Data Lake Insight

User Guide

Issue 01

Date 2024-12-28

HUAWEI TECHNOLOGIES CO., LTD.

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process*. For details about this process, visit the following web page:
https://www.huawei.com/en/psirt/vul-response-process
For vulnerability information, enterprise customers can visit the following web page:
https://securitybulletin.huawei.com/enterprise/en/security-advisory

Contents

1 DLI Job Development Process

This chapter walks you through on how to develop a DLI job.

Creating an IAM User and Granting Permissions

- To manage fine-grained permissions for your DLI resources using IAM, create an IAM user and grant them permissions to DLI if you are an enterprise user. For details, see **Creating an IAM User and Granting Permissions**.
- When using DLI for the first time, you need to update the DLI agency according to the console's guidance so that DLI can use other cloud services and perform resource O&M operations on your behalf. The agency includes permissions to obtain IAM user information, access and use VPCs, CIDR blocks, routes, and peering connections, and send notifications via SMN in case of job execution failure.

  For more information on the specific permissions included in the agency, refer to **Configuring DLI Agency Permissions**.

Creating Compute Resources and Metadata Required for Running Jobs

- Before submitting a job using DLI, you need to create an elastic resource pool and create queues within it. This will provide the necessary compute resources for running the job. For how to create an elastic resource pool and create queues within it, see **Overview of DLI Elastic Resource Pools and Queues**.

  Alternatively, you can enhance DLI's computing environment by creating custom images. Specifically, to enhance the functions and performance of Spark and Flink jobs, you can create custom images by downloading the base images provided by DLI and adding dependencies (files, JAR files, or software) and private capabilities required for job execution. This changes the container runtime environment for the jobs.

  For example, you can add a Python package or C library related to machine learning to a custom image to help you extend functions. For how to create a custom image, see **Using a Custom Image to Enhance the Job Running Environment**.
- DLI metadata is the basis for developing SQL and Spark jobs. Before executing a job, you need to define databases and tables based on your business scenario.

NOTE

Flink allows for dynamic data types, enabling the definition of data structures at runtime without the need for predefined metadata.

- Define your data structures, including data catalogs, databases, and tables. For details, see **Creating Databases and Tables**.
- Create a bucket to store temporary data generated during job running, such as job logs and job results. For details, see **Configuring a DLI Job Bucket**.
- Configure the permission to access metadata. For details, see **Configuring Database Permissions on the DLI Console** and **Configuring Table Permissions on the DLI Console**.

Importing Data to DLI

- DLI allows you to analyze and query data stored in OBS without the need to migrate it. Simply upload your data to OBS and use DLI for data analysis.
- Migrate data from various sources to DLI for central storage and processing.

  For how to migrate data to DLI, see **Using CDM to Migrate Data to DLI**. After the data migration, you can submit jobs.
- Cross-source access can reduce data duplication and latency when real-time access and processing of data from different sources is required for service needs.

  The prerequisites for cross-source access are that DLI can communicate with the data source network and DLI can obtain the access credentials to the data source.
  - Configure network connection between DLI and the data source by referring to **Configuring the Network Connection Between DLI and Data Sources (Enhanced Datasource Connection)**.
  - Manage data source credentials.
    - You can use DLI's datasource authentication to manage the authentication information for accessing a specified datasource.

      This applies to SQL jobs and Flink 1.12 jobs. For details, see **Using DLI Datasource Authentication to Manage Access Credentials for Data Sources**.
    - You can also use DEW to manage access credentials for data sources and use a custom agency to authorize DLI to access DEW.

      This applies to Spark 3.3.1 or later and Flink 1.15 or later. For details, see **Using DEW to Manage Access Credentials for Data Sources** and **Configuring an Agency to Allow DLI to Access Other Cloud Services**.

Submitting a Job Using DLI

- DLI offers a serverless service that integrates stream processing, batch processing, and interactive analytics. It supports various job types to meet different data processing needs.

Table 1-1 Job types supported by DLI

Job Type	Description	Use Case
SQL job	This type is suitable for scenarios where standard SQL statements are used for querying. It is typically used for querying and analyzing structured data. For details, see **Creating and Submitting a SQL Job**.	It applies to scenarios such as data warehouse query, report generation, and online analytical processing (OLAP).
Flink job	This type is specifically designed for real-time data stream processing, making it ideal for scenarios that require low latency and quick response. It is well-suited for real-time monitoring and online analysis. • Flink OpenSource job: DLI provides standard connectors and various APIs to facilitate quick integration with other data systems. For details, see **Creating a Flink OpenSource SQL Job**. • Flink Jar job: allows you to submit Flink jobs compiled into JAR files, providing greater flexibility and customization capabilities. It is suitable for complex data processing scenarios that require user-defined functions (UDFs) or specific library integration. The Flink ecosystem can be utilized to implement advanced stream processing logic and status management. For details, see **Creating a Flink Jar Job**.	It applies to scenarios that require quick response, such as real-time data monitoring and real-time recommender systems. Flink Jar jobs are suitable for data analysis scenarios that require custom stream processing logic, complex state management, or integration with specific libraries.
Spark job	Compute jobs can be submitted through interactive sessions or batch processing. Jobs are submitted to queues created within an elastic resource pool, simplifying resource management and job scheduling. It supports multiple data sources and formats, providing rich data processing capabilities, including but not limited to SQL queries and machine learning. For details, see **Creating a Spark Job**.	It is suitable for large-scale data processing and analysis, such as machine learning training, log analysis, and large-scale data mining.

- Manage program packages of Jar jobs.

DLI allows you to submit Flink or Spark jobs compiled as JAR files, which contain the necessary code and dependency information for executing the job. These files are used for specific data processing tasks such as data query, analysis, and machine learning. You can manage program packages required for jobs on the DLI console.

To submit a Spark Jar or Flink Jar job, you must first upload the program package to OBS, create a program package in DLI, and then submit the program package, data, and job parameters to run the job. For details, see **Managing Program Packages of Jar Jobs**.

NOTE

For Spark 3.3.1 or later and Flink 1.15 or later, when creating a Jar job, you can directly configure the program package in OBS. Program packages cannot be read from DLI.

Using Cloud Eye to Monitor DLI

You can query DLI monitoring metrics and alarms through Cloud Eye management console or APIs.

For example, you can monitor the resource usage and job status of a DLI queue. For details about DLI metrics, see **Using Cloud Eye to Monitor DLI**.

Using CTS to Audit DLI

With CTS, you can log operations related to DLI, making it easier to search, audit, and trace in the future. For the supported operations, see **Using CTS to Audit DLI**.

2 Preparations

2.1 Configuring DLI Agency Permissions

To use DLI, you must first configure permissions.

This section applies to the following scenarios:

- **If you use DLI for the first time, configure DLI agency permissions by referring to this section.**

  DLI needs to work with other cloud services. You must grant DLI basic operation permissions of these services so that DLI can access them and perform resource O&M operations on your behalf.
- **If you are still using the previous-generation agency dli_admin_agency, update it by referring to this section.**

  To balance practical business needs with the risk of excessive delegation, DLI upgraded its system agency to achieve more granular control over permissions. The previous **dli_admin_agency** was upgraded to **dli_management_agency**, which includes permissions for accessing IAM user information, datasource operations, and message notifications. This effectively prevents uncontrolled permission issues related to the services associated with DLI. After the upgrade, the DLI agency is more flexible and more suitable for scenario-based agency customization for medium- and large-sized enterprises.

After agency permissions are configured, the **dli_management_agency** agency is generated on the **Agencies** page of the IAM console. Do not delete this default system agency. Otherwise, the permissions included in the agency will be automatically revoked. The system cannot obtain IAM user information, access network resources required by datasource connections, or access SMN to send notifications.

Notes and Constraints

- Only the tenant account or a member account of user group **admin** can authorize the service.
- DLI authorization needs to be conducted by project. The permissions of required agencies must be updated separately in each project. This means you

need to switch to the corresponding project and then update the agency by following the instructions provided in this section.

Updating DLI Agency Permissions (dli_management_agency)

1. In the navigation pane of the DLI console, choose **Global Configuration** > **Service Authorization**.
2. On the displayed page, select permissions for scenarios.

   Click on a permission card to view its detailed permission policies.

   Table 2-1 describes these agencies.

Table 2-1 Permissions contained in the dli_management_agency agency

Use Case	Agency	Description
Basic usage	IAM ReadOnlyAccess	To authorize IAM users who have not logged in to DLI, you need to obtain their information. So, the permissions contained in the **IAM ReadOnlyAccess** policy are required. **IAM ReadOnlyAccess** is a global policy. Make sure you select this policy. If you do not select it, all its permissions will become invalid in all regions, and the system cannot obtain IAM user information.
Datasource	DLI Datasource Connections Agency Access	Permissions to access and use VPCs, subnets, routes, and VPC peering connections
O&M	DLI Notification Agency Access	Permissions to send notifications through SMN when a job fails to be executed

NOTE

Among the permissions contained in **dli_management_agency**:

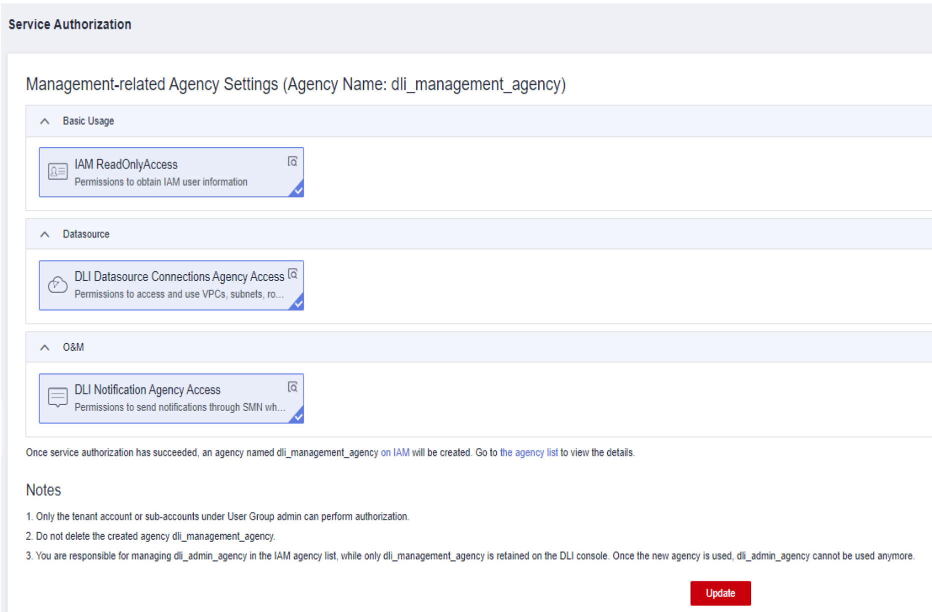
- The authorization scope of the **IAM ReadOnlyAccess** policy covers all global service resources in all regions.
  - If you select this policy when updating a DLI agency in any region, this policy's permissions apply to the projects in all regions.
  - If you do not select this policy when updating an agency in any project, this policy's permissions will be revoked from all regions. This means that all projects cannot obtain IAM user information.
- The authorization scope of the **DLI Datasource Connections Agency Access** and **DLI Notification Agency Access** policies covers the project resources in specified regions.

  These policies' permissions only apply to projects for which these policies are selected and the DLI agency permissions are updated. Projects for which these policies are not selected do not have the permissions required in datasource scenarios and the permission to send notifications using SMN.

Example 1: Configure Permissions for DLI Usage, Datasource Connection, and O&M Scenarios in Project A and **Example 2: Configure Permissions for DLI Usage, Datasource Connection, and O&M Scenarios in Project B** demonstrate the agency permission differences caused by updating a DLI agency for different projects in a region.

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

Figure 2-1 Updating agency permissions



4. View and understand the notes for updating the agency and click **OK**. The DLI agency permissions are updated.
   - The system upgrades your **dli_admin_agency** to **dli_management_agency**.
   - To maintain compatibility with existing job agency permission requirements, **dli_admin_agency** will still be listed in the IAM agency list even after the update.

– Do not delete the agency created by the system by default.

Follow-Up Operations

In addition to the permissions provided by **dli_management_agency**, you need to create an agency on the IAM console and add information about the new agency to the job configuration for scenarios like allowing DLI to read and write data from and to OBS to transfer logs, or allowing DLI to access DEW to obtain data access credentials. For details, see **Creating a Custom DLI Agency** and **Agency Permission Policies in Common Scenarios**.

- When Flink 1.15, Spark 3.3.1 (Spark general queue scenario), or a later version is used to execute jobs, you need to create an agency on the IAM console.
- If the engine version is earlier than Flink 1.15, **dli_admin_agency** is used by default during job execution. If the engine version is earlier than Spark 3.3.1, user authentication information (AK/SK and security token) is used during job execution.

  This means that jobs whose engine versions are earlier than Flink 1.15 or Spark 3.3.1 are not affected by the update of agency permissions and do not require custom agencies.

Common service scenarios where you need to create an agency:

- Data cleanup agency required for clearing data according to the lifecycle of a table and clearing lakehouse table data. You need to create a DLI agency named **dli_data_clean_agency** on IAM and grant permissions to it. You need to create an agency and customize permissions for it. However, the agency name is fixed to **dli_data_clean_agency**.
- **Tenant Administrator** permissions are required to access data from OBS to execute Flink jobs on DLI, for example, obtaining OBS data sources, log dump (including bucket authorization), checkpointing enabling, and job import and export.
- The AK/SK required by DLI Flink jobs is stored in DEW. To allow DLI to access DEW data during job execution, you need to create an agency to delegate the permissions to operate on DEW data to DLI.
- To allow DLI to access DLI catalogs to retrieve metadata when executing jobs, you need to create a new agency that grants DLI catalog data operation permissions to DLI. This will enable DLI to access DLI catalogs on your behalf.
- Cloud data required by DLI Flink jobs is stored in LakeFormation. To allow DLI to access catalogs to retrieve metadata during job execution, you need to create an agency to delegate the permissions to operate on catalog data to DLI.

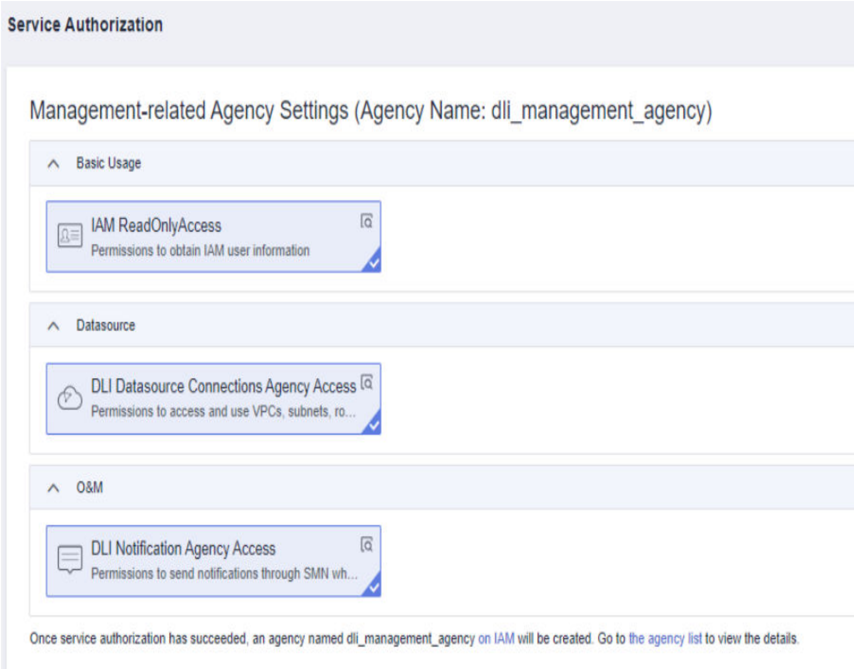
When creating an agency, you cannot use the default agency names **dli_admin_agency**, **dli_management_agency**, or **dli_data_clean_agency**. It must be unique.

For more information about custom agency operations, see **Creating a Custom DLI Agency** and **Agency Permission Policies in Common Scenarios**.

Example 1: Configure Permissions for DLI Usage, Datasource Connection, and O&M Scenarios in Project A

- **Operation instruction**: A DLI user upgrades **dli_admin_agency** to **dli_management_agency** for project A in **CN North-Beijing4**.
  a. On the DLI management console, switch to project A in the **CN North-Beijing4** region and choose **Global Configuration** > **Service Authorization**.
  b. Select the policies under **Basic Usage**, **Datasource**, and **O&M**.

  Figure 2-2 Updating a DLI agency for project A in CN North-Beijing4



  c. Click **Update**.
- **Permission description**: The agency permissions are updated for project A in **CN North-Beijing4**.
  – The **IAM ReadOnlyAccess** policy's permissions are granted to global service resources, meaning that all regions and projects have these permissions.
  – The **DLI Datasource Connections Agency Access** and **DLI Notification Agency Access** policies contain only regional permissions, meaning that their permissions only apply to project A in **CN North-Beijing4**.

Example of Agency Permissions for Project A in CN North-Beijing4	Example of Agency Permissions for Project B in CN North-Beijing4
dli_management_agency The new agency contains the following policy permissions: ● IAM ReadOnlyAccess ● DLI Datasource Connections Agency Access ● DLI Notification Agency Access	dli_management_agency The new agency contains the following policy permissions: ● IAM ReadOnlyAccess

Example 2: Configure Permissions for DLI Usage, Datasource Connection, and O&M Scenarios in Project B

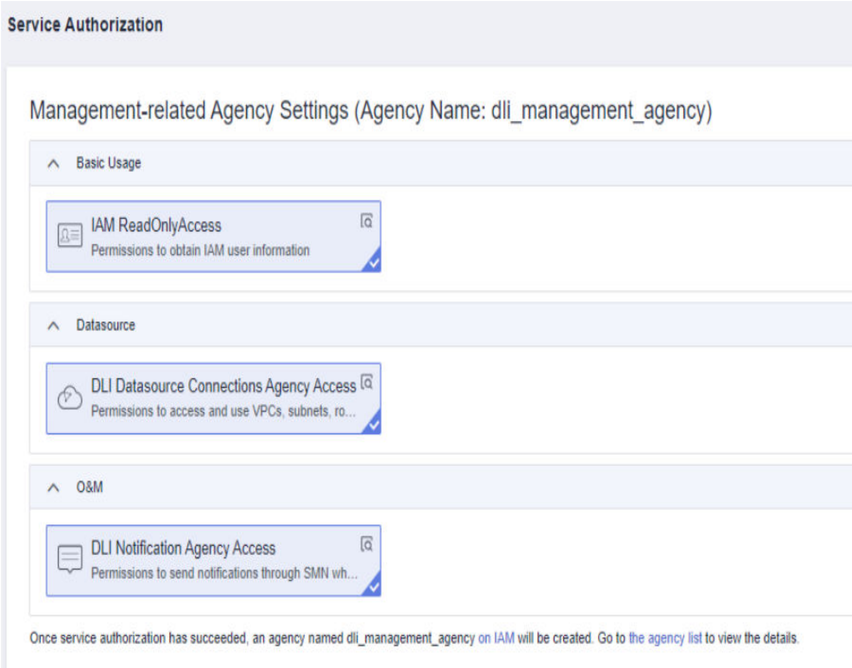
Operation instruction: To assign the permissions of the **DLI Datasource Connections Agency Access** and **DLI Notification Agency Access** policies to project B in the **CN North-Beijing4** region, perform the following operations to update the agency permissions of project B in **CN North-Beijing4**:

1. On the DLI management console, switch to project B in the **CN North-Beijing4** region and choose **Global Configuration** > **Service Authorization**.
2. Select the policies under **Basic Usage**, **Datasource**, and **O&M**.

CAUTION

When updating the agency permissions for project B, you will need to select the **IAM ReadOnlyAccess** policy, as its authorization scope covers global service resources. If you deselect it and update the agency, all its permissions will become invalid in all regions and projects.

Figure 2-3 Updating a DLI agency for project B in CN North-Beijing4



3. Click **Update**.

Permission description:

The authorization scope of the **DLI Datasource Connections Agency Access** and **DLI Notification Agency Access** policies covers the project resources in specified regions. Following updates to agency permissions, project B in **CN North-Beijing4** has the permission to obtain IAM user information, perform datasource operations, and send message notifications.

Example of Agency Permissions in Region A	Example of Agency Permissions in Region B
dli_management_agency The new agency contains the following policy permissions: ● IAM ReadOnlyAccess ● DLI Datasource Connections Agency Access ● DLI Notification Agency Access	dli_management_agency The new agency contains the following policy permissions: ● IAM ReadOnlyAccess ● DLI Datasource Connections Agency Access ● DLI Notification Agency Access

2.2 Creating an IAM User and Granting Permissions

You can use Identity and Access Management (IAM) to implement fine-grained permissions control on DLI resources. For details, see **Overview**.

If your cloud account does not need individual IAM users, then you may skip over this section.

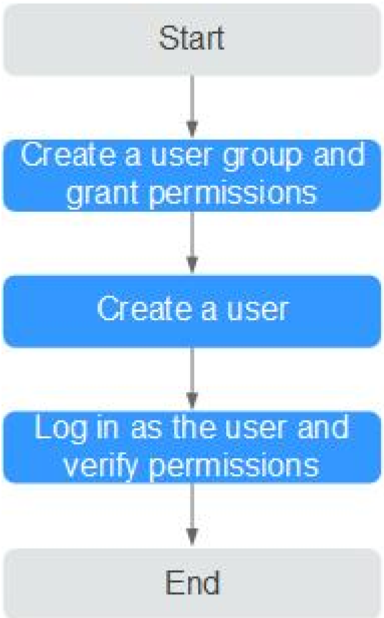
This section describes how to create an IAM user and grant DLI permissions to the user. **Figure 2-4** shows the procedure.

Prerequisites

Before assigning permissions to user groups, you should learn about system policies and select the policies based on service requirements. For details about system permissions supported by DLI, see **DLI System Permissions**.

Process Flow

Figure 2-4 Process for granting DLI permissions



1. **Create a user group and grant permissions to it.**

   Create a user group on the IAM console, and assign the **DLI ReadOnlyAccess** permission to the group.
2. **Create an IAM user.**

   Create a user on the IAM console and add the user to the group created in **1**.
3. **Log in** and verify permissions.

   Log in to the management console using the newly created user, and verify the user permissions.
   - Choose **Service List** > **Data Lake Insight**. The DLI management console is displayed. If you can view the queue list on the **Queue Management** page but cannot buy DLI queues by clicking **Buy Queue** in the upper right corner (assume that the current permission contains only **DLI ReadOnlyAccess**), the **DLI ReadOnlyAccess** permission has taken effect.
   - Choose any other service in **Service List**. If a message appears indicating that you have insufficient permissions to access the service, the **DLI ReadOnlyAccess** permission has already taken effect.

More

- For how to create an IAM user, see **Creating an IAM User**.
- For how to create a custom policy, see **Creating a Custom Policy**.
- For how to modify a user policy, see **Modifying or Deleting a Custom Policy**.

2.3 Configuring a DLI Job Bucket

Before using DLI, you need to configure a DLI job bucket. The bucket is used to store temporary data generated during DLI job running, such as job logs and results.

Configure a DLI job bucket on the **Global Configuration** > **Project** page of the DLI management console.

Preparations

Before the configuration, create an **OBS bucket** or **parallel file system (PFS)**.

In big data scenarios, you are advised to create a PFS. PFS is a high-performance file system provided by OBS, with access latency in milliseconds. PFS can achieve a bandwidth performance of up to TB/s and millions of IOPS, which makes it ideal for processing high-performance computing (HPC) workloads.

For details about PFS, see "Parallel File System Feature Guide" in the *Object Storage Service User Guide*.

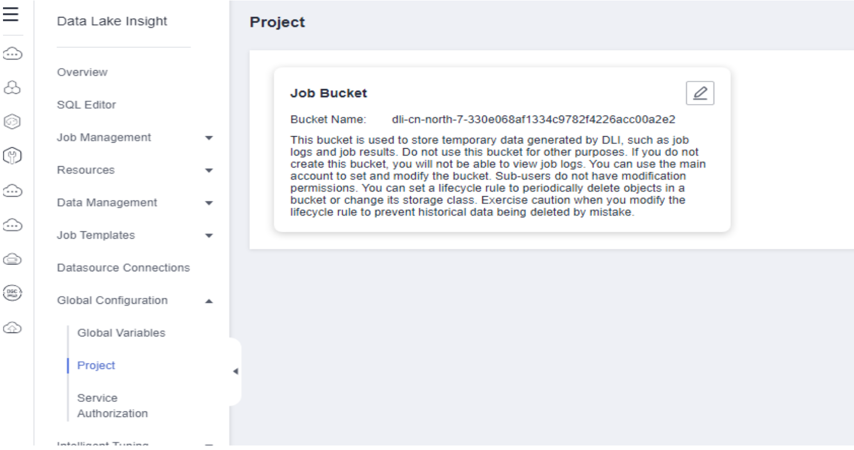
Notes

- Do not use the OBS bucket for other purposes.
- The OBS bucket must be set and modified by the main account. Member users do not have the permission.
- If the bucket is not configured, you will not be able to view job logs.
- You can create lifecycle rules to automatically delete objects or change storage classes for objects that meet specified conditions.
- Inappropriate modifications of the job bucket may lead to loss of historical data.

Procedure

1. In the navigation pane of the DLI console, choose **Global Configuration** > **Project**.
2. On the **Project** page, click  next to **Job Bucket** to configure bucket information.

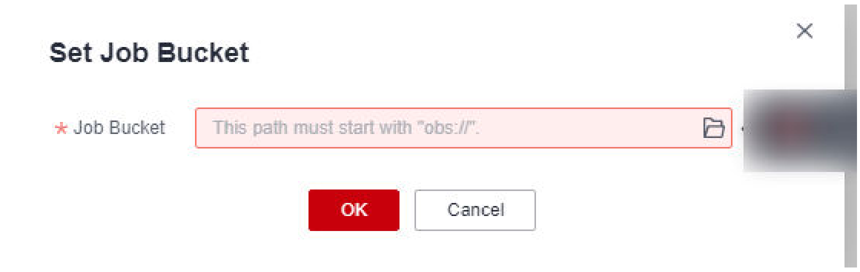
Figure 2-5 Project



3. Click  to view available buckets.
4. Select the bucket for storing the temporary data of the DLI job and click **OK**.

   Temporary data generated during DLI job running will be stored in the OBS bucket.

Figure 2-6 Setting the job bucket



3 Creating an Elastic Resource Pool and Queues Within It

3.1 Overview of DLI Elastic Resource Pools and Queues

DLI compute resources are the foundation to run jobs. This section describes the modes of DLI compute resources and queue types.

What Are Elastic Resource Pools and Queues?

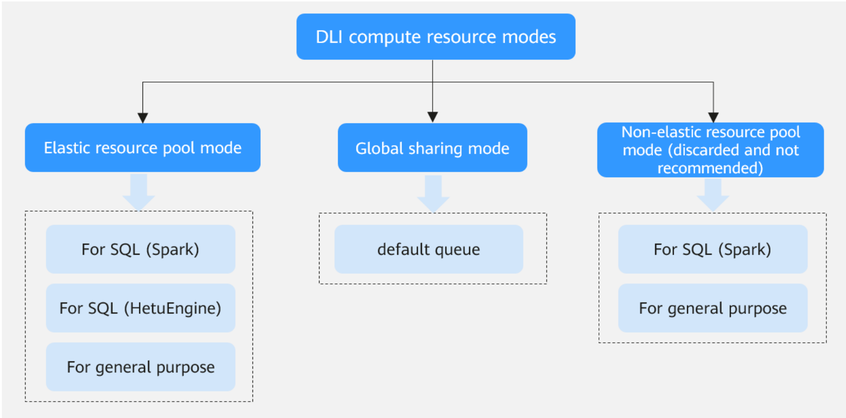
Before we dive into the compute resource modes of DLI, let us first understand the basic concepts of elastic resource pools and queues.

- An **elastic resource pool** is a pooled management mode for DLI compute resources, which can be seen as a collection of DLI compute resources. DLI supports the creation of multiple queues within an elastic resource pool, and these queues can share the resources in the elastic resource pool.
- **Queues** are the basic units of compute resources that are actually used and allocated in DLI. You can create different queues for different jobs or data processing tasks, and allocate and adjust resources for these queues as needed. To learn more about the queue types in DLI, refer to **DLI Queue Types**.

DLI Compute Resource Modes

DLI offers three compute resource management modes, each with unique advantages and use cases.

Figure 3-1 DLI compute resource modes



- **Elastic resource pool mode**: a pooled management mode for compute resources that provides dynamic scaling capabilities. Queues within the same elastic resource pool share compute resources. By setting up a reasonable compute resource allocation policy for queues, you can improve compute resource utilization and meet resource demands during peak hours.
  - Use cases: suitable for scenarios with significant fluctuations in business volume, such as periodic data batch processing tasks or real-time data processing needs.
  - Supported queue types: for SQL (Spark), for SQL (HetuEngine), and for general purpose. To learn more about the queue types in DLI, refer to **DLI Queue Types**.

    NOTE

    General-purpose queues and SQL queues in elastic resource pool mode do not support cross-AZ deployment.
  - Usage: first create an elastic resource pool, then create queues within the pool and allocate compute resources. Queues are associated with specific jobs and data processing tasks.

    For how to buy an elastic resource pool and create queues within it, see **Creating an Elastic Resource Pool and Creating Queues Within It**.
- **Global sharing mode:**

  Global sharing mode is a compute resource allocation mode that allocates resources based on the actual amount of data scanned in SQL queries. It does not support specifying or reserving compute resources.

  The **default** queue, which is preset by DLI, is the compute resource for global sharing mode, and the resource size is allocated on demand. Users who are unsure of the data size or occasionally need to process data can use the **default** queue to run jobs.
  - Use cases: suitable for testing jobs or scenarios with low resource consumption.
  - Supported queue types: Only the preset **default** queue in DLI is the compute resource for global sharing mode.

    The **default** queue is typically used by users who are new to DLI but it may lead to resource contention and prevent you from getting the resources you need for your jobs, as its resources are shared among all users. You are advised to use self-built queues to run production jobs.

– Usage: The **default** queue is only applicable to submitting SQL jobs. When submitting SQL jobs on the DLI management console, select the **default** queue.

- **Non-elastic resource pool mode (discarded and not recommended):**

  The previous-gen of DLI's compute resource management mode is no longer recommended due to its lack of flexibility.

  Non-elastic resource pool mode provides fixed-specification compute resources that are purchased and exclusively used, and cannot be dynamically adjusted according to demand, which may result in resource waste or insufficient resources during peak hours.

Table 3-1 DLI compute resource modes and supported queue types

DLI Compute Resource Mode	Supported Queue Type	Resource Characteristic	Use Case
Elastic resource pool mode	For SQL (Spark) For SQL (HetuEngine) For general purpose	Resources are shared among multiple queues for a single user. Resources are dynamically allocated and can be flexibly adjusted.	Suitable for scenarios with significant fluctuations in business demand, where resources need to be flexibly adjusted to meet peak and off-peak demands.
Global sharing mode	default queue	Resources are shared among multiple queues for multiple users. You are billed on a pay-per-use basis. Resources cannot be reserved.	Suitable for temporary or testing projects where data size is uncertain or data processing is only required occasionally.
Non-elastic resource pool mode **(discarded, not recommended)**	For SQL For general purpose	Resources are exclusively used by a single user and a single queue. Resources cannot be dynamically adjusted and may remain idle.	Discarded and not recommended.

To help you understand the use cases for different DLI compute resource modes, we can compare purchasing DLI compute resources to using car services:

- The elastic resource pool mode can be compared to "renting a car" where you can dynamically adjust the scale of resources based on actual needs.

This mode is suitable for scenarios with significant fluctuations in business demand, allowing for flexible adjustment of resources based on peak and off-peak demands to optimize costs.

- The global sharing mode can be compared to "taking a taxi" where you only pay for the actual amount of data used.

  This mode is suitable for scenarios where data size is uncertain or data processing is only required occasionally, allowing for on-demand use of resources without the need to pre-purchase or reserve resources.

DLI Queue Types

DLI is divided into three queue types: **default** queue, for SQL, and for general purpose. You can choose the most suitable queue type based on your business scenario and job characteristics.

- **default queue:**

  The **default** queue is a preset queue that is shared among all users.

  The **default** queue does not support specifying the size of resources and resources are allocated on-demand during job execution, with billing based on the actual amount of data scanned.

  As resources of the **default** queue are shared among all users, there may be resource contention during use, and it cannot be guaranteed that resources will be available for every job execution.

  The **default** queue is suitable for small-scale or temporary data processing needs. For important jobs or jobs that require guaranteed resources, you are advised to buy an elastic resource pool and create queues within it to execute jobs.

- **For SQL:**

  For SQL queues are used to execute SQL jobs and supports specifying engine types including Spark and HetuEngine.

  This type of queues is suitable for businesses that require fast data query and analysis, as well as regular cache clearing or environment resetting.

- **For general purpose:**

  For general purpose queues are used to execute Spark jobs, Flink OpenSource SQL jobs, and Flink Jar jobs.

  It is suitable for complex data processing, real-time data stream processing, or batch data processing scenarios.

Use Cases of Elastic Resource Pools

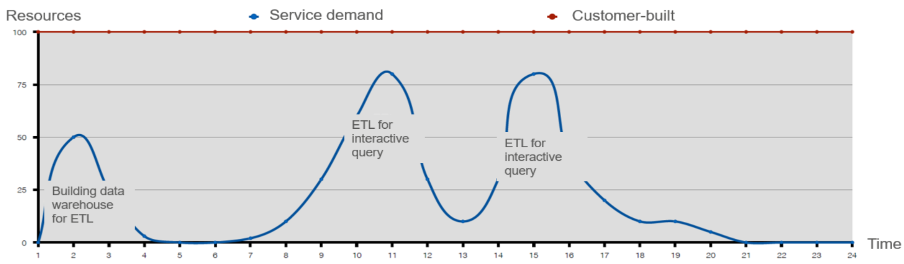
Queues in an elastic resource pool are recommended, as they offer the flexibility to use resources with high utilization as needed. This part describes common use cases of elastic resource pools.

Resources too fixed to meet a range of requirements.

The quantities of compute resources required for jobs change in different time of a day. If the resources cannot be scaled based on service requirements, they may be wasted or insufficient. **Figure 3-2** shows the resource usage during a day.

- After ETL jobs are complete, no other jobs are running during 04:00 to 07:00 in the early morning. The resources could be released at that time.
- From 09:00 to 12:00 a.m. and 02:00 to 04:00 p.m., a large number of ETL report and job queries are queuing for compute resources.

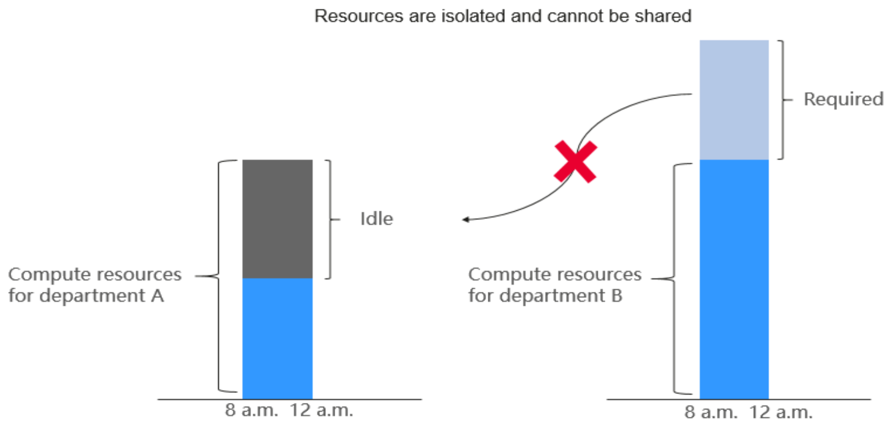
Figure 3-2 Fixed resources



Resources are isolated and cannot be shared.

A company has two departments, and each run their jobs on a DLI queue. Department A is idle from 08:00 to 12:00 a.m. and has remaining resources, while department B has a large number of service requests during this period and needs more resources to meet the requirements. Since the resources are isolated and cannot be shared between department A and B, the idle resources are wasted.

Figure 3-3 Resource waste due to resource isolation



Elastic resource pools can be accessed by different queues and automatically scaled to improve resource utilization and handle resource peaks.

You can use elastic resource pools to centrally manage and allocate resources. Multiple queues can be bound to an elastic resource pool to share the pooled resources.

3.2 Creating an Elastic Resource Pool and Creating Queues Within It

An elastic resource pool offers compute resources (CPU and memory) required for running DLI jobs, which can adapt to the changing demands of services.

You can create multiple queues within an elastic resource pool. These queues are associated with specific jobs and data processing tasks, and serve as the basic unit for resource allocation and usage within the pool. This means queues are specific compute resources required for executing jobs.

Queues within an elastic resource pool can be shared to execute jobs. This is achieved by correctly setting the queue allocation policy. This enhances queue utilization. This section describes how to create an elastic resource pool and create queues within it.

Notes and Constraints

- The billing mode of an elastic resource pool cannot be changed.
- The region of an elastic resource pool cannot be changed.
- For a pay-per-use resource pool, the dedicated resource mode is selected by default. The resource pool is billed by calendar hour since it is created.
- Jobs of Flink 1.10 or later can run in elastic resource pools.
- The CIDR block of an elastic resource pool cannot be changed once set.
- Associating an elastic resource pool with a queue:
  - Only pay-per-use queues (including dedicated queues) can be associated with elastic resource pools.
  - No resources are frozen.
- Currently, only yearly/monthly elastic resource pools can be scaled.
- You can only view the scaling history of resource pools in the last 30 days.
- Elastic resource pools cannot access the Internet.
- Changes to elastic resource pool CUs can occur when setting the CU, adding or deleting queues in an elastic resource pool, or modifying the scaling policies of queues in an elastic resource pool, or when the system automatically triggers elastic resource pool scaling. However, in some cases, the system cannot guarantee that the scaling will reach the target CUs as planned.
  - If there are not enough physical resources, an elastic resource pool may not be able to scale out to the desired target size.
  - The system does not guarantee that an elastic resource pool will be scaled in to the desired target size.

    The system checks the resource usage before scaling in the elastic resource pool to determine if there is enough space for scaling in. If the existing resources cannot be scaled in according to the minimum scaling step, the pool may not be scaled in successfully or only partially.

    The scaling step may vary depending on the resource specifications, usually 16 CUs, 32 CUs, 48 CUs, 64 CUs, etc.

For example, if the elastic resource pool has a capacity of 192 CUs and the queues in the pool are using 68 CUs due to running jobs, the plan is to scale in to 64 CUs.

When executing a scaling in task, the system determines that there are 124 CUs remaining and scales in by the minimum step of 64 CUs. However, the remaining 60 CUs cannot be scaled in any further. Therefore, after the elastic resource pool executes the scaling in task, its capacity is reduced to 128 CUs.

Creating an Elastic Resource Pool

Step 1 In the navigation pane on the left, choose **Resources** > **Resource Pool**.

Step 2 On the displayed page, click **Buy Resource Pool** in the upper right corner.

Step 3 On the displayed page, set the following parameters:

Table 3-2 Parameters

Parameter	Description
Region	Select a region. Select a region near you to ensure the lowest latency possible.
Project	Each region corresponds to a project.
Name	Name of the elastic resource pool. ● Only numbers, letters, and underscores (_) are allowed. The value cannot contain only numbers or start with an underscore (_) or number. ● The value can contain a maximum of 128 characters. **NOTE** The elastic resource pool name is case-insensitive. Uppercase letters will be automatically converted to lowercase letters.
Type	● **Basic edition: offers 16 CUs to 64 CUs** – This edition is suitable for testing scenarios with low resource consumption and low requirements for resource reliability and availability. – High reliability and availability are not supported. – Queue properties and job priorities cannot be set. – Notebook instances cannot be interconnected with. ● **Standard edition: offers at least 64 CUs** This edition offers powerful computing capabilities, high availability, and flexible resource management. It is suitable for large-scale computing tasks and business scenarios with long-term resource planning needs.

Parameter	Description
CU Range	The maximum and minimum CUs allowed for the elastic resource pool. CU settings are used to control the maximum and minimum CU ranges for elastic resource pools to avoid unlimited resource scaling. In **CU Range**, set the minimum CUs on the left and the maximum CUs on the right. ● The total minimum CUs of all queues in an elastic resource pool must be no more than the minimum CUs of the pool. ● The maximum CUs of any queue in an elastic resource pool must be no more than the maximum CUs of the pool. An elastic resource pool should at least ensure that all queues in it can run with the minimum CUs and should try to ensure that all queues in it can run with the maximum CUs. The specifications (yearly/monthly CUs) of an elastic resource pool are equal to the minimum CUs allocated during creation. This means that when the elastic resource pool is first created, the actual CUs will be equal to the specifications, which is also the minimum CUs.
Description	Description of the elastic resource pool
CIDR Block	CIDR block the elastic resource pool belongs to. If you use an enhanced datasource connection, this CIDR block cannot overlap that of the data source. **Once set, this CIDR block cannot be changed.** Recommended CIDR block: 10.0.0.0–10.255.0.0/16–19 172.16.0.0–172.31.0.0/16–19 192.168.0.0–192.168.0.0/16–19
Enterprise Project	If the created elastic resource pool belongs to an enterprise project, select the enterprise project. Enterprise projects let you manage cloud resources and users by project. For how to set enterprise projects, see the **Enterprise Management User Guide**. **NOTE** This parameter is displayed only for users who have enabled the Enterprise Management Service.

Parameter	Description
Tags	Tags used to identify cloud resources. A tag includes the tag key and tag value. If you want to use the same tag to identify multiple cloud resources, that is, to select the same tag from the drop-down list box for all services, you are advised to create predefined tags on the Tag Management Service (TMS). If your organization has configured tag policies for DLI, add tags to resources based on the policies. If a tag does not comply with the tag policies, resource creation may fail. Contact your organization administrator to learn more about tag policies. For details, see **Tag Management Service User Guide**. **NOTE** ● A maximum of 20 tags can be added. ● Only one tag value can be added to a tag key. ● The key name in each resource must be unique. ● Tag key: Enter a tag key name in the text box. **NOTE** A tag key can contain a maximum of 128 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed, but the value cannot start or end with a space or start with **_sys_**. ● Tag value: Enter a tag value in the text box. **NOTE** A tag value can contain a maximum of 255 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed. The value cannot start or end with a space.

Step 4 Click **Buy** and confirm the configurations.

Step 5 Click **Pay**. Wait until the status of the elastic resource pool changes to **Available**. The elastic resource pool is successfully created.

Step 6 Refer to **Example Use Case: Creating an Elastic Resource Pool and Running Jobs** and **Example Use Case: Configuring Scaling Policies for Queues in an Elastic Resource Pool** to perform subsequent operations as needed.

----End

Creating Queues Within an Elastic Resource Pool

Create one or more queues within an elastic resource pool to run jobs. This section describes how to create a queue within an elastic resource pool.

NOTE

Queues added to an elastic resource pool are not billed separately, but billed as billing items of the elastic resource pool.

Step 1 In the navigation pane on the left, choose **Resources** > **Resource Pool**.

Step 2 Locate the elastic resource pool in which you want to create queues and click **Add Queue** in the **Operation** column.

Step 3 On the **Add Queue** page, set basic queue parameters based on the table below.

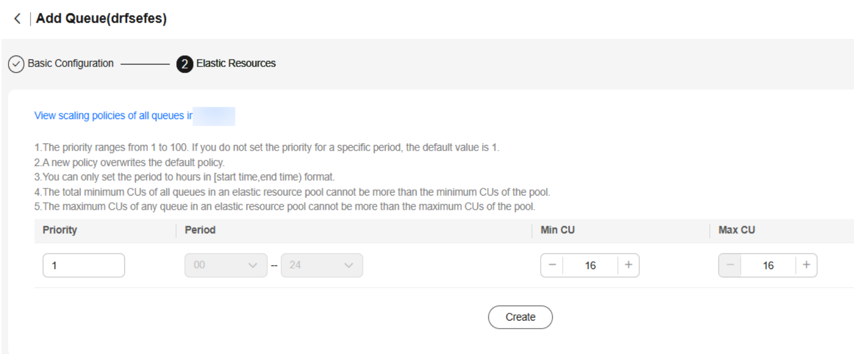
Table 3-3 Basic parameters for adding a queue

Parameter	Description
Name	Name of the queue to add
Type	● **For SQL**: The queue is used to run SQL jobs. ● **For general purpose**: The queue is used to run Spark and Flink jobs.
Engine	If **Type** is **For SQL**, the queue engine can be **Spark** or **HetuEngine**. For a HetuEngine SQL queue, there must be at least 96 CUs.
Enterprise Project	Select the enterprise project the queue belongs to. Queues under different enterprise projects can be added to an elastic resource pool. Enterprise projects let you manage cloud resources and users by project. For how to set enterprise projects, see the **Enterprise Management User Guide**. **NOTE** This parameter is available only for users who have subscribed to the Enterprise Management Service.
Description	Description about the queue.

Parameter	Description
Tags	Tags used to identify cloud resources. A tag includes the tag key and tag value. If you want to use the same tag to identify multiple cloud resources, that is, to select the same tag from the drop-down list box for all services, you are advised to create predefined tags on the Tag Management Service (TMS). If your organization has configured tag policies for DLI, add tags to resources based on the policies. If a tag does not comply with the tag policies, resource creation may fail. Contact your organization administrator to learn more about tag policies. For details, see **Tag Management Service User Guide**. **NOTE** ● You can add a maximum of 20 tags. ● You can only add one tag value to each tag key. ● The key name in each resource must be unique. ● Tag key: Enter a tag key name in the text box. **NOTE** Enter a maximum of 128 characters for each tag key. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed, but the value cannot start or end with a space or start with **_sys_**. ● Tag value: Enter a tag value in the text box. **NOTE** Enter a maximum of 255 characters for each tag value. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed. The value cannot start or end with a space.

Step 4 Click **Next**. On the displayed page, configure a scaling policy for the queue in the elastic resource pool.

Figure 3-4 Configuring a scaling policy when adding a queue



Click **Create** to add a scaling policy with varying priority, period, minimum CUs, and maximum CUs. The parameters of each scaling policy are:

Table 3-4 Scaling policy parameters

Parameter	Description
Priority	Priority of the scaling policy in the current elastic resource pool. A larger value indicates a higher priority. You can set a number ranging from 1 to 100.
Period	Time segment when the policy takes effect. It can be set only by hour. The start time is on the left, and the end time is on the right. ● The time range includes the start time but not the end time, that is, [start time, end time). For example, if you set **Period** to **01** and **17**, the scaling policy takes effect at 01:00 a.m. till 05:00 p.m. ● The periods of scaling policies with different priorities should not overlap.
Min CU	Minimum number of CUs allowed by the scaling policy. ● In any time segment of a day, the total minimum CUs of all queues in an elastic resource pool cannot be more than the minimum CUs of the pool. ● If the minimum CUs of the queue is less than 16 CUs, both **Max. Spark Driver Instances** and **Max. Prestart Spark Driver Instances** set in the queue properties do not apply. Refer to **Setting Queue Properties**. For a HetuEngine SQL queue, there must be at least 96 CUs.
Max CU	Maximum number of CUs allowed by the scaling policy. In any time segment of a day, the maximum CUs of any queue in an elastic resource pool cannot be more than the maximum CUs of the pool.

NOTE

- The first scaling policy is the default policy, and its **Period** parameter configuration cannot be deleted or modified.
- Flink jobs cannot trigger automatic scaling of queues in an elastic resource pool.

Step 5 Click **OK**. View all queues and scaling policies added to the elastic resource pool by referring to **Adjusting Scaling Policies for Queues in an Elastic Resource Pool**.

----End

3.3 Managing Elastic Resource Pools

3.3.1 Viewing Basic Information

After creating an elastic resource pool, you can check and manage it on the management console.

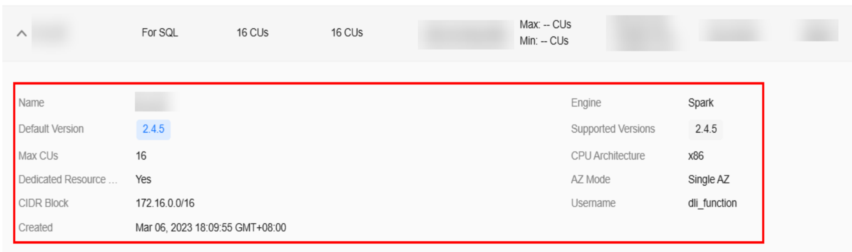
This section describes how to check basic information about an elastic resource pool on the management console, including the VPC CIDR block, IPv6 CIDR block, and creation time.

Procedure

1. Log in to the DLI management console.
2. In the navigation pane on the left, choose **Resources** > **Resource Pool**.
3. On the displayed page, select the elastic resource pool you want to check.
   - In the upper right corner of the list page, click to customize the columns to display and set the rules for displaying the table content and the **Operation** column.
   - In the search box above the list, you can filter the required elastic resource pool by name or tag.
4. Click to expand the basic information card of the elastic resource pool and view detailed information about the pool.

   These details will include the name of the pool, the user who created it, the date it was created, the VPC CIDR block, and whether IPv6 is enabled. If IPv6 is enabled, the subnet's IPv6 CIDR block will also be displayed.

   Figure 3-5 Basic information about an elastic resource pool



3.3.2 Managing Permissions

Administrators can assign permissions of different operation scopes to users for each elastic resource pool.

Precautions

- The administrator and elastic resource pool owner have all permissions, which cannot be set or modified by other users.
- When you set resource pool permissions for a new user, ensure that the region of the user group to which the user belongs has the Tenant Guest permission. For details about the Tenant Guest permission and how to apply for the permission, see **Creating a User Group and Assigning Permissions** and **System Permissions**.

Procedure

Step 1 In the navigation pane on the left of the DLI console, choose **Resources** > **Resource Pool**.

Step 2 Select the desired elastic resource pool and choose **More** > **Permissions** in the **Operation** column. The **User Permissions** area displays the list of users who have permissions of elastic resource pools.

You can assign permissions to new users, modify permissions for users who already have some permissions of elastic resource pools, and revoke all permissions of a user on a pool.

- Assign permissions to a new user.

  A new user does not have permissions on the elastic resource pool.

  a. Click **Set Permission** in the **Operations** column on **User Permissions** page. The **Set Permission** dialog box is displayed.
  b. Set **Username** to the name of the desired IAM user, and select the required permissions for the user.
  c. Click **OK** to.

     Table 3-5 describes the related parameters.

Figure 3-6 Managing permissions

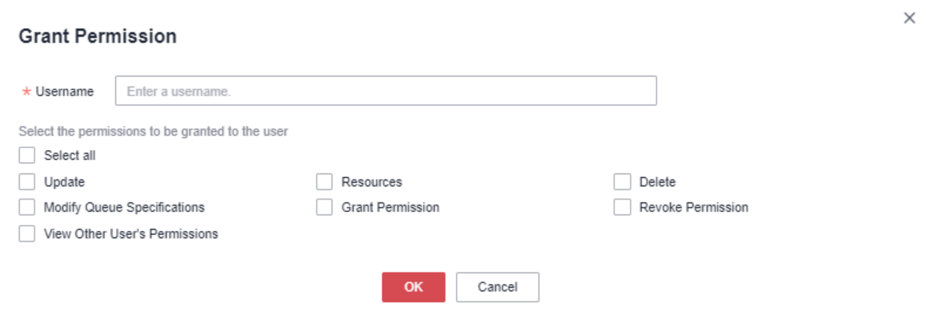


Table 3-5 Parameters

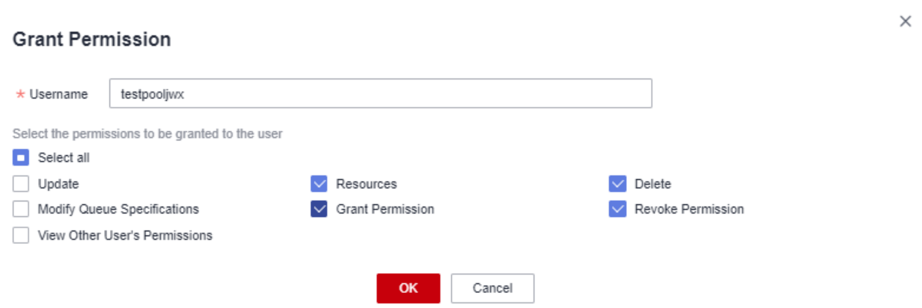
Parameter	Description
Username	Name of the user you want to grant permissions to **NOTE** The username must be an existing IAM username and has logged in to the DLI management console.

Parameter	Description
Select the permissions to be granted to the user	■ Update: Update the description of an elastic resource pool. ■ Resources: Add queues, delete queues, and configure scaling policies for queues in an elastic resource pool. ■ Delete: Delete the elastic resource pool. ■ Modify Specifications: Change the specifications of a yearly/monthly elastic resource pool. ■ Grant Permission: Grant the elastic resource pool permissions to other users. ■ Revoke Permission: Revoke the permissions that other users have but you cannot revoke the owner's permissions. ■ View Other User's Permissions: View the elastic resource pool permissions of other users.

- To assign or revoke permissions of a user who has some permissions on the elastic resource pool, perform the following steps:
  a. In the list under **User Permissions**, select the user whose permissions need to be modified and click **Set Permission** in the **Operation** column.
  b. In the displayed **Set Permission** dialog box, modify the permissions of the user. **Table 3-5** lists the detailed permission descriptions.

     If **Set Permission** is gray, you are not allowed to change permissions on this elastic resource pool. You can apply to the administrator, elastic resource pool owner, or other authorized users for granting and revoking permissions.

     Figure 3-7 Managing permissions

  c. Click **OK**.
- To revoke all permissions of a user on an elastic resource pool, perform the following steps:

In the list under **User Permissions**, select the desired user whose permissions and click **Set Permission** in the **Operation** column. Click **Yes** in the **Revoke Permission** dialog box.

----End

3.3.3 Binding a Queue

Scenario

If you want a queue to use resources in an elastic resource pool, bind the queue to the pool.

You can click **Associate Queue** on the **Resource Pool** page to bind a queue to an elastic resource pool, or bind a queue on the **Queue Management** page.

NOTE

Elastic resource pools support only Flink 1.10 or later. If jobs using Flink 1.7 run on a queue that is bound to an elastic resource pool, errors may occur due to incompatibility.

Prerequisites

- Both the elastic resource pool and queue are available.
- The queue you want to bind must be a dedicated queue in pay-per-use billing mode.
- No resources are frozen.
- Only queues under the same enterprise project can be bound to an elastic resource pool.

Associating a Queue

Step 1 In the navigation pane on the left, choose **Resources** > **Resource Pool**.

Step 2 Locate the row that contains the desired elastic resource pool, click **More** in the **Operation** column, and select **Associate Queue**.

Step 3 In the displayed dialog box, select the desired queue and click **OK**.

----End

Allocating a Queue to an Elastic Resource Pool

Step 1 In the navigation pane on the left, choose **Resources** > **Queue Management**.

Step 2 Locate the target queue and choose **More** > **Bind Resource Pool** in the **Operation** column.

Step 3 Select the desired elastic resource pool and click **OK**.

----End

3.3.4 Setting CUs

CU settings are used to control the maximum and minimum CU ranges for elastic resource pools to avoid unlimited resource scaling.

For example, an elastic resource pool has a maximum of 256 and two queues, and each queue must have at least 64 CUs. If you want to add another queue that needs at lest 256 CUs to the elastic resource pool, the operation is not allowed due to the maximum CUs of the elastic resource pool.

Precautions

- In any time segment of a day, the total minimum CUs of all queues in an elastic resource pool cannot be more than the minimum CUs of the pool.
- In any time segment of a day, the maximum CUs of any queue in an elastic resource pool cannot be more than the maximum CUs of the pool.
- When you change the minimum CUs of a created elastic resource pool, ensure that the value is no more than the current CU value. Otherwise, the modification fails.

Setting CUs

Step 1 In the navigation pane on the left, choose **Resources** > **Resource Pool**.

Step 2 Locate the row that contains the desired elastic resource pool, click **More** in the **Operation** column, and select **Set CUs**.

Step 3 In the **Set CUs** dialog box, set the minimum CUs on the left and the maximum CUs on the right. Click **OK**.

----End

How Do I Increase the Minimum Value of the CU Range?

The minimum value of the CU range of an elastic resource pool is less than or equal to the actual CUs of the elastic resource pool. To adjust the minimum value of the CU range to be greater than the current CUs, increase the actual CUs first.

The following operations apply when the number of target CUs is less than or equal to the maximum value of the CU range. If the number of target CUs is greater than the maximum value of the CU range, increase the maximum CUs of the elastic resource pool first.

- For a yearly/monthly elastic resource pool, you can increase its actual CUs by adjusting the maximum CUs of queues or add queues to the pool. Afterwards, you can modify the specifications to match or exceed the target CUs. This will ensure that the actual CUs match the provided specifications. Finally, set the minimum value of the CU range to be equal to the target CUs.
- For a pay-per-use elastic resource pool, you can increase its actual CUs by adjusting the maximum CUs of queues or add queues to the pool. Afterwards, set the minimum value of the CU range to match the target CUs.

Example

- Example 1: For a yearly/monthly elastic resource pool, the number of actual CUs is 64, the CU range is from 64 to 96, the number of CUs provided by the specifications is 64, and the target CU range is from 80 to 96.

  Procedure

a. You can increase the actual CUs of an elastic resource pool by adjusting the maximum CUs of queues in or adding queues to the elastic resource pool.

   When the total number of maximum CUs of queues in the elastic resource pool is greater than its actual CUs, scale-out is triggered for the elastic resource pool. Actual CUs after scale-out = min (Total number of maximum CUs of queues, Maximum value of the CU range)

b. Change the specifications of the elastic resource pool to 80 CUs. After the change, the minimum value of the CU range is automatically changed to the number of CUs provided by the specifications.

- Example 2: For a pay-per-use elastic resource pool, the number of actual CUs is 64, the CU range is from 64 to 96, and the target CU range is from 80 to 96.

  Procedure

  a. You can increase the actual CUs of an elastic resource pool by adjusting the maximum CUs of queues in or adding queues to the elastic resource pool.

     When the total number of maximum CUs of queues in the elastic resource pool is greater than its actual CUs, scale-out is triggered for the elastic resource pool. Actual CUs after scale-out = min (Total number of maximum CUs of queues, Maximum value of the CU range)

  b. Adjust the CU range to the target one (80 to 96).

NOTE

- The adjustment of the CU range, specifications change, and CU setting of an elastic resource pool take effect on the next hour.
- The adjustment of the actual CUs of an elastic resource pool by adding queues takes effect immediately.

How Do I Decrease the Maximum Value of the CU Range?

The minimum value of the CU range of an elastic resource pool is less than or equal to the actual CUs of the elastic resource pool. To adjust the maximum value of the CU range to be less than the current CUs, decrease the actual CUs first.

- For a yearly/monthly elastic resource pool, you can decrease its actual CUs by adjusting the maximum CUs of queues or delete queues from the pool. Afterwards, you can modify the specifications to be equal to or less than the target CUs. This will ensure that the actual CUs match the provided specifications. Finally, set the minimum value of the CU range to be equal to the target CUs.
- For a pay-per-use elastic resource pool, you can decrease its actual CUs by adjusting the maximum CUs of queues or delete queues from the pool. Afterwards, set the minimum value of the CU range to match the target CUs.

Example

- Example 1: For a yearly/monthly elastic resource pool, the number of actual CUs is 96, the CU range is from 64 to 128, the number of CUs provided by the specifications is 96, and the target CU range is from 64 to 80.

  Procedure

a. You can decrease the actual CUs of an elastic resource pool by deceasing the maximum CUs of queues in or deleting queues from the elastic resource pool.

   When the total number of maximum CUs of queues in the elastic resource pool is less than its actual CUs, scale-in is triggered for the elastic resource pool. Actual CUs after scale-in = min (Total number of maximum CUs of queues, Maximum value of the CU range)

b. Change the specifications of the elastic resource pool to 80 CUs. After the change, the minimum value of the CU range is automatically changed to the number of CUs provided by the specifications.

- Example 2: For a pay-per-use elastic resource pool, the number of actual CUs is 96, the CU range is from 64 to 128, and the target CU range is from 64 to 80.

  Procedure

  a. You can decrease the actual CUs of an elastic resource pool by deceasing the maximum CUs of queues in or deleting queues from the elastic resource pool.

     When the total number of maximum CUs of queues in the elastic resource pool is less than its actual CUs, scale-in is triggered for the elastic resource pool. Actual CUs after scale-in = min (Total number of maximum CUs of queues, Maximum value of the CU range)

  b. Adjust the CU range to the target one (64 to 80).

NOTE

- The adjustment of the CU range, specifications change, and CU setting of an elastic resource pool take effect on the next hour.
- The adjustment of the actual CUs of an elastic resource pool by adding queues takes effect immediately.

3.3.5 Modifying Specifications

Scenario

If the number of CUs of an yearly/monthly elastic resource pool is within the specified range (yearly/monthly CUs), then the resource pool is billed on a yearly/monthly basis. Any excess CUs beyond the specified range (yearly/monthly CUs) are billed by CUH. You can adjust the specifications to make billing more cost-effective based on actual CU usage.

For example, if the current specification for the elastic resource pool (yearly/monthly CUs) is 64 CUs, but most of the time the actual usage exceeds 128 CUs, the 64 CUs will be billed annually/monthly, while the excess 64 CUs will be billed by CUH. To achieve more cost-effective billing, you can change the specification of the elastic resource pool to 128 CUs. Once the specification change is successful, the entire 128 CU range will be billed yearly/monthly, resulting in a more cost-effective billing.

If the current specifications of your elastic resource pool do not meet your service needs, you can modify them using the change yearly/monthly CUs function.

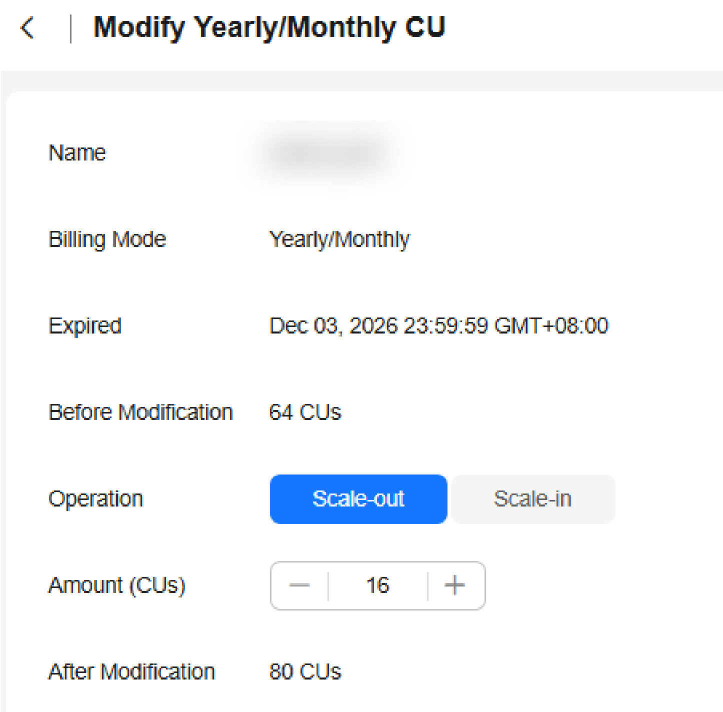
Precautions

Currently, only yearly/monthly elastic resource pools support specification (yearly/monthly CUs) changes.

Scaling Out

1. In the navigation pane on the left of the console, choose **Resources** > **Resource Pool**.
2. Locate the elastic resource pool you want to scale out, click **More** in the **Operation** column, and select **Modify Yearly/Monthly CU**.
3. On the **Modify Yearly/Monthly CU** page, set **Operation** to **Scale-out** and specify the number of CUs you want to add.

Figure 3-8 Scaling out



4. Confirm the changes and click **OK**.
5. Choose **Job Management** > **SQL Jobs** to view the status of the SCALE_POOL SQL job.

   If the job status is **Scaling**, the elastic resource pool is scaling out. Wait until the job status changes to **Finished**.

Scaling In

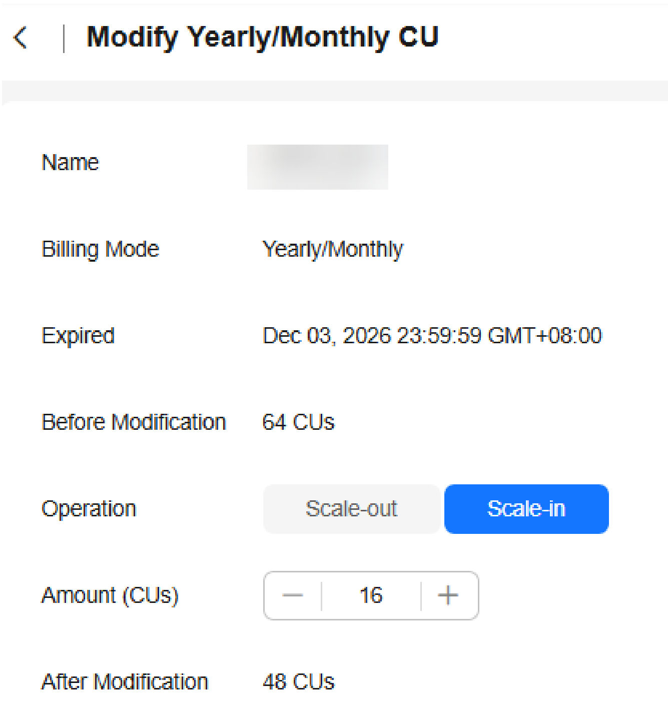
NOTE

By default, the minimum number of CUs is **16**. That is, when the specifications of an elastic resource pool are **16 CUs**, you cannot scale the pool down.

1. In the navigation pane on the left, choose **Resources** > **Resource Pool**.
2. Locate the elastic resource pool you want to scale in, click **More** in the **Operation** column, and select **Modify Yearly/Monthly CU**.

3. On the **Modify Yearly/Monthly CU** page, set **Operation** to **Scale-in** and specify the number of CUs you want to decrease.

Figure 3-9 Scaling in



4. Confirm the changes and click **OK**.
5. Choose **Job Management** > **SQL Jobs** to view the status of the SCALE_POOL SQL job.

   If the job status is **Scaling**, the elastic resource pool is scaling in. Wait until the job status changes to **Finished**.

3.3.6 Managing Tags

Tag Management

A tag is a key-value pair that you can customize to identify cloud resources. It helps you to classify and search for cloud resources. A tag consists of a tag key and a tag value.

If you use tags in other cloud services, you are advised to create the same tag (key-value pairs) for cloud resources used by the same business to keep consistency.

DLI supports the following two types of tags:

- Resource tags: non-global tags created on DLI
- Predefined tags: global tags created on Tag Management Service (TMS).

  For more information about predefined tags, see **Tag Management Service User Guide**.

If your organization has configured tag policies for DLI, add tags to resources based on the policies. If a tag does not comply with the tag policies, resource creation may fail. Contact your organization administrator to learn more about tag policies.

DLI allows you to add, modify, or delete tags for queues.

Step 1 In the left navigation pane of the DLI console, choose **Resources** > **Resource Pool**.

Step 2 In the **Operation** column of the queue, choose **More** > **Tags**.

Step 3 The tag management page is displayed, showing the tag information about the current queue.

Step 4 Click **Add/Edit Tag**. The **Add/Edit Tag** dialog is displayed. Enter a tag and a value, and click **Add**.

Figure 3-10 Adding/Editing tags

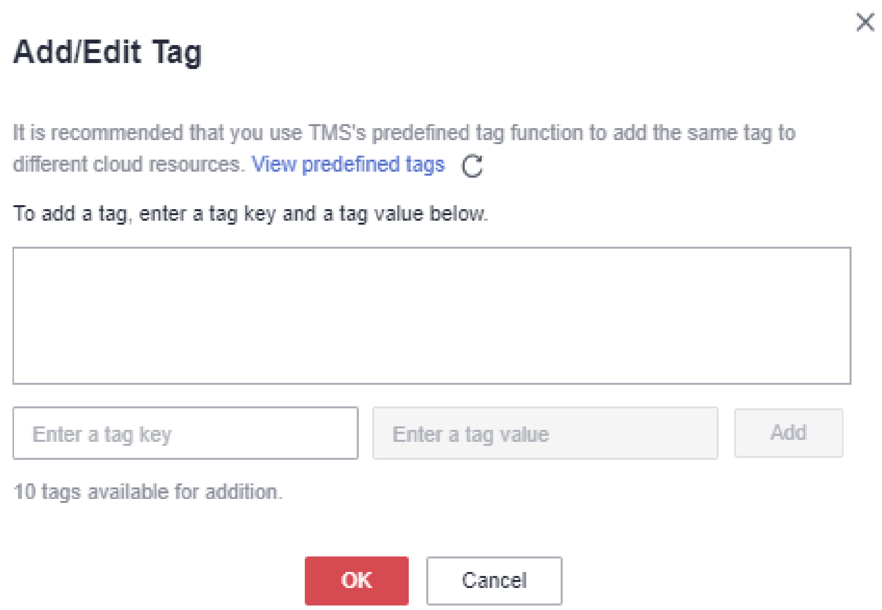


Table 3-6 Tag parameters

Parameter	Description
Tag key	You can specify the tag key in either of the following ways: ● Click the text box and select a predefined tag key from the drop-down list. To add a predefined tag, you need to create one on TMS and then select it from the **Tag key** drop-down list. You can click **View predefined tags** to go to the **Predefined Tags** page of the TMS console. Then, click **Create Tag** in the upper corner of the page to create a predefined tag. For details, see **Creating Predefined Tags** in *Tag Management Service User Guide*. ● Enter a tag key in the text box. **NOTE** A tag key can contain a maximum of 128 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed, but the value cannot start or end with a space or start with **_sys_**.
Tag value	You can specify the tag value in either of the following ways: ● Click the text box and select a predefined tag value from the drop-down list. ● Enter a tag value in the text box. **NOTE** A tag value can contain a maximum of 255 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed. The value cannot start or end with a space.

NOTE

- A maximum of 20 tags can be added.
- Only one tag value can be added to a tag key.
- The key name in each resource must be unique.

Step 5 Click **OK**.

Step 6 (Optional) To delete a tag, locate the row where the tag locates in the tag list and click **Delete** in the **Operation** column to delete the tag.

----End

3.3.7 Adjusting Scaling Policies for Queues in an Elastic Resource Pool

Multiple queues can be added to an elastic resource pool. For how to add a queue, see **Creating an Elastic Resource Pool and Creating Queues Within It**. You can configure the number of CUs you want based on the compute resources used by DLI queues during peaks and troughs and set priorities for the scaling policies to ensure stable running of jobs.

Precautions

- You are advised to implement fine-grained management of resource pools for stream and batch processing jobs by placing Flink real-time stream jobs and SQL batch processing jobs in separate elastic resource pools.

  Flink real-time stream jobs can run stably without forced scale-in, thus avoiding job interruption and system instability.

  SQL batch processing jobs are placed in independent resource pools, which can scale out and in more flexibly, significantly enhancing the success rate and operational efficiency of scaling operations.
- In any time segment of a day, the total minimum CUs of all queues in an elastic resource pool cannot be more than the minimum CUs of the pool.
- In any time segment of a day, the maximum CUs of any queue in an elastic resource pool cannot be more than the maximum CUs of the pool.
- The periods of scaling policies cannot overlap.
- The period of a scaling policy can only be set by hour and specified by the start time and end time. For example, if you set the period to **00-09**, the time range when the policy takes effect is [00:00, 09:00). The period of the default scaling policy cannot be modified.
- In any period, compute resources are preferentially allocated to meet the minimum number of CUs of all queues. The remaining CUs (total CUs of the elastic resource pool – total minimum CUs of all queues) are allocated in accordance with the scaling policy priorities.
- After the queue is scaled out, the system starts billing you for the added CUs. So, if you do not have sufficient requirements, scale in your queue to release unnecessary CUs to save cost.

Table 3-7 CU allocation (without jobs)

Scenario	CUs
An elastic resource pool has a maximum number of 256 CUs for queue A and queue B. The scaling policies are as follows: ● Queue A: priority 5; period: 00:00–09:00; minimum CU: 32; maximum CU: 64 ● Queue B: priority 10; time period: 00:00–09:00; minimum CU: 64; maximum CU: 128	From 00:00 a.m. to 09:00 a.m.: 1. The minimum CUs are allocated to the two queues. Queue A has 32 CUs, and queue B has 64 CUs. There are 160 CUs remaining. 2. The remaining CUs are allocated based on priority. Since queue B has a higher priority than queue A, 64 CUs will be allocated to queue B first, followed by the allocation of 32 CUs to queue A.

Scenario	CUs
An elastic resource pool has a maximum number of 96 CUs for queue A and queue B. The scaling policies are as follows: ● Queue A: priority 5; period: 00:00–09:00; minimum CU: 32; maximum CU: 64 ● Queue B: priority 10; time period: 00:00–09:00; minimum CU: 64; maximum CU: 128	From 00:00 a.m. to 09:00 a.m.: 1. The minimum CUs are allocated to the two queues. Queue A has 32 CUs, and queue B has 64 CUs. There are no remaining CUs. 2. The allocation is complete.
An elastic resource pool has a maximum number of 128 CUs for queue A and queue B. The scaling policies are as follows: ● Queue A: priority 5; period: 00:00–09:00; minimum CU: 32; maximum CU: 64 ● Queue B: priority 10; time period: 00:00–09:00; minimum CU: 64; maximum CU: 128	From 00:00 a.m. to 09:00 a.m.: 1. The minimum CUs are allocated to the two queues. Queue A has 32 CUs, and queue B has 64 CUs. There are 32 CUs remaining. 2. The remaining 32 CUs are preferentially allocated to queue B.
An elastic resource pool has a maximum number of 128 CUs for queue A and queue B. The scaling policies are as follows: ● Queue A: priority 5; period: 00:00–09:00; minimum CU: 32; maximum CU: 64 ● Queue B: priority 5; time period: 00:00–09:00; minimum CU: 64; maximum CU: 128	From 00:00 a.m. to 09:00 a.m.: 1. The minimum CUs are allocated to the two queues. Queue A has 32 CUs, and queue B has 64 CUs. There are 32 CUs remaining. 2. The two queues have the same priority, the remaining 32 CUs are randomly allocated to the two queues.

Table 3-8 CU allocation (with jobs)

Scenario	Actual CUs of Elastic Resource Pool	CUs Allocated to Queue A	CUs Allocated to Queue B	Allocation Description
Queues A and B are added to the elastic resource pool. The scaling policies are as follows: ● Queue A: period: 00:00–09:00; minimum CU: 32; maximum CU: 64 ● Queue B: period: 00:00–09:00; minimum CU: 64; maximum CU: 128	192 CUs	64 CUs	128 CUs	If the actual CUs of the elastic resource pool are greater than or equal to the sum of the maximum CUs of the two queues, the maximum CUs are allocated to both queues.
	96 CUs	32 CUs	64 CUs	The elastic resource pool preferentially meets the minimum CUs of the two queues. After the minimum CUs are allocated to the two queues, no CUs are allocatable.
	128 CUs	32 CUs to 64 CUs	64 CUs to 96 CUs	The elastic resource pool preferentially meets the minimum CUs of the two queues. That is, 32 CUs are allocated to queue A, 64 CUs are allocated to queue B, and the remaining 32 CUs are available. The remaining CUs are allocated based on the queue load and priority. The actual CUs of the queue change within the range listed.

Managing Queues

Step 1 In the navigation pane on the left, choose **Resources** > **Resource Pool**.

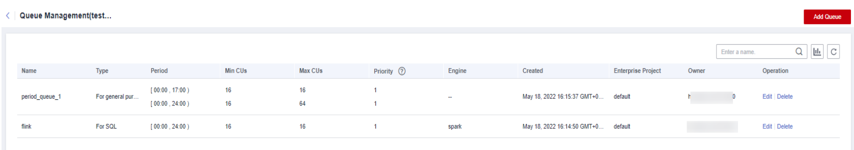
Step 2 Locate the target elastic resource pool and click **Queue MGMT** in the **Operation** column. The **Queue Management** page is displayed.

Step 3 View the queues added to the elastic resource pool.

Table 3-9 Queue parameters

Parameter	Description
Name	Name of the queue to add
Type	Queue type ● For SQL ● For general purpose
Period	The start and end time of the queue scaling policy. This time range includes the start time but not the end time, that is, [start time, end time).
Min CUs	Minimum number of CUs allowed by the scaling policy.
Max CUs	Maximum number of CUs allowed by the scaling policy.
Priority	Priority of the scaling policy for a queue in the elastic resource pool. The priority ranges from 1 to 100. A smaller value indicates a lower priority.
Engine	For a queue running SQL jobs, the engine is Spark. For a queue for general purpose, the engine can be Spark or Flink, but it is displayed by -- in this page.
Created	Time when a queue is added to the elastic resource pool
Enterprise Project	Enterprise project the queue belongs to. Queues under different enterprise projects can be added to an elastic resource pool.
Owner	User who added this queue
Operation	● Edit: Modify or add a scaling policy. ● Delete: Delete the queue.

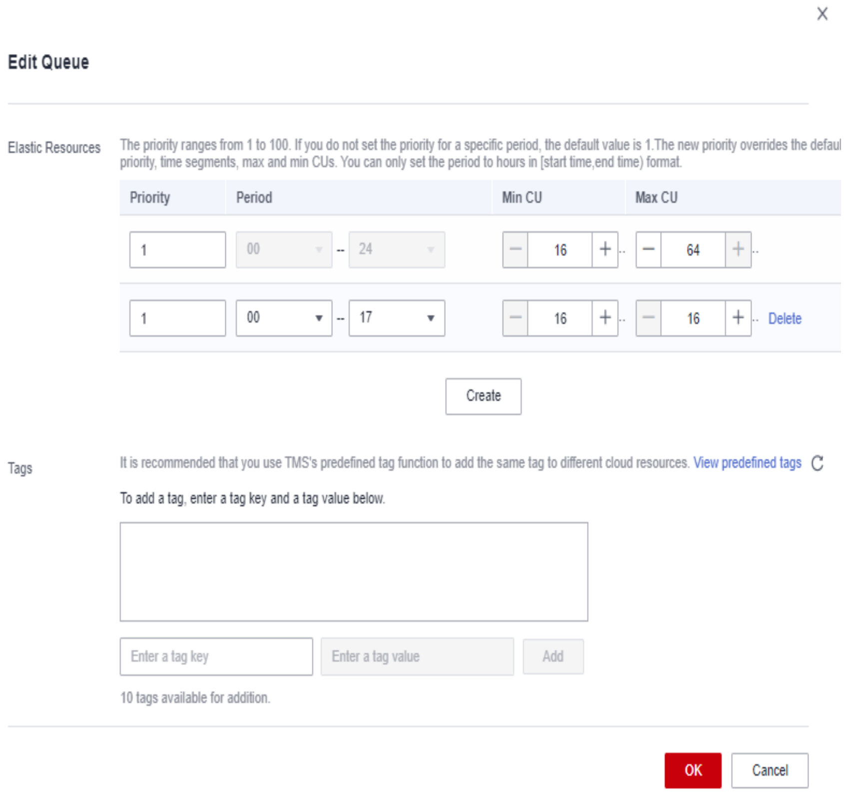
Figure 3-11 Managing queues



Step 4 Locate the target queue and click **Edit** in the **Operation** column.

Step 5 In the displayed **Queue Management** pane, perform the following operations as needed:

Figure 3-12 Editing scaling policies for a queue



- **Add**: Click **Create** to add a scaling policy. Set **Priority**, **Period**, **Min CU**, and **Max CU**, and click **OK**.
- **Modify**: Modify parameters of an existing scaling policy and click **OK**.
- **Delete**: Locate the row that contains the scaling policy you want, click **Delete** and click **OK**.

NOTE

The **Priority** and **Period** parameters must meet the following requirements:

- **Priority**: The default value is **1**. The value ranges from 1 to 100. A larger value indicates a higher priority.
- **Period**:
  - You can only set the period to hours in [start time,end time) format.
  - For example, if the **Period** to **01** and **17**, the scaling policy takes effect at 01:00 a.m. till 05:00 p.m.
  - The periods of scaling policies with different priorities cannot overlap.
- **Max CUs** and **Min CUs**:
  - In any time segment of a day, the total minimum CUs of all queues in an elastic resource pool cannot be more than the minimum CUs of the pool.
  - In any time segment of a day, the maximum CUs of any queue in an elastic resource pool cannot be more than the maximum CUs of the pool.

Step 6 After you finish the settings, click statistics icon in the upper right corner of the queue list to view all scaling policies of all queue in the elastic resource pool.

Figure 3-13 Viewing statistic graphics

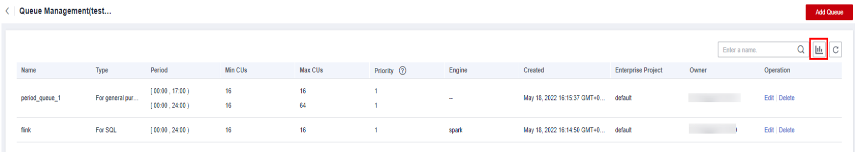


Figure 3-14 Statistic graphics of queues and scaling policies

Step 7 View the scaling task generated when the scaling starts. Go to **Job Management** > **SQL Jobs** and view the jobs of the **SCALE_QUEUE** type.

----End

3.3.8 Viewing Scaling History

Scenario

If you added a queue to or deleted one from an elastic resource pool, or you scaled an added queue, the CU quantity of the elastic resource pool may be changed. You can view historical CU changes of an elastic resource pool on the console.

CAUTION

When scaling in an elastic resource pool, Spark and SQL jobs may automatically retry. However, if the number of retries exceeds the limit, the jobs will fail and need to be manually executed again.

Prerequisites

Currently, you can only view the historical records generated within the last 30 days on the console.

Viewing Scaling History of an Elastic Resource Pool

1. In the navigation pane on the left, choose **Resources** > **Resource Pool**.
2. Select the desired elastic resource pool and choose **More** > **Expansion History** in the **Operation** column.

3. On the displayed page, select a duration to view the CU usage.

   You can view the number of CUs before and after a scaling, and the target number of CUs.

   The historical records can be displayed in charts or tables. Click ☰ in the upper right corner to switch the display.

   For example, the scaling is abnormal according to **Viewing Scaling History**, and the **Figure 3-16** shows that the target number of CUs is 80, the original number of CUs is 64, and the scaled number of CUs is 64. The scaling fails.

Figure 3-15 Scaling history in a chart

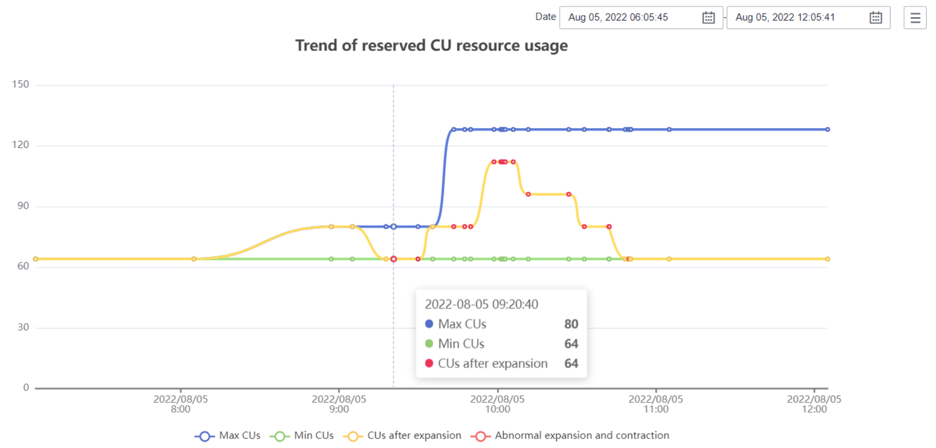
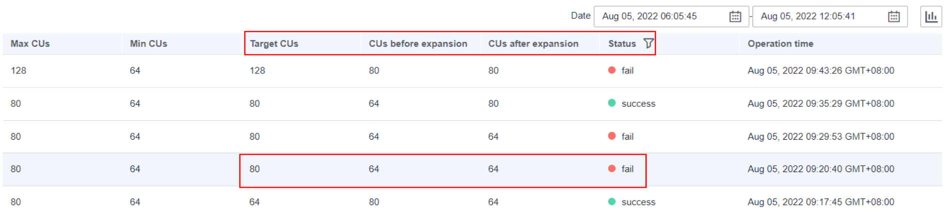


Figure 3-16 Scaling history in a table



Date Aug 05, 2022 06:05:45 - Aug 05, 2022 12:05:41

Max CUs	Min CUs	Target CUs	CUs before expansion	CUs after expansion	Status	Operation time
128	64	128	80	80	fail	Aug 05, 2022 09:43:26 GMT+08:00
80	64	80	64	80	success	Aug 05, 2022 09:35:29 GMT+08:00
80	64	80	64	64	fail	Aug 05, 2022 09:29:53 GMT+08:00
80	64	80	64	64	fail	Aug 05, 2022 09:20:40 GMT+08:00
80	64	64	80	64	success	Aug 05, 2022 09:17:45 GMT+08:00

3.3.9 Allocating to an Enterprise Project

You can create enterprise projects matching the organizational structure of your enterprises to centrally manage cloud resources across regions by project. Then you can create user groups and users with different permissions and add them to enterprise projects.

DLI allows you to select an enterprise project when creating an elastic resource pool. This section describes how to bind an elastic resource pool to and modify an enterprise project.

NOTE

Modifying the enterprise project of an elastic resource pool will modify the enterprise projects of the queues in the elastic resource pool.

Only queues under the same enterprise project can be bound to an elastic resource pool.

Prerequisites

You have logged in to the Enterprise Project Management Service console and created an enterprise project by referring to **Creating an Enterprise Project**.

Binding an Enterprise Project

When creating an elastic resource pool, you can select a created enterprise project for **Enterprise Project**.

Alternatively, you can click **Create Enterprise Project** to go to the Enterprise Project Management Service console to create an enterprise project and check existing ones.

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

Modifying an Enterprise Project

You can modify the enterprise project bound to a created cluster as needed.

1. Log in to the DLI management console.
2. In the navigation pane on the left, choose **Resources** > **Resource Pool**.
3. In the elastic resource pool list, locate the elastic resource pool for which you want to modify the enterprise project, click **More** in the **Operation** column, and select **Allocate to Enterprise Project**.
4. In the **Modify Enterprise Project** dialog box displayed, select an enterprise project.

   Alternatively, you can click **Create Enterprise Project** to go to the Enterprise Project Management Service console to create an enterprise project and check existing ones.
5. After the modification, click **OK** to save the enterprise project information of the elastic resource pool.

Related Operations

For details about how to modify the enterprise project of a queue, see **Allocating a Queue to an Enterprise Project**.

3.4 Managing Queues

3.4.1 Viewing Basic Information About a Queue

This section walks you through how to view basic information about a queue on the management console, including the engine type and version.

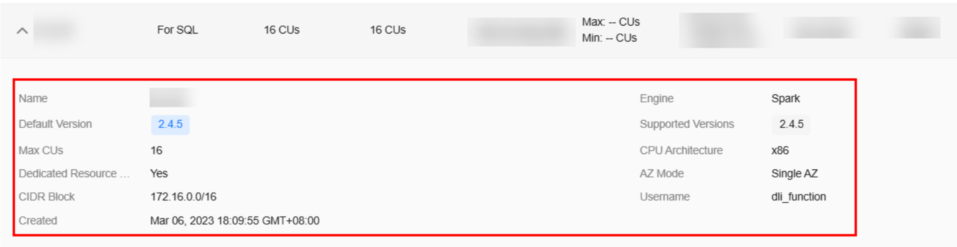
Procedure

1. Log in to the DLI management console.
2. In the navigation pane on the left, choose **Resources** > **Queue Management**.

3. On the displayed page, locate the queue whose basic information you want to view.
   - In the upper right corner of the list page, click to customize the columns to display and set the rules for displaying the table content and the **Operation** column.
   - In the search box above the list, you can filter the required queue by name or tag.
4. Click to view queue details.

   The parameters of the queue are:
   - **Engine**: type of the engine that executes jobs on the queue.
   - **Default Version**: default version of the execution engine or the version the system will use if no specific version is specified.
   - **Supported Versions**: all versions supported by the execution engine. By viewing the supported versions of the queue, you can find out which versions of the execution engine can be used to process jobs on the queue.

   Figure 3-17 Basic queue information



3.4.2 Queue Permission Management

Administrators and queue owners have full operation permissions on queues. They can grant operation permissions to other users based on service needs. This ensures that users can execute their jobs independently without any impact on the performance of other users' job execution. This section describes how to manage queue permissions.

Operation Precautions

- The administrator and queue owner have all permissions, which cannot be set or modified by other users.
- When setting queue permissions for a new user, ensure that the region of the user group to which the user belongs has the **Tenant Guest** permission.

  For details about the **Tenant Guest** permission and how to apply for the permission, see **System Permissions** and **Creating a User Group** in *Identity and Access Management User Guide*.

Operations

Step 1 On the top menu bar of the DLI management console, click **Resources** > **Queue Management**.

Step 2 Select the queue to be configured and choose **Manage Permissions** in the **Operation** column. The **User Permission Info** area displays the list of users who have permissions on the queue.

You can grant queue permissions to new users, modify permissions for users who have some permissions on a queue, and revoke all permissions for a user on a queue.

- **Grant permissions to a new user.**

  A new user does not have permissions on the queue.

  a. Click **Set Permission** on the right of **User Permissions** page. The **Set Permission** dialog box is displayed.

  b. Specify **Username** and select corresponding permissions.

  c. Click **OK**.

  Table 3-10 describes the related parameters.

Figure 3-18 Queue permission granting

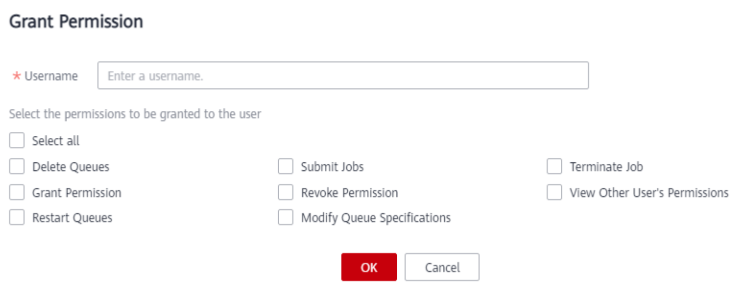


Table 3-10 Parameter description

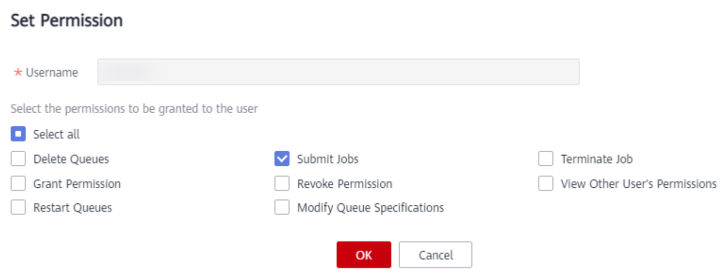
Parameter	Description
Username	Name of the authorized user. **NOTE** The username is an existing IAM user name and has logged in to the DLI management console.

Parameter	Description
Permission Settings	■ **Delete Queues**: This permission allows you to delete the queue. ■ **Submit Jobs**: This permission allows you to submit jobs using this queue. ■ **Terminate Jobs**: This permission allows you to terminate jobs submitted using this queue. ■ **Grant Permission**: This permission allows you to grant queue permissions to other users. ■ **Revoke Permission**: This permission allows you to revoke the queue permissions that other users have but cannot revoke the queue owner's permissions. ■ **View Other User's Permissions**: This permission allows you to view the queue permissions of other users. ■ **Restart Queues**: This permission allows you to restart queues. ■ **Modify Queue Specifications**: This permission allows you to modify queue specifications.

- **To grant or revoke permissions for a user who already has certain permissions on a queue, perform the following steps:**
  a. In the list under **User Permission Info** for a queue, select the user whose permissions need to be modified and click **Set Permission** in the **Operation** column.
  b. In the displayed **Set Permission** dialog box, modify the permissions of the current user. **Table 3-10** lists the detailed permission descriptions.

     If all options under **Set Permission** are gray, you are not allowed to change permissions on this queue. You can apply to the administrator, queue owner, or other authorized users for queue permission granting and revoking.

     Figure 3-19 Setting queue permissions

  c. Click **OK**.

- **To revoke all permissions for a user on a queue, perform the following steps:**

  In the user list under **Permission Info**, select the user whose permission needs to be revoked and click **Revoke Permission** under **Operation**. In the **Revoke Permission** dialog box, click **OK**. All permissions on this queue are revoked.

----End

3.4.3 Allocating a Queue to an Enterprise Project

You can create enterprise projects matching the organizational structure of your enterprises to centrally manage cloud resources across regions by project. Then you can create user groups and users with different permissions and add them to enterprise projects.

DLI allows you to select an enterprise project when creating a queue. This section describes how to bind a DLI queue to and modify an enterprise project.

NOTE

Currently, enterprise projects can be modified only for queues that have not been added to elastic resource pools.

Prerequisites

You have logged in to the Enterprise Project Management Service console and created an enterprise project by referring to **Creating an Enterprise Project**.

Modifying an Enterprise Project

You can change the enterprise project associated with a queue that has been created.

1. Log in to the DLI management console.
2. In the navigation pane on the left, choose **Resources** > **Queue Management**.
3. In the queue list, locate the queue for which you want to modify the enterprise project, click **More** in the **Operation** column, and select **Modify Enterprise Project**.
4. In the **Modify Enterprise Project** dialog box displayed, select an enterprise project.

   Alternatively, you can click **Create Enterprise Project** to go to the Enterprise Project Management Service console to create an enterprise project and check existing ones.
5. After the modification, click **OK** to save the enterprise project information of the queue.

Related Operations

For details about how to modify the enterprise project of an elastic resource pool, see **Allocating to an Enterprise Project**.

3.4.4 Creating an SMN Topic

Scenario

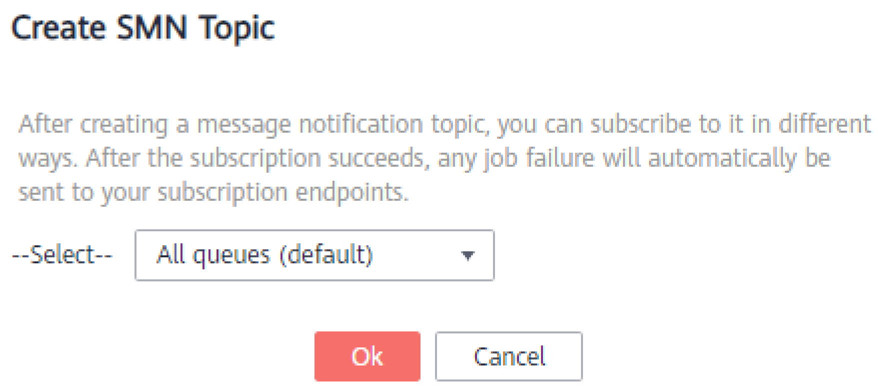
Once you have created an SMN topic, you can easily subscribe to it by going to the **Topic Management** > **Topics** page of the SMN console. You can choose to receive notifications via SMS or email. After the subscription is successful, if a job fails, the system automatically sends a message to the subscription endpoint you specified.

- If a job fails within 1 minute of submission, a message notification is not triggered.
- If a job fails after 1 minute of submission, the system automatically sends a message to the subscriber terminal you specified.

Procedure

1. On the **Resources** > **Queue Management** page, click **Create SMN Topic** on the upper left side. The **Create SMN Topic** dialog box is displayed.

Figure 3-20 Creating an SMN topic

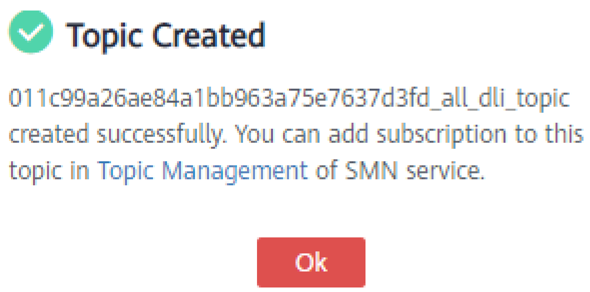


2. Select a queue and click **OK**.

NOTE

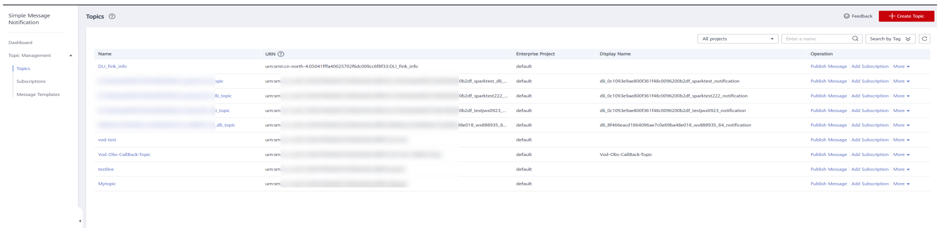
- You can select a single queue or all queues.
- If you create a topic for a queue and another topic for all queues, the SMN of all queues does not include the message of the single queue.
- After a message notification topic is created, you will receive a message notification only when a Spark job created on the subscription queue fails.

Figure 3-21 Successfully created a topic



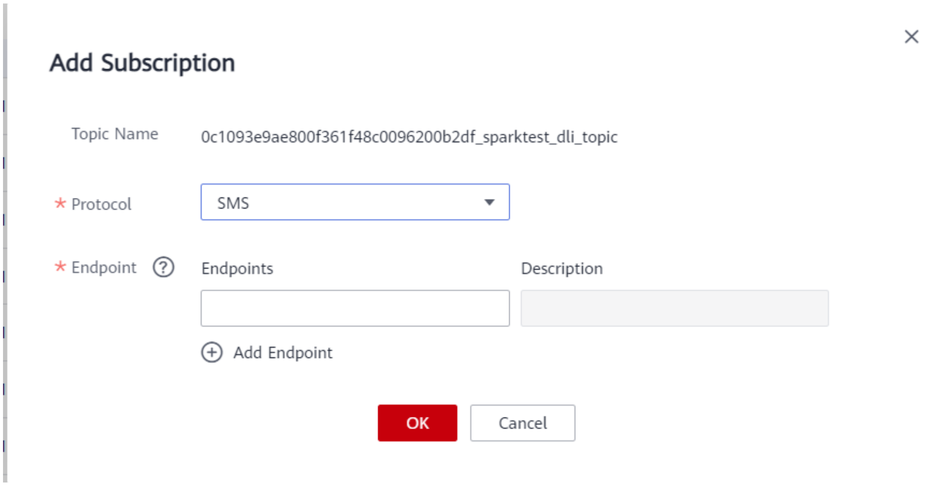
3. Click **Topic Management** in to go to the **Topic Management** page of the SMN service.

Figure 3-22 Topic management page



4. In the **Operation** column of the topic, click **Add Subscription**. Select **Protocol** to determine the subscription mode.
   – If you select **SMS** for the subscription protocol, you need to enter the mobile number for receiving the confirmation SMS message in the **Endpoint** text box.
   – If you select **Email**, enter the email address for receiving the confirmation email in the **Endpoint** text box.

Figure 3-23 Adding a subscription



5. After you click the link in the SMS message or email, you will receive a message indicating that the subscription is successful.
6. Go to the **Subscriptions** page of SMN, and check that subscription status is **Confirmed**.

3.4.5 Managing Queue Tags

Tag Management

A tag is a key-value pair that you can customize to identify cloud resources. It helps you to classify and search for cloud resources. A tag consists of a tag key and a tag value.

If you use tags in other cloud services, you are advised to create the same tag (key-value pairs) for cloud resources used by the same business to keep consistency.

DLI supports the following two types of tags:

- Resource tags: non-global tags created on DLI
- Predefined tags: global tags created on Tag Management Service (TMS).

