the uploaded file.
      .map(_ => println(SparkFiles.get("test")))
      .map(_ => println(Source.fromFile(SparkFiles.get("test")).mkString)).collect()
  }
}
```

7.1.9 Why Can't I Find the Specified Python Environment After Adding the Python Package?

I cannot find the specified Python environment after adding the Python 3 package.

Set **spark.yarn.appMasterEnv.PYSPARK_PYTHON** to **python3** in the **conf** file to specify the Python 3 environment for the compute cluster.

New clusters use the Python 3 environment by default.

7.1.10 Why Is a Spark Jar Job Stuck in the Submitting State?

The remaining CUs in the queue may be insufficient. As a result, the job cannot be submitted.

To view the remaining CUs of a queue, perform the following steps:

1. Check the CU usage of the queue.

   Log in to the Cloud Eye console. In the navigation pane on the left, choose **Cloud Service Monitoring** > **Data Lake Insight**. On the displayed page, locate the desired queue and click **View Metric** in the **Operation** column, and check **CU Usage (queue)** on the displayed page.

2. Calculate the number of remaining CUs.

   Remaining CUs of a queue = Total CUs of the queue – CU usage.

If the number of remaining CUs is less than the number of CUs required by the job, the job submission fails. The submission can be successful only after resources are available.

7.2 Spark Job O&M

7.2.1 What Can I Do When Receiving java.lang.AbstractMethodError in the Spark Job?

The Spark 2.3 has changed the behavior of the internal interface **Logging**. If the user code directly inherits the **Logging** and the earlier version Spark is used during compilation, the **java.lang.AbstractMethodError** is reported when the application runs in the Spark 2.3 environment.

Solutions are as follows:

- You can recompile the application based on Spark 2.3.
- You can use the **sl4j+log4j** to implement the log function instead of inheriting the internal interface **Logging** of the Spark. Details are described as follows:

```
<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>slf4j-api</artifactId>
  <version>1.7.16</version>
</dependency>
<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>slf4j-log4j12</artifactId>
  <version>1.7.16</version>
</dependency>
```

```
<dependency>
  <groupId>log4j</groupId>
  <artifactId>log4j</artifactId>
  <version>1.2.17</version>
</dependency>

private val logger = LoggerFactory.getLogger(this.getClass)
logger.info("print log with sl4j+log4j")
```

7.2.2 Why Do I Get "ResponseCode: 403" and "ResponseStatus: Forbidden" Errors When a Spark Job Accesses OBS Data?

Symptom

The following error is reported when a Spark job accesses OBS data:

```
Caused by: com.obs.services.exception.ObsException: Error message:Request Error.OBS servcie Error
Message. -- ResponseCode: 403, ResponseStatus: Forbidden
```

Solution

Set the AK/SK to enable Spark jobs to access OBS data.

For details, see **How Do I Set Up AK/SK So That a General Queue Can Access Tables Stored in OBS?**

7.2.3 Why Do I Encounter the Error "verifyBucketExists on XXXX: status [403]" When Using a Spark Job to Access an OBS Bucket That I Have Permission to Access?

This error message may be due to the OBS bucket being set as the DLI log bucket, which cannot be used for other purposes.

You can follow these steps to check:

1. Check if the OBS bucket has been set as the DLI log bucket.

   In the left navigation pane of the DLI management console, choose **Global Configuration** > **Job Configurations**. On the displayed page, check if the OBS bucket has been set as the DLI log bucket.

2. Check if the bucket is used for other purposes.

   If so, you can modify the job configuration on the DLI management console and select another OBS bucket that is not being used for DLI log storage.

7.2.4 Why Does a Job Running Timeout Occur When Processing a Large Amount of Data with a Spark Job?

When running large amounts of data in a Spark job, if a timeout exception error occurs, it is usually due to insufficient resource configuration, data skew, network issues, or too many tasks.

Solution:

- Set concurrency: By setting an appropriate concurrency, you can run multiple tasks concurrently, improving the job processing capacity.

  For example, when accessing large amounts of database data in GaussDB(DWS), set the concurrency and run in a multi-task manner to avoid job timeouts.

  For details about how to set the concurrency, see the **partitionColumn** and **numPartitions** fields and **Scala Example Code** for connecting to GaussDB(DWS).
- Adjust the number of Executors in the Spark job and allocate more resources for the Spark job to run.

7.2.5 Why Does a Spark Job Fail to Execute with an Abnormal Access Directory Error When Accessing Files in SFTP?

Spark jobs cannot access SFTP. Upload the files you want to access to OBS and then you can analyze the data using Spark jobs.

1. Upload data to an OBS bucket: Upload data stored in SFTP to an OBS bucket using the OBS management console or command-line tools.

   For how to use a Spark job to read OBS data, see **Using Spark Jar Jobs to Read and Query OBS Data**.
2. Configure the Spark job: Configure the Spark job to access data stored in OBS.
3. Submit the Spark job: After completing the job writing, submit and execute the job.

7.2.6 Why Does the Job Fail to Be Executed Due to Insufficient Database and Table Permissions?

Symptom

When a Spark job is running, an error message is displayed, indicating that the user does not have the database permission. The error information is as follows:

```
org.apache.spark.sql.AnalysisException: org.apache.hadoop.hive.ql.metadata.HiveException: 
MetaException(message:Permission denied for resource: databases.xxx,action:SPARK_APP_ACCESS_META)
```

Solution

You need to assign the database permission to the user who executes the job. The procedure is as follows:

1. In the navigation pane on the left of the management console, choose **Data Management** > **Databases and Tables**.
2. Locate the row where the target database resides and click **Permissions** in the **Operation** column.
3. On the displayed page, click **Grant Permission** in the upper right corner.
4. In the displayed dialog box, select **User** or **Project**, enter the username or select the project that needs the permission, and select the desired permissions.
5. Click **OK**.

7.2.7 Why Is the global_temp Database Missing in the Job Log of Spark 3.x?

Symptom

I cannot find the global_temp database in the Spark 3.*x* job log.

Possible Causes

The global_temp database is the default built-in database of Spark 3.*x* and is Spark's global temporary view.

When a Spark job is registered with ViewManager, the system checks whether the database exists in Metastore. If it is, the Spark job fails to be executed.

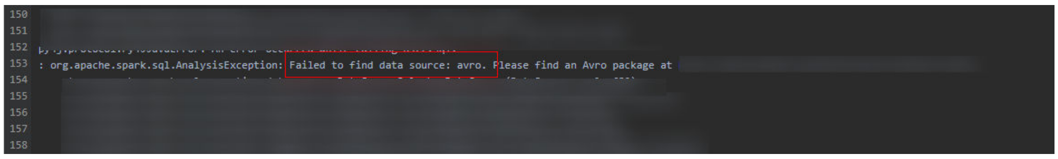
If the Spark 3.*x* job log contains a record of accessing catalogs to query the database and a message indicating that the non-existence of the database is to ensure that Spark jobs can run properly, you do not need to perform any operations.

7.2.8 Why Does Using DataSource Syntax to Create an OBS Table of Avro Type Fail When Accessing Metadata With Spark 2.3.x?

Symptom

I failed to use the DataSource syntax to create an OBS table in Avro format when selecting Spark to access metadata.

Figure 7-8 Failed to create an OBS table in Avro format



