

Data Lake Insight

FAQs

Issue 01
Date 2022-09-19



Copyright © Huawei Technologies Co., Ltd. 2023. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Contents

1 Flink Jobs	1
1.1 Usage.....	1
1.1.1 What Data Formats and Data Sources Are Supported by DLI Flink Jobs?.....	1
1.1.2 How Do I Authorize a Subuser to View Flink Jobs?.....	2
1.1.3 How Do I Set Auto Restart upon Exception for a Flink Job?.....	4
1.1.4 How Do I Save Flink Job Logs?.....	5
1.1.5 How Can I Check Flink Job Results?.....	6
1.1.6 Why Is Error "No such user. userName:xxxx." Reported on the Flink Job Management Page When I Grant Permission to a User?.....	6
1.1.7 How Do I Know Which Checkpoint the Flink Job I Stopped Will Be Restored to When I Start the Job Again?.....	6
1.1.8 Which Flink Version Does DLI Support? Is Flink 1.13 Supported? Which Version Is the Next?.....	7
1.1.9 Why Is a Message Displayed Indicating That the SMN Topic Does Not Exist When I Use the SMN Topic in DLI?.....	7
1.2 Flink SQL.....	7
1.2.1 How Much Data Can Be Processed in a Day by a Flink SQL Job?.....	7
1.2.2 Does Data in the Temporary Stream of Flink SQL Need to Be Cleared Periodically? How Do I Clear the Data?.....	7
1.2.3 Why Is a Message Displayed Indicating That the OBS Bucket Is Not Authorized When I Select an OBS Bucket for a Flink SQL Job?.....	8
1.2.4 How Do I Create an OBS Partitioned Table for a Flink SQL Job?.....	8
1.2.5 How Do I Change the Number of Kafka Partitions of a Flink SQL Job Without Stopping It?.....	9
1.2.6 How Do I Dump Data to OBS and Create an OBS Partitioned Table?.....	9
1.2.7 Why Is Error Message "DLI.0005" Displayed When I Use an EL Expression to Create a Table in a Flink SQL Job?.....	10
1.2.8 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?.....	10
1.2.9 Why Does a Flink SQL Job Fail to Be Executed, and Is "connect to DIS failed java.lang.IllegalArgumentException: Access key cannot be null" Displayed in the Log?.....	13
1.2.10 Why Is Error "Not authorized" Reported When a Flink SQL Job Reads DIS Data?.....	13
1.2.11 Data Writing Fails After a Flink SQL Job Consumed Kafka and Sank Data to the Elasticsearch Cluster.....	14
1.2.12 How Does Flink Opensource SQL Parse Nested JSON?.....	14
1.2.13 Why Is the RDS Database Time Read by a Flink Opensource SQL Job Different from RDS Database Time?.....	16
1.2.14 What Are the Syntax Differences Between Flink SQL and Flink Opensource SQL?.....	17

1.2.15 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?.....	17
1.3 Flink Jar Jobs.....	17
1.3.1 How Do I Configure Checkpoints for Flink Jar Jobs and Save the Checkpoints to OBS?.....	17
1.3.2 Does a Flink JAR Job Support Configuration File Upload? How Do I Upload a Configuration File?...	20
1.3.3 Why Does the Submission Fail Due to Flink JAR File Conflict?.....	20
1.3.4 Why Does a Flink Jar Job Fail to Access GaussDB(DWS) and a Message Is Displayed Indicating Too Many Client Connections?.....	21
1.3.5 Why Is Error Message "Authentication failed" Displayed During Flink Jar Job Running?.....	22
1.3.6 Why Is Error Invalid OBS Bucket Name Reported After a Flink Job Submission Failed?.....	23
1.3.7 Why Does the Flink Submission Fail Due to Hadoop JAR File Conflict?.....	23
1.3.8 How Do I Connect a Flink jar Job to SASL_SSL?.....	24
1.4 Performance Tuning.....	24
1.4.1 What Is the Recommended Configuration for a Flink Job?.....	24
1.4.2 How Do I Optimize Performance of a Flink Job?.....	27
1.4.3 How Do I Write Data to Different Elasticsearch Clusters in a Flink Job?.....	31
1.4.4 How Do I Prevent Data Loss After Flink Job Restart?.....	31
1.5 O&M Guide.....	33
1.5.1 How Do I Locate a Flink Job Submission Error?.....	33
1.5.2 How Do I Locate a Flink Job Running Error?.....	34
1.5.3 How Do I Know Whether a Flink Job Can Be Restored from a Checkpoint After Being Restarted?....	35
1.5.4 Why Does DIS Stream Not Exist During Job Semantic Check?.....	35
1.5.5 Why Is the OBS Bucket Selected for Job Not Authorized?.....	35
1.5.6 Why Are Logs Not Written to the OBS Bucket After a DLI Flink Job Fails to Be Submitted for Running?.....	35
1.5.7 How Do I Configure Connection Retries for Kafka Sink If it is Disconnected?.....	36
1.5.8 Why Is Information Displayed on the FlinkUI/Spark UI Page Incomplete?.....	36
1.5.9 Why Is the Flink Job Abnormal Due to Heartbeat Timeout Between JobManager and TaskManager?	37
1.5.10 Why Is Error "Timeout expired while fetching topic metadata" Repeatedly Reported in Flink JobManager Logs?.....	38
2 Problems Related to SQL Jobs.....	39
2.1 Usage.....	39
2.1.1 SQL Jobs.....	39
2.2 Job Development.....	40
2.2.1 How Do I Merge Small Files?.....	40
2.2.2 How Do I Use DLI to Access Data in an OBS Bucket?.....	40
2.2.3 How Do I Specify an OBS Path When Creating an OBS Table?.....	40
2.2.4 How Do I Create a Table Using JSON Data in an OBS Bucket?.....	41
2.2.5 How Can I Use the count Function to Perform Aggregation?.....	41
2.2.6 How Do I Synchronize DLI Table Data from One Region to Another?.....	41
2.2.7 How Do I Insert Table Data into Specific Fields of a Table Using a SQL Job?.....	42
2.2.8 How Do I Delete Table Data?.....	42

2.3 Job O&M Errors.....	42
2.3.1 Why Is Error "path obs://xxx already exists" Reported When Data Is Exported to OBS?.....	42
2.3.2 Why Is Error "SQL_ANALYSIS_ERROR: Reference 't.id' is ambiguous, could be: t.id, t.id.;" Displayed When Two Tables Are Joined?.....	42
2.3.3 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?.....	42
2.3.4 Why Is Error "There should be at least one partition pruning predicate on partitioned table XX.YYY" Reported When a Query Statement Is Executed?.....	43
2.3.5 Why Is Error "IllegalArgumentException: Buffer size too small. size" Reported When Data Is Loaded to an OBS Foreign Table?.....	43
2.3.6 Why Is Error "DLI.0002 FileNotFoundException" Reported During SQL Job Running?.....	44
2.3.7 Why Is a Schema Parsing Error Reported When I Create a Hive Table Using CTAS?.....	44
2.3.8 Why Is Error "org.apache.hadoop.fs.obs.OBSIOException" Reported When I Run DLI SQL Scripts on DataArts Studio?.....	44
2.3.9 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?.....	45
2.3.10 Why Is Error "File not Found" Reported When I Access a SQL Job?.....	46
2.3.11 Why Is Error "DLI.0003: AccessControlException XXX" Reported When I Access a SQL Job?.....	46
2.3.12 Why Is Error "DLI.0001: org.apache.hadoop.security.AccessControlException: verifyBucketExists on {{bucket name}}: status [403]" Reported When I Access a SQL Job?.....	46
2.3.13 Why Is Error "The current account does not have permission to perform this operation,the current account was restricted. Restricted for no budget" Reported During SQL Statement Execution? Restricted for no budget.....	47
2.4 O&M Guide.....	47
2.4.1 How Do I Troubleshoot Slow SQL Jobs?.....	47
2.4.2 How Do I View DLI SQL Logs?.....	52
2.4.3 How Do I View SQL Execution Records?.....	53
2.4.4 How Do I Eliminate Data Skew by Configuring AE Parameters?.....	53
2.4.5 What Can I Do If a Table Cannot Be Queried on the DLI Console?.....	54
2.4.6 The Compression Ratio of OBS Tables Is Too High.....	54
2.4.7 How Can I Avoid Garbled Characters Caused by Inconsistent Character Codes?.....	55
2.4.8 Do I Need to Grant Table Permissions to a User and Project After I Delete a Table and Create One with the Same Name?.....	55
2.4.9 Why Can't I Query Table Data After Data Is Imported to a DLI Partitioned Table Because the File to Be Imported Does Not Contain Data in the Partitioning Column?.....	55
2.4.10 How Do I Fix the Data Error Caused by CRLF Characters in a Field of the OBS File Used to Create an External OBS Table?.....	56
2.4.11 Why Does a SQL Job That Has Join Operations Stay in the Running State?.....	57
2.4.12 The on Clause Is Not Added When Tables Are Joined. Cartesian Product Query Causes High Resource Usage of the Queue, and the Job Fails to Be Executed.....	57
2.4.13 Why Can't I Query Data After I Manually Add Data to the Partition Directory of an OBS Table?....	58
2.4.14 Why Is All Data Overwritten When insert overwrite Is Used to Overwrite Partitioned Table?.....	58
2.4.15 Why Is a SQL Job Stuck in the Submitting State?.....	58
2.4.16 Why Is the create_date Field in the RDS Table Is a Timestamp in the DLI query result?.....	59
2.4.17 What Can I Do If datasize Cannot Be Changed After the Table Name Is Changed in a Finished SQL Job?.....	59

2.4.18 Why Is the Data Volume Changes When Data Is Imported from DLI to OBS?.....	59
3 Problems Related to Spark Jobs.....	61
3.1 Usage.....	61
3.1.1 Spark Jobs.....	61
3.2 Job Development.....	62
3.2.1 How Do I Use Spark to Write Data into a DLI Table?.....	62
3.2.2 How Do I Set the AK/SK for a Queue to Operate an OBS Table?.....	62
3.2.3 How Do I View the Resource Usage of DLI Spark Jobs?.....	63
3.2.4 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?.....	65
3.2.5 How Do I Run a Complex PySpark Program in DLI?.....	65
3.2.6 How Does a Spark Job Access a MySQL Database?.....	67
3.2.7 How Do I Use JDBC to Set the spark.sql.shuffle.partitions Parameter to Improve the Task Concurrency?.....	67
3.2.8 How Do I Read Uploaded Files for a Spark Jar Job?.....	67
3.3 Job O&M Errors.....	68
3.3.1 What Can I Do When Receiving java.lang.AbstractMethodError in the Spark Job?.....	68
3.3.2 Why Are Errors "ResponseCode: 403" and "ResponseStatus: Forbidden" Reported When a Spark Job Accesses OBS Data?.....	69
3.3.3 Why Is Error "verifyBucketExists on XXXX: status [403]" Reported When I Use a Spark Job to Access an OBS Bucket That I Have Access Permission?.....	69
3.3.4 Why Is a Job Running Timeout Reported When a Spark Job Runs a Large Amount of Data?.....	69
3.3.5 Why Does the Job Fail to Be Executed and the Log Shows that the File Directory Is Abnormal When I Use a Spark Job to Access Files in SFTP?.....	69
3.3.6 Why Does the Job Fail to Be Executed Due to Insufficient Database and Table Permissions?.....	70
3.3.7 Why Can't I Find the global_temp Database in the Spark 3.x Job Log?.....	70
3.3.8 Why Does the DataSource Syntax Fail to Create an OBS Table in Avro Format When Spark 2.3.x Is Used?.....	71
3.4 O&M Guide.....	71
3.4.1 Why Can't I Find the Specified Python Environment After Adding the Python Package?.....	71
3.4.2 Why Is a Spark Jar Job Stuck in the Submitting State?.....	71
4 Product Consultation.....	73
4.1 Usage.....	73
4.1.1 What Is DLI?.....	73
4.1.2 What Are the Application Scenarios of DLI?.....	73
4.1.3 Which Data Formats Does DLI Support?.....	75
4.1.4 What Are the Differences Between DLI Flink and MRS Flink?.....	76
4.1.5 What Are the Differences Between MRS Spark and DLI Spark?.....	78
4.1.6 Where Can DLI Data Be Stored?.....	79
4.1.7 What Are the Differences Between DLI Tables and OBS Tables?.....	79
4.1.8 How Can I Use DLI If Data Is Not Uploaded to OBS?.....	79
4.1.9 Can I Import OBS Bucket Data Shared by Other Tenants into DLI?.....	79

4.1.10 Why Is Error "Failed to create the database. {"error_code":"DLI.1028";"error_msg":"Already reached the maximum quota of databases:XXX"." Reported?.....	79
4.1.11 Can a Member Account Use Global Variables Created by Other Member Accounts?.....	80
4.2 Job Management.....	81
4.2.1 How Do I Manage Tens of Thousands of Jobs Running on DLI?.....	81
4.2.2 How Do I Change the Name of a Field in a Created Table?.....	81
4.3 Privacy and Security.....	81
4.3.1 Does DLI Have the Apache Spark Command Injection Vulnerability (CVE-2022-33891)?.....	81
5 Billing.....	82
5.1 What Is the Billing Mode of DLI?.....	82
5.2 Can I Change a Yearly/Monthly Queue to a Pay-per-Use Queue?.....	83
5.3 Can I Change a Pay-Per-Use Queue to a Yearly/Monthly Queue?.....	83
5.4 When Is a Data Lake Queue Idle?.....	83
5.5 How Do I Check the Billing?.....	83
5.6 What Is the Difference Between the Following Two Payment Modes: One Is to Purchase 4,000-CU Resources for Three Months at a Time, the Other Is to Purchase 4,000-CU Resources for One Month for Three Times?.....	84
5.7 How Do I Purchase 30-CU Resources?.....	85
5.8 How Will I Be Billed if Only Two CUs Are Used to Run a Flink Job on a Queue of 16 CUs?.....	85
5.9 Do I Need to Pay Extra Fees for Purchasing a Queue Billed Based on the Scanned Data Volume?.....	85
6 Quota.....	86
6.1 What Is User Quota?.....	86
6.2 How Do I View My Quotas?.....	86
6.3 How Do I Increase a Quota?.....	87
7 Permission.....	88
7.1 Usage.....	88
7.1.1 How Do I Manage Fine-Grained DLI Permissions?.....	88
7.1.2 What Is Column Permission Granting of a DLI Partition Table?.....	89
7.2 O&M Guide.....	90
7.2.1 Why Does My Account Have Insufficient Permissions Due to Arrears?.....	90
7.2.2 Why Does the System Display a Message Indicating Insufficient Permissions When I Update a Program Package?.....	90
7.2.3 Why Is Error "DLI.0003: Permission denied for resource..." Reported When I Run a SQL Statement?.....	91
7.2.4 Why Can't I Query Table Data After I've Been Granted Table Permissions?.....	91
7.2.5 Will an Error Be Reported if the Inherited Permissions Are Regranted to a Table That Inherits Database Permissions?.....	91
7.2.6 Why Can't I Query a View After I'm Granted the Select Table Permission on the View?.....	92
8 Queue.....	93
8.1 Usage.....	93
8.1.1 Does the Description of a DLI Queue Can Be Modified?.....	93
8.1.2 Will Table Data in My Database Be Lost If I Delete a Queue?.....	93

8.1.3 How Does DLI Ensure the Reliability of Spark Jobs When a Queue Is Abnormal?.....	93
8.1.4 How Do I Monitor Queue Exceptions?.....	93
8.2 O&M Guide.....	93
8.2.1 How Do I View DLI Queue Load?.....	94
8.2.2 How Do I Determine Whether There Are Too Many Jobs in the Current Queue?.....	94
8.2.3 How Do I Switch an Earlier-Version Spark Queue to a General-Purpose Queue?.....	95
8.2.4 Why Cannot I View the Resource Running Status of DLI Queues on Cloud Eye?.....	95
8.2.5 How Do I Allocate Queue Resources for Running Spark Jobs If I Have Purchased 64 CUs?.....	95
8.2.6 Why Is Error "Queue plans create failed. The plan xxx target cu is out of quota" Reported When I Schedule CU Changes?.....	95
8.2.7 Why Is a Timeout Exception Reported When a DLI SQL Statement Fails to Be Executed on the Default Queue?.....	96
9 Datasource Connections.....	97
9.1 Datasource Connections.....	97
9.1.1 Why Do I Need to Create a VPC Peering Connection for an Enhanced Datasource Connection?.....	97
9.1.2 Failed to Bind a Queue to an Enhanced Datasource Connection.....	97
9.1.3 DLI Failed to Connect to GaussDB(DWS) Through an Enhanced Datasource Connection.....	98
9.1.4 How Do I Do if the Datasource Connection Is Created But the Network Connectivity Test Fails?.....	99
9.1.5 How Do I Configure the Network Between a DLI Queue and a Data Source?.....	104
9.1.6 What Can I Do If a Datasource Connection Is Stuck in Creating State When I Try to Bind a Queue to It?.....	105
9.1.7 How Do I Bind a Datasource Connection to a Pay-per-Use Queue?.....	105
9.1.8 How Do I Connect DLI to Data Sources?.....	105
9.2 Cross-Source Analysis.....	105
9.2.1 How Can I Perform Query on Data Stored on Services Rather Than DLI?.....	106
9.2.2 How Can I Access Data Across Regions?.....	106
9.2.3 How Do I Set the Auto-increment Primary Key or Other Fields That Are Automatically Filled in the RDS Table When Creating a DLI and Associating It with the RDS Table?.....	106
9.3 Datasource Connection O&M.....	106
9.3.1 Why Is the Error Message "communication link failure" Displayed When I Use a Newly Activated Datasource Connection?.....	106
9.3.2 Connection Times Out During MRS HBase Datasource Connection, and No Error Is Recorded in Logs.....	109
9.3.3 Why Can't I Find the Subnet When Creating a DLI Datasource Connection?.....	109
9.3.4 Error Message "Incorrect string value" Is Displayed When insert overwrite Is Executed on a Datasource RDS Table.....	110
9.3.5 Null Pointer Error Is Displayed When the System Creates a Datasource RDS Table.....	111
9.3.6 Error Message "org.postgresql.util.PSQLException: ERROR: tuple concurrently updated" Is Displayed When the System Executes insert overwrite on a Datasource GaussDB(DWS) Table.....	111
9.3.7 RegionTooBusyException Is Reported When Data Is Imported to a CloudTable HBase Table Through a Datasource Table.....	112
9.3.8 A Null Value Is Written Into a Non-Null Field When a DLI Datasource Connection Is Used to Connect to a GaussDB(DWS) Table.....	112
9.3.9 An Insert Operation Failed After the Schema of the GaussDB(DWS) Source Table Is Updated.....	114

10 APIs.....	115
10.1 How Do I Obtain the AK/SK Pair?.....	115
10.2 How Do I Obtain the Project ID?.....	115
10.3 Why Is Error "unsupported media Type" Reported When I Subimt a SQL Job?.....	116
10.4 Are Project IDs of Different Accounts the Same When They Are Used to Call APIs?.....	116
10.5 What Can I Do If an Error Is Reported When the Execution of the API for Creating a SQL Job Times Out?.....	116
10.6 What Can I Do If Chinese Characters Returned by an API Are Garbled Characters?.....	116
11 SDKs.....	118
11.1 How Do I Set the Timeout Duration for Querying SQL Job Results Using SDK?.....	118
11.2 How Do I Handle the dli.xxx,unable to resolve host address Error?.....	118
A Change History.....	119

1 Flink Jobs

1.1 Usage

1.1.1 What Data Formats and Data Sources Are Supported by DLI Flink Jobs?

DLI Flink jobs support the following data formats:

Avro, Avro_merge, BLOB, CSV, EMAIL, JSON, ORC, Parquet, and XML.

