Data Lake Compute Operation Guide

Copyright Notice

Trademark Notice

Tencent Cloud

Service Notice

This document provides an overview of the as-is details of Tencent Cloud's products and services in their entirety or part. The descriptions of certain products and services may be subject to adjustments from time to time.

The commercial contract concluded by you and Tencent Cloud will provide the specific types of Tencent Cloud products and services you purchase and the service standards. Unless otherwise agreed upon by both parties, Tencent Cloud does not make any explicit or implied commitments or warranties regarding the content of this document.

Contact Us

We are committed to providing personalized pre-sales consultation and technical after-sale support. Don't hesitate to contact us at 4009100100 or 95716 for any inquiries or concerns.

Contents

Operation Guide
Introduction to Console Operations
Data Development and Exploration
Data Exploration
SQL Editor

Last updated: 2026-05-20 15:13:11

The SQL editor provided by DLC supports data querying using unified SQL statements and is compatible with SparkSQL. You can complete data query tasks using standard SQL. For detailed syntax, see SQL Syntax.
You can go to the SQL editor via Data Exploration. Within the editor, you can perform simple data management, multi-session data queries, query history management, and download history management.

Data Management

Data Management supports adding new data sources and managing databases and data tables.

Creating a Data Catalog

Currently, DLC supports managing data catalogs for COS and EMR Hive. The operation steps are as follows.

1. Log in to the DLC console and select the service region. The login role requires administrator permissions.
2. Go to Data Exploration. Hover over the + icon above the database and table list, and click **Create Data Catalog** to enter the creation process.

   For detailed operation guidelines, see Query Other Data Sources.

Managing a Database

Through the SQL editor, you can create, delete, and view details of databases. For detailed operation guidelines, see Managing Databases via SQL.

Managing a Data Table

Through the SQL editor, you can create, query, and view details of data tables. For detailed operation guidelines, see Data Table Management.

Switching the Default Database

When using the SQL editor, you can specify a default database for query tasks. If no database is declared in the query statement after the default database is specified, the query will be executed under the default database.

1. Log in to the DLC console and select the service region.
2. Go to Data Exploration. Hover over the name of the database you want to specify, click the ••• icon, and then click **Set as Default Database** to designate that database as the default.
3. You can also switch directly in the default database selection box.

Data Query

Adding a Query Page

The SQL editor supports adding multiple pages for data queries. Each query page maintains independent configurations (default database, compute engine in use, query history, and so on), facilitating users in running and managing multiple tasks.

- You can create a new query page by clicking the + icon and switch between editor interfaces by clicking the tab bar.
- For your convenience in querying, you can save frequently used query pages by clicking the **Save** button. At the same time, you can quickly open your saved pages by clicking the ▼ icon.
- For saved query page information, you can click the **Refresh** button to update and synchronize the saved information, ensuring the accuracy of the query statements.
- The editor supports running multiple different SQL statements simultaneously. Clicking the **Run** button will execute all SQL statements in the editor and split them into multiple SQL tasks.
- To run some specific statements, select the statements to be executed and click **Run**.

Configuring Engine Parameters

After selecting a data engine, you can configure its parameters. To do this, go to Advanced Settings and click **Add**.

The following parameters are currently supported for configuration:

Engine	Configuration Name	Initial Value	Configuration Description
SparkSQL	spark.sql.files.maxRecordsPerFile	0	Maximum number of records written to a single file. A value of zero or a negative value indicates no limit.
	spark.sql.autoBroadcastJoinThreshold	10MB	Maximum byte size of a table to be broadcast to all worker nodes when execution connections are configured. By setting this value to -1, the broadcast feature can be disabled.
	spark.sql.shuffle.partitions	200	Default number of partitions.
	spark.sql.sources.partitionOverwriteMode	static	If the value is static and no partition is specified, all matching partitions will be deleted before an overwrite operation is performed. Example: If a partitioned table has a partition 2022-01, when an INSERT OVERWRITE statement is used to write data for the partition 2022-02 without specifying the partition, the data in the partition 2022-01 will also be overwritten. If the partition 2022-02 is specified for writing, then the data in the partition 2022-01 will not be overwritten. When the value is dynamic, partitions will not be deleted in advance. Instead, partitions with data being written will be overwritten at runtime.
	spark.sql.files.maxPartitionBytes	128MB	Maximum number of bytes to be packed into a single partition when files are being read.

Presto	use_mark_distinct	true	Determines whether the engine redistributes data when executing the distinct function. If the distinct function is called multiple times in a query, it is recommended to set this parameter to false.
	USEHIVEFUNCTION	true	Determines whether to use the Hive function when a query is executed. Set the parameter to false to use native Presto functions.
	query_max_execution_time	–	Used to specify the query timeout. Queries will be terminated if the execution time exceeds the set duration. Supported units: d (days), h (hours), m (minutes), s (seconds), and ms (milliseconds) (for example, 1d represents 1 day, and 3m represents 3 minutes).
	dlc.query.execution.mode	async	Engine query execution mode, which defaults to async. In async mode, the task completes full query computation, saves the results to COS, and returns them to the user, enabling users to download query results after the query is completed. Users can also set the value to sync. In sync mode, queries may not perform full computation. As soon as partial results are available, they will be directly returned by the engine to users without being saved to COS. As a result, this mode delivers lower query latency and reduced execution time to users. However, results are retained in the system for only 30 seconds. This mode is recommended for scenarios where downloading full results from COS is unnecessary but lower query latency and reduced execution time are desired, such as during query exploration phases and BI result presentations.

Note:

> The Presto engine is deprecated and only available for existing users.

Presto Execution Modes

When the Presto engine is selected, Data Exploration supports two execution modes: quick query and complete mode.

- Quick query: Executes queries faster, with the results not being persisted. It applies to the exploration phase.
- Complete mode: Executes a full query and saves the data to COS.

Querying Results

You can directly view query results through the SQL editor and click the icon to expand or collapse the display height of query results.

You can configure the query result storage directory using the configuration button in the upper-right corner. It supports configuration to a COS path or built-in storage.

- The console returns a maximum of 1000 results per task. To obtain more results, use the API. For API operations, see API Documentation.
- Query results can be downloaded locally when the COS storage path is not specified. For details, see Obtaining Task Results.

Querying Statistical Data

For results queried by Presto and Spark SQL engines, the optimization and quantification of different features can be displayed.
Spark SQL engine supports viewing:

1. Data scan volume
2. Cache acceleration
3. Adaptive shuffle

Presto engine supports viewing:

1. Data scan volume
2. Cache acceleration

Click the **Statistical Data** tab to view the statistical data and optimization recommendations for the query results.

Historical Execution Query

Each query page can save the execution history within the last 3 months and supports viewing query results from the last 24 hours. You can quickly search for information on previously executed tasks through the execution history. For detailed operations, see Task History.

Download History Management

The download task for each query result can be viewed in **Download History**, where you can query the download task status and related parameter information.

Data Query Task
SELECT Tasks

Last updated: 2026-05-20 15:13:37

Users can query, analyze, and compute data in created databases and tables using SQL statements. For the SQL query statements and functions supported by DLC, refer to SQL Syntax.

Executing SELECT Queries

1. Select the default database and compute resources.
   - If a user selects a default database, operations that do not specify a database in SQL statements will be executed under the default database.
   - You can select a shared cluster or a dedicated cluster for compute resources.
2. Run SQL tasks. After standard SQL statements are entered, click **Run** to start running the SQL tasks.
   - DLC adopts a Serverless architecture, where computing resources are temporarily scheduled. For the first time over a period of time, the result return time for running DML tasks may be slightly slower than that for normal tasks.

   > **Note:**
   > When you submit an SQL task through the console, a single SQL statement supports a maximum of 450,000 characters. If the number of characters exceeds this limit, you can execute the task via the API.

3. After a task is executed, the query results are displayed in the console.

   If a user exits the console page, they cannot view the query results of historical tasks in the console. They need to view the task result files in the historical run or the user-configured query result COS bucket.

Canceling a Running Query

During task execution, you can click the **Run** button. The button text changes to **Stop**, and the current query is canceled. After the query is canceled, DLC does not return query results, but it does count the data scan volume that has already been executed. If you use public_engine as the compute resource, the data scan volume will incur certain charges. For detailed pricing, refer to Billing Overview.

Querying Partitioned Tables

Last updated: 2026-05-20 16:32:51

Storing data in partitioned directories can reduce the data scan volume for DLC computing tasks and significantly improve computational performance. The most common approach to partitioning data is to store it in different directories based on time. For example, data generated on the same day can be placed in one directory, and multi-level data directories can be organized in a "year-month-day" hierarchy. The same table and its partitions in DLC must use the same data format.

Creating a Partitioned Table

To create a partitioned table, you must specify the partition fields in the CREATE TABLE statement. For details, see Table Management.

Adding Partition Data

Specifying partitions when creating a table only configures the partition fields; you cannot immediately execute queries to obtain data. You must add the partitioned data to the table. If new partitioned data is added to the data directory, you need to add the partition information to the table.

Manually Adding a Partition

Using the `ALTER TABLE ADD PARTITION` statement adds the specified partition directory to the table. If the partition directory is compatible with Hive's partitioning rules (**partition column name = partition column value**), you do not need to specify the data path. Otherwise, you must explicitly specify the data path. For details, see SQL Syntax.

> **Example Note:**
> "table_demo" is a created data table. In this example, it is partitioned by time.

- Example 1: A single partition directory that is compliant.

```
ALTER TABLE tabel_demo ADD
PARTITION (dt = '2021-01-01');
```

- Example 2: Nested multi-level partition directories

```
ALTER TABLE tabel_demo ADD
```

```
PARTITION (year = '2021', month='01', day='01');
```

- Example 3: Explicitly specifying the partition path

```
ALTER TABLE tabel_demo ADD
PARTITION (year = '2021', month='01', day='01') LOCATION
'cosn://tablea_demo' ;
```

Automatically Adding a Partition

Use the `MSCK REPAIR TABLE` statement to scan the data directory specified when the table is created. If new partition directories exist, the system automatically adds these partitions to the table's metadata. For details, see SQL Syntax. The example is as follows:

```
MSCK REPAIR TABLE table_demo
```

System Constraints

- The `MSCK REPAIR TABLE` statement only adds partitions to the table's metadata; it does not delete them. To delete partitions that have been added, execute the `ALTER TABLE table-name DROP PARTITION` statement. For details, see SQL Syntax.
- If the data volume is very large, MSCK REPAIR TABLE is not a recommended solution. The system scans all the data, which consumes a significant amount of time. This may cause the task to time out, leaving the table's partition information in an incomplete state.
- Partition directories must be compatible with Hive's partitioning rules: **partition column name = partition column value**. If they are not, use `ALTER TABLE ADD PARTITION` to load the partition. For details, see SQL Syntax.
- Ensure that data for separate tables is kept in separate folder hierarchies. For example, suppose table A's data is at `cosn://tablea_a` and table B's data is at `cosn://tablea_a/tablea_b` in the COS service. If both tables are partitioned by string, `MSCK REPAIR TABLE` will add table B's partitions to table A. To avoid this, use separate folder structures, such as `cosn://tablea_a` and `cosn://tablea_b`.
- Executing this statement may incur data read/write costs for the COS service. For details, see the COS Service Pricing.

Querying JSON Data

Last updated: 2026-05-20 15:14:32

Query Steps

1. Create a data table and specify the JSON parsing format.

```
CREATE EXTERNAL TABLE `order_demo`(
  `docid` string COMMENT 'from deserializer',
  `user` struct < id :int,
  username :string,
  name :string,
  shippingaddress :struct < address1 :string,
  address2 :string,
  city :string,
  state :string > > COMMENT 'from deserializer',
  `children` array < string >
) ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe' LOCATION
'cosn://dlc-bucket/order'
```

2. Execute query statements to query JSON data. DLC supports JSON parsing functions such as `json_parse()`, `json_extract_scalar()`, and `json_extract()`. For details, see SQL Syntax.

```
SELECT `user`.`shippingaddress`.`address1` FROM `order_demo` limit 10;
```

System Constraints

- The content must be in a complete JSON format; otherwise, DLC cannot parse it properly.
- A single row of data must not contain line breaks, and JSON must not be formatted for visual optimization. For example:

```
{"name":"Michael"}
{"name":"Andy", "age":30}
{"name":"Justin", "age":19}
```

- DLC automatically identifies the first level of JSON as attribute columns of the data table and the remaining nested structures as corresponding attribute values.

Querying External Data Sources Through the Standard Spark Engine

Last updated: 2026-05-20 15:14:54

DLC supports querying and analyzing unmanaged data. Currently supported data sources include: MySQL, EMR Hive(COS), EMR Hive(HDFS), EMR Hive(CHDFS), PostgreSQL, SQL Server, ClickHouse, and TCHouse-D. Users can add and manage other data sources through the DLC console. This document uses querying TCHouse-D as an example.

Prerequisites

You have purchased the standard Spark engine from Data Lake Compute (DLC) .
You have created a Tencent Cloud TCHouse-D instance.

Creating a Network Connection

1. Log in to the DLC console and select the region where the service is located.
2. Click Network Connection Configuration in the left sidebar to create a new network connection. Select **Cross-Source** as the network configuration type, select **New Network Configuration** as the instance source, select Tencent Cloud TCHouse-D as the data source type, select the Tencent Cloud TCHouse-D instance you want to query, select **Standard Spark Engine** in the data engine binding area, and save the configuration. If the connection status displays as normal, the creation succeeds.

Adding a Data Catalog

1. Log in to the DLC console and select the region where the service is located. The logged-in user must have the permission to create data catalogs.

Solution 1: Go to the Data Explore page, hover over "+" next to the database and table, and click **Create data catalog**.
Solution 2: Go to the Metadata Management page and click **Create Data Catalog**.

2. After selecting Tencent Cloud TCHouse-D as the data source type, select the corresponding instance to associate with.
3. Enter verification information of the data source and click **Next and then Confirm**. When the creation status of the data catalog displays as successful, you can query and perform other operations in data exploration.

Data Management

Currently, DLC supports the **database information viewing** and **data table preview** features for unmanaged data.

Viewing Database Information

Solution 1: View on the Data Explore Page

1. Log in to the DLC console and select the region where the service is located. The logged-in user must have the permission to view data tables.
2. Go to the Data Exploration page, select the data catalog to be queried, move the pointer over a database name to display a menu, select **Basic Info**, and view the basic database information in the displayed pop-up window.

Solution 2: View on the Data Management Page

1. Log in to the DLC console and select the region where the service is located. The logged-in user must have the permission to view data tables.
2. Go to the Data Explore page. You can go to the details page to view database and table information based on the data catalog name.

Previewing Data Table Data

1. Log in to the DLC console and select the region where the service is located. The logged-in user must have the permission to view data tables.
2. Go to the Data Exploration page, move the pointer over a data table name to display a menu, click **Preview Data**, and execute an SQL statement to query and display the data from the data table.

> ⚠ **Note:**
> You need to select the data engine bound to the VPC network configuration of the data source to query.

Using Views

Last updated: 2026-05-20 15:15:12

A view in DLC is a logical table, not a physical table. Each time a view is referenced in a query, the query that defines the view runs. You can create a view from a `SELECT` query and then reference that view in future queries. For the view operation statements supported by DLC, refer to SQL Syntax.

System Constraints

- A view name is case-insensitive, can only contain English letters and underscores (_), and has a maximum length of 128 characters.
- DLC does not support using views to manage data access permissions.

INSERT INTO

Last updated: 2026-05-20 15:15:37

The INSERT INTO statement can insert the results of a SELECT query run on a source table as new rows into a target table. The INSERT INTO syntax supported by DLC can be found in SQL Syntax.

Querying Script Parameters

Last updated: 2026-05-20 15:15:53

To facilitate querying with script parameters, DLC supports configuring date parameters for queries.
The DLC standard date format is: yyyymmddhh24miss. You can use the ${} command to set the date as a variable. The variable consists of two parts, date and time, and the result is the sum of these two parts.

- The date part can be any date format or a system-predefined variable, such as yyyymmdd, yyyymm, yyyy-mm-dd, yy, or dataDate.
- The date variation part allows adjustments of +/-N cycles and supports formats such as N/Nd, Nm, Nw, Nh, and Nmi. It is also compatible with calculation formulas, for example, 7*N and N/24.

Obtain addition/subtraction cycles.	Methodology	Compatible formats	Example
Next N years	${yyyymmdd+Ny}	-	-
Previous N years	${yyyymmdd-Ny}	-	Previous year: ${yyyymmdd-12m}: 20190920
Next N months	-	${yyyymmdd+Nm}	-
Previous N months	${yyyymmdd-Nm}	$[add_months(yyyymmdd,-N)]	${yyyymmdd-1m}:20200820 ${yyyymm}:202009 ${dataDate-1m}:20200820
Next N weeks	${yyyymmdd+Nw}	${yyyymmdd+7*N}	-
Previous N weeks	${yyyymmdd-Nw}	${yyyymmdd-7*N}	-
Next N days	${yyyymmdd+N/Nd}	-	-

Previous N days	${yyyymmdd-N/Nd}	-	${yyyymmdd-1},${dataDate-1}
Next N hours	${yyyymmddhh24+Nh}	$[yyyymmddhh24+N/24]	-
Previous N hours	${yyyymmddhh24-Nh}	$[yyyymmddhh24-N/24]	${yyyymmddhh24-1h}:2020092014 ${dataDate-1h}: 2020092014
Next N minutes	${yyyymmddhh24mi+Nmi}	$[yyyymmddhh24+N/24/60]	-
Previous N minutes	${yyyymmddhh24mi-Nmi}	$[yyyymmddhh24-N/24/60]	${yyyymmddhh24mi-10mi},${dataDate-10mi}

Note:
When using this feature, ensure that the format of the variable or the part before the +/- sign in the variable conforms to the standard date format. Otherwise, the system will not be able to recognize and use it.

Obtaining Task Results

Last updated: 2026-05-20 15:16:09

Using the Query Editor to Obtain Task Results

Log in to the DLC Console > Data Exploration page. When you query a task, the query results are displayed in real time below the editor. Click **Run History** to view the task list.

- A single SQL task displays a maximum of 1000 data results in the console. SQL tasks submitted using the API or JDBC are not subject to this limit.
- You can query the execution history of the last 3 months for a single Session through the run history. For more ways to query historical records, see Task History.

Configuring the Output Format for Task Results

The results of data exploration are saved in csv format by calling Spark's DataFrame.write method. If your engine version is after April 2023, you can make the exploration results configurable.

1. Configure the format of the results output to csv. The following parameters are supported.

Parameter	Default Value	Remarks
livy.sql.result.format.option.sep livy.sql.result.format.option.delimiter	,	The delimiter between columns in csv storage, defaulting to an English comma.
livy.sql.result.format.option.encoding livy.sql.result.format.option.charset	UTF-8	String encoding format. For example: UTF-8, US-ASCII, ISO-8859-1, UTF-16BE, UTF-16LE, UTF-16.
livy.sql.result.format.option.quote	\"	Whether the quote is a single quote or double quote, and note to use escape characters.
livy.sql.result.format.option.escape	\\	Escape character, and note to use escape sequences.
livy.sql.result.format.option.charToEscapeQuoteEscaping		Characters that need to be escaped inside quotes.
livy.sql.result.format.option.comment	\u0000	Remarks
livy.sql.result.format.option.header	false	Header present or not.
livy.sql.result.format.option.inferSchema	false	Infer the type of each column; if not inferred, each column is a string.
livy.sql.result.format.option.ignoreLeadingWhiteSpace	true	Ignoring leading empty strings.
livy.sql.result.format.option.ignoreTrailingWhiteSpace	true	Ignoring trailing empty strings.
livy.sql.result.format.option.columnNameOfCorruptRecord	_corrupt_record	The column name for columns that cannot be converted. This parameter is affected by spark.sql.columnNameOfCorruptRecord, with table configuration taking precedence.
livy.sql.result.format.option.nullValue		The storage format for null, which defaults to an empty string, is written according to the emptyValue method.
livy.sql.result.format.option.nanValue	NaN	The storage format for non-numeric values.
livy.sql.result.format.option.positiveInf	Inf	The storage format for positive infinity.
livy.sql.result.format.option.negativeInf	-Inf	The storage format for negative infinity.
livy.sql.result.format.option.compression or codec		The class name of the compression algorithm. By default, no compression is applied. Abbreviations such as bzip2, deflate, gzip, lz4, and snappy can be used.
livy.sql.result.format.option.timeZone	system default time zone	The default time zone. The value of this parameter is affected by spark.sql.session.timeZone, for example, Asia/Shanghai, with table configuration taking precedence.
livy.sql.result.format.option.locale	en-US	Locale
livy.sql.result.format.option.dateFormat	yyyy-MM-dd	The default date format.
livy.sql.result.format.option.timestampFormat	yyyy-MM-dd'T'HH:mm:ss.SSSXXX	The default timestamp format, which is yyyy-MM-dd'T'HH:mm:ss[.SSS][XXX] in non-LEGACY mode.
livy.sql.result.format.option.livy.sql.result.format.option.multiLine	false	Allow multiple lines
livy.sql.result.format.option.maxColumns	20480	Maximum number of columns
livy.sql.result.format.option.maxCharsPerColumn	-1	Maximum number of characters per column. A value of -1 indicates no limit.
livy.sql.result.format.option.escapeQuotes	true	Escape quotes
livy.sql.result.format.option.quoteAll	quoteAll	Enclosing the entire content in quotes during writing.
livy.sql.result.format.option.emptyValue	\"\"	The read/write format for null values.
livy.sql.result.format.option.lineSep		Line break

2. Configure the output format to a non-csv format. Note that in this case, the console cannot display the results. However, you can read the result path through other means. For the specific location where the result path is saved, refer to the next section.

The configuration method is livy.sql.result.format, which supports saving results in text, orc, json, or parquet format.

Configuring the Save Location for Task Results

> **Note:** Standard Engine – Presto does not currently support this. Full results can be obtained via JDBC.

DLC allows users to automatically save query results to a COS path or DLC managed storage through configuration. The configuration method is as follows:

1. Log in to the DLC console and select the service region. The account used for login must have COS-related permissions.
2. Go to the Data Exploration page, click **Storage Configuration** in the upper-right corner, and configure the query result saving method.

3. Results can be saved to DLC managed storage or COS. If you need to configure a COS path, the operating account must have the relevant permissions on the COS side. Data storage fees are subject to COS pricing.

   Task results are stored in the subfolders of the following COS paths.

```
The data path for task results: COS directory
path/DLCQueryResults/yyyy/mm/dd/[QueryID]/data/XXXX.csv
The metadata path for task results: COS directory
path/DLCQueryResults/yyyy/mm/dd/[QueryID]/meta/result.meta.json
```

   - COS directory path: the COS directory path configured in the system configuration.
   - /yyyy/mm/dd: the directory is organized according to the task execution date.
   - /data: the directory where query result data is stored, with the file format being csv. DLC may generate multiple data files.
   - /meta: the new directory for storing the metadata of queried data tables, with the file format being json.

> **Note:**
> - SELECT query results are stored in DLC internal storage, which uses COS as the underlying storage. The results are retained for 36 hours.
> - Store SELECT query results in your bucket path on COS. Note: Ensure that the relevant COS permissions have been granted.

Downloading Task Results

> **Note:** Standard Engine – Presto does not currently support this. Full results can be obtained via JDBC.

DLC allows users to manually download query results to their local machine. When the full mode is not enabled, users can click the download result button for tasks with query results to save the results locally or manually save them to COS (COS permissions are required).

- The data downloaded and saved to COS corresponds to the query results of this SQL task, which are limited to a maximum of 500 records.
- The data downloaded to the local machine must not exceed 50 MB.
- If you configure to save results to COS, the results are automatically saved to the COS path, eliminating the need for manual download.

Query Script Analysis

Last updated: 2026-05-20 15:16:27

To help users quickly handle repetitive query tasks, DLC provides script file analysis.

> **Note:**
> The console allows you to save up to 100 SQL scripts.

Creating a Query Directory

1. Log in to the DLC console > Script Query page.
2. On the query page, select **Query**, and click + on the right to add a query catalog.
3. After the directory configuration is filled in, click **OK** to complete the creation.

> **Note:**
> 1. Directory name: It can contain Chinese characters, English letters (both uppercase and lowercase), digits, and special characters including underscore (_), hyphen (-), space ( ), and colon (:), with a maximum length of 50 characters.
> 2. Permission Settings: You can set visibility permissions for this script directory and the scripts within it, based on either the working group or user perspective.

Creating a Query Script

1. Log in to the DLC console > Script Query page. You can then execute and save scripts by clicking the library ••• icon or by creating new ones directly.
2. After selecting the **Compute Engine**, click Run to execute the script.

Saving a Query Script

1. After the query is completed, click **Save**.
2. Queries created through a library are saved in that library's directory. Queries added via the tab bar can be saved directly in the root directory or in a library for which you have permissions.
3. Go to the **Save Query** page. In the **Permission Settings** section, you can customize the selection based on the library's **Public Scope**. It also supports specifying the **Usage Permissions** for tables within the public scope.

Viewing Script Information

1. Hover the mouse over the script name you want to view to see its details.
2. Click the ••• icon next to the table you want to view to select and open the query.

Deleting a Query Script

Click the ••• icon next to the table you want to delete to select and delete the script.

> ⚠ **Note:**
> Once a script is deleted, it cannot be recovered. Please proceed with caution.

Data Job
Data Job Overview

Last updated: 2026-05-20 15:16:49

DLC provides batch processing and stream computing capabilities based on native Spark, enabling users to perform complex data processing, ETL, and other operations through data tasks.
The supported Spark-related versions for data jobs are as follows:

- Scala 2.12.* version.
- Spark 3.2.1 version.

Preparations

Before using data jobs, you must create a data access policy to ensure your data security. This policy specifies the COS paths and files that your data jobs can access. For detailed configuration instructions, see Configure Data Access Policy.
If a data job needs to access other data sources, you must configure the network for the data engine. After the configuration is completed, you can select the corresponding data engine to access and process data. For the network configuration method and detailed instructions, see Engine Network Configuration.

Billing Mode

Data jobs are billed based on the data engine used. Currently, pay-as-you-go and yearly/monthly subscription billing modes are supported. For details, see Data Engine Description.

- Pay-as-you-go: It is applicable to scenarios with a small number of data jobs or periodic usage. A data job is started after creation and automatically suspended after successful execution, after which no fees will be incurred.
- Monthly subscription: It is applicable to scenarios where a large number of data jobs are regularly executed. Resources are reserved in this mode, so you do not need to wait for data engine start.

> ⚠ **Note:**
> As a data job differs from a SQL job in terms of the compute engine type, you need to purchase a separate data engine for Spark jobs; otherwise, you cannot run data jobs on a SparkSQL data engine.

Job Management

Through the **Data Jobs** management menu, you can create, start, modify, and delete data jobs.

1. Log in to DLC Console > Data Jobs and go to the Data Job Management page.
2. Click the **Create Data Job** button to create a new data job. For detailed steps, see Create Data Job.
3. You can view the status of the current data job tasks in the list and manage data jobs. For detailed steps, see Manage Data Jobs.

Creating a Data Job

Last updated: 2026-05-20 15:17:06

Preliminary Preparation

Before you start creating a data job, you must configure the data access policy to ensure that the data job can securely access your data. For configuration steps, see Configuring Data Access Policy.

Steps

1. Log in to the DLC console, click **Data Job** in the left-side menu to go to the Data Job Management page.
2. Click the **Create Job** button to go to the creation page.

Selecting a Data Access Policy

Go to the DLC console > Data Job page. Click the Create Job button to go to the creation page. The data access policy option will be automatically filled with the resident access policy. You can select another access policy from the dropdown option. If you need to add or switch the resident access policy, see Configuring Data Access Policy > Step 4.
Configuration parameters are as follows:

Parameter	Description
Job Name	It supports Chinese, English, numbers, and "_", with a maximum of 100 characters.
Job Type	Batch processing: Spark JAR-based batch data job Stream processing: Spark Streaming-based Streaming data job SQL job: Packaged as a Jar file in the background, which invokes the spark.sql operator to execute SQL statements.
Package	The jar format is supported. You can select a local file of up to 5 MB in size or a file in Cloud Object Storage (COS). If the local file exceeds 5 MB, upload it to cos for use. You can directly enter the cos storage path.
Main Class	It is mandatory when selecting a JAR file. JAR package parameter in the main class. Separate multiple parameters by space.

Program Entry Parameter	Optional. Entry parameter of the program. Multiple entries are supported. Separate multiple parameters by space.
Job Parameter	Optional. –config information of the job, which starts with spark. in the format of k=v. Separate multiple parameters by line break. Example: spark.network.timeout=120s
Spark Image	Required.
Dependency JAR Resource (––jar)	Optional. The JAR format is supported. You can select multiple files. You can select a local file of up to 5 MB in size or a file in Cloud Object Storage (COS). If the local file exceeds 5 MB, upload it to COS for use. You can enter multiple COS paths and separate them by semicolon.
Dependency PY Resource (––py–files)	Optional. The PY, ZIP, and EGG formats are supported. You can select multiple files. You can select a local file of up to 5 MB in size or a file in Cloud Object Storage (COS). If the local file exceeds 5 MB, upload it to COS for use. You can enter multiple COS paths and separate them by semicolon.
Dependency File Resource (––files)	Optional. The JAR and ZIP formats are not supported. You can select multiple files. You can select a local file of up to 5 MB in size or a file in Cloud Object Storage (COS). If the local file exceeds 5 MB, upload it to COS for use. You can enter multiple COS paths and separate them by semicolon.
Dependency Archives Resource (––archives)	Optional. TAR.GZ, TGZ, and TAR formats are supported. You can select multiple files. You can select a local file of up to 5 MB in size or a file in Cloud Object Storage (COS). If the local file exceeds 5 MB, upload it to COS for use. You can enter multiple COS paths and separate them by semicolon.
CAM Role arn	The data access policy configured in the job configurations specifies the data scope accessible to the data job. For detailed configuration instructions, see Configuring Data Access Policy.

Resource Configuration	The engine resources that can be configured with the data job, the number of which cannot exceed the specifications of the selected data engine. Resource description: 1 CU ≈ 1-core 4 GB Billable CUs = Executor resource x Executor quantity + Driver resource Pay-as-you-go data engines are billed by the computing CUs used.

After the configuration is completed, click **Save** to finish the creation.

Managing Data Jobs

Last updated: 2026-05-20 15:17:20

To help you better manage data jobs, we provide the following operations for data job management.

- Edit a data job.
- Start & Stop a Data Job Task
- View Data Job & Task Details
- Delete a data job.

Editing a Data Job

> **Note:**
> Data jobs cannot be edited while they are running.

The type of a data job cannot be edited. To change the type, create a new data job. For detailed steps, see Create a Data Job.

1. Log in to the DLC console, select a service region, and go to the Data Job page in the left-side menu bar.
2. Locate the data job you want to edit by its **Job Name/ID**. Then, click the **Edit** button under the **Operations** column on the far right to go to the job editing page.
3. After the content that needs to be edited is modified, save it to complete the editing.

Starting & Stopping Data Job Tasks

Using a created data job, you can start & stop it to execute the job and generate corresponding tasks. A single data job can generate multiple task instances or be executed multiple times.
Data task statuses are as follows:

Status	Description
Not Started	Initial State After Creation
Running	While a data task is running, its associated data job cannot be edited or deleted.
Successful	Task Completed Successfully

Failed	Task Failed. You can query error information through logs or Spark UI.
Canceled	Manually Canceled by User

To start & stop a data job task, follow these steps:

1. Log in to the DLC console, select a service region, and go to the Data Job page in the left-side menu bar.
2. Locate the data job whose status needs to be changed, and click the **Run** button in the Actions column to change its task status.

> ⚠ **Note:**
> Starting a task instance consumes compute engine resources. If the resources exceed the configured maximum of the compute engine, the task will be queued.

Viewing Data Job & Task Details

1. Log in to the DLC console, select a service region, and go to the Data Job page in the left-side menu bar.
2. Click the **Job Name/ID** of the job you want to view to go to the Data Job Details page.
3. In the **Spark Job Details** page, select **Historical Tasks** to view the basic information and task list of the data job. The task list contains the data task information corresponding to that data job and supports viewing task details and the Spark UI.
4. Click **View Details** or **Task ID** to go to the Task Details page. The Task Details page contains the task's **Basic Information**, **Run Results**, **Task Insights**, and **Task Logs**. Currently, only the most recent 1000 entries of task logs are supported for viewing.
5. Select **Task Logs**, click the **Export Logs** button to export them, then perform a full download after the export. In the pop-up **Export Records** window, click **Save to Local** under the **Actions** column.

> ⚠ **Note:**
> Download records are retained for 3 days. After this period, they cannot be saved locally, and you must create a new download task.

Deleting a Data Job

> ⓘ **Note**
> A data job cannot be deleted if it has running data tasks.

1. Log in to the DLC console, select a service region, and go to the Data Job page in the left-side menu bar.
2. Locate the data job you want to delete, click the **More** button, select **Delete Job** from the expanded menu bar, and confirm the deletion to complete the operation.

> ⚠ **Note:**
> Deleting a data job will also delete its corresponding data task information. Proceed with caution.

PySpark Dependency Management

Last updated: 2026-05-20 15:17:37

Currently, the PySpark base runtime environment in DLC is configured to use Python=3.9.2. Python dependencies for Spark jobs can be managed in the following two ways:

1. Specify dependent modules and files using `--py-files`.
2. Specify a virtual environment using `--archives`.

If your modules or files are implemented in pure Python, it is recommended to specify them using `--py-files`.
Using `--archives` allows you to directly package and use the entire development and testing environment. This method supports compiling and installing C-related dependencies and is recommended when the dependency environment is complex.

> **Note:**
> The two methods described above can be used simultaneously based on your requirements.

Using the `--py-files` Dependency Package

This method applies to modules or files implemented in pure Python that do not contain any C dependencies.

Step 1: Packaging Modules/Files

For external packages from PyPI, you need to install and package the commonly used dependencies in your local environment using the pip command. These dependency packages must be implemented in pure Python and must not rely on C-related libraries.

```
pip install -i https://mirrors.tencent.com/pypi/simple/ <packages...>  -
t dep

cd dep

zip -r ../dep.zip .
```

Single-file modules (functions.py) and custom Python modules can be packaged using the methods described above. Note that custom Python modules must be standardized according to Python official requirements. For details, refer to the Python official documentation Python Packaging User Guide.

Step 2: Importing the Packaged Module

Log in to the Data Lake DLC Console > Data Jobs page and click **Create Job**. In the `--py-files` parameter, import the packaged dep.zip file. This file can be imported by uploading it to COS or from your local machine.

Using a Virtual Environment

Virtual environments can resolve dependency issues for some Python packages that rely on C. Users can compile and install the required packages into a virtual environment as needed, and then upload the entire virtual environment.
Since C-related dependencies involve compilation and installation, it is recommended to use an x86-architecture machine, a Debian 11 (bullseye) system, and a Python = 3.9.2 environment for packaging.

Step 1: Packaging the Virtual Environment

There are two methods for packaging a virtual environment: using Venv and using Conda.

1. Package using Venv.

```
python3 -m venv pyvenv

source pyvenv/bin/activate

(pyvenv)> pip3 install -i [https://mirrors.tencent.com/pypi/simple/]
(https://mirrors.tencent.com/pypi/simple/) packages

(pyvenv)> deactivate

tar czvf pyvenv.tar.gz pyvenv/
```

2. Package using Conda.

```
conda create -y -n pyspark_env conda-pack <packages...> python=<3.9.x>
conda activate pyspark_env
conda pack -f -o pyspark_env.tar.gz
```

After packaging is complete, upload the packaged virtual environment file `pyvenv.tar.gz` to cos.

> ⚠ **Note:**

> Package using the tar command.

3. Use the packaging script.

To use the packaging script, you need to install a docker environment. Currently, Linux / macOS environments are supported.

```
bash pyspark_env_builder.sh -h
Usage:

  pyspark-env-builder.sh [-r] [-n] [-o] [-h]
  -r ARG, the requirements for python dependency.
  -n ARG, the name for the virtual environment.
  -o ARG, the output directory. [default:current directory]
  -h, print the help info.
```

Parameter	Description
-r	Specify the location of requirements.txt.
-n	Specify the virtual environment name. The default is py3env.
-o	Specify the local directory for saving the virtual environment. The default is the current directory.
-h	Print help information.

```
# requirement.txt
requests

# Run the following command.
bash pyspark_env_builder.sh -r requirement.txt -n py3env
```

After the script finishes running, you can obtain py3env.tar.gz from the current directory. Then, upload this file to cos.

Step 2: Specifying the Virtual Environment

Log in to the Data Lake DLC console, go to the Data Job page, click **Create Job**, and refer to the following screenshot for operation.

1. In the `--archives` parameter, enter the full path of the virtual environment. The text after the # sign is the name of the decompressed folder.

   > **Note:**
   > The "#" sign is used to specify the decompression directory. The decompression directory affects the configuration of subsequent runtime environment parameters.

2. Specify runtime environment parameters in the `--config` parameter.

   To use the Venv packaging method, configure: `spark.pyspark.python = venv/pyspark_venv/bin/python3`

   To use the Conda packaging method, configure: `spark.pyspark.python = venv/bin/python3`

   To use the script packaging method, configure: `spark.pyspark.python = venv/bin/python3`

   > **Note:**
   > Because venv and conda use different packaging methods, their directory structures differ. Specifically, you can decompress the .tar.gz file to view the relative path of the python file.

Resource Management
Engine Management
Introduction to Data Engines

Last updated: 2026-07-14 15:37:47

The DLC data engine is the foundation for DLC's data analysis and computing services. All computations performed in DLC require the use of a data engine. You can select the corresponding engine type based on your application scenario.

Engine Type

DLC provides two types of data engines for you to choose from: the **Standard Engine** and the **SuperSQL Engine**. The core difference between the two engine types lies in the SQL syntax they support. The Standard Engine uses the native Spark and Presto syntax from the community. The SuperSQL Engine supports a unified syntax developed by DLC. This means the same set of SuperSQL syntax can be executed on both Spark and Presto engines, eliminating syntax differences between engines. This significantly reduces usage costs in business scenarios that require the combined use of different analytical engines. The main features and selection recommendations for the two engine types are as follows:

Engine Type	Optional Type	Key Features	Use Constraint	Selection Recommendation
Standard engine	• Spark • Presto	• Native syntax: the native syntax from the Spark/Presto communities, which reduces learning and migration costs. • Flexible to use: It supports Hive JDBC and Presto JDBC. • Integrated Spark: Its standard Spark engine can	Currently provides a free gateway with 2cu specification. To upgrade the specification, upgrade the gateway.	1. Use Spark/Presto native syntax. 2. Purchase a Spark engine to complete batch jobs and offline SQL tasks. 3. Use Hive JDBC and Presto JDBC.

		execute SQL and Spark batch tasks.		
SuperSQL engine	• SparkSQL • Spark jobs Presto	• Unified syntax: The same set of syntax applies to both Spark and Presto engines. • Supports federated queries.	Learn the unified SuperSQL syntax. For SQL/batch task scenarios, purchase the corresponding engine type.	1. Use a unified syntax for Spark + Presto. 2. Use federated queries.