  For more information about predefined tags, see **Tag Management Service User Guide**.

If your organization has configured tag policies for DLI, add tags to resources based on the policies. If a tag does not comply with the tag policies, resource creation may fail. Contact your organization administrator to learn more about tag policies.

DLI allows you to add, modify, or delete tags for queues.

Step 1 In the navigation pane of the DLI management console, choose **Resources** > **Queue Management**.

Step 2 In the **Operation** column of the queue, choose **More** > **Tags**.

Step 3 The tag management page is displayed, showing the tag information about the current queue.

Step 4 Click **Add/Edit Tag** to switch to the **Add/Edit Tag** dialog box. Enter a tag and a value, and click **Add**.

Figure 3-24 Adding/Editing tags

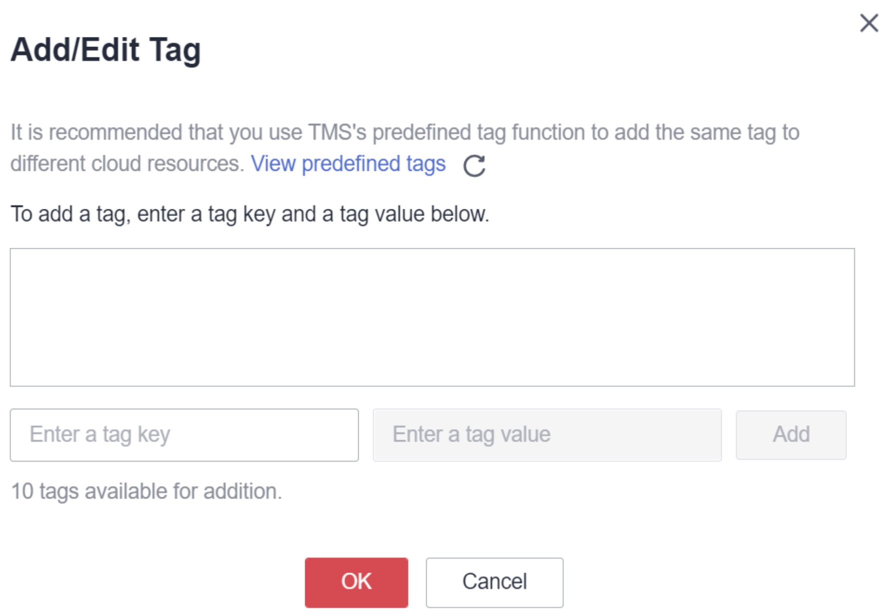


Table 3-11 Tag parameters

Parameter	Description
Tag key	You can specify the tag key in either of the following ways: ● Click the text box and select a predefined tag key from the drop-down list. To add a predefined tag, you need to create one on TMS and then select it from the **Tag key** drop-down list. You can click **View predefined tags** to go to the **Predefined Tags** page of the TMS console. Then, click **Create Tag** in the upper corner of the page to create a predefined tag. For details, see **Creating Predefined Tags** in *Tag Management Service User Guide*. ● Enter a tag key in the text box. **NOTE** A tag key can contain a maximum of 128 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed, but the value cannot start or end with a space or start with **_sys_**.
Tag value	You can specify the tag value in either of the following ways: ● Click the text box and select a predefined tag value from the drop-down list. ● Enter a tag value in the text box. **NOTE** A tag value can contain a maximum of 255 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed. The value cannot start or end with a space.

NOTE

- A maximum of 20 tags can be added.
- Only one tag value can be added to a tag key.
- The key name in each resource must be unique.

Step 5 Click **OK**.

Step 6 (Optional) To delete a tag, locate the row where the tag resides in the tag list and click **Delete** in the **Operation** column to delete the tag.

----End

3.4.6 Setting Queue Properties

Scenario

DLI allows you to set properties for queues.

You can set Spark driver parameters to improve the scheduling efficiency of queues.

This section describes how to set queue properties on the management console.

Notes and Constraints

- Only SQL queues of the Spark engine support configuring queue properties.
- Setting queue properties is only supported after the queue has been created.
- Currently, only queue properties related to the Spark driver can be set.
- Queue properties cannot be set in batches.
- For a queue in an elastic resource pool, if the minimum CUs of the queue is less than 16 CUs, both **Max. Spark Driver Instances** and **Max. Prestart Spark Driver Instances** set in the queue properties do not apply.

Procedure

1. In the navigation pane of the DLI management console, choose **Resources** > **Queue Management**.
2. Locate the queue for which you want to set properties, click **More** in the **Operation** column, and select **Set Property**.
3. Go to the queue property setting page and set property parameters. **Table 3-12** describes the property parameters.

Table 3-12 Queue properties

Property	Description	Value Range
Max. Spark Driver Instances	Maximum number of Spark drivers can be started on this queue, including the Spark driver that is prestarted and the Spark driver that runs jobs.	• For a 16-CU queue, the value is **2**. • For a queue that has more than 16 CUs, the value range is [2, queue CUs/16]. • If the minimum CUs of the queue is less than 16 CUs, this configuration item does not apply.
Max. Prestart Spark Driver Instances	Maximum number of Spark drivers can be prestarted on this queue. When the number of Spark drivers that run jobs exceeds the value of **Max. Concurrency per Instance**, the jobs are allocated to the Spark drivers that are prestarted.	• For a 16-CU queue, the value range is 0 to 1. • For a queue that has more than 16 CUs, the value range is [2, queue CUs/16]. • If the minimum CUs of the queue is less than 16 CUs, this configuration item does not apply.

Property	Description	Value Range
Max. Concurrency per Instance	Maximum number of jobs can be concurrently executed by a Spark driver. When the number of jobs exceeds the value of this parameter, the jobs are allocated to other Spark drivers.	1–32

4. Click **OK**.

3.4.7 Testing Address Connectivity

DLI's address connectivity testing feature can be used to verify network connectivity between DLI queues and destination addresses.

This feature is typically utilized for reading and writing external data sources. Once a datasource connection is configured, the communication capability between the DLI queue and the bound peer address is verified.

Testing the Address Connectivity Between a Queue and the Data Source

1. Log in to the DLI management console. In the navigation pane on the left, choose **Resources** > **Queue Management**.
2. On the **Queue Management** page, locate the row containing the target queue, click **More** in the **Operation** column, and select **Test Address Connectivity**.
3. On the **Test Address Connectivity** page, enter the address to be tested. The domain name and IP address are supported, and the port number can be specified.

   You can input the data source address in the following formats: IPv4 address; IPv4 address + Port number; Domain name; Domain name + Port number.

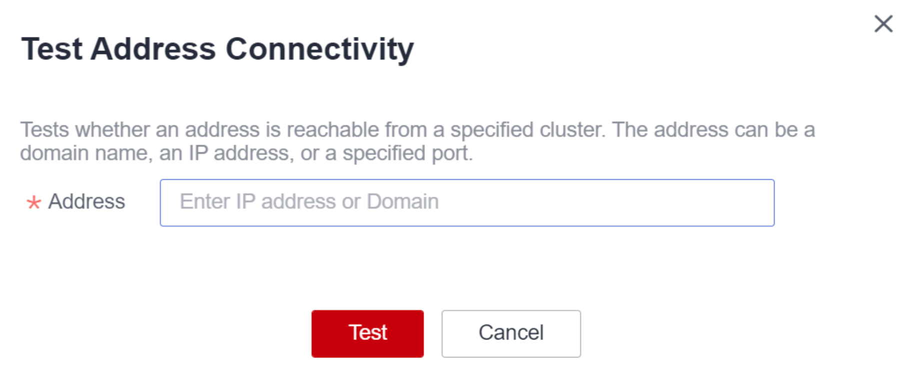
   · IPv4 address: 192.168.x.x

   · IPv4 + Port number: 192.168.x.x:8080

   · Domain name: domain-xxxxxx.com

   · Domain name + Port number: domain-xxxxxx.com:8080

Figure 3-25 Testing address connectivity



4. Click **Test**.
   - If the test address is reachable, a message is displayed on the page, indicating that the address is reachable.
   - If the test address is unreachable, the system displays a message indicating that the address is unreachable. Check the network configurations and retry. Network configurations include the VPC peering and the datasource connection. Check whether they have been activated.

Related Operations

Why Is a Datasource Connection Successfully Created But the Network Connectivity Test Fails?

3.4.8 Deleting a Queue

You can delete a queue based on actual conditions.

NOTE

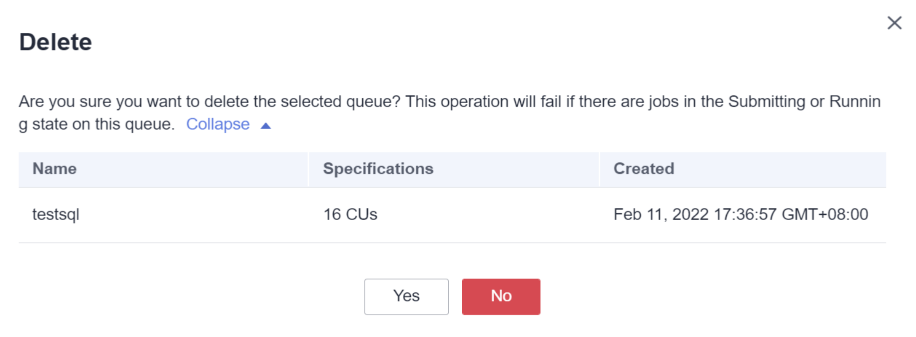
- This operation will fail if there are jobs in the **Submitting** or **Running** state on this queue.
- Deleting a queue does not cause table data loss in your database.

Procedure

Step 1 In the navigation pane on the left of the DLI management console, choose **Resources** > **Queue Management**.

Step 2 Locate the row where the target queue locates and click **Delete** in the **Operation** column.

Figure 3-26 Deleting a Queue



NOTE

If **Delete** in the **Operation** column is gray, the current user does not have the permission of deleting the queue. You can apply to the administrator for the permission.

Step 3 In the displayed dialog box, click **OK**.

----End

3.4.9 Auto Scaling of Standard Queues

Prerequisites

Newly created pay-per-use queues need to run jobs before they can be scaled in or out.

NOTE

The operations described in this section only apply to standard queues.

Notes and Constraints

- Queues with 16 CUs do not support scale-out or scale-in.
- Queues with 64 CUs do not support scale-in.
- Only queues whose billing mode is **Pay-per-use/By CUH** and **Pay-per-use/Dedicated resource mode** support elastic scaling.
- If **Status of queue xxx is assigning, which is not available** is displayed on the **Elastic Scaling** page, the queue can be scaled only after the queue resources are allocated.
- If there are not enough physical resources, a queue may not be able to scale out to the desired target size.
- The system does not guarantee that a queue will be scaled in to the desired target size. Typically, the system checks the resource usage before scaling in the queue to determine if there is enough space for scaling in. If the existing resources cannot be scaled in according to the minimum scaling step, the queue may not be scaled in successfully or only partially.

  The scaling step may vary depending on the resource specifications, usually 16 CUs, 32 CUs, 48 CUs, 64 CUs, etc.

For example, if the queue size is 48 CUs and job execution uses 18 CUs, the remaining 30 CUs do not meet the requirement for scaling in by the minimum step of 32 CUs. If a scaling in task is executed, it will fail.

Scaling Out

If the current queue specifications do not meet service requirements, you can add the number of CUs to scale out the queue.

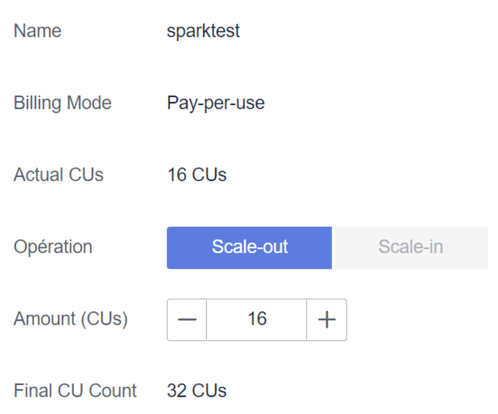
NOTE

Scale-out is time-consuming. After you perform scale-out on the **Elastic Scaling** page of DLI, wait for about 10 minutes. The duration is related to the CU amount to add. After a period of time, refresh the **Queue Management** page and check whether values of **Specifications** and **Actual CUs** are the same to determine whether the scale-out is successful. Alternatively, on the **Job Management** page, check the status of the **SCALE_QUEUE** SQL job. If the job status is **Scaling**, the queue is being scaled out.

The procedure is as follows:

1. In the navigation pane on the left of the DLI management console, choose **Resources** > **Queue Management**.
2. Select the queue to be scaled out, click **More** in the **Operation** column, and select **Elastic Scaling**.
3. On the displayed page, select **Scale-out** for **Operation** and set the scale-out amount.

Figure 3-27 Scale-out



4. Click OK.

Scaling In

If the current queue specifications are too much for your computing service, you can reduce the number of CUs to scale in the queue.

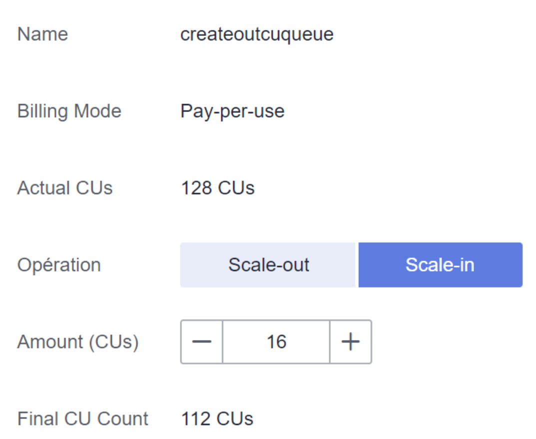
NOTE

- Scale-in is time-consuming. After you perform scale-in on the **Elastic Scaling** page of DLI, wait for about 10 minutes. The duration is related to the CU amount to reduce. After a period of time, refresh the **Queue Management** page and check whether values of **Specifications** and **Actual CUs** are the same to determine whether the scale-in is successful. Alternatively, on the **Job Management** page, check the status of the **SCALE_QUEUE** SQL job. If the job status is **Scaling**, the queue is being scaled in.
- By default, the minimum number of CUs is **16**. That is, when the queue specifications are **16 CUs**, you cannot scale in the queue.

The procedure is as follows:

1. In the navigation pane on the left of the DLI management console, choose **Resources** > **Queue Management**.
2. Select the queue to be scaled in, click **More** in the **Operation** column, and select **Elastic Scaling**.
3. On the displayed page, select **Scale-in** for **Operation** and set the scale-in amount.

Figure 3-28 Manual scale-in



4. Click OK.

3.4.10 Setting a Scheduled Auto Scaling Task for a Standard Queue

Scenario

When services are busy, you might need to use more compute resources to process services in a period. After this period, you do not require the same amount of resources. If the purchased queue specifications are small, resources may be insufficient during peak hours. If the queue specifications are large, resources may be wasted.

DLI provides scheduled tasks for elastic scale-in and -out in the preceding scenario. You can set different queue sizes (CUs) at different time or in different periods based on your service period or usage and the existing queue specifications to meet your service requirements and reduce costs.

NOTE

The operations described in this section only apply to standard queues.

Precautions

- Newly created queues need to run jobs before they can be scaled in or out.
- Scheduled scaling tasks are available only for a queue with more than 64 CUs. That is, the minimum specifications of a queue are 64 CUs.
- A maximum of 12 scheduled tasks can be created for each queue.
- When each scheduled task starts, the actual start time of the specification change has a deviation of 5 minutes. It is recommended that the task start time be at least 20 minutes earlier than the time when the queue is actually used.
- The interval between two scheduled tasks must be at least 2 hours.
- Changing the specifications of a queue is time-consuming. The time required for changing the specifications depends on the difference between the target specifications and the current specifications. You can view the specifications of the current queue on the **Queue Management** page.
- If a job is running in the current queue, the queue may fail to be scaled in to the target CU amount value. Instead, it will be scaled in to a value between the current queue specifications and the target specifications. The system will try to scale in again 1 hour later until the next scheduled task starts.
- If a scheduled task does not scale out or scale in to the target CU amount value, the system triggers the scaling plan again 15 minutes later until the next scheduled task starts.

Creating Periodic Task

- If only scale-out or scale-in is required, you need to create only one task for changing specifications. Set the **Task Name**, **Final CU Count**, and **Executed** parameters. For details, see **Table 3-13**.
- To set both scale-out and scale-in parameters, you need to create two periodic tasks, and set the **Task Name**, **Final CU Count**, and **Executed** parameters. For details, see **Table 3-13**.

The procedure is as follows:

1. In the navigation pane on the left of the DLI management console, choose **Resources** > **Queue Management**.
2. Locate the queue for which you want to schedule a periodic task for elastic scaling, and choose **More** > **Schedule CU Changes** in the **Operation** column.
3. On the displayed page, click **Create Periodic Task** in the upper right corner.
4. On the **Create Periodic Task** page, set the required parameters. Click **OK**.

Figure 3-29 Creating a periodic task

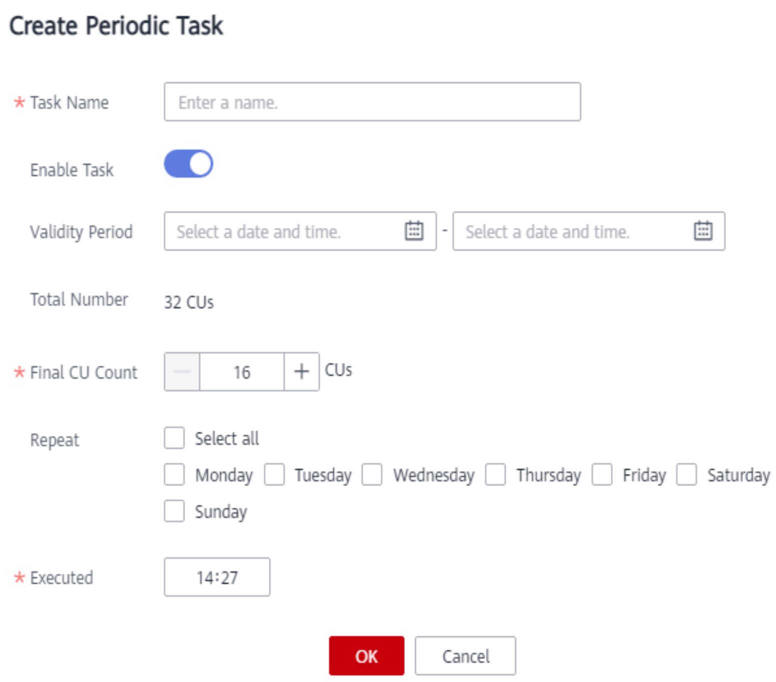


Table 3-13 Parameters

Parameter	Description
Task Name	Enter the name of the periodic task. ● Only numbers, letters, and underscores (_) are allowed. The value cannot contain only numbers or start with an underscore (_) or be left unspecified. ● The value can contain a maximum of 128 characters.
Enable Task	Whether to enable periodic elastic scaling. The task is enabled by default. If disabled, the task will not be triggered on time.

Parameter	Description
Validity Period	Time segment for executing the periodic task. The options include **Date** and **Time**. If there is no time segment restriction, leave this parameter empty, indicating that the task takes effect permanently. If you need to specify the time segment for the task to take effect, set this parameter based on the service requirements. **NOTE** ● The start time of the **Validity Period** must be later than the current system time. ● If only scale-out is configured, the system does not automatically scale in after the **Validity Period** expires. You need to manually modify or configure a periodic scale-in task. Similarly, if only scale-in is configured, you need to manually modify or configure a periodic scale-out task. That is, a scheduled scaling task is executed at a time. ● If both scale-out and scale-in are configured, the system scales in or out resources based on the configured queue specifications within the validity period. After the validity period expires, the system retains the last configured queue specifications.
Actual CUs	Queue specifications before scale-in or scale-out.
Final CUs	Specifications after the queue is scaled in or out. **NOTE** ● By default, the maximum specifications of a queue are **512 CUs**. ● The minimum queue specifications for scheduled scaling are 64 CUs. That is, only when **Actual CUs** are more than 64 CUs, the scheduled scaling can be performed. ● The value of **Actual CUs** must be a multiple of 16.
Repeat	Time when scheduled scale-out or scale-in is repeat. Scheduled tasks can be scheduled by week in **Repeat**. ● By default, this parameter is not configured, indicating that the task is executed only once at the time specified by **Executed**. ● If you select all, the plan is executed every day. ● If you select some options of **Repeat**, the plan is executed once a week at all specified days. **NOTE** ● You do not need to set this parameter if you only need to perform scale-in or scale-out once. ● If you have set scaling, you can set **Repeat** as required. You can also set the repeat period together with the validity period.

Parameter	Description
Executed	Time when scheduled scale-out or scale-in is performed ● When each scheduled task starts, the actual start time of the specification change has a deviation of 5 minutes. It is recommended that the task start time be at least 20 minutes earlier than the time when the queue is actually used. ● The interval between two scheduled tasks must be at least 2 hours.

After a periodic task is created, you can view the specification change of the current queue and the latest execution time on the page for scheduling CU changes.

Alternatively, on the **Queue Management** page, check whether the **Specifications** change to determine whether the scaling is successful.

You can also go to the **Job Management** page and check the status of the **SCALE_QUEUE** job. If the job status is **Scaling**, the queue is being scaled in or out.

Modifying a Scheduled Task

If a periodic task cannot meet service requirements anymore, you can modify it on the **Schedule CU Changes** page.

1. In the navigation pane on the left of the DLI management console, choose **Resources** > **Queue Management**.
2. Locate the queue for which you want to schedule a periodic task for elastic scaling, and choose **More** > **Schedule CU Changes** in the **Operation** column.
3. On the displayed page, click **Modify** in the **Operation** column. In the displayed dialog box, modify the task parameters as needed.

Deleting a Scheduled Task

If you do not need the task anymore, delete the task on the **Schedule CU Changes** page.

1. In the navigation pane on the left of the DLI management console, choose **Resources** > **Queue Management**.
2. Locate the queue for which you want to schedule a periodic task for elastic scaling, and choose **More** > **Schedule CU Changes** in the **Operation** column.
3. On the displayed page, click **Delete** in the **Operation** column. In the displayed dialog box, click **OK**.

3.4.11 Changing the CIDR Block for a Standard Queue

If the CIDR block of the DLI queue conflicts with that of the user data source, you can change the CIDR block of the queue.

If the queue whose CIDR block is to be modified has jobs that are being submitted or running, or the queue has been bound to enhanced datasource connections, the CIDR block cannot be modified.

NOTE

The operations described in this section only apply to standard queues.

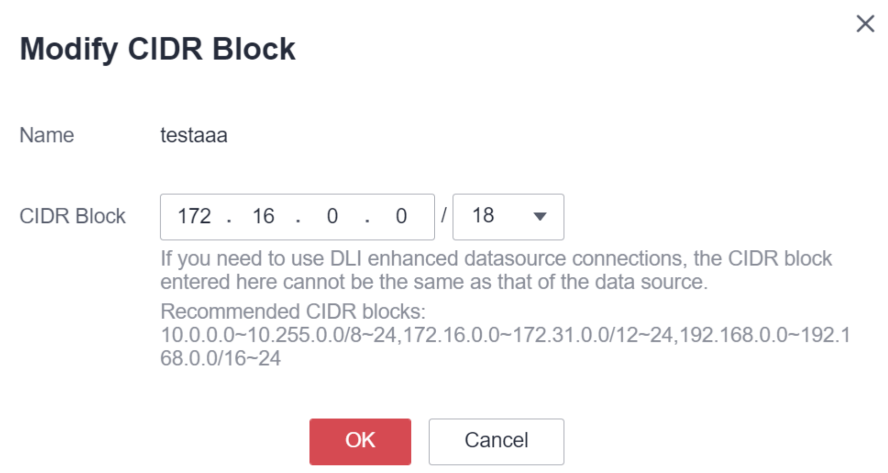
Procedure

NOTE

Currently, you can modify the CIDR block of for queues whose billing mode is **Yearly/Monthly** or **Pay-per-use/Dedicated resource mode**.

1. In the navigation pane on the left of the DLI management console, choose **Resources** > **Queue Management**.
2. Select the queue to be modified and click **Modify CIDR Block** in the **Operation** column.

Figure 3-30 Modifying a CIDR block



3. Enter the required CIDR block and click **OK**. After the CIDR block of the queue is successfully changed, wait for 5 to 10 minutes until the cluster to which the queue belongs is restarted and then run jobs on the queue.

   Recommended CIDR block:

   10.0.0.0–10.255.0.0/8–24

   172.16.0.0–172.31.0.0/12–24

   192.168.0.0–192.168.0.0/16–24

3.5 Example Use Case: Creating an Elastic Resource Pool and Running Jobs

This section walks you through the procedure of adding a queue to an elastic resource pool and binding an enhanced datasource connection to the elastic resource pool.

Figure 3-31 Process of creating an elastic resource pool

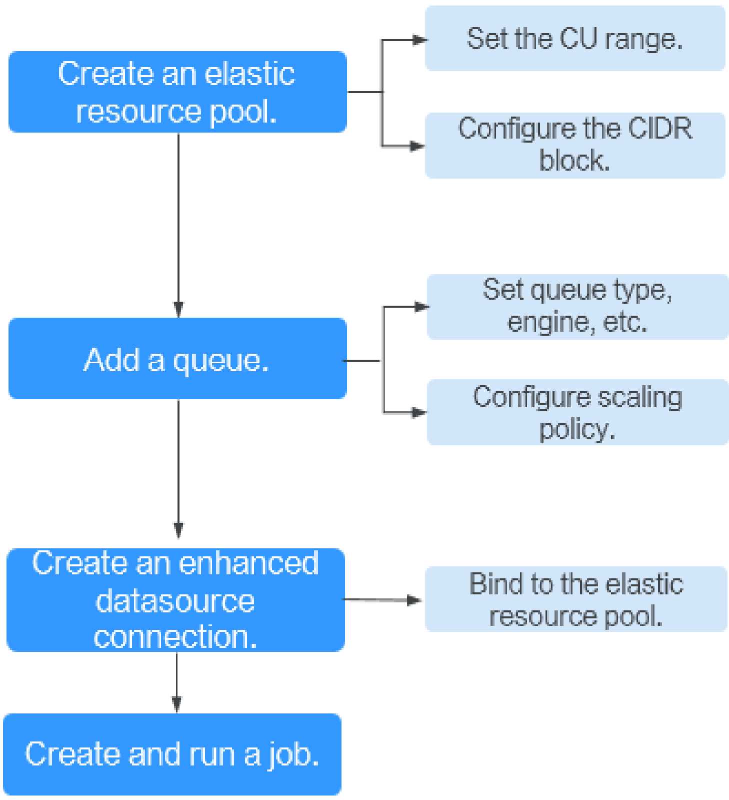


Table 3-14 Procedure

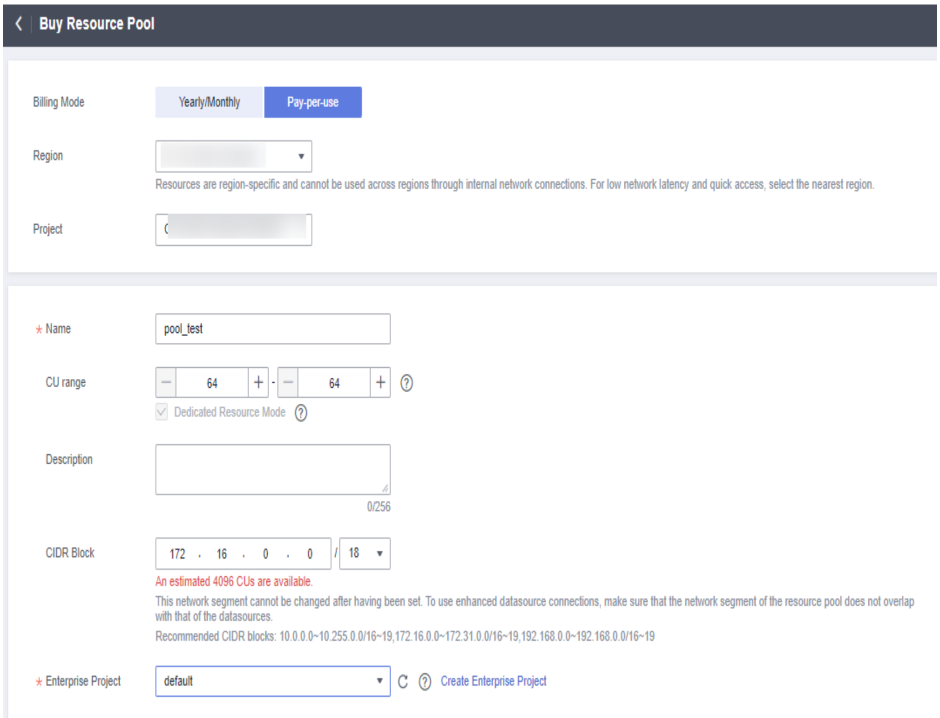
Step	Description	Reference
Create an elastic resource pool	Create an elastic resource pool and configure basic information, such as the billing mode, CU range, and CIDR block.	**Creating an Elastic Resource Pool and Creating Queues Within It**
Add a queue to the elastic resource pool	Add the queue where your jobs will run on to the elastic resource pool. The operations are as follows: 1. Set basic information about the queue, such as the name and type. 2. Configure the scaling policy of the queue, including the priority, period, and the maximum and minimum CUs allowed for scaling.	**Creating an Elastic Resource Pool and Creating Queues Within It** **Adjusting Scaling Policies for Queues in an Elastic Resource Pool**

Step	Description	Reference
(Optional) Create an enhanced datasource connection.	If a job needs to access data from other data sources, for example, GaussDB(DWS) and RDS, you need to create a datasource connection. The created datasource connection must be bound to the elastic resource pool.	**Creating an Enhanced Datasource Connection**
Run a job.	Create and submit the job as you need.	**Managing SQL Jobs** **Flink Job Overview** **Creating a Spark Job**

Step 1: Create an Elastic Resource Pool

1. Log in to the DLI management console. In the navigation pane on the left, choose **Resources** > **Resource Pool**.
2. On the displayed page, click **Buy Resource Pool** in the upper right corner.
3. On the displayed page, set the following parameters:
   - **Name**: Enter the name of the elastic resource pool. For example, **pool_test**.
   - **CU range**: Minimum and maximum CUs of the elastic resource pool.
   - **CIDR Block**: Network segment of the elastic resource pool. For example, **172.16.0.0/18**.
   - Set other parameters as required.

Figure 3-32 Creating an elastic resource pool



For details about how to create an elastic resource pool, see **Creating an Elastic Resource Pool and Creating Queues Within It**.

4. Click **Buy**. Confirm the configuration and click **Pay**.
5. Go to the **Resource Pool** page to view the creation status. If the status is **Available**, the elastic resource pool is ready for use.

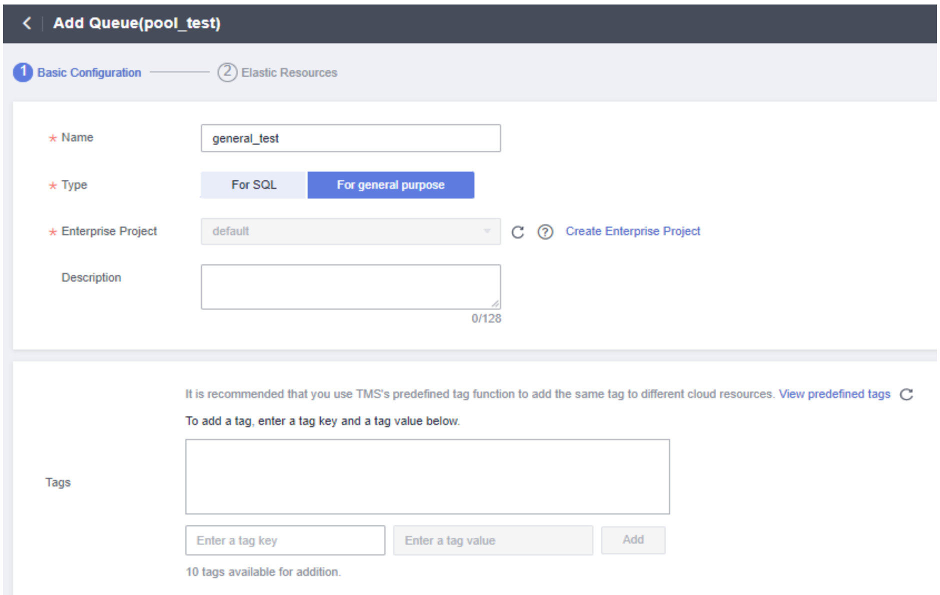
Step 2: Add a Queue to the Elastic Resource Pool

1. In the **Operation** column of the created elastic resource pool, click **Add Queue**.
2. Specify the basic information about the queue. The configuration parameters are as follows:
   - **Name**: Queue name
   - **Type**: Queue type In this example, select **For general purpose**.

     For SQL: The queue is used to run Spark SQL and HetuEngine jobs.

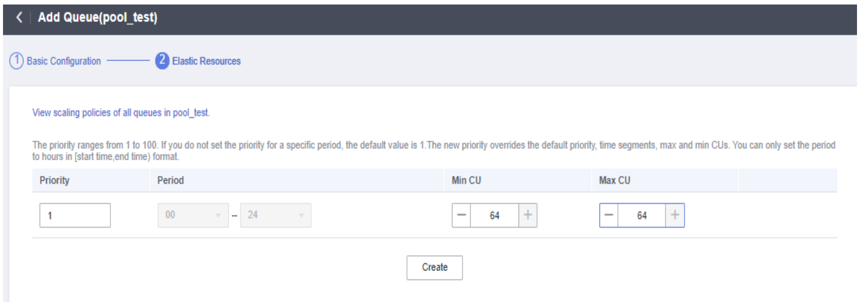
     For general purpose: The queue is used to run Flink and Spark Jar jobs.
   - Set other parameters as required.

Figure 3-33 Creating a queue



3. Click **Next**. On the displayed page, set **Min CU** to **64** and **Max CU** to **64**.

Figure 3-34 Set scaling policy for the queue



4. Click **OK**. The queue is added.

(Optional) Step 3: Create an Enhanced Datasource Connection

In this example, a datasource connection is required to connect to RDS. You need to create a datasource connection. If your job does not need to connect to an external data source, skip this step.

1. Log in to the RDS console and create an RDS DB instance.

   For details, see **Buying an RDS for MySQL DB Instance**.

2. Click **Create Database**. In the dialog box that appears, enter database name **test2**. Then, click **OK**.

3. Locate the row that contains the **test2** database, click **Query SQL Statements** in the **Operation** column. On the displayed page, enter the following statement to create table **tabletest2**. Click **Execute SQL**. The table creation statement is as follows:

```
CREATE TABLE `tabletest2` (
   `id` int(11) unsigned,
   `name` VARCHAR(32)
)   ENGINE = InnoDB    DEFAULT CHARACTER SET = utf8mb4;
```


4. On the RDS console, choose **Instances** form the navigation pane. Click the name of a created RDS DB instance to view its basic information.
5. In the **Connection Information** pane, obtain the floating IP address, database port, VPC, and subnet.
6. Click the security group name. In the **Inbound Rules** tab, add a rule to allow access from the CIDR block of the elastic resource pool. For example, if the CIDR block of the elastic resource pool is **172.16.0.0/18** and the database port is **3306**, set the rule **Priority** to **1**, **Action** to **Allow**, **Protocol** to **TCP** and **Port** to **3306**, Type to **IPv4**, and **Source** to **172.16.0.0/18**.

   Click **OK**. The security group rule is added.
7. Log in to the DLI management console. In the navigation pane on the left, choose **Datasource Connections**. On the displayed page, click **Create** in the **Enhanced** tab.
8. In the displayed dialog box, set the following parameters:
   - **Connection Name**: Name of the enhanced datasource connection
   - **Resource Pool**: Select the elastic resource pool created in **Step 1: Create an Elastic Resource Pool**.

     NOTE

     If you cannot decide the elastic resource pool in this step, you can skip this parameter, go to the **Enhanced** tab, and click **More** > **Bind Resource Pool** in the **Operation** column of the row that contains this datasource connection after it is created.
   - **VPC**: Select the VPC of the RDS DB instance obtained in **5**.
   - **Subnet**: Select the subnet of the RDS DB instance obtained in **5**.
   - Set other parameters as you need.

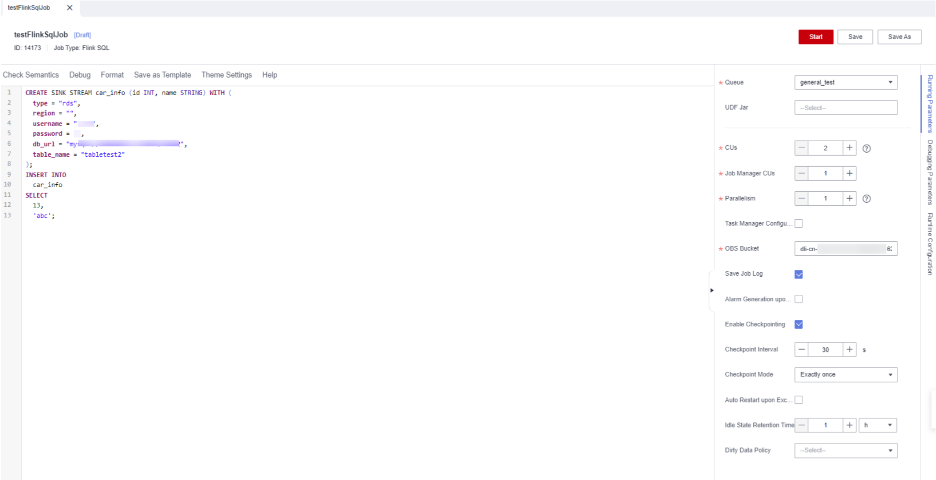
   Click **OK**. Click the name of the created datasource connection to view its status. You can perform subsequent steps only after the connection status changes to **Active**.
9. Click **Resources** > **Queue Management**, select the target queue, for example, **general_test**. In the **Operation** column, click **More** and select **Test Address Connectivity**.
10. In the displayed dialog box, enter *Floating IP address*:*Database port* of the RDS database in the **Address** box and click **Test** to check whether the database is reachable.

Step 4: Run a Job

Run a Flink SQL jab on a queue in an elastic resource pool.

1. On the DLI management console, choose **Job Management** > **Flink Jobs**. On the **Flink Jobs** page, click **Create Job**.
2. In the **Create Job** dialog box, set **Type** to **Flink SQL** and **Name** to **testFlinkSqlJob**. Click **OK**.
3. On the job editing page, set the following parameters:

Figure 3-35 Creating a Flink SQL job

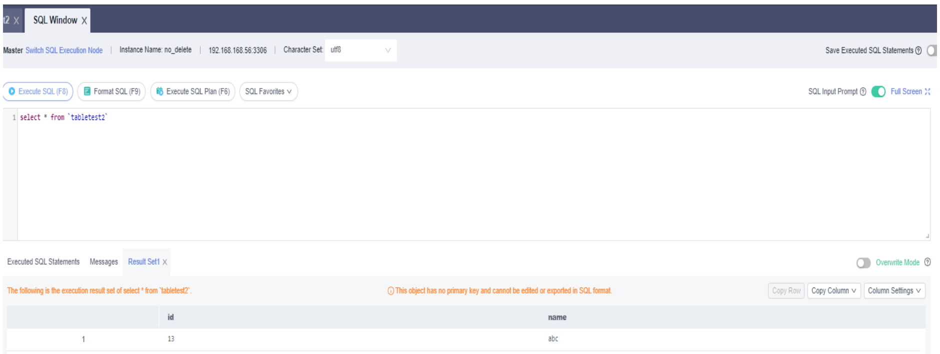


- **Queue**: Select the **general_test** queue added to the elastic resource pool in **Step 2: Add a Queue to the Elastic Resource Pool**.
- **Save Job Log**: Enable this function.
- **OBS Bucket**: Select an OBS bucket for storing job logs and grant access permissions of the OBS bucket as prompted.
- **Enable Checkpointing**: Enable this function.
- Enter the SQL statement in the editing pane. The following is an example. Modify the parameters in bold as you need.

```
CREATE SINK STREAM car_info (id INT, name STRING) WITH (
  type = "rds",
  region = "", /* Change the value to the current region ID. */
   'pwd_auth_name'="xxxxx", // Name of the datasource authentication of the password type created on DLI. If datasource authentication is used, you do not need to set the username and password for the job.
db_url = "mysql://192.168.x.x:3306/test2", /* The format is mysql://floating IP address:port number of the RDS database/database name. */
table_name = "tabletest2" /* Table name in RDS database */
);
INSERT INTO
  car_info
SELECT
  13,
  'abc';
```

4. Click **Check Semantic** and ensure that the SQL statement passes the check. Click **Save**. Click **Start**, confirm the job parameters, and click **Start Now** to execute the job.
5. Wait until the job is complete. The job status changes to **Completed**.
6. Log in to the RDS console, click the name of the RDS DB instance. On the displayed page, click the name of the created database, for example, **test2**, and click **Query SQL Statements** in the **Operation** column of the row that containing the **tabletest2** table.
7. On the displayed page, click **Execute SQL**. Check whether data has been written into the RDS table.

Figure 3-36 Query result



3.6 Example Use Case: Configuring Scaling Policies for Queues in an Elastic Resource Pool

Scenario

A company has multiple departments that perform data analysis in different periods during a day.

- Department A requires a large number of compute resources from 00:00 a.m. to 09:00 a.m. In other time segments, only small tasks are running.
- Department B requires a large number of compute resources from 10:00 a.m. to 10:00 p.m. Some periodical tasks are running in other time segments during a day.

In the preceding scenario, you can add two queues to an elastic resource pool: queue **test_a** for department A, and queue **test_b** for department B. You can add scaling policies for 00:00-09:00 and 10:00-23:00 respectively to the **test_a** and **test_b** queues. For jobs in other periods, you can modify the default scaling policy.

Table 3-15 Scaling policy

Queue	Period	Priority	CUs	Default Period	Default Priority	Default CUs	Remarks
test_a	[00:00, 09:00)	20	Minimum CU: 64 Maximum CU: 128	The time segments beyond [00:00, 09:00)	5	Minimum CU: 16 Maximum CU: 32	Jobs of department A

Queue	Period	Priority	CUs	Default Period	Default Priority	Default CUs	Remarks
test_b	[10:00, 23:00)	20	Minimum CU: 64 Maximum CU: 128	The time segments beyond [10:00, 23:00)	5	Minimum CU: 32 Maximum CU: 64	Jobs of department B

Precautions

- You are advised to implement fine-grained management of resource pools for stream and batch processing jobs by placing Flink real-time stream jobs and SQL batch processing jobs in separate elastic resource pools.

  Flink real-time stream jobs can run stably without forced scale-in, thus avoiding job interruption and system instability.

  SQL batch processing jobs are placed in independent resource pools, which can scale out and in more flexibly, significantly enhancing the success rate and operational efficiency of scaling operations.
- In any time segment of a day, the total minimum CUs of all queues in an elastic resource pool cannot be more than the minimum CUs of the pool.
- In any time segment of a day, the maximum CUs of any queue in an elastic resource pool cannot be more than the maximum CUs of the pool.
- The periods of scaling policies cannot overlap.
- The period of a scaling policy can only be set by hour and specified by the start time and end time. For example, if you set the period to **00-09**, the period when the policy takes effect is [00:00, 09:00). The period of the default scaling policy cannot be modified.
- In any period, compute resources are preferentially allocated to meet the minimum number of CUs of all queues. The remaining CUs (maximum CUs of the elastic resource pool – total minimum CUs of all queues) are allocated in accordance with the scaling policy priorities.
  - Scaling policies with smaller priority values are prior to over those with larger priority values.
  - If the scaling policies of two queues have the same priority, resources are randomly allocated to a queue. If there are remaining resources, they are randomly allocated until there is no more left.

Table 3-16 CU allocation

Scenario	CUs
An elastic resource pool has a maximum number of 256 CUs for queue A and queue B. The scaling policies are as follows: ● Queue A: priority 5; period: 00:00–09:00; minimum CU: 32; maximum CU: 128 ● Queue B: priority 10; time period: 00:00–09:00; minimum CU: 64; maximum CU: 128	From 00:00 a.m. to 09:00 a.m.: 1. The minimum CUs are allocated to the two queues. Queue A has 32 CUs, and queue B has 64 CUs. There are 160 CUs remaining. 2. The remaining CUs are allocated based on the priorities. Queue B is prior to queue A. Therefore, queue B gets 64 CUs, and queue A has 96 CUs.
An elastic resource pool has a maximum number of 96 CUs for queue A and queue B. The scaling policies are as follows: ● Queue A: priority 5; period: 00:00–09:00; minimum CU: 32; maximum CU: 64 ● Queue B: priority 10; time period: 00:00–09:00; minimum CU: 64; maximum CU: 128	From 00:00 a.m. to 09:00 a.m.: 1. The minimum CUs are allocated to the two queues. Queue A has 32 CUs, and queue B has 64 CUs. There are no remaining CUs. 2. The allocation is complete.
An elastic resource pool has a maximum number of 128 CUs for queue A and queue B. The scaling policies are as follows: ● Queue A: priority 5; period: 00:00–09:00; minimum CU: 32; maximum CU: 64 ● Queue B: priority 10; time period: 00:00–09:00; minimum CU: 64; maximum CU: 128	From 00:00 a.m. to 09:00 a.m.: 1. The minimum CUs are allocated to the two queues. Queue A has 32 CUs, and queue B has 64 CUs. There are 32 CUs remaining. 2. The remaining 32 CUs are all preferentially allocated to queue B.
An elastic resource pool has a maximum number of 128 CUs for queue A and queue B. The scaling policies are as follows: ● Queue A: priority 5; period: 00:00–09:00; minimum CU: 32; maximum CU: 64 ● Queue B: priority 5; time period: 00:00–09:00; minimum CU: 64; maximum CU: 128	From 00:00 a.m. to 09:00 a.m.: 1. The minimum CUs are allocated to the two queues. Queue A has 32 CUs, and queue B has 64 CUs. There are 32 CUs remaining. 2. The two queues have the same priority, the remaining 32 CUs are randomly allocated to the two queues.

Setting a Scaling Policy

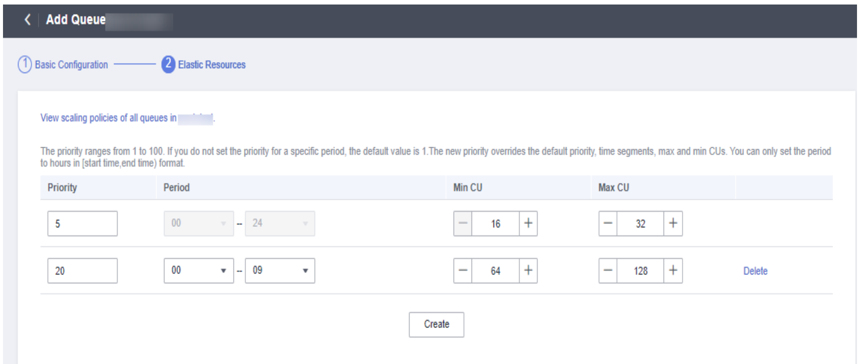
Step 1 Log in to the DLI management console and create an elastic resource pool. Set the minimum and maximum number of CUs of the pool to 128 and 256 respectively. For details, see **Creating an Elastic Resource Pool and Creating Queues Within It**.

Step 2 Choose **Resources** > **Resource Pool**. Locate the row that contains the created elastic resource pool, and click **Queue MGMT** in the **Operation** column.

Step 3 Refer to **Creating an Elastic Resource Pool and Creating Queues Within It** to create the **test_a** queue and set the scaling policy.

1. Set the priority of the default scaling policy to 5, **Min CU** to **16**, and **Max CU** to **32**.
2. Click create to add a scaling policy. Set the priority to **20**, **Period** to **00--09**, **Min CU** to **64**, and **Max CU** to **128**.

Figure 3-37 Adding a scaling policy for queue test_a



Step 4 View the scaling policy on the **Queue Management** page of the specific elastic resource pool.

Figure 3-38 Viewing the new scaling policy

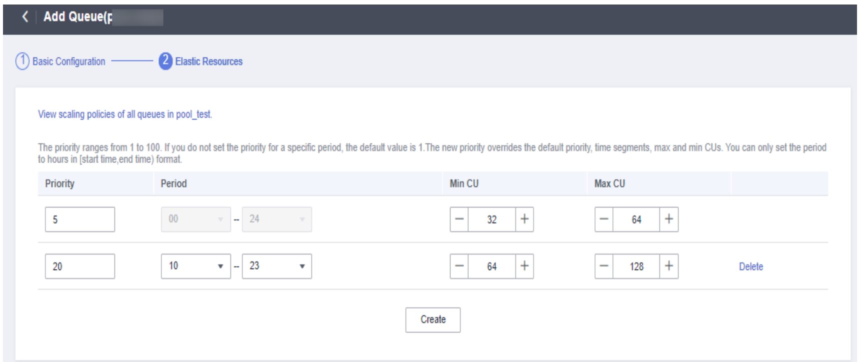
Click  to view graphical statistics of priorities and CU settings for all time segments.

Figure 3-39 Scaling policies

Step 5 Refer to **Creating an Elastic Resource Pool and Creating Queues Within It** to create the **test_b** queue and set the scaling policy.

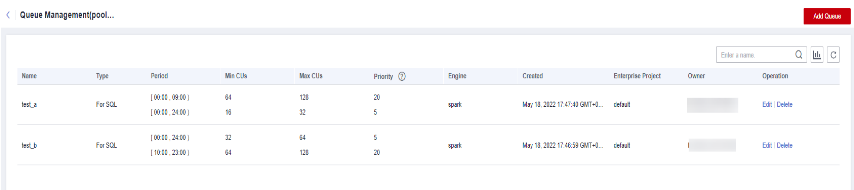
1. Set the priority of the default scaling policy to **5**, **Min CU** to **32**, and **Max CU** to **64**.
2. Click create to add a scaling policy. Set the priority to **20**, **Period** to **10--23**, **Min CU** to **64**, and **Max CU** to **128**.

Figure 3-40 Adding a scaling policy for queue test_b



Step 6 View the scaling policy on the **Queue Management** page of the specific elastic resource pool.

Figure 3-41 Viewing the new scaling policy



Click  to view graphical statistics on priorities and CU settings of the two queues for all time segments.

Figure 3-42 Scaling policies of both queues

----End

3.7 Creating a Non-Elastic Resource Pool Queue (Discarded and Not Recommended)

Queues in the non-elastic resource pool mode are the previous-gen of resource management for DLI. It involved purchasing and releasing resources based on usage demands, requiring estimation of resource needs before making purchases.

Queues in an elastic resource pool are recommended, as they offer the flexibility to use resources with high utilization as needed. For how to buy an elastic resource pool and create queues within it, see **Creating an Elastic Resource Pool and Creating Queues Within It**.

NOTE

- If you use a sub-account to create a queue for the first time, log in to the DLI management console using the main account and keep records in the DLI database before creating a queue.
- It takes 6 to 10 minutes for a job running on a new queue for the first time.
- After a queue is created, if no job is run within one hour, the system releases the queue.
- Queues with 16 CUs do not support scale-out or scale-in.
- Queues with 64 CUs do not support scale-in.

Notes and Constraints

- A queue named **default** is preset in DLI for you to experience. Resources are allocated on demand. You are billed based on the amount of data scanned in each job (unit: GB).
- Queue types:
  - For SQL: Spark SQL jobs can be submitted to SQL queues.
  - For general purpose: The queue is used to run Spark programs, Flink SQL jobs, and Flink Jar jobs.

  The queue type cannot be changed. If you want to use another queue type, purchase a new queue.
- The billing mode of a queue cannot be changed.

- The region of a queue cannot be changed.
- Queues with 16 CUs do not support scale-out or scale-in.
- Queues with 64 CUs do not support scale-in.
- When creating a queue, you can only select cross-AZ active-active for yearly/monthly queues and pay-per-use dedicated queues. The price of a cross-AZ queue is twice that of a single-AZ queue.
- Newly created queues need to run jobs before they can be scaled in or out.
- DLI queues cannot access the Internet.

Procedure

1. You can create a queue on the **Overview**, **SQL Editor**, or **Queue Management** page.
   - In the upper right corner of the **Overview** page, click Purchase Queue.
   - To create a queue on the **Queue Management** page:
     i. In the navigation pane on the left of the DLI management console, choose **Resources** > **Queue Management**.
     ii. In the upper right corner of the **Queue Management** page, click **Buy Queue** to create a queue.
   - To create a queue on the **SQL Editor** page:
     i. In the navigation pane of the DLI management console, click **SQL Editor**.
     ii. Click **Queues**. On the tab page displayed, click  on the right to create a queue.
2. On the **Buy Queue** page displayed, set the parameters according to **Table 3-17**.

Table 3-17 Parameters

Parameter	Description
Billing Mode	**Pay-per-use** Billing for CUH used = Number of CUs x Usage duration x Unit price. You are billed for used CUs on an hourly basis (rounded up to the nearest hour). The pay-per-use billing mode is adopted. You are advised to purchase the CUH package to enjoy preferential price.
Region	Select a region. Select a region near you to ensure the lowest latency possible.
Project	Each region corresponds to a project.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Name</td><td>Queue name
Only numbers, letters, and underscores (_) are allowed. The value cannot contain only numbers, start with an underscore (_), or be left unspecified.
The value can contain a maximum of 128 characters.
NOTE
The queue name is case-insensitive. Uppercase letters will be automatically converted to lowercase letters.</td></tr>
<tr><td>Type</td><td>For SQL: compute resources used for SQL jobs.
For general purpose: compute resources used for Spark and Flink jobs.
NOTE
When a dedicated queue is not in use, its resources are still reserved and not released. This means that resources are constantly being held, regardless of whether the queue is being utilized. By using a dedicated queue, resources can be guaranteed to be available whenever jobs are submitted. You can create enhanced datasource connections for a dedicated queue.
When buying a pay-per-use queue, you have the option to choose a dedicated queue. Dedicated queues are billed 24/7 regardless of whether they are used.</td></tr>
<tr><td>AZ Mode</td><td>Available only when Dedicated Resource Mode is selected for Type.
The deployment mode for the DLI queue. Select dual-AZ if you require high availability.
NOTE
Currently, only SQL queues in dedicated resource mode support the dual-AZ policy.
Dual-AZ improves data availability by creating a duplicate queue in the second AZ, but at an increased cost (twice as much as that of single AZ mode).
This is a one-time configuration and cannot be changed later.</td></tr>
<tr><td>Specifications</td><td>The compute nodes' total number of CUs. One CU equals one vCPU and 4 GB of memory. DLI automatically assigns CPU and memory resources to each compute node, and the client does not need to know how many compute nodes are being used.</td></tr>
<tr><td>Enterprise Project</td><td>If the created queue belongs to an enterprise project, you can select the corresponding enterprise project.
Enterprise projects let you manage cloud resources and users by project.
For how to set an enterprise project, see the Enterprise Management User Guide.
NOTE
This parameter is displayed only for users who have enabled the Enterprise Management Service.</td></tr>
</table>

Parameter	Description
Description	Description of the queue to be created. The value can contain a maximum of 128 characters.
Advanced Settings	In the **Queue Type** area, select **Dedicated Resource Mode** and then click **Advanced Settings**. ● **Default**: The system automatically configures the parameter. ● **Custom** **CIDR Block**: Enter a CIDR block range. If DLI enhanced datasource connection is used, the CIDR block of the DLI queue cannot overlap with that of the data source. Recommended CIDR block: 10.0.0.0–10.255.0.0/8–24 172.16.0.0–172.31.0.0/12–24 192.168.0.0–192.168.0.0/16–24 **Queue Type**: When running an AI-related SQL job, select **AI-enhanced**. When running other jobs, select **Basic**.
Tags	Tags used to identify cloud resources. A tag includes the tag key and tag value. If you want to use the same tag to identify multiple cloud resources, that is, to select the same tag from the drop-down list box for all services, you are advised to create predefined tags on the Tag Management Service (TMS). If your organization has configured tag policies for DLI, add tags to resources based on the policies. If a tag does not comply with the tag policies, resource creation may fail. Contact your organization administrator to learn more about tag policies. For details, see **Tag Management Service User Guide**. NOTE ● A maximum of 20 tags can be added. ● Only one tag value can be added to a tag key. ● The key name in each resource must be unique. ● Tag key: Enter a tag key name in the text box. NOTE A tag key can contain a maximum of 128 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed, but the value cannot start or end with a space or start with **_sys_**. ● Tag value: Enter a tag value in the text box. NOTE A tag value can contain a maximum of 255 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed. The value cannot start or end with a space.

3. Click **Buy** to confirm the configuration.

NOTE

When creating a queue for the first time, select **Agree to the above privacy agreements** and click **OK**.

4. Confirm the configuration and click **Submit**.

   If the queue name already exists, the system displays a message indicating that the queue name already exists when you click **Submit**. In this case, click **Previous** to go back to modify the queue name.

5. Once created, you can check and use it on the **Queue Management** page.

   NOTE

   It takes 6 to 10 minutes for a job running on a new queue for the first time.

4 Creating Databases and Tables

4.1 Understanding Data Catalogs, Databases, and Tables

Databases and tables are the basis for developing SQL and Spark jobs. Before running a job, you need to define databases and tables based on your service scenarios.

NOTE

Flink allows for dynamic data types, enabling the definition of data structures at runtime without the need for predefined metadata.

Data Catalog

A data catalog is a metadata management object that can contain multiple databases.

For how to create a database and a table in a DLI data catalog, see **Creating a Database and Table on the DLI Console**.

Database

A database is a repository of data organized, stored, and managed on computer storage devices according to data structures. Databases are typically used to store, retrieve, and manage structured data, consisting of multiple data tables that are interrelated through keys and indexes.

Table

Tables are one of the most important components of a database, consisting of rows and columns. Each row represents a data item, while each column represents a property or feature of the data. Tables are used to organize and store specific types of data, making it possible to query and analyze the data effectively.

A database is a framework, and tables are its essential content. A database contains one or more tables.

You can create databases and tables on the management console or using SQL statements. For details about using SQL statements, see **Creating a Database**, **Creating an OBS Table**, and **Creating a DLI Table**. This section describes how to create a database and a table on the management console.

NOTE

When creating a database or table, you need to grant permissions to other users so that they can view the new database or table.

Table Metadata

Metadata is data used to define the type of data. It primarily describes information about the data itself, including its source, size, format, or other characteristics. In database fields, metadata is used to interpret the content of a data warehouse.

When creating a table, metadata is defined by three columns: column name, type, and column description.

Table Types Supported by DLI

- **DLI table**

  DLI tables are data tables stored in a DLI data lake. They can store structured, semi-structured, and unstructured data.

  Data in DLI tables is stored internally within the DLI service, resulting in better query performance. This makes it suitable for time-sensitive services, such as interactive queries.

  In the navigation pane on the left of the DLI console, choose **Data Management** > **Databases and Tables**. On the displayed page, click the name of a database. In the displayed table list, tables whose **Type** is **MANAGED** are DLI tables.

- **OBS table**

  Data in OBS tables is stored in the OBS service, which is suitable for latency-insensitive services, such as historical data statistics and analysis.

  An OBS table stores data in the form of objects. Each object contains data and related metadata.

  In the navigation pane on the left of the DLI console, choose **Data Management** > **Databases and Tables**. On the displayed page, click the name of a database. In the displayed table list, tables whose **Type** is **EXTERNAL** and **Storage Location** is **OBS** are OBS tables.

- **View table**

  A view table is a virtual table that does not store actual data. Instead, it dynamically generates data based on the defined query logic. Views are typically used to simplify complex queries or provide customized data views for different users or applications.

  A view table can be created based on one or multiple tables, providing a flexible way to display data without affecting the storage and organization of the underlying data.

  In the navigation pane on the left of the DLI console, choose **Data Management** > **Databases and Tables**. On the displayed page, click the name of a database. In the displayed table list, tables whose **Type** is **VIEW** are view tables.

NOTE

A view can only be created using SQL statements. You cannot create a view on the **Create Table** page. Table or view information in a view cannot be modified. Otherwise, the query may fail.

- **Datasource table**

  A datasource table is a data table that can be queried and analyzed across multiple data sources. This type of tables can integrate data from varying data sources and provide a unified data view.

  Datasource tables are typically used in data warehouse and data lake architectures, allowing users to perform complex queries across multiple data sources.

  In the navigation pane on the left of the DLI console, choose **Data Management** > **Databases and Tables**. On the displayed page, click the name of a database. In the displayed table list, tables whose **Type** is **EXTERNAL** and **Storage Location** is not **OBS** are datasource tables.

Notes and Constraints on Databases and Tables

- **Database**
  - **default** is the database built in DLI. You cannot create a database named **default**.
  - DLI supports a maximum of 50 databases.
- **Table**
  - DLI supports a maximum of 5,000 tables.
  - DLI supports the following table types:
    - **MANAGED**: Data is stored in a DLI table.
    - **EXTERNAL**: Data is stored in an OBS table.
    - **View**: A view can only be created using SQL statements.
    - Datasource table: The table type is also **EXTERNAL**.
  - You cannot specify a storage path when creating a DLI table.
- **Data import**
  - Only OBS data can be imported to DLI or OBS.
  - You can import data in CSV, Parquet, ORC, JSON, or Avro format from OBS to tables created on DLI.
  - To import data in CSV format to a partitioned table, place the partition column in the last column of the data source.
  - The encoding format of imported data can only be UTF-8.
- **Data export**
  - Data in DLI tables (whose table type is **MANAGED**) can only be exported to OBS buckets, and the export path must contain a folder.
  - The exported file is in JSON format, and the text format can only be UTF-8.

- Data can be exported across accounts. That is, after account B authorizes account A, account A has the permission to read the metadata and permission information of account B's OBS bucket as well as the read and write permissions on the path. Account A can export data to the OBS path of account B.

Table Management Page

From the **Data Management** page, click the database name or **Tables** in the **Operation** column to switch to the table management page.

The displayed page lists all tables created in the current database. You can view the table type, data storage location, and other information. Tables are listed in chronological order by default, with the most recently created tables displayed at the top.

4.2 Creating a Database and Table on the DLI Console

A database, built on the computer storage device, is a data warehouse where data is organized, stored, and managed based on its structure.

The table is an important part of the database. It consists of rows and columns. Each column functions as a field. Each value in a field (column) represents a type of data.

The database is a framework and the table contains data content. A database has one or more tables.

You can create databases and tables on the management console or using SQL statements. For details about using SQL statements, see **Creating a Database**, **Creating an OBS Table**, and **Creating a DLI Table**. This section describes how to create a database and a table on the management console.

NOTE

A view can be created only by using SQL statements. You cannot create a view on the **Create Table** page.

Precautions

- If a folder and a file have the same name in the OBS directory, the file path is preferred as the path of the OBS table to be created.
- When creating a database or table, you need to grant permissions to other users so that they can view the created database or table. For details, see **Common Operations Supported by DLI System Policy**.

Creating a Database

Step 1 You can create a database on either the **Data Management** page or the **SQL Editor** page.

- To create a database on the **Data Management** page:
  a. On the left of the management console, choose **Data Management** > **Databases and Tables**.

b. In the upper right corner of the **Databases and Tables** page, click **Create Database** to create a database.

- To create a database on the **SQL Editor** page:

  a. On the left of the management console, click **SQL Editor**.

  b. In the navigation pane on the left, click  next to **Databases**.

Step 2 In the displayed **Create Database** dialog box, specify **Name** and **Description** by referring to **Table 4-1**.

Figure 4-1 Creating a database

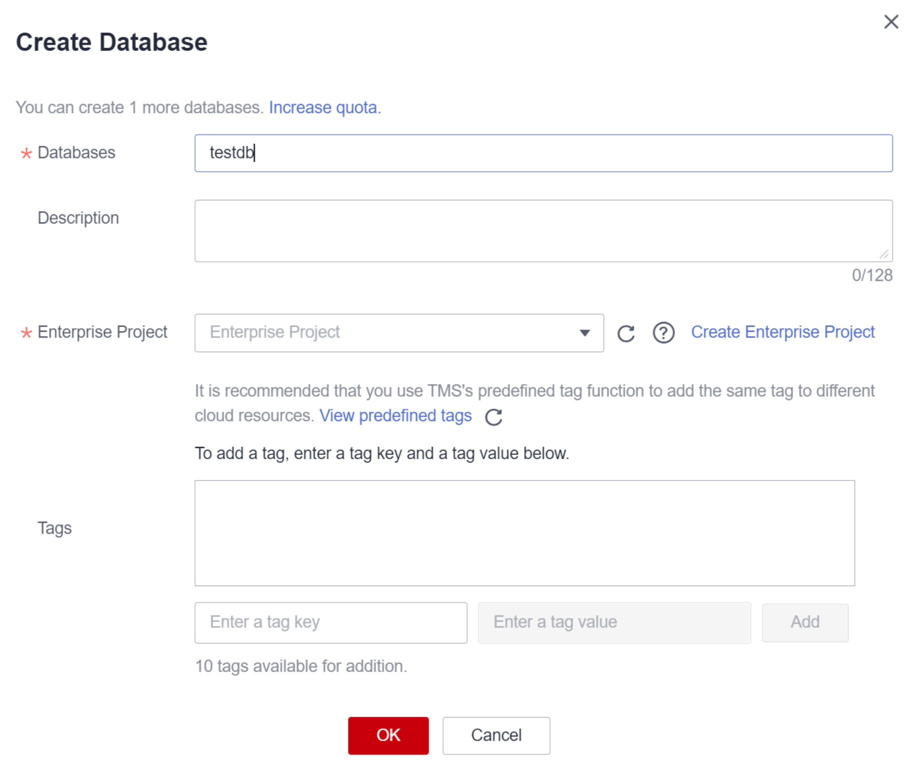


Table 4-1 Description

Parameter	Description
Database Name	• Only numbers, letters, and underscores (_) are allowed. The value cannot contain only numbers or start with an underscore (_). • The database name is case insensitive and cannot be left blank. • The value can contain a maximum of 128 characters. **NOTE** The **default** database is a built-in database. You cannot create the **default**. database.

Parameter	Description
Enterprise Project	If the created queue belongs to an enterprise project, you can select the corresponding enterprise project. Enterprise projects let you manage cloud resources and users by project. For how to set enterprise projects, see the **Enterprise Management User Guide**. **NOTE** This parameter is displayed only for users who have enabled the Enterprise Management Service.
Description	Description of a database.
Tags	Tags used to identify cloud resources. A tag includes the tag key and tag value. If you want to use the same tag to identify multiple cloud resources, that is, to select the same tag from the drop-down list box for all services, you are advised to create predefined tags on the Tag Management Service (TMS). If your organization has configured tag policies for DLI, add tags to resources based on the policies. If a tag does not comply with the tag policies, resource creation may fail. Contact your organization administrator to learn more about tag policies. For details, see **Tag Management Service User Guide**. **NOTE** • A maximum of 20 tags can be added. • Only one tag value can be added to a tag key. • The key name in each resource must be unique. • Tag key: Enter a tag key name in the text box. **NOTE** A tag key can contain a maximum of 128 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed, but the value cannot start or end with a space or start with **_sys_**. • Tag value: Enter a tag value in the text box. **NOTE** A tag value can contain a maximum of 255 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed. The value cannot start or end with a space.

Step 3 Click **OK**.

After a database is created, you can view and select the database for use on the **Databases and Tables** page or **SQL Editor** page.

----End

Creating a Table

Before creating a table, ensure that a database has been created.

Step 1 You can create a table on either the **Databases and Tables** page or the **SQL Editor** page.

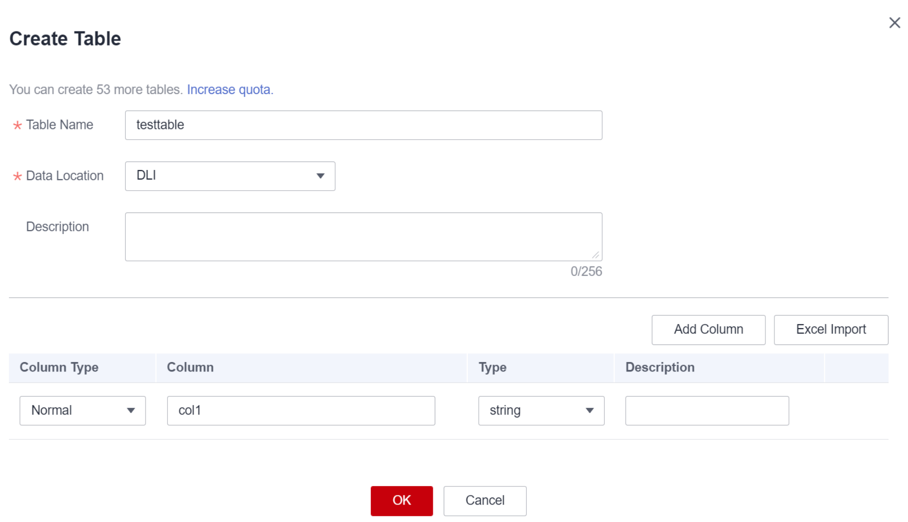
> NOTE
>
> Datasource connection tables, such as View tables, HBase (CloudTable/MRS) tables, OpenTSDB (CloudTable/MRS) tables, GaussDB(DWS) tables, RDS tables, and CSS tables, cannot be created. You can use SQL to create views and datasource connection tables. For details, see the **Data Lake Insight SQL Syntax Reference**.

- To create a table on the **Data Management** page:
  a. On the left of the management console, choose **Data Management** > **Databases and Tables**.
  b. On the **Databases and Tables** page, select the database for which you want to create a table. In the **Operation** column, click **More** > **Create Table** to create a table in the current database.
- To create a table on the **SQL Editor** page:
  a. On the left of the management console, click **SQL Editor**.
  b. In the navigation pane of the displayed **SQL Editor** page, click **Databases**. You can create a table in either of the following ways:
     - Click a database name. In the **Tables** area, click  on the right to create a table in the current database.
     - Click  on the right of the database and choose **Create Table** from the shortcut menu to create a table in the current database.

Step 2 In the displayed **Create Table** dialog box, set parameters as required.

- If you set **Data Location** to **DLI**, set related parameters by referring to **Table 4-2**.

Figure 4-2 Creating a DLI table



- If you set **Data Location** to **OBS**, set related parameters by referring to **Table 4-2** and **Table 4-3**.

Figure 4-3 Creating an OBS table

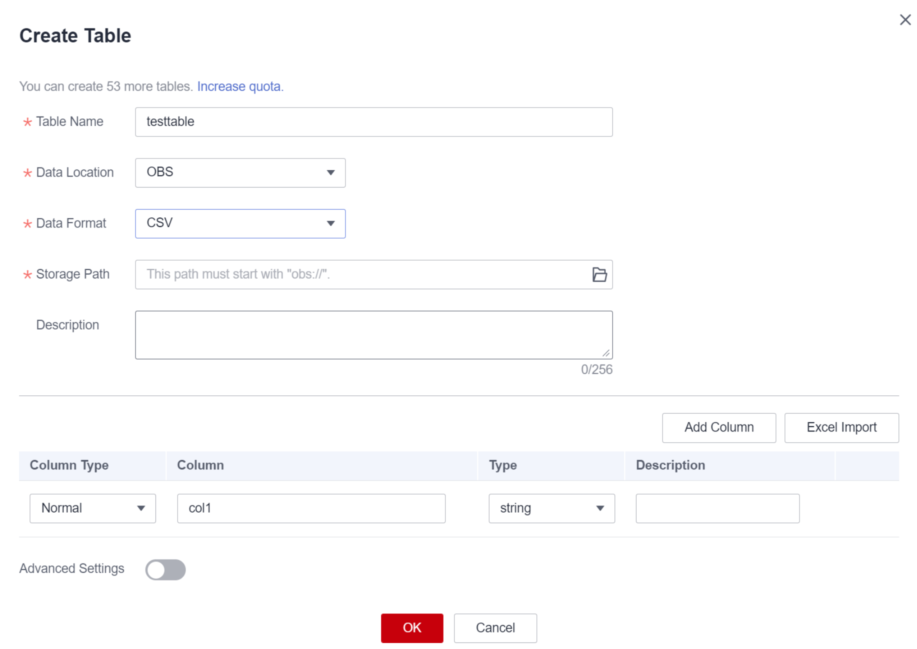


Table 4-2 Common parameters

Parameter	Description	Example
Table Name	– Only numbers, letters, and underscores (_) are allowed. The value cannot contain only numbers or start with an underscore (_). – The table name is case insensitive and cannot be left unspecified. – The table name can contain the dollar sign ($). An example value is **$test**. – The value can contain a maximum of 128 characters.	table01
Data Location	Data storage location. Currently, DLI and OBS are supported.	DLI
Description	Description of the table.	-
Column Type	Available values: **Normal** or **Partition**	Normal

Parameter	Description	Example
Column	Name of a column in a table. The column name must contain at least one letter and can contain underscores (_). It cannot contain only numbers. You can select **Normal** or **Partition**. Partition columns are dedicated to partition tables. User data is partitioned to improve query efficiency. **NOTE** The column name is case-insensitive and must be unique.	name
Type	Data type of a column. This parameter corresponds to **Column Name**. – **string**: The data is of the string type. – **int**: Each integer is stored on four bytes. – **date**: The value ranges from 0000-01-01 to 9999-12-31. – **double**: Each number is stored on eight bytes. – **boolean**: Each value is stored on one byte. – **decimal**: The valid bits are positive integers between 1 to 38, including 1 and 38. The decimal digits are integers less than 10. – **smallint/short**: The number is stored on two bytes. – **bigint/long**: The number is stored on eight bytes. – **timestamp**: The data indicates a date and time. The value can be accurate to six decimal points. – **float**: Each number is stored on four bytes. – **tinyint**: Each number is stored on one byte. Only OBS tables support this data type.	string
Column Description	Description of a column.	-
Operation	– Add Column – Delete **NOTE** If the table to be created includes a great number of columns, you are advised to use SQL statements to create the table or import column information from the local EXCEL file.	-

Table 4-3 Parameter description when **Data Location** is set to **OBS**

Parameter	Description	Example
Data Format	DLI supports the following data formats: – **Parquet**: DLI can read non-compressed data or data that is compressed using Snappy and gzip. – **CSV**: DLI can read non-compressed data or data that is compressed using gzip. – **ORC**: DLI can read non-compressed data or data that is compressed using Snappy. – **JSON**: DLI can read non-compressed data or data that is compressed using gzip. – **Avro**: DLI can read uncompressed Avro data.	CSV
Storage Path	Enter or select an OBS path. The path can be a folder or a path. **NOTE** If you need to import data stored in OBS to the OBS table, set this parameter to the path of a folder. If the table creation path is a file, data fails to be imported.	obs://obs1/sampledata.csv
Table Header: No/Yes	This parameter is valid only when **Data Format** is set to **CSV**. Whether the data source to be imported contains the table header. Click **Advanced Settings** and select the checkbox next to **Table Header: No**. If the checkbox is selected, the table header is displayed. If the checkbox is deselected, no table header is displayed.	-
User-defined Delimiter	This parameter is valid only when **Data Format** is set to **CSV** and you select **User-defined Delimiter**. The following delimiters are supported: – Comma (,) – Vertical bar (|) – Tab character (\t) – Others: Enter a user-defined delimiter.	Comma (,)

Parameter	Description	Example
User-defined Quotation Character	This parameter is valid only when **Data Format** is set to **CSV** and you select **User-defined Quotation Character**. The following quotation characters are supported: – Single quotation mark (') – Double quotation marks (") – Others: Enter a user-defined quotation character.	Single quotation mark (')
User-defined Escape Character	This parameter is valid only when **Data Format** is set to **CSV** and you select **User-defined Escape Character**. The following escape characters are supported: – Backslash (\) – Others: Enter a user-defined escape character.	Backslash (\)
Date Format	This parameter is valid only when **Data Format** is set to **CSV** or **JSON**. This parameter specifies the format of the date in the table and is valid only **Advanced Settings** is selected. The default value is **yyyy-MM-dd**. For definition of characters involved in the date pattern, see Table 3 in **Importing Data to the Table**.	2000-01-01
Timestamp Format	This parameter is valid only when **Data Format** is set to **CSV** or **JSON**. This parameter specifies the format of the timestamp in the table and is valid only **Advanced Settings** is selected. The default value is **yyyy-MM-dd HH:mm:ss**. For definition of characters involved in the time pattern, see Table 3 in **Importing Data to the Table**.	2000-01-01 09:00:00

Step 3 Click **OK**.

After a table is created, you can view and select the table for use on the **Data Management** page or **SQL Editor** page.

Step 4 (Optional) After a DLI table is created, you can decide whether to directly import data to the table.

----End

4.3 Viewing Table Metadata

Metadata Description

- Metadata is used to define data types. It describes information about the data, including the source, size, format, and other data features. In database fields, metadata interprets data content in the data warehouse.
- When you create a table, metadata is defined, consisting of the column name, type, and description.
- The **Metadata** page displays information about the target table, including **Column Name**, **Column Type**, **Data Type**, and **Description**.

Procedure

You can view metadata on either the **Data Management** page or the **SQL Editor** page.

- To view metadata on the **Data Management** page:
  a. On the left of the management console, choose **Data Management** > **Databases and Tables**.
  b. On the displayed **Data Management** page, click the name of the database where the target table whose data you want to export resides to switch to the **Manage Tables** page.
  c. Click **More** in the **Operation** column of the target table and select **View Properties**. In the **Metadata** tab, view the metadata of the table.
- To view metadata on the **SQL Editor** page:
  a. On the left of the management console, click **SQL Editor**.
  b. In the navigation pane of the displayed **SQL Editor** page, click **Databases**.
  c. Click the corresponding database name to view the tables in the database.
  d. Click ≡ on the right of the table and choose **View Properties** from the shortcut menu. On the **Metadata** tab page, view the metadata of the table.

4.4 Managing Database Resources on the DLI Console

4.4.1 Configuring Database Permissions on the DLI Console

Scenario

- By setting permissions, you can assign varying database permissions to different users.
- The administrator and database owner have all permissions, which cannot be set or modified by other users.

- When setting database permissions for a new user, ensure that the region of the user group to which the user belongs has the **Tenant Guest** permission. For details about the Tenant Guest permission and how to apply for the permission, see **Creating a User Group and Assigning Permissions** and **System Permissions**.

Precautions

- By the rules in **Common Operations Supported by DLI System Policy**, you need to grant the current user the permission to view the databases of an administrator or another user.
- Lower-level objects automatically inherit permissions granted to upper-level objects. The hierarchical relationship is database > table > column.
- The database owner, table owner, and **authorized** users can assign permissions on the database and tables.
- Columns can only inherit the query permission. For details about **Inheritable Permissions**, see **Configuring Database Permissions on the DLI Console**.
- The permissions can be revoked only at the initial level to which the permissions are granted. You need to grant and revoke permissions at the same level. You need to grant and revoke permissions at the same level. For example, after you are granted the insertion permission on a database, you can obtain the insertion permission on the tables in the database. Your insertion permission can be revoked only at the database level.
- If you create a database with the same name as a deleted database, the database permissions will not be inherited. In this case, you need to grant the database permissions to users or projects.

  For example, user A is granted with the permission to delete the **testdb** database. Delete the database and create another one with the same name. You need to grant user A the deletion permission of the **testdb** database again.

Viewing Database Permissions

1. On the left of the management console, choose **Data Management** > **Databases and Tables**.
2. Locate the row where the target database resides and click **Manage Permissions** in the **Operation** column.

   Permissions can be granted to new users or projects, modified for users or projects with existing permissions, or revoked from a user or project.

Granting Permissions to a New User or Project

Here, the new user or project refers to a user or a project that does not have permissions on the database.

1. Click a database you need. In the displayed **Database Permission Management** page, click **Grant Permission** in the upper right corner.
2. In the displayed dialog box, select **User** or **Project**, enter the username or select the project to be authorized, and select the required permissions. For details about the permissions, see **Table 4-4**.

Figure 4-4 Granting database permissions to a user

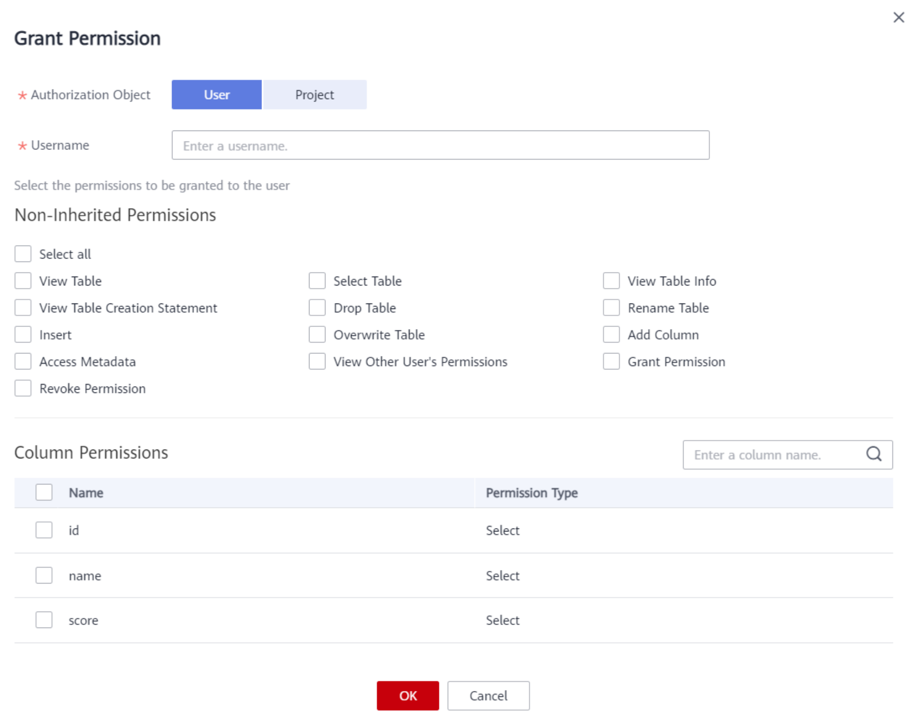


Figure 4-5 Granting database permissions on a project

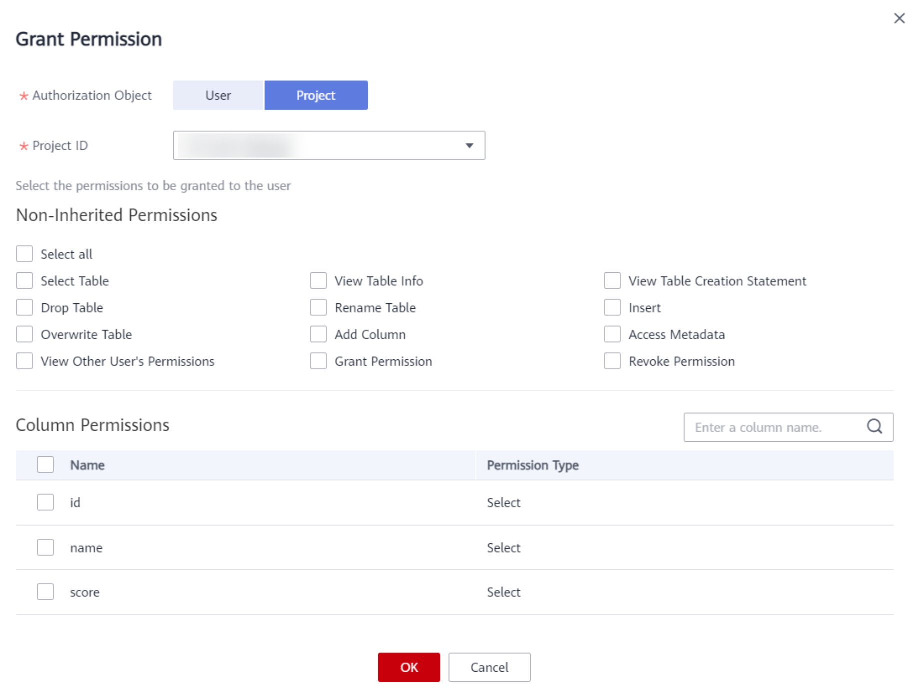


Table 4-4 Parameters

Parameter	Description
Authorization Object	Select **User** or **Project**.
Username/ Project Name	• If you select **User**, enter the IAM username when adding a user to the database. **NOTE** The username is an existing IAM user name and has logged in to the DLI management console. • If you select **Project**, select the project to be authorized in the current region. **NOTE** When **Project** is selected: • If you set **Non-inheritable Permissions**, you cannot view tables in the corresponding database within the project. • If you set **Inheritable Permissions**, you can view all tables in the database within the project.

Parameter	Description
Non-Inherited Permissions	Select a permission to grant it to the user, or deselect a permission to revoke it. Non-inherited permissions apply only to the current database. ● The following permissions are applicable to both user and project authorization: – **Drop Database**: This permission allows you to delete the current database. – **Create Table**: This permission allows you to create tables in the current database. – **Create View**: This permission allows you to create views in the current database. – **Execute SQL EXPLAIN**: This permission allows you to execute an EXPLAIN statement and view information about how this database executes a query. – **Create Role**: This permission allows you to create roles in the current database. – **Delete Role**: This permission allows you to delete roles of the current database. – **View Role**: This permission allows you to view the role of the current user. – **Bind Role**: This permission allows you to bind roles to the current database. – **Unbind Role**: This permission allows you to bind roles from the current database. – **View All Binding Relationships**: This permission allows you to view the binding relationships between all roles and users. – **Create Function**: This permission allows you to create a function in the current database. – **Delete Function**: This permission allows you to delete functions from the current database. – **View All Functions**: This permission allows you to view all functions in the current database. – **View Function Details**: This permission allows you to view details about the current function. ● The following permissions can only be granted to users: – **View All Tables**: This permission allows you to view all tables in the current database. **NOTE** If this permission of a specific database is not granted, all tables in the database will not be displayed. – **View Database**: This permission allows you to view the information about the current database.

Parameter	Description
	NOTE If this permission is not granted, the database will not be displayed.

Parameter	Description
Inherited Permissions	Select a permission to grant it to the user, or deselect a permission to revoke it. Inherited permissions are applicable to the current database and all its tables. However, only the query permission is applicable to table columns. The following permissions can be granted to both user and project. ● **Drop Table**: This permission allows you to delete tables in a database. ● **Select Table**: This permission allows you to query data of the current table. ● **View Table Information**: This permission allows you to view information about the current table. ● **Insert**: This permission allows you to insert data into the current table. ● **Add Column**: This permission allows you to add columns to the current table. ● **Overwrite**: This permission allows you to insert data to overwrite the data in the current table. ● **Grant Permission**: This permission allows you to grant database permissions to other users or projects. ● **Revoke Permission**: This permission allows you to revoke the permissions of the database that other users have but cannot revoke the database owner's permissions. ● **Add Partition to Partition Table**: This permission allows you to add a partition to a partition table. ● **Delete Partition from Partition Table**: This permission allows you to delete existing partitions from a partition table. ● **Configure Path for Partition**: This permission allows you to set the path of a partition in a partition table to a specified OBS path. ● **Rename Table Partition**: This permission allows you to rename partitions in a partition table. ● **Rename Table**: This permission allows you to rename tables. ● **Restore Table Partition**: This permission allows you to export partition information from the file system and save the information to metadata. ● **View All Partitions**: This permission allows you to view all partitions in a partition table. ● **View Other Users' Permissions**: This permission allows you to query other users' permission on the current database.

3. Click **OK**.

Modifying Permissions for an Existing User or Project

For a user or project that has some permissions on the database, you can revoke the existing permissions or grant new ones.

> NOTE
>
> If the options in **Set Permission** are gray, the corresponding account does not have the permission to modify the database. You can apply to the administrator, database owner, or other authorized users for granting and revoking permissions of databases.

1. In the **User Permission Info** list, find the user whose permission needs to be set.
   - If the user is an IAM user, you can set permissions for it.
   - If the user is already an administrator, you can only view the permissions information.

   In the **Project Permission Info** list, locate the project for which you want to set permissions and click **Set Permission**.
2. In the **Operation** column of the IAM user or project, click **Set Permission**. The **Set Permission** dialog box is displayed.

   For details about the permissions of database users or projects, see **Table 4-4**.
3. Click **OK**.

Revoking All Permissions of a User or Project

Revoke all permissions of a user or a project.

- In the user list under **User Permission Info**, locate the row where the target IAM user resides and click **Revoke Permission** in the **Operation** column. In the displayed dialog box, click **OK**. In this case, the user has no permissions on the database.

  > NOTE
  >
  > If a user is an administrator, **Revoke Permission** is gray, indicating that the user's permission cannot be revoked.
- In the **Project Permission Info** area, select the project whose permissions need to be revoked and click **Revoke Permission** in the **Operation** column. After you click **OK**, the project does not have any permissions on the database.

4.4.2 Deleting a Database on the DLI Console

You can delete databases as needed.

Precautions

- You are not allowed to delete databases or tables that are being used for running jobs.
- The administrator, database owner, and users with the database deletion permission can delete the database.

NOTE

If a database or table is deleted, it cannot be recovered. Exercise caution when performing this operation.

Deleting a Database

1. On the left of the management console, choose **Data Management** > **Databases and Tables**.
2. Locate the row where the target database locates and click **More** > **Drop Database** in the **Operation** column.

   NOTE

   You cannot delete databases that contain tables. To delete a database containing tables, delete the tables first.
3. In the displayed dialog box, click **Yes**.

4.4.3 Changing the Database Owner on the DLI Console

In practical use, developers create databases and tables, which are then handed over to testers for testing. Once testing is complete, the databases and tables are handed over to O&M personnel for experience. In this scenario, ownership of the data can be transferred to another owner by changing the database owner.

Changing the Database Owner

You can change the owner of a database on either the **Data Management** page or the **SQL Editor** page.

- On the **Data Management** page, change the database owner.
  a. On the left of the management console, choose **Data Management** > **Databases and Tables**.
  b. On the **Databases and Tables** page, locate the database you want and click **More** > **Modify Database** in the **Operation** column.
  c. In the displayed dialog box, enter a new owner name (an existing username) and click **OK**.
- Change the database owner on the **SQL Editor** page.
  a. On the left of the management console, click **SQL Editor**.
  b. In the navigation tree on the left, click **Databases**, click on the right of the database you want to modify, and choose **Modify Database** from the shortcut menu.
  c. In the displayed dialog box, enter a new owner name (an existing username) and click **OK**.

4.4.4 Managing Tags

Tag Management

A tag is a key-value pair that you can customize to identify cloud resources. It helps you to classify and search for cloud resources. A tag consists of a tag key and a tag value. If you use tags in other cloud services, you are advised to create the same tag (key-value pairs) for cloud resources used by the same business to keep consistency.

If your organization has configured tag policies for DLI, add tags to resources based on the policies. If a tag does not comply with the tag policies, resource creation may fail. Contact your organization administrator to learn more about tag policies.

DLI supports the following two types of tags:

- Resource tags: non-global tags created on DLI
- Predefined tags: global tags created on Tag Management Service (TMS).

  For more information about predefined tags, see **Tag Management Service User Guide**.

This section describes how to add, modify, and delete tags for databases and tables.

Database Tags

Step 1 In the navigation pane on the left, choose **Data Management** > **Databases and Tables**.

Step 2 Locate the row that contains the target database, and click **More** > **Tags** in the **Operation** column.

Step 3 The tag management page is displayed, and the tags (if there are) are displayed.

Step 4 On the displayed page, click **Add/Edit Tag**. The **Add/Edit Tag** dialog box is displayed.

Enter a tag key and a tag value in the text boxes and click **Add**.

Figure 4-6 Adding/Editing tags

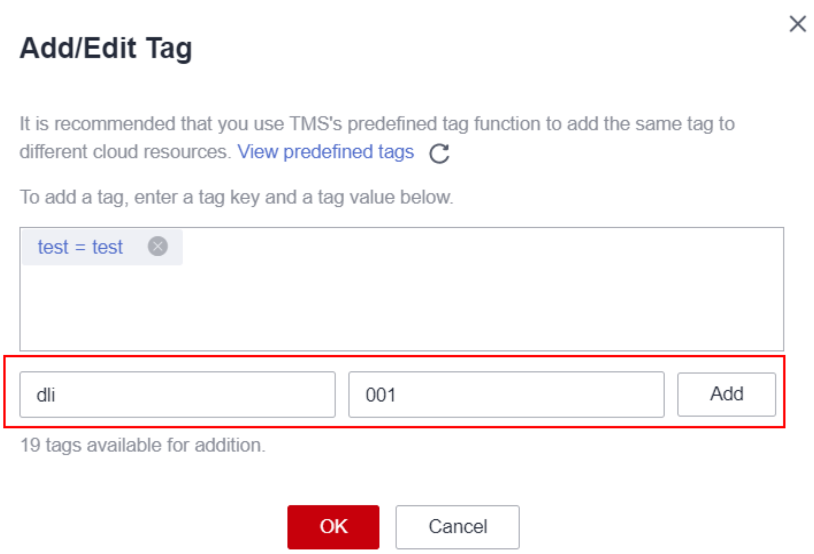


Table 4-5 Tag parameters

Parameter	Description
Tag key	You can specify the tag key in either of the following ways: ● Click the text box for tag key and select a predefined tag key from the drop-down list. To add a predefined tag, you need to create one on TMS and then select it from the **Tag key** drop-down list. You can click **View predefined tags** to go to the **Predefined Tags** page of the TMS console. Then, click **Create Tag** in the upper corner of the page to create a predefined tag. For details, see **Creating Predefined Tags** in *Tag Management Service User Guide*. ● Enter a tag key in the text box. NOTE A tag key can contain a maximum of 128 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed, but the value cannot start or end with a space or start with **_sys_**.
Tag value	You can specify the tag value in either of the following ways: ● Click the tag value text box and select a predefined tag value from the drop-down list. ● Enter a tag value in the text box. NOTE A tag value can contain a maximum of 255 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed. The value cannot start or end with a space.

NOTE

- A maximum of 20 tags can be added.
- Only one tag value can be added to a tag key.
- The key name in each resource must be unique.

Step 5 Click **OK**. The database tag is added.

To delete a tag, click **Delete** in the **Operation** column of the target tag.

----End

Table Tags

Step 1 In the navigation pane on the left, choose **Data Management** > **Databases and Tables**.

Step 2 Click a database name to view the tables in the database.

Step 3 Locate the row that contains the target table and click **More** > **Tag** in the **Operation** column.

Step 4 The tag management page is displayed, and the tags (if there are) are displayed.

Step 5 On the displayed page, click **Add/Edit Tag**. The **Add/Edit Tag** dialog box is displayed.

Enter a tag key and a tag value in the text boxes and click **Add**

Figure 4-7 Adding/Editing tags

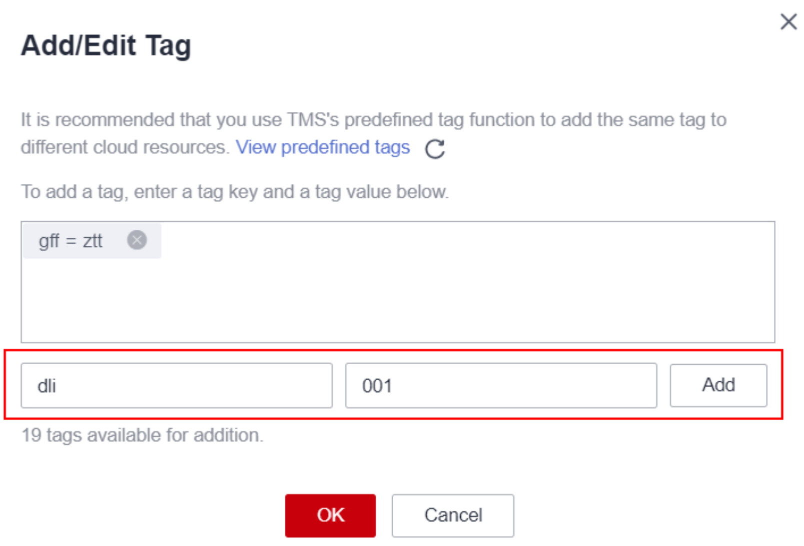


Table 4-6 Tag parameters

Parameter	Description
Tag key	You can specify the tag key in either of the following ways: ● Click the text box for tag key and select a predefined tag key from the drop-down list. To add a predefined tag, you need to create one on TMS and then select it from the **Tag key** drop-down list. You can click **View predefined tags** to go to the **Predefined Tags** page of the TMS console. Then, click **Create Tag** in the upper corner of the page to create a predefined tag. For details, see **Creating Predefined Tags** in *Tag Management Service User Guide*. ● Enter a tag key in the text box. **NOTE** A tag key can contain a maximum of 128 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed, but the value cannot start or end with a space or start with **_sys_**.

Parameter	Description
Tag value	You can specify the tag value in either of the following ways: • Click the tag value text box and select a predefined tag value from the drop-down list. • Enter a tag value in the text box. **NOTE** A tag value can contain a maximum of 255 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed. The value cannot start or end with a space.

NOTE

- A maximum of 20 tags can be added.
- Only one tag value can be added to a tag key.
- The key name in each resource must be unique.

Step 6 Click **OK**. The table tag is added.

To delete a tag, click **Delete** in the **Operation** column of the target tag.

----End

4.5 Managing Table Resources on the DLI Console

4.5.1 Configuring Table Permissions on the DLI Console

Operation Scenario

- By setting permissions, you can assign varying table permissions to different users.
- The administrator and table owner have all permissions, which cannot be set or modified by other users.
- When setting table permissions for a new user, ensure that the region of the user group the user belongs to has the **Tenant Guest** permission. For details about the Tenant Guest permission and how to apply for the permission, see **Permissions Policies** and **Creating a User Group and Assigning Permissions** in the *Identity and Access Management User Guide*.

Precautions

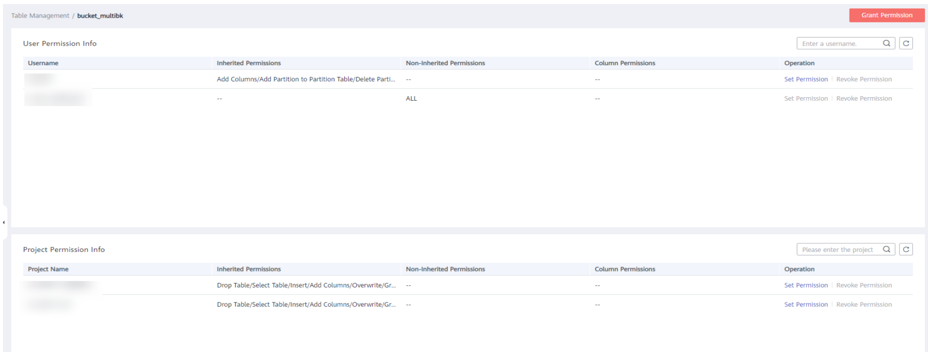
- By the rules in **Common Operations Supported by DLI System Policy**, you need to authorize a user to view tables in a database of the owner account.
- If you create a table with the same name as a deleted table, the table permissions will not be inherited. In this case, you need to grant the table permissions to users or projects.

  For example, user A is granted with the permission to delete the **testTable** table. Delete the table and create another one with the same name. You need to grant user A the deletion permission of the **testTable** table again.

Viewing Table Permissions

1. On the left of the management console, choose **Data Management** > **Databases and Tables**.
2. Click the database name in the table whose authority is to be set. The **Table Management** page of the database is displayed.
3. Locate the row where the target table resides and click **Manage Permissions** in the **Operation** column.

Figure 4-8 Table permission management



Permissions can be granted to new users or projects, modified for users or projects with existing permissions, or revoked from a user or project.

Granting Permissions to a New User or a Project

Here, the new user or project refers to a user or a project that does not have permissions on the database.

1. Click the table you need. In the displayed table permissions page, click **Grant Permission** in the upper right corner.
2. In the displayed **Grant Permission** dialog box, select the required permissions.
   – For details about the DLI table permissions, see **Table 4-7**.

