Data Lake Compute

Operation Guide

Product Documentation

Contents

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

Last updated：2024-07-17 17:36:45

The SQL editor provided by Data Lake Compute (DLC) supports data querying using unified SQL statements, compatible with SparkSQL. You can complete data query tasks using standard SQL.
You can access the SQL editor through data exploration, where you can perform simple data management, multi-session data queries, query record management, and download record management.

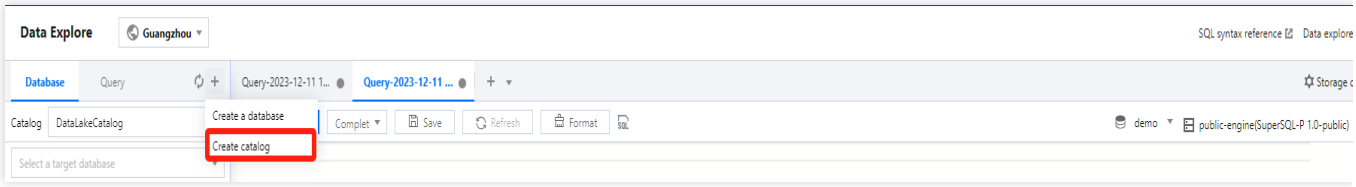
Data Management

Data management supports adding data sources, managing databases, and managing data tables.

Creating a data catalog

Currently, Data Lake Compute supports the management of COS and EMR Hive data catalogs. The directions are as follows:
1. Log in to the Data Lake Compute console and select the service region. You need to have the admin permission.
2. Select **Data Explore** on the left sidebar, hover over

on the **Database & table** tab, and click **Create catalog**.



For detailed directions, see Querying Data from Other Sources.

Managing a database

You can create, delete, and view the details of a database in the SQL editor.

Managing a data table

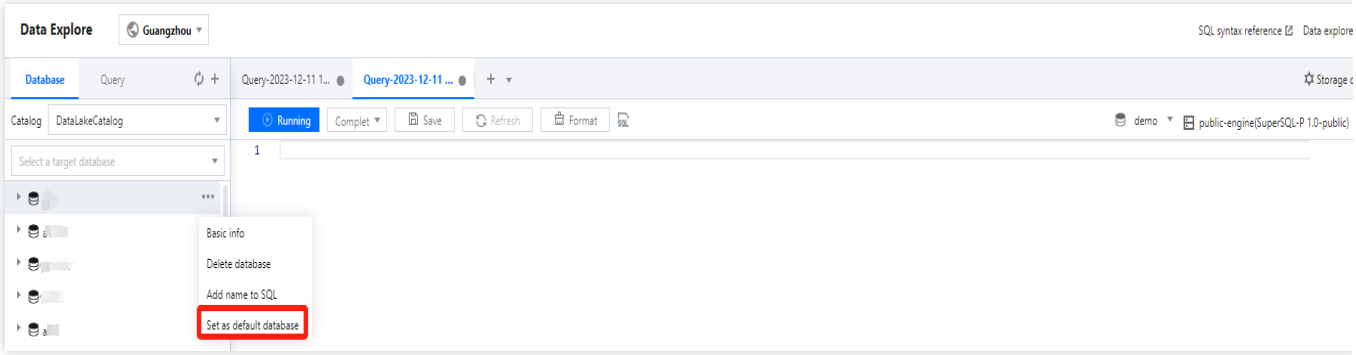
You can create, query, and view the details of a data table in the SQL editor.

Changing the default database

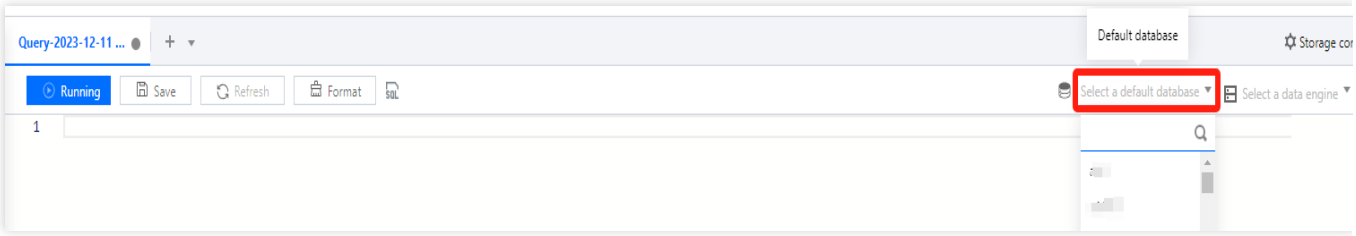
You can use the SQL editor to specify the default database for query tasks. If no database is specified in a query statement, the query will be executed in the default database.

1. Log in to the Data Lake Compute console and select the service region.
2. Select **Data Explore** on the left sidebar, hover over the target database name, click

, and click **Set as default database** to set the database as the default database.



3. You can also change the default database in the **Default database** selection box.



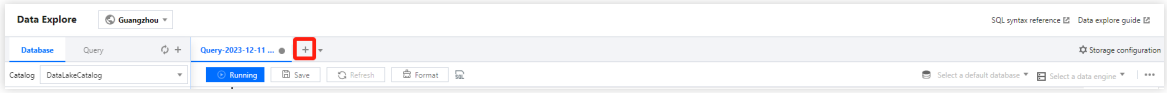
Data Query

Add Query Page

The SQL editor supports adding multiple pages for data querying, with each query page having independent configurations (default database, computation engine used, query records, etc.). This facilitates users in running and managing multiple tasks.

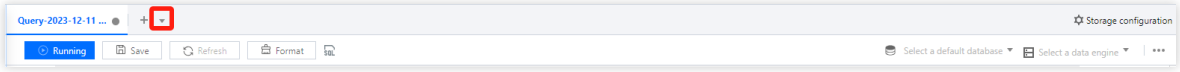
You can create a new query page by clicking on the

icon, and switch the editor interface by clicking on the tab bar.

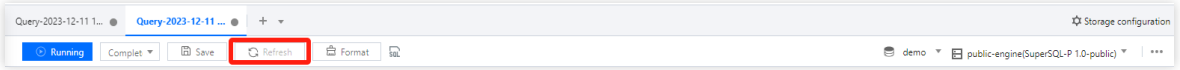


For your convenience, you can save frequently used query pages by clicking the **Save** button. You can also quickly open your saved pages by clicking the

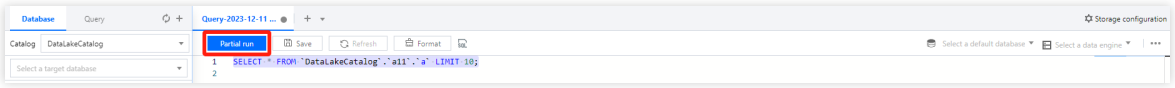
icon.



For saved query page information, you can click the **Refresh** button to update and synchronize the saved information, ensuring the accuracy of the query statement.

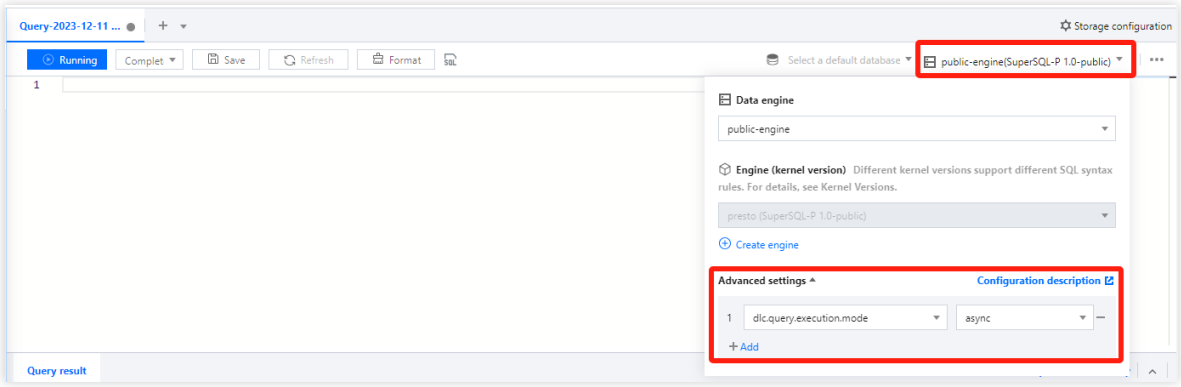


The editor supports running multiple different SQL statements simultaneously. Clicking the **Run** button will execute all SQL statements within the editor, simultaneously dividing them into multiple SQL tasks.
If you need to run a portion of the statement, select the required statement and click **Partial run**.



Engine Parameter Configuration

After selecting the data engine, you can configure parameters for the data engine. After selecting the data engine, click **Add** in Advanced Settings to configure.



The currently supported configuration parameters are as follows:

Engine	Configuration name	Start Value	Configuration Notes
SparkSQL	spark.sql.files.maxRecordsPerFile	0	The maximum number of records that can be written to a single file. If this value is zero or negative, there are no restrictions.
	spark.sql.autoBroadcastJoinThreshold	10MB	Configure the maximum byte size of the table of all working nodes displayed when executing a connection. By setting this value to "-1", the display can be disabled.
	spark.sql.shuffle.partitions	200	Default Partition Count.
	spark.sql.sources.partitionOverwriteMode	static	When the value is set to static, all qualifying partitions will be deleted prior to executing the overwrite operation. For instance, in a partitioned table, there is a partition "2022-01". When using the INSERT OVERWRITE statement to write data to the "2022-02" partition, the data in the "2021-01" partition will also be overwritten. When the value is set to 'dynamic', partitions will not be deleted in advance, but will be overwritten during runtime for those partitions where data is written.

	spark.sql.files.maxPartitionBytes	128MB	The maximum number of bytes to be packaged into a single partition when reading a file.
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 Hive functions when executing a query; if you need to use Presto native functions, please set the parameter to false.
	query_max_execution_time	-	This setting is used to establish a query timeout. If the execution time of a query exceeds the set time, the query will be terminated. The units supported are d-day, h-hour, m-minute, s-second, ms-millisecond (for example, 1d represents 1 day, 3m represents 3 minutes).
	dlc.query.execution.mode	async	The engine query execution mode is set to async mode by default. In this mode, the task will perform a complete query calculation, save the results to COS, and then return them to the user, allowing the user to download the query results after the query is completed. Users can also change this value to sync. In sync mode, queries may not necessarily perform full calculations. Once partial results are available, they will be directly returned to the user by the engine, without being saved to COS. Therefore, users can achieve lower query latency and duration, but the results are only saved in the system for 30 seconds. This mode is recommended for

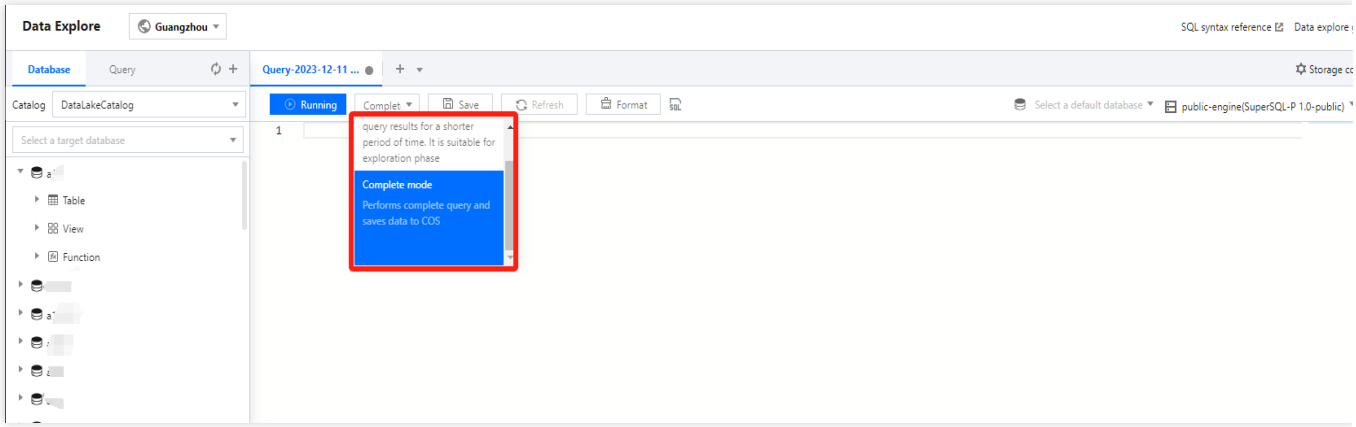
			users who do not need to download the complete query results from COS, but expect lower query latency and duration, such as during the query exploration phase or BI result display.

Presto Execution Mode

When the user selects the Presto engine, Data Exploration supports the user to choose to run in "Fast Mode" or "Full Mode".

Quick Query: This offers faster speed, but the query results cannot be persistently saved. It is suitable for the exploration phase.

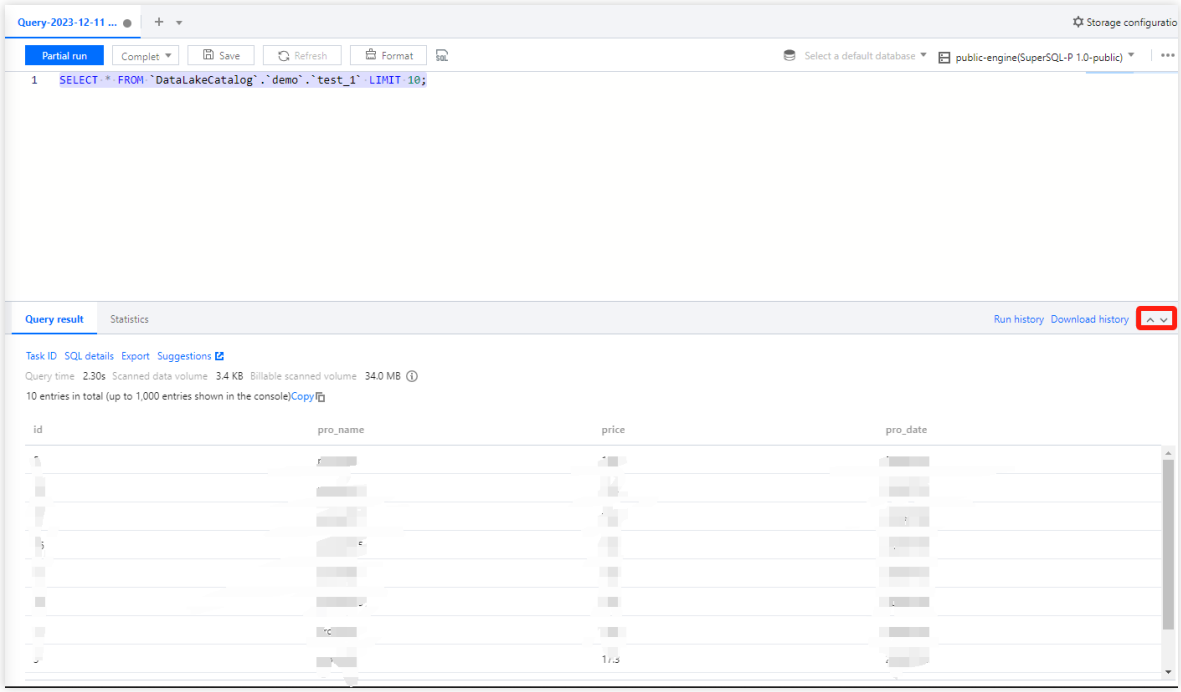
Full Mode: Execute a full query and save the data to object storage.



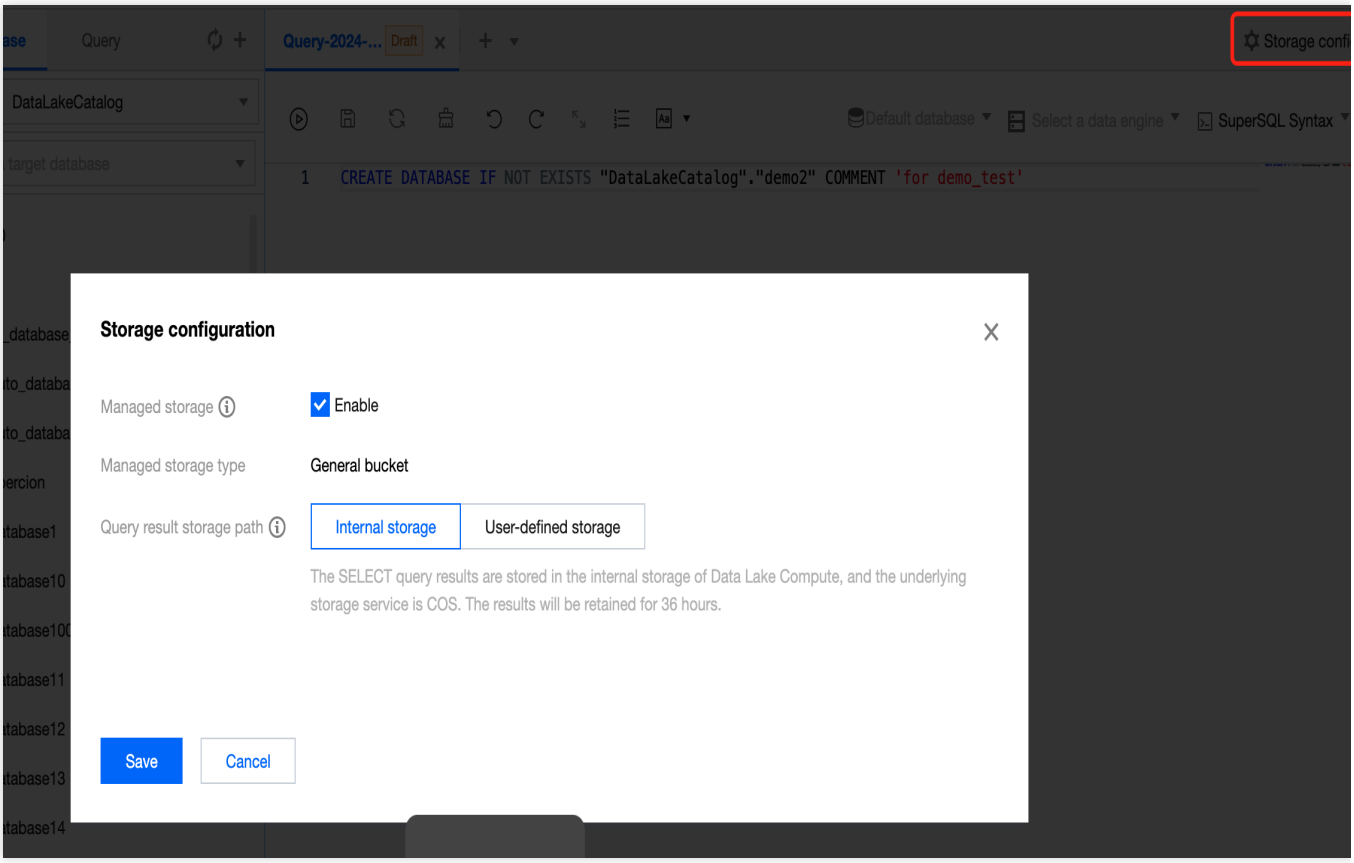
Search results

Through the SQL editor, you can directly view the query results. You can expand or collapse the display height of the query results by clicking the

chart.



You can configure the query result storage directory through the configuration button in the upper right corner, supporting configuration to the COS path or built-in storage.



The console will return a maximum of 1000 results for a single task. If more results are needed, the API can be used. For instructions on API-related operations, refer to the API Documentation.

Query results can be downloaded locally when no COS storage path is specified. For detailed instructions, refer to Obtaining Task Results.

Querying statistical data

The query results under the Presto engine and SparkSQL engine support the display of optimized quantification with different characteristics.

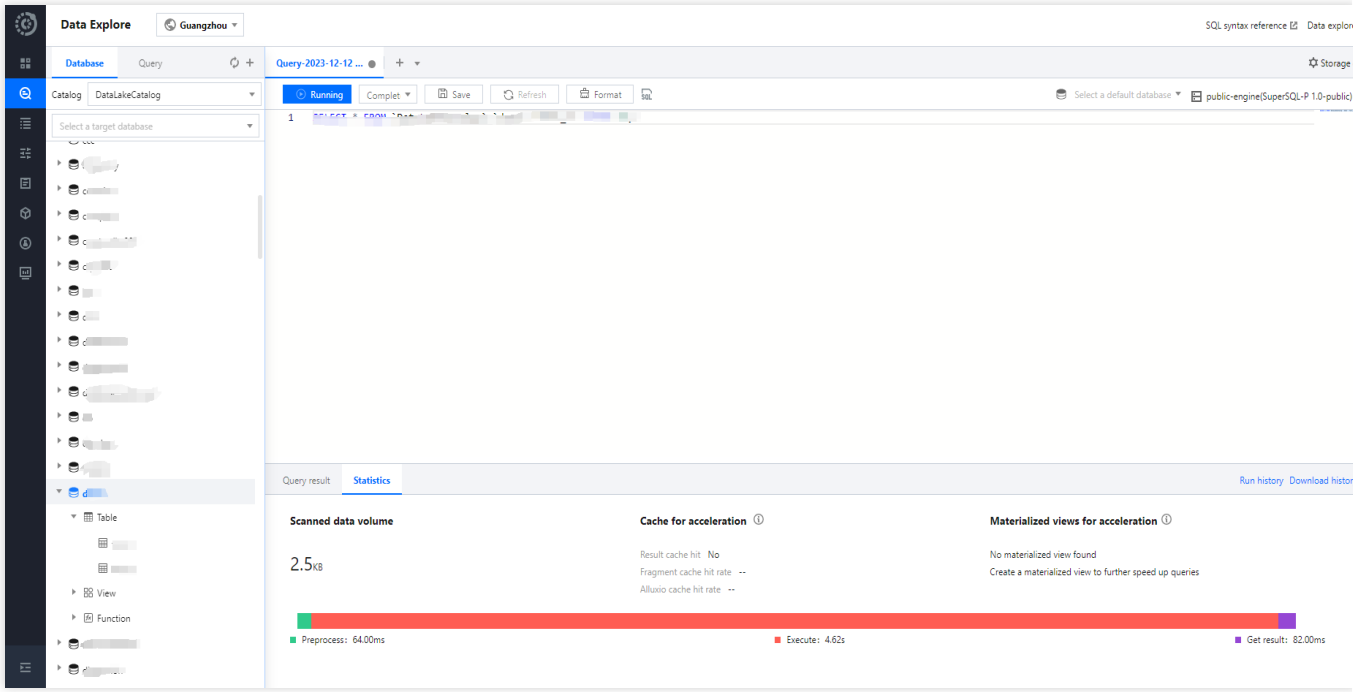
The SparkSQL engine supports viewing:

1. Data Scanning Volume
2. Cache Acceleration
3. Adaptive Shuffle
4. Materialized View Acceleration

The Presto engine supports viewing:

1. Data Scanning Volume
2. Cache Acceleration
3. Materialized View Acceleration

Click on the **Statistics** column to review the statistical data and optimization suggestions for the query results.

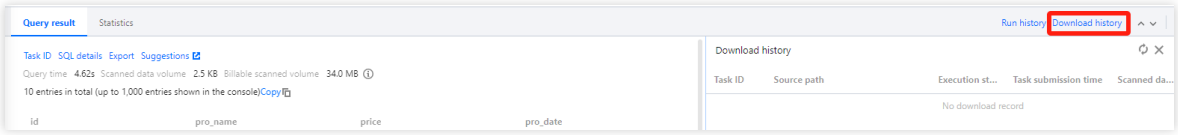


Historical Queries

Each query page can save the running history of the past three months and supports viewing the query results of the past 24 hours. You can quickly find past task information through the running history. For detailed operations, refer to Task History Records.

Download History Management

Each query result's download task can be viewed in the **Download history**, where you can check the status of the download task and related parameter information.



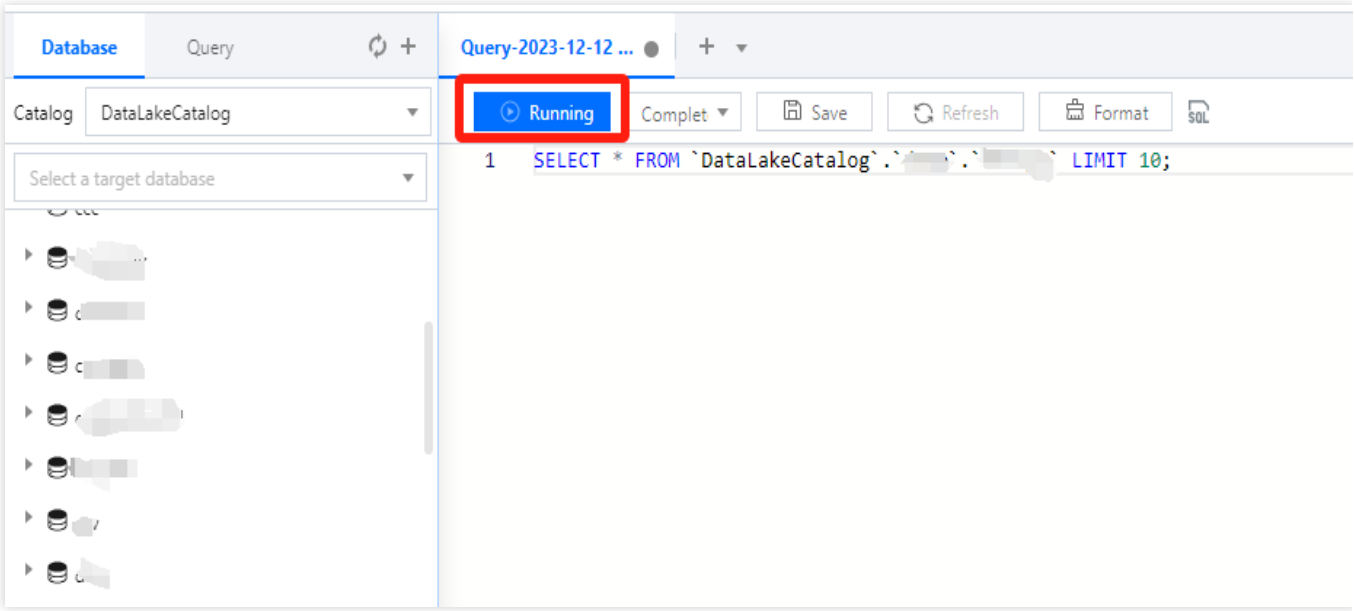
Data Query Task
SELECT Task

Last updated：2024-07-17 16:04:41

You can query, analyze, and compute the data in a created database or data table with SQL statements.

Running a `SELECT` query task

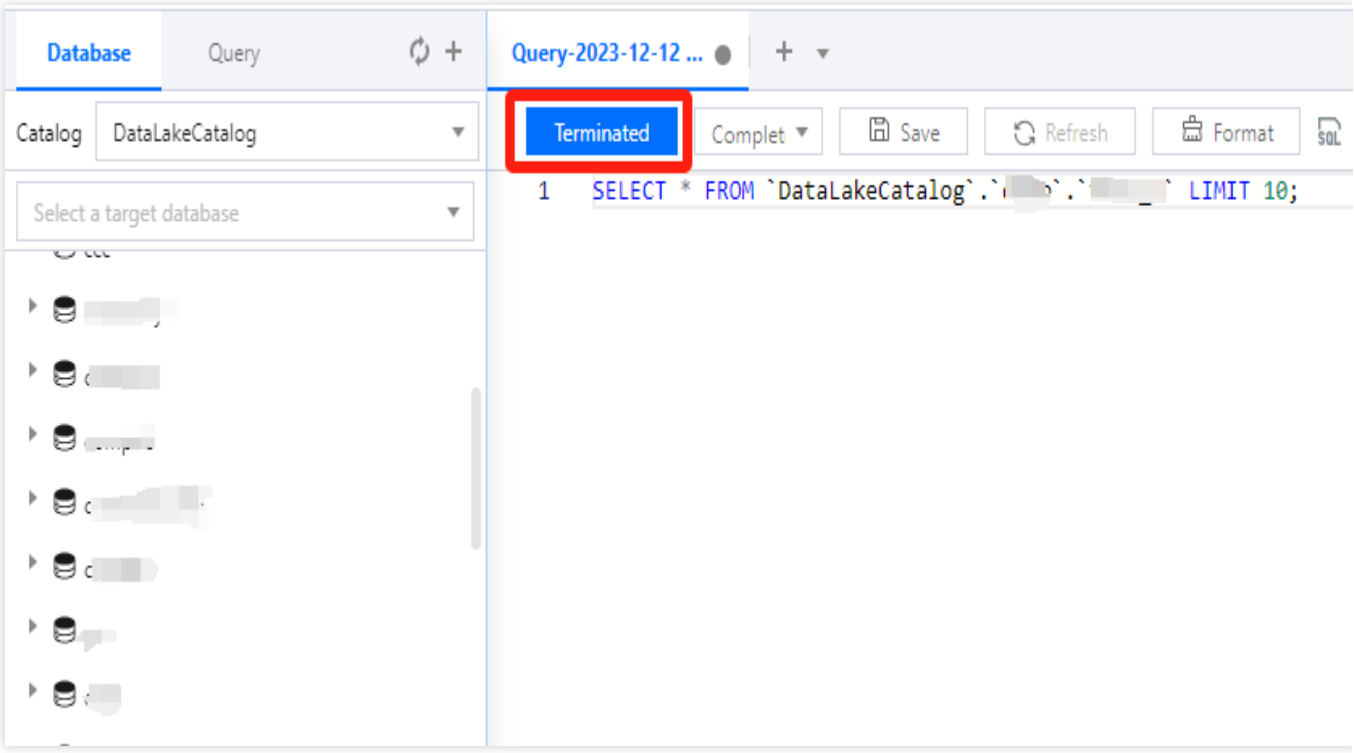
1. Select the default database and compute resource.
You can select a default database. Then, when there is no database specified in a SQL statement, the statement will be executed in the default database.
You can select a public or private cluster as the compute resource.
2. Write a standard SQL statement and click **Running**.



In Data Lake Compute, a task can run for up to 30 minutes.
Data Lake Compute is serverless, so compute resources will be scheduled temporarily. It may take longer than usual to return the result of the first DML task.
3. The query result will be displayed in the console after the task is completed.
If you exit the console page, you cannot view the query result of a historical task there again. In this case, you can view the task result file in **Run history** or the query result COS bucket you configured.

Canceling a running query task

During task running, the **Run** button becomes **Terminated**, which you can click to cancel the task. Then, Data Lake Compute will not return the query result but will calculate the scanned data volume. If you use the public engine, the scanned data volume will incur fees. For billing details, see Billing Overview.



Querying Partition Table

Last updated：2025-03-07 15:27:25

Storing data in partition catalogs can greatly reduce the scanned data volume of a computing task in Data Lake Compute and thereby significantly enhance the computing performance. The general practice of data partitioning is to store data in different catalogs by time. For example, data generated on the same day can be stored in the same catalog, and catalogs can be organized in a "year-month-day" structure. In Data Lake Compute, a table and its partitions must adopt the same data format.

Creating a Partition Table

To create a partition table, you need to specify the partition field in the table creation statement.

Adding Partitioned Data

Specifying a partition during data table creation is only to configure the partition field and doesn't allow running a query statement immediately to get data. You need to add partitioned data to a data table. If new partitioned data is added to the data catalog, you also need to add the partition information to the data table.

Manually adding a partition

Use the `ALTER TABLE ADD PARTITION` statement to add a specified partition catalog to a data table. If the partition catalog is compatible with the Hive partitioning rule (**partition column name=partition column value**), you don't need to specify the data path; otherwise, you need to refer SQL Syntax.

Sample 1: Adding a single partition catalog

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

Sample 2: Adding multi-level nested partition catalogs

```
ALTER TABLE tabel_demo ADD
PARTITION (year = '2021', month='01', day='01');
```

Sample 3: Displaying the specified 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 catalog specified during table creation. If there is a new partition catalog, the system will automatically add the partitions to the metadata of the data table. Details can be found in the SQL Syntax.Below is a sample:

```
MSCK REPAIR TABLE table_demo
```

System Restraints

`MSCK REPAIR TABLE` only adds partitions to the metadata of the data table but does not delete them. To delete an added partition, run the `ALTER TABLE table-name DROP PARTITION` statement.Details can be found in the SQL Syntax.

`MSCK REPAIR TABLE` is not recommended if the data volume is large, as the system will scan all the data, which may take a long time, cause the task to time out, and make the partition information of the data table incomplete.

A partition catalog must be compatible with the Hive partitioning rule of **partition column name=partition column value**; otherwise, use `ALTER TABLE ADD PARTITION` to load a partition.Details can be found in the SQL Syntax.

Make sure that data of a table is stored in a separate folder. For example, if the `cosn://tablea_a` data in table A and the `s3://table_a/table_b` data in table B are stored in COS and both tables are partitioned by string, then `MSCK REPAIR TABLE` will add partitions of table B to table A. To avoid this, use separate folder structures, such as `cosn://tablea_a` and `cosn://tablea_b` .

The statement may incur data read/write fees charged by COS. For more information, see Billing Overview.