```
[redacted]
: org.apache.spark.sql.AnalysisException: Failed to find data source: avro. Please find an Avro package at [redacted]
```

Possible Causes

Spark 2.3.*x* does not support creating OBS tables in Avro format.

Solution

When using the DataSource syntax to create an OBS table in Avro format, select Spark 2.4.*x* or later.

8 DLI Resource Quotas

8.1 What Is User Quota?

Quotas are the preset limits on resource usage established by a cloud platform, encompassing both the quantity and capacity of resources. These quota settings are designed to ensure equitable distribution and utilization of resources, as well as to prevent over-concentration and waste.

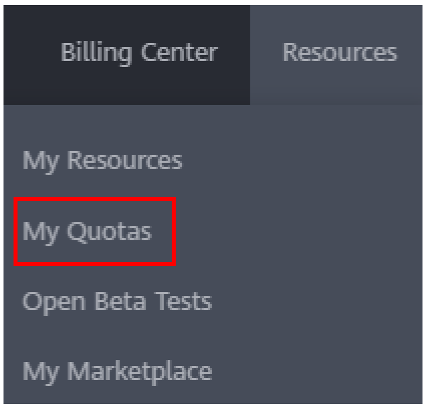
If the existing resource quota cannot meet your service needs, you can submit a service ticket to increase your quota.

Once approved, we will update your resource quota accordingly and send you a notification. For details about the quotas, see **Quotas**.

8.2 How Do I View My Quotas?

1. Log in to the management console.
2. Click in the upper left corner and select a region and a project.
3. In the upper right corner of the page, choose **Resources** > **My Quotas**.
   The **Service Quota** page is displayed.

Figure 8-1 My quotas



4. View the used and total quota of each type of resources on the displayed page.

   If a quota cannot meet service requirements, increase a quota.

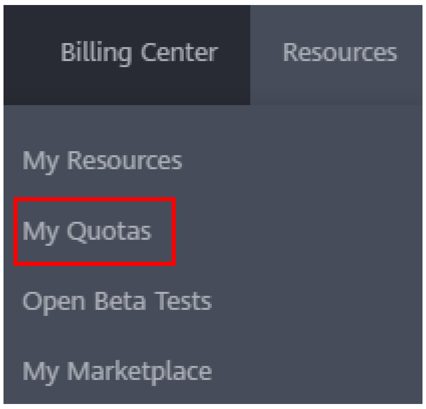
8.3 How Do I Apply for a Higher Quota?

How Do I Apply for a Higher Quota?

1. Log in to the management console.
2. In the upper right corner of the page, choose **Resources** > **My Quotas**.

   The **Service Quota** page is displayed.

   Figure 8-2 My quotas

3. Click **Increase Quota**.
4. On the **Create Service Ticket** page, configure parameters as required.

   In the **Problem Description** area, fill in the content and reason for adjustment.
5. Select the agreement and click **Submit**.

9 DLI Permissions Management

9.1 How Do I Do If I Receive an Error Message Stating That I Do Not Have Sufficient Permissions When Creating a Table After Upgrading the Engine Version of a Queue?

Symptom

After upgrading the queue version from Spark 2.*x* to Spark 3.3.*x* or switching to HetuEngine, I still receive an error message stating that I do not have sufficient permissions when trying to create a table, even though I have been granted table creation permissions.

Possible Causes

The authorization methods supported may differ depending on the engine version of the DLI queue.

HetuEngine does not support IAM user authorization and requires resource authorization provided by the DLI console.

Solution

Grant users the permission to create tables by referring to **Managing Database Permissions**.

9.2 What Is Column-Level Authorization for DLI Partitioned Tables?

You are unable to perform permission operations on the partition columns of partitioned tables.

However, when you grant the permission of any non-partition column in a partitioned table to another user, the user gets the permission of the partition column by default.

When the user views the permission of the partition table, the permission of the partition column will not be displayed.

9.3 How Do I Do If I Encounter Insufficient Permissions While Updating Packages?

Symptom

When the user update an existing program package, the following error information is displayed:

```
"error_code"*DLI.0003","error_msg":"Permission denied for resource 'resources. xxx', User = 'xxx', Action =
"UPDATE_RESOURCE'."
```

Solution

You need to assign the package permission to the user who executes the job. The procedure is as follows:

1. In the left navigation pane of the DLI management console, choose **Data Management** > **Package Management**.
2. On the **Package Management** page, click **Manage Permission** in the **Operation** column of the package. The **User Permissions** page is displayed.
3. Click **Grant Permission** in the upper right corner of the page to authorize a user to access a package group or package. Select the **Update Group** permission.
4. Click **OK**.

9.4 Why Is Error "DLI.0003: Permission denied for resource..." Reported When I Run a SQL Statement?

Symptom

When the SQL query statement is executed, the system displays a message indicating that the user does not have the permission to query resources.

Error information: **DLI.0003: Permission denied for resource 'databases.dli_test.tables.test.columns.col1', User = '{UserName}', Action = 'SELECT'**

Solution

The user does not have the permission to query the table.

In the navigation pane on the left of the DLI console page, choose **Data Management** > **Databases and Tables**, search for the desired database table, view the permission configuration, and grant the table query permission to the user who requires it.

For details about how to grant permission, see **Table Permission Management**.

9.5 How Do I Do If I Can't Query Table Data After Being Granted Table Permissions?

If you have already granted authorization to a table and a test query was successful, but you encounter an error when trying to query it again after some time, you should first check if the current table permissions still exist:

- Check if the permissions still exist:

  If your permissions have been revoked, it may result in a missing permission error when trying to query the table data.

- Check the table creation time:

  Check if the table has been deleted and recreated by others. A table with the same name as a deleted table does not inherit the permissions of the deleted table and is not considered the same table.

9.6 Will Granting Duplicate Permissions to a Table After Inheriting Database Permissions Cause an Error?

If a table inherits database permissions, you do not need to regrant the inherited permissions to the table.

Re-authorizing may cause confusion in table permission management, as the inherited permissions are already sufficient.

When you grant permissions on a table on the console:

- If you set **Authorization Object** to **User** and select the permissions that are the same as the inherited permissions, the system displays a message indicating that the permissions already exist and do not need to be regranted.
- If you set **Authorization Object** to **Project** and select the permissions that are the same as the inherited permissions, the system does not notify you of duplicate permissions.

9.7 Why Can't I Query a View After I'm Granted the Select Table Permission on the View?

Symptom

User A created Table1.

User B created View1 based on Table1.

After the **Select Table** permission on Table1 is granted to user C, user C fails to query View1.

Possible Causes

User B does not have the **Select Table** permission on Table1.

Solution

Grant the **Select Table** permission on Table1 to user B. Then, query View1 as user C again.

9.8 How Do I Do If I Receive a Message Saying I Don't Have Sufficient Permissions to Submit My Jobs to the Job Bucket?

Symptom

Despite configuring a job bucket and authorizing DLI to access it, I still receive an error message stating that DLI is not authorized to access the bucket when attempting to submit a job.

Possible Causes

To use a DLI job bucket, ensure that necessary permissions for the bucket have been configured.

To ensure that DLI can perform necessary operations, check the bucket policy of the DLI job bucket on the OBS management console and verify that it contains the required authorization information.

Make sure that there are no policies explicitly denying DLI access to the bucket. IAM policies prioritize deny permissions over allow permissions, which means that even if there are allow permissions, the presence of deny permissions will result in authorization failure.

Solution

1. Find the DLI job bucket on the OBS management console.
2. View the policy of the bucket you select.

   The authorization information required for DLI Flink jobs to use the bucket is as follows: *domainId* and *userId* are the DLI account and sub-account, respectively, *bucketName* is the user's bucket name, and *timeStamp* is the timestamp when the policy was created.

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "ID": [
                    "domain/domainId:user/userId"
                ]
            },
            "Action": [
                "GetObject",
                "GetObjectVersion",
                "PutObject",
                "DeleteObject",
                "DeleteObjectVersion",
                "ListMultipartUploadParts",
                "AbortMultipartUpload",
                "GetObjectAcl",
                "GetObjectVersionAcl"
            ],
            "Resource": [
                "bucketName/*"
            ],
            "Sid": "Untitled bucket policy-Timestamp-0"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "ID": [
                    "domain/domainId:user/userId "
                ]
            },
            "Action": [
                "HeadBucket",
                "ListBucket",
                "ListBucketVersions",
                "ListBucketMultipartUploads",
                "GetBucketAcl",
                "GetBucketLocation",
                "GetBucketLogging",
                "GetLifecycleConfiguration"
            ],
            "Resource": [
                " bucketName "
            ],
            "Sid": "Untitled bucket policy-Timestamp-1"
        }
    ]
}
```

3. Check the following permission content on the management console and verify if the policy name matches the one in 2.
   - **Effect**: Select **Allow**.
   - **Resources**: Authorize buckets and objects as needed.
   - **Actions**: Select those configured for **Action** in 2.