Figure 4-9 Granting DLI table permissions to a user

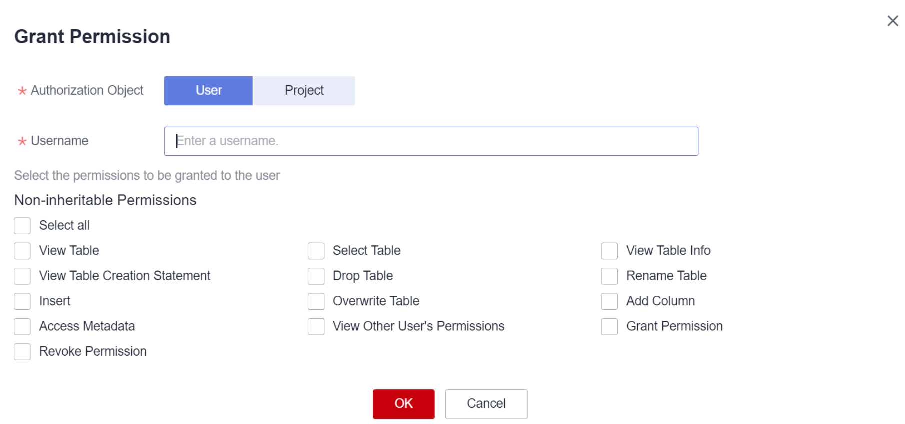


Figure 4-10 Granting DLI table permissions to a project

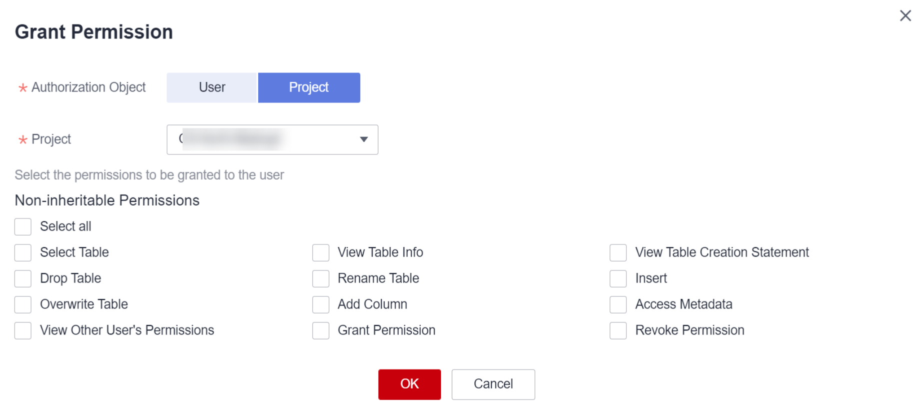


Table 4-7 Parameter description

Parameter	Description
Authorization Object	Select **User** or **Project**.
Username/ Project	• If you select **User**, enter the IAM username when granting table permissions to the user. **NOTE** The username is an existing IAM user name and has logged in to the DLI management console. • If you select **Project**, select the project to be authorized in the current region. **NOTE** If you select **Project**, you can only view information about the authorized tables and their databases.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Non-inheritable Permissions</td><td>Select a permission to grant it to the user, or deselect a permission to revoke it.
The following permissions are applicable to both user and project authorization:
Select Table: This permission allows you to query data of the current table.
View Table Information: This permission allows you to view information about the current table.
View Table Creation Statement: This permission allows you to view the statement for creating the current table.
Drop Table: This permission allows you to delete the current table.
Rename Table: Rename the current table.
Insert: This permission allows you to insert data into the current table.
Overwrite: This permission allows you to insert data to overwrite the data in the current table.
Add Column: Add columns to the current table.
Grant Permission: The current user can grant table permissions to other users.
Revoke Permission: The current user can revoke the table's permissions that other users have but cannot revoke the table owner's permissions.
View Other Users' Permissions: This permission allows you to query other users' permission on the current table.
The partition table also has the following permissions:
Delete Partition: This permission allows you to delete existing partitions from a partition table.
View All Partitions: This permission allows you to view all partitions in a partition table.
The following permissions can only be granted to users:
View Table: This permission allows you to display the current table.</td></tr>
</table>

- For details about the OBS table permissions, see **Table 4-8**.

Figure 4-11 Granting OBS table permissions to a user

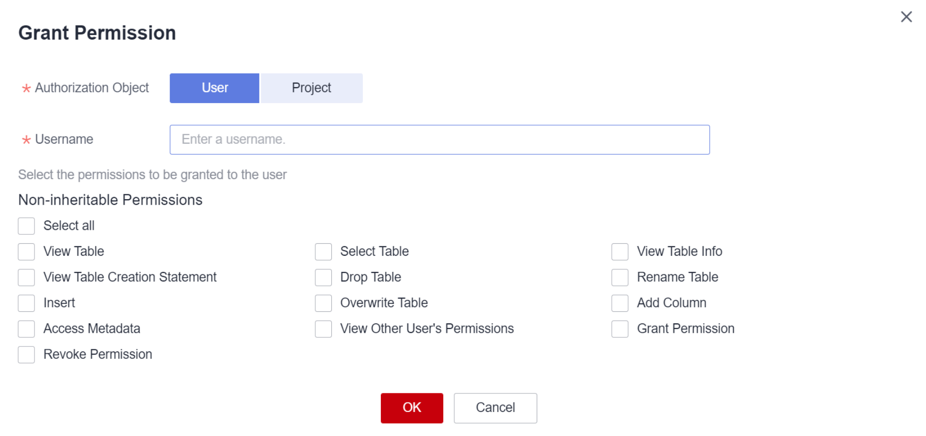


Figure 4-12 Granting OBS table permissions to a project

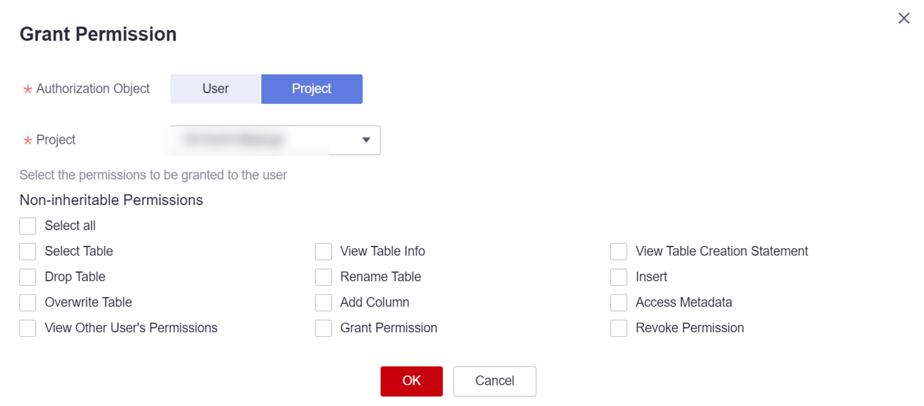


Table 4-8 Parameter description

Parameter	Description
Authorization Object	Select **User** or **Project**.
Username/Project	• If you select **User**, enter the IAM username when granting table permissions to the user. **NOTE** The username is an existing IAM user name and has logged in to the DLI management console. • If you select **Project**, select the project to be authorized in the current region. **NOTE** If you select **Project**, you can only view information about the authorized tables and their databases.

Parameter	Description
Non-inheritable Permissions	Select a permission to grant it to the user, or deselect a permission to revoke it. ● The following permissions are applicable to both user and project authorization: – **View Table Creation Statement**: This permission allows you to view the statement for creating the current table. – **View Table Information**: This permission allows you to view information about the current table. – **Select Table**: This permission allows you to query data of the current table. – **Drop Table**: This permission allows you to delete the current table. – **Rename Table**: Rename the current table. – **Insert**: This permission allows you to insert data into the current table. – **Overwrite**: This permission allows you to insert data to overwrite the data in the current table. – **Add Column**: This permission allows you to add columns to the current table. – **Grant Permission**: This permission allows you to grant table permissions to other users or projects. – **Revoke Permission**: This permission allows you to revoke the table's permissions that other users or projects have but cannot revoke the table owner's permissions. – **View Other Users' Permissions**: This permission allows you to query other users' permission on the current table. The partition table also has the following permissions: – **Add Partition**: This permission allows you to add a partition to a partition table. – **Delete Partition**: This permission allows you to delete existing partitions from a partition table. – **Configure Path for Partition**: This permission allows you to set the path of a partition in a partition table to a specified OBS path. – **Rename Table Partition**: This permission allows you to rename partitions in a partition table. – **Restore Table Partition**: This permission allows you to export partition information from the file system and save the information to metadata.

Parameter	Description
	– **View All Partitions**: This permission allows you to view all partitions in a partition table. ● The following permissions can only be granted to users: – **View Table**: This permission allows you to view the current table.

– For details about the view permissions, see **Table 4-9**.

NOTE

A view can be created only by using SQL statements. You cannot create a view on the **Create Table** page.

Figure 4-13 Granting view permissions to a user

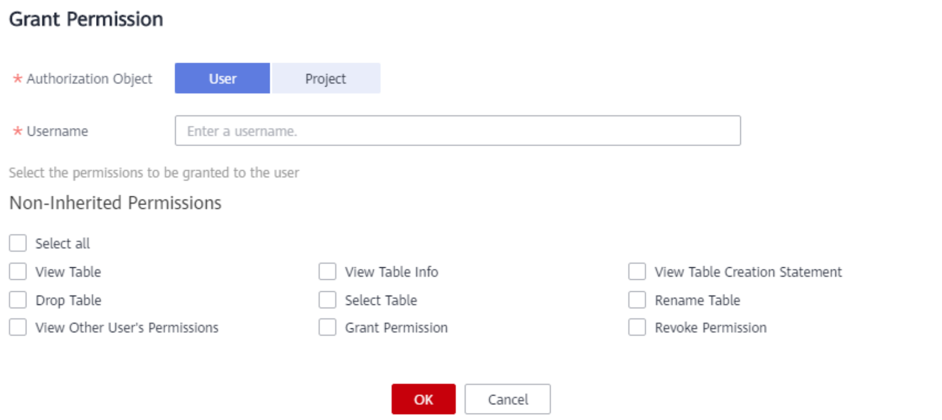


Figure 4-14 Granting view permissions to a project

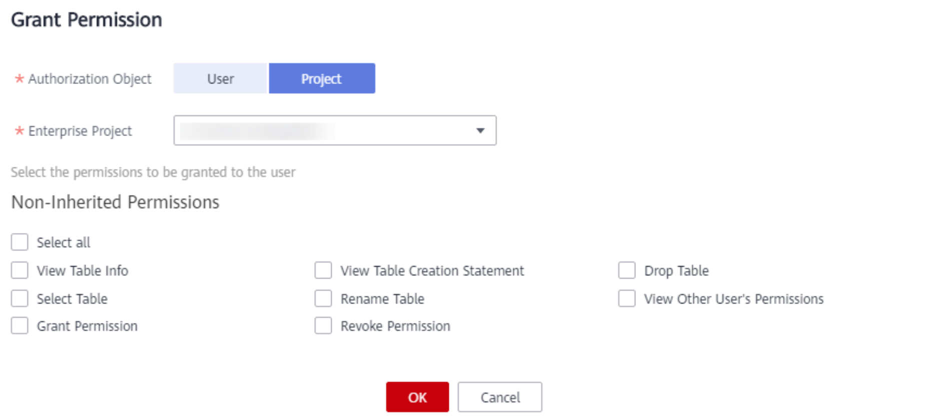


Table 4-9 Parameter description

Parameter	Description
Authorization Object	Select **User** or **Project**.
Username/Project	● If you select **User**, enter the IAM username when adding a user to the database. **NOTE** The username is an existing IAM user name and has logged in to the DLI management console. ● If you select **Project**, select the project to be authorized in the current region. **NOTE** If you select **Project**, you can only view information about the authorized tables and their databases.
Non-inheritable Permissions	Select a permission to grant it to the user, or deselect a permission to revoke it. ● The following permissions are applicable to both user and project authorization: – **View Table Information**: This permission allows you to view information about the current table. – **View Table Creation Statement**: This permission allows you to view the statement for creating the current table. – **Drop Table**: This permission allows you to delete the current table. – **Select Table**: This permission allows you to query data of the current table. – **Rename Table**: Rename the current table. – **Grant Permission**: The current user or project can grant table permissions to other users or projects. – **Revoke Permission**: The current user or project can revoke the table's permissions that other users or projects have but cannot revoke the table owner's permissions. – **View Other Users' Permissions**: This permission allows you to query other users' permission on the current table. ● Only applicable to – **View Table**: This permission allows you to view the current table.

3. Click **OK**.

Modifying Permissions for an Existing User or Project

For a user or project that has some permissions on the database, you can revoke the existing permissions or grant new ones.

NOTE

If all options under **Set Permission** are gray, you are not allowed to change permissions on this table. You can apply to the administrator, table owner, or other authorized users for granting and revoking table permissions.

1. In the **User Permission Info** list, find the user whose permission needs to be set.
   - If the user is an IAM user and is not the owner of the table, you can set permissions.
   - If the user is an administrator or table owner, you can only view permissions.

   In the **Project Permission Info** list, locate the project for which you want to set permissions and click **Set Permission**.
2. In the **Operation** column of the IAM user or project, click **Set Permission**. The **Set Permission** dialog box is displayed.
   - For details about DLI table user or project permissions, see **Table 4-7**.
   - For details about OBS table user or project permissions, see **Table 4-8**.
   - For details about View table user or project permissions, see **Table 4-9**.
3. Click **OK**.

Revoking All Permissions of a User or Project

Revoke all permissions of a user or a project.

- In the user list under **User Permission Info**, locate the row where the target IAM user resides and click **Revoke Permission** in the **Operation** column. In the displayed dialog box, click **OK**. In this case, the user has no permissions on the table.

  NOTE

  In the following cases, **Revoke Permission** is gray, indicating that the permission of the user cannot be revoked.
  - The user is an administrator.
  - The IAM user is the owner of the table.
  - The IAM user has only inheritable permissions.
- In the **Project Permission Info** area, select the project whose permissions need to be revoked and click **Revoke Permission** in the **Operation** column. After you click **OK**, the project does not have any permissions on the table.

  NOTE

  If a project has only inheritable permissions, **Revoke Permission** is gray, indicating that the permissions of the project cannot be revoked.

4.5.2 Deleting a Table on the DLI Console

You can delete tables as needed.

Precautions

- Databases or tables that are being used for running jobs cannot be deleted.
- Only administrators, table owners, and users with table deletion permission can delete tables.

  NOTE

  Deleted data tables cannot be restored. Exercise caution when performing this operation.

Deleting a Table

You can delete a table on either the **Data Management** page or the **SQL Editor** page.

- On the **Data Management** page
  a. In the navigation pane on the left of the console, choose **Data Management** > **Databases and Tables**.
  b. Locate the database whose tables you want to delete and click its name.
  c. On the displayed page, select the table you want to delete, click **More** in the **Operation** column, and select **Drop Table**.
  d. In the displayed dialog box, click **Yes**.
- On the **SQL Editor** page
  a. In the navigation pane on the left of the console, choose **SQL Editor**.
  b. In the middle pane, click the **Databases** tab. Click the name of the database whose tables you want to delete.
  c. Click ☰ next to the table to delete and select **Delete**.
  d. In the **Drop Table** dialog box that appears, click **OK**.

4.5.3 Changing the Table Owner on the DLI Console

In practical use, developers create databases and tables, which are then handed over to testers for testing. Once testing is complete, the databases and tables are handed over to O&M personnel for experience. In this scenario, ownership of the data can be transferred to another owner by changing the table owner.

Changing the Table Owner

1. In the navigation pane on the left of the console, choose **Data Management** > **Databases and Tables**.
2. Click the name of the database whose tables you want to modify.
3. Locate the table for which you want to change the owner, click **More** in the **Operation** column, and select **Modify Owner**.
4. In the displayed **Modify Owner** dialog box, enter a new owner name (an existing username) and click **OK**.

4.5.4 Importing OBS Data to DLI

This section describes how to import data stored in OBS to a table on the DLI console.

Precautions

- Only one path can be specified during data import. The path cannot contain commas (,).
- To import data in CSV format to a partitioned table, place the column to be partitioned in the last column of the data source.
- You are advised not to concurrently import data in to a table. If you concurrently import data into a table, there is a possibility that conflicts occur, leading to failed data import.
- The imported file can be in CSV, Parquet, ORC, JSON, and Avro format. The encoding format must be UTF-8.

Prerequisites

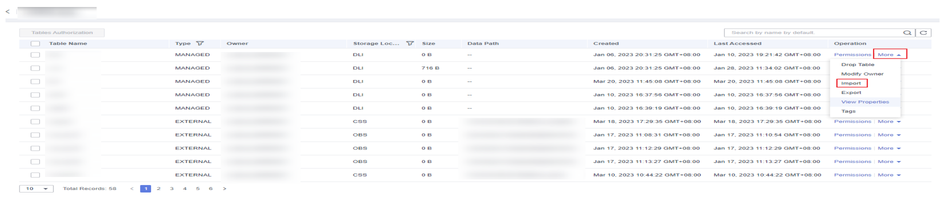
The data to be imported has been stored on OBS.

Procedure

Step 1 You can import data on either the **Data Management** page or the **SQL Editor** page.

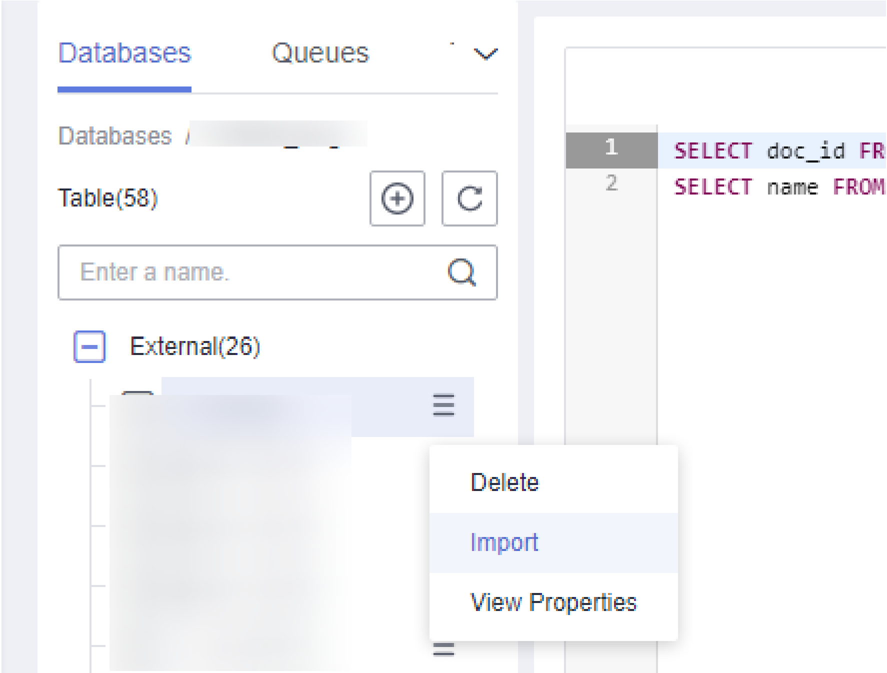
- To import data on the **Data Management** page:
  a. On the left of the management console, choose **Data Management** > **Databases and Tables**.
  b. Click the name of the database corresponding to the table where data is to be imported to switch to the table management page.
  c. Locate the row where the target table resides and choose **More** > **Import** in the **Operation** column. The **Import** dialog box is displayed.

  Figure 4-15 Importing data

- To import data on the **SQL Editor** page:
  a. On the left of the management console, click **SQL Editor**.
  b. In the navigation tree on the left of **SQL Editor**, click **Databases** to see all databases. Click the database where the target table belongs. The table list is displayed.
  c. Click on the right of the table and choose **Import** from the shortcut menu. The **Import** page is displayed.

Figure 4-16 SQL editor - importing data



Step 2 In the **Import** dialog box, set the parameters based on **Table 4-10**.

Table 4-10 Description

Parameter	Description	Example
Databases	Database where the current table is located.	-
Table Name	Name of the current table.	-
Queues	Queue where the imported data will be used	-
File Format	Format of the data source file to be imported. The CSV, Parquet, ORC, JSON, and Avro formats are supported. Encoding format. Only UTF-8 is supported.	CSV
Path	You can directly enter a path or click  and select an OBS path. If no bucket is available, you can directly switch to the OBS management console and create an OBS bucket. ● **When creating an OBS table, you must specify a folder as the directory. If a file is specified, data import may be failed.** ● **If a folder and a file have the same name in the OBS directory, the file path is preferred as the path of the data to be imported.** **NOTE** The path can be a file or folder.	obs://DLI/sampledata.csv

Parameter	Description	Example
Table Header: No/Yes	This parameter is valid only when **File Format** is set to **CSV**. Whether the data source to be imported contains the table header. Click **Advanced Settings** and select the checkbox next to **Table Header: No**. If the checkbox is selected, the table header is displayed. If the checkbox is deselected, no table header is displayed.	-
User-defined Delimiter	This parameter is valid only when **File Format** is set to **CSV** and you select **User-defined Delimiter**. The following delimiters are supported: ● Comma (,) ● Vertical bar (|) ● Tab character (\t) ● Others: Enter a user-defined delimiter.	Default value: comma (,)
User-defined Quotation Character	This parameter is valid only when **File Format** is set to **CSV** and **User-defined Quotation Character** is selected. The following quotation characters are supported: ● Single quotation mark (') ● Double quotation marks (") ● Others: Enter a user-defined quotation character.	Default value: double quotation marks (")
User-defined Escape Character	This parameter is valid only when **File Format** is set to **CSV** and you select **User-defined Escape Character**. The following escape characters are supported: ● Backslash (\) ● Others: Enter a user-defined escape character.	Default value: backslash (\)
Date Format	This parameter is valid only when **File Format** is set to **CSV** or **JSON**. This parameter specifies the format of the date in the table and is valid only **Advanced Settings** is selected. The default value is **yyyy-MM-dd**. For definition of characters involved in the date pattern, see Table 3 in **Importing Data to the Table**.	2000-01-01

Parameter	Description	Example
Timestamp Format	This parameter is valid only when **File Format** is set to **CSV** or **JSON**. This parameter specifies the format of the timestamp in the table and is valid only **Advanced Settings** is selected. The default value is **yyyy-MM-dd HH:mm:ss**. For definition of characters involved in the time pattern, see Table 3 in **Importing Data to the Table**.	2000-01-01 09:00:00
Error Records Path	This parameter is valid only when **File Format** is set to **CSV** or **JSON**. The parameter specifies the error data is stored in the corresponding OBS path and is valid only **Advanced Settings** is selected.	obs://DLI/

Step 3 Click **OK**.

Step 4 You can view the imported data in either of the following ways:

NOTE

Currently, only the first 10 records are displayed.

- Choose **Data Management** > **Databases and Tables** in the navigation pane of the console. Locate the row that contains the database where the target table belongs and click **More** > **View Properties** in the **Operation** column. In the displayed dialog box, click the **Preview** tab to view the imported data.
- On the **Databases** tab of the **SQL Editor**, click the database name to go to the table list. Click ≡ on the right of a table name and choose **View Properties** from the shortcut menu. In the displayed dialog box, click **Preview** to view the imported data.

Step 5 (Optional) View the status and execution result of the importing job on the **Job Management** > **SQL Jobs** page.

----End

4.5.5 Exporting DLI Table Data to OBS

You can export data from a DLI table to OBS. During the export, a folder is created in OBS or the content in the existing folder is overwritten.

Precautions

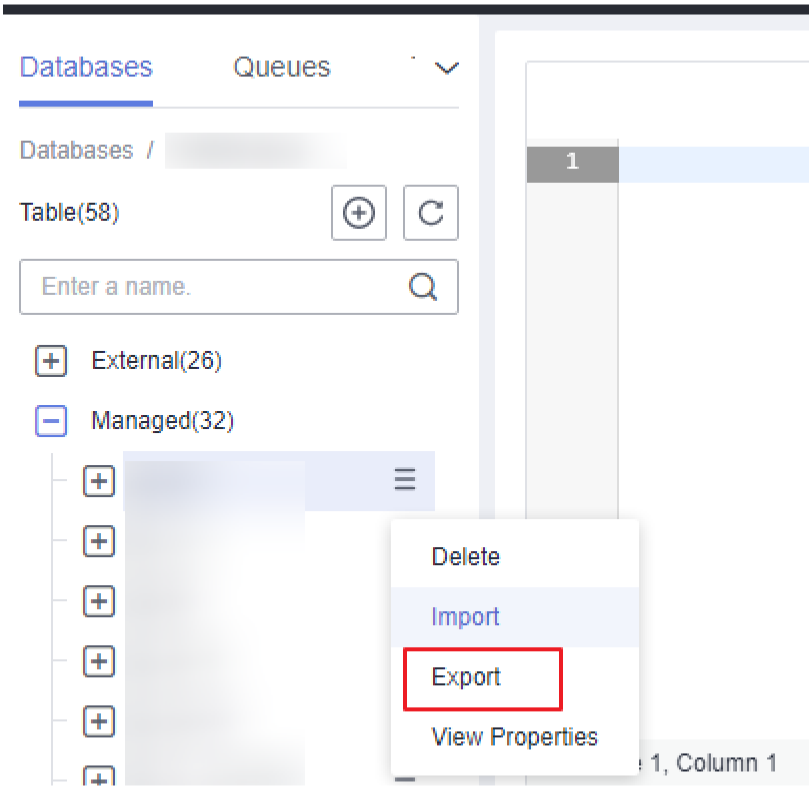
- The exported file can be in JSON format, and the text format can only be UTF-8.
- Only the data in the DLI table (the table type is **Managed**) can be exported to the OBS bucket, and the export path must be specified to the folder level.
- Data can be exported across accounts. That is, after account B authorizes account A, account A can export data to the OBS path of account B if account A has the permission to read the metadata and permission information about the OBS bucket of account B and read and write the path.

Procedure

Step 1 You can export data on either the **Data Management** page or the **SQL Editor** page.

- To export data on the **Data Management** page:
  a. On the left of the management console, choose **Data Management** > **Databases and Tables**.
  b. Click the name of the database corresponding to the table where data is to be exported to switch to the **Manage Tables** page.
  c. Select the corresponding table (DLI table) and choose **More** > **Export** in the **Operation** column. The **Export Data** page is displayed.
- To export data on the **SQL Editor** page:
  a. On the left of the management console, click **SQL Editor**.
  b. In the navigation tree on the left, click **Databases** to see all databases. Click the database name corresponding to the table to which data is to be exported. The tables are displayed.
  c. Click ☰ on the right of the managed table (DLI table) whose data is to be exported, and choose **Export** from the shortcut menu.

Figure 4-17 Exporting the table



Step 2 In the displayed **Export Data** dialog box, specify parameters by referring to **Table 4-11**.

Figure 4-18 Exporting data

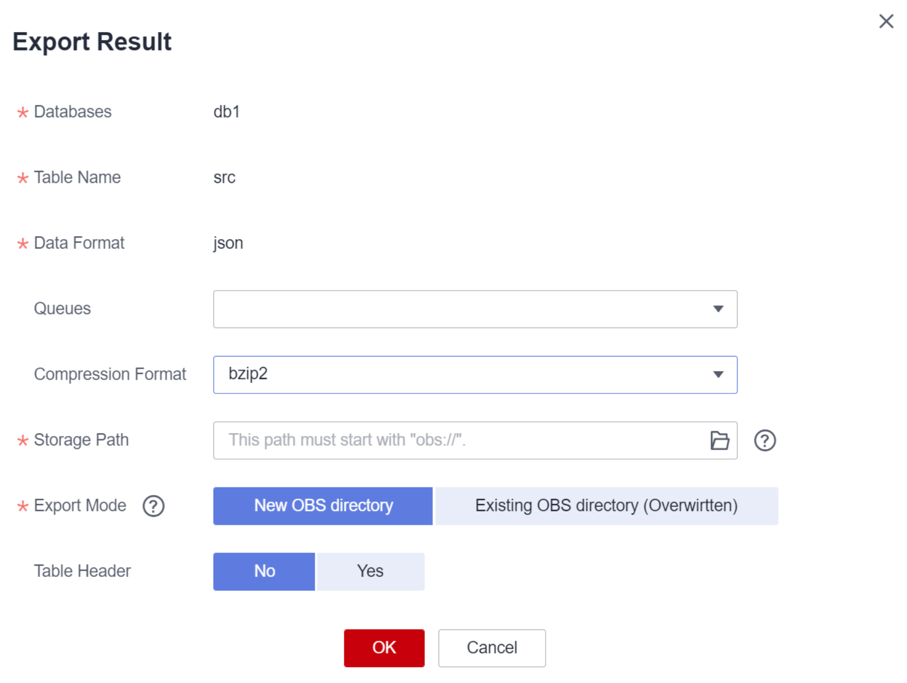


Table 4-11 Parameter description

Parameter	Description
Databases	Database where the current table is located.
Table Name	Name of the current table.
Data Format	Format of the file storing data to be exported. Formats other than JSON will be supported in later versions.
Queue	Select a queue.
Compression Format	Compression format of the data to be exported. The following compression formats are supported: ● **none** ● **bzip2** ● **deflate** ● **gzip**
Storage Path	● Enter or select an OBS path. ● The export path must be a folder that does not exist in the OBS bucket. Specifically, you need to create a folder in the target OBS directory. ● The folder name cannot contain the special characters of **\/:*?"<>|**, and cannot start or end with a dot (.).

Parameter	Description
Export Mode	Storage mode of the data to be exported. ● **New OBS directory**: If the specified export directory exists, an error is reported and the export operation cannot be performed. ● **Existing OBS directory (Overwritten)**: If you create a file in the specified directory, the existing file will be overwritten.
Table Header: No/Yes	Whether the data to be exported contains the table header.

Step 3 Click **OK**.

Step 4 (Optional) You can view the job status (indicated by **Status**), statements (indicated by **Statement**), and other information about exporting jobs on the **SQL Jobs** page.

1. Select **EXPORT** from the **Job Type** drop-down list box and specify the time range for exporting data. The jobs meeting the requirements are displayed in the job list.
2. Click ˅ to view details about an exporting job.

----End

4.5.6 Previewing Table Data on the DLI Console

The **Preview** page displays the first 10 records in the table.

Procedure

You can preview data on either the **Data Management** page or the **SQL Editor** page.

- To preview data on the **Data Management** page:
  a. On the left of the management console, choose **Data Management** > **Databases and Tables**.
  b. On the displayed **Data Management** page, click the name of the database where the target table whose data you want to export resides to switch to the **Manage Tables** page.
  c. Click **More** in the **Operation** column of the target table and select **View Properties**.
  d. Click the **Preview** tab to preview the table data.
- To preview data on the **SQL Editor** page:
  a. On the left of the management console, click **SQL Editor**.
  b. In the navigation pane of the displayed **SQL Editor** page, click **Databases**.
  c. Click the corresponding database name to view the tables in the database.

d. Click on the right of the corresponding table, choose **View Properties** from the list menu, and click the **Preview** tab to preview the data of the table.

4.6 Creating and Using LakeFormation Metadata

4.6.1 Connecting DLI to LakeFormation

Scenario

LakeFormation is an enterprise-level, all-in-one lakehouse construction service that provides unified metadata management capabilities. It supports seamless integration with various compute engines and big data cloud services, enabling you to efficiently build data lakes and operate related businesses, thereby accelerating the extraction of business data value.

In Spark and SQL job scenarios, LakeFormation can be connected to achieve unified metadata management. This section will guide you through the steps to configure the data connection between DLI and LakeFormation.

For the Spark syntax of LakeFormation, see **Spark Syntax Reference**.

For the Flink syntax of LakeFormation, see **Flink Syntax Reference**.

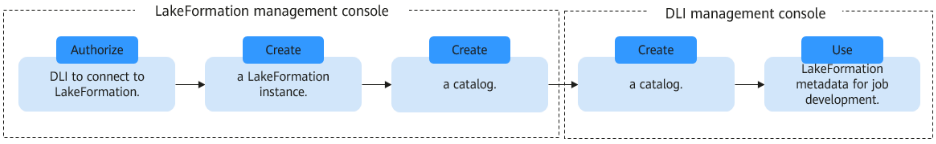
Notes

To use this function, which is currently in the whitelist, submit a request by choosing **Service Tickets** > **Create Service Ticket** in the upper right corner of the management console.

Connecting DLI to LakeFormation depends on the availability of the LakeFormation service. To understand the availability scope of LakeFormation, refer to **Global Products and Services**.

Procedure

Figure 4-19 Procedure



Notes and Constraints

- **Table 4-12** lists the DLI queue and engine types that allow you to connect DLI to LakeFormation to obtain metadata.

  For the engine type and version of a queue, see **Viewing Basic Information About a Queue**.

Table 4-12 Queue and engine types that allow for the connection of DLI to LakeFormation for metadata retrieval

Queue Type	Engine Type and Version
default queue	● Spark 3.3.*x*: can connect to LakeFormation to obtain metadata. ● HetuEngine 2.1.0: can connect to LakeFormation to obtain metadata.
For SQL	● Spark 3.3.*x*: can connect to LakeFormation to obtain metadata. ● HetuEngine 2.1.0: can connect to LakeFormation to obtain metadata.
For general purpose	Flink job scenario: Flink 1.15 or later supports integration with LakeFormation to obtain metadata when using queues within an elastic resource pool.

- DLI can only connect to the default LakeFormation instance. Set the instance in LakeFormation as the default to ensure successful connection.
- DLI can read data in Avro, Json, Parquet, CSV, ORC, Text, and Hudi formats from LakeFormation.
- LakeFormation manages the permissions for databases and tables in the data catalog of LakeFormation.
- After connecting DLI to LakeFormation, the original databases and tables in DLI will be moved to the data directory of DLI.

Step 1: Create a LakeFormation Instance for Metadata Storage

LakeFormation instances provide basic resources for metadata management. DLI can only connect to the default LakeFormation instance.

1. **Creating an instance**
   a. Log in to the LakeFormation management console.
   b. Click **Buy Now** or **Buy Instance** in the upper right corner of the page.
      If there are no instances available on the page, **Buy Now** is displayed. If there are any LakeFormation instances on the page, **Buy Instance** is displayed.
   c. Set LakeFormation instance parameters as needed to complete instance creation.
      In this example, we create a pay-per-use shared instance.
2. **Setting a LakeFormation instance as the default**
   a. View the value of **Default Instance** in the **Basic Information** area.
      - **true**: The instance is the default.
      - **false**: The instance is not the default.
   b. To set an instance as the default, click **Set as Default** in the upper right corner of the page.

c. In the dialog box that appears, select **I understand the consequences of changing the default instance and have still decided to perform this operation.** and click **OK**.

NOTE

DLI can currently only connect to the default LakeFormation instance. Changing an instance to the default may impact the services that use LakeFormation. Exercise caution when performing this operation.

Step 2: Create a Catalog on the LakeFormation Management Console

A data catalog is a metadata management object that can contain multiple databases. You can create and manage multiple catalogs in LakeFormation to isolate metadata of different external clusters.

1. Log in to the LakeFormation management console.
2. In the navigation pane on the left, choose **Metadata** > **Catalog**.
3. On the displayed page, click **Create**.

   Set catalog instance parameters as needed.

   For parameter settings and descriptions, see **Creating a Catalog**.
4. Once created, you can view information about the created catalog on the **Catalog** page.

Step 3: Create a Data Catalog on the DLI Management Console

On the DLI management console, you need to create a link to the catalog to access the catalog stored in the LakeFormation instance.

NOTE

- DLI can only connect to the default LakeFormation instance. Set the instance in LakeFormation as the default to ensure successful connection.
- You can only create one mapping for each data catalog in LakeFormation.

  For example, a user creates a mapping named **catalogMapping1** in DLI, which corresponds to the data catalog **catalogA** in LakeFormation. Once created, you cannot create a mapping to **catalogA** in the same project space.

1. Log in to the DLI management console.
2. In the navigation pane on the left, choose **SQL Editor**.
3. The **Catalog** tab of the SQL editor is displayed.
4. Click  to create a data catalog.
5. In the **Create Catalog** dialog box, set data catalog parameters.

Table 4-13 Data catalog parameters

Parameter	Mandatory	Description
External Catalog Name	Yes	Catalog name of the default LakeFormation instance.

Parameter	Mandatory	Description
Type	Yes	Currently, the only available option is **LakeFormation**. This option is fixed and does not need to be selected.
Catalog Name	Yes	Catalog mapping name used in DLI. When running SQL statements, you need to specify the catalog mapping to identify the external metadata to be accessed. You are advised to set this parameter to the same value as **External Catalog Name**. Currently, DLI can only connect to the data catalog of the default LakeFormation instance.
Description	No	Description of the data catalog.

6. Click **OK**.

Step 4: Authorize to Use LakeFormation Resources

- **SQL job scenarios**

  Before submitting a SQL job, you need to authorize DLI to access LakeFormation resources such as metadata, databases, tables, columns, and functions, to ensure that the job can access required data and resources during execution. **Supported Actions for LakeFormation SQL Resources** describes LakeFormation resources and corresponding permissions.

  To use LakeFormation resources, you need to separately complete IAM fine-grained authorization and LakeFormation SQL resource authorization.

  – **IAM fine-grained authorization for LakeFormation: Authorize DLI to use LakeFormation APIs.**

    IAM offers multiple methods to manage the permissions of users, groups, and roles to access resources. You can create policies on the IAM console to define which users or roles can call LakeFormation APIs. Then, attach these policies to the specified users or roles.

    ▪ Method 1: Role-based authorization

      A coarse-grained authorization strategy that defines permissions by job responsibility. Only a limited number of service-level roles are available for authorization.

      For example, grant users the read-only permission to query LakeFormation metadata resources by referring to **LakeFormation Permissions Management**.

      Alternatively, grant all operation permissions on LakeFormation-related metadata resources.

      Example:

```
{
    "Version": "1.1",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lakeformation:table:*",
        "lakeformation:database:*",
        "lakeformation:catalog:*",
        "lakeformation:function:*",
        "lakeformation:transaction:*",
        "lakeformation:policy:describe",
        "lakeformation:credential:describe"
      ]
    }
  ]
}
```

- Method 2: Policy-based fine-grained authorization

  A policy defines the permissions required to perform actions on a specific cloud resource under certain conditions.

  For LakeFormation permissions policies, see **LakeFormation Permissions and Supported Actions**.

  For how to grant permissions, see **Creating a User and Granting LakeFormation Permissions**.

– **LakeFormation SQL resource authorization: authorizes users to use specific LakeFormation resources (such as metadata, databases, tables, columns, and functions).**

  Users are allowed to access specific resources. This controls access to LakeFormation data and metadata.

  Two methods:

  - Method 1: Authorize access to resources on the LakeFormation management console.

    For details, see **Granting Permissions**.

    For LakeFormation SQL resource permissions, see **Data Permissions**.

  - Method 2: Run the **GRANT** SQL statement on the DLI management console.

    The **GRANT** statement is used for authorization in SQL.

    You can run the **GRANT** statement to grant users or roles the permission to access databases, tables, columns, and functions.

    Supported Actions for LakeFormation SQL Resources describes authorization policies for LakeFormation resources.

    NOTE

    Currently, you cannot authorize access to catalogs by running the **GRANT** statement on the DLI console. To authorize access, use **method 1**.

- **Spark Jar, Flink OpenSource SQL, and Flink Jar job scenarios:**

  – Method 1: Use agency authorization. Before using Spark 3.3.1 or later and Flink 1.15 to run jobs, you need to create an agency on the IAM console and add the new agency information when configuring the job.

  For agency permission examples, see **Creating a Custom DLI Agency** and **Agency Permission Policies in Common Scenarios**.

- Method 2: Use DEW for authorization.
  - You have granted the IAM user the IAM and LakeFormation permissions. For details, see IAM authorization in **SQL job scenarios**.
  - A shared secret has been created on the DEW console and the secret value has been stored. For details, see **Creating a Shared Secret**.
  - An agency has been created and authorized for DLI to access DEW. The agency must have been granted the following permissions:
    - Permission of the **ShowSecretVersion** interface for querying secret versions and secret values in DEW: **csms:secretVersion:get**.
    - Permission of the **ListSecretVersions** interface for listing secret versions in DEW: **csms:secretVersion:list**.
    - Permission to decrypt DEW secrets: **kms:dek:decrypt**

    For agency permission examples, see **Creating a Custom DLI Agency** and **Agency Permission Policies in Common Scenarios**.

Step 5: Use LakeFormation Metadata During DLI Job Development

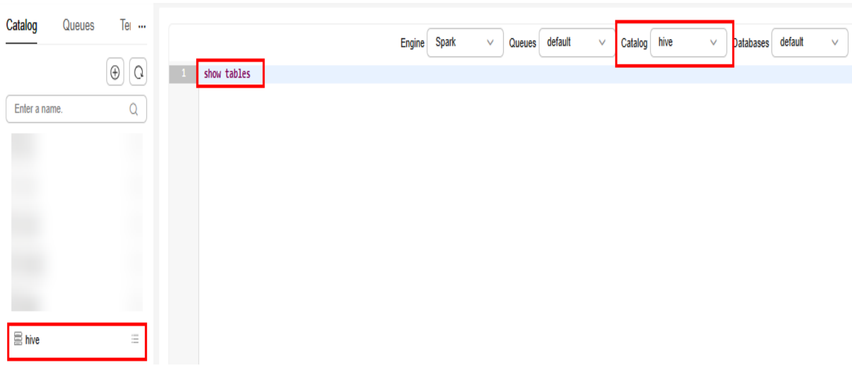
After connecting DLI to the default LakeFormation instance and authorizing access to LakeFormation resources, you can use LakeFormation metadata during DLI job development.

- **DLI SQL:**

  For the SQL syntax of LakeFormation, see **Data Lake Insight Spark SQL Syntax Reference**.

  When running a SQL job, you can select the catalog where the SQL statement is located on the console, as shown in **Figure 4-20**, or specify **catalogName** in the SQL statement. **catalogName** is the mapping name of the data catalog on the DLI console.

  Figure 4-20 Selecting a data catalog on the SQL editor page



NOTE

- When connecting DLI to a LakeFormation instance, you need to specify the OBS path for storing the database when creating it.
- When connecting DLI to a LakeFormation instance, you cannot set table lifecycle and versioning when creating a table.
- When connecting DLI to a LakeFormation instance, the **LOAD DATA** statement does not support datasource tables, and partitions must be specified if the statement is used to import data into a partitioned table.
- If the databases and tables created on the LakeFormation console contain Chinese characters, operations on them cannot be performed on DLI.
- When connecting DLI to a LakeFormation instance, you cannot specify filter criteria to delete partitions.
- When connecting DLI to a LakeFormation instance, you cannot create Truncate Datasource or Hive foreign tables.
- DLI does not currently support the use of LakeFormation row filter criteria.
- When DLI reads binary data for console display, it converts the binary data to Base64.
- DLI does not currently support authorizing access to LakeFormation paths.

- **DLI Spark Jar:**

  This part explains how to configure LakeFormation metadata when submitting a Spark Jar job on the DLI management console.

  – Example Spark Jar

```
SparkSession spark = SparkSession.builder()
    .enableHiveSupport()
    .appName("java_spark_demo")
    .getOrCreate();

spark.sql("show databases").show();
```

  – Configuring a Spark Jar job on the DLI management console

    ■ **(Recommended) Method 1: Configure a Spark Jar job's access to LakeFormation metadata using the parameters (such as agency and metadata source) provided on the console**

    When creating or editing a Spark Jar job, refer to **Table 4-14** to configure the job's access to LakeFormation metadata.

Table 4-14 Configuring a Spark Jar job's access to LakeFormation metadata

Parameter	Description	Example Value
Spark Version	Spark 3.3.*x* or later can connect to LakeFormation.	3.3.1

<table>
<tr><th>Parameter</th><th>Description</th><th>Example Value</th></tr>
<tr><td>Agency</td><td>Before using Spark 3.3.1 or later to run jobs, you need to create an agency on the IAM console and add the new agency information. Once set, the system automatically adds the following agency configuration to your job:
spark.dli.job.agency.name=agency
For agency permission examples, see Creating a Custom DLI Agency and Agency Permission Policies in Common Scenarios.</td><td>-</td></tr>
<tr><td>Access Metadata</td><td>Whether to allow the Spark job to access metadata.</td><td>Yes</td></tr>
<tr><td>Metadata Source</td><td>Type of metadata the Spark job accesses. In this scenario, select lakeformation.
Once set to lakeformation, the system automatically adds the following configurations to your job to load LakeFormation dependencies.
spark.sql.catalogImplementation=hive
spark.hadoop.hive-ext.dlcatalog.metastore.client.enable=true
spark.hadoop.hive-ext.dlcatalog.metastore.session.client.class=com.huawei.cloud.dalf.lakecat.client.hiveclient.LakeCatMetaStoreClientog
// Load LakeFormation dependencies.
spark.driver.extraClassPath=/usr/share/extension/dli/spark-jar/lakeformation/*
spark.executor.extraClassPath=/usr/share/extension/dli/spark-jar/lakeformation/*
You can also set the metadata source in the Spark Arguments(--conf) parameter. This means if you set the metadata source in both Metadata Source and Spark Arguments(--conf), the system preferentially uses the information configured in Spark Arguments(--conf).
You are advised to set the metadata source through Metadata Source.</td><td>LakeFormation</td></tr>
</table>

Parameter	Description	Example Value
Catalog Name	Name of the data catalog the Spark job accesses. The data catalog created on the DLI management console is selected, that is, the mapping between DLI and the data catalog of the default LakeFormation instance. This data catalog is connected to the data catalog of the default LakeFormation instance. To specify other LakeFormation instances, configure the LakeFormation instances and data catalogs to be connected in **Spark Arguments(--conf)**. For details, see **Method 2: Configure a Spark Jar job's access to LakeFormation metadata using the Spark Arguments(--conf) parameter**. Once set to **LakeFormation**, the system automatically adds the following configuration to your job to connect to the data catalog of the default LakeFormation instance: `spark.hadoop.lakecat.catalogname.default=lfcatalog` You can also set the data catalog name in the **Spark Arguments(--conf)** parameter. This means if you set the data catalog name in both **Catalog Name** and **Spark Arguments(--conf)**, the system preferentially uses the information configured in **Spark Arguments(--conf)**. You are advised to set the data catalog name through **Catalog Name**.	-

Parameter	Description	Example Value

Spark Arguments(--conf)	You can also set the metadata source and data catalog name in the **Spark Arguments(--conf)** parameter. This means if you set the metadata source and data catalog name in both **Metadata Source** and **Catalog Name**, as well as **Spark Arguments(--conf)**, the system preferentially uses the information configured in **Spark Arguments(--conf)**.
	● To access a Hudi data table, add the following configurations to the **Spark Arguments(--conf)** parameter:
	`spark.sql.extensions=org.apache.spark.sql.hudi.HoodieSparkSessionExtension`
	`spark.hadoop.hoodie.write.lock.provider=org.apache.hudi.lakeformation.LakeCatMetastoreBasedLockProvider`
	● To access a Delta data table, add the following configurations to the **Spark Arguments(--conf)** parameter:
	`spark.sql.catalog.spark_catalog=org.apache.spark.sql.delta.catalog.DeltaCatalog`
	`spark.sql.extensions=io.delta.sql.DeltaSparkSessionExtension`
		-

- **Method 2: Configure a Spark Jar job's access to LakeFormation metadata using the Spark Arguments(--conf) parameter**

  When creating or editing a Spark Jar job, configure the following information in the **Spark Arguments(--conf)** parameter on the job configuration page to access LakeFormation metadata:

```
spark.sql.catalogImplementation=hive
spark.hadoop.hive-ext.dlcatalog.metastore.client.enable=true
spark.hadoop.hive-
ext.dlcatalog.metastore.session.client.class=com.huawei.cloud.dalf.lakecat.client.hiveclient.LakeCatMetaStoreClient
spark.sql.extensions=org.apache.spark.sql.hudi.HoodieSparkSessionExtension // Hudi is supported, which is optional.
spark.hadoop.hoodie.write.lock.provider=org.apache.hudi.lakeformation.LakeCatMetastoreBasedLockProvider // Hudi is supported, which is optional.
// Access using an agency with the OBS and LakeFormation permissions. You are advised to configure the minimum permission set.
spark.dli.job.agency.name=agencyForLakeformation
// ID of the LakeFormation instance to be accessed, which can be viewed on the LakeFormation console. This parameter is optional. If unset, the default LakeFormation instance is accessed.
spark.hadoop.lakeformation.instance.id=xxx
// Name of the catalog to be accessed on the LakeFormation side, which can be viewed on the LakeFormation console. This parameter is optional. If unset, the default value is hive.
spark.hadoop.lakecat.catalogname.default=lfcatalog
// Load LakeFormation dependencies.
spark.driver.extraClassPath=/usr/share/extension/dli/spark-jar/lakeformation/*
spark.executor.extraClassPath=/usr/share/extension/dli/spark-jar/lakeformation/*
```

- **DLI Flink OpenSource SQL**

– Example 1: Connecting DLI to LakeFormation using an agency

Create a Flink OpenSource SQL job and set the following parameters:

Parameter	Description	Example Value
Flink Version	Flink 1.15 or later can connect to LakeFormation.	1.15
Agency	Before using Flink 1.15 or later to run jobs, you need to create an agency on the IAM console and add the new agency information. Once set, the system automatically adds the following agency configuration to your job: flink.dli.job.agency.name=*agency* For agency permission examples, see **Creating a Custom DLI Agency** and **Agency Permission Policies in Common Scenarios**.	-
Enable Checkpointing	Select it.	Select it.
Runtime Configuration	• Type of metadata the Flink job accesses. In this scenario, select **lakeformation**. flink.dli.job.catalog.type=lakeformation • Name of the data catalog the Flink job accesses. flink.dli.job.catalog.name=[Catalog name in LakeFormation] The data catalog created on the DLI management console is selected, that is, the mapping between DLI and the data catalog of the default LakeFormation instance. This data catalog is connected to the data catalog of the default LakeFormation instance.	-

For the catalog parameters in the example, see **Table 4-15**.

Table 4-15 Catalog parameters in the example Flink OpenSource SQL

Parameter	Description	Mandatory	Example Value
type	Catalog type	Yes	Fixed at hive
hive-conf-dir	hive-conf path, which is fixed at **/opt/flink/conf**.	Yes	Fixed at /opt/flink/conf

Parameter	Description	Mandatory	Example Value
default-database	Default database name	No	Default database

```
CREATE CATALOG hive
WITH
  (
    'type' = 'hive',
    'hive-conf-dir' = '/opt/flink/conf',  -- Fixed at /opt/flink/conf
    'default-database'='default'
  );

USE CATALOG hive;

CREATE TABLE IF NOT EXISTS
  dataGenSource612 (user_id string, amount int)
WITH
  (
    'connector' = 'datagen',
    'rows-per-second' = '1',
    'fields.user_id.kind' = 'random',
    'fields.user_id.length' = '3'
  );

CREATE table IF NOT EXISTS
  printSink612 (user_id string, amount int)
WITH
  ('connector' = 'print');

INSERT INTO
  printSink612
SELECT
  *
FROM
  dataGenSource612;
```

– Example 2: Connecting DLI to LakeFormation using DEW

Create a Flink OpenSource SQL job and set the following parameters:

Parameter	Description	Example Value
Flink Version	Flink 1.15 or later can connect to LakeFormation.	1.15
Agency	Before using Flink 1.15 or later to run jobs, you need to create an agency on the IAM console and add the new agency information. Once set, the system automatically adds the following agency configuration to your job: flink.dli.job.agency.name=*agency* For agency permission examples, see **Creating a Custom DLI Agency** and **Agency Permission Policies in Common Scenarios**.	-
Enable Checkpointing	Select it.	Select it.

Parameter	Description	Example Value
Runtime Configuration	● Type of metadata the Flink job accesses. In this scenario, select **lakeformation**. flink.dli.job.catalog.type=lakeformation ● Name of the data catalog the Flink job accesses. flink.dli.job.catalog.name=[Catalog name in LakeFormation] The data catalog created on the DLI management console is selected, that is, the mapping between DLI and the data catalog of the default LakeFormation instance. This data catalog is connected to the data catalog of the default LakeFormation instance.	-

For the catalog parameters in the example, see **Table 4-16**.

Set **properties.catalog.lakeformation.auth.identity.util.class** to **com.huawei.flink.provider.lakeformation.FlinkDewIdentityGenerator** and configure DEW.

Table 4-16 Catalog parameters in the example Flink OpenSource SQL (using DEW)

Parameter	Description	Mandatory	Example Value
type	Catalog type	Yes	Fixed at hive
hive-conf-dir	hive-conf path, which is fixed at **/opt/flink/conf**.	Yes	Fixed at **/opt/flink/conf**
default-database	Default database name	No	If unset, the default database is used.
properties.catalog.lakecat.auth.identity.util.class	Authentication information acquisition class	Yes	Mandatory when DEW is used, which is fixed at **com.huawei.flink.provider.lakeformation.FlinkDewIdentityGenerator**.

Parameter	Description	Mandatory	Example Value
properties.catalog.dew.projectId	ID of the project DEW belongs to. The default value is the ID of the project where the Flink job is.	Yes	Mandatory when DEW is used
properties.catalog.dew.endpoint	Endpoint of the DEW service to be used.	Yes	Mandatory when DEW is used. Example: kms.xxx.com
properties.catalog.dew.csms.secretName	Name of the shared secret in DEW's secret management.	Yes	Mandatory when DEW is used
properties.catalog.dew.csms.version	Version number of the shared secret in DEW's secret management.	Yes	Mandatory when DEW is used
properties.catalog.dew.access.key	Enter the key corresponding to the **access.key** value in DEW's secret.	Yes	Mandatory when DEW is used
properties.catalog.dew.secret.key	Enter the key corresponding to the **secret.key** value in DEW's secret.	Yes	Mandatory when DEW is used

```
CREATE CATALOG myhive
WITH
  (
    'type' = 'hive',
    'hive-conf-dir' = '/opt/flink/conf',
    'default-database'='default',
    -- The following is the DEW configuration. Change the parameter values based on site requirements.
    'properties.catalog.lakeformation.auth.identity.util.class' = 'com.huawei.flink.provider.lakeformation.FlinkDewIdentityGenerator',
    'properties.catalog.dew.endpoint'='kms.xxx.com',
    'properties.catalog.dew.csms.secretName'='obsAksK',
    'properties.catalog.dew.access.key' = 'myak',
    'properties.catalog.dew.secret.key' = 'mysk',
    'properties.catalog.dew.projectId'='330e068af1334c9782f4226xxxxxxxxx',
    'properties.catalog.dew.csms.version'='v9'
);

USE CATALOG myhive;

create table IF NOT EXISTS dataGenSource_dew612(
  user_id string,
```

```
  amount int
) with (
  'connector' = 'datagen',
  'rows-per-second' = '1',
  'fields.user_id.kind' = 'random',
  'fields.user_id.length' = '3'
);

create table IF NOT EXISTS printSink_dew612(
  user_id string,
  amount int
) with (
  'connector' = 'print'
);

insert into printSink_dew612 select * from dataGenSource_dew612;
```

– Example 3: Connecting DLI to LakeFormation using an agency to write data to a Hudi table

Create a Flink OpenSource SQL job and set the following parameters:

Parameter	Description	Example Value
Flink Version	Flink 1.15 or later can connect to LakeFormation.	1.15
Agency	Before using Flink 1.15 or later to run jobs, you need to create an agency on the IAM console and add the new agency information. Once set, the system automatically adds the following agency configuration to your job: flink.dli.job.agency.name=*agency* For agency permission examples, see **Creating a Custom DLI Agency** and **Agency Permission Policies in Common Scenarios**.	-
Enable Checkpointing	Select it.	Select it.
Runtime Configuration	• Type of metadata the Flink job accesses. In this scenario, select **lakeformation**. flink.dli.job.catalog.type=lakeformation • Name of the data catalog the Flink job accesses. flink.dli.job.catalog.name=[Catalog name in LakeFormation] The data catalog created on the DLI management console is selected, that is, the mapping between DLI and the data catalog of the default LakeFormation instance. This data catalog is connected to the data catalog of the default LakeFormation instance.	-

For the catalog parameters in the example, see **Table 4-17**.

Table 4-17 Parameters for configuring a catalog of the Hudi type

Parameter	Description	Mandatory	Example Value
type	Catalog type	Yes	**hudi** for a Hudi table
hive-conf-dir	hive-conf path. The value is fixed at **/opt/flink/conf**.	Yes	Fixed at **/opt/flink/conf**
default-database	Default database name	No	Default database
mode	The value can be **'hms'** or **'non-hms'**. ● **hms** indicates that the created Hudi catalog uses Hive Metastore to store metadata. ● **non-hms** indicates that Hive Metastore is not used to store metadata.	Yes	Fixed at **hms**

Table 4-18 Connector parameters for a Hudi sink table

Parameter	Description	Mandatory	Example Value
connector	Flink connector type If set to **hudi**, the sink table is a Hudi table.	Yes	hudi
path	Basic path of the table. If the path does not exist, the system will create it.	Yes	Refer to the values configured in the sample code.
hoodie.datasource.write.recordkey.field	Unique key field name of the Hoodie table	No	Set **order_id** to a unique key.

Parameter	Description	Mandatory	Example Value
EXTERNAL	Whether the table is foreign	Yes Mandatory for the Hudi table and must be set to **true**.	true

```
CREATE CATALOG hive_catalog
  WITH (
    'type'='hive',
    'hive-conf-dir' = '/opt/flink/conf',
    'default-database'='test'
  );
USE CATALOG hive_catalog;
create table  if not exists genSource618 (
  order_id STRING,
  order_name STRING,
  price INT,
  weight INT
) with (
  'connector' = 'datagen',
  'rows-per-second' = '1',
  'fields.order_id.kind' = 'random',
  'fields.order_id.length' = '8',
  'fields.order_name.kind' = 'random',
  'fields.order_name.length' = '5'
);

CREATE CATALOG hoodie_catalog
  WITH (
    'type'='hudi',
    'hive.conf.dir' = '/opt/flink/conf',
    'mode'='hms' -- supports 'dfs' mode that uses the DFS backend for table DDLs
persistence
  );
CREATE TABLE  if not exists  hoodie_catalog.`test`.`hudiSink618` (
  `order_id` STRING PRIMARY KEY NOT ENFORCED,
  `order_name` STRING,
  `price` INT,
  `weight` INT,
  `create_time` BIGINT,
  `create_date` String
) PARTITIONED BY (create_date) WITH (
  'connector' = 'hudi',
  'path' = 'obs://xxx/catalog/dbtest3/hudiSink618',
  'hoodie.datasource.write.recordkey.field' = 'order_id',
  'write.precombine.field' = 'create_time',
  'EXTERNAL' = 'true' -- must be set
);

insert into hoodie_catalog.`test`.`hudiSink618`
select
  order_id,
  order_name,
  price,
  weight,
  UNIX_TIMESTAMP() as create_time,
  FROM_UNIXTIME(UNIX_TIMESTAMP(), 'yyyyMMdd') as create_date
from genSource618;
```

- **DLI Flink Jar**
  - **Example 1: Connecting DLI to LakeFormation using an agency**
    1. Develop a Flink Jar program, compile and upload the JAR file to OBS. In this example, the file is uploaded to the **obs://obs-test/dlitest/** directory.

       The sample code is as follows:

       In this example, random data is generated using the DataGen table and then output to the Print result table.

       For other connector types, see **List of Connectors Supported by Flink 1.15**.

```
package com.huawei.test;

import org.apache.flink.api.java.utils.ParameterTool;
import org.apache.flink.contrib.streaming.state.RocksDBStateBackend;
import org.apache.flink.runtime.state.filesystem.FsStateBackend;
import org.apache.flink.streaming.api.CheckpointingMode;
import org.apache.flink.streaming.api.environment.CheckpointConfig;
import org.apache.flink.streaming.api.environment.StreamExecutionEnvironment;
import org.apache.flink.table.api.EnvironmentSettings;
import org.apache.flink.table.api.bridge.java.StreamTableEnvironment;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import java.text.SimpleDateFormat;

@SuppressWarnings({"deprecation", "rawtypes", "unchecked"})
public class GenToPrintTaskAgency {
    private static final Logger LOGGER =
LoggerFactory.getLogger(GenToPrintTaskAgency.class);
    private static final String datePattern = "yyyy-MM-dd_HH-mm-ss";

    public static void main(String[] args) {
        LOGGER.info("Start task.");
        ParameterTool paraTool = ParameterTool.fromArgs(args);
        String checkpointInterval = "180000000";

        // set up execution environment
        StreamExecutionEnvironment env =
StreamExecutionEnvironment.getExecutionEnvironment();
        EnvironmentSettings settings = EnvironmentSettings.newInstance()
                .inStreamingMode().build();
        StreamTableEnvironment tEnv = StreamTableEnvironment.create(env, settings);

env.getCheckpointConfig().setCheckpointingMode(CheckpointingMode.EXACTLY_ONCE);
        env.getCheckpointConfig().setCheckpointInterval(Long.valueOf(checkpointInterval));
        env.getCheckpointConfig().enableExternalizedCheckpoints(
                CheckpointConfig.ExternalizedCheckpointCleanup.RETAIN_ON_CANCELLATION);

        SimpleDateFormat dateTimeFormat = new SimpleDateFormat(datePattern);
        String time = dateTimeFormat.format(System.currentTimeMillis());
        RocksDBStateBackend rocksDbBackend =
                new RocksDBStateBackend(
                        new FsStateBackend("obs://obs/xxx/testcheckpoint/" + time), true);
        env.setStateBackend(rocksDbBackend);

        String createCatalog = "CREATE CATALOG lf_catalog WITH (\n" +
                "    'type' = 'hive',\n" +
                "    'hive-conf-dir' = '/opt/hadoop/conf'\n" +
                "  );";
        tEnv.executeSql(createCatalog);

        String dataSource = "CREATE TABLE if not exists
lf_catalog.`testdb`.`dataGenSourceJar618_1` (\n" +
                "  user_id string,\n" +
```



```
            "  amount int\n" +
            ") WITH (\n" +
            "  'connector' = 'datagen',\n" +
            "  'rows-per-second' = '1',\n" +
            "  'fields.user_id.kind' = 'random',\n" +
            "  'fields.user_id.length' = '3'\n" +
            ")";
/*testdb is a custom database.*/

      tEnv.executeSql(dataSource);

      String printSink = "CREATE TABLE if not exists lf_catalog.`testdb`.`printSinkJar618_1`
(\n" +
            "  user_id string,\n" +
            "  amount int\n" +
            ") WITH ('connector' = 'print')";
      tEnv.executeSql(printSink);
/*testdb is a custom database.*/

      String query = "insert into lf_catalog.`test`.`printSinkJar618_1` " +
            "select * from lf_catalog.`test`.`dataGenSourceJar618_1`";
      tEnv.executeSql(query);
   }
}
```

ii. Create a Flink Jar job and set the following parameters:

Parameter	Description	Example Value
Flink Version	Flink 1.15 or later can connect to LakeFormation.	1.15
Agency	Before using Flink 1.15 or later to run jobs, you need to create an agency on the IAM console and add the new agency information. Once set, the system automatically adds the following agency configuration to your job: flink.dli.job.agency.name=*agency* For agency permission examples, see **Creating a Custom DLI Agency** and **Agency Permission Policies in Common Scenarios**.	-

Parameter	Description	Example Value
Runtime Configuration	• Type of metadata the Flink job accesses. In this scenario, select **lakeformation**. flink.dli.job.catalog.type=lakeformation • Name of the data catalog the Flink job accesses. flink.dli.job.catalog.name=[Catalog name in LakeFormation] The data catalog created on the DLI management console is selected, that is, the mapping between DLI and the data catalog of the default LakeFormation instance. This data catalog is connected to the data catalog of the default LakeFormation instance.	-

– **Example 2: Connecting DLI to LakeFormation using DEW**

i. Develop a Flink Jar program, compile and upload the JAR file to OBS. In this example, the file is uploaded to the **obs://obs-test/dlitest/** directory.

The sample code is as follows:

In this example, random data is generated using the DataGen table and then output to the Print result table.

For other connector types, see **List of Connectors Supported by Flink 1.15**.

```
package com.huawei.test;

import org.apache.flink.api.java.utils.ParameterTool;
import org.apache.flink.contrib.streaming.state.RocksDBStateBackend;
import org.apache.flink.runtime.state.filesystem.FsStateBackend;
import org.apache.flink.streaming.api.CheckpointingMode;
import org.apache.flink.streaming.api.environment.CheckpointConfig;
import org.apache.flink.streaming.api.environment.StreamExecutionEnvironment;
import org.apache.flink.table.api.EnvironmentSettings;
import org.apache.flink.table.api.bridge.java.StreamTableEnvironment;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import java.text.SimpleDateFormat;

@SuppressWarnings({"deprecation", "rawtypes", "unchecked"})
public class GenToPrintTaskDew {
    private static final Logger LOGGER = 
LoggerFactory.getLogger(GenToPrintTaskAgency.class);
    private static final String datePattern = "yyyy-MM-dd_HH-mm-ss";

    public static void main(String[] args) {
        LOGGER.info("Start task.");
        ParameterTool paraTool = ParameterTool.fromArgs(args);
        String checkpointInterval = "180000000";

        // set up execution environment
        StreamExecutionEnvironment env =
```

```
StreamExecutionEnvironment.getExecutionEnvironment();
        EnvironmentSettings settings = EnvironmentSettings.newInstance()
                .inStreamingMode().build();
        StreamTableEnvironment tEnv = StreamTableEnvironment.create(env, settings);

env.getCheckpointConfig().setCheckpointingMode(CheckpointingMode.EXACTLY_ONCE);
        env.getCheckpointConfig().setCheckpointInterval(Long.valueOf(checkpointInterval));
        env.getCheckpointConfig().enableExternalizedCheckpoints(
                CheckpointConfig.ExternalizedCheckpointCleanup.RETAIN_ON_CANCELLATION);

        SimpleDateFormat dateTimeFormat = new SimpleDateFormat(datePattern);
        String time = dateTimeFormat.format(System.currentTimeMillis());
        RocksDBStateBackend rocksDbBackend =
                new RocksDBStateBackend(
                        new FsStateBackend("obs://obs/xxx/testcheckpoint/" + time), true);
        env.setStateBackend(rocksDbBackend);

        String createCatalog = "CREATE CATALOG lf_catalog WITH (\n" +
                "    'type' = 'hive',\n" +
                "    'hive-conf-dir' = '/opt/hadoop/conf',\n" +
                "    'properties.catalog.lakeformation.auth.identity.util.class' =
'com.huawei.flink.provider.lakeformation.FlinkDewIdentityGenerator',\n" +
                "    'properties.catalog.dew.endpoint'='kms.xxx.xxx.com',\n" +
                "    'properties.catalog.dew.csms.secretName'='obsAksK',\n" +
                "    'properties.catalog.dew.access.key' = 'ak',\n" +
                "    'properties.catalog.dew.secret.key' = 'sk',\n" +

"    'properties.catalog.dew.projectId'='330e068af1334c9782f4226xxxxxxxxxx',\n" +
                "    'properties.catalog.dew.csms.version'='v9'\n" +
                "  );";
        tEnv.executeSql(createCatalog);

        String dataSource = "CREATE TABLE if not exists
lf_catalog.`testdb`.`dataGenSourceJarDew618_1` (\n" +
                "  user_id string,\n" +
                "  amount int\n" +
                ") WITH (\n" +
                "  'connector' = 'datagen',\n" +
                "  'rows-per-second' = '1',\n" +
                "  'fields.user_id.kind' = 'random',\n" +
                "  'fields.user_id.length' = '3'\n" +
                ")";
        tEnv.executeSql(dataSource);
/*testdb is a custom database.*/

        String printSink = "CREATE TABLE if not exists
lf_catalog.`testdb`.`printSinkJarDew618_1` (\n" +
                "  user_id string,\n" +
                "  amount int\n" +
                ") WITH ('connector' = 'print')";
        tEnv.executeSql(printSink);
/*testdb is a custom database.*/

        String query = "insert into lf_catalog.`test`.`printSinkJarDew618_1` " +
                "select * from lf_catalog.`test`.`dataGenSourceJarDew618_1`";
        tEnv.executeSql(query);
    }
}
```

ii. Create a Flink Jar job and set the following parameters:

Parameter	Description	Example Value
Flink Version	Flink 1.15 or later can connect to LakeFormation.	1.15

Parameter	Description	Example Value
Agency	Before using Flink 1.15 or later to run jobs, you need to create an agency on the IAM console and add the new agency information. Once set, the system automatically adds the following agency configuration to your job: flink.dli.job.agency.name=*agency* For agency permission examples, see **Creating a Custom DLI Agency** and **Agency Permission Policies in Common Scenarios**.	-
Runtime Configuration	• Type of metadata the Flink job accesses. In this scenario, select **lakeformation**. flink.dli.job.catalog.type=lakeformation • Name of the data catalog the Flink job accesses. flink.dli.job.catalog.name=[Catalog name in LakeFormation] The data catalog created on the DLI management console is selected, that is, the mapping between DLI and the data catalog of the default LakeFormation instance. This data catalog is connected to the data catalog of the default LakeFormation instance.	-

– **Example 3: Flink Jar jobs supporting Hudi tables**

i. Develop a Flink Jar program, compile and upload the JAR file to OBS. In this example, the file is uploaded to the **obs://obs-test/dlitest/** directory.

The sample code is as follows:

In this example, random data is generated using the DataGen table and then output to the Hudi result table.

For other connector types, see **List of Connectors Supported by Flink 1.15**.

```
package com.huawei.test;

import org.apache.flink.api.java.utils.ParameterTool;
import org.apache.flink.contrib.streaming.state.RocksDBStateBackend;
import org.apache.flink.runtime.state.filesystem.FsStateBackend;
import org.apache.flink.streaming.api.CheckpointingMode;
import org.apache.flink.streaming.api.environment.CheckpointConfig;
import org.apache.flink.streaming.api.environment.StreamExecutionEnvironment;
import org.apache.flink.table.api.EnvironmentSettings;
import org.apache.flink.table.api.bridge.java.StreamTableEnvironment;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
```

```
import java.io.IOException;
import java.text.SimpleDateFormat;

public class GenToHudiTask4 {
    private static final Logger LOGGER = LoggerFactory.getLogger(GenToHudiTask4.class);
    private static final String datePattern = "yyyy-MM-dd_HH-mm-ss";

    public static void main(String[] args) throws IOException {
        LOGGER.info("Start task.");
        ParameterTool paraTool = ParameterTool.fromArgs(args);
        String checkpointInterval = "30000";

        // set up execution environment
        StreamExecutionEnvironment env =
StreamExecutionEnvironment.getExecutionEnvironment();
        EnvironmentSettings settings = EnvironmentSettings.newInstance()
                .inStreamingMode().build();
        StreamTableEnvironment tEnv = StreamTableEnvironment.create(env, settings);

env.getCheckpointConfig().setCheckpointingMode(CheckpointingMode.EXACTLY_ONCE);
        env.getCheckpointConfig().setCheckpointInterval(Long.valueOf(checkpointInterval));
        env.getCheckpointConfig().enableExternalizedCheckpoints(
                CheckpointConfig.ExternalizedCheckpointCleanup.RETAIN_ON_CANCELLATION);

        SimpleDateFormat dateTimeFormat = new SimpleDateFormat(datePattern);
        String time = dateTimeFormat.format(System.currentTimeMillis());
        RocksDBStateBackend rocksDbBackend =
                new RocksDBStateBackend(
                        new FsStateBackend("obs://xxx/jobs/testcheckpoint/" + time), true);
        env.setStateBackend(rocksDbBackend);

        String catalog = "CREATE CATALOG hoodie_catalog\n" +
                "  WITH (\n" +
                "    'type'='hudi',\n" +
                "    'hive.conf.dir' = '/opt/hadoop/conf',\n" +
                "    'mode'='hms'\n" +
                "  )";
        tEnv.executeSql(catalog);
        String dwsSource = "CREATE TABLE if not exists genSourceJarForHudi618_1 (\n" +
                "  order_id STRING,\n" +
                "  order_name STRING,\n" +
                "  price INT,\n" +
                "  weight INT\n" +
                ") WITH (\n" +
                "  'connector' = 'datagen',\n" +
                "  'rows-per-second' = '1',\n" +
                "  'fields.order_id.kind' = 'random',\n" +
                "  'fields.order_id.length' = '8',\n" +
                "  'fields.order_name.kind' = 'random',\n" +
                "  'fields.order_name.length' = '8'\n" +
                ")";
        tEnv.executeSql(dwsSource);
/*testdb is a custom database.*/
        String printSinkdws =
                "CREATE TABLE if not exists hoodie_catalog.`testdb`.`hudiSinkJarHudi618_1`
(\n" +
                "  order_id STRING PRIMARY KEY NOT ENFORCED,\n" +
                "  order_name STRING,\n" +
                "  price INT,\n" +
                "  weight INT,\n" +
                "  create_time BIGINT,\n" +
                "  create_date String\n" +
                ") WITH (" +
                "'connector' = 'hudi',\n" +
                "'path' = 'obs://xxx/catalog/dbtest3/hudiSinkJarHudi618_1',\n" +
                "'hoodie.datasource.write.recordkey.field' = 'order_id',\n" +
                "'EXTERNAL' = 'true'\n" +
                ")";
```

```
        tEnv.executeSql(printSinkdws);
/*testdb is a custom database.*/
        String query = "insert into hoodie_catalog.`testdb`.`hudiSinkJarHudi618_1` select\n" +
        " order_id,\n" +
        " order_name,\n" +
        " price,\n" +
        " weight,\n" +
        " UNIX_TIMESTAMP() as create_time,\n" +
        " FROM_UNIXTIME(UNIX_TIMESTAMP(), 'yyyyMMdd') as create_date\n" +
        " from genSourceJarForHudi618_1";
        tEnv.executeSql(query);
    }
}
```

Table 4-19 Connector parameters for a Hudi sink table

Parameter	Description	Mandatory	Example Value
connector	Flink connector type If set to **hudi**, the sink table is a Hudi table.	Yes	hudi
path	Basic path of the table. If the path does not exist, the system will create it.	Yes	Refer to the values configured in the sample code.
hoodie.datasource.write.recordkey.field	Unique key field name of the Hoodie table	No	Set **order_id** to a unique key.
EXTERNAL	Whether the table is foreign	Yes Mandatory for the Hudi table and must be set to **true**.	true

ii. Create a Flink Jar job and set the following parameters:

Parameter	Description	Example Value
Flink Version	Flink 1.15 or later can connect to LakeFormation.	1.15

Parameter	Description	Example Value
Agency	Before using Flink 1.15 or later to run jobs, you need to create an agency on the IAM console and add the new agency information. Once set, the system automatically adds the following agency configuration to your job: flink.dli.job.agency.name=*agency* For agency permission examples, see **Creating a Custom DLI Agency** and **Agency Permission Policies in Common Scenarios**.	-
Runtime Configuration	• Type of metadata the Flink job accesses. In this scenario, select **lakeformation**. flink.dli.job.catalog.type=lakeformation • Name of the data catalog the Flink job accesses. flink.dli.job.catalog.name=[Catalog name in LakeFormation] The data catalog created on the DLI management console is selected, that is, the mapping between DLI and the data catalog of the default LakeFormation instance. This data catalog is connected to the data catalog of the default LakeFormation instance.	-

4.6.2 Permission Policies and Supported Actions for LakeFormation Resources

Supported Actions for LakeFormation SQL Resources

For the list of actions supported by DLI for SQL resource authentication, refer to **Data Permissions List**.

Table 4-20 lists the supported actions for LakeFormation SQL resources.

Table 4-20 Supported actions for LakeFormation SQL resources

Resource Type	Permission Type
Database	ALL
	ALTER
	DROP

Resource Type	Permission Type
	DESCRIBE
	LIST_TABLE
	LIST_FUNC
	CREATE_TABLE
	CREATE_FUNC
Table/View	ALL
	ALTER
	DROP
	DESCRIBE
	UPDATE
	INSERT
	SELECT
	DELETE
Column	SELECT
Function	ALL
	ALTER
	DROP
	DESCRIBE
	EXEC

LakeFormation Permission Policies

Table 4-21 LakeFormation permission policies

Type	SQL Statement	Permission to Authenticate Access to Metadata Using IAM	Permission to Authenticate Access to SQL Resources
DDL statement	ALTER DATABASE	database:describe database:alter	database:DESCRIBE database:ALTER

Type	SQL Statement	Permission to Authenticate Access to Metadata Using IAM	Permission to Authenticate Access to SQL Resources
	ALTER TABLE	database:describe table:describe table:alter database:create	database:DESCRIBE table:DESCRIBE table:ALTER database:CREATE_TABLE column:SELECT or table:SELECT
	ALTER VIEW	database:describe table:describe table:alter	database:DESCRIBE table:DESCRIBE column:SELECT table:ALTER
	CREATE DATABASE	database:describe database:create	database:DESCRIBE catalog:CREATE_DATABASE
	CREATE OR REPLACE FUNCTION (CREATE)	database:describe function:create	database:DESCRIBE database:CREATE_FUNC
	CREATE OR REPLACE FUNCTION (REPLACE)	database:describe function:describe function:alter	database:CREATE_FUNC database:DESCRIBE function:DESCRIBE function:ALTER
	CREATE TABLE	database:describe table:describe table:create	database:DESCRIBE database:CREATE_TABLE
	CREATE VIEW	database:describe table:describe table:drop table:create	database:CREATE_TABLE table:DESCRIBE (source\targer) table:DROP(target) column:SELECT
	DROP DATABASE	database:describe database:drop	database:DESCRIBE database:DROP

Type	SQL Statement	Permission to Authenticate Access to Metadata Using IAM	Permission to Authenticate Access to SQL Resources
	DROP FUNCTION	database:describe function:describe function:drop	database:DESCRIBE function:DESCRIBE function:DROP
	DROP TABLE	database:describe table:describe credential:describe table:drop	database:DESCRIBE table:DESCRIBE table:DROP
	DROP VIEW	database:describe table:describe table:drop	database:DESCRIBE table:DESCRIBE(target\source) table:DROP(target)
	REPAIR TABLE	database:describe table:describe credential:describe table:alter	database:DESCRIBE table:DESCRIBE table:ALTER table:SELECT
	TRUNCATE TABLE	database:describe table:describe table:alter	database:DESCRIBE table:DESCRIBE table:SELECT table:UPDATE
DML statement	INSERT TABLE	database:describe table:describe table:alter credential:describe	database:DESCRIBE table:DESCRIBE table:ALTER table:INSERT column:SELECT or table:SELECT
	LOAD DATA	database:describe table:describe credential:describe	database:DESCRIBE table:DESCRIBE table:UPDATE table:ALTER table:SELECT
DR statement	SELECT	database:describe table:describe credential:describe	database:DESCRIBE table:DESCRIBE column:SELECT
	EXPLAIN	Depends on the SQL statement.	Depends on the SQL statement.

Type	SQL Statement	Permission to Authenticate Access to Metadata Using IAM	Permission to Authenticate Access to SQL Resources
Auxiliary statement	ANALYZE TABLE	database:describe table:describe credential:describe table:alter	database:DESCRIBE table:DESCRIBE table:SELECT table:ALTER
	DESCRIBE DATABASE	database:describe	database:DESCRIBE
	DESCRIBE FUNCTION	database:describe function:describe	database:DESCRIBE function:DESCRIBE
	DESCRIBE QUERY	database:describe table:describe	database:DESCRIBE table:DESCRIBE table:SELECT
	DESCRIBE TABLE	database:describe table:describe	database:DESCRIBE table:DESCRIBE
	REFRESH TABLE	database:describe table:describe credential:describe	database:DESCRIBE table:DESCRIBE table:SELECT
	REFRESH FUNCTION	database:describe function:describe	database:DESCRIBE function:DESCRIBE
	SHOW COLUMNS	database:describe table:describe	database:DESCRIBE table:DESCRIBE
	SHOW CREATE TABLE	database:describe table:describe	database:DESCRIBE table:DESCRIBE
	SHOW DATABASES	database:describe	catalog:LIST_DATABASE database:DESCRIBE
	SHOW FUNCTIONS	database:describe function:describe	database:DESCRIBE
	SHOW PARTITIONS	database:describe table:describe	database:DESCRIBE table:DESCRIBE

Type	SQL Statement	Permission to Authenticate Access to Metadata Using IAM	Permission to Authenticate Access to SQL Resources
	SHOW TABLE EXTENDED	database:describe table:describe	catalog:LIST_DATABASE database:DESCRIBE table:DESCRIBE database:LIST_TABLE
	SHOW TABLES	database:describe table:describe	catalog:LIST_DATABASE database:LIST_TABLE database:DESCRIBE
	SHOW TBLPROPERTIES	database:describe table:describe	database:DESCRIBE table:DESCRIBE
	SHOW VIEWS	database:describe table:describe	catalog:LIST_DATABASE database:LIST_TABLE database:DESCRIBE

5 Data Migration and Transmission

5.1 Overview

Importing Data to OBS

DLI enables direct access to data stored in OBS for query and analysis, eliminating the need for data migration.

To begin using DLI for data analysis, just import your local data into OBS.

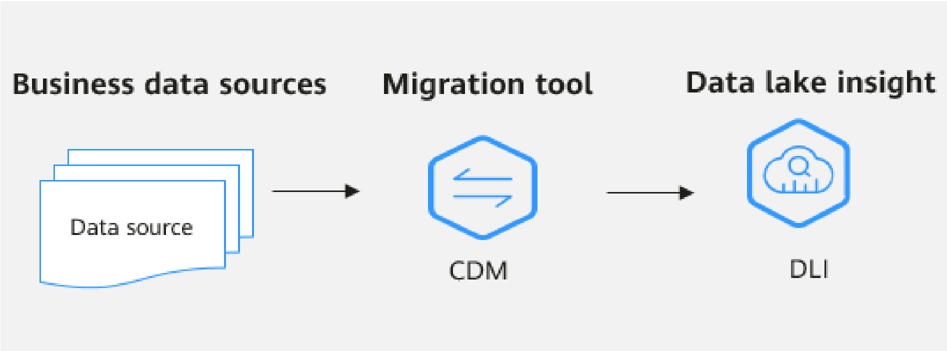
Migrating Data to DLI

To centrally analyze and manage scattered data from various systems, you can utilize migration tools like Cloud Data Migration (CDM) to migrate the data to DLI. Once the migration is complete, you can submit DLI jobs to analyze the data.

CDM supports multiple types of data sources, such as databases, data warehouses, and files. You can configure data source migration tasks on the GUI to enhance the efficiency of data migration and integration.

For details, see **Migrating Data from External Data Sources to DLI**.

Figure 5-1 Migrating data to DLI



Configuring DLI to Read and Write External Data Sources

If you prefer not to import data into OBS or DLI tables, DLI offers cross-source access capabilities, allowing you to connect to data sources for analysis without the need to migrate the data.

For details, see **Configuring DLI to Read and Write External Data Sources**.

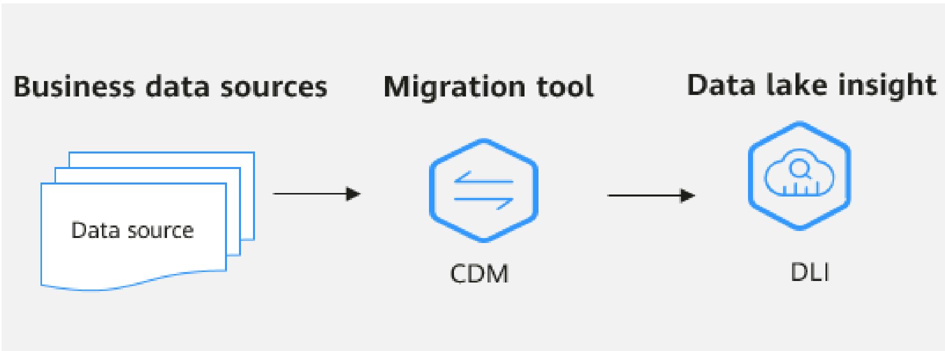
5.2 Migrating Data from External Data Sources to DLI

5.2.1 Overview of Data Migration Scenarios

To centrally analyze and manage scattered data from various systems, you can utilize migration tools like Cloud Data Migration (CDM) to migrate the data to DLI. Once the migration is complete, you can submit DLI jobs to analyze the data.

CDM supports multiple types of data sources, such as databases, data warehouses, and files. You can configure data source migration tasks on the GUI to enhance the efficiency of data migration and integration.

Figure 5-2 Migrating data to DLI



Common Migration Scenarios and Solutions

Table 5-1 Common migration scenarios and solutions

Data Type	Migration Tool	Solution
Hive	CDM	**Example Typical Scenario: Migrating Data from Hive to DLI**
Kafka	CDM	**Example Typical Scenario: Migrating Data from Kafka to DLI**
Elasticsearch	CDM	**Example Typical Scenario: Migrating Data from Elasticsearch to DLI**

Data Type	Migration Tool	Solution
RDS	CDM	**Example Typical Scenario: Migrating Data from RDS to DLI**
GaussDB(DWS)	CDM	**Example Typical Scenario: Migrating Data from GaussDB(DWS) to DLI**

Data Type Mapping

Refer to **Table 5-2** for data type mapping between data sources and destinations during data migration between DLI and other cloud services and platforms. This will aid in data type conversion and mapping.

Table 5-2 Data type mapping

MySQL	Hive	GaussDB(DWS)	Oracle	PostgreSQL	Hologres	DLI Spark
CHAR	CHAR	CHAR	CHAR	CHAR	CHAR	CHAR
VARCHAR	VARCHAR	VARCHAR	VARCHAR	VARCHAR	VARCHAR	VARCHAR/ STRING
DECIMAL	DECIMAL	NUMERIC	NUMERIC	NUMERIC	DECIMAL	DECIMAL
INT	INT	INTEGER	NUMBER	INTEGER	INTEGER	INT
BIGINT	BIGINT	BIGINT	NUMBER	BIGINT	BIGINT	BIGINT/ LONG
TINYINT	TINYINT	SMALLINT	NUMBER	SMALLINT	SMALLINT	TINYINT
SMALLINT	SMALLINT	SMALLINT	NUMBER	SMALLINT	SMALLINT	SMALLINT/SHORT
BINARY	BINARY	BYTEA	RAW	BYTEA	BYTEA	BINARY
VARBINARY	BINARY	BYTEA	RAW	BYTEA	BYTEA	BINARY
FLOAT	FLOAT	FLOAT4	FLOAT	DOUBLE	FLOAT4	FLOAT
DOUBLE	DOUBLE	FLOAT8	FLOAT	REAL/ DOUBLE	FLOAT8	DOUBLE
DATE	DATE	TIMESTAMP	DATE	DATE	DATE	DATE

MySQL	Hive	GaussDB(DWS)	Oracle	PostgreSQL	Hologres	DLI Spark
TIME	Not supported (use String instead)	TIME	DATE	TIME	TIME	Not supported (use String instead)
DATETIME	TIMESTAMP	TIMESTAMP	TIME	TIME	TIMESTAMP	TIMESTAMP
TINYINT	TINYINT	BOOLEAN	Not supported	TINYINT	BOOLEAN	BOOLEAN
Not supported (use TEXT instead)	Not supported (use String instead)	Not supported (use TEXT instead)	Not supported (use VARCHAR instead)	Not supported (use TEXT instead)	Not supported (use TEXT instead)	ARRAY
Not supported (use TEXT instead)	Not supported (use String instead)	Not supported (use TEXT instead)	Not supported (use VARCHAR instead)	Not supported (use TEXT instead)	Not supported (use TEXT instead)	MAP
Not supported (use TEXT instead)	Not supported (use String instead)	Not supported (use TEXT instead)	Not supported (use VARCHAR instead)	Not supported (use TEXT instead)	Not supported (use TEXT instead)	STRUCT

NOTE

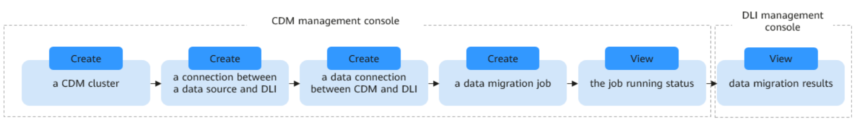
If a service does not support a standard data type, you can use the recommended data type.

5.2.2 Using CDM to Migrate Data to DLI

On its GUI, CDM enables you to create data migration tasks from multiple data sources to a data lake.

This section describes how to use CDM to migrate data from data sources to DLI.

Figure 5-3 Process of migrating data to DLI using CDM



Step 1: Create a CDM Cluster

A CDM cluster is used to execute data migration jobs that migrate data from data sources to DLI.

1. Log in to the CDM management console.
2. Click **Buy CDM Cluster** in the upper right corner. On the displayed page, set CDM cluster parameters.
   - You are advised to set the region, VPC, subnet, security group, and enterprise project to be the same as those of the data source and DLI.
   - Once a cluster is created, its specifications cannot be modified. If you need to use higher specifications, you will need to create a new cluster.

   For how to set CDM cluster parameters, see **Creating a Cluster**.
3. Click **Buy Now**.
4. Confirm the settings and click **Submit**. The system starts to create a CDM cluster. You can check the creation progress on the **Cluster Management** page.

Step 2: Create a Data Connection Between the Data Source and CDM

This step uses the MySQL data source as an example to describe how to create a data connection between the data source and CDM.

Step 1 In the navigation pane on the left of the CDM console, choose **Cluster Management**. Locate the **cdm-aff1** cluster created in **Step 1: Create a CDM Cluster**.

Step 2 Click **Job Management** in the **Operation** column.

Step 3 Click the **Link Management** tab then **Create Link**.

Figure 5-4 Selecting a connector

Step 4 Select **RDS for MySQL** and click **Next** to set the link parameters.

Click **Show Advanced Attributes** to view more optional parameters. For details, see **Link to an RDS for MySQL/MySQL Database**. Retain the default values for the optional parameters and set the mandatory parameters based on **Table 5-3**.

Table 5-3 MySQL link parameters

Parameter	Description	Example Value
Name	Enter a unique link name.	mysqllink
Database Server	IP address or domain name of the MySQL database	-
Port	Port number of the MySQL database	3306
Database Name	Name of the MySQL database	sqoop
Username	User who has the read, write, and delete permissions on the MySQL database	admin
Password	Password of the user	-
Use Local API	Whether to use the local API of the database for acceleration. (The system attempts to enable the **local_infile** system variable of the MySQL database.)	Yes
Use Agent	This parameter does not need to be set as the agent function will be unavailable soon.	-
local_infile Character Set	When using local_infile to import data to MySQL, you can set the encoding format.	utf8
Driver Version	Before connecting CDM to a relational database, you need to upload the JDK 8 .jar driver of the relational database. Download the MySQL driver 5.1.48 from **https://downloads.mysql.com/archives/c-j/**, obtain **mysql-connector-java-5.1.48.jar**, and upload it.	-

Step 5 Click **Test** to check whether the parameters are correctly configured. If the test is successful, click **Save** to create the link and return to the **Links** page.

----End

Step 3: Create a Data Connection Between CDM and DLI

Step 1 In the navigation pane on the left of the CDM console, choose **Cluster Management**. Locate the **cdm-aff1** cluster created in **Step 1: Create a CDM Cluster**.

Step 2 Click **Job Management** in the **Operation** column.

Step 3 Click the **Link Management** tab then **Create Link**.

Figure 5-5 Selecting a connector

Step 4 Select **Data Lake Insight** for **Data Warehouse** and click **Next**. On the displayed page, set DLI link parameters.

Click **Show Advanced Attributes** to view optional parameters. For details, see **Link to DLI**. Retain the default values for the optional parameters and set the mandatory parameters based on **Table 5-4**.

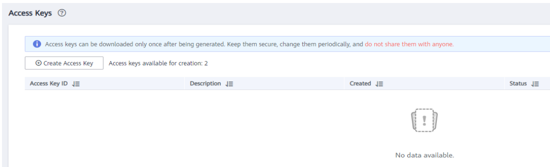
Table 5-4 DLI link parameters

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	dlilink

<table>
<tr><th>Parameter</th><th>Description</th><th>Example Value</th></tr>
<tr><td>Access Key</td><td rowspan="2">AK/SK required for authentication during access to the DLI database.

You need to create an access key for the current account and obtain an AK/SK pair.

1. Log in to the management console, move the cursor to the username in the upper right corner, and select My Credentials from the drop-down list.
2. On the My Credentials page, choose Access Keys, and click Create Access Key. See Figure 5-6.

Figure 5-6 Clicking Create Access Key


3. Click OK and save the access key file as prompted. The access key file will be saved to your browser's configured download location. Open the credentials.csv file to view Access Key Id and Secret Access Key.
NOTE
– Only two access keys can be added for each user.
– To ensure access key security, the access key is automatically downloaded only when it is generated for the first time and cannot be obtained from the management console later. Keep them properly.</td><td>-</td></tr>
<tr><td>Secret Key</td><td>-</td></tr>
<tr><td>Project ID</td><td>Project ID of the region where DLI is

A project is a group of tenant resources, and an account ID corresponds to the current account. The IAM ID corresponds to the current user. You can view the project IDs, account IDs, and user IDs in different regions on the corresponding pages.

1. Register with and log in to the management console.
2. Hover the cursor on the username in the upper right corner and select My Credentials from the drop-down list.
3. On the API Credentials page, obtain the account name, account ID, IAM username, and IAM user ID, and obtain the project and its ID from the project list.</td><td>-</td></tr>
</table>

Step 5 Click **Test** to check whether the parameters are correctly configured. If the test is successful, click **Save** to create the link and return to the **Links** page.

----End

Step 4: Create a Data Migration Job on CDM

After establishing the data connection between the data source and CDM, as well as between CDM and DLI, you need to create a data migration job to migrate the data from the data source to DLI.

Step 1 On the **Cluster Management** page, locate the **cdm-aff1** cluster created in **Step 1: Create a CDM Cluster**.

Step 2 Click **Job Management** in the **Operation** column.

Step 3 Click the **Table/File Migration** tab, click **Create Job**, and configure basic job information.

- **Job Name**: Enter a unique job name, for example, **mysql2dli**.
- Source job parameters
  - **Source Link Name**: Select the MySQL link **mysqllink**.
  - **Use SQL**: Select **No**.
  - **Schema/Tablespace**: Select the MySQL database from which the table is to be exported.
  - **Table Name**: Select the table from which data is to be exported.
  - Retain the default values for other optional parameters. For details, see **From MySQL**.
- Destination job parameters
  - **Destination Link Name**: Select the DLI link **dlilink**.
  - **Schema/Tablespace**: Select the schema to which data is to be imported.
  - **Auto Table Creation**: Select **Auto creation**. If the table specified by **Table Name** does not exist, CDM automatically creates the table in DLI.
  - **Table Name**: Select the table to which data is to be imported.
  - **Advanced Attributes** > **Extend Field Length**: Select **Yes**. The encoding techniques used for storing Chinese characters in MySQL and DLI differ, and the required lengths also vary. In UTF-8 encoding, a Chinese character can take up to three bytes. If this parameter is set to **Yes**, DLI will automatically create tables with character fields that are three times the length of the original table to avoid errors caused by insufficient character field length in DLI tables.
  - Retain the default values for other optional parameters. For details, see **To DWS**.

Step 4 Click **Next**. The **Map Field** tab page is displayed. CDM automatically maps the fields in the source and destination data tables. You need to check if the field mapping is correct.

- If the mapping is incorrect, click the row where the field is located and hold down the left mouse button to drag the field to adjust the mapping.

- When importing data into DLI, you must manually choose the distribution columns. You are advised to select the distribution columns based on the following principles:
  a. Use the primary key as the distribution column.
  b. If multiple data segments are combined as primary keys, specify all primary keys as the distribution column.
  c. In the scenario where no primary key is available, if no distribution column is selected, DWS uses the first column as the distribution column by default. As a result, data skew risks exist.
- To convert the content of the source fields, perform the operations in this step. For details, see **Converting Fields**. In this example, field conversion is not required.

Step 5 Click **Next** and set task parameters. Typically, retain the default values for all parameters.

In this step, you can configure the following optional features:

- **Retry Upon Failure**: If the job fails to be executed, you can determine whether to automatically retry. Retain the default value **Never**.
- **Group**: Select the group to which the job belongs. The default group is **DEFAULT**. On the **Job Management** page, jobs can be displayed, started, or exported by group.
- **Schedule Execution**: Determine whether to automatically execute the job at a scheduled time. Retain the default value **No**.
- **Concurrent Extractors**: Enter the number of concurrent extractors. An appropriate value improves migration efficiency. For details, see **Performance Tuning**. Retain the default value **1**.
- **Write Dirty Data**: Specify this parameter if data that fails to be processed or filtered out during job execution needs to be written to OBS for future viewing. Before writing dirty data, create an OBS link on the CDM console. Retain the default value **No**, meaning dirty data is not recorded.

Step 6 Click **Save and Run**. CDM starts to execute the job immediately.

----End

Step 5: View Data Migration Results

This step describes how to view a job's execution results and its historical information within the past 90 days, including the number of written rows, read rows, written bytes, written files, and log information.

- **Viewing the status of the migration job on CDM**
  a. On the **Cluster Management** page, locate the **cdm-aff1** cluster created in **Step 1: Create a CDM Cluster**.
  b. Click **Job Management** in the **Operation** column.
  c. Locate the **mysql2dli** job created in **Step 4: Create a Data Migration Job on CDM** and check its execution status. If the job status is **Succeeded**, the migration is successful.
- **Viewing data migration results on DLI**

a. After the CDM migration job is complete, log in to the DLI management console.

b. In the navigation pane on the left, choose **SQL Editor**.

   Set **Engine** to **Spark**, **Queues** to the created SQL queue, and **Databases** to the created database. Run the following DLI table query statement to check whether the MySQL data has been successfully migrated to the DLI table:

```
select * from tablename;
```

5.2.3 Example Typical Scenario: Migrating Data from Hive to DLI

This section describes how to use CDM's data synchronization to migrate data from MRS Hive to DLI. All other MRS Hadoop component data can be synchronized bidirectionally with DLI using CDM.

Prerequisites

- You have created a DLI SQL queue.

  CAUTION

  Set **Type** to **For SQL** when buying a queue.

- You have created an MRS security cluster that contains the Hive component.
  - In this example, the MRS cluster and component versions are as follows:
    - Cluster version: MRS 3.1.0
    - Hive version: 3.1.0
    - Hadoop version: 3.1.1
  - In this example, Kerberos authentication is enabled when the MRS cluster is created.
- You have created a CDM cluster. For how to create a CDM cluster, see **Creating a CDM Cluster**.

NOTE

- To connect the cluster to an on-premises database as the destination data source, you can use either Internet or Direct Connect. If the Internet is used, make sure that an EIP has been bound to the CDM cluster, the security group of CDM allows outbound traffic from the host where the on-premises data source is located, the host where the data source is located can access the Internet, and the connection port has been enabled in the firewall rules.
- To successfully connect to cloud services like MRS and GaussDB (DWS) as data sources, the following requirements must be met:

  i. If the CDM cluster and the cloud service are in different regions, they must be connected through either the Internet or Direct Connect. If the Internet is used, make sure that an EIP has been bound to the CDM cluster, the host where the data source is located can access the Internet, and the port has been enabled in the firewall rules.

  ii. If the CDM cluster and the cloud service are in the same region, instances in the same VPC, subnet, and security group can communicate with each other by default. If the CDM cluster and the cloud service are in the same VPC but in different subnets or security groups, you must configure routing and security group rules.

  For how to configure routing rules, see **Configure routes**. For how to configure security group rules, see **Security Group Configuration Examples**.