For more detailed information, see the following comparison table or view the Standard Engine and SuperSQL Engine descriptions.

> **Note:**
> The SuperSQL-type engine and the Presto engine have been deprecated and are only available for existing users.

Detailed Comparison Between Standard Engine and SuperSQL Engine

Function	Standard Engine	SuperSQL Engine	Description
Presto	✓	✓	Both engines support the Presto engine.
Spark	✓	✓	The SuperSQL engine is divided into SparkSQL and Spark job types, where the SparkSQL engine supports SQL jobs, and the Spark job engine supports Spark batch/streaming jobs and SQL jobs; the standard engine is an integrated Spark engine.
SQL Syntax	Native syntax	Unified syntax	The standard engine supports the native syntax of Spark and Presto. The SuperSQL engine supports the unified syntax developed by DLC.
Gateway	✓	-	DLC provides a more stable, secure, and higher-performance task submission experience through its

			self-developed Serverless gateway service based on Apache Kyuubi.
Resource group	✓	–	Resource groups are a queue feature specific to the standard Spark engine. Through resource groups, standard engine resources can be allocated on demand, and SQL tasks can be submitted to designated resource groups for execution.
Shared Engine		✓	The SuperSQL engine supports the shared mode, which is suitable for scenarios with low analysis frequency and small data volumes.
Hive JDBC	✓	–	The standard engine supports submitting tasks using Hive JDBC.
Presto JDBC	✓	–	The standard engine supports submitting tasks using Presto JDBC.
DLC JDBC	✓	✓	Both engines support submitting tasks using DLC JDBC.
Submitting tasks via Tencent Cloud API	✓	✓	Both engines support submitting tasks using TencentCloud API or on the Data Exploration page in the console.
Federated query	–	✓	The SuperSQL engine provides data federation query and analysis capabilities. For the operation guide on adding a federation query data catalog, refer to Data Catalog and DMC. The standard engine does not currently support federation queries.

If you have further questions about choosing the Standard Engine or SuperSQL Engine, you can submit a ticket to contact us.

Engine Pricing

The data engine supports yearly/monthly subscription and pay-as-you-go billing. For more information, see Billing Overview.

Constraints

- The name of a data engine is globally unique and cannot be modified.
- Switching the billing mode of a data engine is not supported.

- A data engine does not support switching regions.

SuperSQL Engine
SuperSQL Engine Overview

Last updated: 2026-07-14 15:38:17

The DLC data engine is the foundation for DLC's data analysis and computing services. All computations performed in DLC require the use of a data engine. You can select a shared engine or a dedicated engine based on your application scenario.

> **Note:**
> The SuperSQL-type engine has been deprecated and is only available for existing users.

Shared Engine

The shared engine (public-engine) is a built-in data engine provided upon DLC service activation. It is suitable for scenarios with low analysis frequency and small data volumes. Users do not need to configure or manage resources. Billing is based on the amount of data scanned per task (for specific pricing, see Billing Overview). No charges are incurred when the engine is not running. The engine offers high flexibility and high availability.
DLC adopts a Serverless architecture. When a task is executed for the first time over a period of time, the data engine needs to be scheduled, which may result in a longer waiting time.

Private Engine

A dedicated engine is a data engine that users purchase for their exclusive use. Resource usage is billed on a pay-as-you-go basis. For specific pricing, see Billing Overview.

- Pay-as-you-go: It is suitable for users whose analysis data is periodic and requires elastic scaling based on business peaks and valleys. It offers high flexibility and stability, with billing based on CU usage.
- Yearly/Monthly subscription: It is suitable for long-term, large-scale, and stable data analysis requirements. It supports elastic scaling based on business peaks and valleys, requires no waiting for resource provisioning, is always available, and is billed monthly based on the cluster specification. When elastic scaling is enabled, the system performs cluster scaling based on the elastic scaling rules. The scaled-out clusters are billed based on CU usage.

Note:

For different application scenarios, a dedicated engine can select different compute engines to address them.

- SparkSQL: It applies to stable and efficient offline SQL tasks.
- Spark jobs: It applies to the processing of Spark-native streaming/batch data jobs.
- Presto: It applies to agile and fast interactive query and analysis.

> ⚠ **Note:**
> The unit price for a dedicated engine is not affected by the type of compute engine selected.

Engine Scaling Rules

Scaling rules are categorized into two types: load-based scaling and time-based scaling. These two types are independent of each other, and users can configure them simultaneously. During the time period when a scheduled scaling rule is active, the system prioritizes the execution of the time-based scaling rule. At other times, the system performs scaling adjustments according to the "load-based scaling" rule.

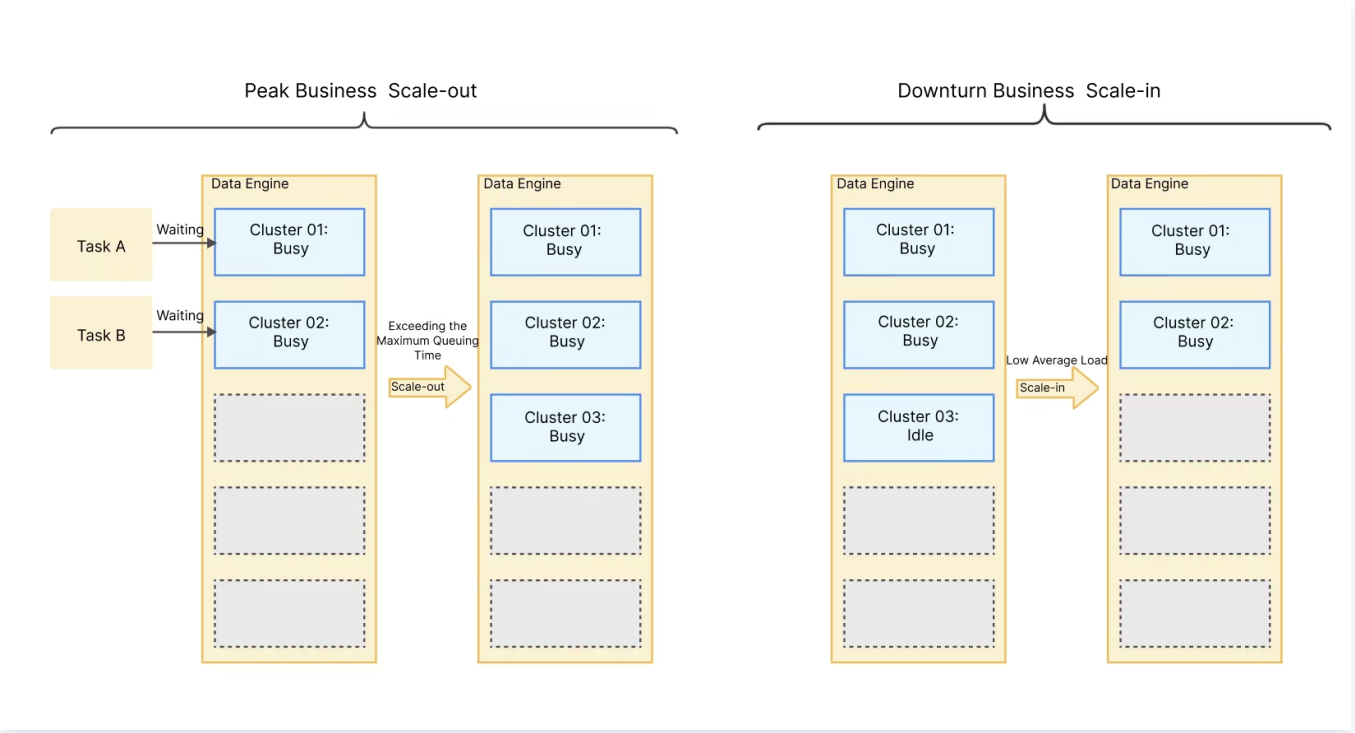
Scaling by Load

Engine load-based scaling rules can be configured on the Create Engine page or in the engine configuration section of the Console Data Engine.
The number of clusters refers to the number of resident clusters in an engine. The number of clusters + the number of elastic clusters = the maximum number of clusters that can be reached when the engine scales.

- Basic rule: Engine scaling occurs only when the number of elastic clusters is greater than 0.
- Scale-out rule: The system scales out the data engine according to the configured rules when the number of queued tasks exceeds the available concurrent capacity, the task queuing time surpasses the maximum queuing time, and no clusters are being initialized.
- Scale-in rule: The system scales in the data engine when the current number of clusters exceeds the number of resident clusters, the overall average cluster load is below 20%, and some clusters are idle.

As shown in the figure below: During purchase, the configuration includes 2 clusters, 3 elastic clusters, and a maximum task queuing time of 5 minutes. When cluster tasks are highly concurrent, if the number of queued tasks exceeds 2 and the queuing time exceeds 5 minutes, the system scales out the data engine to alleviate task queuing. After a successful scale-out and a period of time, when the cluster task queuing situation is alleviated, and there are idle

clusters with low load, the system scales in the data engine.



Under elastic scaling, the number of clusters in a data engine will not be less than the configured number of clusters, nor will it exceed the sum of the configured number of clusters and the number of elastic clusters.
For example: If you configure 2 clusters and 3 elastic clusters during purchase, then after elastic scale-out, the number of clusters will not exceed 5, and after elastic scale-in, the number of clusters will not be less than 2.

> **Note:**
> If a pay-as-you-go cluster is not needed, you can suspend it to avoid resource waste.

Scaling by Time

Notes

1. The scaling-by-time feature is only supported for the SuperSQL Presto and SparkSQL engines.
2. You cannot enable the scaling-by-time feature and scheduled cluster start/stop feature at the same time.

Configuration Path

Navigate to: **Resource Management > SuperSQL Engine > Choose the target engine > Choose the engine name to enter the engine details page > Configuration Information > More > Scale**

Configuration Change > Elasticity Rules > Enable Scheduled Elasticity. You can then configure time-based elasticity.
After you complete the configuration and confirm the change, the configuration takes effect immediately.

Configuration Description

1. Execution type
   1.1 Performing execution once
   - You can self-select a scaling time period and specify the time zone in which it takes effect.
   - The total duration of the effective period cannot exceed 24 hours.
   - The maximum and minimum number of elastic clusters must be specified.
     - When the scaling time period starts, the cluster scales out based on the minimum number.
     - Within the scaling time period, the system can automatically scale out up to the maximum number of clusters allowed based on the load.
     - Up to 10 resident clusters and elastic clusters are allowed.
   - Maximum task queuing time:
     - The maximum wait time for tasks in a queue can be configured.
     - The value can be 0, indicating that auto scale-out is triggered when a queue occurs.
     - When the queuing time exceeds the maximum, the system will trigger auto scale-out of clusters by starting elastic resources. (The start time is subject to the amount of resources.)

1.2 Performing execution repeatedly
   - Multiple scaling plans can be configured under the same execution frequency, and the planned time periods cannot overlap.
   - Scaling plans are executed in chronological order.

   1.2.1 Execution frequencies:
   - **Daily**
     - The total duration of the effective period cannot exceed 24 hours.
     - If the end time is earlier than the start time, the execution is considered as cross-day execution.
   - **Weekly**
     - You can select specific weekdays with multiple selections supported.
   - **Monthly**

- You can select specific dates with multiple selections supported.
- The End of Month option indicates that the execution is performed on the last day of a month (regardless of the number of days in the month).

Engine Status

A cluster is categorized into eight statuses based on its current operational condition: Starting, Running, Paused, Pausing, Resizing, Isolated, Isolating, and Recovering.

- Starting: The cluster resources are being launched, and the pay-as-you-go dedicated engine is not billed during this period. A cluster in the Starting status cannot be selected for data computation.
- Running: The cluster is operational and can be selected for data computation.
- Paused: The cluster is suspended and cannot be selected for data computation.
- Pausing: The cluster is being transitioned to the Paused state, which affects running tasks and prevents the cluster from being selected for data computation.
- Resizing: The cluster is undergoing configuration changes and will be unavailable for data computation during this period.
- Isolated: The cluster is isolated due to account arrears and cannot be selected for data computation.
- Isolating: Due to account arrears, the cluster is being transitioned to the Isolated state. This process affects running tasks and prevents the cluster from being selected for data computation.
- Recovering: The process of a cluster transitioning from the Isolated state back to the Running state after the account is no longer in arrears due to a recharge cannot be selected for data computation.

Purchasing a Dedicated Data Engine

Last updated: 2026-07-14 15:38:44

The dedicated data engine of DLC currently supports pay-as-you-go and yearly/monthly subscription billing modes. For detailed pricing, see Billing Overview.

> **Note:**
> The SuperSQL-type engine has been deprecated and is only available for existing users.

Purchasing a Dedicated Engine

You can purchase DLC by searching for it on the official website purchase page or directly in the console. The console purchase procedure is as follows:

1. Log in to the DLC console and select the region where the service is located. The user who logs in must have Tencent Cloud administrator or financial permissions.
2. In the navigation menu, click SuperSQL Engine to go to the Data Engine Management page.
3. Click **Create Resource** in the upper-left corner to go to the Resource Configuration and Purchase page. After completing the configuration based on your requirements, you can view the estimated price.
4. After confirming the price, you can proceed with the purchase.

Purchase Configuration Parameter Description:

- Region: The private networks of cloud products in different regions are not interconnected. The region cannot be changed after purchase, so choose carefully. We recommend selecting the region closest to your customers to reduce access latency.
- Compute engine: Supports both Presto and Spark engines. The engine cannot be changed after purchase, so choose carefully. The Presto engine applies to faster interactive query and analysis and supports multi-source federated queries. The Spark engine applies to offline tasks with large data volumes, and task execution is more stable.
- Cluster specification: The cluster specification uses compute units (CUs) as the unit of measurement. 1 CU is approximately equivalent to the computing resources of 1 CPU core and 4 GB memory. The specification size determines the amount of computing resources available during task execution. You can make a purchase based on your actual requirements.

> **Note:**

> If you need to purchase resources exceeding 256 CUs, contact us via a ticket so we can prepare inventory in advance. This prevents insufficient stock from affecting your usage at the time of purchase.

- Automatic startup: If the automatic startup is enabled, the data engine will automatically start when it is suspended and receives a task request.

> ⚠ **Note:**
> Pay-as-you-go resources are not reserved resources, so they may fail to be launched. If you need computing resources for stable running over a long period, you can choose to purchase a data engine with a yearly/monthly subscription.

- Suspension policy: You can configure the suspension method for pay-as-you-go data engines. Both automatic and scheduled suspension policies are supported. No fees will be incurred after a pay-as-you-go data engine is suspended.
  - Automatic suspension: The data engine will automatically switch to the paused state after a period of no tasks.
  - Scheduled policy: You can configure weekly scheduled startup and suspension policies. The system will periodically start and suspend clusters according to the configuration rules.
    - Suspend after task ends: If the data engine has tasks in progress at the specified time, it will automatically suspend within 5 minutes after the tasks are completed.
    - Automatic pause and then suspend: When the specified time is reached, if the data engine has tasks in progress, the system will pause the tasks and then immediately suspend the data engine.
- Advanced configuration: If you need to use federated queries, you can configure network segment information in the advanced configuration.
- Tags: This feature supports tagging purchased resources for categorization and cost allocation. For details, see the Associate Tags with Dedicated Engine Resources document.

Query Bills

You can query your product billing through the DLC console. The procedure is as follows:

1. Log in to the DLC console and select the region where the service is located. The user who logs in must have Tencent Cloud administrator or financial permissions.
2. In the navigation menu, click SuperSQL Engine to go to the Data Engine Management page.

3. Click the **Order Query** button to view detailed order and billing information (financial permissions are required).

Renewal Management

For yearly/monthly subscription dedicated data engines, you can go to your resource management page through the renewal management entry in the DLC console to perform operations such as renewal. The procedure is as follows:

1. Log in to the DLC console and select the region where the service is located. The user who logs in must have Tencent Cloud administrator or financial permissions.
2. In the navigation menu, click SuperSQL Engine to go to the Data Engine Management page.
3. Click the **Renewal Management** button to go to the resource list and perform renewal management operations (financial permissions are required).

Renewing a SupersSQL Engine

Last updated: 2026-05-20 15:24:40

You can renew a yearly/monthly subscription data engine that is not expired or is isolated on the DLC console.

1. Log in to the DLC console and select the region where the service is located. The user who logs in must have Tencent Cloud administrator or financial permissions.
2. In the left navigation menu, click SupersSQL Engine to go to the Data Engine Management page.
3. Locate the data engine that needs to be renewed, click **More**, and then select "Renew". For resources nearing expiration (within 7 days), you can also directly click the **Renew** button next to the expiration time to renew them.

4. After confirming the renewal duration and price, click **Confirm Renewal** to submit the renewal order. The renewal is completed once the order is confirmed and the payment is successfully deducted.

> ⚠ **Note:**
> After an isolated data engine is renewed and restored to service, the start date of the new subscription period is the expiration date of the previous period.

Engine-Level Parameter Settings

Last updated: 2026-05-20 15:24:13

> **Note:**
> Currently, the system supports engine configuration for the Spark SQL engine and the Spark job engine.

Spark parameters are settings used to configure and tune Apache Spark applications. In a self-managed Spark environment, these parameters can be set via command-line options, configuration files, or programmatically. In DLC, you can specify Spark parameters within the SQL and code of the SparkSQL engine and Spark job engine, or directly set parameters at the engine level. The engine-level Spark parameter settings are as follows.

Setting Engine-Level Parameters

1. Go to the SuperSQL engine module, click **Parameter Configuration**, and the engine parameter side panel will pop up.
2. For the Spark job engine, you can configure the default resource specifications and parameters for jobs. For the SparkSQL engine, you do not need to pay attention to the default resource specifications for jobs.

Using Engine-Level Parameters

Using Engine-Level Parameters in Spark Jobs

The Spark job engine provides two entry points for job submission: Data Jobs and Data Exploration. Both support the use of engine-level parameters.
When a data job is created, the engine-level parameters and resource configurations are inherited by default. You can override the engine-level parameters using job parameters (--config) and choose whether to inherit the engine-level resource configurations. When the default configuration is selected, the engine-level resource configurations are used.

When SQL is run using the Spark job engine in Data Exploration, the engine-level parameters and resource configurations are inherited by default. You can override the engine-level parameters using the set command in SQL and choose whether to inherit the engine-level resource configurations.

Using Engine-Level Parameters in SparkSQL

The SparkSQL engine does not have engine-level resource parameters, and tasks will use as many cluster resources as possible. Currently, SQL must be submitted in Data Exploration for the SparkSQL engine. When SQL is run using the SparkSQL engine in Data Exploration, the engine-level parameters are inherited by default. You can override the engine-level parameters using the set command in SQL.

Managing a Private Data Engine

Last updated: 2026-05-20 15:52:53

> **Note:**
> The shared data engine is managed uniformly by DLC, requiring no management from users.

Modifying a Private Engine Configuration

> **Note:**
> Changing the configuration of a dedicated engine may result in billing changes. For details, refer to the Configuration Adjustment Fee Description.

Method 1: Data Engine List

1. Log in to the DLC console and select the region where the service is located. The user who logs in must have Tencent Cloud administrator or financial permissions.
2. In the navigation menu bar, click SuperSQL Engine to go to the Data Engine Management page.
3. Locate the dedicated engine you need to modify in the data engine list. Click the **Configure** button on the right to go to the configuration modification page, where you can modify the cluster specifications and scaling policy.
4. After adjusting the parameters as needed, click **Save**. After submitting and paying for the order, you can initiate the configuration change.

Method 2: Data Engine Details

1. Log in to the DLC console and select the region where the service is located. The user who logs in must have Tencent Cloud administrator or financial permissions.
2. In the navigation menu, click SuperSQL Engine to go to the Engine Management page.
3. Locate the dedicated engine you need to modify in the **SuperSQL Engine** list. Click the engine name to go to its details page. Hover your mouse over the **More** option in the upper-right corner of the configuration information, then click **Scale Configuration Change** in the dropdown menu to modify the cluster specifications and scaling policy.
4. After adjusting the parameters as needed, click **Save** to complete the configuration modification.

Modifying Private Engine Information

1. Log in to the DLC console and select the region where the service is located. The user who logs in must have Tencent Cloud administrator permissions.
2. In the navigation menu, click SuperSQL Engine to go to the Engine Management page.
3. Locate the dedicated engine you need to modify in the data engine list. Click the engine name to go to the engine details page. In the Basic Information section, you can modify the **Description**. Hover your mouse over the **More** option in the upper-right corner of the configuration information, then click **Startup and Suspension Policy Configuration** in the dropdown menu to modify the automatic startup and suspension policies.

> **Note:**
> Suspension policy: It supports configurations of suspension methods for the pay-as-you-go **SuperSQL Engine**, including automatic and scheduled suspension policies. No fees will be incurred after suspending a pay-as-you-go engine.
> - Automatic suspension: The SuperSQL engine will automatically switch to the paused state after 15 minutes of no tasks.
> - Scheduled policy: You can configure weekly scheduled startup and suspension policies. The system will periodically start and suspend clusters according to the configuration rules.
>   - Suspend after task ends: If the **SuperSQL Engine** has tasks in progress at the specified time, the system will automatically suspend the engine within 5 minutes after the tasks are completed.
>   - Suspend after automatic pause: If the **SuperSQL Engine** has tasks in progress at the specified time, the system will pause the tasks and immediately suspend the engine.

4. After adjusting the parameters as needed, click **Save** to complete the information modification.

Managing the Startup and Suspension Policies

This feature supports configurations of startup and suspension policies for pay-as-you-go exclusive data engines to facilitate management and cost control.

> **Note:**
> Pay-as-you-go data engines will generate fees if not suspended. Suspend unused data engines promptly.

Startup policy: It supports configurations of startup methods for a pay-as-you-go **SuperSQL Engine**, including automatic startup, manual startup, and scheduled startup.

- Automatic startup: After the configuration, if the data engine is in a suspended state, it will automatically start when a task is submitted to it.
- Manual startup: After the configuration, if the data engine is in a suspended state, you need to manually start the data engine before processing data tasks.
- Scheduled startup: You can configure weekly scheduled startup policies. The system will periodically start clusters according to the configuration rules.

Suspension policy: It supports configurations of suspension methods for pay-as-you-go data engines, including automatic and scheduled suspension policies. No fees will be incurred after suspending a pay-as-you-go SuperSQL engine.

- Automatic suspension: After the configuration, the SuperSQL engine defaults to switching to a suspended state 10 minutes after task completion. The trigger time can also be configured.
- Scheduled policy – Weekly scheduled startup and suspension policies can be configured. The system will periodically start and suspend clusters according to the configured rules.
  - Suspend after task ends: If the data engine has tasks in progress at the specified time, it will automatically suspend within 5 minutes after the tasks are completed.
  - Automatic pause and then suspend: When the specified time is reached, if the data engine has tasks in progress, the system will pause the tasks and then immediately suspend the data engine.

Manually Suspending & Starting a Private Engine

> ⚠ **Note:**
> Yearly/Monthly subscription resources are persistent and do not support suspension.

1. Log in to the DLC console, select the service region, and ensure the logged-in user has administrator permissions.
2. In the navigation menu, click SuperSQL Engine to go to the Data Engine Management page.
3. Locate the dedicated engine that needs to be paused/started in the SuperSQL Engine, hover the mouse over More, and then click the Start/Suspend button in the drop-down menu.

Destroying a Private Engine

If you no longer need to use the **SuperSQL Engine**, you can destroy the data engine. Destroying a yearly/monthly subscription data engine triggers a self-service refund operation. For details, see Refund Instructions.

> **Note:**
> Once a pay-as-you-go data engine is destroyed, it cannot be recovered. Please proceed with caution.

1. Log in to the DLC console and select the region where the service is located. The user who logs in must have Tencent Cloud administrator permissions.
2. In the navigation menu, click SuperSQL Engine to go to the Engine Management page.
3. Locate the dedicated engine that needs to be destroyed in the data engine (only clusters in the "Paused" state can be destroyed), hover the mouse over More, and then click the Destroy button in the drop-down menu.
4. After secondary confirmation, the dedicated engine is successfully destroyed.

Cluster Runtime Logs

For the dedicated engine purchased by the user, **DLC provides cluster operation logs within the last 14 days** to help the user understand the cluster's startup, suspension, and scaling activities. The cluster logs mainly include the following content:

- **Start Time**: The time when the cluster starts working.
- **Suspension Time**: The time when the cluster stops working.
- **Scale-out Records**: The time and quantity of cluster scale-outs.
- **Scale-in Records**: The time and quantity of cluster scale-ins.

日志信息

时间	动作	详情
2023-07-27 16:00:08	引擎挂起	引擎已挂起，原因：手动挂起
2023-07-27 15:13:47	集群扩容	扩容前：集群个数0个，集群规模16CU，扩容后:集群个数1个，集群规模16CU
2023-07-27 15:12:45	集群恢复	引擎已恢复，原因：自动恢复
2023-07-26 16:30:07	引擎挂起	引擎已挂起，原因：手动挂起
2023-07-26 15:55:13	集群扩容	扩容前：集群个数0个，集群规模16CU，扩容后:集群个数1个，集群规模16CU
2023-07-26 15:53:41	集群恢复	引擎已恢复，原因：自动恢复
2023-07-19 16:00:08	引擎挂起	引擎已挂起，原因：手动挂起
2023-07-19 14:38:21	集群扩容	扩容前：集群个数0个，集群规模16CU，扩容后:集群个数1个，集群规模16CU
2023-07-19 14:36:59	集群恢复	引擎已恢复，原因：自动恢复
2023-07-19 14:30:08	引擎挂起	引擎已挂起，原因：手动挂起

共 181 条

Disaster Recovery Cluster

Last updated: 2026-05-20 15:56:39

To ensure stable running of the computing engine in extreme scenarios, DLC provides efficient and agile disaster recovery cluster capabilities. When you need a disaster recovery cluster, you can quickly switch to it to maintain normal service operation. The disaster recovery cluster is charged only during runtime. For details, see the Billing Instructions.

Operation Steps

1. Go to the DLC console. Click SuperSQL Engine on the left sidebar to go to the engine page.
2. Click the SuperSQL engine resource name to go to the engine details page.

3. Click **Enable Disaster Recovery Cluster** and wait for the disaster recovery cluster to initialize.
4. After the disaster recovery cluster is enabled, go to the disaster recovery cluster information and click **Switch to Disaster Recovery Cluster**. This switches the running cluster to the disaster recovery cluster. From then on, all jobs directed to this data engine are submitted to the disaster recovery cluster. The disaster recovery cluster is used as a transition during extreme failures of the data engine.

5. After the extreme failure is resolved, go to the basic information of the data engine and click **Switch to Primary Cluster**. The disaster recovery cluster will be suspended. From then on, all jobs directed to this data engine are submitted to the primary cluster.

Disaster Recovery Cluster Specifications

The disaster recovery cluster always matches the specifications of the SuperSQL engine as closely as possible to ensure that existing tasks can transition and run normally. When AS is enabled for the SuperSQL engine, the AS rules of the disaster recovery cluster match those of the SuperSQL engine. To save costs, the disaster recovery cluster always uses the pay-as-you-go billing mode.

Fee Instructions

Enabling the disaster recovery cluster is free of charge. When you switch to the disaster recovery cluster and it is running, you are charged according to the pay-as-you-go billing standard for data engines of the same specifications.
Example:

1. When the SuperSQL engine itself is a 16-CU SparkSQL engine with yearly/monthly subscription, the disaster recovery cluster enabled is a 16-CU SparkSQL engine with pay-as-you-go billing. In this case, no fee is charged when the disaster recovery cluster is suspended. After you switch to the disaster recovery cluster and it is running, an additional fee is charged for the CU usage duration of the disaster recovery cluster. For details, see the Billing Overview.
2. When the SuperSQL engine itself is a 16-CU SparkSQL engine with pay-as-you-go billing, the disaster recovery cluster enabled is a 16-CU SparkSQL engine with pay-as-you-go billing. In this case, no fee is charged when the disaster recovery cluster is suspended. After you switch to the disaster recovery cluster and it is running, the primary cluster is suspended. Only the fee for the CU usage duration of the disaster recovery cluster is charged.

Engine Kernel Version

Last updated: 2026-07-14 15:39:14

DLC provides different kernel versions for various application scenarios and incorporates extensive feature and performance optimizations for the kernel. The kernel versions are listed below.

- For scenarios primarily involving interactive queries, we recommend using the latest kernel versions for the Presto engine and the SparkSQL engine.
- For scenarios primarily involving batch jobs, we recommend using the Spark job engine and the Spark 3.2 kernel version.

> **Note:**
> The SuperSQL-type engine has been deprecated and is only available for existing users.

Engine Type	Kernel Version	Description
Presto	SuperSQL-P 1.0	Based on the native Presto 0.242 version, it supports dynamic data source loading, Dynamic Filter enhancement, Iceberg V2 tables, INSERT OVERWRITE for non-partition tables, and execution of Hive UDFs.
SparkSQL	SuperSQL-S 1.0	Based on the native Spark3.2 version, it supports Iceberg 1.1.0, Hudi 0.12.0, and adaptive Shuffle Manager.
	SuperSQL-S 1.0(Native)	Based on the native Spark3.2 version, it supports Iceberg 1.1.0. The current version is a beta version, leveraging the Native runtime environment provided by Tencent Cloud Meson Engine and using technologies such as vectorization to accelerate the execution performance of SQL-based applications, delivering over 2x performance improvement in certain scenarios compared to the SuperSQL 1.0 version.
	SuperSQL-S 3.5	Based on the native Spark3.5 version, it supports Iceberg 1.5.0 and adaptive Shuffle Manager. The current version is a beta version, delivering over 33% performance improvement in certain scenarios compared to the SuperSQL 1.0 version.
SparkBatch	Spark 3.5	Based on the native Spark3.5 version, it supports Iceberg 1.5.0, Python3, and adaptive Shuffle Manager.

		The current version is a beta version, delivering over 33% performance improvement in certain scenarios compared to Spark 3.2.
	Spark 3.2	Based on the native Spark3.2 version, it supports Iceberg 1.1.0, Hudi 0.12.0, and Python3, and supports adaptive Shuffle Manager.
	Spark 3.2(Native)	Based on the native Spark3.2 version, it supports Iceberg 1.1.0. The current version is a beta version, leveraging the Native runtime environment provided by Tencent Cloud Meson Engine and using technologies such as vectorization to accelerate the execution performance of SQL-based applications, delivering over 2x performance improvement in certain scenarios compared to Spark 3.2.
	Spark 2.4	Based on the native Spark2.4 version, it supports Iceberg 0.13.1, Python2, and Python3.

You can also learn about the differences between versions through Engine Kernel Updates.

Associating Tag with Private Engine Resource

Last updated: 2026-05-20 15:57:26

Overview

Tags are used to categorize and organize resources. A Tag consists of a Tag key and a Tag value, and a single Tag key can correspond to multiple values. Users can create and bind Tags to cloud resources to manage them. DLC supports binding Tags to dedicated engines on the product console and purchase pages, enabling multi-dimensional categorization management and billing breakdown for dedicated engine resources through Tags.

Creating a Tag and Associating It with Resources

You can categorize and uniformly manage resources by creating Tags and binding them to dedicated engines.

Operation Steps

1. Log in to the Tag Console to create Tags. For detailed steps, refer to Create Tags and Bind Resources.
2. Log in to the DLC console.
3. In the left sidebar, click the SuperSQL Engine module to go to the Engine List page.
4. Click the resource name to go to the Resource Details page. In the **Basic Information** -> **Tags** section, click the **Edit** icon. The Tag editing window will pop up. Select the Tags you want to bind.
5. Click **Confirm** to successfully bind the Tag to the engine. Click Edit again to unbind or modify the Tag.

Tag Binding on the Purchase Page

Users can bind Tags when purchasing dedicated engine resources. Tag binding is supported in both yearly/monthly subscription and pay-as-you-go billing modes.

Filtering by Tag

You can filter resources by Tag in the DLC console SuperSQL Engine module.

Operation Steps

1. Log in to the DLC console. From the left sidebar, go to the SuperSQL Engine module.

2. Select Tags in the Tag search box. Filtering is supported by **Tag** or **Tag Key**.
3. After selecting a Tag, click **Search** to filter the engine list to show only those containing that Tag.

Cost Allocation by Tag

You can plan Tags for resources based on organizational or business dimensions (such as department, project group, or region) to implement cost allocation management.

Operation Steps

1. Create Tags. Log in to the Tag Console to create Tags.
2. Bind resources. Bind Tags to engine resources in the Tag Console, the DLC console > SuperSQL Engine, or on the purchase page.
3. Set a cost allocation tag: Go to the Billing Center to set a cost allocation tag. For details, see Cost Allocation Tags.
4. Go to the Billing Overview page. Select the By Tag Summary tab to view the bar chart and list of related resources aggregated by Tag Key.

Engine Local Cache

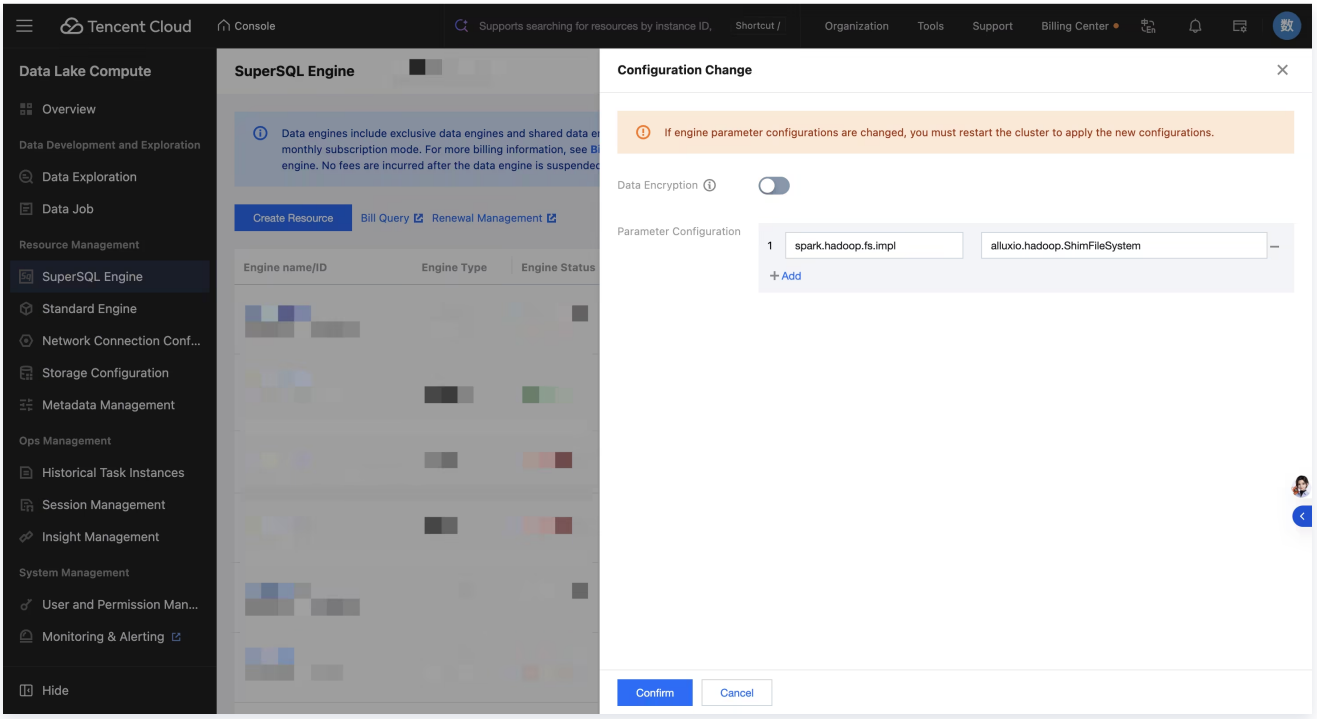
Last updated: 2026-05-20 15:58:04

To ensure stable running of Spark engine query analysis when network bandwidth is limited (for example, when storage system throttling is triggered), the DLC Spark engine provides a Local Cache feature. When you need to cache table data, you can quickly enable the caching feature by adding engine configurations.

Operation Steps

1. To create a Spark engine, see Purchase a Dedicated Data Engine for details.
2. To add caching configurations, go to DLC Console > SuperSQL Engine. Select the engine created in Step 1, click **Parameter Configuration**, and add the configuration items from Cache Configuration Item Description to the engine configurations.

   Spark SQL Engine Configuration:

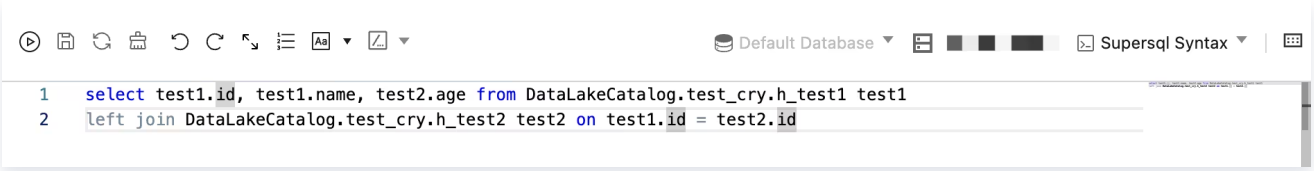


> **Note:**
> After you add configurations, the engine cluster will be restarted. We recommend that you enable caching when no tasks are running to avoid affecting running tasks.