Querying JSON Data

Last updated：2024-07-17 16:18:53

Query steps

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

```
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. Run a query statement to query the JSON data. Data Lake Compute supports `json_parse()`, `json_extract_scalar()`, and `json_extract()` parsing functions.

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

System restraints

The data must be in complete JSON format; otherwise, Data Lake Compute cannot parse it.

A data row cannot contain a line break, and the JSON format cannot be optimized visually; for example:

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

Data Lake Compute will automatically recognize the first JSON level as the attribute column of a data table and recognize other nested structures as corresponding attribute values.

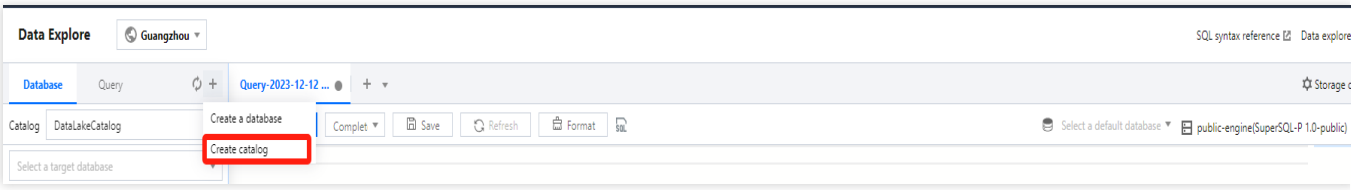
Querying Data from Other Sources

Last updated：2025-01-03 15:40:27

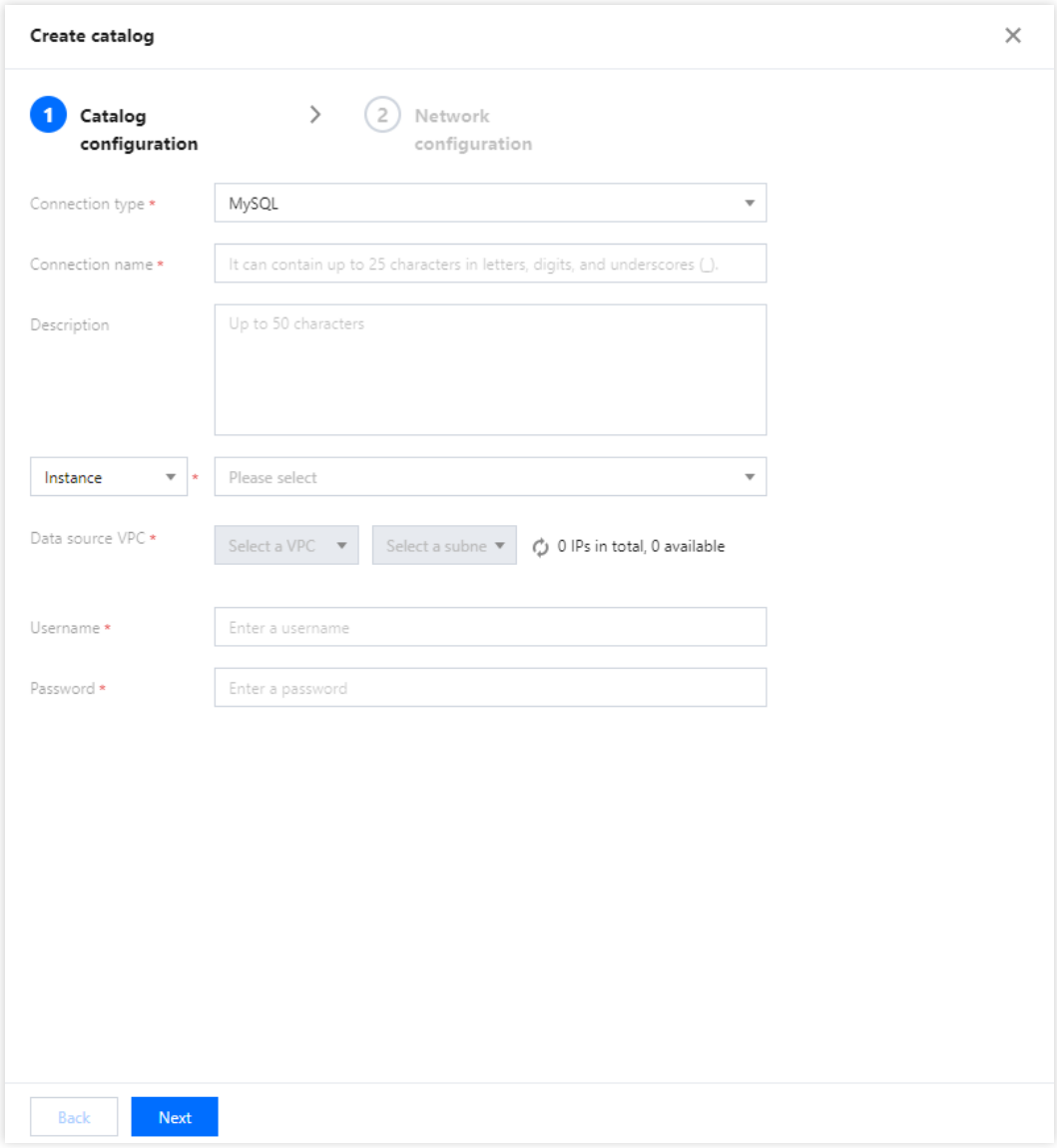
Data Lake Compute allows you to query and analyze data in an external table. Currently, data from MySQL and EMR Hive can be connected to it. You can add and manage other data sources in the Data Lake Compute console.

Adding a data source

1. Log in to the Data Lake Compute console and select the service region. You need to have the permission to create data catalogs.
2. Select **Data Explore** on the left sidebar, hover over **+**, and click **Create data catalog**.



3. Select the data source type. Currently, MySQL and EMR Hive are supported. Before configuring MySQL, you need to add the Data Lake Compute subnet to the database's allowlist. Two configuration methods are supported: database instance and JDBC connection.



Supported EMR Hive versions are 2.0.1, 2.1.0, 2.2.0, 2.2.1, 2.3.0, 2.4.0, 2.5.0, 2.5.1, and 2.6.0. The configuration is performed through the EMR access address.

4. Enter the data source information and click **Create connection**.

Note：

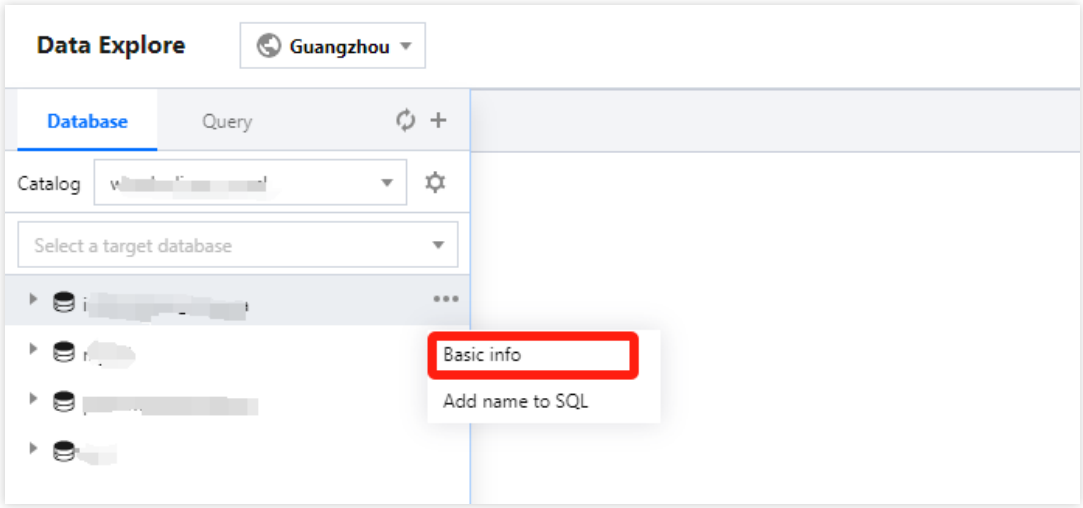
A data engine must be bound to the network configuration of the VPC where the data source resides. You can view the bound data engine during creation or create a network configuration and bind the data engine. For more information about network configuration, see Engine Network Configuration.

Managing Data

Currently, Data Lake Compute allows you to **view the database information of** and **preview data in** external tables.

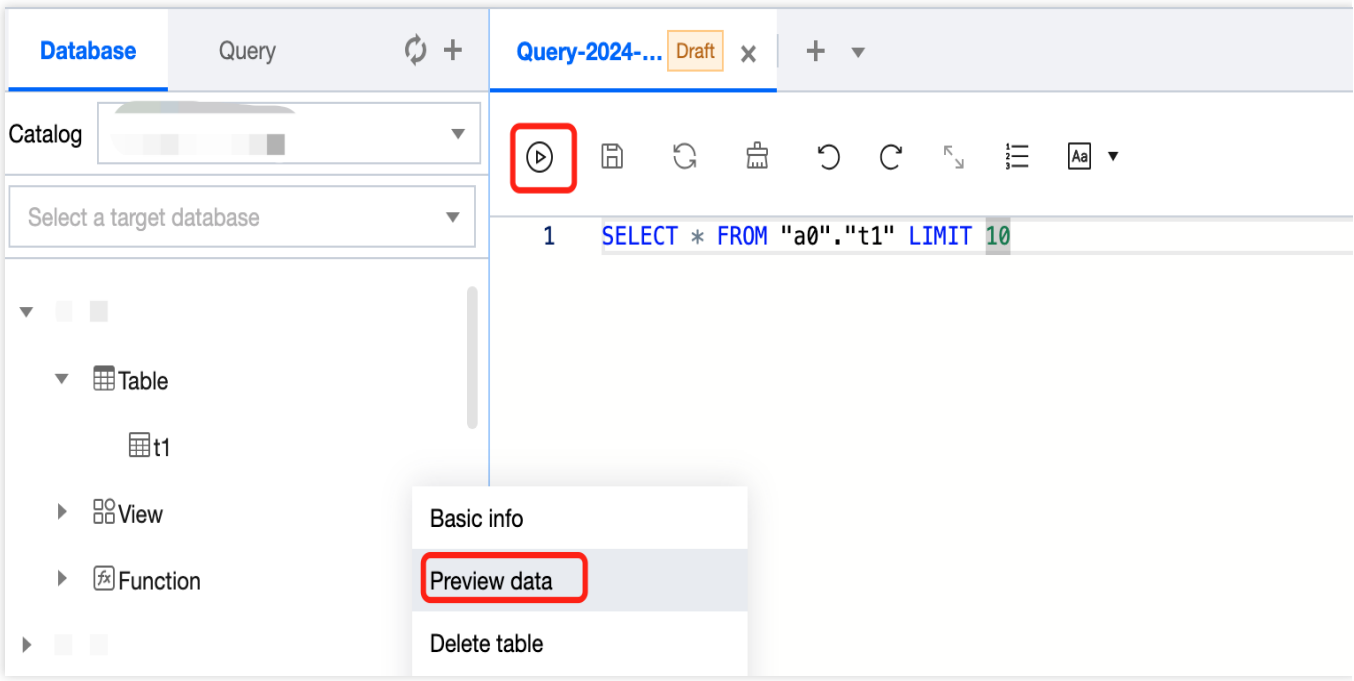
Viewing database information

1. Log in to the Data Lake Compute console and select the service region. You need to have the permission to view data tables.
2. Select **Data Explore** on the left sidebar, hover over **+**, and click **Basic info**. You can view the basic information of a data table in the pop-up window.



Previewing data in a data table

1. Log in to the Data Lake Compute console and select the service region. You need to have the permission to view data tables.
2. Select **Data Explore** > **Data table**, hover over **...**, and click **Preview data**. Then, you can run a SQL statement to query and display data in the data table.



Note：

Select the data engine bound to the network configuration of the VPC of the data source.

Using View

Last updated：2025-01-03 15:27:27

In Data Lake Compute, a view is a logical table rather than a physical table. Whenever a view is referenced during a query, the query that defines the view will be executed. You can create a view through `SELECT` and reference it in future queries.Details can be found in the SQL Syntax.

System restraints

A view name is case-insensitive and can contain up to 128 letters and underscores.

Data Lake Compute doesn't support managing data access permissions through views.

INSERT INTO

Last updated：2024-07-17 16:23:11

The `INSERT INTO` statement can insert a `SELECT` query result in the source table to the target table as a new row.

Querying Script Parameters

Last updated：2024-07-17 16:23:47

Data Lake Compute allows you to configure date parameters to facilitate queries with scripts.
Data Lake Compute adopts the standard date format of `yyyymmddhh24miss` and uses the `${}` command to set a date as a variable consisting of the date and time.
Date: It can be in any date format or a predefined system variable, such as `yyyymmdd`, `yyyymm`, `yyyy-mm-dd`, `yy`, and `dataDate`.
Time: It can be +/-N cycles and supports `N/Nd`, `Nm`, `Nw`, `Nh`, and `Nmi`. It is compatible with various calculation formulas, such as `7*N` and `N/24`.
Examples

+/- N Cycle	Method	Compatible Format	Example
N years later	${yyyymmdd+Ny}	-	-
N years ago	${yyyymmdd-Ny}	-	One year ago: ${yyyymmdd-12m}: 20190920
N months later	-	${yyyymmdd+Nm}	-
N months ago	${yyyymmdd-Nm}	$[add_months(yyyymmdd,-N)]	${yyyymmdd-1m}: 20200820 ${yyyymm}: 202009 ${dataDate-1m}: 20200820
N weeks later	${yyyymmdd+Nw}	${yyyymmdd+7*N}	-
N weeks ago	${yyyymmdd-Nw}	${yyyymmdd-7*N}	-
N days later	${yyyymmdd+N/Nd}	-	-
N days ago	${yyyymmdd-N/Nd}	-	${yyyymmdd-1}, ${dataDate-1}
N hours later	${yyyymmddhh24+Nh}	$[yyyymmddhh24+N/24]	-
N hours ago	${yyyymmddhh24-Nh}	$[yyyymmddhh24-N/24]	${yyyymmddhh24-1h}: 2020092014

			${dataDate-1h}: 2020092014
N minutes later	${yyyymmddhh24mi+Nmi}	$[yyyymmddhh24+N/24/60]	-
N minutes ago	${yyyymmddhh24mi-Nmi}	$[yyyymmddhh24-N/24/60]	${yyyymmddhh24mi-10mi}, ${dataDate-10mi}

Note:

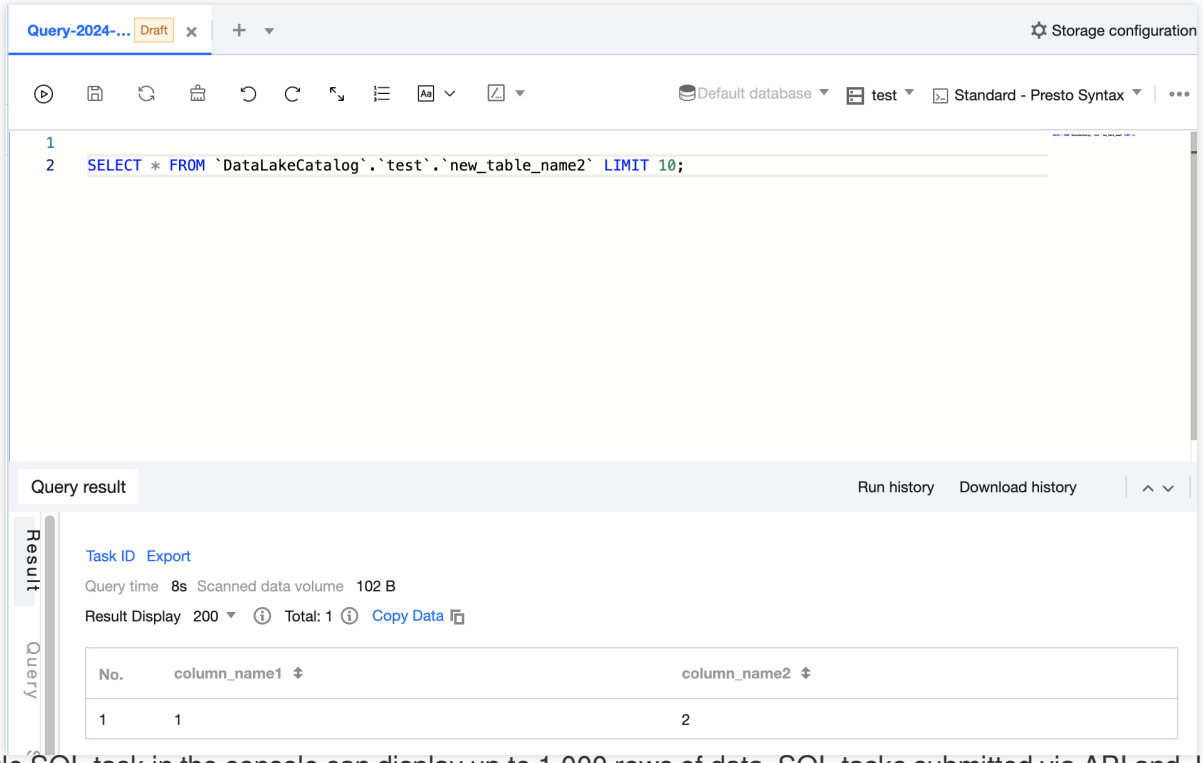
Make sure that the variable or the part before `+/-` in the variable is in line with the standard date format; otherwise, the system cannot recognize and use it.

Obtaining Task Results

Last updated：2024-09-18 17:59:35

Using the Query Editor to Obtain Task Results

When you use the DLC console for task queries, the query results will be displayed in real-time below the editor.



A single SQL task in the console can display up to 1,000 rows of data. SQL tasks submitted via API and JDBC are not subject to this limitation.

You can view the query history for a single Session for up to 3 months by checking the running history. For more methods to query historical records, see History.

Output Format Configuration for Task Results

The results of data exploration are saved in CSV format by calling Spark's DataFrame.write. If the engine version is released later than April 2023, you can configure the output format of the exploration results.

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

Parameter	Default Value	Remark
livy.sql.result.format.option.sep livy.sql.result.format.option.delimiter	,	The separator bet the result is storec

		comma by default
livy.sql.result.format.option.encoding livy.sql.result.format.option.charset	UTF-8	String encoding fo For example: UTF 8859-1, UTF-16B UTF-16.
livy.sql.result.format.option.quote	\\"	Specifies whether quotation marks, v of escape charact
livy.sql.result.format.option.escape	\\\\	Escape character of escape charact
livy.sql.result.format.option.charToEscapeQuoteEscaping		The characters th within quotation m
livy.sql.result.format.option.comment	\\u0000	Remark informatio
livy.sql.result.format.option.header	false	Specifies whether
livy.sql.result.format.option.inferSchema	false	Infers the data typ not inferred, all co strings.
livy.sql.result.format.option.ignoreLeadingWhiteSpace	true	Ignores leading en
livy.sql.result.format.option.ignoreTrailingWhiteSpace	true	Ignores trailing en
livy.sql.result.format.option.columnNameOfCorruptRecord	_corrupt_record	The name for the converted. This pa by spark.sql.columnN with table configur precedence.
livy.sql.result.format.option.nullValue		Specifies the stora values. The defau which case it can emptyValue types
livy.sql.result.format.option.nanValue	NaN	The storage forma values.
livy.sql.result.format.option.positiveInf	Inf	The storage forma
livy.sql.result.format.option.negativeInf	-Inf	The storage forma

livy.sql.result.format.option.compression or codec		The class name o algorithm. By defa applied. Short nar gzip, lz4, and sna
livy.sql.result.format.option.timeZone	System default time zone	The default time z spark.sql.session. example, Asia/Sh configuration take
livy.sql.result.format.option.locale	en-US	Specifies the lang
livy.sql.result.format.option.dateFormat	yyyy-MM-dd	The default forma
livy.sql.result.format.option.timestampFormat	yyyy-MM-dd'T'HH:mm:ss.SSSXXX	The default forma LEGACY mode, it yyyy-MM-dd'T'HH
livy.sql.result.format.option.livy.sql.result.format.option.multiLine	false	Allows multiple lin
livy.sql.result.format.option.maxColumns	20480	The maximum nur
livy.sql.result.format.option.maxCharsPerColumn	-1	The maximum nur column. -1 means
livy.sql.result.format.option.escapeQuotes	true	Escapes quotatio
livy.sql.result.format.option.quoteAll	quoteAll	Encloses the entir marks when writir
livy.sql.result.format.option.emptyValue	\\"\\"	The format used f empty values.
livy.sql.result.format.option.lineSep		The newline chara separation.

2. Configure the output format to a non-CSV format. Note that in this case, the console will not be able to display the results. However, you can read the result path using other methods. For details on where the result path is saved, see the next section.
The configuration option livy.sql.result.format supports saving in formats such as text, ORC, JSON, and Parquet.

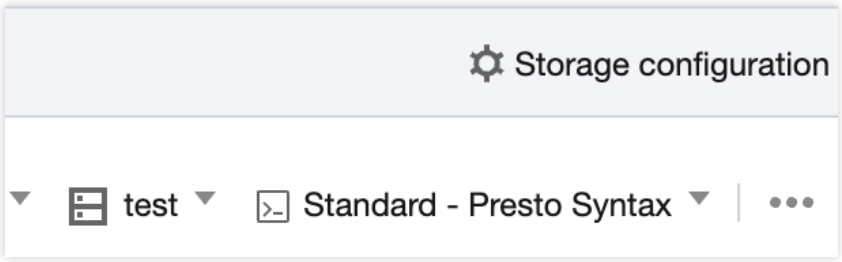
Task Result Storage Location Configuration

Note: The Standard Engine - Presto is not supported. Full results can be obtained via JDBC.

DLC supports automatically saving query results to a COS path or DLC's managed storage through configuration. The configuration steps are as follows:

1. Log in to the DLC console, select the service region, and ensure that the login account has necessary COS-related permissions.

2. Go to the **Data Exploration Page**, click **Storage Configuration** in the upper right corner, and configure the settings for saving query results.



3. You can save the results to DLC's managed storage or COS. If you want to configure the path to COS, the operating account should have necessary COS-related permissions. Data storage fees will be based on COS pricing. The task results are stored in subfolders under the following COS path:

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

COS directory path: This is the COS directory path configured in the system settings.

/yyyy/mm/dd: The directory is organized based on the task execution date.

/data: This directory stores the query result data, with files in CSV format. DLC may generate multiple data files.

/meta: This directory stores the metadata for the queried data tables, with files in JSON format.

Note:

Storing SELECT query results in DLC's internal storage, with Cloud Object Storage as the underlying storage, and the results are retained for 36 hours.

When SELECT query results are stored in your COS bucket path, ensure that you have necessary COS-related permissions.

Downloading Task Results

Note: The Standard Engine - Presto is not supported. Full results can be obtained via JDBC.

DLC allows users to manually download query results to their local devices. If full result mode is not enabled, users can download the results of tasks with available query results to their local devices or manually save them to COS

(COS permissions are required).
The data downloaded or saved to COS correspond to the query results of the current SQL task, with a maximum of 500 results.
The maximum size for the local download is 50 MB.
If the results are configured to be saved to COS, they will be automatically stored in the COS path without the need for manual downloads.

Query Script Analysis

Last updated：2024-08-07 17:08:48

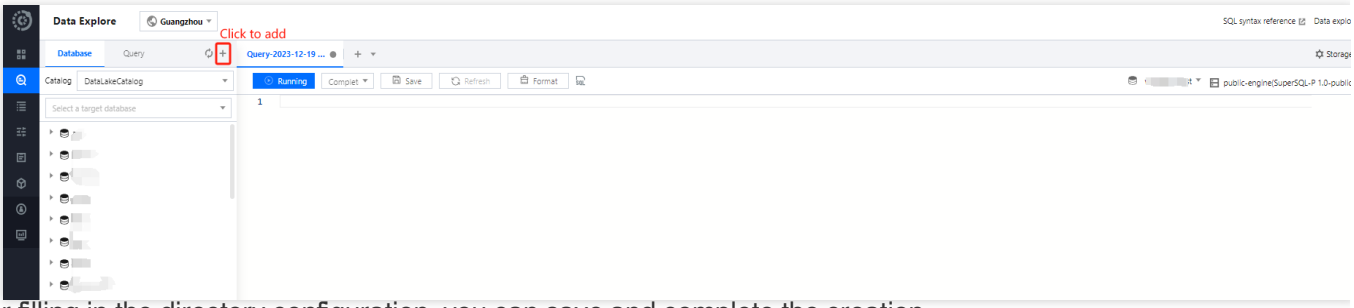
To facilitate users in quickly handling repetitive query tasks, DLC provides script file analysis.

Note

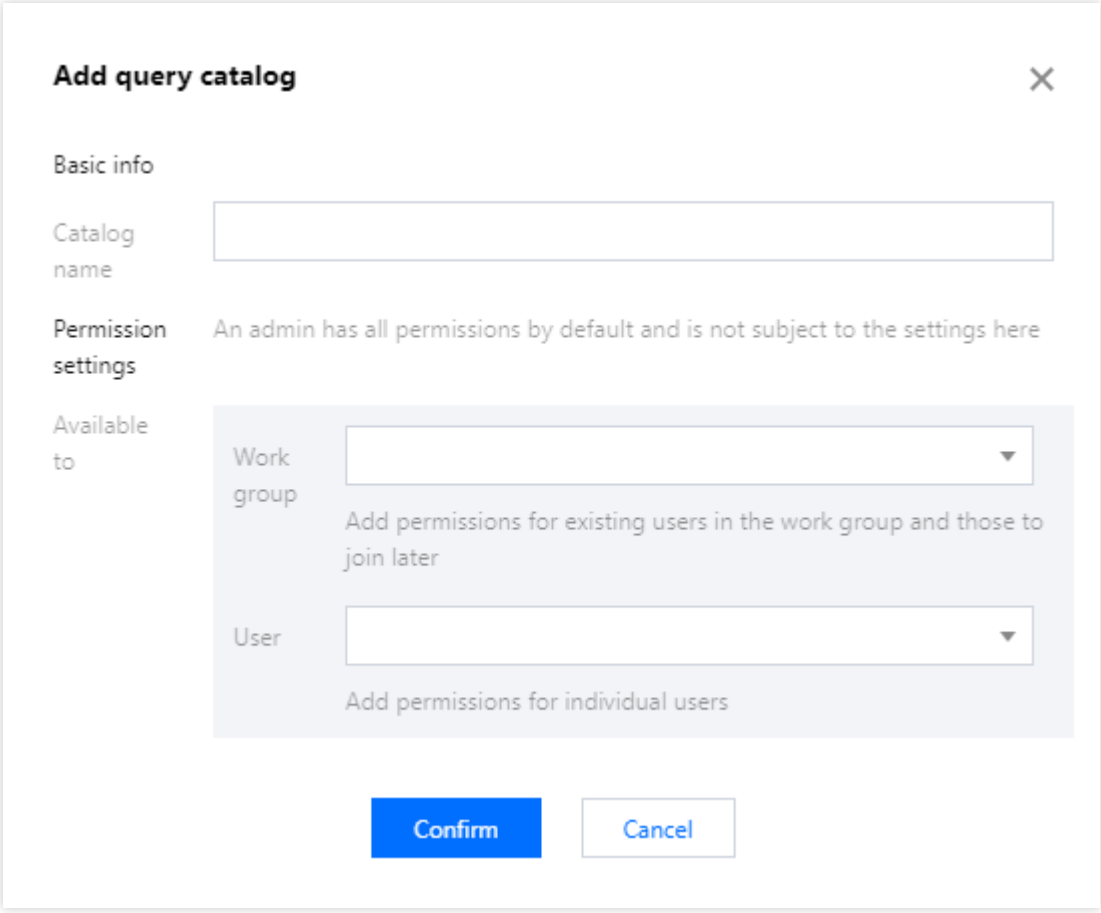
The console allows saving up to 100 SQL scripts.

Creating a New Query Directory

1. Log in to DLC Console > Script Query Page.
2. On the query page, click Add Query Directory.



3. After filling in the directory configuration, you can save and complete the creation.

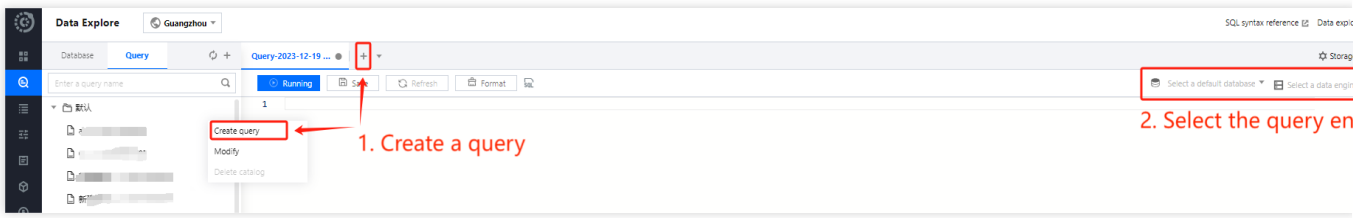


Directory name: Supports Chinese characters, letters, and underscores (_), up to 25 characters.
Permission settings: You can set the visibility permissions for the script directory and the scripts within it based on the perspective of the workgroup or user.

Creating a New Query Script

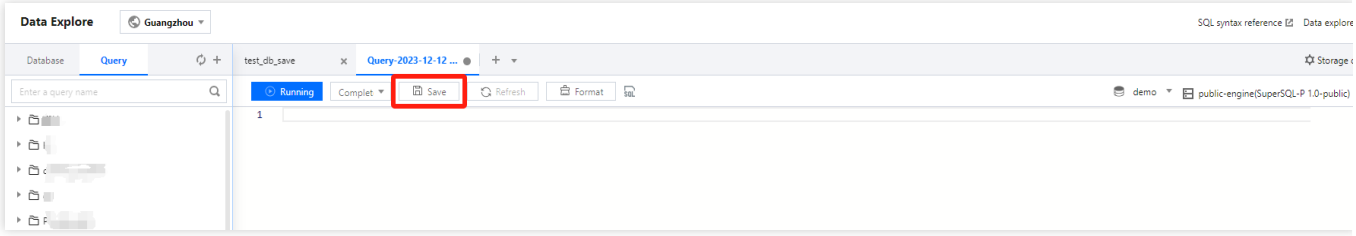
1. Log in to DLC Console > Script Query Page, You can click the library


icon or directly add execution and save.
2. After the computation engine is selected, click Run to execute the script.

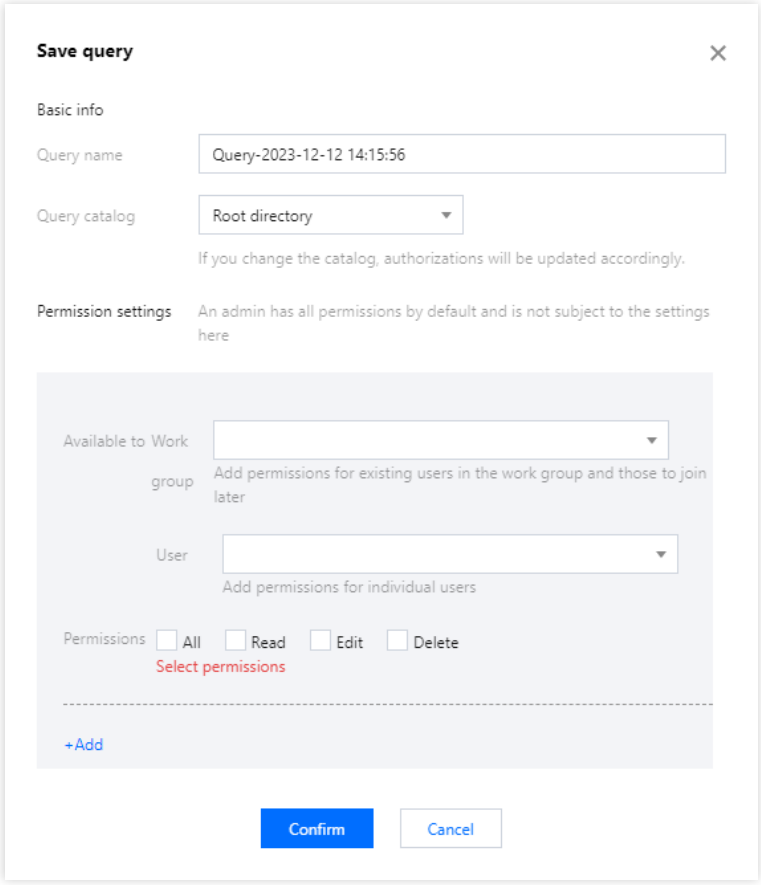


Saving a Query Script

1. After the query is completed, click the Save button.
2. Queries created through the library will be saved under the directory of that library. Queries added through the tab bar can be saved directly in the root directory or an authorized library.

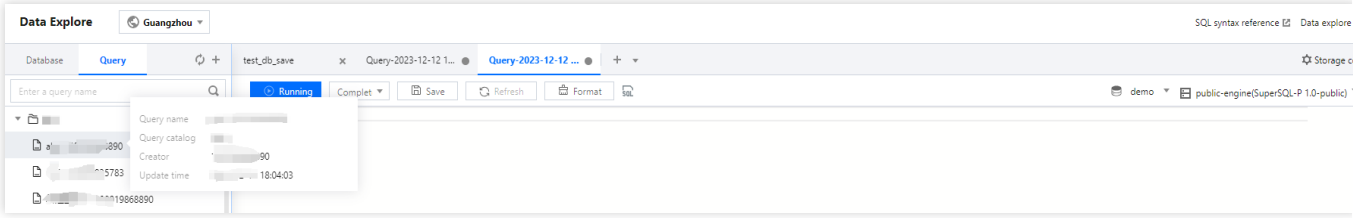


3. Query table permissions can be customized according to the public scope of the library, and table usage permissions can be specified for the public scope.



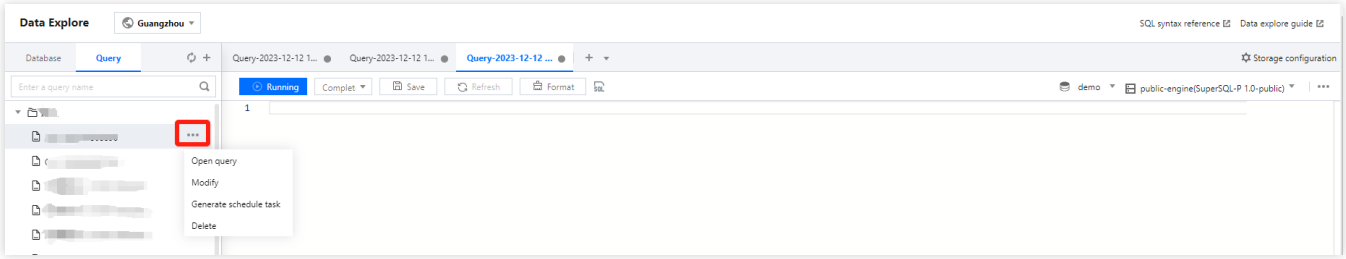
Viewing script information

1. Hover the mouse pointer over the script name to view the script details.



2. Click the

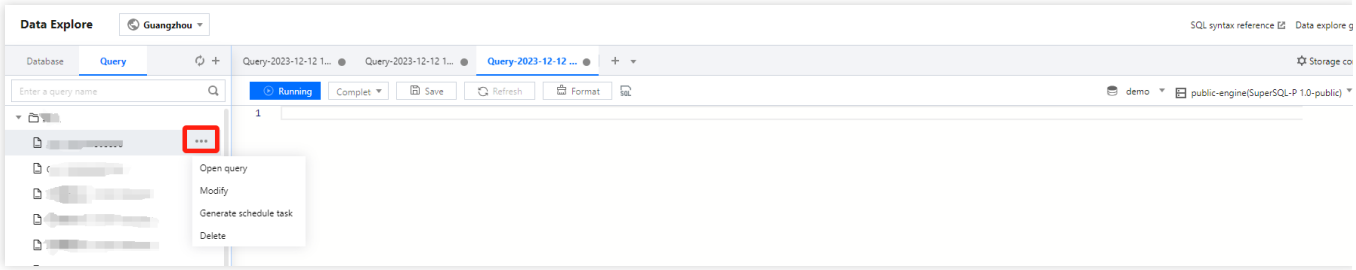
icon next to the table you want to view, and select to open or query it.



Deleting a Query Script

Click the

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



Note:

Deleted scripts cannot be restored. Operate with caution.

Data Job Overview

Last updated：2024-07-17 16:36:54

Data Lake Compute provides Spark-based batch and flow computing capabilities for you to perform complex data processing and ETL operations through data jobs.
Currently, data jobs support the following versions:
Scala 2.12
Spark 3.1.2

Preparations

Before starting a data job, you need to create a data access policy to ensure data security as instructed in Configuring Data Access Policy.
Currently, only CKafka data source is supported for data job configuration, with more data sources to come in the future.

Billing mode

A data job is billed by the data engine usage. Currently, pay-as-you-go and monthly subscription billing modes are supported. For more information, see Data Engine Overview.
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 don't 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 can’t run data jobs on a SparkSQL data engine.

Job management

On the **Data job** management page, you can create, start, modify, and delete a data job.
1. Log in to the Data Lake Compute console and select **Data job** on the left sidebar.
2. Click **Create job**. For detailed directions, see Creating Data Job.

3. In the list, you can view the current task status of the data job. You can also manage the job as instructed in Managing Data Job.

Configuring Data Access Policy

Last updated：2024-07-17 17:44:52

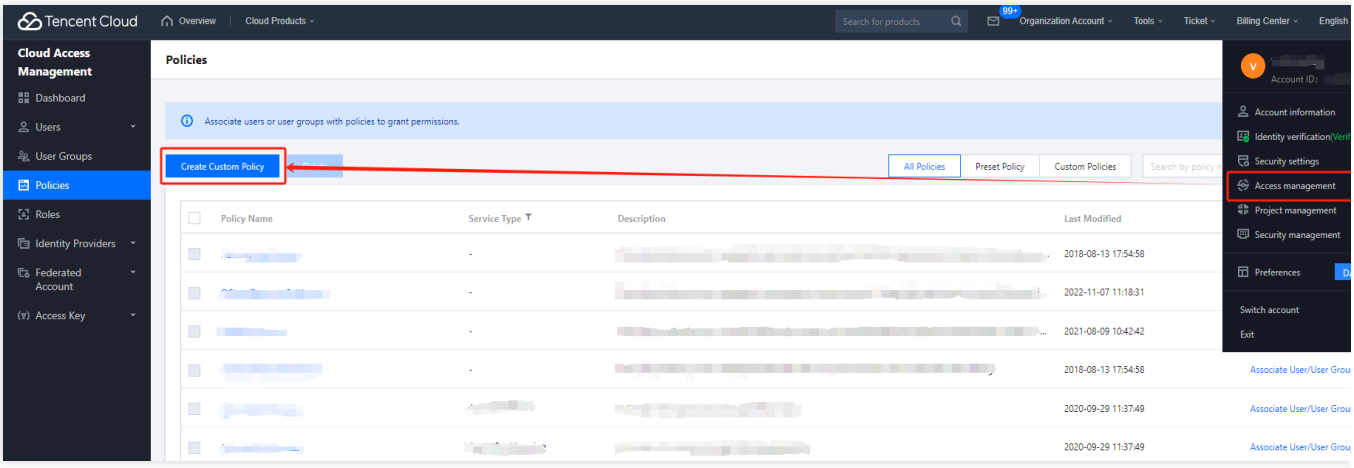
Data Access Policy (CAM role arn) Overview

A data access policy (CAM role arn) allows you to configure permissions in CAM for accessing data in data sources and COS during data job execution.
When configuring a data job in Data Lake Compute, you need to specify the data access policy to protect data security.

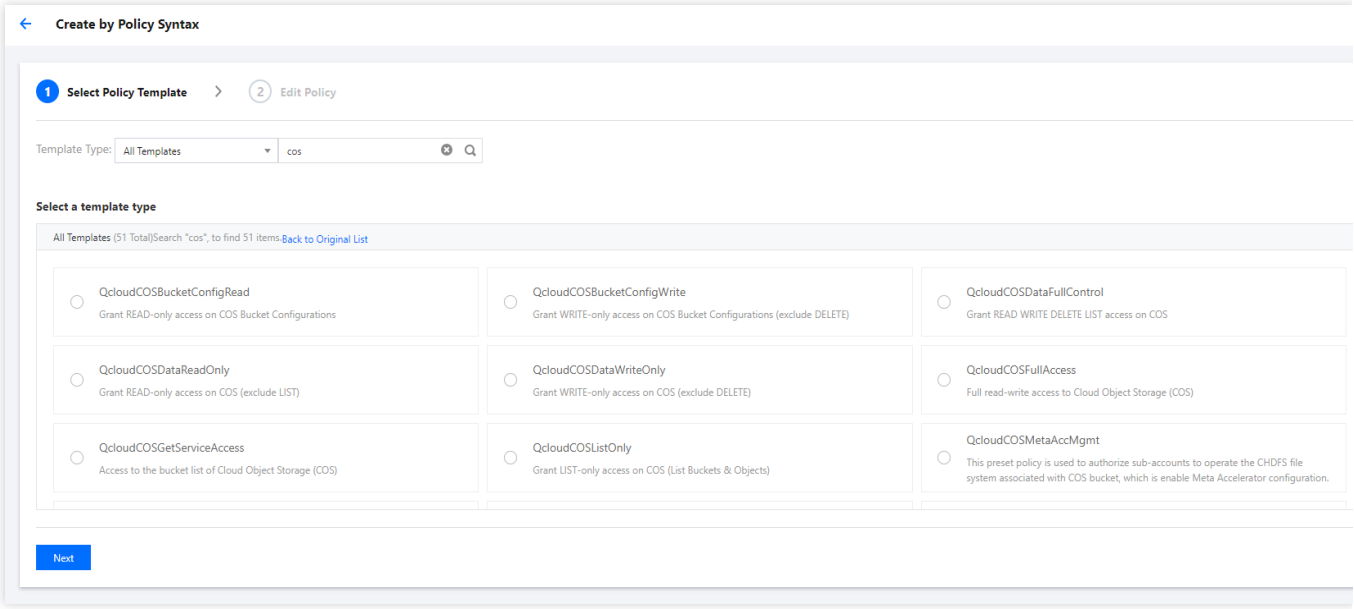
Directions

Step 1. Create a policy in CAM

1. Log in to the Tencent Cloud console and select **Cloud Access Management**. The logged-in account needs to have permissions to configure CAM; therefore, we recommend you use a root account or admin account.
2. Select **Policies** on the left sidebar to enter the policy management page. Click **Create Custom Policy** and select **Create by Policy Syntax**.



3. Search for COS in the policy template and select **COS permission templates**.



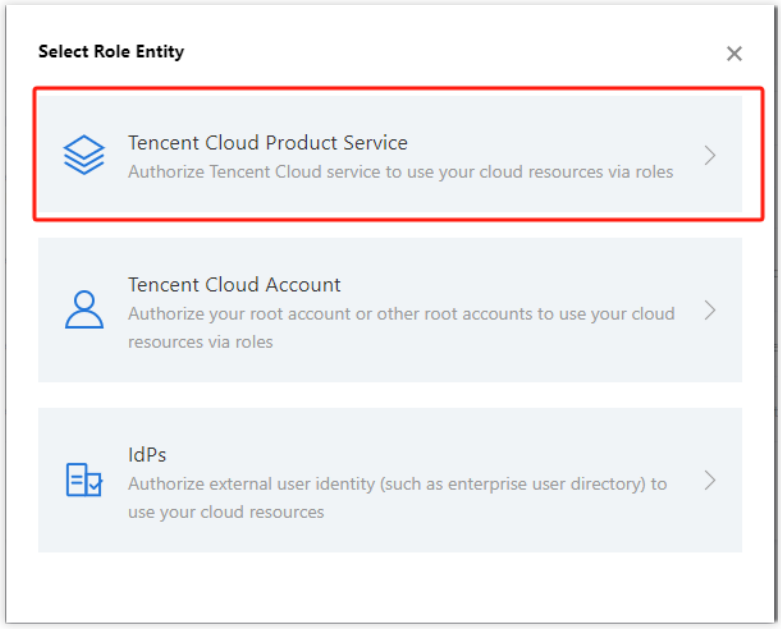
The preset templates define read-only and read/write permission policies. If they don't meet your needs, create a custom policy template as instructed in Appendix.

4. Select the template, set a name for the policy, and click **Save**.

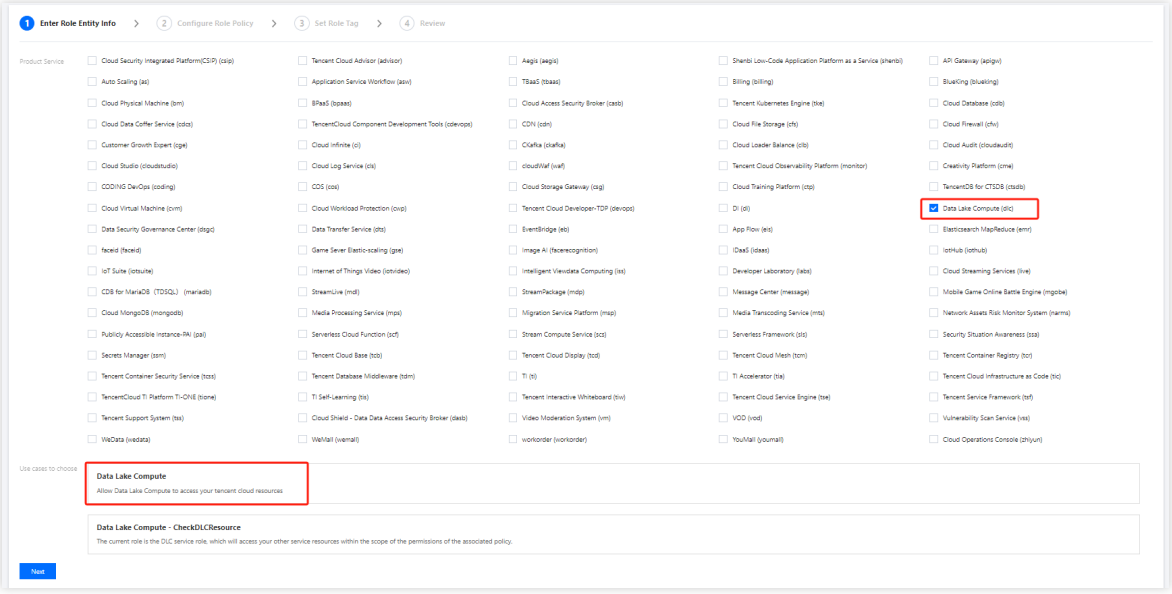
Step 2. Create a service role

1. Log in to the Tencent Cloud console and select **Cloud Access Management**. The logged-in account needs to have permissions to configure CAM; therefore, we recommend you use a root account or admin account.

2. Select **Role** on the left sidebar to enter the role management page. Click **Create Role** and select **Tencent Cloud Product Service**.



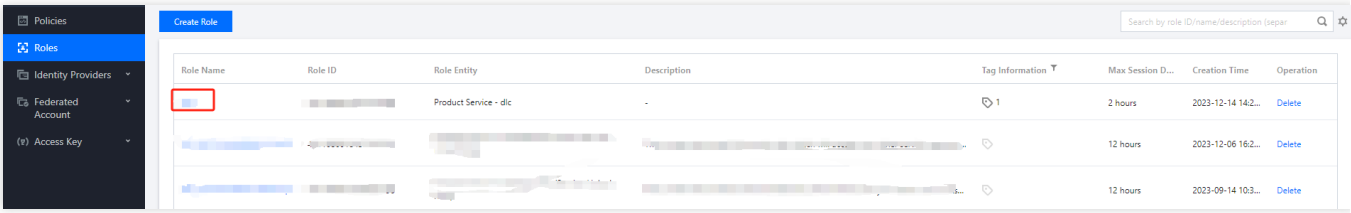
3. In the **Role Entity** service list, find and select **Data Lake Compete** and click **Next**.



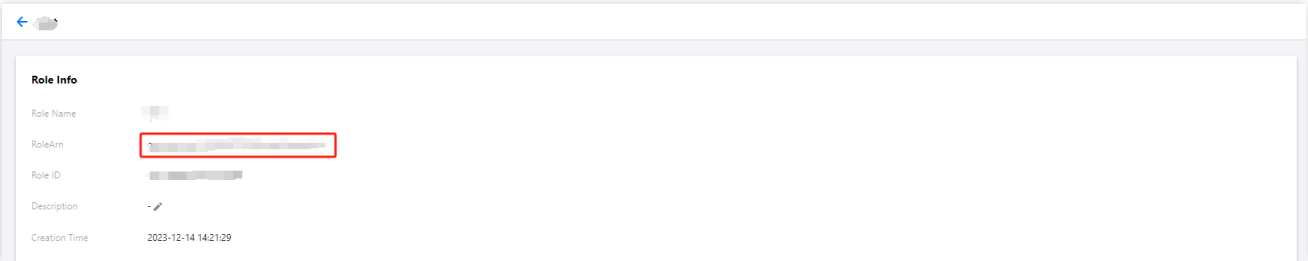
4. In the policy configuration, find and select the policy created in Step 1 and click **Next**.
5. Set a name for the role and click **Save**.

Step 3. Get the role arn information

1. After creating the role in Step 2, return to the role list and find the created role.
2. Click **Role Name** to enter the role details page.



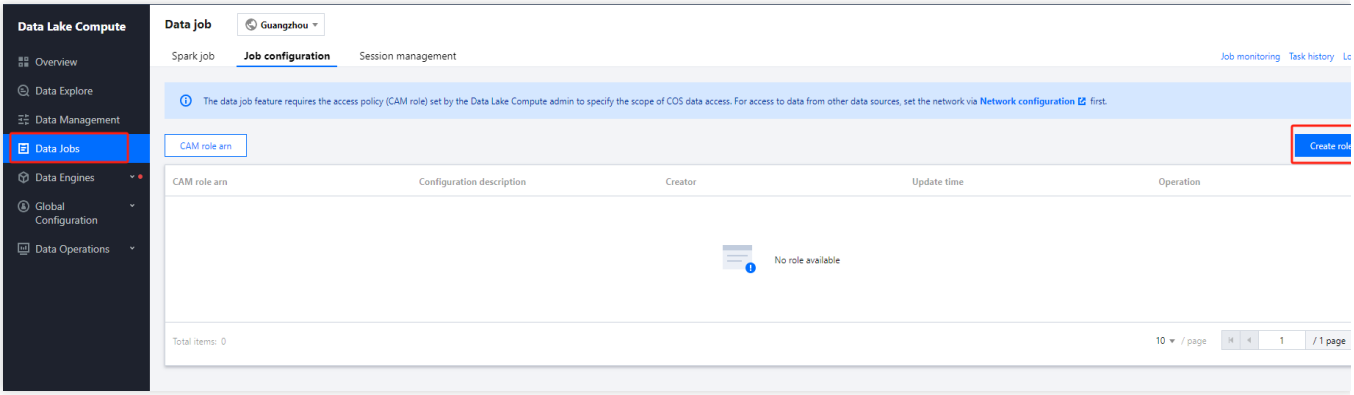
3. Find and copy the role arn information.



Step 4. Configure the role arn in Data Lake Compute

1. Log in to the Data Lake Compute console with an admin account.

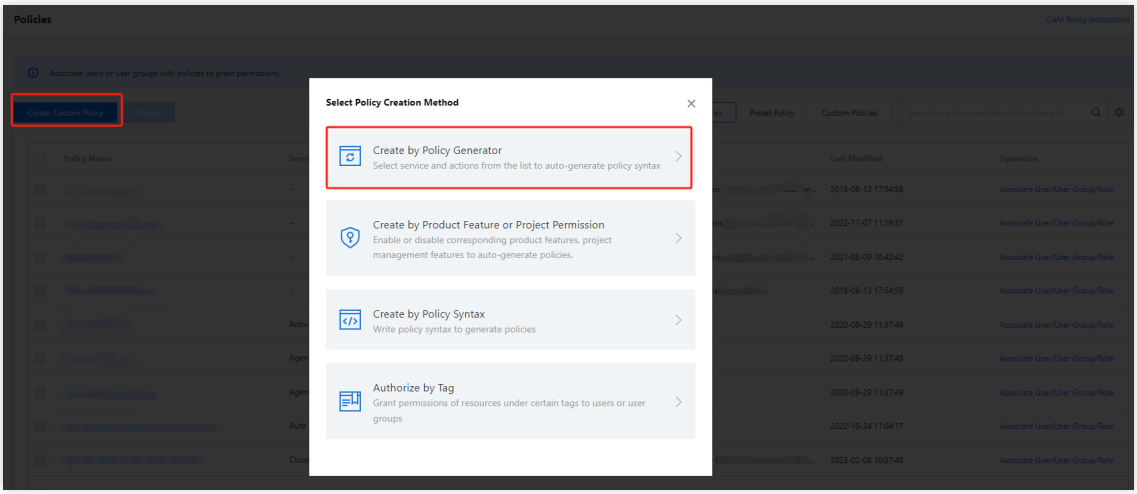
2. Select **Data job** on the left sidebar to enter the data job management page. Click **Job configuration** and select **CAM role arn**.
3. Click **Create role arn**.



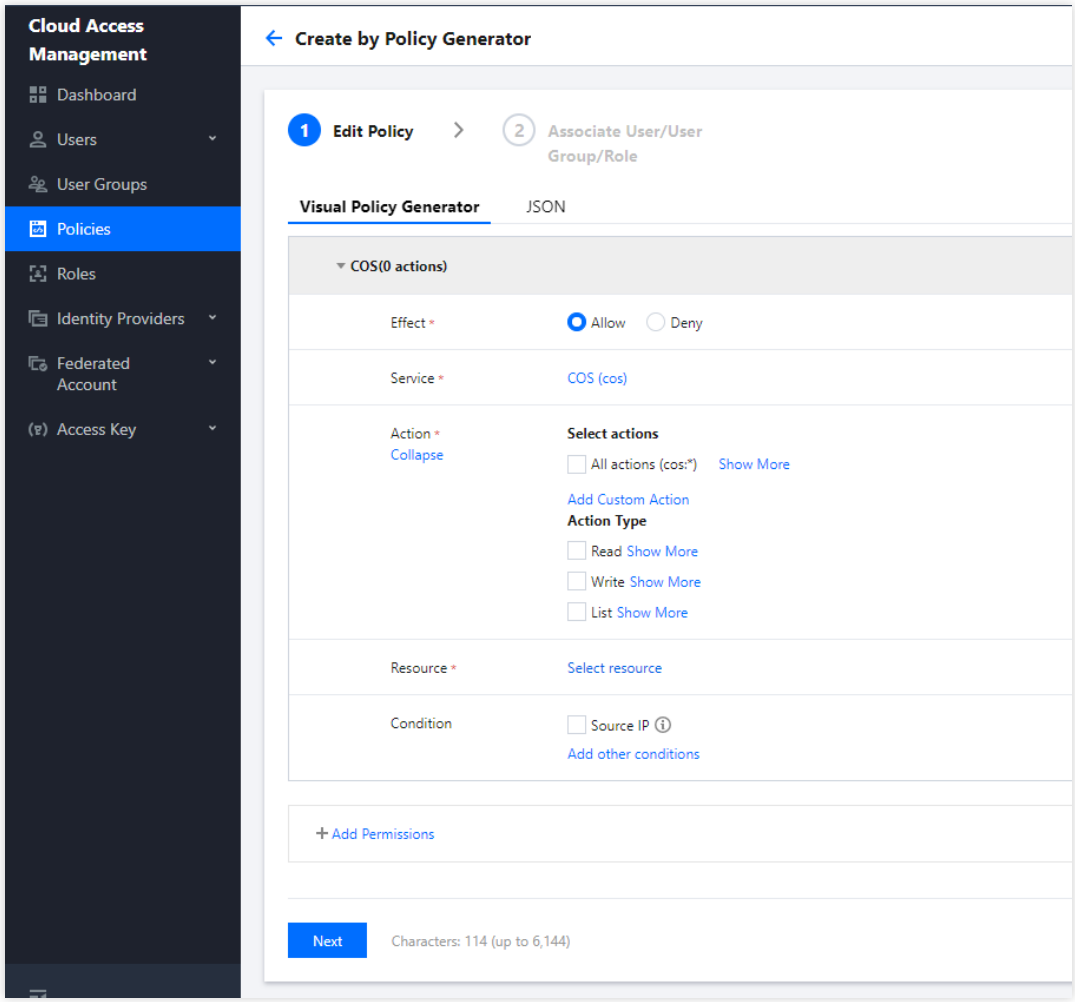
4. Paste the role arn information obtained in Step 3 in the input box and click **Save**.

Appendix: Custom Policy Template

If the preset templates cannot meet your data management needs, you can configure a custom template in the following steps.
1. Log in to the Tencent Cloud console and select **Cloud Access Management**. The logged-in account needs to have permissions to configure CAM; therefore, we recommend you use a root account or admin account.
2. Select **Policies** on the left sidebar to enter the policy management page. Click **Create Custom Policy** and select **Create by Policy Generator**.



3. Select **Allow** as **Effect** and **COS** as **Service**. Select the resource scope as needed.



If you need to manage specific resources, click **Add a six-segment resource description** to add resources. You can use `*` to indicate all the resources. For more information, see Resource Description Method.

4. After completing the configuration, set a name for the policy and click **Save**. You can also select **Authorized Users** to authorize the policy to existing users.

Creating Data Job

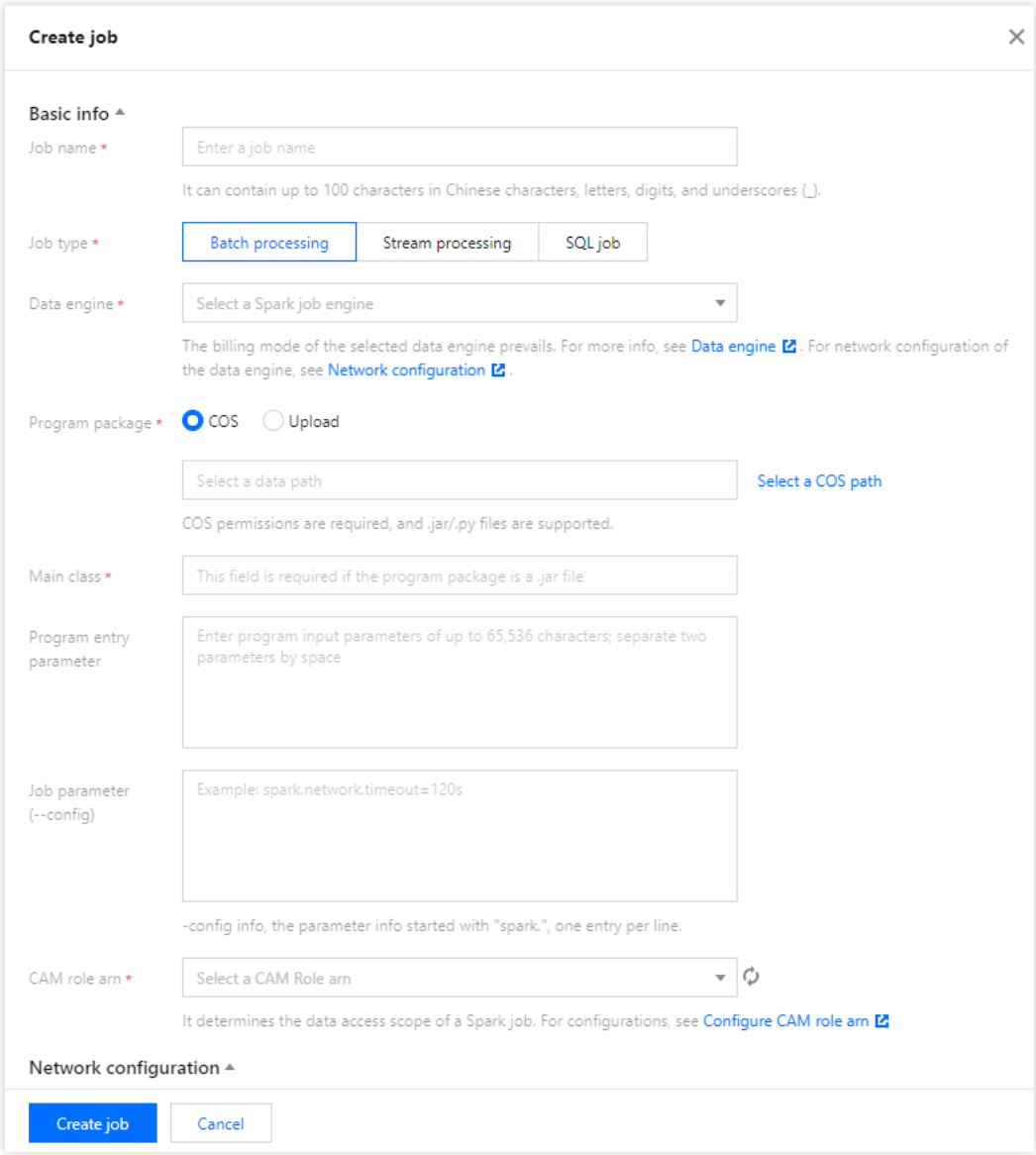
Last updated：2024-07-17 17:45:32

Preparations

Before creating a data job, you need to configure the CAM role arn to secure the data access from the data job. For detailed directions, see Configuring Data Access Policy.

Directions

1. Log in to the Data Lake Compute console and select **Data job** on the left sidebar.
2. Click **Create job**.



Configure parameters as follows:

Parameter	Description
Job name	It can contain up to 40 letters, digits, and underscores.
Job type	**In batch**: Batch data jobs based on Spark JAR **In flow**: Flow data jobs based on Spark Streaming
Data source connection	Data source for **In batch** data jobs. Currently, it can only be CKafka, which needs to be configured in advanced in **Job configuration**.
Data engine	It can be a Spark job data engine for which you have the permission. If you select **Data source**, you can only select a data engine connected to the data source.
Program package	The JAR format is supported.

	You can select a local file of up to 5 MB in size or a file in COS. If the local file exceeds 5 MB, upload it to COS for use. You can directly enter a COS path.
Dependency JAR resource	The JAR format is supported. You can select multiple resources. You can select a local file of up to 5 MB in size or a file in COS. If the local file exceeds 5 MB, upload it to COS for use. You can directly enter multiple COS paths and separate them by semicolon.
Dependency file resource	You can select a local file of up to 5 MB in size or a file in COS. If the local file exceeds 5 MB, upload it to COS for use. You can directly enter multiple COS paths and separate them by semicolon.
CAM role arn	The data access policy configured in **Job configuration**, which specifies the scope of data accessible to a data job. For more information, see Configuring Data Access Policy.
Main class	JAR package parameter in the main class. Separate multiple parameters by space.
Job parameter	`--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
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 MEM Billable CUs = executor resource * executor quantity + driver resource Pay-as-you-go data engines are billed by the billable CUs.

3. After configuring the parameters, click **Save**.

Managing Data Job

Last updated：2025-03-07 15:27:25

This document describes how to manage a data job.

Edit a data job.

Start and stop a data job task.

View the data job and task details.

Delete a data job.

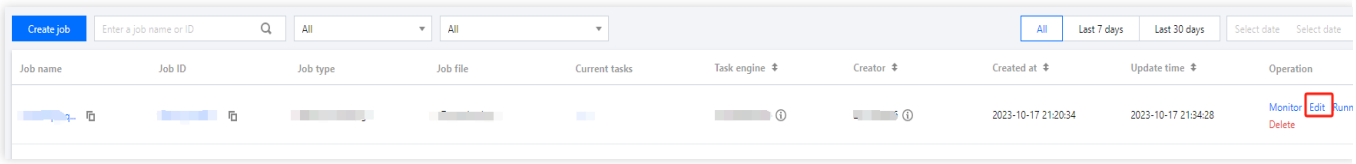
Editing a data job

Note:

A running data job cannot be edited.

The type of a data job cannot be changed. To change it, create a new data job as instructed in Creating Data Job.

1. Log in to the Data Lake Compute console, select the service region, and select **Data job** on the left sidebar.
2. Find the target data job and click **Edit**.



3. Edit the content and click **Save**.

Starting and stopping a data job task

You can start and stop a created data job to generate corresponding tasks. A data job can generate multiple task instances and be executed multiple times.

Data task statuses are as follows:

Status	Description
Not started	Initial status after creation.
Running	The data task is running, during which the data job cannot be edited or deleted.
Successful	The task is executed successfully.
Failed	Failed to run the task. You can query the error message through the log or SparkUI.

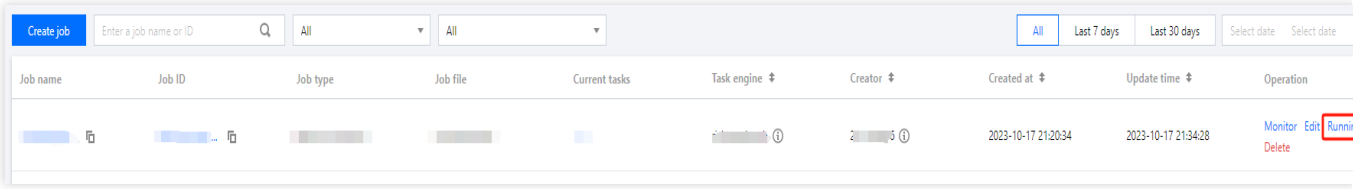
Canceled	The task is manually canceled.

You can start and stop a data job task in the following steps:

1. Log in to the Data Lake Compute console, select the service region, and select **Data job** on the left sidebar.
2. Find the target data job and click **Start** or **Stop** to change the task status.

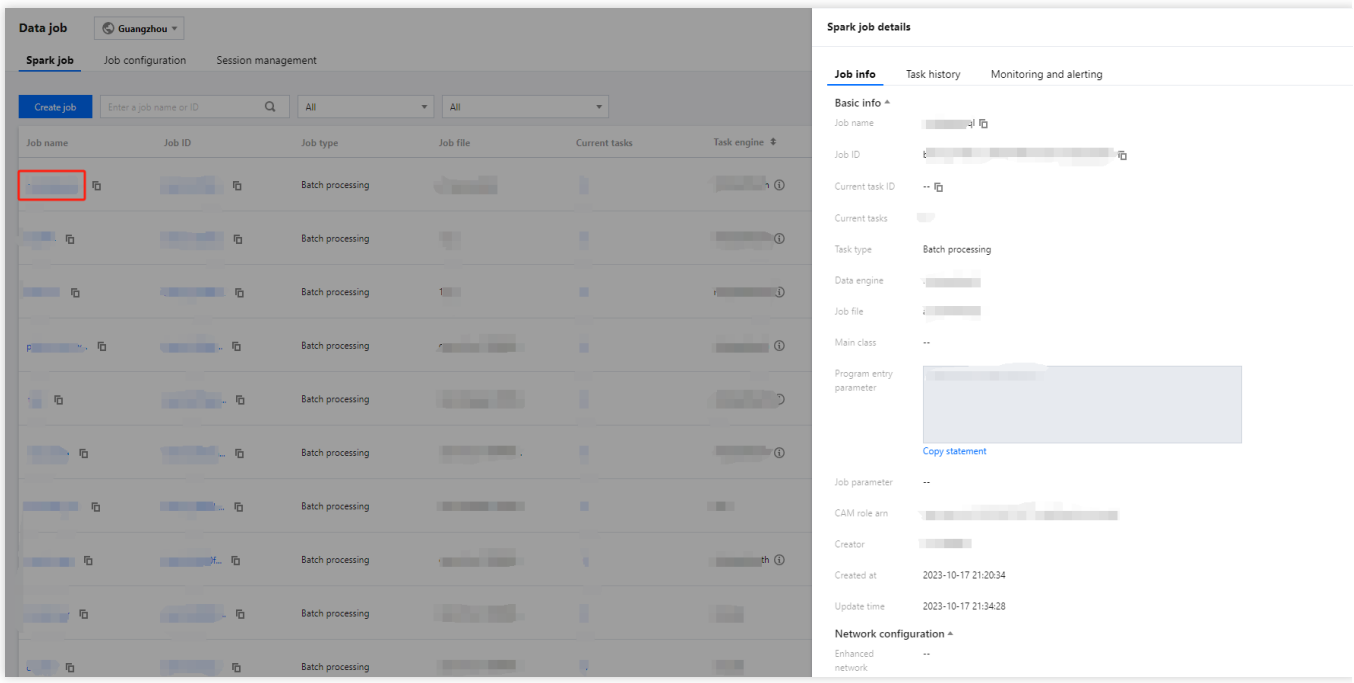
Note:

Starting a task instance will use compute engine resources. If the usage exceeds the configured upper limit, the task will be put into a queue.



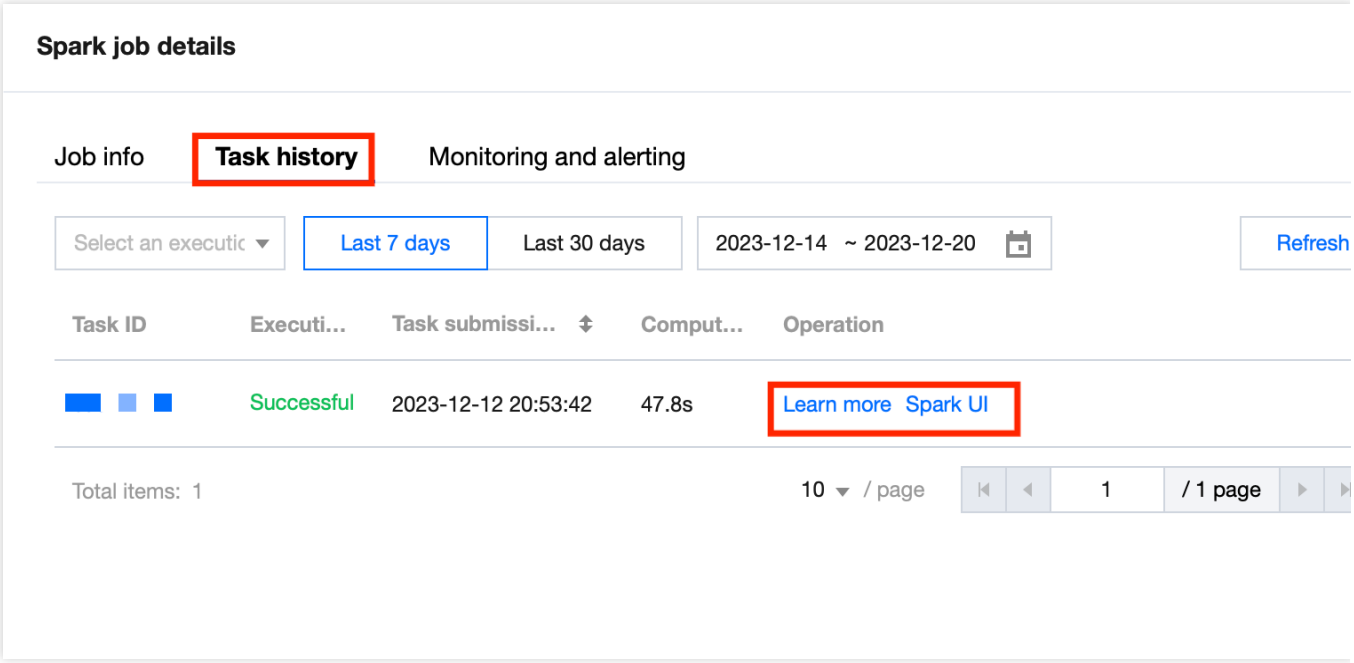
Viewing the Data Job and Task Details

1. Log in to the Data Lake Compute console, select the service region, and select **Data job** on the left sidebar.
2. Click **Job name** to enter the data job details page.

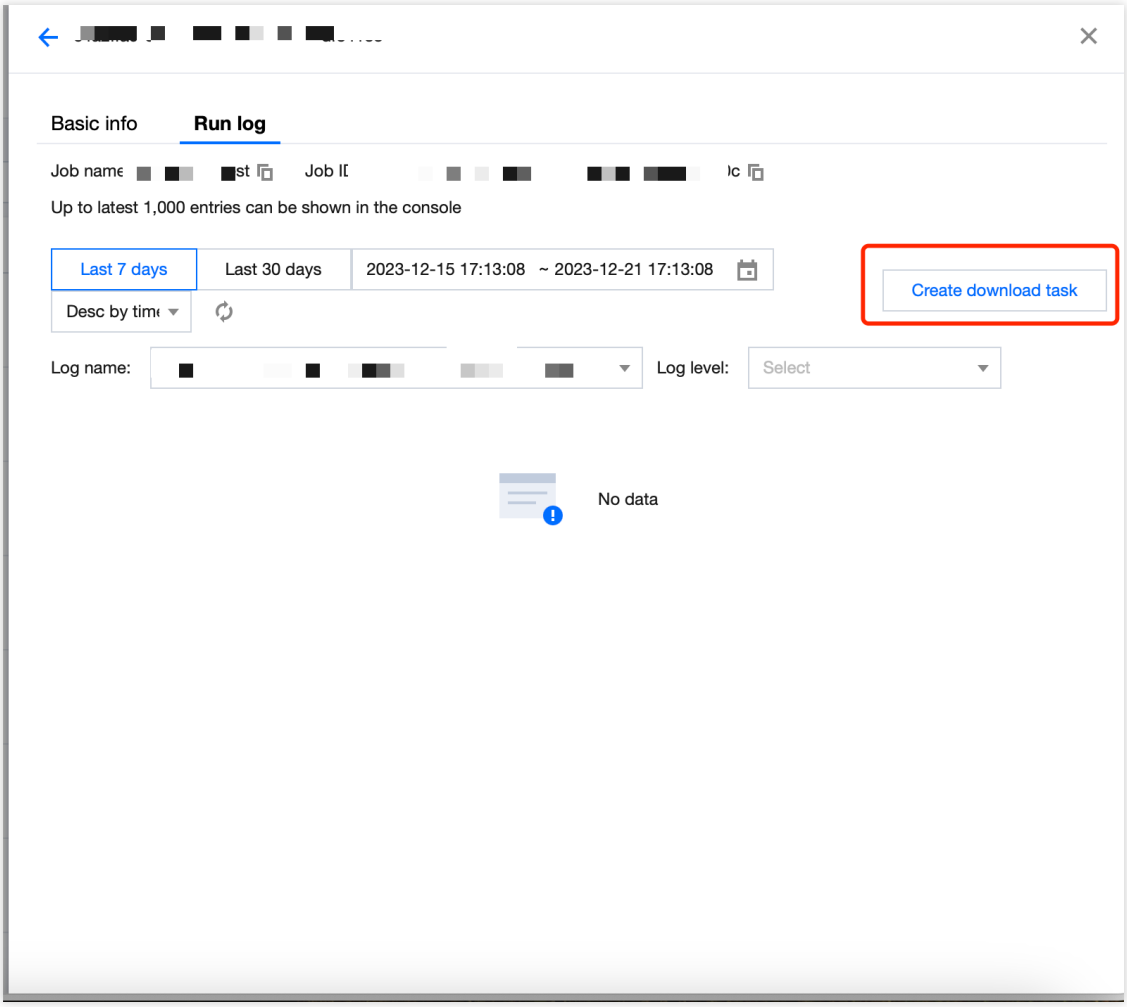


On the details page, you can view the basic information and task list of the data job. The task list contains the data

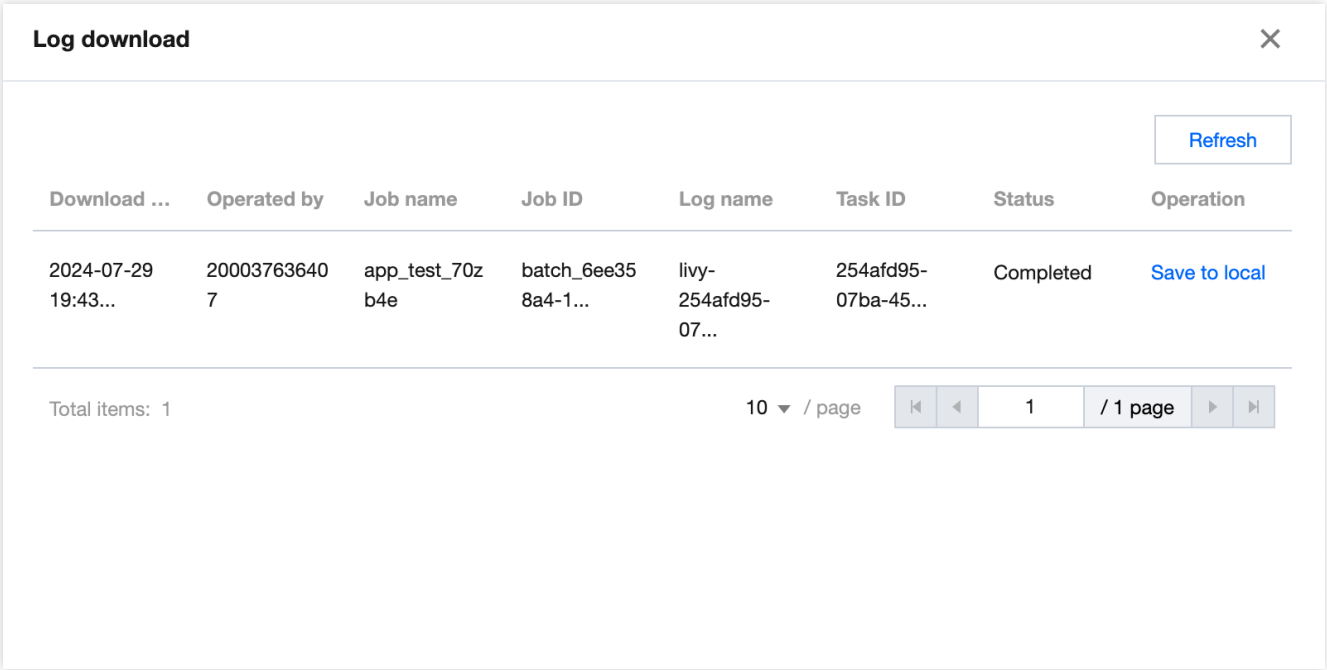
task information of the data job. You can view the task run log and SparkUI.



Click **Learn more** or **Task ID** to view the task details, which include the basic information and run log of the task. Currently, the run log allows you to view the last 1,000 data entries.



You can click **Create download task** to download the full log and click **Log download** to save the log locally.



Note:

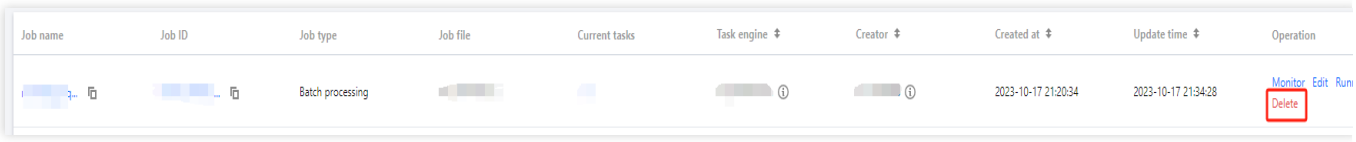
The download record will be saved for three days, after which you cannot save the log locally and need to create a new download task.

Deleting a data job

Note:

A data job with a running data task cannot be deleted.

1. Log in to the Data Lake Compute console, select the service region, and select **Data job** on the left sidebar.
2. Find the target data job, click **Delete** > **OK**.



Note:

Note that deleting a data job will delete its data task information. Proceed with caution.

PySpark Dependency Package Management

Last updated：2024-09-18 17:59:53

Currently, the basic running environment for DLC's PySpark uses Python 3.9.2.
Python dependencies for Spark jobs can be specified in the following two methods:
1. Use `--py-files` to specify dependency modules and files.
2. Use `--archives` to specify a virtual environment.
If your module or file is compiled by using pure Python to implement customized function, it is recommended to specify Python dependencies using the `--py-files` .
The `--archives` option allows you to package and use the entire development and test environment. This method supports compiled installations of C-related dependencies and is recommended when the environment is more complex.
Note:
The two methods mentioned above can be used simultaneously based on your needs.

Using `--py-files` to Specify Dependency Packages

This method is suitable for modules or files implemented in pure Python, without any C dependencies.

Step 1: Packaging Modules/Files

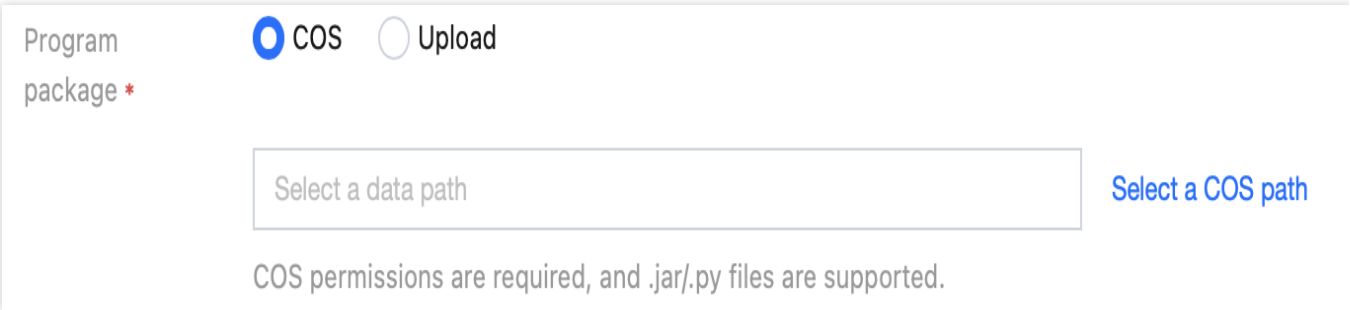
For external PyPI packages, use the pip command to install and package common dependencies in the local environment. The dependencies should be implemented in pure Python and should not be dependent on any C-related databases.

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

cd dep

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

The single-file module (e.g., functions.py) and custom Python modules can be packaged by using the method mentioned above. It is important to ensure that custom Python modules are standardized according to Python's official requirements. For more details, see the official Python Packaging User Guide.

Step 2: Importing the Packaged Module

In the Data Lake DLC Console, create a job in the Data Job module. Use the `--py-files` parameter to import the packaged dep.zip file, which can be uploaded either through COS or directly from your local device.



Using a Virtual Environment

A virtual environment can resolve issues with some Python dependency packages that are dependent on C databases. Users can compile and install dependency packages into the virtual environment as needed, and then upload the entire environment.
Since C-related dependencies involve compilation and installation, it is recommended to use an x86 architecture machine, Debian 11 (Bullseye) system, and Python 3.9.2 environment for packaging.

Step 1: Packaging the Virtual Environment

There are two methods to package a virtual environment: using Venv or Conda.
1. Packaging with 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. Packaging with 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 completed, upload the packaged virtual environment file `pyvenv.tar.gz` to COS.

Note:

Use the tar command for packaging.

3. Use the provided packaging script.

To use the packaging script, you need to have docker installed. The script currently supports Linux and macOS environments.