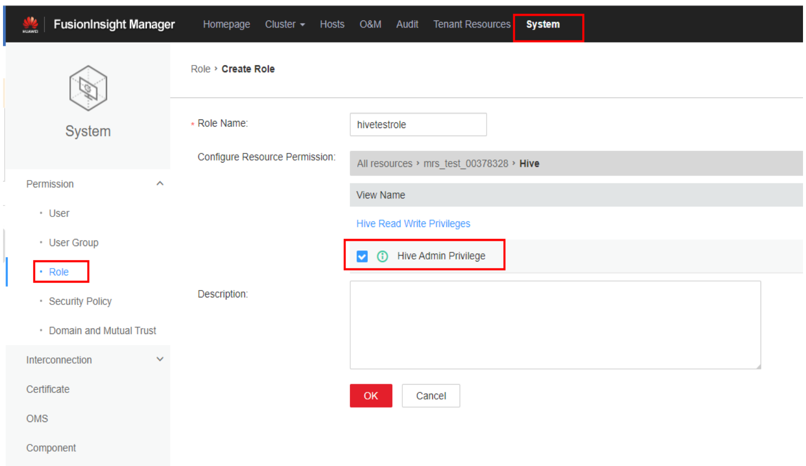
  iii. The cloud service instance and the CDM cluster belong to the same enterprise project. If they do not, change the enterprise project of the workspace.

In this example, the VPC, subnet, and security group of the CDM cluster match those of the MRS cluster.

Step 1: Prepare Data

- Create a Hive table in the MRS cluster and insert data into the table.
  a. Log in to MRS Manager by referring to **Accessing FusionInsight Manager**.
  b. Click **System** and choose **Permission** > **Role**. On the displayed page, set the following parameters:
     - **Role Name**: Enter a role name, for example, **hivetestrole**.
     - **Configure Resource Permission**: Select the MRS cluster name and then **Hive**. Select **Hive Admin Privilege**.

Figure 5-7 Creating a Hive role



For how to create a role, see **Creating a Role**.

c. Click **System** and choose **Permission** > **User**. On the displayed page, set the following parameters:

i. **Username**: Enter a username. In this example, enter **hivetestusr**.

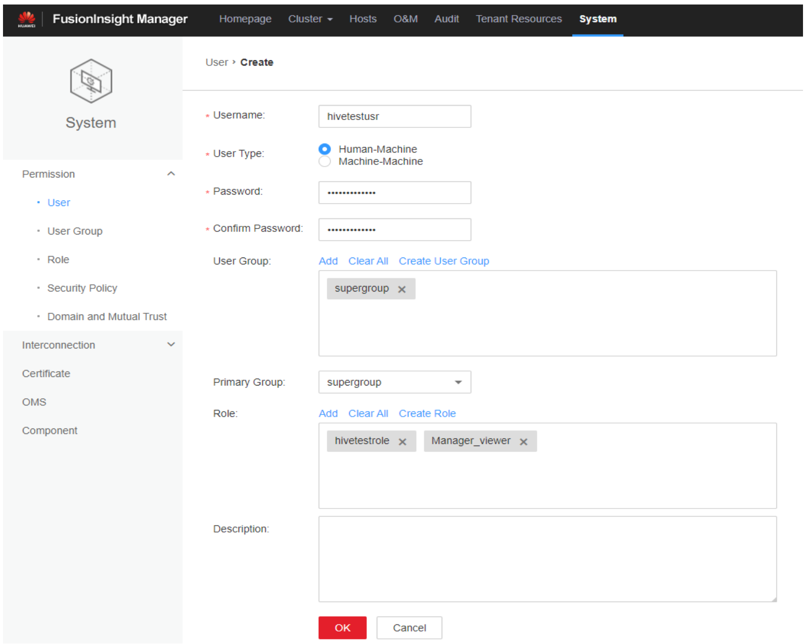
ii. **User Type**: Select **Human-Machine**.

iii. **Password** and **Confirm Password**: Enter the current user's password and confirm it by entering it again.

iv. **User Group** and **Primary Group**: Select **supergroup**.

v. **Role**: Select the role created in **b** and the **Manager_viewer** role.

Figure 5-8 Creating a Hive user



d. Download and install the Hive client by referring to **Installing an MRS Client**. For example, the current Hive client is installed in the **/opt/hiveclient** directory of the active MRS node.

e. Go to the client installation directory as user **root**.

Example: **cd /opt/hiveclient**

f. Configure environment variables.

source bigdata_env

g. Authenticate the user created in **c** as Kerberos authentication has been enabled for the cluster:

kinit *Username in c*

Example: **kinit hivetestusr**

h. Connect to Hive.

beeline

i. Create a table and insert data into it.

Create a table.

```
create table user_info(id string,name string,gender string,age int,addr string);
```

Insert data into the table.

```
insert into table user_info(id,name,gender,age,addr) values("12005000201","A","Male",19,"City A");
insert into table user_info(id,name,gender,age,addr) values("12005000202","B","Male",20,"City B");
insert into table user_info(id,name,gender,age,addr) values("12005000202","B","Male",20,"City B");
```

NOTE

In the preceding example, data is migrated by creating a table and inserting data into the table. To migrate existing Hive databases and table data, obtain information about them.

- Run the following command on the Hive client to obtain database information:

  show databases

- Switch to the Hive database to migrate.

  use *Hive database name*

- Displays information about all tables in the current database.

  show tables

- Query the Hive table creation statement.

  show create table *Hive table name*

  The table creation statement obtained from the query needs to be modified to match DLI's syntax before running it on DLI.

- Create a database and table on DLI.

  a. Log in to the DLI management console. In the navigation pane on the left, choose **SQL Editor**. On the displayed page, set **Engine** to **Spark** and **Queues** to the created SQL queue.

     Create a database, for example, **testdb**. For the syntax to create a DLI database, see **Creating a Database**.

     ```
     create database testdb;
     ```

  b. Create a table in the database.

     NOTE

     You need to modify the table creation statement obtained by running **show create table hive_table_name** to comply with DLI's table creation syntax. For the table creation syntax, see **Creating a DLI Table Using the DataSource Syntax**.

     ```
     create table user_info(id string,name string,gender string,age int,addr string);
     ```

Step 2: Migrate Data

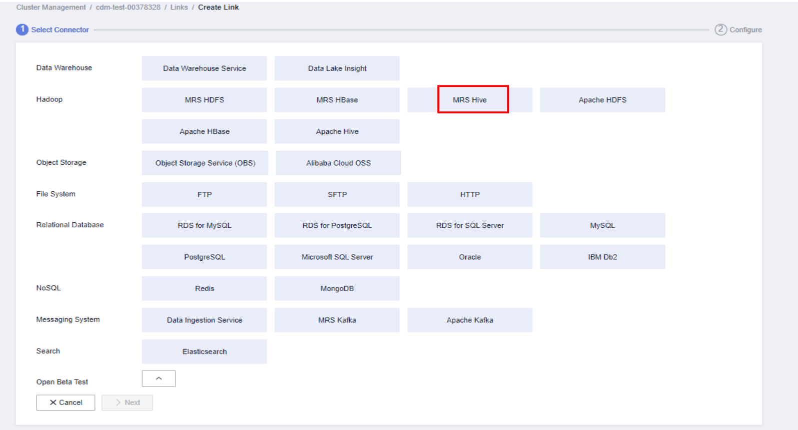
1. Create a CDM connection.

   a. Create a connection to link CDM to MRS Hive.

      i. Log in to the CDM console. In the navigation pane on the left, choose **Cluster Management**. On the displayed page, locate the created CDM cluster and click **Job Management** in the **Operation** column.

      ii. On the **Job Management** page, click the **Links** tab. On this tab page, click **Create Link**. On the displayed page, select **MRS Hive** and click **Next**.

Figure 5-9 Selecting the MRS Hive connector



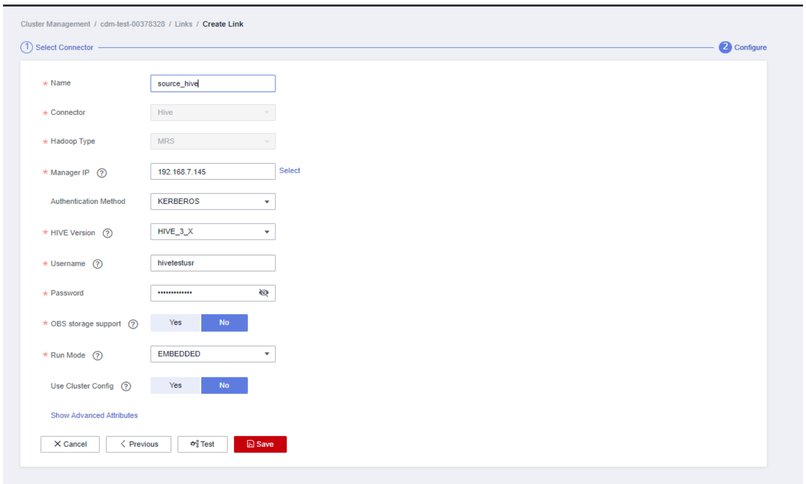
iii. Configure the connection as follows:

Table 5-5 MRS Hive connection configurations

Parameter	Value
Name	Name of the MRS Hive data source, for example, **source_hive**.
Manager IP	Manager IP address, which is automatically filled in after you click **Select** next to the text box and select the MRS Hive cluster.
Authentication Method	Set it to **KERBEROS** if Kerberos authentication is enabled for the MRS cluster or to **SIMPLE** if the MRS cluster is a common cluster. In this example, set it to **KERBEROS**.
Hive Version	Set it to the Hive version you selected when creating the MRS cluster. Select **HIVE_3_X** if the current Hive version is 3.1.0.
Username	Name of the MRS Hive user created in **c**.
Password	Password of the MRS Hive user.

Retain the default values for other parameters.

Figure 5-10 Configuring the connection to MRS Hive



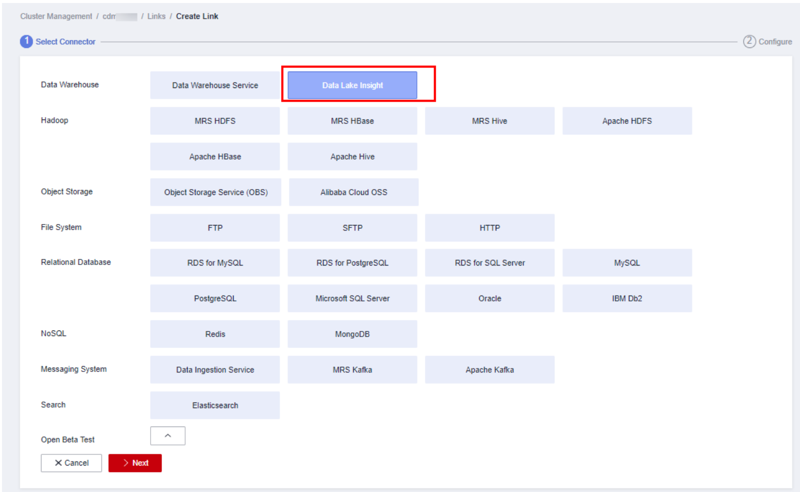
iv. Click **Save**.

b. Create a connection to link CDM to DLI.

i. Log in to the CDM console. In the navigation pane on the left, choose **Cluster Management**. On the displayed page, locate the created CDM cluster and click **Job Management** in the **Operation** column.

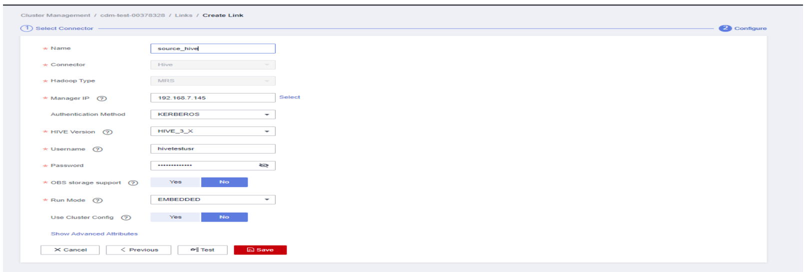
ii. On the **Job Management** page, click the **Links** tab, and click **Create Link**. On the displayed page, select **Data Lake Insight** and click **Next**.

Figure 5-11 Selecting the DLI connector



iii. Set the connection parameters.

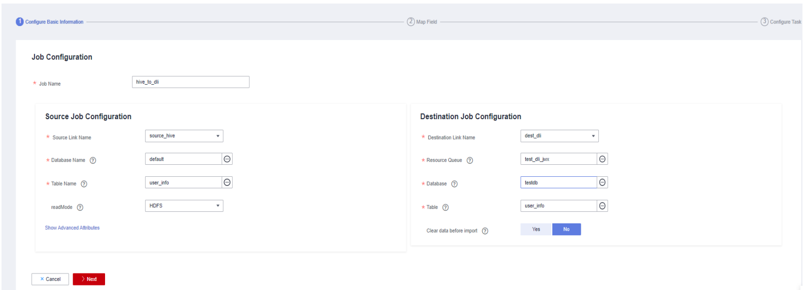
Figure 5-12 Setting connection parameters



Click **Save**.

2. Create a CDM migration job.
   a. Log in to the CDM console. In the navigation pane on the left, choose **Cluster Management**. On the displayed page, locate the created CDM cluster and click **Job Management** in the **Operation** column.
   b. On the **Job Management** page, click the **Table/File Migration** tab. On the displayed tab, click **Create Job**.
   c. On the **Create Job** page, set job parameters.

Figure 5-13 Setting CDM job parameters



   i. **Job Name**: Name of the data migration job, for example, **hive_to_dli**.
   ii. Set the parameters in the **Source Job Configuration** area as follows:

Table 5-6 Source job parameters

Parameter	Value
Source Link Name	Select the name of the data source created in **1.a**.
Database Name	Select the name of the MRS Hive database you want to migrate to DLI. For example, the **default** database.
Table Name	Select the name of the Hive table. In this example, a database created on DLI and the **user_info** table are selected.

Parameter	Value
readMode	In this example, set it to **HDFS**. The options are described as follows: There are two read modes available: HDFS and JDBC. HDFS is used by default. If you do not need to use the WHERE condition to filter data or add fields on the field mapping page, select **HDFS**. When using the HDFS mode, data reading performance is good, but it does not support filtering data using WHERE conditions or adding fields on the field mapping page. The JDBC mode allows you to use WHERE conditions to filter data or add fields on the field mapping page.

For details about parameter settings, see **From Hive**.

iii. Set the parameters in the **Destination Job Configuration** area as follows:

Table 5-7 Destination job parameters

Parameter	Value
Destination Link Name	Select the DLI data source connection created in **1.b**.
Resource Queue	Select a created DLI SQL queue.
Database Name	Select a created DLI database. In this example, database **testdb** created in **Create a database and table on DLI** is selected.
Table Name	Select the name of a table in the database. In this example, table **user_info** created in **Create a database and table on DLI** is selected.
Clear data before import	Whether to clear data in the destination table before data import. In this example, set it to **No**. If set to **Yes**, data in the destination table will be cleared before the task is started.

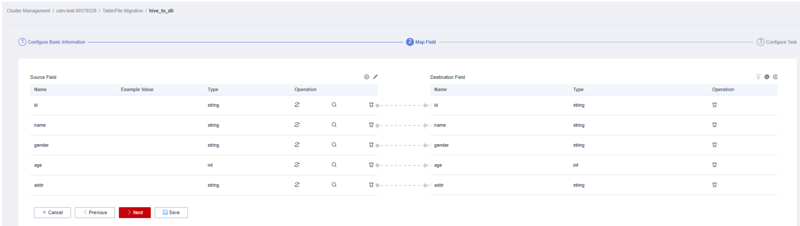
For details about parameter settings, see **To DLI**.

3. Click **Next**. The **Map Field** page is displayed. CDM automatically matches the source and destination fields.
   - You can drag any unmatched fields to match them.

– If the type is automatically created at the migration destination, you need to configure the type and name of each field.

– CDM allows for field conversion during migration. For details, see **Field Conversion**.

Figure 5-14 Field mapping



4. Click **Next** and set task parameters. Typically, retain the default values for all parameters.

   In this step, you can configure the following optional features:

   – **Retry Upon Failure**: If the job fails to be executed, you can determine whether to automatically retry. Retain the default value **Never**.

   – **Group**: Select the group to which the job belongs. The default group is **DEFAULT**. On the **Job Management** page, jobs can be displayed, started, or exported by group.

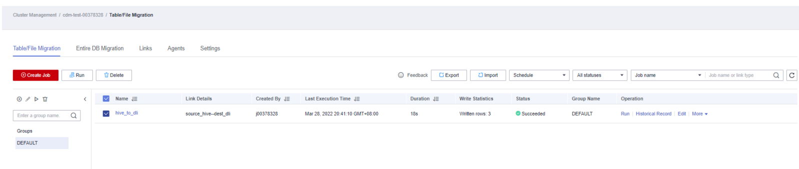
   – **Scheduled Execution**: For how to configure scheduled execution, see **Scheduling Job Execution**. Retain the default value **No**.

   – **Concurrent Extractors**: Enter the number of extractors to be concurrently executed. Retain the default value **1**.

   – **Write Dirty Data**: Set this parameter if data that fails to be processed or filtered out during job execution needs to be written to OBS. Before writing dirty data, create an OBS link. You can view the data on OBS later. Retain the default value **No**, meaning dirty data is not recorded.

5. Click **Save and Run**. On the **Job Management** page, you can view the job execution progress and result.

Figure 5-15 Job progress and execution result

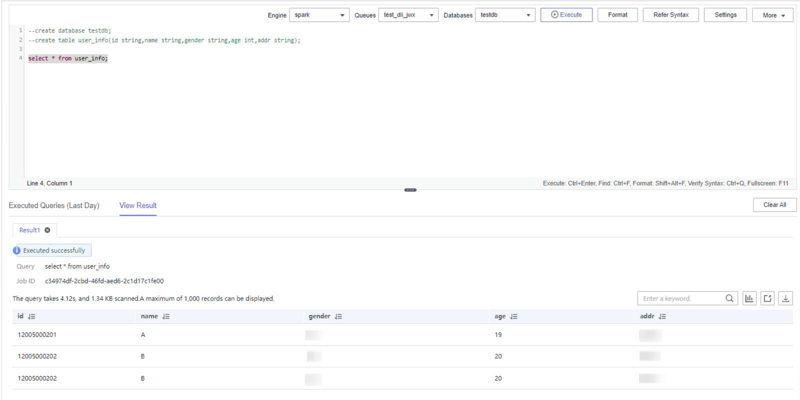


Step 3: Query Results

Once the migration job is complete, check whether the Hive table data has been migrated to the **user_info** table. Specifically, do as follows: Log in to the DLI management console and choose **SQL Editor**. On the displayed page, set **Engine** to **Spark**, **Queues** to the created SQL queue, and **Databases** to the database created in **a**. Then, execute the following query statement:

```
select * from user_info;
```

Figure 5-16 Querying migrated data



5.2.4 Example Typical Scenario: Migrating Data from Kafka to DLI

This section describes how to use CDM's data synchronization to migrate data from MRS Kafka to DLI.

Prerequisites

- You have created a DLI SQL queue. For how to create a DLI queue, see **Creating a Queue**.

> **CAUTION**
>
> Set **Type** to **For SQL** when buying a queue.

- You have created an MRS security cluster that contains the Kafka component. For how to create an MRS cluster, see **Purchasing a Custom Cluster**.
  - In this example, the version of the MRS cluster is 3.1.0.
  - You have enabled Kerberos authentication for the MRS cluster.
- You have created a CDM cluster. For how to create a CDM cluster, see **Creating a CDM Cluster**.

NOTE

- To connect the cluster to an on-premises database as the destination data source, you can use either Internet or Direct Connect. If the Internet is used, make sure that an EIP has been bound to the CDM cluster, the security group of CDM allows outbound traffic from the host where the on-premises data source is located, the host where the data source is located can access the Internet, and the connection port has been enabled in the firewall rules.
- To successfully connect to cloud services MRS and GaussDB (DWS) as data sources, the following requirements must be met:

  i. If the CDM cluster and the cloud service are in different regions, they must be connected through either the Internet or Direct Connect. If the Internet is used, make sure that an EIP has been bound to the CDM cluster, the host where the data source is located can access the Internet, and the port has been enabled in the firewall rules.

  ii. If the CDM cluster and the cloud service are in the same region, instances in the same VPC, subnet, and security group can communicate with each other by default. If the CDM cluster and the cloud service are in the same VPC but in different subnets or security groups, you must configure routing and security group rules.

  For how to configure routing rules, see **Configure routes**. For how to configure security group rules, see **Security Group Configuration Examples**.

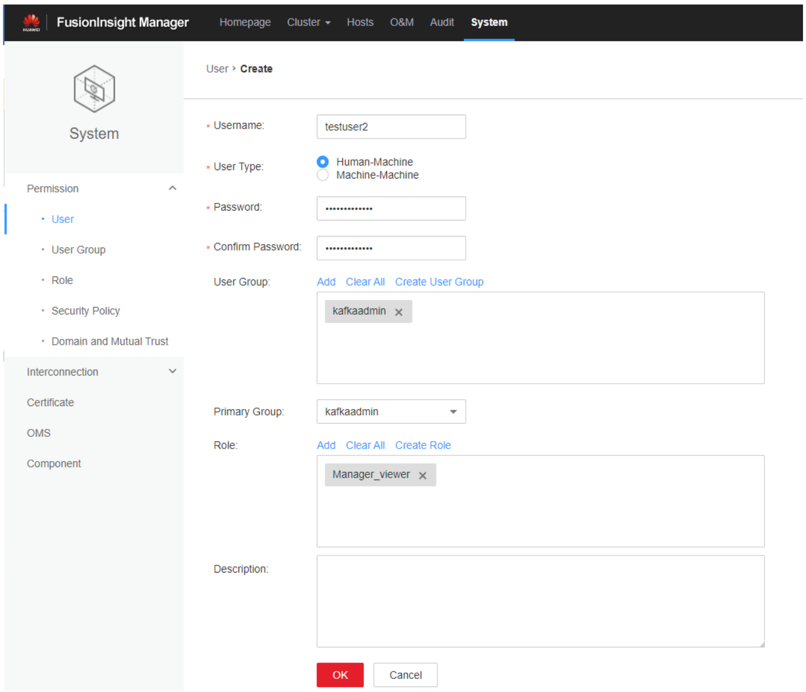
  iii. The cloud service instance and the CDM cluster belong to the same enterprise project. If they do not, change the enterprise project of the workspace.

In this example, the VPC, subnet, and security group of the CDM cluster match those of the MRS cluster.

Step 1: Prepare Data

- Create a Kafka topic for the MRS cluster and send messages to the topic.
  a. Log in to MRS Manager by referring to **Accessing FusionInsight Manager**.
  b. Click **System** and choose **Permission** > **User**. On the displayed page, set the following parameters:
     i. **Username**: Enter a username. In this example, enter **testuser2**.
     ii. **User Type**: Select **Human-Machine**.
     iii. **Password** and **Confirm Password**: Enter the current user's password and confirm it by entering it again.
     iv. **User Group** and **Primary Group**: Select **kafkaadmin**.
     v. **Role**: Select **Manager_viewer**.

Figure 5-17 Creating a Kafka user



c. On MRS Manager, choose **Cluster** > *Name of the desired cluster* > **Service** > **ZooKeeper** > **Instance**. On the displayed page, obtain the IP address of the ZooKeeper instance.

d. On MRS Manager, choose **Cluster** > *Name of the desired cluster* > **Service** > **Kafka** > **Instance**. On the displayed page, obtain the IP address of the Kafka instance.

e. Download and install the Kafka client by referring to **Installing an MRS Client**. For example, the current Hive client is installed in the **/opt/kafkaclient** directory of the active MRS node.

f. Go to the client installation directory as user **root**.

Example: **cd /opt/kafkaclient**

g. Configure environment variables.

source bigdata_env

h. Authenticate the user created in **b** as Kerberos authentication has been enabled for the cluster:

kinit *Username in* ***b***

Example: **kinit testuser2**

i. Create a Kafka topic named **kafkatopic**.

```
kafka-topics.sh --create --zookeeper IP address 1 of the node where the ZooKeeper role is:2181,IP address 2 of the node where the ZooKeeper role is:2181,IP address 3 of the node where the ZooKeeper role is:2181/kafka --replication-factor 1 --partitions 1 --topic kafkatopic
```

In this command, *IP address of the node where the ZooKeeper role is* is the IP address of the ZooKeeper instance obtained in **c**.

j. Send a test message to **kafkatopic**.

```
kafka-console-producer.sh --broker-list IP address 1 of the node where the Kafka role is::21007;IP address 2 of the node where the Kafka role is::21007;IP address 3 of the node where the Kafka role is::21007 --topic kafkatopic --producer.config /opt/kafkaclient/Kafka/kafka/config/producer.properties
```

In this command, *IP address of the node where the Kafka role is* is the IP address of the Kafka instance obtained in **d**.

The content of the test message is as follows:

```
{"PageViews":5, "UserID":"4324182021466249494", "Duration":146,"Sign":-1}
```

- Create a database and table on DLI.
  a. Log in to the DLI management console. In the navigation pane on the left, choose **SQL Editor**. On the displayed page, set **Engine** to **Spark** and **Queues** to the created SQL queue.

     Create a database, for example, **testdb**. For the syntax to create a DLI database, see **Creating a Database**.

     ```
     create database testdb;
     ```

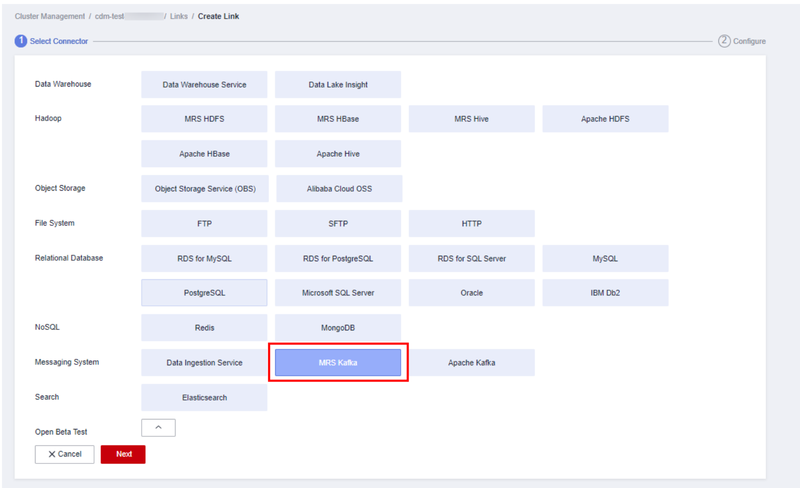
  b. Create a table in the database. For the table creation syntax, see **Creating a DLI Table Using the DataSource Syntax**.

     ```
     CREATE TABLE testdlitable(value STRING);
     ```

Step 2: Migrate Data

1. Create a CDM connection.
   a. Create a connection to link CDM to MRS Kafka.
      i. Log in to the CDM console. In the navigation pane on the left, choose **Cluster Management**. On the displayed page, locate the created CDM cluster and click **Job Management** in the **Operation** column.
      ii. On the **Job Management** page, click the **Links** tab and click **Create Link**. On the displayed page, select **MRS Kafka** and click **Next**.

Figure 5-18 Selecting the MRS Kafka connector



      iii. Configure the connection as follows:

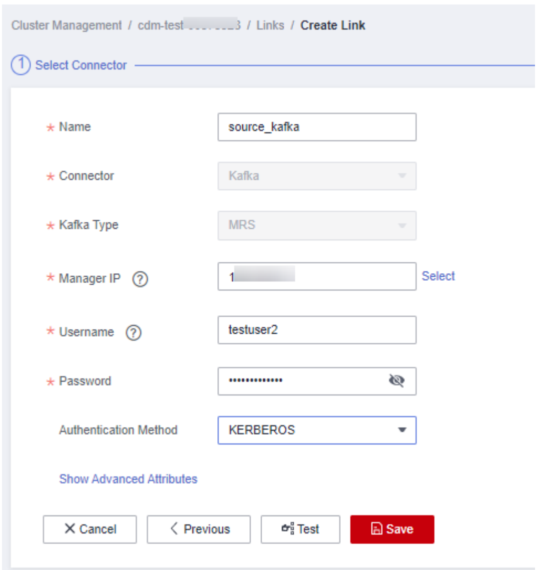
Table 5-8 MRS Kafka connection configurations

Parameter	Value
Name	Name of the MRS Kafka data source, for example, **source_kafka**.

Parameter	Value
Manager IP	Manager IP address, which is automatically filled in after you click **Select** next to the text box and select the MRS Kafka cluster.
Username	Name of the MRS Kafka user created in **b**.
Password	Password of the MRS Kafka user.
Authenticatio n Method	Set it to **KERBEROS** if Kerberos authentication is enabled for the MRS cluster or to **SIMPLE** if the MRS cluster is a common cluster. In this example, set it to **KERBEROS**.

For more details about the parameters, see **Link to Kafka**.

Figure 5-19 Configuring the MRS Kafka connection



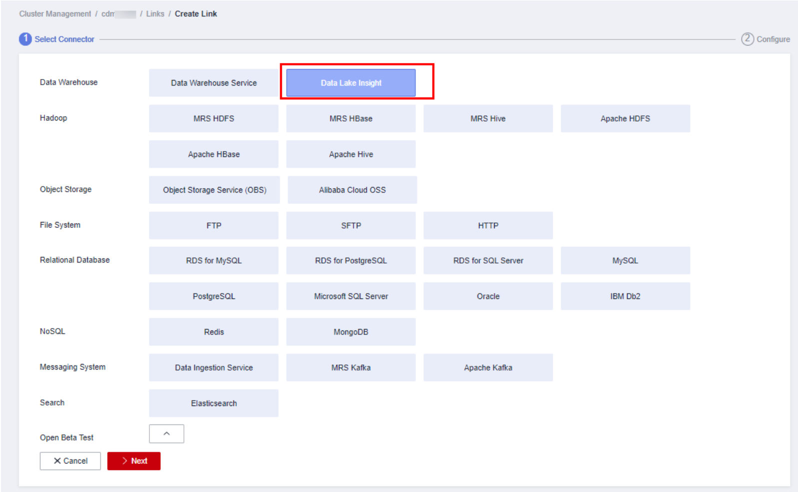
iv. Click **Save**.

b. Create a connection to link CDM to DLI.

i. Log in to the CDM console. In the navigation pane on the left, choose **Cluster Management**. On the displayed page, locate the created CDM cluster and click **Job Management** in the **Operation** column.

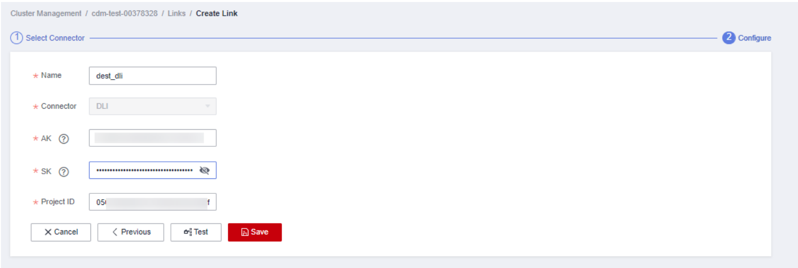
ii. On the **Job Management** page, click the **Links** tab, and click **Create Link**. On the displayed page, select **Data Lake Insight** and click **Next**.

Figure 5-20 Selecting the DLI connector



iii. Set the connection parameters. For details about parameter settings, see **Link to DLI**.

Figure 5-21 Setting connection parameters



iv. Click **Save**.

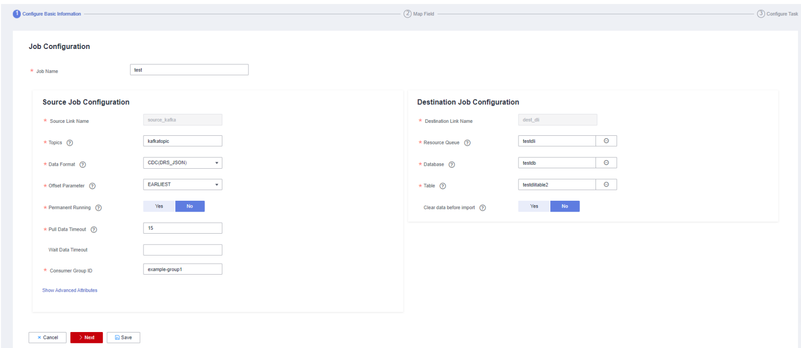
2. Create a CDM migration job.

   a. Log in to the CDM console. In the navigation pane on the left, choose **Cluster Management**. On the displayed page, locate the created CDM cluster and click **Job Management** in the **Operation** column.

   b. On the **Job Management** page, click the **Table/File Migration** tab. On the displayed tab, click **Create Job**.

   c. On the **Create Job** page, set job parameters.

Figure 5-22 Setting CDM job parameters



i. **Job Name**: Name of the data migration job, for example, **test**.

ii. Set the parameters in the **Source Job Configuration** area as follows:

Table 5-9 Source job parameters

Parameter	Value
Source Link Name	Select the name of the data source created in **1.a**.
Topics	Name of the topics you want to migrate to DLI. You can select one or more topics. Example: **kafkatopic**.
Data Format	Select the message format as needed. In this example, **CDC (DRS_JSON)** is selected, indicating that the source data will be parsed in DRS_JSON format.
Offset Parameter	Initial offset when data is pulled from Kafka. In this example, select **EARLIEST**. The options are: • **Latest**: Maximum offset, meaning that the latest data will be pulled. • **Earliest**: Minimum offset, meaning that the earliest data will be pulled. • **Submitted**: The submitted data is pulled. • **Time Range**: Data within a time range is pulled.
Permanent Running	Whether a job runs permanently. In this example, set it to **No**.
Pull Data Timeout	Maximum minutes allowed for a continuous data pulling. In this example, set it to **15**.
Wait Data Timeout	(Optional) Maximum seconds allowed for waiting data reading. In this example, leave this parameter blank.
Consumer Group ID	Consumer group ID. The default Kafka message group ID **example-group1** is used.

For details about parameter settings, see **From Apache Kafka**.

iii. Set the parameters in the **Destination Job Configuration** area as follows:

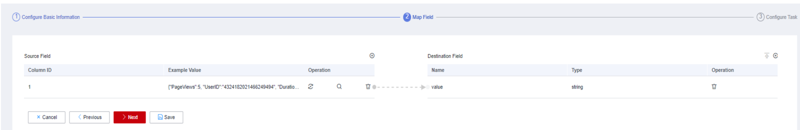
Table 5-10 Destination job parameters

Parameter	Value
Destination Link Name	Select the DLI data source connection created in **1.b**.
Resource Queue	Select a created DLI SQL queue.
Database Name	Select a created DLI database. In this example, database **testdb** created in **Create a database and table on DLI** is selected.
Table Name	Select the name of a table in the database. In this example, table **testdlitable** created in **Create a database and table on DLI** is selected.
Clear data before import	Whether to clear data in the destination table before data import. In this example, set it to **No**. If set to **Yes**, data in the destination table will be cleared before the task is started.

For details about parameter settings, see **To DLI**.

3. Click **Next**. The **Map Field** page is displayed. CDM automatically matches the source and destination fields.
   – You can drag any unmatched fields to match them.
   – If the type is automatically created at the migration destination, you need to configure the type and name of each field.
   – CDM allows for field conversion during migration. For details, see **Field Conversion**.

Figure 5-23 Field mapping



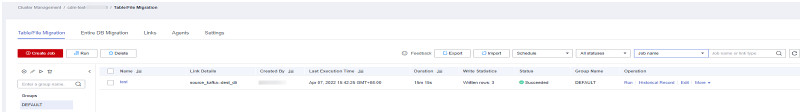
4. Click **Next** and set task parameters. Typically, retain the default values for all parameters.

   In this step, you can configure the following optional features:
   – **Retry Upon Failure**: If the job fails to be executed, you can determine whether to automatically retry. Retain the default value **Never**.
   – **Group**: Select the group to which the job belongs. The default group is **DEFAULT**. On the **Job Management** page, jobs can be displayed, started, or exported by group.
   – **Scheduled Execution**: For how to configure scheduled execution, see **Scheduling Job Execution**. Retain the default value **No**.
   – **Concurrent Extractors**: Enter the number of extractors to be concurrently executed. Retain the default value **1**.

   - **Write Dirty Data**: Set this parameter if data that fails to be processed or filtered out during job execution needs to be written to OBS. Before writing dirty data, create an OBS link. You can view the data on OBS later. Retain the default value **No**, meaning dirty data is not recorded.

5. Click **Save and Run**. On the **Job Management** page, you can view the job execution progress and result.

Figure 5-24 Job progress and execution result



Step 3: Query Results

Once the migration job is complete, check whether the Kafka table data has been migrated to the **testdlitable** table. Specifically, do as follows: Log in to the DLI management console and choose **SQL Editor**. On the displayed page, set **Engine** to **Spark**, **Queues** to the created SQL queue, and **Databases** to the database created in **a**. Then, execute the following query statement:

```
select * from testdlitable;
```

5.2.5 Example Typical Scenario: Migrating Data from Elasticsearch to DLI

This section describes how to use CDM's data synchronization to migrate data from a CSS Elasticsearch cluster to DLI. Data in a self-built Elasticsearch cluster can also be bidirectionally synchronized with DLI using CDM.

Prerequisites

- You have created a DLI SQL queue. For how to create a DLI queue, see **Creating a Queue**.

  CAUTION

  Set **Type** to **For SQL** when buying a queue.

- You have created a CSS Elasticsearch cluster. For how to create a CSS cluster, see **Creating a CSS Cluster**.

  In this example, the version of the created CSS cluster is 7.6.2, and the cluster is a non-security cluster.

- You have created a CDM cluster. For how to create a CDM cluster, see **Creating a CDM Cluster**.

NOTE

- To connect the cluster to an on-premises database as the destination data source, you can use either Internet or Direct Connect. If the Internet is used, make sure that an EIP has been bound to the CDM cluster, the security group of CDM allows outbound traffic from the host where the on-premises data source is located, the host where the data source is located can access the Internet, and the connection port has been enabled in the firewall rules.
- To successfully connect to the cloud service CSS as a data source, the following requirements must be met:

  i. If the CDM cluster and the cloud service are in different regions, they must be connected through either the Internet or Direct Connect. If the Internet is used, make sure that an EIP has been bound to the CDM cluster, the host where the data source is located can access the Internet, and the port has been enabled in the firewall rules.

  ii. If the CDM cluster and the cloud service are in the same region, instances in the same VPC, subnet, and security group can communicate with each other by default. If the CDM cluster and the cloud service are in the same VPC but in different subnets or security groups, you must configure routing and security group rules.

  For how to configure routing rules, see **Configure routes**. For how to configure security group rules, see **Security Group Configuration Examples**.

  iii. The cloud service instance and the CDM cluster belong to the same enterprise project. If they do not, change the enterprise project of the workspace.

In this example, the VPC, subnet, and security group of the CDM cluster match those of the CSS cluster.

Step 1: Prepare Data

- Create an index for the CSS cluster and import data.

  a. Log in to the CSS management console and choose **Clusters** > **Elasticsearch** from the navigation pane on the left.

  b. On the **Clusters** page, locate the created CSS cluster and click **Access Kibana** in the **Operation** column.

  c. In the navigation pane of Kibana, choose **Dev Tools**.

  d. On the displayed **Console** page, run the following command to create index **my_test**:

```
PUT /my_test
{
  "settings": {
    "number_of_shards": 1
  },
  "mappings": {
          "properties": {
        "productName": {
          "type": "text",
          "analyzer": "ik_smart"
        },
        "size": {
          "type": "keyword"
        }
      }
    }
  }
```

  e. Import data to the **my_test** index.

```
POST /my_test/_doc/_bulk
{"index":{}}
{"productName":"2017 Autumn New Shirts for Women", "size":"L"}
```

```
{"index":{}}
{"productName":"2017 Autumn New Shirts for Women", "size":"M"}
{"index":{}}
{"productName":"2017 Autumn New Shirts for Women", "size":"S"}
{"index":{}}
{"productName":"2018 Spring New Jeans for Women","size":"M"}
{"index":{}}
{"productName":"2018 Spring New Jeans for Women","size":"S"}
{"index":{}}
{"productName":"2017 Spring Casual Pants for Women","size":"L"}
{"index":{}}
{"productName":"2017 Spring Casual Pants for Women","size":"S"}
```

If **errors** is **false** in the command output, the data is imported.

- Create a database and table on DLI.
  a. Log in to the DLI management console. In the navigation pane on the left, choose **SQL Editor**. On the displayed page, set **Engine** to **Spark** and **Queues** to the created SQL queue.

     Create a database, for example, **testdb**. For the syntax to create a DLI database, see **Creating a Database**.

     ```
     create database testdb;
     ```

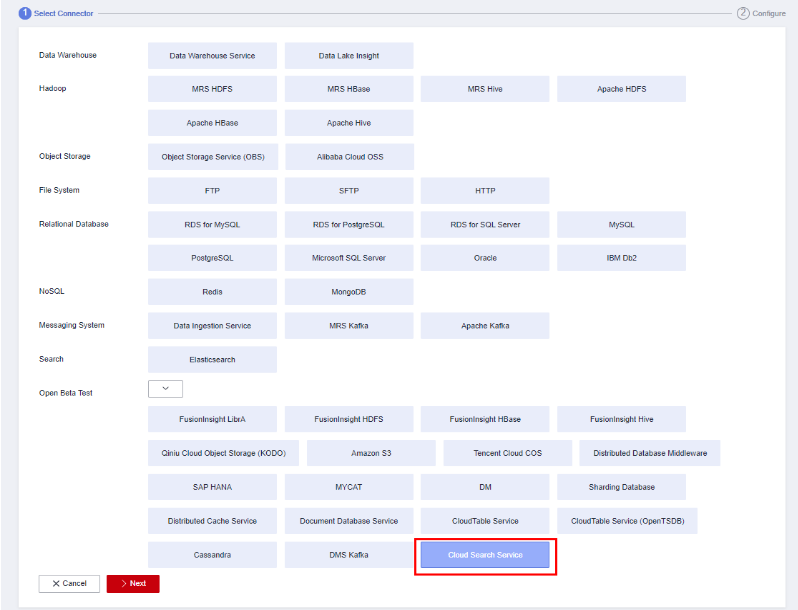
  b. Create a table in the database. For the table creation syntax, see **Creating a DLI Table Using the DataSource Syntax**.

     ```
     create table tablecss(size string, productname string);
     ```

Step 2: Migrate Data

1. Create a CDM connection.
   a. Create a connection to link CDM to the data source CSS.
      i. Log in to the CDM console. In the navigation pane on the left, choose **Cluster Management**. On the displayed page, locate the created CDM cluster and click **Job Management** in the **Operation** column.
      ii. On the **Job Management** page, click the **Links** tab, and click **Create Link**. On the displayed page, select **Cloud Search Service** and click **Next**.

Figure 5-25 Selecting the CSS connector

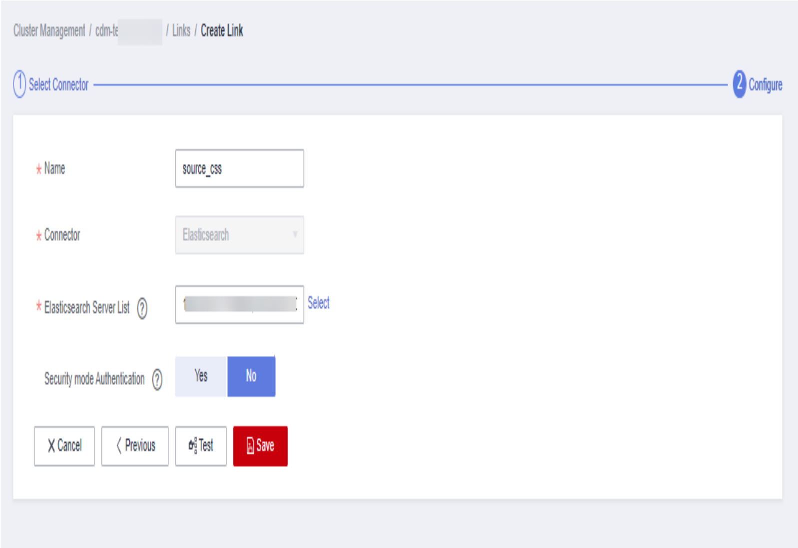


iii. Configure the connection. For details about parameter settings, see **Link to Elasticsearch/CSS**.

Table 5-11 CSS data source configuration

Parameter	Value
Name	Name of the CSS data source, for example, **source_css**.
Elasticsearch Server List	Elasticsearch server list, which is automatically displayed after you click **Select** next to the text box and select the CSS cluster.
Security Mode Authentication	If you have enabled the security mode for the CSS cluster, set this parameter to **Yes**. Otherwise, set this parameter to **No**. In this example, set it to **No**.

Figure 5-26 Configuring the CSS connection



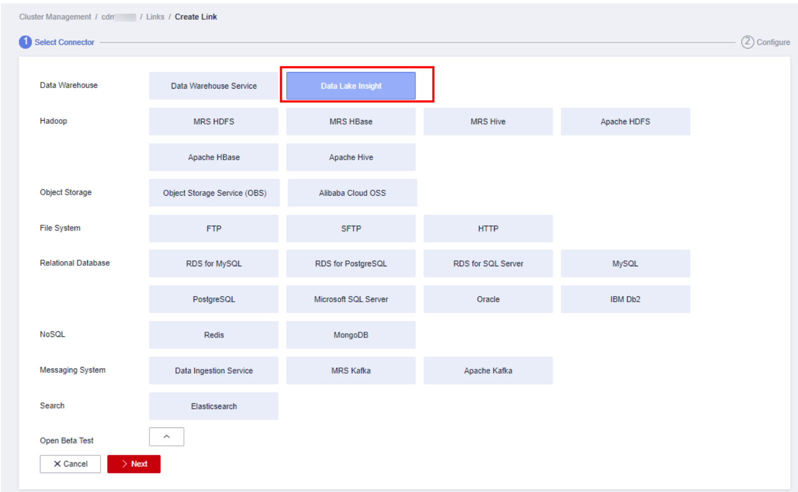
iv. Click **Save**.

b. Create a connection to link CDM to DLI.

i. Log in to the CDM console. In the navigation pane on the left, choose **Cluster Management**. On the displayed page, locate the created CDM cluster and click **Job Management** in the **Operation** column.

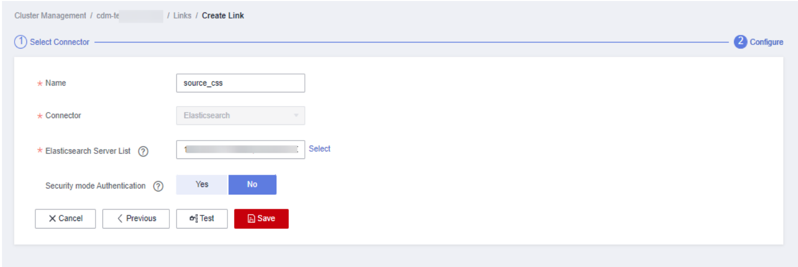
ii. On the **Job Management** page, click the **Links** tab, and click **Create Link**. On the displayed page, select **Data Lake Insight** and click **Next**.

Figure 5-27 Selecting the DLI connector



iii. Set the connection parameters. For details about parameter settings, see **Link to DLI**.

Figure 5-28 Setting connection parameters



iv. Click **Save**.

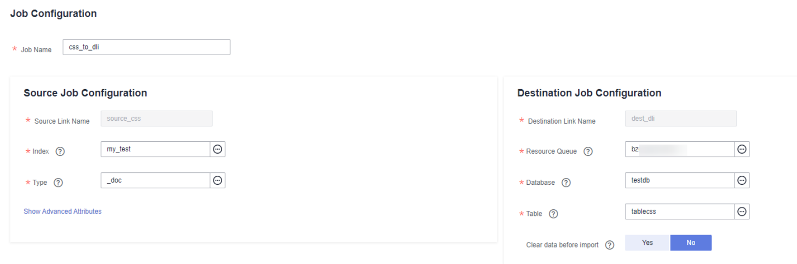
2. Create a CDM migration job.

   a. Log in to the CDM console. In the navigation pane on the left, choose **Cluster Management**. On the displayed page, locate the created CDM cluster and click **Job Management** in the **Operation** column.

   b. On the **Job Management** page, click the **Table/File Migration** tab. On the displayed tab, click **Create Job**.

   c. On the **Create Job** page, set job parameters.

Figure 5-29 Setting CDM job parameters



i. **Job Name**: Name of the data migration job, for example, **css_to_dli**.

ii. Set the parameters in the **Source Job Configuration** area as follows:

Table 5-12 Source job parameters

Parameter	Value
Source Link Name	Select the name of the data source created in **1.a**.
Index	Select the Elasticsearch index created for the CSS cluster. In this example, the **my_test** index created in **Create an index for the CSS cluster and import data** is used. Only lowercase letters are allowed.
Type	Elasticsearch type, which is similar to the table name of a relational database. Only lowercase letters are allowed. Example: **_doc**.

For details about other parameters, see **From Elasticsearch or CSS**.

iii. Set the parameters in the **Destination Job Configuration** area as follows:

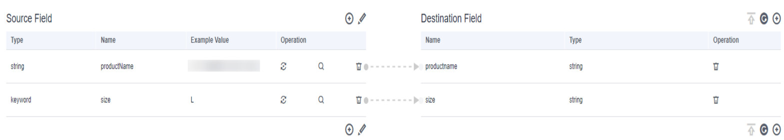
Table 5-13 Destination job parameters

Parameter	Value
Destination Link Name	Select the DLI data source connection created in **1.b**.
Resource Queue	Select a created DLI SQL queue.
Database Name	Select a created DLI database. In this example, the database **testdb** created in **Create a database and table on DLI** is selected.
Table Name	Select the name of a table in the database. In this example, the table **tablecss** created in **Create a database and table on DLI** is used.
Clear data before import	Whether to clear data in the destination table before data import. In this example, set it to **No**. If set to **Yes**, data in the destination table will be cleared before the task is started.

For details about parameter settings, see **To DLI**.

3. Click **Next**. The **Map Field** page is displayed. CDM automatically matches the source and destination fields.
   - You can drag any unmatched fields to match them.
   - If the type is automatically created at the migration destination, you need to configure the type and name of each field.
   - CDM allows for field conversion during migration. For details, see **Field Conversion**.

Figure 5-30 Field mapping



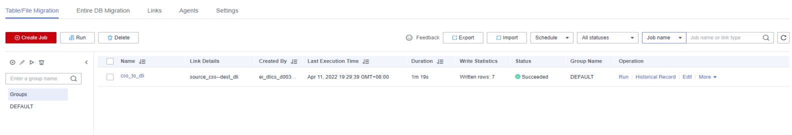
4. Click **Next** and set task parameters. Typically, retain the default values for all parameters.

   In this step, you can configure the following optional features:
   - **Retry Upon Failure**: If the job fails to be executed, you can determine whether to automatically retry. Retain the default value **Never**.
   - **Group**: Select the group to which the job belongs. The default group is **DEFAULT**. On the **Job Management** page, jobs can be displayed, started, or exported by group.

– **Scheduled Execution**: For how to configure scheduled execution, see **Scheduling Job Execution**. Retain the default value **No**.
– **Concurrent Extractors**: Enter the number of extractors to be concurrently executed. Retain the default value **1**.
– **Write Dirty Data**: Set this parameter if data that fails to be processed or filtered out during job execution needs to be written to OBS. Before writing dirty data, create an OBS link. You can view the data on OBS later. Retain the default value **No**, meaning dirty data is not recorded.

5. Click **Save and Run**. On the **Job Management** page, you can view the job execution progress and result.

Figure 5-31 Job progress and execution result

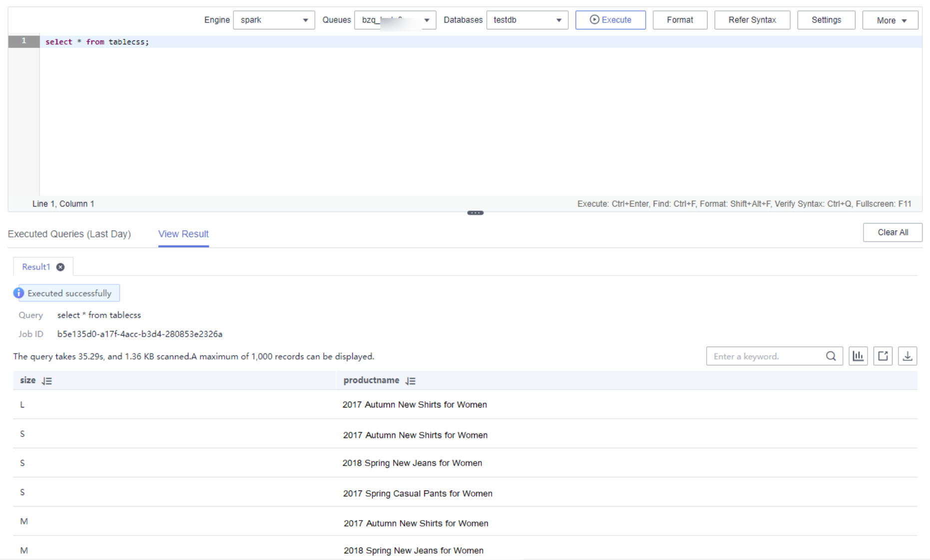


Step 3: Query Results

Once the migration job is complete, check whether the CSS table data has been migrated to the **tablecss** table. Specifically, do as follows: Log in to the DLI management console and choose **SQL Editor**. On the displayed page, set **Engine** to **Spark**, **Queues** to the created SQL queue, and **Databases** to the database created in **a**. Then, execute the following query statement:

```
select * from tablecss;
```

Figure 5-32 Querying migrated data



5.2.6 Example Typical Scenario: Migrating Data from RDS to DLI

This section describes how to use CDM's data synchronization to migrate data from an RDS DB instance to DLI. Data in other relational databases can also be bidirectionally synchronized with DLI using CDM.

Prerequisites

- You have created a DLI SQL queue. For how to create a DLI queue, see **Creating a Queue**.

> **CAUTION**
>
> Set **Type** to **For SQL** when buying a queue.

- You have created an RDS for MySQL DB instance. For how to create an RDS cluster, see **Buying an RDS for MySQL DB Instance**.
  - In this example, the RDS DB engine is MySQL.
  - In this example, the DB engine version is 5.7.
- You have created a CDM cluster. For how to create a CDM cluster, see **Creating a CDM Cluster**.

> **NOTE**
>
> - To connect the cluster to an on-premises database as the destination data source, you can use either Internet or Direct Connect. If the Internet is used, make sure that an EIP has been bound to the CDM cluster, the security group of CDM allows outbound traffic from the host where the on-premises data source is located, the host where the data source is located can access the Internet, and the connection port has been enabled in the firewall rules.
> - If the data source is RDS or MRS, the network must meet the following requirements:
>
>   i. If the CDM cluster and the cloud service are in different regions, they must be connected through either the Internet or Direct Connect. If the Internet is used, make sure that an EIP has been bound to the CDM cluster, the host where the data source is located can access the Internet, and the port has been enabled in the firewall rules.
>
>   ii. If the CDM cluster and the cloud service are in the same region, instances in the same VPC, subnet, and security group can communicate with each other by default. If the CDM cluster and the cloud service are in the same VPC but in different subnets or security groups, you must configure routing and security group rules.
>
>   For how to configure routing rules, see **Configure routes**. For how to configure security group rules, see **Security Group Configuration Examples**.
>
>   iii. The cloud service instance and the CDM cluster belong to the same enterprise project. If they do not, change the enterprise project of the workspace.

In this example, the VPC, subnet, and security group of the CDM cluster match those of the RDS for MySQL DB instance.

Step 1: Prepare Data

- Create a database and table on the RDS for MySQL DB instance.
  a. Log in to the RDS management console. In the navigation pane on the left, choose **Instances**. On the displayed page, locate the target DB instance and click **Log In** in the **Operation** column.
  b. In the displayed login dialog box, enter the correct username and password and click **Log In**.

c. On the **Databases** page, click **Create Database**. In the displayed dialog box, enter **testrdsdb** as the database name and retain default values for the rest parameters. Then, click **OK**.

d. In the **Operation** column of row where the created database is, click **SQL Window** and enter the following statement to create a table:

```
CREATE TABLE tabletest (
   `id` VARCHAR(32) NOT NULL,
   `name` VARCHAR(32) NOT NULL,
   PRIMARY KEY (`id`)
)    ENGINE = InnoDB
   DEFAULT CHARACTER SET = utf8mb4;
```

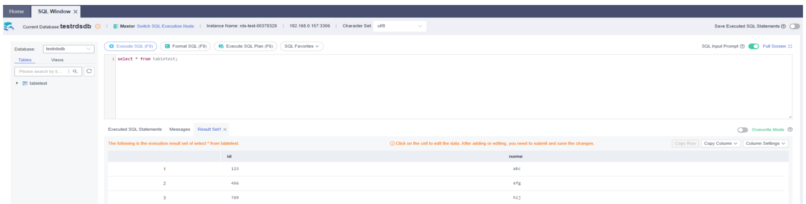
e. Insert data into the table.

```
insert into tabletest VALUES ('123','abc');
insert into tabletest VALUES ('456','efg');
insert into tabletest VALUES ('789','hij');
```

f. Query table data.

```
select * from tabletest;
```

Figure 5-33 Querying table data



- Create a database and table on DLI.

a. Log in to the DLI management console. In the navigation pane on the left, choose **SQL Editor**. On the displayed page, set **Engine** to **Spark** and **Queues** to the created SQL queue.

Create a database, for example, **testdb**. For the syntax to create a DLI database, see **Creating a Database**.

```
create database testdb;
```

b. On the **SQL Editor** page, set **Databases** to **testdb** and run the following table creation statement to create a table in the database. For the table creation syntax, see **Creating a DLI Table Using the DataSource Syntax**.

```
create table tabletest(id string,name string);
```

Step 2: Migrate Data

1. Create a CDM connection.

a. Create a connection to the RDS database.

i. Log in to the CDM console. In the navigation pane on the left, choose **Cluster Management**. On the displayed page, locate the created CDM cluster and click **Job Management** in the **Operation** column.

ii. If this is your first time creating a connection to RDS for MySQL, you need to upload the MySQL driver. Click the **Links** tab and click **Driver Management**.

iii. Download the MySQL driver to your local PC by referring to **Managing Drivers** and extract the driver package to obtain the JAR file.

For example, download the **mysql-connector-java-5.1.48.zip** package and extract it to obtain the driver file **mysql-connector-java-5.1.48.jar**.

iv. Return to the **Driver Management** page. Locate the **MYSQL** driver and click **Upload** in the **Operation** column. In the **Import Driver File** dialog box, click **Select File** and upload the driver file obtained in **1.a.iii**.

v. On the **Driver Management** page, click **Back** in the lower left corner to return to the **Links** tab. Click **Create Link**, select **RDS for MySQL**, and click **Next**.

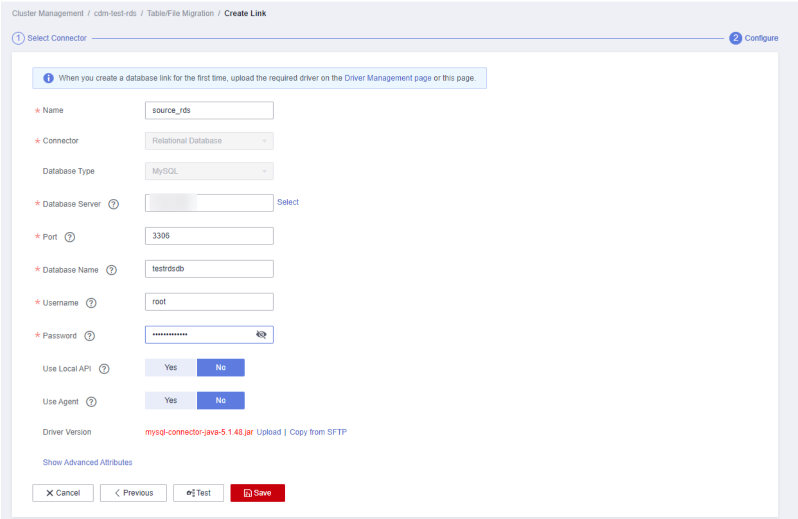
vi. Configure the connection as follows:

Table 5-14 Connection parameters

Parameter	Value
Name	Name of the RDS data source, for example, **source_rds**.
Database Server	Click **Select** next to the text box and click the name of the created RDS DB instance. The database server address is automatically entered.
Port	Port number of the RDS DB instance. The value is automatically filled in after you select the database server.
Database Name	Name of the RDS DB instance you want to migrate. In this example, the **testrdsdb** database created in **c** is used.
Username	Username used for accessing the database. This user must have the permissions to read and write data tables and metadata. In this example, the default user **root** for creating the RDS for MySQL DB instance is used.
Password	Password of the user.

For other parameters, retain the default values. For details, see **Link to Relational Databases**. Click **Save**.

Figure 5-34 Configuring the connection to the RDS for MySQL DB instance

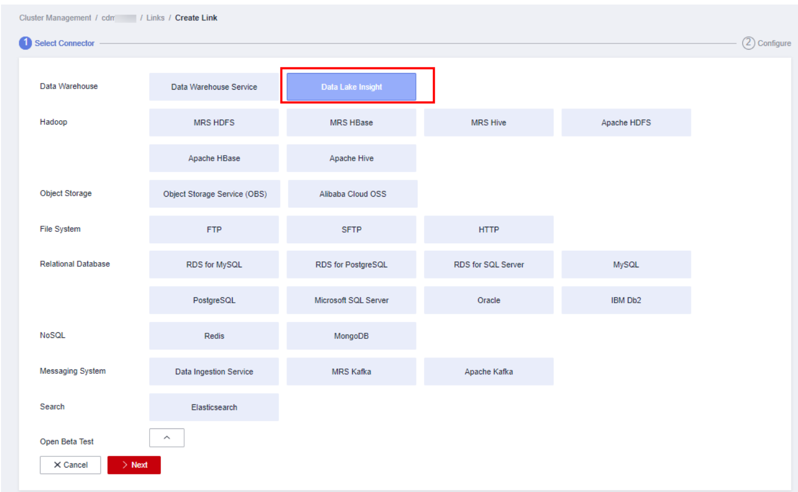


b. Create a connection to DLI.

i. Log in to the CDM console. In the navigation pane on the left, choose **Cluster Management**. On the displayed page, locate the created CDM cluster and click **Job Management** in the **Operation** column.

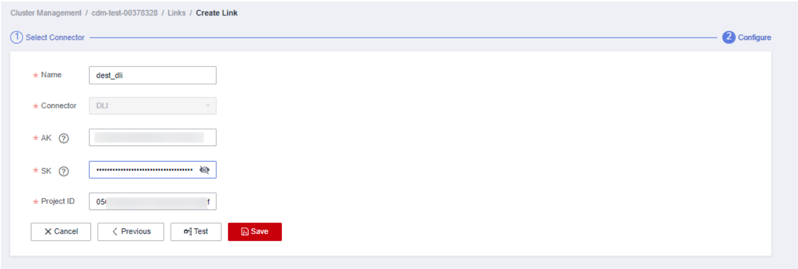
ii. On the **Job Management** page, click the **Links** tab, and click **Create Link**. On the displayed page, select **Data Lake Insight** and click **Next**.

Figure 5-35 Selecting the DLI connector



i. Create a connection to link CDM to DLI. For details about parameter settings, see **Link to DLI**.

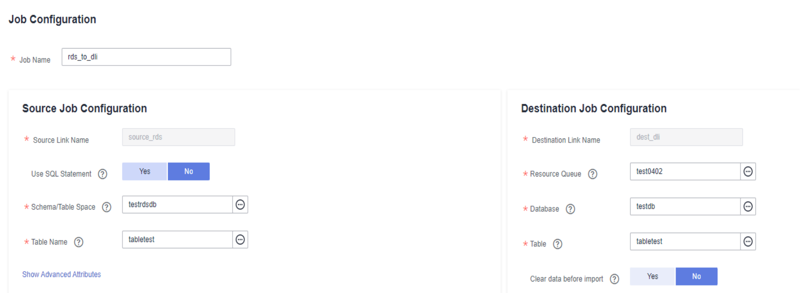
Figure 5-36 Selecting the DLI connector



Click **Save**.

2. Create a CDM migration job.
   a. Log in to the CDM console. In the navigation pane on the left, choose **Cluster Management**. On the displayed page, locate the created CDM cluster and click **Job Management** in the **Operation** column.
   b. On the **Job Management** page, click the **Table/File Migration** tab. On the displayed tab, click **Create Job**.
   c. On the **Create Job** page, set job parameters.

Figure 5-37 Configuring the migration job



i. **Job Name**: Name of the data migration job, for example, **rds_to_dli**.

ii. Set the parameters in the **Source Job Configuration** area as follows:

Table 5-15 Source job parameters

Parameter	Value
Source Link Name	Select the name of the data source created in **1.a**.
Use SQL Statement	When set to **Yes**, enter a SQL statement. CDM exports data based on the statement. In this example, set it to **No**.
Schema/Table Space	Select the name of the RDS for MySQL database you want to migrate to DLI. For example, the **testrdsdb** database.
Table Name	Name of the table you want to migrate. In this example, use **tabletest** created in **d**.

For details about parameter settings, see **From PostgreSQL/SQL Server**.

iii. Set the parameters in the **Destination Job Configuration** area as follows:

Table 5-16 Destination job parameters

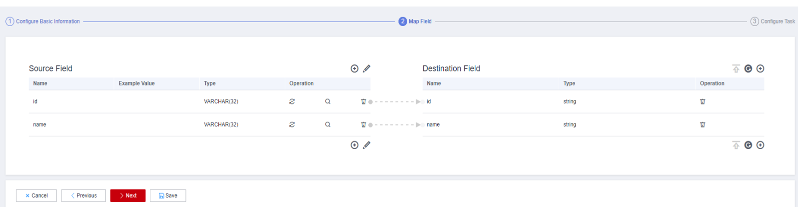
Parameter	Value
Destination Link Name	Select the DLI data source connection.
Resource Queue	Select a created DLI SQL queue.
Database Name	Select a created DLI database. In this example, the database **testdb** created in **Create a database and table on DLI** is selected.
Table Name	Select the name of a table in the database. In this example, the table **tabletest** created in **Create a database and table on DLI** is selected.
Clear data before import	Whether to clear data in the destination table before data import. In this example, set it to **No**. If set to **Yes**, data in the destination table will be cleared before the task is started.

For details about parameter settings, see **To DLI**.

iv. Click **Next**. The **Map Field** page is displayed. CDM automatically matches the source and destination fields.

- You can drag any unmatched fields to match them.
- If the type is automatically created at the migration destination, you need to configure the type and name of each field.
- CDM allows for field conversion during migration. For details, see **Field Conversion**.

Figure 5-38 Field mapping



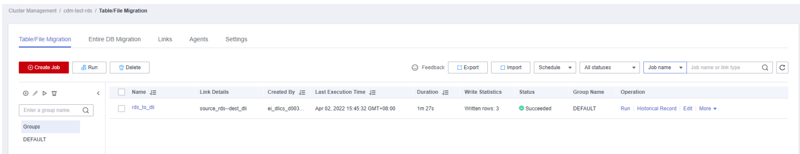
v. Click **Next** and set task parameters. Typically, retain the default values for all parameters.

In this step, you can configure the following optional features:

- **Retry Upon Failure**: If the job fails to be executed, you can determine whether to automatically retry. Retain the default value **Never**.
- **Group**: Select the group to which the job belongs. The default group is **DEFAULT**. On the **Job Management** page, jobs can be displayed, started, or exported by group.
- **Scheduled Execution**: For how to configure scheduled execution, see **Scheduling Job Execution**. Retain the default value **No**.
- **Concurrent Extractors**: Enter the number of extractors to be concurrently executed. Retain the default value **1**.
- **Write Dirty Data**: Set this parameter if data that fails to be processed or filtered out during job execution needs to be written to OBS. Before writing dirty data, create an OBS link. You can view the data on OBS later. Retain the default value **No**, meaning dirty data is not recorded.

vi. Click **Save and Run**. On the **Job Management** page, you can view the job execution progress and result.

Figure 5-39 Job progress and execution result

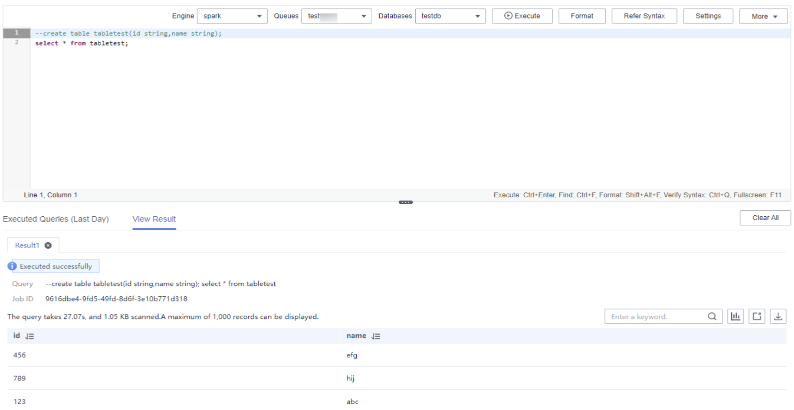


Step 3: Query Results

Once the migration job is complete, check whether the RDS for MySQL table data has been migrated to the **tabletest** table. Specifically, do as follows: Log in to the DLI management console and choose **SQL Editor**. On the displayed page, set **Engine** to **Spark**, **Queues** to the created SQL queue, and **Databases** to the database created in **Create a database and table on DLI**. Then, execute the following query statement:

```
select * from tabletest;
```

Figure 5-40 Querying data in the table



5.2.7 Example Typical Scenario: Migrating Data from GaussDB(DWS) to DLI

This section describes how to use CDM's data synchronization to migrate data from GaussDB(DWS) to DLI.

Prerequisites

- You have created a DLI SQL queue. For how to create a DLI queue, see **Creating a Queue**.

  CAUTION

  Set **Type** to **For SQL** when buying a queue.

- You have created a GaussDB(DWS) cluster. For how to create a GaussDB(DWS) cluster, see **Creating a Cluster**.
- You have created a CDM cluster. For how to create a CDM cluster, see **Creating a CDM Cluster**.

  NOTE

  - To connect the cluster to an on-premises database as the destination data source, you can use either Internet or Direct Connect. If the Internet is used, make sure that an EIP has been bound to the CDM cluster, the security group of CDM allows outbound traffic from the host where the on-premises data source is located, the host where the data source is located can access the Internet, and the connection port has been enabled in the firewall rules.
  - If the data source is GaussDB(DWS) or MRS, the network must meet the following requirements:

    i. If the CDM cluster and the cloud service are in different regions, they must be connected through either the Internet or Direct Connect. If the Internet is used, make sure that an EIP has been bound to the CDM cluster, the host where the data source is located can access the Internet, and the port has been enabled in the firewall rules.

    ii. If the CDM cluster and the cloud service are in the same region, instances in the same VPC, subnet, and security group can communicate with each other by default. If the CDM cluster and the cloud service are in the same VPC but in different subnets or security groups, you must configure routing and security group rules.

    For how to configure routing rules, see **Configure routes**. For how to configure security group rules, see **Security Group Configuration Examples**.

    iii. The cloud service instance and the CDM cluster belong to the same enterprise project. If they do not, change the enterprise project of the workspace.

  In this example, the VPC, subnet, and security group of the CDM cluster match those of the GaussDB(DWS) cluster.

Step 1: Prepare Data

- Create a database and table in the GaussDB(DWS) cluster.
  a. Connect to the existing GaussDB(DWS) cluster by referring to **Using the gsql CLI Client to Connect to a Cluster**.

b. Connect to the default database **gaussdb** of the GaussDB(DWS) cluster.

```
gsql -d gaussdb -h Connection address of the GaussDB(DWS) cluster -U dbadmin -p 8000 -W password -r
```

- **gaussdb**: Default database of the GaussDB(DWS) cluster.
- **Connection address of the DWS cluster**: If a public address is used, set it to **Public Network Address** or **Public Network Access Domain Name**. If a private address is used, set it to **Private Network Address** or **Private Network Access Domain Name**. For details, see **Obtaining the Cluster Connection Address**. If an ELB is used, set it to the ELB address.
- **dbadmin**: Default administrator username used during cluster creation.
- **-W**: Default password of the administrator.

c. Create the **testdwsdb** database.

```
CREATE DATABASE testdwsdb;
```

d. Exit the **gaussdb** database and connect to **testdwsdb**.

```
\q
gsql -d testdwsdb -h Connection address of the GaussDB(DWS) cluster -U dbadmin -p 8000 -W password -r
```

e. Create a table and import data into it.

Create a table.

```
CREATE TABLE table1(id int, a char(6), b varchar(6),c varchar(6)) ;
```

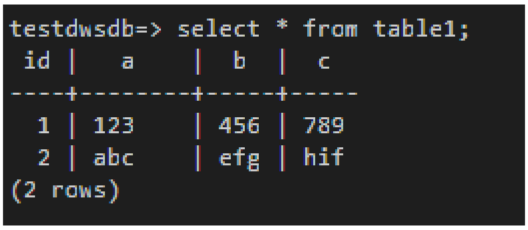
Insert data into the table.

```
INSERT INTO table1 VALUES(1,'123','456','789');
INSERT INTO table1 VALUES(2,'abc','efg','hif');
```

f. Query the table data to verify that the data is inserted.

```
select * from table1;
```

Figure 5-41 Querying data in the table



- Create a database and table on DLI.

a. Log in to the DLI management console. In the navigation pane on the left, choose **SQL Editor**. On the displayed page, set **Engine** to **Spark** and **Queues** to the created SQL queue.

Create a database, for example, **testdb**. For the syntax to create a DLI database, see **Creating a Database**.

```
create database testdb;
```

b. On the **SQL Editor** page, set **Databases** to **testdb** and run the following table creation statement to create a table in the database. For the table creation syntax, see **Creating a DLI Table Using the DataSource Syntax**.

```
create table tabletest(id INT, name1 string, name2 string, name3 string);
```

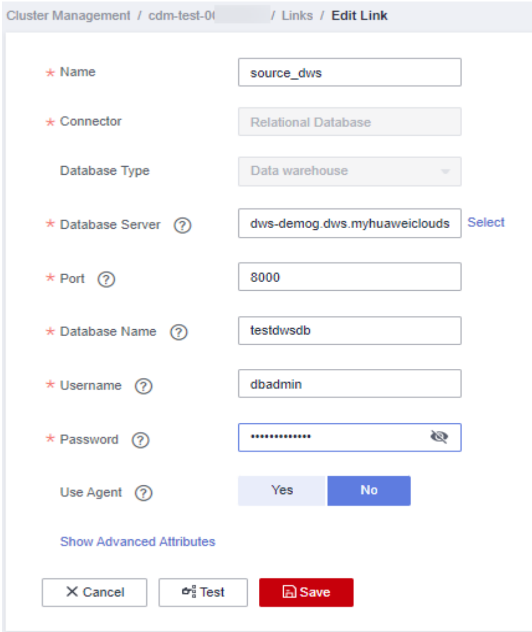
Step 2: Migrate Data

1. Create a CDM connection.
   a. Create a connection to the GaussDB(DWS) database.
      i. Log in to the CDM console. In the navigation pane on the left, choose **Cluster Management**. On the displayed page, locate the created CDM cluster and click **Job Management** in the **Operation** column.
      ii. On the **Job Management** page, click the **Links** tab, and click **Create Link**. On the displayed page, select **Data Warehouse Service** and click **Next**.
      iii. Configure the connection as follows:

Table 5-17 GaussDB(DWS) data source configuration

Parameter	Value
Name	Name of the GaussDB(DWS) data source, for example, **source_dws**.
Database Server	Click **Select** next to the text box and select the name of the created GaussDB(DWS) cluster.
Port	Port number of the GaussDB(DWS) database, which is **8000** by default.
Database Name	Name of the GaussDB(DWS) database you want to migrate. In this example, the **testdwsdb** database created in **Create a database and table in the GaussDB(DWS) cluster** is used.
Username	Username used for accessing the database. This user must have the permissions to read and write data tables and metadata. In this example, the default administrator **dbadmin** specified when you create the GaussDB(DWS) database is used.
Password	Password of the GaussDB(DWS) database user.

Figure 5-42 Configuring the GaussDB(DWS) connection



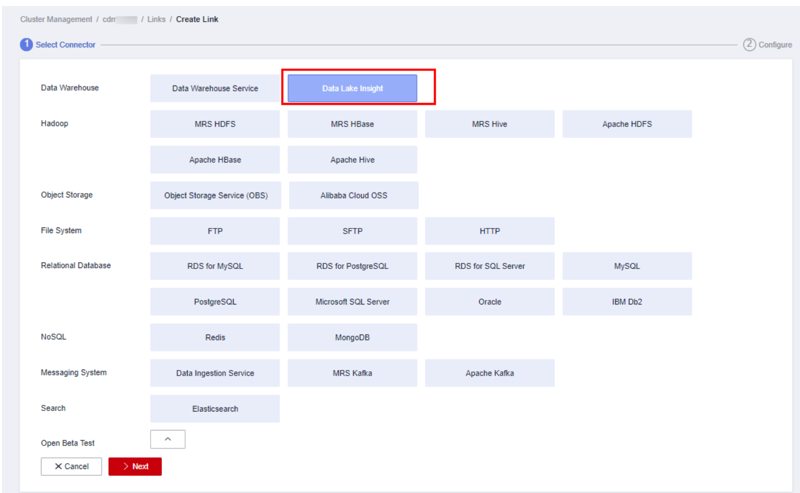
For other parameters, retain the default values. For details, see **Link to Relational Databases**. Click **Save**.

b. Create a connection to the DLI.

i. Log in to the CDM console. In the navigation pane on the left, choose **Cluster Management**. On the displayed page, locate the created CDM cluster and click **Job Management** in the **Operation** column.

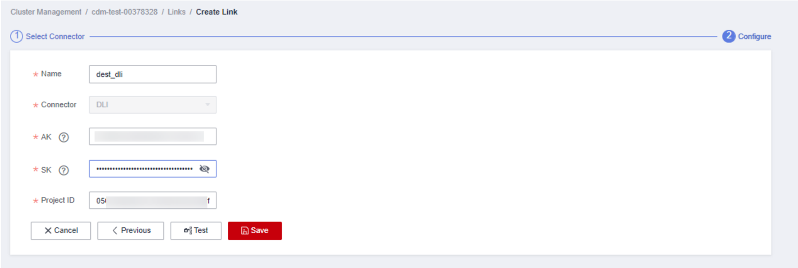
ii. On the **Job Management** page, click the **Links** tab, and click **Create Link**. On the displayed page, select **Data Lake Insight** and click **Next**.

Figure 5-43 Selecting the DLI connector



i. Create a connection to link CDM to DLI. For details about parameter settings, see **Link to DLI**.

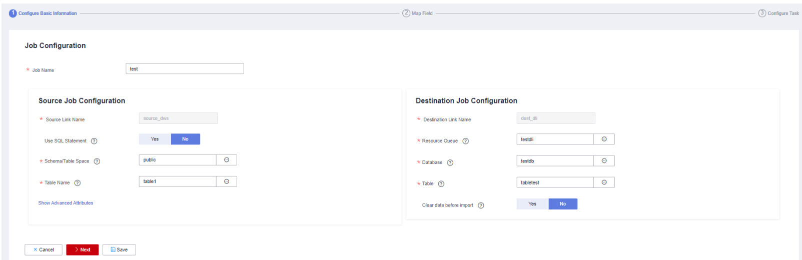
Figure 5-44 Selecting the DLI connector



Click **Save**.

2. Create a CDM migration job.
   a. Log in to the CDM console. In the navigation pane on the left, choose **Cluster Management**. On the displayed page, locate the created CDM cluster and click **Job Management** in the **Operation** column.
   b. On the **Job Management** page, click the **Table/File Migration** tab. On the displayed tab, click **Create Job**.
   c. On the **Create Job** page, set job parameters.

Figure 5-45 Configuring the migration job



i. **Job Name**: Name of the data migration job, for example, **test**.

ii. Set the parameters in the **Source Job Configuration** area as follows:

Table 5-18 Source job parameters

Parameter	Value
Source Link Name	Select the name of the data source created in **1.a**.
Use SQL Statement	When set to **Yes**, enter a SQL statement. CDM exports data based on the statement. In this example, set it to **No**.

Parameter	Value
Schema/Table Space	Name of the schema or tablespace from which data will be extracted. This parameter is available when **Use SQL Statement** is set to **No**. Click the icon next to the text box to select a schema or tablespace or directly enter a schema or tablespace. In this example, set this parameter to the default value **public** as there is no schema created in **Create a database and table in the GaussDB(DWS) cluster**. If there are no schemas or tablespaces available, check if the account has the permission to query metadata. **NOTE** The parameter value can contain wildcard characters (*), which allows for the export of all databases with names starting or ending with a certain prefix or suffix, respectively. For example: **SCHEMA*** indicates that all databases with names starting with **SCHEMA** are exported. ***SCHEMA** indicates that all databases with names ending with **SCHEMA** are exported. ***SCHEMA*** indicates that all databases with names containing **SCHEMA** are exported.
Table Name	Name of the table you want to migrate. In this example, **table1** created in **Create a database and table in the GaussDB(DWS) cluster** is used.

For details about parameter settings, see **From a Relational Database**.

iii. Set the parameters in the **Destination Job Configuration** area as follows:

Table 5-19 Destination job parameters

Parameter	Value
Destination Link Name	Select the DLI data source connection.
Resource Queue	Select a created DLI SQL queue.
Database Name	Select a created DLI database. In this example, the database **testdb** created in **Create a database and table on DLI** is used.

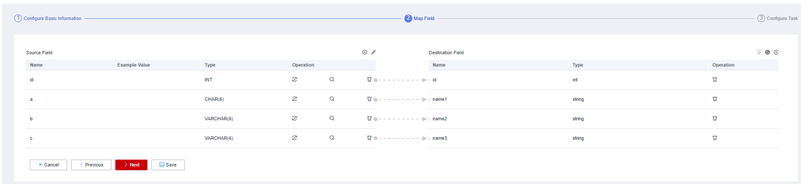
Parameter	Value
Table Name	Select the name of a table in the database. In this example, the table **tabletest** created in **Create a database and table on DLI** is used.
Clear data before import	Whether to clear data in the destination table before data import. In this example, set it to **No**. If set to **Yes**, data in the destination table will be cleared before the task is started.

For details about parameter settings, see **To DLI**.

iv. Click **Next**. The **Map Field** page is displayed. CDM automatically matches the source and destination fields.

- You can drag any unmatched fields to match them.
- If the type is automatically created at the migration destination, you need to configure the type and name of each field.
- CDM allows for field conversion during migration. For details, see **Field Conversion**.

Figure 5-46 Field mapping



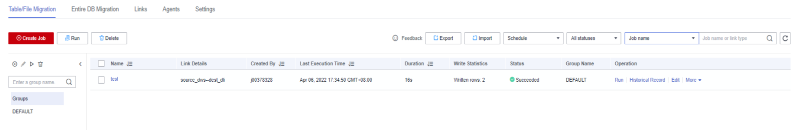
v. Click **Next** and set task parameters. Typically, retain the default values for all parameters.

In this step, you can configure the following optional features:

- **Retry Upon Failure**: If the job fails to be executed, you can determine whether to automatically retry. Retain the default value **Never**.
- **Group**: Select the group to which the job belongs. The default group is **DEFAULT**. On the **Job Management** page, jobs can be displayed, started, or exported by group.
- **Scheduled Execution**: For how to configure scheduled execution, see **Scheduling Job Execution**. Retain the default value **No**.
- **Concurrent Extractors**: Enter the number of extractors to be concurrently executed. Retain the default value **1**.
- **Write Dirty Data**: Set this parameter if data that fails to be processed or filtered out during job execution needs to be written to OBS. Before writing dirty data, create an OBS link. You can view the data on OBS later. Retain the default value **No**, meaning dirty data is not recorded.

vi. Click **Save and Run**. On the **Job Management** page, you can view the job execution progress and result.

Figure 5-47 Job progress and execution result

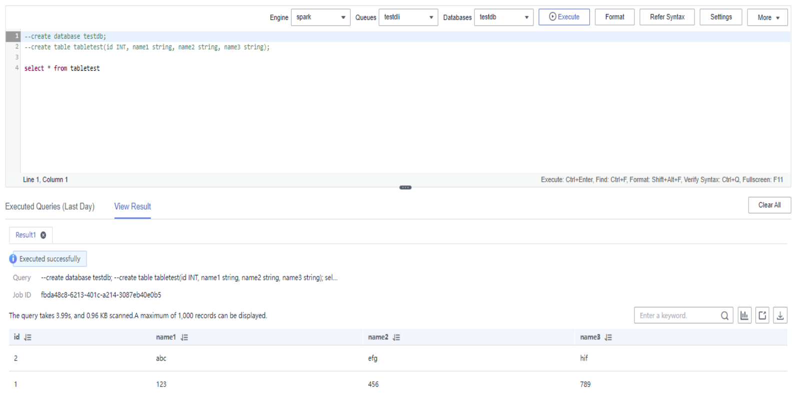


Step 3: Query Results

Once the migration job is complete, check whether the GaussDB(DWS) table data has been migrated to the **tabletest** table. Specifically, do as follows: Log in to the DLI management console and choose **SQL Editor**. On the displayed page, set **Engine** to **Spark**, **Queues** to the created SQL queue, and **Databases** to the database created in **Create a database and table on DLI**. Then, execute the following query statement:

```
select * from tabletest;
```

Figure 5-48 Querying data in the table



5.3 Configuring DLI to Read and Write Data from and to External Data Sources

5.3.1 Configuring DLI to Read and Write External Data Sources

To read and write external data sources when running DLI jobs, two conditions must be met:

- Establish network connectivity between DLI and the external data source to ensure that the DLI queue is connected to the data source network.
- Securely store the access credentials for the data source to ensure authentication security and facilitate secure DLI access to the data source.

This section describes how to configure DLI to read and write external data sources.

- Configure network connection between DLI and the data source by referring to **Configuring the Network Connection Between DLI and Data Sources (Enhanced Datasource Connection)**.
- Manage credentials for DLI to access data sources.
  - Spark 3.3.1 or later and Flink 1.15 or later jobs accessing data sources using datasource connections
    - You are advised to use Data Encryption Workshop (DEW) to store authentication information of data sources, addressing data security, key security, and complex key management issues.

      For details, see **Using DEW to Manage Access Credentials for Data Sources**.
    - To manage data source access credentials using DEW, you also need to create a DLI agency to grant DLI access to read access credentials for other services (DEW).
  - When SQL and Flink 1.12 jobs access data sources using datasource connections, use DLI's datasource authentication feature to manage data source access credentials. For details, see **Using DLI Datasource Authentication to Manage Access Credentials for Data Sources**.

5.3.2 Configuring the Network Connection Between DLI and Data Sources (Enhanced Datasource Connection)

5.3.2.1 Overview of Enhanced Datasource Connections

Why Create Enhanced Datasource Connections?

In cross-source data analysis scenarios, DLI needs to connect to external data sources. However, due to the different VPCs between the data source and DLI, the network cannot be connected, which results in DLI being unable to read data from the data source. DLI's enhanced datasource connection feature enables network connectivity between DLI and the data source.

This section will introduce a solution for cross-VPC data source network connectivity:

- Creating an enhanced datasource connection: Establish a VPC peering connection to connect DLI and the data source's VPC network.
- Testing network connectivity: Verify the connectivity between the queue and the data source's network.

For details about the data sources that support cross-source access, see **Common Development Methods for DLI Cross-Source Analysis**.

CAUTION

In cross-source development scenarios, there is a risk of password leakage if datasource authentication information is directly configured. You are advised to use Data Encryption Workshop (DEW) to store authentication information of data sources when Spark 3.3.1 or later and Flink 1.15 or later jobs access data sources using datasource connections. This will help you address issues related to data security, key security, and complex key management. For details, see **Using DEW to Manage Access Credentials for Data Sources**.

Notes and Constraints

- Datasource connections cannot be created for the **default** queue.
- Flink jobs can directly access DIS, OBS, and SMN data sources without using datasource connections.
- Enhanced connections can only be created for yearly/monthly and pay-per-use queues.
- **VPC Administrator** permissions are required for enhanced connections to use VPCs, subnets, routes, VPC peering connections.

  You can set these permissions by referring to **Service Authorization**.
- If you use an enhanced datasource connection, the CIDR block of the elastic resource pool or queue cannot overlap with that of the data source.
- Only queues bound with datasource connections can access datasource tables.
- Datasource tables do not support the preview function.
- When checking the connectivity of datasource connections, the notes and constraints on IP addresses are:
  - The IP address must be valid, which consists of four decimal numbers separated by periods (.). The value ranges from 0 to 255.
  - During the test, you can add a port after the IP address and separate them with colons (:). The port can contain a maximum of five digits. The value ranges from 0 to 65535.

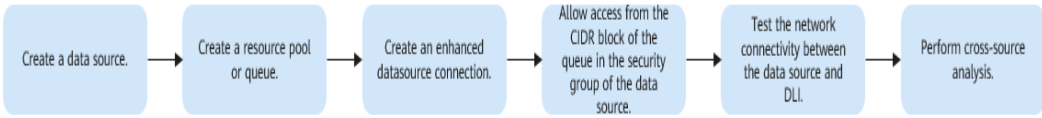
    For example, **192.168.*xx.xx*** or **192.168.*xx.xx*:8181**.
- When checking the connectivity of datasource connections, the notes and constraints on domain names are:
  - The domain name can contain 1 to 255 characters. Only letters, numbers, underscores (_), and hyphens (-) are allowed.
  - The top-level domain name must contain at least two letters, for example, **.com**, **.net**, and **.cn**.
  - During the test, you can add a port after the domain name and separate them with colons (:). The port can contain a maximum of five digits. The value ranges from 0 to 65535.

    For example, **example.com:8080**.

Cross-Source Analysis Process

To use DLI for cross-source analysis, you need to create a datasource connection to connect DLI to the data source, and then develop jobs to access the data source.

Figure 5-49 Cross-source analysis flowchart



Helpful Links

Creation Method	Enhanced Datasource Connection
Console	**Creating an Enhanced Datasource Connection**
API	**Creating an Enhanced Datasource Connection**

5.3.2.2 Creating an Enhanced Datasource Connection

Scenario

Create an enhanced datasource connection for DLI to access, import, query, and analyze data of other data sources.

For example, to connect DLI to the MRS, RDS, CSS, Kafka, or GaussDB(DWS) data source, you need to enable the network between DLI and the VPC of the data source.

Create an enhanced datasource connection on the console.

Notes and Constraints

- Datasource connections cannot be created for the **default** queue.
- Flink jobs can directly access DIS, OBS, and SMN data sources without using datasource connections.
- Enhanced connections can only be created for yearly/monthly and pay-per-use queues.
- **VPC Administrator** permissions are required for enhanced connections to use VPCs, subnets, routes, VPC peering connections.

  You can set these permissions by referring to **Service Authorization**.
- If you use an enhanced datasource connection, the CIDR block of the elastic resource pool or queue cannot overlap with that of the data source.
- Only queues bound with datasource connections can access datasource tables.
- Datasource tables do not support the preview function.
- When checking the connectivity of datasource connections, the notes and constraints on IP addresses are:
  - The IP address must be valid, which consists of four decimal numbers separated by periods (.). The value ranges from 0 to 255.

  - During the test, you can add a port after the IP address and separate them with colons (:). The port can contain a maximum of five digits. The value ranges from 0 to 65535.

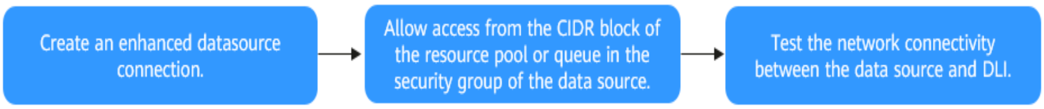
    For example, **192.168.***xx.xx* or **192.168.***xx.xx***:8181**.

- When checking the connectivity of datasource connections, the notes and constraints on domain names are:
  - The domain name can contain 1 to 255 characters. Only letters, numbers, underscores (_), and hyphens (-) are allowed.
  - The top-level domain name must contain at least two letters, for example, **.com**, **.net**, and **.cn**.
  - During the test, you can add a port after the domain name and separate them with colons (:). The port can contain a maximum of five digits. The value ranges from 0 to 65535.

    For example, **example.com:8080**.

Process

Figure 5-50 Enhanced datasource connection creation flowchart



Prerequisites

- An elastic resource pool or queue has been created.
- You have obtained the VPC, subnet, private IP address, port, and security group information of the external data source.
- The security group of the external data source has allowed access from the CIDR block of the elastic resource pool or queue.

Procedure

Step 1 **Create an Enhanced Datasource Connection**

1. Log in to the DLI management console.
2. In the navigation pane on the left, choose **Datasource Connections**.
3. On the displayed **Enhanced** tab, click **Create**.

   Set parameters based on **Table 5-20**.

Table 5-20 Parameters

Parameter	Description
Connection Name	Name of the created datasource connection. – Only letters, numbers, and underscores (_) are allowed. The parameter must be specified. – A maximum of 64 characters are allowed.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Resource Pool</td><td>It binds an elastic resource pool or queue that uses a datasource connection. This parameter is optional.
Only dedicated queues charged in yearly/monthly or pay-per-use billing mode can be bound to elastic resource pools.
In regions where this function is available, an elastic resource pool with the same name is created by default for the queue created in "Creating a Queue."
NOTE
Before using an enhanced datasource connection, you must bind a queue and ensure that the VPC peering connection is in the Active state.</td></tr>
<tr><td>Bind Queue</td><td>It binds a queue that requires datasource connections. This parameter is optional.
Only dedicated queues charged in yearly/monthly or pay-per-use billing mode can be bound to elastic resource pools.
NOTE
Before using an enhanced datasource connection, you must bind a queue and ensure that the VPC peering connection is in the Active state.</td></tr>
<tr><td>VPC</td><td>VPC used by the data source.</td></tr>
<tr><td>Subnet</td><td>Subnet used by the data source.</td></tr>
<tr><td>Route Table</td><td>Route table of the subnet.
NOTE
– The route table is associated with the subnet used by the destination data source, which is not the table containing the route you add by Manage Route in the Operation column. The route you add on the Manage Route page is contained in the route table associated with the subnet used by the queue to be bound.
– The subnet used by the destination data source must be different from that used by the queue to be bound. Otherwise, a segment conflict occurs.</td></tr>
</table>

Parameter	Description
Host Information	In this text field, you can configure the mapping between host IP addresses and domain names so that jobs can only use the configured domain names to access corresponding hosts. This parameter is optional. For example, when accessing the HBase cluster of MRS, you need to configure the host name (domain name) and IP address of the ZooKeeper instance. Enter one record in each line in the format of *IP address Host name*/*Domain name*. Example: 192.168.0.22 node-masterxxx1.com 192.168.0.23 node-masterxxx2.com For details about how to obtain host information, see **How Do I Obtain MRS Host Information?**.
Tags	Tags used to identify cloud resources. A tag includes the tag key and tag value. If you want to use the same tag to identify multiple cloud resources, that is, to select the same tag from the drop-down list box for all services, you are advised to create predefined tags on the Tag Management Service (TMS). If your organization has configured tag policies for DLI, add tags to resources based on the policies. If a tag does not comply with the tag policies, resource creation may fail. Contact your organization administrator to learn more about tag policies. For details, see **Tag Management Service User Guide**. **NOTE** – A maximum of 20 tags can be added. – Only one tag value can be added to a tag key. – The key name in each resource must be unique. – Tag key: Enter a tag key name in the text box. **NOTE** A tag key can contain a maximum of 128 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed, but the value cannot start or end with a space or start with _**sys**_. – Tag value: Enter a tag value in the text box. **NOTE** A tag value can contain a maximum of 255 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed. The value cannot start or end with a space.

4. Click **OK**.

   After the creation is complete, the enhanced datasource connection is in the **Active** state, indicating that the connection is successfully created.

Step 2 Security Group Where the Data Source Belongs Allows Access from the CIDR Block of the Elastic Resource Pool

1. On the DLI management console, obtain the network segment of the elastic resource pool or queue.

   Choose **Resources** > **Queue Management** from the left navigation pane. On the page displayed, locate the queue on which jobs are running, and click the button next to the queue name to obtain the CIDR block of the queue.

2. Log in to the VPC console and find the VPC the data source belongs to.
3. On the network console, choose **Virtual Private Cloud** > **Network Interfaces**. On the **Network Interfaces** tab page displayed, search for the security group name, click **More** in the **Operation** column, and select **Change Security Group**.
4. In the navigation pane on the left, choose **Access Control** > **Security Groups**.
5. Click the name of the security group to which the external data source belongs.
6. Click the **Inbound Rules** tab and add a rule to allow access from the CIDR block of the queue. See **Figure 5-51**.

   Configure the inbound rule parameters according to **Table 5-21**.

Figure 5-51 Adding an inbound rule

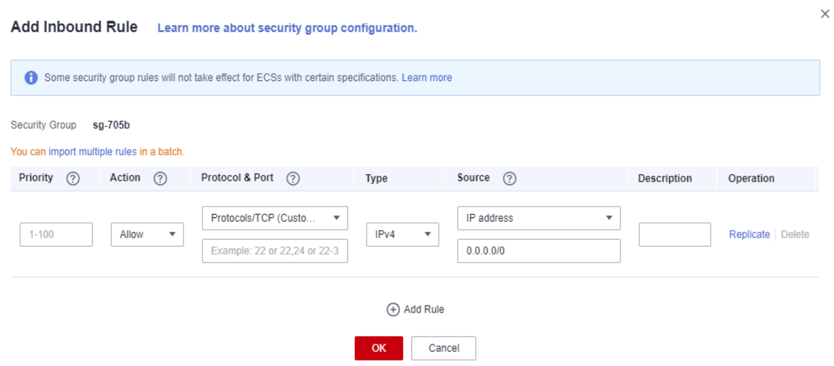


Table 5-21 Inbound rule parameters

Parameter	Description	Example Value
Priority	Priority of a security group rule. The priority value ranges from 1 to 100. The default value is **1**, indicating the highest priority. A smaller value indicates a higher priority of a security group rule.	1
Action	Action of the security group rule.	Allow

Parameter	Description	Example Value
Protocol & Port	– Network protocol. The value can be **All**, **TCP**, **UDP**, **ICMP**, or **GRE**. – Port: Port or port range over which the traffic can reach your instance. The port ranges from 1 to 65535.	In this example, select **TCP**. Leave the port blank or set it to the data source port.
Type	Type of IP addresses.	IPv4
Source	Allows access from IP addresses or instances in another security group.	In this example, enter the obtained queue CIDR block.
Description	Supplementary information about the security group rule. This parameter is optional.	–

Step 3 Test the Connectivity Between the DLI Queue and the Data Source

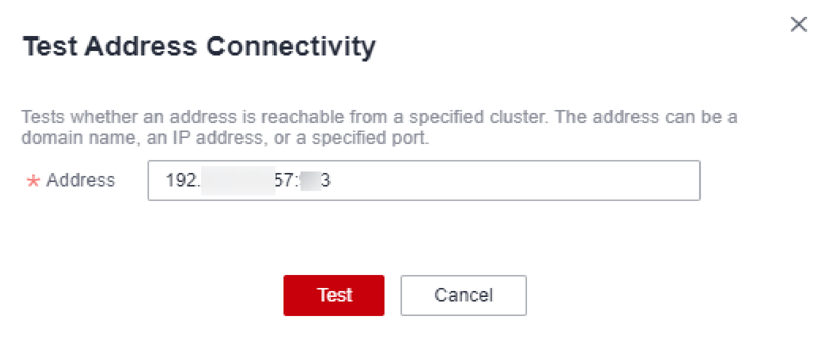
1. Obtain the private IP address and port number of the data source.

   Take the RDS data source as an example. On the **Instances** page, click the target DB instance. On the page displayed, locate the **Connection Information** pane and view the private IP address. In the **Connection Information** pane, locate the **Database Port** to view the port number of the RDS DB instance.
2. In the navigation pane of the DLI management console, choose **Resources** > **Queue Management**.
3. Locate the queue bound with the enhanced datasource connection, click **More** in the **Operation** column, and select **Test Address Connectivity**.
4. Enter the data source connection address and port number to test the network connectivity.

   Format: *IP address*:*Port number*

> **CAUTION**
>
> Before testing the connection, ensure that the security group of the external data source has allowed access from the CIDR block of the queue.