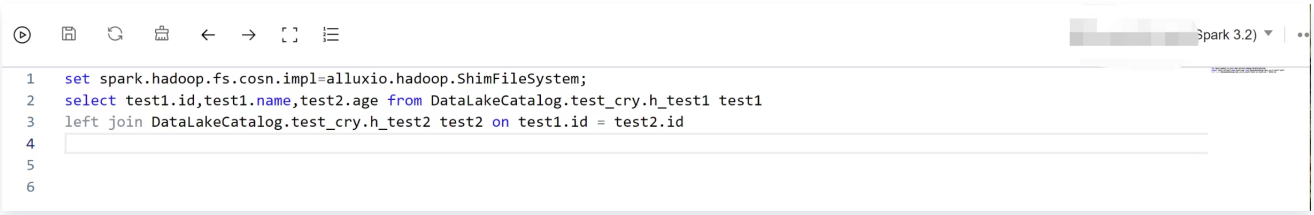
3. To use engine caching, go to the Data Exploration page and the SQL interface to write a query SQL. Select an engine with caching enabled and execute the SQL. After the execution is complete, the engine caches the DLC external tables involved in the SQL to

the local storage. When you execute the SQL again, the data is obtained from the local cache, improving query efficiency.

Spark SQL Engine Query:



Spark Batch Engine Query:



Cache Description

Cache Configuration Items

Configuration Item	Value	Description
spark.hadoop.fs.cosn.impl	alluxio.hadoop.ShimFileSystem	Fixed value. The configuration value is the implementation class for the caching feature. Setting this value enables the caching feature. When the caching feature is enabled, if you set a value other than this one, the engine cannot access COS data. Please strictly follow the guide to configure it. If you need to disable caching after enabling it, delete this configuration item.

Cache Usage Instructions

1. Engine Type Description

SparkSQL Engine: The originally cached data becomes invalid after the engine restarts because the cache is local.

SparkBatch Engine: The SparkBatch engine runs tasks at the Session level. After a task is executed, the cached data becomes invalid.

2. Table Type Description

Currently, only DLC external tables are cached.

Custom Task Scheduling Pools and Resource Usage Limits

Last updated: 2026-05-20 15:58:27

Use Cases

Applicable Engines: Spark SQL engine, resource groups of the standard engine.
When you submit multiple tasks to the engine, such as multiple SQL tasks to a Spark SQL cluster simultaneously, the engine schedules these tasks fairly using the FAIR approach by default during their execution, as the submitted tasks may have dependencies.
However, in certain special cases, you may need to define priorities for specific tasks yourself, as in the following scenarios:

- The submitted task has a high priority and must be executed with the highest priority to avoid queuing for cluster resources.
- The submitted task has a low priority. It is expected to avoid preempting resources from other tasks as much as possible, executing only when resources are available and queuing when they are not.

Custom Scheduling Rules

In the Spark SQL engine, each executed SQL Job is split into a TaskSet, which is a collection of multiple tasks. Our scheduling is based on TaskSets. Whenever cluster resources are idle, a Task is selected from the TaskSets of all jobs according to the scheduling algorithm and dispatched for execution.
Our scheduling algorithm defines multiple scheduling pools, places Jobs/TaskSets into their corresponding pools, and obtains Tasks that need to be dispatched for execution based on these pools.

Limiting the core Usage Limit for Tasks

Applicable Engine: Spark engine. The kernel must be upgraded to a version released after August 20, 2024.
When a large task preempts excessive resources, you can configure a limit on the maximum number of cores a single task can use (resource usage limit). This configuration can take effect at both the engine global level and the task level, with the task-level configuration having higher priority.
For example, configuring `spark.sql.execution.coresLimitNumber=50` means that the task can concurrently use a maximum of 50 cores for execution.

Scheduling Pools and Their Attributes

You can define multiple scheduling pools, each of which has four attributes:

- name: The name of the scheduling pool, which you can assign yourself. You can name it `default`, indicating the default scheduling pool.
- schedulingMode: The scheduling rule, which supports two modes: FIFO and FAIR. It is the scheduling algorithm used when there are multiple TaskSets within a scheduling pool.
  - FAIR: (Recommended, as it avoids starvation issues) Tasks from multiple TaskSets are dispatched fairly. The specific dispatch rules are related to the minShare and weight attributes of the scheduling pool.
  - FIFO: Tasks are dispatched sequentially in the order their TaskSets were submitted.
- minShare: The minimum number of cores that must be satisfied, which must be greater than 0. This is also the minimum number of Tasks that can run. During scheduling, priority is given to ensuring that the number of Tasks running within this scheduling pool reaches the minShare value.
- weight: The weight. Tasks in a scheduling pool with a higher weight are scheduled with priority. Weight comparison occurs only after the minShare requirement is met.

To configure scheduling, you need to author an XML file in the following format:

```
<?xml version="1.0"?>
<allocations>
  <pool name="production">
    <schedulingMode>FAIR</schedulingMode>
    <weight>1</weight>
    <minShare>2</minShare>
  </pool>
  <pool name="test">
    <schedulingMode>FIFO</schedulingMode>
    <weight>2</weight>
    <minShare>3</minShare>
  </pool>
</allocations>
```

Scheduling Configuration Reference Examples

You can configure three scheduling pools as a reference. This configuration defines three resource pools (default, slowTrigger, and fastTrigger).

- slowTrigger low-priority task resource pool: Scheduling mode (schedulingMode): FAIR, which means tasks within this resource pool will have resources allocated evenly. Weight (weight): 1. This indicates that the priority of this resource pool during resource allocation is 1, making it the relatively lowest priority.

- default default resource pool: Scheduling mode (schedulingMode): FAIR, which means tasks within this resource pool will have resources allocated evenly. Weight (weight): 10, which means the priority of this resource pool during resource allocation is 10. Minimum resource share (minShare): 5, which means this resource pool has the opportunity to obtain 5 cores, even when resources are constrained.
- fastTrigger high-priority task resource pool: Scheduling mode (schedulingMode): FAIR, which means tasks within this resource pool will have resources allocated evenly. Weight (weight): 100. This indicates that the priority of this resource pool during resource allocation is 100, making it the relatively highest priority. Minimum resource share (minShare): 5, which means this resource pool has the opportunity to obtain 5 cores, even when resources are constrained.

Based on this configuration, the fastTrigger resource pool will have the highest priority because its weight is set to 100. This means that during resource allocation, it will have the opportunity to obtain more resources compared to other resource pools.

```
<?xml version="1.0"?>
<allocations>
  <pool name="slowTrigger">
    <schedulingMode>FAIR</schedulingMode>
    <weight>1</weight>
    <minShare>1</minShare>
  </pool>
  <pool name="default">
    <schedulingMode>FAIR</schedulingMode>
    <weight>10</weight>
    <minShare>5</minShare>
  </pool>
  <pool name="fastTrigger">
    <schedulingMode>FAIR</schedulingMode>
    <weight>100</weight>
    <minShare>5</minShare>
  </pool>
</allocations>
```

Operation Method

1. After preparing the scheduling pool XML file, place it in a COS path, for example, cosn://bucket-appid/fairscheduler.xml.

2. In SuperSQL Engine Configuration, click **Parameter Configuration** for the corresponding engine and add the following configurations.

   Configure the parameter spark.scheduler.allocation.file with the value set to the path of your scheduling pool XML file: cosn://bucket-appid/fairscheduler.xml.

   This operation requires a cluster restart.
3. When submitting a task, specify the following parameters as task parameters: spark.scheduler.pool = the name of the scheduling pool to which the task is to be submitted. If the parameter is the default scheduling pool, you can omit this specification.

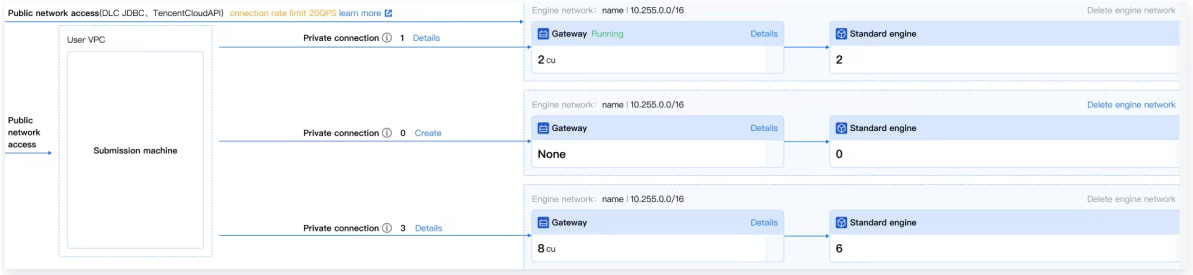
Must-Knows

Scheduling occurs when: the cluster has idle resources and there are tasks that need to be scheduled. Consequently, if the cluster is fully occupied by a task, such as a slow task, the scheduler must wait for one Task of that job to complete before it can begin scheduling other higher-priority tasks. Therefore, it is important to note that the execution time of a single Task for a slow job should be relatively reasonable. Otherwise, it may still lead to prolonged occupation of cluster resources.

Standard Engine
Standard Engine Architecture

Last updated: 2026-05-20 16:08:11

The standard engine architecture consists of several core components: the engine network, gateway, resource group, endpoint, and executor. Before using the DLC standard engine, you also need to understand these concepts:



Note:
The Presto engine is deprecated and only available for existing users.

Concepts

The table provides a brief introduction to several core concepts of the standard engine architecture. For more details, click the relevant links.

Concept	Description
Engine network	An engine network is a managed Private Link for deploying gateways and standard engines, providing a logically isolated network environment. Users can customize the IP address range and subnet of the engine network based on business requirements.
Gateway	The gateway is implemented based on the big data component kyuubi, serving as the access entry for the standard engine service and providing users with a more efficient and stable task submission experience.
Standard engine	A standard engine is a type of computing resource provided by DLC, helping users quickly launch computing clusters of specific specifications. It offers comprehensive support for native syntax and behavior, enabling users familiar with the big data ecosystem to get started more quickly and use it with ease.

Resource group	The standard Spark engine supports using resource groups to further partition engine resources on demand. A resource group is a collective term for a portion of the standard Spark engine's computing resources and their corresponding configurations. SQL tasks can be submitted to a specified resource group for execution.
Private Link	Private Link enables network connectivity between a user account's VPC and the standard engine, allowing users to submit tasks from servers within that VPC.
Submission server	After a VPC endpoint is created, any server within the user account's VPC bound to that endpoint can serve as a submission server for task submission.

Task Submission Methods

Users can submit tasks in multiple ways:

1. Access via JDBC on the submission server, as shown in the figure.
2. Submit SQL tasks through the Data Exploration page of the DLC console.
3. Submit Spark batch/streaming jobs through the Data Jobs page of the DLC console.
4. Submit tasks via TencentCloud API.

Quickly Purchasing and Configuring a Standard Engine

1. If you are purchasing a Standard Engine for the first time, DLC recommends that you quickly configure it by following the Standard Engine Configuration Guide in the documentation.
2. After completing the purchase, you can submit tasks through the Data Exploration page or the submission server.

Standard Engine Introduction

Last updated: 2026-05-20 16:08:37

The Standard Engine is a type of computing resource provided by DLC. It helps users quickly launch compute clusters of specified specifications. The Standard Engine offers comprehensive support for native syntax and behavior, enabling users familiar with the big data ecosystem to get started more rapidly and use it with ease.

Standard Engine Types

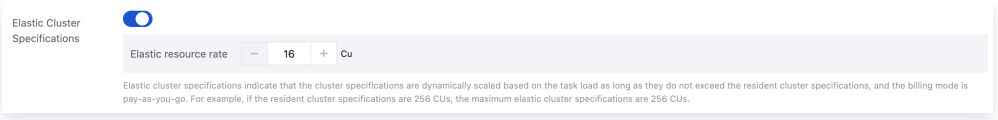
Users can select different standard engine kernels based on their requirements to address different application scenarios. Standard engines are categorized into the following types:

- **Spark**: It applies to stable and efficient offline SQL tasks and native Spark stream/batch data job processing.
- **Presto**: It applies to agile and fast interactive query and analysis.
- Gateway: It is a special type of standard engine, implemented based on native Kyuubi. The gateway is used for users to connect to Spark/Presto compute engines and submit tasks, and it is a prerequisite for using other compute engines.

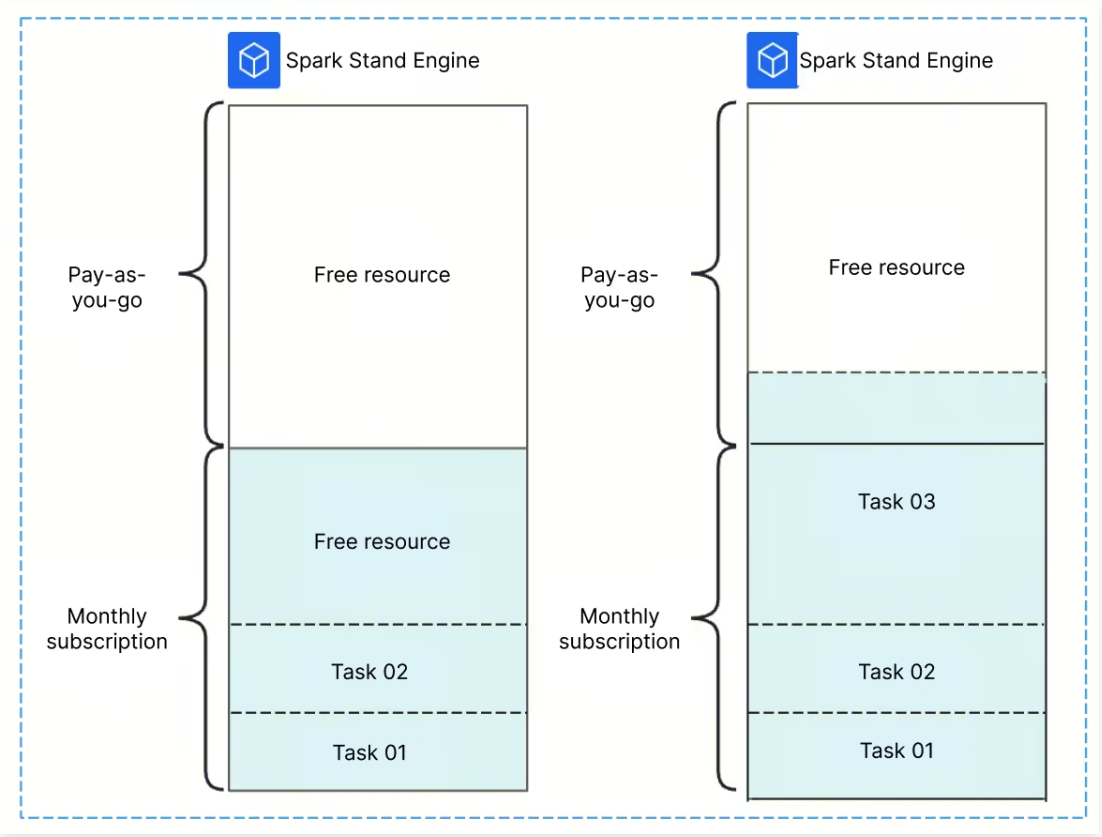
> **Note:**
> 1. The unit price for an engine is not affected by the engine type. For detailed pricing, see Billing Overview.
> 2. The Presto engine is deprecated and only available for existing users.

Engine Elasticity

Currently, only the yearly/monthly subscription Spark standard engine supports configuring pay-as-you-go resource elasticity.



As shown in the figure, tasks and resource groups first use the yearly/monthly subscription resources. If the yearly/monthly subscription resources are exhausted by user-submitted tasks, subsequently submitted tasks will use the configured pay-as-you-go elastic resources. In the figure, after the resources allocated to Task 03 have consumed all the yearly/monthly subscription resources and are still insufficient, the pay-as-you-go resources will continue to be used.



Note:

1. Pay-as-you-go elastic resources are billed based on the actual compute resources used.
2. If started tasks and resource groups are scheduled to pay-as-you-go resources, the resource groups will continue to use the pay-as-you-go resources even after the yearly/monthly subscription resources are released. They will be scheduled to the yearly/monthly subscription resources only after the resource groups are restarted.
3. The upper limit for elastic resources that can be configured for a single Spark standard engine does not exceed the yearly/monthly subscription resource quantity. For example, a 128CU yearly/monthly subscription engine can have at most 128CU of elastic resources. If you need to configure more elastic resources, contact us by submitting a ticket.

Standard Engine Terminology

Term	Description
Cluster Type	When purchasing a Standard Spark engine, you can select a cluster type. For the Standard type, 1 CU is approximately equal to 1-core 4 GB, and for

	the Memory type, 1 CU is approximately equal to 1-core 8 GB. Different types have different unit prices. For details, see Billing Overview.
Elastic Cluster Specifications	The yearly/monthly subscription Spark engine allows users to configure elastic specifications. After the yearly/monthly subscription resource package is exhausted, the system automatically launches pay-as-you-go resource usage based on the user configuration.
Gateway Name	The name of a gateway, which is globally unique. It cannot be the same as the names of other gateways or compute engines.
Engine Name	The name of an engine, which is globally unique. It cannot be the same as the names of other gateways or compute engines.
Engine Type	Standard engine types are divided into Presto engines and Spark engines, and a gateway is also a special type of standard engine.
Engine Status	The status of a standard engine. Based on the current operational status of the cluster, it is divided into eight states: Starting, Running, Ready, Paused, Pausing, Resizing, Isolated, Isolating, and Recovering. • Starting: The cluster resources are being launched, and the pay-as-you-go engine is not billed during this period. A cluster in the Starting status cannot be selected for data computation. • Running: The cluster is operational and can be selected for data computation. • Ready: It is a state identical to Running, indicating that the engine is available for use. • Paused: The cluster is suspended and cannot be selected for data computation. • Pausing: The cluster is being transitioned to the Paused state, which affects running tasks and prevents the cluster from being selected for data computation. • Resizing: The cluster is undergoing configuration changes and will be unavailable for data computation during this period. • Isolated: The cluster is isolated due to account arrears and cannot be selected for data computation. • Isolating: Due to account arrears, the cluster is being transitioned to the Isolated state. This process affects running tasks and prevents the cluster from being selected for data computation. • Recovering: The cluster cannot be selected for data computation during the process of transitioning from the Isolated state back to the Running state after the account is no longer in arrears due to a recharge.
Number of	Current number of resource groups under Standard Spark.

Resource Groups	
Used/Total Resources	Quantity of resources currently used by the engine and total resources available to the engine. • The total resource quantity is the sum of the resident resources and the elastic resources. • The used resources include the resource usage statistics of the DLC deployment service system. The statistical data has a certain delay.
Billing Type	• } • The gateway supports only the yearly/monthly subscription mode. Standard Spark and Presto engines support yearly/monthly subscription and pay-as-you-go.
Auto-renewal or Not	Whether a yearly/monthly subscription engine is automatically renewed upon expiration.
Engine Scale	Total amount of resources available to the engine, in CU. The scale of a yearly/monthly subscription engine includes the engine's resident specification and the pay-as-you-go elastic specification. **Note:** 1. A yearly/monthly subscription engine is charged a one-time fee upon purchase. The engine status does not affect the billing cost. 2. The pay-as-you-go engine charges based on user usage: • The standard Presto engine is not charged when suspended, but is charged while running. During engine startup, partial charges are incurred. • No charges are incurred for the standard Spark engine in the Ready state. Charges are incurred only when a user submits a task or starts a resource group to run.

Standard Engine Kernel Version

Last updated: 2026-05-20 16:08:55

The kernel version used by the DLC standard engine is described below:

Engine Type	Kernel Version	Description
Spark	Standard-S 1.0	Standard-S 1.0 is an engine kernel self-developed based on Spark 3.2, compatible with native Spark syntax and behavior, and applies to offline SQL tasks. On this basis, it supports Iceberg 1.1.0, Hudi 0.12.0, and Python3, and supports the adaptive Shuffle Manager.
Presto	Standard-P 1.0	Standard-P 1.0 is an engine kernel self-developed based on Presto 0.242, compatible with native Presto syntax and behavior, and applies to interactive query and analysis. On this basis, it supports dynamic data source loading, Dynamic Filter enhancement, Iceberg V2 tables, INSERT OVERWRITE for non-partition tables, and execution of Hive UDFs.

Note:
The Presto engine is deprecated and only available for existing users.

Standard Engine Kernel Management

Last updated: 2026-05-20 16:09:15

The DLC (Data Lake Compute) Standard Engine provides the kernel management feature. It supports users in upgrading and rolling back the engine base image editions, helping them flexibly manage the computing environment to obtain the latest features or revert to a stable version.

Feature Entry

In the DLC console, go to Standard Engine > Engine Details > Basic Configuration > Kernel Management to manage image versions.

Version upgrade

Feature Overview

The version upgrade feature supports automatically upgrading the current engine base image to the latest version, allowing you to obtain the latest features and performance optimizations.

Operation Steps

1. Go to **Standard Engine > Engine Details > Basic Configuration > Kernel Management.**
2. Click **Version Upgrade.**
3. Review the image configuration comparison to confirm the upgrade details.
4. Select whether to automatically restart running resource groups as needed (selected by default).
5. After confirmation, perform the upgrade operation.

Effective Scope

Range type	Description
Will be upgraded	Engine base image Standard S1.1 / Standard – S1.1(native)
Will not be changed	Machine learning image
Will not be changed	User custom image

Effective Time

The effective timing of version upgrades varies for different types of resource groups:

Resource group type	Effective Time
SQL analytics resource group	Takes effect immediately after restart.
Job resource group	Takes effect upon the next task submission.
Machine learning resource group (if the Standard S1.1 image is selected)	Requires rebuilding the corresponding Spark application or creating a new Notebook to take effect.

Effective Scope

- Select (default): Automatically restart running resource groups after the upgrade to make the new version take effect immediately.
- Do not select: No automatic restart after the upgrade. The new version will take effect upon the next restart or task submission.

Version Rollback

Feature Overview

The version rollback feature supports rolling back the current engine base image to the version previously used by the user, addressing potential compatibility issues that may arise after an upgrade.

Operation Steps

1. Go to **Standard Engine > Engine Details > Basic Configuration > Kernel Management.**
2. Click **Version Rollback.**
3. Review the image configuration comparison to confirm the rollback details.
4. Select whether to automatically restart running resource groups as needed (selected by default).
5. After confirmation, perform the rollback operation.

Effective Scope

Range type	Description
Rollback Supported	Engine base image Standard S1.1 / Standard – S1.1(native)

Rollback Not Supported	Machine learning image
Rollback Not Supported	User custom image

Effective Time

The effective timing of version rollback varies for different types of resource groups:

Resource group type	Effective Time
SQL analytics resource group	Takes effect immediately after restart.
Job resource group	Takes effect upon the next task submission.
Machine learning resource group (if the Standard S1.1 image is selected)	Requires rebuilding the corresponding Spark application or creating a new Notebook to take effect.

Effective Scope

- Select (default): Automatically restart running resource groups after the rollback to make the rolled-back version take effect immediately.
- Do not select: No automatic restart after the rollback. The rolled-back version will take effect upon the next restart or task submission.

Must-Knows

Compatibility Risks

Important Note: Differences in syntax compatibility and runtime dependencies may exist between different versions. Upgrading or rolling back may cause historical tasks to fail to run or behave abnormally. It is recommended that you perform the following before the operation:

- Evaluate the dependency of current tasks on specific versions.
- Verify the compatibility of key tasks in the test environment.
- Keep backups of important tasks.

Handling Upgrade Failures

If the image upgrade fails, the system will display an upgrade failure message. You can:

1. Check the failure reason, troubleshoot the issue, and then retry.

2. Contact technical support to obtain assistance.

Best Practices

Scenario	Recommended Action
Obtaining new features	Perform the version upgrade during off-peak business hours.
Encountering compatibility issues	Use version rollback to revert to the previous version.
Upgrading in a production environment	Verify in a test environment first before upgrading the production environment.
Batch task scenario	Do not select automatic restart. Choose an appropriate time to restart manually.

FAQs

What to Do If Tasks Report Errors After Upgrade?

If you encounter task errors after an upgrade, you can use the version rollback feature to revert to the previous version. It is also recommended that you check the error information to confirm whether it is a version compatibility issue.

Will Custom Images Be Affected?

No. Version upgrades and rollbacks only affect the engine base images (Standard S1.1 / Standard - S1.1(native)). User-defined images and machine learning images are not affected.

How Long Does It Take to Take Effect After Upgrade?

The effective time depends on the resource group type:

- SQL Analysis Resource Group: Takes effect immediately after a restart.
- Job Resource Group: Takes effect when the next task is submitted.
- Machine Learning Resource Group: Takes effect after you rebuild the Spark application or create a new Notebook.

If the "Auto Restart" option is selected, the SQL Analysis Resource Group is restarted and takes effect immediately.

Can I Roll Back to Any Historical Version?

Currently, the version rollback feature supports rolling back to the version previously used by the user. If you need to roll back to an earlier version, contact technical support.

Standard Engine Parameter Configuration

Last updated: 2026-05-20 16:09:36

Spark parameters are used to configure and tune Apache Spark application settings.

- In a self-managed Spark deployment, these parameters can be set via command-line options, configuration files, or programmatically.
- In the DLC standard engine, you can set Spark parameters. These parameters take effect when a user submits a Spark job or an interactive SQL query with a custom configuration.

> **Note:**
> 1. Configuration at the Standard Engine level only takes effect on Spark jobs and Batch SQL tasks.
> 2. New tasks take effect only after engine-level configuration is added.

Configuring Standard Spark Engine Parameters

1. Go to the Standard Engine feature.
2. On the list page, select the engine that you want to configure.
3. Click **Parameter Configuration**. An engine parameter side panel pops up.
4. In "Parameter Configuration", click **Add**. After the target configuration is added, click **Confirm**.

Resource Group-Level Parameter Configuration

SQL Analysis Resource Group Parameters Only

Adding Parameters During Creation

When a resource group is created, select "SQL Analysis Only". In the final "Parameter Management" section, select to add parameters.

> **Note:**
> 1. Static parameters take effect only after a resource group is restarted. Dynamic parameters take effect without requiring a resource group restart.
> 2. For dynamic and static parameters, refer to the Spark official website.

3. Configuration for the SQL Analysis Only resource group takes effect only when you use that resource group to run SQL tasks.

Modifying Resource Group Parameters

1. On the Standard Engine List Page, select the engine you want to modify, and click **Go To**.
2. On the engine's resource group management page, select an SQL Analysis Only resource group, and click **Details**.
3. On the details page, click **Edit** in the configuration management module to add parameters or modify or delete existing parameters. Similarly, static parameters take effect only after a resource group is restarted, while dynamic parameters take effect without requiring a resource group restart.
4. After completing the modifications, click **Save**. At this point, you can choose to restart immediately, or choose to save without restarting the resource group. You can then restart the resource group at an appropriate time later to make the configuration take effect.

AI Resource Group Parameters

Note:

1. Currently, only AI resource groups of the Spark MLlib type support adding configurations.
2. Currently, only static configurations can be added. These configurations take effect only on new notebook sessions. They do not take effect on existing sessions.
3. The AI resource group feature is an allowlist feature. To ensure it meets your application scenario, please submit a ticket to contact us for evaluation before enabling it.
4. This resource group supports only the Standard-S 1.1 version of the standard Spark engine.

Adding Parameters During Creation

When creating an AI resource group, select the Spark MLlib type. In the final "Parameter Management" section, choose to add parameters.

Modifying AI Resource Group Parameters

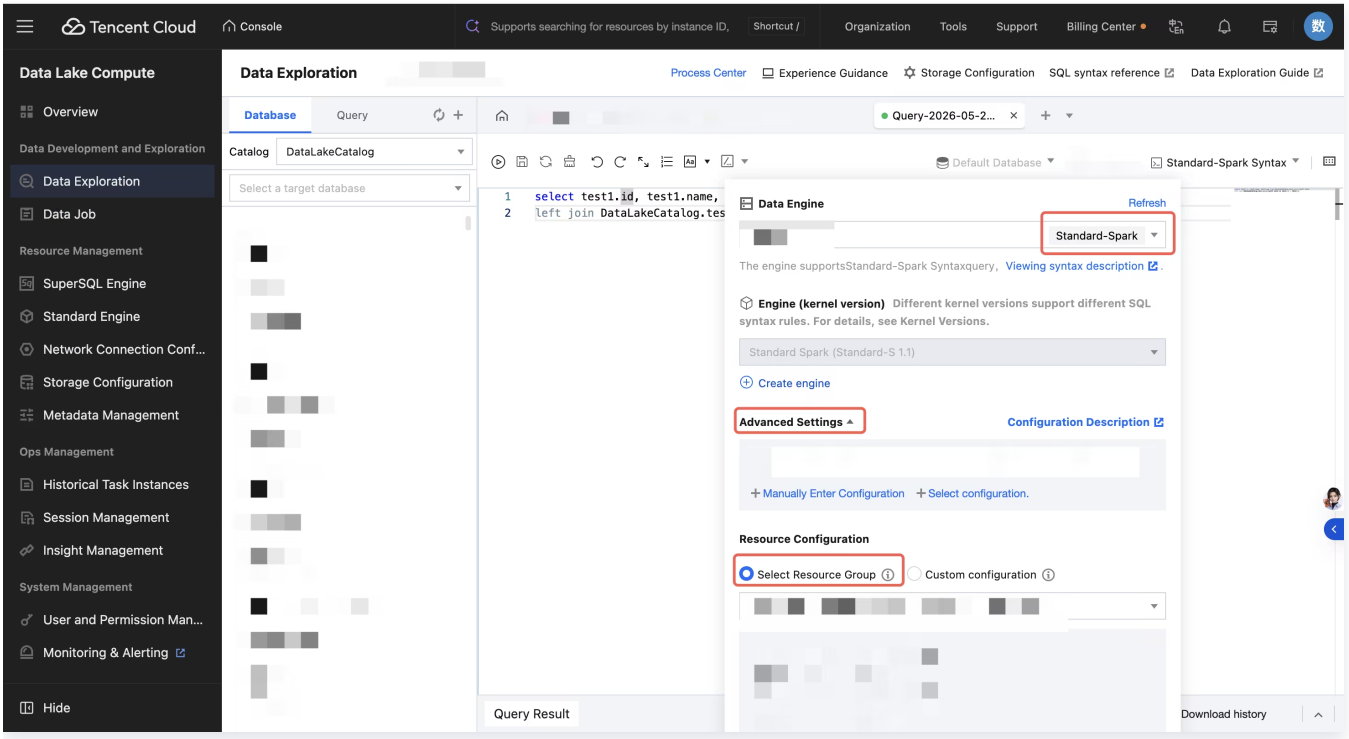
1. On the Standard Engine List Page, select the engine you want to modify, and click **Go To**.
2. On the engine's resource group management page, select a Spark MLlib type resource group, and click **Details**.

3. On the details page, click **Edit** in the configuration management module to add parameters or modify or delete existing parameters. Note that modified parameters take effect only on notebook sessions launched after the modification. They do not take effect on existing sessions.

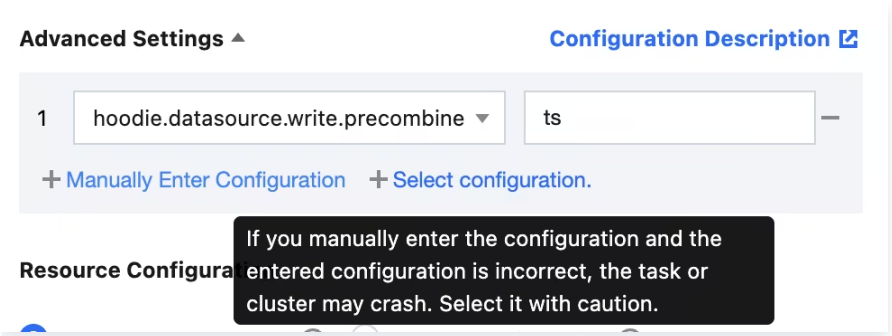
Data Exploration Parameters

> **Note:**
> 1. Currently, only SQL Analysis resource groups support adding parameters on the Data Exploration page.
> 2. Note that only dynamic spark configurations take effect in subsequently executed SQL statements. Static parameters do not take effect.
> 3. The configuration priority at Data Exploration is higher than that of parameter configurations at the engine level and resource group level.

As shown in the figure, on the Data Exploration page, select the standard Spark engine for the data engine, select the resource group for resource configuration, and then select "Advanced Configuration" on the page to add configurations.



As shown in the figure, you can select a built-in configuration or manually enter a configuration.



Spark Job Parameters

> **Note:**
> 1. Modified job parameters take effect only on jobs launched subsequently. They do not take effect on jobs that are already running.
> 2. Job parameter priority is higher than engine-level parameter priority.

Adding Parameters When Creating a Job

Go to Data Jobs, click **Create Job**, and add parameters in the job parameters section.

Editing Parameters of an Existing Job

1. Click Data Jobs, select an existing job, and click **Edit**.
2. On the job editing page, modify the job parameters. After modification, click **Complete Modification**.

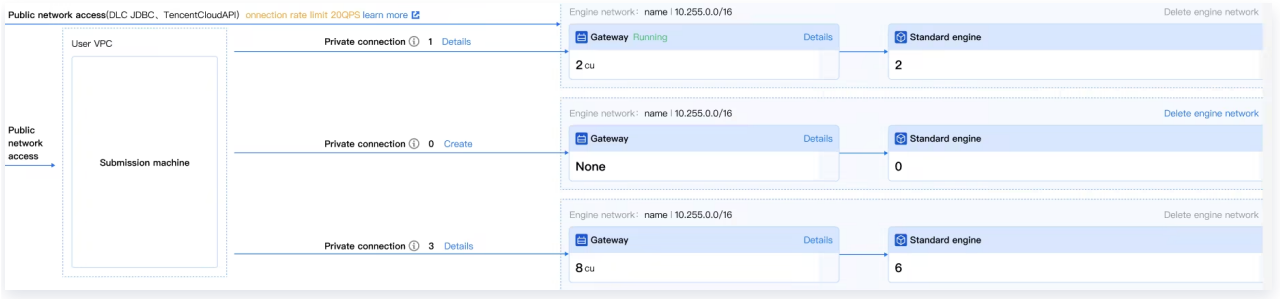
Engine Network Overview

Last updated: 2026-05-20 16:10:18

Concept

The engine network is built on a VPC and assigns fixed network addresses, such as 10.255.0.0/16, to compute engines (standard spark engine, standard presto engine). Each engine network provides a gateway for external access to the standard engines within it. For example, you can access the compute engines via JDBC from a private network (VPC) or a public network.

> **Note:**
> 1. If you need to access resources in different VPCs, for example, to access EMR HDFS data via a DLC engine, select a subnet with sufficient IP addresses and no conflicts with other products. You can purchase multiple compute engines under the same engine network and manage them uniformly through the gateway.
> 2. The Presto engine is deprecated and only available for existing users.



Use Limits

> **Note:**
> The subnet configuration must align with the VPC subnet settings. You create it yourself, and once created, it cannot be modified.

1. You can use any of the following private subnets:
   - 10.0.0.0 – 10.255.255.255 (The subnet mask must be between 12 and 28 bits.)
   - 172.16.0.0 – 172.31.255.255 (The subnet mask must be between 12 and 28 bits.)
   - 192.168.0.0 – 192.168.255.255 (The subnet mask must be between 16 and 28 bits.)
2. Ensure that you allocate a subnet with sufficient IP addresses to the engine network to prevent Pod creation failures due to IP address resource exhaustion when creating large-

scale workloads. If you are uncertain about the scale, use the default configuration.

3. When using federated queries, ensure that the engine subnet does not overlap with the data source subnet.
4. Engine network configurations: You can customize network configurations during the initial activation and purchase. If you need to make changes later, submit a ticket for application.

Network Segmentation

Each engine network is managed by a single gateway, which handles all standard engines within that network. Therefore, properly using engine networks to partition resources can effectively balance the load on gateways and prevent single points of failure. We recommend that you partition resources by business unit or by task type.

Segmentation by Business

We recommend that you partition resources by business unit. For example: allocate at least one engine network per business unit.

Segmentation by Task

We recommend that you partition resources by task type. For example: create separate engine networks for different task types, such as BI analysis, data governance, and data analytics.

> **Note:**
> The engine network partitioning suggestions above are provided as a reference based on our experience. You can also partition resources according to your actual situation. For example, you can create a dedicated engine network to handle exceptionally large task sets.

Private network

You can create a Private Link to establish a secure and stable connection between your VPC and the gateway, enabling access to the standard engine. On the standard engine's CAM page, you can create a new Private Link, select the source VPC and subnet you need to access. After the creation is successful, you will obtain an access link. You can then directly access the standard engine in that engine network from any machine within the source VPC.

Public Network Access

You can also access the standard engine in the engine network via a public network. For example, some BI systems deployed on a public network need to connect to the engine over a

public network.

1. Create a Private Link by referring to the private network access method. For example: a JDBC connection string for private network access.

```
jdbc:hive2://172.22.0.202:10009/?spark.engine=
{DataEngineName};spark.resourcegroup={ResourceGroupName};secretkey=
{SecretKey};secretid={SecretId};region=ap-
singapore;kyuubi.engine.type=SPARK_SQL;kyuubi.engine.share.level=ENGIN
E
```

2. Go to CLB, create a new public network access instance, and select "Configure Listener".
3. Go to the Configure Listener page, select Listener Management, and click Create TCP Listener. The port defaults to the same as the Private Link port: 10009 (for accessing the standard Spark) or 10999 (for accessing the standard Presto).
4. Bind the backend service to the created listener: select the IP address type and enter the Private Link IP address created earlier, for example: 172.22.0.202, with port 10009 (for accessing the standard Spark) or 10999 (for accessing the standard Presto).
5. You can access engine resources via the public network VIP and port 10009 or 10999 provided by the CLB. The access link then becomes a public network access link.

```
jdbc:hive2://{PublicNetworkVIP}:10009/?spark.engine=
{DataEngineName};spark.resourcegroup={ResourceGroupName};secretkey=
{SecretKey};secretid={SecretId};region=ap-
singapore;kyuubi.engine.type=SPARK_SQL;kyuubi.engine.share.level=ENGIN
E
```

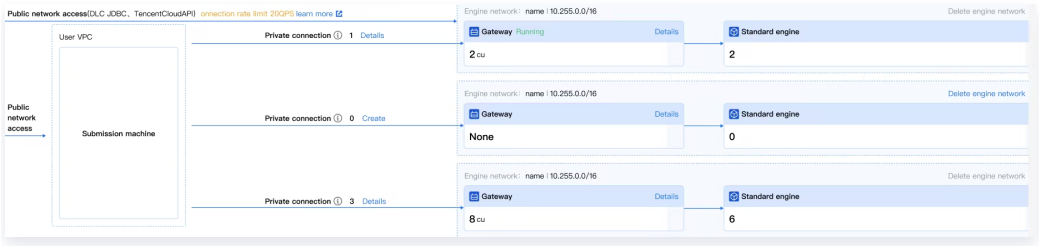
Accessing the Public Network from Within the Engine

The standard engine does not support public network access by default. If you require public network access, for example, to install Python packages via the magic %pip command in a notebook, please submit a ticket to apply.