```
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	Specifies the location of the requirements.txt file.
-n	Specifies the name of the virtual environment (default: py3env).
-o	Specifies the local directory to save the virtual environment (default: the current directory).
-h	Prints help information.

```
# requirement.txt
requests

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

After the script running is completed, you can obtain py3env.tar.gz in the current directory and then upload this file to COS.

Step 2: Specifying the Virtual Environment

In the Data Lake DLC console, create a job in the Data Operation Module following the instructions as shown in the screenshot below.

1. For the `--archives` parameter, enter the full path to the virtual environment. The name of the decompressed folder is After the #.

Note:

The # symbol is used to specify the decompression directory. The decompression directory will affect the configuration of the subsequent running environment parameters.

2. In the `--config` parameter, specify the running environment settings.

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

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

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

Note:

Due to the differences in packaging methods between Venv and Conda, the directory structure will vary. You can decompress the .tar.gz file to check the relative path of the Python file.

Resource Management
Engine Management
Data Engine Introduction

Last updated：2025-04-15 16:25:35

The DLC data engine is the foundation of DLC's data analysis and computation services. All calculations performed by users within DLC require the use of this data engine. Depending on the specific use case, users can select the appropriate engine type.

Engine Types

DLC offers two types of data engines for users to choose from: **Standard Engine** and **SuperSQL Engine**. The primary difference between these two engines lies in the SQL syntax they support. The Standard Engine uses native Spark and Presto syntax from the community, while the SuperSQL Engine supports DLC's independently developed unified syntax. This unified SuperSQL syntax can run on both Spark and Presto engines, effectively masking the syntax differences between them. This feature can significantly reduce usage costs in scenes where different analytics engines need to be used together. Below are the main characteristics of each engine and recommendations for selection:

Engine Types	Available Types	Main Features	Usage Requirements	Purchase Recommendations
Standard Engine	Spark Presto	Native syntax: Uses the native syntax from the Spark/Presto community, ensuring low learning and migration costs. Flexible usage: Supports both Hive JDBC and Presto JDBC. Integrated Spark: The standard Spark engine can execute SQL and Spark batch tasks.	Currently, a 2 CU specification free gateway is provided. If you need to upgrade the specification, upgrade the Gateway	1. Require the use of native Spark/Presto syntax. 2. Need to purchase a Spark engine for batch processing and offline SQL tasks. 3. Prefer to use Hive JDBC and Presto JDBC.
SuperSQL Engine	SparkSQL	Unified syntax: A set of syntax applies to both	You need to learn the SuperSQL unified	1. Prefer to use a unified syntax for both Spark

	Spark jobs\\nPresto	Spark and Presto engines. Supports federated queries.	syntax.\\nFor SQL/batch task scenes, it is recommended to purchase the corresponding engine type.	and Presto. 2. Need to perform federated queries.

For more detailed information, see the comparison table below or review the documentation for the Standard Engine and SuperSQL Engine Description.

Detailed Comparison of Standard Engine and SuperSQL Engine

Feature	Standard Engine	SuperSQL Engine	Description
Presto	✓	✓	Both engines support the Presto engine.
Spark	✓	✓	The SuperSQL Engine is divided into SparkSQL and Spark job. The SparkSQL engine supports SQL jobs, while the Spark job engine supports Spark batch and streaming jobs as well as SQL jobs. The Standard Engine is an integrated Spark engine.
SQL Syntax	Native syntax	Unified syntax	The Standard Engine supports native Spark and Presto syntax. The SuperSQL Engine supports DLC's self-developed unified syntax.
Gateway	✓		DLC, based on Apache Kyuubi, has developed its own Serverless gateway service, providing a more stable, secure, and high-performance task submission experience.
Resource Group	✓		Resource groups are a unique feature of the Standard Spark Engine, allowing resources to be allocated as needed. SQL tasks can be submitted to a designated resource group for execution.
Shared Engine		✓	The SuperSQL Engine supports a shared mode, which is suitable for scenes with low analysis frequency and smaller 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 types of engines support submitting tasks using DLC JDBC.
TencentCloud API Task Submission	✓	✓	Both types of engines support submitting tasks using TencentCloud API or through the data exploration page in the console.
Federated Query		✓	The SuperSQL Engine provides federated query analysis capabilities. For instructions on adding a federated query data catalog, see Data Directory and DMC. The Standard Engine currently does not support federated queries.

If you have any questions about choosing between the Standard Engine or SuperSQL Engine, you can Submit a Ticket to contact us.

Engine Pricing

Data engines support both monthly subscription and pay-as-you-go subscription. For more information, see Billing Overview.

Limitations

The name of the data engine should be globally unique and cannot be changed.
The billing mode of the data engine cannot be switched.
The data engine does not support changing regions.

SuperSQL Engine

SuperSQL Engine Overview

Last updated：2025-03-07 15:27:25

Data engines empower the data analysis and computing service in Data Lake Compute. They are used in all computing operations and can be public or private based on your needs.

Public engine

The Data Lake Compute service comes with the shared public engine, which is applicable to low-frequency analysis use cases with small data volumes. With this highly flexible and available engine, you don't need to configure or manage resources. Fees are charged by the scanned data volume of running tasks. For billing details, see Billing Overview.
Since Data Lake Compute adopts serverless architecture, it needs to schedule the data engine for task execution for the first time over a period of time, which may take a longer time.

Private engine

A private engine is a dedicated data engine that you purchase on a pay-as-you-go basis. For billing details, see Billing Overview.
Pay-as-you-go: This billing mode is highly flexible and stable, where fees are charged by the CU usage. It is applicable to use cases where data is analyzed regularly, with compute resources elastically scaled based on the business load.
Monthly subscription: This billing mode is applicable to use cases where large amounts of data require long-term and stable analysis, with compute resources elastically scaled based on the business load. It guarantees always available resources with no need to wait for resource startup. Fees are charged by month based on the cluster specification (elastic clusters are billed by CU usage though).

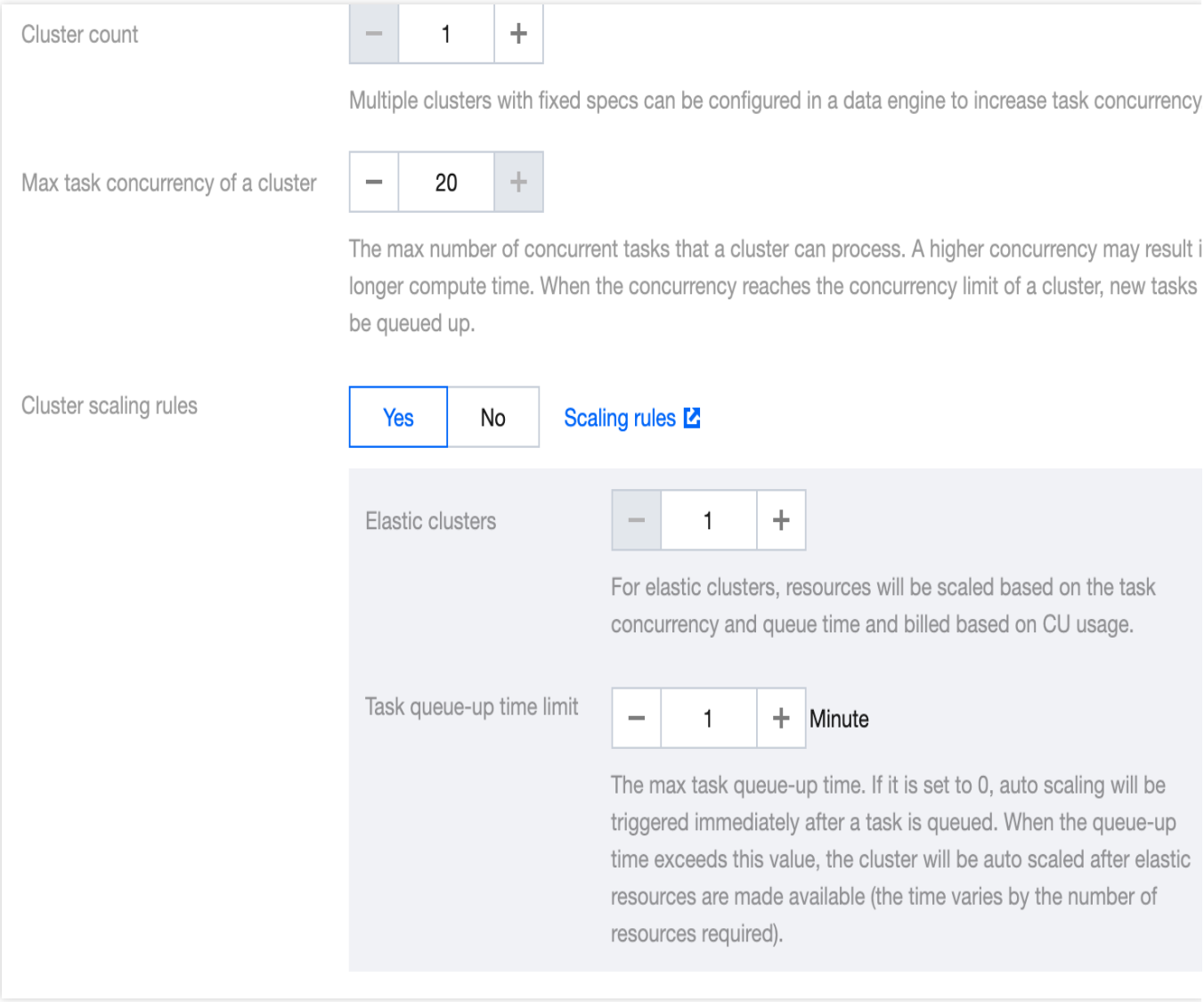
Compute engine types

A private engine can work with different compute engines in different use cases.
SparkSQL: It is suitable for stable and efficient offline SQL tasks.
Spark job: It is suitable for native Spark stream/batch data job processing.
Presto: It is suitable for agile and fast interactive query and analysis.
Note:

The compute engine type does not affect the unit price of a private engine.

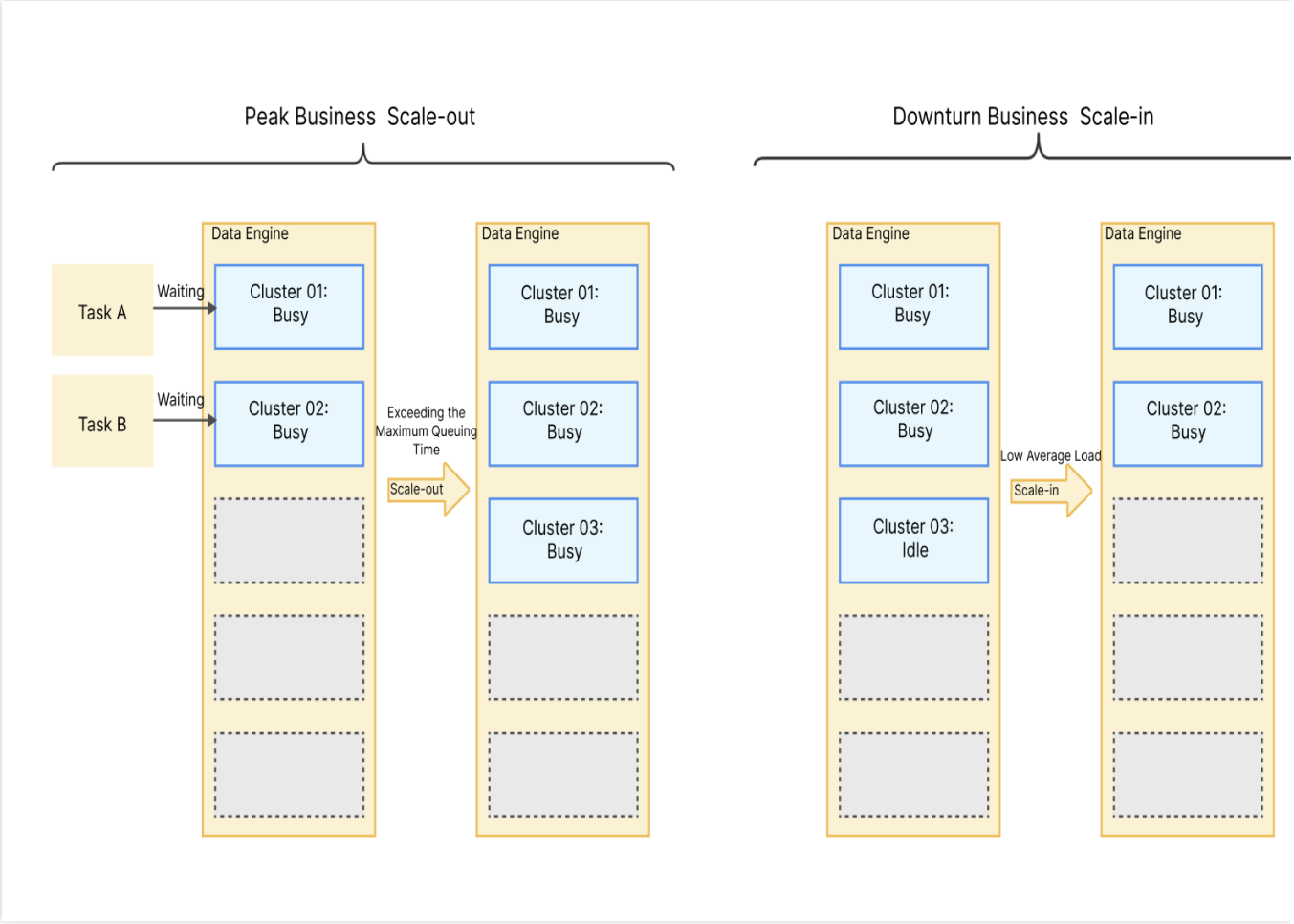
Engine scaling rules

The elastic scaling rules for the engine can be configured either in Create Engine or in the SuperSQL Engine.within the Console Data Engine.



The number of clusters refers to the number of resident clusters in the engine. The sum of the total number of clusters and elastic clusters represents the maximum number of clusters the engine can scale to during elastic scaling.
Basic rule: Engine scaling will only occur when the number of elastic clusters is greater than zero.
Scale-out rule: The system will scale out the data engine based on the configured rules when the number of queued tasks exceeds the available concurrent capacity, the task queue time surpasses the queue time limit, and no clusters are being initialized.

Scale-in rule: The system will scale in the data engine when the current number of clusters exceeds the number of resident clusters, the overall average load of the clusters is below 20%, and there are idle clusters.
As shown in the figure below: During the purchase, the number of clusters is set to 2, the number of elastic clusters to 3, and the task queue time limit to 5 minutes. During high concurrency of cluster tasks, if the number of queued tasks exceeds 2 and the queue time exceeds 5 minutes, the system will scale out the data engine to alleviate the task queuing situation. After successful scale-out, if the task queuing situation is alleviated, clusters become idle, and the load is low, the system will scale in the data engine.



In the case of elastic scaling, the number of clusters in the data engine will not be less than the configured cluster count and will not exceed the sum of the configured cluster count and the elastic clusters.
For example, if the configured number of clusters is 2 and the number of elastic clusters is 3, after scaling out, the number of clusters will not exceed 5, and after scaling in, the number of clusters will not be fewer than 2.

Note:

The cluster count of a data engine cannot be smaller than the minimum cluster count. A pay-as-you-go cluster can be suspended if it is not needed.

Engine running status

A cluster may be in one of the following eight statuses: Starting, Running, Suspended, Suspending, Changing configuration, Isolated, Isolating, Recovering.
Starting: The cluster is being started. In this case, a pay-as-you-go private engine is not billed. A starting cluster cannot be selected for data computing.
Running: The cluster is running and can be selected for data computing.
Suspended: The cluster is suspended and cannot be selected for data computing.
Suspending: The cluster is being suspended and cannot be selected for data computing. This will affect running tasks.
Changing configuration: The cluster is undergoing a configuration change and cannot be selected for data computing.
Isolated: The cluster is isolated due to overdue payments and cannot be selected for data computing.
Isolating: The cluster is being isolated due to overdue payments and cannot be selected for data computing. This will affect running tasks.
Recovering: The cluster is being recovered from the **Isolated** status to the **Running** status after the account is topped up. It cannot be selected for data computing.

Purchasing Private Data Engine

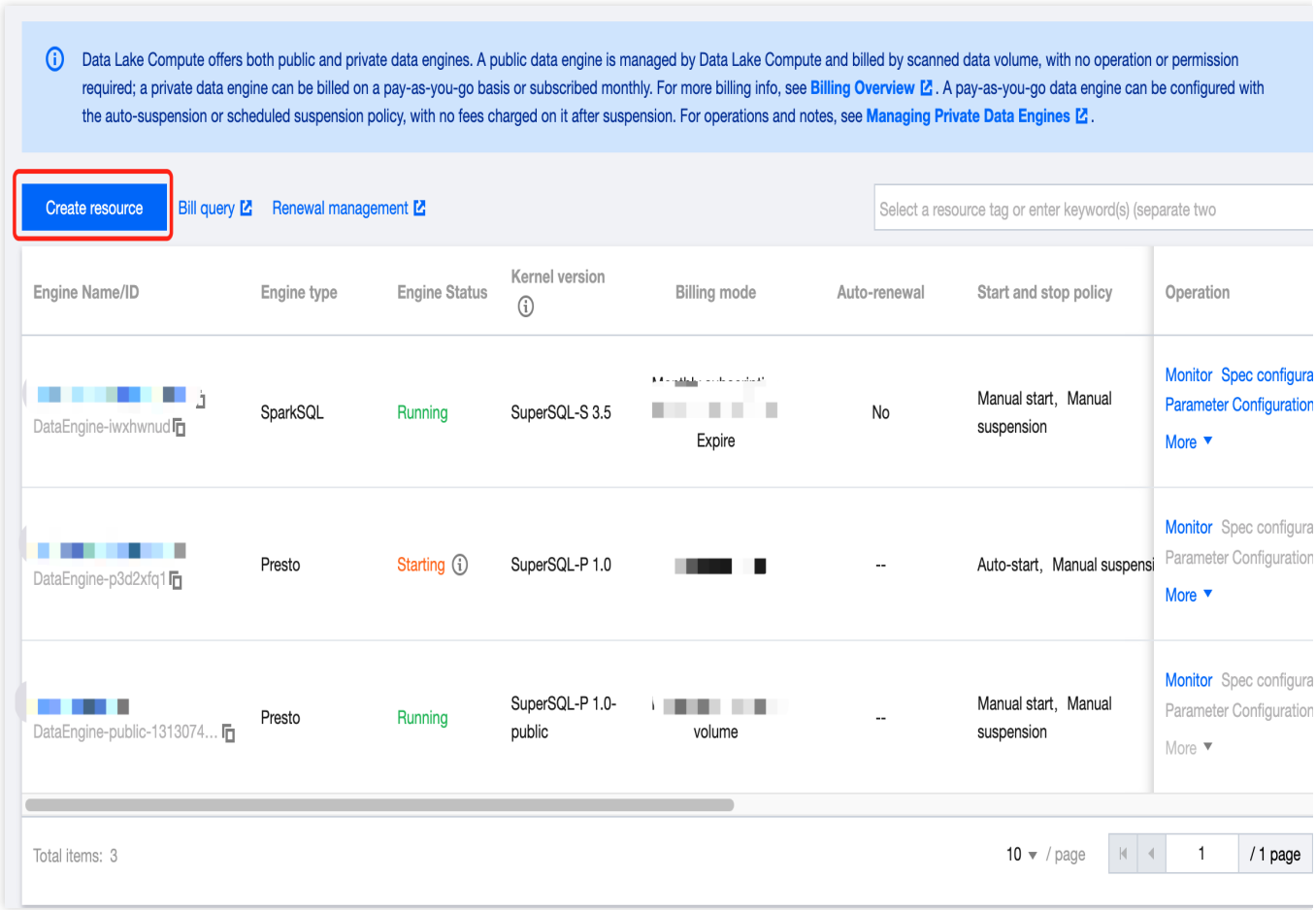
Last updated：2024-07-17 17:55:49

A private data engine in Data Lake Compute supports pay-as-you-go and monthly subscription billing modes. For billing details, see Billing Overview.

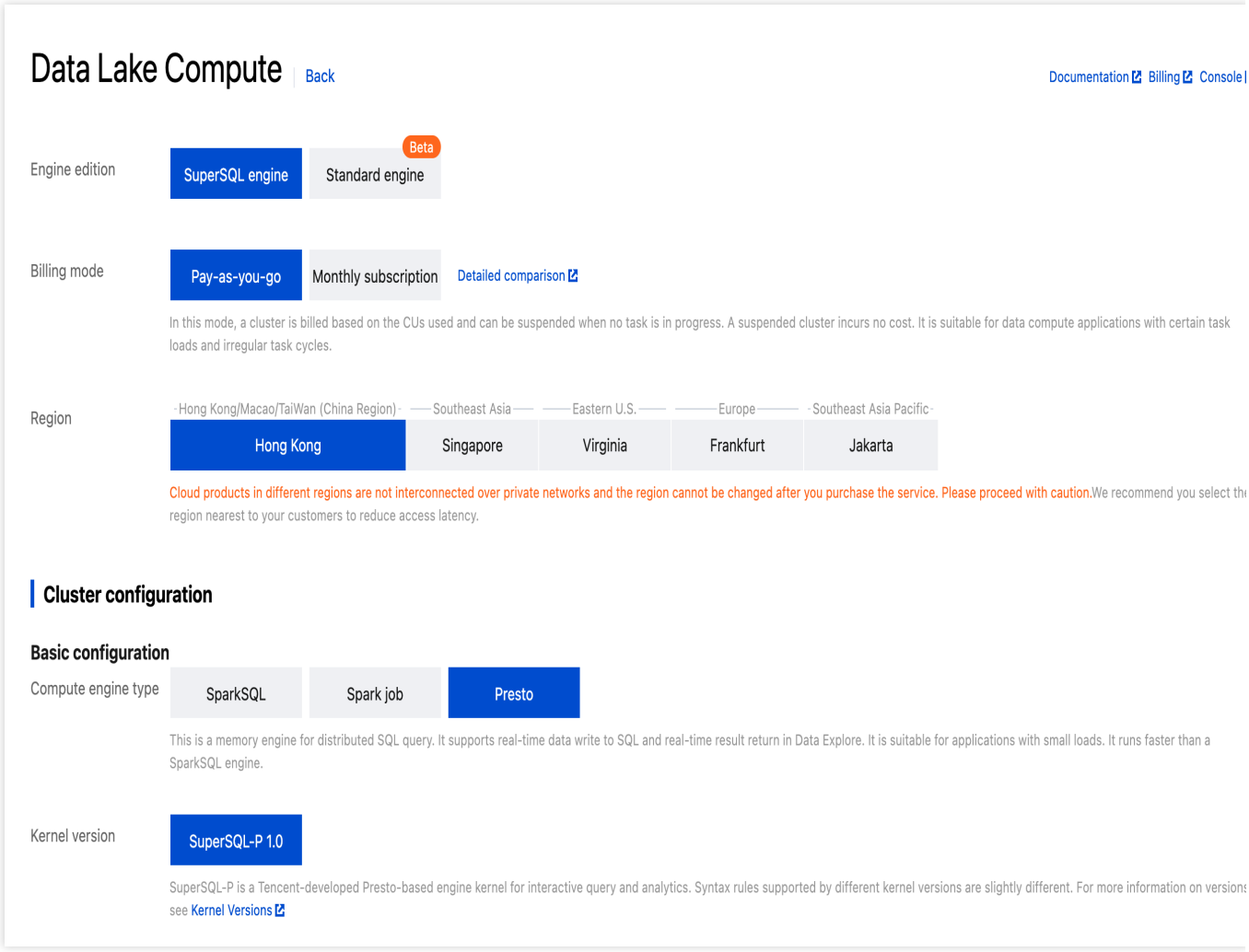
Private engine purchase

You can purchase on the Data Lake Compute purchase page or in the console as instructed below:

1. Log in to the Data Lake Compute console and select the service region. You need to have the Tencent Cloud admin or financial collaborator permission.
2. Click **Data engine** on the left sidebar to enter the data engine management page.
3. Click **Create resource** in the top-left corner to enter the **Resource configuration** page. Configure the resource as needed and view the estimated price.



4. Confirm the price and make the purchase.



Configuration parameter description:

Region: Cloud products in different regions are not interconnected over private networks and the region cannot be changed after you purchase the service. Proceed with caution.

Compute engine: Presto and Spark engines are supported. Note that the engine cannot be changed once purchased. Presto is suitable for faster interactive query and analysis and multi-source federated query, while Spark is suitable for more stable offline tasks with large data volumes.

Cluster spec: Cluster specification is measured in CU. 1 CU equals to 1 CPU core and 4 GB memory of compute resources. The specification determines the amount of compute resources during task execution and can be purchased as needed.

Note:

If you need more than 152 CUs, submit a ticket for assistance.

Min cluster count: Set the minimum number of clusters during cluster start or resident resources in a monthly subscribed cluster. Multiple clusters can deliver a higher concurrency.

Max cluster count: Set the maximum number of clusters for elastic scaling. If it is the same as the minimum cluster count, elastic scaling is not enabled for the cluster.

Auto-start: If it is enabled, a suspended data engine will be automatically started after receiving a task request.

Note:

As pay-as-you-go resources are not reserved, it is possible that they cannot be started right away. If you need resident and stable compute resources, purchase a monthly subscribed data engine instead.

Suspension policy: Configure the suspension method of a pay-as-you-go data engine. Automatic suspension and scheduled suspension are supported. A suspended pay-as-you-go data engine will not incur fees.

Auto-suspension: The data engine will be automatically switched to the **Suspended** status after it has been idle for a certain period of time.

Timing policy: You can configure scheduled start and suspension policies by week. The system will start or suspend clusters regularly as configured.

Suspension after task end: After the specified time elapses, if a task is running, the system will automatically suspend the data engine within five minutes after the task ends.

Suspension after task pause: After the specified time elapses, if a task is running, the system will pause the task and suspend the data engine immediately.

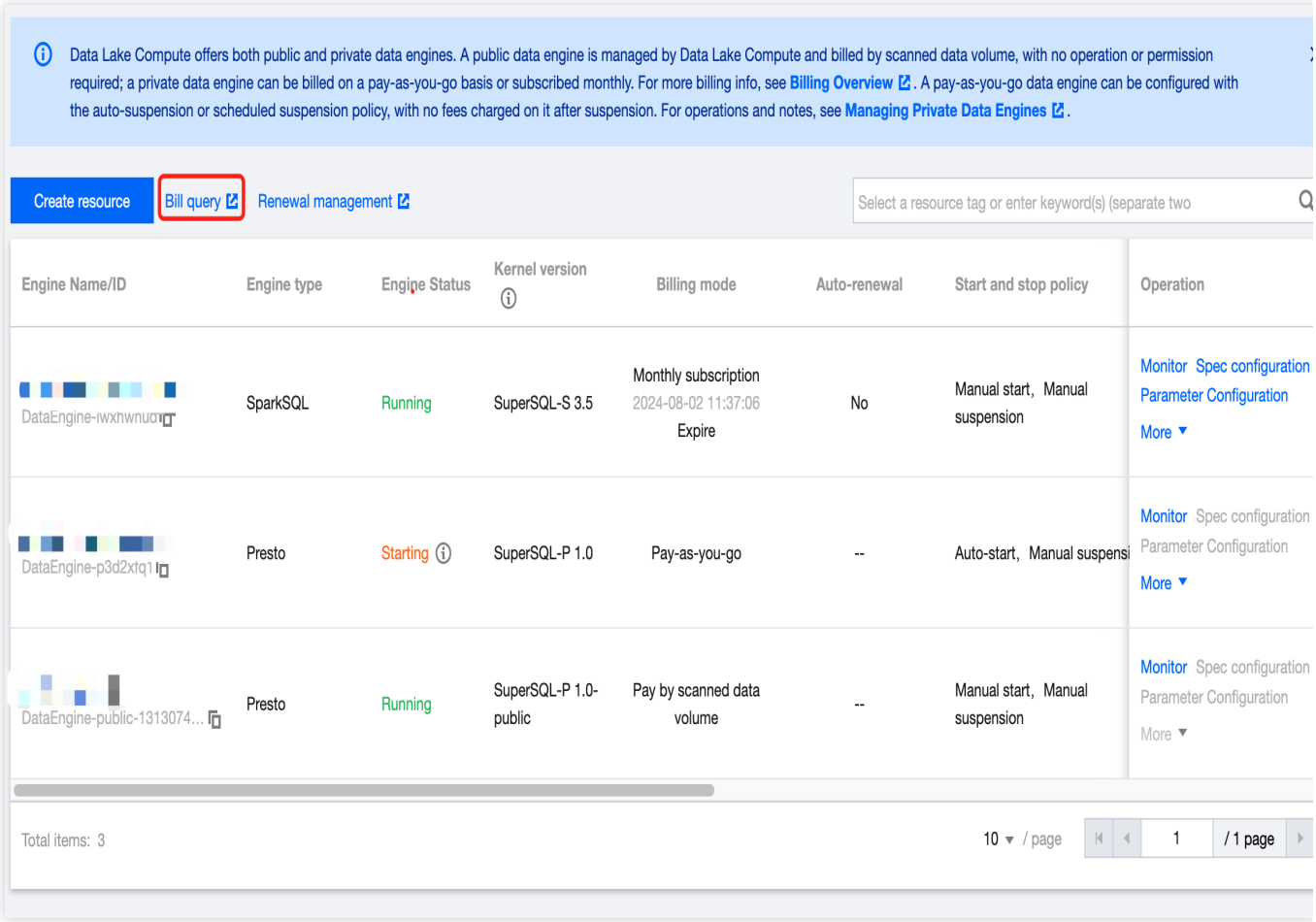
Advanced configuration: If you need to use federated query, configure the IP range in the advanced configuration.

Tag: Set tags to categorize purchased resources and allocate costs. For more information, see Associating Tag with Private Engine Resource.

Bill query

You can query bills in the Data Lake Compute console in the following steps:

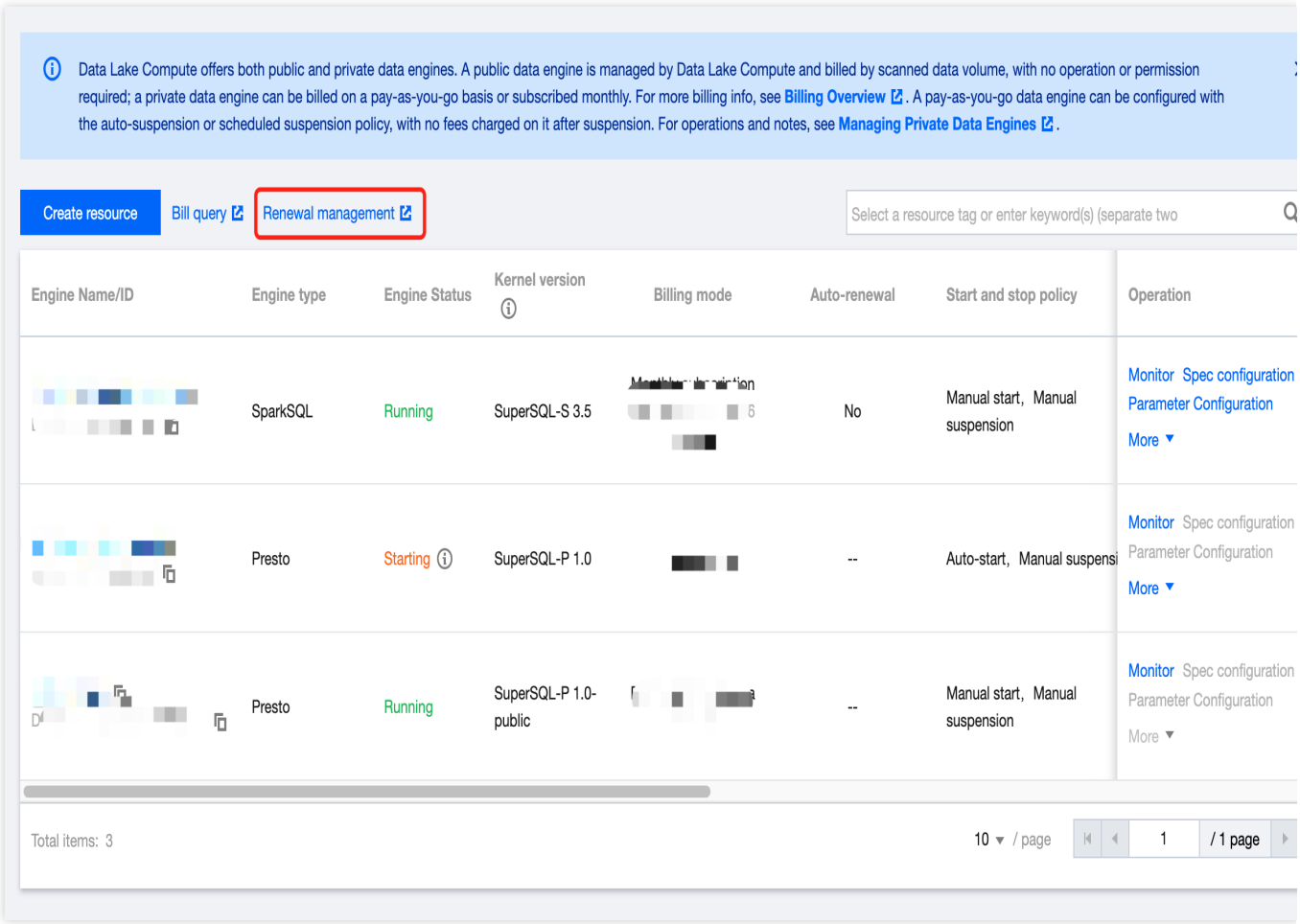
1. Log in to the Data Lake Compute console and select the service region. You need to have the Tencent Cloud admin or financial collaborator permission.
2. Click **Data engine** on the left sidebar to enter the data engine management page.
3. Click **Bill query** to view the detailed bill and settlement information (the financial collaborator permission is required).



Renewal management

For a monthly subscribed private data engine, you can perform renewal and other operations in the Data Lake Compute console > Renewal management > Resource management in the following steps:

1. Log in to the Data Lake Compute console and select the service region. You need to have the Tencent Cloud admin or financial collaborator permission.
2. Click **Data engine** on the left sidebar to enter the data engine management page.
3. Click **Renewal management** to enter the resource list and renew resources (the financial collaborator permission is required).

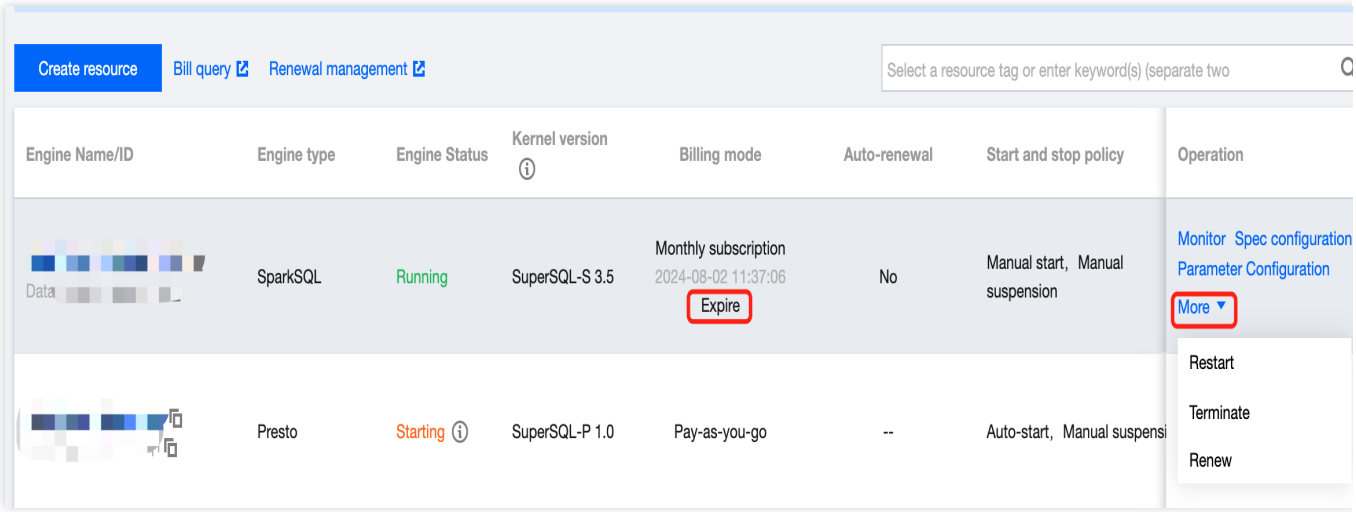

Data Lake Compute offers both public and private data engines. A public data engine is managed by Data Lake Compute and billed by scanned data volume, with no operation or permission required; a private data engine can be billed on a pay-as-you-go basis or subscribed monthly. For more billing info, see Billing Overview. A pay-as-you-go data engine can be configured with the auto-suspension or scheduled suspension policy, with no fees charged on it after suspension. For operations and notes, see Managing Private Data Engines.
Create resource
Bill query
Renewal management
Select a resource tag or enter keyword(s) (separate two
Engine Name/ID
Engine type
Engine Status
Kernel version
Billing mode
Auto-renewal
Start and stop policy
Operation
SparkSQL
Running
SuperSQL-S 3.5
No
Manual start, Manual suspension
Monitor Spec configuration Parameter Configuration More
Presto
Starting
SuperSQL-P 1.0
--
Auto-start, Manual suspensi
Monitor Spec configuration Parameter Configuration More
Presto
Running
SuperSQL-P 1.0-public
--
Manual start, Manual suspension
Monitor Spec configuration Parameter Configuration More
Total items: 3
10 / page
1
/ 1 page

Renewing SuperSQL Engine

Last updated：2024-07-31 17:55:25

You can renew a monthly subscribed data engine that has not expired or is isolated in the Data Lake Compute console.

1. Log in to the Data Lake Compute console and select the service region. You need to have the Tencent Cloud admin or financial collaborator permission.
2. Click **Data engine** on the left sidebar to enter the data engine management page.
3. Find the target data engine and click **More** > **Renew**. You can also renew resources that will expire soon (in seven days) by clicking **Renew** next to the expiration time.



4. Check the renewal term and price and click **Confirm**. The renewal will be completed after the order is confirmed and paid.

Note:

The billing cycle of a data engine that is renewed from the isolated status will start from the expiration date of the previous cycle.

Managing Private Data Engine

Last updated：2024-07-17 18:02:09

Note:

You don't need to manage the public engine, as it is managed by Data Lake Compute in a unified manner.

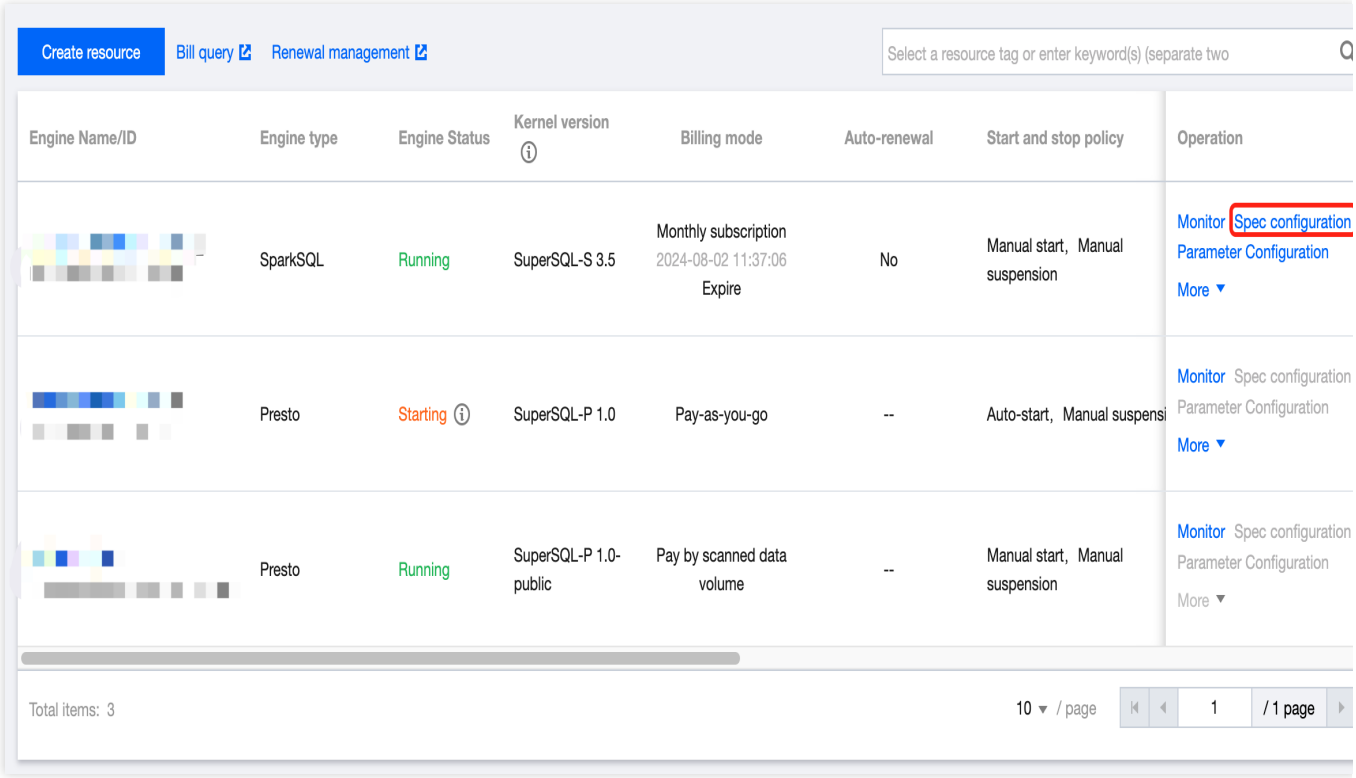
Modifying the private engine configuration

Note:

Fees may change as the private engine configuration changes. For more information, see Configuration Adjustment Fees Description.

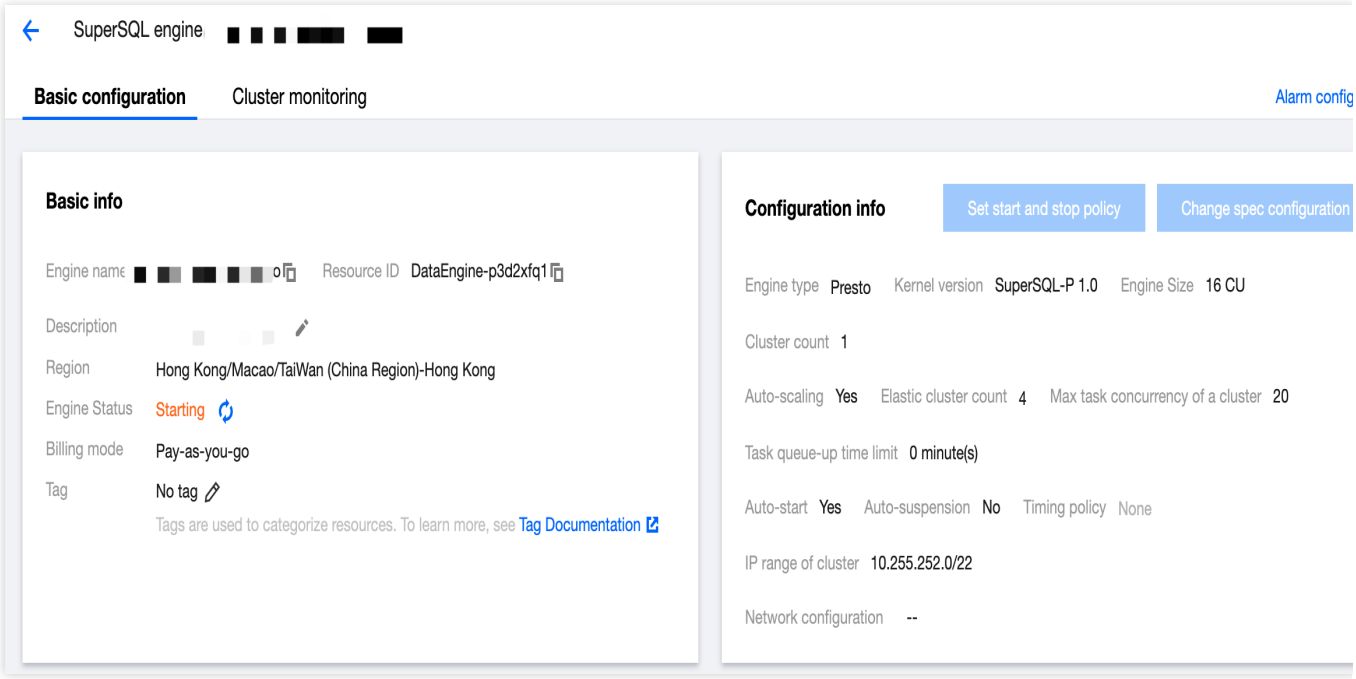
Option 1. Data engine list

1. Log in to the Data Lake Compute console and select the service region. You need to have the Tencent Cloud admin or financial collaborator permission.
2. Click **Data engine** on the left sidebar to enter the data engine management page.
3. Find the target private engine and click **Spec configuration** on the right to enter the configuration modification page, where you can modify the cluster specification and elastic scaling policy.
4. After making changes, click **Save** to submit the order and make the payment.



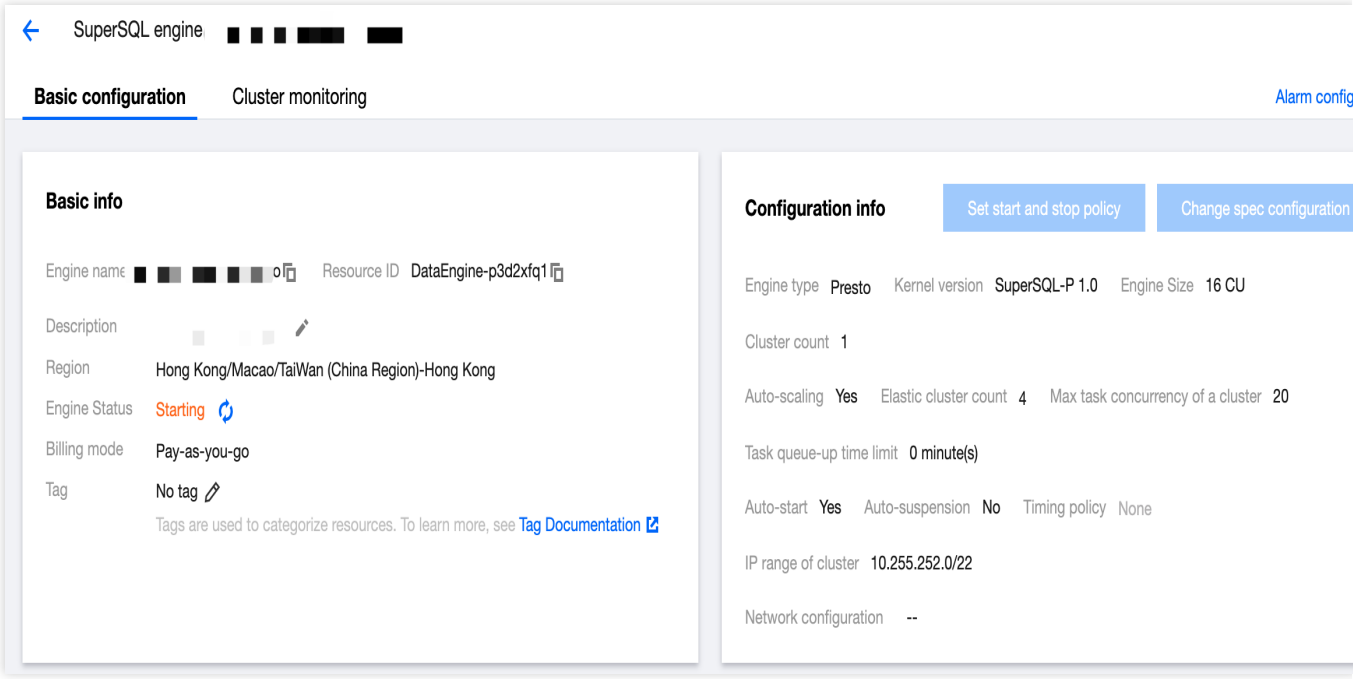
Option 2. Data engine details

1. Log in to the Data Lake Compute console and select the service region. You need to have the Tencent Cloud admin or financial collaborator permission.
2. Click **Data engine** on the left sidebar to enter the data engine management page.
3. Find the target private engine and click the cluster name to enter the cluster details page, where you can modify the cluster specification and elastic scaling policy.
4. Adjust the parameters as needed and click **Save**.



Modifying the private engine information

1. Log in to the Data Lake Compute console and select the service region. You need to have the Tencent Cloud admin permission.
2. Click **Data engine** on the left sidebar to enter the data engine management page.
3. Find the target private engine and click the cluster name to enter the cluster details page, **where you can modify the cluster description, automatic start policy, and suspension policy**.
4. Adjust the parameters as needed and click **Save**.



Suspension policy: Configure the suspension method of a pay-as-you-go data engine. Automatic suspension and scheduled suspension are supported. A suspended pay-as-you-go data engine will not incur fees.

Auto-suspension: The data engine will be automatically switched to the **Suspended** status after it has been idle for 15 minutes.

Timing policy: You can configure scheduled start and suspension policies by week. The system will start or suspend clusters regularly as configured.

Suspension after task end: After the specified time elapses, if a task is running, the system will automatically suspend the data engine within five minutes after the task ends.

Suspension after task pause: After the specified time elapses, if a task is running, the system will pause the task and suspend the data engine immediately.

Enable suspension policy management

It supports the configuration of start & suspend policies for the exclusive data engine of billing by volume, which facilitates management and cost control.

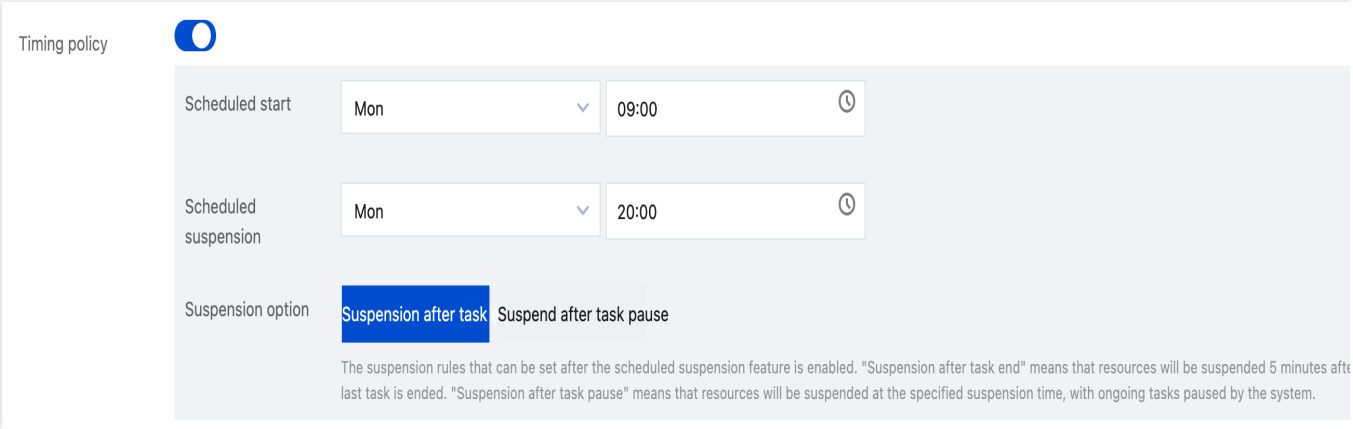
Note：

If the pay-as-you-go data engine is not suspended, charges will be generated. If the data engine is not needed, suspend it in time.

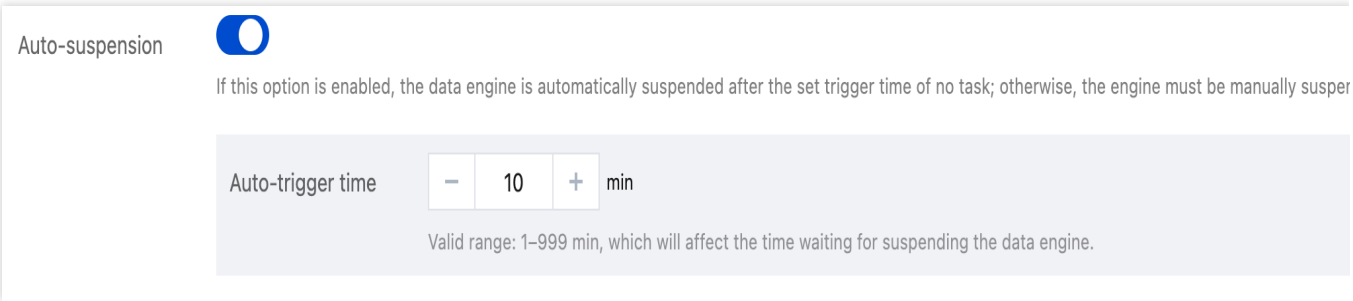
Startup policy: Supports automatic start, manual start, and scheduled start of the data engine.

Automatic start: After the configuration, if the data engine is in the suspended state and a task is submitted to the data engine, the data engine will automatically start.

Manual start: After the configuration, if the data engine is in the suspended state, you need to manually start the data engine before processing data tasks.
Periodic startup: You can configure a weekly periodic startup policy. The system periodically starts the cluster based on the configuration rules.