Figure 5-52 Testing the network connectivity between the queue and the data source



----End

Related Operations

- **Why Is a Datasource Connection Successfully Created But the Network Connectivity Test Fails?**

5.3.2.3 Establishing a Network Connection Between DLI and Resources in a Shared VPC

VPC Sharing Overview

VPC sharing allows sharing VPC resources created in one account with other accounts using Resource Access Manager (RAM). For example, account A can share its VPC and subnets with account B. After accepting the share, account B can view the shared VPC and subnets and use them to create resources.

For more information about VPC sharing, see **VPC Sharing** in *Virtual Private Cloud User Guide*.

DLI Use Cases

An enterprise IT management account creates a VPC and subnets and shares them with other service accounts to facilitate centralized configuration of VPC security policies and orderly resource management.

Service accounts use the shared VPC and subnets to create resources and want to use DLI to submit jobs and access resources in the shared VPC. To do this, they need to establish a network connection between DLI and the resources in the shared VPC.

For example, account A is the enterprise IT management account and the owner of VPC resources. It creates the VPC and subnets and shares them with service account B.

Account B is a service account that uses the shared VPC and subnets to create resources and uses DLI to access them.

Prerequisites

- Account A has been configured with a DLI agency, which includes the **DLI Datasource Connections Agency Access** permission. This will grant the necessary permissions to access and use VPCs, subnets, routes, and VPC peering connections. For details, see **Configuring DLI Agency Permissions**.
- Account A, as the resource owner, has created a VPC and subnets and designated account B as the principal.

  For details, see **Creating a Resource Share**.

Establishing a Network Connection Between DLI and Resources in a Shared VPC

Step 1 Account A creates an enhanced datasource connection.

1. Log in to the DLI management console using account A.
2. In the navigation pane on the left, choose **Datasource Connections**.
3. On the displayed **Enhanced** tab, click **Create**.

   Set parameters based on **Table 5-22**.

   Table 5-22 Parameters for creating an enhanced datasource connection

Parameter	Description
Connection Name	Name of the datasource connection to be created
Resource Pool	You do not need to set this parameter in this scenario.
VPC	VPC shared by account A to account B
Subnet	Subnet shared by account A to account B
Route Table	You do not need to set this parameter in this scenario.
Host Information	You do not need to set this parameter in this scenario.
Tags	Tags used to identify cloud resources. A tag includes the tag key and tag value.

4. Click **OK**.

Step 2 Account A grants account B access to the enhanced datasource connection created in **Step 1**.

1. In the enhanced datasource connection list, locate the row containing the newly created one, click **More** in the **Operation** column, and select **Manage Permission** from the drop-down list.
2. In the displayed **Permissions** dialog box, select **Grant Permission** for **Set Permission**, enter the ID of the project account B belongs to in **Project ID**, and click **OK**.

Step 3 Account B binds a DLI elastic resource pool to the shared enhanced datasource connection.

1. Log in to the DLI management console using account B.
2. In the navigation pane on the left, choose **Datasource Connections**.
3. On the displayed **Enhanced** tab, locate the row containing the enhanced datasource connection shared by account A, click **More** in the **Operation** column, and select **Bind Resource Pool** from the drop-down list.
4. In the displayed **Bind Resource Pool** dialog box, select the created elastic resource pool for **Resource Pool** and click **OK**.

   If there is no elastic resource pool available, create one by referring to **Creating an Elastic Resource Pool and Creating Queues Within It**.

Step 4 Account B tests the network connectivity between the elastic resource pool and resources in the VPC.

NOTE

If there are resources in the shared VPC, ensure that the security group the resources belong to has allowed access to the elastic resource pool's CIDR block.

1. Obtain the private IP address and port number of the data source in the shared VPC.

   Take the RDS data source as an example. On the **Instances** page, click the target DB instance. On the displayed page, locate the **Connection Information** pane and view the private IP address. In the **Connection Information** pane, locate the **Database Port** to view the port number of the RDS DB instance.
2. In the navigation pane of the DLI management console, choose **Resources** > **Queue Management**.
3. Locate the queue under the elastic resource pool bound with the enhanced datasource connection, click **More** in the **Operation** column, and select **Test Address Connectivity**.
4. Enter the data source connection address and port number to test the network connectivity.

   If the address is reachable, it means that account B has established a network connection between the DLI resource and the resources in the shared VPC. Account B can then submit jobs to the elastic resource pool's queue and access the resources in the shared VPC.

----End

5.3.2.4 Common Development Methods for DLI Cross-Source Analysis

Cross-Source Analysis

If DLI needs to access external data sources, you need to establish enhanced datasource connections to enable the network between DLI and the data sources, and then develop different types of jobs to access the data sources. This is the process of DLI cross-source analysis.

This section describes how to develop data sources supported by DLI for cross-source analysis.

Notes

- Flink jobs can directly access DIS, OBS, and SMN data sources without using datasource connections.
- You are advised to use enhanced datasource connections to connect DLI to data sources.

Cross-Source Analysis Development Methods

Table 5-23 lists the data sources supported by DLI and the corresponding development methods.

Table 5-23 Syntax reference for cross-source analysis

Service	Spark SQL Job	Spark Jar Job	Flink OpenSource SQL Job	Flink Jar Job
CloudTable HBase	• **Create an HBase association table** • **Insert data** • **Query data**	• **Scala sample code** • **PySpark sample code** • **Java sample code**	• **HBase source table** • **HBase result table** • **HBase dimension table**	-
CloudTable OpenTSDB	• **Create an OpenTSDB association table** • **Insert data** • **Query data**	• **Scala sample code** • **PySpark sample code** • **Java sample code**	-	-
CSS	• **Create a CSS association table** • **Insert data** • **Query data**	• **Scala sample code** • **PySpark sample code** • **Java sample code**	• **Elasticsearch result table**	-

Service	Spark SQL Job	Spark Jar Job	Flink OpenSource SQL Job	Flink Jar Job
DCS Redis	• Create a DCS association table • Insert data • Query data	• Scala sample code • PySpark sample code • Java sample code	• Redis source table • Redis result table • Redis dimension table	Flink job sample
DDS	• Create a DDS association table • Insert data • Query data	• Scala sample code • PySpark sample code • Java sample code	-	-
DMS	-	-	• Kafka source table • Kafka result table	-
GaussDB(DWS)	• Create a GaussDB(DWS) association table • Insert data • Query data	• Scala sample code • PySpark sample code • Java sample code	• GaussDB(DWS) source table • GaussDB(DWS) result table • GaussDB(DWS) dimension table	Flink job sample
MRS HBase	• Create an HBase association table • Insert data • Query data	• Scala sample code • PySpark sample code • Java sample code	• HBase source table • HBase result table • HBase dimension table	Flink job sample

Service	Spark SQL Job	Spark Jar Job	Flink OpenSource SQL Job	Flink Jar Job
MRS Kafka	-	-	• **Kafka source table** • **Kafka result table**	• **Flink job sample**
MRS OpenTSDB	• **Create an OpenTSDB association table** • **Insert data** • **Query data**	• **Scala sample code** • **PySpark sample code** • **Java sample code**	-	-
RDS for MySQL	• **Create an RDS association table** • **Insert data** • **Query data**	• **Scala sample code** • **PySpark sample code** • **Java sample code**	• **MySQL CDC source table**	-
RDS PostGre	• **Create an RDS association table** • **Insert data** • **Query data**	• **Scala sample code** • **PySpark sample code** • **Java sample code**	• **Postgres CDC source table**	-

5.3.3 Using DEW to Manage Access Credentials for Data Sources

When using DLI to submit jobs that involve reading and writing data from external sources, it is crucial to securely access these sources by properly storing their access credentials. This ensures the authentication of the data source and enables secure access by DLI. DEW is a comprehensive cloud-based encryption service that addresses data security, key security, and complex key management issues. This section describes how to use DEW to store authentication information for a data source.

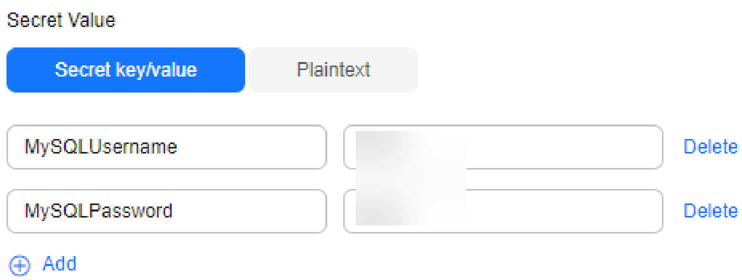
For details, see **Data Encryption Workshop (DEW)**.

Creating a Shared Secret in DEW

This example describes how to configure a credential for accessing RDS DB instances in a DLI job and store the credential in DEW.

1. Log in to the DEW management console.
2. In the navigation pane on the left, choose **Cloud Secret Management Service** > **Secrets**.
3. Click **Create Secret**. On the displayed page, configure basic secret information.
   - **Secret Name**: Enter a secret name. In this example, the name is **secretInfo**.
   - **Secret Value**: Enter the username and password for logging in to the RDS for MySQL DB instance.
     - The key in the first line is **MySQLUsername**, and the value is the username for logging in to the DB instance.
     - The key in the second line is **MySQLPassword**, and the value is the password for logging in to the DB instance.

Figure 5-53 Secret Value



4. Set other parameters as required and click **OK**.

Using the Secret Created in DEW in a DLI Job

This part uses a Flink job as an example to describe how to use credentials created in DEW.

```
WITH (
 'connector' = 'jdbc',
  'url? = 'jdbc:mysql://MySQLAddress:MySQLPort/flink',--flink is the MySQL database where the orders table locates.
 'table-name' = 'orders',
 'username' = 'MySQLUsername',  -- Shared secret in DEW whose name is secretInfo and version is v1. The key MySQLUsername defines the secret value. The value is the user's sensitive information.
 'password' = 'MySQLPassword',  -- Shared secret in DEW whose name is secretInfo and version is v1. The key MySQLPassword defines the secret value. The value is the user's sensitive information.
 'sink.buffer-flush.max-rows' = '1',
 'dew.endpoint'='kms.xxxx.com', --Endpoint information for the DEW service being used
 'dew.csms.secretName'='secretInfo', --Name of the DEW shared secret
 'dew.csms.decrypt.fields'='username,password', --The password field value must be decrypted and replaced using DEW secret management.
```

```
 'dew.csms.version'='v1'
);
```

Related Operations

For how to use a DLI agency to obtain access credentials, see **Table 5-24**.

Table 5-24 Guidelines for configuring DLI agency permissions in specific scenarios

Type	Instruction	Description
Flink job	**Flink OpenSource SQL Jobs Using DEW to Manage Access Credentials**	Guideline for using DEW to manage and access credentials for Flink OpenSource SQL jobs. When writing the output data of Flink jobs to MySQL or GaussDB(DWS), set attributes such as the username and password in the connector.
	Flink Jar Jobs Using DEW to Acquire Access Credentials for Reading and Writing Data from and to OBS	Guideline for Flink Jar jobs to acquire an AK/SK to read and write data from and to OBS.
	Obtaining Temporary Credentials for Flink Job Agencies	DLI provides a common interface to obtain temporary credentials for Flink job agencies set by users during job launch. The interface encapsulates the obtained temporary credentials for the job agency in the **com.huaweicloud.sdk.core.auth.BasicCredentials** class. Guideline for obtaining a temporary credential for a Flink job agency.
Spark job	**Spark Jar Jobs Using DEW to Acquire Access Credentials for Reading and Writing Data from and to OBS**	Guideline for Spark Jar jobs to acquire an AK/SK to read and write data from and to OBS.
	Obtaining Temporary Credentials for Spark Job Agencies	Guideline for obtaining a temporary credential for a Spark Jar job agency.

5.3.4 Using DLI Datasource Authentication to Manage Access Credentials for Data Sources

5.3.4.1 Overview

What Is Datasource Authentication?

When analyzing across multiple sources, you are not advised to configure authentication information directly in a job as it can lead to password leakage. Instead, you are advised to use either Data Encryption Workshop (DEW) or datasource authentication provided by DLI to securely store data source authentication information.

- DEW is a comprehensive cloud-based encryption service that addresses data security, key security, and complex key management issues. You are advised to use DEW to store authentication information for data sources.

  You are advised to use DEW to store authentication information of data sources when Spark 3.3.1 or later and Flink 1.15 or later jobs access data sources using datasource connections. This will help you address issues related to data security, key security, and complex key management. For details, see **Using DEW to Manage Access Credentials for Data Sources**.
- Datasource authentication is used to manage authentication information for accessing specified data sources. After datasource authentication is configured, you do not need to repeatedly configure data source authentication information in jobs, improving data source authentication security while enabling DLI to securely access data sources.

  When SQL and Flink 1.12 jobs access data sources using datasource connections, use DLI's datasource authentication feature to manage data source access credentials.

This section describes how to use datasource authentication provided by DLI.

Notes and Constraints

- Only Spark SQL and Flink OpenSource SQL 1.12 jobs support datasource authentication.
- Flink jobs can use datasource authentication only on queues created after May 1, 2023.
- DLI supports four types of datasource authentication. Select an authentication type specific to each data source.
  - CSS: applies to 6.5.4 or later CSS clusters with the security mode enabled.
  - Kerberos: applies to MRS security clusters with Kerberos authentication enabled.
  - Kafka_SSL: applies to Kafka with SSL enabled.
  - Password: applies to GaussDB(DWS), RDS, DDS, and DCS.

Datasource Authentication Types

DLI supports four types of datasource authentication. Select an authentication type specific to each data source.

- CSS: applies to 6.5.4 or later CSS clusters with the security mode enabled. During the configuration, you need to specify the username, password, and authentication certificate of the cluster and store the information in DLI through datasource authentication so that DLI can securely access CSS data sources. For details, see **Creating a CSS Datasource Authentication**.

- Kerberos: applies to MRS security clusters with Kerberos authentication enabled. During the configuration, you need to specify MRS cluster authentication credentials, including the **krb5.conf** and **user.keytab** files. For details, see **Creating a Kerberos Datasource Authentication**.
- Kafka_SSL: applies to Kafka with SSL enabled. During the configuration, you need to specify the KafkaTruststore path and password. For details, see **Creating a Kafka_SSL Datasource Authentication**.
- Password: applies to GaussDB(DWS), RDS, DDS, and DCS data sources. During the configuration, you need to store the passwords of the data sources in DLI. For details, see **Creating a Password Datasource Authentication**.

Jobs That Can Connect to Data Sources Through Datasource Authentication

Different types of jobs can connect to data sources through different types of datasource authentication.

- For details about the data sources that Spark SQL jobs can connect to through datasource authentication and their constraints, see **Table 5-25**.
- For details about the data sources that Flink OpenSource SQL 1.12 jobs can connect to through datasource authentication and their constraints, see **Table 5-26**.

Table 5-25 Data sources that Spark SQL jobs can connect to through datasource authentication

Datasource Authentication Type	Data Source	Notes and Constraints
CSS	CSS	The CSS cluster version must be 6.5.4 or later. The security mode has been enabled for the CSS cluster.
Password	GaussDB(DWS), RDS, DDS, and Redis	-

Table 5-26 Data sources that Flink OpenSource SQL 1.12 jobs can connect to through datasource authentication

Table Type	Datasource Authentication Type	Data Source	Notes and Constraints
Source table	Kerberos	HBase	Kerberos authentication has been enabled for the MRS cluster.
		Kafka	Kerberos authentication has been enabled for MRS Kafka.

Table Type	Datasource Authentication Type	Data Source	Notes and Constraints
	Kafka_SSL	Kafka	SASL_SSL authentication has been enabled for DMS Kafka. SASL authentication has been enabled for MRS Kafka. SSL authentication has been enabled for MRS Kafka.
	Password	GaussDB(DWS), RDS, and Redis	-
Result table	Kerberos	HBase	Kerberos authentication has been enabled for the MRS cluster.
		Kafka	Kerberos authentication has been enabled for MRS Kafka.
	Kafka_SSL	Kafka	SASL_SSL authentication has been enabled for DMS Kafka. SASL authentication has been enabled for MRS Kafka. SSL authentication has been enabled for MRS Kafka.
	Password	GaussDB(DWS), RDS, CSS, and Redis	-
Dimension table	Kerberos	HBase	Kerberos authentication has been enabled for the MRS cluster.
	Password	GaussDB(DWS), RDS, and Redis	-

5.3.4.2 Creating a CSS Datasource Authentication

Scenario

Create a CSS datasource authentication on the DLI console to store the authentication information of the CSS security cluster to DLI. This will allow you to access to the CSS security cluster without having to configure a username and password in SQL jobs.

Create a datasource authentication for a CSS security cluster on the DLI console.

Notes

A CSS security cluster has been created and has met the following conditions:

- The cluster version is 6.5.4 or later.
- The security mode has been enabled for the cluster.

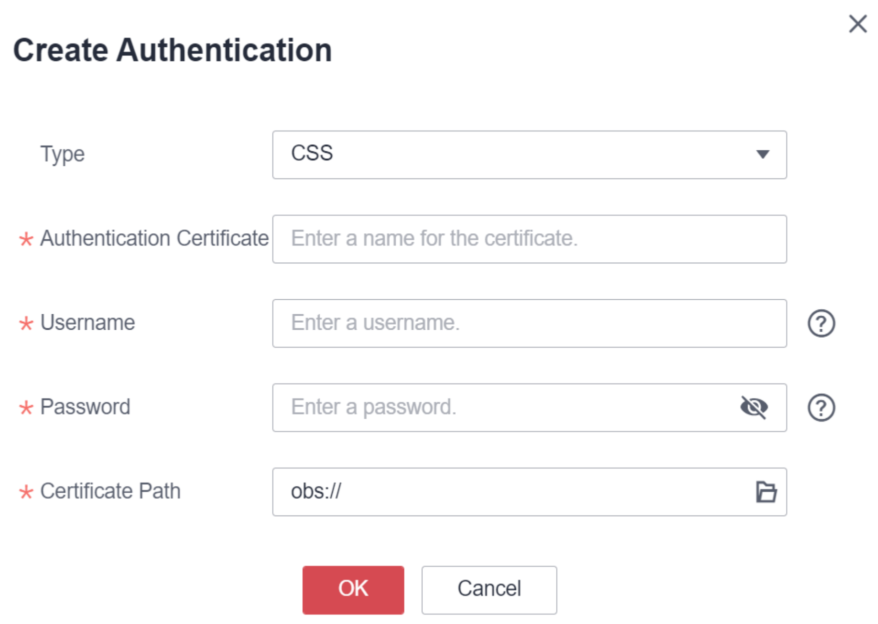
Procedure

1. Download the authentication credential of the CSS security cluster.
   a. Log in to the CSS management console and choose **Clusters** > **Elasticsearch**.
   b. On the **Clusters** page displayed, click the cluster name.
   c. On the **Cluster Information** page displayed, find the security mode and download the certificate of the CSS security cluster.
2. Upload the authentication credential to the OBS bucket.
3. Create a datasource authentication.
   a. Log in to the DLI management console.
   b. Choose **Datasource Connections**. On the page displayed, click **Datasource Authentication**.
   c. Click **Create**.
   Configure CSS authentication parameters according to **Table 5-27**.

Table 5-27 Parameters

Parameter	Description
Authentication Certificate	Name of the datasource authentication information to be created. ● The name can contain only digits, letters, and underscores (_), but cannot contain only digits or start with an underscore (_). ● The length of the database name cannot exceed 128 characters. ● It is recommended that the name contain the CSS security cluster name to distinguish security authentication information of different clusters.
Type	Select **CSS**.
Username	Username for logging in to the security cluster.
Password	The password of the security cluster
Certificate Path	Enter the OBS path to which the security certificate is uploaded, that is, the OBS bucket address in 2.

Figure 5-54 Creating a datasource authentication-CSS



4. Create a table to access the CSS cluster.

   When creating a table, associate the table with the created datasource authentication to access the CSS cluster.

   For example, when using Spark SQL to create a table for accessing the CSS cluster, configure **es.certificate.name** to set the datasource authentication name and then connect to the CSS security cluster.

   Use Spark SQL to create a table for accessing the CSS cluster by referring to **Creating a DLI Table and Associating It with CSS**.

5.3.4.3 Creating a Kerberos Datasource Authentication

Scenario

Create a Kerberos datasource authentication on the DLI console to store the authentication information of the data source to DLI. This will allow you to access to the data source without having to configure a username and password in SQL jobs.

NOTE

- When Kerberos authentication is enabled for MRS Kafka but SSL authentication is disabled, create a Kerberos authentication. When creating a table, configure **krb_auth_name** to associate the datasource authentication.
- If Kerberos authentication and SSL authentication are both enabled for MRS Kafka, you need to create Kerberos and Kafka_SSL authentications. When creating a table, configure **krb_auth_name** and **ssl_auth_name** to associate the datasource authentications.
- Datasource authentication is not required when Kerberos authentication is disabled but SASL authentication is enabled for MRS Kafka (for example, when a username and a password are used for PlainLoginModule authentication).
- When Kerberos authentication is disabled but SSL authentication is enabled for MRS Kafka, you need to create a Kafka_SSL authentication. When creating a table, configure **ssl_auth_name** to associate the datasource authentication.
- When Kerberos authentication is disabled but SASL authentication and SSL authentication are enabled for MRS Kafka, you need to create a Kafka_SSL authentication. When creating a table, configure **ssl_auth_name** to associate the datasource authentication.

Data Sources Supported by Kerberos Datasource Authentication

Table 5-28 lists the data sources supported by Kerberos datasource authentication.

Table 5-28 Data sources supported by Kerberos datasource authentication

Job Type	Table Type	Data Source	Notes and Constraints
Flink OpenSource SQL	Source table	HBase	Kerberos authentication has been enabled for the MRS cluster.
		Kafka	Kerberos authentication has been enabled for MRS Kafka.
	Result table	HBase	Kerberos authentication has been enabled for the MRS cluster.
		Kafka	Kerberos authentication has been enabled for MRS Kafka.
	Dimension table	HBase	Kerberos authentication has been enabled for the MRS cluster.

Procedure

1. Download the authentication credential of the data source.
   a. Log in to MRS Manager.
   b. Choose **System** > **Permission** > **User**.

c. Click **More**, select **Download Authentication Credential**, save the file, and decompress it to obtain the **keytab** and **krb5.conf** files.

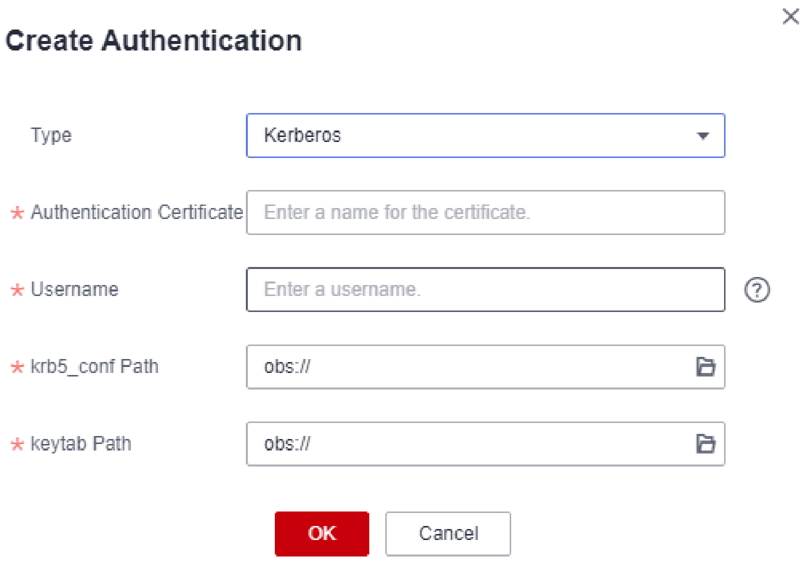
2. Upload the authentication credential to the OBS bucket.
3. Create a datasource authentication.
   a. Log in to the DLI management console.
   b. Choose **Datasource Connections**. On the page displayed, click **Datasource Authentication**.
   c. Click **Create**.

   Configure Kerberos authentication parameters according to **Table 5-29**.

Table 5-29 Parameters

Parameter	Description
Type	Select **Kerberos**.
Authentication Certificate	Name of the datasource authentication to be created. ● The name can contain only digits, letters, and underscores (_), but cannot contain only digits or start with an underscore (_). ● The name can contain a maximum of 128 characters. ● It is recommended that the name contain the MRS security cluster name to distinguish security authentication information of different clusters.
Username	Username for logging in to the security cluster.
krb5_conf Path	OBS path to which the **krb5.conf** file is uploaded. **NOTE** The **renew_lifetime** configuration item under **[libdefaults]** must be removed from **krb5.conf**. Otherwise, the "Message stream modified (41)" error may occur.
keytab Path	OBS path to which the **user.keytab** file is uploaded.

Figure 5-55 Creating a datasource authentication – Kerberos



4. Create a table to access the MRS cluster.

   When creating a data source, associate the data source with the created datasource authentication to access the data source.

   Table 5-30 lists the fields used to associate with the datasource authentication during table creation.

Table 5-30 Fields that are used to associate with Kerberos datasource authentication during table creation

Job Type	Data Source	Parameter	Mandatory	Data Type	Description
Flink OpenSource SQL	HBase	krb_auth_name	No	String	This field is used to associate datasource authentications when source, result, and dimension tables are created.

Job Type	Data Source	Parameter	Mandatory	Data Type	Description
	Kafka	krb_auth_name	No	String	This field is used to associate datasource authentications when source and result tables are created. Name of the created Kerberos datasource authentication. If SASL_PLAINTEXT and Kerberos authentication are both used, you need to configure the following parameters: ● 'properties.sasl.mechanism' = 'GSSAPI' ● 'properties.security.protocol' = 'SASL_PLAINTEXT'

For details about how to create a table, see *Data Lake Insight Syntax Reference*.

– Flink OpenSource SQL Syntax Reference: **Creating an HBase Source Table**

5.3.4.4 Creating a Kafka_SSL Datasource Authentication

Scenario

Create a Kafka_SSL datasource authentication on the DLI console to store the Kafka authentication information to DLI. This will allow you to access to Kafka instances without having to configure a username and password in SQL jobs.

NOTE

- When Kerberos authentication is enabled for MRS Kafka but SSL authentication is disabled, create a Kerberos authentication. When creating a table, configure **krb_auth_name** to associate the datasource authentication.
- If Kerberos authentication and SSL authentication are both enabled for MRS Kafka, you need to create Kerberos and Kafka_SSL authentications. When creating a table, configure **krb_auth_name** and **ssl_auth_name** to associate the datasource authentications.
- Datasource authentication is not required when Kerberos authentication is disabled but SASL authentication is enabled for MRS Kafka (for example, when a username and a password are used for PlainLoginModule authentication).
- When Kerberos authentication is disabled but SSL authentication is enabled for MRS Kafka, you need to create a Kafka_SSL authentication. When creating a table, configure **ssl_auth_name** to associate the datasource authentication.
- When Kerberos authentication is disabled but SASL authentication and SSL authentication are enabled for MRS Kafka, you need to create a Kafka_SSL authentication. When creating a table, configure **ssl_auth_name** to associate the datasource authentication.

Data Sources Supported by Kafka_SSL Datasource Authentication

Table 5-31 lists the data sources supported by Kafka_SSL datasource authentication.

Table 5-31 Data sources supported by Kafka_SSL datasource authentication

Job Type	Table Type	Data Source	Notes and Constraints
Flink OpenSource SQL	Source table and result table	Kafka	SASL_SSL authentication has been enabled for DMS Kafka. SASL authentication has been enabled for MRS Kafka. SSL authentication has been enabled for MRS Kafka.

Procedure

1. Download the authentication credential.
   - **DMS Kafka**
     i. Log in to the DMS (for Kafka) console and click a Kafka instance to access its details page.
     ii. In the connection information, find the SSL certificate and click **Download**.
     Decompress the downloaded **kafka-certs** package to obtain the **client.jks** and **phy_ca.crt** files.
   - **MRS Kafka**
     i. Log in to MRS Manager.

ii. Choose **System** > **Permission** > **User**.

iii. Click **More**, select **Download Authentication Credential**, save the file, and decompress it to obtain the truststore file.

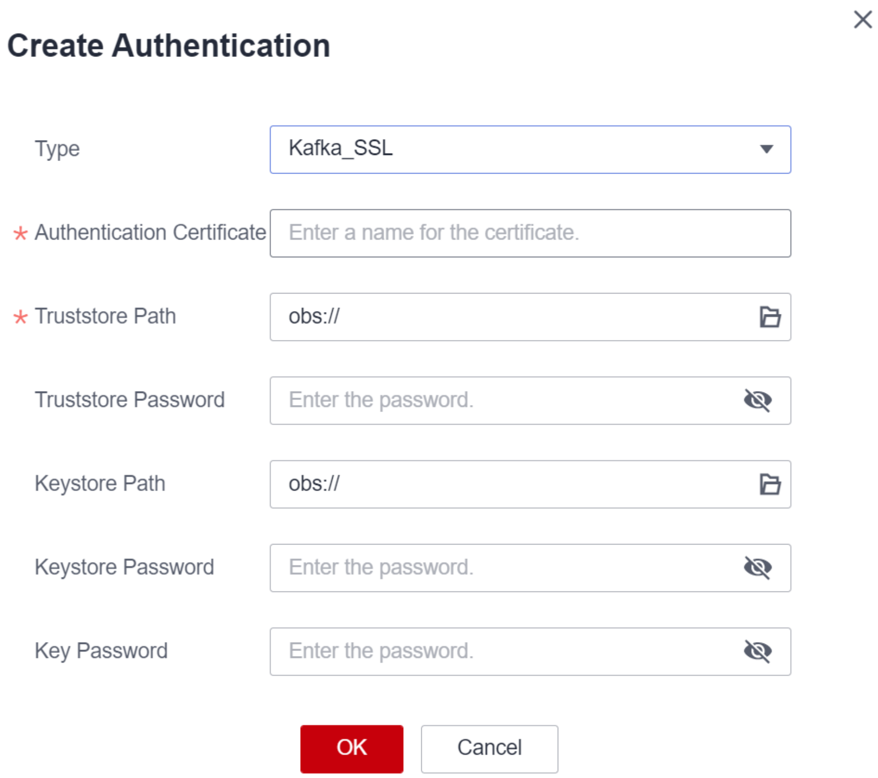
2. Upload the authentication credential to the OBS bucket.
3. Create a datasource authentication.
   a. Log in to the DLI management console.
   b. Choose **Datasource Connections**. On the page displayed, click **Datasource Authentication**.
   c. Click **Create**.

   Configure Kafka authentication parameters according to **Table 5-32**.

Table 5-32 Parameters

Parameter	Description
Type	Select **Kafka_SSL**.
Authentication Certificate	Name of the datasource authentication to be created. ● The name can contain only digits, letters, and underscores (_), but cannot contain only digits or start with an underscore (_). ● The name can contain a maximum of 128 characters.
Truststore Path	OBS path to which the SSL truststore file is uploaded. ● For MRS Kafka, enter the OBS path of the **Truststore.jks** file. ● For DMS Kafka, enter the OBS path of the **client.jks** file.
Truststore Password	Truststore password.
Keystore Path	OBS path to which the SSL keystore file (key and certificate) is uploaded.
Keystore Password	Keystore (key and certificate) password.
Key Password	Password of the private key in the keystore file.

Figure 5-56 Creating a datasource authentication – Kafka_SSL



4. Access Kafka with SASL_SSL authentication enabled.

   When creating a data source, associate the data source with the created datasource authentication to access the data source.

   Table 5-33 lists the fields used to associate with the datasource authentication during table creation.

Table 5-33 Fields that are used to associate with Kafka_SSL datasource authentication during table creation

Parameter	Mandatory	Data Type	Description
ssl_auth_name	No	String	This field is used to associate datasource authentications when source, result, and dimension tables are created. Name of the created Kafka_SSL datasource authentication. This configuration is used when SSL is configured for Kafka. ● If only SSL is used, configure the following parameter: 'properties.security.protocol '= 'SSL'; ● If SASL_SSL is used, configure the following parameters: – 'properties.security.protocol' = 'SASL_SSL', – 'properties.sasl.mechanism' ='GSSAPI or PLAIN' – 'properties.sasl.jaas.config' = 'org.apache.kafka.common.security.plain.PlainLoginModule required username=\"xxx\" password=\"xxx\";'

For details about how to create a table, see *Data Lake Insight Syntax Reference*.

– Flink OpenSource SQL Syntax Reference: **Creating a Kafka Source Table**

5.3.4.5 Creating a Password Datasource Authentication

Scenario

Create a password datasource authentication on the DLI console to store passwords of the GaussDB(DWS), RDS, DCS, and DDS data sources to DLI. This will allow you to access to the data sources without having to configure a username and password in SQL jobs.

Data Sources Supported by Password Datasource Authentication

Table 5-34 lists the data sources supported by password datasource authentication.

Table 5-34 Data sources supported by password datasource authentication

Job Type	Table Type	Data Source
Spark SQL	-	GaussDB(DWS), RDS, DDS, and Redis
Flink OpenSource SQL	Source table	GaussDB(DWS), RDS, and Redis
	Result table	GaussDB(DWS), RDS, CSS, and Redis
	Dimension table	GaussDB(DWS), RDS, and Redis

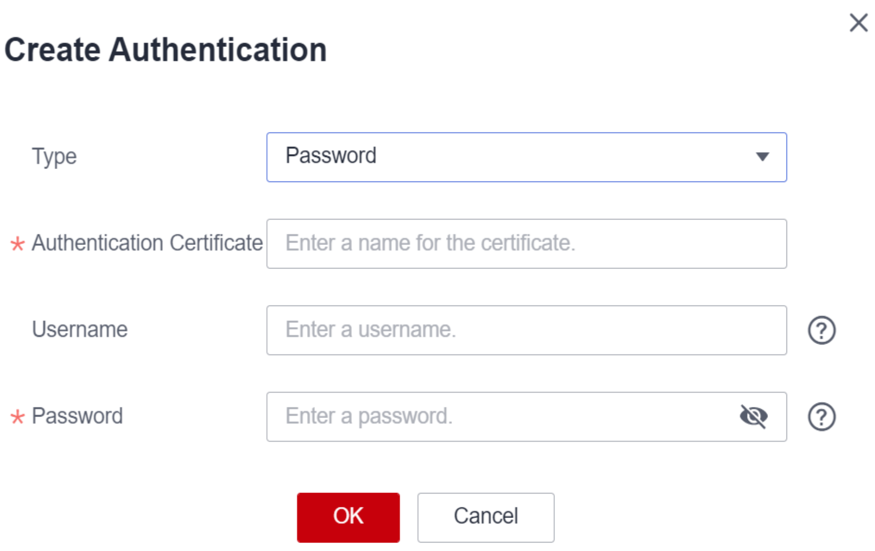
Procedure

1. Create a datasource authentication.
   a. Log in to the DLI management console.
   b. Choose **Datasource Connections**. On the page displayed, click **Datasource Authentication**.
   c. Click **Create**.
   Configure authentication parameters according to **Table 5-35**.

Table 5-35 Parameters

Parameter	Description
Type	Select **Password**.
Authentication Certificate	Name of the datasource authentication to be created. ● The name can contain only digits, letters, and underscores (_), but cannot contain only digits or start with an underscore (_). ● The name can contain a maximum of 128 characters.
Username	Username for accessing the data source.
Password	Password for accessing the data source.

Figure 5-57 Creating a datasource authentication – Password



2. Access the data source.

   When creating a data source, associate the data source with the created datasource authentication to access the data source.

   Table 5-36 lists the fields used to associate with the datasource authentication during table creation.

Table 5-36 Fields that are used to associate with password datasource authentication during table creation

Job Type	Parameter	Mandatory	Data Type	Description
Spark SQL	passwdauth	No	String	Name of datasource authentication. It is applicable to GaussDB(DWS), RDS, DDS, and Redis data sources.
Flink OpenSource SQL	pwd_auth_name	No	String	This field is used to associate datasource authentications when source, result, and dimension tables are created. Set **pwd_auth_name** to the name of the password datasource authentication. If this parameter is set, you do not need to configure a username and a password of the data source in SQL jobs.

For details about how to create a table, see *Data Lake Insight Syntax Reference*.

– Flink OpenSource SQL Syntax Reference: **Creating a GaussDB(DWS) Source Table**

5.3.4.6 Datasource Authentication Permission Management

Scenario

Grant permissions on a datasource authentication to users so multiple user jobs can use the datasource authentication without affecting each other.

Notes

- The administrator and the datasource authentication owner have all permissions. You do not need to set permissions for them, and their datasource authentication permissions cannot be modified by other users.
- When setting datasource authentication permissions for a new user, ensure that the user group to which the user belongs has the **Tenant Guest** permission.

  For details about the **Tenant Guest** permission and how to apply for the permission, see **System Permissions** and **Creating a User Group** in *Identity and Access Management User Guide*.

Granting Permissions on Datasource Connections

1. Log in to the DLI management console.
2. Choose **Datasource Connections**. On the page displayed, click **Datasource Authentication**.
3. Locate the row containing the datasource authentication to be authorized and click **Manage Permission** in the **Operation** column. The **User Permissions** page is displayed.
4. Click **Grant Permission** in the upper right corner of the page. On the **Grant Permission** dialog box displayed, grant permissions on this datasource authentication to other users.

Figure 5-58 Granting permissions on datasource connections

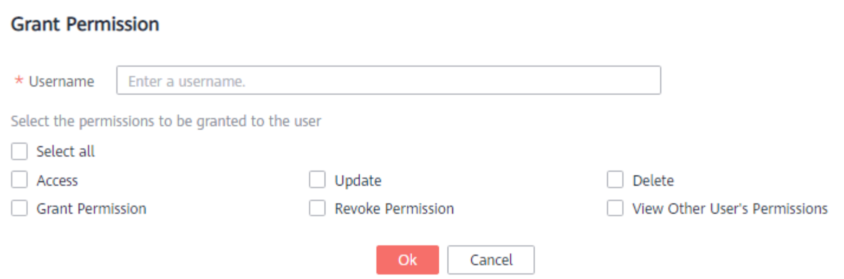


Table 5-37 Permission granting parameters

Parameter	Description
Username	Name of the IAM user to whom permissions on the datasource connection are to be granted. **NOTE** The username is the name of an existing IAM user.
Select the permissions to be granted to the user	• Access: This permission allows you to access the datasource connection. • Update: This permission allows you to update the datasource connection. • Delete: This permission allows you to delete the datasource connection. • Grant Permission: This permission allows you to grant the datasource connection permission to other users. • Grant Permission: This permission allows you to revoke the datasource connection permission to other users. However, you cannot revoke the permissions of the datasource connection owner. • View Other User's Permissions: This permission allows you to view the datasource connection permissions of other users.

Modifying the Permissions of Current User

1. Log in to the DLI management console.
2. Choose **Datasource Connections**. On the page displayed, click **Datasource Authentication**.
3. Locate the row containing the datasource authentication to be authorized and click **Manage Permission** in the **Operation** column. The **User Permissions** page is displayed.
4. Click **Set Permission** in the **Operation** column to modify the permissions of the current user. **Table 5-37** lists the detailed permission descriptions.

NOTE

- If all options under **Set Permission** are gray, you are not allowed to change permissions on this datasource connection. You can apply to the administrator, group owner, or other users who have the permission to grant permissions for the permissions to grant and revoke the datasource authentication permissions.
- To revoke all permissions of the current user, click **Revoke Permission** in the **Operation** column. The IAM user will no longer have any permission on the datasource authentication.

5.3.5 Managing Enhanced Datasource Connections

5.3.5.1 Viewing Basic Information About an Enhanced Datasource Connection

After creating an enhanced datasource connection, you can view and manage it on the management console.

This section describes how to view basic information about an enhanced datasource connection on the management console, including whether the enhanced datasource connection supports IPv6, host information, and more.

Procedure

1. Log in to the DLI management console.
2. In the navigation pane on the left, choose **Datasource Connections**.
3. On the displayed **Enhanced** tab, locate the enhanced datasource connection whose basic information you want to view.
   - In the upper right corner of the list page, click to customize the columns to display and set the rules for displaying the table content and the **Operation** column.
   - In the search box above the list, you can filter the required enhanced datasource connection by name or tag.
4. Click to expand details about the enhanced datasource connection.

   You can view the following information:

   a. **IPv6 Support**: If you selected a subnet with IPv6 enabled when creating the enhanced datasource connection, then your enhanced datasource connection will support IPv6.

   b. **Host Information**: When accessing an MRS HBase cluster, you need to configure the host name (domain name) and the corresponding IP address of the instance. For details, see **Modifying Host Information in an Elastic Resource Pool**.

5.3.5.2 Enhanced Connection Permission Management

Scenario

Enhanced connections support user authorization by project. After authorization, users in the project have the permission to perform operations on the enhanced connection, including viewing the enhanced connection, binding a created resource pool to the enhanced connection, and creating custom routes. In this way, the enhanced connection can be used across projects. Grant and revoke permissions to and from a user for an enhanced connection.

NOTE

- If the authorized projects belong to different users in the same region, you can use the user account of the authorized projects to log in.
- If the authorized projects belong to the same user in the same region, you can use the current account to switch to the corresponding project.

Use Cases

Project B needs to access the data source of project A. The operations are as follows:

- For Project A:
  a. Log in to DLI using the account of project A.
  b. Create an enhanced datasource connection **ds** in DLI based on the VPC information of the corresponding data source.
  c. Grant project B the permission to access the enhanced datasource connection **ds**.
- For Project B:
  a. Log in to DLI using the account of project B.
  b. Bind the enhanced datasource connection **ds** to a queue.
  c. (Optional) Set host information and create a route.

After creating a VPC peering connection and route between the enhanced datasource connection of project A and the queue of project B, you can create a job in the queue of project B to access the data source of project A.

Procedure

1. Log in to the DLI management console.
2. In the left navigation pane, choose **Datasource Connections**.
3. On the **Enhanced** tab page displayed, locate the desired enhanced connection, click **More** in the **Operation** column, and select **Manage Permission**.
   - **Granting permission**
     i. In the **Permissions** dialog box displayed, select **Grant Permission** for **Set Permission**.
     ii. Enter the project ID.
     iii. Click **OK** to grant the resource pool operation permission to the project.
   - **Revoking permission**
     i. In the **Permissions** dialog box displayed, select **Revoke Permission** for **Set Permission**.
     ii. Select a project ID.
     iii. Click **OK** to revoke the resource pool operation permission from the specified project.

5.3.5.3 Binding an Enhanced Datasource Connection to an Elastic Resource Pool

Scenario

To connect other resource pools to data sources through enhanced datasource connections, bind enhanced datasource connections to resource pools on the **Enhanced** tab page.

Constraints

- Enhanced datasource connections support only dedicated pay-per-use resource pools and queues.
- The CIDR block of the DLI queue to be bound with a datasource connection cannot overlap with that of the data source.
- The **default** queue preset in the system cannot be bound with a datasource connection.

Procedure

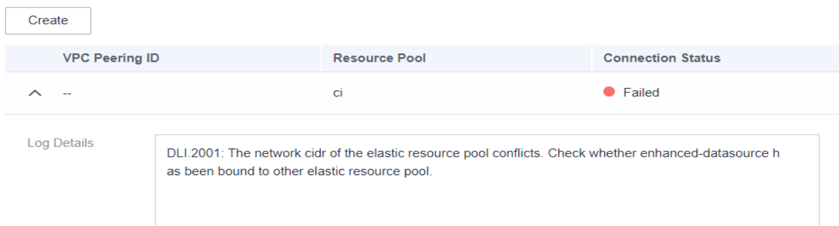
1. Log in to the DLI management console.
2. In the left navigation pane, choose **Datasource Connections**.
3. On the **Enhanced** tab page displayed, bind an enhanced datasource connection to an elastic resource pool:
   a. Locate your desired enhanced datasource connection, click **More** in the **Operation** column, and select **Bind Resource Pool**.
   b. In the **Bind Resource Pool** dialog box, select the resource pool to be bound for **Resource Pool**.
   c. Click **OK**.
4. View the connection status on the **Enhanced** tab page.
   - After an enhanced datasource connection is created, the status is **Active**, but it does not indicate that the queue is connected to the data source. Go to the queue management page to check whether the data source is connected. The procedure is as follows:
     i. In the navigation pane on the left, choose **Resources** > **Queue Management**. On the page displayed, locate a desired queue.
     ii. Click **More** in the **Operation** column and select **Test Address Connectivity**.
     iii. Enter the IP address and port number of the data source.
   - On the details page of an enhanced datasource connection, you can view information about the VPC peering connection.
     - VPC peering ID: ID of the VPC peering connection created in the cluster to which the queue belongs.

       A VPC peering connection is created for each queue bound to an enhanced datasource connection. The VPC peering connection is used for cross-VPC communication. Ensure that the security group used by the data source allows access from the CIDR block of the DLI queue, and do not delete the VPC peering connection during the datasource connection.
     - Status of the VPC peering connection:

       The status of a datasource connection can be **Creating**, **Active**, or **Failed**.

       If the connection status is **Failed**, click on the left to view the detailed error information.

Figure 5-59 Viewing details



5.3.5.4 Unbinding an Enhanced Datasource Connection from an Elastic Resource Pool

Scenario

Unbind an enhanced datasource connection from an elastic resource pool that does not need to access a data source through an enhanced datasoruce connection.

Constraints

If the status of the VPC peering connection created for binding an enhanced datasource connection to an elastic resource pool is **Failed**, the elastic resource pool cannot be unbound.

Procedure

1. Log in to the DLI management console.
2. In the left navigation pane, choose **Datasource Connections**.
3. On the **Enhanced** tab page displayed, use either of the following methods to unbind an enhanced datasource connection from an elastic resource pool:
   - Method 1:
     i. Locate your desired enhanced datasource connection, click **More** in the **Operation** column, and select **Unbind Resource Pool**.
     ii. In the **Unbind Resource Pool** dialog box, select the resource pool to be unbound for **Resource Pool**.
     iii. Click **OK**.
   - Method 2:
     i. Click your desired enhanced datasource connection in the list.
     ii. Locate your desired resource pool and click **Unbind Resource Pool** in the **Operation** column.
     iii. Click **OK**.

5.3.5.5 Adding a Route for an Enhanced Datasource Connection

Scenario

A route is configured with the destination, next hop type, and next hop to determine where the network traffic is directed. Routes are classified into system routes and custom routes.

After an enhanced connection is created, the subnet is automatically associated with the default route. You can add custom routes as needed to forward traffic destined for the destination to the specified next hop.

NOTE

- When an enhanced connection is created, the associated route table is the one associated with the subnet of the data source.
- The route to be added in the **Add Route** dialog box must be one in the route table associated with the subnet of the resource pool.
- The subnet of the data source must be different from that used by the resource pool. Otherwise, a network segment conflict occurs.

Procedure

1. Log in to the DLI management console.
2. In the left navigation pane, choose **Datasource Connections**.
3. On the **Enhanced** tab page displayed, locate the row containing the enhanced connection to which a route needs to be added, and add the route.
   - Method 1:
     i. On the **Enhanced** tab page displayed, locate the enhanced datasource connection to which a route needs to be added and click **Manage Route** in the **Operation** column.
     ii. Click **Add Route**.
     iii. In the **Add Route** dialog box, enter the route information. For details about the parameters, see **Table 5-38**.
     iv. Click **OK**.
   - Method 2:
     i. On the **Enhanced** tab page displayed, locate the enhanced datasource connection to which a route needs to be added, click **More** in the **Operation** column, and select **Add Route**.
     ii. In the **Add Route** dialog box, enter the route information. For details about the parameters, see **Table 5-38**.
     iii. Click **OK**.

Table 5-38 Parameters for adding a custom route

Parameter	Description
Route Name	Name of a custom route, which is unique in the same enhanced datasource scenario. The name can contain 1 to 64 characters. Only digits, letters, underscores (_), and hyphens (-) are allowed.
IP Address	Custom route CIDR block. The CIDR block of different routes can overlap but cannot be the same. Do not add the CIDR blocks **100.125.***xx.xx* and **100.64.***xx.xx* to prevent conflicts with the internal CIDR blocks of services such as SWR. This can lead to failure of the enhanced datasource connection.

4. After adding a route, you can view the route information on the route details page.

5.3.5.6 Deleting the Route for an Enhanced Datasource Connection

Scenario

Delete a route that is no longer used.

Constraints

A custom route table cannot be deleted if it is associated with a subnet.

Procedure

1. Log in to the DLI management console.
2. In the left navigation pane, choose **Datasource Connections**.
3. On the **Enhanced** tab page displayed, locate the row containing the enhanced connection from which the route needs to be deleted, and delete the route.
   - Method 1:
     i. On the **Enhanced** tab page displayed, locate the enhanced connection from which the route needs to be deleted and click **Manage Route** in the **Operation** column.
     ii. Locate the route to be deleted and click **Delete** in the **Operation** column.
     iii. In the dialog box displayed, click **OK**.
   - Method 2:
     i. On the **Enhanced** tab page displayed, locate the enhanced connection from which the route needs to be deleted, click **More** in the **Operation** column, and select **Delete Route**.
     ii. In the **Delete Route** dialog box displayed, confirm the route information.
     iii. Click **Yes**.

5.3.5.7 Modifying Host Information in an Elastic Resource Pool

Scenario

Host information is the mapping between host IP addresses and domain names. After you configure host information, jobs can only use the configured domain names to access corresponding hosts. After a datasource connection is created, you can modify the host information.

When accessing the HBase cluster of MRS, you need to configure the host name (domain name) and IP address of the instance.

Constraints

You have obtained the MRS host information by referring to **How Do I Obtain MRS Host Information?**

Modifying Host Information

1. Log in to the DLI management console.
2. In the left navigation pane, choose **Datasource Connections**.
3. On the **Enhanced** tab page displayed, locate the enhanced datasource connection to be modified, click **More** in the **Operation** column, and select **Modify Host**.
4. In the **Modify Host** dialog box displayed, enter the obtained host information.

   Enter host information in the format of *Host IP address Host name*. Information about multiple hosts is separated by line breaks.

   Example:

   192.168.0.22 node-masterxxx1.com

   192.168.0.23 node-masterxxx2.com

   Obtain the MRS host information by referring to **How Do I Obtain MRS Host Information?**
5. Click **OK**.

How Do I Obtain MRS Host Information?

- **Method 1: View MRS host information on the management console.**

  To obtain the host name and IP address of an MRS cluster, for example, MRS 3.*x*, perform the following operations:

  a. Log in to the MRS management console.
  b. On the **Active Clusters** page displayed, click your desired cluster to access its details page.
  c. Click the **Components** tab.
  d. Click **ZooKeeper**.
  e. Click the **Instance** tab to view the corresponding service IP addresses. You can select any service IP address.
  f. Modify host information by referring to **Modifying Host Information**.

  NOTE

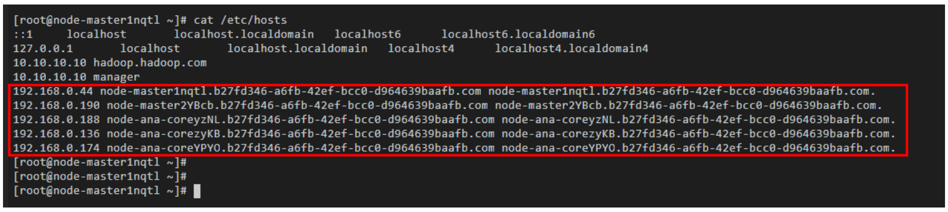
  If the MRS cluster has multiple IP addresses, enter any service IP address when creating a datasource connection.

- **Method 2: Obtain MRS host information from the /etc/hosts file on an MRS node.**

  a. Log in to any MRS node as user **root**.
  b. Run the following command to obtain MRS hosts information. Copy and save the information.

     cat /etc/hosts

Figure 5-60 Obtaining hosts information



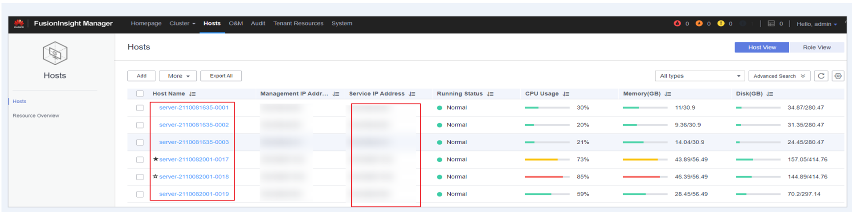
c. Modify host information by referring to **Modifying Host Information**.

- **Method 3: Log in to FusionInsight Manager to obtain host information.**

  a. Log in to FusionInsight Manager.

  b. On FusionInsight Manager, click **Hosts**. On the **Hosts** page, obtain the host names and service IP addresses of the MRS hosts.

  Figure 5-61 FusionInsight Manager



  c. Modify host information by referring to **Modifying Host Information**.

5.3.5.8 Enhanced Datasource Connection Tag Management

Scenario

A tag is a key-value pair customized by users and used to identify cloud resources. It helps users to classify and search for cloud resources. A tag consists of a tag key and a tag value.

If you use tags in other cloud services, you are advised to create the same tag key-value pairs for cloud resources used by the same business to keep consistency.

If your organization has configured tag policies for DLI, add tags to resources based on the policies. If a tag does not comply with the tag policies, resource creation may fail. Contact your organization administrator to learn more about tag policies.

DLI supports the following two types of tags:

- Resource tags: non-global tags created on DLI.
- Predefined tags: global tags created on Tag Management Service (TMS).

  For more information about predefined tags, see **Tag Management Service User Guide**.

DLI allows you to add, modify, or delete tags for datasource connections.

Procedure

1. In the left navigation pane of the DLI management console, choose **Datasource Connections**.

2. In the **Operation** column of the link, choose **More** > **Tags**.
3. The tag management page is displayed, showing the tag information about the current connection.
4. Click **Add/Edit Tag**. The **Add/Edit Tag** dialog is displayed. Add or edit tag keys and values and click **OK**.

Figure 5-62 Adding/Editing a tag

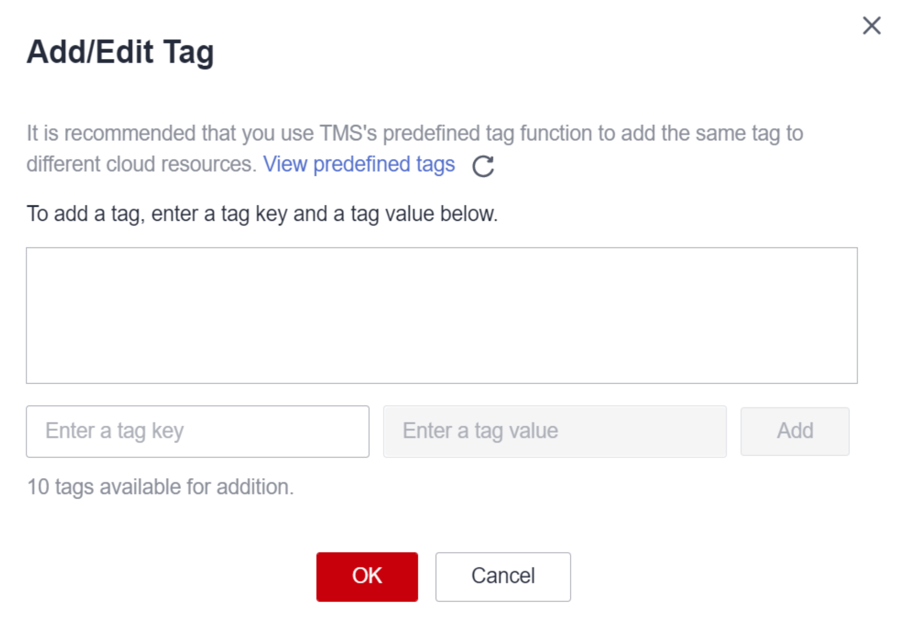


Table 5-39 Tag parameters

Parameter	Description
Tag key	You can perform the following operations: ● Click the text box and select a predefined tag key from the drop-down list. To add a predefined tag, you need to create one on TMS and then select it from the **Tag key** drop-down list. You can click **View predefined tags** to go to the **Predefined Tags** page of the TMS console. Then, click **Create Tag** in the upper corner of the page to create a predefined tag. For details, see **Creating Predefined Tags** in *Tag Management Service User Guide*. ● Enter a tag key in the text box. **NOTE** A tag key can contain a maximum of 128 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed, but the value cannot start or end with a space or start with **_sys_**.

Parameter	Description
Tag value	You can perform the following operations: ● Click the text box and select a predefined tag value from the drop-down list. ● Enter a tag value in the text box. **NOTE** A tag value can contain a maximum of 255 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed. The value cannot start or end with a space.

5. Click **OK**.
6. (Optional) To delete a tag, locate the row where the tag locates in the tag list and click **Delete** in the **Operation** column to delete the tag.

5.3.5.9 Deleting an Enhanced Datasource Connection

Scenario

Delete an enhanced datasource connection that is no longer used on the console.

Procedure

1. Log in to the DLI management console.
2. In the left navigation pane, choose **Datasource Connections**.
3. On the **Enhanced** tab page displayed, locate the enhanced datasource connection to be deleted and click **Delete** in the **Operation** column.
4. In the dialog box displayed, click **Yes**.

5.3.6 Example Typical Scenario: Connecting DLI to a Data Source on a Private Network

Scenario

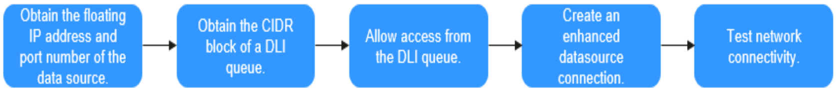
Typically, connecting DLI to a data source on a private network means connecting it to a Huawei Cloud service, such as MRS, RDS, CSS, Kafka, or GaussDB(DWS). DLI's enhanced datasource connections use VPC peering connections to directly connect DLI to VPC networks of destination data sources.

This section describes how to connect DLI to a data source on a private network using an enhanced datasource connection.

If the network is disconnected when an enhanced datasource connection is created, you can rectify the fault based on the overall process and procedure in this section.

Overall Process

Figure 5-63 Configuration process of an enhanced datasource connection



Prerequisites

- You have created a queue. For details about how to create a queue, see **Creating a Queue**.

CAUTION

The queue billing mode must be **Pay-per-use**, and **Dedicated Resource Mode** must be selected after you select a queue type.

Enhanced datasource connections can be created only for pay-per-use resources in dedicated resource mode.

- A cluster of the external data source has been created. You can select a data source as needed.

Table 5-40 Reference for creating clusters of other data sources

Service Name	Reference Documents
RDS	**Buying an RDS for MySQL DB Instance**
GaussDB(DWS)	**Creating a GaussDB(DWS) Cluster**
DMS Kafka	**Creating a Kafka Instance** **CAUTION** When you create the instance, do not enable **Kafka SASL_SSL**.
CSS	**Creating a CSS Cluster**
MRS	**Creating an MRS Cluster**

CAUTION

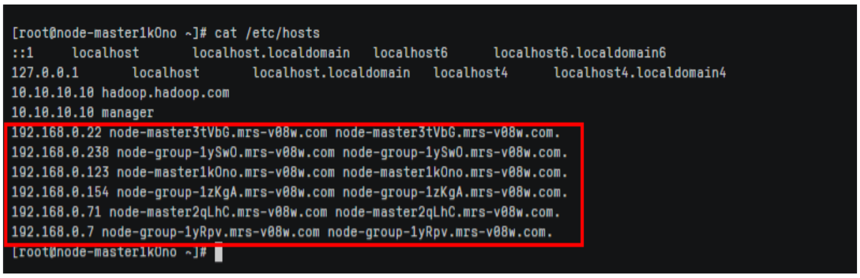
- The CIDR block of the DLI queue bound with a datasource connection cannot overlap with the CIDR block of other data sources.
- Datasource connections cannot be bound with the **default** queue.

Step 1: Obtain the Floating IP Address, Port Number, and Security Group of an External Data Source

Table 5-41 Data source information to be obtained

Data Source	Obtain Method
DMS Kafka	1. On the Kafka management console, click an instance name on the **DMS for Kafka** page. Basic information of the Kafka instance is displayed. 2. In the **Connection** pane, obtain the **Instance Address (Private Network)** value. In the **Network** pane, obtain the VPC and subnet of the instance. 3. In the **Network** pane, obtain the security group of the instance.
RDS	On the **Instances** page of the RDS console, click the target DB instance name. In the displayed page, locate the **Connection Information** pane and obtain the **Floating IP Address**, **VPC**, **Subnet**, **Database Port**, and **Security Group**.
CSS	1. On the CSS management console, choose **Clusters** > **Elasticsearch**. On the displayed page, click the name of the created CSS cluster to view basic information. 2. On the **Cluster Information** page, obtain the **Private Network Address**, **VPC**, **Subnet**, and **Security Group**.
GaussDB(DWS)	1. On the GaussDB(DWS) management console, choose **Clusters**. On the displayed page, click the name of the created GaussDB(DWS) cluster to view basic information. 2. On the **Basic Information** tab, locate the **Database Attributes** pane and obtain the private IP address and port number of the DB instance. In the **Network** pane, obtain the VPC, subnet, and security group information.

<table>
<tr><th>Data Source</th><th>Obtain Method</th></tr>
<tr><td>MRS HBase</td><td>
An MRS 3.x cluster is used as an example.

Log in to the MRS management console, click a cluster name on the Clusters > Active Clusters page to view basic information.
On the dashboard, obtain VPC, subnet, and security group from the Basic Information pane.
The ZooKeeper instance and its port of the MRS cluster are required for creating a job that connects DLI to MRS HBase. You need to obtain the host information of the MRS cluster.
<ol type="a">
Log in to MRS Manager by referring to Accessing FusionInsight Manager. On MRS Manager, choose Cluster > Name of the desired cluster > Services > ZooKeeper. Click the Instance tab and obtain the ZooKeeper host information such as the host name and service IP address.
On MRS Manager, choose Cluster and click the name of the desired cluster. Choose Services > ZooKeeper. Click the Configurations tab and select All Configurations, search for the clientPort parameter, and obtain its value, that is, the ZooKeeper port number.
Log in to any MRS node as user root in SSH mode. For details, see Logging In to an ECS.
Run the following command to obtain MRS hosts information. Copy and save the information.
cat /etc/hosts
An example query result is as follows:


</td></tr>
</table>

Step 2: Obtain the CIDR Block of the DLI Queue

On the DLI management console, choose **Resources** > **Queue Management** from the navigation pane. Locate the queue you have created, and click  next to the queue name to view the CIDR block of the queue.

Step 3: Add a Rule to the Security Group of the External Data Source to Allow Access from the DLI Queue

1. Log in to the VPC console.
2. In the navigation pane on the left, choose **Access Control** > **Security Groups**.
3. Click the name of the security group to which the external data source belongs.

   Obtain the security group name of the data source on the management console of the data source by referring to **Step 1: Obtain the Floating IP Address, Port Number, and Security Group of an External Data Source**.
4. On the **Inbound Rules** tab, add a rule to allow access from the queue network segment.

   Set the inbound rule parameters based on **Table 5-42**.

Figure 5-64 Adding an inbound rule

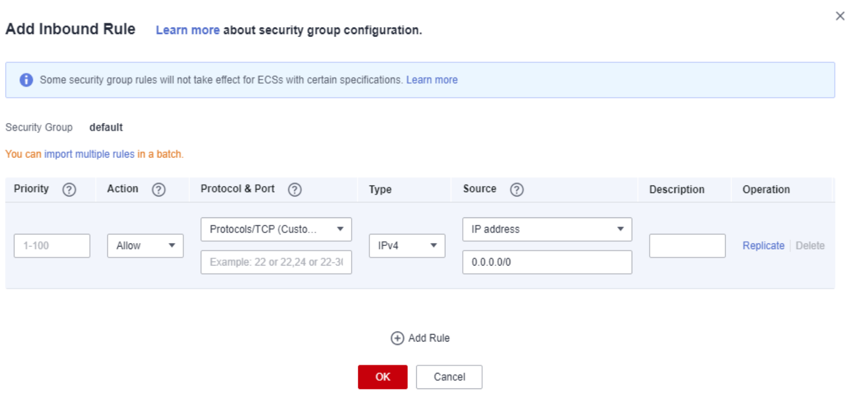


Table 5-42 Inbound rule parameters

Parameter	Description	Example
Priority	The security group rule priority. The priority value ranges from 1 to 100. The default value is **1**, indicating the highest priority. A smaller value indicates a higher priority of a security group rule.	1
Action	Action of the security group rule.	Select **Allow**.

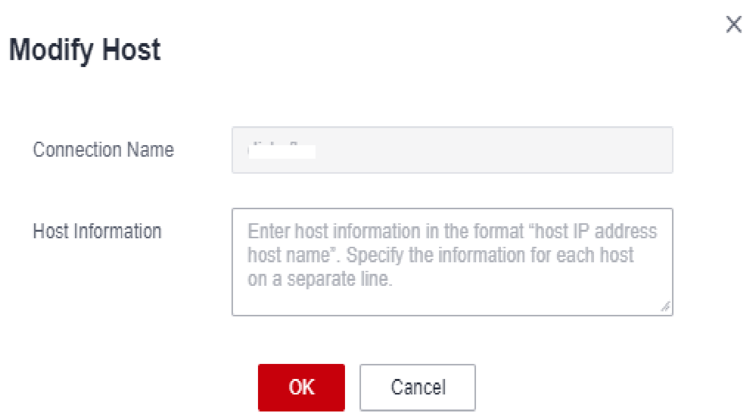
Parameter	Description	Example
Protocol &Port	• Network protocol: The value can be **All**, **TCP**, **UDP**, **ICMP**, or **GRE**. • Port: Port or port range over which the traffic can reach your instance. The port ranges from 1 to 65535.	In this example, select **TCP**. Leave the port blank or set it to the data source port obtained in **Step 1: Obtain the Floating IP Address, Port Number, and Security Group of an External Data Source**.
Type	Type of IP addresses.	IPv4
Source	Allow access from IP addresses or instances in another security group.	In this example, enter the queue CIDR block obtained in **Step 2: Obtain the CIDR Block of the DLI Queue**.
Description	Supplementary information about the security group rule. This parameter is optional.	–

Step 4: Create an Enhanced Datasource Connection

1. Log in to the DLI management console. In the navigation pane on the left, choose **Datasource Connections**. On the displayed page, click **Create** in the **Enhanced** tab.
2. In the displayed dialog box, set the following parameters:
   – **Connection Name**: Name of the enhanced datasource connection
   – **Resource Pool**: Select the target DLI queue. (Queues that are not added to a resource pool are displayed in this list.)
   – VPC: VPC of the data source obtained in **Step 1: Obtain the Floating IP Address, Port Number, and Security Group of an External Data Source**
   – Subnet: Subnet of the data source obtained in **Step 1: Obtain the Floating IP Address, Port Number, and Security Group of an External Data Source**
   – Set other parameters as you need.
3. Click **OK**. Click the name of the created datasource connection to view its status. You can perform subsequent steps only after the connection status changes to **Active**.
4. To connect to MRS HBase, you need to add MRS host information. The procedure is as follows:
   a. On the **Datasource Connections** page, click the **Enhanced** tab and locate the row that contains the created enhanced datasource connection. Click **More** > **Modify Host** in the **Operation** column.
   b. In the dialog box that appears, enter the MRS HBase host information obtained in **Step 1: Obtain the Floating IP Address, Port Number, and**

Security Group of an External Data Source to the **Host Information** box.

Figure 5-65 Modifying host information



c. Click **OK**.

Step 5: Test Network Connectivity

1. Choose **Resources** > **Queue Management** from the left navigation pane, locate the target queue. In the **Operation** column, click **More** > **Test Address Connectivity**.
2. In the displayed dialog box, enter the IP address and port number of the data source obtained in **Step 1: Obtain the Floating IP Address, Port Number, and Security Group of an External Data Source** in the address box and click **Test**. If the queue passes the test, it can access the data source.

   NOTE

   For MRS HBase, use **ZooKeeper IP address**:**ZooKeeper port** or **ZooKeeper host information**:**ZooKeeper port** for the test.

5.3.7 Example Typical Scenario: Connecting DLI to a Data Source on a Public Network

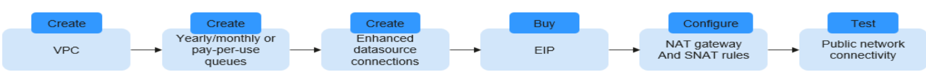
Scenario

A public network data source is one that is accessible over the Internet and has a public IP address. By connecting DLI to the public network, you can access these data sources.

This section explains how to connect DLI to a public network by setting up SNAT rules and configuring routing information.

Procedure

Figure 5-66 Configuration process



Step 1: Create a VPC

Log in to the VPC console and create a VPC. The created VPC is used for NAT to access the public network.

For details about how to create a VPC, see **Creating a VPC**.

Figure 5-67 Creating a VPC

Step 2: Create an Elastic Resource Pool and Add Queues Within It

1. Log in to the DLI management console.
2. In the navigation pane on the left, choose **Resources** > **Resource Pool**.
3. On the displayed page, click **Buy Resource Pool** in the upper right corner.
4. On the displayed page, set the parameters.

Table 5-43 Parameter descriptions

Parameter	Description
Region	Select a region where you want to buy the elastic resource pool.
Project	Project uniquely preset by the system for each region
Name	Name of the elastic resource pool
Specifications	Specifications of the elastic resource pool
CU Range	The maximum and minimum CUs allowed for the elastic resource pool
CIDR Block	CIDR block the elastic resource pool belongs to. If you use an enhanced datasource connection, this CIDR block cannot overlap that of the data source. **Once set, this CIDR block cannot be changed.**
Enterprise Project	Select an enterprise project for the elastic resource pool.

5. Click **Buy**.
6. Click **Submit**.
7. In the elastic resource pool list, locate the pool you just created and click **Add Queue** in the **Operation** column.

8. Set the basic parameters listed below.

Table 5-44 Basic parameters for adding a queue

Parameter	Description
Name	Name of the queue to add
Type	Type of the queue ● To execute SQL jobs, select **For SQL**. ● To execute Flink or Spark jobs, select **For general purpose**.
Engine	SQL queue engine. The options are **Spark** and **HetuEngine**.
Enterprise Project	Select an enterprise project for the elastic resource pool.

9. Click **Next** and configure scaling policies for the queue.

   Click **Create** to add a scaling policy with varying priority, period, minimum CUs, and maximum CUs.

   Figure 5-68 shows the scaling policy configured in this example.

Figure 5-68 Configuring a scaling policy when adding a queue

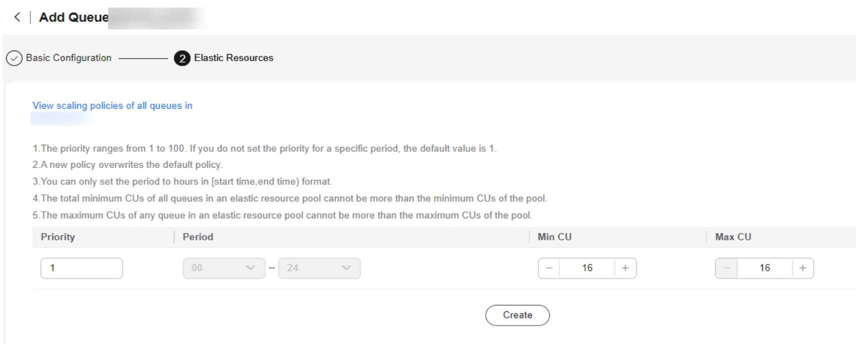


Table 5-45 Scaling policy parameters

Parameter	Description	Example Value
Priority	Priority of the scaling policy in the current elastic resource pool. A larger value indicates a higher priority. In this example, only one scaling policy is configured, so its priority is set to **1** by default.	1

Parameter	Description	Example Value
Period	The first scaling policy is the default policy, and its **Period** parameter configuration cannot be deleted or modified. The period for the scaling policy is from 00 to 24.	00-24
Min CU	Minimum number of CUs allowed by the scaling policy	16
Max CU	Maximum number of CUs allowed by the scaling policy	64

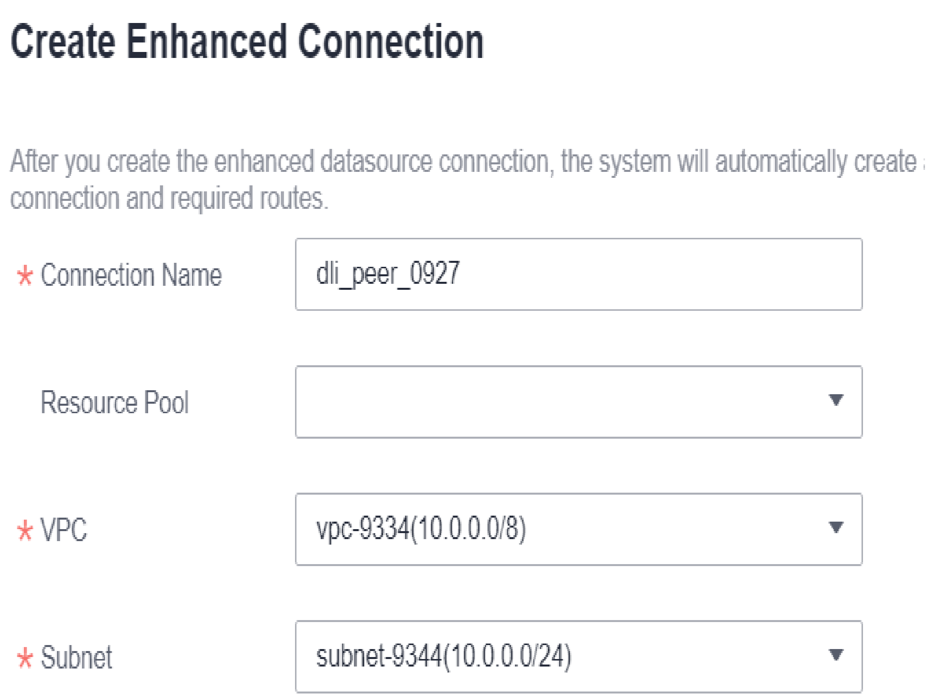
10. Click **OK**.

Step 3: Create an Enhanced Datasource Connection Between the Queue and a VPC

1. In the navigation pane of the DLI management console, choose **Datasource Connections**.
2. In the **Enhanced** tab, click **Create**.

   Enter the connection name, select the created queue, VPC, and subnet, and enter the host information (optional).

   Figure 5-69 Creating an enhanced datasource connection



Step 4: Buy an EIP

1. Log in to the **EIPs** page of the network console, click **Buy EIP**.
2. In the displayed page, configure the parameters as required.

   For details about how to set the parameters, see **Buy EIP**.

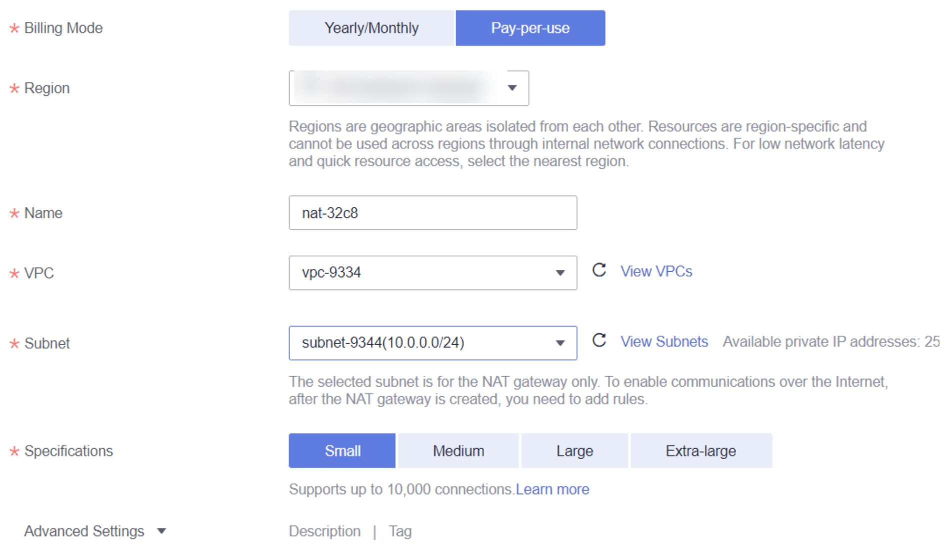
Step 5: Configure a NAT Gateway

Step 1 Create a NAT gateway.

1. Log in to the console and search for **NAT Gateway** in the Service List. The **Public NAT Gateways** page of the network console is displayed.
2. Click **Buy Public NAT Gateway** and configure the required parameters.

   For details, see **Buying a Public NAT Gateway**.

   Figure 5-70 Buying a NAT gateway

3. Click **Next**, confirm the configurations, and click **Submit**.

   NOTE

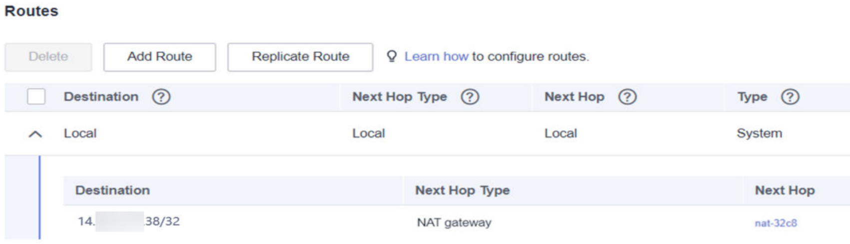
   During the configuration, set **VPC** to the one created in **Step 1: Create a VPC**.

Step 2 Add a route.

In the navigation pane on the left of the network console, choose Virtual **Private Cloud** > **Route Tables**. After a NAT gateway instance is created, a route to that gateway is automatically created. Click the route table name to view the automatically created route.

The destination address is the public IP address you want to access, and the next hop is the NAT gateway.

Figure 5-71 Viewing the route



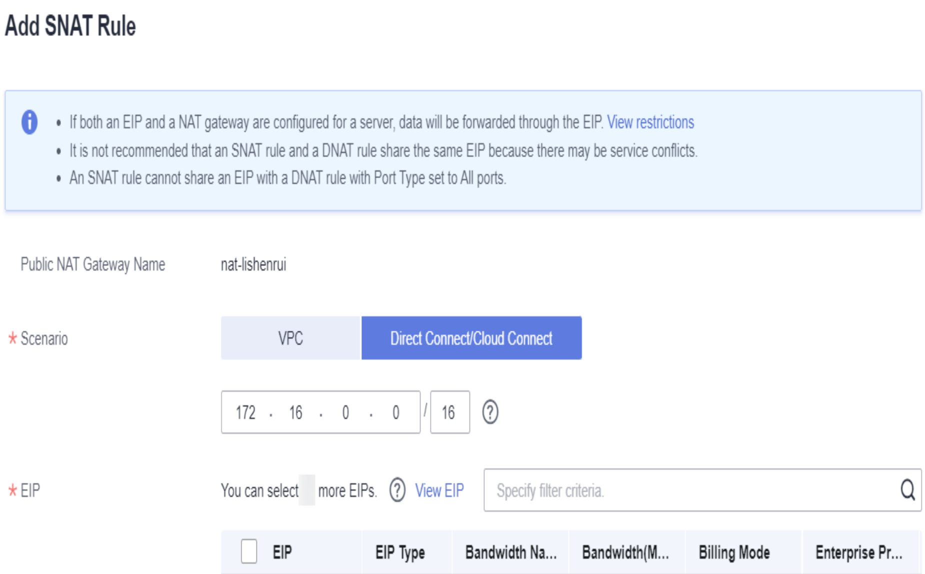
Step 3 Add an SNAT rule.

You need to add SNAT rules for the new NAT gateway to allow the hosts in the subnet to communicate with the Internet.

1. Click the name of the created NAT gateway on the **Public NAT Gateways** page of the network console.
2. On the **SNAT Rules** tab, click **Add SNAT Rule**.

   For details, see **Adding an SNAT Rule**.
3. **Scenario**: Select **Direct Connect/Cloud Connect**.
4. **Subnet**: Select the subnet where the queue you want to connect locates.
5. **EIP**: Select the target EIP.

Figure 5-72 Adding an SNAT rule



6. Click **OK**.

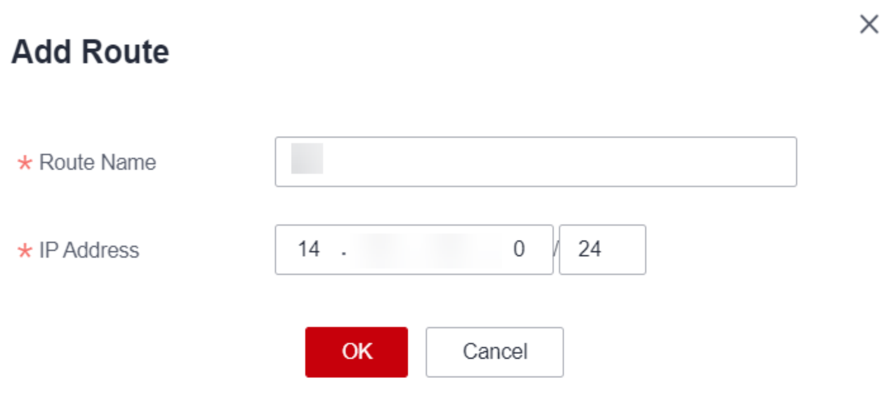
----End

Step 6: Add a Custom Route

Add a custom route for the enhanced datasource connection you have created. Specify the route information of the IP address you want to access.

For details, see **Custom Route Information**.

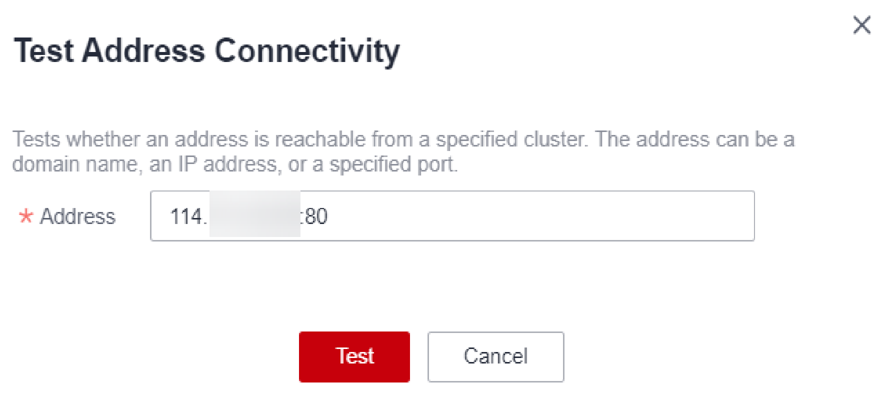
Figure 5-73 Adding route information for test



Step 7: Test the Connectivity to the Public Network

Test the connectivity between the queue and the public network. Click **More** > **Test Address Connectivity** in the **Operation** column of the target queue and enter the public IP address you want to access.

Figure 5-74 Testing address connectivity



6 Configuring an Agency to Allow DLI to Access Other Cloud Services

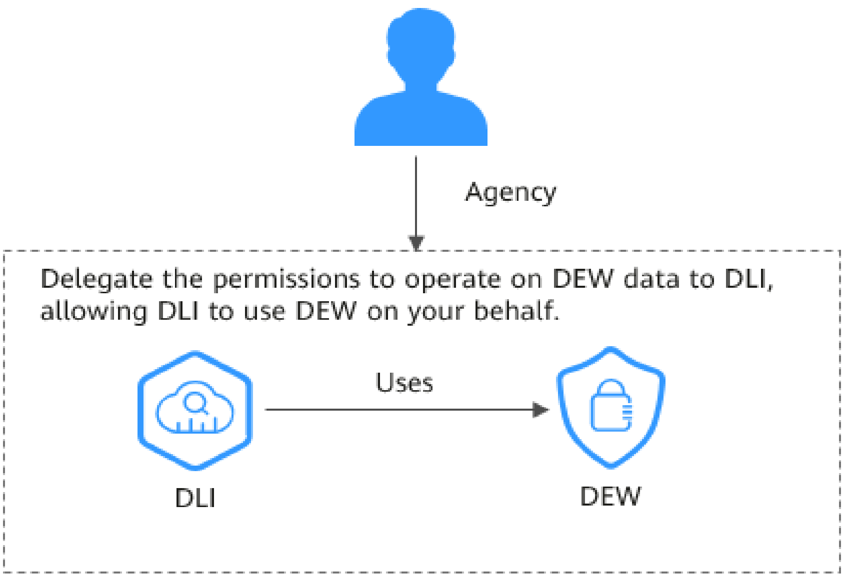
6.1 DLI Agency Overview

What Is an Agency?

Cloud services often interact with each other, with some of which dependent on other services. You can create an agency to delegate DLI to use other cloud services and perform resource O&M on your behalf.

For example, the AK/SK required by DLI Flink jobs is stored in DEW. To allow DLI to access DEW data during job execution, you need to provide an IAM agency to delegate the permissions to perform operations on DEW data to DLI.

Figure 6-1 DLI service agency



DLI Agencies

Before using DLI, you are advised to set up DLI agency permissions to ensure the proper functioning of DLI.

- By default, DLI provides the following agencies: **dli_admin_agency**, **dli_management_agency**, and **dli_data_clean_agency**. The names of these agencies are fixed, but the permissions contained in them can be customized. In other scenarios, you need to create custom agencies. For details about the agencies, see **Table 6-1**.
- DLI upgrades **dli_admin_agency** to **dli_management_agency** to meet the demand for fine-grained agency permissions management. The new agency has the necessary permissions for datasource operations, notifications, and user authorization operations. For details, see **Configuring DLI Agency Permissions**.
- To use Flink 1.15, Spark 3.3.1 (Spark general queue scenario), or a later version to execute jobs, perform the following operations:

  Create an agency on the IAM console and add the agency information to the job configuration. For details, see **Creating a Custom DLI Agency**.

  - Common scenarios for creating an agency: DLI is allowed to read and write data from and to OBS, dump logs, and read and write Flink checkpoints. DLI is allowed to access DEW to obtain data access credentials and access catalogs to obtain metadata.
  - You cannot use the default agency names **dli_admin_agency**, **dli_management_agency**, or **dli_data_clean_agency**. It must be unique.
- If the engine version is earlier than Flink 1.15, **dli_admin_agency** is used by default during job execution. If the engine version is earlier than Spark 3.3.1, user authentication information (AK/SK and security token) is used during job execution.

  This means that jobs whose engine versions are earlier than Flink 1.15 or Spark 3.3.1 are not affected by the update of agency permissions and do not require custom agencies.
- To maintain compatibility with existing job agency permission requirements, **dli_admin_agency** will still be listed in the IAM agency list even after the update.

NOTE

- Only the tenant account or a member account of user group **admin** can authorize the service.
- Do not delete the agency created by the system by default.

Table 6-1 DLI agencies

Permission	Type	Description
dli_admin_agency	Default agency	This agency has been discarded and is not recommended. Upgrade the agency to **dli_management_agency** as soon as possible. For details about how to update an agency, see **Configuring DLI Agency Permissions**.
dli_management_agency	Default agency	DLI system agency, which is used to delegate operation permissions to DLI so that DLI can use other cloud services and perform resource O&M operations on your behalf. This agency grants permissions for datasource operations, message notifications, and user authorization operations. For details about the permissions of an agency, see **Table 6-2**.
dli_data_clean_agency	Default agency, which needs to be authorized by users	Data cleanup agency, which is used to clean up data according to the lifecycle of a table and clean up lakehouse table data. You need to create a DLI agency named **dli_data_clean_agency** on IAM and grant permissions to it. You need to create an agency and customize permissions for it. However, the agency name is fixed to **dli_data_clean_agency**. For details about the permission policies of an agency, see **Agency Permission Policies in Common Scenarios**.

Permission	Type	Description
Other custom agencies	Custom agency	When using Flink 1.15, Spark 3.3, or a later version to execute jobs, create an agency on the IAM console and add new agency information to the job configuration. For details, see **Creating a Custom DLI Agency**. Common scenarios for creating an agency: DLI is allowed to read and write data from and to OBS to transfer logs. DLI is allowed to access DEW to obtain data access credentials and access catalogs to obtain metadata. You cannot use the default agency names **dli_admin_agency**, **dli_management_agency**, or **dli_data_clean_agency**. It must be unique. For details about the permission policies of an agency, see **Agency Permission Policies in Common Scenarios**.

Table 6-2 Permissions contained in the dli_management_agency agency

Policy	Description
IAM ReadOnlyAccess	To authorize IAM users who have not logged in to DLI, you need to obtain their information. So, the permissions contained in the **IAM ReadOnlyAccess** policy are required.
DLI Datasource Connections Agency Access	Permissions to access and use VPCs, subnets, routes, and VPC peering connections
DLI Notification Agency Access	Permissions to send notifications through SMN when a job fails to be executed

6.2 Creating a Custom DLI Agency

When Flink 1.15, Spark 3.3, or a later version is used to execute jobs and the required agency is not included in the DLI system agency **dli_management_agency**, you need to create an agency on the IAM console and add information about the new agency to the job configuration. **dli_management_agency** contains the permissions required for datasource operations, message notifications, and user authorization operations. For other agency permission requirements, you need to create custom DLI agencies. For details about **dli_management_agency**, see **DLI Agency Overview**.

This section walks you through on how to create a custom agency, complete service authorization, and add information about the new agency to the job configuration.

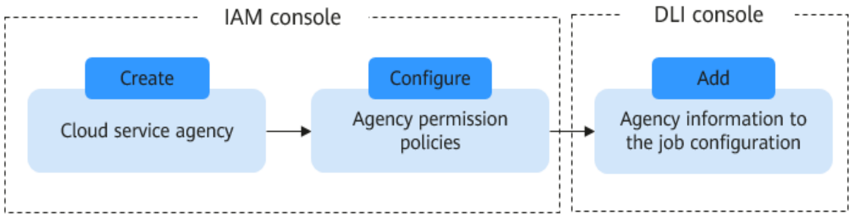
DLI Custom Agency Scenarios

Table 6-3 DLI custom agency scenarios

Scenario	Agency Name	Description	Permission Policy
Allowing DLI to clear data according to the lifecycle of a table	dli_data_clean_agency	Data cleanup agency, which is used to clean up data according to the lifecycle of a table and clean up lakehouse table data. You need to create an agency and customize permissions for it. However, the agency name is fixed to **dli_data_clean_agency**.	**Data Cleanup Agency Permission Configuration**
Allowing DLI to read and write data from and to OBS to transfer logs	Custom	For DLI Flink jobs, the permissions include downloading OBS objects, obtaining OBS/GaussDB(DWS) data sources (foreign tables), transferring logs, using savepoints, and enabling checkpointing. For DLI Spark jobs, the permissions allow downloading OBS objects and reading/writing OBS foreign tables.	**Permission Policies for Accessing and Using OBS**
Allowing DLI to obtain data access credentials by accessing DEW	Custom	DLI jobs use DEW-CSMS' secret management.	**Permission to Use DEW's Encryption Function**
Allowing DLI to access DLI catalogs to retrieve metadata	Custom	DLI accesses catalogs to retrieve metadata.	**Permission to Access DLI Catalog Metadata**
Allowing DLI to access LakeFormation catalogs to retrieve metadata	Custom	DLI accesses LakeFormation catalogs to retrieve metadata.	**Permission to Access LakeFormation Catalog Metadata**

Procedure

Figure 6-2 Process of creating a custom agency



Notes and Constraints

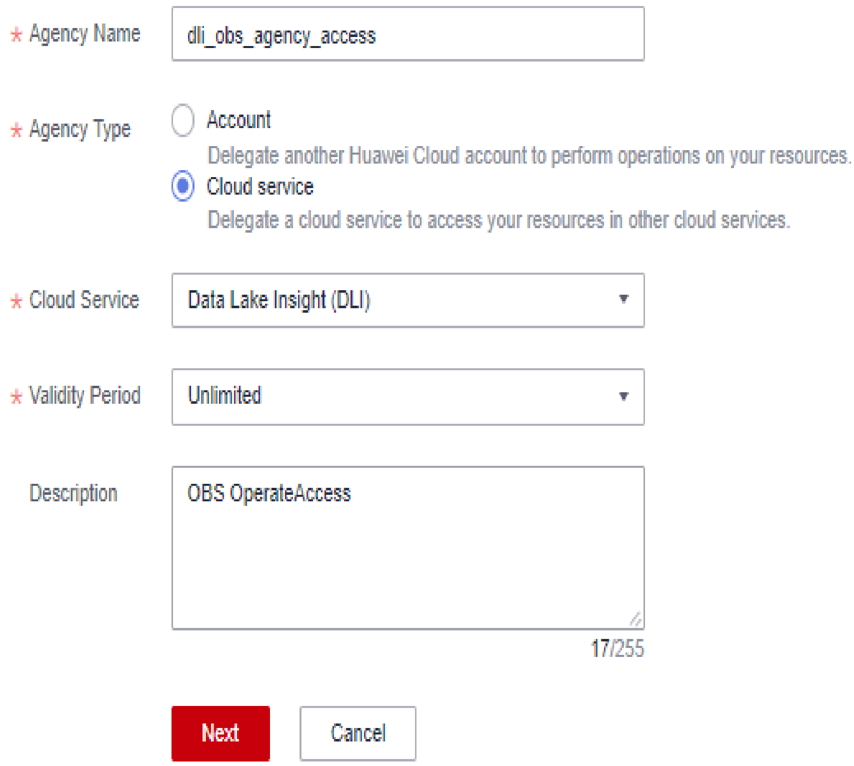
- The custom agency name cannot be the default agency names **dli_admin_agency**, **dli_management_agency**, or **dli_data_clean_agency**. It must be unique.
- The name of the agency that allows DLI to clear data according to the lifecycle of a table must be **dli_data_clean_agency**.
- Custom agencies can be configured only for jobs executed by engines like Flink 1.15, Spark 3.3.1 (Spark general queue scenario), or later versions.
- Once the agency permissions are updated, your **dli_admin_agency** will be upgraded to **dli_management_agency**. This new agency will have the necessary permissions for datasource operations, notifications, and user authorization operations. If you have other agency permission needs, you will need to create custom agencies. For details about **dli_management_agency**, see **DLI Agency Overview**.
- Common scenarios for creating an agency: DLI is allowed to read and write data from and to OBS, dump logs, and read and write Flink checkpoints. DLI is allowed to access DEW to obtain data access credentials and access catalogs to obtain metadata. For details about agency permissions in these scenarios, see **Agency Permission Policies in Common Scenarios**.

Step 1: Create a Cloud Service Agency on the IAM Console and Grant Permissions

1. Log in to the management console.
2. In the upper right corner of the page, hover over the username and select **Identity and Access Management**.
3. In the navigation pane of the IAM console, choose **Agencies**.
4. On the displayed page, click **Create Agency**.
5. On the **Create Agency** page, set the following parameters:
   - **Agency Name**: Enter an agency name, for example, **dli_obs_agency_access**.
   - **Agency Type**: Select **Cloud service**.
   - **Cloud Service**: This parameter is available only when you select **Cloud service** for **Agency Type**. Select **Data Lake Insight (DLI)** from the drop-down list.
   - **Validity Period**: Select **Unlimited**.

– **Description**: You can enter **Agency with OBS OperateAccess permissions**. This parameter is optional.

Figure 6-3 Creating an agency



6. Click **Next**.
7. Click the agency name. On the displayed page, click the **Permissions** tab. Click **Authorize**. On the displayed page, click **Create Policy**.
8. Configure policy information.
   a. Enter a policy name, for example, **dli-obs-agency**.
   b. Select **JSON**.
   c. In the **Policy Content** area, paste a custom policy.

      In this example, the permissions allow access and usage of OBS in various scenarios. For DLI Flink jobs, this includes downloading OBS objects, obtaining OBS/GaussDB(DWS) data sources (foreign tables), transferring logs, using savepoints, and enabling checkpointing. For DLI Spark jobs, the permissions allow downloading OBS objects and reading/writing OBS foreign tables.

      For how to configure common agency permissions for Flink jobs, see **Agency Permission Policies in Common Scenarios**.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "obs:bucket:GetBucketPolicy",
                "obs:bucket:GetLifecycleConfiguration",
                "obs:bucket:GetBucketLocation",
```

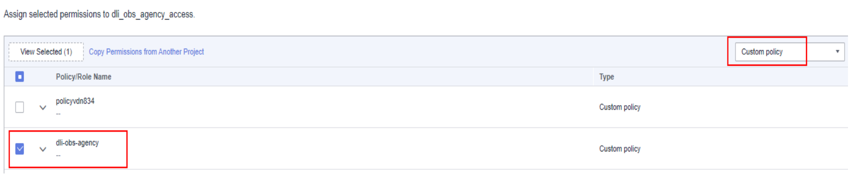
```
                "obs:bucket:ListBucketMultipartUploads",
                "obs:bucket:GetBucketLogging",
                "obs:object:GetObjectVersion",
                "obs:bucket:GetBucketStorage",
                "obs:bucket:GetBucketVersioning",
                "obs:object:GetObject",
                "obs:object:GetObjectVersionAcl",
                "obs:object:DeleteObject",
                "obs:object:ListMultipartUploadParts",
                "obs:bucket:HeadBucket",
                "obs:bucket:GetBucketAcl",
                "obs:bucket:GetBucketStoragePolicy",
                "obs:object:AbortMultipartUpload",
                "obs:object:DeleteObjectVersion",
                "obs:object:GetObjectAcl",
                "obs:bucket:ListBucketVersions",
                "obs:bucket:ListBucket",
                "obs:object:PutObject"
            ],
            "Resource": [
                "OBS:*:*:bucket:bucketName",// Replace bucketName with the actual bucket name.
                "OBS:*:*:object:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "obs:bucket:ListAllMyBuckets"
            ]
        }
    ]
}
```

d. Enter a policy description as required.

9. Click **Next**.

10. On the **Select Policy/Role** page, select **Custom policy** from the first drop-down list and select the custom policy created in **8**.

Figure 6-4 Selecting the created custom policy



11. Click **Next**. On the **Select Scope** page, set the authorization scope.

    For details about authorization operations, see **Creating a User Group and Assigning Permissions**.

    – **All resources**: IAM users will be able to use all resources, including those in enterprise projects, region-specific projects, and global services under your account based on assigned permissions.

    – **Global services**: Global services are deployed in all regions, so you can access them seamlessly without having to switch between regions. However, global services cannot be authorized based on regional projects. They include services like Object Storage Service (OBS) and Content Delivery Network (CDN). After authorization, you can use the selected services based on your permissions.

    – **Region-specific projects**: IAM users will be able to use resources in the selected region-specific projects based on assigned permissions.

   - **Enterprise projects**: IAM users will be able to use resources in the selected enterprise projects based on assigned permissions. For example, an enterprise project may contain resources that are deployed in different regions. After you associate the enterprise project with the IAM users, they can access the resources in this enterprise project based on the assigned permissions.

   In this example, the custom policy is an OBS agency. So, select **Global services**. If a DLI agency is used, you are advised to select **Region-specific projects**.

12. Click **OK**.

   It takes 15 to 30 minutes for the authorization to be in effect.

Step 2: Set Agency Permissions for a Job

When Flink 1.15, Spark 3.3, or a later version is used to execute jobs, you need to add information about the new agency to the job configuration.

Otherwise, If you do not specify an agency for Spark 3.3.1 jobs, the jobs cannot use OBS. If you do not specify an agency for a Flink 1.15 job, checkpointing cannot be enabled, savepoints cannot be used, logs cannot be transferred, and data sources such as OBS and GaussDB(DWS) cannot be used.

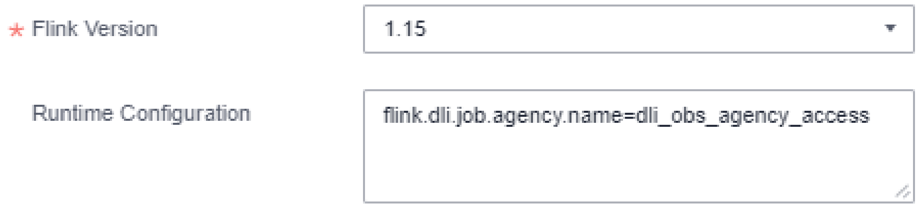
CAUTION

- You can only specify an agency for Flink 1.15 and Spark 3.3.1 jobs running on queues in an elastic resource pool.
- After specifying an agency for a job, be careful when modifying the permissions granted to the agency. Any changes made may impact the job's normal operation.

- **Specifying an agency for a Flink Jar job**
  a. Log in to the DLI console. In the navigation pane, choose **Job Management** > **Flink Jobs**.
  b. Select a desired job and click **Edit** in the **Operation** column.
  c. In the job configuration area on the right, configure agency information.
     - **Flink Version**: Select **1.15**.
     - **Runtime Configuration**: Configure the key-value information of the new agency. The key is fixed to **flink.dli.job.agency.name**, and the value is the custom agency name.

       In this example, set the key-value information to **flink.dli.job.agency.name=dli_obs_agency_access**.

Figure 6-5 Specifying an agency for a Flink Jar job



- **Specifying an agency for a Flink OpenSource SQL job**
  a. Log in to the DLI console. In the navigation pane, choose **Job Management** > **Flink Jobs**.
  b. Select a desired job and click **Edit** in the **Operation** column.
  c. In the job configuration area on the right, configure agency information.
     - On the **Running Parameters** tab, ensure that the selected Flink version is **1.15**.
     - Click **Runtime Configuration**. Configure the key-value information of the new agency. The key is fixed to **flink.dli.job.agency.name**, and the value is the custom agency name.

       In this example, set the key-value information to **flink.dli.job.agency.name=dli_obs_agency_access**.

Figure 6-6 Specifying an agency for a Flink OpenSource SQL job

- **Specifying an agency for a Spark job**
  a. Log in to the DLI console. In the navigation pane, choose **Job Management** > **Spark Jobs**.
  b. Select the target job and click **Edit** in the **Operation** column.
  c. In the job configuration area on the right, configure agency information.
     - Ensure that the selected Spark version is **3.3.1**.
     - In the **Spark Arguments(--conf)** area, configure the key-value information of the new agency. The key is fixed to **spark.dli.job.agency.name**, and the value is the custom agency name.

       In this example, set the key-value information to **spark.dli.job.agency.name=dli_obs_agency_access**.

Figure 6-7 Specifying an agency for a Spark job

6.3 Agency Permission Policies in Common Scenarios

This section provides agency permission policies for common scenarios, which can be used to configure agency permission policies when you customize your permissions. The "Resource" in the agency policy should be replaced according to specific needs.

Data Cleanup Agency Permission Configuration

Application scenario: Data cleanup agency, which is used to clean up data according to the lifecycle of a table and clean up lakehouse table data. You need to create an agency and customize permissions for it. However, the agency name is fixed to **dli_data_clean_agency**.

NOTE

Set the authorization scope of an agency as follows:

- For an OBS agency, select **Global services**.
- For a DLI agency, select **Region-specific projects**.