Gateway Overview

Last updated: 2026-05-20 16:10:37

The DLC Gateway is a Serverless unified access gateway service deeply optimized based on Apache Kyuubi. Through the gateway, you can achieve stable and secure access to DLC data and standard compute engines using standard interfaces such as Hive JDBC/Presto JDBC/DLC JDBC/TencentCloud API. This reduces the complexity of managing access to compute engines at scale. For example, you can submit SQL tasks and ETL jobs to specified standard compute engines via the gateway.



> **Note:**
> The Presto engine is deprecated and only available for existing users.

Gateway Value

The Gateway is a service exclusive to the DLC Standard Engine. It offers users advantages such as **reduced query latency, secure high availability, and flexible access**:

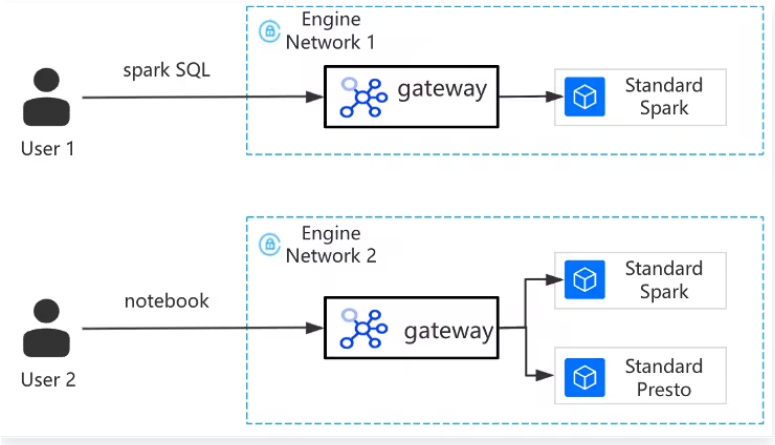
- **Reduced Query Latency**: The DLC Gateway can significantly reduce query link latency and improve performance, especially for interactive analysis of small data volumes.
- **Support for More Access Methods**: The gateway supports connecting to the DLC Standard Engine via Hive JDBC/Presto JDBC, meeting the needs of various query scenarios.
- **Enterprise-Grade High Security**: Authentication and sub-user engine permission control are performed using CAM authentication parameters (AK/SK).
- **High Availability**: The gateway provides higher availability and CLB, supporting scale-out to handle extremely high concurrent queries.

Gateway Architecture

As shown in the figure, only one gateway can be created under an engine network. This gateway manages all standard Spark engines and Presto engines created within that engine network.
By default, a user can only have one engine network and create one gateway. If your business scenario is complex, has high performance requirements such as for concurrency, or if

certain critical business operations require environmental isolation, we recommend that you create multiple engine networks and multiple gateways to physically isolate different tasks.

> **Note:**
> 1. Creating multiple engine networks and gateways requires the backend to open an allowlist. Please submit a ticket to apply.
> 2. Different engine networks and gateways are physically isolated from each other. They cannot communicate with each other or access each other's engines.



Creating an Engine Network and a Gateway

When the allowlist is not enabled, a user has one engine network by default. The user does not need to manually create a gateway. When the user creates the first engine or submits the first task under this engine network, DLC automatically creates a free gateway with a 2CU specification under that engine network.
After the allowlist is enabled, users can create multiple engine networks, as shown in the following figure. Users can create a new engine network by clicking **New Engine Network**. No gateway exists under the newly created engine network initially. Similarly, when a user creates the first engine or submits the first task under this engine network, DLC automatically creates a free gateway with a 2CU specification under that engine network.
Users can see which engine network the current engine belongs to via the "Engine Network Name/ID" column on the engine list page.
Click **Expand** on the upper right to view the engine network list information. Click **Details** to view the detailed information of the current engine network, including the number of standard engines under the network, how many users' VPCs the network is connected to, and the gateway specifications.

> **Note:**
> To avoid accidental deletion, users cannot directly delete an engine network. Users can only click **Delete Engine Network** to delete the network when the number of

standard engines under the current engine network is 0.

Gateway Specifications

DLC automatically creates a 2CU-specification gateway for each engine network, and this 2CU gateway is free of charge. However, the 2CU gateway is only applicable to test environments. For production environments, it is recommended that users scale out the gateway.
DLC provides multiple gateway specifications. It is recommended that you select a gateway specification based on the number of engines you need to manage and the maximum query concurrency QPS of your business scenario. For details, see the following table:

Gateway specification	Gateway HA or Not	Number of Managed Spark Resource Groups	Number of Presto engines managed	Concurrent number of queries for Spark SQL / Presto SQL	Instantaneous Concurrent number of creations / Maximum number of creations for Spark ML Notebook Session	Spark Batch Instantaneous Concurrent number of task submissions / Parallel Running number of tasks
2CU	No	50	4	100	10/20	30/50
16CU	Yes	150	12	200	20/80	80/150
32CU	Yes	400	35	600	100/200	220/400
64CU	Yes	700	70	1000	200/300	400/600

Upgrading Specifications

DLC provides a 2CU specification for users by default. When the business scenario cannot be met and you need to upgrade the specification, you must purchase and obtain it.

Note:

1. Gateway specification change causes all currently running tasks to be interrupted and fail. Proceed with caution!

2. The entire change process is expected to take 10 to 15 minutes. If the gateway status does not return to Running for an extended period, submit a ticket to resolve the issue.

To upgrade the gateway specification, follow these steps:

1. Click Standard Engine on the left navigation bar to go to the engine list page.
2. On the overview card, click **Engine Network > Gateway > Details** to go to the engine network details page.
3. On the details page, scroll down to the bottom and click **Specification Configuration** under Gateway.
4. On the "Configuration Change" page that pops up, select the target specification and click **Confirm Change**.

FAQs

How to Resolve API Timeout Errors When Submitting Tasks via JDBC?

First, check whether the gateway status is normal and whether it is running through the console. If the gateway is in a suspended state, you can retry after starting the gateway using the Start button. Go to the engine network details page, locate the gateway details at the bottom, and click **Start**.

How to Determine Whether the Current Gateway Load Is Normal?

DLC provides basic monitoring for the gateway. You can assess the gateway's health status based on the monitoring information. Go to the engine network details page, locate the gateway details at the bottom, and click **Monitoring** to go to the gateway monitoring page. Users can also configure alarms on the Tencent Cloud Observability Platform (TCOP). When the gateway's CPU or memory utilization exceeds a certain limit, the alarm can immediately notify the customer, enabling proactive operations such as gateway scale-out.
The configuration process is as follows:

1. Go to Tencent Cloud Observability Platform (TCOP), select "Alarm Configuration", and click **Create Policy**.
2. Policy Name: Any name.

Policy Type: DLC/Gateway (Multi-dimensional).

Alarm Targets: For Region, select the region where the gateway is located. For Gateway, select the gateways for which you want to configure alarms. Multiple gateways can be selected.

Trigger Condition: Configure manually. As shown in the figure, an alarm is triggered when either the cpu load or memory utilization exceeds 70%. Users can configure other alarms based on their own requirements.

3. Click **Next to Configure Alarm Notifications**. If an alarm notification template already exists, you can reuse it. If you choose not to create a new template, configure the users to be notified when an alarm is triggered or distribute the alarm to a WeChat group.
4. After the notification template is configured, click **Finish**.

Standard Engine Start/Stop Logs

Last updated: 2026-05-20 16:10:52

The Standard Engine Start/Stop Log feature records the start and suspend events of each engine, facilitating engine status monitoring, troubleshooting, and resource management optimization.

Operation Steps

1. Log in to the DLC console > Resource Management > Standard Engine and select the service region.
2. Start/Stop logs for different operation objects:
   - Gateway: In the engine list, select the engine instance you want to view, click **Resource Group Management**, and scroll down the page to view the gateway start/stop logs.
   - Presto engine: In the engine list, select the engine instance you want to view, click **Engine Name**, go to the Basic Configuration page, and view the compute engine start/stop logs.
   - Spark engine resource group: In the engine list, select the engine instance you want to view, click **Resource Group Management**, select the resource group you want to view, click **Resource Group Name**, go to the resource group details page, and view the resource group start/stop logs.

Start/Stop Log List

Note:

1. The feature supports start/stop logs for Spark engine resource groups, but requires a gateway restart operation after March 20, 2025. Procedure: On the overview card, click **Engine Network > Gateway > Details** for the target operation, go to the engine network details page, click **Suspend**, and then click **Start**.
2. The Presto engine is deprecated and only available for existing users.

Field Name	Description
TraceId	TraceId is a unique identifier for a start-stop process, which can associate log records of different actions within the same process, helping users identify which logs belong to the same operation or request.
Time	The time when an action starts corresponds to the operation start time, and the time when an action

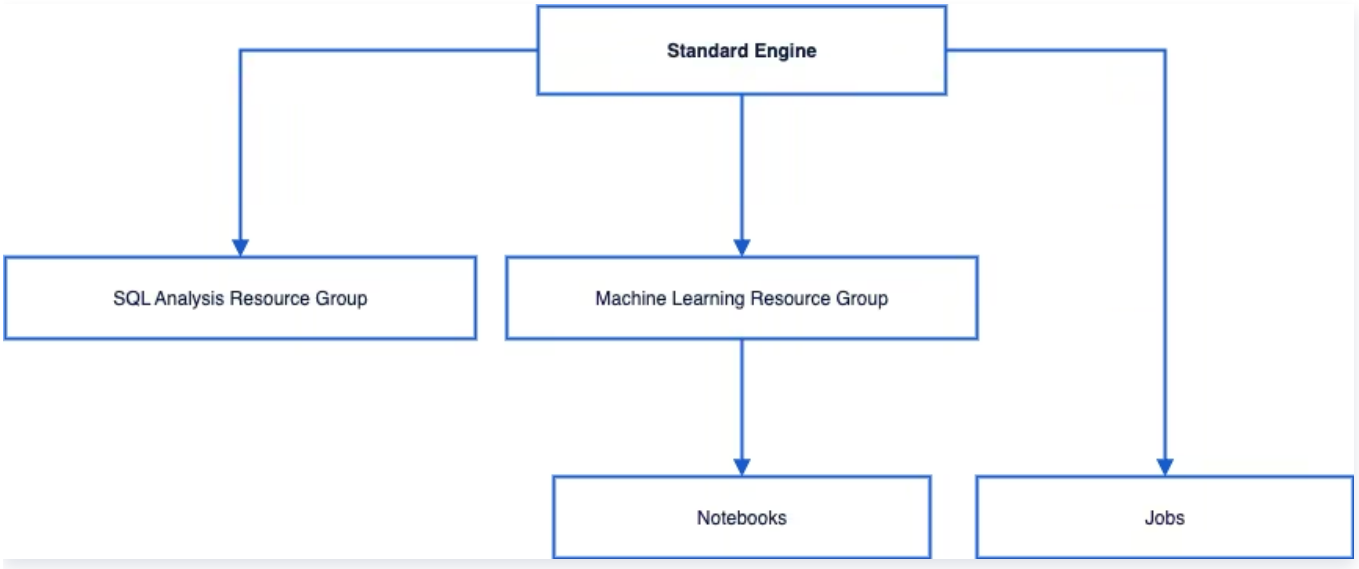
	completes corresponds to the operation end time.
Action.	Start, Suspend, Starting, and so on
Details	CU change before and after the object operation.

Dependency Package Management Guide

Last updated: 2026-05-20 16:11:10

This document introduces the three-tier model and operational methods for dependency package management in DLC, helping you efficiently and controllably configure dependency environments in scenarios such as standard engines, resource groups/jobs, and Notebooks.

Core Concepts



Hierarchical Model

To ensure the consistency and controllability of dependency environments, DLC maintains a protected dependency set pre-configured by the engine kernel and standard images above the "Standard Engine" level. This set is called the Engine Kernel Baseline Dependencies (referred to as "Baseline Dependencies"). It serves as the starting point for all dependency resolution and merging. For details about the baseline dependencies of different images, see Runtime Environment.

- **Engine Kernel Baseline Dependencies (Baseline Dependencies)**
  - Description: A pre-configured dependency list delivered with the engine kernel version and standard images.
  - Characteristics: Cannot be uninstalled; cannot be directly overridden via PyPI. Version adjustments must be performed at the engine level through requirements.txt to implement controlled "version pinning/overriding".
  - Function: Serves as the upper limit of the global baseline to ensure platform stability and consistency during runtime.

- **Standard Engine**
  - Function: Performs limited dependency supplementation and controlled overriding on top of the "Baseline Dependencies" to form a tenant-engine-level global dependency baseline.
  - Scope of Impact: All resource groups, jobs, and Notebooks inherit it by default.
- **Resource Group**
  - Resource Groups: SQL Analysis Only resource groups and Machine Learning resource groups.
  - Function: Serves as an isolation layer for teams/business lines, hosting common but non-global dependency configurations.
- **Notebooks and Jobs**
  - Function: Serves as a personalized and experimental minimal-increment dependency, tailored to development and debugging scenarios.

Inheritance and Customization Rules

- **Inheritance Relationship**: Child entities inherit the dependency configurations of their parent by default.
  - For example: Resource groups inherit the Standard Engine dependencies by default.
- **Customized Addition**: Dependencies can be added at any level. Newly added dependencies take effect only at that level and its child levels.
  - For example: Python packages added to the "Machine Learning Resource Group" take effect only for jobs/Notebooks within that resource group.

The final dependency environment for a task instance = Baseline dependencies + Engine dependencies + Resource Group/Job dependencies + Notebook dependencies.

Installation Sequence

- **Installation Order**: Newly added dependencies at the engine/resource group/job level are installed in chronological order based on their "addition time".

Conflict Rules

- **Conflict Semantics Aligned with the Native Ecosystem**:
  - Maven (Jar): Follows the Maven resolution and conflict handling semantics.
  - PyPI (Python): Follows the pip/PEP specifications.
- For details, see Dependency Conflict Handling.

Overrides and Rewrites

- **Cannot Be Directly Overridden**: Engine-built-in dependencies cannot be overridden via PyPI.
- **Controlled Override**: At the engine level, built-in dependencies can have their versions overridden via `requirements.txt`.

Effective Time and Installation Status

- **Effective Actions**: Installation, uninstallation, and cloning take effect when the compute instance starts.
- **Trigger Scenario**:
  - Start a Job
  - Restart the SQL Resource Group
  - Machine Learning Resource Group:
    - ML Open Source Frameworks and Python Dependencies: On the **Wedata Notebook Exploration** page, click `Restart`.
    - Spark MLlib Dependencies: Recreate the Spark session on the **Wedata Notebook Exploration** page.
- **View Installation Status**:
  - Engine Level: Displays the status after the most recent installation on any compute instance (for example, if only the SQL resource group and Job A run sequentially, the status corresponding to Job A is displayed).
  - Resource Group/Job/Notebook Level: Displays the installation status for each respective dimension.
  - You can click a dependency entry to view the installation success/failure details and logs.

Prerequisites

> ⚠ **Note:**
> The dependency package feature is an allowlist feature. To use this feature, submit a ticket to contact after-sales support for activation.

- Purchase Standard Engine Spark.
- If you need to use an existing engine, submit a ticket to contact after-sales support to upgrade the engine image and gateway image to the version dated 2025-09-30 or later.

Operation Guide

1. Engine-Level Dependency Management

- **Navigation Path**: Resource Management → Standard Engine → Select Engine → Dependency Package Management

Installation

- **COS / Local File**
  - Supports Jar and Python packages; upload from a COS bucket or locally.
- **PyPI**
  - Enter the package name and version (following PyPI naming conventions). By default, the Tencent Cloud repository is used, and external sources are supported.
- **Maven**
  - Enter the Maven coordinates. By default, the Tencent Cloud repository is used, and custom remote repositories and dependency exclusions are supported.

Uninstalling

- Select the dependency → Click "Uninstall". The uninstallation is an asynchronous operation and takes effect the next time the compute instance starts.

Cloning

- Clone the dependency environment from another engine with one click. Duplicate dependencies are automatically skipped.

2. Resource Group–Level Dependency Management

- **Navigation Path**: Resource Management → Standard Engine → Select Engine → Resource Group Management → Select Resource Group → Dependency Package Management.
- **Installation/Uninstallation**: Consistent with the engine level.

3. Job–Level Dependency Management

- **Navigation Path**: Data Development and Exploration → Data Jobs → Create/Edit Job → Dependent Resources
- **Description**: Dependencies customized for a single job take effect only for that job.
- **Installation Method**: Consistent with the engine level.

4. Overview of Task Dependency Environments

- **Navigation Path**: Ops Management → Historical Task Instances → Select Task → Dependent Environment.
- **Description**: Displays the final dependency set and installation status of the task instance (covering the engine/resource group/job levels), which is used for rapid troubleshooting.

Dependency Conflict Resolution

Maven(Jar)

- **Direct Dependency Conflict**: The package installed later fails to be installed.
- **Transitive Dependency Conflict**: The system automatically removes conflicting transitive dependencies, and the installation succeeds. You can view the dependency removal records.
- **Class Conflict**: The installation succeeds but may conflict at runtime.
  - Unused Conflict Class
  - Using a Conflict Class

PyPI(Python)

- **Peer-level Repeated Installation**: When a dependency with the same name or a different version is installed repeatedly at the same level, the later installation fails.
- **Cross-level Repeated Installation**:
  - Standard Engine/Resource Group/Job Level
  - Notebook Level

Best Practices

Core Principles

- **Minimize Engine-level Dependencies.**
  - The engine level serves as the global baseline for tenant engines, affecting the entire system. Avoid adding excessive dependencies at the engine level unless you confirm they are public dependencies required for "all scenarios." This helps reduce global coupling and cross-team compatibility costs.
- **Prioritize Using Resource Groups for Isolation and Layered Evolution**
  - Place common dependencies for teams/business lines at the resource group level. This naturally isolates the differentiated requirements of different teams/projects and avoids global pollution.
- **Make Minimal Additions at the Job/Notebook Level**
  - Place only dependencies that are unique to the job/Notebook, intended for short-term validation, or for small-scale personalization. When multiple jobs reuse the same dependencies, move them up to the resource group level for unified management.

Scenario-Based Recommendations

- **Multiple Teams Share the Same Engine.**

- The engine maintains the "enterprise common base."
- Place team-specific dependencies in their respective resource groups to avoid mutual interference.

- **Algorithm Team Rapid Experimentation**
  - The Notebook phase allows for rapid trial installations. Before release, solidify dependencies at the resource group level.

FAQs

- **Q: How to Handle Maven Transitive Dependency Conflicts?**
- A: The system automatically removes conflicting transitive dependencies and records the details. Please verify the impact scope in the installation logs and the "Removal Records".
- **Q: Why Do Installations/Uninstallations Not Take Effect Immediately?**
- A: Dependency changes take effect when the compute instance starts. Please restart the resource group or recreate the job/session according to the "Effective Actions".

DNS Domain Resolution

Last updated: 2026-05-20 16:11:31

Configure DNS services to enable the engine to securely and conveniently access external services via domain names.

Prerequisites

1. **The DNS service has been created.**
   1.1 The DNS service has been created and the corresponding domain names and resolution records (the mapping relationship between domain names and IP addresses) have been configured in the **Private DNS** console. For the operation guide, refer to Create a Private Zone.
   1.2 The custom DNS service is deployed in the Tencent Cloud VPC.
2. **Network connectivity has been configured.**
   In the **DLC console**, use the network connection configuration to enable the network where the engine resides to access the **VPC** where the target IP address is located.

Operation Steps

Step 1: Binding a Domain for Resolution

1. Log in to the DLC console and go to the target **Standard Engine Details page**.
2. On the details page, go to the **Domain Resolution Binding** settings tab.

Step 2: Selecting a Resolution Type

Domain resolution supports the following two types:

1. Private Domain Resolution

- When a user has already created a private domain resolution service in **Private DNS**, you can directly select this type.
- During the binding process, select the **VPC** corresponding to the private domain resolution service.
- After binding, the system will automatically synchronize the latest domain name and resolution configurations.

2. Custom Domain Resolution

- If you use a self-built DNS service or a third-party resolution solution, you can select **Custom Domain Resolution**.

- Manually specify the **VPC** where the DNS service resides and complete the binding.

Step 3: Verifying Access

1. After a successful binding, the engine will automatically load the resolution configuration.
2. By accessing the configured domain name within the engine, you can resolve it to the corresponding IP network, enabling access to external services.

Must-Knows

- Private domain resolution takes effect only within the same VPC or within VPCs that have established network connectivity.
- For custom domain resolution, ensure that the DNS service is available and has correct resolution records.

Resource Groups

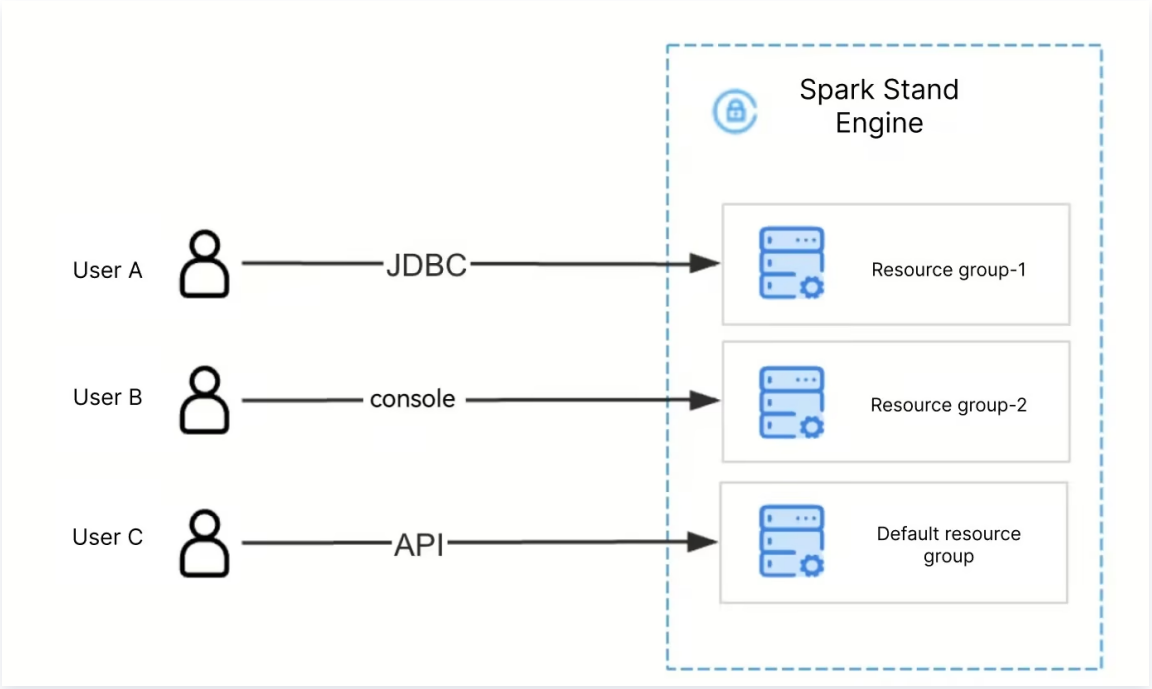
Last updated: 2026-05-20 16:11:51

A resource group is a secondary queue division of the computing resources for the Spark standard engine. It belongs to a parent standard engine, and resource groups under the same engine share resources with each other. The computing units (CUs) of the DLC Spark standard engine can be allocated to multiple resource groups as needed. For each resource group, you can set the minimum and maximum number of CUs that can be used, start-stop policies, concurrency, and dynamic/static parameters. This enables efficient management of computing resource isolation and workloads in complex scenarios such as multi-tenancy and multi-tasking.
For example, you can create a reporting resource group, a data warehouse resource group, and a historical backfill resource group for a Spark standard engine. You can set the minimum and maximum number of CUs for each resource group. In your business operations, you can then submit relevant SQL tasks or jobs, such as reports and data warehouse workloads, to their corresponding resource groups. This achieves resource isolation between different types of tasks and prevents individual large queries from monopolizing resources for extended periods.

Features

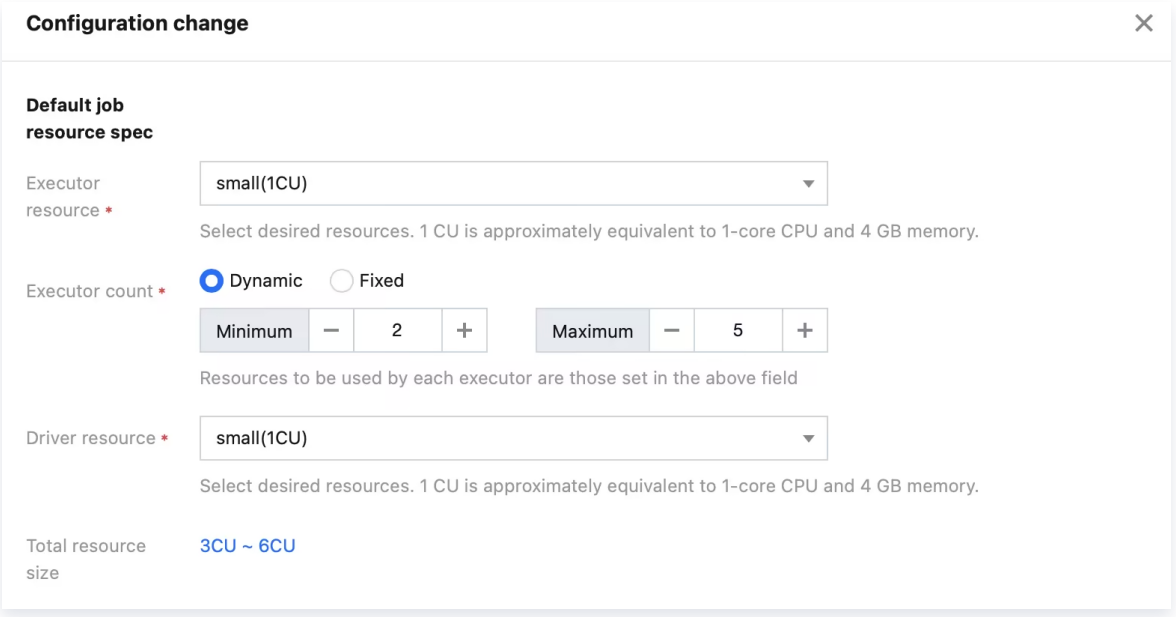
Resource Group Isolation

Using resource groups enables resource isolation for the Spark standard engine. You can assign corresponding resource groups to different users or queries. This achieves resource isolation and prevents a single user or large query from monopolizing most of the computing engine resources.

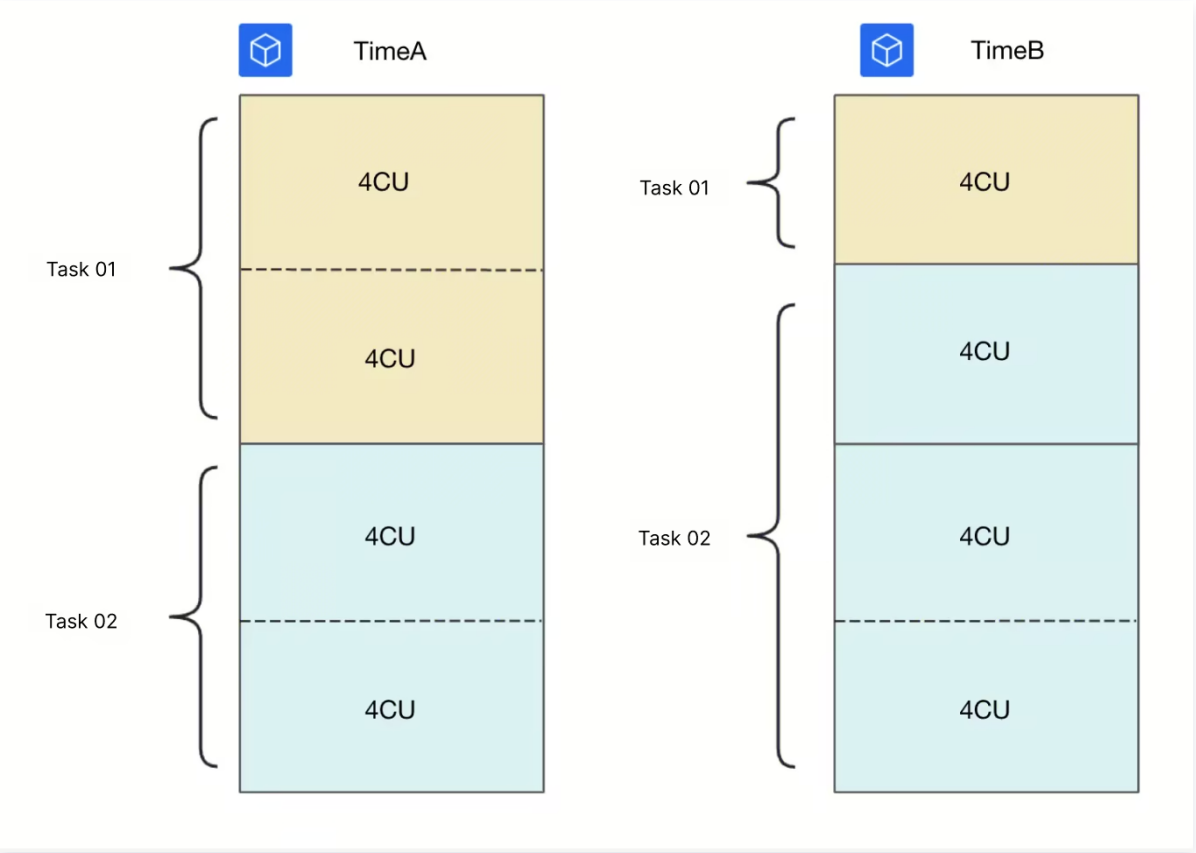


Resource Group Elasticity

Configure the number of Executors in a resource group for dynamic allocation. The resource group dynamically adjusts the resources occupied by SQL tasks or jobs based on the load, effectively improving resource utilization.
The dynamic allocation configuration is shown in the following figure:



Both Task 01 and Task 02 are configured for dynamic allocation. At TimeA, each task is allocated 8 CUs for execution. When execution reaches TimeB, Task 01 requires only 4 CUs. The released 4 CUs of idle resources are then allocated to Task 02. This process improves overall resource utilization. The scenario is illustrated in the following figure:



Resource group type

DLC currently supports three types of resource groups: SQL Analysis Only, Job, and Machine Learning. When you purchase a standard engine, DLC supports the default creation of "SQL Analysis Only" and "Job" type resource groups. It also supports your independent creation of "SQL Analysis Only" and "Machine Learning" type resource groups. The application scenarios and features of the three resource groups are described separately below:

- SQL Analysis Only resource group: It can be configured and used in modules such as Data Exploration, supporting SQL query and analysis scenarios.
- Job resource group: It is used for data job scenarios. When you create a data job and select an engine, the job resource group of that engine is used by default. The resource configurations for that job can be set during data job creation.
- Machine Learning resource group: It is used for scenarios involving AI model training using Python, ML machine learning frameworks, and PySpark.

Quick Configuration

The resource group resource provides a quick configuration option. Users can conveniently set the total specification of the resource group (unit: number of CUs). The backend then automatically allocates resources based on the policy.
The specific policies are as follows:

1. Dynamic allocation is enabled by default.
2. For a total specification of [4,8): the driver and executor use 2 CUs.
3. For a total specification of [8,64): the driver and executor use 4 CUs.
4. For a total specification of [64, ∞): the driver and executor use 8 CUs.

Monitoring Resource Groups

Resource group monitoring provides users with comprehensive insights into task and resource usage, helping optimize resource allocation and scheduling efficiency.

Resource Group Overview

1. Resource limit: If dynamic allocation is disabled for the resource group, this value represents the fixed resource capacity of the resource group. If dynamic allocation is enabled for the resource group, this value represents the Max resource set for the resource group.
2. Occupied resources: This metric obtains, in real time, the resources occupied by the Pods launched by the corresponding resource group.
3. Remaining available resources: The resources available to users in the current resource group, which is the resource limit minus the occupied resources.

Detailed Metrics for Resource Group Monitoring

- **Task Metrics (Task)**
  - Count the number of tasks in different statuses, including Canceled, Failed, Initializing, Running, Queued, and Succeeded.
  - Task duration metrics: Average initialization duration, maximum initialization duration, average queuing duration, and maximum queuing duration help analyze task startup and scheduling efficiency.
- **Resource Aggregation Metrics**
  - The total number of Executor Cores launched by the resource group reflects the scale of computing resources requested by the resource group.
  - The total number of actively used Executor Cores in the resource group shows the current amount of computing resources actually in use.
- **Compute Unit (CU) Metrics**
  - Resource group CU usage quantifies the compute unit resources consumed by the resource group.
  - CU utilization reflects the utilization efficiency of the resource group's computing resources.

> **Note:**
> Monitoring for machine learning resource groups currently supports only **Compute Unit (CU) metrics.**

Use Limits

Resource group names must be globally unique. It is recommended to use fully English names.

Related Terms

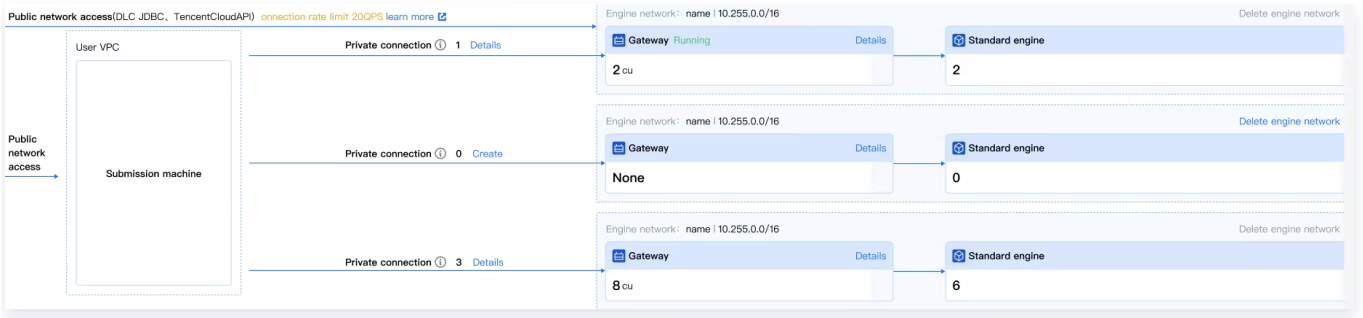
Term	Description
Default Resource Group (Created by the system by default)	SQL analysis only resource group: Exists upon engine creation and named default-rg-xxx. • This default resource group is initially suspended and configured with auto-start and auto-suspend. • This default resource group supports modifying its resource configuration. • This default resource group supports configuring start-stop policies, concurrency limits, and dynamic/static parameters. • This default resource group supports the dependency package management feature. • This default resource group cannot be deleted. Job resource group: Exists upon engine creation, does not support operations such as suspend, start, and restart, and is named default-job-rg-xxx. • This default resource group is initially in a ready state. Auto-start and auto-suspend cannot be configured. • This default resource group does not support modifying its resource configuration. Its configuration is set to the engine resource limit by default. • This default resource group does not support configuring start-stop policies or concurrency limits, but it does support configuring dynamic/static parameters. • This default resource group supports the dependency package management feature. • This default resource group cannot be deleted.
Custom Resource Group	• Custom resource groups support modifying their resource configuration.

(Manually created by users)	• Custom resource groups support configuring start-stop policies, concurrency limits, and dynamic/static parameters. • Custom resource groups can be deleted. • Job resource groups do not support custom operations or operations related to custom resource groups.

Private Link

Last updated: 2026-05-20 16:12:09

An endpoint is built on Private Link. If you need to access the engine and data through methods such as JDBC, you can create an endpoint to establish a secure and stable Private Link between your VPC and the access point.



Note:
The Presto engine is deprecated and only available for existing users.

Use Limits

1. The number of instances that can be created is limited to four.
2. For Private Link billing, see Private Link Billing.

Meson Engine

Last updated: 2026-05-20 16:12:26

The Meson engine is a high-performance vectorized query engine built into the DLC standard engine Spark. It transparently accelerates Spark SQL workloads and DataFrame API calls, reducing the total cost of workloads. Compared to open-source Spark, Meson delivers a 2.7x performance improvement in the TPC-DS 1TB benchmark. Meson is fully compatible with Apache Spark APIs and requires no changes to existing business code.

How It Works

With the widespread adoption of SSDs and significant improvements in NIC performance, the performance bottleneck of the Spark engine has shifted from the traditionally perceived I/O to compute resources, primarily CPU. However, CPU optimization solutions around the JVM, such as Codegen, have many constraints. For example, there are limitations on bytecode length and number of parameters, and developers also find it difficult to leverage some features of modern CPUs on the JVM.
The Meson engine transforms Spark physical plans, executes computations using a C++-implemented vectorized acceleration library, and returns the processed data in a columnar format. This improves memory and bandwidth utilization efficiency, thereby breaking through performance bottlenecks and effectively boosting Spark job efficiency.

Use Limits

The Meson engine currently has application scenario limitations. When a scenario is limited, the Meson engine performs a Fallback, reverting to the native Spark engine for execution. Because a Fallback requires necessary data conversion, excessive fallbacks may result in a longer overall runtime compared to the native Spark engine.
Please familiarize yourself with the main usage limitations of the Meson engine in advance:
It supports the Parquet data format. Support for ORC is currently incomplete, and other data formats are not currently supported.
ANSI mode is not currently supported.
Applications based on RDD are not currently supported.
Structured Streaming is not currently supported.
Custom Python code based on PySpark is not currently supported.
CacheTable with MEMORY_ONLY is not currently supported.

Applicable Scenarios

The following support capabilities are provided based on the Standard Engine Standard-S version 1.1 or later.

> **Note:**
> Storage formats, data types, operators, and functions that are not fully supported or unsupported by Meson are Fallbacked to the native Spark engine for execution.