DLI Flink jobs support data from the following data sources:

CloudTable HBase, CloudTable OpenTSDB, CSS Elasticsearch, DCS, DDS, DIS, DMS, GaussDB(DWS), EdgeHub, MRS HBase, MRS Kafka, open-source Kafka, file systems, OBS, RDS, and SMN

Table 1-1 Supported source and sink streams of corresponding data formats

Data Format	Supported Source Stream	Supported Sink Stream
Avro	-	OBS Sink Stream
Avro_merge	-	OBS Sink Stream
BLOB	<ul style="list-style-type: none">• DIS Source Stream• MRS Kafka Source Stream• Open-Source Kafka Source Stream	-

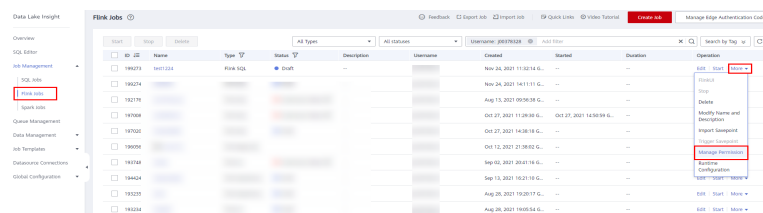
Data Format	Supported Source Stream	Supported Sink Stream
CSV	<ul style="list-style-type: none"> DIS Source Stream OBS Source Stream Open-Source Kafka Source Stream 	<ul style="list-style-type: none"> DIS Sink Stream OBS Sink Stream GaussDB(DWS) Sink Stream (OBS Mode) Open-Source Kafka Sink Stream File System Sink Stream
EMAIL	DIS Source Stream	-
JSON	<ul style="list-style-type: none"> DIS Source Stream OBS Source Stream MRS Kafka Source Stream Open-Source Kafka Source Stream 	<ul style="list-style-type: none"> DIS Sink Stream OBS Sink Stream MRS Kafka Sink Stream
ORC	-	<ul style="list-style-type: none"> GaussDB(DWS) Sink Stream (OBS Mode)
Parquet	-	<ul style="list-style-type: none"> OBS Sink Stream File System Sink Stream
XML	DIS Source Stream	-

1.1.2 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.

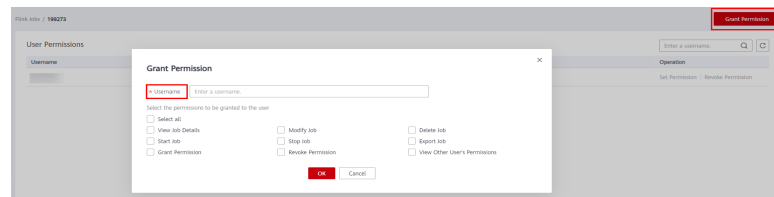
- Authorization on DLI
 - Log in to the DLI console using a tenant account, a job owner account, or an account with the **DLI Service Administrator** permission.
 - Choose **Job Management > Flink Jobs**. On the displayed page, locate the target job.
 - In the **Operation** column of the target job,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

Figure 1-1 Managing Flink job permissions



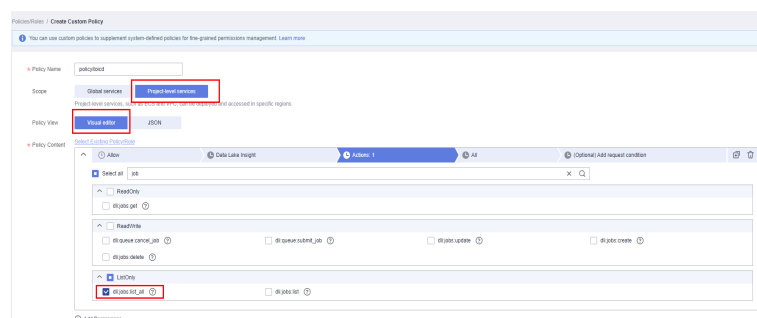
- On the displayed page, click **Grant Permission**. Enter the name of the user to be authorized and select the required permissions. Click **OK**. The authorized user can view the job and perform related operations.

Figure 1-2 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 1-3 Creating a policy



- 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 Admin** (with all DLI permissions) for the subuser to view Flink jobs.

1.1.3 How Do I Set 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 1-4 Editing a Flink SQL job

Running Parameters	Debugging Parameters
* Job Manager CUs	<input type="text" value="1"/>
* Parallelism	<input type="text" value="1"/> <input style="border: none; border-radius: 50%; padding: 2px 5px;" type="button" value="?"/>
Task Manager Configu...	<input type="checkbox"/>
OBS Bucket	<input type="text" value=""/>
Save Job Log	<input type="checkbox"/>
Alarm Generation upo...	<input checked="" type="checkbox"/>
* SMN Topic	<input type="text" value=""/> <input type="button" value="Configure Topic"/>
Enable Checkpointing	<input type="checkbox"/>
Auto Restart upon Exc...	<input type="checkbox"/>
Idle State Retention Ti...	<input type="text" value="1"/> <input type="button" value="h"/>
Dirty Data Policy	<input type="text" value="Save"/>
* Dirty Data Dump Addr...	<input type="text" value=""/>

* Queue	<input type="text" value="Shared queue (Free for Ope..."/>

1.1.4 How Do I Save Flink Job Logs?

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 details about how to create a Flink SQL job or Flink Jar job, see [Creating a Flink SQL Job](#).

1.1.5 How Can I Check Flink Job Results?

- DLI can output Flink job results to RDS. You can view the results in RDS. For details, see [Relational Database Service Getting Started](#).
- DLI can output Flink job results to SMN, and SMN sends the results to the user's terminal. For details, see [Simple Message Notification Getting Started](#).
- DLI can output Flink job results to Kafka. You can view the results in Kafka clusters. For details, visit the [Kafka official website](#).
- DLI can output Flink job results to CloudTable. You can view the results in CloudTable. For details, see *CloudTable Service User Guide*.
- DLI can export Flink job results to CSS. You can view the results in CSS. For details, see "Getting Started" in *Cloud Search Service User Guide*.
- DLI can export Flink job results to DCS. You can view the results in DCS. For details, see [Distributed Cache Service Getting Started](#).

1.1.6 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

Check whether the username exists and whether the user has logged in to the system before authorization.

1.1.7 How Do I Know Which Checkpoint the Flink Job I Stopped Will Be Restored to When I Start the Job Again?

Symptom

Checkpoint was enabled when a Flink job is created, and the OBS bucket for storing checkpoints was specified. After a Flink job is manually stopped, no message is displayed specifying the checkpoint where the Flink job will be restored if the Flink job is started again.

Solution

The generation mechanism and format of Flink checkpoints are the same as those of savepoints. You can import a savepoint of the job to restore it from the latest checkpoint saved in OBS.

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 from the imported savepoint.

1.1.8 Which Flink Version Does DLI Support? Is Flink 1.13 Supported? Which Version Is the Next?

Currently, Flink 1.12 and later versions are supported. Flink 1.13 is not supported. DLI will support Flink 1.15. Stay tuned.

1.1.9 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 the SMN topic you select for your DLI job does not exist, log in to the Identity and Access Management (IAM) console, select the user group to which the IAM user belongs, and add the SMN policy for your region to the group.

1.2 Flink SQL

1.2.1 How Much Data Can Be Processed in a Day by a Flink SQL Job?

The consumption capability of a Flink SQL job depends on the data source transmission, queue size, and job parameter settings. The peak consumption is 10 Mbit/s.

1.2.2 Does Data in the Temporary Stream of Flink SQL Need to Be Cleared Periodically? How Do I Clear the Data?

The temp stream in Flink SQL is similar to a subquery. It is a logical stream used to simplify the SQL logic and does not generate data storage. Therefore, there is no need to clean data.

1.2.3 Why Is a Message Displayed Indicating That the OBS Bucket Is Not Authorized When I Select an OBS Bucket for a Flink SQL Job?

- Symptom

When you create a Flink SQL job and configure the parameters, you select an OBS bucket you have created. The system displays a message indicating that the OBS bucket is not authorized. After you click **Authorize**, the system displays a message indicating that an internal error occurred on the server and you need to contact Huawei Cloud customer service or try again later.

- Solution

On the settings page, press F12 to view the error details. The following is an example:

```
{"error_msg":"An internal error occurred. {0} Contact Huawei Cloud customer services or try again later ","error_json_opt":{"error": "Unexpected exception[NoSuchElementException: None.get]"},"error_code":"DLI.10001"}
```

Check whether a DLI agency has been created. If you do not have the permission to create an agency. On the DLI console, choose **Global Configuration > Service Authorization**, select **Tenant Administrator (Global service)**, and click **Update**.

1.2.4 How Do I Create an OBS Partitioned Table for a Flink SQL Job?

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 (only the parquet format is supported currently) to dump **car_info** data to OBS.

```
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;
```

1.2.5 How Do I Change the Number of Kafka Partitions of 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.

1.2.6 How Do I Dump Data to OBS and Create an OBS Partitioned Table?

In this example, the **day** field is used as the partition field with the parquet encoding format (only the parquet format is supported currently) to dump **car_info** data to OBS. For more information, see .

```
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 partition 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;
```

1.2.7 Why Is Error Message "DLI.0005" Displayed When I Use an EL Expression 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 `#{exp}`.

1.2.8 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 1-5 Enable checkpointing

Running Parameters	Debugging Parameters
* CUs	<input type="text" value="2"/> <input type="button" value="−"/> <input type="button" value="+"/> <input type="button" value="ⓘ"/>
* Job Manager CUs	<input type="text" value="1"/> <input type="button" value="−"/> <input type="button" value="+"/>
* Parallelism	<input type="text" value="1"/> <input type="button" value="−"/> <input type="button" value="+"/> <input type="button" value="ⓘ"/>
Task Manager Configu...	<input type="checkbox"/>
* OBS Bucket	<input type="text" value="--Select--"/>
Save Job Log	<input checked="" type="checkbox"/>
Alarm Generation upo...	<input type="checkbox"/>
Enable Checkpointing	<input checked="" type="checkbox"/>
* Checkpoint Interval	<input type="text" value="30"/> <input type="button" value="−"/> <input type="button" value="+"/> s
Checkpoint Mode	<input type="text" value="Exactly once"/> <input type="button" value="▼"/>
Auto Restart upon Exc...	<input type="checkbox"/>
Idle State Retention Ti...	<input type="text" value="1"/> <input type="button" value="−"/> <input type="button" value="+"/> <input type="text" value="h"/> <input type="button" value="▼"/>
Dirty Data Policy	<input type="text" value="--Select--"/> <input type="button" value="▼"/>

* Queue	<input type="text" value="Shared queue (Free for Ope..."/> <input type="button" value="▼"/>

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;
```

1.2.9 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.

1.2.10 Why Is Error "Not authorized" Reported When a Flink SQL Job Reads DIS Data?

Symptom

Semantic verification for a Flink SQL job (reading DIS data) fails. The following information is displayed when the job fails:

```
Get dis channel xxx info failed. error info: Not authorized, please click the overview page to do the authorize action
```

Possible Causes

Before running a Flink job, the permission to obtain DIS data is not granted to the user.

Solution

1. Log in to the DLI management console. Choose **Global Configuration > Service Authorization** in the navigation pane on the left.

2. On the **Service Authorization** page, select **DIS Administrator** and click **Update**.
3. Choose **Job Management > Flink Jobs**. On the displayed page, locate the desired Flink SQL job and restart the job.

1.2.11 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

1.2.12 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,lq 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.

1.2.13 Why Is the RDS Database Time Read by a Flink Opensource SQL Job Different from RDS Database Time?

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 execution result is as follows:

Variable_name	Value
system_time_zone	CST
time_zone	SYSTEM

- **time_zone** indicates the database time zone. The value **SYSTEM** indicates the system time of the database server ('system_time_zone'). The value of the system time zone is **CST**. Therefore, the database time zone is **CST**.
- **system_time_zone** indicates the 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**.

The real problem here is that **time_zone** is set to **SYSTEM** and **system_time_zone** is set to **CST** for an RDS for MySQL database. In the MySQL database, CST is China Standard Time (UTC+8), but it is Central Standard Time (USA) (UTC-5) in Java. 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 (either of the following method)

1. Do not set the value of `time_zone` to `SYSTEM`. You can set it to, for example, `+08:00`.
2. Add the time zone to `jdbcUrl`.
For example, `jdbc:mysql://localhost:3306/test?serverTimezone=Asia/Shanghai`.

1.2.14 What Are the Syntax Differences Between Flink SQL and Flink OpenSource SQL?

Flink SQL is an early in-house syntax of DLI and is incompatible with open-source syntax.

Flink OpenSource SQL is fully compatible with Flink syntax and is continuously iterated with official Flink updates.

Flink OpenSource SQL jobs are recommended.

Reference documents

- [Flink OpenSource SQL 1.12 Syntax](#) (recommended)
- [Flink OpenSource SQL1.10 Syntax](#).

1.2.15 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`.

1.3 Flink Jar Jobs

1.3.1 How Do I Configure Checkpoints for Flink Jar Jobs and Save the Checkpoints to OBS?

The procedure is as follows:

1. Add the following code to the JAR file code of the Flink Jar job:

```
// Configure the pom file on which the StreamExecutionEnvironment depends.
StreamExecutionEnvironment env = StreamExecutionEnvironment.getExecutionEnvironment();

    env.getCheckpointConfig().setCheckpointingMode(CheckpointingMode.EXACTLY_ONCE);
    env.getCheckpointConfig().setCheckpointInterval(40000);

env.getCheckpointConfig().enableExternalizedCheckpoints(CheckpointConfig.ExternalizedCheckpointCleanup.RETAIN_ON_CANCELLATION);
    RocksDBStateBackend rocksDbBackend = new RocksDBStateBackend(new
FsStateBackend("obs://${bucket}/jobs/checkpoint/my_jar"), false);
    rocksDbBackend.setOptions(new OptionsFactory() {
        @Override
        public DBOptions createDBOptions(DBOptions currentOptions) {
            return currentOptions
                .setMaxLogFileSize(64 * 1024 * 1024)
                .setKeepLogFileNum(3);
        }

        @Override
        public ColumnFamilyOptions createColumnOptions(ColumnFamilyOptions currentOptions) {
            return currentOptions;
        }
    });
env.setStateBackend(rocksDbBackend);
```

 **NOTE**

The preceding code saves the checkpoint to the **`\${bucket}`** bucket in **jobs/checkpoint/my_jar** path every 40 seconds in **EXACTLY_ONCE** mode.

Pay attention to the checkpoint storage path. Generally, the checkpoint is stored in the OBS bucket. The path format is as follows:

- Path format: **obs://`\${bucket}`/xxx/xxx/xxx**
- Example

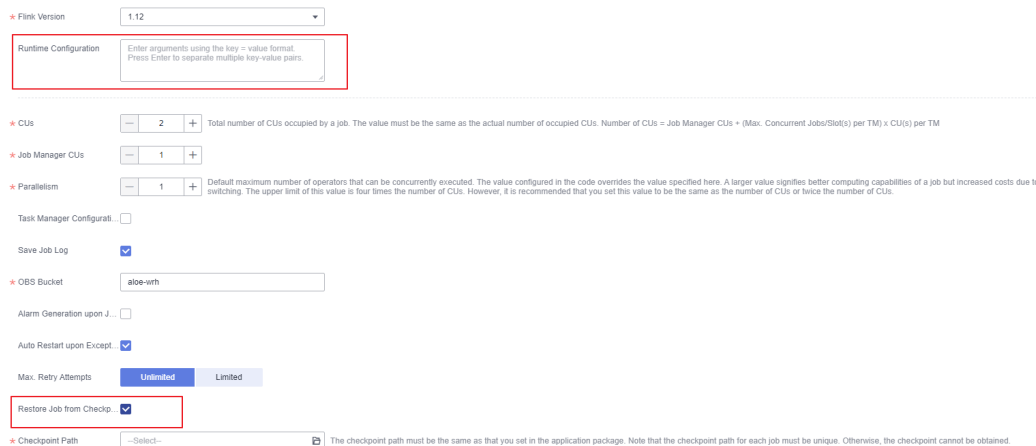
obs://mybucket/jobs/checkpoint/jar-3

- Add the following configuration to the POM file for the packages on which the StreamExecutionEnvironment depends:

```
<dependency>
    <groupId>org.apache.flink</groupId>
    <artifactId>flink-streaming-java_${scala.binary.version}</artifactId>
    <version>${flink.version}</version>
    <scope>provided</scope>
</dependency>
```

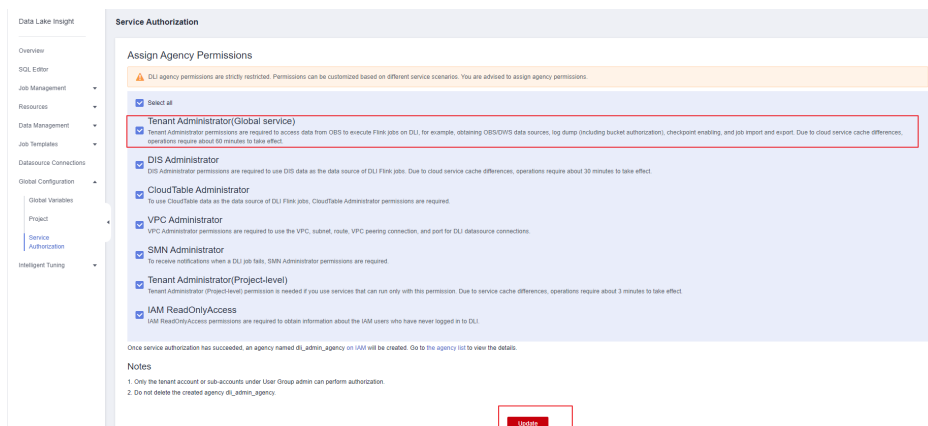
2. Configure **Runtime Configuration** and **Restore Job from Checkpoint** for a DLI Flink Jar job.

Figure 1-6 Configuring Runtime Configuration and Restore Job from Checkpoint



- Constraints on parameter optimization
 - i. In the left navigation pane of the DLI console, choose **Global Configuration > Service Authorization**. On the page displayed, select **Tenant Administrator(Global service)** and click **Update**.

Figure 1-7 Selecting Tenant Administrator(Global service)



- ii. The bucket to which data is written must be an OBS bucket created by a main account.

- **Configuring Restore Job from Checkpoint**

- i. Select **Auto Restart upon Exception**.
- ii. Select **Restore Job from Checkpoint** and set the **Checkpoint Path**.

The checkpoint path is the same as that you set in JAR file code. The format is as follows:

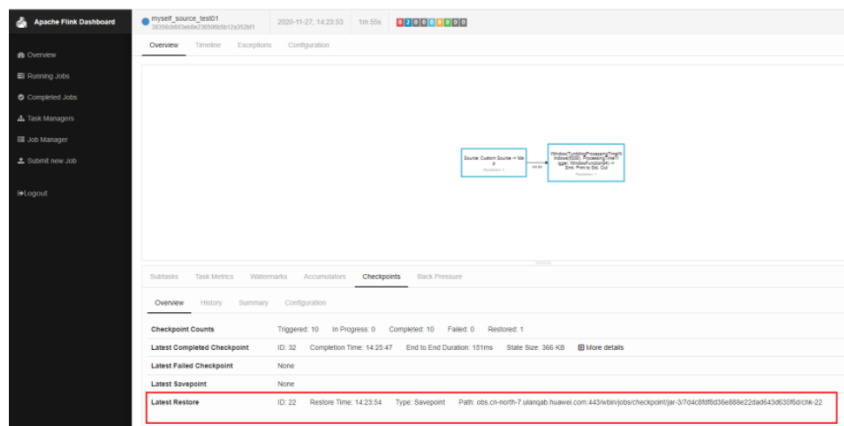
- o `${bucket}/xxx/xxx/xxx`
- o Example

If the path in the JAR file is **obs://mybucket/jobs/checkpoint/jar-3**,

Set **Checkpoint Path** to **mybucket/jobs/checkpoint/jar-3**.

NOTE

- The checkpoint path for each Flink Jar job must be unique. Otherwise, data cannot be restored.
 - DLI can access files in the checkpoint path only after DLI is authorized to access the OBS bucket.
3. Check whether the job is restored from the checkpoint.

Figure 1-8 Checking the restoration

1.3.2 Does a Flink JAR Job Support Configuration File Upload? How Do I Upload a Configuration File?

Configuration files can be uploaded for user-defined 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. In the file name, **fileName** indicates the name of the file to be accessed, and **ClassName** indicates the name of the class that needs to access the file.

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

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

Delete your JAR file that is the same as an existing one of the DLI Flink platform.

For details about DLI built-in dependencies, see [Built-in Dependencies](#).

1.3.4 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
```

Possible Causes

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](#).

1.3.5 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
```

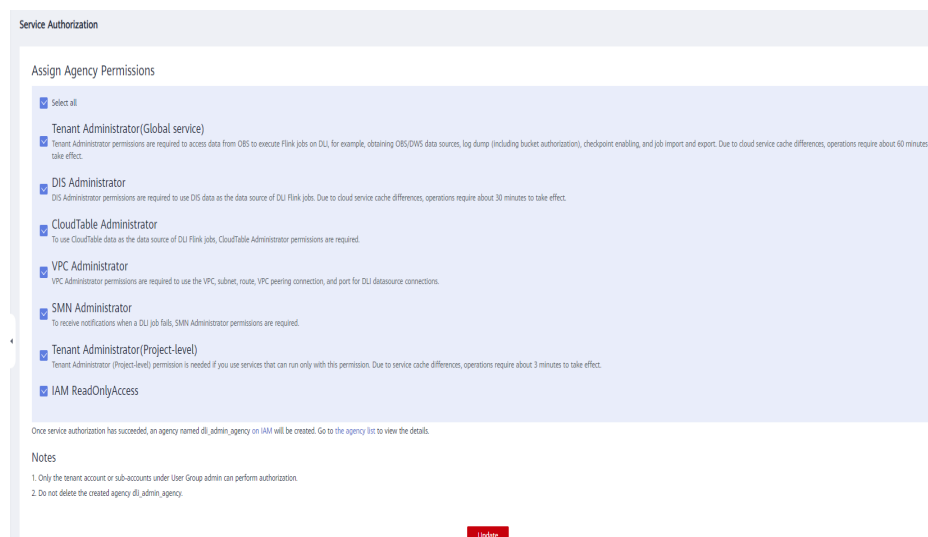
Possible Causes

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. Choose **Global Configuration > Service Authorization** in the navigation pane.
- Step 2** On the **Service Authorization** page, select all agency permissions.

Figure 1-9 Assigning agency permissions



- Step 3** Click **Update**. If the message "Agency permissions updated successfully" is displayed, the modification is successful.
- Step 4** After the authorization is complete, create a datasource connection and run the job again.

----End

1.3.6 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.

For details about how to save checkpoints to OBS, see [How Do I Configure Checkpoints for Flink Jar Jobs and Save the Checkpoints to OBS?](#)

Cause Analysis

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

1.3.7 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

1.3.8 How Do I Connect a Flink jar Job to SASL_SSL?

You can use Flink Jar to connect to Kafka with SASL SSL authentication enabled.

For details, see [Using Flink Jar to Connect to Kafka with SASL_SSL Authentication Enabled](#).

1.4 Performance Tuning

1.4.1 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 1-10 Creating a topic

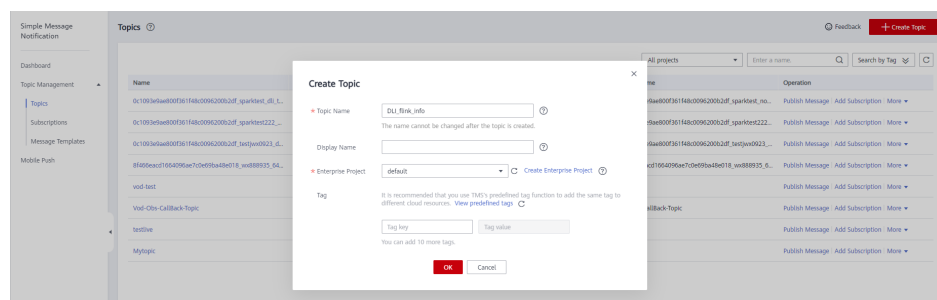


Figure 1-11 Adding a subscription

The screenshot shows a dialog box titled "Add Subscription" with a close button (X) in the top right corner. The dialog contains the following fields and controls:

- Topic Name:** A text input field containing "DLI_fink_info" and a "+" button to its right.
- Protocol:** A dropdown menu currently showing "SMS".
- Endpoints:** A table with two columns: "Endpoints" and "Description".

Endpoints	Description
12312345678	name
- Buttons:** An "Add Endpoint" button with a plus icon, a red "OK" button, and a "Cancel" button.

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 1-12 Specifying a bucket

The screenshot shows a form with the following elements:

- OBS Bucket:** A text input field containing "dli-test-obs01".
- Save Job Log:** A checkbox that is checked, indicated by a blue checkmark.

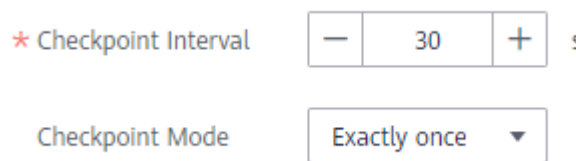
- c. Select **Alarm Generation upon Job Exception** and select the SMN topic created in 1. This allows DLI to send notifications to your email box or phone when a job exception occurs, so you can be aware of any exceptions in time.

Figure 1-13 Alarm generation upon job exception



- 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 1-14 Checkpoint parameters



NOTE

- Checkpoint interval indicates the interval between two triggers. Checkpointing hurts real-time computing performance. To minimize the performance loss, you need to allow for the recovery duration when configuring the interval. It is recommended that the checkpoint interval **be greater than the checkpointing duration**. The recommended value is 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 1-15 Cloud service monitoring



Figure 1-16 Creating an alarm rule

The screenshot shows the 'Create Alarm Rule' configuration page. It includes the following sections:

- Name:** Flink-job-alarm
- Description:** Flink xxxAlarm (15/256 characters)
- Enterprise Project:** default
- Resource Type:** Data Lake Insight
- Dimension:** Flink job
- Monitoring Scope:** Specific resources
- Monitored Object:** long_run
- Method:** Use template (selected) / Configure manually
- Alarm Policy:** A table with columns: Metric Name, Alarm Policy, Alarm Severity, Operation. The current policy is: Metric Name: --Select--, Alarm Policy: Raw d..., Alarm Severity: 3 consecuti..., Operation: >=.
- Alarm Notification:** Enabled (toggle)
- Notification Object:** Account contact
- Validity Period:** 00:00 - 23:59
- Trigger Condition:** Generated alarm (checked), Cleared alarm (checked)

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

1.4.2 How Do I Optimize Performance of a Flink Job?

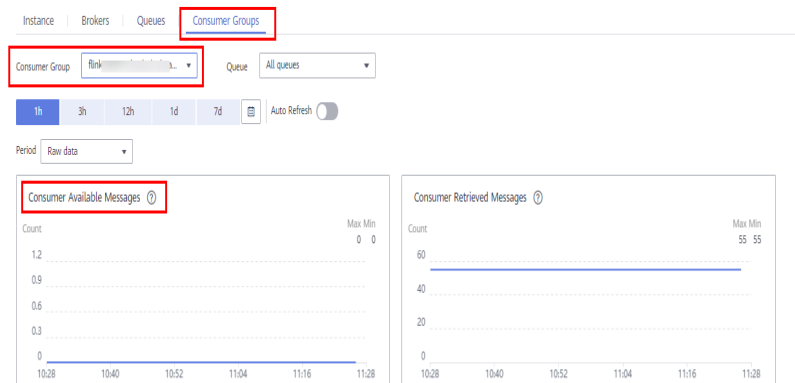
Basic Concepts and Job Monitoring

- 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** from 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 1-17 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 1-18 Back pressure status and delay

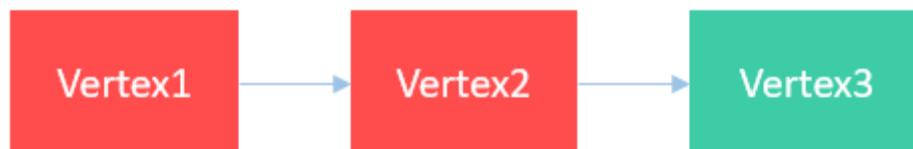
Task List Execution Plan Commit Logs Run Log Tags

Duration	Max Concur...	Task	Status	Back Pressur...	Delay	Sent Records	Sent Bytes	Received Rec...	Received Bytes	Started	Ended
10m 15.62s	1	hadoopTableSource[cat_id, car_owner, car_brand, ...]	Running	HIGH	-	3040744	290.594 MB	0	0 B	Dec 03, 2021 ...	-
10m 15.61s	1	version[table=default, catalog=default, database=...]	Running	HIGH	70	3036914	290.201 MB	3036614	290.113 MB	Dec 03, 2021 ...	-
10m 15.61s	1	fileSink[cat_id, car_owner, car_brand, car_speed]	Running	OK	225	0	0 B	3032630	290 MB	Dec 03, 2021 ...	-

Performance Analysis

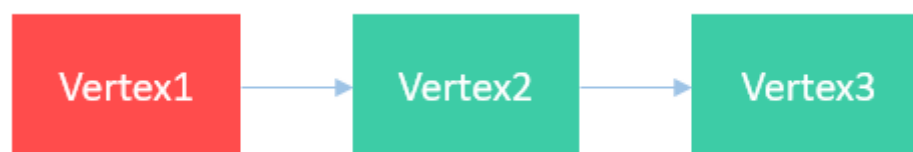
Due to Flink back pressure, the data source consumption rate can be lower than the production rate when performance of a Flink job is low. As a result, data is stacked in a Kafka consumer group. In this case, you can use back pressure and delay of the operator to find its performance bottleneck.

- 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).



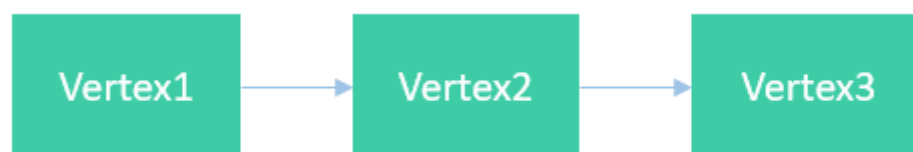
In this scenario, the performance bottleneck is the sink and the optimization is specific to the data source. For example, for the JDBC data source, you can adjust the write batch using **connector.write.flush.max-rows** and JDBC rewriting parameter **rewriteBatchedStatements=true** to optimize the performance.

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



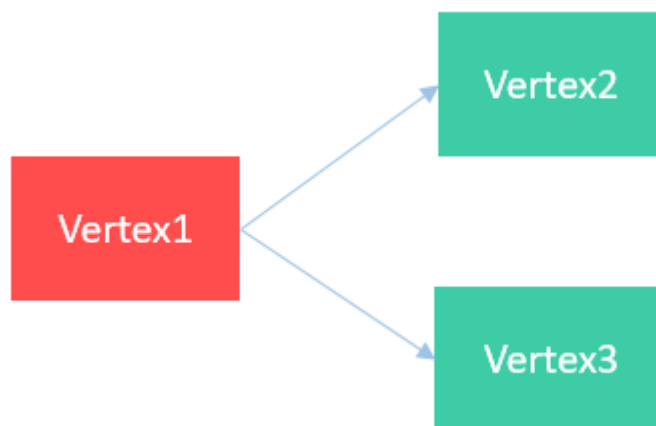
In this scenario, the performance bottleneck is the Vertex2 operator. You can view the description about the function of the operator for further optimization.

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



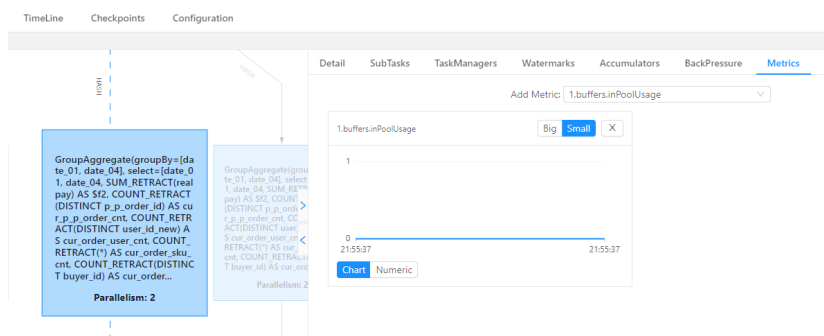
In this scenario, the performance bottleneck is the source, and the performance is mainly affected by the data read speed. In this case, you can increase the number of Kafka partitions and the number of concurrent sources to solve the problem.

- 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 1-19 inPoolUsage monitoring



Performance Tuning

- Rocksdb state tuning

Top N sorting, window aggregate calculation, and stream-stream join involve a large number of state 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-Global 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 is 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.

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

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;
```

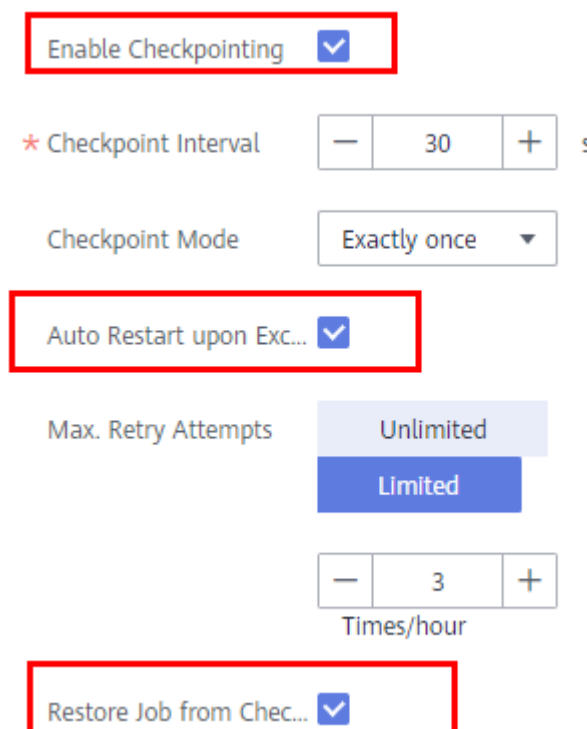
1.4.4 How Do I Prevent Data Loss After Flink Job Restart?