```
{
   "Version": "1.1",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "obs:object:GetObject",
            "obs:object:DeleteObject",
            "obs:bucket:HeadBucket",
            "obs:bucket:ListBucket",
            "obs:object:PutObject"
         ]
      }
   ]
}
```

```
{
   "Version": "1.1",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dli:table:showPartitions",
                "dli:table:select",
                "dli:table:dropTable",
                "dli:table:alterTableDropPartition"
            ]
        }
    ]
}
```

Permission Policies for Accessing and Using OBS

Application scenario: For DLI Flink jobs, the permissions include downloading OBS objects, obtaining OBS/GaussDB(DWS) data sources (foreign tables), transferring logs, using savepoints, and enabling checkpointing. For DLI Spark jobs, the permissions allow downloading OBS objects and reading/writing OBS foreign tables.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "obs:bucket:GetBucketPolicy",
                "obs:bucket:GetLifecycleConfiguration",
                "obs:bucket:GetBucketLocation",
                "obs:bucket:ListBucketMultipartUploads",
                "obs:bucket:GetBucketLogging",
                "obs:object:GetObjectVersion",
                "obs:bucket:GetBucketStorage",
                "obs:bucket:GetBucketVersioning",
                "obs:object:GetObject",
                "obs:object:GetObjectVersionAcl",
                "obs:object:DeleteObject",
                "obs:object:ListMultipartUploadParts",
                "obs:bucket:HeadBucket",
                "obs:bucket:GetBucketAcl",
                "obs:bucket:GetBucketStoragePolicy",
                "obs:object:AbortMultipartUpload",
                "obs:object:DeleteObjectVersion",
                "obs:object:GetObjectAcl",
                "obs:bucket:ListBucketVersions",
                "obs:bucket:ListBucket",
                "obs:object:PutObject"
            ],
            "Resource": [
                "OBS:*:*:bucket:bucketName",// Replace bucketName with the actual bucket name.
                "OBS:*:*:object:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "obs:bucket:ListAllMyBuckets"
            ]
        }
    ]
}
```

Permission to Use DEW's Encryption Function

Application scenario: DLI Flink and Spark jobs use DEW-CSMS' secret management.


```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "csms:secretVersion:get",
                "csms:secretVersion:list",
                "kms:dek:decrypt"
            ]
        }
    ]
}
```

Permission to Access DLI Catalog Metadata

Application scenario: DLI Flink and Spark jobs are authorized to access DLI metadata.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dli:table:showPartitions",
                "dli:table:alterTableAddPartition",
                "dli:table:alterTableAddColumns",
                "dli:table:alterTableRenamePartition",
                "dli:table:delete",
                "dli:column:select",
                "dli:database:dropFunction",
                "dli:table:insertOverwriteTable",
                "dli:table:describeTable",
                "dli:database:explain",
                "dli:table:insertIntoTable",
                "dli:database:createDatabase",
                "dli:table:alterView",
                "dli:table:showCreateTable",
                "dli:table:alterTableRename",
                "dli:table:compaction",
                "dli:database:displayAllDatabases",
                "dli:database:dropDatabase",
                "dli:table:truncateTable",
                "dli:table:select",
                "dli:table:alterTableDropColumns",
                "dli:table:alterTableSetProperties",
                "dli:database:displayAllTables",
                "dli:database:createFunction",
                "dli:table:alterTableChangeColumn",
                "dli:database:describeFunction",
                "dli:table:showSegments",
                "dli:database:createView",
                "dli:database:createTable",
                "dli:table:showTableProperties",
                "dli:database:showFunctions",
                "dli:database:displayDatabase",
                "dli:table:alterTableRecoverPartition",
                "dli:table:dropTable",
                "dli:table:update",
                "dli:table:alterTableDropPartition"
            ]
        }
    ]
}
```

Permission to Access LakeFormation Catalog Metadata

Application scenario: DLI Spark jobs are authorized to access LakeFormation metadata.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "lakeformation:table:drop",
                "lakeformation:table:create",
                "lakeformation:policy:create",
                "lakeformation:database:create",
                "lakeformation:database:drop",
                "lakeformation:database:describe",
                "lakeformation:catalog:alter",
                "lakeformation:table:alter",
                "lakeformation:database:alter",
                "lakeformation:catalog:create",
                "lakeformation:function:describe",
                "lakeformation:catalog:describe",
                "lakeformation:function:create",
                "lakeformation:table:describe",
                "lakeformation:function:drop",
                "lakeformation:transaction:operate"
            ]
        }
    ]
}
```

6.4 Example of Configuring DLI Agency Permissions in Typical Scenarios

Table 6-4 Guidelines for configuring DLI agency permissions in specific scenarios

Type	Instruction	Description
Flink job	**Flink OpenSource SQL Jobs Using DEW to Manage Access Credentials**	Guideline for using DEW to manage and access credentials for Flink OpenSource SQL jobs. When writing the output data of Flink jobs to MySQL or GaussDB(DWS), set attributes such as the username and password in the connector.
	Flink Jar Jobs Using DEW to Acquire Access Credentials for Reading and Writing Data from and to OBS	Guideline for Flink Jar jobs to acquire an AK/SK to read and write data from and to OBS.

Type	Instruction	Description
	Obtaining Temporary Credentials for Flink Job Agencies	DLI provides a common interface to obtain temporary credentials for Flink job agencies set by users during job launch. The interface encapsulates the obtained temporary credentials for the job agency in the **com.huaweicloud.sdk.core.auth.BasicCredentials** class. Guideline for obtaining a temporary credential for a Flink job agency.
Spark job	**Spark Jar Jobs Using DEW to Acquire Access Credentials for Reading and Writing Data from and to OBS**	Guideline for Spark Jar jobs to acquire an AK/SK to read and write data from and to OBS.
	Obtaining Temporary Credentials for Spark Job Agencies	Guideline for obtaining a temporary credential for a Spark Jar job agency.

7 Submitting a SQL Job Using DLI

7.1 Creating and Submitting a SQL Job

Introduction

The SQL Editor allows you to execute data query operations using SQL statements.

It supports SQL:2003 and is compatible with Spark SQL. For detailed syntax descriptions, refer to **Data Lake Insight SQL Syntax Reference**.

To access the SQL editor, choose **SQL Editor** in the left navigation pane of the DLI console, or click **Create Job** in the upper right corner of the **Job Management** > **SQL Jobs** page.

This section describes how to create and submit a SQL job using the DLI SQL editor.

Notes

- If you access the SQL editor for the first time, the system prompts you to set a bucket for DLI jobs. The created bucket is used to store temporary data generated by DLI, such as job logs.

  You cannot view job logs if you choose not to create the bucket. The bucket name will be set by the system.

  On the OBS console, you can configure lifecycle rules for a bucket to periodically delete objects in it or change object storage classes.

  For details, see **Configuring a Lifecycle Rule**.
- SQL statements can be executed in batches on the SQL editor page.
- Commonly used syntax in the job editing window is highlighted in different colors.
- Both single-line comment and multi-line comment are allowed. Use two consecutive hyphens (--) in each line to comment your statements.

Creating and Submitting a SQL Job Using the SQL Editor

1. Log in to the DLI management console. In the navigation pane on the left, choose **SQL Editor**.

   NOTE

   On the SQL editor page, the system prompts you to create an OBS bucket to store temporary data generated by DLI jobs. In the **Set Job Bucket** dialog box, click **Setting**. On the page displayed, click the edit button in the upper right corner of the job bucket card. In the displayed **Set Job Bucket** dialog box, enter the job bucket path and click **OK**.

2. Above the SQL job editing window, set the parameters required for running a SQL job, such as the queue and database. For how to set the parameters, refer to **Table 7-1**.

Table 7-1 Setting SQL job parameters

Button & Drop-Down List	Description
Engine	SQL jobs support the Spark and HetuEngine engines. ● Spark is suitable for offline analysis. ● HetuEngine is suitable for interactive analysis.
Queues	Select a queue from the drop-down list box. If there is no queue available, the **default** queue is displayed. However, this queue is shared among all users for experience only. This may lead to resource contention, preventing you from obtaining the required resources for your jobs. So, you are advised to create your own queue for executing your jobs. For how to create a queue, see **Creating an Elastic Resource Pool and Creating Queues Within It**. SQL jobs can only be executed on SQL queues.
Catalog	A data catalog is a metadata management object that can contain multiple databases. You can create and manage multiple catalogs in DLI to isolate different metadata.
Databases	Select a database from the drop-down list box. If no database is available, the **default** database is displayed. For how to create a database, see **Creating a Database and Table on the DLI Console**. **NOTE** If you have specified a database where tables are located in SQL statements, the database you choose here does not apply.
Settings	Add parameters and tags. **Parameter Settings**: Set parameters in key/value format for SQL jobs. For details, see the **Data Lake Insight SQL Syntax Reference**. **Tags**: Assign key-value pairs as tags to a SQL job.

3. Create a database and a table.

   Create them in advance by referring to **Creating a Database and Table on the DLI Console**. For example, create a table named **qw**.

4. In the SQL job editing window, enter the following SQL statement:

   ```
   SELECT * FROM qw.qw LIMIT 10;
   ```

   Alternatively, you can double-click the table name **qw**. The query statement is automatically entered in the SQL job editing window.

   DLI offers a range of SQL templates that come with use cases, code examples, and usage guides. You can also use these templates to quickly implement your service logic. For more information about templates, see **Creating a SQL Job Template**.

5. On top of the editing window, click **More** > **Verify Syntax** to check whether the SQL statement is correct.

   a. If the verification fails, check the SQL statement syntax by referring to **Data Lake Insight SQL Syntax Reference**.

   b. If the syntax verification is successful, click **Execute**. Read and agree to the privacy agreement. Click **OK** to execute the SQL statement.

   Once successfully executed, you can check the execution result on the **View Result** tab below the SQL job editing window.

6. View job execution results.

   On the **View Result** tab, click to display execution results in a chart. Click to switch back to the table view.

   You can view a maximum of 1,000 data records on this **View Result** tab. To view more or full data, click to export the data to OBS.

   NOTE

   - If no column of the numeric type is displayed in the execution result, the result cannot be represented in charts.
   - You can view the data in a bar chart, line chart, or fan chart.
   - In the bar chart and line chart, the X axis can be any column, while the Y axis can only be columns of the numeric type. The fan chart displays the corresponding legends and indicators.

Functions of SQL Editor

- **Parameter settings for SQL jobs**

  Click **Settings** in the upper right corner of the **SQL Editor** page. You can set parameters and tags for the SQL job.

  – **Parameter Settings**: Assign key-value pairs as parameter settings.

    For details, see **Data Lake Insight SQL Syntax Reference**.

  – **Tags**: Assign key-value pairs as tags to a SQL job.

Table 7-2 Parameters for SQL job running

Parameter	Default Value	Description
spark.sql.files.maxRecordsPerFile	0	Maximum number of records to be written into a single file. If the value is zero or negative, there is no limit.
spark.sql.autoBroadcastJoinThreshold	209715200	Maximum size, in bytes, of the table that displays all working nodes when a connection is executed. You can set this parameter to **-1** to disable the display. **NOTE** Currently, only configuration units that store tables analyzed using the **ANALYZE TABLE COMPUTE statistics noscan** command and file-based data source tables that calculate statistics directly from data files are supported.
spark.sql.shuffle.partitions	200	Default number of partitions used to filter data for join or aggregation.
spark.sql.dynamicPartitionOverwrite.enabled	false	When set to **false**, DLI will delete all partitions that meet the conditions before overwriting them. For example, if there is a partition named **2021-01** in a partitioned table and you use the **INSERT OVERWRITE** statement to write data to the **2021-02** partition, the data in the **2021-01** partition will also be overwritten. When set to **true**, DLI will not delete partitions in advance, but will overwrite partitions with data written during runtime.
spark.sql.files.maxPartitionBytes	134217728	Maximum number of bytes to be packed into a single partition when a file is read.
spark.sql.badRecordsPath	-	Path of bad records.
dli.sql.sqlasync.enabled	true	Whether DDL and DCL statements are executed asynchronously. The value **true** indicates that asynchronous execution is enabled.
dli.sql.job.timeout	-	Job running timeout interval, in seconds. If the job times out, it will be canceled.

- **Switching to the SparkUI page to view the SQL statement execution process**

The SQL editor allows you to switch to the SparkUI to view the SQL statement execution process.

- You can view only the latest 100 job records on DLI's SparkUI.
- If a job is running on the **default** queue or is a synchronization one, you cannot switch to the SparkUI to view the SQL statement execution process.

NOTE

When you execute a job on a created queue, the cluster is restarted. It takes about 10 minutes. If you click **SparkUI** before the cluster is created, an empty **projectID** will be cached. The SparkUI page cannot be displayed. You are advised to use a dedicated queue so that the cluster will not be released. Alternatively, wait for a while after the job is submitted (the cluster is created), and then check **SparkUI**.

- **Archiving SQL run logs**

  On the **Executed Queries (Last Day)** tab of the **SQL Editor** page, click **More** and select **View Log** in the **Operation** column of the SQL job. The system automatically switches to the OBS path where logs are stored. You can download logs as needed.

  NOTE

  The **View Log** button is not available for synchronization jobs and jobs running on the **default** queue.

- **SQL Editor shortcuts**

Table 7-3 Keyboard shortcuts

Shortcut	Description
Ctrl+Enter	Execute SQL statements. You can run SQL statements by pressing **Ctrl+R** or **Ctrl + Enter** on the keyboard.
Ctrl+F	Search for SQL statements. You can press Ctrl+F to search for a required SQL statement.
Shift+Alt+F	Format SQL statements. You can press **Shift + Alt + F** to format a SQL statement.
Ctrl+Q	Syntax verification. You can press **Ctrl + Q** to verify the syntax of SQL statements.
F11	Full screen. You can press **F11** to display the SQL Job Editor window in full screen. Press **F11** again to leave the full screen.

7.2 Exporting SQL Job Results

Store the data results of analyzed SQL jobs in a specified location in the desired format.

By default, DLI stores SQL job results in its job bucket. You can also download job results to a local host or export job results to a specified OBS bucket.

Exporting Job Results to the DLI Job Bucket

DLI specifies a default OBS bucket for storing job results. You can configure the bucket information on the **Global Configuration** > **Project** page of the DLI management console. Once a job is complete, the system automatically stores its results to this bucket.

The following conditions must be met if you want to read job results from the DLI job bucket:

- You have configured the job bucket on the **Global Configuration** > **Project** page of the DLI management console by referring to **Configuring a DLI Job Bucket**.
- You have submitted a service ticket to request the whitelisting of the feature that allows writing job results to buckets.
- The user who executes jobs has been granted read and write permissions either on the job bucket or on the **jobs/result** path of the job bucket.

  For details, see **Creating a Custom Bucket Policy**.

For how to obtain job results from the DLI job bucket, see "Object Management" > **Downloading Objects** in *Object Storage Service User Guide*.

Exporting Job Results to a Specified Location in Another Bucket

In addition to storing job results in the default bucket, you can also export them to a specified location in another bucket, increasing the flexibility of job result management and making it easier to organize and manage them.

On the console, you can only view a maximum of 1,000 job results. To view additional results, you can export them to an OBS path. The procedure is as follows:

You can export job results on either the **SQL Jobs** or the **SQL Editor** page.

- **SQL Jobs** page: In the navigation pane on the left, choose **Job Management** > **SQL Jobs**. On the displayed page, locate the row containing a desired job, click **More** in the **Operation** column, and select **Export Result**.
- **SQL Editor** page: In the navigation pane on the left, choose **SQL Editor**. On the displayed page, once query statements are successfully executed, click next to the **View Result** tab to export job results.

NOTE

- If there are no numerical columns in the query results, job results cannot be exported.
- Ensure that the user who exports job results has the read and write permissions on the OBS bucket.

Table 7-4 Parameters

<table>
<tr><th>Parameter</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>Data Format</td><td>Yes</td><td>Choose a data format for the job results you want to export. The options include json and csv.</td></tr>
<tr><td>Queues</td><td>Yes</td><td>Select the queue where the job is executed. SQL jobs can only be executed on SQL queues.</td></tr>
<tr><td>Compression Format</td><td>No</td><td>Compression format of the data to be exported. The options are:
● none
● bzip2
● deflate
● gzip</td></tr>
<tr><td>Storage Path</td><td>Yes</td><td>Path in an OBS bucket where the job results are exported
● If Export Mode is set to New OBS directory, then
You need to manually enter a directory name and ensure that the directory name does not exist. Otherwise, the system returns an error message and the export operation cannot be performed.
NOTE
The folder name cannot contain special characters (\/:*?"<>|) and cannot start or end with a period (.).
For example, after selecting the storage path obs://bucket/src1/, you need to manually enter a directory name to change the path to obs://bucket/src1/src2/ and ensure that the src2 directory name does not exist under src1.
The job result export path is obs://bucket/src1/src2/test.csv.
● If Export Mode is set to Existing OBS directory (Overwritten), then
After selecting a bucket path, the job results are exported to that path. If there are files with the same name, they will be automatically overwritten.
For example, if you select obs://bucket/src1/ as the bucket path, then
The job result export path is obs://bucket/src1/test.csv.</td></tr>
</table>

Parameter	Mandatory	Description
Export Mode	Yes	• **New OBS directory** If you select this mode, a new folder path is created and the job results are saved to this path. This mode is used when you need to save exported results to a new location, making it easier to manage and track job results. If you select this option, you must manually enter an export directory in **Storage Path** and ensure that the directory must not exist. If the directory already exists, the system displays an error message and the export operation cannot be performed. • **Existing OBS directory (Overwritten)**: When exporting job results, you can choose an existing file path as the output directory. If there is a file with the same name in that path, it will be automatically overwritten by the new exported job result file. This mode is used when you only need to save a single job result file in the same path, and you do not need to keep old job results.
Number of Results	No	Number of results to be exported If you do not specify or set it to **0**, all results will be exported.
Table Header	No	Whether the job results to be exported contain table headers

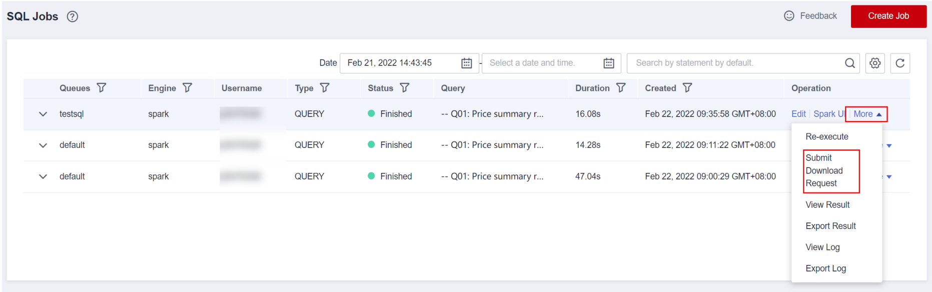
Exporting Job Results to a Local Host

You can download the results of asynchronous DDL and QUERY statements to a local directory. By default, you can download a maximum of 1,000 data records to a local host.

The procedure is as follows:

1. Locate the row containing a desired job whose asynchronous DDL or QUERY statement has been successfully executed, click **More** in the **Operation** column, and select **Submit Download Request**. In the displayed dialog box, click **OK**. After a few seconds, the **Submit Download Request** button would change to **Download**.

Figure 7-1 Selecting Submit Download Request



2. Click **Download** to download the results to your local host.

7.3 Creating a SQL Inspection Rule

What Is SQL Inspection?

There are numerous SQL engines in the big data field, which bring diversity to the solutions but also expose some issues such as varying quality of SQL input statements, difficult SQL problem localization, and excessive resource consumption by large SQL statements.

Poor quality SQL can have unforeseeable impacts on data analysis platforms, affecting system performance or platform stability.

DLI offers this feature to allow you to create inspection rules for the Spark SQL engine. This helps prevent common issues like large or low-quality SQL statements by providing pre-event information, blocking, and in-event circuit breaker. You do not need to change your SQL submission method or syntax, so it is easy to implement without affecting service operations.

- You can configure SQL inspection rules in a visualized manner and also have the ability to query and modify these rules.
- During query execution and response, each SQL engine proactively inspects SQL statements based on the rules.
- Administrators can choose to display hints on, intercept, or block SQL statements. The system logs SQL inspection events in real time for SQL audit. O&M engineers can analyze the logs, assess the quality of SQL statements on the live network, identify potential risks, and take preventive measures.

This section describes how to create a SQL inspection rule to enhance SQL defense capabilities.

Notes and Constraints of DLI SQL Inspection Rules

- SQL inspection is only supported by Spark 3.3.*x* or later.
- Only one inspection rule can be created for an action or a queue.
- Each rule can be associated with a maximum of 50 SQL queues.
- A maximum of 1,000 rules can be created for each project.

Creating a SQL Inspection Rule

You can create SQL inspection rules for specified SQL queues on the **SQL Inspector** page. The system will prompt, block, or perform circuit breaking on SQL requests that trigger the rules.

NOTE

When creating or modifying a SQL inspection rule, evaluate the appropriateness of enabling the rules and setting the threshold based on the service scenario. This will help avoid the negative impact of unreasonable inspection rules blocking or performing circuit breaking on relevant SQL requests on the services.

1. Log in to the DLI management console.
2. In the navigation pane on the left, choose **Global Configuration** > **SQL Inspector**.
3. On the displayed **SQL Inspector** page, click **Create Rule** in the upper right corner. In the **Create Rule** dialog box, set parameters based on the table below.

Table 7-5 Parameters for creating a SQL inspection rule

Parameter	Description
Rule Name	Name of a SQL inspection rule
System Rules	Select an inspection rule. For details about the system inspection rules supported by DLI, see **SQL Inspection System Rules That DLI Supports**.
Queues	Select the queues the rules are bound to.
Description	Enter a rule description.
Rule Action	Actions that the current SQL inspection rule supports. SQL rules support the following types of actions: ● **Info**: Record logs and provide a hint for handling the SQL request. If the rule has parameters, you need to configure the threshold. ● **Block**: Intercept the SQL request that meets the rule. If the rule has parameters, you need to configure the threshold. ● **Circuit Breaker**: Perform circuit breaking on the SQL request that meets the inspection rules. If the rule has parameters, you need to configure the threshold.

4. Click **OK**.

   View the added inspection rule on the **SQL Inspector** page. The rule takes effect dynamically.

   To modify a rule, click **Modify** in its **Operation** column.

SQL Inspection System Rules That DLI Supports

This part describes the system inspection rules supported by DLI. For details, see **Table 7-6**.

- Default system rules are SQL inspection rules automatically created by DLI when a queue is created. These rules are bound to the queue and cannot be deleted.
- Default system rules include **Scan files number**, **Scan partitions number**, **Shuffle data(GB)**, **Count(distinct) occurrences**, and **Not in<Subquery>**.
- Only one inspection rule can be created for an action or a queue.
- For every supported action, a default system rule is created. For example, when a queue is created, a **Scan files number** rule is automatically created for both the **Info** and **Block** actions.
- Different engine versions support different inspection rules.

  To view the engine version of a queue, choose **Resources** > **Queue Management** in the navigation pane on the left, select the queue, double-click the pane at the bottom of the page, and check the value of **Default Version**.

Figure 7-2 Viewing the engine version of a queue

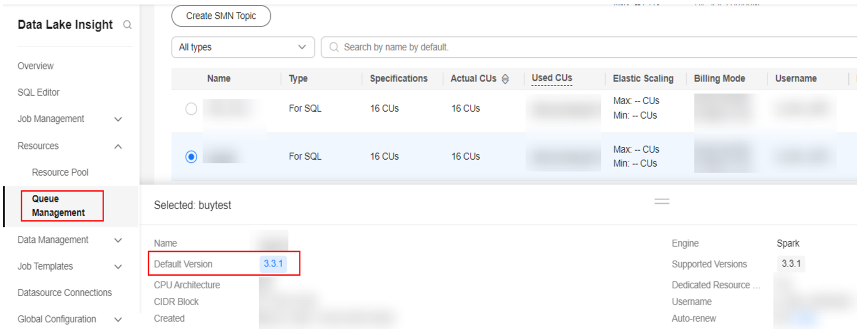


Table 7-6 System inspection rules supported by DLI

Rule ID	Rule Name	Description	Type	Applicable Engine	Action	Value	Default System Rule	Example SQL Statement
dynamic_0001	Scan files number	Maximum number of files to be scanned	Dynamic	Spark Hetu Engine	Info Block	Value range: 1–2000000 Default value: **200000**	Yes	N/A

Rule ID	Rule Name	Description	Type	Applicable Engine	Action	Value	Default System Rule	Example SQL Statement
dynamic_0002	Scan partitions number	Maximum number of partitions involved in the operations (select, delete, update, and alter) that can be performed on a table	Dynamic	Spark	Info Block	Value range: 1–500000 Default value: **5000**	Yes	select * from Partitioned table
running_0002	Memory used(MB)	Peak memory usage of the SQL statement	Running	Spark	Circuit Breaker	Value range: 1–8388608	No	N/A
running_0003	Run time(S)	Maximum running duration of the SQL statement	Running	Spark	Circuit Breaker	Unit: second Value range: 1–43200	No	N/A
running_0004	Scan data(GB)	Maximum amount of data to be scanned	Running	Spark	Circuit Breaker	Unit: GB Value range: 1–10240	No	N/A
running_0005	Shuffle data(GB)	Maximum amount of data to be shuffled	Running	Spark	Circuit Breaker	Unit: GB Value range: 1–10240 Default value: **2048**	Yes	N/A

Rule ID	Rule Name	Description	Type	Applicable Engine	Action	Value	Default System Rule	Example SQL Statement
static_0001	Count(distinct) occurrences	Maximum number of occurrences of count(distinct) in the SQL statement	Static	Spark	Info Block	Value range: 1–100 Default value: **10**	Yes	SELECT COUNT(DISTINCT deviceId), COUNT(DISTINCT collDeviceId) FROM table GROUP BY deviceName, collDeviceName, collCurrentVersion;
static_0002	Not in<Subquery>	Check whether **not in <subquery>** is used in the SQL statement.	Static	Spark	Info Block	Value range: Yes, No Default value: **Yes**	Yes	SELECT * FROM Orders o WHERE Orders.Order_ID not in (Select Order_ID FROM HeldOrders h where h.order_id = o.order_id);

Rule ID	Rule Name	Description	Type	Applicable Engine	Action	Value	Default System Rule	Example SQL Statement
static_0003	Join occurrences	Maximum number of joins in the SQL statement	Static	Spark	Info Block	Value range: 1–50	No	SELECT name, text FROM table_1 JOIN table_2 ON table_1.Id = table_2.Id
static_0004	Union occurrences	Maximum number of union all times in the SQL statement	Static	Spark	Info Block	Value range: 1–100	No	select * from tables t1 union all select * from tables t2 union all select * from tables t3
static_0005	Subquery nesting layers	Maximum number of subquery nesting layers	Static	Spark	Info Block	Value range: 1–20	No	select * from (with temp1 as (select * from tables) select * from temp1);
static_0006	Sql size(KB)	Maximum size of a SQL file	Static	Spark	Info Block	Unit: KB Value range: 1–1024	No	N/A

Rule ID	Rule Name	Description	Type	Applicable Engine	Action	Value	Default System Rule	Example SQL Statement
static_0007	Cartesian product	Limitation of Cartesian products when multiple tables are being associated	Static	Spark	Info Block	Value range: 0–1	No	select * from A,B;

7.4 Setting the Priority for a SQL Job

Scenario

In actual job running, it is necessary to prioritize and ensure the normal running of important and urgent tasks due to their varying levels of importance and urgency. This requires providing the necessary compute resources for their normal operations.

DLI offers a feature to set job priorities for each SQL job, which prioritizes the allocation of compute resources to higher priority jobs when resources are limited.

Notes

- You can assign a priority level of 1 to 10 for each job, with a larger value indicating a higher priority. Compute resources are preferentially allocated to high-priority jobs. That is, if compute resources required for high-priority jobs are insufficient, compute resources for low-priority jobs are reduced.
- Jobs running on a SQL queue have a default priority level of 3.
- To change the priority for a job, you must first stop the job, change the priority level, and then submit the job for the modification to take effect.

Setting the Priority for a SQL Job

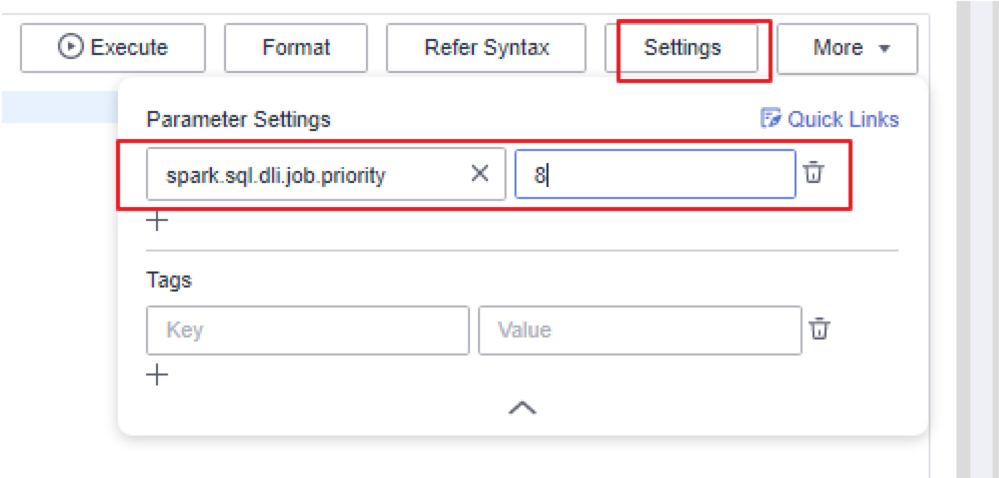
Click **Settings**. In the **Parameter Settings** area, configure the following parameter. *x* indicates the priority value.

```
spark.sql.dli.job.priority=x
```

1. Log in to the DLI management console.
2. In the navigation pane on the left, choose **Job Management** > **SQL Jobs**.
3. Locate the row containing the job for which you want to set the priority and click **Edit** in the **Operation** column.

4. Click **Settings**. In the **Parameter Settings** area, configure the **spark.sql.dli.job.priority** parameter.

Figure 7-3 Example configuration for a SQL job



7.5 Querying Logs for SQL Jobs

Scenario

DLI job buckets are used to store temporary data generated during DLI job running, such as job logs and results.

This section describes how to configure a bucket for DLI jobs on the DLI console and obtain SQL job logs.

Notes

- To avoid disordered job results, do not use the OBS bucket configured for DLI jobs for any other purposes.
- DLI jobs must be set and modified by the main account as IAM users do not have required permissions.
- You cannot view the logs for DLI jobs before configuring a bucket.
- You can configure lifecycle rules to periodically delete objects from buckets or change storage classes of objects.
- Exercise caution when modifying the job bucket, as it may result in the inability to retrieve historical data.

Prerequisites

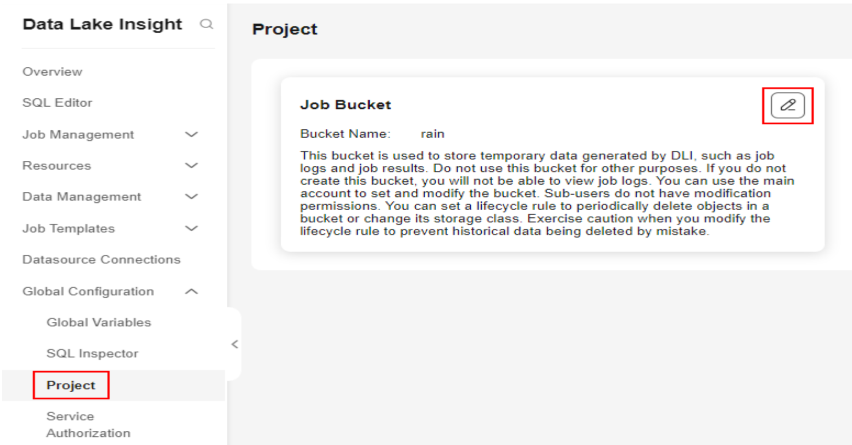
Before the configuration, create an OBS bucket or parallel file system (PFS). In big data scenarios, you are advised to create a PFS. PFS is a high-performance file system provided by OBS, with access latency in milliseconds. PFS can achieve a bandwidth performance of up to TB/s and millions of IOPS, which makes it ideal for processing high-performance computing (HPC) workloads.

For details about PFS, see "Parallel File System Feature Guide" in the *Object Storage Service User Guide*.

Configuring a Bucket for DLI Jobs

1. In the navigation pane of the DLI console, choose **Global Configuration** > **Project**.
2. On the **Project** page, click next to **Job Bucket** to configure bucket information.

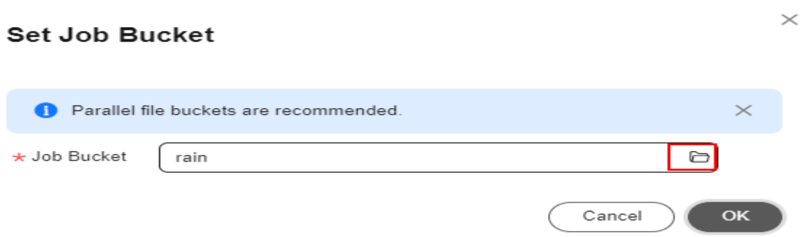
Figure 7-4 Project



3. Click to view available buckets.
4. In the displayed **OBS** dialog box, click the name of a bucket or search for and click a bucket name and then click **OK**. In the **Set Job Bucket** dialog box, click **OK**.

   Temporary data generated during DLI job running will be stored in the OBS bucket.

Figure 7-5 Setting the job bucket

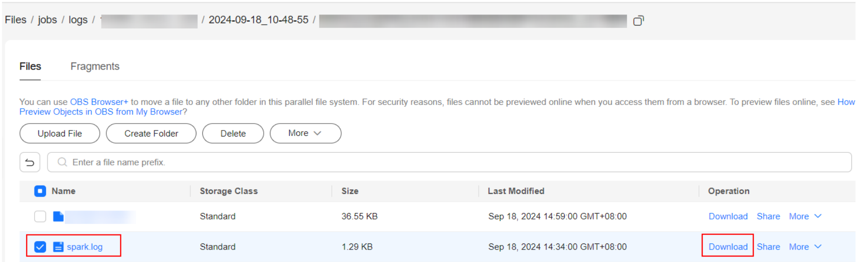


Querying Logs for SQL Jobs

1. Log in to the DLI console. In the navigation pane on the left, choose **Job Management** > **SQL Jobs**.
2. Select the SQL job whose jobs you want to query, click **More** in the **Operation** column, and select **View Log**.

   The system automatically switches to the log path of the DLI job bucket.
3. On the **Files** tab, select the log file of the desired date and time and click **Download** in the **Operation** column to download the file to your local host.

Figure 7-6 Downloading SQL job logs



7.6 Managing SQL Jobs

Viewing Basic Job Information on the SQL Jobs Page

The **SQL Jobs** page displays all SQL jobs, which may span multiple pages if there are many jobs. You can navigate to a specific page as needed. DLI allows you to view jobs in all statuses. Jobs in the job list are sorted by creation time in descending order by default.

Table 7-7 SQL Job management parameters

Parameter	Description
Queues	Name of the queue to which a job belongs
Engine	SQL jobs support the Spark and HetuEngine engines. ● **Spark**: displays jobs whose execution engine is Spark. ● **HetuEngine**: displays jobs whose execution engine is HetuEngine.
Username	Name of the user who executed the job.
Type	Job type. The following types are supported: ● **IMPORT**: A job that imports data to DLI ● **EXPORT**: A job that exports data from DLI ● **DCL**: Conventional DCLs and operations related to queue permissions ● **DDL**:Conventional DDLs, including creating and deleting databases and tables ● **QUERY**: A job that queries data by running SQL statements ● **INSERT**: A job that inserts data by running SQL statements ● **UPDATE**: A job that updates data. ● **DELETE**: A job that deletes a SQL job. ● **DATA_MIGRATION**: A job that migrates data. ● **RESTART_QUEUE**: A job that restarts a queue. ● **SCALE_QUEUE**: A job that changes queue specifications, including sale-out and scale-in.

Parameter	Description
Status	Job status. Possible values are as follows: • **Submitting** • **Running** • **Finished** • **Canceled** • **Failed** • **Scaling**
Query	SQL statements for operations such as exporting and creating tables You can click to copy the query statement.
Duration	Running duration of a job
Created	Time when a job is created. Jobs can be displayed in ascending or descending order of the job creation time.

Parameter	Description
Operation	• **Edit**: Edit the job. • **Cancel** – You can terminate a job only when the job is in **Submitting** or **Running** status. – A job whose status is **Finished**, **Failed**, or **Canceled** cannot be terminated. – If the **Cancel** button is gray, you are not allowed to perform this operation. • **Re-execute**: Execute the job again. • **SparkUI**: Display the Spark job execution page. NOTE • When you execute a job on a created queue, the cluster is restarted. It takes about 10 minutes. If you click **SparkUI** before the cluster is created, an empty **projectID** will be cached. The SparkUI page cannot be displayed. You are advised to use a dedicated queue so that the cluster will not be released. Alternatively, wait for a while after the job is submitted (the cluster is created), and then check **SparkUI**. • SparkUI can currently only display the latest 100 jobs. • In addition to the preceding operations, the following operations are available for QUERY jobs and asynchronous DDL jobs. – **Submit Download Request**: Download the results of asynchronous DDL and QUERY statements to a local directory. For details, see **Exporting Job Results to a Local Host**. – **View Result**: View the job running result. – **Export Result**: Export the job running result to the created OBS bucket. • In addition to the preceding operations, the EXPORT job also includes the following operations: – **Download** • **View Log**: Save job logs to the temporary OBS bucket created by DLI. NOTE The **View Log** button is not available for synchronization jobs and jobs running on the default queue.

Viewing Job Details

On the **SQL Jobs** page, you can click in front of a job record to view details about the job.

Job details vary with job types. The job details vary depending on the job types, status, and configuration options. The following describes how to load data, create a table, and select a job. For details about other job types, see the information on the management console.

- **Load data** (job type: IMPORT) include the following information: queue, job ID, username, type, status, execution statement, running duration, creation time, end time, parameter settings, label, number of results, scanned data, number of scanned data, number of error records, storage path, data format, database, table, table header, separator, reference character, escape character, date format, timestamp format, total CPU used, and output bytes.
- **Create table** (job type: DDL) include the following information: queue, job ID, username, type, status, execution statement, running duration, creation time, end time, parameter settings, tags, number of results, scanned data, and database.
- **Select** (job type: QUERY) include the following information: queue, job ID, username, type, status, execution statement, running duration, creation time, end time, parameter setting, label, number of results (results of successful executions can be exported), and scanned data, username, result status (results of successful tasks can be viewed. Failure causes of failed tasks are displayed), database, total CPU used, and output bytes.

  NOTE

  - **Total CPU Used (Core x ms)**: total CPU used during job execution.
  - **Output Bytes**: number of output bytes after the job is executed.

Searching for a Job

On the **SQL Jobs** page, you can search jobs with any of the following operations.

- Select a queue name.
- Select an engine.
- Set the date range.
- Enter a username, statement, tag, or job ID.
- Select the creation time in ascending or descending order.
- Select a job type.
- Select a job status.
- Select the job execution duration in ascending or descending order.

Terminating a SQL Job

On the **SQL Jobs** page, you can click **Terminate** in the **Operation** column to stop a submitting or running job.

7.7 Creating and Managing SQL Job Templates

7.7.1 Creating a SQL Job Template

To facilitate SQL operation execution, DLI allows you to customize query templates or save the SQL statements in use as templates. After templates are saved, you do not need to compile SQL statements. Instead, you can directly perform the SQL operations using the templates.

SQL templates include sample templates and custom templates. The default sample template contains 22 standard TPC-H query statements, which can meet most TPC-H test requirements. For details, see **TPC-H Sample Data in the SQL Templates Preset on DLI**.

NOTE

In the navigation pane on the left, choose **Job Templates** > **SQL Templates**. In the upper right corner of the displayed page, click **Settings**. In the displayed **Settings** dialog box, choose whether to display templates by group.

If you enable **Display by Group**, the display options are **Expand the first group**, **Expand all**, and **Collapse all**.

Creating a SQL Job Template

You can create a template on either the **Job Templates** or the **SQL Editor** page.

- To create a template on the **Job Templates** page:
  a. On the left of the management console, choose **Job Templates** > **SQL Templates**.
  b. On the **SQL Templates** page, click **Create Template** to create a template.

     Enter the template name, SQL statement, and description information. For details, see **Table 7-8**.

Figure 7-7 Creating a template

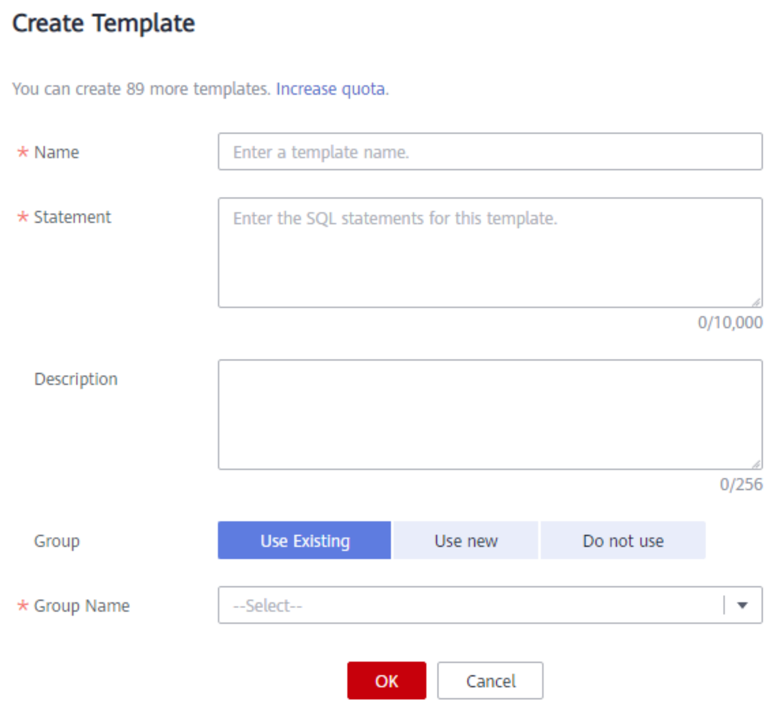


Table 7-8 Parameter description

Parameter	Description
Name	Indicates the template name. ● A template name can contain only digits, letters, and underscores (_), but cannot start with an underscore (_) or contain only digits. It cannot be left empty. ● The template name can contain a maximum of 50 characters.
Statement	SQL statement to be saved as a template.
Description	Description of the template you create.
Group	● Use existing ● Use new ● Do not use
Group Name	If you select **Use existing** or **Use new**, you need to enter the group name.

c. Click **OK**.

- To create a template on the **SQL Editor** page:
  a. On the left of the management console, click **SQL Editor**.
  b. In the SQL job editing area of the displayed **SQL Editor** page, click **More** in the upper right corner and choose **Save as Template**.
     Enter the template name, SQL statement, and description information. For details, see **Table 7-8**.
  c. Click **OK**.

Submitting a SQL Job Using a Template

Perform the template operation as follows:

1. On the left of the management console, choose **Job Templates** > **SQL Templates**.
2. On the **SQL Templates** page, select a template and click **Execute** in the **Operation** column. The **SQL Editor** page is displayed, and the corresponding SQL statement is automatically entered in the SQL job editing window.
3. In the upper right corner of the SQL job editing window, Click **Execute** to run the SQL statement. After the execution is complete, you can view the execution result below the current SQL job editing window.

Searching for a SQL Job Template

On the **SQL Templates** page, you can enter the template name keyword in the search box on the upper right corner to search for the desired template.

Modifying a SQL Job Template

Only custom templates can be modified. To modify a template, perform the following steps:

Step 1 On the **SQL Templates** page, locate the target template and click **Modify** in the **Operation** column.

Step 2 In the displayed **Modify Template** dialog box, modify the template name, statement, and description as required.

Step 3 Click **OK**.

----End

Deleting a Template

On the **SQL Templates** page, select one or more templates to be deleted and click **Delete** to delete the selected templates.

7.7.2 Developing and Submitting a SQL Job Using a SQL Job Template

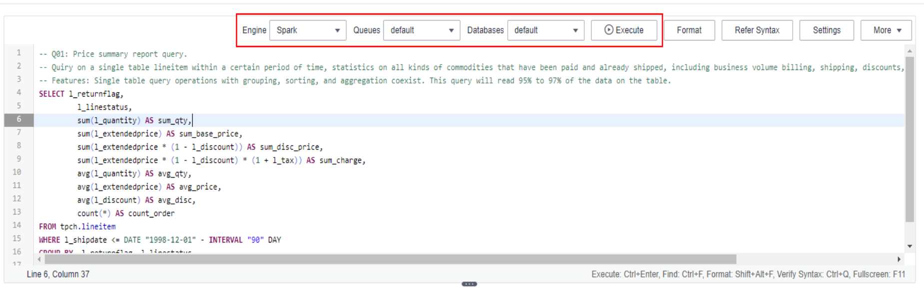
DLI allows you to create custom templates or save currently used SQL statements as templates for quick and convenient SQL operations. Once a template is saved, you can execute SQL operations directly through the template without the need to write SQL statements.

The system offers sample templates that include various standard TPC-H query statements. You can choose to use one of these templates or create a custom template to create a SQL job.

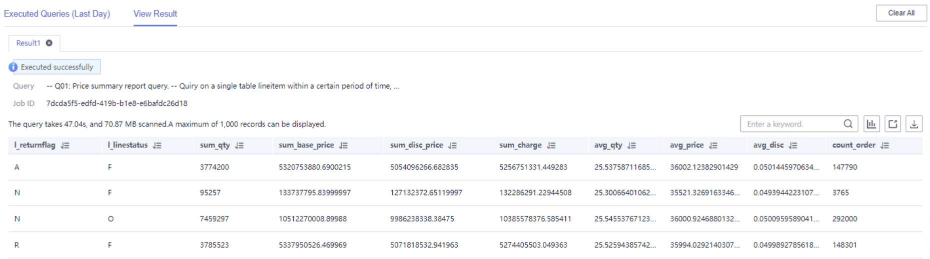
This example shows how to use a TPC-H sample template to develop and submit a SQL job.

Procedure

1. Log in to the DLI management console.
2. In the navigation pane on the left, choose **Job Templates** > **SQL Templates**.
3. On the displayed **Sample Templates** tab, find a sample template that matches your service scenario under **tpchQuery** and click **Execute** in the **Operation** column.
4. In the upper part of the editing window, set **Engine** to **Spark**, **Queues** to **default**, and **Databases** to **default**, and click **Execute**.



5. Check the query result on the **View Result** tab below the editing window.



This example uses the **default** queue and database preset in the system as an example. You can also run the command in a self-created queue and database.

For details, see **Creating a Queue**. For how to create a database, see **Creating a Database**.

7.7.3 TPC-H Sample Data in the SQL Templates Preset on DLI

TPC-H Sample Data

TPC-H is a test set developed by the Transaction Processing Performance Council (TPC) to simulate decision-making support applications. It is widely used in academia and industry to evaluate the performance of decision-making support technology. This business test has higher requirements on vendors, because it can comprehensively evaluate the overall business computing capability. With universal business significance, is widely used in analysis of bank credit, credit card, telecom operation, tax, as well as tobacco industry decision-making analysis.

The TPC-H benchmark test is derived from the TPC-D standard, which was established by the TPC organization in 1994 for decision support system testing. TPC-H implements a 3NF data warehouse that contains eight basic relationships, with a data volume range from 1 GB to 3 TB. The TPC-H benchmark test includes 22 queries (Q1 to Q22). The main evaluation indicator is the response time of each query (from submission to result return). The unit of the TPC-H benchmark test is the query number per hour (QphH@size). **H** indicates the average number of complex queries per hour. **size** indicates the size of database, which reflects the query processing capability of the system. TPC-H can evaluate key performance parameters that other tests cannot evaluate, because it is modeled based on the actual production and operation environment. In a word, the TPC-H standard by TPC meets the test requirements of data warehouse and motivate vendors and research institutes to stretch the limit of this technology.

In this example, DLI directly queries the TPC-H dataset on OBS. DLI has generated a standard TPC-H-2.18 dataset of 100 MB which is uploaded to the tpch folder on OBS. The read-only permission is granted to you to facilitate query operations.

TPC-H Test and Metrics

TPC-H test is divided into three sub-tests: data loading test, Power test, and Throughput test. Data loading indicates the process of setting up a test database, and the loading test is to test the data loading ability of DBMS. The first test is data loading test that tests data loading time, which is time-consuming. The second test is Power test, also called raw query. After data loading test is complete, the database is in the initial state without any other operation, especially the data in the buffer is not tested. Power test requires that the 22 queries be executed once in sequence and a pair of RF1 and RF2 operations be executed at the same time. The third test is Throughput test, the core and most complex test, more similar to the actual application environment. With multiple query statement groups and a pair of RF1 and RF2 update flows, Throughput test pose greater pressure on the SUT system than Power test does.

The basic data in the test is related to the execution time (the time of each data loading step, each query execution, and each update execution), based on which you can calculate the data loading time, Power@Size, Throughput@Size, qphH@Size and $/QphH@Size.

Power@Size is the result of the Power test, which is defined as the reciprocal of the geometric average value of the query time and change time. The formula is as follows:

$$\text{TPC-H Power@Size} = \frac{3600 * SF}{\sqrt[24]{\prod_{i=1}^{i=22} QI(i,0) * \prod_{j=1}^{j=2} RI(j,0)}}$$

Size indicates the data size. SF is the scaling factor of data scale. QI (i, 0) indicates the time of the ith query, in seconds. R (I j, 0) is the update time of RFj, in seconds.

Throughput@Size is the Throughput test result, which is defined as the reciprocal of the average value of all query execution time. The formula is as follows:

$$\text{QphH@Size} = \sqrt{Power\ @\ Size * Throughput\ @\ Size}$$

Service Scenario

You can use the built-in TPC-H test suite of DLI to perform interactive query without uploading data.

Advantages of DLI Built-in TPC-H

- You can log in to DLI and get permission to run SQL statements without creating tables or import data.
- The 22 preset TPC-H SQL query templates with rich functions meet the requirements of most business scenarios. You do not need to download TPC-H query statements, which saves your time and energy.
- Data Lake gives you brand-new experience of serverless DLI product within the minimum time.

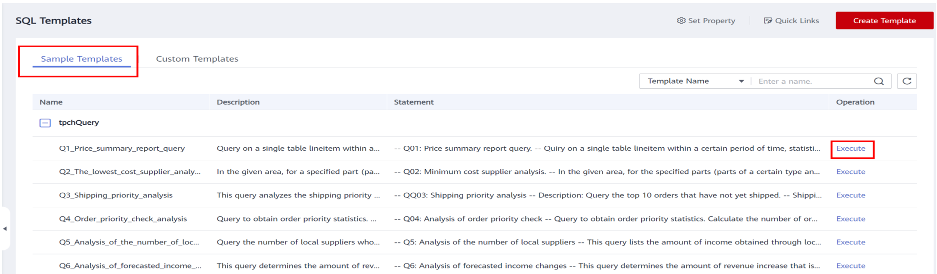
Precautions

When a sub-account uses the TPC-H test suite, the main account needs to grant the sub-account the OBS access permission and the permission to view the main account table. If the master account has not logged in to DLI, the sub-account needs to have the permissions to create databases and tables in addition to the preceding permissions.

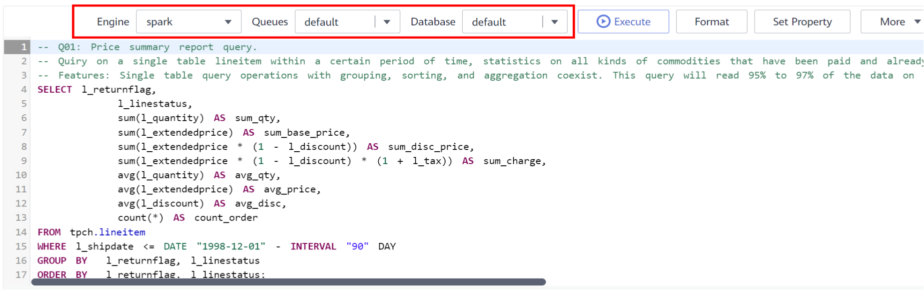
Developing and Submitting a SQL Job Using the TPC-H Sample Template

DLI allows you to create custom templates or save currently used SQL statements as templates for quick and convenient SQL operations. Once a template is saved, you can execute SQL operations directly through the template without the need to write SQL statements.

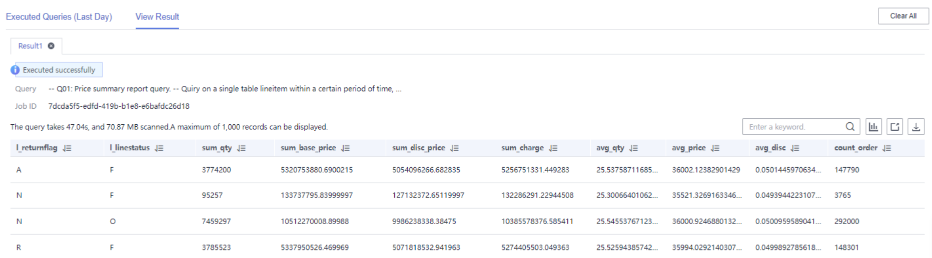
1. Log in to the DLI management console.
2. On the DLI management console, choose **Job Templates** > **SQL Templates** in the navigation pane on the left. On the displayed **Sample Templates** tab, click the plus sign next to **tpchQuery** to find the **Q1_Price_summary_report_query** template. Then, click **Execute** in the **Operation** column.



3. In the upper part of the editing window, set **Engine** to **Spark**, **Queues** to **default**, and **Databases** to **default**, and click **Execute**.



4. Check the query result on the **View Result** tab below the editing window.



This example uses the **default** queue and database preset in the system as an example. You can also run the command in a self-created queue and database.

8 Submitting a Flink Job Using DLI

8.1 Flink Job Overview

DLI supports two types of Flink jobs:

- **Flink OpenSource SQL job:**
  - It is fully compatible with Flink of the community edition, ensuring that jobs can run smoothly on these Flink versions.
  - DLI Flink has expanded the support for connectors based on Flink of the community edition, supporting Redis and GaussDB(DWS) as new data source types. With this expansion, you can now utilize a wider range of data source types, providing greater flexibility and convenience when working with datasets.
  - Flink OpenSource SQL jobs are ideal for scenarios where stream processing logic can be defined and executed through SQL statements. This simplifies stream processing, allowing developers to focus more on implementing service logic.

  For how to create a Flink OpenSource SQL job, see **Creating a Flink OpenSource SQL Job**.
- **Flink Jar job:**
  - DLI allows you to submit Flink jobs compiled into JAR files, providing higher flexibility and customization capabilities. It is applicable to scenarios where complex data processing is required.
  - If the connectors provided by Flink of the community edition cannot meet specific needs, you can use Jar jobs to implement custom connectors or data processing logic.
  - It is ideal for scenarios where user-defined functions (UDFs) or specific library integration are required. You can use the Flink ecosystem to implement advanced stream processing logic and status management.

  For how to create a Flink Jar job, see **Creating a Flink Jar Job**.

8.2 Creating a Flink OpenSource SQL Job

This section describes how to create a Flink OpenSource SQL job.

DLI Flink OpenSource SQL jobs are fully compatible with the syntax of Flink provided by the community. In addition, Redis and GaussDB(DWS) data source types are added based on the community connector. For the syntax and constraints of Flink SQL DDL, DML, and functions, see **Table API & SQL**.

- For the Flink OpenSource SQL 1.15 syntax, see **Flink OpenSource SQL 1.15 Syntax**.
- For the Flink OpenSource SQL 1.12 syntax, see **Flink OpenSource SQL 1.12 Syntax**.

Prerequisites

- You have prepared the source and sink streams.
- A datasource connection has been created to enable the network between the queue where the job is about to run and external data sources.
  - For details about the external data sources that can be accessed by Flink jobs, see **Common Development Methods for DLI Cross-Source Analysis**.
  - For how to create a datasource connection, see **Configuring the Network Connection Between DLI and Data Sources (Enhanced Datasource Connection)**.

    On the **Resources** > **Queue Management** page, locate the queue you have created, click **More** in the **Operation** column, and select **Test Address Connectivity** to check if the network connection between the queue and the data source is normal. For details, see **Testing Address Connectivity**.

Precautions

Before creating jobs and submitting tasks, you are advised to enable CTS to record DLI operations for queries, audits, and tracking. **Using CTS to Audit DLI** lists DLI operations that can be recorded by CTS.

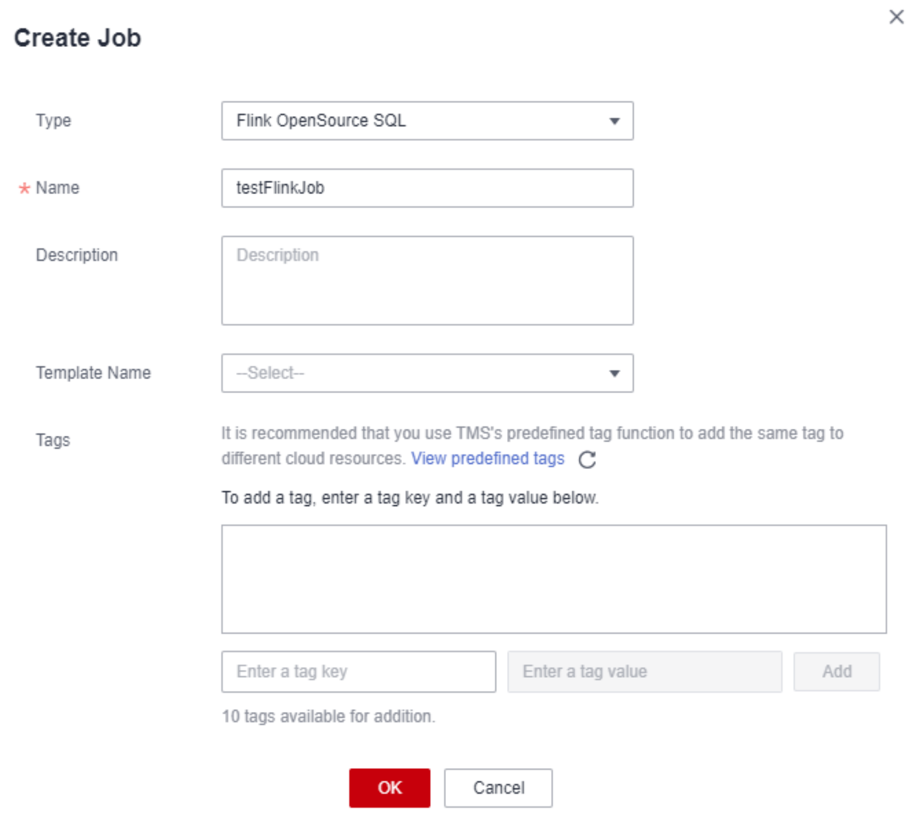
For how to enable CTS and view trace details, see the **Cloud Trace Service Getting Started**.

Creating a Flink OpenSource SQL Job

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 In the upper right corner of the **Flink Jobs** page, click **Create Job**.

Figure 8-1 Creating a Flink OpenSource SQL job



Step 3 Set job parameters.

Table 8-1 Job parameters

Parameter	Description
Type	Set **Type** to **Flink OpenSource SQL**. You will need to start jobs by compiling SQL statements.
Name	Job name. Enter 1 to 57 characters. Only letters, numbers, hyphens (-), and underscores (_) are allowed. **NOTE** The job name must be globally unique.
Description	Description of a job. It can contain a maximum of 512 characters.
Template Name	You can select a sample template or a custom job template. For details about templates, see **Managing Flink Job Templates**.

Parameter	Description
Tags	Tags used to identify cloud resources. A tag includes the tag key and tag value. If you want to use the same tag to identify multiple cloud resources, that is, to select the same tag from the drop-down list box for all services, you are advised to create predefined tags on the Tag Management Service (TMS). If your organization has configured tag policies for DLI, add tags to resources based on the policies. If a tag does not comply with the tag policies, resource creation may fail. Contact your organization administrator to learn more about tag policies. For details, see **Tag Management Service User Guide**. **NOTE** ● A maximum of 20 tags can be added. ● Only one tag value can be added to a tag key. ● The key name in each resource must be unique. ● Tag key: Enter a tag key name in the text box. **NOTE** A tag key can contain a maximum of 128 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed, but the value cannot start or end with a space or start with **_sys_**. ● Tag value: Enter a tag value in the text box. **NOTE** A tag value can contain a maximum of 255 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed. The value cannot start or end with a space.

Step 4 Click **OK** to enter the editing page.

Step 5 Edit an OpenSource SQL job.

Enter detailed SQL statements in the statement editing area. For details about SQL statements, see the **Data Lake Insight Flink OpenSource SQL Syntax Reference**.

Step 6 Click **Check Semantics**.

- You can **Start** a job only after the semantic verification is successful.
- If verification is successful, the message "The SQL semantic verification is complete. No error." will be displayed.
- If verification fails, a red "X" mark will be displayed in front of each SQL statement that produced an error. You can move the cursor to the "X" mark to view error details and change the SQL statement as prompted.

NOTE

Flink 1.15 does not support syntax verification.

Step 7 Set job running parameters.

Figure 8-2 Setting running parameters for Flink OpenSource SQL

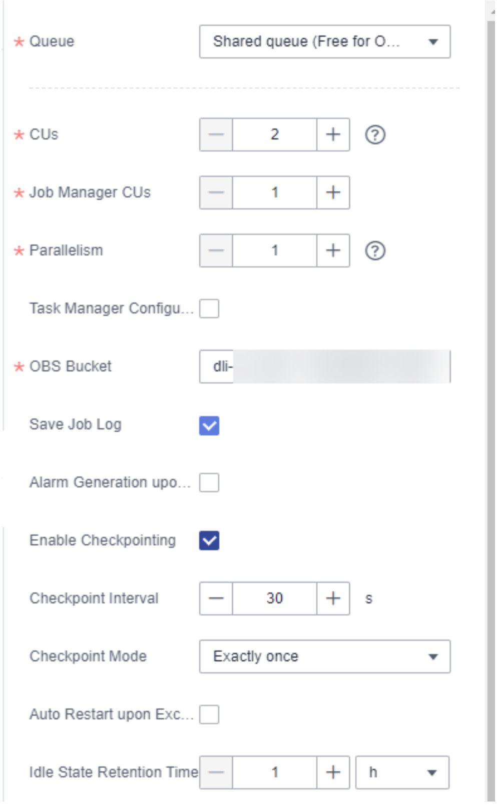


Table 8-2 Running parameters

Parameter	Description
Queue	Select a queue to run the job.
UDF Jar	UDF JAR file, which contains UDFs that can be called in subsequent jobs. There are the following ways to manage UDF JAR files: ● Upload packages to OBS: Upload Jar packages to an OBS bucket in advance and select the corresponding OBS path. ● Upload packages to DLI: Upload JAR files to an OBS bucket in advance and create a package on the **Data Management** > **Package Management** page of the DLI management console. For details, see **Creating a DLI Package**. For Flink 1.15 or later, only OBS packages can be selected when creating jobs, and DLI packages are not supported.

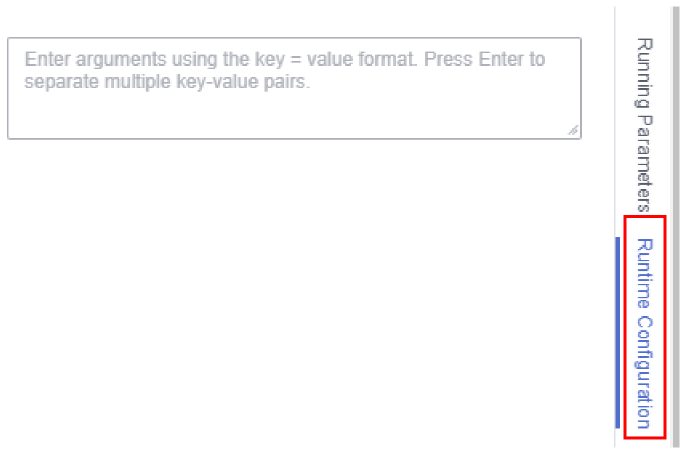
Parameter	Description
Flink Version	Flink version used for job running. Flink versions have varying feature support. If you choose to use Flink 1.15, make sure to configure the agency information for the cloud service that DLI is allowed to access in the job. For the syntax of Flink 1.15, see **Flink OpenSource SQL 1.15 Usage** and **Flink OpenSource SQL 1.15 Syntax**. For the syntax of Flink 1.12, see **Flink OpenSource SQL 1.12 Syntax**. **NOTE** You are not advised to use Flink of different versions for a long time. ● Doing so can lead to code incompatibility, which can negatively impact job execution efficiency. ● Doing so may result in job execution failures due to conflicts in dependencies. Jobs rely on specific versions of libraries or components.
Agency	If you choose Flink 1.15 to execute your job, you can create a custom agency to allow DLI to access other services.
CUs	Sum of the number of compute units and job manager CUs of DLI. CU is also the billing unit of DLI. One CU equals 1 vCPU and 4 GB. The value is the number of CUs required for job running and cannot exceed the number of CUs in the bound queue.
Job Manager CUs	Number of CUs of the management unit.
Parallelism	Number of tasks concurrently executed by each operator in a job. **NOTE** This value cannot be greater than four times the compute units (number of CUs minus the number of job manager CUs).
Task Manager Config	Whether Task Manager resource parameters are set ● If selected, you need to set the following parameters: – **CU(s) per TM**: Number of resources occupied by each Task Manager. – **Slot(s) per TM**: Number of slots contained in each Task Manager. ● If not selected, the system automatically uses the default values. – **CU(s) per TM**: The default value is **1**. – **Slot(s) per TM**: The default value is (Parallelism x CU(s) per TM)/(CUs – Job Manager CUs).
OBS Bucket	OBS bucket to store job logs and checkpoint information. If the OBS bucket you selected is unauthorized, click **Authorize**.

Parameter	Description
Save Job Log	Whether job running logs are saved to OBS. The logs are saved in the following path: *Bucket name*/**jobs/logs/***Directory starting with the job ID*. **CAUTION** You are advised to configure this parameter. Otherwise, no run log is generated after the job is executed. If the job fails, the run log cannot be obtained for fault locating. If this option is selected, you need to set the following parameters: **OBS Bucket**: Select an OBS bucket to store job logs. If the OBS bucket you selected is unauthorized, click **Authorize**. **NOTE** If **Enable Checkpointing** and **Save Job Log** are both selected, you only need to authorize OBS once.
Alarm on Job Exception	Whether to notify users of any job exceptions, such as running exceptions or arrears, via SMS or email. If this option is selected, you need to set the following parameters: **SMN Topic** Select a custom SMN topic. For how to create a custom SMN topic, see **Creating a Topic**.
Enable Checkpointing	Whether to enable job snapshots. If this function is enabled, jobs can be restored based on the checkpoints. If this option is selected, you need to set the following parameters: ● **Checkpoint Interval**: interval for creating checkpoints, in seconds. The value ranges from 1 to 999999, and the default value is **30**. ● **Checkpoint Mode** can be set to either of the following values: – **At least once**: Events are processed at least once. – **Exactly once**: Events are processed only once. If you select **Enable Checkpointing**, you also need to set **OBS Bucket**. **OBS Bucket**: Select an OBS bucket to store your checkpoints. If the OBS bucket you selected is unauthorized, click **Authorize**. The checkpoint path is *Bucket name*/**jobs/checkpoint/***Directory starting with the job ID*. **NOTE** If **Enable Checkpointing** and **Save Job Log** are both selected, you only need to authorize OBS once.

Parameter	Description
Auto Restart upon Exception	Whether automatic restart is enabled. If enabled, jobs will be automatically restarted and restored when exceptions occur. If this option is selected, you need to set the following parameters: ● **Max. Retry Attempts**: maximum number of retries upon an exception. The unit is times/hour. – **Unlimited**: The number of retries is unlimited. – **Limited**: The number of retries is user-defined. ● **Restore Job from Checkpoint**: This parameter is available only when **Enable Checkpointing** is selected.
Idle State Retention Time	Clears intermediate states of operators such as **GroupBy**, **RegularJoin**, **Rank**, and **Depulicate** that have not been updated after the maximum retention time. The default value is 1 hour.
Dirty Data Policy	Policy for processing dirty data. The following policies are supported: **Ignore**, **Trigger a job exception**, and **Save**. If you set this field to **Save**, the **Dirty Data Dump Address** must be set. Click the address box to select the OBS path for storing dirty data. This parameter is available only when a DIS data source is used.

Step 8 (Optional) Set the runtime configuration as required. For details about related parameters, see**How Do I Optimize Performance of a Flink Job?**

Figure 8-3 Runtime configuration



Step 9 Click **Save**.

Step 10 Click **Start**. On the displayed **Start Flink Jobs** page, confirm the job specifications and the price, and click **Start Now** to start the job.

After the job is started, the system automatically switches to the **Flink Jobs** page, and the created job is displayed in the job list. You can view the job status in the **Status** column. Once a job is successfully submitted, its status changes from **Submitting** to **Running**. After the execution is complete, the status changes to **Completed**.

If the job status is **Submission failed** or **Running exception**, the job fails to submit or run. In this case, you can hover over the status icon in the **Status** column of the job list to view the error details. You can click ❐ to copy these details. Rectify the fault based on the error information and resubmit the job.

NOTE

Other buttons are as follows:

- **Save As**: Save the created job as a new job.
- **Static Stream Graph**: Provide the static concurrency estimation function and stream graph display function. See **Figure 8-5**.
- **Simplified Stream Graph**: Display the data processing flow from the source to the sink. See **Figure 8-4**.
- **Format**: Format the SQL statements in the editing box.
- **Set as Template**: Set the created SQL statements as a job template.
- **Theme Settings**: Set the theme related parameters, including **Font Size**, **Wrap**, and **Page Style**.
- **Help**: Redirect to the help center to provide you with the SQL syntax for stream jobs.

----End

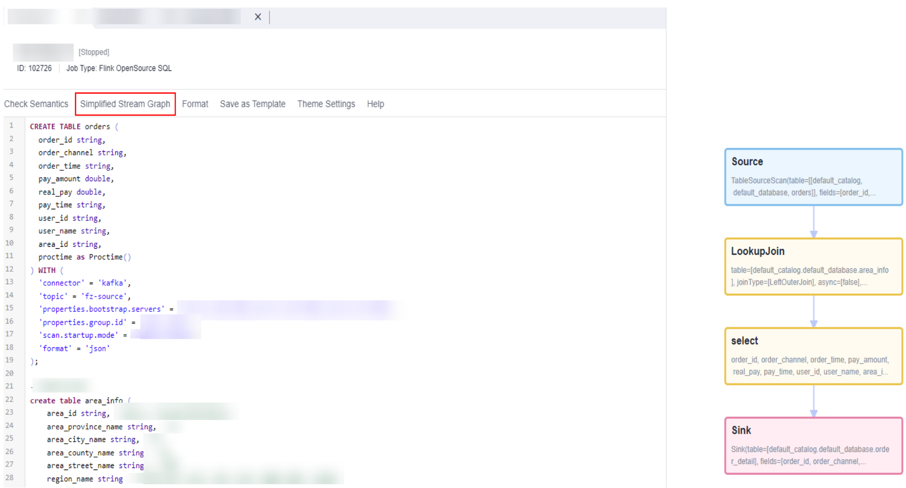
Simplified Stream Graph

On the OpenSource SQL job editing page, click **Simplified Stream Graph**.

NOTE

Simplified stream graph viewing is only supported in Flink 1.12 and Flink 1.10.

Figure 8-4 Simplified stream graph



Static Stream Graph

On the OpenSource SQL job editing page, click **Static Stream Graph**.

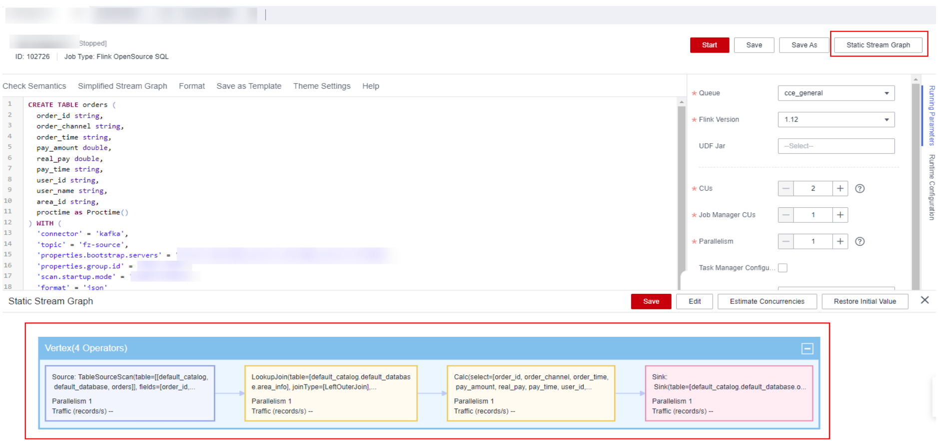
NOTE

- Simplified stream graph viewing is only supported in Flink 1.12 and Flink 1.10.
- If you use a UDF in a Flink OpenSource SQL job, it is not possible to generate a static stream graph.

The **Static Stream Graph** page also allows you to:

- Estimate concurrencies. Click **Estimate Concurrencies** on the **Static Stream Graph** page to estimate concurrencies. Click **Restore Initial Value** to restore the initial value after concurrency estimation.
- Zoom in or out the page.
- Expand or merge operator chains.
- You can edit **Parallelism**, **Output rate**, and **Rate factor**.
  - **Parallelism**: indicates the number of concurrent tasks.
  - **Output rate**: indicates the data traffic of an operator. The unit is piece/s.
  - **Rate factor**: indicates the retention rate after data is processed by operators. Rate factor = Data output volume of an operator/Data input volume of the operator (Unit: %)

Figure 8-5 Static stream graph



8.3 Creating a Flink Jar Job

A Flink Jar job involves developing a custom application Jar package based on Flink's capabilities and submitting it to a DLI queue for execution.

To create a Flink Jar job, you need to write and build your own application Jar package. This is suitable for users who require stream data processing and are proficient in Flink's secondary development capabilities.

This section describes how to create a Flink Jar job on the DLI management console.

Prerequisites

- When you use a Flink Jar job to access other external data sources, such as OpenTSDB, HBase, Kafka, GaussDB(DWS), RDS, CSS, CloudTable, DCS Redis, and DDS, you need to create a datasource connection to connect the job running queue to the external data source.
  - For details about the external data sources that can be accessed by Flink jobs, see **Common Development Methods for DLI Cross-Source Analysis**.
  - For how to create a datasource connection, see **Configuring the Network Connection Between DLI and Data Sources (Enhanced Datasource Connection)**.

    On the **Resources** > **Queue Management** page, locate the queue you have created, click **More** in the **Operation** column, and select **Test Address Connectivity** to check if the network connection between the queue and the data source is normal. For details, see **Testing Address Connectivity**.
- To run a Flink Jar job, you need to build your custom application code into a JAR file and upload it to the OBS bucket that has already been created.
- 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** in Maven or SBT)
  - Log configuration files (example, **log4j.properties**/**logback.xml**)
  - JAR package for log output implementation (example, **log4j**).

Precautions

Before creating jobs and submitting tasks, you are advised to enable CTS to record operations associated with DLI for later query, audit, and backtrack operations. To view the DLI operations that can be recorded by CTS, see **Using CTS to Audit DLI**.

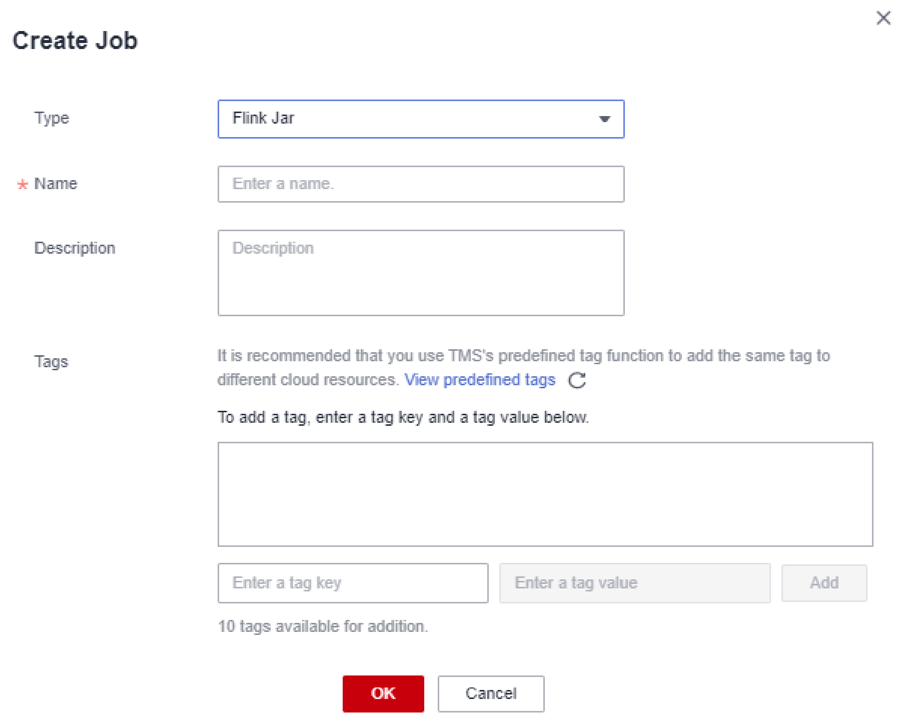
For how to enable CTS and view trace details, see the **Cloud Trace Service Getting Started**.

Creating a Flink Jar Job

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 In the upper right corner of the **Flink Jobs** page, click **Create Job**.

Figure 8-6 Creating a Flink Jar job



Step 3 Specify job parameters.

Table 8-3 Job configuration information

Parameter	Description
Type	Select **Flink Jar**.
Name	Job name. Enter 1 to 57 characters. Only letters, numbers, hyphens (-), and underscores (_) are allowed. **NOTE** The job name must be globally unique.
Description	Description of a job. It can be up to 512 characters long.

Parameter	Description
Tags	Tags used to identify cloud resources. A tag includes the tag key and tag value. If you want to use the same tag to identify multiple cloud resources, that is, to select the same tag from the drop-down list box for all services, you are advised to create predefined tags on the Tag Management Service (TMS). If your organization has configured tag policies for DLI, add tags to resources based on the policies. If a tag does not comply with the tag policies, resource creation may fail. Contact your organization administrator to learn more about tag policies. For details, see **Tag Management Service User Guide**. **NOTE** • A maximum of 20 tags can be added. • Only one tag value can be added to a tag key. • The key name in each resource must be unique. • Tag key: Enter a tag key name in the text box. **NOTE** A tag key can contain a maximum of 128 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed, but the value cannot start or end with a space or start with **_sys_**. • Tag value: Enter a tag value in the text box. **NOTE** A tag value can contain a maximum of 255 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed. The value cannot start or end with a space.

Step 4 Click **OK** to enter the editing page.

Step 5 Select a queue.

Step 6 Configuring Flink Jar Job parameters

Figure 8-7 Configuring Flink Jar Job parameters

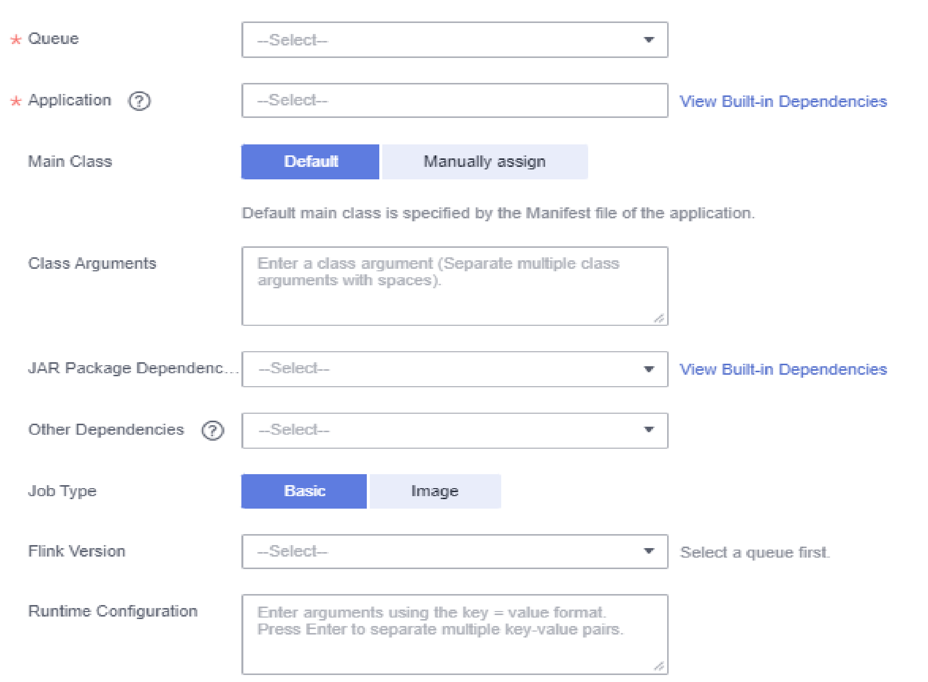


Table 8-4 Parameters

Parameter	Description
Queue	Select a queue where you want to run your job.
Application	Select a Jar job package. There are the following ways to manage JAR files: ● Upload packages to OBS: Upload Jar packages to an OBS bucket in advance and select the corresponding OBS path. ● Upload packages to DLI: Upload JAR files to an OBS bucket in advance and create a package on the **Data Management** > **Package Management** page of the DLI management console. For details, see **Creating a DLI Package**. For Flink 1.15 or later, you can only select a package from OBS, instead of DLI.
Main Class	The name of the JAR package to be loaded, for example, **KafkaMessageStreaming**. ● **Default**: Specified based on the **Manifest** file in the JAR package. ● **Manually assign**: You must enter the class name and confirm the class arguments (separated by spaces). **NOTE** When a class belongs to a package, the main class path must contain the complete package path, for example, **packagePath.KafkaMessageStreaming**.

Parameter	Description
Class Arguments	List of arguments of a specified class. The arguments are separated by spaces. Flink parameters support replacement of non-sensitive global variables. For example, if you add the global variable **windowsize** in **Global Configuration** > **Global Variables**, you can add the **-windowsSize {{windowsize}}** parameter for the Flink Jar job.
JAR Package Dependencies	Select a user-defined package dependency. The dependent program packages are stored in the classpath directory of the cluster. There are the following ways to manage JAR files: ● Upload packages to OBS: Upload Jar packages to an OBS bucket in advance and select the corresponding OBS path. ● Upload packages to DLI: Upload JAR files to an OBS bucket in advance and create a package on the **Data Management** > **Package Management** page of the DLI management console. For details, see **Creating a DLI Package**. For Flink 1.15 or later, you can only select a package from OBS, instead of DLI. When creating a JAR file for a Flink Jar job, you do not need to upload existing built-in dependency packages to avoid package information conflicts. For details about built-in dependency packages, see **DLI Built-in Dependencies**.
Other Dependencies	User-defined dependency files. Other dependency files need to be referenced in the code. There are the following ways to manage dependency files: ● Upload packages to OBS: Upload dependency files to an OBS bucket in advance and select the corresponding OBS path. ● Upload packages to DLI: Upload dependency files to an OBS bucket in advance and create a package on the **Data Management** > **Package Management** page of the DLI management console. For details, see **Creating a DLI Package**. For Flink 1.15 or later, you can only select a package from OBS, instead of DLI. You can add the following command to the application to access the corresponding dependency file. In the command, **fileName** indicates the name of the file to be accessed, and **ClassName** indicates the name of the class that needs to access the file. `ClassName.class.getClassLoader().getResource("userData/fileName")`

Parameter	Description
Job Type	This parameter is displayed when the queue type is CCE. ● **Basic** ● **Image**: Select the image name and image version. Image set on the Software Repository for Container (SWR) console. For details, see **Using a Custom Image to Enhance the Job Running Environment**.
Flink Version	Set **Queue** before setting this parameter. **NOTE** You are not advised to use Flink of different versions for a long time. ● Doing so can lead to code incompatibility, which can negatively impact job execution efficiency. ● Doing so may result in job execution failures due to conflicts in dependencies. Jobs rely on specific versions of libraries or components.
Agency	If you choose Flink 1.15 to execute your job, you can create a custom agency to allow DLI to access other services.
Runtime Configuration	User-defined optimization parameters. The parameter format is **key=value**. Flink optimization parameters support replacement non-sensitive global variable. For example, if you create global variable **phase** in **Global Configuration** > **Global Variables**, optimization parameter **table.optimizer.agg-phase.strategy={{phase}}** can be added to the Flink Jar job. Flink 1.15 supports minimal submission of Flink Jar jobs. Enable this by configuring **flink.dli.job.jar.minimize-submission.enabled=true** in the runtime optimization parameters. **NOTE** Minimal submission means Flink only submits the necessary job dependencies, not the entire Flink environment. By setting the scope of non-Connector Flink dependencies (starting with **flink-**) and third-party libraries (like Hadoop, Hive, Hudi, and MySQL-CDC) to **provided**, you ensure these dependencies are excluded from the Jar job, avoiding conflicts with Flink core dependencies. ● Only Flink 1.15 supports minimal submission of Flink Jar jobs. ● For Flink-related dependencies, use the **provided** scope by adding **<scope>provided</scope>** in the dependencies, especially for non-Connector dependencies under the **org.apache.flink** group starting with **flink-**. ● For dependencies related to Hadoop, Hive, Hudi, and MySQL-CDC, also use the **provided** scope by adding **<scope>provided</scope>** in the dependencies. ● In the Flink source code, only methods marked with **@Public** or **@PublicEvolving** are intended for user invocation. DLI guarantees compatibility with these methods.

Step 7 Set compute resource specification parameters.

Figure 8-8 Configuring job parameters

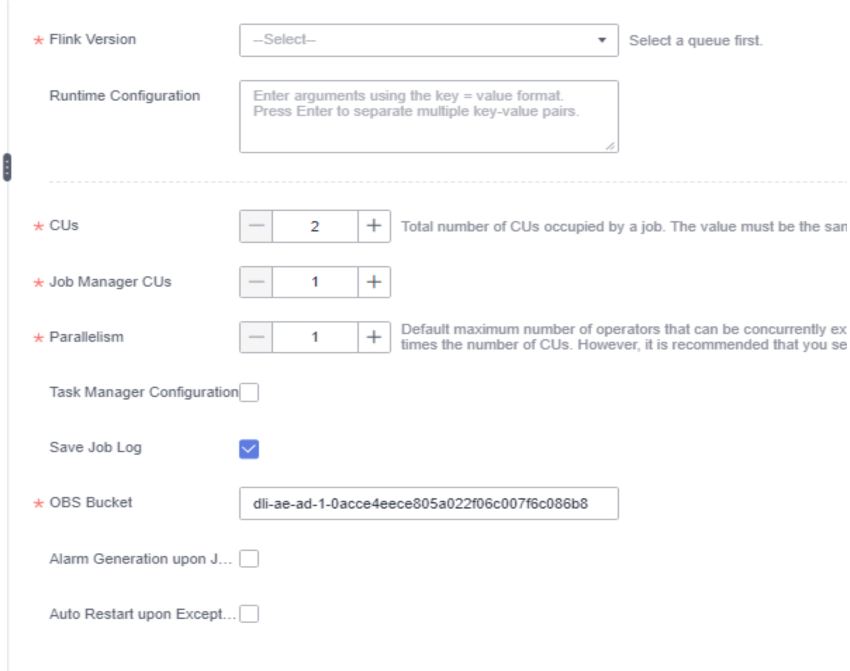


Table 8-5 Parameters

Parameter	Description
CUs	One CU consists of one vCPU and 4 GB of memory. The number of CUs ranges from 2 to 10000.
Job Manager CUs	Number of CUs for the job management unit.
Parallelism	Number of tasks concurrently executed by each operator in a job. **NOTE** • The value must be less than or equal to four times the number of compute units (CUs minus the number of job manager CUs). • Set this parameter to a value greater than that configured in the code to avoid job submission failures.

Parameter	Description
Task Manager Config	Whether Task Manager resource parameters are set ● If this option is selected, you need to set the following parameters: – **CU(s) per TM**: Number of resources occupied by each Task Manager. – **Slot(s) per TM**: Number of slots contained in each Task Manager. ● If not selected, the system automatically uses the default values. – **CU(s) per TM**: The default value is **1**. – **Slot(s) per TM**: The default value is (Parallelism x CU(s) per TM)/(CUs – Job Manager CUs).
Save Job Log	Whether to save the job running logs to the OBS bucket. If you select this parameter, you also need to set **OBS Bucket**. **CAUTION** You are advised to select this parameter. Otherwise, no run log is generated after the job is executed. If the job is abnormal, the run log cannot be obtained for fault locating.

Parameter	Description
Enable Checkpointing	Checkpoints are used to periodically save the job state. Enabling checkpointing allows for the quick recovery of a specific job state in case of system failure. There are two ways to enable checkpointing in DLI: ● Configure checkpoint-related parameters in the job code, suitable for Flink 1.15 or earlier. ● Enable checkpointing on the Jar job configuration page of the DLI management console, suitable for Flink 1.15 or later. **For Flink 1.15, do not configure checkpoint-related parameters both in the job code and the Jar job configuration page. The configurations in the job code have higher priority. Duplicate configurations may lead to the use of incorrect checkpoint paths during abnormal restarts, causing recovery failures or data inconsistencies.** After selecting **Enable Checkpointing**, set the following parameters to enable checkpointing: ● **Checkpoint Interval**: The interval between checkpoints, in seconds. ● **Checkpoint Mode**: Select a mode for checkpoints. The options are: – **At least once**: Events are processed at least once. – **Exactly once**: Events are processed exactly once. **CAUTION** ● After selecting **Enable Checkpointing**, you need to set **OBS Bucket** to save the checkpoint information. The default checkpoint save path is *Bucket name*/**jobs/checkpoint/***Directory with job ID prefix*. ● Once checkpointing is enabled, do not set checkpoint parameters in the job code, as the parameters configured in the job code have a higher priority than those configured on the job configuration page. Duplicate configurations may cause the job to use incorrect checkpoint paths during abnormal restarts, resulting in recovery failures or data inconsistencies. ● After enabling checkpointing, if **Auto Restart on Exception** and **Restore Job from Checkpoint** are both selected, you do not need to set **Checkpoint Path**. The system will automatically determine the path based on the **Enable Checkpointing** configuration.
OBS Bucket	This parameter is mandatory when you select **Save Job Log** or **Enable Checkpointing**. Select an OBS bucket to store job logs. If the OBS bucket you select is not authorized, click **Authorize**.
Alarm on Job Exception	Whether to notify users of any job exceptions, such as running exceptions or arrears, via SMS or email. If this option is selected, you need to set the following parameters: **SMN Topic** Select a custom SMN topic. For how to create a custom SMN topic, see **Creating a Topic**.

Parameter	Description
Auto Restart upon Exception	Whether automatic restart is enabled. If enabled, jobs will be automatically restarted and restored when exceptions occur. If this option is selected, you need to set the following parameters: ● **Max. Retry Attempts**: maximum number of retries upon an exception. The unit is times/hour. – **Unlimited**: The number of retries is unlimited. – **Limited**: The number of retries is user-defined. ● **Restore Job from Checkpoint**: Restore the job from the saved checkpoint. If you select this parameter, you need to set **Checkpoint Path**. **Checkpoint Path**: Select the checkpoint save path. This path must match that configured in the application package. Each job must have a unique checkpoint path, or, you will not be able to obtain the checkpoint. **NOTE** – If you also select **Enable Checkpointing**, you do not need to set **Checkpoint Path**. The system will automatically determine the path based on the **Enable Checkpointing** configuration. – If you do not select **Enable Checkpointing**, you need to set **Checkpoint Path**.

Step 8 Click **Save** on the upper right of the page.

Step 9 Click **Start** in the upper right corner. On the displayed **Start Flink Job** page, confirm the job specifications and the price, and click **Start Now** to start the job. After the job is started, the system automatically switches to the **Flink Jobs** page, and the created job is displayed in the job list. You can view the job status in the **Status** column.

- Once a job is successfully submitted, its status changes from **Submitting** to **Running**. After the execution is complete, the status changes to **Completed**.
- If the job status is **Submission failed** or **Running exception**, the job fails to submit or run. In this case, you can hover over the status icon in the **Status** column of the job list to view the error details. You can click to copy these details. Rectify the fault based on the error information and resubmit the job.

----End

8.4 Configuring Flink Job Permissions

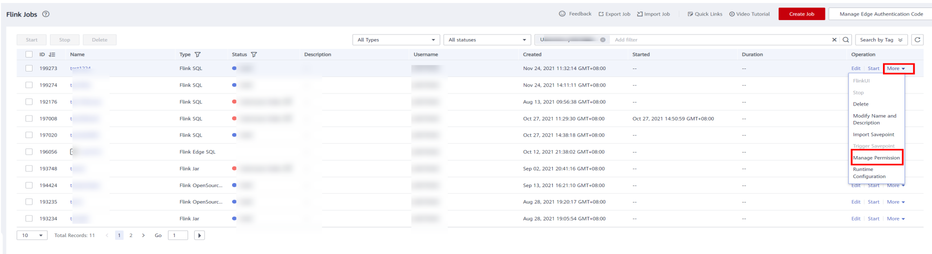
Scenario

- You can isolate Flink jobs allocated to different users by setting permissions to ensure data query performance.
- The administrator and job creator have all permissions, which cannot be set or modified by other users.
- When setting job permissions for a new user, ensure that the region of the user group to which the user belongs has the **Tenant Guest** permission. For

details about the Tenant Guest permission and how to apply for the permission, see **Permissions Policies** and **Creating a User Group and Assigning Permissions** in the *Identity and Access Management User Guide*.

Flink Job Permission Operations

1. On the left of the DLI management console, choose **Job Management** > **Flink Jobs**.
2. Select the job to be configured and choose **More** > **Permissions** in the **Operation** column. The **User Permissions** area displays the list of users who have permissions on the job.



You can assign queue permissions to new users, modify permissions for users who have some permissions of a queue, and revoke all permissions of a user on a queue.

– Assign permissions to a new user.

A new user does not have permissions on the job.

i. Click **Grant Permission** on the right of **User Permissions** page. The **Grant Permission** dialog box is displayed.

ii. Specify **Username** and select corresponding permissions.

iii. Click **OK**.

Table 8-6 describes the related parameters.

Figure 8-9 Granting permissions

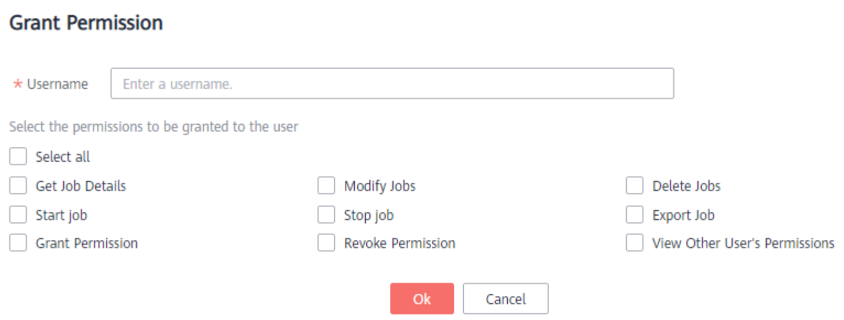


Table 8-6 Permission parameters

Parameter	Description
Username	Name of the user you want to grant permissions to. **NOTE** This username must be an existing IAM username. In addition, the user can perform authorization operations only after logging in to the Huawei Cloud platform.
Permissions to be granted to the user	• **Select all**: All permissions are selected. • **View Job Details**: This permission allows you to view the job details. • **Modify Job**: This permission allows you to modify the job. • **Delete Job**: This permission allows you to delete the job. • **Start Job**: This permission allows you to start the job. • **Stop Job**: This permission allows you to stop the job. • **Export Job**: This permission allows you to export the job. • **Grant Permission**: This permission allows you to grant job permissions to other users. • **Revoke Permission**: This permission allows you to revoke the job permissions that other users have but cannot revoke the job creator's permissions. • **View Other User's Permissions**: This permission allows you to view the job permissions of other users.

- To assign or revoke permissions of a user who has some permissions on the job, perform the following steps:
  i. In the list under **User Permissions** for a job, select the user whose permissions need to be modified and click **Set Permission** in the **Operation** column.
  ii. In the displayed **Set Permission** dialog box, modify the permissions of the current user. **Table 8-6** lists the detailed permission descriptions.

     If all options under **Set Permission** are gray, you are not allowed to change permissions on this job. You can apply to the administrator, job creator, or other authorized users for job permission granting and revoking.
  iii. Click **OK**.
- To revoke all permissions of a user on a job, perform the following steps:

  In the list under **User Permissions** for a job, locate the user whose permissions need to be revoked, click **Revoke Permission** in the

Operation column, and click **Yes**. After this operation, the user does not have any permission on the job.

Flink Job Permissions

- **View Job Details**
  - Tenants and the admin user can view and operate all jobs.
  - Subusers and users with the read-only permission can only view their own jobs.