Storage Formats

Data storage formats supported by the Meson engine:

- Supported data formats: Parquet, ORC
- Supported table formats: Iceberg, Hive

Data Types

Data types supported by the Meson engine:

- Byte,Short,Int,Long
- Boolean
- String,Binary
- Decimal
- Float,Double
- Date,Timestamp

Operators

Type	Supported Operators	Unsupported Operators
Source	FileSourceScanExec,HiveTableScanExec,BatchScanExec,InMemoryTableScanExec	-
Sink	DataWritingCommandExec,InsertIntoHiveTable,	-
Common	FilterExec,ProjectExec,SortExec,UnionExec	-
Aggregate	HashAggregateExec	SortAggregateExec,ObjectHashAggregateExec
Join	BroadcastHashJoinExec,ShuffledHashJoinExec,SortMergeJoinExec,BroadcastNestedLoopJoinExec,CartesianProductExec	-
Window	WindowExec	WindowGroupLimitExec
Exchange	ShuffleExchangeExec,ReusedExchangeExec,BroadcastExchangeExec,CoalesceExec	CustomShuffleReaderExec
Limit	GlobalLimitExec,LocalLimitExec,TakeOrderedAndProjectExec,CollectLimitExec	-
Subquery	SubqueryBroadcastExec	-
Other	ExpandExec,GenerateExec,CollectTailExec,RangeExec	RangeExec,SampleExec

Functions

Type	Supported Functions
Generator Functions	explode,explode_outer,inline,inline_outer,posexplode,posexplode_outer,stack
Window Functions	cume_dist,dense_rank,lag,lead,nth_value,ntile,percent_rank,rank,row_number
Aggregate Functions	any,any_value,approx_count_distinct,approx_percentile,array_agg,avg,bit_and,bit_or,bit_xor,bool_and,bool_or,collect_list,collect_set,corr,count,count_if,covar_pop,covar_samp,every,first,first_value,grouping,grouping_id,kurtosis,last,last_value,max,max_by,mean,median,min,min_by,percentile,percentile_approx,regr_avgx,regr_avgy,regr_count,regr_intercept,regr_r2,regr_slope,regr_sxx,regr_sxy,regr_syy,skewness,some,std,stddev,stddev_pop,stddev_samp,sum,try_avg,try_sum,var_pop,var_samp,variance
Array Functions	array,array_append,array_compact,array_contains,array_distinct,array_except,array_insert,array_intersect,array_join,array_max,array_min,array_position,array_prepend,array_remove,array_repeat,array_union,arrays_overlap,arrays_zip,flatten,get,shuffle,slice,sort_array
Bitwise Functions	&,^,bit_count,bit_get,getbit,shiftright,| ,~
Collection Functions	array_size,cardinality,concat,reverse,size
Conditional Functions	coalesce,if,ifnull,nanvl,nullif,nvl,nvl2,when
Conversion Functions	bigint,binary,boolean,cast,date,decimal,double,float,int,smallint,string,timestamp,tinyint
Date and Timestamp Functions	add_months,date_add,date_diff,date_format,date_from_unix_date,date_sub,date_trunc,dateadd,datediff,day,dayofmonth,dayofweek,dayofyear,extract,from_unixtime,from_utc_timestamp,hour,last_day,make_date,make_timestamp,make_ym_interval,minute,month,next_day,quarter,second,timestamp_micros,timestamp_millis,to_unix_timestamp,to_utc_timestamp,trunc,unix_date,unix_micros,unix_millis,unix_seconds,unix_timestamp,weekday,weekofyear,year
Hash Functions	crc32,hash,md5,sha,sha1,sha2,xxhash64
JSON Functions	from_json,get_json_object,json_array_length,json_object_keys,json_tuple,schema_of_json,to_json
Lambda Functions	aggregate,array_sort,exists,filter,forall,map_filter,map_zip_with,reduce,transform,transform_keys,transform_values,zip_with
Map Functions	element_at,map,map_concat,map_contains_key,map_entries,map_keys,map_values,str_to_map,try_element_at
Mathematical Functions	%,*,+,-,/,abs,acos,acosh,asin,asinh,atan,atan2,atanh,bin,cbrt,ceil,ceiling,conv,cos,cosh,cot,csc,degrees,e,exp,expm1,factorial,floor,greatest,hex,hypot,least,log,log10,log1p,log2,mod,negative,pi,pmod,positive,pow,power,rand,random,rint,round,sec,shiftleft,sign,signum,sinh,sqrt,try_add,unhex,width_bucket
Misc Functions	assert_true,equal_null,spark_partition_id,uuid,version,||
Predicate Functions	!,!=,<,<=,<=>,<>,=,==,>,>=,and,between,case,ilike,in,isnan,isnotnull,isnull,like,not,or,regexp,regexp_like
String Functions	ascii,base64,bit_length,btrim,char,char_length,character_length,chr,concat_ws,contains,endswith,find_in_set,format_number,format_string,initcap,instr,lcase,left,len,length,levenshtein,locate,lower,lpad,ltrim,luhn_check,mask,overlay,position,regexp_extract,regexp_extract_all,regexp_replace,repeat,replace,right,rpad,rtrim,soundex,split,split_part,startswith,substr,substring,substring_index,translate,trim,ucase,unbase64,upper
Struct Functions	named_struct,struct
URL Functions	url_decode,url_encode

Using the Meson Engine

The Standard Engine Standard-S 1.1 (native) supports the Meson engine by default.

> **Note:**
> When creating resources, ensure you select the correct engine version. Standard-S 1.1 does not support the Meson engine.

Configuring Network Connection

Last updated: 2026-05-20 16:12:44

DLC supports configuring networks (VPC) for data engines, facilitating the management of data engine access to different data source networks.

Network Configuration Types

DLC provides two network configuration types for different business scenarios.

- Enhanced network configuration: applies to scenarios requiring high-speed, stable access to data within a single VPC.

  > ⚠ **Note:**
  > 1. A data engine of a non-Spark job type can be bound to only one enhanced network configuration.
  > 2. If you use the enhanced network configuration, it will consume subnet IPs from your VPC. Ensure that sufficient subnet IPs are available.

- Cross-source network configuration: applies to cross-source federated queries that require access to data across multiple VPCs. A data engine supports binding to multiple cross-source network configurations.

Network Configuration Status

- Initializing: The network configuration is being initialized, and the network is not yet active.
- Success: The network configuration is applied to the bound engine.
- Failed: The network configuration failed. You can delete it and reconfigure.

Network Configuration Security Policy

If you have configured a security group policy for your VPC, you must add inbound rules for different network configuration types.

- Enhanced network configuration: You must add inbound rules for the CIDR block of the VPC where the data source resides to the security group.
- Cross-source network configuration: You must add inbound rules for the CIDR block of the engine bound to the network configuration to the security group.

Creating a Network Configuration

1. Log in to the DLC console and select the service region.
2. Go to Network Connection Configuration from the left sidebar.

3. Click the **Create Network Configuration** button to go to the configuration creation page.

Configuration parameters are as follows:

Configuration Content	Required	Description
Network Configuration Type	Yes	Select based on the application scenario: • Enhanced network configuration: applies to scenarios requiring high-speed, stable access to data within a single VPC. • Cross-source network configuration: applies to cross-source federated query and analysis scenarios that require access to data across multiple VPCs.
Configuration Name	Yes	It supports Chinese, English, and "_", with a maximum of 35 characters.
Instance Source	Yes	Supporting two sources: • DLC Data Catalog: You can select data catalogs that have been connected in DLC Data Management. • New network configuration: Select a new data source to create a network connection. Currently supported data sources include MySQL, Kafka, EMR HDFS (COS, HDFS, Chdfs), PostgreSQL, SQL Server, and ClickHouse. If the data source associated with the network configuration you need to create is not yet supported, select Other and manually specify a VPC.
Data Catalog	Yes	Select the corresponding data catalog based on the selected instance source. The range of available data catalogs depends on your account permissions.
Bind Data Engine.	Yes	Select the data engine associated with this network configuration. You cannot select a data engine if it is in an isolated or starting state.
Configuration Description	No	No more than 100 characters.

4. After completing the form, save it to create the network configuration.

> ⚠ **Note:**
> After creation, the network is in an initializing state. You can view its status in the list later.

Deleting a Network Configuration

You can manage and delete network configurations that are no longer needed or have failed to configure by performing a delete operation. The steps are as follows:

1. Log in to the DLC console and select the service region.
2. Go to Network Connection Configuration from the left sidebar.
3. Locate the network configuration you want to delete. You can filter and search for it. Note that you need to select the network configuration type.
4. Click the **Delete** button. After the action is confirmed, the deletion is completed.

> ⚠ **Note:**
> After deletion, this data engine will no longer be able to use this network configuration. If access is required, you need to reconfigure it. Please proceed with caution.

Modifying Description Information

You can modify the description of a configured network configuration. The steps are as follows:

1. Log in to the DLC console and select the service region.
2. Go to Network Connection Configuration from the left sidebar.
3. Locate the network configuration you want to delete. You can filter and search for it. Note that you need to select the network configuration type.
4. Click the **Modify Description** button to edit it.

Storage configuration
Archive Storage and Deep Archive Storage Overview

Last updated: 2026-05-20 16:17:04

In addition to supporting Standard Storage, DLC managed storage also supports Archive Storage and Deep Archive Storage for Iceberg native partitioned tables. Based on data access frequency, you can set the storage type of specific tables or partitions to Archive Storage or Deep Archive Storage. This enables data tiering between hot and cold data to reduce storage costs.

Note:
1. The supported regions are Beijing, Shanghai, Guangzhou, Nanjing, and Chengdu.
2. For details of billable items related to the archive storage and deep archive storage, see Billing Overview.

Introduction to Managed Storage Types

Storage Type	Description
STANDARD Storage	It is the default storage type for DLC managed storage and suitable for scenarios where data is frequently accessed and frequent read/write operations are performed.
Archive storage	It is suitable for data not requiring frequent access, effectively reducing storage costs. After archiving, the storage period cannot be less than 90 days.
Deep archive storage	It is suitable for data not requiring frequent access, effectively reducing storage costs. After archiving, the storage period cannot be less than 180 days.

Methods to Enable the Archive Storage or Deep Archive Storage

Note:
1. Prerequisite: You have been granted the ALTER TABLE permission. Otherwise, you cannot configure a storage policy.
2. Only partition tables can be configured with tiered storage policies.

3. You can configure up to 500 policies. Effective table-level policies consume the quota, while effective partition-level policies do not count against it. If the number of existing policies exceeds 500, delete existing policies before creating tiered storage policies.
4. After a table-level policy takes effect for a table, the policy applies to all partitions in the table.
5. If multiple policies are configured for a partition, the coldest policy takes effect.
6. Policies cannot be scheduled to take effect on a deferred date (T+1).

Method 1: Configuring an Archive Policy at the Table Level

1. Go to the Database tab page, select the database where the archive table resides, and click the database to go to the details page.
2. Click the database name in the upper left corner of the page to go to the tiered storage policy configuration page, click **Create Policy**, and select partition tables in batches under the database to configure the archive policy.

Method 2: Configuring an Archive Policy by Partition Field

Note:

1. If you configure an archive policy for a partition, the policy takes effect on the current snapshot. When the snapshot data is archived during the configuration, data written later will not be archived. In this case, change the configuration to archive newly generated data.
2. If an archive policy is being created, you need to wait until it takes effect before creating a new one.
3. If you find that Location is not displayed in partition information when configuring an archive policy for a partition, click the refresh button to recollect partition information.
4. If a table-level archive policy is configured, no more archive policy is required for partition fields.

1. Go to the Database tab page, click the **partition table** for which you want to configure a policy, and go to the **Tiered Storage** page.
2. Click **Create Archive** and select multiple partition fields in the table to configure an archive policy.

Data Restoration from the Archive Storage and Deep Archive Storage

Note:

1. The restoration operation cannot be scheduled to take effect on a deferred date (T+1).
2. After data is restored from the archive storage, you will be charged for the storage of object copies based on the standard storage. For data restored from the deep archive storage, you will be charged for the retrieval request of object copies based on the labeled unit price. For details, see the product pricing documentation.

1. Go to the Database tab page, select the database where an archived table resides, click the database to go to the details page, click the **name of the partition table** you want to restore, and switch to the **Tiered Storage** page.
2. In the operation column on the right side, click Restore. If you need to restore multiple archives in batches, select them, and click Batch Restore above.
3. In the pop-up window, configure the validity period of copies generated after restoration. You need to configure the period (at least 1 day) after which the copies will expire and be deleted automatically. If you want to modify the valid period in days of the copies after restoration, click Restore again and change the validity period in the pop-up window.
4. If the status changes to **Restoring**, the operation succeeds.
5. You can view relevant information about the restoration operations from **View Restoration Record**.

Configuring a Managed Bucket

Last updated: 2026-05-20 16:19:19

Managed storage refers to the storage space hosted by users on the data lake product, with its underlying storage being COS. Native tables, user program packages, query results, and other data are stored in managed storage. Therefore, to use capabilities such as native tables and their data optimization, you must first enable managed storage. Native tables in managed storage are Iceberg format tables by default, and you do not need to manage the underlying file content. For details on managed storage billing, see Billing Overview.

Note:

- In addition to billing for user data usage, the managed bucket also incurs minimal fees for requests from daily storage routine inspections (including user storage activity level and amount inspections) performed by Data Lake Compute (DLC). To avoid fees from these routine inspections, promptly terminate any managed bucket that is no longer in use.
- DLC supports creating metadata acceleration buckets and regular buckets. For regular buckets, object versioning is enabled by default, and historical versions are retained for 30 days by default.

This document describes the steps for enabling and configuring managed storage.

Enabling Managed Storage

Step 1: Go to the Managed Storage Configuration Page

You can log in to the DLC console, click the Storage Management module in the left-side menu bar, and go to the managed storage configuration.

Step 2: Creating a Managed Bucket

1. Click Create Managed Bucket to go to the creation page, select the managed bucket type, and click Confirm to complete the creation.

   Here, you can specify the managed bucket type as a metadata acceleration bucket or a regular bucket. Metadata acceleration buckets and regular buckets have the same billing method. However, Metadata acceleration buckets require additional configurations to grant engine access permissions. For details, refer to Binding of the Metadata Acceleration Bucket.
2. You can modify the query result storage path above. The query result path is used to temporarily store data such as SQL query results and Spark job Shuffle data. You must

specify a path to ensure the normal operation of jobs and tasks. If you have already created a managed bucket, it is recommended that you configure the query result path as **Managed Storage**. Alternatively, you can configure the query result path to a COS bucket path under your own account.

Viewing Managed Buckets

After you enable managed storage, a bucket is created. You can view the buckets and data in managed storage in the Storage Management module.

Deleting Managed Storage

Destroying data is a high-risk operation. You can destroy managed storage only after all database and table data has been deleted. Destroying managed storage requires administrator permissions.

Step 1: Deleting Database Table Data

To destroy managed storage, you must first delete all database and table data in the managed storage.
You can refer to the documents Data Catalog and Database Management and Data Table Management to delete database and table data. Alternatively, you can run the DROP statement in the Data Exploration module to delete the data.

Step 2: Deleting Managed Storage

After deleting the database and table data, you can go to the Managed Storage Configuration tab under the Storage Management module to destroy the managed storage.
Destroying managed storage deletes all DLC managed buckets. Proceed with caution.

Tags Associated with a Managed Bucket

Tags are used to categorize and organize resources. A Tag consists of a Tag key and a Tag value, and a single Tag key can correspond to multiple values. Users can create and bind Tags to cloud resources to manage them. DLC supports binding Tags to managed buckets in the **Managed Bucket List on the Product Console** to implement billing breakdown for managed buckets through Tags.

Creating Tags and Allocating Costs by Tag

You can categorize and centrally manage storage resources by creating and binding Tags to managed buckets. You can plan Tags for resources (at the managed bucket level) based on organizational or business dimensions, such as department, project group, or region, to implement cost allocation management.

> **Note:**
> Managed bucket Tags currently do not support resource isolation and authorization. They only support the Tag-based cost allocation feature.

Operation Steps:

1. Create Tags. Log in to the Tag Console to create Tags. For detailed steps, refer to Create Tags and Bind Resources.
2. Bind resources. Bind Tags in the Tag Console, the DLC console > Storage Management > Managed Bucket List page.
3. Set a cost allocation tag: Go to the Billing Center to set a cost allocation tag. For details, see Cost Allocation Tags.
4. Go to the Billing Overview page. Select the By Tag Summary tab to view the bar chart and list of related resources aggregated by Tag Key.

> **Note:**
> Currently, Tags can only be bound to managed buckets from the Managed Bucket List. Binding Tags during the creation of a new managed bucket is not yet supported.

Binding of the Metadata Acceleration Bucket

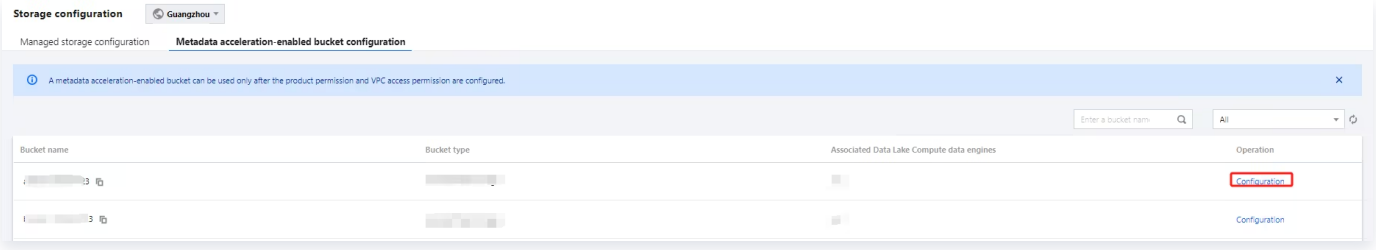
Last updated: 2024-01-10 16:30:47

Data Lake Compute supports binding to fusion buckets to accelerate query analysis performance. To use this feature, you need to create a metadata acceleration bucket. Data Lake Compute's managed storage provides a metadata acceleration bucket, using the object storage bucket under the user's account. For more details, please refer to Object Storage > Metadata Acceleration.
When the engine accesses the Data Lake Compute metadata acceleration bucket, it needs to bind permissions. The process for binding permissions is as follows.

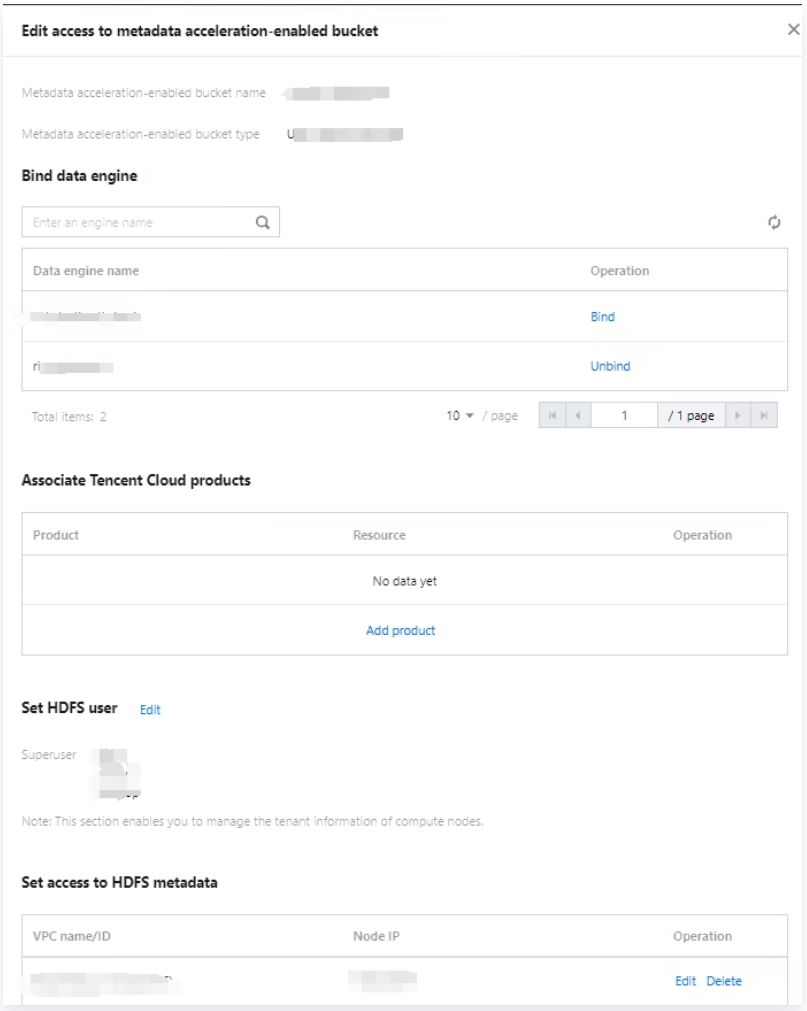
Binding Data Engine and Metadata Acceleration Bucket

1. Log in to the Data Lake Compute Console and navigate to Public Management > Storage Configuration.
2. Navigate to the **Metadata Acceleration Bucket Configuration Page**, select the bucket you need to bind, and click **Configure**.

> **Note:**
> Only metadata acceleration buckets are displayed on the Metadata Acceleration Bucket page. Regular buckets (buckets without metadata acceleration enabled) are not displayed here.

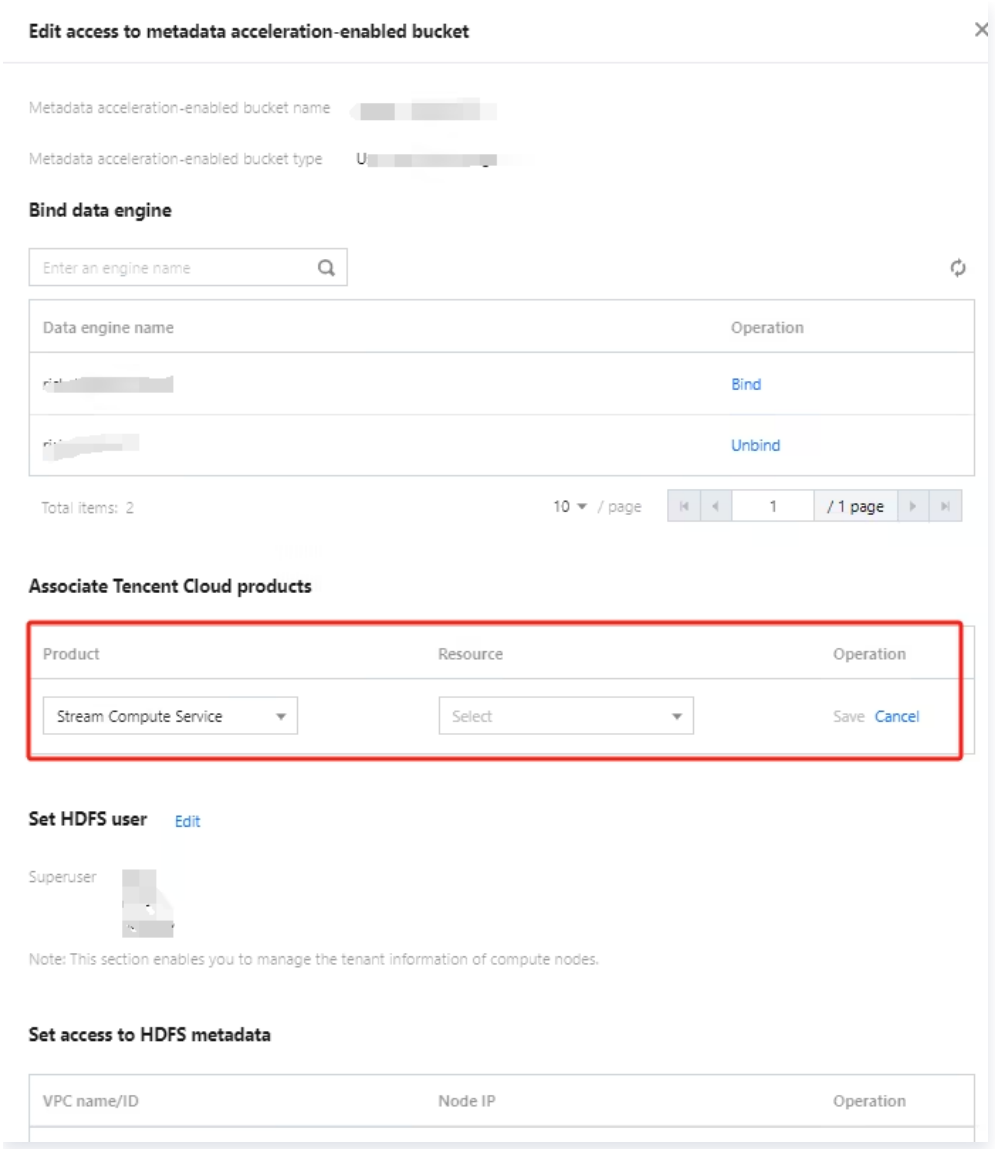


3. Click **Bind** to associate the data engine that needs to access this bucket with the metadata acceleration bucket.



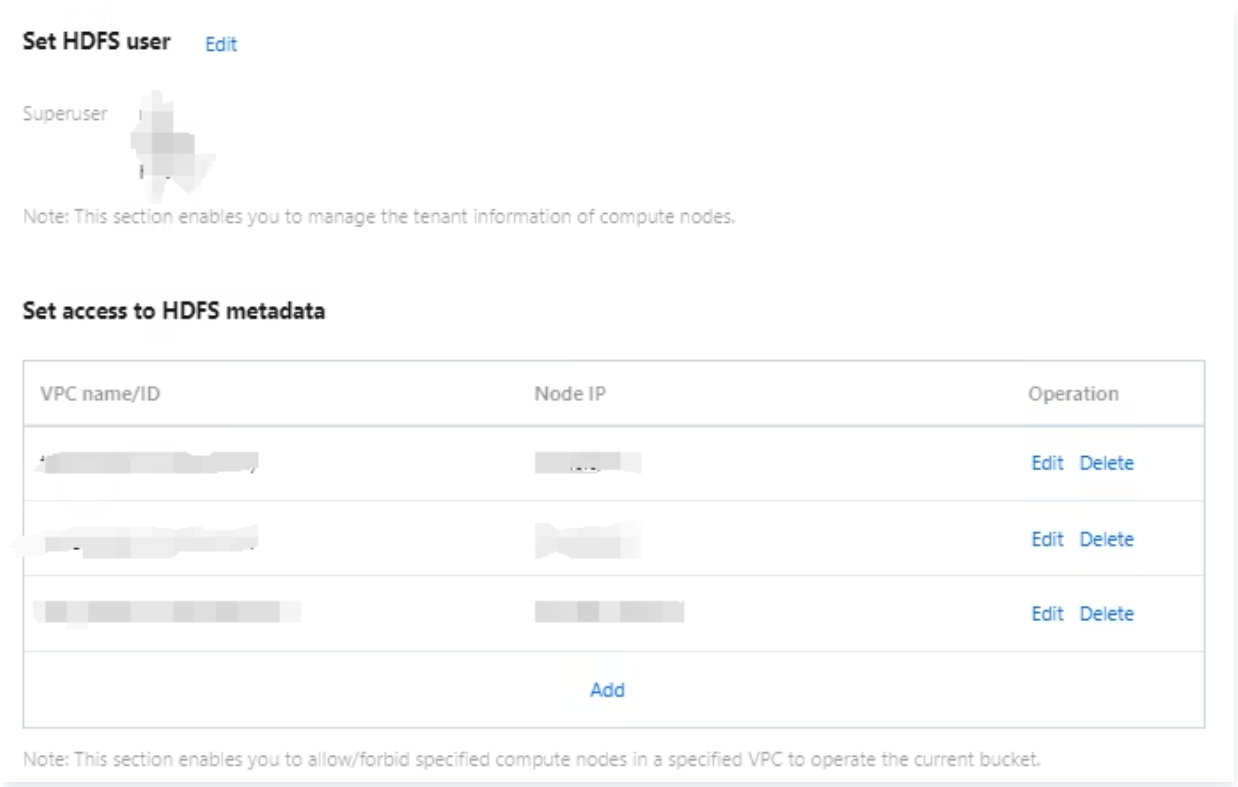
Bind the computational resources of Stream Computing Oceanus.

If you are using Stream Computing Oceanus to stream data into the data lake, and the storage being written to is a metadata acceleration bucket, you will need to configure access permissions to the metadata acceleration bucket under Storage Configuration. Under the Tencent Cloud Product Binding column, create a new product, select Stream Computing Oceanus and the corresponding resources, then click save.



Binding Compute Resources of Non-Data Lake Compute Data Engines

At times, the computational resources you need to access the metadata acceleration bucket are not from the DLC's compute engine. In such instances, you can configure access permissions to the metadata acceleration bucket under Storage Configuration.
HDFS User Configuration is used to set up the superuser for your computational resources accessing the DLC, typically root/hadoop/presto/flink.
HDFS Metadata Permission Configuration is used to set up the VPC network environment you permit to access the DLC, typically the VPC where the aforementioned non-DLC data engine's computational resources are located.



Data Management
Managing Data Catalogs and Databases

Last updated: 2026-05-20 16:20:15

External data and managed storage data in DLC can be managed through the Data Management page, by executing standard SQL statements, or via APIs. From the console's Data Management page, you can create and edit data catalogs, as well as create, query, and delete database tables.

Creating a data catalog

> **Note:**
> The platform creates a DataLakeCatalog for you by default for lake data management. When you have an external data source that you want to perform federated analysis on, you can refer to the following procedure to create a data catalog for the external data source.

1. Log in to the DLC console and select a service region. The account you use to log in must have permissions to create catalogs. To enable permissions for a sub-account, refer to Sub-account Permission Management.
2. Go to Metadata Management and click **Create Data Catalog**.
3. Go to the data source creation page.
   After entering the connection information, complete the network configuration to establish connectivity between the engine and the external data source.
4. Enter the data source information and click **Confirm** to create the data source.
5. In the data catalog list, you can view the connection information, status, creator, and other details.

Editing a Data Catalog

1. In the **data catalog list**, click the **Edit** button on the right side of a data catalog. You can modify data catalog information, including the description, network configuration, username, password, advanced settings, and so on.
2. After modifications, click **Confirm** to create the data catalog again.

Rolling Back the Data Catalog Configuration

To ensure security and operational flexibility, DLC supports rolling back the custom configuration of a data catalog. You can roll back the configuration to quickly restore the data catalog to the previous version.

Creating a Database

1. Log in to the DLC console and select a service region. The account you use to log in must have permissions to create databases.
2. Go to the Metadata Management page, click the name of a data catalog on the Catalog tab page, and view databases in the catalog.
3. Click **Create Database** to go to the database creation page.
4. After entering the database information and saving it, you can complete the database creation.
   - Database name: It must be globally unique, support uppercase and lowercase letters, digits, and "_", cannot start with a digit, and can contain a maximum of 128 characters.
   - Description: It supports Chinese and English and can contain a maximum of 2048 characters.
   - A root account can create a maximum of 100 databases.

Viewing the Database

1. Log in to the DLC console and select a service region. The account you use to log in must have database query permissions.
2. Go to Metadata Management > **Database**, choose a data catalog, and click the **database name** to access the database details. You can manage the data tables in the database. For detailed instructions, see Data Table Management.

Deleting a Database

1. Log in to the DLC console and select a service region. The account you use to log in must have database deletion permissions.
2. Go to Metadata Management, click **Delete**, and confirm the deletion to delete the database.

Data Table Management

Last updated: 2026-05-20 16:20:35

Users can use the DLC console or API to execute DDL statements to create databases.

Creating Data Table

Method 1: Creating a Data Table in Data Exploration

1. Log in to the DLC console and select the region where the service is located. The logged-in user must have the permission to create data tables.
2. Go to the Data Exploration module. In the list on the left, click an existing database, hover over a table row, then click the ••• icon. Click **Create Native Table** or **Create External Table**.

> **Note:**
> - A native table refers to a table on DLC managed storage. You do not need to focus on the underlying Iceberg storage format for native tables, and they have capabilities such as data optimization. To use native tables, you must first enable managed storage. For details, see Managed Storage Configuration.
> - The underlying data of an external table resides in your own COS. To create an external table, you must specify the data path.

3. After you click **Create Native Table/Create External Table**, the system automatically generates an SQL template for creating a data table. You can modify the SQL template to create the data table. After you click the **Run** button, the SQL statement for creating the data table is executed to complete the creation.

Method 2: Creating a Data Table in Data Management

The Data Management module supports managing native tables and external tables that are stored on DLC managed storage.

1. Log in to the DLC console and select the region where the service is located. The logged-in user must have the permission to create data tables.
2. Go to Metadata Management via the left sidebar, enter **Database**, click the name of the database where the data table resides, and go to the DMC page.
3. Click the **Create Native Table** or **Create External Table** button to go to the Data Table Configuration page.

4. Native tables support three different data source types: empty tables, local uploads, and COS. Selecting a different data source corresponds to a different creation process. Native tables provide capabilities such as data optimization. You can choose to inherit library governance rules or enable/disable them individually.
   4.1 Create an empty table: Create an empty table that contains no records.
   - Data table name: It cannot start with a digit, can contain uppercase and lowercase letters, digits, and underscores, and can contain a maximum of 128 characters.
   - Supports entering description information for the data table.
   - You can manually add and enter column names and field types. It supports the configuration of three complex field types: array/map/struct.

   4.2 Local upload: Upload a local form file to DLC to create a data table. It supports files smaller than 100 MB.
   - CSV: It supports visually configuring CSV parsing rules, including compression format, column delimiter, and field delimiter. It also supports automatically inferring the Schema of data files and parsing the first row as column names.
   - Json: DLC identifies only the first level of Json as columns. It supports automatically inferring the Schema of Json files, and the system identifies the first-level Json fields as column names.
   - It supports data files in common big data formats such as Parquet, ORC, and AVRO.
   - You can manually add and enter column names and field types.
   - If you select automatic structure inference, DLC automatically populates the identified columns, column names, and field types. If they are incorrect, modify them manually.

   4.3 Create a data table through COS.
   - Create a data table by reading the COS data bucket under the current account.
   - CSV: It supports visually configuring CSV parsing rules, including compression format, column delimiter, and field delimiter. It also supports automatically inferring the Schema of data files and parsing the first row as column names.
   - Json: DLC identifies only the first level of Json as columns. It supports automatically inferring the Schema of Json files, and the system identifies the first-level Json fields as column names.
   - It supports data files in common big data formats such as Parquet, ORC, and AVRO.
   - You can manually add and enter column names and field types.
   - If you select automatic structure inference, DLC automatically populates the identified columns, column names, and field types. If they are incorrect, modify them manually.
5. Data partitioning is typically performed on tables with large data volumes to improve query performance. DLC supports querying data based on partitions, and users need to add partition information at this step. By partitioning your data, you can limit the amount of data

scanned per query, thereby improving query performance and reducing usage costs. DLC adheres to Apache Hive's partitioning rules.

- A partition column corresponds to a subdirectory under the COS path of the table. The naming rule for the directory is **partition_column_name=partition_column_value**. Example:

> **Note:**
> The example code is for reference only and should be modified based on the actual business scenario. For example, replace "bucket_name" with your bucket name.

```
cosn://nanjin-bucket/CSV/year=2021/month=10/day=10/demo1.csv
cosn://nanjin-bucket/CSV/year=2021/month=10/day=11/demo2.csv
```

- If there are multiple partition columns, you must nest them sequentially according to the order specified in the CREATE TABLE statement.

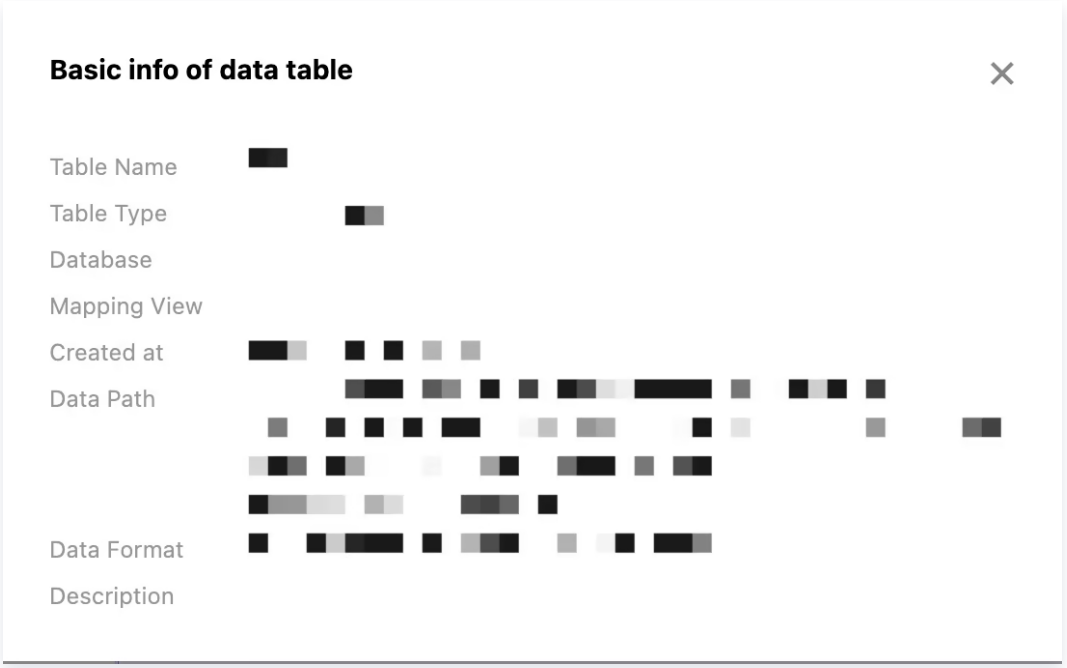
```
CREATE EXTERNAL TABLE IF NOT EXISTS
`COSDataCatalog`.`dlc_demo`.`table_demo` (
    `_c0` string,
    `_c1` string,
    `_c2` string,
    `_c3` string
) PARTITIONED BY (`year` string, `month` string, `day` string)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES ('separatorChar' = ',', 'quoteChar' = '"')
STORED AS TEXTFILE
LOCATION 'cosn://bucket_name/folder_name/';
```

Querying Basic Information of a Data Table

Method 1: Querying in Data Exploration

In the Data Table project, hover the pointer over the **Data Table Name** row, click the ••• icon, select **Basic Info** from the drop-down menu, and you can view the basic information of the created data table.

The basic information of the data table is as follows:



Method 2: Viewing in Data Management

1. Log in to the DLC console, select the region where the service is located. The logged-in user must have the permission to view data tables.
2. Go to the Metadata Management module via the left-side menu. On the **Database** page, click the **database name** where the data table resides to enter the Database Management page. This page supports querying information such as the number of rows, storage space, creator, fields, and partitions of the data table.

Self-Service Querying for Data Table Partition Information

> **Note:**
> Replace the database name and table name in the example below according to your actual business scenario.

- SuperSQL Spark SQL Engine:

```
select * from `DataLakeCatalog`.`db`.`tb$partitions`
```

- SuperSQL Job Engine and Standard Engine:

```
select * from `DataLakeCatalog`.`db`.`tb`.`partitions`
```

Querying Data Table Partition Information

Data table management supports querying partition information related to data tables. With the partition information, users can view details, including record quantity, file quantity, data storage capacity, and update time for each partition of the table.