Suspension policy: Supports the suspension mode of the data engine for charging by volume, including automatic suspension and scheduled suspension. Pay-as-you-go data engines do not incur any costs when suspended.
Automatic suspension: After the configuration, the data engine automatically switches to the suspended state 10 minutes after there is no task, and the triggering time can be configured.



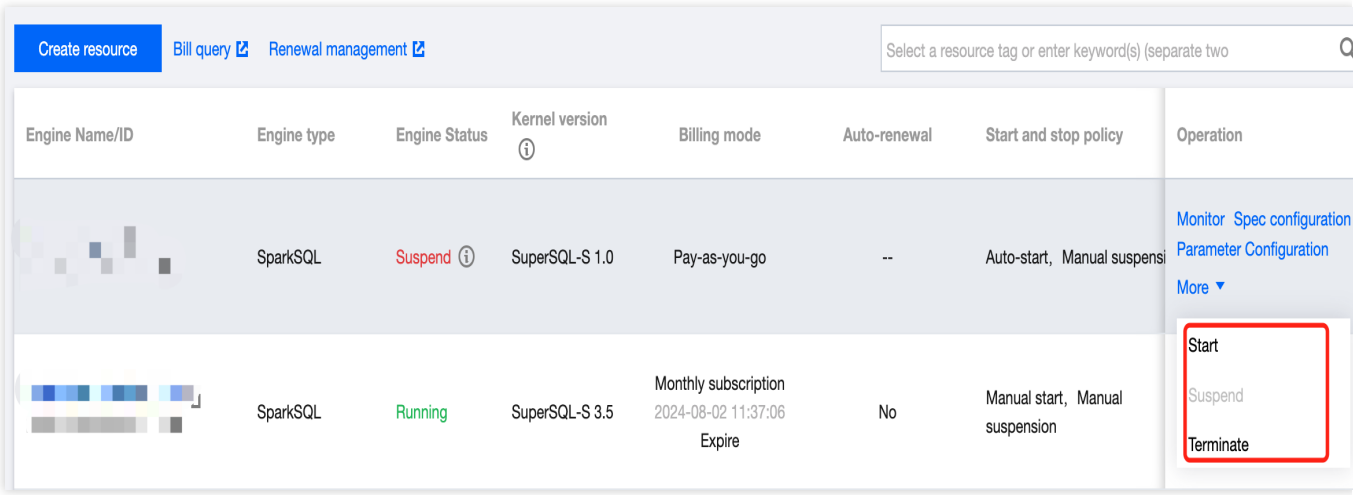
Periodic policy - You can configure weekly periodic start and suspension policies. The system starts and suspends the cluster periodically based on the configuration rules.
Suspend after Completion: If a task is being executed by the data engine within the specified time, the data engine automatically suspends the task within 5 minutes after the task is completed.
Suspend after Automatic pause: If a task is being executed on the data engine within the specified time, the system suspends the task and immediately suspends the data engine.

Manually suspending/starting a private engine

Note:

Monthly subscribed resources are resident and cannot be suspended.

1. Log in to the Data Lake Compute console and select the service region. You need to have the Tencent Cloud admin permission.
2. Click **Data engine** on the left sidebar to enter the data engine management page.
3. Find the target private engine, click **More**, and select **Start** or **Suspend** in the drop-down list.



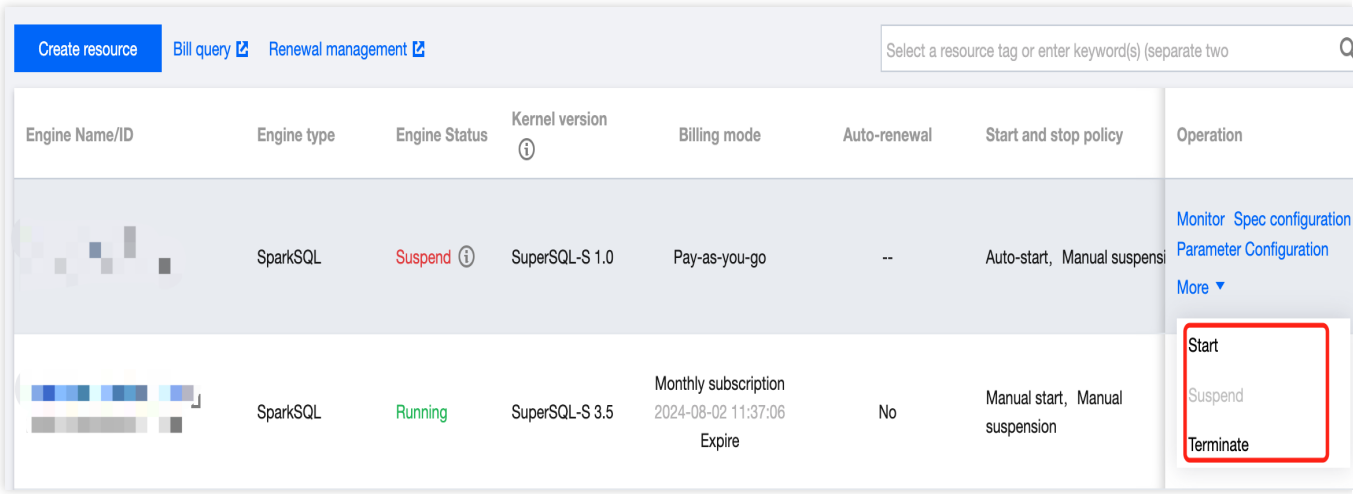
Terminating a private engine

You can terminate a data engine that is no longer needed. A monthly subscribed data engine will be returned automatically after termination. For more information, see Refund.

Note:

Note that a pay-as-you-go data engine cannot be recovered once terminated. Proceed with caution.

1. Log in to the Data Lake Compute console and select the service region. You need to have the Tencent Cloud admin permission.
2. Click **Data engine** on the left sidebar to enter the data engine management page.
3. Find the target private engine (only suspended clusters can be terminated), click **More**, and select **Terminate** in the drop-down list.
4. Confirm the termination.



Cluster running logs

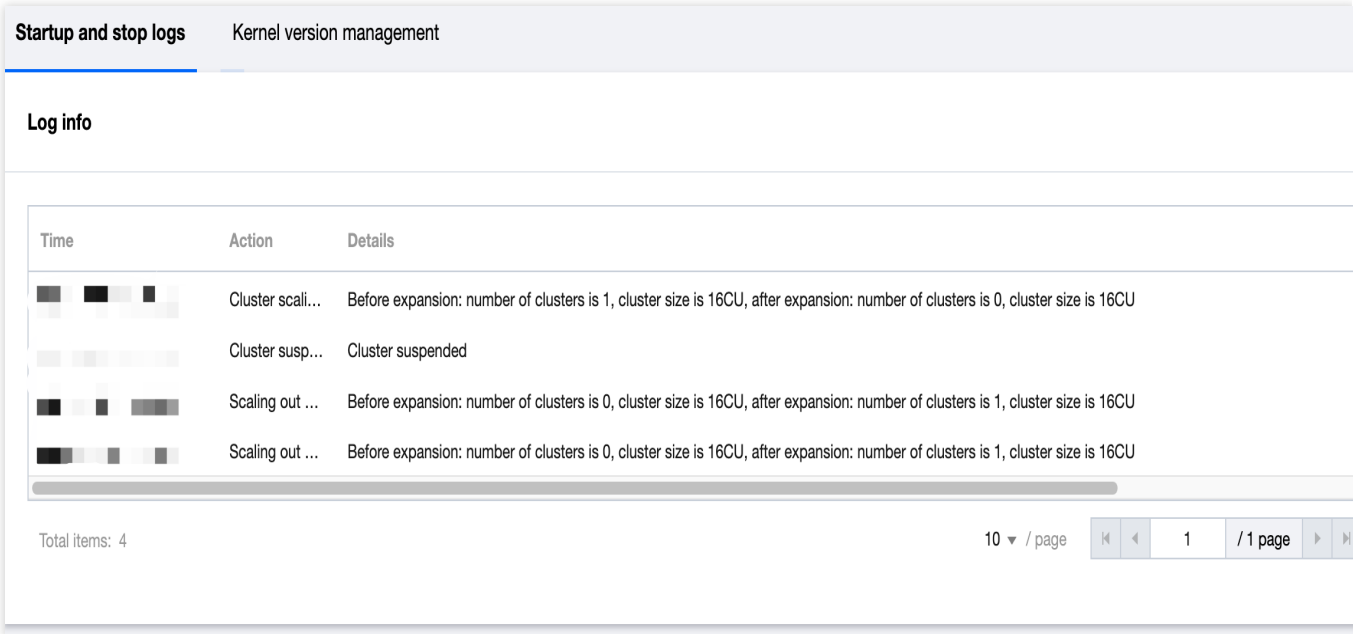
Data Lake Compute provides running logs within 14 days for private engines to help you stay informed of the start, suspension, and scaling of clusters. Cluster logs mainly include the following content:

Start time: The time when the cluster starts working.

Suspension time: The time when the cluster stops working.

Scale-out record: The time of the cluster scale-out and the number of added clusters.

Scale-in record: The time of the cluster scale-in and the number of removed clusters.



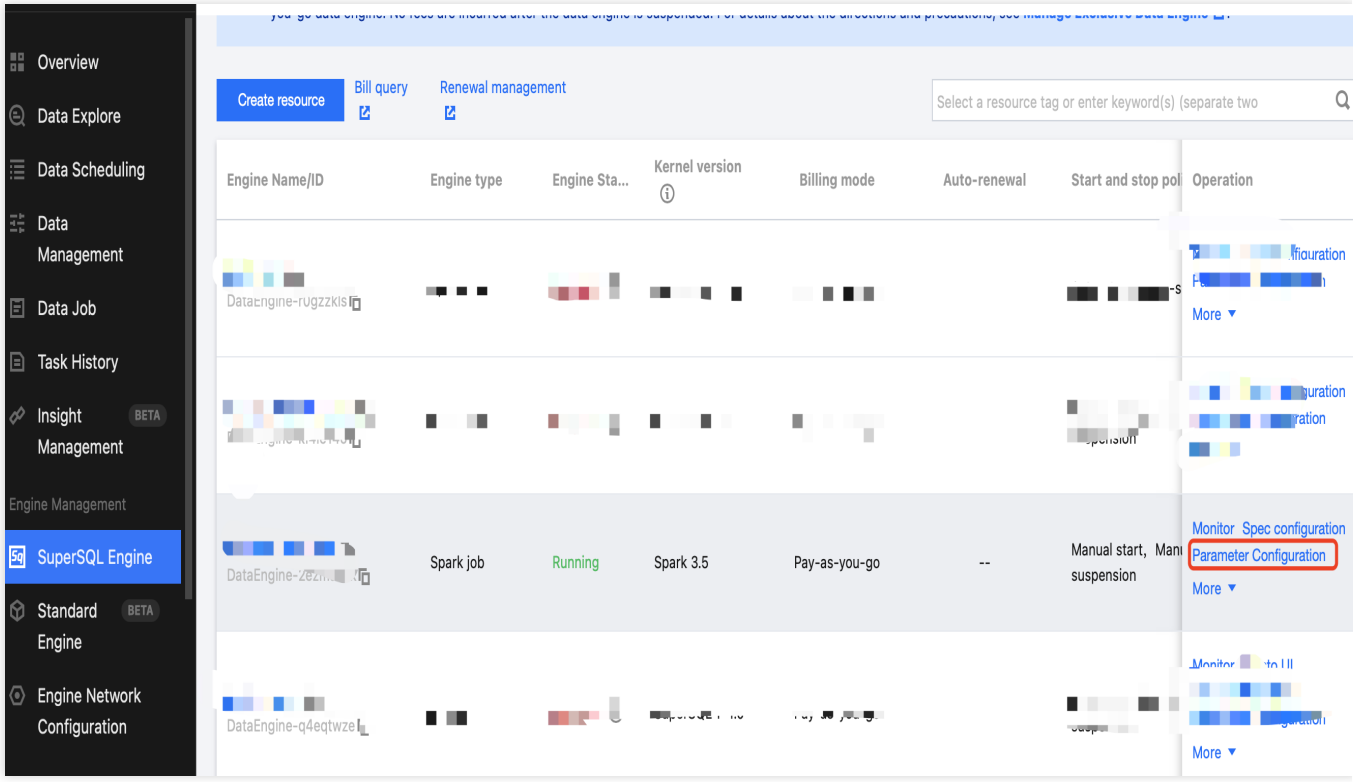
Engine-Level Parameter Settings

Last updated：2024-09-04 11:22:53

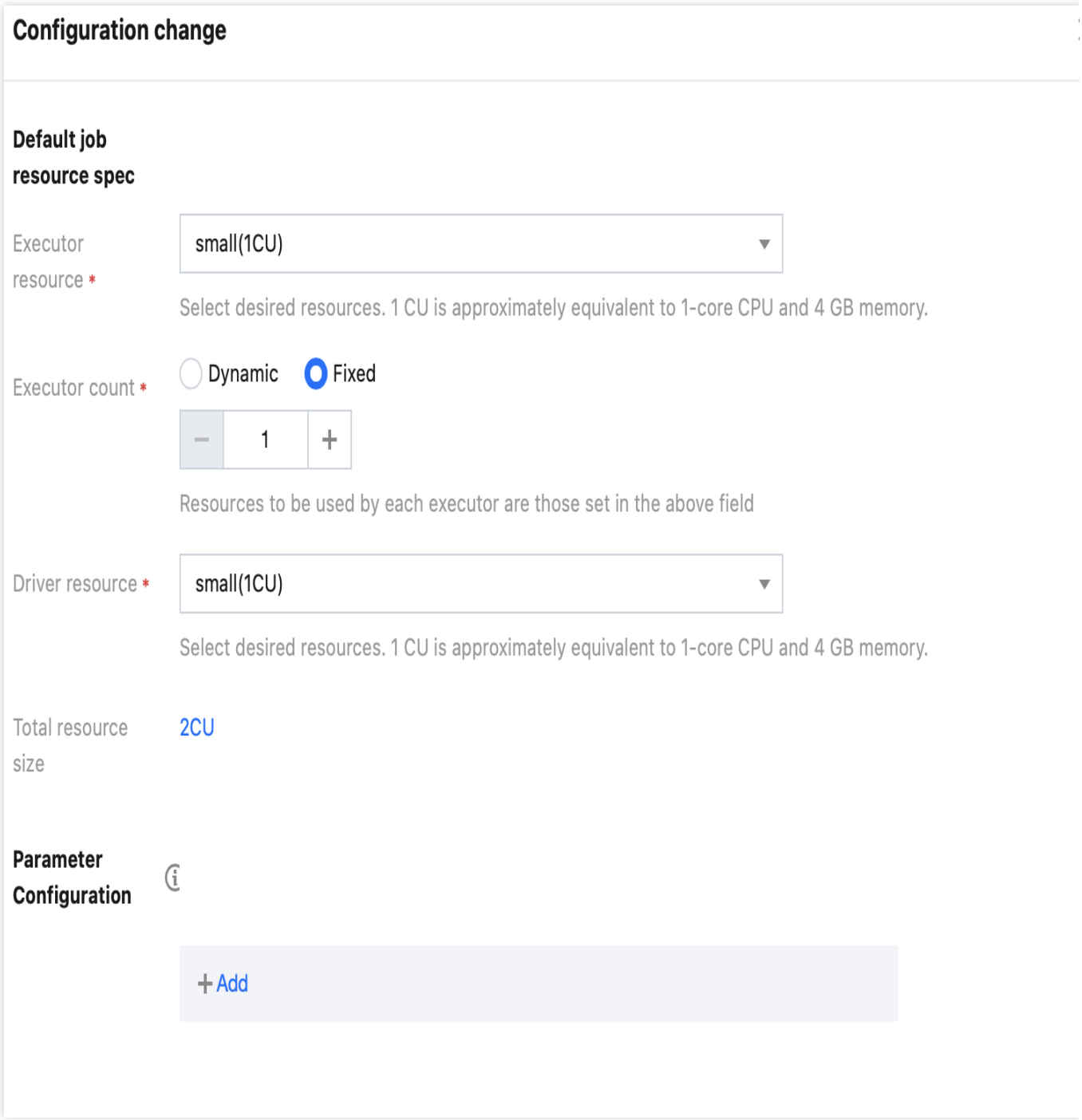
Note:
Currently, only the SparkSQL Engine and Spark Job Engine are supported for engine configuration.
Spark parameters are used to configure and optimize the settings of Apache Spark applications. In a self-built Spark environment, these parameters can be set via command-line options, configuration files, or programmatically. In DLC, you can specify Spark parameters within the SQL and code of the SparkSQL Engine and Spark Job Engine, or you can directly set engine-level parameters. The engine-level Spark parameter configuration is as follows.

Setting Engine-Level Parameters

1. Enter the SupersSQL Engine module, click **Parameter Configuration**, and the engine parameter side window will appear.



2. Under the Spark Job Engine, you can configure the default resource specifications and parameters for jobs. In the SparkSQL Engine, there's no need to adjust the default resource specifications for jobs.



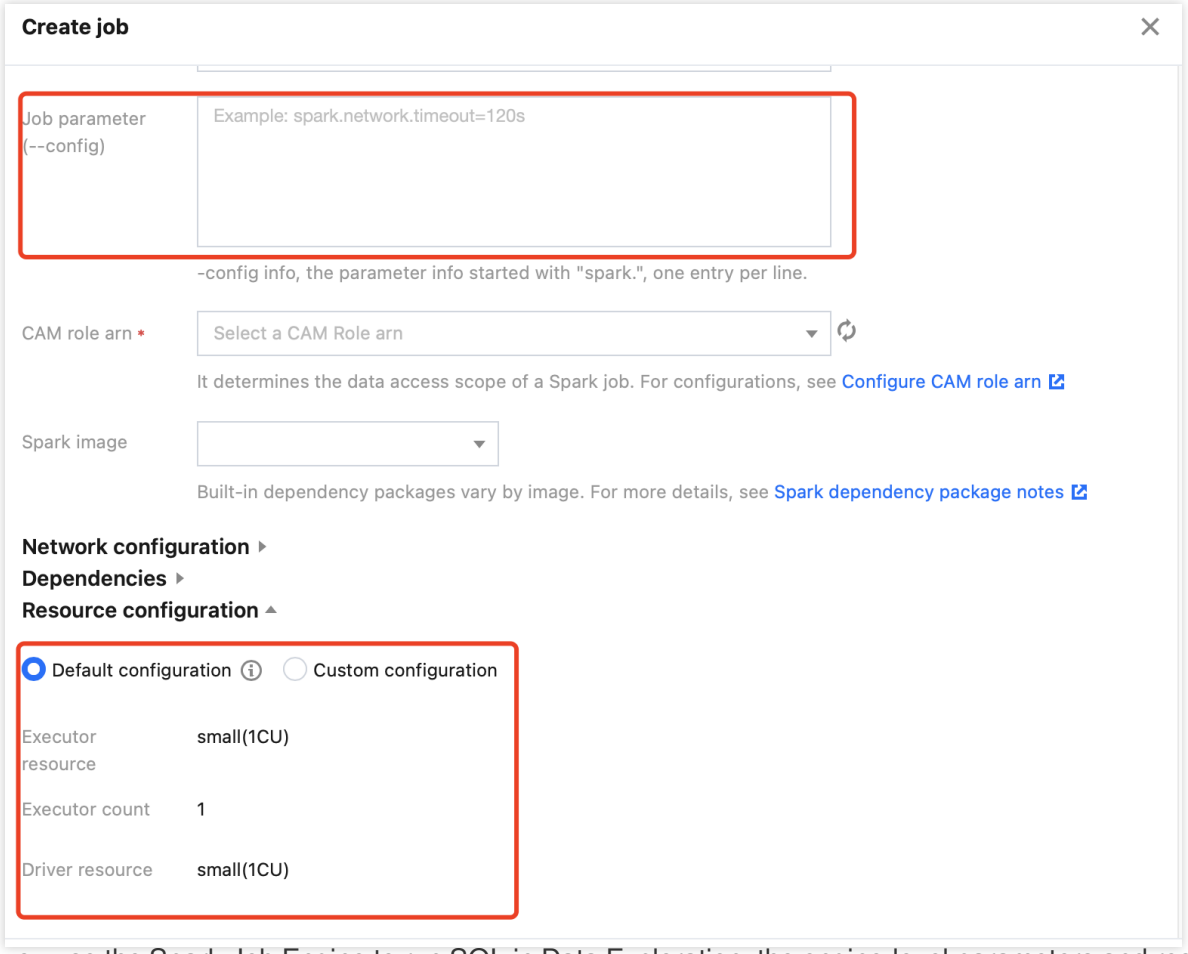
Using Engine-Level Parameters

Spark Job Engine Using Engine-Level Parameters

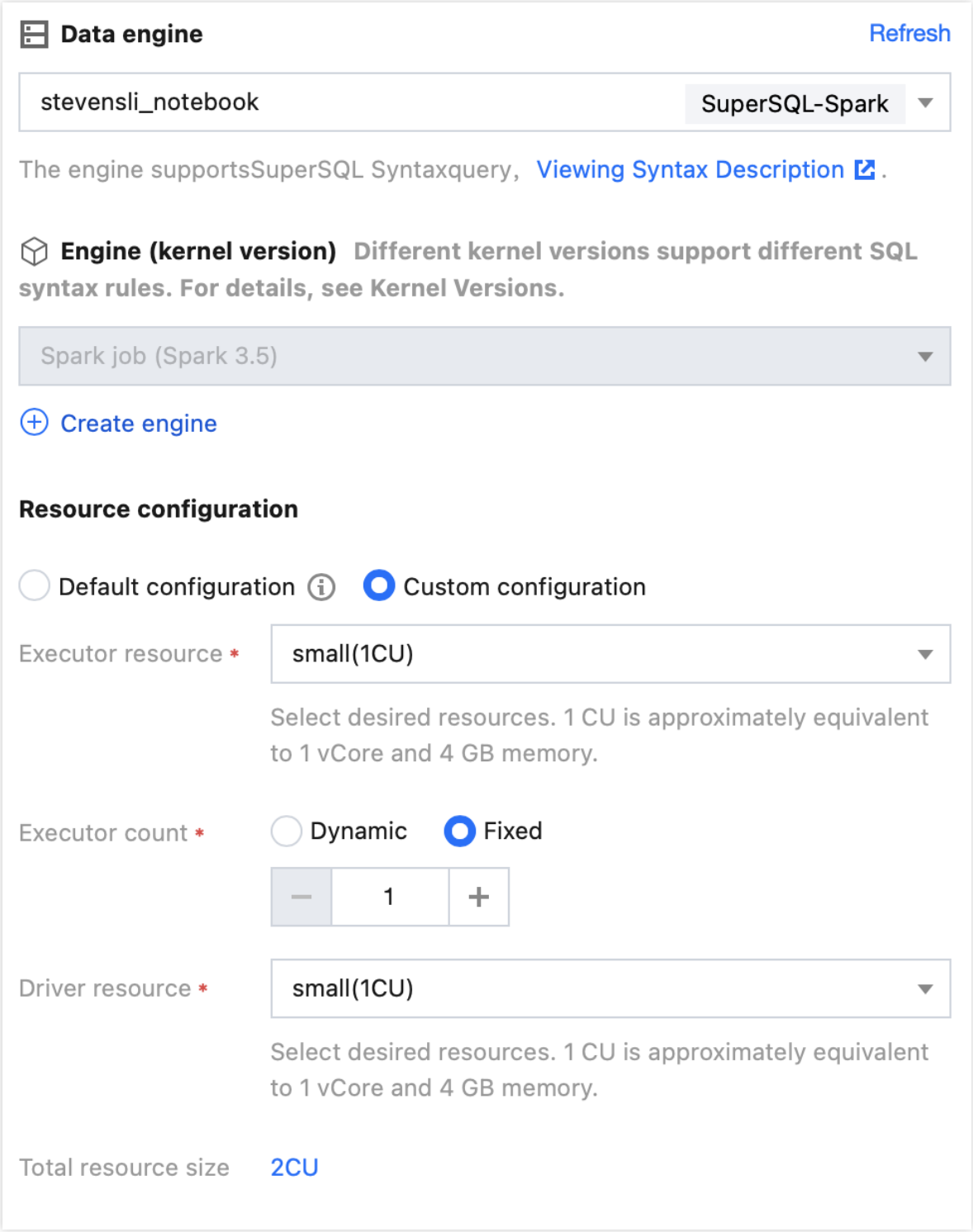
There are two entry points for submitting jobs in the Spark Job Engine: Data Job and Data Exploration. Both support the use of engine-level parameters.

When you create a data job, 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. If the default configuration is selected, the engine-level resource configuration will be used.



When you use the Spark Job Engine to run SQL 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 within the SQL, and choose whether to inherit the engine-level resource configurations.



SparkSQL Engine Using Engine-Level Parameters

The SparkSQL Engine does not have engine-level resource parameters, so tasks will use as much of the cluster's resources as possible. Currently, SQL needs to be submitted using the SparkSQL Engine within Data Exploration. When you run SQL in Data Exploration with the SparkSQL Engine, engine-level parameters are inherited by default. You can override these parameters using the set command within the SQL.

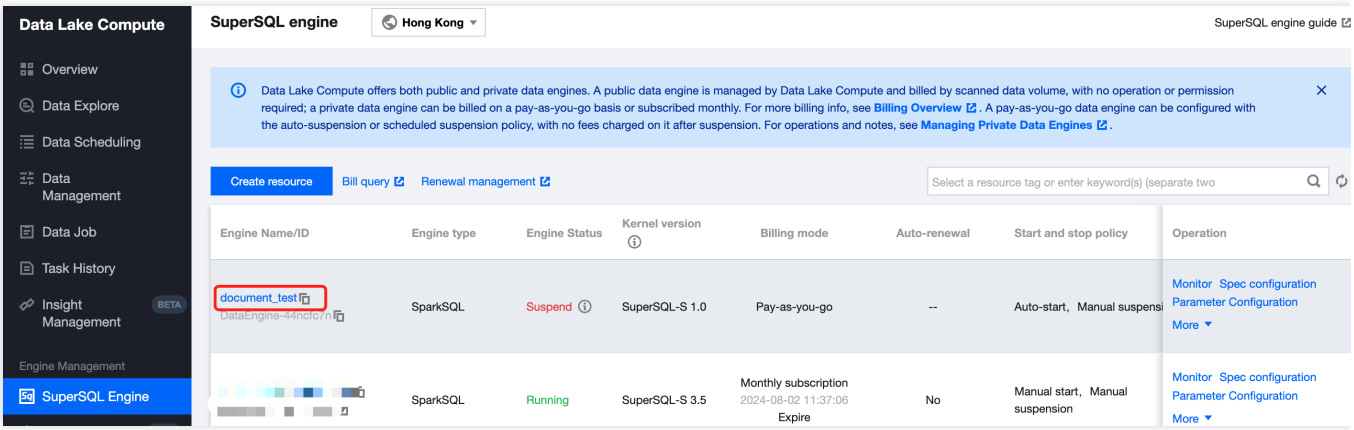
Disaster Recovery Cluster

Last updated：2024-07-31 17:47:09

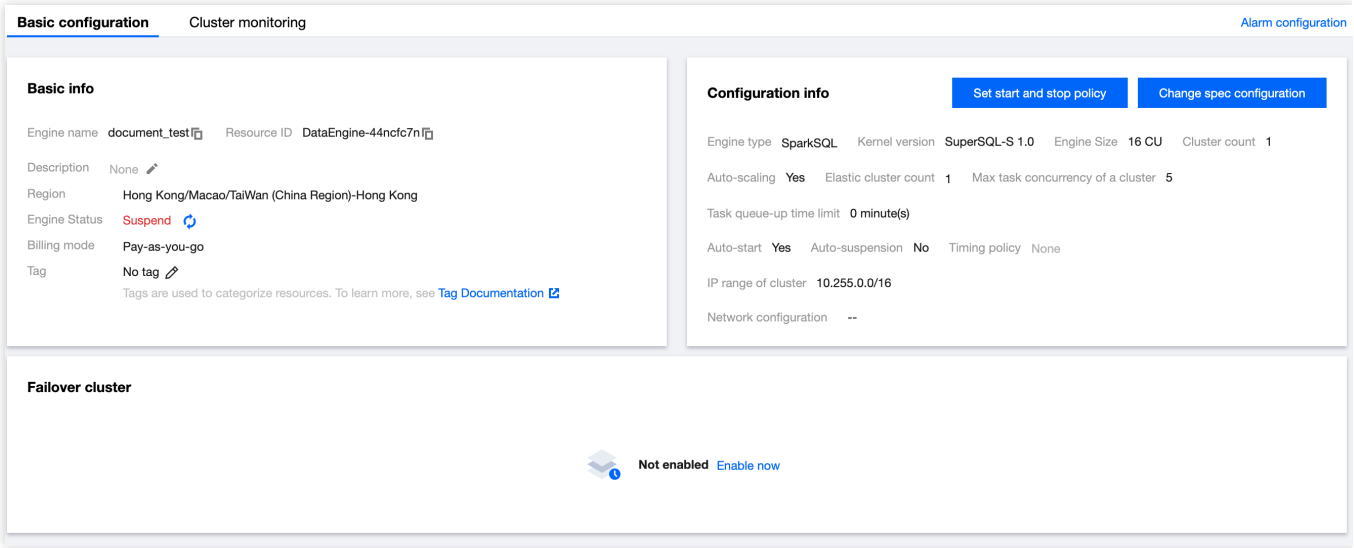
To ensure the stable operation of the compute engine under extreme scenarios, DLC provides an efficient and agile disaster recovery cluster capability. When you need a disaster recovery cluster, you can quickly switch to it to ensure normal service operation. The disaster recovery cluster is only charged during operation, for more details, please see Cost Description.

Operation step

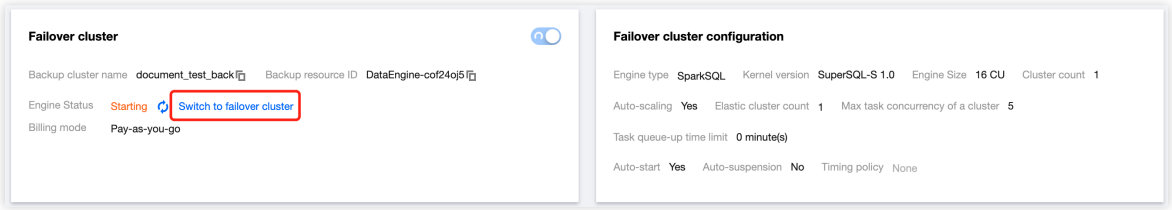
1. Enter the DLC Console, click Data Engine to access the Data Engine Page.
2. Click on the Data Engine Resource Name to enter the Data Engine Detail Page.



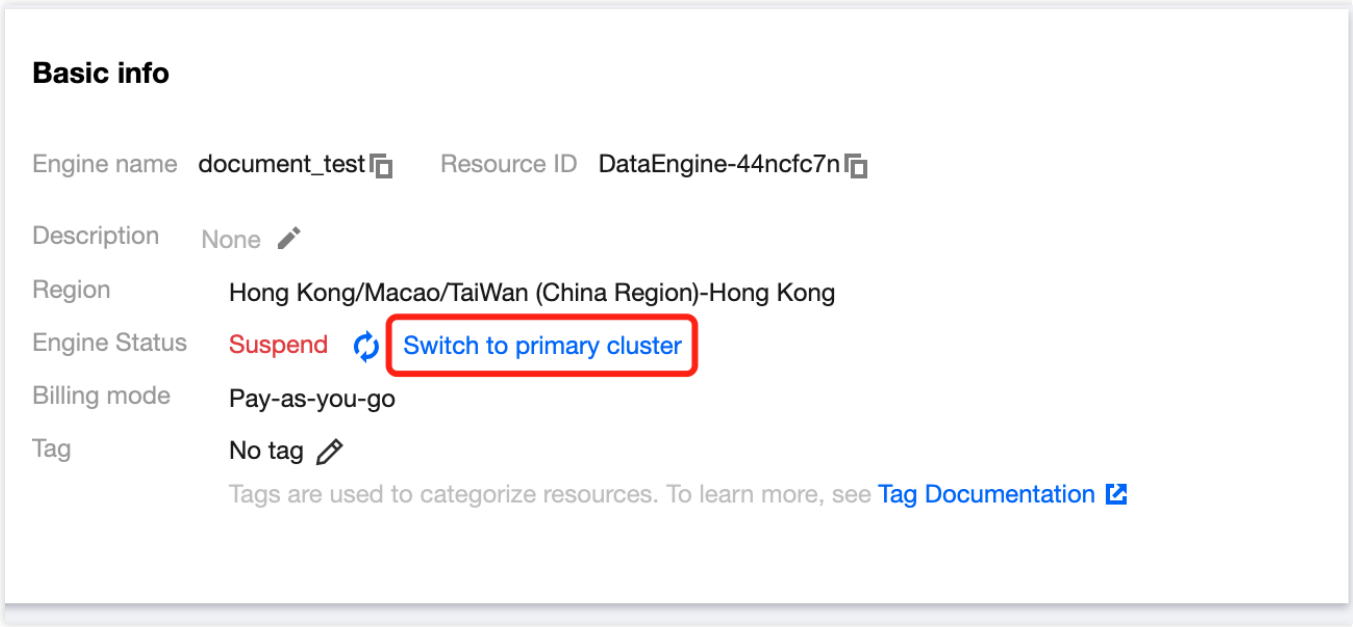
3. Click **Enable Disaster Recovery Cluster** and wait for the disaster recovery cluster to initialize.



4. After the disaster recovery cluster is enabled, in the disaster recovery cluster information, click **Switch to Disaster Recovery Cluster** to adjust the running cluster to the disaster recovery cluster. Subsequently, jobs directed to this data engine will be submitted to the disaster recovery cluster. The disaster recovery cluster serves as a transition during extreme failures of the data engine.



5. Once the extreme failure is resolved, in the basic information of the data engine, click **Switch to Primary Cluster**, and the disaster recovery cluster will be suspended. Subsequently, jobs directed to this data engine will be submitted to the primary cluster.



Disaster Recovery Cluster Specifications

The disaster recovery cluster always tries to match the specifications of the data engine itself to ensure that the original tasks can transition and run normally. When AS is enabled on the data engine itself, the AS rules of the disaster recovery cluster will be consistent with the data engine. At the same time, to save costs, the disaster recovery cluster always operates on a pay-as-you-go basis.

Note on Fees

There is no charge for enabling the disaster recovery cluster. When switching to the disaster recovery cluster and it is running, charges will be applied according to the pay-as-you-go rates for the same specifications as the data engine.
Example:
1. When the data engine itself is a 16 CU SparkSQL engine with an annual and monthly subscription. After enabling the disaster recovery cluster, it becomes a 16 CU SparkSQL engine on a pay-as-you-go basis, and there is no charge while the disaster recovery cluster is suspended. When users switch to the disaster recovery cluster and it is running, additional charges for the disaster recovery cluster's use of CU duration will apply. For specific fees, please refer to Billing Overview.
2. When the data engine itself is a 16 CU SparkSQL engine on a pay-as-you-go basis. After enabling the disaster recovery cluster, it remains a 16 CU SparkSQL engine on a pay-as-you-go basis, and there is no charge while the disaster recovery cluster is suspended. When users switch to the disaster recovery cluster and it is running, with the primary cluster suspended, only the fees for the disaster recovery cluster's use of CU duration will be charged.

Engine Kernel Version

Last updated：2024-07-31 17:47:29

DLC provides different kernel versions optimized for various use cases, with numerous features and performance enhancements. The available kernel versions are listed below.
If your scenario primarily involves interactive queries, it is recommended to use the Presto engine and SparkSQL engine with the latest kernel versions.
If your scenario primarily involves batch jobs, it is recommended to use the Spark job engine with the Spark 3.2 kernel version.

Engine Type	Kernel Version	Description
Presto	SuperSQL-P 1.0	Based on the native Presto 0.242 version, this implementation supports dynamic data source loading, enhanced Dynamic Filter, Iceberg V2 tables, INSERT OVERWRITE for non-partitioned tables, and execution of Hive UDFs.
SparkSQL	SuperSQL-S 1.0	Based on the native Spark 3.2 version, this implementation supports Iceberg 1.1.0, Hudi 0.12.0, and Adaptive Shuffle Manager.
	SuperSQL-S 3.5	Based on the native Spark3.5 version, this implementation supports Iceberg 1.5.0 and Adaptive Shuffle Manager. The current beta version is backward compatible with various SQL and data governance tasks of SuperSQL-S 1.0, providing a performance improvement of more than 33% over the S1.0 version.
SparkBatch	Spark 3.5	Based on the native Spark3.5 version, this implementation supports Iceberg 1.5.0, Python3 and Adaptive Shuffle Manager. The current beta version is backward compatible with various SQL, jar, pyspark and data governance tasks of Spark 3.2, with a performance improvement of more than 33% over Spark 3.2.
	Spark 3.2	Based on the original Spark3.2 version, this implementation supports Iceberg 1.1.0, Hudi 0.12.0, Python3, and Adaptive Shuffle Manager.
	Spark 2.4	Based on the native Spark2.4 version, this implementation supports Iceberg 0.13.1, Python2, and Python3.

Engine Network Configuration

Last updated：2024-07-31 17:47:50

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

Network Configuration Type

Based on different business scenarios, Data Lake Computing offers two types of network configurations.
Enhanced Network Configuration: Suitable for situations requiring high-speed, stable access to data within a single VPC.
Caution
Data engines of non-Spark job types can only be bound to one Enhanced Network Configuration.
Cross-origin Network Configuration: Suitable for cross-origin federated data queries requiring access to multiple VPCs. A data engine can be bound to multiple Cross-origin Network Configurations.

Network Configuration Status

Initial: The network configuration is being initialized, and the network is not yet effective.
Success: The network configuration is effective for the bound engine.
Failure: Network configuration failed, it can be deleted and reconfigured.

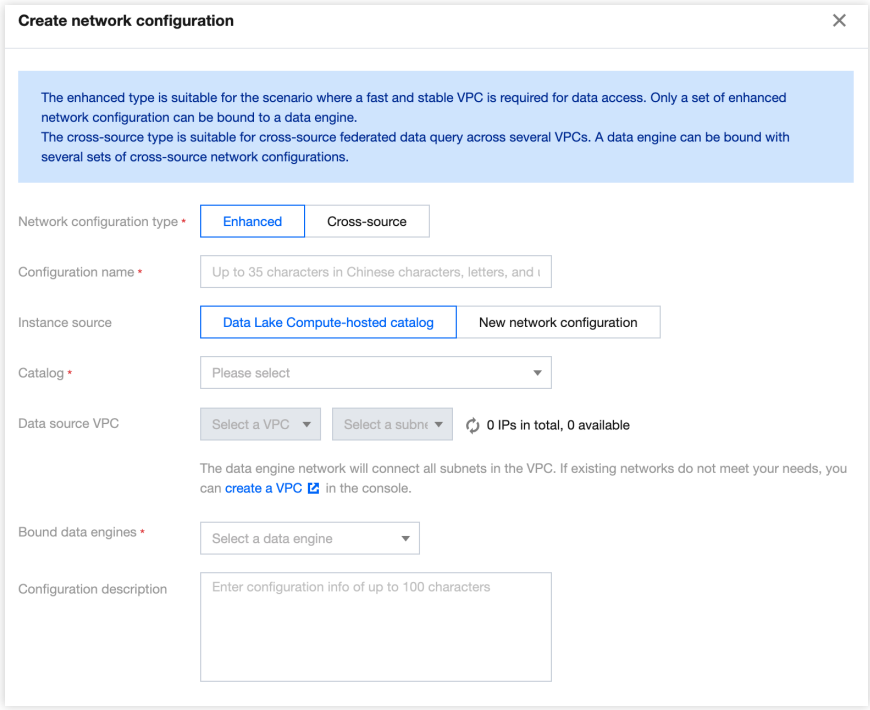
Network Configuration Security Policies

If you have configured a Security Group Policy for the VPC, inbound rules need to be added for different types of network configurations.
Enhanced Network: In the Security Group, add inbound rules for the IP range of the VPC where the data source is located.
Cross-origin Network: In the Security Group, add inbound rules for the IP range where the network configuration's bound engine is located.

Create Network Configuration

1. log in to DLC console, select the service region.

2. Access **Engine Management> Engine Network Configuration** through the left navigation menu.
3. Click the **Create Network Configuration** button to enter the creation page.



Configure parameters as follows:

Configuration	Required	Filling Instructions
Network Configuration Type	Yes	Select based on use case: Enhanced Network Configuration: Suitable for scenarios requiring high-speed, stable access to data within a single VPC Cross-origin Network Configuration: Suitable for scenarios involving cross-origin federated query analysis requiring access to data across multiple VPCs
Configuration Name	Yes	Supports Chinese, English, and _, with a maximum of 35 characters
Instance Source	Yes	Supports two sources: DLC data directory: You can select the data directory that has been created under DLC's Data Management New Network Configuration: Choose a new data source to create a network connection. Currently, supported data sources include MySQL, Kafka, EMR HDFS (COS, HDFS, Chdfs), PostgreSQL, SQLServer, and ClickHouse. If the data source required for the network configuration is not yet supported, select Other and manually specify the VPC
Data directory	Yes	Based on the selected instance source, choose the corresponding data directory. The range of available data directories will be related to your account

		permissions
Bind data engine	Yes	Select the data engine associated with this network configuration. If the data engine is in an isolated or initializing status, it cannot be selected
Configuration description	No	No more than 100 characters

4. Fill out and save to create a network configuration.

Caution

After creation, the network will be in an initialization state, and its status can be viewed in the list afterward.

Delete network configuration

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

1. DLC Console, select the service region.
2. Access **Engine Management> Engine Network Configuration** through the left navigation menu.
3. Find the network configuration you wish to delete. You can filter search results, but be sure to select the correct Network Configuration Type.
4. Click the **Delete** button. After a secondary confirmation, the deletion will be complete.

Caution

After deletion, the data engine will not be able to use this network configuration. If access is required, it must be reconfigured. Please proceed with caution.

Modifying description information

You can modify the description of an existing network configuration by following these steps:

1. DLC Console, select the service region.
2. Access **Engine Management> Engine Network Configuration** through the left navigation menu.
3. Find the network configuration you wish to delete. You can filter search results, but be sure to select the correct Network Configuration Type.
4. Click the **Modify description information** button to edit and modify.

Associating Tag with Private Engine Resource

Last updated：2025-01-03 15:27:27

Overview

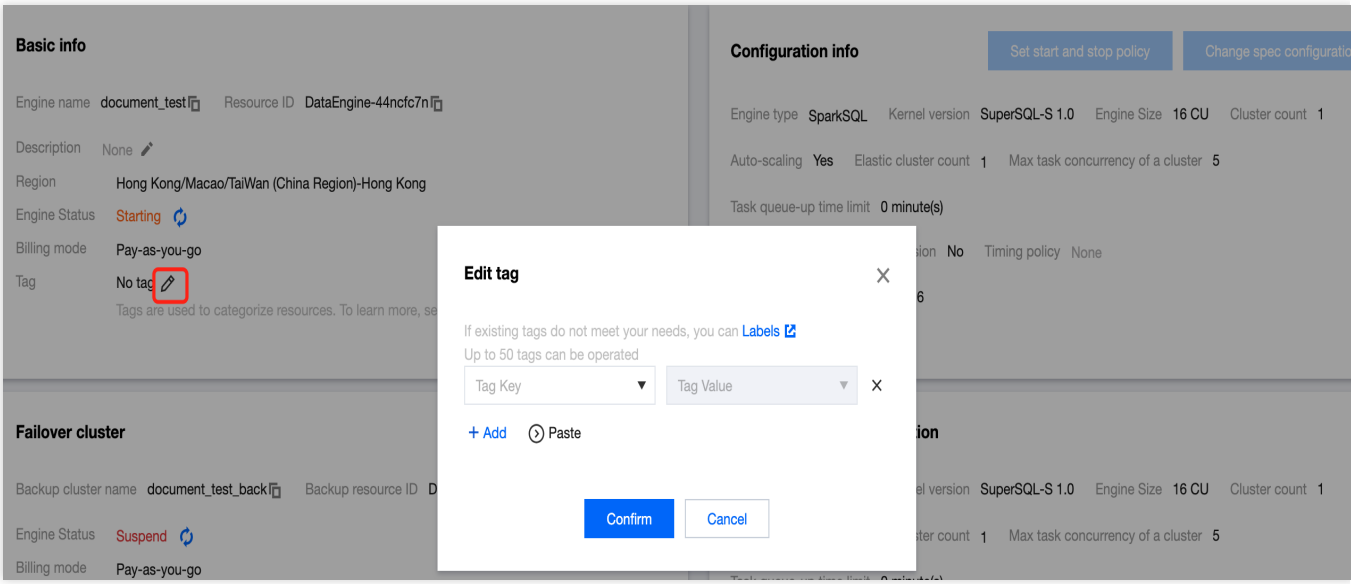
A tag is used to categorize and manage resources. It consists of a tag key and a tag value. A tag key can correspond to multiple values. You can create tags and bind them to cloud resources for easier management. Data Lake Compute supports binding tags to private engines in the console or on the purchase page, thereby enabling multidimensional category management and bill breakdown for private engine resources.

Creating a Tag and Binding a Resource

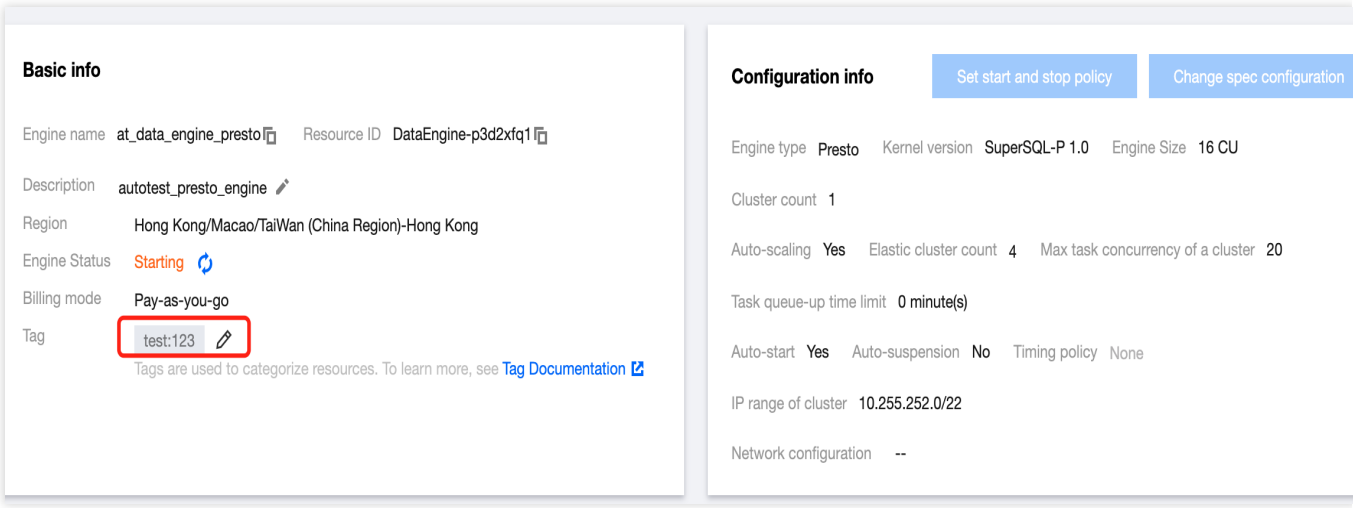
Create a tag and bind it to a private engine for resource categorization and unified management.

Directions

1. Log in to the Tag console to create a tag as instructed in Creating Tags and Binding Resources.
2. Log in to the Data Lake Compute console.
3. Click **SuperSQL Engine** on the left sidebar to enter the **Data engine list** page.
4. Click a resource name to enter the resource details page. Click **Edit** to pop up the tag edit window and select a tag for binding.

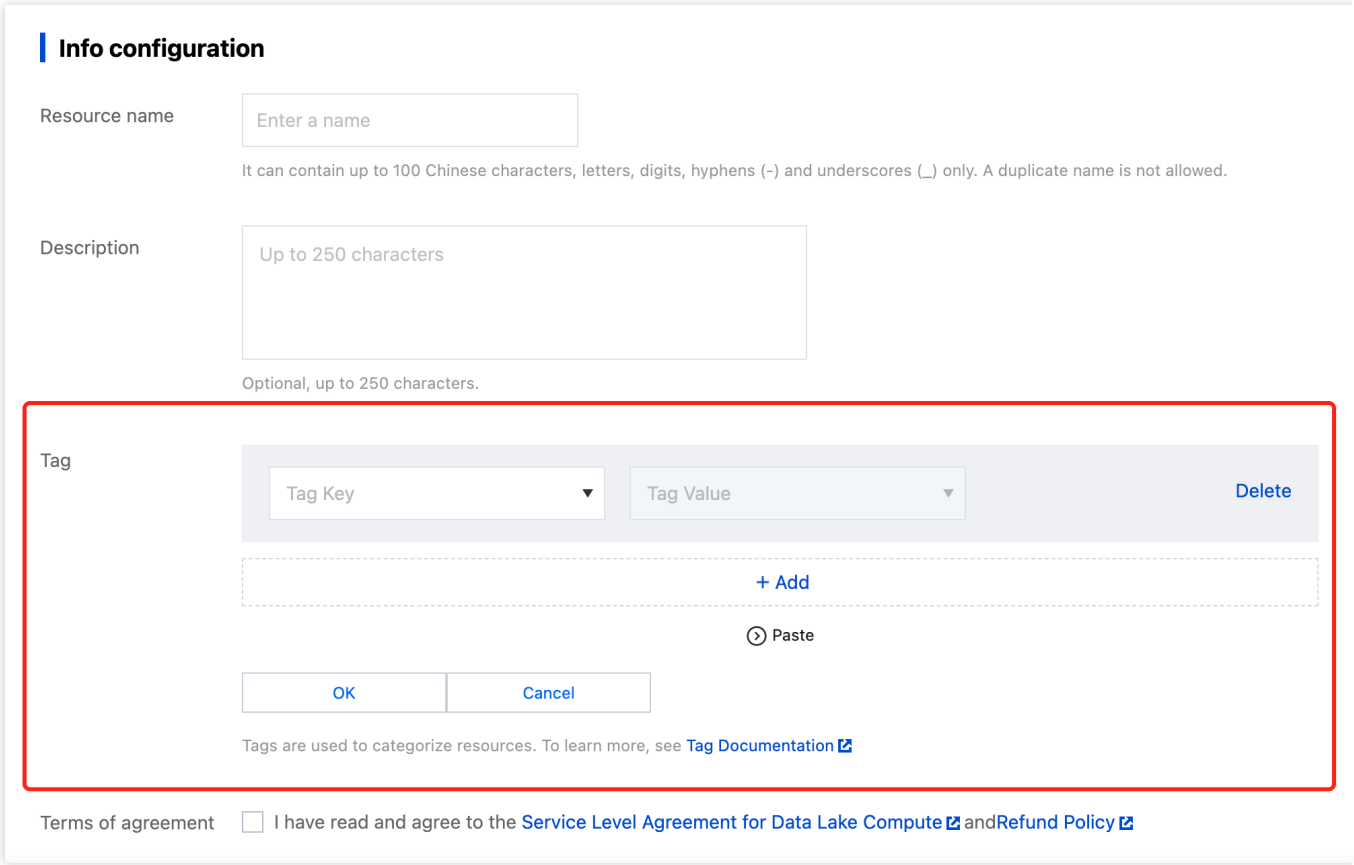


5. Click **Confirm** to bind the tag to the private engine. You can click **Edit** again to unbind or modify the tag.



Binding a Tag on the Purchase Page

You can bind a tag when purchasing a private engine resource in both monthly subscription and pay-as-you-go billing modes.

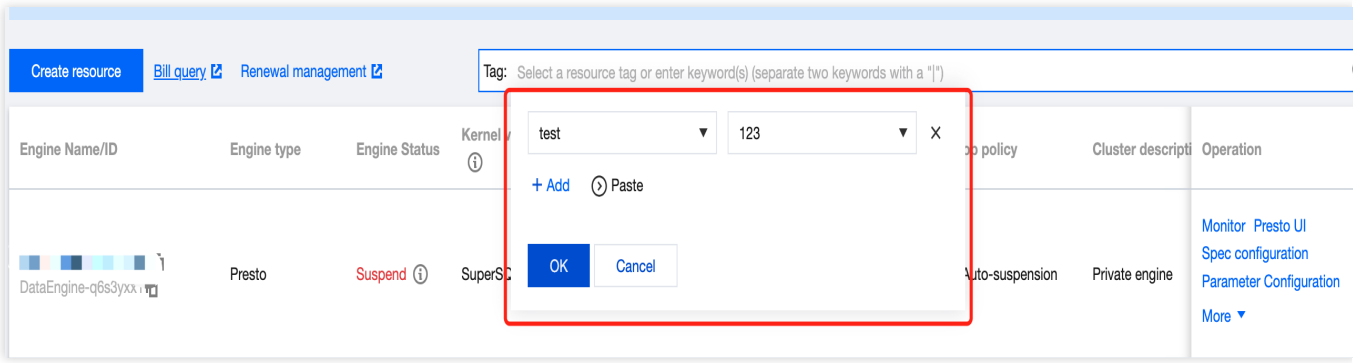


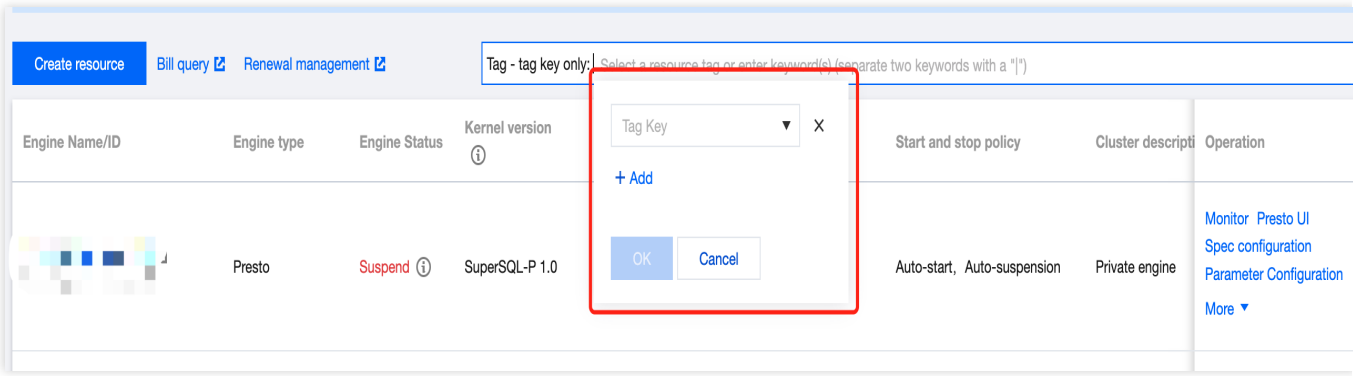
Filtering Resources by Tag

You can filter resources by tag on the **SuperSQL Engine** page in the Data Lake Compute console.

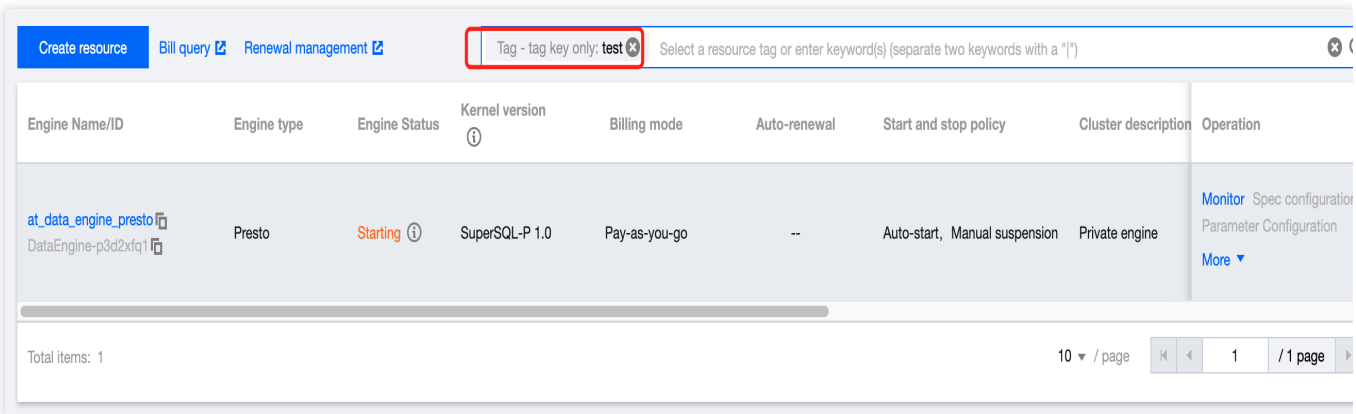
Directions

1. Log in to the Data Lake Compute console and select **SuperSQL Engine**.
2. Select a tag in the tag search box. You can filter resources by tag key or tag key-value.





3. Click the search icon to get the list of engines with that tag.



Allocating Costs by Tag

You can bind tags in the organization or business dimension for cost allocation by department, project team, region, etc.

Directions

1. Log in to the Tag console and create a tag.
2. Bind the tag to an engine resource in the tag console, on the **SuperSQL Engine** page in the Data Lake Compute console, or on the purchase page.
3. Go to the Billing Center to set a cost allocation tag. For more information, see Cost Allocation Tags.
4. Go to the Bill Overview page, select the aggregation by tag tab, and view the column chart and list of resources aggregated by tag key.

Engine Local Cache

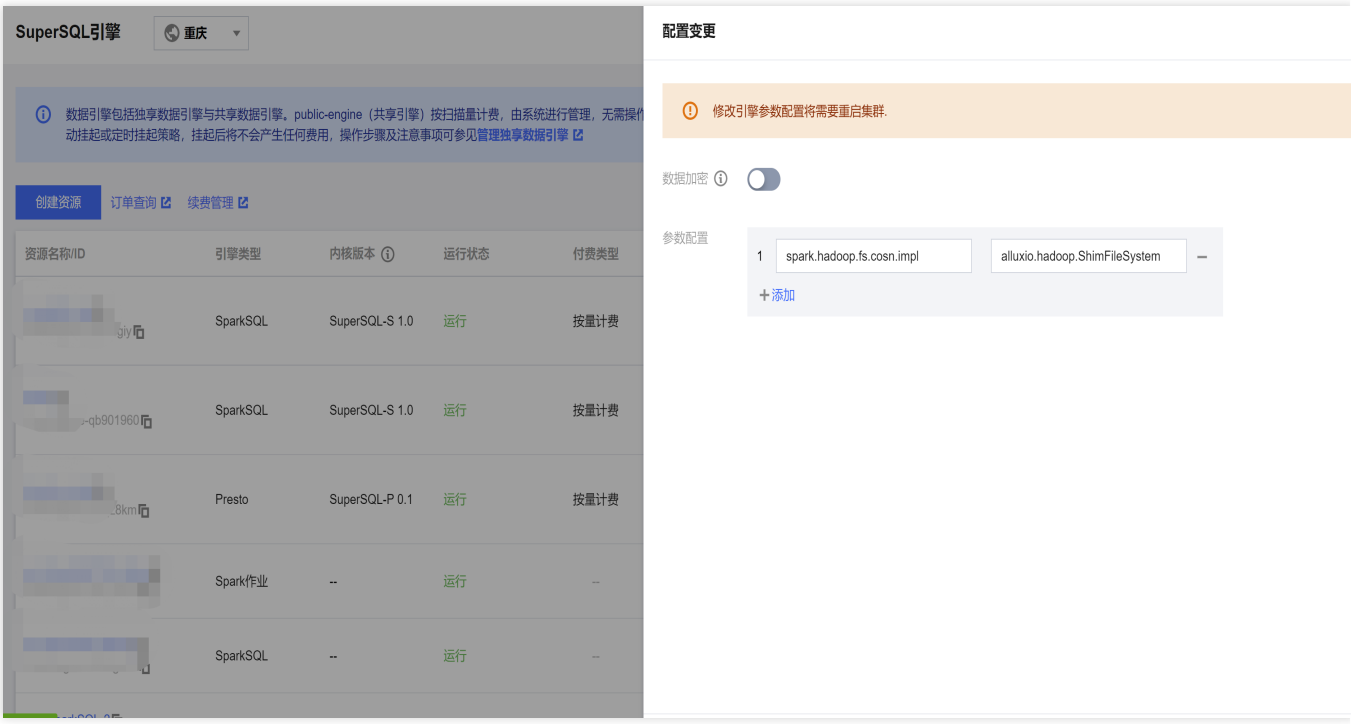
Last updated：2024-07-31 17:48:05

To ensure stable operation of Spark engine query analysis when network bandwidth is limited (e.g. during storage system throttling), the DLC Spark engine provides a local cache capability. When you need to cache table data, you can quickly enable caching by adding engine configuration.

Directions

1. Create a Spark Engine: For details, see Purchase Exclusive Data Engine.
2. Add Cache Configuration: Go to the DLC Console > Data Engine. Select the engine created in Step 1, click **Parameter Configuration**, and add the configuration items from Cache Configuration Item Explanation.

Spark SQL Engine Configuration:

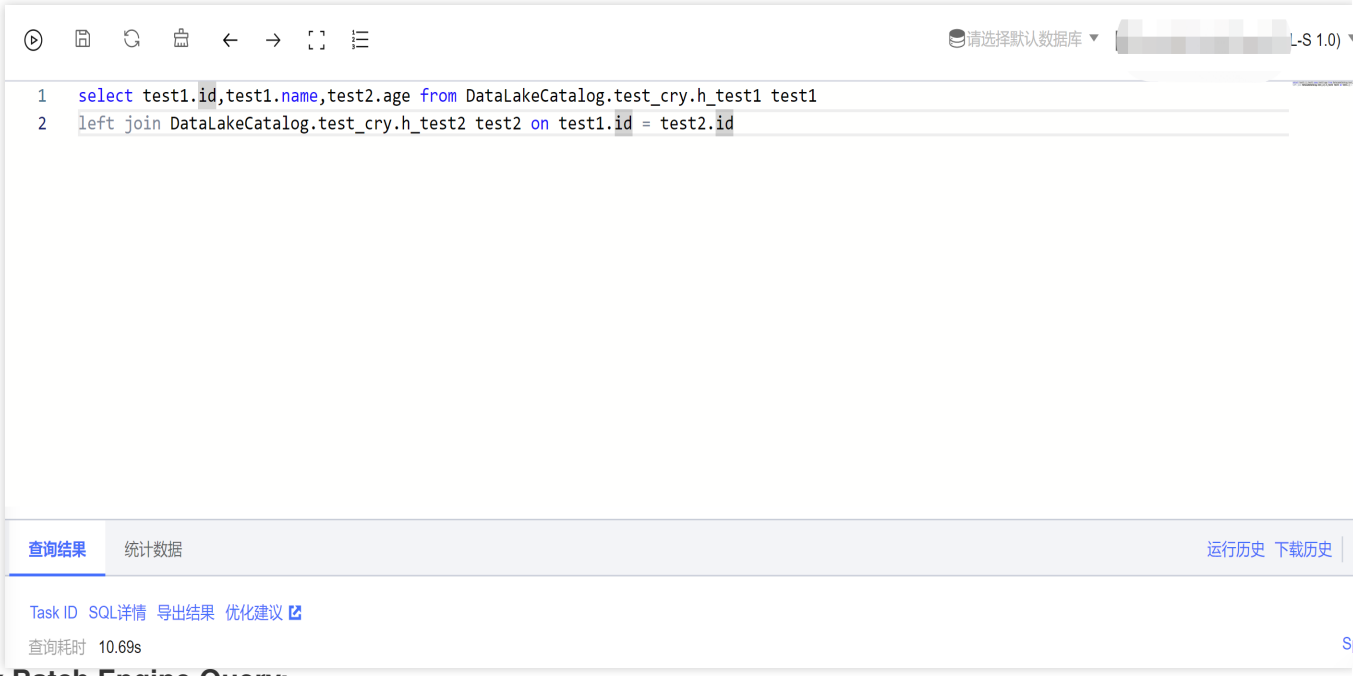


Note:

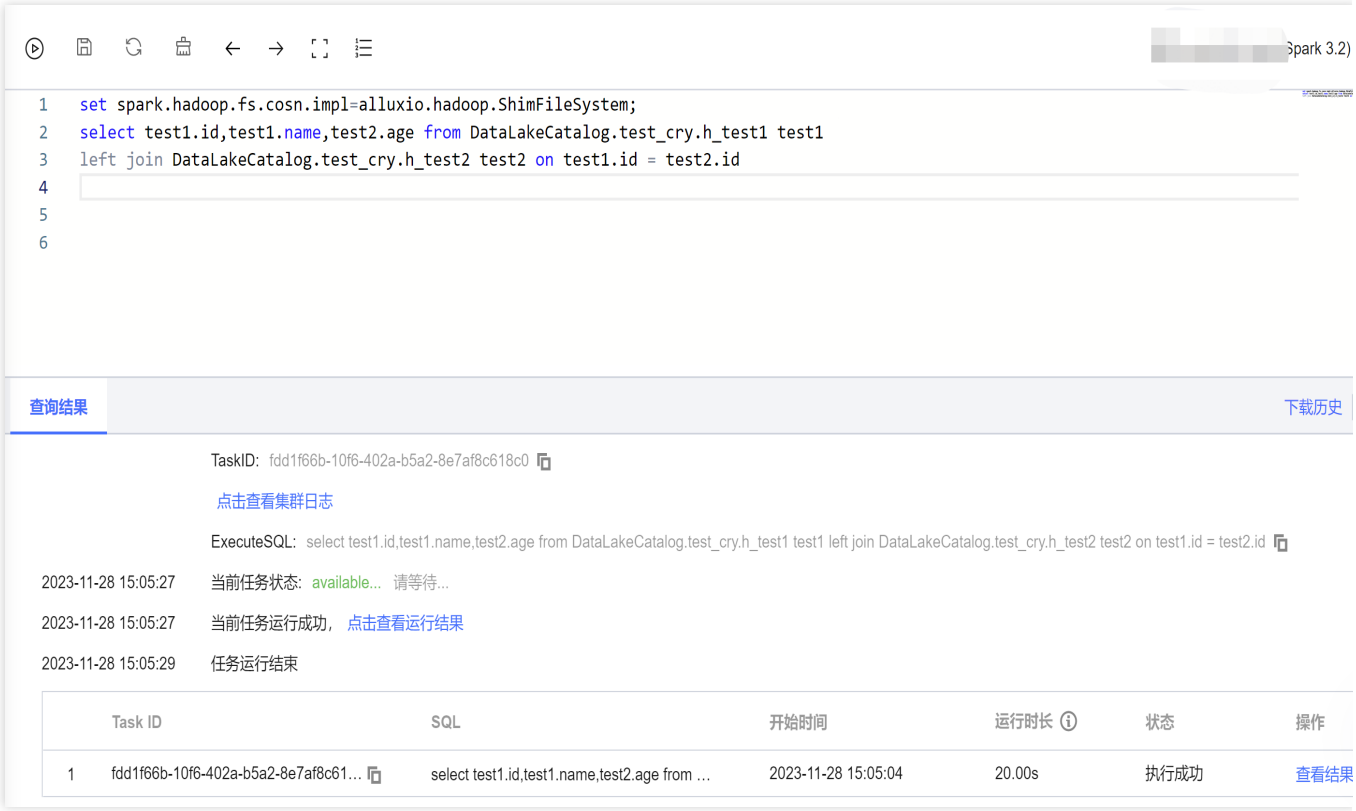
After the configuration is added, the engine cluster will restart. It is recommended to enable the cache when no tasks are running to avoid affecting ongoing tasks.

3. To use the engine cache, go to Data Exploration, write the query SQL in the SQL interface, select the engine with the cache enabled, and execute the SQL. Once executed, the engine will cache the DLC external table data locally. When the SQL is executed again, the data will be fetched from the local cache, improving query efficiency.

Spark SQL Engine Query:



Spark Batch Engine Query:



Cache Description

Cache Configuration Items Description

Configuration Items	Configuration Values	Configuration Items Description

spark.hadoop.fs.cosn.impl	alluxio.hadoop.ShimFileSystem	Fixed value; the configuration value is the cache implementation class. Configure this value to enable the cache feature. If the cache feature is enabled, configuring a value other than this will result in the engine not being able to access COS data. Please follow the instructions carefully. If you need to disable the cache after enabling it, please delete this configuration item.

Cache Usage Instructions

1. Engine Type Description

SparkSQL Engine: When the engine restarts, the cached data becomes invalid because it is a local cache.

SparkBatch Engine: The SparkBatch engine runs tasks at the session level. Once the task execution is complete, the cached data becomes invalid.

2. Table Type Description

Currently, only DLC external tables are cached.

Custom Task Scheduling Pool

Last updated：2024-07-31 17:48:18

Application scenario

Applicable Engine: Spark SQL Engine.
When you submit multiple tasks to the engine, for example, submitting multiple SQL tasks to the Spark SQL cluster simultaneously, the tasks submitted by the business may have dependencies, so the engine will default to scheduling these tasks in a FIFO manner when scheduling and executing.
However, in some special cases, you may need to define the priorities of certain tasks yourself, for example in the following scenario:
The submitted task has a high priority and needs to be executed with the highest priority, not wanting it to queue for cluster resources.
The submitted task has a low priority, hoping that it will not preempt resources from other tasks as much as possible. It will be executed when resources are available, and it will queue when resources are not.

Customize Scheduling Rules

In the Spark SQL Engine, each executed SQL task Job is split into a collection of multiple tasks, TaskSet, and our scheduling is based on TaskSet. Whenever the cluster has idle resources, it takes a Task from all Job's TaskSet according to the scheduling algorithm for dispatch execution.
Our scheduling algorithm is to define multiple scheduling pools, placing Job/TaskSet in the corresponding scheduling pool, and obtaining the Task that needs to be dispatched for execution according to the scheduling pool.

Scheduling Pool and Its Attributes

You can define multiple scheduling pools, each with four attributes:
name: The name of the scheduling pool, which you can name yourself. It can be named default, indicating the default scheduling pool.
schedulingMode: The scheduling rule, supporting two modes: FIFO and FAIR. The scheduling algorithm when there are multiple TaskSets within a scheduling pool.
FIFO: Tasks are dispatched in the order that TaskSets are submitted.
FAIR: Tasks from multiple TaskSets are dispatched fairly. The specific dispatch rules are related to the minShare and weight attributes of the scheduling pool.
minShare: The minimum number of cores required, must be greater than 0, that is, the minimum number of Tasks that can run. During scheduling, priority is given to the number of Tasks running in the scheduling pool reaching minShare.

weight: The weight. Scheduling pools with a higher weight will have their Tasks prioritized. Weight comparison will only occur after minShare is met.

The scheduling configuration requires you to write an xml file, in the following formats:

```
<?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 Example

You can refer to the settings for three scheduling pools:

Default Scheduling Pool default:schedulingMode = FIFO, weight = 1, minShare = (Cluster Cores - Driver Cores). This scheduling pool is the default submission pool for tasks, with ordinary priority. Execution is in sequential order, and it can utilize all of the cluster's computing resources.

Slow Task Scheduling Pool straggler:schedulingMode = FAIR, weight = 1, minShare = 1. This scheduling pool is dedicated to slow task submissions, with ordinary priority. Since minShare = 1, it does not preempt resources from tasks submitted to the default pool. Tasks in the straggler scheduling pool are executed when the cluster has more available resources.

High Priority Scheduling Pool special:schedulingMode = FIFO, weight = 1000, minShare = (Cluster Cores - Driver Cores). This scheduling pool is for tasks that need priority execution in special circumstances. However, due to the presence of minShare, this pool does not monopolize all cluster resources. Tasks in both the default and special pools continue to be executed, typically dispatching an equal number of Tasks from each pool.

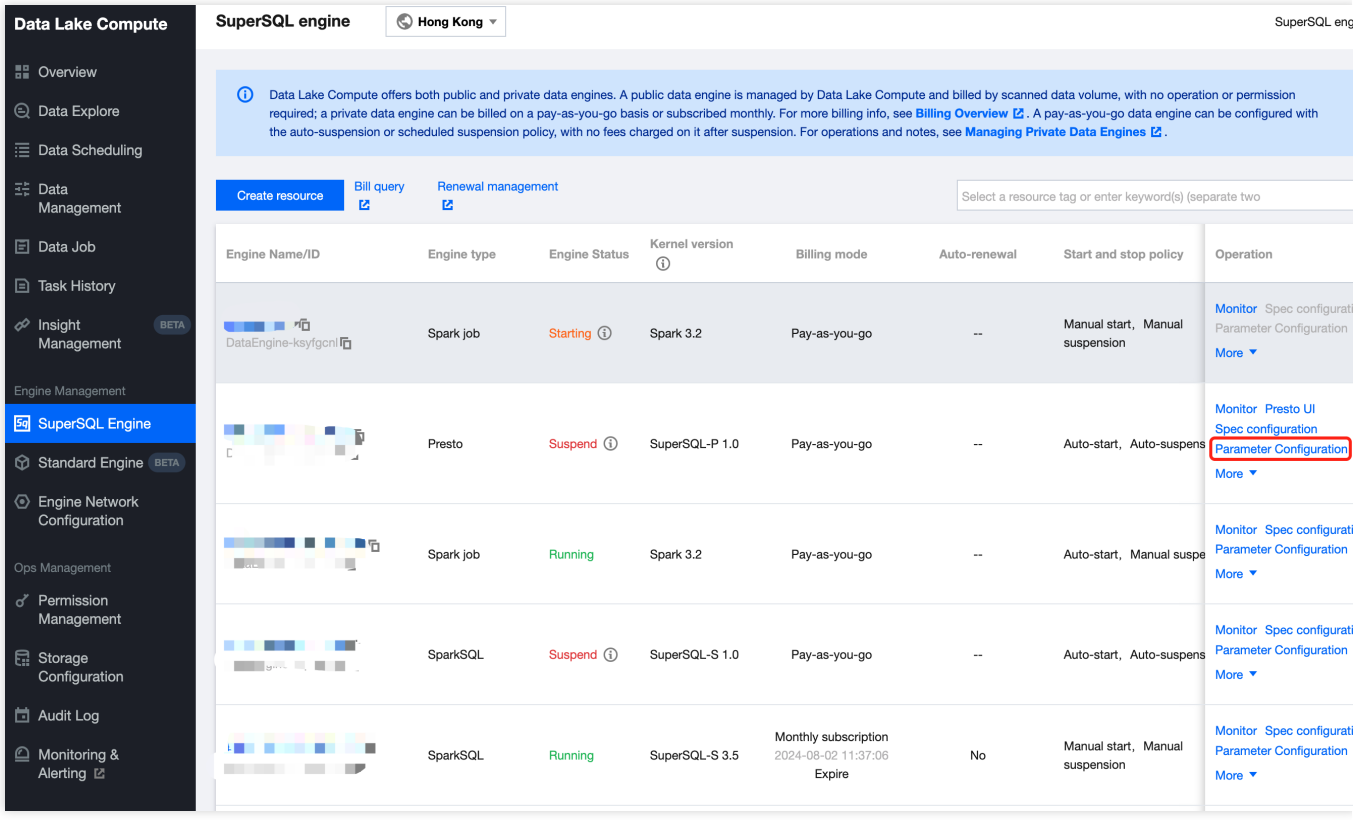
Taking a 16CU cluster (with the driver being 4CU) as an example, the configuration for this reference example is as follows:

```
<?xml version="1.0"?>
<allocations>
  <pool name="default">
    <schedulingMode>FIFO</schedulingMode>
    <weight>1</weight>
    <minShare>12</minShare>
  </pool>
  <pool name="straggler">
```

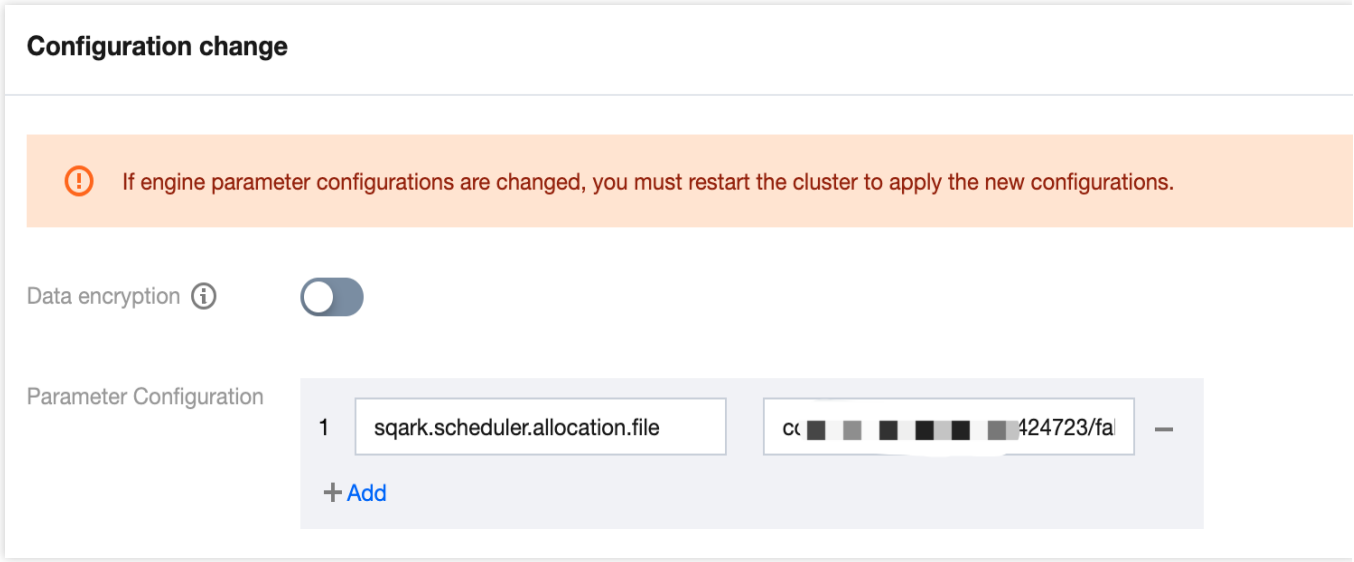
```
    <schedulingMode>FAIR</schedulingMode>
    <weight>1</weight>
    <minShare>1</minShare>
  </pool>
  <pool name="special">
    <schedulingMode>FIFO</schedulingMode>
    <weight>1000</weight>
    <minShare>12</minShare>
  </pool>
</allocations>
```

Operation method

1. After preparing the xml file for the scheduling pool, place it in a path on cos, for example cosn://bucket-appid/fairscheduler.xml.
2. Add the following configuration in the engine settings.