    NOTE

    If another user grants any permission other than the job viewing permission to a subuser, the job is displayed in the job list, but the details cannot be viewed by the subuser.
- **Start Job**

  You must have the permission to submit and start jobs.
- **Stop Job**

  You must have the permission to stop queues and jobs.
- **Delete Job**
  - If a job can be deleted, you can delete the job if you were granted this permission.
  - If a job cannot be deleted, the system stops the job before you delete it. For details about how to stop a job, see **Stop Job**. In addition, you must have the permission to delete the job.
- **Create Job**
  - By default, sub-users cannot create jobs.
  - To create a job, you must have this permission. Currently, only the admin user has the permission to create jobs. In addition, the user must have the permission of the related package group or package used by the job.
- **Modify Job**

  When modifying a job, you need to have the permission to update the job and the permission to the package group or package used by the job belongs.

8.5 Managing Flink Jobs

8.5.1 Viewing Flink Job Details

After creating a Flink job, you can check the basic information, job details, task list, and execution plan of the job on the DLI console.

This section describes how to check information about a Flink job.

Table 8-7 Viewing Flink job information

Type	Description	Instruction
Basic information	Includes the job ID, job type, job execution status, and more.	**Viewing Basic Information**
Job details	Includes SQL statements and the parameter settings for Flink Jar jobs.	**Viewing Details**
Job monitoring	You can use Cloud Eye to check job data input and output details.	**Viewing Monitoring Information**
Task list	You can view details about each task running on a job, including the task start time, number of received and transmitted bytes, and running duration.	**Viewing the Task List**
Execution plan	You can understand the operator flow direction of a running job.	**Viewing the Execution Plan**

Viewing Basic Information

In the navigation pane of the DLI console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page displays all Flink jobs. You can check basic information about any Flink jobs in the list.

Table 8-8 Basic information about a Flink job

Parameter	Description
ID	ID of a submitted Flink job, which is generated by the system by default.
Name	Name of the submitted Flink job.
Type	Type of the submitted Flink job, which includes: ● **Flink SQL** ● **Flink Jar** ● **Flink OpenSource SQL**
Status	Job status, which is subject to the console.
Description	Description of the submitted Flink job.
Username	Name of the user who submits the job.
Created	Time when the job was created.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Started</td><td>Time when the Flink job started to run.</td></tr>
<tr><td>Duration</td><td>Time consumed by job running.</td></tr>
<tr><td>Operation</td><td>Edit: Edit a created job.
Start: Start and run a job.
More
FlinkUI: Selecting this will display the Flink job execution page.
NOTE
If you select FlinkUI immediately after submitting a job to a new queue, an empty projectID will be cached and the FlinkUI page cannot be displayed as it takes about 10 minutes to create a cluster in the background.
You are advised to use a dedicated queue to ensure immediate availability of clusters for your jobs. Alternatively, select FlinkUI when the job is in the Running state.
Stop: Stop a Flink job. If it is grayed out, jobs in the current state cannot be stopped.
Delete: Delete a job.
NOTE
Deleted jobs cannot be restored.
Modify Name and Description: You can modify the name and description of a job.
Import Savepoint: Import the data exported from the original Cloud Stream Service (CS) job.
Trigger Savepoint: You can select this operation for jobs in the Running state to save the job status.
Permissions: You can view the user permissions corresponding to the job and grant permissions to other users.
Runtime Configuration: You can enable Alarm Generation on Job Exception and Auto Restart on Exception.</td></tr>
</table>

Viewing Details

This section describes how to view job details. After you create and save a job, you can click the job name to view job details, including SQL statements and parameter settings. For a Jar job, you can only view its parameter settings.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Click the name of the job to be viewed. The **Job Detail** tab is displayed.

In the **Job Details** tab, you can view SQL statements, configured parameters, and total cost for the job.

The following uses a Flink SQL job as an example.

Table 8-9 Parameter descriptions

Parameter	Description
Type	Job type, for example, **Flink SQL**
Name	Flink job name
Description	Description of a Flink job
Status	Running status of a job
Running Mode	The dedicated resource mode is used by default.
Flink Version	Version of Flink selected for the job.
Queue	Name of the queue where the Flink job runs
UDF Jar	This parameter is displayed when **UDF Jar** is set.
Runtime Configuration	Displayed when a user-defined parameter is added to a job
CUs	Number of CUs configured for a job
Job Manager CUs	Number of job manager CUs configured for a job.
Parallelism	Number of jobs that can be concurrently executed by a Flink job
CU(s) per TM	Number of CUs occupied by each Task Manager configured for a job
Slot(s) per TM	Number of Task Manager slots configured for a job
OBS Bucket	OBS bucket name. After **Enable Checkpointing** and **Save Job Log** are enabled, checkpoints and job logs are saved in this bucket.
Save Job Log	Whether the job running logs are saved to OBS
Alarm on Job Exception	Whether job exceptions are reported
SMN Topic	Name of the SMN topic. This parameter is displayed when **Alarm Generation upon Job Exception** is enabled.
Auto Restart upon Exception	Whether automatic restart is enabled.
Max. Retry Attempts	Maximum number of retry times upon an exception. **Unlimited** means the number is not limited.
Restore Job from Checkpoint	Whether the job can be restored from a checkpoint
ID	Job ID
Savepoint	OBS path of the savepoint

Parameter	Description
Enable Checkpointing	Whether checkpointing is enabled
Checkpoint Interval	Interval between storing intermediate job running results to OBS. The unit is second.
Checkpoint Mode	Checkpoint mode. Available values are as follows: ● **At least once**: Events are processed at least once. ● **Exactly once**: Events are processed only once.
Idle State Retention Time	Clears intermediate states of operators such as **GroupBy**, **RegularJoin**, **Rank**, and **Depulicate** that have not been updated after the maximum retention time. The default value is 1 hour.
Dirty Data Policy	Policy for processing dirty data. The value is displayed only when there is a dirty data policy. Available values are as follows: **Ignore** **Trigger a job exception** **Save**
Dirty Data Dump Address	OBS path for storing dirty data when **Dirty Data Policy** is set to **Save**.
Created	Time when a job is created
Updated	Time when a job was last updated

----End

Viewing Monitoring Information

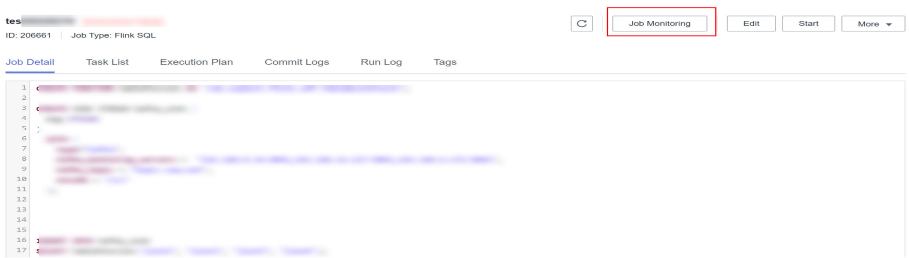
You can use Cloud Eye to view details about job data input and output.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Click the name of the job you want. The job details are displayed.

Click **Job Monitoring** in the upper right corner of the page to switch to the Cloud Eye console.

Figure 8-10 Monitoring a Job



The following table describes monitoring metrics related to Flink jobs.

Table 8-10 Monitoring metrics related to Flink jobs

Name	Description
Flink Job Data Read Rate	Displays the data input rate of a Flink job for monitoring and debugging. Unit: record/s.
Flink Job Data Write Rate	Displays the data output rate of a Flink job for monitoring and debugging. Unit: record/s.
Flink Job Total Data Read	Displays the total number of data inputs of a Flink job for monitoring and debugging. Unit: records
Flink Job Total Data Write	Displays the total number of output data records of a Flink job for monitoring and debugging. Unit: records
Flink Job Byte Read Rate	Displays the number of input bytes per second of a Flink job. Unit: byte/s
Flink Job Byte Write Rate	Displays the number of output bytes per second of a Flink job. Unit: byte/s
Flink Job Total Read Byte	Displays the total number of input bytes of a Flink job. Unit: byte
Flink Job Total Write Byte	Displays the total number of output bytes of a Flink job. Unit: byte
Flink Job CPU Usage	Displays the CPU usage of Flink jobs. Unit: %
Flink Job Memory Usage	Displays the memory usage of Flink jobs. Unit: %
Flink Job Max Operator Latency	Displays the maximum operator delay of a Flink job. The unit is **ms**.
Flink Job Maximum Operator Backpressure	Displays the maximum operator backpressure value of a Flink job. A larger value indicates severer backpressure. **0**: OK **50**: low **100**: high

----End

Viewing the Task List

You can view details about each task running on a job, including the task start time, number of received and transmitted bytes, and running duration.

NOTE

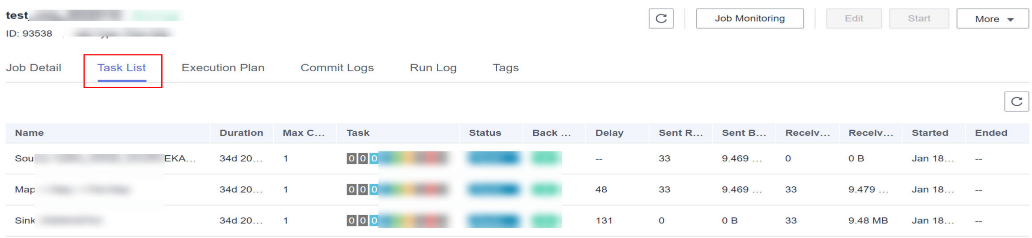
If the value is **0**, no data is received from the data source.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Click the name of the job you want. The job details are displayed.

Step 3 On **Task List** and view the node information about the task.

Figure 8-11 Task list



View the operator task list. The following table describes the task parameters.

Table 8-11 Parameter descriptions

Parameter	Description
Name	Name of an operator.
Duration	Running duration of an operator.
Max Concurrent Jobs	Number of parallel tasks in an operator.
Task	Operator tasks are categorized as follows: ● The digit in red indicates the number of failed tasks. ● The digit in light gray indicates the number of canceled tasks. ● The digit in yellow indicates the number of tasks that are being canceled. ● The digit in green indicates the number of finished tasks. ● The digit in blue indicates the number of running tasks. ● The digit in sky blue indicates the number of tasks that are being deployed. ● The digit in dark gray indicates the number of tasks in a queue.
Status	Status of an operator task.

Parameter	Description
Back Pressure Status	Working load status of an operator. Available options are as follows: ● **OK**: indicates that the operator is in normal working load. ● **LOW**: indicates that the operator is in slightly high working load. DLI processes data quickly. ● **HIGH**: indicates that the operator is in high working load. The data input speed at the source end is slow.
Delay	Duration from the time when source data starts being processed to the time when data reaches the current operator. The unit is millisecond.
Sent Records	Number of data records sent by an operator.
Sent Bytes	Number of bytes sent by an operator.
Received Bytes	Number of bytes received by an operator.
Received Records	Number of data records received by an operator.
Started	Time when an operator starts running.
Ended	Time when an operator stops running.

----End

Viewing the Execution Plan

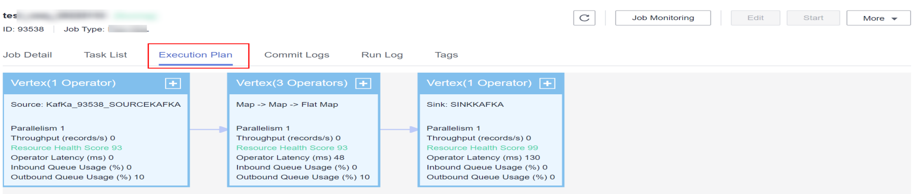
You can view the execution plan to understand the operator stream information about the running job.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Click the name of the job you want. The job details are displayed.

Step 3 Click the **Execution Plan** tab to view the operator flow direction.

Figure 8-12 Execution plan



Click a node. The corresponding information is displayed on the right of the page.

- Scroll the mouse wheel to zoom in or out.

- The stream diagram displays the operator stream information about the running job in real time.

----End

8.5.2 Setting the Priority for a Flink Job

Scenario

In actual job running, it is necessary to prioritize and ensure the normal running of important and urgent tasks due to their varying levels of importance and urgency. This requires providing the necessary compute resources for their normal operations.

DLI offers a feature to set job priorities for each Flink job, which prioritizes the allocation of compute resources to higher priority jobs when resources are limited.

NOTE

You can set the priority for Flink 1.12 or later jobs.

Notes

- You can assign a priority level of 1 to 10 for each job, with a larger value indicating a higher priority. Compute resources are preferentially allocated to high-priority jobs. That is, if compute resources required for high-priority jobs are insufficient, compute resources for low-priority jobs are reduced.
- Flink jobs running on a general-purpose queue have a default priority level of 5.
- The job priority change will only be in effect once the job has been stopped, edited, and resubmitted.
- You can set the priority for Flink jobs only after enabling dynamic scaling by setting **flink.dli.job.scale.enable** to **true**. For details, see **Enabling Dynamic Scaling for Flink Jobs**.
- To change the priority for a job, you must first stop the job, change the priority level, and then submit the job for the modification to take effect.

Setting the Priority for a Flink OpenSource SQL Job

1. Log in to the DLI management console.
2. In the navigation pane on the left, choose **Job Management** > **Flink Jobs**.
3. Select the job for which you want to set the priority and click **Edit** in the **Operation** column.
4. On the far right of the displayed page, click **Runtime Configuration**.
5. Enter statements in the text box to enable dynamic scaling and then set the job priority.

   NOTE

   To set the priority for Flink jobs, you must first enable dynamic scaling by setting **flink.dli.job.scale.enable** to **true**.

   For more parameter settings, see **Enabling Dynamic Scaling for Flink Jobs**.

```
flink.dli.job.scale.enable=true
flink.dli.job.priority=x
```

Figure 8-13 Example configuration for a Flink OpenSource SQL job

Setting the Priority for a Flink Jar Job

Enter the following statement in the **Runtime Configuration** text box, where *x* indicates the priority value:

```
flink.dli.job.priority=x
```

1. Log in to the DLI management console.
2. In the navigation pane on the left, choose **Job Management** > **Flink Jobs**.
3. Select the job for which you want to set the priority and click **Edit** in the **Operation** column.
4. In the **Runtime Configuration** text box, enter the following statements to enable dynamic scaling and set the job priority:

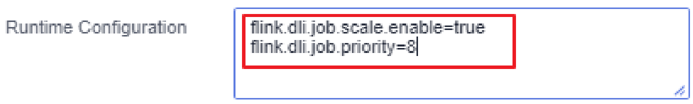
   NOTE

   To set the priority for Flink jobs, you must first enable dynamic scaling by setting **flink.dli.job.scale.enable** to **true**.

   For more parameter settings, see **Enabling Dynamic Scaling for Flink Jobs**.

```
flink.dli.job.scale.enable=true
flink.dli.job.priority=x
```

Figure 8-14 Example configuration for a Flink Jar job



8.5.3 Enabling Dynamic Scaling for Flink Jobs

Scenario

In actual job operations, the compute resources required by a job vary depending on the data volume. As a result, compute resources are wasted when the volume is small and are insufficient when the volume is large.

DLI provides dynamic scaling to dynamically adjust the compute resources used by a job based on the job load, such as the data input and output volume, data input and output rate, and backpressure, to improve resource utilization.

After enabling dynamic scaling for Flink jobs, the system will adjust resource allocation based on the actual resource requirements of the Flink jobs. When the remaining pod resources in the elastic resource pool are sufficient to meet the minimum resource requirements of the job, the system will automatically reduce the number of nodes where the job is running. This ensures efficient job running while improving resource utilization.

NOTE

Currently, dynamic scaling can only be enabled for Flink 1.12 jobs.

Notes

- During dynamic scaling of a Flink job, if queue resources are preempted and the remaining resources are insufficient for starting the job, the job may fail to be restored.
- When the resources that can be used by a Flink job are dynamically scaled in or out, the background job needs to be stopped and then restored from the savepoint. So, the job cannot process data before the restoration is successful.
- Savepoints need to be triggered during scaling. So, you must configure an OBS bucket, save logs, and enable checkpointing.
- Do not set the scaling detection period to a small value to avoid frequent job start and stop.
- The restoration duration of a scaling job is affected by the savepoint size. If the savepoint size is large, the restoration may take a long time.
- To adjust the configuration items of dynamic scaling, you need to stop the job, edit the job, and submit the job for the modification to take effect.

Procedure

Dynamic scaling applies to Flink OpenSource SQL and Flink Jar jobs.

1. Log in to the DLI management console.
2. In the navigation pane on the left, choose **Job Management** > **Flink Jobs**.
3. Select the job for which you want to enable dynamic scaling and click **Edit** in the **Operation** column.
   – For a Flink OpenSource SQL job, click **Runtime Configuration** on the right to configure dynamic scaling parameters.
   – For a Flink Jar job, click the **Runtime Configuration** box to configure dynamic scaling parameters.

Table 8-12 Dynamic scaling parameters

Parameter	Default Value	Description
flink.dli.job.scale.enable	false	Whether to enable dynamic scaling, that is, whether to allow DLI to adjust the resources used by jobs based on job loads and job priorities. If this parameter is set to **false**, the function is disabled. If this parameter is set to **true**, the function is enabled. The default value is **false**.
flink.dli.job.scale.interval	30	Interval for checking whether to scale the resources for the current job, in minutes. The default value is **30**. For example, **30** indicates that the job is checked every 30 minutes to determine whether to scale in or out the resources used by the job. Note: This configuration is effective only when dynamic scaling is enabled.
flink.dli.job.cu.max	Initial CU value	Maximum number of CUs that can be used by the current job during dynamic scaling. If this parameter is not set, the default value is the initial total number of CUs of the job. Note: The value of this parameter cannot be smaller than the total number of CUs configured by the user. In addition, this parameter is effective only when dynamic scaling is enabled.
flink.dli.job.cu.min	2	Minimum number of CUs that can be used by the current job during dynamic scaling. The default value is **2**. Note: The value of this parameter cannot be greater than the total number of CUs configured by the user. In addition, this parameter is effective only when dynamic scaling is enabled.

8.5.4 Querying Logs for Flink Jobs

Scenario

DLI job buckets are used to store temporary data generated during DLI job running, such as job logs and results.

This section describes how to configure a bucket for DLI jobs on the DLI console and obtain Flink job logs.

Notes

- To avoid disordered job results, do not use the OBS bucket configured for DLI jobs for any other purposes.
- DLI jobs must be set and modified by the main account as IAM users do not have required permissions.
- You cannot view the logs for DLI jobs before configuring a bucket.
- You can configure lifecycle rules to periodically delete objects from buckets or change storage classes of objects.
- Exercise caution when modifying the job bucket, as it may result in the inability to retrieve historical data.

Prerequisites

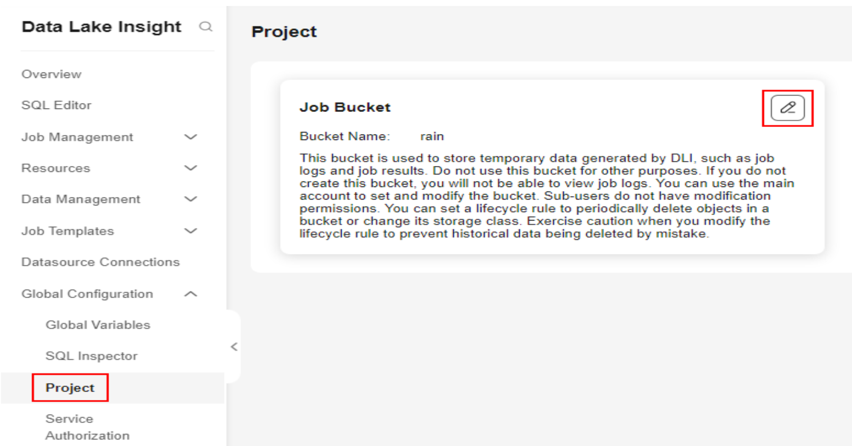
Before the configuration, create an OBS bucket or parallel file system (PFS). In big data scenarios, you are advised to create a PFS. PFS is a high-performance file system provided by OBS, with access latency in milliseconds. PFS can achieve a bandwidth performance of up to TB/s and millions of IOPS, which makes it ideal for processing high-performance computing (HPC) workloads.

For details about PFS, see "Parallel File System Feature Guide" in the *Object Storage Service User Guide*.

Configuring a Bucket for DLI Jobs

1. In the navigation pane of the DLI console, choose **Global Configuration** > **Project**.
2. On the **Project** page, click  next to **Job Bucket** to configure bucket information.

Figure 8-15 Project

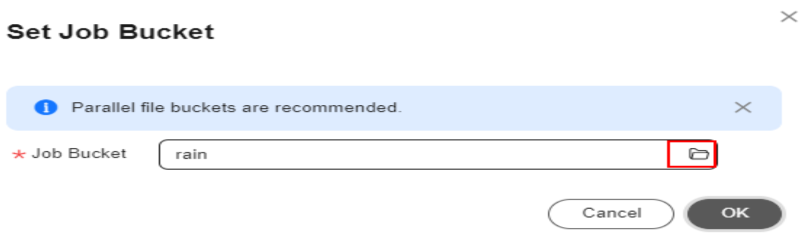


3. Click  to view available buckets.

4. In the displayed **OBS** dialog box, click the name of a bucket or search for and click a bucket name and then click **OK**. In the **Set Job Bucket** dialog box, click **OK**.

   Temporary data generated during DLI job running will be stored in the OBS bucket.

   Figure 8-16 Setting the job bucket



Viewing Commit Logs

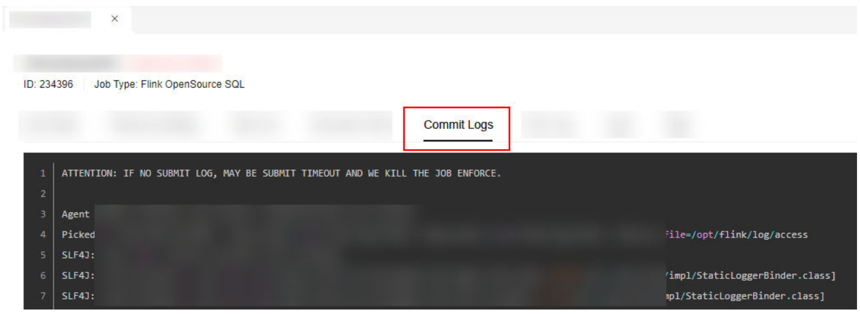
You can check commit logs to locate commit faults.

Step 1 In the navigation pane of the DLI console, choose **Job Management** > **Flink Jobs**.

Step 2 Click the name of the Flink job whose commit logs you want to check.

Step 3 Click the **Commit Logs** tab and check the job commit process.

Figure 8-17 Commit logs



----End

Viewing Run Logs

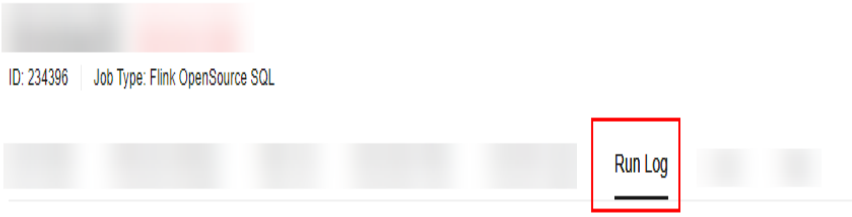
You can check run logs to locate job running faults.

Step 1 In the navigation pane of the DLI console, choose **Job Management** > **Flink Jobs**.

Step 2 Click the name of the Flink job whose commit logs you want to check.

Step 3 Click the **Run Log** tab and check the JobManager and TaskManager information of the running job.

Figure 8-18 Run logs



JobManager and TaskManager information is updated every minute. By default, the run logs generated in the last minute are displayed.

If you have configured an OBS bucket to store job logs, you can access it to download and check historical logs.

NOTE

For how to upload files to an OBS bucket, see **Uploading an Object** in the *Object Storage Service Getting Started*.

If the job is not running, you cannot check Task Manager information.

----End

8.5.5 Common Operations of Flink Jobs

After creating a job, you can manage it by performing various operations such as editing its basic information, starting or stopping it, and importing or exporting it.

Editing a Job

You can edit a created job, for example, by modifying the SQL statement, job name, job description, or job configurations.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 In the row where the job you want to edit locates, click **Edit** in the **Operation** column to switch to the editing page.

Step 3 Edit the job as required.

For details, see **Creating a Flink OpenSource SQL Job** and **Creating a Flink Jar Job**.

----End

Starting a Job

You can start a saved or stopped job.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Use either of the following methods to start jobs:

- Starting a single job

Select a job and click **Start** in the **Operation** column.

Alternatively, you can select the row where the job you want to start locates and click **Start** in the upper left of the job list.

- Batch starting jobs

  Select the rows where the jobs you want to start locate and click **Start** in the upper left of the job list.

After you click **Start**, the **Start Flink Jobs** page is displayed.

Step 3 On the **Start Flink Jobs** page, confirm the job information and price. If they are correct, click **Start Now**.

After a job is started, you can view the job execution result in the **Status** column.

----End

Stopping a Job

You can stop a job in the **Running** or **Submitting** state.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Stop a job using either of the following methods:

- Stopping a job

  Locate the row that contains the job to be stopped, click **More** in the **Operation** column, and select **Stop**.

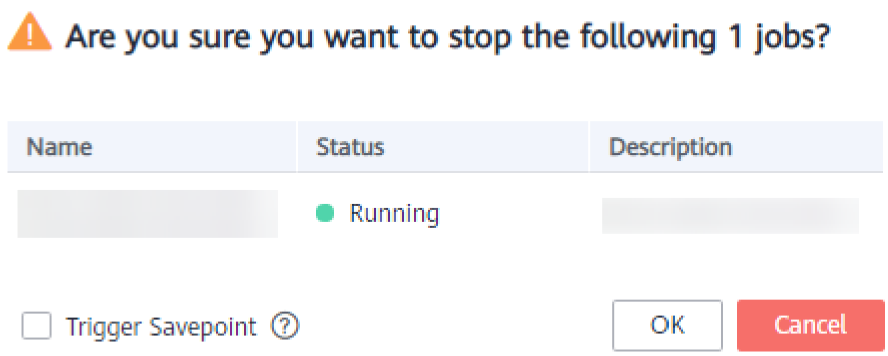
  Alternatively, you can select the row where the job you want to stop locates and click **Stop** in the upper left of the job list.

- Batch stopping jobs

  Locate the rows containing the jobs you want to stop and click **Stop** in the upper left of the job list.

Step 3 In the displayed **Stop Job** dialog box, click **OK** to stop the job.

Figure 8-19 Stopping a job



NOTE

- Before stopping a job, you can trigger a savepoint to save the job status information. When you start the job again, you can choose whether to restore the job from the savepoint.
- If you select **Trigger savepoint**, a savepoint is created. If **Trigger savepoint** is not selected, no savepoint is created. By default, the savepoint function is disabled.
- The lifecycle of a savepoint starts when the savepoint is triggered and stops the job, and ends when the job is restarted. The savepoint is automatically deleted after the job is restarted.

When a job is being stopped, the job status is displayed in the **Status** column of the job list. The details are as follows:

- **Stopping**: indicates that the job is being stopped.
- **Stopped**: indicates that the job is stopped successfully.
- **Stop failed**: indicates that the job failed to be stopped.

----End

Deleting a Job

If you do not need to use a job, perform the following operations to delete it. A deleted job cannot be restored. Therefore, exercise caution when deleting a job.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Perform either of the following methods to delete jobs:

- Deleting a single job

  Locate the row containing the job you want to delete and click **More > Delete** in the **Operation** column.

  Alternatively, you can select the row containing the job you want to delete and click **Delete** in the upper left of the job list.
- Deleting jobs in batches

  Select the rows containing the jobs you want to delete and click **Delete** in the upper left of the job list.

Step 3 Click **Yes**.

----End

Exporting a Job

You can export the created Flink jobs to an OBS bucket.

This mode is applicable to the scenario where a large number of jobs need to be created when you switch to another region, project, or user. In this case, you do not need to create a job. You only need to export the original job, log in to the system in a new region or project, or use a new user to import the job.

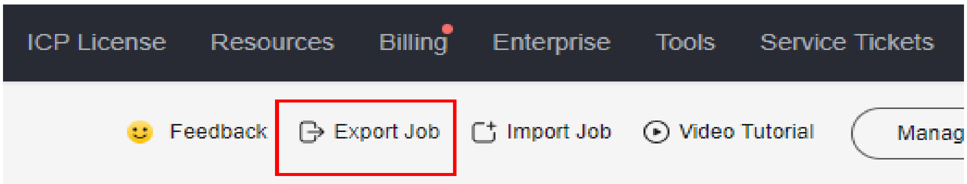
NOTE

When switching to another project or user, you need to grant permissions to the new project or user. For details, see **Configuring Flink Job Permissions**.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Click **Export Job** in the upper right corner. The **Export Job** dialog box is displayed.

Figure 8-20 Exporting a job



Step 3 Select the OBS bucket where the job is stored. Click **Next**.

Step 4 Select job information you want to export.

By default, configurations of all jobs are exported. You can enable the **Custom Export** function to export configurations of the desired jobs.

Step 5 Click **Confirm** to export the job.

----End

Importing a Job

You can import the Flink job configuration file stored in the OBS bucket to the **Flink Jobs** page of DLI.

This mode is applicable to the scenario where a large number of jobs need to be created when you switch to another region, project, or user. In this case, you do not need to create a job. You only need to export the original job, log in to the system in a new region or project, or use a new user to import the job.

To import a self-created job, use the job creation function.

For details, see **Creating a Flink OpenSource SQL Job** and **Creating a Flink Jar Job**.

NOTE

- When switching to another project or user, you need to grant permissions to the new project or user. For details, see **Configuring Flink Job Permissions**.
- Only jobs whose data format is the same as that of Flink jobs exported from DLI can be imported.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Click **Import Job** in the upper right corner. The **Import Job** dialog box is displayed.

Step 3 Select the complete OBS path of the job configuration file to be imported. Click **Next**.

Step 4 Configure the same-name job policy and click next. Click **Next**.

- Select **Overwrite job of the same name**. If the name of the job to be imported already exists, the existing job configuration will be overwritten and the job status switches to **Draft**.

- If **Overwrite job of the same name** is not selected and the name of the job to be imported already exists, the job will not be imported.

Step 5 Ensure that **Config File** and **Overwrite Same-Name Job** are correctly configured. Click **Confirm** to import the job.

----End

Modifying the Name and Description of a Flink Job

You can change the job name and description as required.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 In the **Operation** column of the job whose name and description need to be modified, choose **More > Modify Name and Description**. The **Modify Name and Description** dialog box is displayed. Change the name or modify the description of a job.

Step 3 Click **OK**.

----End

Triggering a Savepoint

Before stopping a job, you can trigger a savepoint to save the status information of your job. When you restart the job, you can choose whether to quickly recover it from the most recent savepoint.

Step 1 In the navigation pane of the DLI console, choose **Job Management** > **Flink Jobs**.

Step 2 Locate the job you want to stop, click **More** in the **Operation** column, and select **Trigger Savepoint**. In the displayed dialog box, select a save path.

Step 3 Click **OK**.

----End

NOTE

- You can click **Trigger Savepoint** for jobs in the **Running** status to save the job status.
- The lifecycle of a savepoint starts when the savepoint is triggered and stops the job, and ends when the job is restarted. The savepoint is automatically deleted after the job is restarted.

Importing to a Savepoint

Flink jobs can be restored based on imported savepoints.

Step 1 In the navigation pane of the DLI console, choose **Job Management** > **Flink Jobs**.

Step 2 Locate the job you want to stop, click **More** in the **Operation** column, and select **Import Savepoint**. In the displayed dialog box, select a save path.

Step 3 Click **OK**.

----End

Runtime Configuration

You can configure job exception alarms and restart options by selecting **Runtime Configuration**.

NOTE

This configuration is only available for Flink OpenSource SQL jobs and Flink Jar jobs.

1. Locate the desired Flink job, click **More** in the **Operation** column, and select **Runtime Configuration**.
2. In the **Runtime Configuration** dialog box, set the following parameters:

Figure 8-21 Runtime configuration

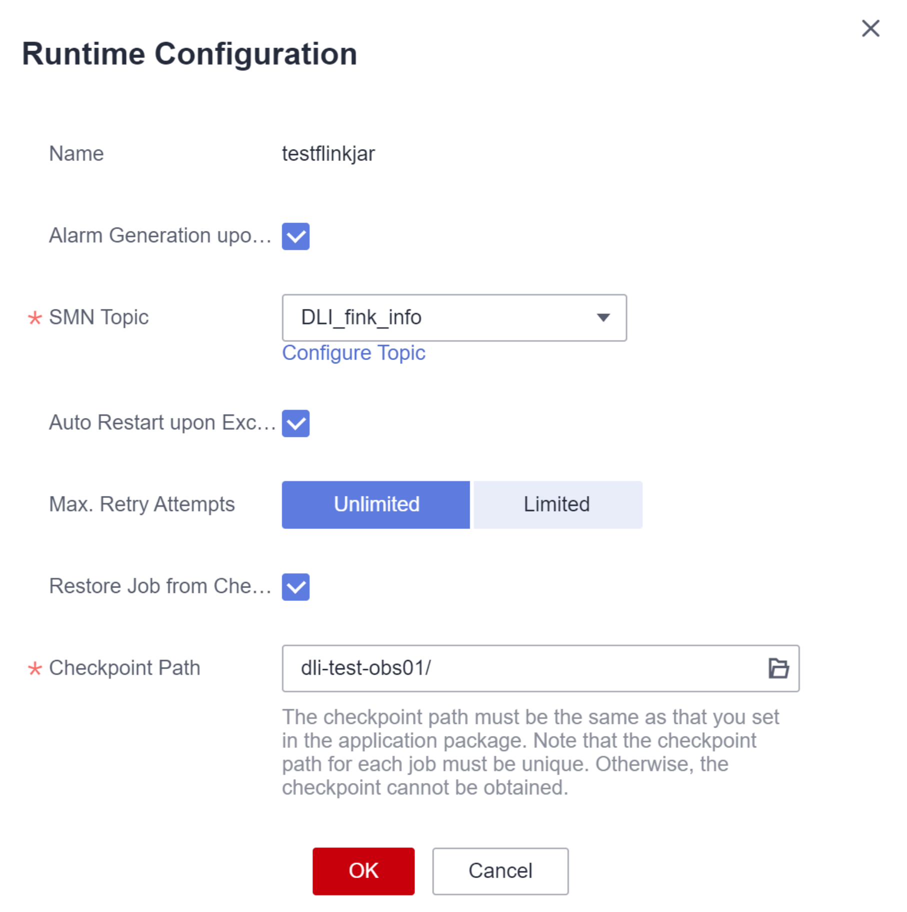


Table 8-13 Running parameters

Parameter	Description
Name	Job name.

Parameter	Description
Alarm Generation upon Job Exception	Whether to report job exceptions, for example, abnormal job running or exceptions due to an insufficient balance, to users via SMS or email. If this option is selected, you need to set the following parameters: **SMN Topic** Select a custom SMN topic. For how to create a custom SMN topic, see **Creating a Topic**.
Auto Restart upon Exception	Whether to enable automatic restart. If this function is enabled, any job that has become abnormal will be automatically restarted. If this option is selected, you need to set the following parameters: ● **Max. Retry Attempts**: maximum number of retries upon an exception. The unit is times/hour. – **Unlimited**: The number of retries is unlimited. – **Limited**: The number of retries is user-defined. ● **Restore Job from Checkpoint**: Restore the job from the saved checkpoint. **NOTE** This parameter cannot be configured for Flink SQL jobs or Flink OpenSource SQL jobs. If this parameter is selected, you need to set **Checkpoint Path** for Flink Jar jobs. **Checkpoint Path**: Select the checkpoint saving path. The checkpoint path must be the same as that you set in the application package. Note that the checkpoint path for each job must be unique. Otherwise, the checkpoint cannot be obtained.

8.6 Managing Flink Job Templates

Flink templates include sample templates and custom templates. You can modify an existing sample template to meet the actual job logic requirements and save time for editing SQL statements. You can also customize a job template based on your habits and methods so that you can directly invoke or modify the template in later jobs.

Flink template management provides the following functions:

- **Flink SQL Sample Template**
- **Flink OpenSource SQL Sample Template**
- **Custom Templates**
- **Creating a Template**

- Creating a Job Based on a Template
- Modifying a Template
- Deleting a Template

Flink SQL Sample Template

The template list displays existing sample templates for Flink SQL jobs. **Table 1** describes the parameters in the template list.

The scenarios of sample templates can be different, which are subject to the console.

Table 8-14 Parameters in the Flink SQL sample template list

Parameter	Description
Name	Template name. Enter 1 to 64 characters. Only letters, numbers, hyphens (-), and underscores (_) are allowed.
Description	Description of a template. It contains 0 to 512 characters.
Operation	**Create Job**: Create a job directly by using the template. After a job is created, the system switches to the **Edit** page under **Job Management**.

Flink OpenSource SQL Sample Template

The template list displays existing sample templates for Flink SQL OpenSource jobs. **Table 1** describes the parameters in the template list.

Table 8-15 Parameters in the sample template list for Flink OpenSource SQL jobs

Parameter	Description
Name	Template name. Enter 1 to 64 characters. Only letters, numbers, hyphens (-), and underscores (_) are allowed.
Description	Description of a template. It contains 0 to 512 characters.
Operation	**Create Job**: Create a job directly by using the template. After a job is created, the system switches to the **Edit** page under **Job Management**.

The existing sample templates apply to the following scenarios:

- Create a wide table of order information from the dimension table of address information.
- Generate statistics on indicators such as the daily transaction amount, number of orders, and number of paid users in real time.

- Generate statistics on offerings with the highest real-time click-through rate.

Custom Templates

The custom template list displays all Jar job templates. **Table 1** describes parameters in the custom template list.

Table 8-16 Parameters in the custom template list

Parameter	Description
Name	Template name. Enter 1 to 64 characters. Only letters, numbers, hyphens (-), and underscores (_) are allowed.
Description	Description of a template. It contains 0 to 512 characters.
Created	Time when a template is created.
Updated	Latest time when a template is modified.
Operation	• **Edit**: Modify a template that has been created. • **Create Job**: Create a job directly by using the template. After a job is created, the system switches to the **Edit** page under **Job Management**. • **More**: – **Delete**: Delete a created template. – **Tags**: View or add tags.

Creating a Template

You can create a template using any of the following methods:

- Creating a template on the **Template Management** page
  a. In the left navigation pane of the DLI management console, choose **Job Templates** > **Flink Templates**.
  b. Click **Create Template** in the upper right corner of the page. The **Create Template** dialog box is displayed.
  c. Specify **Name** and **Description**.

Figure 8-22 Creating a Flink template

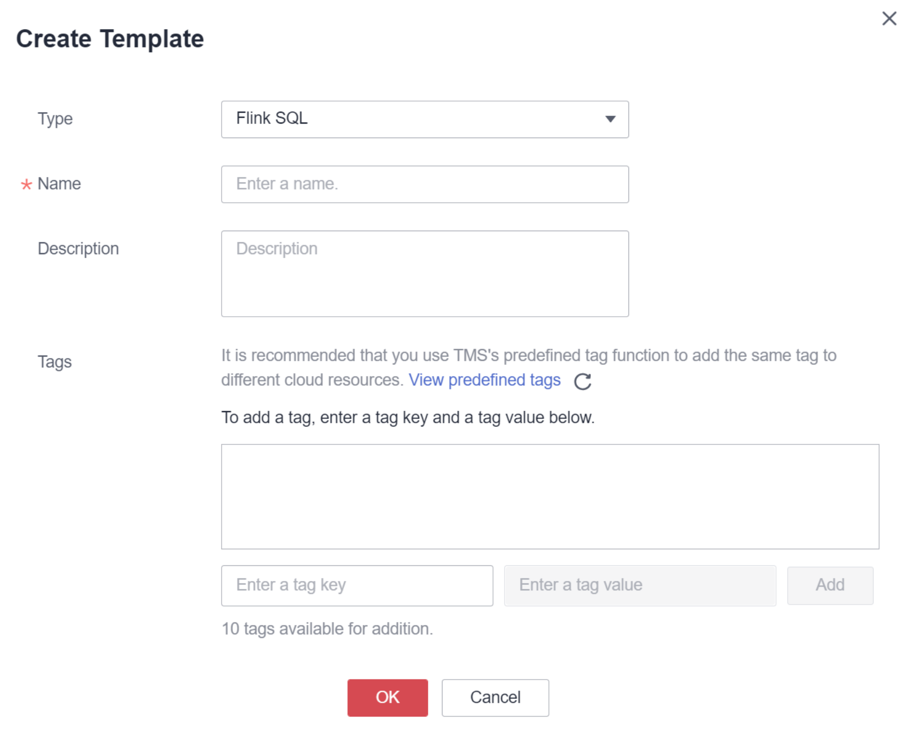


Table 8-17 Template parameters

Parameter	Description
Type	Template type ● Flink SQL job template ● Flink OpenSource SQL job template
Name	Template name. Enter 1 to 64 characters. Only letters, numbers, hyphens (-), and underscores (_) are allowed. **NOTE** The template name must be unique.
Description	Description of a template. It contains 0 to 512 characters.

Parameter	Description
Tags	Tags used to identify cloud resources. A tag includes the tag key and tag value. If you want to use the same tag to identify multiple cloud resources, that is, to select the same tag from the drop-down list box for all services, you are advised to create predefined tags on the Tag Management Service (TMS). If your organization has configured tag policies for DLI, add tags to resources based on the policies. If a tag does not comply with the tag policies, resource creation may fail. Contact your organization administrator to learn more about tag policies. For details, see **Tag Management Service User Guide**. **NOTE** ● A maximum of 20 tags can be added. ● Only one tag value can be added to a tag key. ● The key name in each resource must be unique. ● Tag key: Enter a tag key name in the text box. **NOTE** A tag key can contain a maximum of 128 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed, but the value cannot start or end with a space or start with **_sys_**. ● Tag value: Enter a tag value in the text box. **NOTE** A tag value can contain a maximum of 255 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed. The value cannot start or end with a space.

d. Click **OK** to enter the editing page.

The **Table 8-18** describes the parameters on the template editing page.

Table 8-18 Template parameters

Parameter	Description
Name	You can modify the template name.
Description	You can modify the template description.
Saving Mode	● **Save Here**: Save the modification to the current template. ● **Save as New**: Save the modification as a new template.
SQL statement editing area	In the area, you can enter detailed SQL statements to implement business logic. For how to compile SQL statements, see **Data Lake Insight SQL Syntax Reference**.

Parameter	Description
Save	Save the modifications.
Create Job	Use the current template to create a job.
Format	Format SQL statements. After SQL statements are formatted, you need to compile SQL statements again.
Theme Settings	Change the font size, word wrap, and page style (black or white background).

e. In the SQL statement editing area, enter SQL statements to implement service logic. For how to compile SQL statements, see **Data Lake Insight SQL Syntax Reference**.

f. After the SQL statement is edited, click **Save** in the upper right corner to complete the template creation.

g. (Optional) If you do not need to modify the template, click **Create Job** in the upper right corner to create a job based on the current template. For how to create a job, see **Creating a Flink Jar Job**.

- Creating a template based on an existing job template
  a. In the left navigation pane of the DLI management console, choose **Job Templates** > **Flink Templates**. Click the **Custom Templates** tab.
  b. In the row where the desired template is located in the custom template list, click **Edit** under **Operation** to enter the **Edit** page.
  c. After the modification is complete, set **Saving Mode** to **Save as New**.
  d. Click **Save** in the upper right corner to save the template as a new one.
- Creating a template using a created job
  a. In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.
  b. Click **Create Job** in the upper right corner. The **Create Job** page is displayed.
  c. Specify parameters as required.
  d. Click **OK** to enter the editing page.
  e. After the SQL statement is compiled, click **Set as Template**.
  f. In the **Set as Template** dialog box that is displayed, specify **Name** and **Description** and click **OK**.
- Creating a template based on the existing job
  a. In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.
  b. In the job list, locate the row where the job that you want to set as a template resides, and click **Edit** in the **Operation** column.
  c. After the SQL statement is compiled, click **Set as Template**.
  d. In the **Set as Template** dialog box that is displayed, specify **Name** and **Description** and click **OK**.

Creating a Job Based on a Template

You can create jobs based on sample templates or custom templates.

1. In the left navigation pane of the DLI management console, choose **Job Templates** > **Flink Templates**.
2. In the sample template list, click **Create Job** in the **Operation** column of the target template. For how to create a job, see **Creating a Flink OpenSource SQL Job** and **Creating a Flink Jar Job**.

Modifying a Template

After creating a custom template, you can modify it as required. The sample template cannot be modified, but you can view the template details.

1. In the left navigation pane of the DLI management console, choose **Job Templates** > **Flink Templates**. Click the **Custom Templates** tab.
2. In the row where the template you want to modify is located in the custom template list, click **Edit** in the **Operation** column to enter the **Edit** page.
3. In the SQL statement editing area, modify the SQL statements as required.
4. Set **Saving Mode** to **Save Here**.
5. Click **Save** in the upper right corner to save the modification.

Deleting a Template

You can delete a custom template as required. The sample templates cannot be deleted. Deleted templates cannot be restored. Exercise caution when performing this operation.

1. In the left navigation pane of the DLI management console, choose **Job Templates** > **Flink Templates**. Click the **Custom Templates** tab.
2. In the custom template list, select the templates you want to delete and click **Delete** in the upper left of the custom template list.

   Alternatively, you can delete a template by performing the following operations: In the custom template list, locate the row where the template you want to delete resides, and click **More** > **Delete** in the **Operation** column.
3. In the displayed dialog box, click **Yes**.

8.7 Adding Tags to a Flink Job

A tag is a key-value pair customized by users and used to identify cloud resources. It helps users to classify and search for cloud resources. A tag consists of a tag key and a tag value.

DLI allows you to add tags to Flink jobs. You can add tags to Flink jobs to identify information such as the project name, service type, and background. If you use tags in other cloud services, you are advised to create the same tag key-value pairs for cloud resources used by the same business to keep consistency.

DLI supports the following two types of tags:

- Resource tags: indicate non-global tags created on DLI.

- Predefined tags: global tags created on Tag Management Service (TMS).
  For more information about predefined tags, see **Tag Management Service User Guide**.

If your organization has configured tag policies for DLI, add tags to resources based on the policies. If a tag does not comply with the tag policies, resource creation may fail. Contact your organization administrator to learn more about tag policies.

This section includes the following content:

- **Managing a Job Tag**
- **Searching for a Job by Tag**

Managing a Job Tag

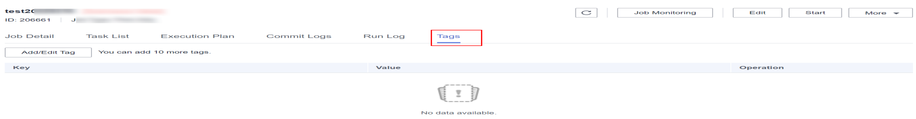
DLI allows you to add, modify, or delete tags for jobs.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Click the name of the job to be viewed. The **Job Details** page is displayed.

Step 3 Click **Tags** to display the tag information about the current job.

Figure 8-23 Managing a job tag



Step 4 Click **Add/Edit Tag** to open to the **Add/Edit Tag** dialog box.

Step 5 Configure the tag parameters in the **Add/Edit Tag** dialog box.

Figure 8-24 Adding a tag

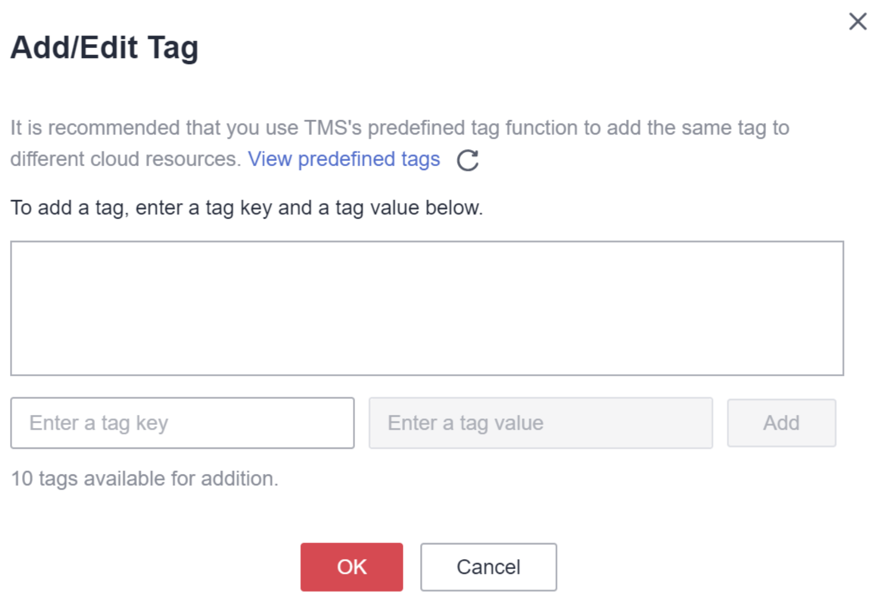


Table 8-19 Tag parameters

Parameter	Description
Tag key	You can perform the following operations: ● Click the text box and select a predefined tag key from the drop-down list. To add a predefined tag, you need to create one on TMS and then select it from the **Tag key** drop-down list. You can click **View predefined tags** to go to the **Predefined Tags** page of the TMS console. Then, click **Create Tag** in the upper corner of the page to create a predefined tag. For details, see **Creating Predefined Tags** in *Tag Management Service User Guide*. ● Enter a tag key in the text box. **NOTE** A tag key can contain a maximum of 128 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed, but the value cannot start or end with a space or start with **_sys_**.
Tag value	You can perform the following operations: ● Click the text box and select a predefined tag value from the drop-down list. ● Enter a tag value in the text box. **NOTE** A tag value can contain a maximum of 255 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed. The value cannot start or end with a space.

NOTE

- A maximum of 20 tags can be added.
- Only one tag value can be added to a tag key.
- The key name in each resource must be unique.

Step 6 Click **OK**.

Step 7 (Optional) In the tag list, locate the row where the tag you want to delete resides, click **Delete** in the **Operation** column to delete the tag.

----End

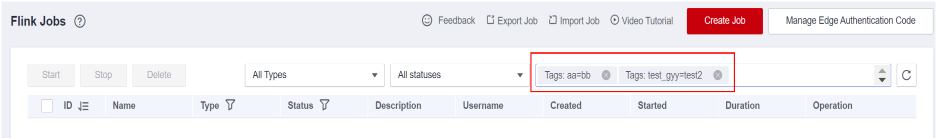
Searching for a Job by Tag

If tags have been added to a job, you can search for the job by setting tag filtering conditions to quickly find it.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 In the upper right corner of the page, click the search box and select **Tags**.

Figure 8-25 Searching for jobs by tag



Step 3 Choose a tag key and value as prompted. If no tag key or value is available, create a tag for the job. For details, see **Managing a Job Tag**.

Step 4 Choose other tags to generate a tag combination for job search.

Step 5 Click search icon. The target job will be displayed in the job list.

----End

9 Submitting a Spark Job Using DLI

9.1 Creating a Spark Job

DLI provides fully-managed Spark computing services by allowing you to execute Spark jobs.

On the **Overview** page, click **Create Job** in the upper right corner of the **Spark Jobs** tab or click **Create Job** in the upper right corner of the **Spark Jobs** page. The Spark job editing page is displayed.

On the Spark job editing page, a message is displayed, indicating that a temporary DLI data bucket will be created. The created bucket is used to store temporary data generated by DLI, such as job logs and job results. You cannot view job logs if you choose not to create the bucket. You can **configure a lifecycle rule** to periodically delete objects in a bucket or transit objects between different storage classes.The bucket will be created and the default bucket name is used.

If you do not need to create a DLI temporary data bucket and do not want to receive this message, select **Do not show again** and click **Cancel**.

Prerequisites

- You have uploaded the dependencies to the corresponding OBS bucket on the **Data Management > Package Management** page. For details, see **Creating a DLI Package**.
- Before creating a Spark job to access other external data sources, such as OpenTSDB, HBase, Kafka, GaussDB(DWS), RDS, CSS, CloudTable, DCS Redis, and DDS, you need to create a datasource connection to enable the network between the job running queue and external data sources.
  - For details about the external data sources that can be accessed by Spark jobs, see **Common Development Methods for DLI Cross-Source Analysis**.
  - For how to create a datasource connection, see **Configuring the Network Connection Between DLI and Data Sources (Enhanced Datasource Connection)**.

    On the **Resources** > **Queue Management** page, locate the queue you have created, click **More** in the **Operation** column, and select **Test**

Address Connectivity to check if the network connection between the queue and the data source is normal. For details, see **Testing Address Connectivity**.

Procedure

1. In the left navigation pane of the DLI management console, choose **Job Management** > **Spark Jobs**. The **Spark Jobs** page is displayed.

   Click **Create Job** in the upper right corner. In the job editing window, you can set parameters in **Fill Form** mode or **Write API** mode.

   The following uses the **Fill Form** as an example. In **Write API** mode, refer to the **Data Lake Insight API Reference** for parameter settings.

2. Select a queue.

   Select the queue you want to use from the drop-down list box.

3. Select a Spark version. Select a supported Spark version from the drop-down list. The latest version is recommended.

   NOTE

   You are not advised to use Spark/Flink engines of different versions for a long time.

   - Doing so can lead to code incompatibility, which can negatively impact job execution efficiency.
   - Doing so may result in job execution failures due to conflicts in dependencies. Jobs rely on specific versions of libraries or components.

4. Configure the job.

   Configure job parameters by referring to **Table 9-1**.

Figure 9-1 Spark job configuration

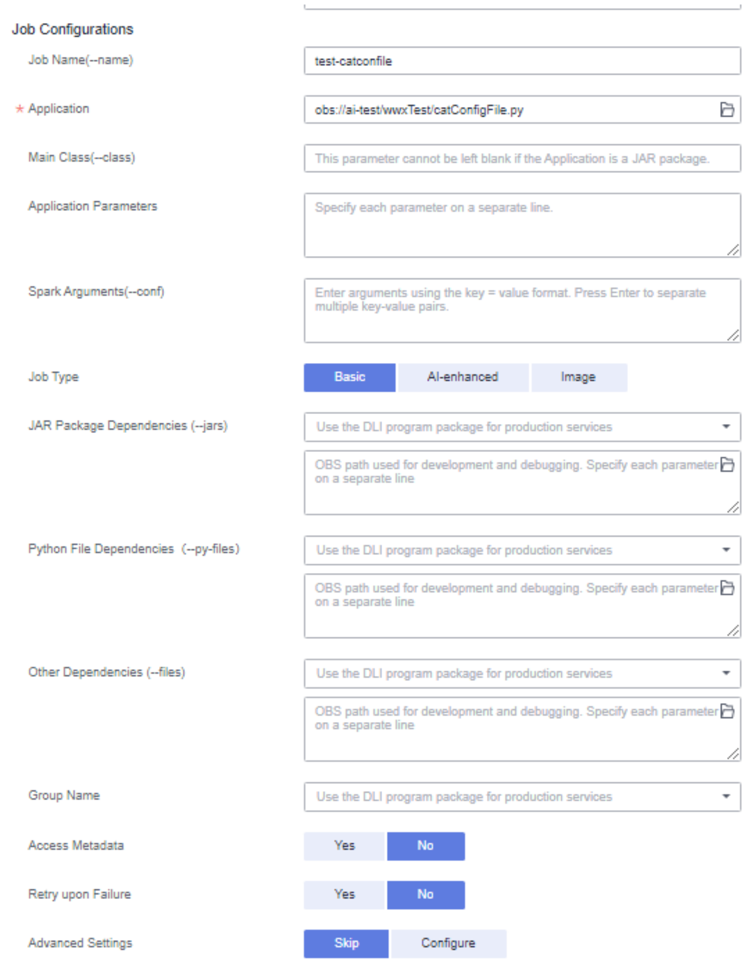


Table 9-1 Job configuration parameters

Parameter	Description
Job Name (--name)	Set a job name.
Application	Select the package to be executed. The value can be **.jar** or **.py**. There are the following ways to manage JAR files: ● Upload packages to OBS: Upload Jar packages to an OBS bucket in advance and select the corresponding OBS path. ● Upload packages to DLI: Upload JAR files to an OBS bucket in advance and create a package on the **Data Management** > **Package Management** page of the DLI management console. For details, see **Creating a DLI Package**. For Spark 3.3.*x* or later, you are advised to select packages in OBS paths.

Parameter	Description
Main Class (--class)	Enter the name of the main class. When the application type is **.jar**, the main class name cannot be empty.
Application Parameters	User-defined parameters. Separate multiple parameters by **Enter**. These parameters can be replaced with global variables. For example, if you create a global variable **batch_num** on the **Global Configuration** > **Global Variables** page, you can use **{{batch_num}}** to replace a parameter with this variable after the job is submitted.
Spark Arguments (--conf)	Enter a parameter in the format of **key=value**. Press Enter to separate multiple key-value pairs. These parameters can be replaced with global variables. For example, if you create a global variable **custom_class** on the **Global Configuration** > **Global Variables** page, you can use **"spark.sql.catalog"={{custom_class}}** to replace a parameter with this variable after the job is submitted. **NOTE** • The JVM garbage collection algorithm cannot be customized for Spark jobs. • If the Spark version is **3.1.1**, configure **Spark parameters (--conf)** to select a dependent module. For details about the example configuration, see **Table 9-2**.
Job Type	Set this parameter when you select a CCE queue. Type of the Spark image used by a job. The options are as follows: • **Basic**: Base images provided by DLI. Select this option for non-AI jobs. • **Image**: Custom Spark images. Select an existing image name and version on SWR.
JAR Package Dependencies (--jars)	JAR file on which the Spark job depends. You can enter the JAR file name or the OBS path of the JAR file in the format of **obs://***Bucket name***/***Folder path***/***JAR file name*.
Python File Dependencies (--py-files)	py-files on which the Spark job depends. You can enter the Python file name or the corresponding OBS path of the Python file. The format is as follows: **obs://*Bucket name*/*Folder name*/*File name***.
Other Dependencies (--files)	Other files on which the Spark job depends. You can enter the name of the dependency file or the corresponding OBS path of the dependency file. The format is as follows: **obs://*Bucket name*/*Folder name*/*File name***.
Group Name	If you select a group when creating a package, you can select all the packages and files in the group. For how to create a package, see **Creating a DLI Package**.

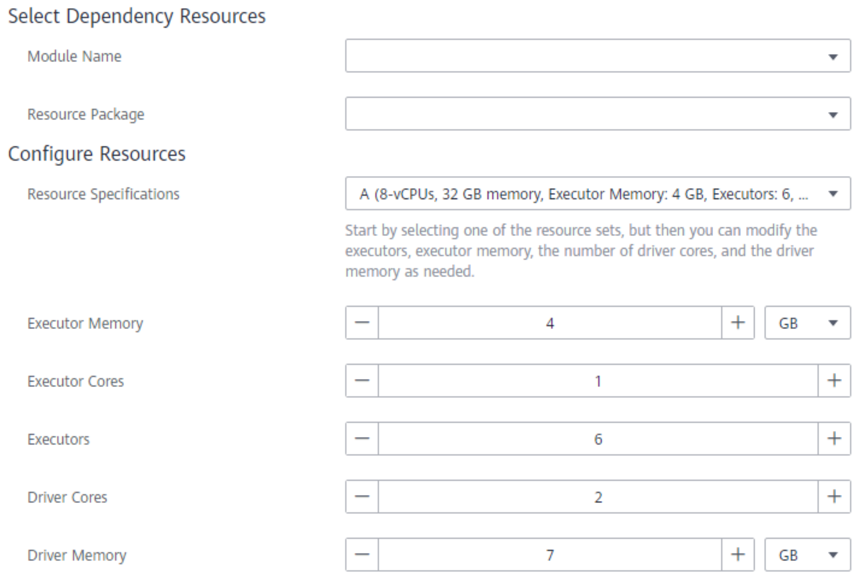
Parameter	Description
Access Metadata	Whether to access metadata through Spark jobs. For details, see the **Data Lake Insight Developer Guide**.
Retry upon Failure	Indicates whether to retry a failed job. If you select **Yes**, you need to set the following parameters: **Maximum Retries**: Maximum number of retry times. The maximum value is **100**.
Advanced Settings	● Skip ● Configure – **Select Dependency Resources**: For details about the parameters, see **Table 9-3**. – **Configure Resources**: For details about the parameters, see **Table 9-4**.

Table 9-2 Spark Parameter (--conf) configuration

Datasource	Example Value
CSS	spark.driver.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/css/* spark.executor.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/css/*
GaussDB(DWS)	spark.driver.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/dws/* spark.executor.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/dws/*
HBase	spark.driver.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/hbase/* spark.executor.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/hbase/*
OpenTSDB	park.driver.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/opentsdb/* spark.executor.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/opentsdb/*
RDS	spark.driver.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/rds/* spark.executor.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/rds/*

Datasource	Example Value
Redis	spark.driver.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/redis/* spark.executor.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/redis/*
Mongo	spark.driver.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/mongo/* spark.executor.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/mongo/*

Figure 9-2 Creating a Spark job - advanced settings



5. Set the following parameters in advanced settings:
   - **Select Dependency Resources**: For details about the parameters, see **Table 9-3**.
   - **Configure Resources**: For details about the parameters, see **Table 9-4**.

NOTE

The parallelism degree of Spark resources is jointly determined by the number of Executors and the number of Executor CPU cores.

Maximum number of tasks that can be concurrently executed = Number of Executors x Number of Executor CPU cores

You can properly plan compute resource specifications based on the compute CUs of the queue you have purchased.

Note that Spark tasks need to be jointly executed by multiple roles, such as **driver** and **executor**. So, the number of executors multiplied by the number of executor CPU cores must be less than the number of compute CUs of the queue to prevent other roles from failing to start Spark tasks. For more information about roles for Spark tasks, see **Apache Spark**.

Calculation formula for Spark job parameters:

- CUs = Driver Cores + Executors x Executor Cores
- Memory = Driver Memory + (Executors x Executor Memory)

Table 9-3 Parameters for selecting dependency resources

Parameter	Description
modules	If the Spark version is **3.1.1**, you do not need to select a module. Configure **Spark parameters (--conf)**. Dependency modules provided by DLI for executing datasource connection jobs. To access different services, you need to select different modules. • CloudTable/MRS HBase: sys.datasource.hbase • DDS: sys.datasource.mongo • CloudTable/MRS OpenTSDB: sys.datasource.opentsdb • DWS: sys.datasource.dws • RDS MySQL: sys.datasource.rds • RDS PostGre: sys.datasource.rds • DCS: sys.datasource.redis • CSS: sys.datasource.css DLI internal modules include: • sys.res.dli-v2 • sys.res.dli • sys.datasource.dli-inner-table
Resource Package	JAR package on which the Spark job depends.

Table 9-4 Resource specification parameters

Parameter	Description
Resource Specifications	Select a resource specification from the drop-down list box. The system provides three resource specification options for you to choose from. Resource specifications involve the following parameters: ● Executor Memory ● Executor Cores ● Executors ● Driver Cores ● Driver Memory If modified, your modified settings of the items are used.
Executor Memory	Customize the configuration item based on the selected resource specifications. Memory of each Executor. It is recommended that the ratio of Executor CPU cores to Executor memory be 1:4.
Executor Cores	Number of CPU cores of each Executor applied for by Spark jobs, which determines the capability of each Executor to execute tasks concurrently.
Executors	Number of Executors applied for by a Spark job
Driver Cores	Number of CPU cores of the driver
Driver Memory	Driver memory size. It is recommended that the ratio of the number of driver CPU cores to the driver memory be 1:4.

6. Click **Execute** in the upper right corner of the Spark job editing page.

   After the message "Batch processing job submitted successfully" is displayed, you can view the status and logs of the submitted job on the **Spark Jobs** page.

9.2 Setting the Priority for a Spark Job

Scenario

In actual job running, it is necessary to prioritize and ensure the normal running of important and urgent tasks due to their varying levels of importance and urgency. This requires providing the necessary compute resources for their normal operations.

DLI offers a feature to set job priorities for each Spark job, which prioritizes the allocation of compute resources to higher priority jobs when resources are limited.

NOTE

You can set the priority for Spark 2.4.5 or later jobs.

Notes

- You can assign a priority level of 1 to 10 for each job, with a larger value indicating a higher priority. Compute resources are preferentially allocated to high-priority jobs. That is, if compute resources required for high-priority jobs are insufficient, compute resources for low-priority jobs are reduced.
- Spark jobs running on a general-purpose queue have a default priority level of 3.
- To change the priority for a job, you must first stop the job, change the priority level, and then submit the job for the modification to take effect.

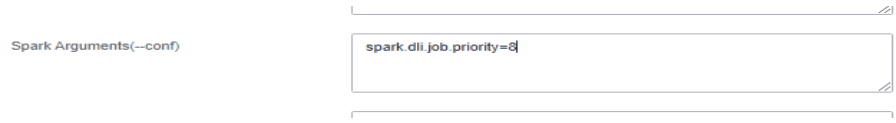
Procedure

Enter the following statement in the **Spark Arguments(--conf)** text box, where *x* indicates the priority value:

```
spark.dli.job.priority=x
```

1. Log in to the DLI management console.
2. In the navigation pane on the left, choose **Job Management** > **Spark Jobs**.
3. Select the job for which you want to set the priority and click **Edit** in the **Operation** column.
4. Set the **spark.dli.job.priority** parameter in the **Spark Arguments(--conf)** text box.

Figure 9-3 Example configuration for a Spark job



9.3 Querying Logs for Spark Jobs

Scenario

DLI job buckets are used to store temporary data generated during DLI job running, such as job logs and results.

This section describes how to configure a bucket for DLI jobs on the DLI console and obtain Spark job logs.

Notes

- To avoid disordered job results, do not use the OBS bucket configured for DLI jobs for any other purposes.
- DLI jobs must be set and modified by the main account as IAM users do not have required permissions.

- You cannot view the logs for DLI jobs before configuring a bucket.
- You can configure lifecycle rules to periodically delete objects from buckets or change storage classes of objects.
- Exercise caution when modifying the job bucket, as it may result in the inability to retrieve historical data.
- Spark log splitting rules:
  - Split by size: By default, each log file has a maximum size of 128 MB.
  - Split by time: A new log file is automatically created every hour.

Prerequisites

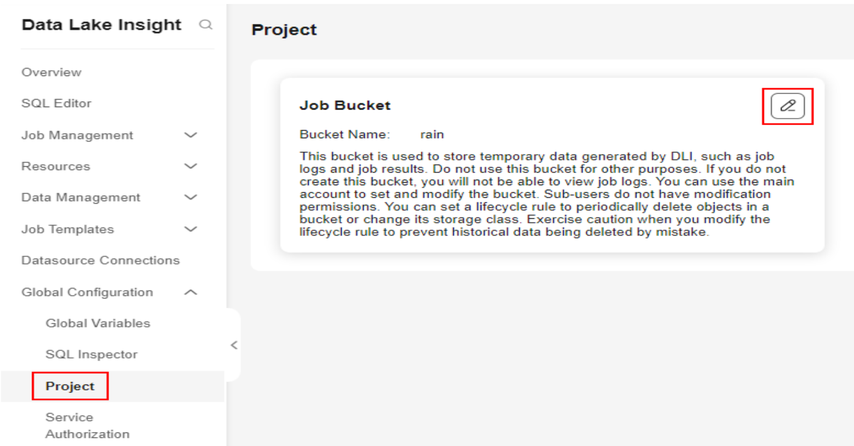
Before the configuration, create an OBS bucket or parallel file system (PFS). In big data scenarios, you are advised to create a PFS. PFS is a high-performance file system provided by OBS, with access latency in milliseconds. PFS can achieve a bandwidth performance of up to TB/s and millions of IOPS, which makes it ideal for processing high-performance computing (HPC) workloads.

For details about PFS, see "Parallel File System Feature Guide" in the *Object Storage Service User Guide*.

Configuring a Bucket for DLI Jobs

1. In the navigation pane of the DLI console, choose **Global Configuration** > **Project**.
2. On the **Project** page, click next to **Job Bucket** to configure bucket information.

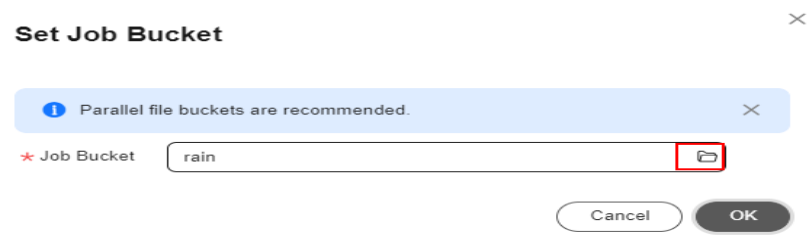
   Figure 9-4 Project

3. Click to view available buckets.
4. In the displayed **OBS** dialog box, click the name of a bucket or search for and click a bucket name and then click **OK**. In the **Set Job Bucket** dialog box, click **OK**.

   Temporary data generated during DLI job running will be stored in the OBS bucket.

Figure 9-5 Setting the job bucket

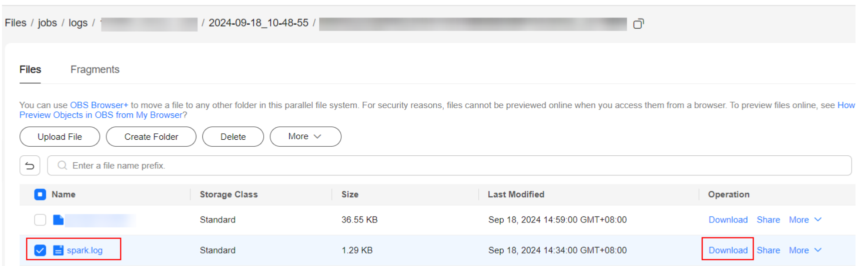


Querying Logs for Spark Jobs

1. Log in to the DLI console. In the navigation pane on the left, choose **Job Management** > **Spark Jobs**.
2. Select the Spark job whose jobs you want to query, click **More** in the **Operation** column, and select **View Log**.

   The system automatically switches to the log path of the DLI job bucket.
3. On the **Files** tab, select the log file of the desired date and time and click **Download** in the **Operation** column to download the file to your local host.

Figure 9-6 Downloading Spark job logs



9.4 Managing Spark Jobs

Viewing Basic Information

On the **Overview** page, click **Spark Jobs** to go to the SQL job management page. Alternatively, you can click **Job Management** > **Spark Jobs**. The page displays all Spark jobs. If there are a large number of jobs, they will be displayed on multiple pages. DLI allows you to view jobs in all statuses.

Table 9-5 Job management parameters

Parameter	Description
Job ID	ID of a submitted Spark job, which is generated by the system by default.
Name	Name of a submitted Spark job.
Queues	Queue where the submitted Spark job runs
Username	Name of the user who executed the Spark job

Parameter	Description
Status	Job status. The following values are available: ● **Starting**: The job is being started. ● **Running**: The job is being executed. ● **Failed**: The session has exited. ● **Finished**: The session is successfully executed. ● **Restoring**: The job is being restored.
Created	Time when a job is created. Jobs can be displayed in ascending or descending order of the job creation time.
Last Modified	Time when a job is completed.
Operation	● **Edit**: You can modify the current job configuration and re-execute the job. ● **SparkUI**: After you click this button, the Spark job execution page is displayed. **NOTE** ● The SparkUI page cannot be viewed for jobs in the **Starting** state. ● Currently, only the latest 100 job information records are displayed on the SparkUI of DLI. ● **Terminate Job**: Cancel a job that is being started or running. ● **Re-execute**: Run the job again. ● **Archive Log**: Save job logs to the temporary bucket created by DLI. ● **Commit Log**: View the logs of submitted jobs. ● **Driver Log**: View the logs of running jobs.

Re-executing a Job

On the **Spark Jobs** page, click **Edit** in the **Operation** column of the job. On the Spark job creation page that is displayed, modify parameters as required and execute the job.

Searching for a Job

On the **Spark Jobs** page, select **Status** or **Queues**. The system displays the jobs that meet the filter condition in the job list.

Terminating a Job

On the **Spark Jobs** page, choose **More** > **Terminate Job** in the **Operation** column of the job that you want to stop.

9.5 Managing Spark Job Templates

Scenario

You can modify a sample template to meet the Spark job requirements, saving time for editing SQL statements.

Currently, the cloud platform does not provide preset Spark templates. You can customize Spark job templates. This section describes how to create a Spark job template.

Creating a Management Job Template

To create a Spark job template, you can save a Spark job information as a template.

1. In the left navigation pane of the DLI management console, choose **Job Templates** > **Spark Templates**. The **Spark Templates** page is displayed.
2. Configure job parameters by referring to **Creating a Spark Job**.
3. After you finish editing the job, click **Save as Template**.
4. Enter the template name and other information as you need.
5. Set a template group for future management.
6. Click **OK**.

10 Using Cloud Eye to Monitor DLI

Description

This section describes metrics reported by DLI to Cloud Eye as well as their namespaces and dimensions. You can use the management console or APIs provided by Cloud Eye to query the metrics of the monitored object and alarms generated for DLI.

Namespace

SYS.DLI

Metric

Table 10-1 DLI metrics

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
queue_cu_num	Queue CU Usage	Displays the number of CUs applied by the user queue	≥ 0	Queues	5 minutes
queue_job_launching_num	Number of Jobs Being Submitted	Displays the number of jobs in the Submitting state in the user queue.	≥ 0	Queues	5 minutes

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
queue_job_running_num	Number of Running Jobs	Displays the number of running jobs in the user queue.	≥ 0	Queues	5 minutes
queue_job_succeed_num	Number of Finished Jobs	Displays the number of completed jobs in the user queue.	≥ 0	Queues	5 minutes
queue_job_failed_num	Failed Jobs	Displays the number of failed jobs in the user queue.	≥ 0	Queues	5 minutes
queue_job_cancelled_num	Number of Canceled Jobs	Displays the number of canceled jobs in the user queue.	≥ 0	Queues	5 minutes
queue_alloc_cu_num	Allocated CUs (queue)	Displays the CU allocation for user queues.	≥ 0	Queues	5 minutes
queue_min_cu_num	Minimum CUs for Queue	Displays the minimum number of CUs for a user queue.	≥ 0	Queues	5 minutes

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
queue_max_cu_num	Maximum CUs for Queue	Displays the maximum number of CUs for a user queue.	≥ 0	Queues	5 minutes
queue_priority	Queue Priority	Displays the priority of a user queue.	1–100	Queues	5 minutes
queue_cpu_usage	Queue CPU Usage	Displays the CPU usage of user queues. This metric only applies to standard queues.	0–100	Queues	5 minutes
queue_disk_usage	Queue Disk Usage	Displays the disk usage of user queues. This metric only applies to standard queues.	0–100	Queues	5 minutes
queue_disk_used	Max Disk Usage	Displays the maximum disk usage of user queues. This metric only applies to standard queues.	0–100	Queues	5 minutes

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
queue_mem_usage	Queue Memory Usage	Displays the memory usage of user queues. This metric only applies to standard queues.	0–100	Queues	5 minutes
queue_mem_used	Used Memory	Displays the memory usage rate of the user queues. This metric only applies to standard queues.	≥ 0	Queues	5 minutes
flink_read_records_per_second	Flink Job Data Read Rate	Displays the data input rate of a Flink job for monitoring and debugging.	≥ 0	Flink jobs	10 seconds
flink_write_records_per_second	Flink Job Data Write Rate	Displays the data output rate of a Flink job for monitoring and debugging.	≥ 0	Flink jobs	10 seconds

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
flink_read_records_total	Flink Job Total Data Read	Displays the total number of data inputs of a Flink job for monitoring and debugging.	≥ 0	Flink jobs	10 seconds
flink_write_records_total	Flink Job Total Data Write	Displays the total number of output data records of a Flink job for monitoring and debugging.	≥ 0	Flink jobs	10 seconds
flink_read_bytes_per_second	Flink Job Byte Read Rate	Displays the number of input bytes per second of a Flink job.	≥ 0	Flink jobs	10 seconds
flink_write_bytes_per_second	Flink Job Byte Write Rate	Displays the number of output bytes per second of a Flink job.	≥ 0	Flink jobs	10 seconds
flink_read_bytes_total	Flink Job Total Read Byte	Displays the total number of input bytes of a Flink job.	≥ 0	Flink jobs	10 seconds

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
flink_write_bytes_total	Flink Job Total Write Byte	Displays the total number of output bytes of a Flink job.	≥ 0	Flink jobs	10 seconds
flink_cpu_usage	Flink Job CPU Usage	Displays the CPU usage of Flink jobs.	0–100	Flink jobs	10 seconds
flink_mem_usage	Flink Job Memory Usage	Displays the memory usage of Flink jobs.	0–100	Flink jobs	10 seconds
flink_max_op_latency	Flink Job Max Operator Latency	Displays the maximum operator delay of a Flink job. The unit is **ms**.	≥ 0	Flink jobs	10 seconds
flink_max_op_backpressure_level	Flink Job Maximum Operator Backpressure	Displays the maximum operator backpressure value of a Flink job. A larger value indicates severer backpressure. **0**: OK **50**: low **100**: high	0–100	Flink jobs	10 seconds

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
elastic_resource_pool_cpu_usage	CPU Usage of Elastic Resource Pool	Displays the CPU usage of elastic resource pools.	0–100	Elastic resource pools	5 minutes
elastic_resource_pool_mem_usage	Memory Usage of Elastic Resource Pool	Displays the memory usage of elastic resource pools.	0–100	Elastic resource pools	5 minutes
elastic_resource_pool_disk_usage	Disk Usage of Elastic Resource Pool	Displays the disk usage of elastic resource pools.	0–100	Elastic resource pools	5 minutes
elastic_resource_pool_disk_max_usage	Maximum Disk Usage of Elastic Resource Pool	Displays the maximum disk usage of elastic resource pools.	0–100	Elastic resource pools	5 minutes
elastic_resource_pool_cu_num	CU Usage of Elastic Resource Pool	Displays the CU usage of elastic resource pools.	≥ 0	Elastic resource pools	5 minutes
elastic_resource_pool_alloc_cu_num	Allocated CUs of Elastic Resource Pool	Displays the CU allocation of elastic resource pools.	≥ 0	Elastic resource pools	5 minutes

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
elastic_resource_pool_min_cu_num	Minimum CUs of Elastic Resource Pool	Displays the minimum number of CUs of elastic resource pools.	≥ 0	Elastic resource pools	5 minutes
elastic_resource_pool_max_cu_num	Maximum CUs of Elastic Resource Pool	Displays the maximum number of CUs of elastic resource pools.	≥ 0	Elastic resource pools	5 minutes

Dimension

Table 10-2 Dimension

Key	Value
queue_id	Queue
flink_job_id	Flink job

Viewing DLI Monitoring Metrics on Cloud Eye

1. Search for Cloud Eye on the management console.
2. In the navigation pane on the left of the Cloud Eye console, click **Cloud Service Monitoring** > **Data Lake Insight**.
3. Select a queue to view its information.

11 Using CTS to Audit DLI

With CTS, you can log operations related to DLI, making it easier to search, audit, and trace in the future.

Table 11-1 DLI operations that can be recorded by CTS

Operation	Resource Type	Trace Name
Creating a database	database	createDatabase
Deleting a database	database	deleteDatabase
Changing the database owner	database	alterDatabaseOwner
Creating a table	table	createTable
Deleting tables	table	deleteTable
Exporting table data	table	exportData
Importing table data	table	importData
Changing the owner of a table	table	alterTableOwner
Creating a queue	queue	createQueue
Deleting a queue	queue	deleteQueue
Granting permissions to a queue	queue	queueAuthorize
Modifying the CIDR block of a queue	queue	replaceQueue
Restarting a queue	queue	queueActions
Scaling out/in a queue	queue	queueActions
Submitting a job (SQL)	queue	submitJob
Canceling a job (SQL)	jobs	cancelJob

Operation	Resource Type	Trace Name
Granting DLI the permission to access OBS buckets	obs	authorizeObsBuckets-ForStream
Checking SQL syntax	jobs	checkSQL
Deleting a job	jobs	deleteStreamJob
Creating a Flink OpenSource SQL job	jobs	createStreamSqlJob
Updating a Flink OpenSource SQL job	jobs	updateStreamSqlJob
Deleting Flink jobs in batches	jobs	deleteStreamJobs
Stopping a Flink job	jobs	stopStreamJobs
Submitting a Flink job	jobs	submitStreamJobs
Creating a Flink Jar job	jobs	createStreamJarJob
Updating a Flink Jar job	jobs	updateStreamJarJob
Viewing Flink jobs	jobs	checkStreamJob
Importing a savepoint	jobs	dealSavepoint
Purchasing CUH packages	order	orderPackage
Granting data permissions	authorization	dataAuthorize
Granting permissions on other projects	authorization	projectDataAuthorize
Exporting query results	jobs	storeJobResult
Saving a SQL template	template	createTemplate
Updating a SQL template	template	updateTemplate
Deleting a SQL template	template	deleteTemplates
Creating a Flink template	template	createStreamTemplate
Updating a Flink template	template	updateStreamTemplate
Viewing Flink templates	template	checkStreamTemplate
Deleting a Flink template	template	deleteStreamTemplate
Creating a datasource authentication and uploading a certificate	datasourceauth	uploadAuthInfo

Operation	Resource Type	Trace Name
Updating datasource authentication information	datasourceauth	updateAuthInfo
Deleting datasource authentication information	datasourceauth	deleteAuthInfo
Uploading a resource package	resource	uploadResources
Deleting a resource package	resource	deleteResource
Creating an enhanced datasource connection	edsconnection	createConnection
Deleting an enhanced datasource connection	edsconnection	deleteConnection
Creating a basic datasource connection	edsconnection	createConnection
Deleting a basic datasource connection	edsconnection	deleteConnection
Binding a queue	edsconnection	associateQueueToConnection
Unbinding a queue	edsconnection	disassociateQueueToConnection
Modifying host information	edsconnection	updateHostInfo
Adding a route	edsconnection	addRoute
Deleting a route	edsconnection	deleteRoute
Creating a batch processing job	jobs	createBatch
Canceling a batch processing job	jobs	cancelBatch
Creating a global variable	variable	createGlobalVariable
Deleting a global variable	variable	deleteGlobalVariable
Modifying a global variable	variable	updateGlobalVariable

For how to enable CTS and view trace details, see the **Cloud Trace Service Getting Started**.

For details about key fields in the CTS trace structure, see **Trace StructureTrace Structure** in the *Cloud Trace Service User Guide*.

12 Permissions Management

12.1 Overview

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. You can use both the DLI's permission control mechanism and the IAM service for permission management.

Application Scenarios of IAM Authentication

When using DLI on the public cloud, enterprise users need to manage DLI resources (queues) used by employees in different departments, including creating, deleting, using, and isolating resources. In addition, data of different departments needs to be managed, including data isolation and sharing.

DLI uses IAM for refined enterprise-level multi-tenant management. IAM provides identity authentication, permissions management, and access control, helping you securely access to your public cloud resources.

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 is free to use, and you only need to pay for the resources in your account. For more information about IAM, see the **IAM Service Overview**.

If your Huawei Cloud account does not need individual IAM users for permissions management, skip over this section.

DLI System Permissions

Table 12-1 lists all the system-defined roles and policies supported by DLI.

Type: There are roles and policies.

- Roles: A type of coarse-grained authorization mechanism that defines permissions related to user responsibilities. Only a limited number of service-level roles are available. When using roles to grant permissions, you also need to assign other roles on which the permissions depend. However, roles are not an ideal choice for fine-grained authorization and secure access control.
- Policies: A type of fine-grained authorization mechanism that defines permissions required to perform operations on specific cloud resources under certain conditions. This mechanism allows for more flexible policy-based authorization, meeting requirements for secure access control. For example, you can grant DLI users only the permissions for managing a certain type of ECSs.

For details about the system policies you need to perform common SQL operations, see **Common Operations Supported by DLI System Policy**.

Table 12-1 DLI system permissions

Role/Policy Name	Description	Category	Dependency
DLI FullAccess	Full permissions for DLI.	System-defined policy	This role depends on other roles in the same project. • Creating a datasource connection: **VPC ReadOnlyAccess** • Creating yearly/monthly resources: **BSS Administrator** • Creating a tag: **TMS FullAccess** and **EPS EPS FullAccess** • Using OBS for storage: **OBS OperateAccess** • Creating an agency: **Security Administrator**

Role/Policy Name	Description	Category	Dependency
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	None
Tenant Administrator	Tenant administrator ● Job execution permissions for DLI resources. 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	None
DLI Service Administrator	DLI administrator. ● Job execution permissions for DLI resources. 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	None

For details, see **Creating an IAM User and Granting Permissions**, **How Do I Create an IAM user?** and **How Do I Modify a User Policy?**

DLI Permission Types

Table 12-2 lists the DLI service permissions. For details about the resources that can be controlled by DLI, see **Table 12-7**.

Table 12-2 DLI permission types

Permission Type	Subtype	Console Operations	SQL Syntax
Queue Permissions	Queue management permissions	For details, see **Queue Permission Management**.	None
	Queue usage permission		
Data Permissions	Database permissions	For details, see **Configuring Database Permissions on the DLI Console** and **Configuring Table Permissions on the DLI Console**.	For details, see **Data Permissions List**.
	Table permissions		
	Column permissions		
Job Permissions	Flink job permissions	For details, see **Configuring Flink Job Permissions**.	None
Package Permissions	Package group permissions	For details, see **Configuring DLI Package Permissions**.	None
	Package permissions		
Datasource Connection Permissions	Datasource connection permissions	For details, see **Datasource Authentication Permission Management**.	None

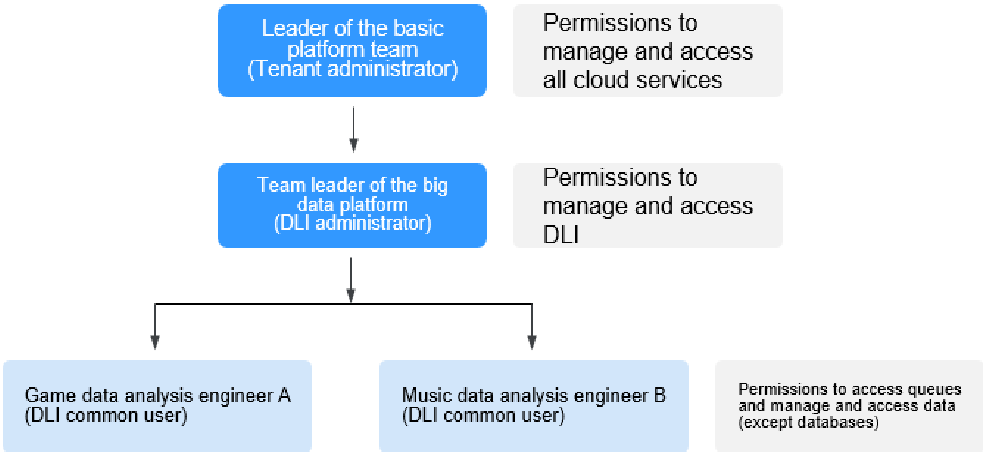
Examples

An Internet company mainly provides game and music services. DLI is used to analyze user behaviors and assist decision making.

As shown in **Figure 12-1**, the **Leader of the Basic Platform Team** has applied for a **Tenant Administrator** account to manage and use public cloud services. The **Leader of the Basic Platform Team** creates a subaccount with the **DLI Service Administrator** permission to manage and use DLI, as the **Big Data Platform**

Team requires DLI for data analysis. The **Leader of the Basic Platform Team** creates a **Queue A** and assigns it to **Data Engineer A** to analyze the gaming data. A **Queue B** is also assigned to **Data Engineer B** to analyze the music data. Besides granting the queue usage permission, the **Leader of the Basic Platform Team** grants data (except the database) management and usage permissions to the two engineers.

Figure 12-1 Granting permissions



The **Data Engineer A** creates a table named **gameTable** for storing game prop data and a table named **userTable** for storing game user data. The music service is a new service. To explore potential music users among existing game users, the **Data Engineer A** assigns the query permission on the **userTable** to the **Data Engineer B**. In addition, **Data Engineer B** creates a table named **musicTable** for storing music copyrights information.

Table 12-3 describes the queue and data permissions of **Data Engineer A** and **Data Engineer B**.

Table 12-3 Permission description

User	Data Engineer A (game data analysis)	Data Engineer B (music data analysis)
Queues	Queue A (queue usage permission)	Queue B (queue usage permission)
Data (Table)	gameTable (table management and usage permission)	musicTable (table management and usage permission)
	userTable (table management and usage permission)	userTable (table query permission)

NOTE

The queue usage permission includes job submitting and terminating permissions.

12.2 Creating a Custom Policy

Custom policies can be created as a supplement to the system policies of DLI. You can add actions to custom policies. For the actions supported for custom policies, see .

You can create custom policies in either of the following two ways:

- Visual editor: Select cloud services, actions, resources, and request conditions without the need to know policy syntax.
- JSON: Create a policy in the JSON format from scratch or based on an existing policy.

For details, see **Creating a Custom Policy**. This section describes common DLI custom policies.

Policy Field Description

The following example assumes that the authorized user has the permission to create tables in all databases in all regions:

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dli:database:createTable"
            ],
            "Resource": [
                "dli:*:*:database:*"
            ]
        }
    ]
}
```

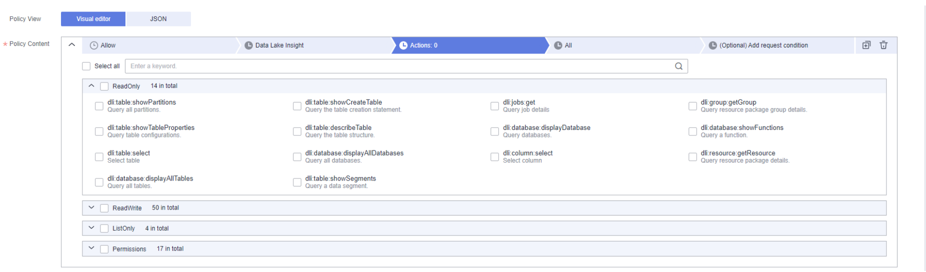
- Version

  1.1 indicates a fine-grained permission policy that defines permissions required to perform operations on specific cloud resources under certain conditions.
- Effect

  The value can be **Allow** and **Deny**. If both **Allow** and **Deny** are found in statements, the **Deny** overrides the **Allow**.
- Action

  Specific operation on a resource. A maximum of 100 actions are allowed, as shown in **Figure 12-2**.

Figure 12-2 DLI actions



NOTE

- The format is *Service name*:*Resource type*:*Action*, for example, **dli:queue:submit_job**.
- *Service name*: product name, such as **dli**, **evs**, and **vpc**. Only lowercase letters are allowed. Resource types and operations are not case-sensitive. You can use an asterisk (*) to represent all operations.
- *Resource type*: For details, see **Table 12-7**.
- *Action*: action registered in IAM.

- Condition

  Conditions determine when a policy takes effect. A condition consists of a condition key and operator.

  A condition key is a key in the **Condition** element of a statement. There are global and service-level condition keys.

  – Global condition keys (prefixed with **g:**) apply to all actions. For details, see condition key description in **Policy Syntax**.

  – Service-level condition keys apply only to operations of the specific service.

  An operator is used together with a condition key to form a complete condition statement. For details, see **Table 12-4**.

  IAM provides a set of DLI predefined condition keys. The following table lists the predefined condition keys of DLI.

Table 12-4 DLI request conditions

Condition Key	Type	Operator	Description
g:CurrentTime	Global	Date and time	Time when an authentication request is received **NOTE** The time is expressed in the format defined by **ISO 8601**, for example, **2012-11-11T23:59:59Z**.
g:MFAPresent	Global	Boolean	Whether multi-factor authentication is used during user login
g:UserId	Global	String	ID of the current login user
g:UserName	Global	String	Current login user
g:ProjectName	Global	String	Project that you have logged in to
g:DomainName	Global	String	Domain that you have logged in to
g:ResourceTag	Global	StringEquals	Resource tag value.

- Resource

  The format is *Service name:Region:Domain ID:Resource type:Resource path*. The wildcard (*) indicates all options. For details about the resource types and path, see **Table 12-7**.

  Example:

  dli:*:*:queue:* indicates all queues.

Creating a Custom Policy

You can set actions and resources of different levels based on scenarios.

1. Define an action.

   The format is *Service name:Resource type:Action*. The wildcard is *****. Example:

   Table 12-5 Action

Action	Description
dli:queue:submit_job	Submission operations on a DLI queue
dli:queue:*	All operations on a DLI queue
dli:*:*	All operations on all DLI resource types

   For more information about the relationship between operations and system permissions, see **Common Operations Supported by DLI System Policy**.

2. Define a resource.

   The format is *Service name:Region:Domain ID:***Resource type:Resource path**. The wildcard (*) indicates all resources. The five fields can be flexibly set. Different levels of permission control can be set for resource paths based on scenario requirements. If you need to set all resources of the service, you do not need to specify this field. For details about the definition of Resource, see **Table 12-6**. For details about the resource types and paths in Resource, see **Table 12-7**.

   Table 12-6 Resource

Resource	Description
DLI:*:*:table:databases.dbname.tables.*	DLI, any region, any account ID, all table resources of database **dbname**
DLI:*:*:database:databases.dbname	DLI, any region, any account ID, resource of database **dbname**
DLI:*:*:queue:queues.*	DLI, any region, any account ID, any queue resource

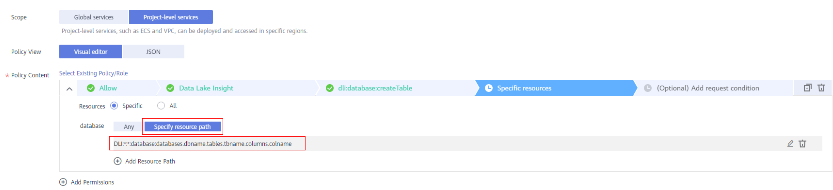
Resource	Description
DLI:*:*:jobs:jobs.flink.1	DLI, any region, any account ID, Flink job whose ID is 1

Table 12-7 DLI resources and their paths

Resource Type	Resource Names	Path
queue	DLI queue	queues.queuename
database	DLI database	databases.dbname
table	DLI table	databases.dbname.tables.tbname
column	DLI column	databases.dbname.tables.tbname.columns.colname
jobs	DLI Flink job	jobs.flink.jobid
resource	DLI package	resources.resourcename
group	DLI package group	groups.groupname
datasourceauth	DLI cross-source authentication information	datasourceauth.name
edsconnections	Enhanced datasource connection	edsconnections.*connection ID*

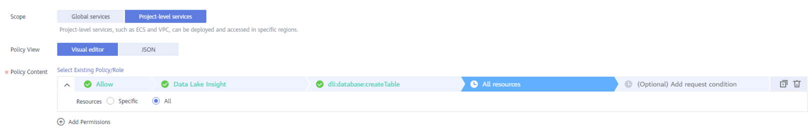
– Specific resources:

Figure 12-3 Specific resources



– All resources: all resources of the service

Figure 12-4 All resources



3. Combine all the preceding fields into a JSON file to form a complete policy. You can set multiple actions and resources. You can also create a policy on the visualized page provided by IAM. For example:

   Create a policy that grants users the permission to create and delete databases, submit jobs for any queue, and delete tables under any account ID in any region of DLI.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": " Allow",
            "Action": [
                  "dli:database:createDatabase",
                  "dli:database:dropDatabase",
                  "dli:queue:submitJob",
                  "dli:table:dropTable"
            ],
            "Resource": [
                  "dli:*:*:database:*",
                  "dli:*:*:queue:*",
                  "dli:*:*:table:*"
            ]
        }
    ]
}
```

Example Custom Policies

- Example 1: Allow policies
  - Allow users to create tables across all databases in all regions:

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dli:database:createTable"
            ],
            "Resource": [
                "dli:*:*:database:*"
            ]
        }
    ]
}
```

  - Allow users to query column **col** in the table **tb** of the database **db** in the region where the user is located:

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dli:column:select"
            ],
            "Resource": [
                "dli:xxx:*:column:databases.db.tables.tb.columns.col"
            ]
        }
    ]
}
```

- Example 2: Deny policies

A deny policy must be used together with other policies. That is, a user can set a deny policy only after being assigned some operation permissions. Otherwise, the deny policy does not take effect.

If the permissions assigned to a user contain both Allow and Deny actions, the Deny actions take precedence over the Allow actions.

– Deny users to create or delete databases, submit jobs (except the default queue), or delete tables.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "dli:database:createDatabase",
                "dli:database:dropDatabase",
                "dli:queue:submitJob",
                "dli:table:dropTable"
            ],
            "Resource": [
                "dli:*:*:database:*",
                "dli:*:*:queue:*",
                "dli:*:*:table:*"
            ]
        }
    ]
}
```

– Deny users to submit jobs in the demo queue.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "dli:queue:submitJob"
            ],
            "Resource": [
                "dli:*:*:queue:queues.demo"
            ]
        }
    ]
}
```

- Example 3: Tag authentication. You need to specify an action and bind it to a condition, and specify the key and value of **g:ResourceTag**.

  Condition g: ResourceTag indicates a resource with the *key=value* tag. Only the operation on this resource contained in the policy action list is allowed.