1. Log in to the DLC console and select the region where the service is located. The logged-in user must have the permission to view data tables.
2. Go to Metadata Management via the left sidebar, enter the **Database**, and click the name of the database where the data table resides to access the **Database Management** page.
3. Select **Database** to go to the **Data Table** management page. Select and click the **Data Table**, and then select **Partition Information** to go to the **Partition Information** page.

The data partition page displays partition information of the table in paginated form. You can query partition details through sorting partitions by fields, including names, record quantity, file data size, and file storage. For example, to view a certain fixed partition, enter the partition name to search.

> **Note:**
> 1. Partition information statistics are currently only available for DLC native tables.
> 2. Partition information statistics are currently in the Beta testing phase. To enable the partition information statistics, you may contact us.

Note:

In the data table project, hover the pointer over the **data table name** row, click the ••• icon, and then click **Preview Data** in the drop-down menu. DLC automatically generates an SQL statement to preview 10 data records and executes it to query the first 10 rows of the data table.
The data preview feature displays the first 100 data records by default.

Note:

You can edit the description information of a data table in the Data Management module.

1. Log in to the DLC console and select the region where the service is located. The logged-in user must have the permission to edit data tables.
2. Go to the Metadata Management > **Database** page via the left sidebar, click the name of the database where the data table resides, and go to the DMC page.
3. Locate the data to be edited, and click the **Edit** button on the right to edit it.
4. After making modifications, click the **OK** button to complete the editing.

Delete Data Table

Method 1: Deleting in Data Exploration

In the data table project, hover the pointer over the **data table name** row, click the ••• icon, and then click **Delete Table** in the drop-down menu. DLC automatically generates an SQL statement to delete the data table and executes it to delete the table.

- Deleting an external table or a data table only removes the metadata information stored in DLC and does not affect the data source files.
- Deleting a data table under the DataLakeCatalog directory will clear all data of that table. Please proceed with caution.

Method 2: Deleting in Data Management

Currently, Data Management only supports managing database tables that are hosted and stored in DLC. For external tables, please use Method 1 to delete them.

1. Log in to the DLC console and select the region where the service is located. The logged-in user must have the permission to delete data tables.
2. Go to Metadata Management > **Database** via the left sidebar, click the **database name** where the data table resides, and go to the DMC page.
3. Click the **Delete** button to the right of the data table you want to delete. After a secondary confirmation, the corresponding data table can be deleted, and its data will be cleared simultaneously.

Displaying the CREATE TABLE Statement

In the data table project, hover the pointer over the **data table name** row, click the ••• icon, and then click **Show CREATE TABLE Statement** in the drop-down menu. DLC automatically generates an SQL statement to view the CREATE TABLE statement for that data table and executes it to query the statement.

System Constraints

- DLC allows a maximum of 4,096 data tables per database. Each data table supports up to 100,000 partitions, and each data table can have a maximum of 4,096 attribute columns.
- DLC identifies data files under the same COS path as belonging to the same table. Please ensure that data for individual tables is kept in separate folder hierarchies.
- DLC does not support COS multi-version data and can only query the latest version of data in a COS bucket.
- All tables created on DLC are external tables. The SQL statement for creating a table must include the EXTERNAL keyword.
- Table names must be unique within the same database.

- A table name is case-insensitive, can only contain English letters, numbers, and underscores (_), and has a maximum length of 128 characters.
- If a table is a partitioned table, you must manually execute an ADD PARTITION or MSCK statement to add partition information before you can query the data in that partition. For details, see Querying Partitioned Tables.
- When you create a table using a CSV file, DLC converts all field types to string by default. This does not affect the calculation and querying of the original data fields.

Data View Management

Last updated: 2026-05-20 16:20:50

DLC provides data view query capabilities. Users can query and use data conveniently and quickly by managing data views.

Creating a View

1. Log in to the DLC console and select the service region. The user who logs in must have view creation permissions.
2. Go to Data Exploration. You can create a view using SQL statements. For statement details, see SQL Syntax.
3. Select computing resources. After you click **Run**, the view creation is completed.

Viewing the View

You can view views using SQL statements in Data Exploration. For specific syntax, see SQL Syntax. DLC also provides a visual interface for managing views. The operations are as follows.

1. Log in to the DLC console and select the service region. The user who logs in must have view query permissions.
2. Go to Metadata Management. Click the **database name** where the view resides to access the Database Management page.
3. Click **View** to go to view management.
4. Click the **view name** you want to view. You can then view the view information and copy the SQL statement.

Deleting a View

You can view views using SQL statements in Data Exploration. For specific syntax, see SQL Syntax. DLC also provides a visual interface for managing views. The operations are as follows.

1. Log in to the DLC console and select the service region. The user who logs in must have view deletion permissions.
2. Go to the Metadata Management page. Click the **database name** where the view resides to access the Database Management page.
3. Click **View** to go to view management, and click **Delete** on the right to delete the view.

> ⚠ **Note:**

Deleting a view clears all data under the view, and the data cannot be recovered. Proceed with caution.

Managing Functions

Last updated: 2026-05-20 16:21:05

DLC supports using **user-defined functions** to process and build data, and it also supports managing these functions.

Creating a Function

1. Log in to the DLC console and select the service region. The account used for login must have database operation permissions.
2. Method 1: Go to the Metadata Management page, switch to the **Database** page, click the **database name** for which you want to build the function, and then switch to the **Function** page.

   Method 2: Go to Metadata Management and switch to the **Function** page.
3. Click the **Create Function** button to go to the function creation menu.
   - The function package supports uploading files locally or using existing JAR and Python files from COS. For local uploads, JAR files are supported up to a maximum of 5 MB, and Python files are supported up to a maximum of 2 MB.
   - Registering a Python UDF makes it globally effective. The configuration entry is: go to the **Metadata Management** page, **switch to the Function page, and click Create**. For the creation and management process, refer to the UDF Function Development Guide.
   - You must select a spark cluster to run the function. No charges will be incurred during the runtime.
   - It is recommended to save the function package to the system for easier management and use. Mounting it to a specified COS path is also supported.

Viewing Function Information

1. Log in to the DLC console. The account used for login must have database operation permissions.
2. Go to Metadata Management >**Database** and click the **database name** that contains the function you want to view. Alternatively, go to Metadata Management, switch to the **Function** tab, and view all functions globally.
3. Select a function to view its build status. If the build fails, you can click the **Edit** button on the right and submit it again.
4. Click the **function name** to directly view its details.

Editing Function Information

1. Log in to the DLC console and select the service region. The account used for login must have database operation permissions.
2. Go to the Metadata Management page and click the **database name** of the function you want to view.
3. Select a **function** and click the **Edit** button to go to the function information editing page.
   - You cannot modify the function name, storage method, or upload method for now. If you need to modify such information, you can create a new function.
   - After you modify the function information, the system will rebuild it. Please proceed with caution.

Deleting a Function

For functions that no longer need to be used or managed, you can delete them.

1. In the DLC console, select the service region. The account used for login must have database operation permissions.
2. Go to Metadata Management and click the **database name** of the function you want to view.
3. Select a **function** and click the **Delete** button to delete functions that are no longer needed.

> ⚠ **Note:**
> After deletion, the data under this function will be cleared and cannot be recovered. Please proceed with caution.

Partition Field Policy

Last updated: 2026-05-20 16:21:22

In Hive, partition information is presented in the form of directories. In Iceberg, partition information is recorded in the final underlying data files. This makes Iceberg partitions more flexible and allows the partitioning policy to evolve as data volume changes. In DLC, you can create Iceberg tables to use features such as hidden partitions.

> **Note:**
> Native tables are Iceberg tables by default. External tables can be created as Hive tables or Iceberg tables based on the file format. For detailed syntax, refer to the CREATE TABLE documentation.

With hidden partitioning, you do not need to specify partition information additionally, as is required in Hive, when inserting or querying data.
The Iceberg partitioning policy supports the use of the following functions. The different fields and their corresponding partition transformation policies are listed in the table below:

Partitioning Policies	Field Type	Result Type
identity	Any	source type
bucket	int, long, decimal, date, time, timestamp, timestamptz, string, uuid, fixed, binary	int
truncate	int, long, decimal, string	source type
year	date, timestamp, timestamptz	int
month	date, timestamp, timestamptz	int
day	date, timestamp, timestamptz	int
hour	timestamp, timestamptz	int

Data Recycle Bin

Last updated: 2026-05-20 16:21:39

The data recycle bin feature provides temporary storage for deleted Iceberg native tables under the DataLakeCatalog data catalog, to retain accidentally deleted data. Once this feature is enabled, all deleted data files will be stored into the recycle bin, instead of being permanently deleted. Administrators can perform recovery and deletion management on these data tables, while regular users can recover data tables depending on their permissions. This effectively reduces the risk of data loss and improves the efficiency of data management. You can enable the data recycle bin feature for your DataLakeCatalog data catalog through the metadata management feature in DLC.

Note:

1. Once the feature is enabled, files in the data recycle bin will still incur storage fees. When the files are permanently deleted from the recycle bin, no charges will be applied. For details, see Storage Fees.
2. The retention duration for the files in the data recycle bin can be configured to 7 days, 15 days, or 30 days according to the use cases.

Enabling the Data Recycle Bin Feature

Note:

1. Administrator confirmation is required to enable the feature.
2. Once the feature is enabled, any deleted data tables will be retained in the recycle bin by default, and storage fees will be charged.

1. Log in to the DLC console.
2. In the left sidebar, select Metadata Management. In the data catalogs, find the DataLakeCatalog data catalog.
3. Click More in the operation column of this catalog. Select Data Recycle Bin, confirm and click the Enable Recycle Bin button.
4. Configure the retention duration for the files in the recycle bin.

Recovering Tables in the Data Recycle Bin

Note:

1. Administrators can recover all data tables in the recycle bin, while common users can recover data tables for which they have edit permissions on the tables' parent databases.
2. The recovery operation is only supported for a single table, and batch operations are not supported.
3. When another table with the same name already exists in the database to which the table to be recovered belongs, you need to change the name of the existing duplicate table before the recovery.
4. If the database to which the table belongs is deleted, the table cannot be recovered.

1. Log in to the DLC console.
2. In the left sidebar, select Metadata Management. In the data catalogs, find the DataLakeCatalog data catalog.
3. Click More in the operation column of this catalog and select Data Recycle Bin.
4. Select the data table to be recovered from the list and click Recover in the right operation column.

Deleting Tables in the Data Recycle Bin

⚠ Note:

1. Only administrators are allowed to delete the tables in the recycle bin.
2. Once a table in the recycle bin is deleted, it cannot be recovered. Confirm before proceeding.
3. If the dwell time of a data table in the recycle bin exceeds the retention duration, it will be automatically deleted.
4. Table deletion is supported for a single table and batch operations.

1. Log in to the DLC console.
2. In the left sidebar, select Metadata Management. In the data catalogs, find the DataLakeCatalog data catalog.
3. Click More in the operation column of this catalog and select Data Recycle Bin.
4. Select the data table to be deleted from the list and click the Delete button in the right operation column. You can also select multiple tables and click the Batch Delete button in the top-left corner.

Managing the Data Recycle Bin Feature

⚠ Note:

Only administrators are allowed to manage the data recycle bin feature.

Setting Retention Duration for Files in the Data Recycle Bin

Note:
The retention duration change takes effect immediately. If the duration is set shorter than the original duration, the existing tables in the recycle bin will be permanently deleted immediately. For example, if the retention duration is changed from 15 days to 7 days, tables older than 7 days will be deleted immediately. Please confirm before proceeding.

1. Log in to the DLC console.
2. In the left sidebar, select Metadata Management. In the data catalogs, find the DataLakeCatalog data catalog.
3. Click More in the operation column of this catalog and select Data Recycle Bin.
4. Click the Recycle Bin Configurations button and set the retention duration for tables in the recycle bin.

Disabling the Data Recycle Bin Feature

Note:
1. Before disabling the feature, ensure all tables in the recycle bin are removed.
2. Once the feature is disabled, the data table deletion action will delete the table immediately and permanently without retention and recovery.

1. Log in to the DLC console.
2. In the left sidebar, select Metadata Management. In the data catalogs, find the DataLakeCatalog data catalog.
3. Click More in the operation column of this catalog and select Data Recycle Bin.
4. Click the Recycle Bin Configurations button to disable the feature.

Data Masking

Last updated: 2026-05-20 16:22:05

Overview

DLC supports column-level DMask. You can use the DMask feature to associate masking rules with columns containing sensitive data and configure a series of masking algorithms for different user groups, enabling fine-grained, role-based DMask application. For example, for phone number data, you might want to grant full access to your customer service group, grant permission to view only the last four digits to your analytics group, and apply strict DMask permissions that display as NULL to your finance group.

Restrictions and Limitations

The current restrictions and limitations of the DLC data masking feature are as follows.

Supported Engine Types

Currently, only the following engine types with a kernel version later than October 1, 2024, are supported: Spark standard engine, SuperSQL Spark job engine, and SuperSQL SparkSQL engine. Among these, data desensitization for the SuperSQL SparkSQL engine is an allowlist feature. If you need to use it, submit a ticket for activation. For information on DLC engine categories, refer to Data Engine Introduction.

Applicable Scope

1. The DMask feature takes effect only for **all DLC users** (including administrators). For DLC user types, see DLC Permission Overview.
2. The configuration for a masking policy tag takes about 1 minute to take effect.
3. Only effective for the metadata under the default data catalog, DataLakeCatalog.

Special Restrictions

1. When the SuperSQL SparkSQL engine is used, views are not supported by masking policies.
2. Users or working groups should have the SELECT permission on databases and tables. Otherwise, queries by the user or working group on that table will return an error. For DLC permission types, refer to Sub-Account Permission Management.

Supported Masking Methods

Currently, DLC supports the following data masking methods for column values:

Default Value

Returns a default masking value for the column based on the column's data type. Use this rule when you want to hide the value of the column but display the data type.

- **Supported data types:** STRING, BINARY, INT, BIGINT, DOUBLE, FLOAT, DECIMAL, BOOLEAN, TIMESTAMP, DATE and ARRAY

Data Type	Default Masking Value
STRING	""
BINARY	[]
INT	0
DECIMAL	0
BIGINT	0
FLOAT	0
DOUBLE	0
BOOLEAN	false
TIMESTAMP	1970-01-01 08:00:00
DATE	1970-01-01
ARRAY	[]

Retaining the First 4 Characters

Returns the first 4 characters of the column's value, replacing the rest of the string with XXXXX. If the column's value is equal to or less than 4 characters in length, return the column's value after the value has been run through the SHA-256 hash function. You can only use this rule with columns that use the STRING data type.

- **Supported data type:** STRING

Example Value	Masking Value
abcd@example.com	abcdxxxxxxxxxx
abc	ba7816bf8f01cfea414140de5dae2223b00361a396177a9cb410ff61f20015ad

Retaining the Last Four Characters

Returns the last 4 characters of the column's value, replacing the rest of the string with XXXXX. If the column's value is equal to or less than 4 characters in length, then return the column's value after the value has been run through the SHA-256 hash function. You can only use this rule with columns that use the STRING data type.

- **Supported data type:** STRING

Example Value	Masking Value
abcd@example.com	xxxxxxxxxx.com
abc	ba7816bf8f01cfea414140de5dae2223b00361a396177a9cb410ff61f20015ad

Hashing

Returns the column's value after the value has been run through the SHA-256 hash function. Use this rule when you want the end user to be able to use this column in a JOIN operation for a query. You can only use this rule with columns that use the STRING or BYTES data types.

- **Supported data types:** STRING and BYTES

Example Value	Masking Value
abcd@example.com	3f7768839b5bcba43f589cc3af54efaea18bceb1df8b05a7dffaec3e7b43b269

Setting as NULL

Returns NULL regardless of the column's data type. Use this rule if you want to hide both the column value and its data type.

- **Supported data types:** Unlimited

Example Value	Masking Value
abcd@example.com	NULL

Date Masking

Displays only the year part of a date string and defaults the month and date to 01/01. You can only use this rule with columns that use the DATE and TIMESTAMP data types.

- **Supported data types:** TIMESTAMP and DATE

Example Value	Masking Value

2015-03-05T09:32:05.359	2015-01-01 00:00:00

No Masking

Displays the plaintext of the column value without any masking.

- **Supported data types:** All

Masking Method Selection Recommendations

You can flexibly select the masking method for the column value based on the following recommendations:

Masking Method	Recommended Scenario
Default	Column value is expected to be hidden but data type needs to be displayed to users.
Retaining the first 4 characters	Plaintext of the column value is expected to be hidden but but part of the characters for information confirmation needs to be displayed, for example, the first 4 characters of the customer's email address used by the customer service personnel for confirmation.
Retaining the last 4 characters.	Plaintext of the column value is expected to be hidden but part of the characters for information confirmation needs to be displayed, for example, the last 4 characters of the customer's mobile number used by the customer service personnel for confirmation.
Hashing	Use this rule if you want users to be able to use this column in JOIN operations of queries or perform GROUP BY statistics.
Setting as NULL	Use this rule if you want to hide the column value and its data type.
Date masking	This rule is used for the scenario where only the year part is displayed and the rest of the date information is hidden, for example, the year of birth for confirmation.
No masking	Recommended for users who need to use the plaintext

Data Masking Workflow

You can configure DLC data masking in the steps below:

Step 1: Creating a Masking Policy Tag

Overview of Masking Policy Tags

A masking policy tag refers to a tag customized by users and associated with columns containing sensitive data. In a masking policy tag, you can configure refined masking methods for multiple user groups. The following example shows the procedure to set a masking policy tag named "mobile number" and set the masking methods for 3 DLC workgroups:

- **Name of masking policy tag:** Mobile number masking
- **Working Group Masking Method Configuration:**

Job group	Masking Method
Customer service personnel group	No masking
Analysis personnel group	Displaying the last 4 characters
Finance personnel group	Setting as NULL

> **Effect:**
> After associating the above masking policy tag with two columns, such as Phone_number1 and Phone_number2, DLC users in the customer service personnel group get the plaintext when querying by Phone_number1 and Phone_number2, DLC users in the analysis personnel group get the result displaying as *********4320, and DLC users in the finance personnel group get the result displaying as NULL.

Masking Policy Execution Priority

Assuming a DLC user is added to multiple working groups, and different masking methods are associated with these different working groups, multiple masking policies may exist for a specific user. If conflicting masking methods exist for a user, the system will **apply the method with the highest priority among the working groups of the user, where working groups higher in the list have a higher priority.**
For example, in the masking policy tag above, if Zhang San belongs to both the analyst personnel group and the finance personnel group, the query for Zhang San will display data desensitized as, for example, *********4320.

Step 1: Creating a Masking Policy Tag

Creating a Masking Policy Tag

1. Go to the DLC console, choose **Metadata Management > DMask**, click **Create Masking Policy Tag**, and create a new masking policy tag in the dialog box that appears:
2. In Configure Masking Method, select the corresponding working groups to add them to the Selected dialog box, and configure a masking method for each working group.
3. After configuration, you can drag and drop working group modules to sort them from top to bottom, which determines the policy priority. Working groups positioned higher in the list have a higher priority. If no manual sorting is performed, policies will be applied based on the default top-down order.
4. Click **OK** to complete the creation.

> **Note:**
> 1. The DLC data desensitization feature currently only takes effect for users within user groups that have been configured with a masking method. It does not take effect for **user groups without a configured masking method**, or for **users not added to any user group**.
> 2. To apply desensitization to all DLC users, it is recommended to:
>
> 2.1 Configure a masking method for all user groups.
>
> 2.2 Add all DLC users to a user group.

Step 2: Binding the Tag to a Data Column

1. Go to the DLC console, select Metadata Management, and then select the data table in the data catalog that contains the field requiring desensitization.
2. Locate the field requiring desensitization, or search for the field name in the top-right search box. Then, in the Masking Policy Tag column for that field, click , and select the desired masking policy tag for binding from the dialog box.
3. If no matching masking policy tag is found, click **Create Masking Policy Tag** in the dialog box to quickly create one.

> **Note:**
> 1. The various built-in DLC masking methods have field type restrictions. For example, date desensitization can only be associated with columns of TIMESTAMP or DATE data types. For details, refer to Supported Masking Methods.
> 2. A field can only be associated with one masking policy tag. However, **a single masking policy tag can be reused across multiple data columns.**

Effective time:
After you configure a masking policy tag and bind it to a column, the configuration takes about 1 minute to take effect. During this period, users may still get plaintext in a query. Just wait a moment for the configuration to take effect.

Step 3: Executing a Query

After you bind a masking policy tag to a column requiring desensitization, the configuration is successful if the column's data type supports all masking methods within that tag. When relevant users query data from that column, the results are displayed with the corresponding desensitization based on the masking method of the user group to which the user belongs. The following virtual case further demonstrates the effects achievable with the DLC DMask feature.

Case Description

Assume that company A has a customer list customer_list containing sensitive information, with detailed fields as follows:

Mobile Number	Customer Level	Consumption Amount	Email Address
123456789	High	45,600	abc@example.com
234567891	Medium	15,000	bcd@example.com
345678912	Low	2,000	cde@example.com
456789123	Low	1,000	def@example.com

There are 3 user groups in company A, including the customer service personnel group, the finance personnel group, and the analysis personnel group. Currently, company A hopes that the Mobile Number and Email Address fields containing users' sensitive PII information can only be used by the customer service personnel group, the Consumption Amount field and the Customer Level tag can only be used by the finance personnel group, but the analysis personnel can view the hashed value of the Customer Level to conduct statistical analysis of customer hierarchy.
Based on the above requirements, create the following 3 masking policy tags:

Contact Information Desensitization (Bind the policy tag to the Mobile Number and Email	Consumption Amount Desensitization (Bind the policy tag to the Consumption Amount	Customer Level Desensitization (Bind the policy tag to the Customer Level field)

Address fields)	field)	
Customer service personnel group: No masking Finance personnel group: NULL Analysis personnel group: NULL	Customer service personnel group: NULL Finance personnel group: No masking Analysis personnel group: NULL	Customer service personnel group: NULL Finance personnel group: No masking Analysis personnel group: Hashing

Assuming that the masking tags are bound to the corresponding columns according to the table above:

1. For a user existing only within a specific working group, running SELECT * FROM customer_list returns the following results:

- **Customer service personnel group:** This group has been granted a rule of no desensitization for contact information. The following results will be returned:

Mobile Number	Customer Level	Consumption Amount	Email Address
123456789	NULL	NULL	abc@example.com
234567891	NULL	NULL	bcd@example.com
345678912	NULL	NULL	cde@example.com
456789123	NULL	NULL	def@example.com

- **Finance personnel group:** This group has been granted a rule of no desensitization for consumption amount and customer level. The following results will be returned:

Mobile Number	Customer Level	Consumption Amount	Email Address
NULL	High	45,600	NULL
NULL	Medium	15,000	NULL
NULL	Low	2,000	NULL
NULL	Low	1,000	NULL

- **Analyst personnel group:** This group has been assigned hash desensitization for customer level. The following results will be returned. While the actual customer level is obscured, statistical analysis can still be performed using the hash value:

Mobile Number	Customer Level	Consumption Amount	Email Address
NULL	4fa3c0d004d0750fc7bf8631993bd7c668fd33f8d089e0103ad8ef3fc1d9f4bb	45,600	NULL
NULL	35d8f8d59e2630de970e35271547d087278074addd61ce31940da69d82d19929	15,000	NULL
NULL	49542bc83b9d59935686144f352b6acb2264992720d0dbe780be50b56b87fef7	2,000	NULL
NULL	49542bc83b9d59935686144f352b6acb2264992720d0dbe780be50b56b87fef7	1,000	NULL

2. If a user exists in both the **customer service personnel group** and the **finance personnel group**, according to the top-down priority of masking rules, the effective masking rules for this user are as follows:

Contact Information Desensitization (Bind the policy tag to the Mobile Number and Email Address fields)	**Consumption Amount Desensitization** (Bind the policy tag to the Consumption Amount field)	**Customer Level Desensitization** (Bind the policy tag to the Customer Level field)
Customer service personnel group: No masking	Customer service personnel group: NULL	Customer service personnel group: NULL

Running SELECT * FROM customer_list returns the following results:

Mobile Number	Customer Level	Consumption Amount	Email Address
123456789	NULL	NULL	abc@example.com
234567891	NULL	NULL	bcd@example.com
345678912	NULL	NULL	cde@example.com
456789123	NULL	NULL	def@example.com

Maintenance
Historical Task Instances

Last updated: 2026-05-20 16:22:39

The Historical Task Instances feature focuses on recording and managing various tasks that users execute in DLC, facilitating subsequent tracking, review, and optimization. Using this feature, users can quickly view task execution details, including start and end times, execution status (such as success or failure), input and output specifics, and generated logs or error messages. It provides users with the convenience of auditing and search, helping them identify task health status, potential issues, and optimize resource allocation.

Operation Steps

1. Log in to DLC Console > Ops Management > Historical Task Instances and select the service region.
2. Go to the Historical Task Instances page. Administrators can view all historical execution tasks from the past 45 days, while common users can query tasks related to themselves from the past 45 days.
3. You can filter and view tasks by task type, execution status, creator, task time range, task name, task ID, task content, sub-channel, and other criteria.
4. Click the task ID/name to view the task details, which include modules such as Basic Information, Run Results, Task Insights, and Task Logs.
5. Users can click to modify task configurations and quickly go to the job details page to adjust configurations for optimization.

Historical Task Instances

Note:
*Fields are supported only after the Insights feature is enabled (statistics are available only after the task is completed). For how to enable it, see How to Enable the Insights Feature.

Field Name	Description
Task ID	Unique identifier of the task.
Task name	prefix_yyyymmddhhmmss_8-digit uuid, where yyyymmddhhmmss is the task execution time. Prefix rule:

	1. For a job task submitted from the console, the task ID prefix is the job name. For example, if a user creates a job named `customer_segmentation_job` and executes it at 2024.11.26 21:25:10, the task ID is `customer_segmentation_job_20241126212510_f2a65wk1`. According to the current data format limitation, the job name must be <= 100 characters. 2. For SQL types submitted from the Data Exploration page, the prefix is `sql_query`. Example: `sql_query_20241126212510_f2a65wk1`. 3. For data optimization tasks, the prefix varies depending on the specific sub-type of the optimization task: 3.1 The optimizer prefix is only `optimizer`. 3.2 The SQL type for an optimized instance is `optimizer_sql`. 3.3 The batch type for an optimized instance is `optimizer_batch`. 3.4 The configuration task created when a data optimization policy is configured is `optimizer_config`. 4. For a data import task, the prefix is `import`. Example: `import_20241126212510_f2a65wk1`. 5. For a data export task, the prefix is `export`. Example: `export_20241126212510_f2a65wk1`. 6. For a Wedata submission, the prefix is `wd`. Example: `wd_20241126212510_f2a65wk1`. 7. For submissions from other APIs, the prefix is `customized`. Example: `customized_20241126212510_f2a65wk1`. 8. For tasks created by performing operations on metadata from the Metadata Management page, the prefix is `metadata`. Example: `metadata_20241126212510_f2a65wk1`.
Task Status	• Starting • Executing • Queued • Successful. • Failed. • Canceled • Expired • Task Timeout
Task content.	Detailed content of the task. For job-type tasks, it is a hyperlink to the job details; for SQL-type tasks, it is the complete SQL statement.
Task type	Job type and SQL type.

Task Sources	The source from which the task is generated. Supported types include data exploration tasks, data job tasks, data optimization tasks, import tasks, export tasks, metadata management, Wedata tasks, and API-submitted tasks.
Sub-channel	Users can customize the sub-channel when submitting tasks via the API.
Computing Resource	The compute engine/resource group used to run the task.
* Accumulated CPU Hours (Consumed CU*Hours)	Specifies the sum of the CPU execution duration of each core of the Spark Executor used in computing, per hour (not equivalent to the duration of starting machines in the cluster, because the machines may not participate in task computing after they start. Eventually, the cluster's CU fee is subject to the bill). In the Spark scenario, it is approximately equal to the sum of the execution durations of the Spark task (seconds) / 3600 (per hour). (This metric can only be calculated after the task is completed.)
Total Execution Duration	Time from the start to the end of a task, which may include the waiting time caused by resource shortages. 1. For a SparkSQL task, it is the platform scheduling time + queuing time within the engine + execution duration within the engine. 2. For a Spark job task, it is the platform scheduling time + engine startup duration + queuing time within the engine + execution duration within the engine.
* Engine Execution Duration	If the task has an insight result, it is the execution time within the engine, reflecting the time actually required for computing, which is the duration from the start of the first Task to the completion of the Spark job. Specifically, it calculates the sum of the duration from the start of the first task to the completion of the last task for each Spark Stage. This sum excludes the queuing time before the task starts (that is, it removes other time, including the time required for scheduling between task submission and the start of execution of the Spark task), and also excludes the time spent waiting for task execution due to insufficient executor resources between multiple Spark stages during the task execution process. (This metric can only be calculated after the task is completed.)
* Data Scan Volume	The amount of physical data read from storage by this task. In the Spark scenario, it is approximately equal to the sum of Stage Input Size in Spark UI.
* Data Scan	The number of physical data records read from storage by this task. In the Spark scenario, it is approximately equal to the sum of Stage Input

Number of Records	Records in Spark UI.
Creator.	If the task is of the job type, it is the creator of that job.
Executor	The user who runs the task.
Submission Time	The time when the user submits the task.
* Engine Execution Time	The time when the task first preempts the CPU to start execution, which is the time when the first Task starts execution within the Spark engine. (This metric can only be calculated after the task is completed.)
* Number of Output Files	The collection of this metric requires upgrading the Spark engine kernel to a version after November 16, 2024. The total number of files written by the task through INSERT and other statements. (This metric can only be calculated after the task is completed.)
* Number of Small Output Files	The collection of this metric requires upgrading the Spark engine kernel to a version after November 16, 2024. Small file definition: an output file is defined as a small file if its size is < 4 MB (controlled by the parameter spark.dlc.monitorFileSizeThreshold, which defaults to 4 MB and can be configured at either the engine global or task level). This metric definition: the total number of small files written by the task through INSERT and other statements. (This metric can only be calculated after the task is completed.)
* Total Output Rows	The number of records output after this task processes data. In the Spark scenario, it is approximately equal to the sum of Stage Output Records in Spark UI.
* Total Output Size	The size of the records output after this task processes data. In the Spark scenario, it is approximately equal to the sum of Stage Output Size in Spark UI.
* Shuffle Read Records	In the Spark scenario, it approximately equals to the sum of the Stage Shuffle Read Records in Spark UI. (This metric can only be calculated after the task is completed.)
* Shuffle Read Data Size	In the Spark scenario, it is approximately equal to the sum of Stage Shuffle Read Size in Spark UI. (This metric can only be calculated after the task is completed.)

* Shuffle Write Data Size	In the Spark scenario, it is approximately equal to the sum of Stage Shuffle Write size in Spark UI.
* Current core Usage	SQL tasks (Standard Engine Resource Group / SuperSQL SQL Engine): count the number of cores currently used by Executors (excluding Driver cores). Job tasks: count the total number of cores currently in use (including Driver cores).
* Health Status	Analyze the task to judge the health status of the task and determine whether optimization is required. For details, see Task Insight. (This metric can only be calculated after the task is completed.)

Historical Task Instance Details

Basic Info

1. Users can view specific task details in **Execution Content**. For SQL tasks, you can view the complete SQL statement. For job tasks, you can view job details and parameters.
2. Users can view task resource details in **Resource Consumption**, including consumed CU*hours, total execution duration, engine execution duration, data scanned, compute resources, kernel version, Driver resources, Executor resources, and the number of Executors.
3. Users can view basic task information in **Basic Information**, including the task name, task ID, task type, task source, creator, executor, submission time, and engine execution time.
4. For tasks run by the SuperSQL SparkSQL or SuperSQL Presto engine, users can view the task execution progress bar in **Query Statistics**, which includes the time spent on stages such as task creation, task scheduling, task execution, and result retrieval.

Execution result

After a task is completed, you can query the task results on the Run Results page. There are two types of task results:

1. File Write Information: For file write tasks run by the SuperSQL or Standard Engine Spark kernel engine, users can view the file write information.
   - Average File Size
   - Minimum File Size
   - Maximum File Size
   - File Size

2. Execution Result: For SQL task queries, the query results of the current task can be displayed, and users are supported in downloading the query results.

Task Insights

After a task is completed, you can view the task insight result on the Task Insights page. It supports analyzing aggregated metrics from each task execution and identifying potential optimization issues. Based on the actual execution of the current task, DLC Task Insights provides corresponding optimization suggestions by combining data analysis and algorithmic rules. For details, see Task Insights.

Task Logs

You can view logs of the current task on the task log page.

> **Note:**
> - Only the job type and BatchSQL type support task log viewing.
> - The SQL type tasks run on permanent clusters, so logs cannot be displayed at the task level.
> - Optimization-related tasks are internal optimization tasks and do not require displaying cluster logs.

1. You can switch logs on different cluster nodes by Pod name, for example, logs of Driver, Executor, and so on.
2. You can switch log types, including All, Log4j, Stdout, and Stderr.

> **Note:**
> - Spark engine images upgraded on or after July 4, 2025, support the display of Log4j, Stdout, and Stderr logs individually.
> - For historic version images of the Spark engine:
>   - When "All" or "Stderr" is selected, all logs will be displayed.
>   - When "Log4j" or "Stdout" is selected, no content will be displayed (the output will be empty).
> - If you want to upgrade the engine, Submit a ticket to contact the after-sales for upgrade.

- **All**: Displays all log content of a task for comprehensive troubleshooting.
- **Log4j**: Displays logs generated by the Spark cluster itself, helping you understand the cluster running status and internal information.

- **Stdout**: Displays service logs, usually containing normal output information of user programs.
- **Stderr**: Displays standard error logs, helping quickly capture exceptions and error information.

3. The following three log levels are supported for filtering: All, Error, and WARN.
4. This page only displays the latest 1,000 logs. To view all logs, export them.
5. Log export history and status of exported tasks can be viewed. In log export history, you can save log files locally.

Resource Usage Statistics

The real-time fluctuation chart provides an intuitive display of resource consumption status during task execution. The chart automatically updates every minute and reports the latest data (due to collection frequency limitations, no fluctuation chart is displayed for tasks running less than 5 seconds). This helps you to dynamically monitor resource usage trends.

> **Note:**
> The feature is only supported in Spark engine versions released on July 1, 2025, or later.

- SQL tasks: Statistics on the real-time usage of Executor cores (excluding Driver cores) are collected, accurately reflecting the computing resources consumption of SQL tasks.
- Batch tasks: Statistics on all startup cores (including Driver cores) are collected, providing a comprehensive display of resource usage for batch tasks.

History (Legacy)

Last updated: 2026-05-20 16:24:54

DLC provides three methods for querying historical task records to facilitate user access.

Viewing Run History Tasks in the Query Editor

1. Log in to the DLC console and select the service region.
2. Go to the **Data Exploration page**. Within a single Session, click **Execution History** to view the task execution history for that Session.
3. Click the **Batch ID** in the history record to view the corresponding execution result on the left .
   - The execution history for each Session is isolated and is retained for a maximum of 45 days.
   - Historical task result data is retained for 24 hours. To view task results older than 24 hours, you must rerun the task.

Viewing Data Import History in the Data Management Feature

1. Log in to DLC Console > Data Management and select the service region.

> ⚠ **Note:**
> To log in to an account, you need database-related permissions.

2. Click **Task History** in the upper-right corner to query data import historical tasks.
3. You can view historical tasks from the past 45 days.

Viewing Historical Tasks in the Run History Feature

> ⓘ **Note:**
> The history record was upgraded to the new edition of historical task instances on March 20, 2025. Users can switch back to the old version within 90 days. For details about the new edition, see Historical Task Instances .

1. Log in to DLC Console > Ops Management > Historical Task Instances and select the service region. Click the button in the upper-right corner to return to the old version.
2. Go to the Historical Execution page. Administrators can view all historical execution tasks from the past 45 days, while common users can query tasks related to themselves from

the past 45 days.

3. You can filter and view tasks by task type, execution status, creator, data type, and other criteria.

4. Click **View Details** to view task execution details and results.

   - Historical task result data is retained for 24 hours. To view task results older than 24 hours, you must rerun the task. You can directly **copy the statement** to Data Exploration to run the task.
   - You can directly click **Task ID** to quickly switch to and view task execution details.
   - You can cancel a running task by clicking **Cancel**.

Session Management

Last updated: 2026-05-20 16:25:09

The session management feature records and tracks interactive notebook sessions that are submitted to the DLC engine via APIs or Wedata. Users can use these sessions to perform operations such as SQL queries, data processing, and model training.

Prerequisites

Prepare the DLC environment.

- Activate the DLC engine service.
- To create a session, you must purchase a job-type engine.
  - SuperSQL Job Engine.
  - Standard Spark Engine or Machine Learning Resource Group.

Operation Steps

1. Log in to DLC Console > Ops Management > Session Management and choose the service region.
2. Go to the Session Management page to view all historical session records.
3. You can filter and view data by engine type, status, category, engine name, resource group name, session ID, and Spark application name.
4. Click **Spark Application Name/ID** to view the Spark application details.
5. Users can close a session by clicking the terminate operation in the console.
6. Users can view the Spark UI of the application on the Spark Application Details page.

Session List

Field Name	Description

Session ID	Unique identifier for the session.
	• A session created by the SuperSQL Job Engine has only a Session ID. Session ID rule: livy-session-uuid.
	• A session created by the Standard Spark Engine:
	  ○ The Notebook submitted by the user is automatically generated by the platform.
	  ○ The batch SQL submitted by the user is automatically generated by the platform.

Spark Application Name/ID	In the SuperSQL scenario, only the Spark application ID is displayed. The prefix for the Spark application ID: spark.
	Sessions created by the standard Spark engine:
	• Notebook submitted by the user:
	  ○ Spark Application Name: the Spark application name entered by the user when the user creates a kernel; the default prefix is sparkapp.
	  ○ Spark Application ID: The prefix is spark. It is the unique identifier for the Spark application.
	• Batch SQL submitted by the user:
	  ○ Spark Application Name: The prefix is temporary-rg.
	  ○ Spark Application ID: The prefix is spark. It is the unique identifier for the Spark application.