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:

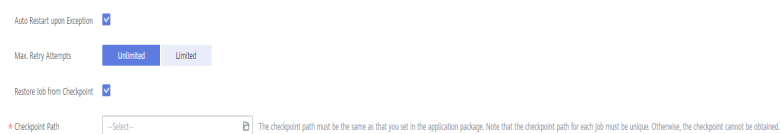
- For Flink SQL jobs, select **Enable Checkpointing** and set a proper checkpoint interval that allows for the impact on service performance and the exception recovery duration. Select **Auto Restart upon Exception** and **Restore Job from Checkpoint**. After the configuration, if a job is restarted 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 1-20 Flink job configuration parameters



- For Flink Jar jobs, you need to enable the checkpoint in the code. If a user-defined state needs to be saved, you need to implement the ListCheckpointed API, set a unique ID for each operator. In the job configuration, select **Restore Job from Checkpoint** and configure the checkpoint path.

Figure 1-21 Enable checkpointing



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 1-22 Stopping a Job

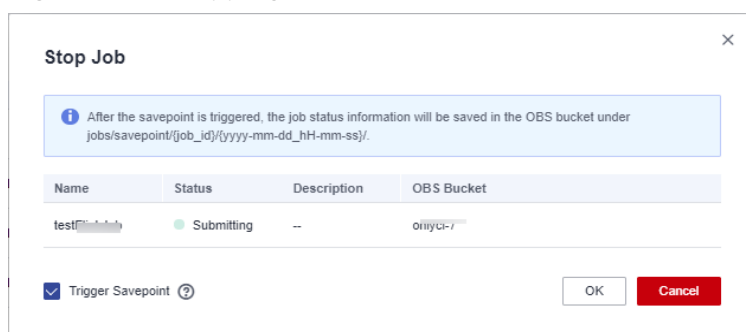
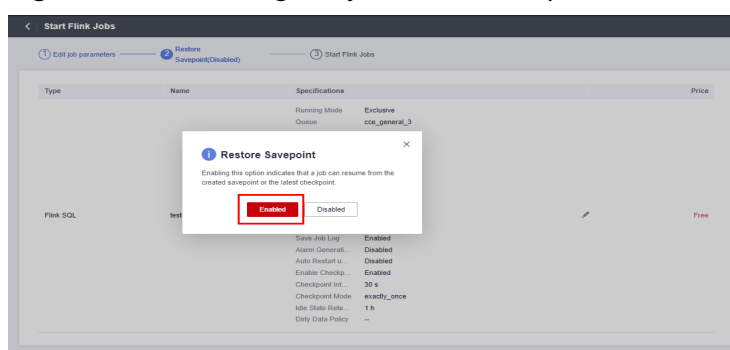


Figure 1-23 Restoring the job from a savepoint or checkpoint



1.5 O&M Guide

1.5.1 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.

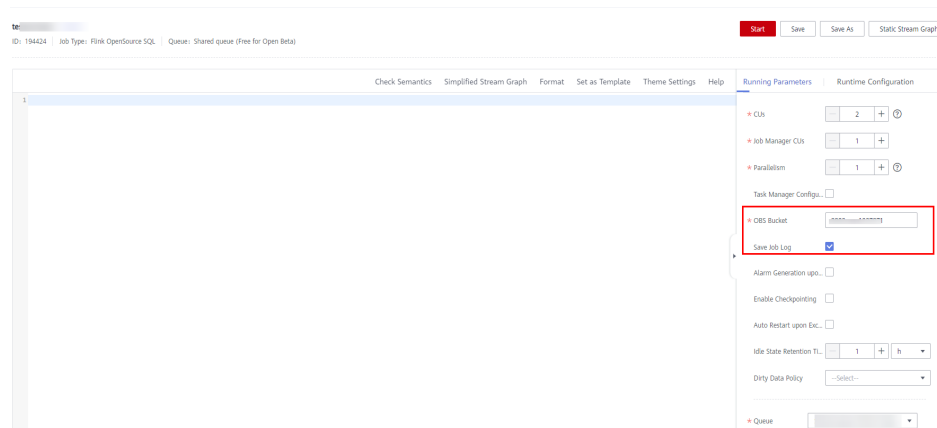
The possible causes are as follows:

- 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.

1.5.2 How Do I Locate a Flink Job Running Error?

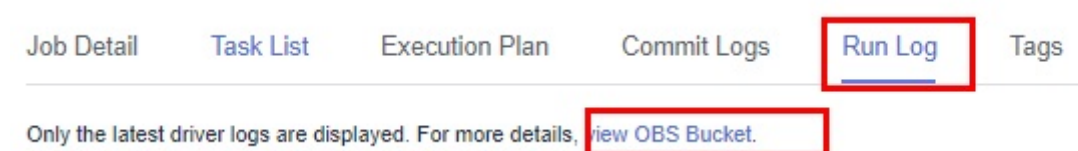
1. On the Flink job management, click **Edit** in the **Operation** column of the target job. On the displayed page, check whether **Save Job Log** in the **Running Parameters** tab is enabled.

Figure 1-24 Enabling Save Job Logs



- 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 1-25 Viewing run logs



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.

1.5.3 How Do I Know Whether a Flink Job Can Be Restored from a Checkpoint After Being Restarted?

Check the following operations:

- Adjusting or adding optimization parameters or the number of concurrent threads of a job, or modifying Flink SQL statements or a Flink Jar job: The job cannot be restored from the checkpoint.
- Modifying the number of CUs occupied by a TaskManager: The job can be restored from the checkpoint.

1.5.4 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 .
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.

1.5.5 Why Is the OBS Bucket Selected for Job Not Authorized?

If the OBS bucket selected for a job is not authorized, perform the following steps:

- Step 1** On the DLI management console, choose **Global Configuration > Service Authorization** and select the **Tenant Administrator (Global service)** permission. Update the permission settings.
- Step 2** In the navigation pane, choose **Job Management > Flink Jobs**.
- Step 3** In the row where the target job resides, click **Edit** under **Operation** to switch to the **Edit** page.
- Step 4** Configure parameters under **Running Parameters** on the **Edit** page.
1. Select **Enable Checkpointing** or **Save Job Log**.
 2. Specify **OBS Bucket**.
 3. Select **Authorize OBS**.

----End

1.5.6 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.

1.5.7 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"  
);
```

1.5.8 Why Is Information Displayed on the FlinkUI/Spark UI Page Incomplete?

Symptom

The Flink/Spark UI was displayed with incomplete information.

Possible Causes

The queue is a pay-per-use one. When the queue is used to run a job, the system releases the cluster and takes about 10 minutes to create a new one. Accessing the Flink UI before completion of the creation will empty the project ID in the cache. As a result, the UI cannot be displayed. The possible cause is that the cluster was not created.

Solution

Change the queue to dedicated, so that the cluster will not be released when the queue is idle. Alternatively, submit a job, wait for a while, and then access FlinkUI.

1.5.9 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 1-26 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 643331ms for sessionId 0x18000013858becca, closing socket connection
and attempting reconnection
2021-05-17 22:46:30.547 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-203666189689296518.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 1-27 Full GC

```
28461 [GC00000000:MarkSweepRate=1:21496328->21496328(21496328), 0.0265116 secs] 2456318K->2456328K(2456328K), (MemSpace= 792238->792238(1120208)), 0.0270750 secs] [Times: user=0.03 sys=0.00, real=0.03 secs]
28462 2021-05-17T22:47:03.020000: 26995.9591 [Full GC (Allocation Failure) 2021-05-17T22:47:03.020000: 26995.9591 [CMS:2021-05-17T22:47:03.020000: 26995.9591 [CMS-concurrent-mark: 0.236(0.236) secs], (MemSpace= 792238->792238(1120208)), 0.5422651 secs] [Times: user=0.07 sys=0.00, real=0.07 secs]
28463 2021-05-17T22:47:03.020000: 26995.9591 [Full GC (Allocation Failure) 2021-05-17T22:47:03.020000: 26995.9591 [CMS:2021-05-17T22:47:03.020000: 26995.9591 [CMS-concurrent-mark: 0.236(0.236) secs], (MemSpace= 792238->792238(1120208)), 0.5422651 secs] [Times: user=0.07 sys=0.00, real=0.07 secs]
28464 2021-05-17T22:47:03.020000: 26995.9591 [Full GC (Allocation Failure) 2021-05-17T22:47:03.020000: 26995.9591 [CMS:2021-05-17T22:47:03.020000: 26995.9591 [CMS-concurrent-mark: 0.236(0.236) secs], (MemSpace= 792238->792238(1120208)), 0.5422651 secs] [Times: user=0.07 sys=0.00, real=0.07 secs]
28465 2021-05-17T22:47:03.020000: 26995.9591 [Full GC (Allocation Failure) 2021-05-17T22:47:03.020000: 26995.9591 [CMS:2021-05-17T22:47:03.020000: 26995.9591 [CMS-concurrent-mark: 0.236(0.236) secs], (MemSpace= 792238->792238(1120208)), 0.5422651 secs] [Times: user=0.07 sys=0.00, real=0.07 secs]
28466 [GC00000000:MarkSweepRate=1:21496328->21496328(21496328), 0.0396028 secs] 2456328K->2456328K(2456328K), (MemSpace= 792238->792238(1120208)), 0.0403887 secs] [Times: user=0.04 sys=0.00, real=0.04 secs]
28467 2021-05-17T22:47:04.5510000: 26996.9591 [Full GC (Allocation Failure) 2021-05-17T22:47:04.5510000: 26996.9591 [CMS:2021-05-17T22:47:04.5510000: 26996.9591 [CMS-concurrent-mark: 0.236(0.236) secs], (MemSpace= 792238->792238(1120208)), 0.5422651 secs] [Times: user=0.07 sys=0.00, real=0.07 secs]
28468 2021-05-17T22:47:04.5510000: 26996.9591 [Full GC (Allocation Failure) 2021-05-17T22:47:04.5510000: 26996.9591 [CMS:2021-05-17T22:47:04.5510000: 26996.9591 [CMS-concurrent-mark: 0.236(0.236) secs], (MemSpace= 792238->792238(1120208)), 0.5422651 secs] [Times: user=0.07 sys=0.00, real=0.07 secs]
28469 2021-05-17T22:47:04.5510000: 26996.9591 [Full GC (Allocation Failure) 2021-05-17T22:47:04.5510000: 26996.9591 [CMS:2021-05-17T22:47:04.5510000: 26996.9591 [CMS-concurrent-mark: 0.236(0.236) secs], (MemSpace= 792238->792238(1120208)), 0.5422651 secs] [Times: user=0.07 sys=0.00, real=0.07 secs]
28470 2021-05-17T22:47:04.5510000: 26996.9591 [Full GC (Allocation Failure) 2021-05-17T22:47:04.5510000: 26996.9591 [CMS:2021-05-17T22:47:04.5510000: 26996.9591 [CMS-concurrent-mark: 0.236(0.236) secs], (MemSpace= 792238->792238(1120208)), 0.5422651 secs] [Times: user=0.07 sys=0.00, real=0.07 secs]
```

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.

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

1. Test address connectivity.
2. If the network is unreachable, rectify the network connection first. Ensure that the network connection between the DLI queue and the external data source is normal.

2 Problems Related to SQL Jobs

2.1 Usage

2.1.1 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.

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.

2.2 Job Development

2.2.1 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.

1. Set the configuration item as follows:
spark.sql.shuffle.partitions = Number of partitions (number of the generated small files in this case)
2. Execute the following SQL statements:
INSERT OVERWRITE TABLE tablename
select * FROM tablename distribute by rand()

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

1. Create an OBS Table.
For details about the syntax, see [Creating an OBS Table Using the DataSource Syntax](#).
2. Add partitions.
For details about the 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](#).

2.2.3 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.

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'
```

Incorrect 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.

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

DLI allows you to associate JSON data in an OBS bucket to create tables in asynchronous mode.

The statement for creating the table is as follows:

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

2.2.5 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
```

2.2.6 How Do I Synchronize DLI Table Data from One Region to Another?

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" in *Data Lake Insight User Guide*.
2. Use the OBS cross-region replication function to replicate data to the OBS bucket in region 2. For details, see "Configuring Cross-Region Replication" in *Data Lake Insight User Guide*.

3. Import or use the corresponding data as required.

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

Currently, DLI does not allow you to insert table data into specific fields. To insert table data, you must insert data of all table fields at a time.

2.2.8 How Do I Delete Table Data?

Use TRUNCATE to clear data in DLI or OBS tables.

For details, see [Clearing Data](#).

2.3 Job O&M Errors

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

Create an OBS directory with a unique name. Alternatively, you can manually delete the existing OBS directory and submit the job again. However, exercise caution when deleting the existing OBS directory because the operation will delete all data in the directory.

2.3.2 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;
```

2.3.3 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?

Check whether the account is in arrears. If yes, recharge the account.

If the error persists, log out and log in again.

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

- Cause Analysis

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
```

2.3.5 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

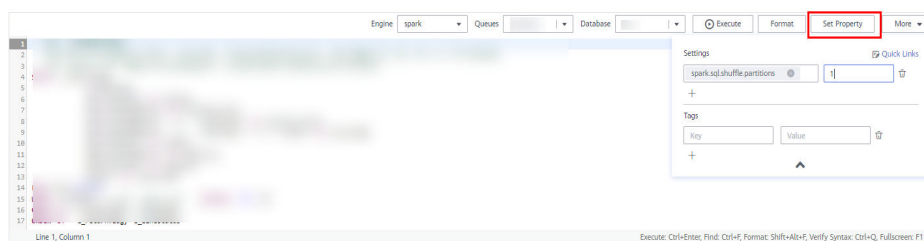
The data volume of the file to be imported is large and the value of **spark.sql.shuffle.partitions** is too large. As a result, the cache size is insufficient.

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 2-1 Modifying the parameter



3. Execute the job again.

2.3.6 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.

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

Currently, DLI supports the Hive syntax for creating tables of the TEXTFILE, SEQUENCEFILE, RCFILE, ORC, AVRO, and PARQUET file types. If the file format specified for creating a table in the CTAS is AVRO and digits are directly used as the input of the query statement (SELECT), for example, if the query is **CREATE TABLE tb_avro STORED AS AVRO AS SELECT 1**, a schema parsing exception is 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.

2.3.8 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.

2.3.9 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. Click **Buy Queue** to create a queue. Set **Type** to **For SQL**, set other parameters required, and click **Buy**.
 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.

2.3.10 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.

Solution

Generally, the file cannot be found due to a read/write conflict. Check whether a job is overwriting the data when the error occurs.

2.3.11 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

View the OBS bucket in the AccessControlException and check whether you are using an account that has the permission to access the bucket.

2.3.12 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

The current account does not have the permission to access the OBS bucket where the foreign table is located. Obtain the OBS permission and perform the query again.

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

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

Check whether your account is in arrears. If it is, renew your account and try again.

If the error persists after renewal, log out and log in again.

2.4 O&M Guide

2.4.1 How Do I Troubleshoot Slow SQL Jobs?

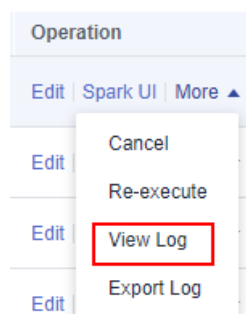
If the job runs slowly, perform the following steps to find the causes and rectify the fault:

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 2-2 Viewing logs



3. Obtain the folder of the archived logs in the OBS directory. The details are as follows:
 - Spark SQL jobs:


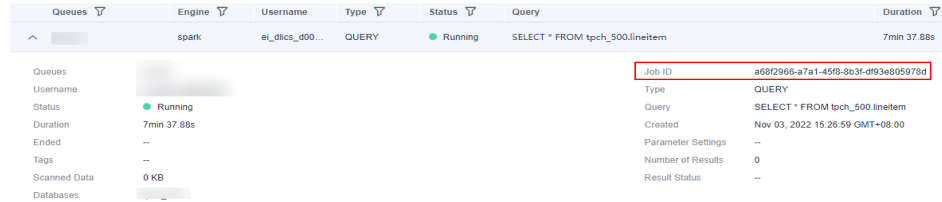
- On the **SQL Jobs** page, locate the row that contains the target job and click  to view the job details and obtain the job ID.

Figure 2-7 Obtaining the job ID

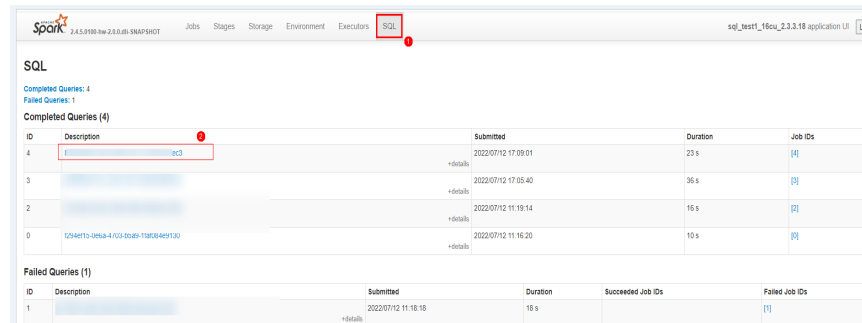


Queues	Engine	Username	Type	Status	Query	Duration
spark	spark	el_dlics_d00...	QUERY	Running	SELECT * FROM tpch_500.lineitem	7min 37.88s

Job ID	a682966-a7a1-45f8-8b3f-df93e805978d
Type	QUERY
Query	SELECT * FROM tpch_500.lineitem
Created	Nov 03, 2022 15:26:59 GMT+08:00
Parameter Settings	--
Number of Results	0
Result Status	--

- In the **Operation** column of the job, click **Spark UI**.
- 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 2-8 Clicking the job link



spark 2.4.5.0100-0aa-2.0.0.0b-08AP09KCT Jobs Stages Storage Environment Executors **SQL** sql_test1_16ou_2.3.3.18 application UI

SQL

Completed Queries: 4
Failed Queries: 1

Completed Queries (4)

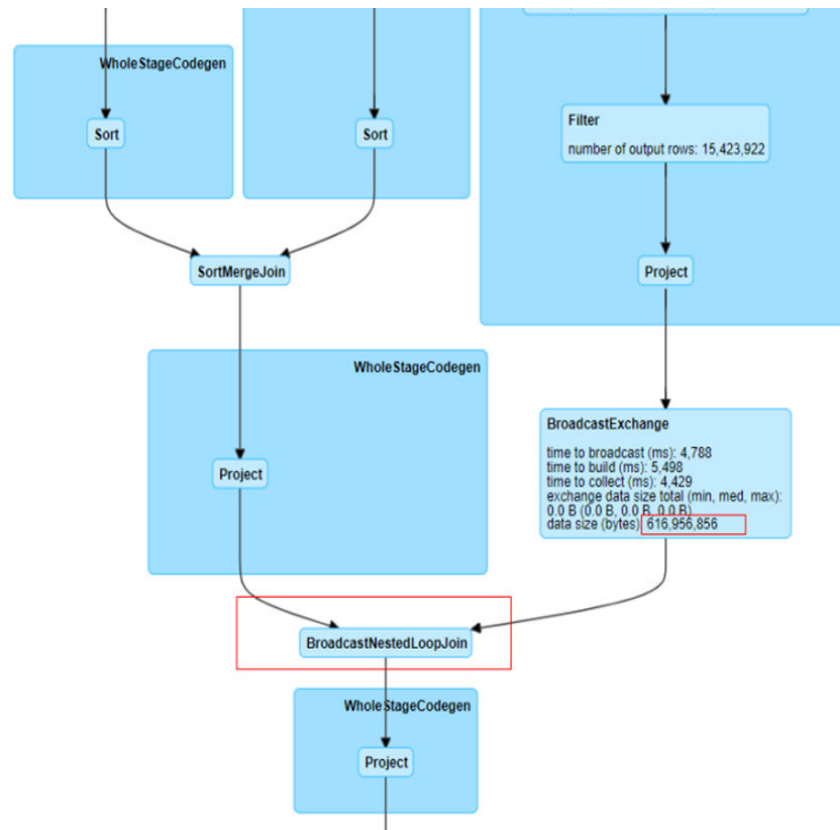
ID	Description	Submitted	Duration	Job IDs
4	[Link]	2022/07/12 17:09:01	23 s	[4]
3	[Link]	2022/07/12 17:05:40	36 s	[3]
2	[Link]	2022/07/12 11:19:14	16 s	[2]
0	[Link]	2022/07/12 11:16:20	10 s	[0]

Failed Queries (1)

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

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

Figure 2-9 DAG



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 2-10 Obtaining the job ID

Queues	Engine	Username	Type	Status	Query	Duration
	spark	el_dlcs_d00...	QUERY	Running	SELECT * FROM tpch_500.lineitem	7min 37.88s

Queues		Job ID	a682966-a7a1-45f8-8c3f-df93e805978d
Username		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			

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.

The screenshot shows the Spark Jobs interface. At the top, there are tabs for Jobs, Stages, Storage, Environment, Executors, and SQL. Below the navigation, there's a summary for Spark Jobs (7) with metrics: User: omm, Total Uptime: 6.5 h, Scheduling Mode: FAIR, Completed Jobs: 4, Failed Jobs: 1. There are two main sections: 'Completed Jobs (4)' and 'Failed Jobs (1)'. The 'Completed Jobs' table has columns: Job Id (Job Group), Description, Submitted, Duration, Stages: Succeeded/Total, and Tasks (for all stages): Succeeded/Total. The 'Failed Jobs' table has columns: Job Id (Job Group), Description, Submitted, Duration, Stages: Succeeded/Total, and Tasks (for all stages): Succeeded/Total. A red box highlights the 'Description' column in the failed job row, which contains a hyperlink.

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.

The screenshot shows the 'Active Stages (1)' table. It has columns: Stage Id, Pool Name, Description, Submitted, Duration, Tasks: Succeeded/Total, Input, Output, Shuffle Read, and Shuffle Write. A red box highlights the 'Description' column, which contains a hyperlink.

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 2-11**, when data skew occurs, the data volume of shuffle reads of a task is much greater than that of other tasks.

Figure 2-11 Data skew

The screenshot shows the 'Aggregated Metrics by Executor' table with 200 tasks. The table has columns: 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), and Shuffle Sp (Disk). A red box highlights the 'Shuffle Read Size / Records' column for the first task (Index 42), which shows a value of 52.9 GB / 47870629, significantly higher than other tasks.

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 2-12 shows the result. There are much more **num** parameters whose value are **1** than other values.

Figure 2-12 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 Eliminate Data Skew by Configuring AE Parameters?](#)

2.4.2 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 2-13 Viewing logs

Name	Type	Running Type	Running Dur...	Actual Start Time	Retry Count	Error Message	Versions	Operation
DLI SQL	DLI SQL	Run successfully	0:00	2022/Jan/30 15:13:57 GMT+08:00	0		1	View Log Manual Retry Succeeded More

Figure 2-14 Job ID