  The key is case insensitive. Fuzzy match for the value is not supported.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dli:database:dropDatabase",
                "dli:table:select",
                "dli:database:createTable",
                "dli:table:dropTable"
            ],
            "Condition": {
                "StringEquals": {
                    "g:ResourceTag/key": [
                        "value"
                    ]
                }
```

```
            }
        }
    ]
}
```

12.3 DLI Resources

A resource is an object that exists within a service. You can select DLI resources by specifying their paths.

Table 12-8 DLI resources and their paths

Resource Type	Resource Names	Path
queue	DLI queue	queues.queuename
database	DLI database	databases.dbname
table	DLI table	databases.dbname.tables.tbname
column	DLI column	databases.dbname.tables.tbname.columns.colname
jobs	DLI Flink job	jobs.flink.jobid
resource	DLI package	resources.resourcename
group	DLI package group	groups.groupname
datasourceauth	DLI cross-source authentication information	datasourceauth.name
edsconnections	Enhanced datasource connection	edsconnections.*connection ID*

12.4 DLI Request Conditions

Request conditions are useful in determining when a custom policy takes effect. A request condition consists of a condition key and operator. Condition keys are either global or service-level and are used in the Condition element of a policy statement. **Global condition keys** (starting with **g:**) are available for operations of all services, while service-level condition keys (starting with a service name such as **dli**) are available only for operations of a specific service. An operator is used together with a condition key to form a complete condition statement.

IAM provides a set of DLI predefined condition keys. The following table lists the predefined condition keys of *DLI*.

Table 12-9 DLI request conditions

Condition Key	Type	Operator	Description
g:CurrentTime	Global	Date and time	Time when an authentication request is received **NOTE** The time is expressed in the format defined by **ISO 8601**, for example, **2012-11-11T23:59:59Z**.
g:MFAPresent	Global	Boolean	Whether multi-factor authentication is used during user login
g:UserId	Global	String	ID of the current login user
g:UserName	Global	String	Current login user
g:ProjectName	Global	String	Project that you have logged in to
g:DomainName	Global	String	Domain that you have logged in to
g:ResourceTag	Global	StringEquals	Resource tag value.

12.5 Common Operations Supported by DLI System Policy

Table 12-10 lists the common operations supported by each system policy of DLI. Choose proper system policies according to this table. For details about the SQL statement permission matrix in DLI in terms of permissions on databases, tables, and roles, see **Data Permission List**.

Table 12-10 Common operations supported by each system permission

Resource	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Administrator
Queue	DROP_QUEUE	Deleting a Queue	√	×	√	√
	SUBMIT_JOB	Submitting a job	√	×	√	√

Resource	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Administrator
	CANCEL_JOB	Terminating a Job	√	×	√	√
	RESTART	Restarting a queue	√	×	√	√
	GRANT_PRIVILEGE	Granting permissions to a queue	√	×	√	√
	REVOKE_PRIVILEGE	Revoking permissions to a queue	√	×	√	√
	SHOW_PRIVILEGES	Viewing the queue permissions of other users	√	×	√	√
Database	DROP_DATABASE	Deleting a database	√	×	√	√
	CREATE_TABLE	Creating a table	√	×	√	√
	CREATE_VIEW	Creating a view	√	×	√	√
	EXPLAIN	Explaining the SQL statement as an execution plan	√	×	√	√
	CREATE_ROLE	Creating a role	√	×	√	√
	DROP_ROLE	Deleting a role	√	×	√	√
	SHOW_ROLES	Displaying a role	√	×	√	√
	GRANT_ROLE	Binding a role	√	×	√	√
	REVOKE_ROLE	Unbinding a role	√	×	√	√

Resource	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Administrator
	SHOW_USERS	Displaying the binding relationships between all roles and users	√	×	√	√
	GRANT_PRIVILEGE	Granting permissions to the database	√	×	√	√
	REVOKE_PRIVILEGE	Revoking permissions to the database	√	×	√	√
	SHOW_PRIVILEGES	Viewing database permissions of other users	√	×	√	√
	DISPLAY_ALL_TABLES	Displaying tables in a database	√	√	√	√
	DISPLAY_DATABASE	Displaying databases	√	√	√	√
	CREATE_FUNCTION	Creating a function	√	×	√	√
	DROP_FUNCTION	Deleting a function	√	×	√	√
	SHOW_FUNCTIONS	Displaying all functions	√	×	√	√
	DESCRIBE_FUNCTION	Displaying function details	√	×	√	√
Table	DROP_TABLE	Deleting tables	√	×	√	√
	SELECT	Querying tables	√	×	√	√
	INSERT_INTO_TABLE	Inserting table data	√	×	√	√

Resource	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Administrator
	ALTER_TABLE_ADD_COLUMNS	Adding a column	√	×	√	√
	INSERT_OVERWRITE_TABLE	Overwriting a table	√	×	√	√
	ALTER_TABLE_RENAME	Renaming a table	√	×	√	√
	ALTER_TABLE_ADD_PARTITION	Adding partitions to the partition table	√	×	√	√
	ALTER_TABLE_RENAME_PARTITION	Renaming a table partition	√	×	√	√
	ALTER_TABLE_DROP_PARTITION	Deleting partitions from a partition table	√	×	√	√
	SHOW_PARTITIONS	Displaying all partitions	√	×	√	√
	ALTER_TABLE_RECOVER_PARTITION	Restoring table partitions	√	×	√	√
	ALTER_TABLE_SET_LOCATION	Setting the partition path	√	×	√	√
	GRANT_PRIVILEGE	Granting permissions to the table	√	×	√	√
	REVOKE_PRIVILEGE	Revoking permissions to the table	√	×	√	√
	SHOW_PRIVILEGES	Viewing table permissions of other users	√	×	√	√

Resource	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Administrator
	DISPLAY_TABLE	Displaying a table	√	√	√	√
	DESCRIBE_TABLE	Displaying table information	√	×	√	√
Elastic resource pool	DROP	Deleting an elastic resource pool	√	×	√	√
	RESOURCE_MANAGEMENT	Managing an elastic resource pool	√	×	√	√
	SCALE	Scaling an elastic resource pool	√	×	√	√
	UPDATE	Updating an elastic resource pool	√	×	√	√
	CREATE	Creating an elastic resource pool	√	×	√	√
	SHOW_PRIVILEGES	Viewing elastic resource pool permissions of other users	√	×	√	√
	GRANT_PRIVILEGE	Granting elastic resource pool permissions	√	×	√	√

Resource	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Administrator
	REVOKE_PRIVILEGE	Retrieving elastic resource pool permissions	√	×	√	√
Enhanced datasource connection	BIND_QUEUE	Binding an enhanced datasource connection to a queue It is only used to grant permissions across projects.	×	×	×	×

13 Common DLI Management Operations

13.1 Using a Custom Image to Enhance the Job Running Environment

Scenario

To enhance the functions and performance of Spark and Flink jobs, you can create custom images by downloading the base images provided by DLI and adding dependencies (files, JAR files, or software) and private capabilities required for job execution. This changes the container runtime environment for the jobs.

For example, you can add a Python package or C library related to machine learning to a custom image to help you extend functions.

NOTE

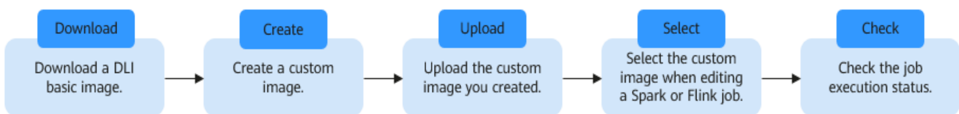
To use the custom image function, you need to have basic knowledge of **Docker**.

Notes and Constraints

- The base images provided by DLI must be used to create custom images.
- You cannot modify the DLI components and directories in the base images.
- Only Spark Jar and Flink Jar jobs are supported.

Use Process

Figure 13-1 Process of using a custom image



1. Obtain DLI base images.
2. Use Dockerfile to pack dependencies (files, JAR files, or software) required for job execution into the base image to create a custom image.
3. Publish the custom image to SoftWare Repository for Container (SWR).
4. On the DLI job editing page, select the created image and run the job.
5. Check the job execution status.

Obtaining DLI Base Images

Table 13-1 Obtaining DLI base images

Image Type	Architecture	URL
General image	x86	swr.ap-southeast-3.myhuaweicloud.com/dli-public/spark_general-x86_64:3.3.1-2.3.7.1720240419835647952528832.202404250955
General image	Arm	swr.ap-southeast-3.myhuaweicloud.com/dli-public/spark_general-aarch64:3.3.1-2.3.7.1720240419835647952528832.202404250955
Notebook image	x86	swr.ap-southeast-3.myhuaweicloud.com/dli-public/spark_notebook-x86_64:3.3.1-2.3.7.1720240419835647952528832.202404250955
Notebook image	Arm	swr.ap-southeast-3.myhuaweicloud.com/dli-public/spark_notebook-aarch64:3.3.1-2.3.7.1720240419835647952528832.202404250955

Creating a Custom Image

The following describes how to package TensorFlow into an image to generate a custom image with TensorFlow installed. Then, you can use the image to run jobs in DLI.

Step 1 Prepare the container environment.

For details, see "Step 1: Install the Container Engine" in **Uploading an Image Through a Container Engine Client**.

Step 2 Log in to the prepared container environment as user **root** and run a command to obtain the base image.

In this example, the Spark base image is used and downloaded to the container image environment in **Step 1** by running the following command:

docker pull *Address for downloading the base image*

For details about the address, see **Using a Custom Image to Enhance the Job Running Environment**.

For example, to download the Spark base image, run the following command:

```
docker pull swr.xxx/dli-public/spark_general-x86_64:3.3.1-2.3.7.1720240419835647952528832.202404250955
```

Step 3 Access SWR.

1. Log in to the SWR management console.
2. In the navigation pane on the left, choose **Dashboard** and click **Generate Login Command** in the upper right corner. On the displayed page, click to copy the login command.
3. Run the login command on the VM where the container engine is installed.

Step 4 Create an organization. If an organization has been created, skip this step.

1. Log in to the SWR management console.
2. In the navigation pane on the left, choose **Organization Management**. On the displayed page, click **Create Organization** in the upper right corner.
3. Enter the organization name and click **OK**.

Step 5 Write a Dockerfile.

vi Dockerfile

Pack TensorFlow into the image as follows:

```
ARG BASE_IMG=swr.xxx/dli-public/spark_general-
x86_64:3.3.1-2.3.7.1720240419835647952528832.202404250955//Replace xxx with the URL of the base
image.

FROM ${BASE_IMG} as builder
USER omm //Run this command as user omm.
RUN set -ex && \
    mkdir -p /home/omm/.pip && \
    pip3 install tensorflow==1.13.1 \
    --user --no-cache-dir  --trusted-host pypi.cloudartifact.dgg.dragon.tools.huawei.com \
    -i https://pypi.cloudartifact.dgg.dragon.tools.huawei.com/artifactory/api/pypi/cbu-pypi-public/simple
Copy the content to the base image.
USER omm
```

The following steps are included:

1. Set the available repository address of pip.
2. Use pip3 to install the TensorFlow algorithm package.
3. Copy the content in the temporary image builder where the algorithm package is installed to the base image (this step is to reduce the image size) to generate the final custom image.

Step 6 Use Dockerfile to generate a custom image.

Format of the image packaging command:

```
docker build -t [Custom organization name]/[Custom image name]: [Image version] --build-arg
BASE_IMG= [DLI base image path] -f Dockerfile .
```

The DLI base image path is the image path in **Obtaining DLI Base Images**.

The following is an example:

```
docker build -t mydli/spark:2.4 --build-arg BASE_IMG=swr.xxx/dli-public/spark_general-
x86_64:3.3.1-2.3.7.1720240419835647952528832.202404250955 -f Dockerfile .
```

Step 7 Add a tag to the custom image.

docker tag [Organization name]/[Image name]:[Image version][Image repository address]/[Organization name]/[Image name:version] in **Step 6**

The following is an example:

```
docker tag mydli/spark:2.4 swr.xxx/testdli0617/spark:2.4.5.tensorflow
```

Step 8 Upload the custom image.

docker push [Image repository address]/[Organization name]/[Image name:Version]

Set [Image repository address]/[Organization name]/[Image name:Version] the same as those in **Step 7**.

The following is an example:

```
docker push swr.xxx/testdli0617/spark:2.4.5.tensorflow
```

Step 9 When submitting a Spark or Flink JAR job in DLI, select a custom image.

- Open the Spark job or Flink job editing page on the management console, select the uploaded and shared image from the custom image list, and run the job.

  If you select a non- shared image, the system displays a message indicating that the image is not authorized. You can use the image only after it is authorized. Click **Authorize** as prompted. Set other job execution parameters and execute the job.

Figure 13-2 Selecting a custom image on the DLI Spark job editing page

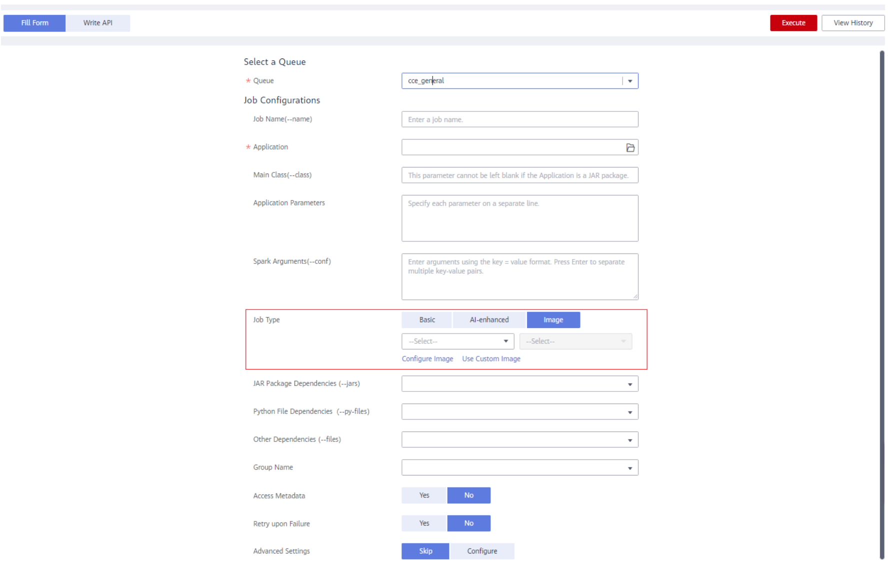


Figure 13-3 Authorizing a Spark job image

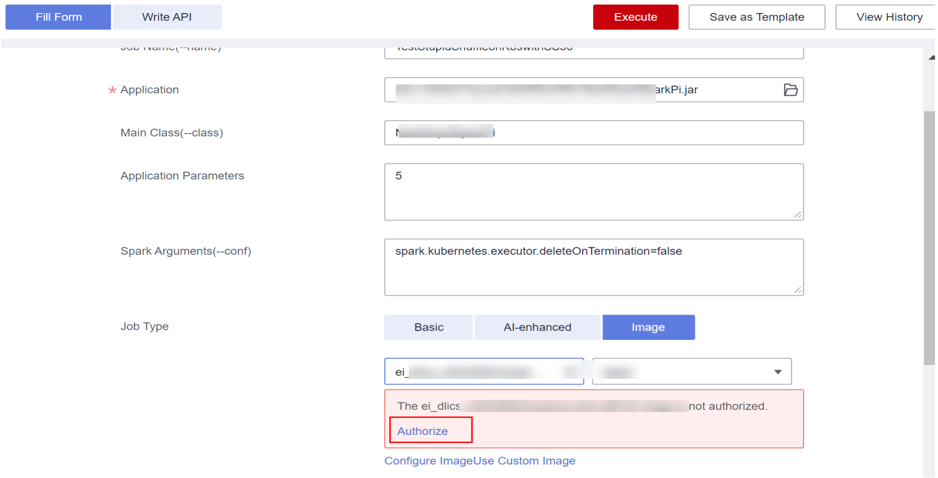
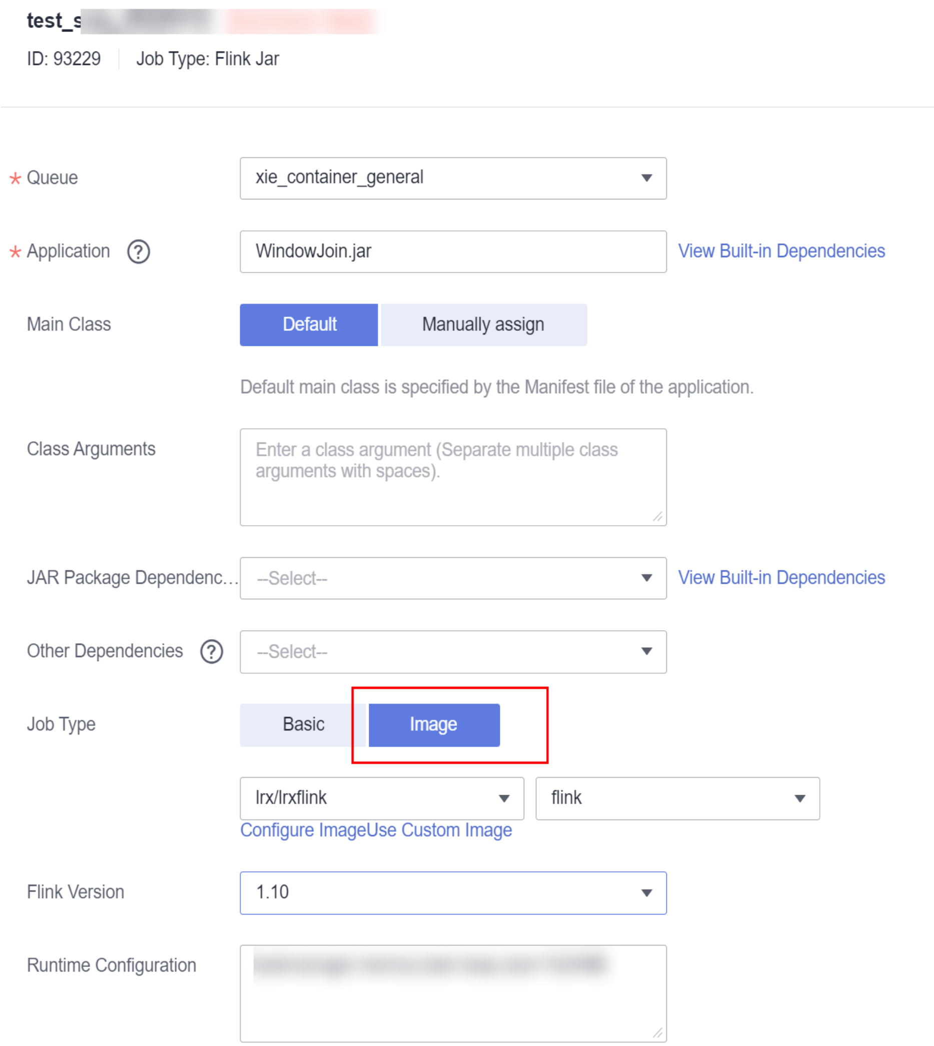


Figure 13-4 Selecting a custom image on the DLI Flink Jar job editing page



- Specify the image parameter in job parameters on API to use a custom image to run a job. For details about Spark jobs, see **Creating a Batch Processing Job**. For details about Flink Jar jobs, see **Creating a Flink Jar Job**.

----End

13.2 Managing DLI Global Variables

What Is a Global Variable?

DLI allows you to set variables that are frequently used during job development as global variables on the DLI management console. This avoids repeated definitions during job editing and reduces development and maintenance costs. Global variables can be used to replace long and difficult variables, simplifying complex parameters and improving the readability of SQL statements.

This section describes how to create a global variable.

Creating a Global Variable

1. In the navigation pane of the DLI console, choose **Global Configuration** > **Global Variables**.
2. On the **Global Variables** page, click **Create** in the upper right corner to create a global variable.

Table 13-2 Global variable parameters

Parameter	Description
Variable	Name of the created global variable.
Value	Global variable value.

3. After creating a global variable, use **{{xxxx}}** in the SQL statement to replace the parameter value set as the global variable. **xxxx** indicates the variable name. For example, if you set global variable **abc** to represent the table name, replace the actual table name with **{{abc}}** in the table creation statement.

```
create table {{table_name}} (String1 String, int4 int, varchar1 varchar(10))
 partitioned by (int1 int,int2 int,int3 int)
```

NOTE

Do not use global variables in **OPTIONS** of the table creation statements.

Related operations:

- **Modifying a global variable**

  On the **Global Variables** page, locate a desired variable and click **Modify** in the **Operation** column.

NOTE

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.

- **Deleting a global variable**

  On the **Global Variables** page, click **Delete** in the **Operation** column of a variable to delete the variable value.

  NOTE

  - 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.
  - After a variable is deleted, the variable cannot be used in SQL statements.

Permissions Management for Global Variables

You can assign different users different global variables through permission settings. The administrator and owners of global variables have all permissions. You do not need to set permissions for them, and their global variable permissions cannot be modified by other users.

When setting global variables for a new user, the region hosting the user's group must have the **Tenant Guest** permission. For details about the **Tenant Guest** permission and how to apply for the permission, see **System-defined Permissions** and **Creating a User Group and Assigning Permissions**.

- **Granting permissions on a global variable to a user**
  a. In the navigation pane on the left of the DLI console, choose **Global Configuration** > **Global Variables**. On the displayed page, locate a desired global variable and click **Set Permission** in the **Operation** column. On the displayed **User Permissions** page, you can grant, set, and revoke permissions on the global variable to users.
  b. Click **Grant Permission** in the upper right corner.

Figure 13-5 Granting permissions of a global variable to a user

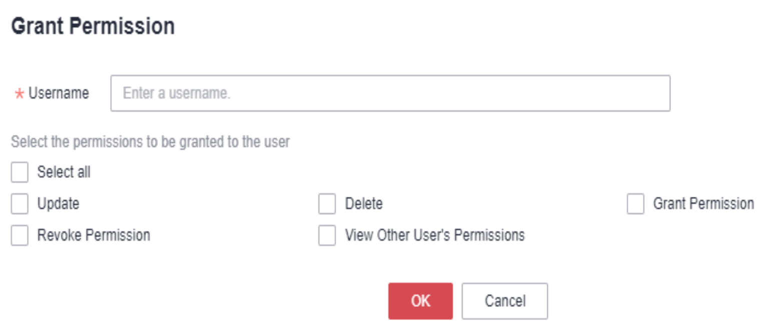


Table 13-3 Global variable parameters

Parameter	Description
Username	Name of the IAM user who is granted permissions **NOTE** This username must be an existing IAM username.
Permissions	● **Update**: This permission allows you to update the global variable. ● **Delete**: This permission allows you to delete the global variable. ● **Grant Permission**: This permission allows you to grant permissions of the global variable to other users. ● **Revoke Permission**: This permission allows you to revoke the global variable permissions that other users have but cannot revoke the global variable owner's permissions. ● **View Other User's Permissions**: This permission allows you to view the global variable permissions of other users.

- **Granting permissions on a global variable to a user**

  On the **User Permissions** page, locate a desired IAM user and click **Set Permission** in the **Operation** column. **Table 13-3** lists the permission parameters.

  If all permission options are grayed out, it means you do not have the authority to modify the permissions on this global variable. You can request the modification permission from users who have authorization, such as the administrator or group owners.
- **Revoking permissions on a global variable from a user**

  On the **User Permissions** page, locate a desired IAM user and click **Revoke Permission** in the **Operation** column. Once the revoke operation is complete, the IAM user will no longer have any permissions on the global variable.

13.3 Managing Program Packages of Jar Jobs

13.3.1 Package Management Overview

Before running DLI jobs, UDF JAR files or Jar job packages need to be uploaded to the cloud platform for unified management and maintenance.

There are two ways to manage packages:

- (Recommended) Upload packages to OBS: Upload Jar packages to an OBS bucket in advance and select the OBS path when configuring a job.
- (The DLI package function will soon be discontinued.) Upload packages to DLI: Upload Jar packages to an OBS bucket in advance, create a package on

the **Data Management** > **Package Management** page of the DLI management console, and select the DLI package when configuring a job.

This section describes how to upload and manage packages on the DLI management console.

NOTE

- The DLI package function will soon be discontinued. When using Spark 3.3.1 or later or Flink1.15 or later to run jobs, you are advised to select packages stored in OBS.
- When packaging Spark or Flink Jar jobs, do not upload the dependency packages that the platform already has to avoid conflicts with the built-in dependency packages of the platform. Refer to **DLI Built-in Dependencies** for built-in dependency packages.

Notes and Constraints

- A package can be deleted, but a package group cannot be deleted.
- The following types of packages can be uploaded:
  - **JAR**: JAR file
  - **PyFile**: User Python file
  - **File**: User file
  - **ModelFile**: User AI model file

Package Management Page

Table 13-4 Package management parameters

Parameter	Description
Group Name	Name of the group to which the package belongs. If the package is not grouped, **--** is displayed.
Package Name	Name of a package.
Owner	Name of the user who uploads the package.
Type	Type of a package. The following package types are supported: ● JAR: JAR file ● PyFile: User Python file ● File: User file ● ModelFile: User AI model file
Status	Status of the package to be created. ● Uploading: The file is being uploaded. ● Finished: The resource package has been uploaded. ● Failed: The resource package upload failed.
Created	Time when a package is created.
Updated	Time when the package is updated.

Parameter	Description
Operation	Manage Permissions: Manage user permissions for a package. Delete: Delete the package. **More**: ● **Modify Owner**: Modify the owner of the package. ● **Tags**: Add or edit package tags.

13.3.2 Creating a DLI Package

DLI allows you to submit program packages in batches to the general-use queue for running.

NOTE

If you need to update a package, you can use the same package or file to upload it to the same location (in the same group) on DLI to overwrite the original package or file.

Prerequisites

All software packages must be uploaded to OBS for storage in advance.

Procedure

1. On the left of the management console, choose **Data Management** > **Package Management**.
2. On the **Package Management** page, click **Create** in the upper right corner to create a package.
3. In the displayed **Create Package** dialog box, set related parameters by referring to **Table 13-5**.

Figure 13-6 Creating a package

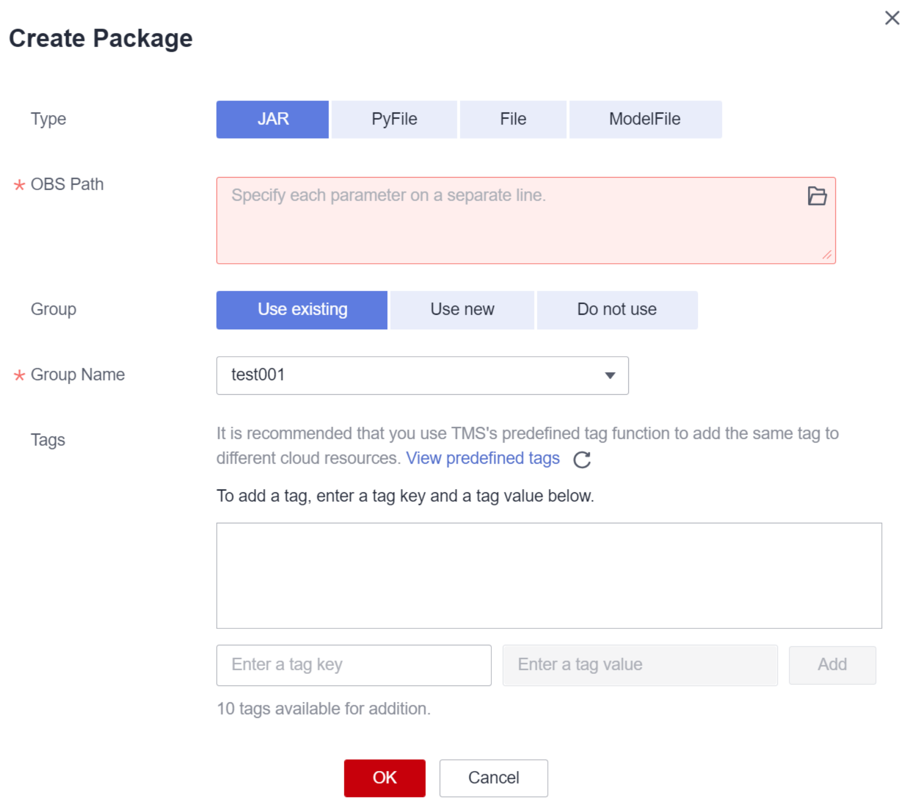


Table 13-5 Parameter description

Parameter	Description
Package Type	The following package types are supported: • JAR: JAR file • PyFile: User Python file • File: User file • ModelFile: User AI model file
Package File Path	Select the OBS path of the corresponding packages. **NOTE** • The packages must be uploaded to OBS for storage in advance. • Only files can be selected.
Group Policy	You can select **Use existing group**, **Use new group**, or **No grouping**.

Parameter	Description
Group Name	● **Use existing group**: Select an existing group. ● **Use new group**: Enter a custom group name. ● **No grouping**: No need to select or enter a group name. NOTE ● If you select a group, the permission management refers to the permissions of the corresponding package group. ● If no group is selected, the permission management refers to the permissions of the corresponding package. For details about how to manage permissions on package groups and packages, see **Managing Permissions on Packages and Package Groups**.
Tag	Tags used to identify cloud resources. A tag includes the tag key and tag value. If you want to use the same tag to identify multiple cloud resources, that is, to select the same tag from the drop-down list box for all services, you are advised to create predefined tags on the Tag Management Service (TMS). If your organization has configured tag policies for DLI, add tags to resources based on the policies. If a tag does not comply with the tag policies, resource creation may fail. Contact your organization administrator to learn more about tag policies. For details, see **Tag Management Service User Guide**. NOTE ● A maximum of 20 tags can be added. ● Only one tag value can be added to a tag key. ● The key name in each resource must be unique. ● Tag key: Enter a tag key name in the text box. NOTE A tag key can contain a maximum of 128 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed, but the value cannot start or end with a space or start with **_sys_**. ● Tag value: Enter a tag value in the text box. NOTE A tag value can contain a maximum of 255 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed. The value cannot start or end with a space.

4. Click **OK**.

   After a package is created, you can view and select the package for use on the **Package Management** page.

   Once a job is executed, you can free up DLI storage space by promptly deleting the job package from the package management page if it is no longer needed.

13.3.3 Configuring DLI Package Permissions

By configuring permissions, you can grant different package groups or packages to various users, ensuring that job efficiency remains unaffected and job performance is maintained.

- Administrators and package group owners have all permissions for the package group. No permission settings are required, and other users cannot modify their package group permissions.
- Administrators and package owners have all permissions for the package. No permission settings are required, and other users cannot modify their package permissions.
- Package groups are used to manage packages with consistent behavior, so they support granting related permissions to package groups, but do not support granting individual permissions to packages within a package group.
- When an administrator assigns package group or package permissions to a new user, the region of the administrator's user group must have Tenant Guest permissions.

  For details about the **Tenant Guest** permission and how to apply for the permission, see **System Permissions** and **Creating a User Group** in *Identity and Access Management User Guide*.

Configuring Permissions on Package Groups or Packages

1. On the **Data Management** > **Package Management** page, locate the package group or package whose permissions you want to grant and click **Manage Permission** in the **Operation** column.
2. On the displayed **User Permissions** page, click **Grant Permission** in the upper right corner of the page. In the dialog box that appears, enter a username in **Username**, select required permissions, and click **OK**.

   NOTE

   - If you select a group when creating a package, you can manage permissions of the corresponding program package group.
   - If you select **No grouping** when creating a package, you can manage permissions of the corresponding package.

   – Granting permissions on package groups

   Figure 13-7 Granting permissions on package groups

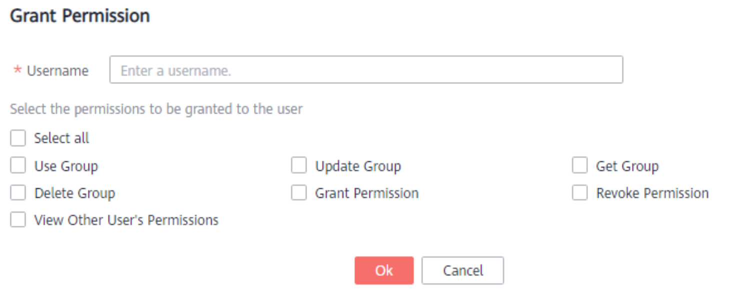
   

Table 13-6 Permission parameters

Parameter	Description
Username	Name of the authorized IAM user. **NOTE** The username is the name of an existing IAM user.
Select the permissions to be granted to the user	● **Use Group**: This permission allows you to use the package of this group. ● **Update Group**: This permission allows you to update the packages in the group, including creating a package in the group. ● **Query Group**: This permission allows you to query the details of a package in a group. ● **Delete Group**: This permission allows you to delete the package of the group. ● **Grant Permission**: This permission allows you to grant group permissions to other users. ● **Revoke Permission**: This permission allows you to revoke the group permissions that other users have but cannot revoke the group owner's permissions. ● **View Other User's Permissions**: This permission allows you to view the group permissions of other users.

– Granting permissions on packages

Figure 13-8 Granting permissions on package groups

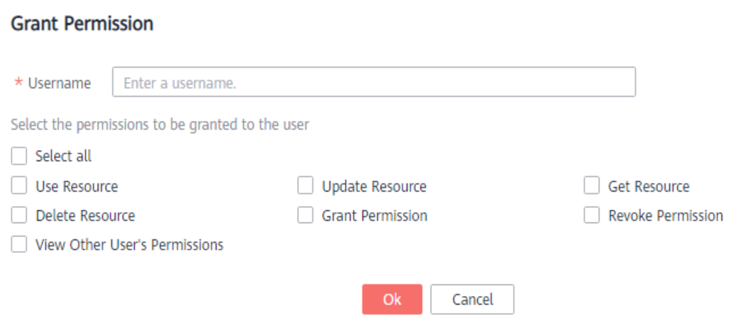


Table 13-7 Permission parameters

Parameter	Description
Username	Name of the authorized IAM user. **NOTE** The username is the name of an existing IAM user.

Parameter	Description
Select the permissions to be granted to the user	• **Use Package**: This permission allows you to use the package. • **Update Package**: This permission allows you to update the package. • **Query Package**: This permission allows you to query the package. • **Delete Package**: This permission allows you to delete the package. • **Grant Permission**: This permission allows you to grant package permissions to other users. • **Revoke Permission**: This permission allows you to revoke the package permissions that other users have but cannot revoke the package owner's permissions. • **View Other User's Permissions**: This permission allows you to view the package permissions of other users.

Modifying Permissions on Package Groups or Packages

1. On the **Data Management** > **Package Management** page, locate the desired package group or package and click **Manage Permission** in the **Operation** column.
2. On the displayed **User Permissions** page, click **Set Permission** in the **Operation** column of the IAM user for whom you want to modify the permissions.

   Table 13-6 and **Table 13-7** list the detailed permission descriptions.

   NOTE

   - If you set **Group** to **Use existing** or **Use new** when creating a package, you will modify the permissions on the group you selected.
   - If you set **Group** to **Do not use** when creating a package, you will modify the permissions on the package.

   If **Set Permission** on the **User Permissions** page is grayed out, you do not have the permission to modify the permissions on the package group or package.

   You can request the **Grant Permission** and **Revoke Permission** permissions on package groups or packages from users who have authorization privileges, such as administrators or group owners.

Revoking Permissions on Package Groups or Packages

DLI allows you to revoke permissions on package groups or packages with just one click.

1. On the **Data Management** > **Package Management** page, locate the desired package group or package and click **Manage Permission** in the **Operation** column.

2. On the displayed **User Permissions** page, click **Revoke Permission** in the **Operation** column of the IAM user for whom you want to revoke the permissions.

   Once the permissions are revoked, the IAM user does not have any permissions on the package group or package.

   NOTE

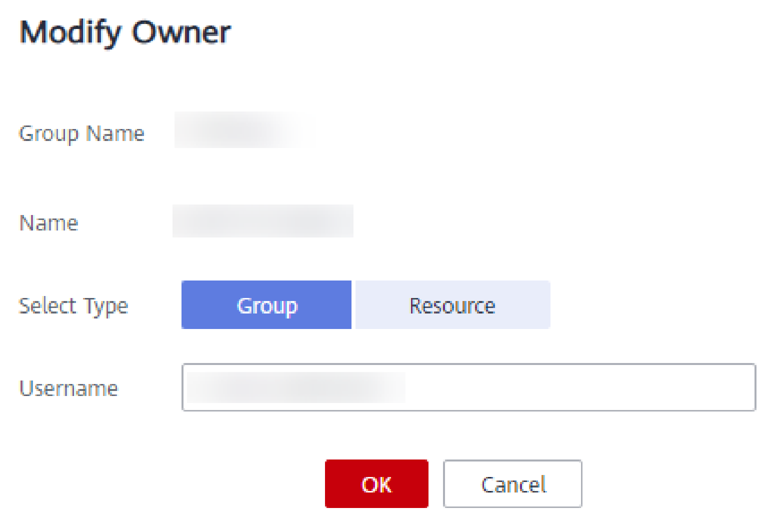
   - If you set **Group** to **Use existing** or **Use new** when creating a package, you will revoke the permissions on the group you selected.
   - If you set **Group** to **Do not use** when creating a package, you will revoke the permissions on the package.

13.3.4 Changing the DLI Package Owner

DLI allows you to change the owner of a package group or package.

1. Log in to the DLI management console and choose **Data Management** > **Package Management**.
2. On the **Package Management** page, locate the package whose owner you want to change, click **More** in the **Operation** column, and select **Modify Owner**.
   - If the package has been grouped, you can change the owner of the group or package by selecting **Group** or **Packages** for **Select Type** and entering the new owner's username in **Username**.

     Figure 13-9 Modifying the package owner

   - If the package has not been grouped, change its owner directly.

Figure 13-10 Modifying the owner of a package

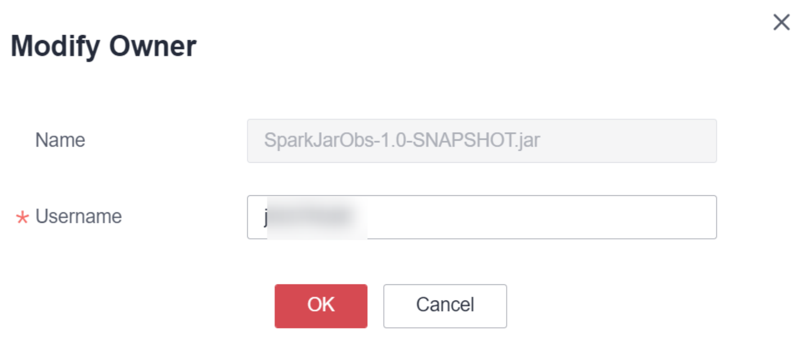


Table 13-8 Description

Parameter	Description
Group Name	● If you select a group when creating a package, the name of the group is displayed. ● If no group is selected when creating a package, this parameter is not displayed.
Name	Name of a package.
Select Type	● If you select a group when creating a package, you can change the owner of the group or package. ● If no group is selected when creating a package, this parameter is not displayed.
Username	Name of the package owner. **NOTE** The username is the name of an existing IAM user.

3. Click **OK**.

13.3.5 Managing DLI Package Tags

Tags are key-value pairs that you can define to identify cloud resources. They assist you in categorizing and searching for cloud resources. A tag consists of a key and a value.

DLI allows you to add tags to package groups or packages.

1. Log in to the DLI management console and choose **Data Management** > **Package Management**.
2. On the **Package Management** page, locate the desired package, click **More** in the **Operation** column, and select **Tags**.
3. On the page that appears, click **Add/Edit Tag** in the upper left corner.
4. In the **Add/Edit Tag** dialog box, set parameters.

Table 13-9 Tag parameters

Parameter	Description
Tag key	You can specify the tag key in either of the following ways: ● Click the text box and select a predefined tag key from the drop-down list. To add a predefined tag, you need to create one on TMS and then select it from the tag key drop-down list. You can click **View predefined tags** to go to the **Predefined Tags** page of the TMS console. Then, click **Create Tag** in the upper corner of the page to create a predefined tag. For details, see **Creating Predefined Tags** in *Tag Management Service User Guide*. ● Enter a tag key in the text box. **NOTE** A tag key can contain a maximum of 128 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed, but the value cannot start or end with a space or start with _**sys**_.
Tag value	You can specify the tag value in either of the following ways: ● Click the text box and select a predefined tag value from the drop-down list. ● Enter a tag value in the text box. **NOTE** A tag value can contain a maximum of 255 characters. Only letters, numbers, spaces, and special characters (_.:=+-@) are allowed. The value cannot start or end with a space.

NOTE

- A maximum of 20 tags can be added.
- Only one tag value can be added to a tag key.
- The key name in each resource must be unique.

5. Click **OK**.
6. (Optional) To delete a tag, locate the tag in the tag list and click **Delete** in its **Operation** column.

13.3.6 DLI Built-in Dependencies

DLI built-in dependencies are provided by the platform by default. In case of conflicts, you do not need to upload them when packaging JAR packages of Spark or Flink Jar jobs.

Spark 3.1.1 Dependencies

Table 13-10 Spark 3.1.1 dependencies

Dependency		
accessors-smart-1.2.jar	hive-shims-scheduler-3.1.0-h0.cbu.mrs.321.r10.jar	metrics-graphite-4.1.1.jar
activation-1.1.1.jar	hive-spark-client-3.1.0-h0.cbu.mrs.321.r10.jar	metrics-jmx-4.1.1.jar
aggdesigner-algorithm-6.0.jar	hive-standalone-metastore-3.1.0-h0.cbu.mrs.321.r10.jar	metrics-json-4.1.1.jar
aircompressor-0.16.jar	hive-storage-api-2.7.2.jar	metrics-jvm-4.1.1.jar
algebra_2.12-2.0.0-M2.jar	hive-vector-code-gen-3.1.0-h0.cbu.mrs.321.r10.jar	minlog-1.3.0.jar
annotations-17.0.0.jar	hk2-api-2.6.1.jar	netty-3.10.6.Final.jar
ant-1.10.9.jar	hk2-locator-2.6.1.jar	netty-all-4.1.86.Final.jar
ant-launcher-1.10.9.jar	hk2-utils-2.6.1.jar	netty-buffer-4.1.86.Final.jar
antlr4-runtime-4.8-1.jar	hppc-0.7.2.jar	netty-codec-4.1.86.Final.jar
antlr-runtime-3.5.2.jar	httpclient-4.5.6.jar	netty-codec-dns-4.1.86.Final.jar
aopalliance-1.0.jar	httpcore-4.4.10.jar	netty-codec-haproxy-4.1.86.Final.jar
aopalliance-repackaged-2.6.1.jar	istack-commons-runtime-3.0.8.jar	netty-codec-http2-4.1.86.Final.jar
apiguardian-api-1.1.0.jar	ivy-2.5.0.jar	netty-codec-http-4.1.86.Final.jar
arpack_combined_all-0.1.jar	jackson-annotations-2.13.2.jar	netty-codec-memcache-4.1.86.Final.jar
arrow-format-2.0.0.jar	jackson-core-2.13.2.jar	netty-codec-mqtt-4.1.86.Final.jar
arrow-memory-core-2.0.0.jar	jackson-core-asl-1.9.13-atlassian-4.jar	netty-codec-redis-4.1.86.Final.jar
arrow-memory-netty-2.0.0.jar	jackson-databind-2.13.2.2.jar	netty-codec-smtp-4.1.86.Final.jar

Dependency		
arrow-vector-2.0.0.jar	jackson-dataformat-yaml-2.13.2.jar	netty-codec-socks-4.1.86.Final.jar
asm-5.0.4.jar	jackson-datatype-jsr310-2.11.2.jar	netty-codec-stomp-4.1.86.Final.jar
audience-annotations-0.5.0.jar	jackson-mapper-asl-1.9.13-atlassian-4.jar	netty-codec-xml-4.1.86.Final.jar
automaton-1.11-8.jar	jackson-module-jaxb-annotations-2.13.2.jar	netty-common-4.1.86.Final.jar
avatica-1.22.0.jar	jackson-module-scala_2.12-2.13.2.jar	netty-handler-4.1.86.Final.jar
avatica-core-1.16.0.jar	jaeger-client-1.6.0.jar	netty-handler-proxy-4.1.86.Final.jar
avatica-metrics-1.16.0.jar	jaeger-core-1.6.0.jar	netty-handler-ssl-ocsp-4.1.86.Final.jar
avatica-server-1.16.0.jar	jaeger-thrift-1.6.0.jar	netty-resolver-4.1.86.Final.jar
avro-1.8.2.jar	jaeger-tracerresolver-1.6.0.jar	netty-resolver-dns-4.1.86.Final.jar
avro-ipc-1.8.2.jar	jakarta.activation-api-1.2.1.jar	netty-resolver-dns-classes-macos-4.1.86.Final.jar
avro-mapred-1.8.2.jar	jakarta.annotation-api-1.3.5.jar	netty-resolver-dns-native-macos-4.1.86.Final-osx-aarch_64.jar
java-sdk-bundle-1.11.856.jar	jakarta.el-3.0.3.jar	netty-resolver-dns-native-macos-4.1.86.Final-osx-x86_64.jar
base64-2.3.8.jar	jakarta.el-api-3.0.3.jar	netty-transport-4.1.86.Final.jar
bcpkix-jdk15on-1.69.jar	jakarta.inject-2.6.1.jar	netty-transport-classes-epoll-4.1.86.Final.jar
bcprov-jdk15on-1.69.jar	jakarta.servlet-api-4.0.3.jar	netty-transport-classes-kqueue-4.1.86.Final.jar
bcutil-jdk15on-1.69.jar	jakarta.validation-api-2.0.2.jar	netty-transport-native-epoll-4.1.86.Final-linux-aarch_64.jar

Dependency		
bonecp-0.8.0.RELEASE.jar	jakarta.ws.rs-api-2.1.6.jar	netty-transport-native-epoll-4.1.86.Final-linux-x86_64.jar
breeze_2.12-1.0.jar	jakarta.xml.bind-api-2.3.2.jar	netty-transport-native-kqueue-4.1.86.Final-osx-aarch_64.jar
breeze-macros_2.12-1.0.jar	jamon-runtime-2.4.1.jar	netty-transport-native-kqueue-4.1.86.Final-osx-x86_64.jar
caffeine-2.8.1.jar	janino-3.0.16.jar	netty-transport-native-unix-common-4.1.86.Final.jar
calcite-core-1.22.0.jar	JavaEWAH-0.3.2.jar	netty-transport-rxtx-4.1.86.Final.jar
calcite-druid-1.19.0.jar	java-sdk-core-3.0.12.jar	netty-transport-sctp-4.1.86.Final.jar
calcite-linq4j-1.22.0.jar	javassist-3.25.0-GA.jar	netty-transport-udt-4.1.86.Final.jar
cats-kernel_2.12-2.0.0-M4.jar	javax.activation-api-1.2.0.jar	nimbus-jose-jwt-8.19.jar
checker-qual-3.5.0.jar	javax.annotation-api-1.3.2.jar	objenesis-2.5.1.jar
chill_2.12-0.9.5.jar	javax.inject-1.jar	okhttp-3.14.9.jar
chill-java-0.9.5.jar	javax.jdo-3.2.0-m3.jar	okio-1.17.2.jar
classmate-1.5.1.jar	java-xmlbuilder-1.1.jar	opencsv-2.3.jar
commons-beanutils-1.9.4.jar	javax.servlet-api-3.1.0.jar	opentelemetry-api-1.16.0.jar
commons-cli-1.2.jar	javax.transaction-api-1.3.jar	opentelemetry-context-1.16.0.jar
commons-codec-1.15.jar	javax.ws.rs-api-2.1.1.jar	opentelemetry-semconv-1.16.0-alpha.jar
commons-collections-3.2.2.jar	javolution-5.5.1.jar	opentracing-api-0.33.0.jar
commons-compiler-3.0.16.jar	jaxb-api-2.2.11.jar	opentracing-noop-0.33.0.jar
commons-compress-1.21.jar	jaxb-runtime-2.3.2.jar	opentracing-tracerresolver-0.1.8.jar

Dependency		
commons-configuration2-2.1.1.jar	jboss-logging-3.4.1.Final.jar	opentracing-util-0.33.0.jar
commons-crypto-1.0.0-20191105.jar	jboss-threads-2.3.3.Final.jar	orc-core-1.6.8.jar
commons-daemon-1.0.13.jar	jcip-annotations-1.0-1.jar	orc-mapreduce-1.6.8.jar
commons-dbcp-1.4.jar	jcl-over-slf4j-1.7.36.jar	orc-shims-1.6.8.jar
commons-dbcp2-2.6.0.jar	jcodings-1.0.57.jar	orc-tools-1.6.7-h0.cbu.mrs.321.r10.jar
commons-digester-2.1.jar	jdo-api-3.2.jar	oro-2.0.8.jar
commons-httpclient-3.1.jar	jersey-client-2.34.jar	osgi-resource-locator-1.0.3.jar
commons-io-2.8.0.jar	jersey-common-2.34.jar	paranamer-2.8.jar
commons-lang-2.4.jar	jersey-container-servlet-2.34.jar	parquet-column-1.12.2.jar
commons-lang-2.6.jar	jersey-container-servlet-core-2.34.jar	parquet-common-1.12.2.jar
commons-lang3-3.10.jar	jersey-hk2-2.34.jar	parquet-encoding-1.12.2.jar
commons-logging-1.2.jar	jersey-server-2.34.jar	parquet-format-structures-1.12.2.jar
commons-math3-3.4.1.jar	jets3t-0.9.4-1.0.0.jar	parquet-hadoop-1.12.2.jar
commons-net-3.1.jar	jettison-1.1.jar	parquet-hadoop-bundle-1.12.0-ei-2.0.jar
commons-pool2-2.6.1.jar	jetty-http-9.4.41.v20210516.jar	parquet-jackson-1.12.2.jar
commons-text-1.10.0.jar	jetty-io-9.4.41.v20210516.jar	postgresql-42.3.5.jar
commons-validator-1.7.jar	jetty-rewrite-9.4.43.v20210629.jar	protobuf-java-2.5.0.jar
compress-lzf-1.0.3.jar	jetty-security-9.4.41.v20210516.jar	py4j-0.10.9.jar

Dependency		
core-1.1.2.jar	jetty-server-9.4.41.v20210516.jar	pyrolite-4.30.jar
curator-client-2.13.0.jar	jetty-servlet-9.4.41.v20210516.jar	re2j-1.1.jar
curator-framework-2.13.0.jar	jetty-util-9.4.41.v20210516.jar	RoaringBitmap-0.9.0.jar
curator-recipes-2.13.0.jar	jetty-util-ajax-9.4.41.v20210516.jar	scala-collection-compat_2.12-2.1.1.jar
datanucleus-api-jdo-4.2.4.jar	jetty-webapp-9.4.41.v20210516.jar	scala-compiler-2.12.16.jar
datanucleus-core-4.1.17.jar	jetty-xml-9.4.41.v20210516.jar	scala-library-2.12.16.jar
datanucleus-rdbms-fi-4.1.19-302022.jar	JLargeArrays-1.5.jar	scala-parser-combinators_2.12-1.1.2.jar
derby-10.14.2.0.jar	jline-3.21.0.jar	scala-reflect-2.12.16.jar
disruptor-3.4.2.jar	joda-time-2.10.5.jar	scala-xml_2.12-1.2.0.jar
dli-catalog-client-2.3.7-20240108.090504-101.jar	jodd-core-3.5.2.jar	secComponentApi-1.1.8.jar
dli-catalog-hive3-client-2.3.7-20240108.090513-100.jar	jodd-util-6.0.0.jar	serializer-2.7.2.jar
dli-catalog-hive-extension-2.3.7-20240108.090517-100.jar	joni-2.1.43.jar	shapeless_2.12-2.3.3.jar
dnsjava-2.1.7.jar	jpam-1.1.jar	shims-0.9.0.jar
dropwizard-metrics-hadoop-metrics2-reporter-0.1.2.jar	jsch-0.1.72.jar	sketches-core-0.9.0.jar
error_prone_annotations-2.18.0.jar	json-20210307.jar	slf4j-api-1.7.30.jar
esdk-obs-java-optimised-3.22.10.2.jar	json4s-ast_2.12-3.7.0-M5.jar	slf4j-log4j12-1.7.25.jar
esri-geometry-api-2.2.0.jar	json4s-core_2.12-3.7.0-M5.jar	snakeyaml-1.30.jar

Dependency		
fastutil-6.5.6.jar	json4s-jackson_2.12-3.7.0-M5.jar	snappy-java-1.1.8.2.jar
flatbuffers-java-1.9.0.jar	json4s-scalap_2.12-3.7.0-M5.jar	spark-avro_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar
generex-1.0.2.jar	json-path-2.4.0.jar	spark-catalyst_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar
glassfish-corba-omgapi-4.2.2.jar	json-smart-2.3.jar	spark-core_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar
gson-2.8.9.jar	jsr305-3.0.0.jar	spark-graphx_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar
gson-fire-1.8.5.jar	JTransforms-3.1.jar	spark-hive_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar
guava-14.0.1.jar	jul-to-slf4j-1.7.36.jar	spark-kubernetes_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar
guice-3.0.jar	kafka-clients-2.8.0.jar	spark-kvstore_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar
guice-assistedinject-3.0.jar	kerb-admin-2.0.2.jar	spark-launcher_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar
guice-servlet-4.0.jar	kerb-client-2.0.2.jar	spark-mllib_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar
hadoop-annotations-3.1.1-h0.cbu.mrs.313.r9.jar	kerb-common-2.0.2.jar	spark-mllib-local_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar
hadoop-archives-3.3.1-h0.cbu.mrs.321.r10.jar	kerb-core-2.0.2.jar	spark-network-common_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar
hadoop-auth-3.3.1-h0.cbu.mrs.321.r16.jar	kerb-crypto-2.0.2.jar	spark-network-shuffle_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar
hadoop-3.3.1-h0.cbu.mrs.321.r16.jar	kerb-identity-2.0.2.jar	spark-quota-manager_2.12-3.1.1-2.3.7.dli-SNAPSHOT.jar
hadoop-client-3.1.1-h0.cbu.mrs.313.r9.jar	kerb-server-2.0.2.jar	spark-repl_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar
hadoop-common-3.3.1-h0.cbu.mrs.321.r10.jar	kerb-simplekdc-2.0.2.jar	spark-sketch_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar

Dependency		
hadoop-distcp-3.3.1-h0.cbu.mrs.321.r10.jar	kerb-util-2.0.2.jar	spark-sql_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar
hadoop-hdfs-3.3.1-h0.cbu.mrs.321.r16.jar	kerby-asn1-2.0.2.jar	spark-streaming_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar
hadoop-hdfs-client-3.3.1-h0.cbu.mrs.321.r10.jar	kerby-config-2.0.2.jar	spark-tags_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar
hadoop-3.1.1-52.1.jar	kerby-pkix-2.0.2.jar	spark-unsafe_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar
hadoop-mapreduce-client-common-3.1.1-h0.cbu.mrs.313.r9.jar	kerby-util-2.0.2.jar	spark-uquery_2.12-3.1.1-2.3.7.dli-SNAPSHOT.jar
hadoop-mapreduce-client-core-3.1.1-h0.cbu.mrs.313.r9.jar	kerby-xdr-2.0.2.jar	spire_2.12-0.17.0-M1.jar
hadoop-mapreduce-client-jobclient-3.1.1-h0.cbu.mrs.313.r9.jar	kotlin-stdlib-1.4.21.jar	spire-macros_2.12-0.17.0-M1.jar
hadoop-mapreduce-client-nativetask-3.3.1-h0.cbu.mrs.321.r10.jar	kotlin-stdlib-common-1.4.21.jar	spire-platform_2.12-0.17.0-M1.jar
hadoop-registry-3.3.1-h0.cbu.mrs.321.r10.jar	kryo-shaded-4.0.2.jar	spire-util_2.12-0.17.0-M1.jar
hadoop-shaded-guava-1.1.1.jar	kubernetes-client-5.4.1-20211025.jar	sqlline-1.3.0.jar
hadoop-shaded-protobuf_3_7-1.1.1.jar	kubernetes-model-admissionregistration-5.4.1-20211025.jar	ST4-4.0.4.jar
hadoop-yarn-api-3.1.1-h0.cbu.mrs.313.r9.jar	kubernetes-model-apiextensions-5.4.1-20211025.jar	stax2-api-4.2.1.jar
hadoop-yarn-client-3.1.1-h0.cbu.mrs.313.r9.jar	kubernetes-model-apps-5.4.1-20211025.jar	stax-api-1.0.1.jar
hadoop-yarn-registry-3.3.1-h0.cbu.mrs.321.r10.jar	kubernetes-model-autoscaling-5.4.1-20211025.jar	stream-2.9.6.jar
hbase-asyncfs-2.4.14-h0.cbu.mrs.321.r10.jar	kubernetes-model-batch-5.4.1-20211025.jar	streamingClient

Dependency		
hbase-client-2.4.14-h0.cbu.mrs.321.r10.jar	kubernetes-model-certificates-5.4.1-20211025.jar	streamingClient010
hbase-common-2.4.14-h0.cbu.mrs.321.r10.jar	kubernetes-model-common-5.4.1-20211025.jar	swagger-annotations-2.2.8.jar
hbase-hadoop2-compat-2.4.14-h0.cbu.mrs.321.r10.jar	kubernetes-model-coordination-5.4.1-20211025.jar	tephra-api-0.6.0.jar
hbase-hadoop-compat-2.4.14-h0.cbu.mrs.321.r10.jar	kubernetes-model-core-5.4.1-20211025.jar	tephra-core-0.6.0.jar
hbase-http-2.4.14-h0.cbu.mrs.321.r10.jar	kubernetes-model-discovery-5.4.1-20211025.jar	tephra-hbase-compat-1.0-0.6.0.jar
hbase-logging-2.4.14-h0.cbu.mrs.321.r10.jar	kubernetes-model-events-5.4.1-20211025.jar	threetenbp-1.3.5.jar
hbase-metrics-2.4.14-h0.cbu.mrs.321.r10.jar	kubernetes-model-extensions-5.4.1-20211025.jar	threeten-extra-1.5.0.jar
hbase-metrics-api-2.4.14-h0.cbu.mrs.321.r10.jar	kubernetes-model-flowcontrol-5.4.1-20211025.jar	tink-1.6.0.jar
hbase-procedure-2.4.14-h0.cbu.mrs.321.r10.jar	kubernetes-model-metrics-5.4.1-20211025.jar	token-provider-2.0.2.jar
hbase-protocol-2.4.14-h0.cbu.mrs.321.r10.jar	kubernetes-model-networking-5.4.1-20211025.jar	tomcat-servlet-api-8.5.61.jar
hbase-protocol-shaded-2.4.14-h0.cbu.mrs.321.r10.jar	kubernetes-model-node-5.4.1-20211025.jar	transaction-api-1.1.jar
hbase-replication-2.4.14-h0.cbu.mrs.321.r10.jar	kubernetes-model-policy-5.4.1-20211025.jar	twill-api-0.6.0-incubating.jar
hbase-server-2.4.14-h0.cbu.mrs.321.r10.jar	kubernetes-model-rbac-5.4.1-20211025.jar	twill-common-0.6.0-incubating.jar
hbase-shaded-gson-4.1.4.jar	kubernetes-model-scheduling-5.4.1-20211025.jar	twill-core-0.6.0-incubating.jar

Dependency		
hbase-shaded-jersey-4.1.4.jar	kubernetes-model-storageclass-5.4.1-20211025.jar	twill-discovery-api-0.6.0-incubating.jar
hbase-shaded-jetty-4.1.4.jar	leveldbjni-all-1.8-20191105.jar	twill-discovery-core-0.6.0-incubating.jar
hbase-shaded-miscellaneous-4.1.4.jar	libfb303-0.9.3.jar	twill-zookeeper-0.6.0-incubating.jar
hbase-shaded-netty-4.1.4.jar	libthrift-0.14.1-ei-311001.jar	univocity-parsers-2.9.1.jar
hbase-shaded-protobuf-4.1.4.jar	log4j-1.2.17-cloudera1.jar	us-common-1.0.66.jar
hbase-unsafe-4.1.4.jar	log4j-api-2.17.1.jar	velocity-1.7.jar
hbase-zookeeper-2.4.14-h0.cbu.mrs.321.r10.jar	log4j-rolling-appender-20131024-2017.jar	velocity-engine-core-2.3.jar
hibernate-validator-6.2.5.Final.jar	logging-interceptor-3.14.9.jar	wildfly-client-config-1.0.1.Final.jar
HikariCP-2.6.1.jar	luxor-cluster-quota-manager-transport_2.12-2.3.7-20231226.034700-559.jar	wildfly-common-1.5.2.Final.jar
hive-classification-3.1.0-h0.cbu.mrs.321.r10.jar	luxor-encrypt-2.3.7-20231226.034423-1046.jar	woodstox-core-5.4.0.jar
hive-common-3.1.0-h0.cbu.mrs.321.r10.jar	luxor-fs3-2.3.7-20231226.034438-1039.jar	xalan-2.7.2.jar
hive-exec-3.1.0-h0.cbu.mrs.321.r10-core.jar	luxor-obs-fs3-2.3.7-20231226.034443-1038.jar	xbean-asm7-shaded-4.15.jar
hive-llap-client-2.3.3-ei-12-20210120.005053-2.jar	luxor-rpc_2.12-2.3.7-20231226.034653-560.jar	xercesImpl-2.12.2.jar
hive-llap-common-3.1.0-h0.cbu.mrs.321.r10.jar	luxor-scc-adapter-2.3.7-20231226.034418-1045.jar	xml-apis-1.4.01.jar
hive-llap-tez-3.1.0-h0.cbu.mrs.321.r10.jar	luxor-transport-2.3.7-20231226.034433-1038.jar	xnio-api-3.8.4.Final.jar

Dependency		
hive-metastore-3.1.0-h0.cbu.mrs.321.r10.jar	lz4-java-1.7.1.jar	xz-1.5.jar
hive-serde-3.1.0-h0.cbu.mrs.321.r10.jar	machinist_2.12-0.6.8.jar	zjsonpatch-0.3.0.jar
hive-service-rpc-3.1.0-h0.cbu.mrs.321.r10.jar	macro-compat_2.12-1.1.1.jar	zookeeper-3.5.6-ei-302002.jar
hive-shims-0.23-3.1.0-h0.cbu.mrs.321.r10.jar	memarts-ccsdk-1.0.jar	zookeeper-jute-3.5.6-ei-302002.jar
hive-shims-3.1.0-h0.cbu.mrs.321.r10.jar	memory-0.9.0.jar	zstd-jni-1.4.9-1.jar
hive-shims-common-3.1.0-h0.cbu.mrs.321.r10.jar	metrics-core-4.1.1.jar	-

Spark 2.4.5 Dependencies

Table 13-11 Spark 2.4.5 dependencies

Dependency		
JavaEWAH-1.1.7.jar	httpclient-4.5.6.jar	lucene-queryparser-7.7.2.jar
RoaringBitmap-0.7.45.jar	httpcore-4.4.10.jar	lucene-sandbox-7.7.2.jar
ST4-4.3.1.jar	ivy-2.4.0.jar	luxor-encrypt-2.0.0-20220623.010726-213.jar
accessors-smart-1.2.jar	jackson-annotations-2.11.4.jar	luxor-fs3-2.0.0-20220623.010750-209.jar
activation-1.1.1.jar	jackson-core-2.11.4.jar	luxor-obs-fs3-2.0.0-20220623.010756-209.jar
aircompressor-0.16.jar	jackson-core-asl-1.9.13-atlassian-4.jar	luxor-rpc_2.11-2.0.0-20220623.010737-182.jar
alluxio-2.3.1-luxor-SNAPSHOT-client.jar	jackson-databind-2.11.4.jar	luxor-transport-2.0.0-20220623.010744-71.jar

Dependency		
annotations-17.0.0.jar	jackson-dataformat-yaml-2.11.4.jar	lz4-java-1.7.1.jar
antlr-2.7.7.jar	jackson-datatype-jsr310-2.11.2.jar	machinist_2.11-0.6.1.jar
antlr-runtime-3.4.jar	jackson-jaxrs-base-2.10.3.jar	macro-compat_2.11-1.1.1.jar
antlr4-runtime-4.8-1.jar	jackson-jaxrs-json-provider-2.10.3.jar	metrics-core-3.1.5.jar
aopalliance-1.0.jar	jackson-mapper-asl-1.9.13-atlassian-4.jar	metrics-graphite-3.1.5.jar
aopalliance-repackaged-2.4.0-b34.jar	jackson-module-jaxb-annotations-2.10.3.jar	metrics-jmx-4.1.12.1.jar
apache-log4j-extras-1.2.17.jar	jackson-module-paranamer-2.11.4.jar	metrics-json-3.1.5.jar
arpack_combined_all-0.1.jar	jackson-module-scala_2.11-2.11.4.jar	metrics-jvm-3.1.5.jar
arrow-format-0.12.0.jar	jakarta.activation-api-1.2.1.jar	minlog-1.3.0.jar
arrow-memory-0.12.0.jar	jakarta.xml.bind-api-2.3.2.jar	mssql-jdbc-6.2.1.jre7.jar
arrow-vector-0.12.0.jar	janino-3.0.9.jar	netty-all-4.1.51.Final.jar
asm-5.0.4.jar	java-util-1.9.0.jar	nimbus-jose-jwt-8.19.jar
audience-annotations-0.5.0.jar	java-xmlbuilder-1.1.jar	objenesis-2.5.1.jar
automaton-1.11-8.jar	javassist-3.18.1-GA.jar	okhttp-3.14.9.jar
avro-1.8.2.jar	javax.annotation-api-1.2.jar	okio-1.17.2.jar
avro-ipc-1.8.2.jar	javax.inject-1.jar	opencsv-2.3.jar
avro-mapred-1.8.2.jar	javax.inject-2.4.0-b34.jar	opencsv-4.6.jar
java-sdk-bundle-1.11.856.jar	javax.servlet-api-3.1.0.jar	opencv-4.3.0-2.jar
base64-2.3.8.jar	javax.ws.rs-api-2.0.1.jar	orc-core-1.6.8-nohive.jar
bcpkix-jdk15on-1.66.jar	javolution-5.3.1.jar	orc-mapreduce-1.6.8-nohive.jar
bcprov-jdk15on-1.67.jar	jaxb-api-2.2.11.jar	orc-shims-1.6.8.jar

Dependency		
bonecp-0.8.0.RELEASE.jar	jcip-annotations-1.0-1.jar	oro-2.0.8.jar
breeze-macros_2.11-0.13.2.jar	jcl-over-slf4j-1.7.30.jar	osgi-resource-locator-1.0.1.jar
breeze_2.11-0.13.2.jar	jdo-api-3.0.1.jar	paranamer-2.8.jar
calcite-avatica-1.2.0-incubating.jar	jersey-client-2.23.1.jar	parquet-column-1.12.2.jar
chill-java-0.9.3.jar	jersey-common-2.23.1.jar	parquet-common-1.12.2.jar
chill_2.11-0.9.3.jar	jersey-container-servlet-2.23.1.jar	parquet-encoding-1.12.2.jar
commons-beanutils-1.9.4.jar	jersey-container-servlet-core-2.23.1.jar	parquet-format-structures-1.12.2.jar
commons-cli-1.2.jar	jersey-guava-2.23.1.jar	parquet-hadoop-1.12.2.jar
commons-codec-1.15.jar	jersey-media-jaxb-2.23.1.jar	parquet-hadoop-bundle-1.6.0.jar
commons-collections-3.2.2.jar	jersey-server-2.23.1.jar	parquet-jackson-1.12.2.jar
commons-collections4-4.2.jar	jets3t-0.9.4.jar	postgresql-42.2.14.jar
commons-compiler-3.0.9.jar	jettison-1.1.jar	protobuf-java-2.5.0.jar
commons-compress-1.4.1.jar	jetty-http-9.4.34.v20201102.jar	py4j-0.10.7.jar
commons-configuration2-2.1.1.jar	jetty-io-9.4.34.v20201102.jar	pyrolite-4.13.jar
commons-crypto-1.0.0-20191105.jar	jetty-security-9.4.34.v20201102.jar	re2j-1.1.jar
commons-daemon-1.0.13.jar	jetty-server-9.4.34.v20201102.jar	scala-compiler-2.11.12.jar
commons-dbcp2-2.7.0.jar	jetty-servlet-9.4.34.v20201102.jar	scala-library-2.11.12.jar
commons-httpclient-3.1.jar	jetty-util-9.4.34.v20201102.jar	scala-parser-combinators_2.11-1.1.2.jar

Dependency		
commons-io-2.5.jar	jetty-util-ajax-9.4.34.v20201102.jar	scala-reflect-2.11.12.jar
commons-lang-2.6.jar	jetty-webapp-9.4.34.v20201102.jar	scala-xml_2.11-1.0.5.jar
commons-lang3-3.5.jar	jetty-xml-9.4.34.v20201102.jar	secComponentApi-1.0.6.jar
commons-logging-1.2.jar	joda-time-2.9.3.jar	shapeless_2.11-2.3.2.jar
commons-math3-3.4.1.jar	jodd-core-3.5.2.jar	shims-0.7.45.jar
commons-net-3.1.jar	json-20200518.jar	slf4j-api-1.7.30.jar
commons-pool2-2.8.0.jar	json-io-2.5.1.jar	slf4j-log4j12-1.7.30.jar
commons-text-1.3.jar	json-sanitizer-1.2.1.jar	snakeyaml-1.26.jar
compress-lzf-1.0.3.jar	json-smart-2.3.jar	snappy-java-1.1.8.2.jar
core-1.1.2.jar	json4s-ast_2.11-3.5.3.jar	solr-core-7.7.2.jar
crypter-0.0.6.jar	json4s-core_2.11-3.5.3.jar	solr-solrj-7.7.2.jar
curator-client-4.2.0.jar	json4s-jackson_2.11-3.5.3.jar	spark-avro_2.11-2.4.5.0100-2.0.0.dli-20220617.085536-9.jar
curator-framework-4.2.0.jar	json4s-scalap_2.11-3.5.3.jar	spark-avro_2.11-4.0.0.jar
curator-recipes-2.7.1.jar	jsp-api-2.1.jar	spark-catalyst_2.11-2.4.5.0100-2.0.0.dli-20220617.085405-16.jar
datanucleus-api-jdo-3.2.6.jar	jsr305-1.3.9.jar	spark-core_2.11-2.4.5.0100-2.0.0.dli-20220617.085327-16.jar
datanucleus-core-3.2.10.jar	jta-1.1.jar	spark-graphx_2.11-2.4.5.0100-.0.dli-20220617.085336-16.jar

Dependency		
datanucleus-rdbms-3.2.9.jar	jtransforms-2.4.0.jar	spark-hive_2.11-2.4.5.0100-2.0.0.dli-20220617.085423-16.jar
derby-10.14.2.0.jar	jts-core-1.16.1.jar	spark-kubernetes_2.11-2.4.5.0100-2.0.0.dli-20220617.085519-16.jar
dnsjava-2.1.7.jar	jul-to-slf4j-1.7.30.jar	spark-kvstore_2.11-2.4.5.0100-2.0.0.dli-20220617.085249-16.jar
ecj-3.21.0.jar	junit-4.11.jar	spark-launcher_2.11-2.4.5.0100-2.0.0.dli-20220617.085435-16.jar
ehcache-3.3.1.jar	kerb-admin-1.0.1.jar	spark-mllib-local_2.11-2.4.5.0100-2.0.0.dli-20220617.085349-16.jar
expiringmap-0.5.9.jar	kerb-client-1.0.1.jar	spark-mllib_2.11-2.4.5.0100-2.0.0.dli-20220617.085342-16.jar
fastutil-8.2.3.jar	kerb-common-1.0.1.jar	spark-network-common_2.11-2.4.5.0100-2.0.0.dli-20220617.085254-16.jar
flatbuffers-java-1.9.0.jar	kerb-core-1.0.1.jar	spark-network-shuffle_2.11-2.4.5.0100-2.0.0.dli-20220617.085300-16.jar
fst-2.50.jar	kerb-crypto-1.0.1.jar	spark-om_2.11-2.4.5.0100-2.0.0.dli-20220617.085316-16.jar
generex-1.0.2.jar	kerb-identity-1.0.1.jar	spark-repl_2.11-2.4.5.0100-2.0.0.dli-20220617.085430-16.jar

Dependency		
geronimo-jcache_1.0_spec-1.0-alpha-1.jar	kerb-server-1.0.1.jar	spark-sketch_2.11-2.4.5.0100-2.0.0.dli-20220617.085243-16.jar
gson-2.2.4.jar	kerb-simplekdc-1.0.1.jar	spark-sql_2.11-2.4.5.0100-2.0.0.dli-20220617.085414-16.jar
guava-14.0.1.jar	kerb-util-1.0.1.jar	spark-streaming_2.11-2.4.5.0100-.0.dli-20220617.085359-16.jar
guice-4.0.jar	kerby-asn1-1.0.1.jar	spark-tags_2.11-2.4.5.0100-2.0.0.dli-20220617.085322-16.jar
guice-servlet-4.0.jar	kerby-config-1.0.1.jar	spark-unsafe_2.11-2.4.5.0100-2.0.0.dli-20220617.085311-16.jar
hadoop-annotations-3.1.1-ei-302002.jar	kerby-pkix-1.0.1.jar	spark-uquery_2.11-2.4.5.0100-2.0.0.dli-SNAPSHOT.jar
hadoop-auth-3.1.1-ei-302002.jar	kerby-util-1.0.1.jar	spark-yarn_2.11-2.4.5.0100-2.0.0.dli-20220617.085531-16.jar
hadoop-3.1.1-ei-302002.jar	kerby-xdr-1.0.1.jar	spire-macros_2.11-0.13.0.jar
hadoop-client-3.1.1-ei-302002.jar	kryo-shaded-4.0.2.jar	spire_2.11-0.13.0.jar
hadoop-common-3.1.1-ei-302002.jar	kubernetes-client-5.4.1-20211025.jar	stax-api-1.0-2.jar
hadoop-hdfs-3.1.1-ei-302002.jar	kubernetes-model-admissionregistration-5.4.1-20211025.jar	stax2-api-3.1.4.jar
hadoop-hdfs-client-3.1.1-ei-302002.jar	kubernetes-model-apiextensions-5.4.1-20211025.jar	stream-2.7.0.jar
hadoop-3.1.1-46.jar	kubernetes-model-apps-5.4.1-20211025.jar	stringtemplate-3.2.1.jar

Dependency		
hadoop-mapreduce-client-common-3.1.1-ei-302002.jar	kubernetes-model-autoscaling-5.4.1-20211025.jar	threeten-extra-1.5.0.jar
hadoop-mapreduce-client-core-3.1.1-ei-302002.jar	kubernetes-model-batch-5.4.1-20211025.jar	tink-1.6.0.jar
hadoop-mapreduce-client-jobclient-3.1.1-ei-302002.jar	kubernetes-model-certificates-5.4.1-20211025.jar	token-provider-1.0.1.jar
hadoop-minikdc-3.1.1-ei-302002.jar	kubernetes-model-common-5.4.1-20211025.jar	tomcat-api-9.0.39.jar
hadoop-yarn-api-3.1.1-ei-302002.jar	kubernetes-model-coordination-5.4.1-20211025.jar	zookeeper-jute-3.5.6-ei-302002.jar
hadoop-yarn-client-3.1.1-ei-302002.jar	kubernetes-model-core-5.4.1-20211025.jar	tomcat-el-api-9.0.39.jar
hadoop-yarn-common-3.1.1-ei-302002.jar	kubernetes-model-discovery-5.4.1-20211025.jar	tomcat-jasper-9.0.39.jar
hadoop-yarn-registry-3.1.1-ei-302002.jar	kubernetes-model-events-5.4.1-20211025.jar	tomcat-jasper-el-9.0.39.jar
hadoop-yarn-server-applicationhistoryservice-3.1.1-ei-302002.jar	kubernetes-model-extensions-5.4.1-20211025.jar	tomcat-jsp-api-9.0.39.jar
hadoop-yarn-server-common-3.1.1-ei-302002.jar	kubernetes-model-flowcontrol-5.4.1-20211025.jar	tomcat-juli-9.0.39.jar
hadoop-yarn-server-resourcemanager-3.1.1-ei-302002.jar	kubernetes-model-metrics-5.4.1-20211025.jar	tomcat-servlet-api-9.0.39.jar
hadoop-yarn-server-web-proxy-3.1.1-ei-302002.jar	kubernetes-model-networking-5.4.1-20211025.jar	tomcat-util-9.0.39.jar
hamcrest-core-1.3.jar	kubernetes-model-node-5.4.1-20211025.jar	tomcat-util-scan-9.0.39.jar
hive-common-1.2.1-2.0.0.dli-20220528.090500-402.jar	kubernetes-model-policy-5.4.1-20211025.jar	univocity-parsers-2.7.3.jar

Dependency		
hive-exec-1.2.1-2.0.0.dli-20220528.090521-401.jar	kubernetes-model-rbac-5.4.1-20211025.jar	zstd-jni-1.4.9-1.jar
hive-metastore-1.2.1-2.0.0.dli-20220528.090509-402.jar	kubernetes-model-scheduling-5.4.1-20211025.jar	validation-api-1.1.0.Final.jar
hive-shims-0.23-1.2.1-2.0.0.dli-20220528.090445-403.jar	kubernetes-model-storageclass-5.4.1-20211025.jar	velocity-1.7.jar
hive-shims-1.2.1-2.0.0.dli-20220528.090455-403.jar	leveldbjni-all-1.8-20191105.jar	woodstox-core-5.0.3.jar
hive-shims-common-1.2.1-2.0.0.dli-20220528.090441-404.jar	libfb303-0.9.3.jar	xbean-asm6-shaded-4.8.jar
hive-shims-scheduler-1.2.1-2.0.0.dli-20220528.090450-403.jar	libthrift-0.12.0.jar	xercesImpl-2.12.0.jar
hk2-api-2.4.0-b34.jar	log4j-1.2.17-cloudera1.jar	xml-apis-1.4.01.jar
hk2-locator-2.4.0-b34.jar	log4j-rolling-appender-20131024-2017.jar	xz-1.0.jar
hk2-utils-2.4.0-b34.jar	logging-interceptor-3.14.9.jar	zjsonpatch-0.3.0.jar
hppc-0.7.2.jar	lucene-analyzers-common-7.7.2.jar	zookeeper-3.5.6-ei-302002.jar
htrace-core4-4.2.0-incubating-1.0.0.jar	lucene-core-7.7.2.jar	-

Spark 2.3.2 Dependencies

Table 13-12 Spark 2.3.2 dependencies

Dependency		
accessors-smart-1.2.jar	HikariCP-java7-2.4.12.jar	logging-interceptor-3.14.4.jar

Dependency		
activation-1.1.1.jar	hive-common-1.2.1-2.1.0.dli-20201111.064115-91.jar	luxor-encrypt-2.1.0-20201106.065437-53.jar
aircompressor-0.8.jar	hive-exec-1.2.1-2.1.0.dli-20201111.064444-91.jar	luxor-fs3-2.1.0-20201106.065612-53.jar
alluxio-2.3.1-luxor-SNAPSHOT-client.jar	hive-metastore-1.2.1-2.1.0.dli-20201111.064230-91.jar	luxor-obs-fs3-2.1.0-20201106.065616-53.jar
antlr-2.7.7.jar	hk2-api-2.4.0-b34.jar	luxor-rpc_2.11-2.1.0-20201106.065541-53.jar
antlr4-runtime-4.8-1.jar	hk2-locator-2.4.0-b34.jar	luxor-rpc-protobuf2-2.1.0-20201106.065551-53.jar
antlr-runtime-3.4.jar	hk2-utils-2.4.0-b34.jar	lz4-java-1.7.1.jar
aopalliance-1.0.jar	hppc-0.7.2.jar	machinist_2.11-0.6.1.jar
aopalliance-repackaged-2.4.0-b34.jar	htrace-core4-4.2.0-incubating-1.0.0.jar	macro-compat_2.11-1.1.1.jar
apache-log4j-extras-1.2.17.jar	httpclient-4.5.4.jar	metrics-core-3.1.5.jar
arpack_combined_all-0.1.jar	httpcore-4.4.7.jar	metrics-graphite-3.1.5.jar
arrow-format-0.8.0.jar	ivy-2.4.0.jar	metrics-jmx-4.1.12.1.jar
arrow-memory-0.8.0.jar	j2objc-annotations-1.3.jar	metrics-json-3.1.5.jar
arrow-vector-0.8.0.jar	jackson-annotations-2.10.0.jar	metrics-jvm-3.1.5.jar
asm-5.0.4.jar	jackson-core-2.10.0.jar	minlog-1.3.0.jar
audience-annotations-0.5.0.jar	jackson-core-asl-1.9.13-atlassian-4.jar	mssql-jdbc-6.2.1.jre7.jar
automaton-1.11-8.jar	jackson-databind-2.10.0.jar	netty-3.10.6.Final.jar
avro-1.7.7.jar	jackson-dataformat-yaml-2.10.0.jar	netty-all-4.1.51.Final.jar
avro-ipc-1.7.7.jar	jackson-datatype-jsr310-2.10.3.jar	nimbus-jose-jwt-8.19.jar
avro-ipc-1.7.7-tests.jar	jackson-jaxrs-base-2.10.3.jar	objenesis-2.1.jar

Dependency		
avro-mapred-1.7.7-hadoop2.jar	jackson-jaxrs-json-provider-2.10.3.jar	okhttp-3.14.4.jar
java-sdk-bundle-1.11.271.jar	jackson-mapper-asl-1.9.13-atlassian-4.jar	okio-1.17.2.jar
base64-2.3.8.jar	jackson-module-jaxb-annotations-2.10.3.jar	opencsv-2.3.jar
bcpkix-jdk15on-1.66.jar	jackson-module-paranamer-2.10.0.jar	opencsv-4.6.jar
bcprov-jdk15on-1.66.jar	jackson-module-scala_2.11-2.10.0.jar	opencv-4.3.0-2.jar
bonecp-0.8.0.RELEASE.jar	jakarta.activation-api-1.2.1.jar	orc-core-1.4.4-nohive.jar
breeze_2.11-0.13.2.jar	jakarta.xml.bind-api-2.3.2.jar	orc-mapreduce-1.4.4-nohive.jar
breeze-macros_2.11-0.13.2.jar	janino-3.0.8.jar	oro-2.0.8.jar
calcite-avatica-1.2.0-incubating.jar	javacpp-1.5.4.jar	osgi-resource-locator-1.0.1.jar
calcite-core-1.2.0-incubating.jar	javacpp-1.5.4-linux-x86_64.jar	paranamer-2.8.jar
calcite-linq4j-1.2.0-incubating.jar	javacv-1.5.4.jar	parquet-column-1.8.3.jar
checker-qual-2.11.1.jar	JavaEWAH-1.1.7.jar	parquet-common-1.8.3.jar
chill_2.11-0.8.4.jar	javassist-3.18.1-GA.jar	parquet-encoding-1.8.3.jar
chill-java-0.8.4.jar	javax.annotation-api-1.2.jar	parquet-format-2.3.1.jar
commons-beanutils-1.9.4.jar	javax.inject-1.jar	parquet-hadoop-1.8.3.jar
commons-cli-1.2.jar	javax.inject-2.4.0-b34.jar	parquet-hadoop-bundle-1.6.0.jar
commons-codec-2.0-20130428.202122-59.jar	javax.servlet-api-3.1.0.jar	parquet-jackson-1.8.3.jar
commons-collections-3.2.2.jar	javax.ws.rs-api-2.0.1.jar	parquet-format-2.3.1.jar

Dependency		
commons-collections4-4.2.jar	java-xmlbuilder-1.1.jar	parquet-hadoop-1.8.3.jar
commons-compiler-3.0.8.jar	javolution-5.3.1.jar	parquet-hadoop-bundle-1.6.0.jar
commons-compress-1.4.1.jar	jaxb-api-2.2.11.jar	parquet-jackson-1.8.3.jar
commons-configuration2-2.1.1.jar	jcip-annotations-1.0-1.jar	postgresql-42.2.14.jar
commons-crypto-1.0.0-20191105.jar	jcl-over-slf4j-1.7.26.jar	protobuf-java-2.5.0.jar
commons-daemon-1.0.13.jar	jdo-api-3.0.1.jar	py4j-0.10.7.jar
commons-dbcp-1.4.jar	jersey-client-2.23.1.jar	pyrolite-4.13.jar
commons-dbcp2-2.7.0.jar	jersey-common-2.23.1.jar	re2j-1.1.jar
commons-httpclient-3.1.jar	jersey-container-servlet-2.23.1.jar	RoaringBitmap-0.5.11.jar
commons-io-2.5.jar	jersey-container-servlet-core-2.23.1.jar	scala-compiler-2.11.12.jar
commons-lang-2.6.jar	jersey-guava-2.23.1.jar	scala-library-2.11.12.jar
commons-lang3-3.5.jar	jersey-media-jaxb-2.23.1.jar	scalap-2.11.0.jar
commons-logging-1.2.jar	jersey-server-2.23.1.jar	scala-parser-combinators_2.11-1.1.0.jar
commons-math3-3.4.1.jar	jets3t-0.9.4.jar	scala-reflect-2.11.12.jar
commons-net-2.2.jar	jetty-http-9.4.31.v20200723.jar	scala-xml_2.11-1.0.5.jar
commons-pool-1.5.4.jar	jetty-io-9.4.31.v20200723.jar	secComponentApi-1.0.5c.jar
commons-pool2-2.8.0.jar	jetty-security-9.4.31.v20200723.jar	shapeless_2.11-2.3.2.jar
commons-text-1.3.jar	jetty-server-9.4.31.v20200723.jar	slf4j-api-1.7.30.jar
compress-lzf-1.0.3.jar	jetty-servlet-9.4.31.v20200723.jar	slf4j-log4j12-1.7.30.jar

Dependency		
core-1.1.2.jar	jetty-util-9.4.31.v20200723.jar	snakeyaml-1.24.jar
curator-client-4.2.0.jar	jetty-util-ajax-9.4.31.v20200723.jar	snappy-java-1.1.7.5.jar
curator-framework-4.2.0.jar	jetty-webapp-9.4.31.v20200723.jar	spark-catalyst_2.11-2.3.2.0101-2.1.0.dli-20201111.073826-143.jar
curator-recipes-2.7.1.jar	jetty-xml-9.4.31.v20200723.jar	spark-core_2.11-2.3.2.0101-.0.dli-20201111.073836-134.jar
datanucleus-api-jdo-3.2.6.jar	joda-time-2.9.3.jar	spark-graphx_2.11-2.3.2.0101-2.1.0.dli-20201111.073847-129.jar
datanucleus-core-3.2.10.jar	jodd-core-4.2.0.jar	spark-hive_2.11-2.3.2.0101-.0.dli-20201111.073854-132.jar
datanucleus-rdbms-3.2.9.jar	json-20200518.jar	spark-kubernetes_2.11-2.3.2.0101-2.1.0.dli-20201111.073916-85.jar
derby-10.12.1.1.jar	json4s-ast_2.11-3.2.11.jar	spark-kvstore_2.11-2.3.2.0101-2.1.0.dli-20201111.073933-127.jar
dnsjava-2.1.7.jar	json4s-core_2.11-3.2.11.jar	spark-launcher_2.11-2.3.2.0101-2.1.0.dli-20201111.073940-127.jar
ehcache-3.3.1.jar	json4s-jackson_2.11-3.2.11.jar	spark-mllib_2.11-2.3.2.0101-2.1.0.dli-20201111.073946-127.jar
eigenbase-properties-1.1.5.jar	json-sanitizer-1.2.1.jar	spark-mllib-local_2.11-2.3.2.0101-2.1.0.dli-20201111.073953-127.jar

Dependency		
error_prone_annotations-2.3.4.jar	json-smart-2.3.jar	spark-network-common_2.11-2.3.2.0101-2.1.0.dli-20201111.073959-127.jar
failureaccess-1.0.1.jar	jsp-api-2.1.jar	spark-network-shuffle_2.11-2.3.2.0101-2.1.0.dli-20201111.074007-127.jar
fastutil-8.2.3.jar	jsr305-3.0.2.jar	spark-om_2.11-2.3.2.0101-.0.dli-20201111.074019-125.jar
ffmpeg-4.3.1-1.5.4.jar	jta-1.1.jar	spark-repl_2.11-2.3.2.0101-2.1.0.dli-20201111.074028-125.jar
ffmpeg-4.3.1-1.5.4-linux-x86_64.jar	jtransforms-2.4.0.jar	spark-sketch_2.11-2.3.2.0101-2.1.0.dli-20201111.074035-125.jar
flatbuffers-1.2.0-3f79e055.jar	jul-to-slf4j-1.7.26.jar	spark-sql_2.11-2.3.2.0101-2.1.0.dli-20201111.074041-126.jar
generex-1.0.2.jar	junit-4.11.jar	spark-streaming_2.11-2.3.2.0101-2.1.0.dli-20201111.074100-123.jar
geronimo-jcache_1.0_spec-1.0-alpha-1.jar	kerb-admin-1.0.1.jar	spark-tags_2.11-2.3.2.0101-2.1.0.dli-20201111.074136-123.jar
gson-2.2.4.jar	kerb-client-1.0.1.jar	spark-tags_2.11-2.3.2.0101-2.1.0.dli-20201111.074141-124-tests.jar
guava-29.0-jre.jar	kerb-common-1.0.1.jar	spark-unsafe_2.11-2.3.2.0101-2.1.0.dli-20201111.074144-123.jar

Dependency		
guice-4.0.jar	kerb-core-1.0.1.jar	spark-uquery_2.11-2.3.2.0101-2.1.0.dli-20201111.074906-210.jar
guice-servlet-4.0.jar	kerb-crypto-1.0.1.jar	spark-yarn_2.11-2.3.2.0101-2.1.0.dli-20201111.074151-123.jar
hadoop-annotations-3.1.1-ei-302002.jar	kerb-identity-1.0.1.jar	spire_2.11-0.13.0.jar
hadoop-auth-3.1.1-ei-302002.jar	kerb-server-1.0.1.jar	spire-macros_2.11-0.13.0.jar
hadoop-3.1.1-ei-302002.jar	kerb-simplekdc-1.0.1.jar	ST4-4.3.1.jar
hadoop-client-3.1.1-ei-302002.jar	kerb-util-1.0.1.jar	stax2-api-3.1.4.jar
hadoop-common-3.1.1-ei-302002.jar	kerby-asn1-1.0.1.jar	stax-api-1.0-2.jar
hadoop-hdfs-3.1.1-ei-302002.jar	kerby-config-1.0.1.jar	stream-2.7.0.jar
hadoop-hdfs-client-3.1.1-ei-302002.jar	kerby-pkix-1.0.1.jar	stringtemplate-3.2.1.jar
hadoop-3.1.1-41.jar	kerby-util-1.0.1.jar	token-provider-1.0.1.jar
hadoop-mapreduce-client-common-3.1.1-ei-302002.jar	kerby-xdr-1.0.1.jar	univocity-parsers-2.5.9.jar
hadoop-mapreduce-client-core-3.1.1-ei-302002.jar	kryo-shaded-3.0.3.jar	validation-api-1.1.0.Final.jar
hadoop-mapreduce-client-jobclient-3.1.1-ei-302002.jar	kubernetes-client-4.9.2-20200804.jar	woodstox-core-5.0.3.jar
hadoop-minikdc-3.1.1-ei-302002.jar	kubernetes-model-4.9.2-20200804.jar	xbean-asm5-shaded-4.4.jar
hadoop-yarn-api-3.1.1-ei-302002.jar	kubernetes-model-common-4.9.2-20200804.jar	xercesImpl-2.12.0.jar
hadoop-yarn-client-3.1.1-ei-302002.jar	leveldbjni-all-1.8-20191105.jar	xml-apis-1.4.01.jar

Dependency		
hadoop-yarn-common-3.1.1-ei-302002.jar	libfb303-0.9.3.jar	xz-1.0.jar
hadoop-yarn-registry-3.1.1-ei-302002.jar	libthrift-0.12.0.jar	zjsonpatch-0.3.0.jar
hadoop-yarn-server-common-3.1.1-ei-302002.jar	listenablefuture-9999.0-empty-to-avoid-conflict-with-guava.jar	zookeeper-3.5.6-ei-302002.jar
hadoop-yarn-server-web-proxy-3.1.1-ei-302002.jar	log4j-1.2.17-cloudera1.jar	zookeeper-jute-3.5.6-ei-302002.jar
hamcrest-core-1.3.jar	log4j-rolling-appender-20131024-2017.jar	zstd-jni-1.4.4-11.jar

Flink 1.15 Dependencies

Obtain information about the Flink 1.15 dependencies from the logs of a Flink job.

1. Check the logs of a Flink job.
   a. Log in to the DLI console. In the navigation pane on the left, choose **Job Management** > **Flink Jobs**.
   b. Click the name of the desired job. On the displayed page, click the **Run Log** tab.
   c. Check the latest run logs. For more logs, check the OBS bucket where the job logs are stored.
2. Search for dependency information in the logs.

   Search for **Classpath:** in the logs to check the dependencies.

Flink 1.12 Dependencies

Table 13-13 Flink 1.12 dependencies

Dependency		
bcpkix-jdk15on-1.60.jar	flink-json-1.12.2-ei-313001-dli-2022011002.jar	libtensorflow-1.12.0.jar
bcprov-jdk15on-1.60.jar	flink-kubernetes_2.11-1.12.2-ei-313001-dli-2022011002.jar	log4j-1.2-api-2.17.1.jar

Dependency		
clickhouse-jdbc-0.3.1-ei-313001-SNAPSHOT.jar	flink-metrics-prometheus_2.11-1.12.2-ei-313001-dli-2022011002.jar	log4j-api-2.17.1.jar
commons-codec-1.9.jar	flink-obs-hadoop-fs-2.0.0-20220226.034421-73.jar	log4j-core-2.17.1.jar
commons-configuration-1.7.jar	flink-s3-fs-hadoop-1.12.2.jar	log4j-slf4j-impl-2.17.1.jar
dataflow-fs-obs-2.0.0-20220226.034402-190.jar	flink-shaded-zookeeper-3.6.3-ei-313001-SNAPSHOT.jar	luxor-encrypt-2.0.0-20220405.072004-199.jar
deeplearning4j-core-0.9.1.jar	flink-sql-avro-1.12.2-ei-313001-dli-2022011002.jar	luxor-fs3-2.0.0-20220405.072025-195.jar
deeplearning4j-nlp-0.9.1.jar	flink-sql-avro-confluent-registry-1.12.2-ei-313001-dli-2022011002.jar	luxor-obs-fs3-2.0.0-20220405.072030-195.jar
deeplearning4j-nn-0.9.1.jar	flink-table_2.11-1.12.2-ei-313001-dli-2022011002.jar	manager-hadoop-security-crypter-8.1.3-313001-SNAPSHOT.jar
ejml-cdense-0.33.jar	flink-table-blink_2.11-1.12.2-ei-313001-dli-2022011002.jar	manager-wc2frm-8.1.3-313001-SNAPSHOT.jar
ejml-core-0.33.jar	guava-18.0.jar	mrs-obs-provider-3.1.1.49.jar
ejml-ddense-0.33.jar	guava-26.0-jre.jar	nd4j-api-0.9.1.jar
ejml-dsparse-0.33.jar	hadoop-hdfs-client-3.1.1-ei-302002.jar	nd4j-native-0.9.1.jar
ejml-experimental-0.33.jar	hadoop-3.1.1-46.jar	nd4j-native-api-0.9.1.jar
ejml-fdense-0.33.jar	hadoop-plugins-8.1.3-313001-SNAPSHOT.jar	nd4j-native-platform-0.9.1.jar
ejml-simple-0.33.jar	httpasyncclient-4.1.2.jar	okhttp-3.14.8.jar
ejml-zdense-0.33.jar	httpclient-4.5.3.jar	okio-1.14.0.jar

Dependency		
elsa-3.0.0-M7.jar	httpcore-4.4.4.jar	ranger-obs-client-0.1.1.jar
flink-changelog-json-1.12.2-ei-313001-dli-2022011002.jar	httpcore-nio-4.4.4.jar	secComponentApi-1.0.5.jar
flink-csv-1.12.2-ei-313001-dli-2022011002.jar	java-xmlbuilder-1.1.jar	slf4j-api-1.7.26.jar
flink-dist_2.11-1.12.2-ei-313001-dli-2022011002.jar	jna-4.1.0.jar	tensorflow-1.12.0.jar

Flink 1.10 Dependencies

For details about the sample code for a Flink 1.10 program, see **Writing Data to OBS Using Flink Jar**.

Only queues created after December 2020 can use the Flink 1.10 dependencies.

Table 13-14 Flink 1.10 dependencies

Dependency		
bcpkix-jdk15on-1.60.jar	esdk-obs-java-3.20.6.1.jar	java-xmlbuilder-1.1.jar
bcprov-jdk15on-1.60.jar	flink-cep_2.11-1.10.0.jar	jna-4.1.0.jar
commons-codec-1.9.jar	flink-cep-scala_2.11-1.10.0.jar	libtensorflow-1.12.0.jar
commons-configuration-1.7.jar	flink-dist_2.11-1.10.0.jar	log4j-over-slf4j-1.7.26.jar
deeplearning4j-core-0.9.1.jar	flink-python_2.11-1.10.0.jar	logback-classic-1.2.3.jar
deeplearning4j-nlp-0.9.1.jar	flink-queryable-state-runtime_2.11-1.10.0.jar	logback-core-1.2.3.jar
deeplearning4j-nn-0.9.1.jar	flink-sql-client_2.11-1.10.0.jar	nd4j-api-0.9.1.jar
ejml-cdense-0.33.jar	flink-state-processor-api_2.11-1.10.0.jar	nd4j-native-0.9.1.jar
ejml-core-0.33.jar	flink-table_2.11-1.10.0.jar	nd4j-native-api-0.9.1.jar
ejml-ddense-0.33.jar	flink-table-blink_2.11-1.10.0.jar	nd4j-native-platform-0.9.1.jar

Dependency		
ejml-dsparse-0.33.jar	guava-26.0-jre.jar	okhttp-3.14.8.jar
ejml-experimental-0.33.jar	hadoop-3.1.1-41.jar	okio-1.14.0.jar
ejml-fdense-0.33.jar	httpasyncclient-4.1.2.jar	secComponentApi-1.0.5.jar
ejml-simple-0.33.jar	httpclient-4.5.3.jar	slf4j-api-1.7.26.jar
ejml-zdense-0.33.jar	httpcore-4.4.4.jar	tensorflow-1.12.0.jar
elsa-3.0.0-M7.jar	httpcore-nio-4.4.4.jar	-

Flink 1.7.2 Dependencies

For details about the sample code of a Flink 1.7.2 program, see **luxor-demo\dli-flink-demo** in **DLI examples**.

Table 13-15 Flink 1.7.2 dependencies

Dependency		
bcpkix-jdk15on-1.60.jar	esdk-obs-java-3.1.3.jar	httpcore-4.4.4.jar
bcprov-jdk15on-1.60.jar	flink-cep_2.11-1.7.0.jar	httpcore-nio-4.4.4.jar
commons-codec-1.9.jar	flink-cep-scala_2.11-1.7.0.jar	java-xmlbuilder-1.1.jar
commons-configuration-1.7.jar	flink-dist_2.11-1.7.0.jar	jna-4.1.0.jar
deeplearning4j-core-0.9.1.jar	flink-gelly_2.11-1.7.0.jar	libtensorflow-1.12.0.jar
deeplearning4j-nlp-0.9.1.jar	flink-gelly-scala_2.11-1.7.0.jar	log4j-over-slf4j-1.7.21.jar
deeplearning4j-nn-0.9.1.jar	flink-ml_2.11-1.7.0.jar	logback-classic-1.2.3.jar
ejml-cdense-0.33.jar	flink-python_2.11-1.7.0.jar	logback-core-1.2.3.jar
ejml-core-0.33.jar	flink-queryable-state-runtime_2.11-1.7.0.jar	nd4j-api-0.9.1.jar
ejml-ddense-0.33.jar	flink-shaded-curator-1.7.0.jar	nd4j-native-0.9.1.jar
ejml-dsparse-0.33.jar	flink-shaded-hadoop2-uber-1.7.0.jar	nd4j-native-api-0.9.1.jar

Dependency		
ejml-experimental-0.33.jar	flink-table_2.11-1.7.0.jar	nd4j-native-platform-0.9.1.jar
ejml-fdense-0.33.jar	guava-26.0-jre.jar	okhttp-3.14.8.jar
ejml-simple-0.33.jar	hadoop-3.1.1-41-20201014.085840-4.jar	okio-1.14.0.jar
ejml-zdense-0.33.jar	httpasyncclient-4.1.2.jar	slf4j-api-1.7.21.jar
elsa-3.0.0-M7.jar	httpclient-4.5.12.jar	tensorflow-1.12.0.jar
log4j-api-2.16.0.jar	log4j-core-2.16.0.jar	log4j-api-2.8.2.jar
log4j-core-2.8.2.jar	-	-

13.4 Managing DLI Resource Quotas

What Is a Quota?

A quota limits the quantity of a resource available to users, thereby preventing spikes in the usage of the resource.

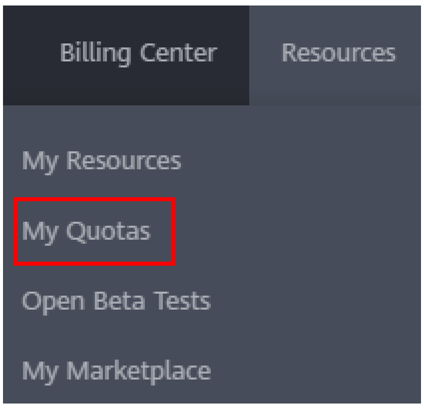
You can also request for an increased quota if your existing quota cannot meet your service requirements.

How Do I View My Quotas?

1. Log in to the management console.
2. Click in the upper left corner and select a region and a project.
3. In the upper right corner of the page, choose **Resources** > **My Quotas**.

   The **Service Quota** page is displayed.

   Figure 13-11 My quotas

4. View the used and total quota of each type of resources on the displayed page.

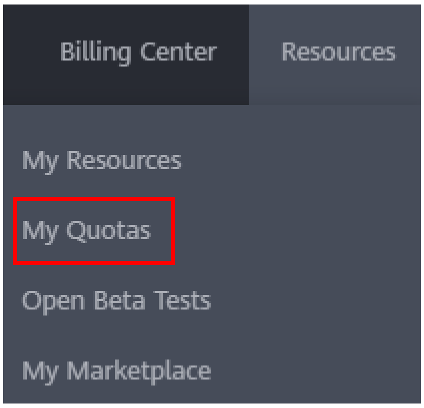
If a quota cannot meet service requirements, increase a quota.

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 13-12 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**.