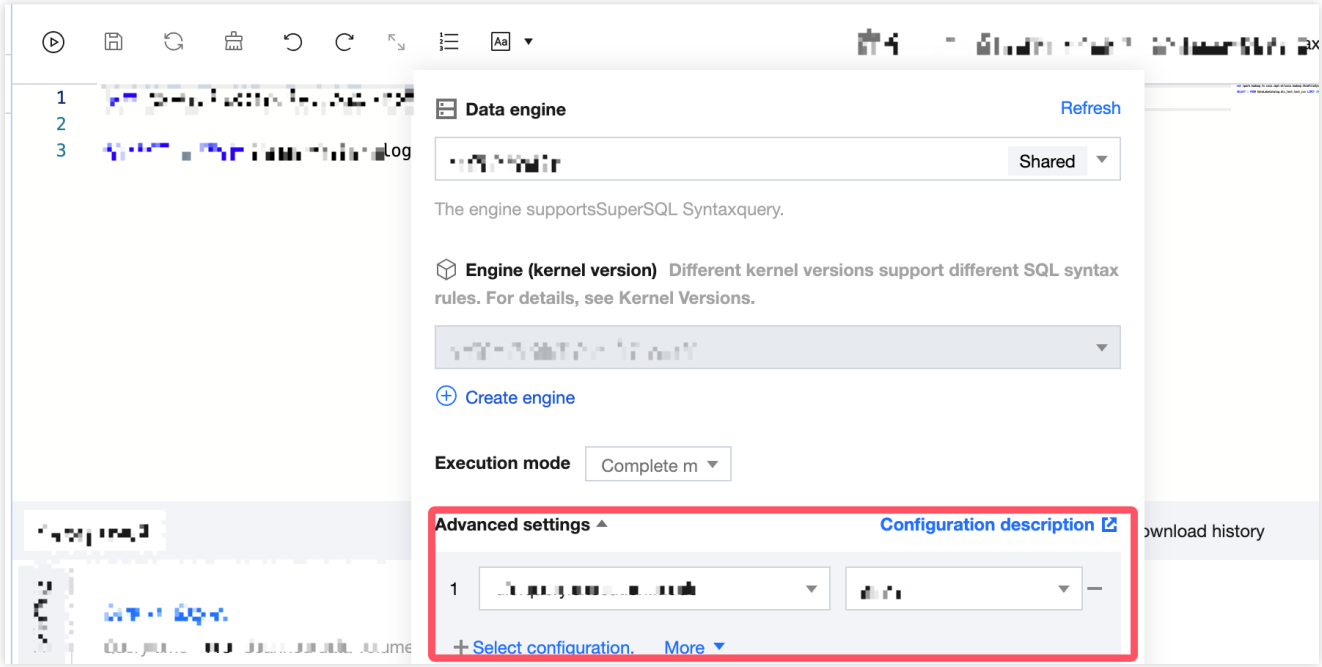


Parameter configuration spark.scheduler.allocation.file, set to the path of your scheduling pool xml file cosn://bucket-appid/fairscheduler.xml.



This operation requires restarting the cluster.

3. When submitting a task, specify the following parameters as task parameters: spark.scheduler.pool = the name of the scheduling pool to submit to. If it is the default scheduling pool, it does not need to be specified.



Notes

Scheduling occurs at the time node when: the cluster has idle resources and there is a task that needs scheduling. Therefore, if the cluster is already fully occupied by a task, for example, a slow task, it must wait for one Task of that task to be completed before beginning to schedule other tasks with higher priority. Therefore, it is important to note

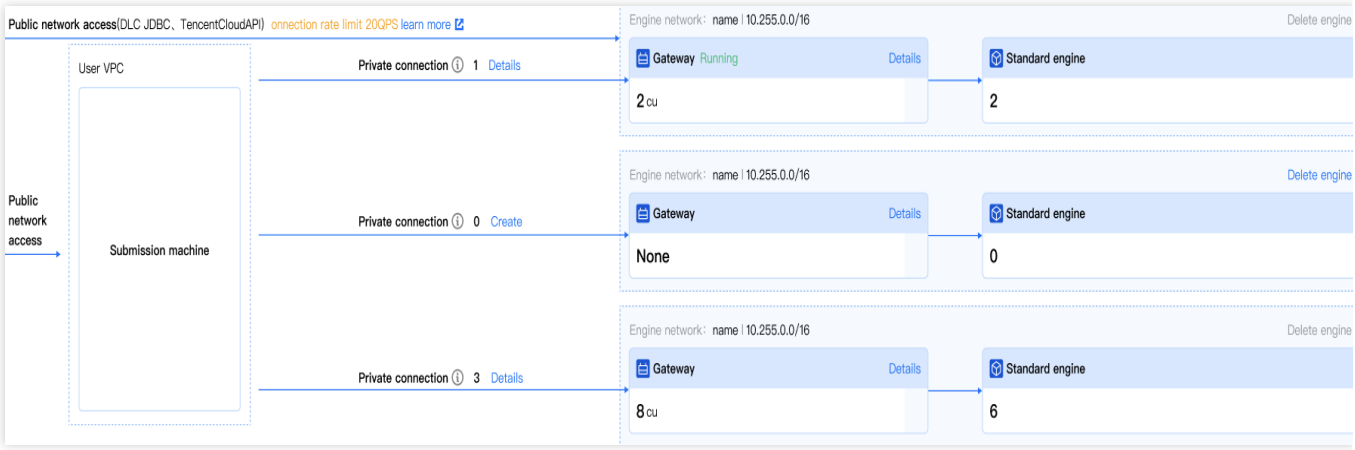
that the time consumption of a single Task of a slow task should be relatively reasonable; otherwise, it might still lead to long periods of occupying cluster resources.

Standard Engine

Introduction of the Standard Engine System

Last updated：2024-09-04 11:12:04

The Standard Engine system is composed of several key components: Engine Network, Gateway, Resource Group, Endpoint, and Executor. Before using the DLC Standard Engine, you should understand these concepts:



Concepts

The table below provides a brief introduction to several key concepts within the Standard Engine system. For more detailed information, you can click the relevant links.

Concept	Description
Engine Network	The Engine Network is a managed private connection that deploys the gateway and the Standard Engine within a logically isolated network environment. Users can customize the IP address range and subnet of the Engine Network according to their business needs.
Gateway	The gateway, implemented based on the Kyuubi big data component, serves as the access point for the Standard Engine services, providing users with a more efficient and stable task submission experience.
Standard Engine	The Standard Engine is a type of computing resource provided by DLC that helps users quickly launch compute clusters of a certain scale. It offers comprehensive support for native syntax and behavior, allowing users familiar with the big data ecosystem to get started more quickly and use the system with ease.
Resource Group	The Standard Spark Engine supports further on-demand division of engine resources through the use of resource groups. A resource group is a collection of a portion of the Standard Spark Engine's computing resources and corresponding configurations. SQL tasks can be submitted to a designated resource group for execution.

Private Link	Through a private connection, users can establish a link between their account's VPC and the Standard Engine's network, allowing tasks to be submitted via servers within that VPC.
Executor	After an endpoint is created, any server within the user account's VPC associated with that endpoint can serve as an executor for task submissions.

Task Submission Methods

Users can submit tasks in various ways:

1. Through JDBC on the executor, as shown in the diagram.
2. Submit SQL tasks via the Data Exploration page in the DLC console.
3. Submit Spark batch and streaming jobs via the Data Jobs page in the DLC console.
4. Submit tasks through the TencentCloud API.

Quick Purchase and Configuration of the Standard Engine

1. If you are purchasing the Standard Engine for the first time, DLC recommends following the Standard Engine Configuration Guide in the documentation to quickly set up the Standard Engine.
2. Once the purchase is completed, you can submit tasks via the Data Exploration page or the executor.

Standard Engine Introduction

Last updated：2024-09-04 11:13:49

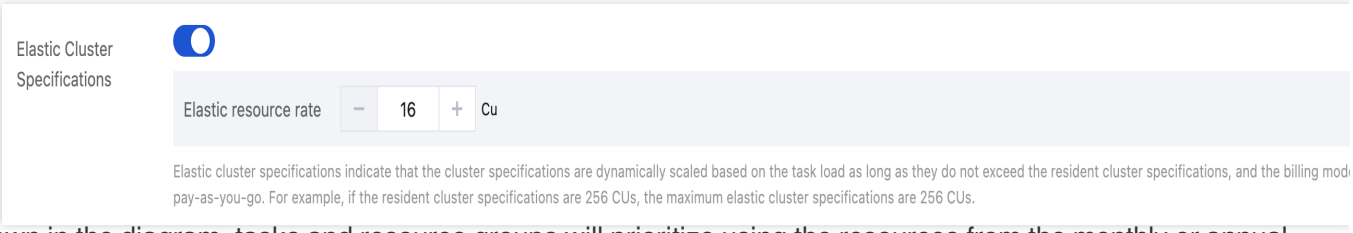
The Standard Engine is a type of computing resource provided by DLC that helps users quickly launch compute clusters of a certain scale. It offers comprehensive support for native syntax and behavior, enabling users who are familiar with the big data ecosystem to get started quickly and use it with ease.

Types of Standard Engine

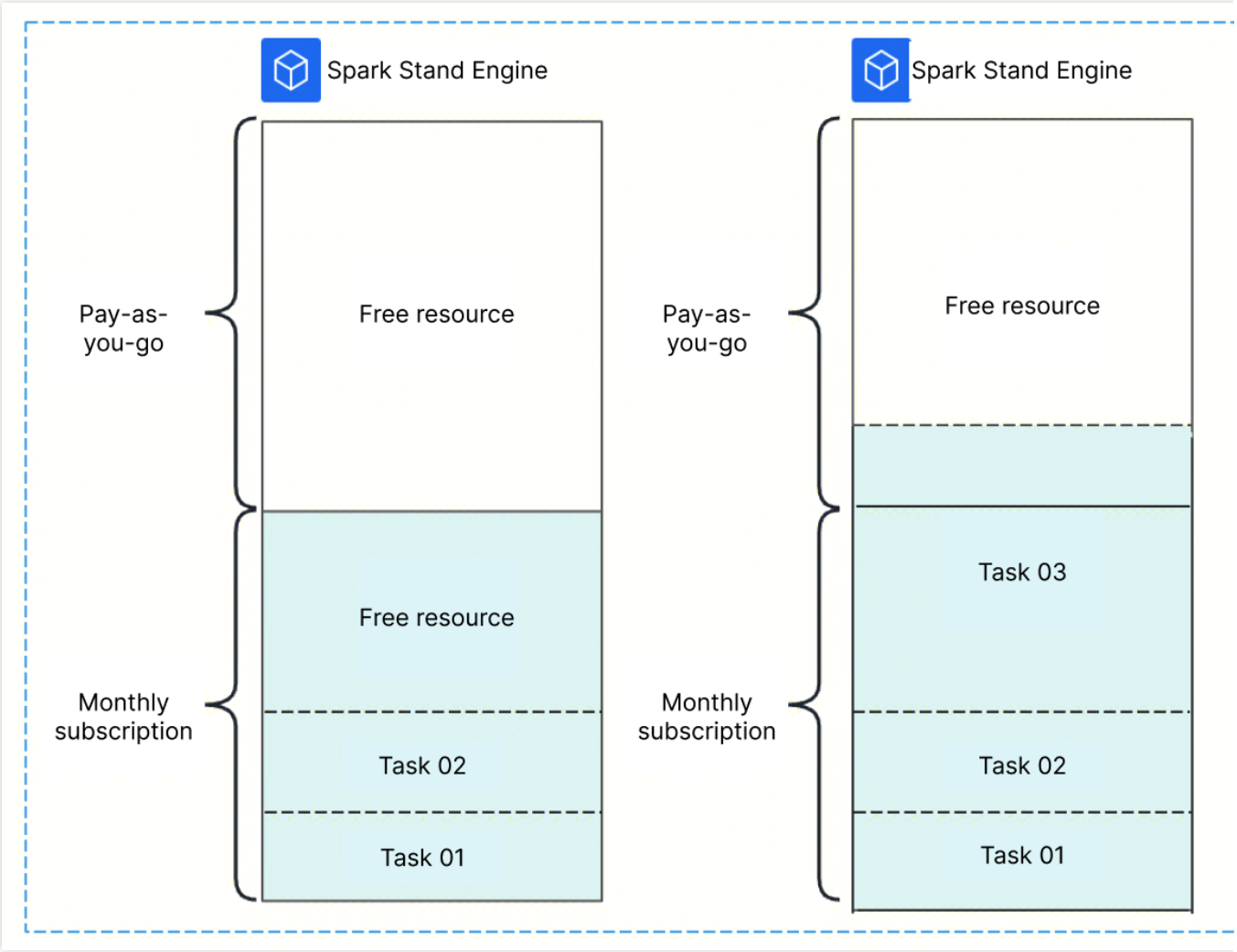
Users can choose different Standard Engine kernels based on their needs to address various use cases. The Standard Engine is divided into the following types:
Spark: Suitable for stable and efficient offline SQL tasks, as well as native Spark streaming/batch data processing jobs.
Presto: Suitable for agile and rapid interactive query analysis.
Gateway: The Gateway is a special type of Standard Engine implemented based on native Kyuubi. The Gateway is used to connect users to the Spark/Presto computing engines and submit tasks, serving as a prerequisite for using other computing engines.
Note:
Different types of engines do not affect the unit price of engine billing. For detailed pricing information, see the Billing Overview.

Engine Elasticity

Currently, only the annual subscription Spark Standard Engine supports the configuration of pay-as-you-go for resource elasticity.



As shown in the diagram, tasks and resource groups will prioritize using the resources from the monthly or annual subscription. If a user's submitted task exhausts the resources from this subscription, any subsequent tasks will automatically use the configured pay-as-you-go elastic resources. In the diagram, after Task 03 depletes the subscription resources, it continues to use the pay-as-you-go resources.



Note:

1. Pay-as-you-go elastic resources are charged based on the actual computing resources used.
2. If a task or resource group is scheduled to use pay-as-you-go resources, it will continue to use those resources even if the monthly or annual subscription resources are later freed up. The resource group will only be rescheduled to use the subscription resources after it has been restarted.
3. A single Spark Standard Engine cannot set elastic resources exceeding the amount of resources in the annual or monthly subscription. For example, a 128 CU annual or monthly subscription engine can set up to 128 CU of elastic resources. If you need to configure more elastic resources, contact us through a ticket.

Standard Engine Terminology

Terminology	Description
Cluster Type	When purchasing a Standard Spark Engine, you can choose the cluster type. The standard type is 1 CU ≈ 1 core with 4 GB memory, and the memory type is 1 CU ≈ 1 core with 8 GB memory. Different types have different unit prices. For more details, see the Billing Overview.

Elastic Cluster Specifications	The monthly or annual subscription Spark Engine allows users to configure elastic specifications. Once the resources from the subscription package are exhausted, the system will automatically allocate pay-as-you-go resources based on user configuration.
Gateway Name	The name of the gateway must be globally unique. It cannot share the same name as any other gateway or compute engine.
Engine Name	The name of the engine must be globally unique. It cannot share the same name as any other gateway or compute engine.
Engine Type	The Standard Engine types are categorized into Presto Engine and Spark Engine. The gateway is also a special type of Standard Engine.
Engine Status	The status of the Standard Engine varies based on the current operation of the cluster. The statuses include: Starting, Running, Ready, Paused, Pausing, Modifying, Isolated, Isolating, and Recovering. Starting: The cluster resources are being initiated. Pay-as-you-go for the engine does not occur during this time. Clusters in the starting status cannot be selected for data computation tasks. Running: The cluster is running and can be selected for data computation tasks. Ready: Similar to the running status, this status indicates that the engine is available for use. Paused: The cluster is paused and cannot be selected for data computation tasks. Pausing: The cluster is in the process of switching to the paused status. This transition may affect any running tasks, and the cluster cannot be selected for data computation during this time. Modifying: The cluster is undergoing configuration changes. During this period, it cannot be selected for data computation tasks. Isolated: The cluster has been isolated due to account arrears and cannot be selected for data computation tasks. Isolating: The cluster is in the process of being isolated due to account arrears. This transition may affect any running tasks, and the cluster cannot be selected for data computation during this time. Recovering: The process of restoring the cluster from an isolated status to a running status after the account has been recharged and is no longer in arrears. The cluster cannot be selected for data computation during this process.
Resource Group Count	The current number of resource groups under the Standard Spark Engine.
Used Resources / Total Resources	The quantity of resources currently used by the engine and the total available resources of the engine. The total resource count includes both the persistent resources and the elastic resources. Used resources include those occupied by the DLC deployment service system. There may be some delay in the reported data.
Payment Type	Payment types include annual/monthly subscription and pay-as-you-go. The gateway only supports the annual/monthly subscription model.

	The Standard Spark and Presto engines support both annual/monthly subscription and pay-as-you-go.
Auto-Renewal	Indicates whether the monthly or annual subscription engine will automatically renew as it approaches expiration.
Engine Size	The total available resources of the engine, measured in CUs. For monthly or annual subscription engines, the size includes both the engine's persistent capacity and the elastic capacity billed on a pay-as-you-go basis. **Note:** 1. For monthly or annual subscription engines, a one-time payment is required at the time of purchase. The engine's status does not affect billing costs. 2. For pay-as-you-go engines, charges are based on the user's usage: The Standard Presto Engine incurs charges while running, but not when suspended. Some costs may be incurred during the engine's startup phase. The Standard Spark Engine does not incur charges while in a ready status. Costs are only incurred when tasks are submitted or when a resource group is started and running.

Standard Engine Kernel Versions

Last updated：2024-09-04 11:14:22

The kernel versions used by the DLC Standard Engine are described as follows:

Engine Type	Kernel Version	Description
Spark	Standard-S 1.0	Standard-S 1.0 is a self-developed engine kernel based on Spark 3.2, compatible with native Spark syntax and behavior, and suitable for offline SQL tasks. It also supports Iceberg 1.1.0, Hudi 0.12.0, and Python 3, and includes support for Adaptive Shuffle Manager.
Presto	Standard-P 1.0	Standard-P 1.0 is a self-developed engine kernel based on Presto 0.242, compatible with native Presto syntax and behavior, and suitable for interactive query analysis. It also supports dynamic data source loading, enhanced Dynamic Filtering, Iceberg V2 tables, INSERT OVERWRITE for non-partitioned tables, and the execution of Hive UDFs.

Standard Engine Parameter Configuration

Last updated：2025-03-12 18:03:39

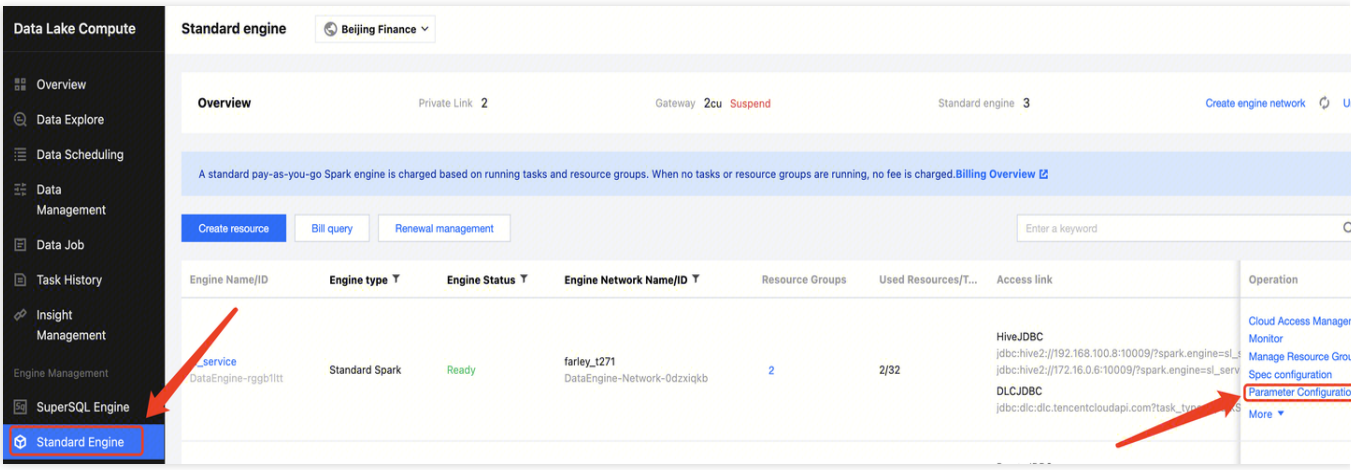
Spark parameters are used to configure and optimize settings for Apache Spark applications.
In a self-built Spark, these parameters can be set through command line options, configuration files, or programmatically.
In the DLC standard engine, you can set Spark parameters on the engine, which will take effect when users submit Spark jobs or submit interactive SQL using custom configurations.

Note:

1. The standard engine dimension configuration only takes effect for Spark jobs and Batch SQL tasks.
2. Only after the engine dimension configuration is added will the new tasks take effect.

Setting Standard Spark Engine Parameters

1. Enter the standard engine feature.
2. Select the engine that needs to be configured on the list page.
3. Click **Parameter Configuration** , and the engine parameter side window pops up.
4. In "Parameter Configuration", click **Add** , add the target configuration and then click **Confirm** .



Resource Group Dimension Parameters

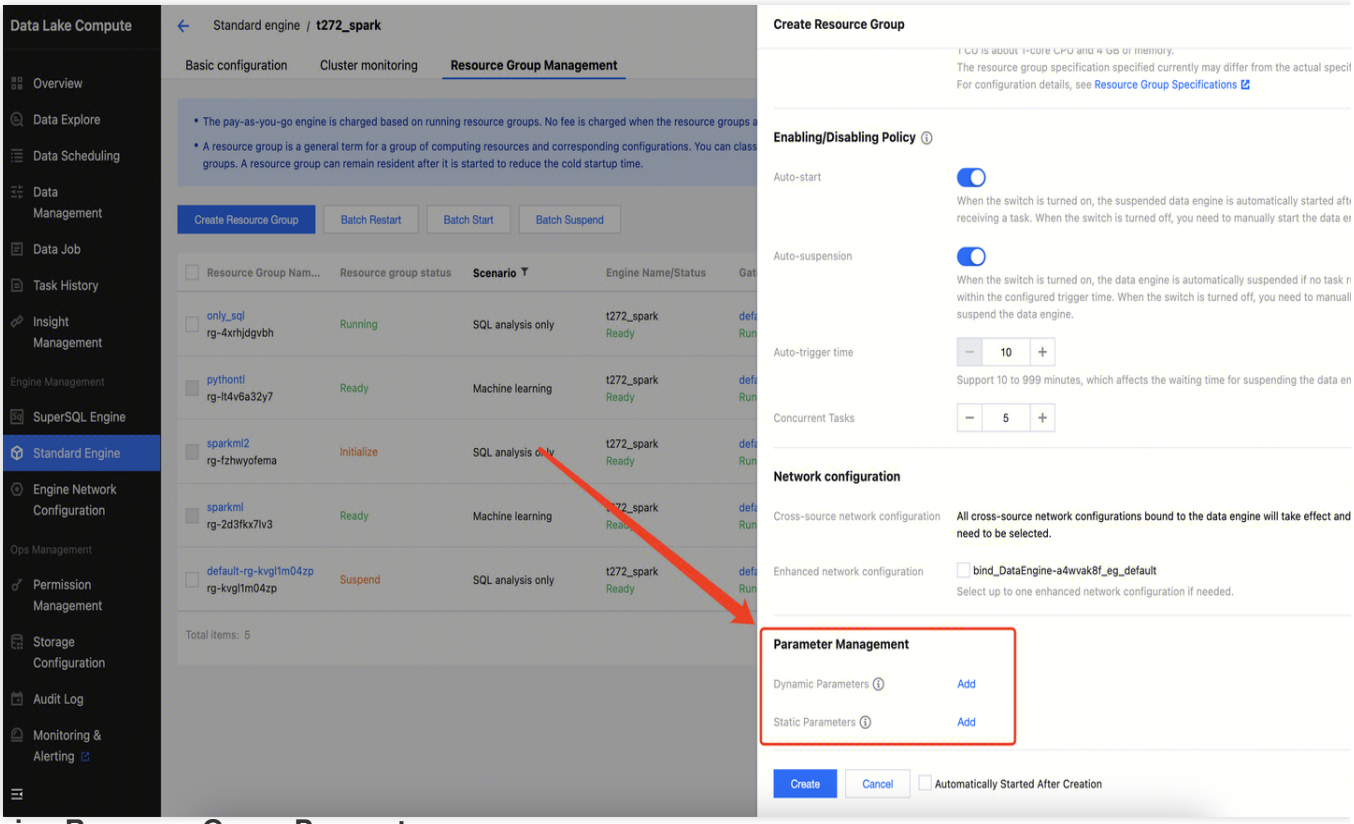
Parameters of Resource Group for SQL Analysis Only Scenario

Adding Parameters When a Resource Group Is Created

When a resource group is created, select SQL analysis only and add parameters in the Parameter Management at the bottom.

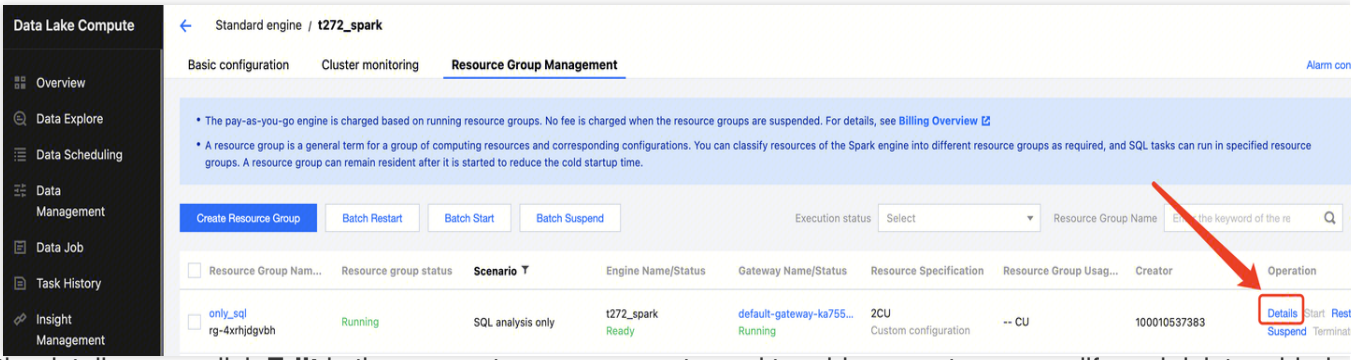
Note:

1. Static parameters can only take effect after the resource group restarts, while dynamic parameters do not require a restart of the resource group to take effect.
2. For details on dynamic parameters and static parameters, see the official website of Spark.
3. The configuration of resource group for SQL analysis only scenario takes effect only when SQL tasks are run using that resource group.



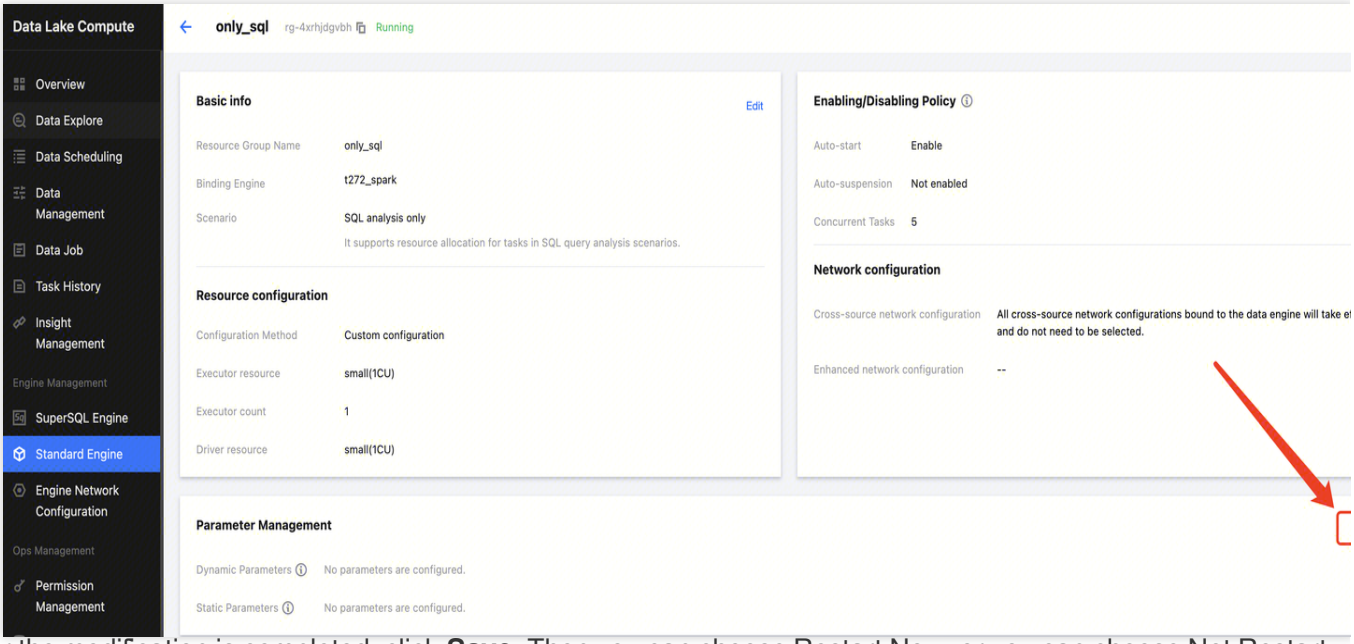
Modifying Resource Group Parameters

1. In Standard Engine List Page , select the engine to be modified and click **Enter** .
2. On the resource group management page, select a resource group for SQL analysis only scenario and click the **Details** button.

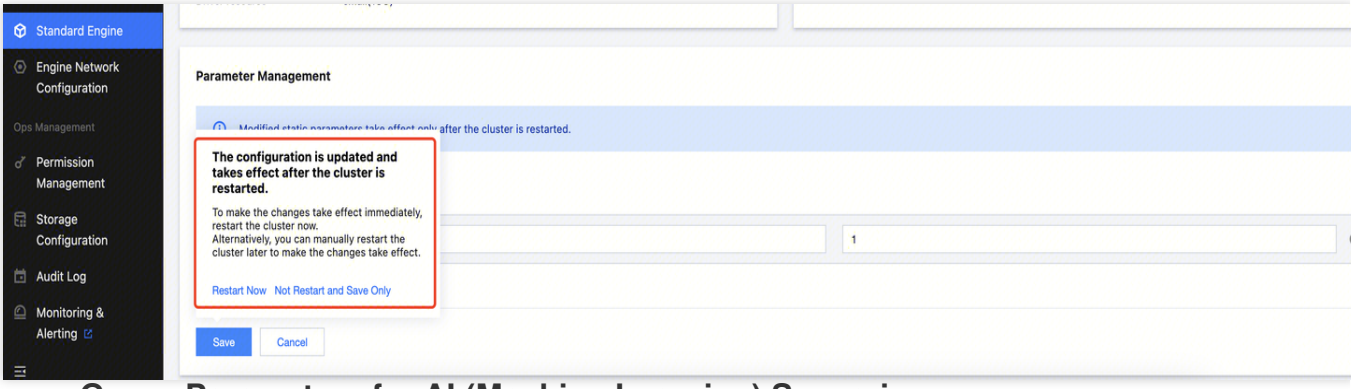


3. On the details page, click **Edit** in the parameter management panel to add parameters or modify and delete added parameters. Similarly, static parameters can only take effect after the resource group is restarted, while dynamic

parameters do not require a restart of the resource group to take effect.



4. After the modification is completed, click **Save**. Then you can choose Restart Now, or you can choose Not Restart and Save Only and then restart the resource group at an appropriate time later to make the configuration take effect.



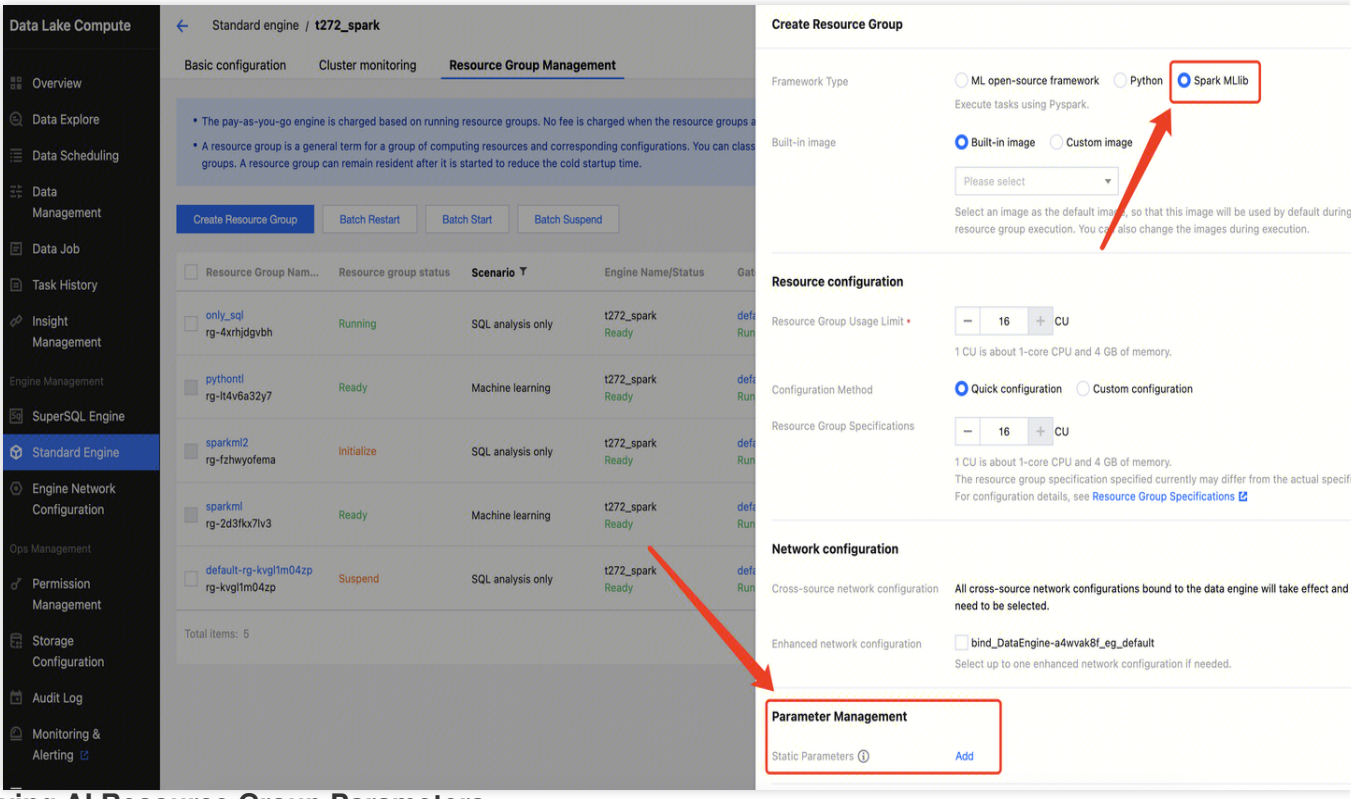
Resource Group Parameters for AI (Machine Learning) Scenario

Note:

1. Currently, only the Spark MLlib-type AI resource groups support adding configurations.
2. Currently, only static configurations can be added, which only take effect on new notebook sessions and do not take effect on existing sessions.
3. The AI Resource Group feature is a whitelist feature. To ensure that it meets your usage scenarios, please submit a ticket contact us for assessment and enablement.
4. This resource group only supports Standard Spark engine Standard-S version 1.1.

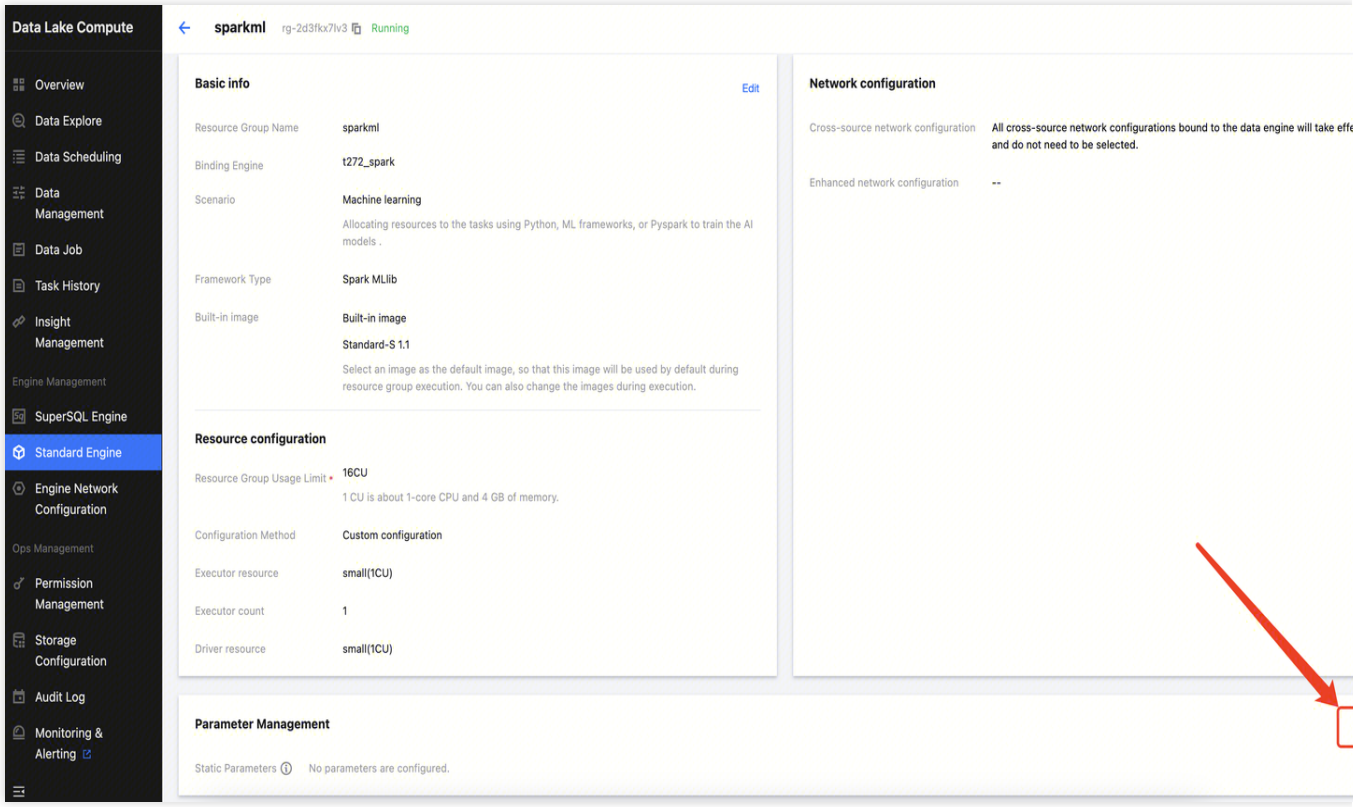
Adding Parameters When an AI Resource Group Is Created

As shown in the figure below, select the Spark MLlib type when the AI resource group is created, and choose to add parameters in the Parameter Management panel at the bottom.



Modifying AI Resource Group Parameters

1. In Standard Engine List Page , select the engine to be modified and click **Enter**.
2. On the resource group management page, select a Spark MLlib resource group and click **Details**.
3. On the details page, click **Edit** in the parameter management panel to add parameters or modify and delete the added parameters. Note that the modified parameters only take effect on the notebook session pulled after modification, and do not take effect on the existing sessions.

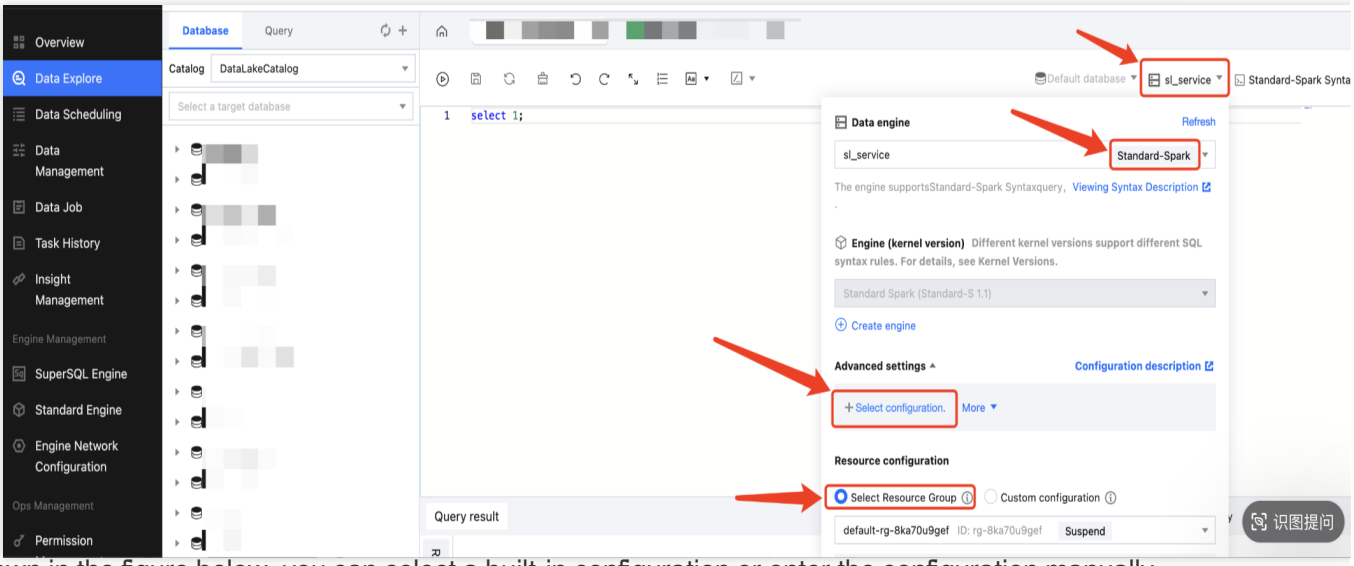


Data Exploration Parameters

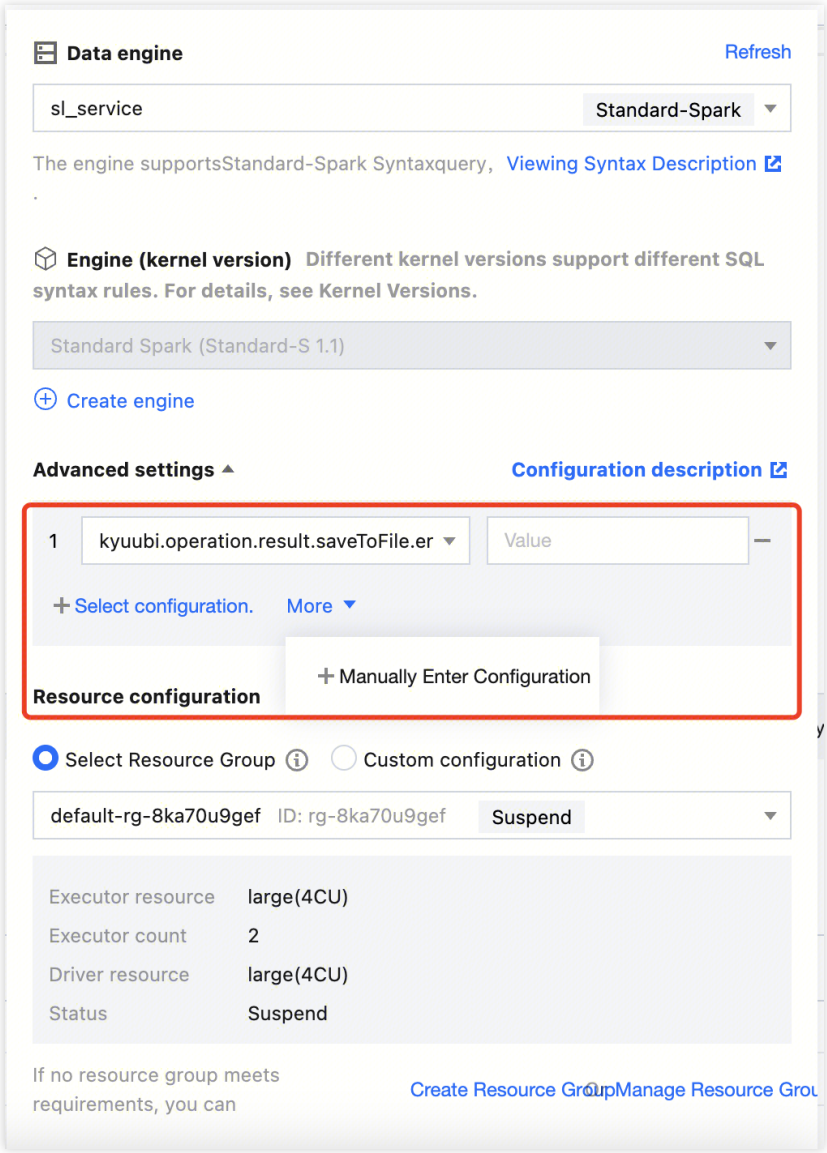
Note:

1. Currently, only the resource group for SQL analysis only scenario supports adding parameters on the Data Explore page.
2. Note that only dynamic Spark configurations take effect in the subsequent executions against SQL, and static parameters cannot take effect.
3. The parameter configuration at the data exploration level is of higher priority than that at the engine level and resource group level.

As shown in the figure below, on the Data Explore page, select the Standard-Spark engine for Data engine, select the option Select Resource Group for Resource configuration, and click Advanced settings on the page to add configurations.



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



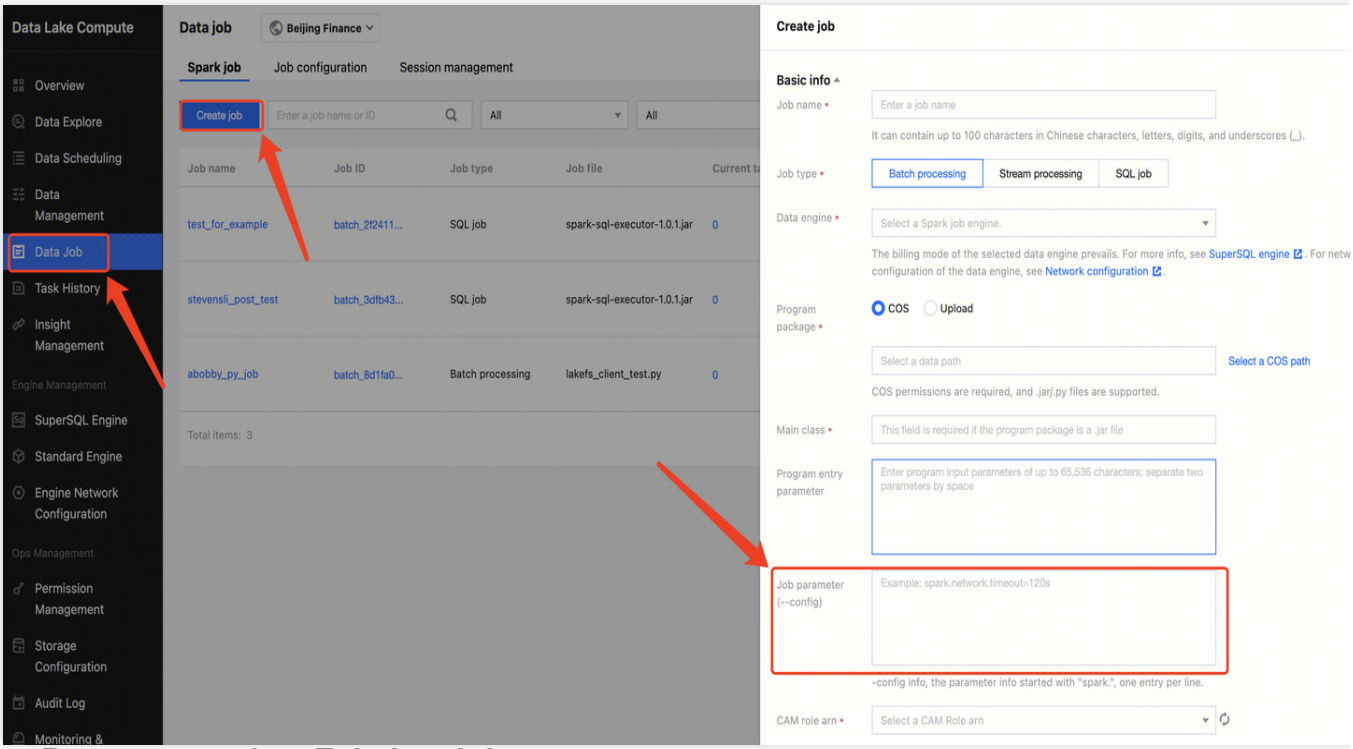
Spark Job Parameters

Note:

1. Modifications to job parameters only take effect in the jobs that are launched subsequently and will not take effect in the running jobs.
2. The priority of job parameters is higher than that of engine-level parameters.