```
);
[2021/03/19 10:02:59 GMT+0800] [INFO] dli.sql.badRecordsPath=obs://dli-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 2-15 Viewing logs

Owner	Engine	Executed By	Job Type	Status	Statement	Duration	Created	Operation
default	spark		QUERY	Finished		14.77s	Jan 25, 2022 10:08:34 GMT+08:00	Modify Cancel Logout More
default	spark		QUERY	Finished		3.25s	Jan 25, 2022 17:53:27 GMT+08:00	Modify Cancel Logout More
default	spark		QUERY	Finished		1.81s	Jan 25, 2022 15:39:24 GMT+08:00	Modify Cancel Logout More
default	spark		QUERY	Finished		0.71s	Jan 26, 2022 15:58:12 GMT+08:00	Modify Cancel Logout More
default	spark		DDL	Finished		1.59s	Jan 25, 2022 15:58:05 GMT+08:00	Modify Cancel Logout More
default	spark		DDL	Finished		0.40s	Jan 25, 2022 15:56:19 GMT+08:00	Modify Cancel Logout More
default	spark	00178103	QUERY	Finished	SELECT * FROM dms_tspart LIMIT 1000	8.11s	Jan 24, 2022 15:55:07 GMT+08:00	Modify Cancel Logout More

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

2.4.3 How Do I View SQL Execution Records?

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.

2.4.4 How Do I Eliminate Data Skew by Configuring AE Parameters?

Scenario

If the execution of an SQL statement takes a long time, you need to access the Spark UI to check the execution status.

If data skew occurs, the running time of a stage exceeds 20 minutes and only one task is running.

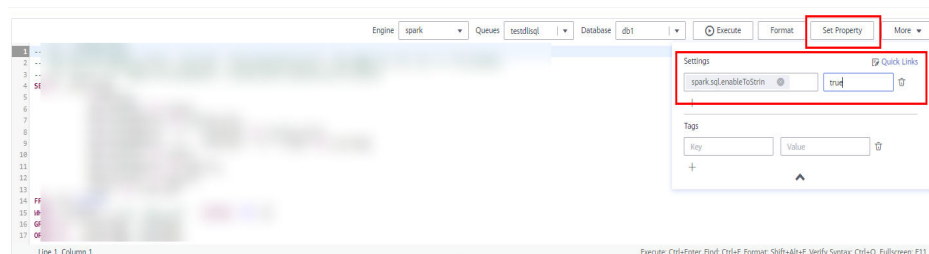
Figure 2-16 Data skew example

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

1 Pa

Procedure

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:



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.

2.4.5 What Can I Do If a Table Cannot Be Queried 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** from 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.

2.4.6 The Compression Ratio of OBS Tables Is Too High

A high compression ratio of OBS tables in the Parquet or ORC format (for example, a compression ratio of 5 or higher compared with text compression) will lead to large data volumes to be processed by a single task. In this case, you are advised to set **dli.sql.files.maxPartitionBytes** to **33554432** (default: **134217728**) in the **conf** field in the **submit-job** request body to reduce the data to be processed per task.

2.4.7 How Can I Avoid Garbled Characters Caused by Inconsistent Character Codes?

DLI supports only UTF-8-encoded texts. Ensure that data is encoded using UTF-8 during table creation and import.

2.4.8 Do I Need to Grant Table Permissions to a User and Project After I Delete a Table and Create One 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**.

2.4.9 Why Can't I Query Table Data After Data Is Imported to a DLI Partitioned Table Because the File to Be Imported Does Not Contain Data in the Partitioning Column?

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.

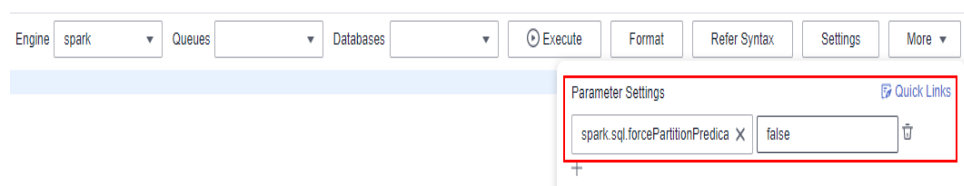
Possible Causes

When data is imported to a DLI partitioned table, 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 2-17 Parameters



3. Query the entire table or the partitioned table.

2.4.10 How Do I Fix the Data Error Caused by CRLF Characters in a Field of the OBS File Used to Create an External OBS Table?

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 as follows:

```
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, the 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);
```

2.4.11 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.

2.4.12 The on Clause Is Not Added When Tables Are Joined. Cartesian Product Query Causes High Resource Usage of the Queue, and the Job Fails to Be Executed

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)
```

2.4.13 Why Can't I Query Data After I Manually Add Data 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.

2.4.14 Why Is All Data Overwritten When insert overwrite Is Used to Overwrite Partitioned Table?

To dynamically overwrite the specified partitioned data in the DataSource table, set `dli.sql.dynamicPartitionOverwrite.enabled` to `true` and then run the insert overwrite statement. (The default value of `dli.sql.dynamicPartitionOverwrite.enabled` is `false`.)

For details, see [Inserting Data](#).

2.4.15 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.

2.4.16 Why Is the create_date Field in the RDS Table Is a Timestamp in the DLI query result?

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](#).

2.4.17 What Can I Do If datasize Cannot Be Changed After the Table Name Is Changed in a Finished SQL Job?

If the table name is changed immediately after SQL statements are executed, the data size of the table may be incorrect.

If you need to change the table name, change it 5 minutes after the SQL job is complete.

2.4.18 Why Is the Data Volume Changes When Data Is Imported 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.

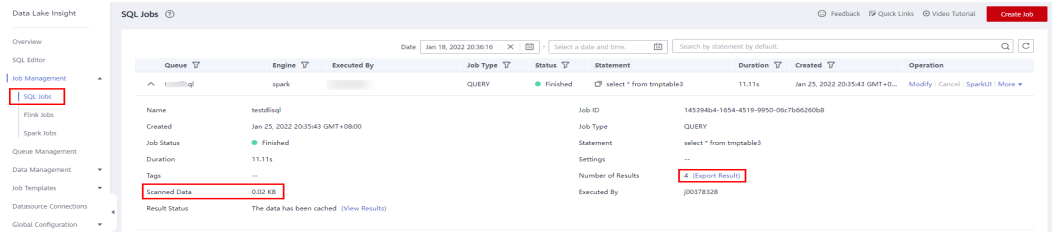
Run a query statement to check whether the amount of imported data is correct.

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 2-18 Checking the amount of data



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*.

3 Problems Related to Spark Jobs

3.1 Usage

3.1.1 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.

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.

3.2 Job Development

3.2.1 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)
```

3.2.2 How Do I Set the AK/SK for a Queue to Operate an OBS Table?

The temporary AK/SK is recommended. For details, see [Obtaining a Temporary Access Key and Security Token](#) in *Identity and Access Management API Reference*.

- If the AK and SK are obtained, set the parameters as follows:

- Create 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 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()
```
- If **ak**, **sk**, and **securitytoken** are obtained, the temporary AK/SK and security token must be used at the same time during authentication. The setting is as follows:
 - Create 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 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()
```

NOTE

For security purposes, you are advised not to include the AK and SK information in the OBS path. In addition, if a table is created in the OBS directory, the OBS path specified by the **Path** field cannot contain the AK and SK information.

3.2.3 How Do I View the Resource Usage of DLI Spark Jobs?

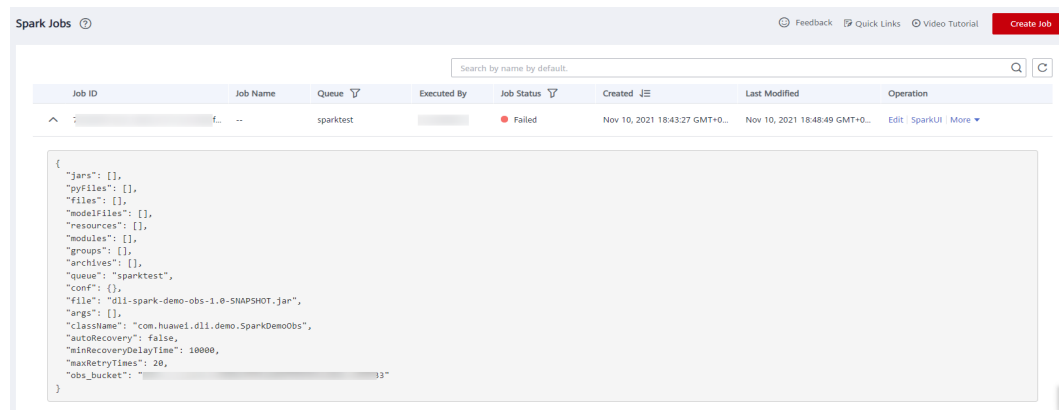
Viewing the Configuration of a Spark Job

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.

Figure 3-1 Viewing the configuration of a Spark job

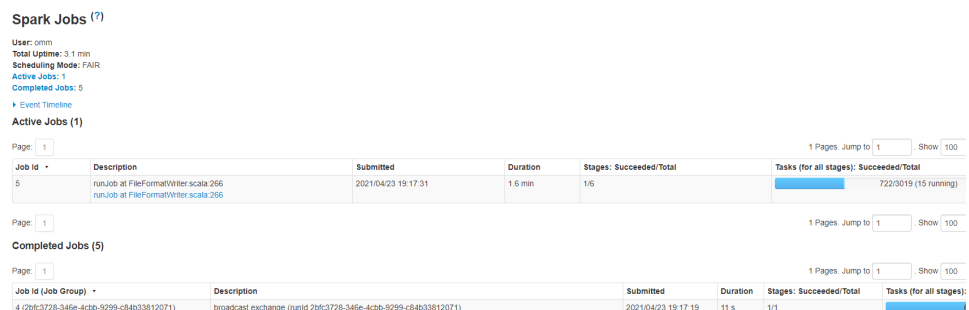


Viewing Real-Time Resource Usage of a Spark Job

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 3-2 SparkUI



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 3-3 Driver information

spark.driver.cores	2
spark.driver.extraJavaOptions	-XX:CIcC XX:+ExitT XX:+Use Dlog4j.co Dcarbon. Dsccon
spark.driver.extraLibraryPath	Bigdata/c
spark.driver.host	spark-4c
spark.driver.memory	7G
spark.driver.port	7078
spark.driver.userClassPathFirst	false
spark.dynamicAllocation.cachedExecutorIdleTimeout	600s

Figure 3-4 Executor information

spark.executor.cores	2
spark.executor.extraClassPath	hadoop/c
spark.executor.extraJavaOptions	-XX:CI Co XX:+ExitC XX:+User Dlog4j.co Dcarbon.l Dsccon
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

3.2.4 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 details about how to create a datasource connection on the management console, see "Enhanced Datasource Connections" in *Data Lake Insight User Guide*.

For details about how to call an API to create a datasource connection, see "Creating an Enhanced Datasource Connection" in *Data Lake Insight API Reference*.

3.2.5 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, the Python library is installed on the execution machine based on pip. For serverless services such as DLI, you do not need to and are unaware of the underlying compute resources. In this case, how does DLI ensure that you run their programs perfectly?

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 3-5 Spark job editor page

The screenshot shows the Spark job editor interface. The 'Python File Dependencies (--py-files)' section is highlighted with a red box. The interface includes the following sections and fields:

- Select a Queue:** A dropdown menu labeled 'Select a queue.'
- Job Configurations:**
 - Job Name(--name):** A text input field labeled 'Enter a job name.'
 - Application:** A text input field with a folder icon on the right.
 - Main Class(--class):** A text input field with a warning message: 'This parameter cannot be left blank if the Application is a JAR package.'
 - Application Parameters:** A text area with the instruction: 'Specify each parameter on a separate line.'
 - Spark Arguments(--conf):** A text area with the instruction: 'Enter arguments using the key = value format. Press Enter to separate multiple key-value pairs.'
- JAR Package Dependencies (--jars):** A dropdown menu with the option 'Use the DLI program package for production services' and a text area for 'OBS path used for development and debugging. Specify each parameter on a separate line'.
- Python File Dependencies (--py-files):** A dropdown menu with the option 'Use the DLI program package for production services' and a text area for 'OBS path used for development and debugging. Specify each parameter on a separate line'.
- Other Dependencies (--files):** A dropdown menu with the option 'Use the DLI program package for production services' and a text area for 'OBS path used for development and debugging. Specify each parameter on a separate line'.
- Group Name:** A dropdown menu.
- Access Metadata:** Two buttons labeled 'Yes' and '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 3-6 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.

3.2.6 How Does a Spark Job Access a MySQL Database?

You can use DLI Spark jobs to access data in the MySQL database using either of the following methods:

- Solution 1: Buy a yearly/monthly or pay-per-use dedicated queue, create an enhanced datasource connection, and read data from the MySQL database through a datasource table. You need to write Java or Scala code to implement this solution.
- Solution 2: Use CDM to import data from the MySQL database to an OBS bucket, and then use a Spark job to read data from the OBS bucket. If you already have a CDM cluster, this solution is simpler than solution 1 and does not involve any other database.