   Common check items:
   - Check if any deny operations are configured for all accounts, and if these operations are the required authorization operations for DLI.
   - Check if any deny operations are configured for the authorized users of DLI, and if these operations are the required authorization operations for DLI.

9.9 How Do I Resolve an Unauthorized OBS Bucket Error?

DLI's agency **dli_admin_agency** is upgraded to **dli_management_agency**.

dli_management_agency contains the permissions required for datasource operations, message notifications, and user authorization operations. For other agency permission requirements, you need to create custom DLI agencies.

dli_management_agency does not contain the permission for DLI to read and write OBS. You need to create a custom agency and configure the agency in the job. (This is required when Flink 1.15, Spark 3.3, or later is used to execute jobs.)

For details about **dli_management_agency**, see **DLI Agency Overview**.

For how to create a custom agency and configure an agency in a job, see **Customizing DLI Agency Permissions**.

10 DLI APIs

10.1 How Do I Obtain the AK/SK Pair?

The access key ID (AK) and secret access key (SK) are a pair of access keys used together to authenticate users who wish to make API requests. The AK/SK pair provides functions similar to a password. When users make API requests to manage cloud resources (for example, creating a cluster), the AK/SK pair is required to sign the requests. This mechanism ensures the confidentiality and integrity of the requests as well as the correctness of the identities of both parties. Access keys can be generated and managed on the **My Credentials** page. To obtain the AK/SK pair, perform the following steps:

1. Register with and log in to the Huawei Cloud management console.
2. Move the cursor over your username in the upper right corner of the management console and click **My Credentials** from the drop-down list.
3. In the navigation pane, select **Access Keys**.
4. Click **Create Access Key**. The **Create Access Key** dialog box is displayed.
5. Enter the required information as prompted and click **OK**. On the displayed page, click **Download**.
6. Open the file to obtain the AK/SK information.

NOTE

Keep the AK/SK file somewhere safe to prevent information leakage.

10.2 How Do I Obtain the Project ID?

A project ID is the ID of the region where a system resides. When you access the public cloud system through APIs to perform operations on cloud resources (for example, creating a cluster), you must provide a project ID.

To view the project ID, perform the following steps:

1. Register with and log in to the Huawei Cloud management console.
2. Move the cursor over your username in the upper right corner of the management console and click **My Credentials** from the drop-down list.

On the displayed **My Credentials** page, view the project ID on the **Projects** page, View project IDs. For example, **5a3314075bfa49b9ae360f4ecd333695**.

10.3 Why Is Error "unsupported media Type" Reported When I Subimt a SQL Job?

In the REST API provided by DLI, the request header can be added to the request URI, for example, **Content-Type**.

Content-Type indicates the request body type or format. The default value is **application/json**.

URI for submitting a SQL job: **POST /v1.0/{project_id}/jobs/submit-job**

Content-Type can be only **application/json**. If **Content-Type** is set to **text**, "unsupported media Type" is displayed.

10.4 What Can I Do If an Error Is Reported When the Execution of the API for Creating a SQL Job Times Out?

Symptom

When the API call for submitting a SQL job times out, and the following error information is displayed:

```
There are currently no resources tracked in the state, so there is nothing to refresh.
```

Possible Causes

The timeout of API calls in synchronous is two minutes. If a call times out, an error will be reported.

Solution

When you make a call of the API for submitting a SQL job, set **dli.sql.sqlasync.enabled** to **true** to run the job asynchronously.

For details, see **Submitting a SQL Job**.

10.5 How Can I Fix Garbled Chinese Characters Returned by an API?

When Chinese characters appear as garbled text in the API response, it is usually due to a mismatch in character encoding formats.

The results returned by DLI APIs are encoded in UTF-8. You need to convert the encoding format of the returned information to UTF-8.

The following example converts the encoding format of the returned **response.content** to UTF-8 to ensure that the Chinese characters are displayed properly.

```
print(response.content.decode("utf-8"))
```