Status	Status of the current session, which can be categorized into • Not Started (not_started): The session has not been started. This status indicates that the session request has been received, but the session has not begun due to certain reasons (such as resource shortages or configuration issues). Users need to check the relevant configuration or resource status to start the session. • Starting (starting): The session is being started. This status indicates that Livy is allocating resources and initializing the environment for the new Spark session. • Idle (idle): The session has been successfully started and is in an idle state. At this point, you can submit Spark jobs, and the Livy session is ready to process requests. • Busy (busy): The session is processing one or more jobs. This status indicates that the session is executing tasks and cannot accept new job requests until the current job is completed. • Shutting Down (shutting_down): The session is being shut down. This status indicates that the user has requested to stop the session, and Livy is performing cleanup and resource release operations. The session may remain in this state for a period of time until all running jobs are completed and resources are released. • Error (error): The session encountered an error during startup or execution. This status typically indicates that the session cannot function properly, which may be caused by resource shortages, configuration errors, or other issues. • Dead (dead): The session is dead and cannot be recovered. • Killed (killed): The session has been forcibly terminated. This status indicates that the user has actively terminated the session,

	possibly because the session is no longer needed or there is an issue with the running job. A killed session cannot be recovered. • Success (success): The session has been successfully completed. This status is typically used to indicate that all jobs in the session have been successfully executed and completed. In this state, the session can be considered successful, and users can view the results or output.
Engine	The mounted compute engine.
Resource group	Standard Spark engine scenario: the resource group used when a compute engine is mounted. When users submit batch SQL, temporary resources are used by default, and the resource group is displayed as -- by default. SuperSQL scenario: the resource group information column is not displayed.
Category	Session type: • Spark • PySpark • SQL • Machine Learning • Python • MLlib
Creator.	The user who created the session.
Start time	The time when the session was started.
End time.	The time when the session ended or was terminated.
Operation	Supports termination operations, allowing sessions to be manually closed.

insight management
Insight Management Overview

Last updated: 2026-05-20 16:25:43

DLC provides you with agile and efficient Serverless data lake analysis and computing services. As a distributed computing platform, the query performance of DLC is affected by multiple internal and external factors, such as: engine CU scale, the number of concurrently submitted and queued tasks, SQL writing patterns, and Spark parameter configurations. DLC insight management offers a visual and intuitive interface to help you quickly understand current query performance and the potential factors affecting it, and to obtain performance optimization recommendations.
DLC provides the insight management feature, which includes **task insight, engine usage insight, and intelligent storage** features, to help users better adjust resources or optimize task logic.
Applicable Business Scenarios:

1. **There is a need for insight into the overall runtime status of the Spark engine**. For example: metrics such as resource contention during task execution under the engine, resource usage within the engine, engine execution duration, data scan size, and data shuffle size are all displayed and analyzed intuitively.
2. **There is a need for convenient self-service troubleshooting and analysis of task runtime status**. For example: you can filter and sort numerous tasks by duration to quickly identify problematic large tasks, and pinpoint the causes of slow or failed Spark tasks, such as resource contention, shuffle exceptions, or insufficient disk space, all with clear identification.
3. **There is a need for insight into table storage distribution**. For example: it enables observation of storage distribution, rankings, and usage trends, intelligent diagnosis of risks, and helps identify tables that require storage optimization.

Task Insights

Last updated: 2026-05-20 16:26:04

Task Insights, based on the task perspective, helps users quickly locate optimization analysis and recommendations for completed tasks.

Prerequisites

1. SuperSQL SparkSQL and Spark Job Engine:
   1.1 Task Insights is enabled by default for newly purchased engines after July 18, 2024.
   1.2 For Spark kernel versions released before July 18, 2024, you must upgrade the engine kernel to enable Task Insights. For the upgrade procedure, see How to Enable the Insight Feature below.
2. Standard Spark Engine:
   2.1 Engines purchased after December 20, 2024, support Task Insights by default.
   2.2 For engines purchased before December 20, 2024, Task Insights cannot be manually enabled by users. To enable it, please submit a ticket to contact after-sales support.

Task Insights is not currently supported on other engine types.

Operation Steps

Log in to the DLC console, select the insight management feature, and then switch to the Task Insights page.

Task Insights

Daily statistics on the distribution and trends of tasks identified for optimization provide a more intuitive understanding of daily tasks.

Task Insights

The Task Insights feature supports analyzing aggregated metrics from each executed task and identifying optimizable issues.
After a task is completed, you only need to confirm the task you want to gain insights into and then click **Task Insights** in the operation column to view the details.

Based on the actual execution of the current task, DLC Task Insights provides corresponding optimization suggestions by combining data analysis and algorithmic rules.

How to Enable the Insight Feature

Upgrading the Kernel Image for Existing SuperSQL Engines

> **Note:**
> For engines newly purchased after July 18, 2024, or existing engines that have been upgraded to a kernel version released after July 18, 2024, insights have been automatically enabled. You can skip this step.

Operation Steps

1. Go to the SuperSQL engine list page and select the engine you want to gain insights into.

2. On the engine details page, click **Kernel Management > Version Upgrade** (which upgrades to the latest kernel by default).

Overview of Key Insight Metrics

Metric Name	Metric Meaning
Engine execution time	Reflects the time the first task was executed on the Spark engine (the time when the task first preempted the CPU for execution).
Engine execution duration	Reflects the actual time used for computation, which is the duration from the start of the first Task to the completion of the Spark job. More specifically, it is the sum of the duration from the start of the first task to the completion of the last task for each Spark stage. This sum does not include the queuing time of the task before it starts (that is, excluding other time, but including the time required for scheduling between task submission and the start of execution of the Spark task), nor include the time spent waiting for task execution due to insufficient executor resources between multiple Spark stages during the task execution process.
Execution wait time (queue time)	Specifies the time taken from task submission to the start execution of the first Spark task. The time taken may include the cold startup duration of the first execution of the engine, the queuing time caused by the concurrent limit of the configuration task, the time waiting for executor resources due to full resources within the engine, and the time taken to generate and optimize the Spark execution plan.

Accumulated CPU * hours (consumed CU * hours)	Specifies the sum of the CPU execution duration of each core of the Spark Executor used in computing, per hour (not equivalent to the duration of starting machines in the cluster, because the machines may not participate in task computing after they start. Eventually, the cluster's CU fee is subject to the bill). In the Spark scenario, it is approximately equal to the sum of the execution durations of the Spark task (seconds) / 3600 (per hour).
Data scan size	The amount of physical data read from storage by this task. In the Spark scenario, it approximately equals to the sum of the Stage Input Size in Spark UI.
Total output size	The size of the records output after this task processes data. In the Spark scenario, it is approximately equal to the sum of the Stage Output Size in Spark UI.
Data shuffle size	In the Spark scenario, it approximately equals to the sum of the Stage Shuffle Read Records in Spark UI.
Number of output files	(The collection of this metric requires upgrading the Spark engine kernel to a version after November 16, 2024.) Total number of files written by the task through insert and other statements.
Number of small output files	(The collection of this metric requires upgrading the Spark engine kernel to a version after November 16, 2024.) Small file definition: an output file is defined as a small file if its size is < 4 MB (controlled by the parameter spark.dlc.monitorFileSizeThreshold, which defaults to 4 MB and can be configured at either the engine global or task level). This metric definition: the total number of small files written by the task through insert and other statements.
Parallel tasks	Displays the concurrent execution status of tasks, facilitating the analysis of affected tasks (up to 200 entries).

Overview of Insight Algorithms

Insight Type	Algorithm Description (Continuously Being Improved and Added)
Resource Preemption	The delay time of the task that starts executing the SQL is greater than the stage submission time by 1 minute, or the delay duration exceeds 20% of the total runtime (the threshold formula is dynamically adjusted based on the runtime and data volume of the task).

Shuffle Anomaly	Shuffle-related error stack information occurs during stage execution.
Slow Task	The task duration in a stage is greater than twice the average duration of other tasks in the same stage (the threshold formula is dynamically adjusted based on task runtime and data volume).
Data skew	Task shuffle data is greater than twice the average shuffle data size of other tasks (the threshold formula is dynamically adjusted based on task runtime and data volume).
Disk or Memory Insufficiency	Error stack information during stage execution includes OOM, insufficient disk space, or bandwidth limitation errors of Cloud Object Storage (COS).
Excessive Small File Output	(The collection of this insight type requires upgrading the Spark engine kernel to a version after November 16, 2024) See the metric "Number of Output Small Files" in the list. The presence of excessive small file output is determined if one of the following conditions is met: 1. Partitioned tables: The number of small files written out by a partition exceeds 200. 2. Non-partitioned tables: The total number of output small files exceeds 1000. 3. Partitioned or non-partitioned tables: The total number of files written out exceeds 3000, and the average file size is less than 4 MB.

Engine Usage Insights

Last updated: 2026-05-20 16:26:19

When cluster resources are tight, tasks submitted to the engine may be queued within it. However, users are unaware of this and may continue submitting tasks, leading to severe task blocking.
Engine Usage Insights presents a unified view of the distribution of all tasks under an engine, from submission to execution, based on the engine dimension. This helps customers quickly analyze the general usage of the engine.

Prerequisites

1. SuperSQL SparkSQL and Spark Job Engine:
   1.1 Engine Usage Insights is enabled by default for newly purchased engines after July 18, 2024.
   1.2 For Spark kernel editions released before July 18, 2024, you must upgrade the engine kernel to enable Task Insights. For the upgrade procedure, see How to Enable the Insight Feature below.
2. Standard Spark Engine:
   2.1 Engines purchased after December 20, 2024, support Engine Usage Insights by default.
   2.2 For engines purchased before December 20, 2024, Engine Usage Insights cannot be manually enabled by users. To enable this feature, submit a ticket to contact after-sales support.

Engine Usage Insights is not currently supported for other engine types.

Operation Steps

Log in to the DLC console and select the Insight Management feature. Then, navigate to the Engine Usage Insights page, select the data engine you want to view, and hover your mouse over the task duration waterfall chart to view the details.

- The gray progress bar represents queuing time.
- The blue progress bar represents engine execution time.
- The green progress bar represents the time taken to obtain results.
- The overall bar chart shifts over time.

> ⚠ **Note:**

> The SparkSQL engine of SuperSQL supports real-time engine execution duration and data scan volume. For other engine types, queuing time and engine execution duration data are available only after the insight process is complete.

Intelligent Storage

Last updated: 2026-05-20 16:26:37

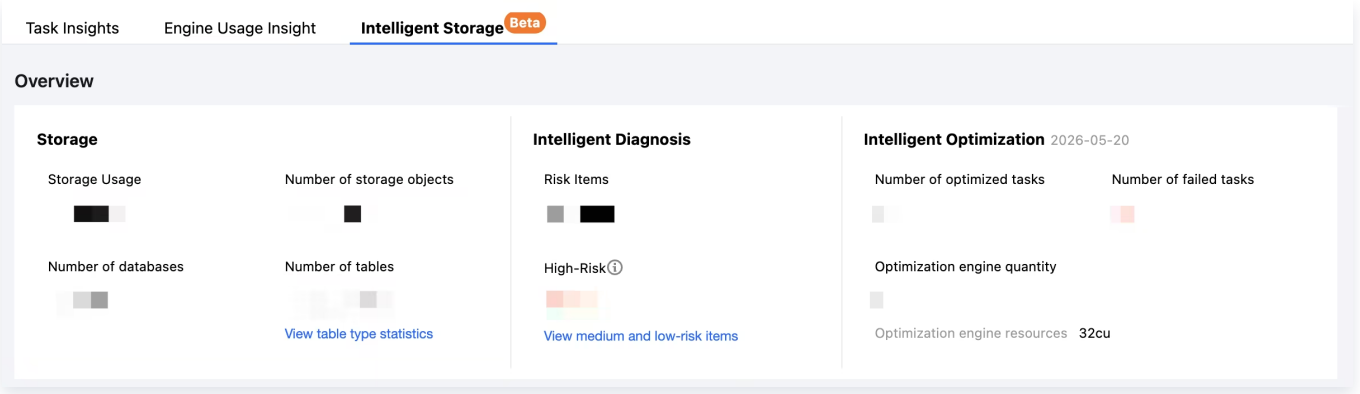
Intelligent Storage helps users observe storage usage, distribution, and storage views at the COS bucket directory and table levels in a one-stop manner, enabling them to understand storage distribution and reduce storage costs. It provides a holistic view of the current storage status and governance optimization. This helps users diagnose risks, configure data optimization, and resolve potential risks.

Operation Steps

Log in to the Data Lake Compute (DLC) console > Insight Management and switch to the Intelligent Storage page.

Overview

Metric

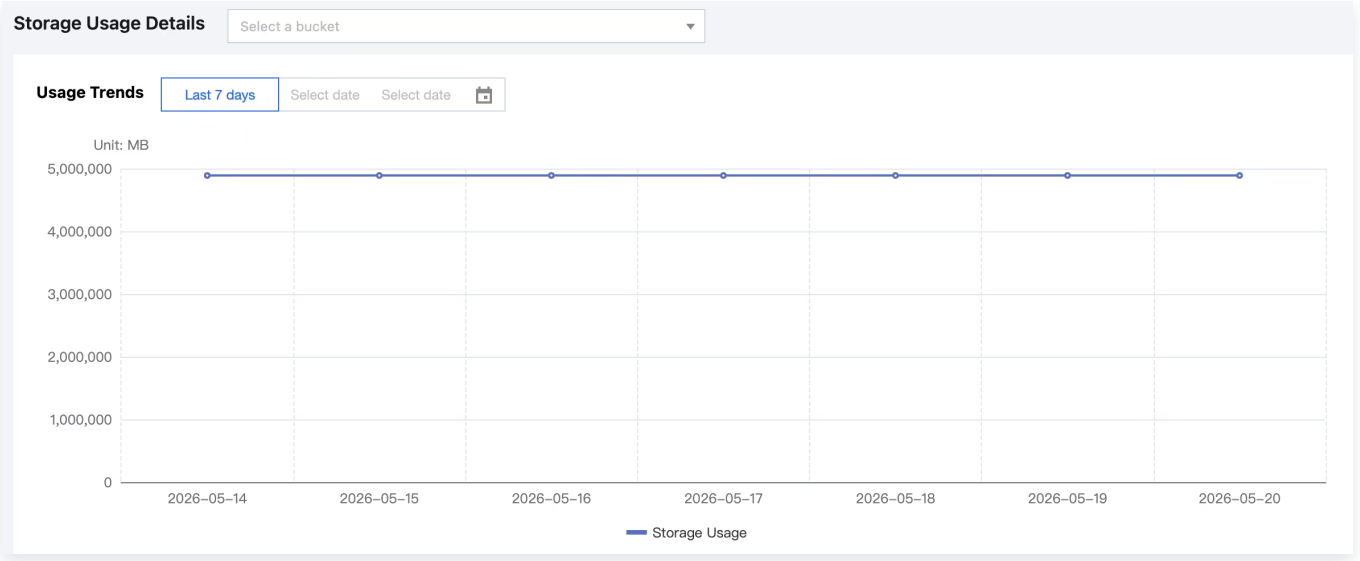


Category	Metric	Description
Storage	Storage usage	Total storage capacity
	Number of Storage Objects	Total number of actual storage objects under COS storage
	Number of Databases	Total number of databases under DatalakeCatalog
	Number of Tables	Total number of data tables under DatalakeCatalog. Supports viewing native tables

		(Iceberg), native tables (hive), external tables (Iceberg), and external tables (hive) via Hover.
Intelligent diagnosis	Risk Items	Scans the internal tables within DLC under DatalakeCatalog to identify the total number of potential risk items.
	High-Risk Items	Risk items are categorized into three levels based on severity: high, medium, and low. The total number of high-risk items is displayed by default. The total numbers of medium- and low-risk items can be viewed via Hover.
Intelligent optimization	Optimization Engine	If the user enables intelligent optimization, the quantity and specifications of engines used for optimization tasks are displayed.
	Number of Optimization Tasks	Total number of optimization tasks for the day
	Number of Failed Tasks	Number of Failed Tasks for the Day

Storage Usage Details

Users can view storage usage trends by bucket. (By default, data for the past seven days is displayed, with a maximum of 30 days available for retrospective analysis.)

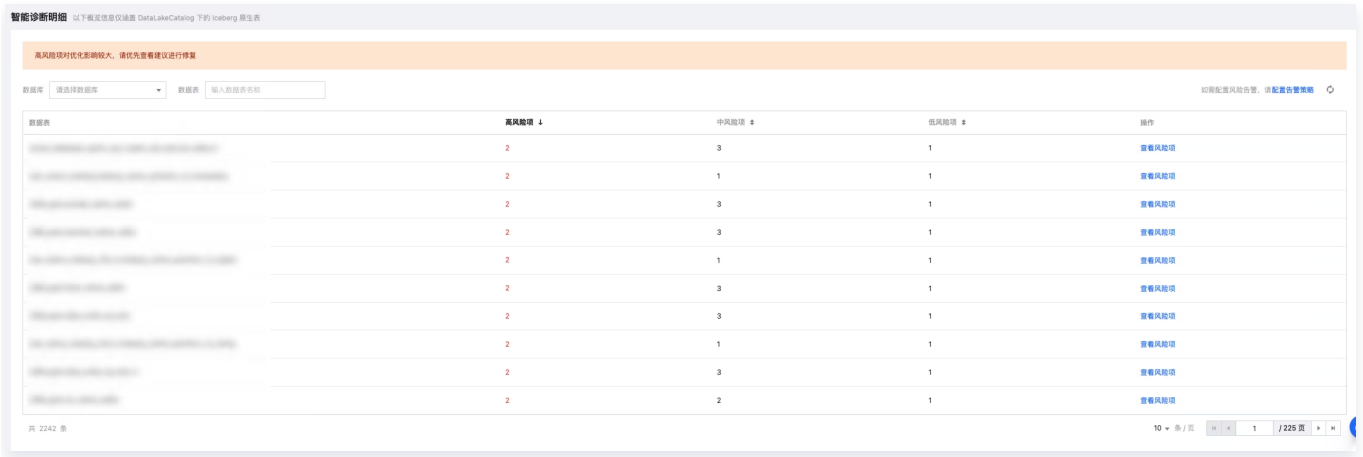


Intelligent Diagnosis Details

Note:

> Only Iceberg native tables under DataLakeCatalog support intelligent diagnosis.

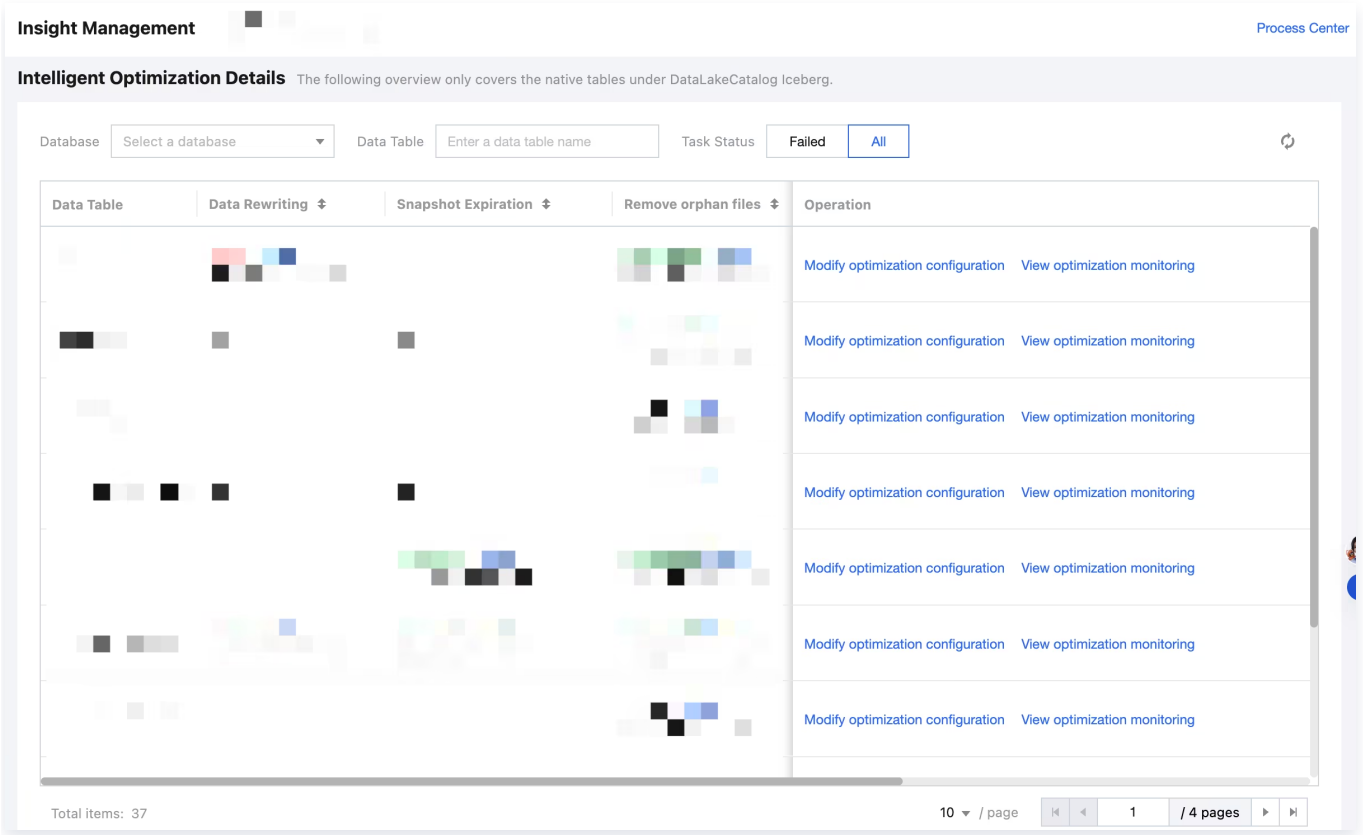
Users can quickly view tables with numerous risk items and perform further targeted optimization for each risk item.



Intelligent Optimization Details

> **Note:**
> Only Iceberg native tables under DataLakeCatalog support intelligent diagnosis.

Users can view the execution status of table-level optimization tasks and then modify optimization configurations or view optimization monitoring accordingly.



System Management
User and Permission Management
CAM service

Last updated: 2026-07-14 15:39:52

DLC has a comprehensive access permission mechanism. DLC permissions are categorized into operation permissions and data permissions. Operation permissions are managed by the CAM service, while data permissions are managed by the DLC permission module.

- A root account has all DLC operation permissions and data permissions by default.
- If a sub-user is granted the operation permission for DLC data permission management, that sub-user can grant data permissions to other sub-users. Such users can be considered "administrators".
- If a sub-user is granted data read/write permissions, that sub-user can run query tasks on the data for which they have permissions. Their data permissions are assigned by an "administrator".
- Data permissions for all sub-users, except for the root account, are assigned by an "administrator". Users cannot query data for which they do not have permissions.

A root account has all DLC operation permissions by default. A root account grants DLC access permissions to sub-users through CAM, enabling those sub-users to have the corresponding DLC operation permissions. This document primarily explains operation permissions.

CAM Permissions Policy

DLC's current operation permissions are implemented through CAM. DLC supports two permission control methods: preset policies provided by the DLC product and custom policies managed by customers.
Preset policies:
DLC currently defines two preset policies:

- QcloudDLCFullAccess: Full read/write access to Data Lake Compute (DLC). This policy includes read/write permissions for all APIs that DLC accesses through CAM. Users can directly grant this permission policy to sub-accounts, but this may result in excessive permissions. Please proceed with caution.
- QcloudDLCReadOnlyAccess: Read-only access to Data Lake Compute (DLC). This policy includes permissions for all read-only APIs that DLC accesses through CAM. Users can directly grant this permission policy to sub-accounts, but it does not include any modification permissions.

> **Attention:**
> 1. The specific APIs included in QcloudDLCFullAccess and QcloudDLCReadOnlyAccess can be viewed by navigating to CAM > Policies in the left menu > QcloudDLCFullAccess / QcloudDLCReadOnlyAccess details page > Service > Data Lake Compute (DLC).
> 2. QcloudDLCFullAccess and QcloudDLCReadOnlyAccess only include permissions for DLC's own APIs. Some DLC capabilities rely on and are implemented by other cloud products. When using these capabilities, you must additionally grant sub-accounts permissions for the corresponding products' APIs. For the required permission APIs and authorization operations for each product, see the appendix of this document.

CAM custom policies:
Users can create custom permission policies through the CAM console to achieve flexible permission management. CAM provides three methods for authorizing custom policies. Taking the policy generator method as an example, refer to Creating Custom Policies via the Policy Generator.

Operation Permission Categories

As shown in the table below, DLC operations are categorized according to DLC APIs. For details, see API Documentation.

Permission Type	Description
Metadata Management	Operate metadata of databases and tables managed by DLC
Task Management	Submit and view DLC tasks
Permission Management	Manage user data access permissions
System Configuration	Basic configuration of DLC service

Preset Policy Authorization Operations

This procedure uses <Granting QcloudDLCFullAccess to a Sub-account> as an example. The operation steps are as follows:

1. Create a sub-user.
2. Grant the QcloudDLCFullAccess permission to the corresponding sub-user. For details, see Sub-account Authorization via Policy.
3. Sub-users log in to the DLC console and verify permissions. If a sub-user successfully logs in to the console and performs a data permission authorization operation, the CAM

permission authorization takes effect.

Custom Policy Authorization Operations

If you access DLC using a root account, you can skip this step. Sub-accounts must be authorized before the DLC service can be enabled. The authorization process is as follows:

1. Create a sub-account. For the operation steps, see Create Sub-account and Authorize.
2. Create a custom policy.
   - On the Policy page in the CAM console, click **Create Custom Policy** to create a policy.
   - In the pop-up window, click **Create by Policy Syntax** to enter the edit policy page.
   - On the Create by Policy Syntax page, select **Blank Template** and click **Next**.
   - In the template, enter a policy name and description (it is recommended to name the policy DLCDataAccess), and paste the copied policy into **Policy Content**. After the form is completed, click **Finish** to successfully create the custom policy. Sub-users granted this custom policy can log in to the DLC console to execute SQL tasks, but they cannot manage data permissions.

```
{
  "version": "2.0",
  "statement": [
      {
          "effect": "allow",
          "action": [
              "dlc:DescribeStoreLocation",
              "dlc:DescribeTable",
              "dlc:DescribeViews",
              "dlc:CancelTask",
              "dlc:CreateDatabase",
              "dlc:CreateScript",
              "dlc:CreateTable",
              "dlc:CreateTask",
              "dlc:DeleteScript",
              "dlc:DescribeDatabases",
              "dlc:DescribeScripts",
              "dlc:DescribeTables",
              "dlc:DescribeTasks",
              "dlc:DescribeQueue",
              "dlc:DescribeTaskResult"
          ],
```

```
            "resource": [
                "*"
            ]
        }
    ]
}
```

3. After the custom policy DLCDataAccess is bound to the sub-account that accesses DLC, the sub-account can log in to and access DLC. For details, see Sub-user Permission Settings.

Appendix

Some capabilities of DLC rely on other cloud products, such as Cloud Monitor and Tag. When using these capabilities, you must additionally grant sub-accounts the corresponding product API permissions. Root accounts can grant the corresponding operation permissions to sub-accounts as needed. The related operation APIs are listed in the following table:

DLC Feature Page	Other Cloud Product	Operation	Description
Data Jobs/Storage Management	COS	GetService	Pulls the bucket list. Whether bucket data pulling is supported is managed by COS. For details, see Authorizing Sub-Accounts to Access COS.
SuperSQL engine > Cluster monitoring Standard engine > Cluster monitoring Resource group > Resource group monitoring	TCOP	DescribeDashboardMetrics	Queries the Dashboard2.0 metric list.
		DescribeMonitorProductByIds	Queries the monitoring product list by ID.
		DescribeDashboardMetricData	Queries Dashboard2.0 metric monitoring data.

SuperSQL engine Standard engine Storage management	TAG	DescribeTagKeys	Queries tag keys.
		AttachResourcesTag	Binds tags to resources in batches.
		DetachResourcesTag	Unbinds tags from resources in batches.
Network Connection Configuration/Metadata Management	MySQL	DescribeDBInstances	Queries the list
	TencentDB for PostgreSQL	DescribeDBInstances	Queries the instance list
	TencentDB for SQL Server	DescribeDBInstances	Queries the instance list

This section describes the process for a root account to grant TCOP permissions to a sub-account, using the permissions required to view monitoring data on the DLC console page as an example.

1. Prerequisite: Create a sub-account. For the operation steps, see Create Sub-account and Authorize.
2. Create a custom policy. This example uses the Create Custom Policy via Policy Generator method. The operation flow is: CAM > Policy > Create Custom Policy > Create via Policy Generator. The creation details are shown in the figure below. For other creation methods, see the official CAM documentation.

   Additional API permissions, policy names, Tag settings, associated users/user groups/roles, and other content can be configured as needed. This document only presents the minimum configuration required to enable monitoring view in the DLC console.
3. Grant the policy defined in Step 2 to the sub-account. This operation can be authorized when the policy is created or performed on the User, User Group, and Role Authorization page after the full policy is created.

DLC Permissions Overview

Last updated: 2026-05-20 16:27:45

DLC permissions include data permissions and data engine permissions. Users with administrator permissions can log in to the DLC console or use APIs to grant data and data engine permissions to sub-users (SuperSQL engine only). Permissions for any sub-user must be granted; otherwise, the sub-user cannot perform operations such as using, modifying, or deleting data and engines.

Users and Working Groups

DLC provides two user management modes for customers: User Mode and Working Group Mode. Both modes can be used to manage permissions for DLC users.
User: A user in CAM, including sub-accounts and collaborator accounts.
Working group: A group managed within the product that includes a set of users. Users in the same group have the same permissions.

> **Note**
> When the permissions granted to a user differ from the permissions of the working group to which the user belongs, the system takes the union of the two sets of permissions.

Through working groups, you can assign permissions to users quickly without the need to grant permissions to each user individually. Enterprise users are recommended to use the working group mode for authorization. For detailed steps, see Users and Working Groups.

User Type

DLC user types are categorized into **DLC Administrators** and **Common Users**.

- DLC Administrators: They are responsible for managing users who use DLC, possess all permissions for data, engines, tasks, and more, and can perform management operations (such as adding, granting permissions to, and removing) on users and working groups.
- Common Users: They are ordinary users of DLC, added by DLC administrators. By default, they have no DLC permissions and require authorization to obtain the corresponding permissions. They can only grant permissions for data and engine permissions that have the **re-grant** permission.

Permission & Operation	DLC Administrator	Ordinary user

Data permission	All permissions	Not granted by default, permissions are granted by the DLC administrator.
Data engine permissions	All permissions	Not granted by default for the SuperSQL engine, permissions are granted by the DLC administrator. Granted by default for the standard engine, can be managed through CAM. For details, see Standard Engine Permission Management.
User management	Perform	Not operable.
Working group management	Perform	Not operable.
Authorization scope	All permissions	Permissions that can be granted within the scope.

Note

The permissions described above only include those defined within DLC. Operations related to billing, such as purchase/configuration change/refund, require you to go to CAM to grant the financial permission QCloudFinanceFullAccess (for the authorization steps, see: Create Sub-Account and Grant Permissions). These operations are not included in the permissions described above.

Data permission

DLC data permissions include operation permissions for data catalogs, databases, and data tables. To facilitate your management and configuration, we provide two permission granting modes: regular permission settings and advanced permission settings.

- In the regular permission setting mode, you can directly grant roles to users (for the mapping between roles and permissions, see Sub-account Permission Management) without needing to focus on specific permission configurations. The authorization granularity covers data catalogs, databases, and data tables. This mode is suitable for scenarios requiring quick authorization and no complex permission management.
- In Advanced permission setting mode, you can grant comprehensive permissions to a single database and the data tables, views, and functions within it. This mode is suitable for scenarios that require fine-grained permission management.

The SQL operations corresponding to the operation permissions are as follows:

Action	CREATE	ALTER	DROP	SELECT	INSERT	DELETE	Target
CREATE DATABASE	✓	–	–	–	–	–	Catalog
ALTER DATABASE	–	✓	–	–	–	–	Database
DROP DATABASE	–	–	✓	–	–	–	Database
CREATE TABLE	✓	–	–	–	–	–	Database
CREATE TABLE AS SELECT	✓	–	–	✓	✓	–	Database/Table
DROP TABLE	–	–	✓	–	–	–	Table
ALTER TABLE LOCATION	–	✓	–	–	–	–	Table
ALTER PARTITION LOCATION	–	✓	–	–	–	–	Table
ALTER TABLE ADD PARTITION	–	✓	–	–	–	–	Table
ALTER TABLE DROP PARTITION	–	✓	–	–	–	–	Table
ALTER TABLE	–	✓	–	–	–	–	Table
CREATE VIEW	✓	–	–	–	–	–	Database
ALTER VIEW PROPERTIES	–	✓	–	–	–	–	View
ALTER VIEW RENAME	–	✓	–	–	–	–	View
DROP VIEW PROPERTIES	–	✓	✓	–	–	–	View
DROP VIEW	–	–	✓	–	–	–	View
SELECT TABLE	–	–	–	✓	–	–	Table
INSERT	–	–	–	–	✓	–	Table

INSERT OVERWRITE	–	–	–	–	✓	✓	Table
CREATE FUNCTION	✓	–	–	–	–	–	Database
DROP FUNCTION	–	–	✓	–	–	–	Function
SELECT VIEW	–	–	–	✓	–	–	View
SELECT FUNCTION	–	–	–	✓	–	–	Function

Note:
Specifically, if you use the unified data catalog TCCatalog service as the DLC built-in metadata service (**currently in beta testing and only available to invited users on the allowlist**), the data operation permissions will change to the following model:

Action	USE	CREATE	ALTER	DROP	SELECT	INSERT	DELETE	Target
USE CATALOG	✓	–	–	–	–	–	–	CATALOG
ALTER CATALOG	–	–	✓	–	–	–	–	CATALOG
DROP CATALOG	–	–	–	✓	–	–	–	CATALOG
USE SCHEMA	✓	–	–	–	–	–	–	CATALOG
CREATE SCHEMA	–	✓	–	–	–	–	–	SCHEMA
ALTER SCHEMA	–	–	✓	–	–	–	–	SCHEMA
DROP SCHEMA	–	–	–	✓	–	–	–	SCHEMA
CREATE TABLE	–	✓	–	–	–	–	–	SCHEMA
CREATE FUNCTION	–	✓	–	–	–	–	–	SCHEMA

SELECT TABLE/VIEW	–	–	–	–	✓	–	–	TABLE /VIEW
INSERT TABLE/VIEW	–	–	–	–	–	✓	–	TABLE /VIEW
DELETE TABLE/VIEW	–	–	–	–	–	–	✓	TABLE /VIEW
ALTER TABLE/VIEW	–	–	✓	–	–	–	–	TABLE /VIEW
DROP TABLE/VIEW	–	–	–	✓	–	–	–	TABLE /VIEW
USE FUNCTION	✓	–	–	–	–	–	–	FUNC TION
DROP FUNCTION	–	–	–	✓	–	–	–	FUNC TION
ALTER FUNCTION	–	–	✓	–	–	–	–	FUNC TION

Data Engine Permissions

DLC data engines are categorized into two types: the **SuperSQL Engine** and the **Standard Engine**. For detailed differences and application scenarios, see Data Engines. The permissions for the earlier-released **SuperSQL Engine** are managed by the DLC console. You can perform quick permission management for the SuperSQL engine in DLC Console > User and Permission Management. In contrast, the permission management for the **Standard Engine** is uniformly controlled by Tencent Cloud CAM. The Standard Engine can be configured as a cloud resource through CAM for highly flexible resource-level authentication settings, meeting enterprise-level security management requirements. For more information on resource-level authentication, see the document Resource-Level Authentication Guide.

SuperSQL Engine Permission Management

The operational permissions for the DLC SuperSQL data engine include permissions to use, modify, operate, monitor, and delete the data engine. The specific permissions are as follows:

- Use: The permission to select and use this engine for tasks.
- Modify: You can modify the basic information and configuration information of the engine. (Modifying configuration information also requires CAM financial permissions.)
- Operate: The permission to suspend and restart the engine.

- Monitor: The permission to view the engine's running tasks and monitoring information.
- Delete: The permission to request a refund for the engine.

A single user can be granted multiple sets of permissions. For detailed operations on authorizing the SuperSQL engine, see Sub-account Permission Management.

Standard Engine Permission Management

The permissions for the DLC Standard Engine are uniformly controlled by Tencent Cloud CAM. To ensure that sub-accounts can use the DLC Standard Engine smoothly, the root account needs to grant permissions to the sub-accounts. After a Standard Engine is created, all sub-users with the **QcloudDLCFullAccess** (Data Lake Compute (DLC) Full Read/Write Access) policy have permissions for the Standard Engine. If you need to manage the permissions of the Standard Engine in a fine-grained manner, for example, granting User A permissions only for Engine A, you can achieve this by **creating a custom policy**.

Requirement Scenarios	Operation
Scenario 1: The sub-account has **all** standard engine permissions.	Associate the QcloudDLCFullAccess preset policy with the sub-account.
Scenario 2: The sub-account has **partial** standard engine permissions.	Create a custom policy.

> **Note:**
> Based on your specific requirements and the two scenarios described above, you can choose one of the following methods to manage user permissions for the Standard Engine.

Scenario 1: Granting All Standard Engine Permissions to a Sub-user

After logging in with a root account or a sub-account that has CAM operation permissions, locate the corresponding sub-account in the sub-account list, click **Grant Permissions** in the Operation column, search for and select **QcloudDLCFullAccess**, and then click **OK**. For detailed operations, see New User Onboarding Full Process.

> **Note:**
> The QcloudDLCFullAccess policy grants sub-accounts full read/write permissions for DLC, including the use and management of all Standard Engines.

Scenario 2: Granting Partial Standard Engine Permissions to a Sub-user

DLC supports **resource-level authentication** based on CAM Tags. You can use Tags to classify and manage existing DLC Standard Engine resources and engine-related API permissions, thereby achieving multi-dimensional classification management and fine-grained authorization for resources. For information about Tencent Cloud Tags, see Tencent Cloud Tags.

> **Note:**
> **Based on Tencent Cloud Tags, you can quickly achieve the following effects in the DLC Standard Engine:**
> - All users in Department A can only use Standard Engine resources tagged with the Department A Tag and cannot use Standard Engines tagged with other Tags.
> - When creating DLC Standard Engine resources, users in Department A must tag them with the Department A Tag. If they do not tag the resources or tag them with a Tag other than the Department A Tag, the creation will fail (optional).