3.2.7 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.createStatement();
st.execute("set spark.sql.shuffle.partitions=20");
```

3.2.8 How Do I Read Uploaded Files for a Spark Jar Job?

You can use `SparkFiles` to read the file submitted using `--file` from 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 3-7 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()
  }
}
```

3.3 Job O&M Errors

3.3.1 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")
```

3.3.2 Why Are Errors "ResponseCode: 403" and "ResponseStatus: Forbidden" Reported 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 the AK/SK for a Queue to Operate an OBS Table?](#)

3.3.3 Why Is Error "verifyBucketExists on XXXX: status [403]" Reported When I Use a Spark Job to Access an OBS Bucket That I Have Access Permission?

Check whether the OBS bucket is used to store DLI logs on the **Global Configuration > Job Configurations** page. The job log bucket cannot be used for other purpose.

3.3.4 Why Is a Job Running Timeout Reported When a Spark Job Runs a Large Amount of Data?

When a Spark job accesses a large amount of data, for example, accessing data in a GaussDB(DWS) database, you are advised to set the number of concurrent tasks and enable multi-task processing.

3.3.5 Why Does the Job Fail to Be Executed and the Log Shows that the File Directory Is Abnormal When I Use a Spark Job to Access 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.

3.3.6 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**.

3.3.7 Why Can't I Find the global_temp Database in the Spark 3.x Job Log?

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.

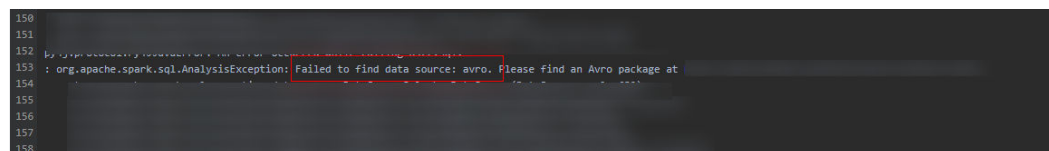
If the Spark 3.x job log contains a record of accessing catalogs to query the database and a message indicating that the non-existence of the database is to ensure that Spark jobs can run properly, you do not need to perform any operations.

3.3.8 Why Does the DataSource Syntax Fail to Create an OBS Table in Avro Format When Spark 2.3.x Is Used?

Symptom

I failed to use the DataSource syntax to create an OBS table in Avro format when selecting Spark to access metadata.

Figure 3-8 Failed to create an OBS table in Avro format



```
150
151
152
153 : org.apache.spark.sql.AnalysisException: Failed to find data source: avro. Please find an Avro package at
154
155
156
157
158
```

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.

3.4 O&M Guide

3.4.1 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.

3.4.2 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.

4 Product Consultation

4.1 Usage

4.1.1 What Is DLI?

Data Lake Insight (DLI) is a serverless data processing and analysis service fully compatible with Apache Spark, Trino, and Apache Flink ecosystems. It frees you from managing any server. DLI supports standard SQL and is compatible with Spark and Flink SQL. It also supports multiple access modes, and is compatible with mainstream data formats. DLI supports SQL statements and Spark applications for heterogeneous data sources, including CloudTable, RDS, GaussDB(DWS), CSS, OBS, custom databases on ECSs, and offline databases.

4.1.2 What Are the Application Scenarios of DLI?

DLI is applicable to large-scale log analysis, federated analysis of heterogeneous data sources, and big data ETL processing.

Large-scale Log Analysis

- Game operation data analysis

Different departments of a game company analyze daily new logs via the game data analysis platform to obtain required metrics and make decision based on the obtained metric data. For example, the operation department obtains required metric data, such as new players, active players, retention rate, churn rate, and payment rate, through the platform to learn the current game status and determine follow-up actions. The placement department obtains the channel sources of new players and active players through the platform to determine the platforms for placement in the next cycle.

- Advantages

- Efficient Spark programming model: DLI uses Spark Streaming to directly ingest data from DIS and perform preprocessing such as data cleaning. You only need to edit the processing logic, without the need to pay attention to the multi-thread model.

- Easy to use: You can use standard SQL statements to compile metric analysis logic without paying attention to the complex distributed computing platform.
- Pay-per-use: Log analysis is scheduled periodically based on the time requirements. There is a long idle period between each two scheduling operations. DLI adopts the pay-per-use billing mode, which saves the cost by more than 50% compared with the exclusive queue mode.
- It is recommended that you use the following related services:
OBS, DIS, GaussDB(DWS), and RDS

Federated Analysis of Heterogeneous Data Sources

- Digital service transformation of car companies
In the face of new competition pressures and changes in travel services, car companies build the IoV cloud platform and IVI OS to streamline Internet applications and vehicle use scenarios, completing digital service transformation for car companies. This delivers better travel experience for vehicle owners, increases the competitiveness of car companies, and promotes sales growth. For example, DLI can be used to collect and analyze daily vehicle metric data (such as batteries, engines, tire pressure, and airbags), and give feedback on maintenance suggestions to vehicle owners in time.
- Advantages
 - No need to migrate data for cross-source data analysis: RDS stores the basic information about vehicles and vehicle owners, CloudTable stores real-time vehicle location and health status information, and GaussDB(DWS) stores periodic metric statistics. DLI allows federated analysis on data from multiple sources without data migration.
 - Tiered data storage: Car companies need to retain all historical data to support auditing and other services that requiring infrequent data access. Warm and cold data is stored in OBS and frequently accessed data is stored in CloudTable and GaussDB(DWS), reducing the overall storage cost.
 - Rapid and agile alarm triggering: There are no special requirements for the CPU, memory, hard disk space, and bandwidth.
- It is recommended that you use the following related services:
DIS, CDM, OBS, GaussDB(DWS), RDS, and CloudTable

Big Data ETL Processing

- Carrier big data analysis
Carriers typically require petabytes, or even exabytes of data storage, for both structured (base station details) and unstructured (messages and communications) data. They need to be able to access the data with extremely low data latency. Extracting value from this data efficiently is a major challenge. DLI provides multi-mode engines such as batch processing and stream processing to break down data silos and perform unified data analysis.
- Advantages

- Big Data ETL: You can enjoy TB to EB-level data governance capabilities to quickly perform ETL processing on massive carrier data. Distributed datasets are provided for batch processing.
- High Throughput, Low Latency: DLI uses the Dataflow model of Apache Flink, a real-time computing framework. High-performance computing resources are provided to consume data from your created Kafka, DMS Kafka, and MRS Kafka clusters. A single CU processes 1,000 to 20,000 messages per second.
- Fine-grained Permissions Management: Your company may have numerous departments, where data needs to be shared and isolated. Using DLI, you can apply for resource queues by tenant to isolate computing resources (CPUs and memory), ensuring job SLA. DLI supports table- or column-level data permission control, allowing for secure access for different departments.
- It is recommended that you use the following related services:
OBS, DIS, and DataArts Studio

Geographic Big Data Analysis

- Geographic big data analysis
Geographic big data has big data characteristics. It features large data volume (for example, PB-scale global satellite remote sensing image data is generated) and numerous data varieties (for example, structured remote sensing image raster data, vector data, unstructured spatial location data, and 3D modeling data). Users focus on how to use efficient mining tools or mining methods to get insights from the large volume of geographic big data.
- Advantages
 - Spatial Data Analysis Operators: With full-stack Spark capabilities and rich Spark spatial data analysis Spatial Data Analysis Operators With full-stack Spark capabilities and rich Spark spatial data analysis algorithm operators, DLI delivers comprehensive support for real-time processing of dynamic streaming data with location attributes and offline batch processing. DLI can handle massive data, including structured remote sensing image data, unstructured 3D modeling, and laser point cloud data.
 - CEP SQL: DLI delivers geographical location analysis functions to analyze geospatial data in real time. You can fulfill yaw detection and geo-fencing through SQL statements.
 - Big Data Processing: DLI allows you to quickly migrate remote sensing image data at the TB to EB scale to the cloud and perform image data slicing to offer resilient distributed datasets (RDDs) for distributed batch computing.
- It is recommended that you use the following related services:
DIS, CDM, DES, OBS, RDS, and CloudTable

4.1.3 Which Data Formats Does DLI Support?

DLI supports the following data formats:

- Parquet

- CSV
- ORC
- Json
- Avro

4.1.4 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 4-1 Function comparison

Category	Features	DLI Flink	MRS Flink
Feature d Capabilities	Deployment	Full hosting	Semi-hosting (cluster O&M required)
	Elastic scaling	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none"> Open-source connectors and out-of-the-box connectors for data sources including databases (RDS and GaussDB), message queues (DMS), data warehouses (GaussDB 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	<ul style="list-style-type: none"> 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.

Category	Features	DLI Flink	MRS Flink
	Usability	<p>Out-of-the-box, serverless architecture, and cross-AZ disaster recovery</p> <ul style="list-style-type: none"> • 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. 	<p>Technical capabilities are required for coding, cluster setup, configuration, and O&M.</p> <ul style="list-style-type: none"> • 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
Enterprise 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

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

The Spark component of DLI is a fully managed service. You can only use the DLI Spark through its APIs. .

The Spark component of MRS is built on the VM in an MRS cluster. You can develop the Spark component to optimize it as needed and make API calls to use it.

4.1.6 Where Can DLI Data Be Stored?

DLI data can be stored in either of the following:

- 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.

4.1.7 What Are the Differences Between DLI Tables and OBS Tables?

- DLI tables store data within the DLI service, and you do not need to know the data storage path.
- OBS tables store data in your OBS buckets, and you need to manage the source data files.
- Different from OBS tables, DLI tables give you more permission control and cache acceleration functions. The performance of DLI tables is better than that of foreign tables, but you will be charged for the storage.

4.1.8 How Can I Use DLI If Data Is Not Uploaded to OBS?

Currently, DLI supports analysis only on the data uploaded to the cloud. In scenarios where regular (for example, on a per day basis) one-off analysis on incremental data is conducted for business, you can do as follows: Anonymize data to be analyzed and store anonymized data on OBS temporarily. After analysis is complete, export the analysis report and delete the data temporarily stored on OBS.

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

Data in the OBS bucket shared by IAM users under the same account can be imported. You cannot import data in the OBS bucket shared with other IAM account.

4.1.10 Why Is Error "Failed to create the database. {"error_code":"DLI.1028";"error_msg":"Already reached the maximum quota of databases:XXX"." Reported?

Viewing My Quotas


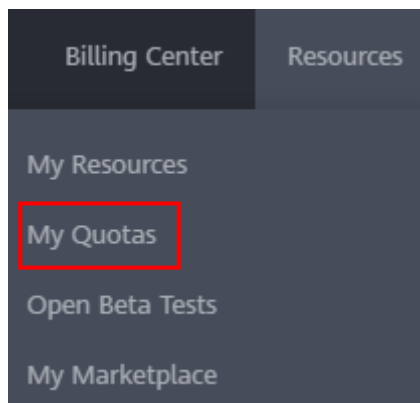
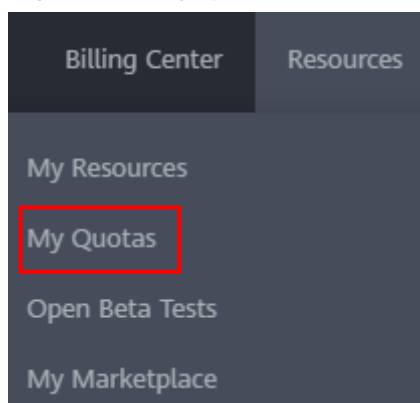
1. Log in to the management console.
2. Click  in the upper left corner and select **Region** and **Project**.
3. In the upper right corner of the page, choose **Resources > My Quotas**. The **Service Quota** page is displayed.

Figure 4-1 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.

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 4-2 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**.

4.1.11 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.

4.2 Job Management

4.2.1 How Do I Manage Tens of Thousands of Jobs Running on DLI?

If you are suggested to perform following operations to run a large number of DLI jobs:

- Group the DLI jobs by type, and run each group on a queue.
- Alternatively, create IAM users to execute different types of jobs. .

4.2.2 How Do I Change the Name of a Field in a Created Table?

The field names of tables that have been created cannot be changed.

You can create a table, define new table fields, and migrate data from the old table to the new one.

4.3 Privacy and Security

4.3.1 Does DLI Have 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.

5 Billing

5.1 What Is the Billing Mode of DLI?

DLI queues support the following billing modes. You can create queues on different billing modes as you needed.

- **Pay-per-use queues:**

Select this billing mode if you seldom use a queue. You will be billed by calendar hour.

Compute resources will be released one hour after the queue is idle (no job is running). When the queue is used again, it takes about 5 to 10 minutes for the system to reallocate compute resources to the queue.

- **Yearly/Monthly queues:**

If you use the queue frequently, you are advised to purchase a yearly/monthly queue. The compute resources are dedicated and will not be released when the queue is idle (no job is running). The usage experience is better and the cost is lower than that of the pay-per-use queue.

- **Pay-per-use queues with dedicated resources:**

If you need dedicated resources for a short period of time, you can select the dedicated resource mode when purchasing a pay-per-use queue. In this mode, resources in idle will not be released (the billing period is equal to the life cycle of the queue).

- **Queues billed based on scanned data volume:**

This is valid available for the default queue only.

For details about how to create a queue, see [Creating a Queue](#).

For details about the billing modes, see [Billing Description](#) in *Data Lake Insight Service Overview*.

For pricing details, see [Data Lake Insight Pricing Details](#).

5.2 Can I Change a Yearly/Monthly Queue to a Pay-per-Use Queue?

The billing mode of a queue cannot be changed. You can buy a pay-per-use queue.

- To buy a queue, perform the following steps:
[Create a Queue](#).

5.3 Can I Change a Pay-Per-Use Queue to a Yearly/Monthly Queue?

Currently, you are not allowed to change the pay-per-use queue to a yearly/monthly queue.

5.4 When Is a Data Lake Queue Idle?

An idle queue means that there are no SQL job, Flink job, or Spark job running on the queue.

If no job is running within a natural hour, the queue in that natural hour is idle. You will not be charged during idle hours for a pay-per-use queue.

The computing resources of a pay-per-use queue are released after the queue is idle for one hour. When the queue is used again, the computing resources need to be reallocated, which may take 5 to 10 minutes.

In pay-per-use mode, fees are deducted by hour. For example, 58 minutes of usage will be rounded to the hour.

For example, if a queue is used during 12:10 to 12:30 or 12:10 to 12:55, the user is charged for one hour.

For details about the billing details, visit [Billing](#).

5.5 How Do I Check the Billing?

If you feel that the billing is incorrect when you use DLI, perform the following steps:

NOTE

The following operations are performed on the DLI management console. For details, see [Job Management](#) in *Data Lake Insight User Guide*.

- SQL Job
 - a. Log in to the DLI management console.
 - b. Choose **Job Management** > **SQL Jobs**.
 - c. View the job details to be confirmed and check whether the following operations are performed within the fee deduction period:

- i. Use the self-created queue.
 - ii. Executed SQL jobs.
 - If you used queues created by yourself to execute jobs, you are billed based on CUH by default.
 - If you used the default queue to execute jobs, you are billed based on the amount of data scanned.
 - If no SQL job is executed, continue to check the Spark jobs.
- Spark Job
 - a. Choose **Job Management > Spark Jobs**.
 - b. Check whether the job is completed. If not, the job is billed by CUH. If no Spark job is executed, continue to check the Flink jobs.
- Flink Job
 - a. Choose **Job Management > Flink Jobs**.
 - b. Check whether the job is completed. If not, the job is billed by CUH. If no job is executed within the fee deduction period, check whether the billing is caused by the storage of a large amount of data.
- Data Storage
 - a. Choose **Data Management > Databases and Tables**.
 - b. Check whether the created database contains data stored in DLI.
 - i. Click the name of the database to be viewed. The **Table Management** page is displayed.
 - ii. Check whether the **Data Location** of the corresponding table is DLI. If you have data stored in DLI, you are billed based on the amount of data stored.

If no problem is found after checking the preceding items, you can submit a service ticket. Service support personnel will help you check the problem.

5.6 What Is the Difference Between the Following Two Payment Modes: One Is to Purchase 4,000-CU Resources for Three Months at a Time, the Other Is to Purchase 4,000-CU Resources for One Month for Three Times?

If you purchase a queue of 4,000 CUs for three months at a time, the validity period of the queue is three months.

If you purchase queues for three times and each time purchase a queue with 4,000 CUs for one month, three queues with 4,000 CUs are purchased. When the one-month validity period expires, the three queues expire.

5.7 How Do I Purchase 30-CU Resources?

You can purchase two 16-CU queues. They are equivalent to a 30-CU queue.

For details about how to purchase a queue, see [Creating a Queue](#) in *Data Lake Insight User Guide*.

5.8 How Will I Be Billed if Only Two CUs Are Used to Run a Flink Job on a Queue of 16 CUs?

Although only two CUs are required for running the Flink job, the actual fee is calculated based on the queue of 16 CUs. The following formula shows how you will be billed (assuming that the unit price of a Flink job is ¥0.35/CU):

Compute resource fee = Number of CUs of the queue x Usage duration (hours) x Unit price

The fee depends on the CU size of the queue running your Flink job, instead of the number of CUs actually required by the job. In this case, the billing is 16 x ¥0.35/CU hour.

The unit of the usage duration is hour. If the duration is less than one hour, it is rounded to one hour.

5.9 Do I Need to Pay Extra Fees for Purchasing a Queue Billed Based on the Scanned Data Volume?

Billing based on the volume of scanned data is applicable to the default queue. When the default queue is used, you are billed by the volume of scanned data.

You can select pay-per-use (by CUH) or yearly/monthly billing for a queue that you purchase.

For details about the billing details, visit [Billing](#).

6 Quota

6.1 What Is User Quota?

Huawei Cloud restricts the number and capacity of user resources. If the existing resource quota cannot meet your service requirements, you can submit a service ticket to increase your quota. Once your application is approved, we will update your resource quota accordingly and send you a notification. For details about the quotas, see [Quotas](#).

6.2 How Do I View My Quotas?


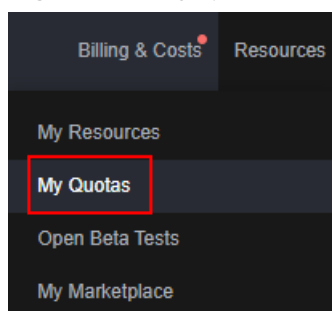
1. Log in to the management console.
2. Click  in the upper left corner and select **Region** and **Project**.
3. In the navigation pane on the top, choose **Resources** > **My Quotas**.
The **Service Quota** page is displayed.

Figure 6-1 My quotas



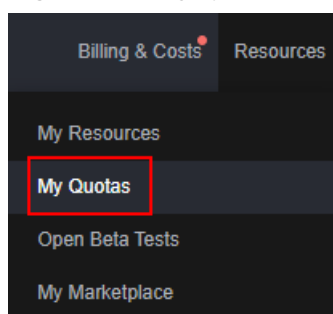
4. View the used and total quota of each type of resources on the displayed page.
If a quota cannot meet service requirements, increase a quota.

6.3 How Do I Increase a 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 6-2 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**.

7 Permission

7.1 Usage

7.1.1 How Do I Manage Fine-Grained DLI Permissions?

DLI has a comprehensive permission control mechanism and supports fine-grained authentication through Identity and Access Management (IAM). You can create policies in IAM to manage DLI permissions.

With IAM, you can use your Huawei Cloud account to create IAM users for your employees, and assign permissions to the users to control their access to specific resource types. For example, some software developers in your enterprise need to use DLI resources but must not delete them or perform any high-risk operations. To achieve this result, you can create IAM users for the software developers and grant them only the permissions required for using DLI resources.

NOTE

For a new user, you need to log in for the system to record the metadata before using DLI.

IAM can be used free of charge. You pay only for the resources in your account.

If the Huawei Cloud account has met your requirements, you do not need to create an independent IAM user for permission management. Then you can skip this section. This will not affect other functions of DLI.

DLI System Permissions

[Table 7-1](#) lists all the system-defined roles and policies supported by DLI.

You can grant users permissions by using roles and policies.

- **Roles:** A type of coarse-grained authorization that defines permissions related to user responsibilities. This mechanism provides only a limited number of service-level roles for authorization. When using roles to grant permissions, you need to also assign other roles on which the permissions depend to take effect. Roles are not an ideal choice for fine-grained authorization and secure access control.

- **Policies:** A type of fine-grained authorization that defines permissions required to perform operations on specific cloud resources under certain conditions. This type of authorization is more flexible and ideal for secure access control. For example, you can grant DLI users only the permissions for managing a certain type of cloud servers.

Table 7-1 DLI system permissions

Role/Policy Name	Description	Category
DLI FullAccess	Full permissions for DLI.	System-defined policy
DLI ReadOnlyAccess	Read-only permissions for DLI. With read-only permissions, you can use DLI resources and perform operations that do not require fine-grained permissions. For example, create global variables, create packages and package groups, submit jobs to the default queue, create tables in the default database, create datasource connections, and delete datasource connections.	System-defined policy
Tenant Administrator	Tenant administrator <ul style="list-style-type: none">• Administer permissions for managing and accessing all cloud services. After a database or a queue is created, the user can use the ACL to assign rights to other users.• Scope: project-level service	System-defined role
DLI Service Admin	DLI administrator <ul style="list-style-type: none">• Administer permissions for managing and accessing the queues and data of DLI. After a database or a queue is created, the user can use the ACL to assign rights to other users.• Scope: project-level service	System-defined role

You can create custom policies to supplement system-defined policies and implement more refined access control. For details about how to create a custom policy, see [Creating a Custom Policy](#).

7.1.2 What Is Column Permission Granting of a DLI Partition Table?

You cannot perform permission-related operations on the partition column of a partitioned table.

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.

7.2 O&M Guide

7.2.1 Why Does My Account Have Insufficient Permissions Due to Arrears?

When you submit a job, a message is displayed indicating that the job fails to be submitted due to insufficient permission caused by arrears. In this case, you need to check the roles in your token:

- **op_restrict**: The account permission is restricted due to insufficient balance. If your account balance is insufficient, the tokens of all online users under this account are revoked. If a user logs in to the system again, the **op_restrict** permission is added to the obtained token.
- **op_suspended**: Your account is suspended due to arrears or other reasons. If your account is in arrears, the tokens of all online users under this account are revoked. If a user logs in to the system again, the **op_suspended** permission is added to the obtained token, and user operations (excluding cloud service users) will be restricted.

If the two roles described about are in your token, user operations are restricted.

7.2.2 Why Does the System Display a Message Indicating Insufficient Permissions When I Update a Program Package?

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**.

7.2.3 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.

7.2.4 Why Can't I Query Table Data After I've Been Granted Table Permissions?

The table permission has been granted and verified. However, after a period of time, an error is reported indicating that the table query fails.

There are two possible reasons:

- The granted permission has been canceled.
- View the table creation time to check whether the table is deleted or re-created by others. If it has been deleted or re-created, the permission becomes invalid.

7.2.5 Will an Error Be Reported if the Inherited Permissions Are Regranted to a Table That Inherits Database Permissions?

If a table inherits database permissions, you do not need to regrant the inherited permissions to the table.

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.

7.2.6 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.

8 Queue

8.1 Usage

8.1.1 Does the Description of a DLI Queue Can Be Modified?

Currently, you are not allowed to modify the description of a created queue. You can add the description when purchasing the queue.

8.1.2 Will Table Data in My Database Be Lost If I Delete a Queue?

Deleting a queue does not cause table data loss in your database.

8.1.3 How Does DLI Ensure the Reliability of Spark Jobs When a Queue Is Abnormal?

You need to develop a mechanism to retry failed jobs. When a faulty queue is recovered, your application tries to submit the failed jobs to the queue again.

8.1.4 How Do I Monitor Queue Exceptions?

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. .

8.2 O&M Guide

8.2.1 How Do I View DLI Queue Load?

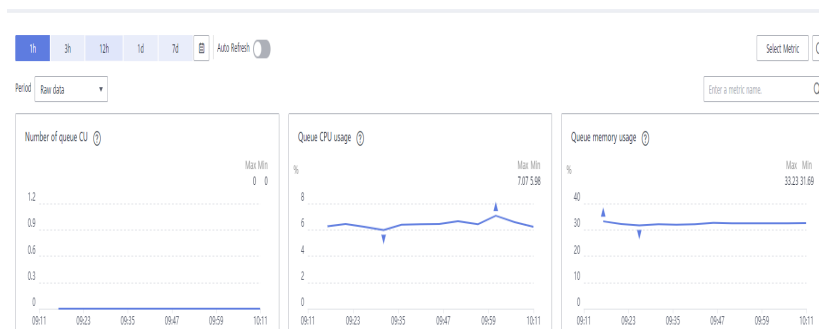
Scenario

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.
2. In the navigation pane on the left, choose **Cloud Service Monitoring** > **Data Lake Insight**.
3. Select the queue you want to view.

Figure 8-1 Viewing the queue load



8.2.2 How Do I Determine Whether There Are Too Many Jobs in the Current Queue?

Description

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

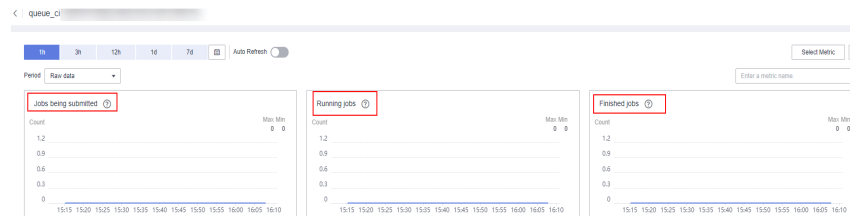
Solution

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 8-2 Viewing monitoring metrics of a queue



8.2.3 How Do I Switch an Earlier-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.

8.2.4 Why Cannot I View the Resource Running Status of DLI Queues on Cloud Eye?

DLI queues do not use resources or bandwidth when no job is running. In this case, the running status of DLI queues is not displayed on CES.

8.2.5 How Do I Allocate Queue Resources for Running Spark Jobs If I Have Purchased 64 CUs?

In DLI, 64 CU = 64 cores and 256 GB memory.

In a Spark job, if the driver occupies 4 cores and 16 GB memory, the executor can occupy 60 cores and 240 GB memory.

8.2.6 Why Is Error "Queue plans create failed. The plan xxx target cu is out of quota" Reported When I Schedule CU Changes?

Scenario

Queue plans create failed. The plan xxx target cu is out of quota is displayed when you create a scheduled scaling task.

Solution

The CU quota of the current account is insufficient. You need to apply for more quotas.

8.2.7 Why Is a Timeout Exception Reported When a DLI SQL Statement Fails to Be Executed 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. The procedure is as follows:

1. In the navigation pane of the DLI management console, choose **Queue Management**.
2. In the upper right corner of the **Queue Management** page, click **Buy Queue** to create a queue.
3. On the **Buy Queue** page, set the required parameters as you need. Especially, set **Type** to **For SQL**.
4. Submit your SQL jobs to the newly created queue.

9 Datasource Connections

9.1 Datasource Connections

9.1.1 Why Do I Need to Create a VPC Peering Connection for an Enhanced Datasource Connection?

You need to create a VPC peering connection to enable network connectivity. Take MRS as an example. If DLI and MRS clusters are in the same VPC, and the security group is enabled, you do not need a VPC peering connection for communications between MRS and DLI.

9.1.2 Failed to Bind a Queue to an Enhanced Datasource Connection

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."}
```