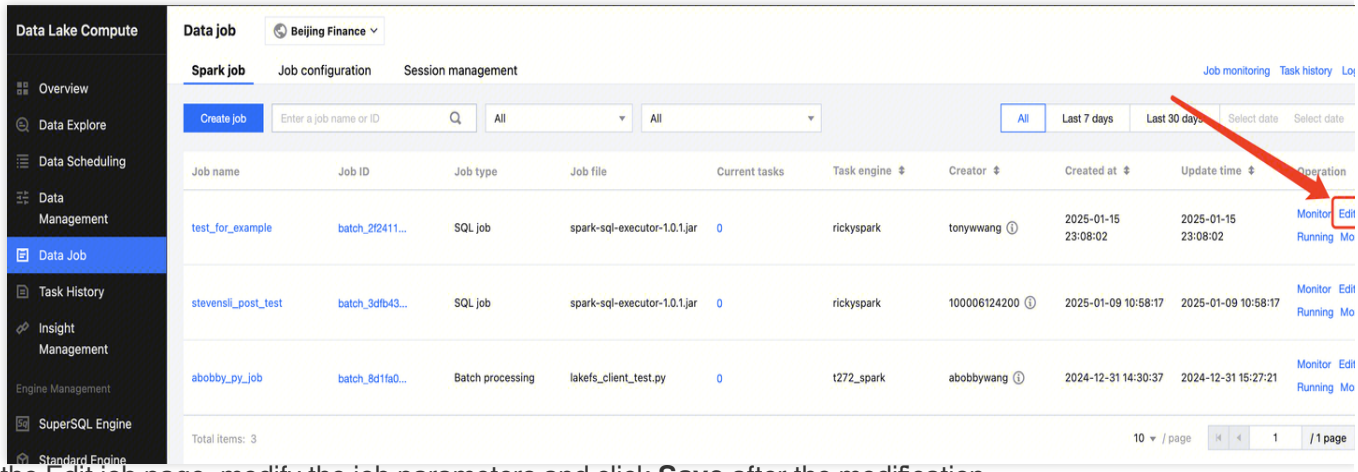
Adding Parameters When a Job Is Created

Enter Data Job, click **Create job**, and add parameters in Job parameter.

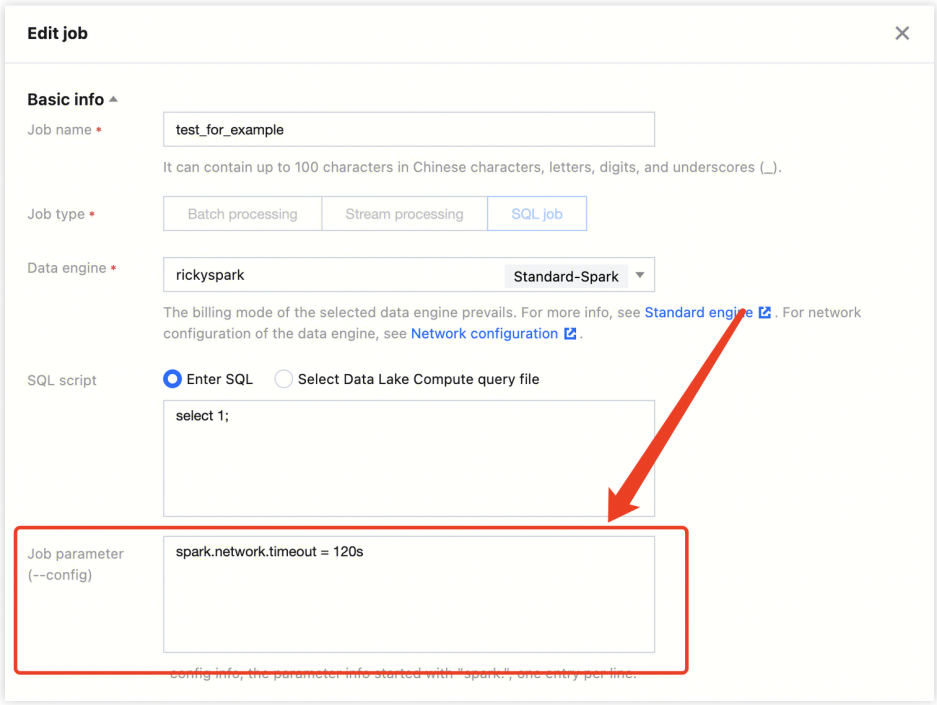


Editing Parameters of an Existing Job

1. Click Data Job,select an existing job and click **Edit**.



2. On the Edit job page, modify the job parameters and click **Save** after the modification.



Engine Network Introduction

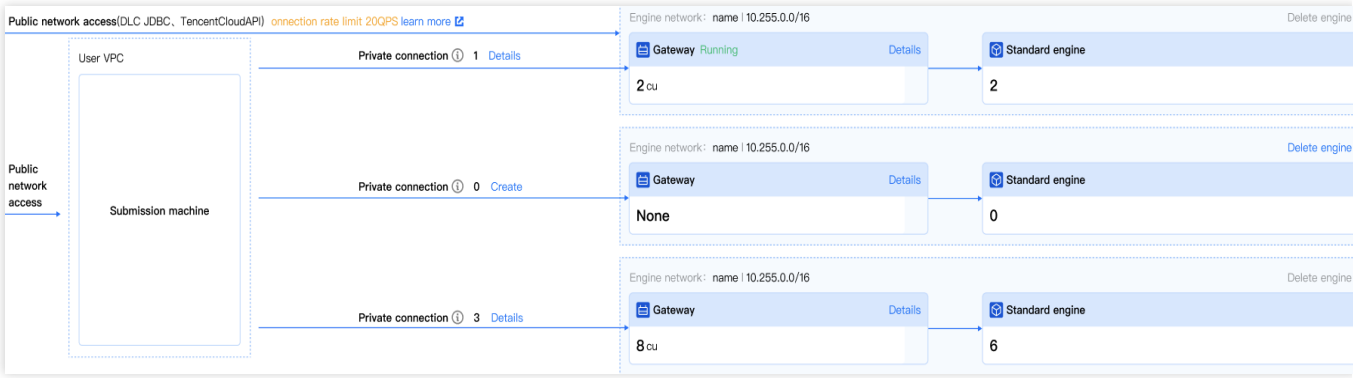
Last updated：2025-03-12 18:03:39

Concept

The engine network is built on a Virtual Private Cloud (VPC) and assigns computing engines (such as the standard Spark engine and the standard Presto engine) with fixed network addresses, for example, 10.255.0.0/16. Each engine network is provided with a gateway for external access to standard engines within the network. This allows computing engines to be accessed via JDBC from either a private network (VPC) or a public network.

Note：

If you need to access resources in different VPCs, such as using a DLC engine to access EMR HDFS data, it is recommended to select an IP range with sufficient available addresses that do not conflict with those used by other products. You can purchase multiple computing engines under the same engine network and manage them centrally through the gateway.



Use Limits

Note：

The IP range should be consistent with the VPC IP range settings and created manually. Once created, it cannot be modified.

1. Use any of the following private IP ranges:10.0.0.0 - 10.255.255.255 (mask range: 12-28)

172.16.0.0 - 172.31.255.255 (mask range: 12-28)

192.168.0.0 - 192.168.255.255 (mask range: 16-28)

2. Make sure that a subnet with sufficient IP addresses is allocated to the engine network to prevent IP address exhaustion, which could hinder Pod creation in large-scale workloads. If the required scale is uncertain, it is

recommended to use the default configuration.
3. When federated queries is used, ensure that the engine IP range does not overlap with the data source IP range.
4. Engine network configuration: Custom network settings can be configured during the initial purchase. To make changes later, submit a ticket to apply for that.

Network Segmentation

Standard engines under each engine network are managed by a gateway. Proper segmentation of engine networks helps balance the gateway load efficiently and mitigates the risk of single point of failure. We recommend segmenting networks based on business departments or task types.

Segmentation by Business

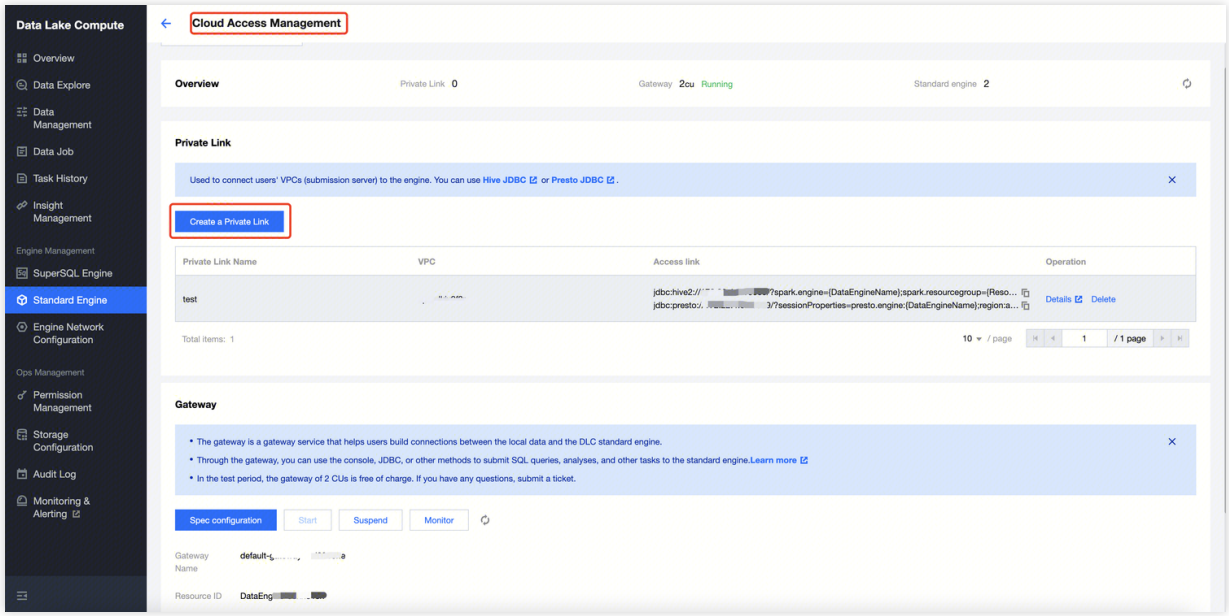
We recommend segmenting engine networks based on business departments. For example, each business department should have at least one engine network.

Segmentation by Task

We recommend segmenting engine networks based on task types. For example, you can create separate engine networks for different tasks such as BI analysis, data governance, and data analysis.
Note：
The above engine network segmentation recommendations are provided based on our experience for reference. You can also adjust the segmentation based on your actual needs, such as creating a dedicated engine network for handling of large-scale tasks according to the task scale.

Private Network Access

Creating a private link allows you to establish a secure and stable connection between your VPC and the gateway, enabling access to standard engines. On the Cloud Access Management page, you can create a private link, select the source VPC and subnet to be accessed, and obtain an access link upon completion. Any machine within the source VPC can then be connected to standard engines in the engine network.



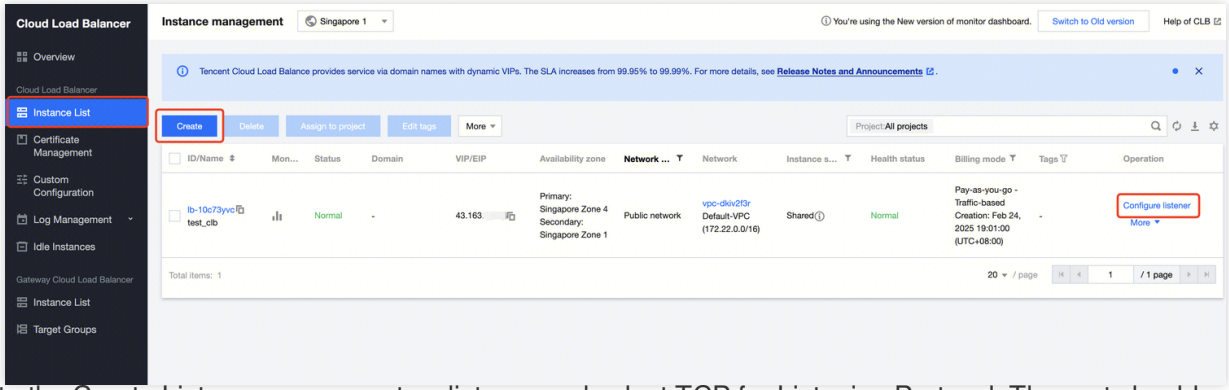
Public Network Access

Standard engines in the engine network can also be accessed via the public network. For example, certain BI tools deployed on the public network may require a public network connection to the engine.

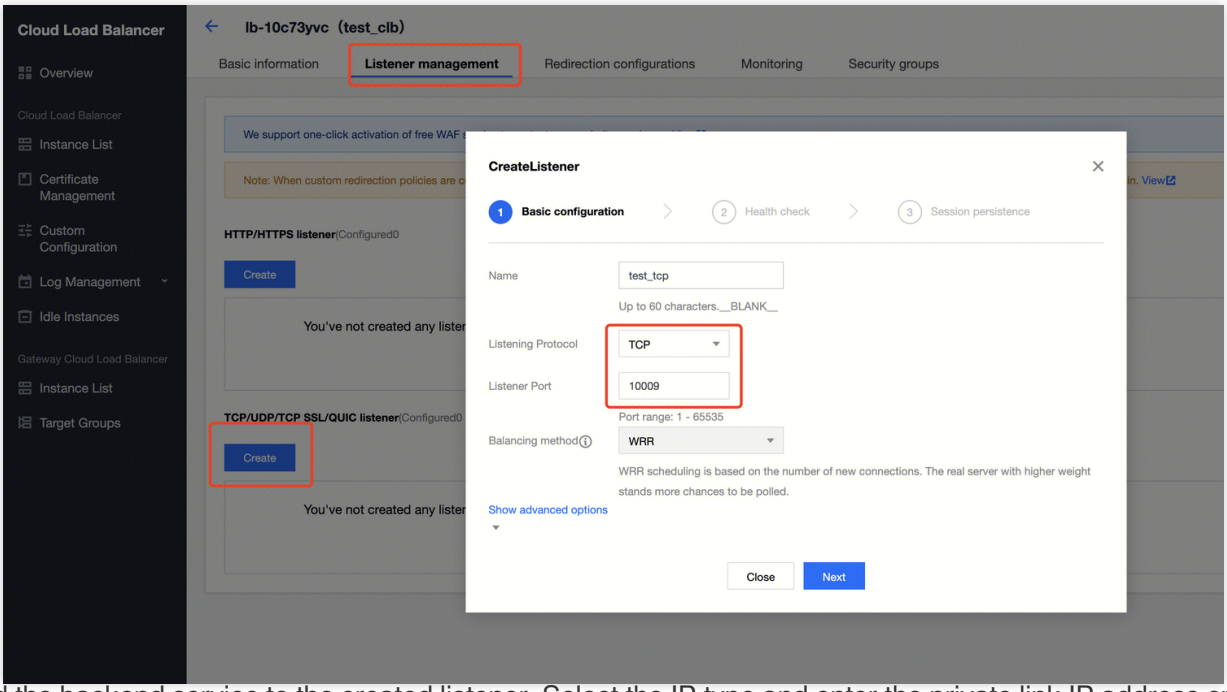
1. See Private Network Access to create a private link. For example: private network access JBDC link string.

```
jdbc:hive2://172.22.0.202:10009/?spark.engine={DataEngineName};spark.resourcegrou
```

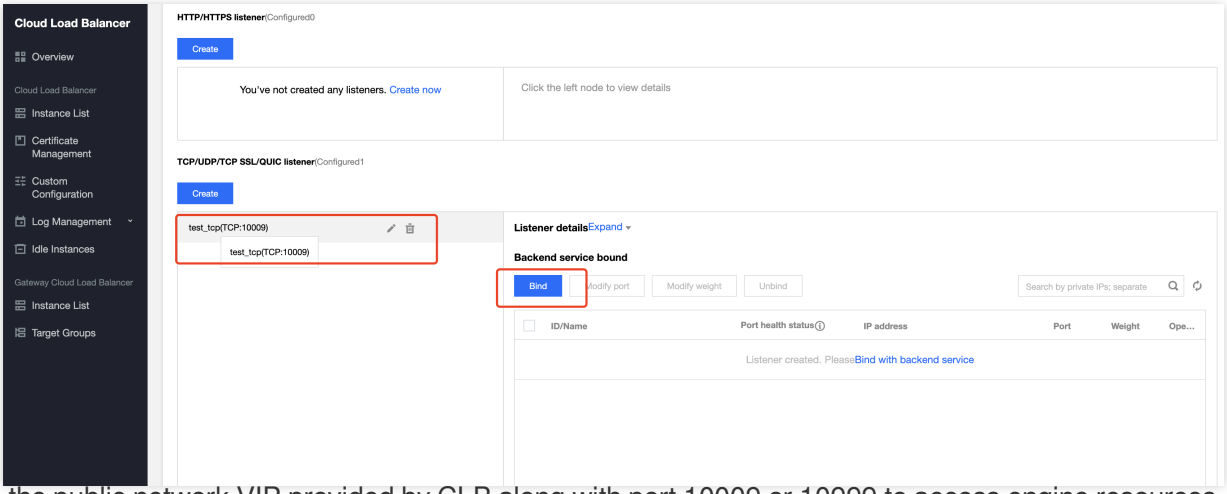
2. Go to the Cloud Load Balancer console, create a public network access instance, and select Configure listener.



3. Go to the Create Listener page, create a listener and select TCP for Listening Protocol. The port should match the private link port by default: 10009 (for accessing the standard Spark engine) or 10999 (for accessing the standard Presto engine).



4. Bind the backend service to the created listener. Select the IP type and enter the private link IP address created earlier, such as 172.22.0.202. Use port 10009 (for accessing the standard Spark engine) or port 10999 (for accessing the standard Presto engine).



5. Use the public network VIP provided by CLB along with port 10009 or 10999 to access engine resources. This converts the access link into a public network connection.

```
jdbc:hive2://{public network VIP}:10009/?spark.engine={DataEngineName};spark.reso
```

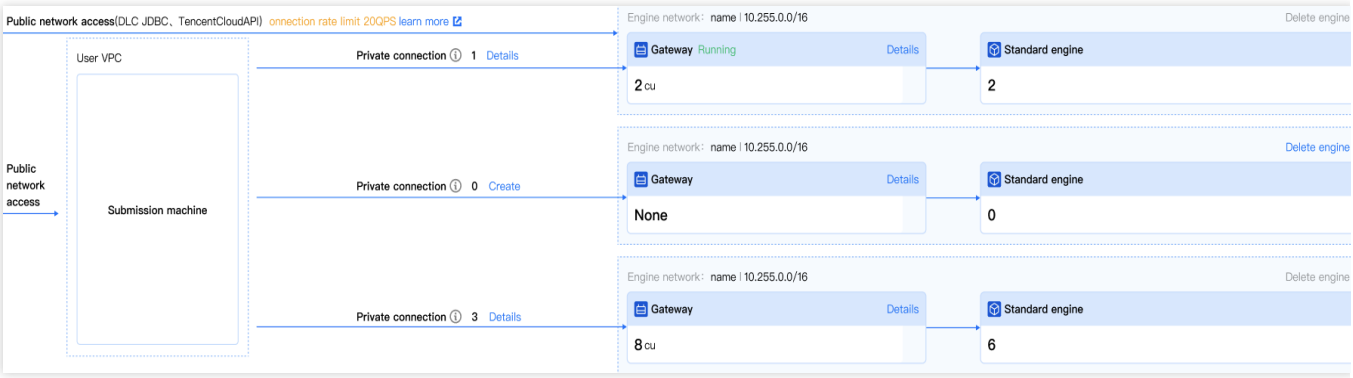
Accessing the Public Network in the Engine

By default, standard engines do not support public network access. If you need to access the public network, such as for installing Python packages in the notebook using magic %pip, submit a ticket to apply.

Gateway Introduction

Last updated：2025-03-12 18:03:39

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 computing engines based on Hive JDBC/Presto JDBC/DLC JDBC/TencentCloud API standard interfaces, reducing the complexity of managing access to large-scale computing engines. For example, you can submit SQL tasks and ETL jobs to specified standard computing engines through the gateway.



DLC Gateway

The gateway is a unique service of the DLC standard engine, offering users strengths such as reduced query latency, security and high availability, and flexible integration:

Reduced query latency: The DLC gateway can significantly reduce the time taken on the query link, and improve performance of data interactive analysis, especially for small data volumes.

Support for more access methods: The gateway supports Hive JDBC/Presto JDBC connects to the DLC standard engine, catering to various query scenarios.

Enterprise-level security: Identity authentication and sub-user engine permission control are performed through CAM authentication parameters (AK/SK).

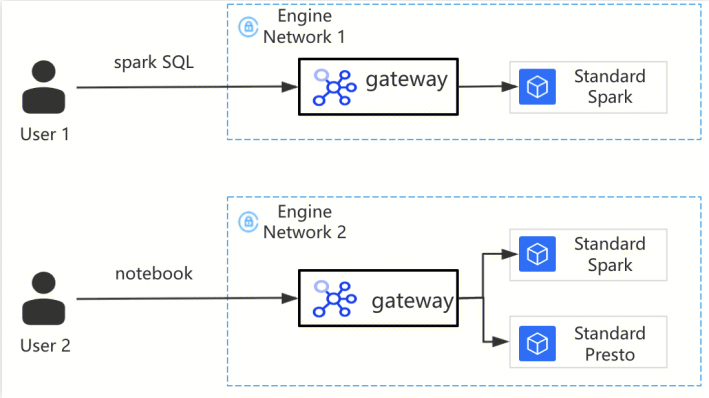
High availability: The gateway provides higher availability and load balancing and supports scaling out for extremely high query concurrency.

Architecture

As shown in the figure below, only one gateway can be created under an engine network. This gateway can simultaneously manage all standard Spark engines and Presto engines created under the engine network.
By default, a user can only have one engine network and can only create one gateway. If the business scenario is complex and there are high requirements for concurrency and other performances, or if some more important businesses require environment isolation, it is recommended that users create multiple engine networks and multiple gateways to physically isolate different tasks.

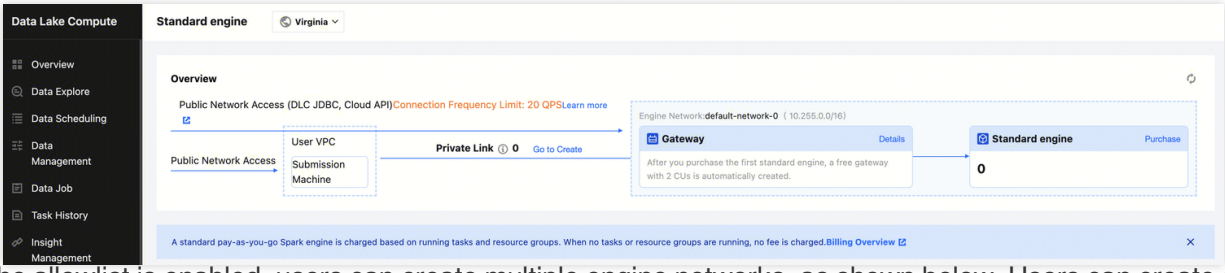
Note：

1. Creating multiple engine networks and gateways requires the backend to enable the allowlist. Contact DLC development personnel to conduct the operations.
2. Different engine networks and gateways are physically isolated and cannot communicate with each other or access each other's engines.



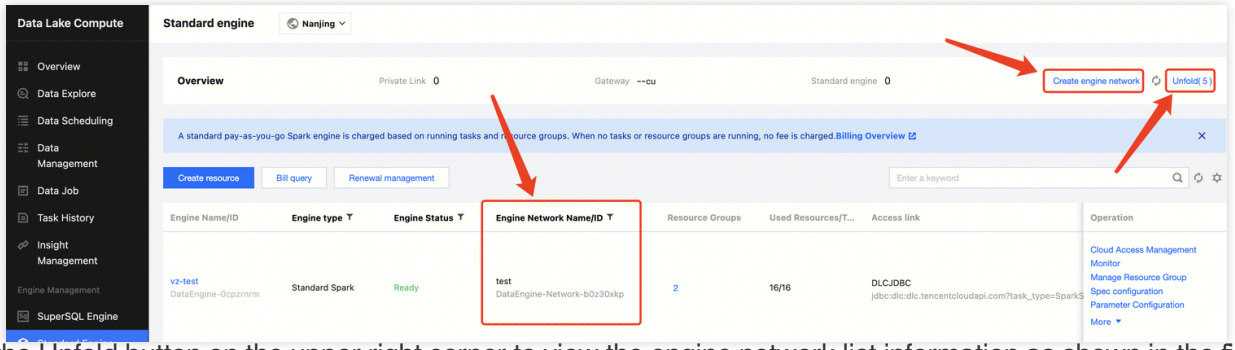
Creating an Engine Network and Gateway

When the allowlist is not enabled, users have one engine network by default and cannot create another engine network, as shown in the figure below. Users do not need to manually create gateways. When users create the first engine or submit the first task under that engine network, DLC will create a free gateway with specifications of 2 CUs by default under that engine network.



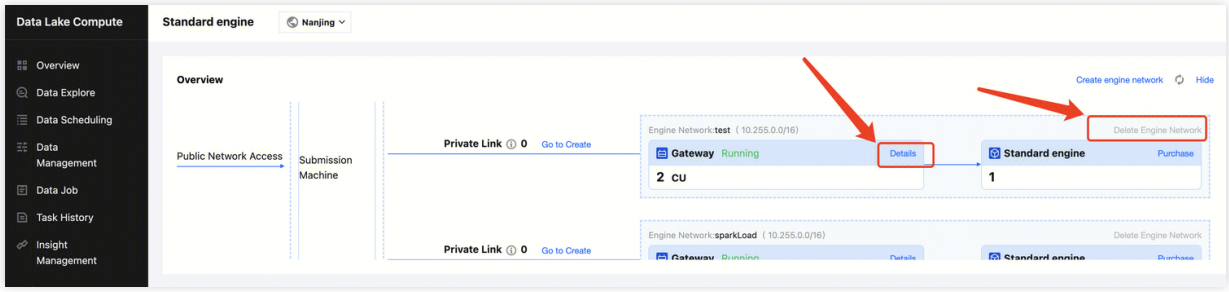
After the allowlist is enabled, users can create multiple engine networks, as shown below. Users can create an engine network by clicking **Create engine network**. The created engine network does not have a gateway initially. Similarly, when users create the first engine or submit the first task under that engine network, DLC will create a free gateway with the specifications of 2 CUs by default under that engine network.

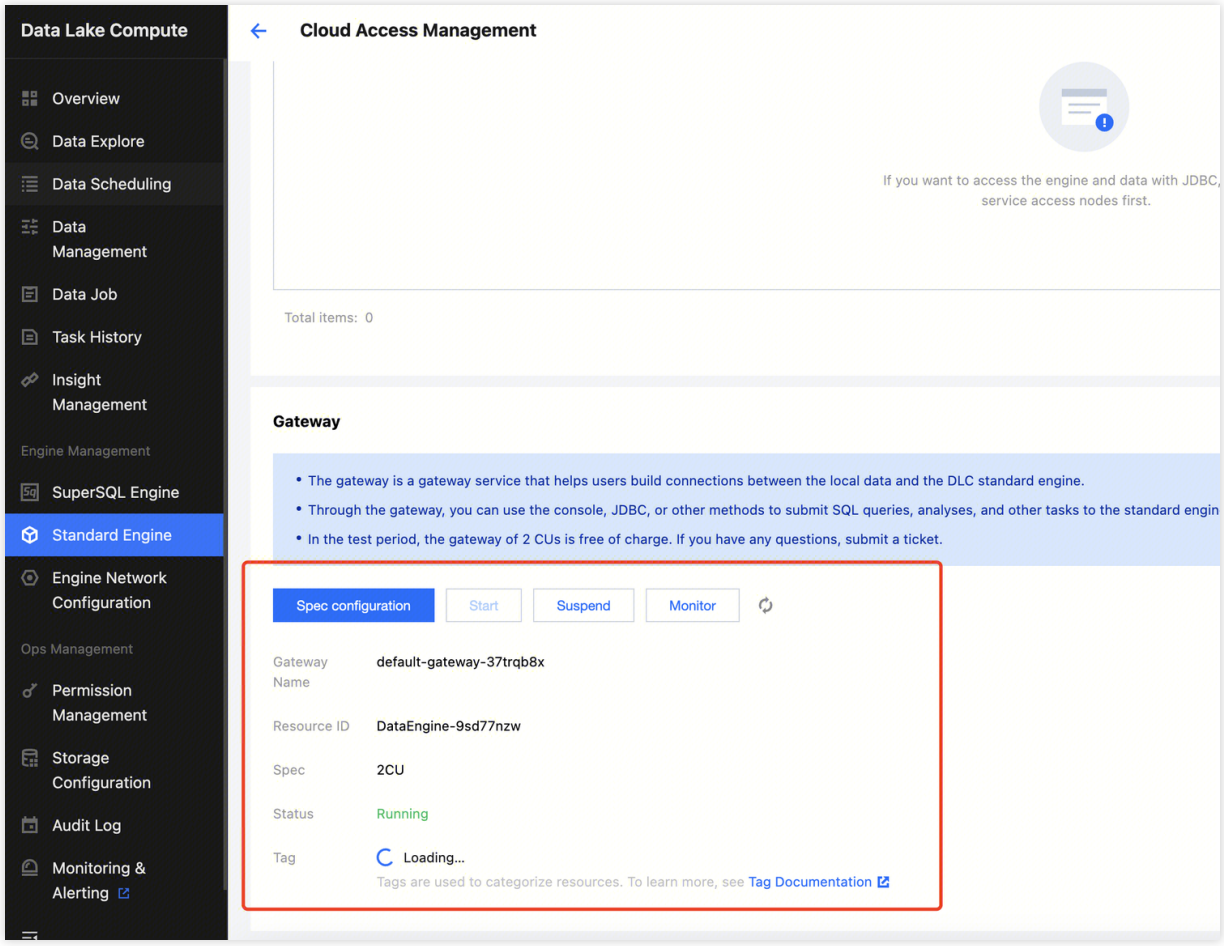
Users can see which engine network the current engine belongs to through the Engine Network Name/ID column on the engine list page.



Click the Unfold button on the upper right corner to view the engine network list information as shown in the figure below. Click the Details button to view the detailed information of the current engine network, including the number of standard engines under the current engine network, the number of user VPCs connected with the engine network, and the specifications of the gateway.

To avoid wrong cancellation, the system does not allow users to directly delete the engine network. Only when the number of standard engines under the current engine network is 0 can users click Delete Engine Network to delete the engine network.





Gateway Specifications

The DLC will automatically create a gateway with the specifications of 2 CUs for each engine network, and this gateway will not incur any fees. However, the gateway of 2 CUs is only suitable for the testing environment. It is recommended that users scale out the gateway for the production environment.

The DLC offers various gateway specifications. It is recommended to select the gateway specifications based on the number of engines to be managed, the maximum query concurrency QPS of the business scenario, and others. See the following table for details.

Gateway Specifications	Whether the Gateway Supports HA	Number of Managed Spark Resource Groups	Number of Managed Presto Engines	Number of Spark SQL/Presto SQL Concurrent Queries	Number of Concurrent Spark MLlib Notebook Sessions Created Transiently/Max Recommended	Number of Concurrent Spark Batch Tasks Submitted Transiently/Numb of Spark Batch Tasks Running Simultaneously
2 CU	No	50	4	100	10/20	30/50

16 CU	Yes	150	12	200	20/80	80/150
32 CU	Yes	400	35	600	100/200	220/400
64 CU	Yes	700	70	1000	200/300	400/600

Upgrading Specifications

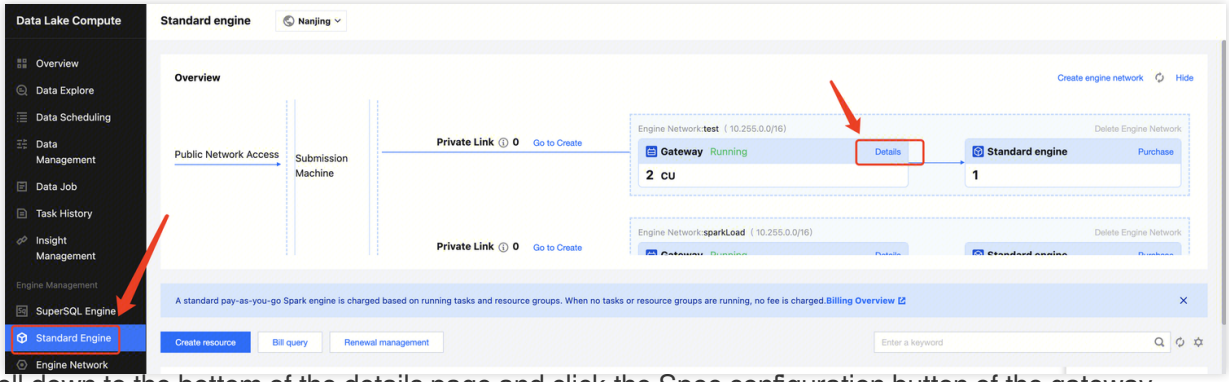
Data Lake Compute (DLC) provides 2 CU specifications for users by default. When the business scenario cannot be met and it is necessary to upgrade the specifications, purchase is required to obtain them.

Note：

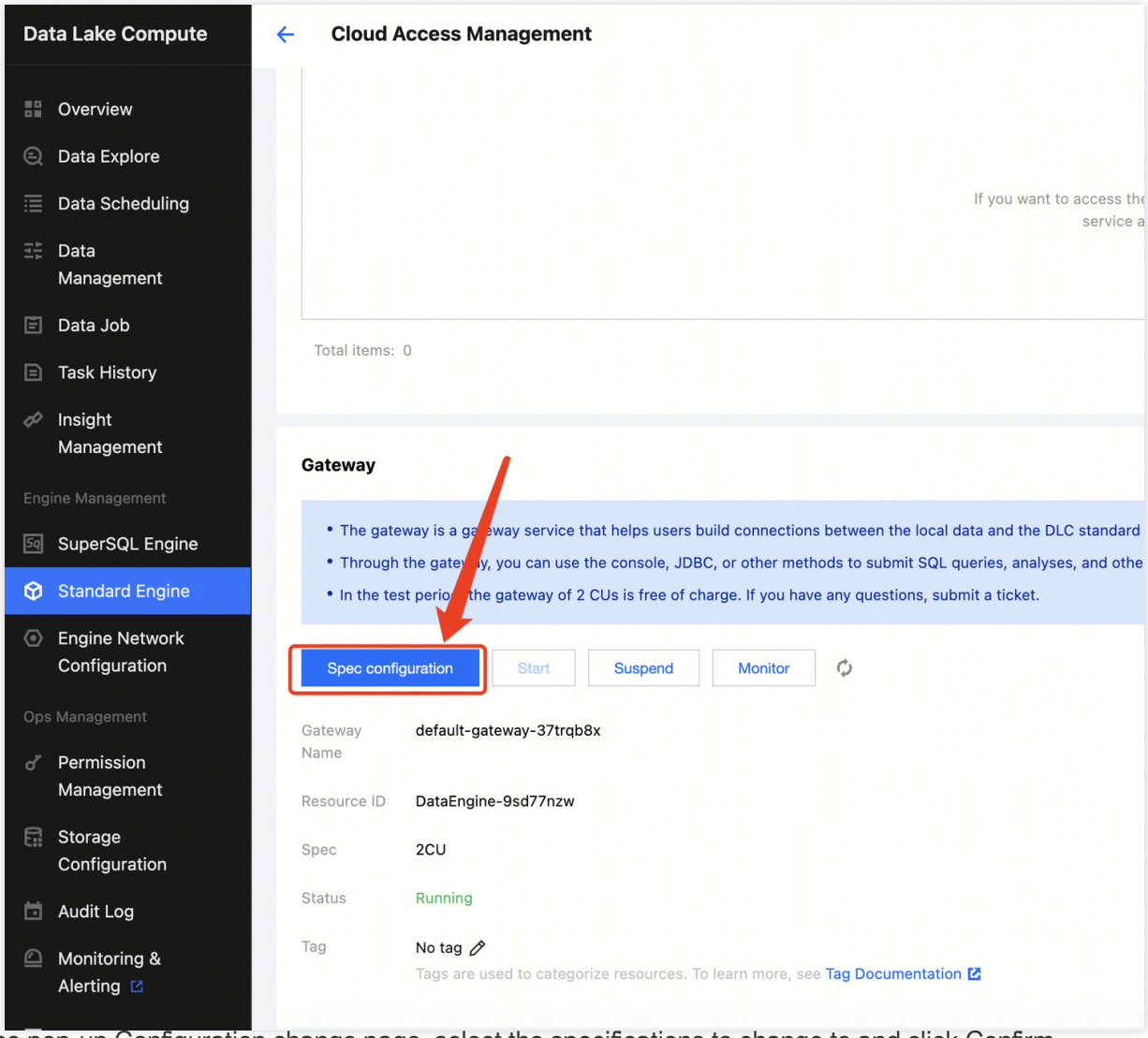
1. Gateway configuration adjustment will lead to interruption and failure of all currently running tasks. Proceed with caution.
2. The entire change process is expected to take 10 to 15 minutes. If the gateway status does not return to running for a long time, submit a ticket for resolution.

If users need to upgrade the configuration of the gateway, they can follow the steps below.

1. Click on the left side of the sidebar. Standard engine to enter the engine list page.
2. Click Standard Engine on the left to enter the engine list page. At the top of the page, find the to-be-operated engine network and click **Gateway> Details** to enter the engine network details page.



3. Scroll down to the bottom of the details page and click the Spec configuration button of the gateway.

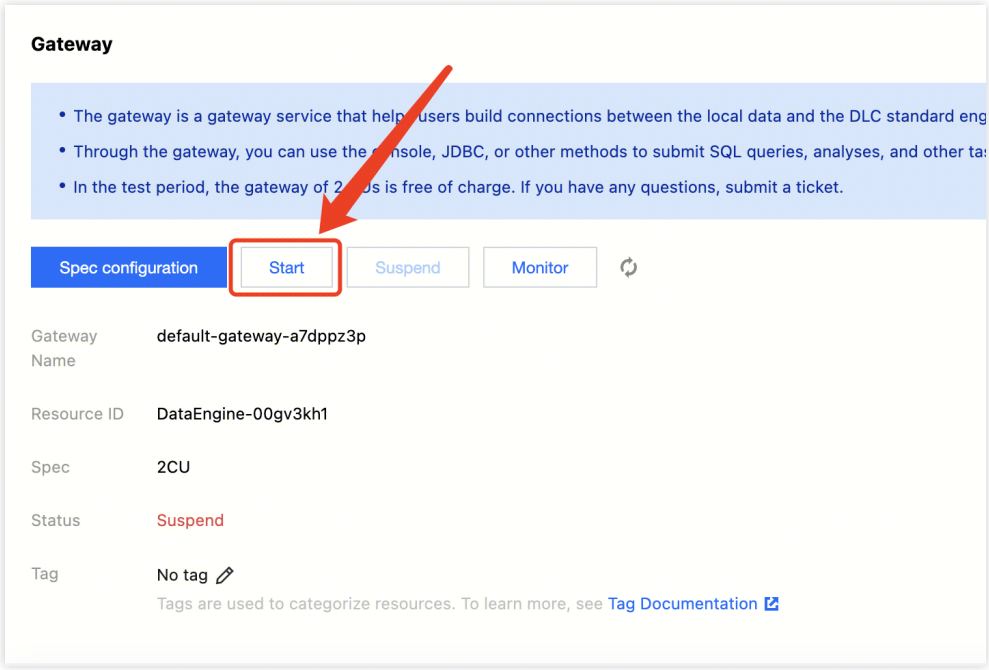


4. In the pop-up Configuration change page, select the specifications to change to and click Confirm.

FAQs

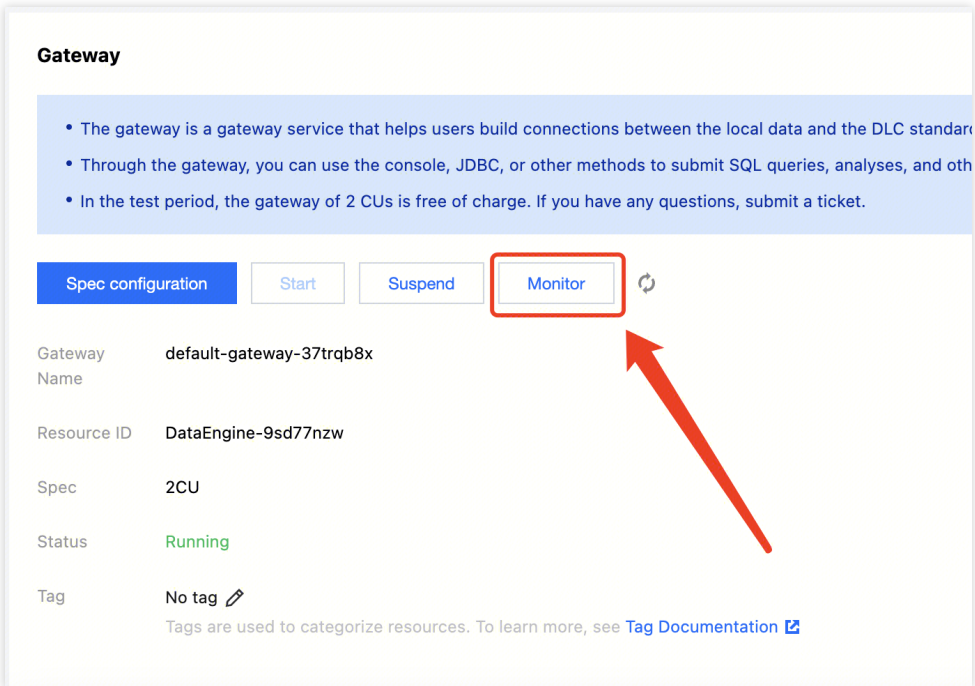
How to solve the API timeout error when tasks are submitted via JDBC?

First, check the gateway status through the console to see if it is normal and running. If the status of the gateway is Suspend, you can click the Start button to start the gateway and try again. Enter the engine network details page, go to the gateway details at the bottom, and click theStart button.

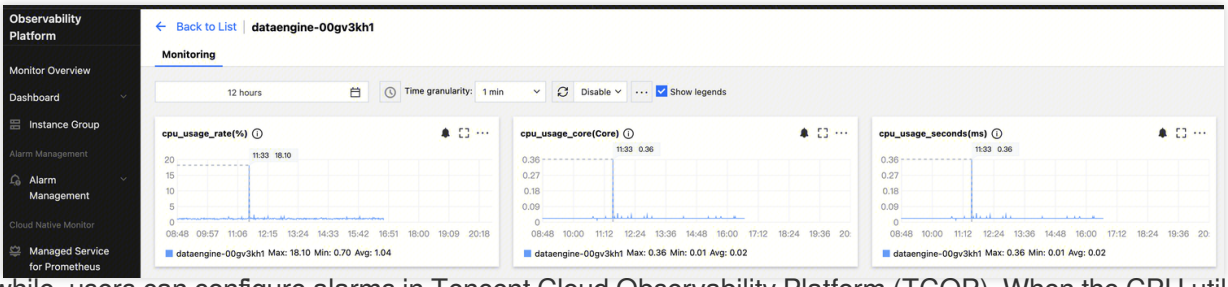


How to determine whether the current gateway load is normal?

The DLC provides basic monitoring of the gateway, and the health status of the gateway can be judged through the monitoring information. Enter the engine network details page, go to the gateway details at the bottom, and click the **Monitor** button to enter the gateway monitoring page.



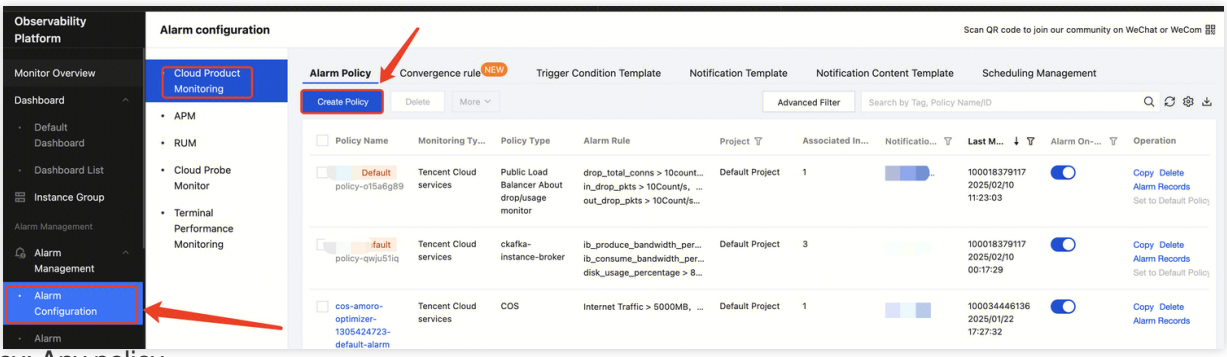
As shown in the figure below, you can see the monitoring information of the gateway's CPU, memory, task threads and other aspects. If the CPU or memory load exceeds 70%, you need to consider whether the gateway load is high and scale out for the gateway.



Meanwhile, users can configure alarms in Tencent Cloud Observability Platform (TCOP). When the CPU utilization and the memory usage of the gateway exceed certain limits, the alarms can reach customers in the first place, enabling them to carry out operations such as scale-out of the gateway in advance.