Operation Steps

Step 1: Create a Tag
Go to Tag Management > Tag List. Click the **Create Tag** button.
Enter the Tag key and Tag value, and click **OK** to create the Tag successfully. For example, to create a Tag with the department name **Analyze**, you can enter "department" for the Key and "Analyze" for the Value.
Step 2: Tag the Standard Engine
Go to DLC console > Standard Engine. Select the appropriate standard engine to go to the **Basic Configurations** page. Select a tag from the **Tag Options** to bind. For detailed steps on binding tags to engines, see Associating Tag with Private Engine Resource.

> **Note:**
> Once a specific Tag, such as "department: Analyze" in the example above, is applied to a Standard Engine, **only sub-users associated with this Tag can view and use this Standard Engine thereafter.**

Step 3: Create a Custom Policy
1. Go to the Tencent Cloud CAM console, select Policies in the left sidebar, click **Create Custom Policy**, and select **Tag-based Authorization** in the pop-up dialog box.
2. In the Custom Policy Creation Wizard, search for and select DLC under Service, and **select All actions (dlc:*) under Action.**

> **Note:**

> **All actions (dlc:*)** means that users are granted permissions to operate all DLC TencentCloud APIs. If you need to restrict users from having permissions to destroy, create, or modify Standard Engines, simply deselect the corresponding APIs under the aforementioned **All actions (dlc:*)**.

Scenario	DLC Interface to Deselect
Engine Deletion Failure	DeleteDataEngine
Engine Creation Failure	CreateDataEngine
Engine Modification Failure	UpdateDataEngine

3. Select the previously created "department: Analyze" Tag. By default, select resource_tag for Condition Key. **Note: For whether to grant the "resource": "*" permission to APIs that do not support Tags,** you need to select **"Yes"**. If you do not select it, the associated sub-users will be unable to access normally because they lack permissions for non-Tag-level authorization APIs in the DLC console.

> **Note:**
> If you need to enforce the association of the "department: Analyze" Tag to newly created engines when users from the Analyze department create DLC Standard Engine resources, you can select the request_tag option. For further details and usage of request_tag, refer to the document Enforcing Fixed Tag Key-Value Binding During Resource Creation.

4. Click Next. Because this involves many DLC APIs, **CAM will create multiple split sub-policies**. You can assign a search-friendly name to the split custom policy, such as DLC-department-analyze-tag-policy. In the Associate User or Associate User Group section, select the users/user groups to be associated with this custom policy. For example, in this sample, associate all employee sub-accounts from the Analyze department.
5. Click **Finish** to complete the custom policy creation. After the custom policy is successfully created, all DLC users associated with this custom policy can only access Standard Engines tagged with "department: Analyze".

> **Note:**
> 1. To facilitate subsequent user maintenance, it is recommended to use **user groups** for association.
> 2. If a user is associated with a custom policy and has already been associated with the QcloudDLCFullAccess preset policy, the user will still have all Standard Engine

permissions.

Users and Working Groups

Last updated: 2026-05-20 16:28:07

DLC provides two user management modes for customers: User Mode and Working Group Mode. Permissions for DLC users can be managed in both modes. For details about DLC permissions, see DLC Permissions Overview.

Basic Information

- User: Users in CAM, including sub-accounts and collaborator accounts.
- Working group: A group managed within the product that includes a batch of users. Users within the group have the same permissions.

> **Note**
> When the permissions granted to a user differ from the permissions of the working group to which the user belongs, the system takes the union of the two sets of permissions.

Through working groups, you can assign permissions to users quickly without the need to grant permissions to each user individually. Enterprise users are recommended to use the working group mode for authorization.

User Management

To perform user management, you must have the relevant DLC operation permissions. For details, see CAM Service.

Adding Users

1. Log in to the DLC console, select a service region, and go to the User and Permission Management **page** in the left sidebar.
2. Click **Add User**, and enter user information. Select Tencent Cloud users (select to add all sub-accounts or collaborator accounts under the root account). You can also manually create a user (add by entering the user UIN), and select the user type.

> **Note:**
> Select user type as "**Regular User**" to authorize personally or obtain all permissions of the specified working group. Select user type as "**Administrator**" without binding a working group to obtain all permissions.

3. Bind a working group: Select a working group to bind (optional).

Viewing User Information

DLC administrators can modify user basic information and permission information.

1. Log in to the DLC console, select a service region, and go to the User and Permission Management **page** in the left sidebar.
2. To find the user **account ID** that you need to view, you can quickly locate it through search. Click the **user name** to view the user's related information and permissions.

Editing User Information

You can modify user description information and working groups through the edit feature. For permission-related change operations, see Sub-account Data Permission Authorization.

1. Log in to the DLC console, select a service region, and go to the **Permission Management** page.
2. To find the user account ID that needs to be edited, you can quickly locate it through search. Click **Edit** under the **operation column** on the far right to go to the edit page.

Removing a user

If you do not want the user to continue using DLC, you can remove the user with an administrator account. After removal, the user's DLC access permissions will be revoked and the user will no longer be able to use DLC.

1. Log in to the DLC console, select a service region, and go to the User and Permission Management **page** in the left sidebar.
2. To find the user account ID that needs to be removed, you can quickly locate it through search.
   - Delete an individual user: Click **Delete** under the **operation column** on the far right to remove the user according to the user ID.
   - To delete multiple users: Click the checkbox in front of the user entry to enable batch removal. After selection, click **Batch Delete Users** to remove the selected users from DLC.

Working Group Management

To manage working groups, you must have the relevant DLC operation permissions. For details, see CAM Service.

Adding a Working Group

You can manage permissions that need to be repeatedly granted in the form of working groups. The operation process for adding a working group is as follows.

1. Log in to the DLC console, select a service region, and go to the User and Permission Management page in the left sidebar.
2. Click **Working Group** to go to the working group management page.
3. Click **Add Working Group**, fill in the relevant information and confirm to complete the addition of the working group.

Viewing Working Group Information

The edit working group feature allows you to modify the working group description and the included users. The operation process for editing a working group is as follows.

1. Log in to the DLC console, select a service region, and go to the User and Permission Management page in the left sidebar.
2. Click **Working Group** to go to the working group management page.
3. To find the working group you need to view, you can quickly locate it by entering the working group ID in the **search bar**. Click the **Working Group ID** or **Working Group Name** to view the working group information.

Editing Working Group Information

The edit working group feature allows you to modify the working group description and the included users. The operation process for editing a working group is as follows.

1. Log in to the DLC console, select a service region, and go to the **Permission Management** page.
2. Click **Working Group** to go to the working group management page.
3. Locate the **working group name** you need to edit, and click **Edit** in the **Operation** column to go to the edit page.
   - To edit the description, click  .
   - You can add users in DLC to the working group by clicking the Bind User button.
   - You can remove a user from the working group by selecting the user and clicking Batch Remove, or by clicking the Remove button in the user's operation column. After removal, the user will no longer have the permissions granted by that working group, but other granted permissions will not be affected.

Deleting a Working Group

DLC administrators can remove working groups.

> ⚠ **Note:**

> After removal, all permissions related to the working group granted to its authorized users will be revoked. This action is irreversible, so proceed with caution.

1. Log in to the DLC console, select a service region, and go to the **Permission Management** page.
2. Click **Working Group** to go to the working group management page.
3. To remove a working group, locate its name, then either select the checkbox in the table header and click Batch Remove, or click the Remove button in the operation column.

Sub-account Permissions

Last updated: 2026-05-20 16:30:13

User Permissions

User permissions consist of data permissions and engine permissions (for details, see DLC Permissions Overview). To access data in DLC, you must have the corresponding data permissions. DLC provides permissions management at the database/table level and fine-grained permissions management at the column level. This facilitates granting data permissions to users or groups in different scenarios and managing data permissions with precision. In addition, you can use the corresponding engine permissions to manage SuperSQL engine resources.

Users and Working Groups

You can grant permissions to individual users, or you can create a working group that includes a batch of users to grant permissions. For user and working group management operations, see Users and Working Groups.

- User: Users in CAM, including sub-accounts and collaborator accounts.
- Working group: A group managed within the product that includes a batch of users. Users within the group have the same permissions.

> **Note**
> When the permissions granted to a user differ from the permissions of the working group to which the user belongs, the system takes the union of the two sets of permissions.

Through working groups, you can assign permissions to users quickly without the need to grant permissions to each user individually. Enterprise users are recommended to use the working group mode for authorization.

Adding User Permissions

Grant permissions to the specified user.

1. Set a user as an administrator/common user. For the procedure, see the Users and Working Groups: Add a User section. Users of the administrator type have all permissions on all resources, including data and engines, without binding a working group. They can also manage administrator users other than the root account. **This permission must be set with caution.**

2. Bind to a working group: Under **User Type**, select **Working Group Options** and then select **Bind to Working Group**. Common users need to be granted the corresponding permissions or bound to a working group to access the corresponding resources.
3. Add data permissions: On the **Users** list page, select the edit option under the **Operations** column on the far right to go to the **Edit User** page. Then, select the authorization operation and grant the user permissions on data directories or databases/tables on the edit page.
   - Add data catalog permissions. Through the data catalog, you can set permissions to create databases under DataLakeCatalog and to create other data catalogs.
   - Add database/table permissions. In the database/table permission settings module, click Add Permission. This includes two modes: Basic Settings and Advanced Settings. Basic permission settings: You can add permissions for database tables under a specified directory and set query analysis, data editing, and owner permissions.

     The specific permissions are as follows:

Permission Type	Database	Data Table	Views and Functions
Query and analysis permission	• Query permissions for all tables, views, and functions in the database. • Permissions to create data tables.	Query data.	Query data.
Data editing permission	• Permissions to modify, delete, and create tables in the database. • All permissions for all tables, views, and functions.	• Data query, insertion, update, and deletion. • Table modification and deletion.	Query, create, modify, and delete.
Owner permission (Based on data editing permission, permissions can be regranted.)	• Database modification, deletion, and table creation. • All permissions for all tables,	• Data query, insertion, update, and deletion. • Table modification and deletion.	Query, create, modify, and delete.

	views, and functions.		

- Advanced permission settings: When you select a single database, you can further set permissions for querying, inserting, updating, and deleting tables, views, and functions under the database. When you select multiple databases, only database-level permissions are set.
- In Advanced mode, column-level permission settings are supported. When you select a single data table, you can add query permissions for its columns. You can grant permissions by selecting one/multiple columns, or all columns. On the Data Exploration page, if you query a column for which you have query permission, the query result returns the content of that column. If you query a column for which you do not have query permission, the system returns an error message indicating that you are not authorized to query that column.

4. Add engine permissions (only the SuperSQL engine is supported): Under the Edit option in the user's corresponding operation column, go to the Edit page, select Engine Permissions, and grant the specified resources the permissions to use, modify, operate, monitor, and delete.

> **Important Note:**
> In the DLC console > Permission Management > Engine Management, permission management is only supported for the earlier-released SuperSQL engine. The permission management for the **Standard Engine** is uniformly controlled by Tencent Cloud CAM. The Standard Engine can be configured as a cloud resource through CAM for highly flexible resource-level authentication settings, meeting enterprise-level security management requirements. For practices on Standard Engine permission management, refer to the **Standard Engine Permission Management** section in DLC Permission Overview.

Modifying User Permissions

1. In the user list, select the Edit option under the operation column, go to the authorization page, and select database/table permissions or engine permissions (only the SuperSQL engine is supported).
2. On the authorization page (using database/table permissions as an example), you can modify permissions by adding or removing them. The same operation applies to engine permission modifications.
3. To modify a working group or user type, click **Operation > Edit** to go to the user editing page, where you can modify the user name, user type, and description. Common users can

add or remove working groups.

4. Click the **Edit** button to change the user type.

View user permissions

1. Click the **user ID** in the user list to go to the user details page.
2. View information such as the user's working group, data permissions, and engine permissions.

Removing User Permissions

To revoke a permission, you can click Delete under the operation column in the user's permission list to remove that permission. Removing a user requires administrator operation.

Adding and Removing Group Permissions

Adding or removing a working group requires administrator operation, similar to user data permission operations. On the Permission Management page, select Working Group to view the working group list. Users in a group have all the permissions owned by that group. You can manage user permissions in batches by binding a group of users to a working group and granting the group permissions for resources such as data and engines. Administrators do not need to be bound to a working group.

Monitoring and Alarming
Data Engine Monitoring

Last updated: 2026-05-20 16:30:41

DLC provides a monitoring service for data engines based on TCOP. This service enables you to understand the operational status of your data engines in real time and configure data alarms. For details on how to configure alarms, refer to Monitoring and Alarm Configuration.

Usage Instructions

Before using DLC monitoring services, you need to activate Tencent Cloud Observability Platform (TCOP) services. For usage details, refer to Tencent Cloud Observability Platform Documentation. If you have not activated this service, you can use your root account to activate it.
Usage of Tencent Cloud Observability Platform services may incur related charges. For detailed billing information, refer to Billing Overview.

Monitoring Entrance

Method 1: DLC Console

> ⚠ **Note:**
> The account must have monitoring permissions for the data engine.

1. Log in to the DLC console and select the service region.
2. Based on the type of the purchased engine, go to the SuperSQL Engine page or Standard Engine page from the left sidebar.
3. Go to the page and you can see the engine list. Select the **Monitor** option under the **operation column** on the far right of the corresponding engine.

Method 2: TCOP

1. Log in to Tencent Cloud Observability Platform (TCOP). The account used for login must have the relevant permissions.
2. From the left menu, select **Cloud Product Monitoring** > **Instance Monitoring**, locate DLC, and then select the type of monitoring you want to view.
3. After selecting the monitoring type, go to the Monitoring page. Then, select **the corresponding region** to view the monitoring resource information for that region.
4. Click **Engine ID** to go to the monitoring details page.

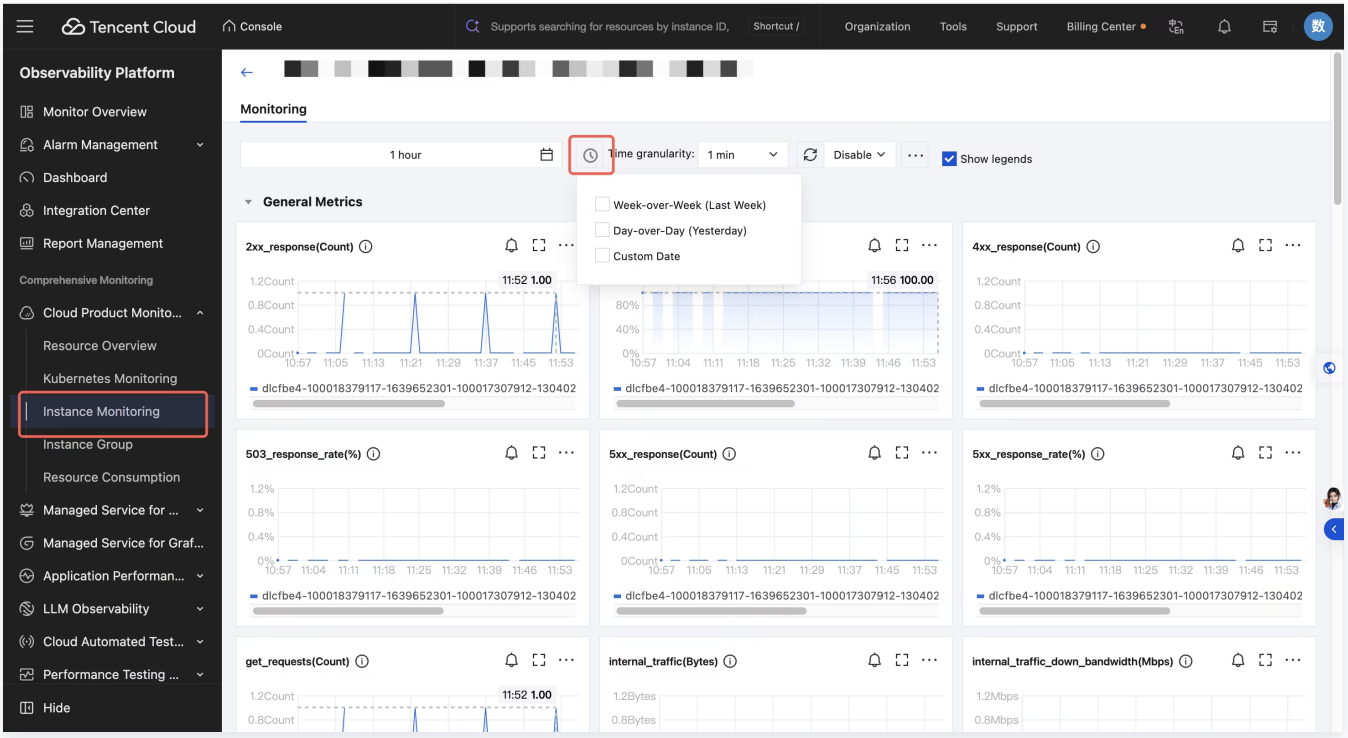
Configuring Monitoring Granularity

You can configure the time range, time granularity, and auto-refresh time range for monitoring data at the top of the monitoring page.

- Monitoring data time range: It is accurate to the minute and supports selecting data for a specific time period.
- Time granularity: The interval between monitoring points, which can be configured as 1 minute or 5 minutes.
- Auto-refresh data: The configuration for automatically refreshing page data supports setting it to Disable, 30s, 5min, 30min, or 1h.

Monitoring Data Comparison

You can select data for a specific time period for monitoring comparison. After clicking to select the comparison time range, you can view the comparison data in the data compass below.



Monitoring Metrics

Monitoring Type	Monitoring Metric
CPU	Maximum CPU utilization across all Driver nodes
	Maximum CPU utilization across all Executor nodes

	Average CPU utilization across all Driver nodes
	Average CPU utilization across all Executor nodes
	Maximum CPU utilization across all clusters
	Average CPU utilization across all clusters
Memory	Maximum memory utilization across all Driver nodes
	Maximum memory utilization across all Executor nodes
	Average memory utilization across all Driver nodes
	Average memory utilization across all Executor nodes
	Maximum memory utilization across all clusters
	Average memory utilization across all clusters
Task	Number of canceled tasks
	Number of failed tasks
	Number of initializing tasks
	Average task initialization duration
	Maximum task initialization duration
	Number of queued tasks
	Average task queuing duration
	Maximum task queuing duration
	Number of running tasks
	Number of successful tasks
Networking	Maximum inbound network bandwidth across all Driver nodes
	Maximum inbound network bandwidth across all Executor nodes
	Average inbound network bandwidth across all Driver nodes
	Average inbound network bandwidth across all Executor nodes
	Maximum outbound network bandwidth across all Driver nodes

	Maximum outbound network bandwidth across all Executor nodes
	Average outbound network bandwidth across all Driver nodes
	Average outbound network bandwidth across all Executor nodes
Cloud disk	Maximum cloud disk utilization across all Driver nodes
	Maximum cloud disk utilization across all Executor nodes
	Average cloud disk utilization across all Driver nodes
	Average cloud disk utilization across all Executor nodes
CU	Number of CUs for the job engine
	CU utilization

Data Job Monitoring

Last updated: 2026-05-20 16:30:57

DLC provides monitoring services for data jobs based on Tencent Cloud Observability Platform (TCOP). This ensures you can understand the operational status of your data jobs in real time and configure data alarms.

Usage Instructions

Before using DLC monitoring services, you need to activate Tencent Cloud Observability Platform (TCOP) services. For usage details, refer to Tencent Cloud Observability Platform Documentation. If you have not activated this service, you can use your root account to activate it.
Usage of Tencent Cloud Observability Platform services may incur related charges. For detailed billing information, refer to Tencent Cloud Observability Platform Billing Overview.

Monitoring Entrance

Method 1: DLC Console

1. Log in to DLC Console > Data Jobs and choose the service region.
2. Alternatively, go to the Data Jobs page from the left menu bar.
3. Click **Job Monitoring** in the upper-right corner to go to the Monitoring page. Alternatively, click the **Monitoring** feature of the target job to go to the Target Job Monitoring page.

Method 2: TCOP

1. Log in to the Tencent Cloud Observability Platform (TCOP) console. The account you use to log in must have the relevant permissions.
2. From the left menu, select Cloud Product Monitoring, locate DLC, and then select the type of monitoring you want to view.
3. After the monitoring type is selected, go to the Monitoring page. At the top of the page, select **the corresponding region** to view the monitoring job information for that region.
4. Click **Job ID** to go to the Monitoring Details page.

Configuring Monitoring Granularity

You can configure the time range, time granularity, and auto-refresh time range for monitoring data at the top of the monitoring page.

- Monitoring data time range: It is accurate to the minute and supports selecting data for a specific time period.

- Time granularity: The interval between monitoring points, which can be configured as 1 minute or 5 minutes.
- Auto-refresh data: The configuration for automatically refreshing page data supports setting it to Disable, 30s, 5min, 30min, or 1h.

Monitoring Data Comparison

You can select data for a specific time period for monitoring comparison. After clicking to select the comparison time range, you can view the comparison data in the data compass below.

Monitoring Metrics

Monitoring Type	Monitoring Metric
Job	Job ERROR Logs Number
	Job WARN Logs Number

Access Point Gateway Engine Monitoring

Last updated: 2026-05-21 16:57:34

DLC provides monitoring services for the access point gateway engine based on Tencent Cloud Observability Platform (TCOP). This ensures you can understand the gateway status in real time.

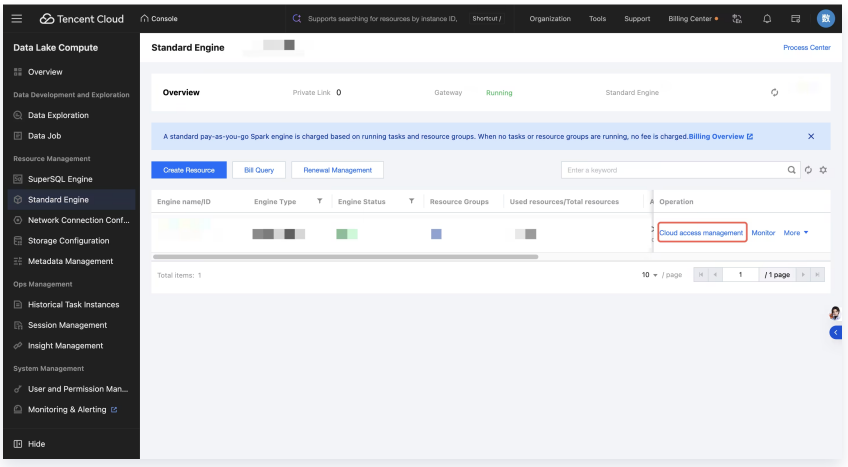
Usage Instructions

Before using DLC monitoring services, you need to activate Tencent Cloud Observability Platform (TCOP) services. For usage details, refer to Tencent Cloud Observability Platform Documentation. If you have not activated this service, you can use your root account to activate it.
Usage of Tencent Cloud Observability Platform services may incur related charges. For detailed billing information, refer to Tencent Cloud Observability Platform Billing Overview.

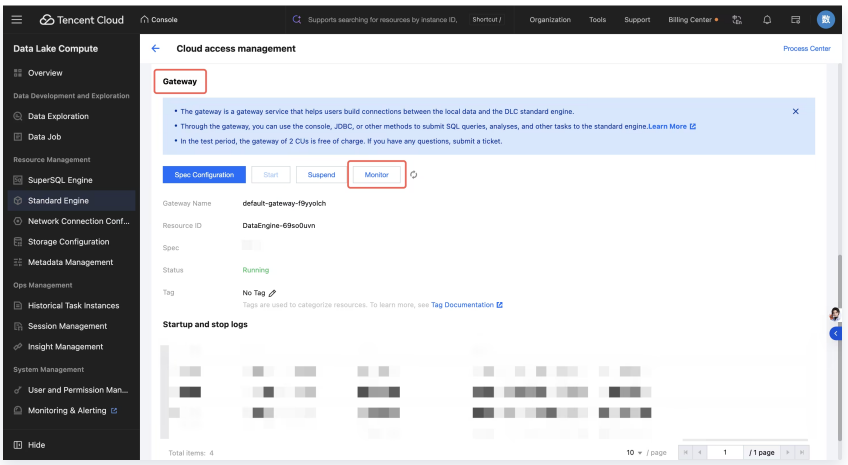
Monitoring Entrance

Method 1: DLC Console

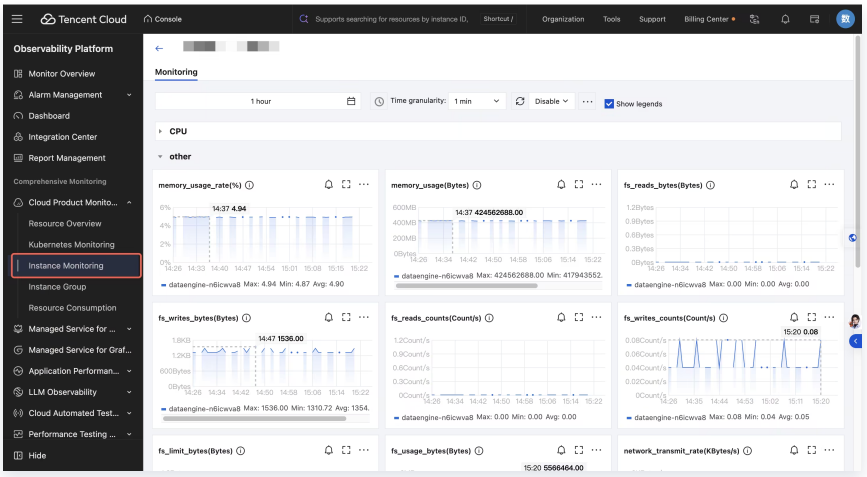
1. Log in to the DLC console page and select the service region.
2. Go to the Standard Engine page in the left sidebar. Then, select **CAM** under the **Operations** column on the far right of the corresponding engine.



3. Go to the **CAM** page, scroll to the **Gateway** module, and select the **Monitor** button to enter the monitoring data display page.

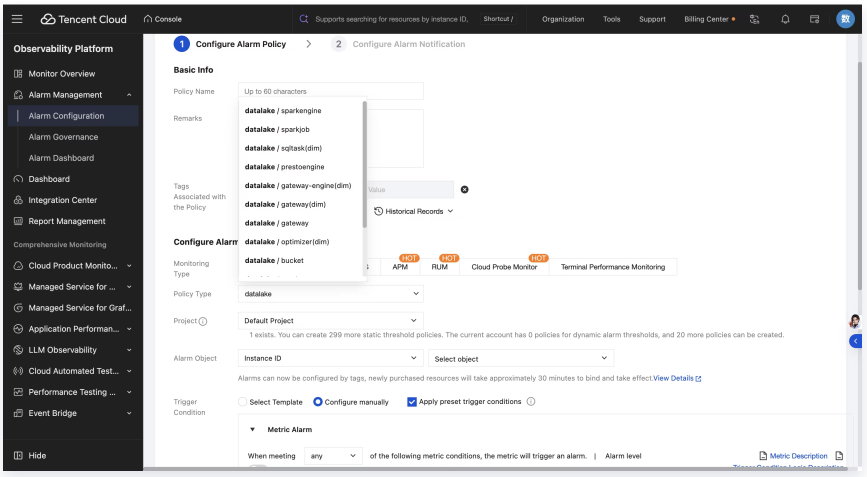


The monitoring data display page is shown below:



Configuration Method: TCOP

1. Log in to the Tencent Cloud Observability Platform (TCOP) console. The account you use to log in must have the relevant permissions.
2. In the left-side menu, select Alarm Configurations under **Alarm Management**. Then, click **New Policy** and select the corresponding access point gateway engine for DLC.



Access Point Gateway Engine Monitoring Configuration Types

Creating an alarm policy

1. The DLC access point gateway supports alarm capabilities. To configure alarms, go to the Tencent Cloud Observability Platform (TCOP). Then, in the left-side menu, select Alarm Configurations under **Alarm Management**.
2. Click **New Policy** and select "DLC" as the policy type. The access point gateway supports alarms in three dimensions:
   - The "Gateway" alarm dimension is: appid/gatewayid.
   - The "Gateway (Multi-dimensional)" alarm dimension is: appid/gatewayid/instanceid.
   - The "Gateway Engine (Multi-dimensional)" alarm dimension is: appid/gatewayid/engineid/processid.

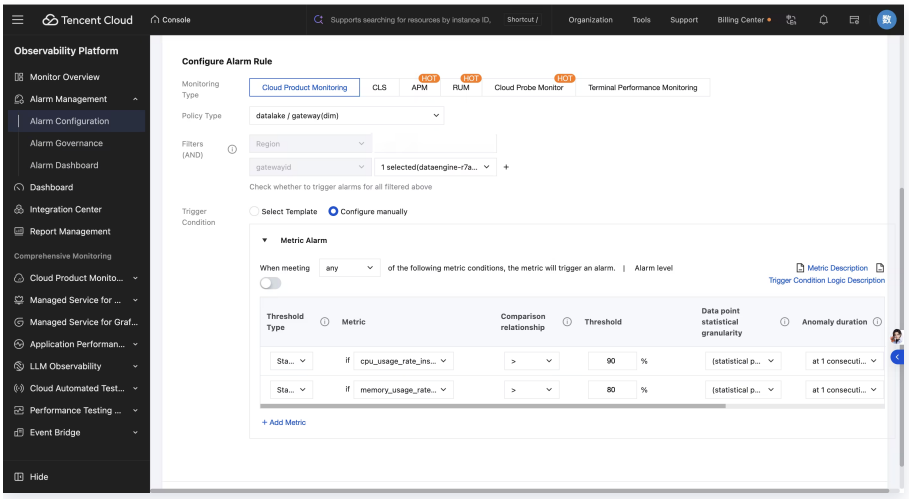
Name	Supported Dimension	Advantages and Applicable Scenarios
Gateway (Multi-dimensional)	Supports: CPU, memory, disk, and network fine-grained alarms. For example: to configure an alarm for the CPU utilization of an access point gateway, you can choose to configure the alarm to be triggered when one, multiple, or any instance node under a specific access point gateway reaches the threshold.	Alarms support more dimensions, and the alarm methods are more flexible. This method is recommended for basic metrics.
Gateway	Primarily monitors the overall load status of the current gateway, aggregates basic metrics by access point gateway node, and also supports service-level metric alarms. For example: execute_statement_num (number of statements executed), opened_operation_num (number of operations opened), launch_engine_num (number of engines launched), engine_process_thread_num (number of threads started by the engine).	Supports Dashboard. Applies to single-node access point gateways or scenarios requiring service metric alarms.
Gateway engine (Multi-dimen	A gateway engine refers to the access point gateway monitoring and alarming the processes that start DLC engines. For example: engine_process_thread_num (number of threads started by the engine), which primarily monitors the process information of	Supports fine-grained alarms, for example: it is common to configure an alarm to be triggered when the number of processes of any engine under a specific access point gateway ID reaches

sional)	engines started by the current access point gateway.	the threshold. Applies to alarms for process metrics of access point gateways.

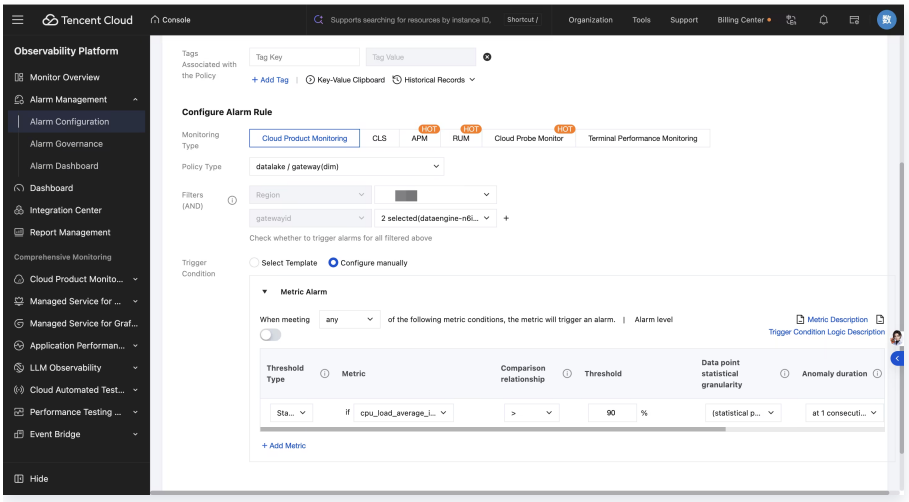
Configuring Monitoring Items

Gateway (Multi-dimensional) Configuration

Example 1: Configure an alarm for when the CPU utilization of any instance of a specific access point gateway reaches 90% or its memory utilization reaches 80%. The configuration is shown in the following figure:

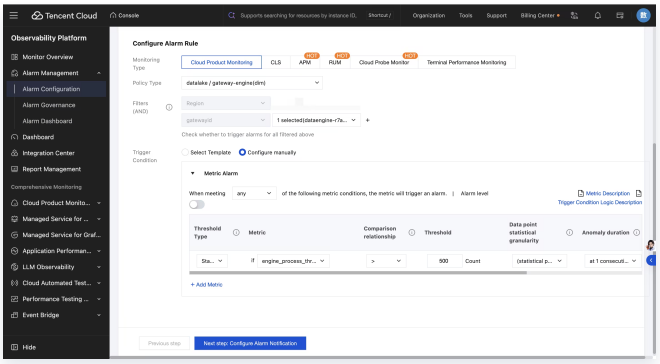


Example 2: To configure an alarm for when the CPU utilization of one or more instances of a specific access point gateway reaches 90% (this configuration method only triggers alarms for the selected instances and requires reconfiguration after cluster rebuild), the configuration is shown in the following figure:



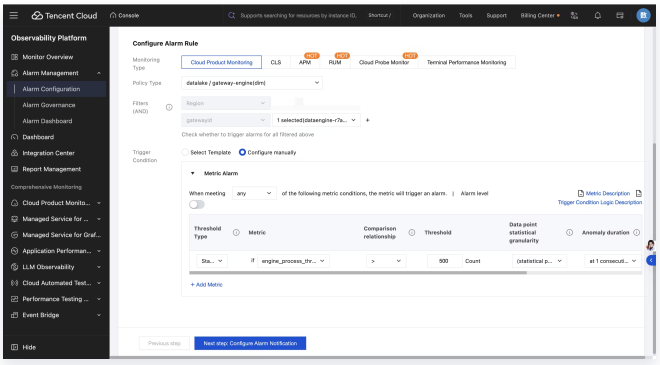
Gateway Configuration

Example 1: Configure an alarm for when the number of executed statements in a specific access point gateway ID exceeds 100. The configuration is shown in the following figure:

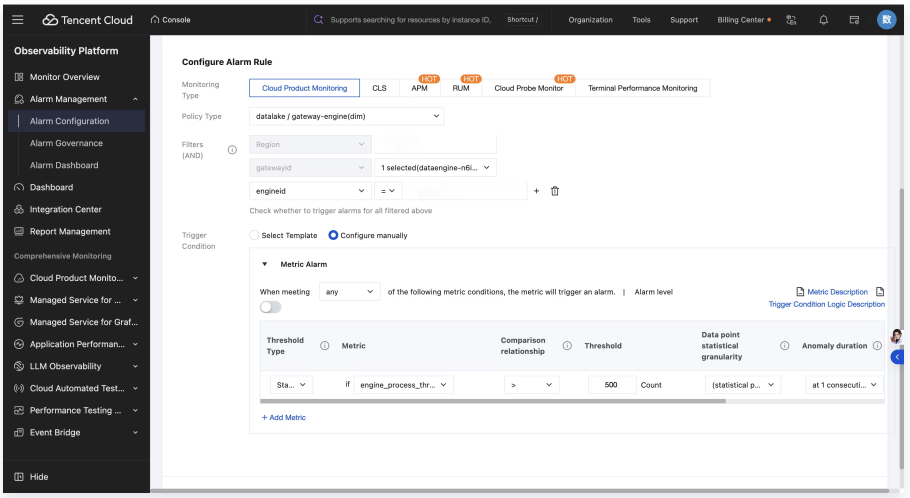


Gateway Engine (Multi-dimensional)

Example 1: Configure an alarm for when the number of threads in any running engine process within a specific access point gateway ID reaches 500. The configuration is shown in the following figure:



Example 2: Configure an alarm for when the number of threads in any process within a specific engine reaches 500 for a specific access point gateway ID. The configuration is shown in the following figure:



Configuring Monitoring Alarms

Last updated: 2026-05-20 16:31:43

Creating an Alarm Policy

Solution 1: You can configure monitoring and alarms for specific metrics. Log in to Tencent Cloud Observability Platform (TCOP). Click Alarm Configurations under **Alarm Management** in the left sidebar. Click the **Create Policy** button, and go to Creating Alarm Policy to configure alarm content.

Solution 2: Click the monitoring item for which you want to configure an alarm to go to the configuration page, where you can configure the alarm content.

Managing an Alarm Policy

To manage the configured alarm policies, you can go to **Alarm Management** and Alarm Configurations to manage the configurations.

Configuration Instructions

Configuration Item	Configuration Description
Policy Name	Policy Name (up to 60 characters)
Remarks	Remarks for the alarm policy (up to 100 characters)
Monitoring Type	Select a cloud product to monitor.
Policy type	Select DLC.
Policy label.	You can configure policy content via Tags. Relevant permissions are required to perform this operation.
Alarm object	You can configure alarms for instance IDs (multiple selection supported), grouped instances, or all instances.

Alarm Configuration Template	You can select a template or configure manually. Templates must be created in advance by an administrator before they can be used. Multiple alarm rules can be configured. For details, see Create an Alarm Policy.
Notification Templates	You can create a new notification template or select an existing one. Up to three templates can be configured. For details, see Create a Notification Template.

Audit Logs

Last updated: 2026-05-20 16:32:01

DLC provides an operation log auditing service based on Tencent Cloud's CloudAudit service. This enables you to review system operation records in real time and verify operation details.

Usage Instructions

Before using the DLC audit log service, you must activate Tencent Cloud's CloudAudit service. For details on using CloudAudit, refer to the CloudAudit documentation. If you have not activated this service, you can use your root account to activate it.

Usage Instructions

The DLC console currently displays log information for up to the last three months. To view logs from earlier periods, go to CloudAudit.
Audit logs include console operations and API call operations. Currently, you can view logs for engine management, task management, data source management, working group management, user management, scheduled task instance management, scheduled task management, and scheduled plan management.

Operation Guide

1. Log in to the DLC console and select the **service region**.
2. Go to the Audit Log page in the left sidebar.
3. Enter user UIN or request ID to perform log query. Detailed log information can be viewed by clicking **Query Details** under the **operation column** on the far right.