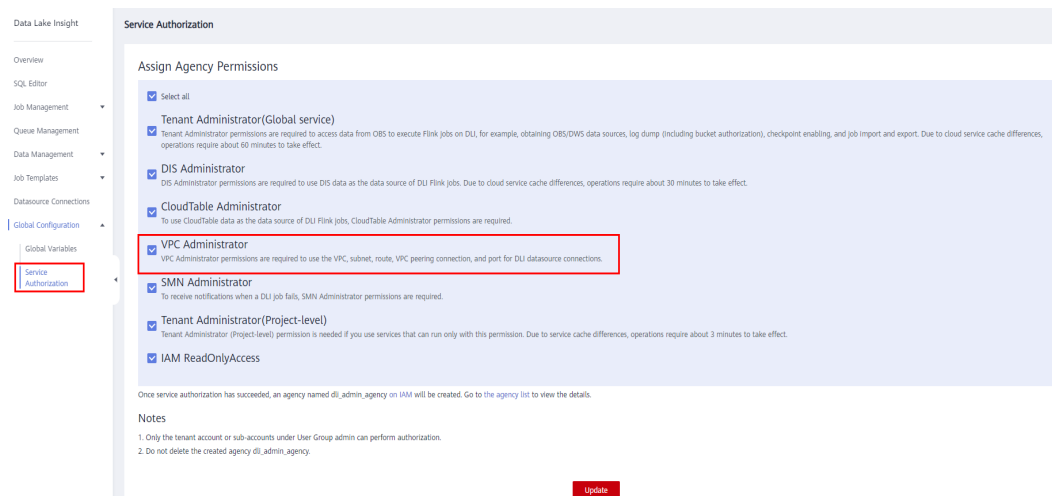
Cause Analysis

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

On the DLI console, choose **Global Configuration** > **Service Authorization**, select the required VPC permission, and click **Update**.

Figure 9-1 Selecting VPC administrator

9.1.3 DLI Failed to Connect 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: PSQLError: 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

9.1.4 How Do I Do if the Datasource Connection Is Created But the Network Connectivity Test Fails?

Description

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 That of the Data Source](#)
- [Check Whether the VPC Administrator 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](#)

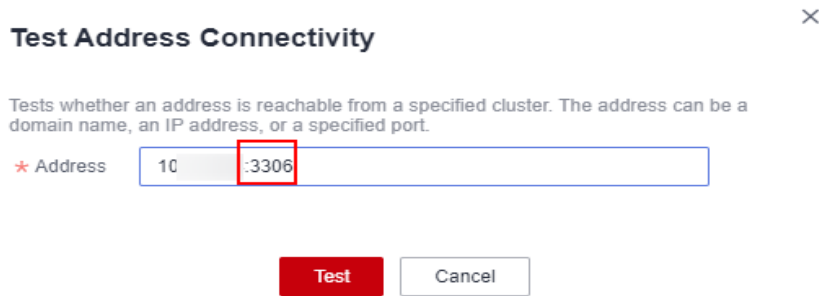
Check Whether a Port Number Is Added to the End of the Domain Name or IP Address

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 9-2 Testing address connectivity

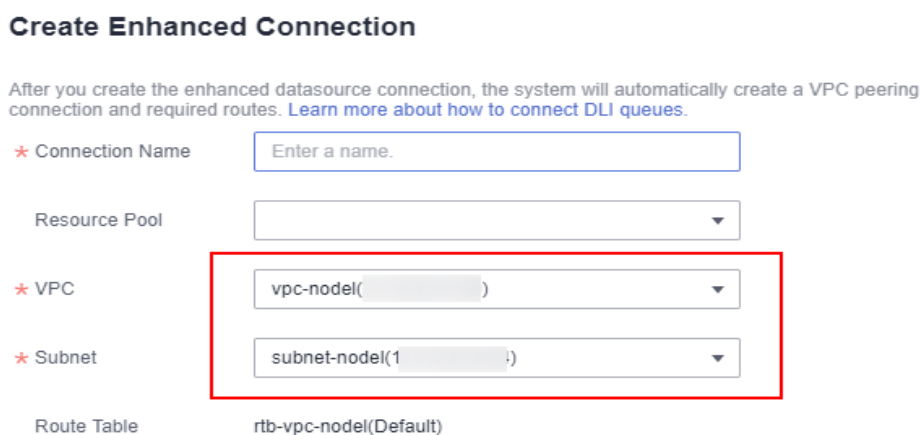


Check Whether the Information of the Peer VPC and Subnet Are Correct.

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 9-3 Creating a connection



Check Whether the CIDR Block of the Queue Overlaps 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 9-4 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 9-4 Viewing connection logs



Check Whether the VPC Administrator Permission Is Granted to DLI

View the connection logs to check whether there is the required permission.

Figure 9-5 and Figure 9-6 show the logs when subnet ID and route ID of the destination cannot be obtained because there is no permission.

Solution: Grant DLI the VPC Administrator permission and cancel the IAM ReadOnlyAccess authorization.

Figure 9-5 Viewing connection logs



Figure 9-6 Viewing connection logs

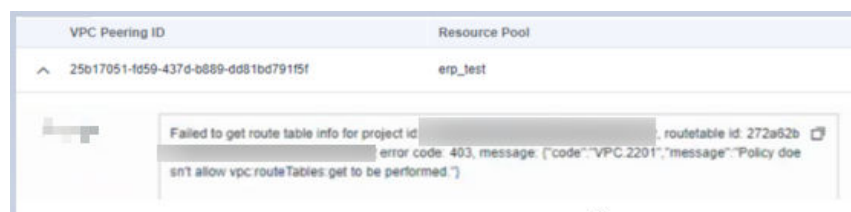
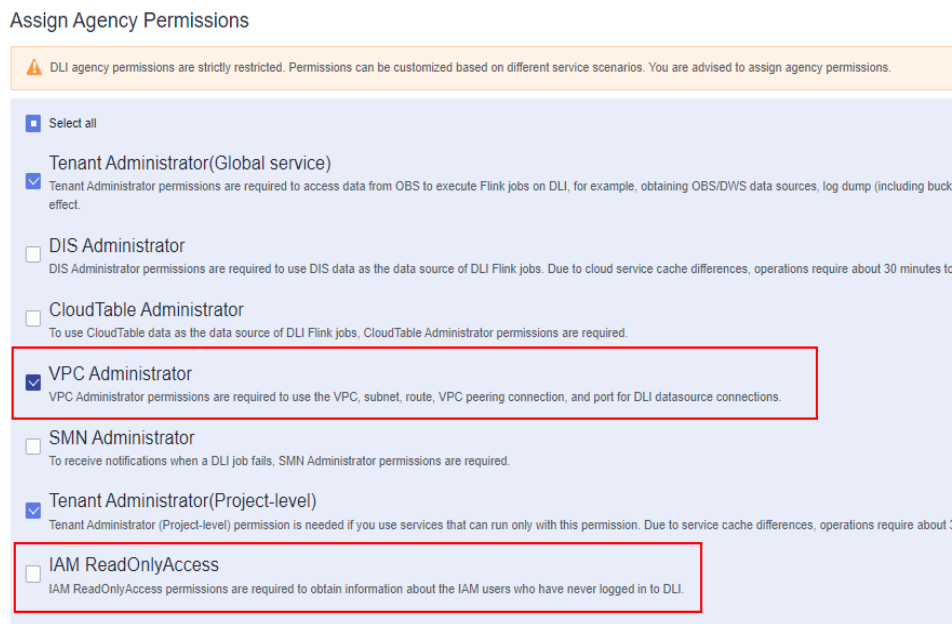


Figure 9-7 Selecting VPC administrator

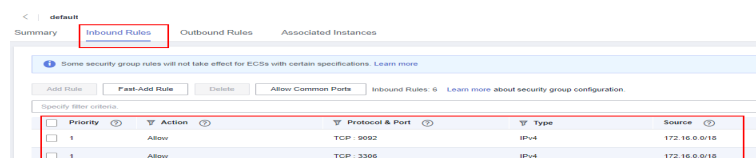


Check Whether the Destination Security Group Allows Access from the CIDR Block of the Queue

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 9-8 VPC security group rules

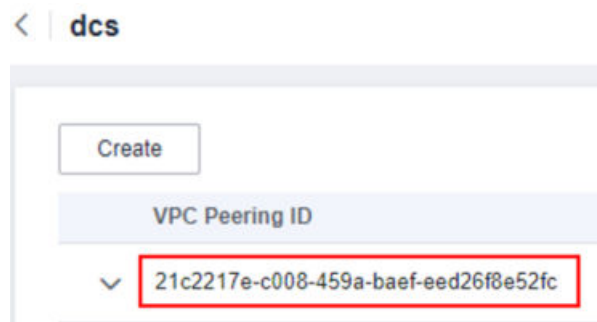


Check the Route Information of the VPC Peering Connection Corresponding to an Enhanced Datasource Connection

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 9-9 Obtaining the VPC peering connection ID



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

Figure 9-10 Viewing a VPC peering connection

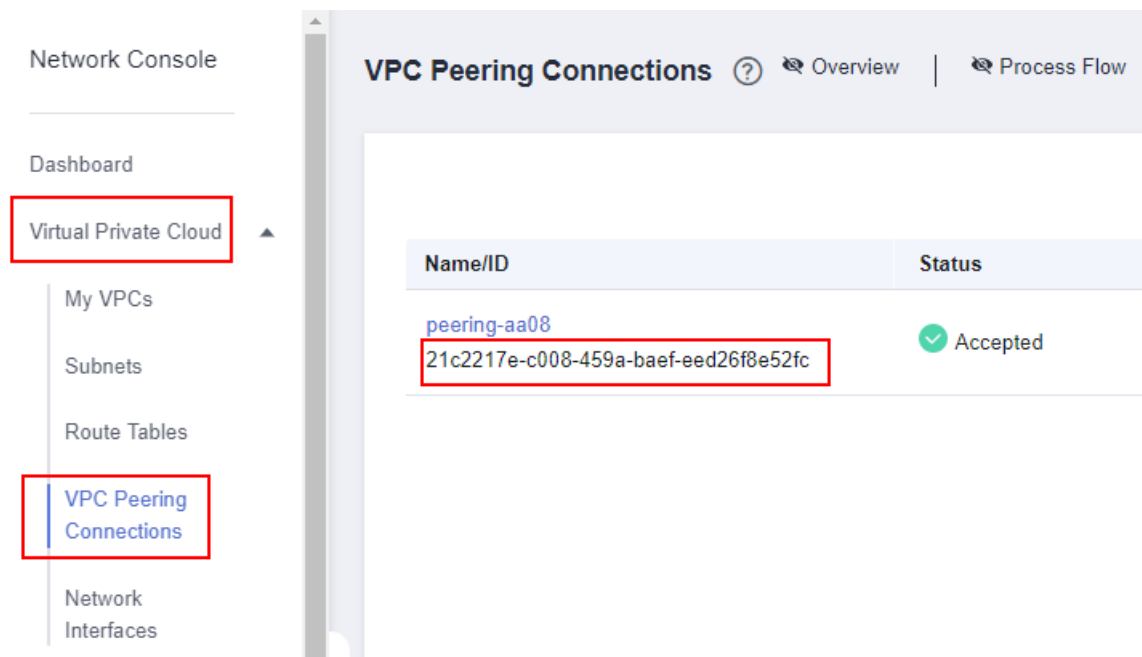
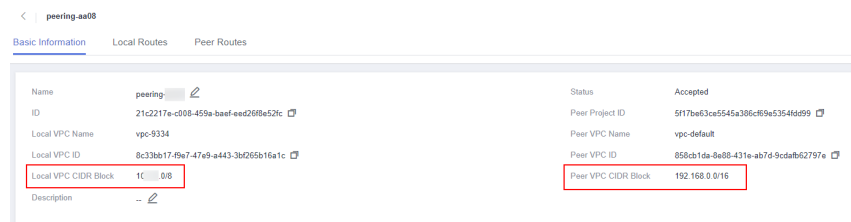
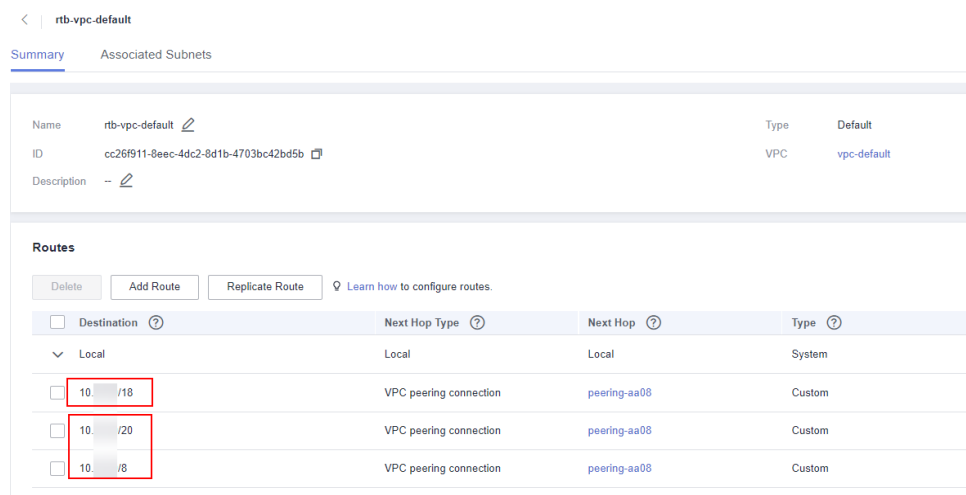


Figure 9-11 Viewing the CIDR block of the queue



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

Figure 9-12 Viewing the destination addresses in the routing table



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.

9.1.5 How Do I Configure the Network 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 9-13 Configuration process

For details about how to configure an enhanced datasource connection, see [Creating an Enhanced Datasource Connection to Other Data Sources](#).

- **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 9-14 Configuration process

9.1.6 What Can I Do If a Datasource Connection Is Stuck in Creating State When I Try to Bind a Queue to It?

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.

9.1.7 How Do I Bind a Datasource Connection to a Pay-per-Use Queue?

Enhanced datasource connections support only yearly/monthly and pay-per-use queues.

You can bind a datasource connection only when you select **Dedicated Resource Mode** when purchasing a pay-per-use queue. The resources are dedicated and used on demand (the billing period is equal to the lifecycle of the queue).

9.1.8 How Do I Connect DLI to Data Sources?

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

For details, see [Creating an Enhanced Datasource Connection](#).

9.2 Cross-Source Analysis

9.2.1 How Can I Perform Query on Data Stored on Services Rather Than DLI?

To perform query on data stored on services rather than DLI, perform the following steps:

1. Assume that the data to be queried is stored on multiple services (for example, OBS) on cloud.
2. On the DLI management console, create a table and set the **Path** parameter to the save path of the target data, for example, path of an OBS bucket. The data is actually stored on OBS and data migration is not required.
3. On the DLI management console, edit SQL statements to query and analyze the data.

9.2.2 How Can I Access Data Across Regions?

1. Connect VPCs in different regions.
2. Create an enhanced datasource connection on DLI and bind it to a queue.
3. Add a DLI route.

9.2.3 How Do I Set the Auto-increment Primary Key or Other Fields That Are Automatically Filled in the RDS Table When Creating a DLI and Associating It with the RDS Table?

When data is inserted into DLI, set the ID field to NULL.

9.3 Datasource Connection O&M

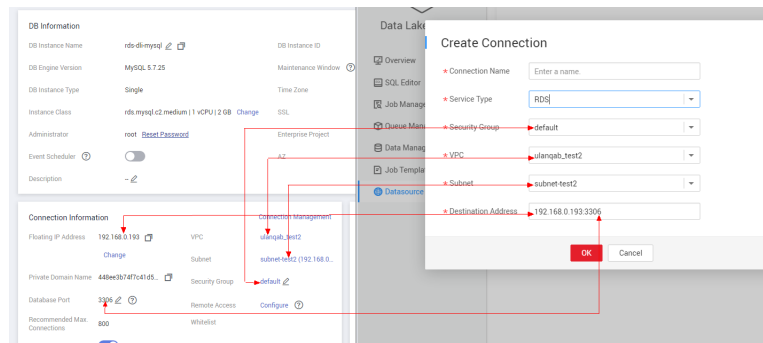
9.3.1 Why Is the Error Message "communication link failure" Displayed When I Use a Newly Activated Datasource Connection?

- 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**.
 - a. Delete the original datasource connection and create a new one. When you create a new 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 9-15 Creating a basic datasource connection-RDS

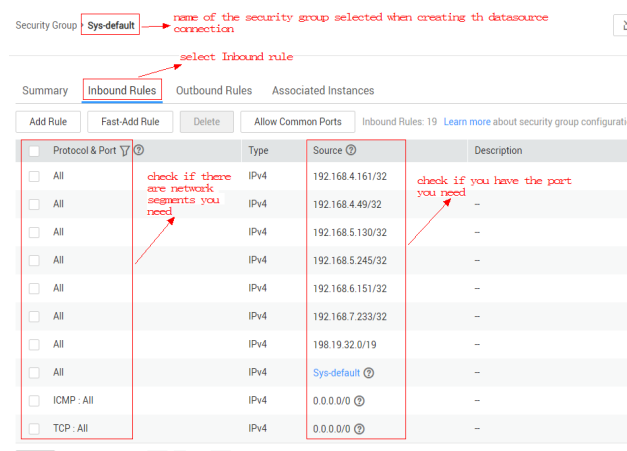


b. Check the configurations of VPC.

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

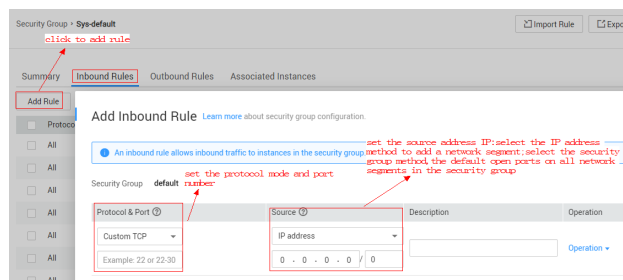
- Classic datasource connections:
 - Inbound rule: Check whether the inbound CIDR block and port in the security group have been enabled. If not, create the CIDR block and port you need.
- Check whether the CIDR block and port are configured.

Figure 9-16 Checking whether the CIDR block and port are configured



If no, add it.

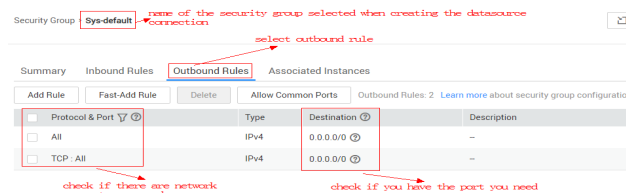
Figure 9-17 Adding an inbound rule



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

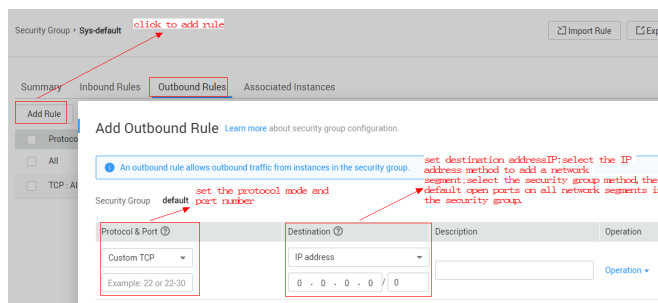
Check whether the CIDR block and port are configured.

Figure 9-18 Checking whether the CIDR block and port are configured



If no, add it.

Figure 9-19 Adding an outbound rule

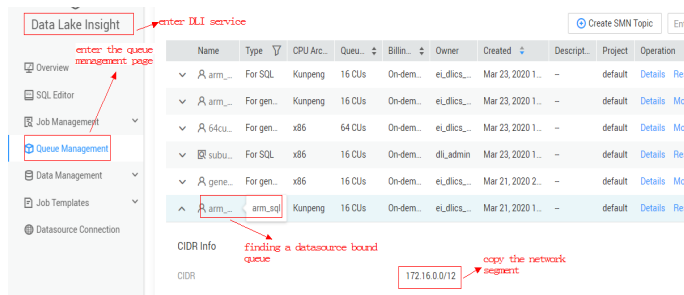


- Enhanced datasource connections:

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

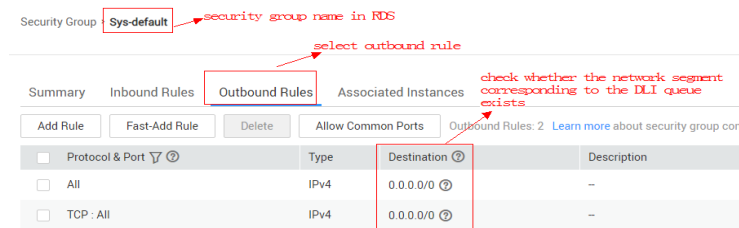
- 1) Find the CIDR block corresponding to the queue bound to the datasource connection in DLI.

Figure 9-20 Searching for the CIDR block corresponding to the queue bound to the datasource connection



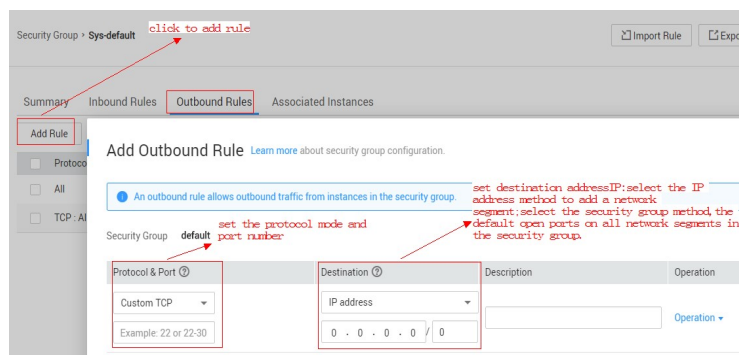
- 2) In the VPC security group, check whether the CIDR block corresponding to the DLI queue has been configured.

Figure 9-21 Checking the CIDR block corresponding to the DLI queue in the security group in the VPC



If no, add it.

Figure 9-22 Adding the corresponding CIDR to the VPC



9.3.2 Connection Times Out During MRS HBase Datasource Connection, and No Error Is Recorded in Logs

The cluster host information is not added to the datasource connection. As a result, the KRB authentication fails, the connection times out, and no error is recorded in logs. Configure the host information and try again.

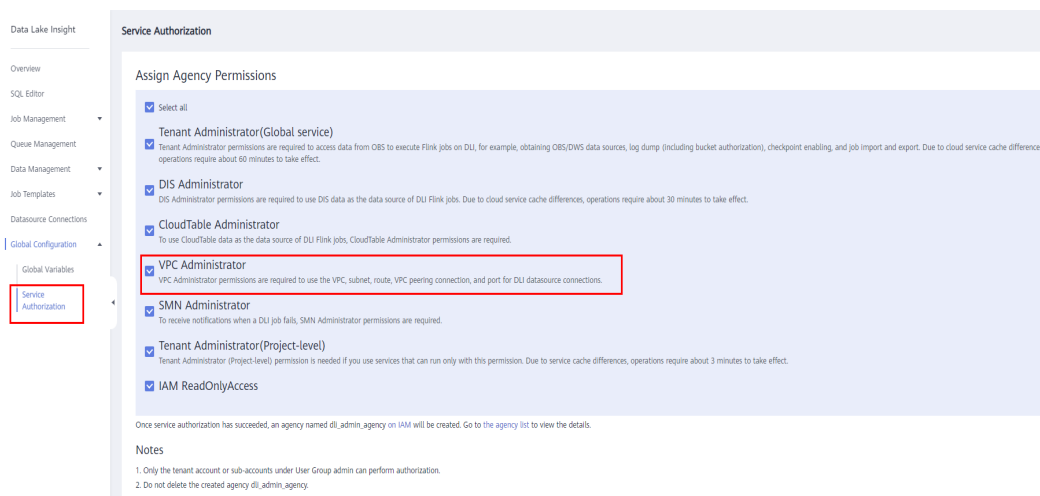
On the **Enhanced** page, select the connection and click **Modify Host**. In the dialog box that is displayed, enter the host information. The format is **Host IP address Host name/Domain name**. Multiple records are separated by line breaks.

9.3.3 Why Can't I Find the Subnet When Creating a DLI Datasource Connection?

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."}
```

Before you create a datasource connection, check whether **VPC Administrator** is selected. If only the global **Tenant Administrator** is selected, the system cannot find the subnet.

Figure 9-23 Selecting VPC administrator

9.3.4 Error Message "Incorrect string value" Is Displayed When insert overwrite Is Executed 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

9.3.5 Null Pointer Error Is Displayed When the System Creates a Datasource RDS 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](#) in *Data Lake Insight SQL Syntax Reference*.

9.3.6 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.

9.3.7 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.

9.3.8 A Null Value Is Written Into a Non-Null Field When a DLI Datasource Connection Is Used 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 9-24 Creation statement of the source table

```
CREATE EXTERNAL TABLE ... (  
    ...  
    'ctr' DOUBLE,  
    ...  
)
```

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

Figure 9-25 Creation statement of the target table

```
CREATE TABLE ... (  
    ...  
    'ctr' DECIMAL(9, 6),  
    ...  
) USING DWS OPTIONS (  
    ...  
) TBLPROPERTIES (  
    ...  
)
```

3. View the source table data. The CTR value that causes the problem is **1675**, which exceeded 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.

9.3.9 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.

10 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 Submit 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 Are Project IDs of Different Accounts the Same When They Are Used to Call APIs?

If different IAM accounts call APIs in the same enterprise project in the same region, the accounts use the same project ID.

10.5 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.sqlsync.enabled** to **true** to run the job asynchronously.

10.6 What Can I Do If Chinese Characters Returned by an API Are Garbled Characters?

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"))
```

11 SDKs

11.1 How Do I Set the Timeout Duration for Querying SQL Job Results Using SDK?

When you query the SQL job results using SDK, the system checks the job status when the job is submitted. The timeout interval set in the system is 300s. If the job is not in the **FINISHED** state, a timeout error is reported after 300s.

You are advised to use **getJobId()** to obtain the job ID and then call **queryJobResultInfo(String jobId)** or **cancelJob(String jobId)** to obtain the result or cancel the job.

11.2 How Do I Handle the dli.xxx,unable to resolve host address Error?

1. Run the ping command to check whether **dli.xxx** can be accessed.
If **dli.xxx** can be accessed, check whether DNS resolution is correctly configured.
2. DLI does not support cross-region access.

A Change History

Released On	What's New
2023-04-25	Added the following sections: <ul style="list-style-type: none">• Will an Error Be Reported if the Inherited Permissions Are Re granted to a Table That Inherits Database Permissions?• Why Can't I Query a View After I'm Granted the Select Table Permission on the View?
2023-03-31	<ul style="list-style-type: none">• Added Why Can't I Find the global_temp Database in the Spark 3.x Job Log?.
2022-09-15	This issue is the first official release.