The configuration process is as follows:

1. Enter the TCOP console, select Alarm Configuration, and click Create Policy.



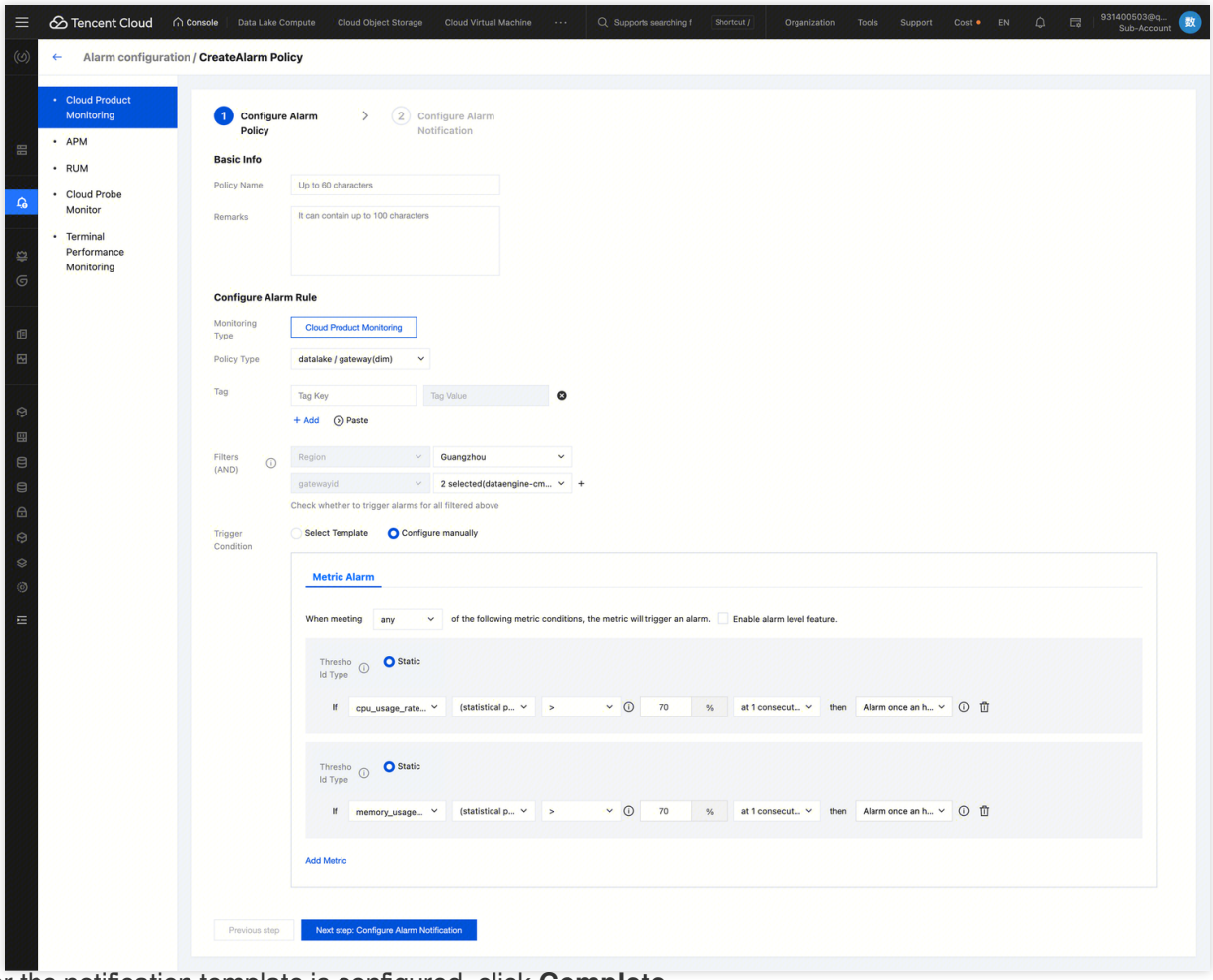
2. Policy: Any policy

Policy Type: datalake/gateway (dim)

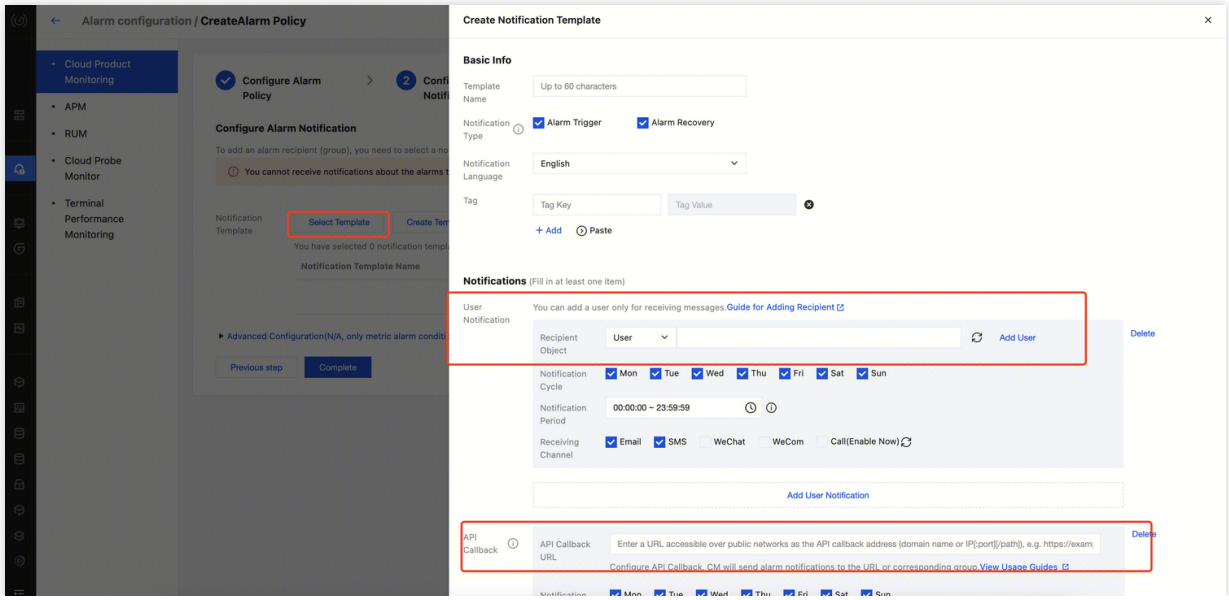
Filters (AND): Select the region where the gateway resides and select the gateway that requires alarm enabled. Multiple filters are allowed.

Trigger Condition: Manually configure the trigger conditions. As shown in the figure below, it It is configured that if either the CPU load or the memory usage exceeds 70%, an alarm will be triggered. Users can configure other alarm trigger conditions according to their needs.

3. Click **Next step:Configure Alarm Notification**.~~As shown in the figure below, if~~ If there is an alarm notification template, you can reuse the existing template. If there is not, you can create a template and select the users to be notified after the alarm is triggered or select the WeChat group that the alarms are to be distributed to.



4. After the notification template is configured, click **Complete**.



Standard Engine Startup and Stop Logs

Last updated：2025-03-21 12:29:26

The log feature of Standard Engine Startup and Stop Logs records the startup and suspension events of each engine, making it easy to monitor engine status, troubleshoot, and optimize resource management.

Operation Steps

1. Log in to Data Lake Compute (DLC) Console > Resource Management > Standard Engine, choose service region.
2. Startup and stop logs of different operation objects:
Gateway: Unfold the overview, click **Details**, and view the startup and stop logs of the gateway on the details page.
Presto engine: Select the engine instance you want to view in the engine list, click **engine name**, and enter the basic configuration page to view the startup and stop logs of the computing engine.
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, and click **resource group name** to enter the resource group details page to view the startup and stop logs of the resource group.

Startup and Stop Log List

Note:
Support for Spark engine resource group startup and shutdown logs requires a gateway restart operation after March 20, 2025. Specific operation steps: Click on **Engine Network > Gateway > Details** on the overview card to enter the engine network details page, click **Suspend**, and then click **Start**.

Field Name	Description
TraceId	TraceId is a unique identifier for a start-stop process. It can associate the logs of different actions within the same process, helping users identify which logs belong to the same operation or request.
Time	Starting an action corresponds to the operation start time, and completing an action corresponds to the operation completion time.
Action	The actions include CLUSTER_SCALE_IN、CLUSTER_SUSPEND、CLUSTER_SCALE_UP, etc.
Details	CU adjustment of objects before and after operation.

Resource Group
Resource Group Introduction

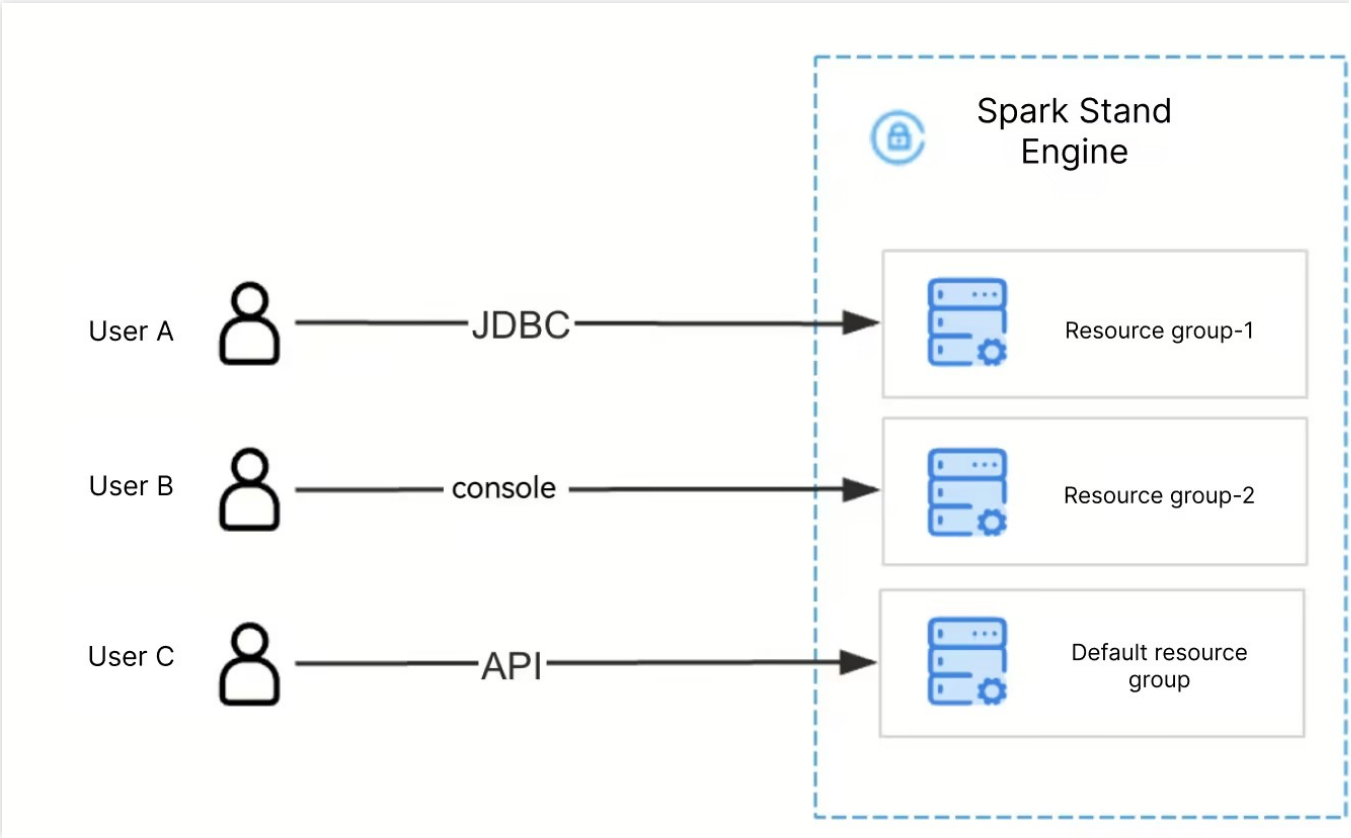
Last updated：2025-01-23 17:05:12

The resource group is a secondary queue division of the computing resources within a Standard Spark Engine. Resource groups belong to a parent Standard Engine, and resource groups under the same engine share resources with each other. The computing units (CUs) of the DLC Standard Spark Engine can be allocated to multiple resource groups as needed. You can configure each resource group's minimum and maximum CU limits, start and stop policies, concurrency, and dynamic/static parameters to efficiently manage resource isolation and workload in complex scenes such as multi-tenancy and multi-tasking.
For example, you can create separate resource groups within a Standard Spark Engine, such as a Report Resource Group, a Data Warehouse Resource Group, and a Historical Backfill Resource Group. You can set the upper and lower limits of computing units (CUs) for each resource group and assign relevant SQL tasks or jobs, such as reports and data warehouse tasks, to the appropriate resource group, ensuring resource isolation between different types of tasks and preventing individual large queries from monopolizing resources for extended periods.

Features

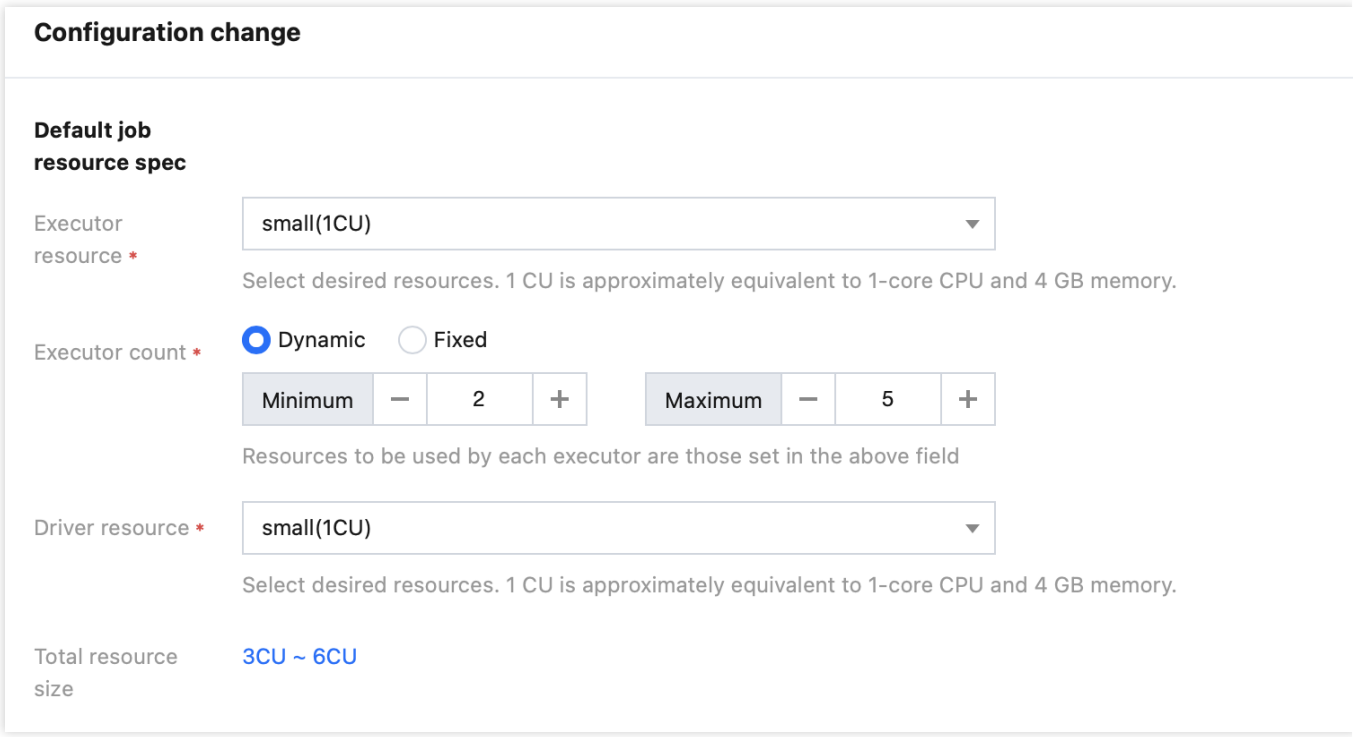
Resource Group Isolation

Resource groups enable resource isolation within the Standard Spark Engine. You can assign specific resource groups to different users or queries, effectively isolating resources and preventing a single user or large query from monopolizing most of the computing engine's resources.

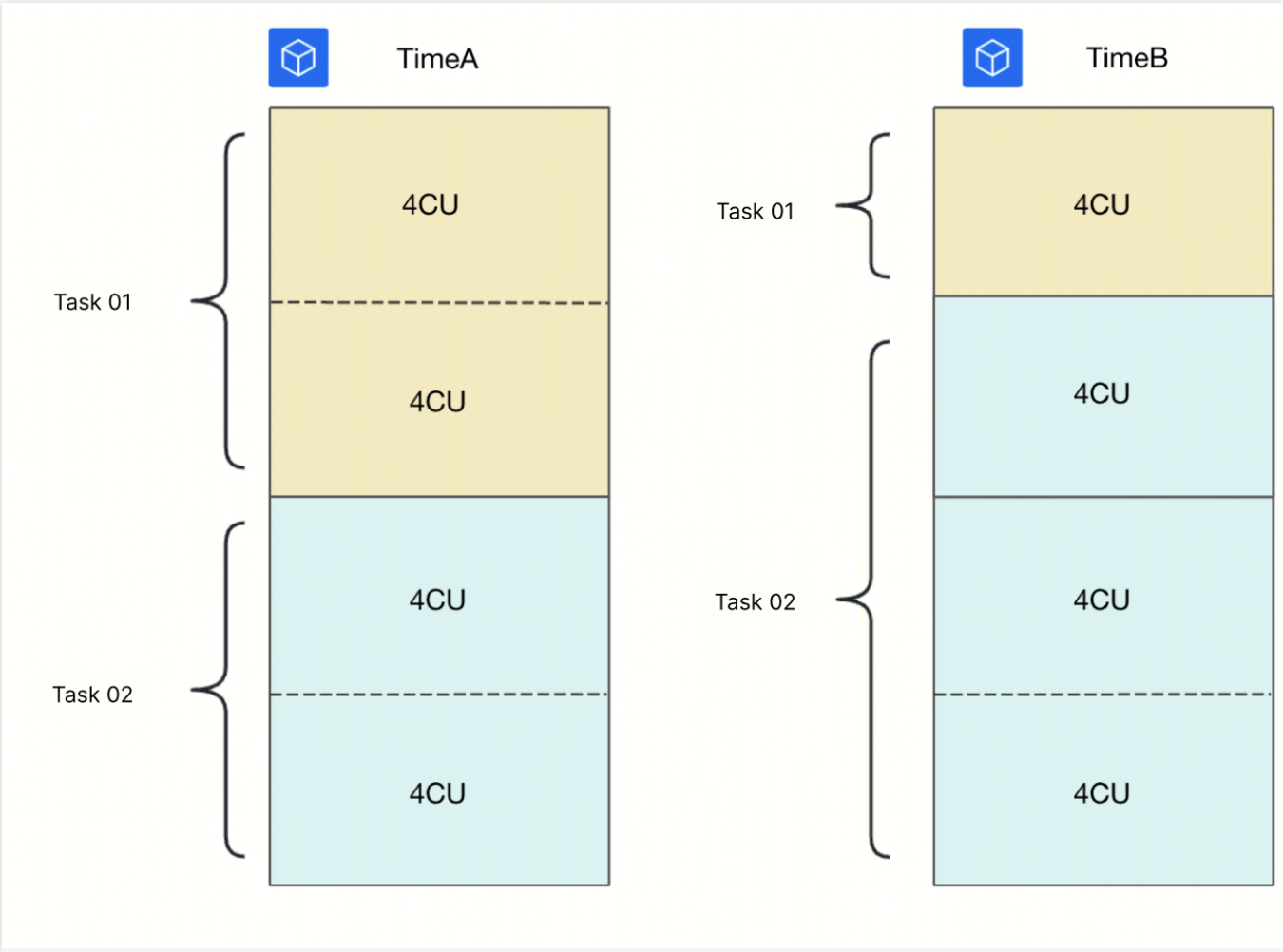


Resource Group Elasticity

By configuring the number of Executors in a resource group for dynamic allocation, the resource group can adjust the resources used by SQL tasks or jobs based on the workload, effectively improving resource utilization.
The dynamic allocation configuration is shown in the diagram below:



Both Task 01 and Task 02 are set to dynamic allocation, each using 8 CUs at Time A. By Time B, Task 01 only requires 4 CUs, releasing 4 CUs of idle resources for Task 02 to use, thereby improving overall resource utilization. This process is illustrated in the diagram below:



Usage Limitations

The resource group name should be globally unique. It is recommended to use an all-English name.

Terminology

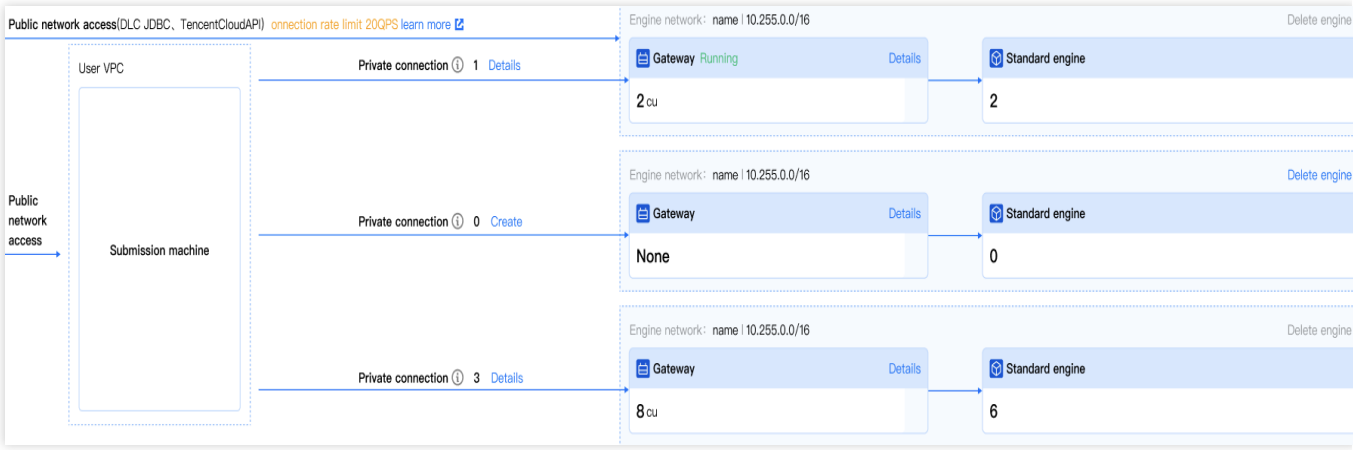
Description	Illustration	Default Resource Groups
(System created by default) Exist upon engine creation, and	The default resource group starts in a suspended status, with settings for automatic start and automatic suspension. The default resource group supports modification of resource configurations.	default-rg-i0wg28z6... rg-i0wg28z6p4

named as default-rg-xxx.	The default resource group supports configuring start/stop policies, setting concurrency limits, and adjusting dynamic/static parameters. The default resource group cannot be deleted.	
(User manually created) The custom resource group supports the modification of resource configurations.	The custom resource group supports configuring start/stop policies, setting concurrency limits, and adjusting dynamic/static parameters. The custom resource group can be deleted.	-

Private Connection
Private Connection Introduction

Last updated：2024-09-04 11:15:28

Endpoints are built on Private Link. If you need to access engines and data through JDBC or other methods, you can create an endpoint to establish a secure and stable private connection between your VPC and the access point.



Usage Limitations

1. A maximum of 4 endpoints can be created.
2. For private connection billing, see Private Link Billing.

Network Connection Configuration

Last updated：2025-04-09 20:49:29

Data Lake Compute (DLC) supports configuring network (VPC) for data engine, facilitating management of engine access to different data source networks.

Network Configuration Type

According to different business scenarios, DLC provides two network configuration types.
Enhanced network configuration: suitable for accessing the data under one VPC with high speed and stability.
Note：
1. A data engine of a non-Spark job type can only be bound to one enhanced network configuration.
2. If you use an enhanced network, the subnet IP address under your VPC will be used. Please ensure sufficient subnet IP addresses.
Cross-origin network configuration: suitable for cross-origin federated data query that needs to access multiple VPCs. A data engine can support binding multiple cross-origin network configurations.

Network Configuration Status

Initializing: The network configuration is being initialized. At this point, the network is not active.
Success: The network configuration takes effect on the bound engine.
Failure: The network configuration fails and can be deleted and reconfigured.

Network Configuration Security Policy

If you have configured a security group policy for the VPC, you need to add inbound rules for different network configuration types.
Enhanced network: Add inbound rules for the IP range of the VPC where the data source is located to the security group.
Cross-source network: Add inbound rules for the IP range of the engine bound to the network configuration to the security group.

Create Network Configuration

1. Log in to the DLC console and choose service region.
2. Enter **Resource Management > Network Connection Configuration** through the left sidebar.
3. Click the **Create Network Configuration** button to enter the Create Configuration page.

The configuration parameters are as follows:

Configuration Content	Required or Not	Filling Instructions
Network Configuration Type	Yes	Select according to the use case Enhanced network configuration: suitable for data scenarios that require high-speed and stable access to a VPC. Cross-origin network configuration: suitable for cross-origin federated query analysis scenarios that need to access data under multiple VPCs.
Configuration Name	Yes	Supports Chinese, English, and _, with a number of characters not more than 35.
Instance source	Yes	Two sources are supported: Data catalog of DLC: Option the data catalog that has created a connection in the data management of DLC currently New network configuration: Select a new data source to create a network connection. Currently, the data source supports MySQL, Kafka, EMR HDFS (COS, HDFS, Chdfs), Postgresql, SqlServer, Clickhouse. If the data source associated with the network configuration to be created is not yet supported, you can select another option and manually specify a VPC.
Catalog	Yes	Select the corresponding data catalog according to the source of the selected instance. The range of selectable data catalogs will be related to your account permission.
Data source VPC	No	The data engine network will connect all subnets in the VPC.
Bound data engine	Yes	Select the data engine associated with this network configuration. If the data engine is in isolated or initializing status, it will be unable to select.
Configuration Description	No	Not more than 100 characters.

4. Fill in, complete the settings and save. Then you can create a network configuration.

Note：

Once created, the network is in the initialization state. Subsequently, you can view the status in the list.

Delete Network Configuration

You can perform a deletion operation to manage the deletion of network configurations that are no longer needed or have failed to configure. Directions:

1. DLC console, choose service region.
2. Enter **Engine Management > Engine Network Configuration** through the left sidebar.
3. Find the network configuration that needs to be deleted. Support filtering search. Note the selection of network configuration type.
4. Click the **Delete** button. Just complete the deletion after secondary confirmation.

Note：

After deletion, this data engine will not be able to use this network configuration. If you need access, reconfiguration is required. Proceed with caution.

Modify Description Information

You can modify the description information of the configured network configuration by modifying the description information. Directions:

1. DLC console, choose service region.
2. Enter **Engine Management > Engine Network Configuration** through the left sidebar.
3. Find the network configuration that needs to be deleted. Support filtering search. Note the selection of network configuration type.
4. Click the **Modify Description Information** button to edit.

Storage Configuration
Managed Storage Configuration

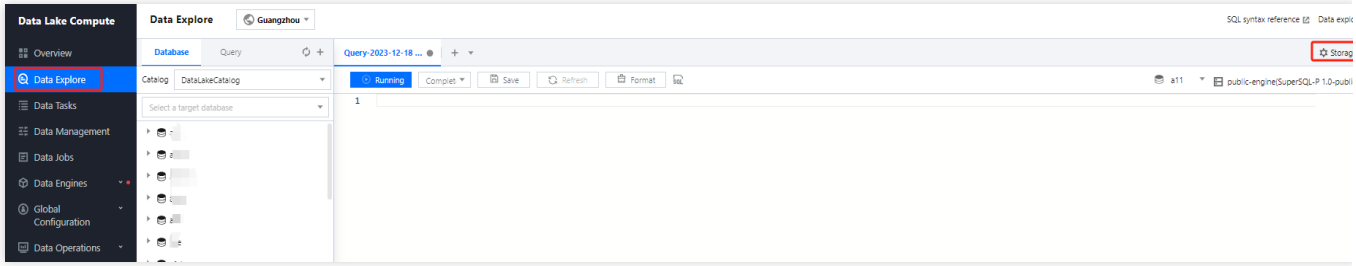
Last updated：2024-07-31 17:30:11

Managed storage refers to the storage space hosted on the Data Lake product, with COS as the underlying storage. Managed storage contains data such as native tables, user program packages, and query results. Therefore, to utilize the capabilities of native tables and data optimization, it is necessary to enable managed storage first. The native tables on managed storage are by default in the Iceberg format, so you don't need to manage the underlying file contents. For details on managed storage billing, please refer to Billing Overview.
This document introduces how to enable and configure managed storage.

Enable Managed Storage

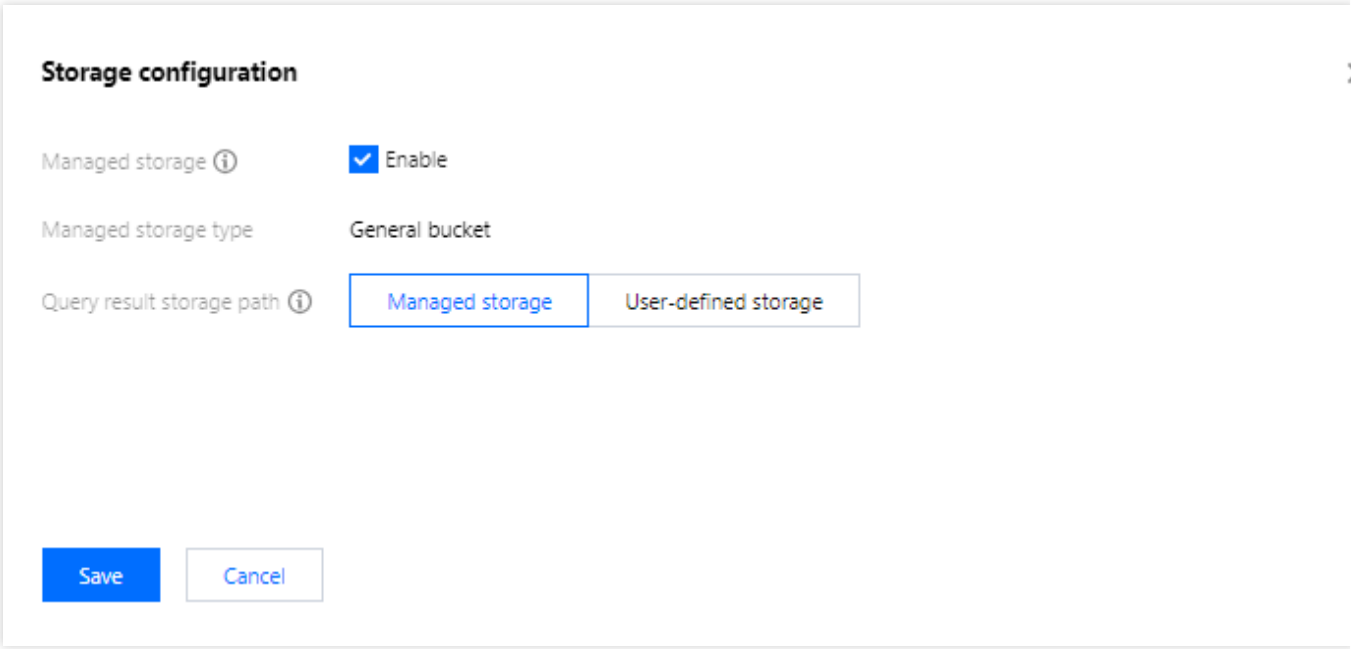
Step 1: Enter Managed Storage Configuration

You can enter the managed storage configuration in the Data Exploration module or the Global Configuration > Storage Configuration module.



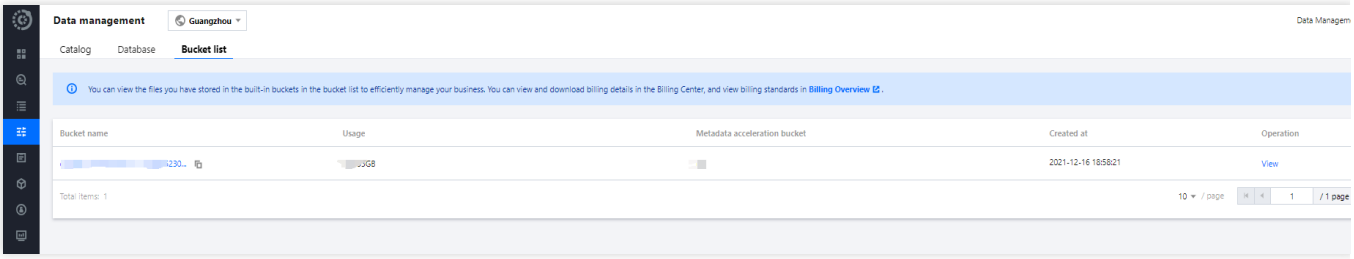
Step 2: Open Managed Storage

1. Check to enable managed storage and save.
Here, you can specify the managed storage type as either a Metadata Acceleration Bucket or an Ordinary Bucket. The billing for both is consistent, but it is necessary to separately configure engine access permissions for the Metadata Acceleration Bucket. For details, please refer to Binding of Metadata Acceleration Bucket.
2. The query result path is used to temporarily store SQL query results, Spark Job Shuffle data, etc. You need to specify a path to ensure the normal operation of jobs and tasks. If you have enabled managed storage, it is recommended to configure the query result path as **Managed Storage**. You can also configure the query result path to your own account's COS bucket path.



View managed bucket

After enabling managed storage, a bucket will be created, and you can view the buckets and data on managed storage in the Data Management module.



Destroy Managed Storage

Destroying data is a high-risk action; only after all database table data has been deleted, can you proceed to destroy managed storage. Destroying managed storage requires administrator privileges.

Step one: Delete database table data

To destroy managed storage, you must first delete all database table data on the managed storage.

You can refer to the Data Catalog and DMC and Data Table Management documents to delete the database table data, or you can run the DROP Syntax in the Data Exploration module to delete the database table data.

Step two: Destroy Managed Storage

After deleting the database table data, you can destroy managed storage on the managed storage configuration tab under the Storage Configuration module.
Destroying managed storage will delete all DLC managed buckets, so please proceed with caution.

Binding a Metadata Acceleration Bucket

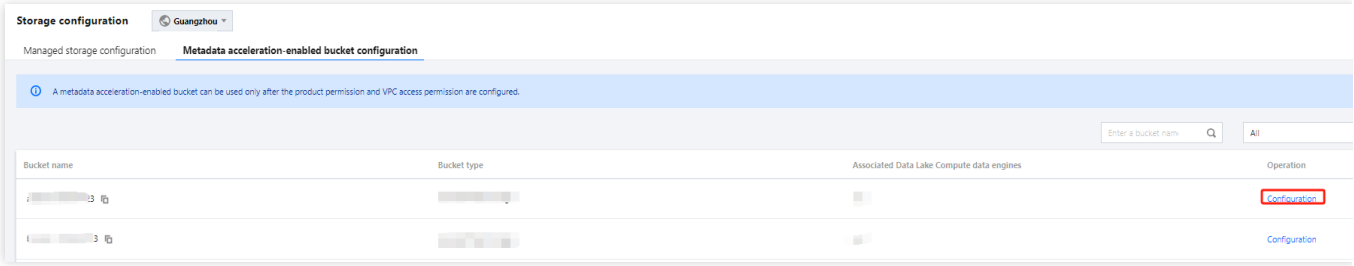
Last updated：2024-07-31 17:30:27

DLC supports the binding of Fusion Bucket to accelerate Query Analysis Performance. To use this feature, you need to create a Metadata Acceleration Bucket. DLC Managed Storage provides Metadata Acceleration Bucket. Use COS Bucket under the user's account. For details, please see COS>Metadata Acceleration.
When accessing the DLC Metadata Acceleration Bucket, binding of permissions is necessary. The Permission Binding Process is as follows.

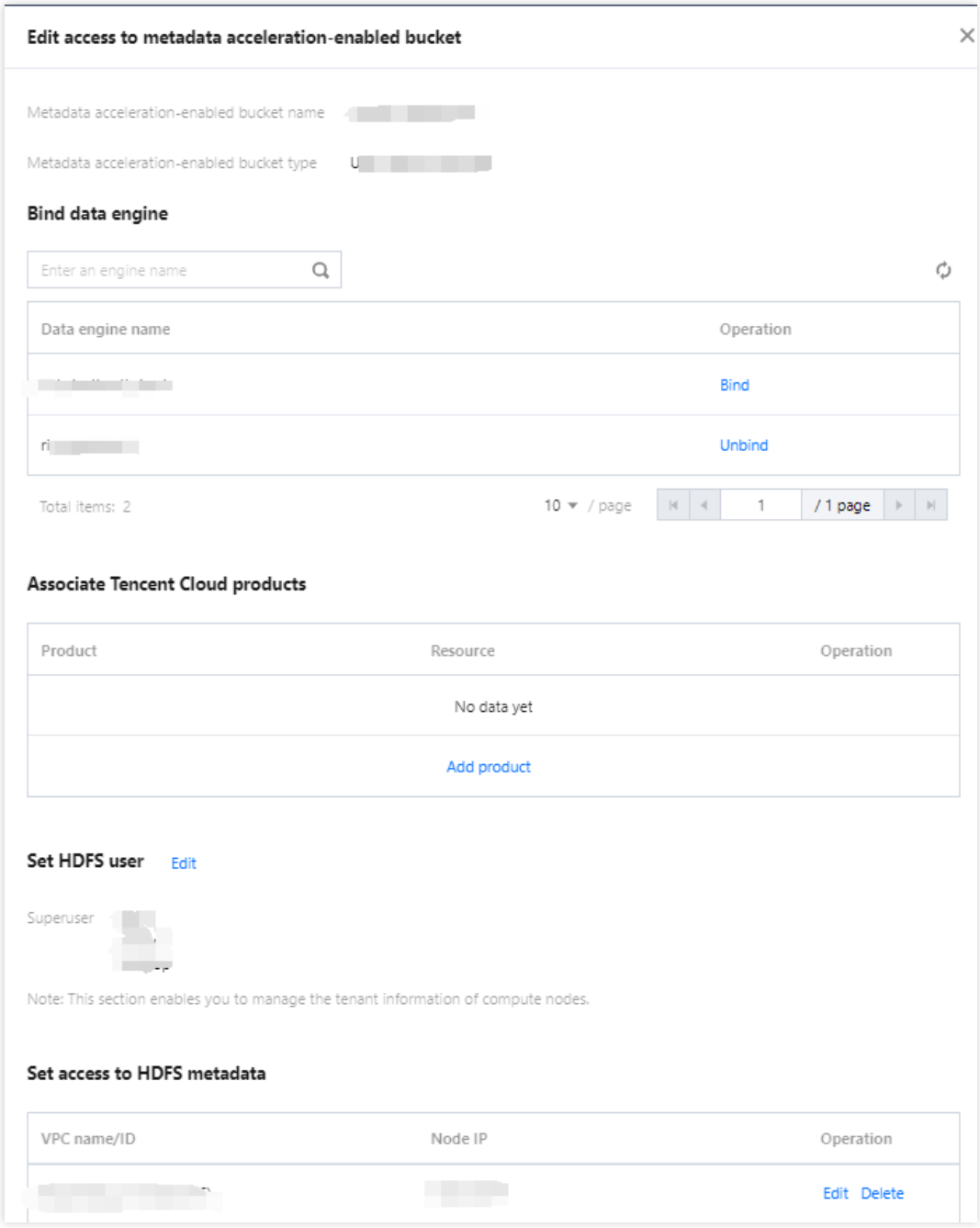
Bind Data Engine and Metadata Acceleration Bucket

1. log in to Data Lake Computing Console, enter Common Management > Storage Configuration.
2. Enter the **Metadata Acceleration Bucket Configuration Page**, select the bucket you want to bind, and click **Configure**.

Note:
Only Metadata Acceleration Buckets are displayed on the Metadata Acceleration Bucket page; ordinary buckets (buckets without the metadata acceleration feature enabled) will not be shown.

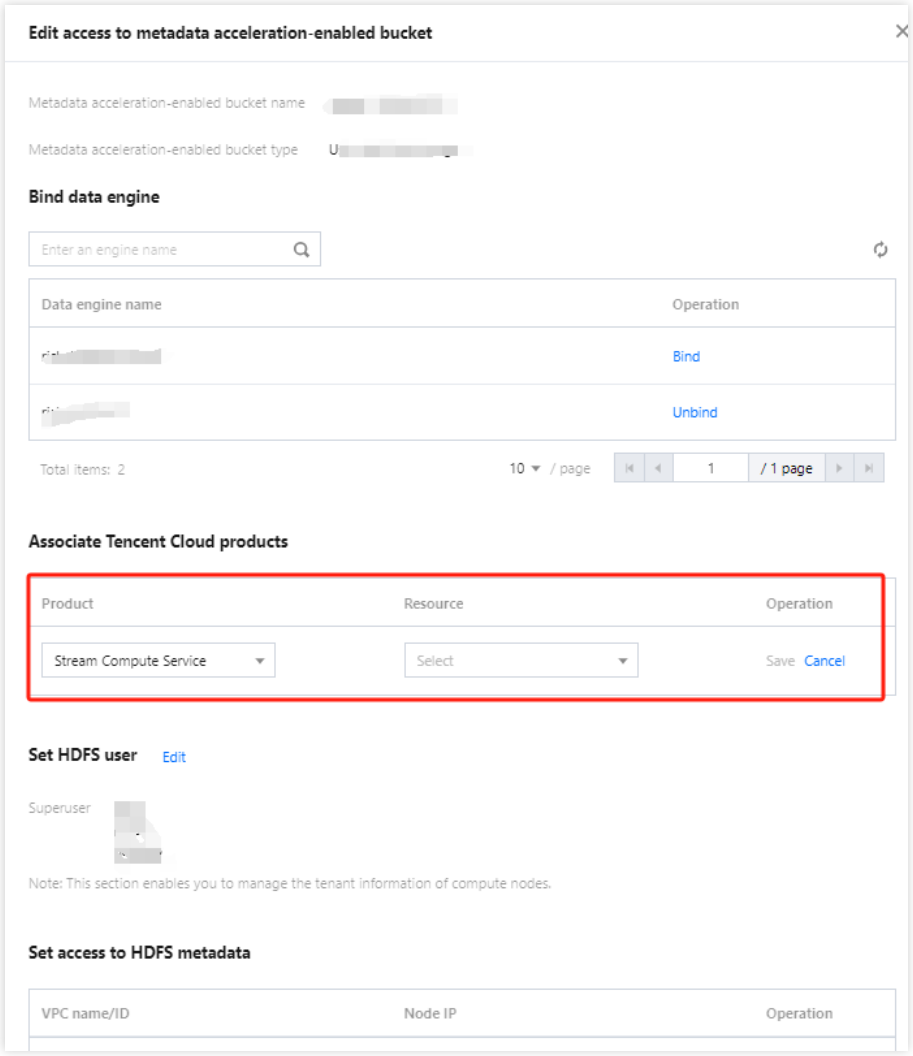


3. Click **Bind** to bind the data engine that needs to access this bucket to the Metadata Acceleration Bucket.



Bind computing resources of SCS

If you use SCS to stream data into the lake, and the storage written to is a Metadata Acceleration Bucket, then you need to configure access permissions for the Metadata Acceleration Bucket under Storage Configuration. Under the Tencent Cloud Product Binding section, create a new product, select Stream Computing Oceanus and the corresponding resources, then click save.

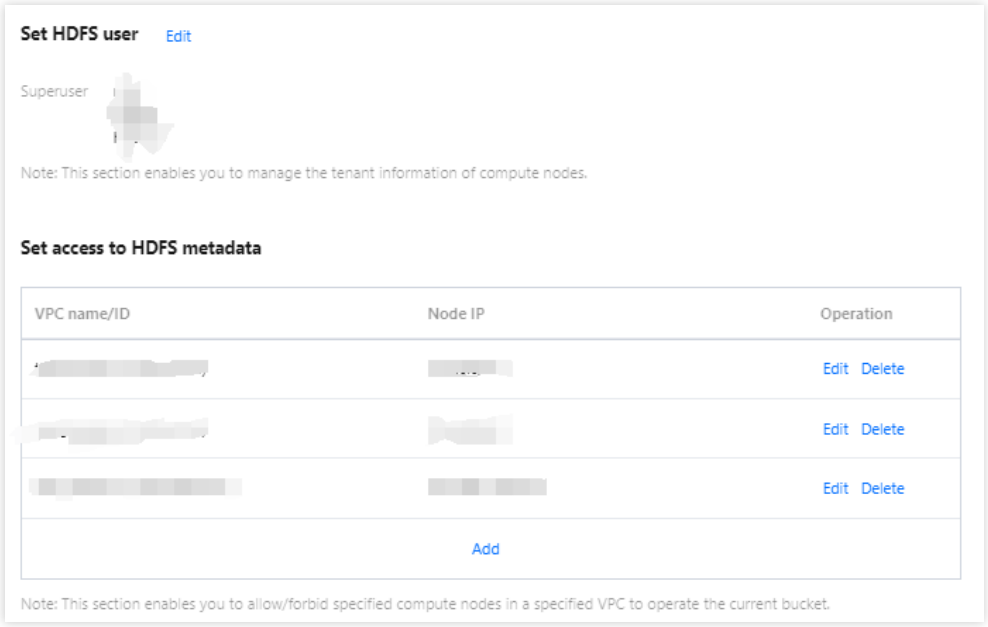


Bind computing resources of non-DLC data engines

Sometimes, the computing resources you need to access the Metadata Acceleration Bucket are not from a DLC data engine. In this case, you can configure access permissions for the Metadata Acceleration Bucket under Storage Configuration.

HDFS User Configuration is used to configure the super user of your computing resources accessing DLC, usually root/hadoop/presto/flink.

HDFS Metadata Permissions Configuration is used to configure the VPC Network Environment you allow to access DLC, usually the VPC where the computing resources of the above mentioned non-DLC data engines are located.


Set HDFS user
Edit
Superuser
Note: This section enables you to manage the tenant information of compute nodes.
Set access to HDFS metadata
VPC name/ID
Node IP
Operation
Edit Delete
Edit Delete
Edit Delete
Add
Note: This section enables you to allow/forbid specified compute nodes in a specified VPC to operate the current bucket.

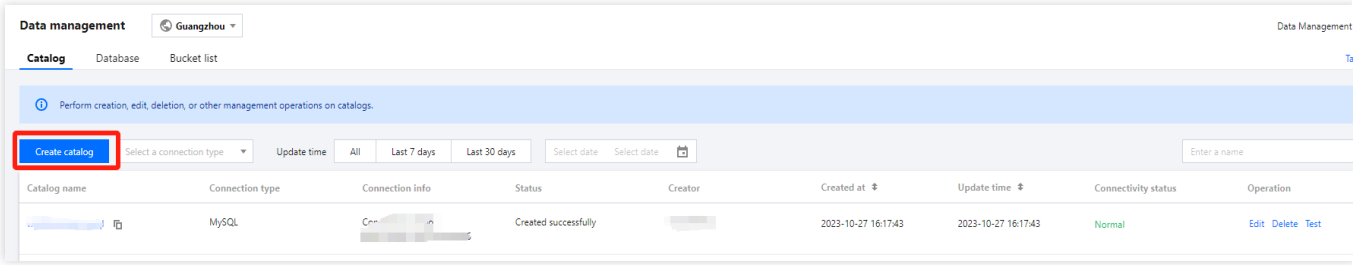
Metadata Management
Data Catalogs and DMC

Last updated：2024-07-31 17:27:26

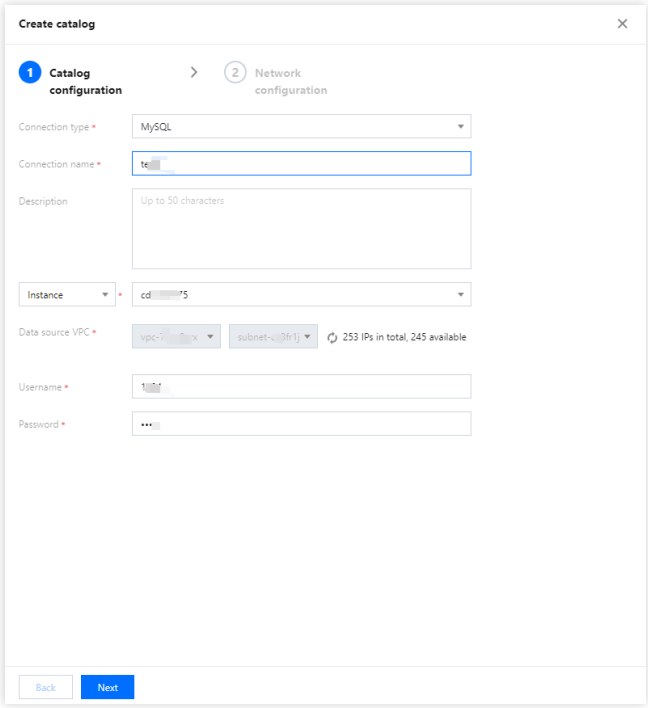
External data and managed storage data in DLC can be managed through the Data Management Page by executing standard SQL statements and APIs. Through the Console Data Management Page, you can create, edit data catalogs, and create, query, delete databases and tables.

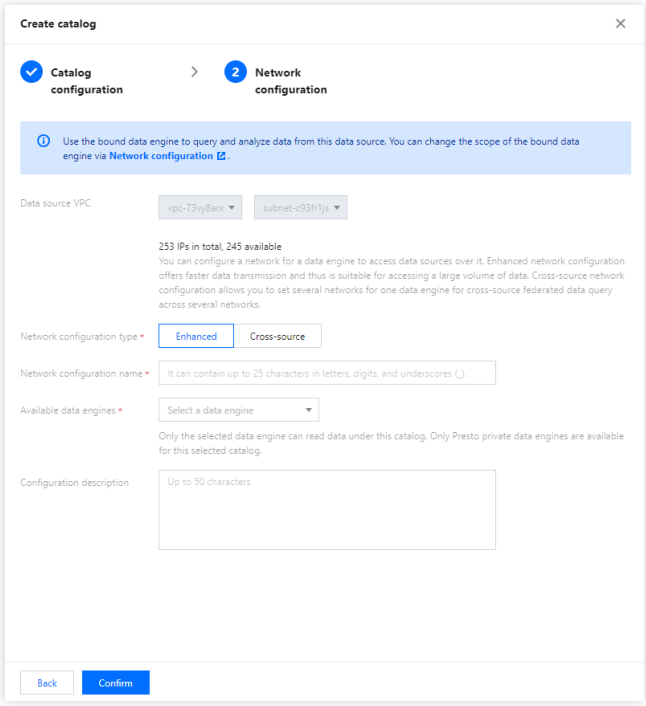
Creating a data catalog

Note:
The platform will automatically create a DataLakeCatalog for you for data management on the lake.
When you have external data sources and wish to perform federated analysis, you can follow the process below to create a data catalog for external data sources.
1. Log in to DLC console, select the service region. The account used to log in must have the permission to create a catalog. For enabling sub-account permissions, refer to Sub-account Permission Management.
2. Enter Data Management, click **Create Catalog**.



3. Enter the data source creation visual interface. After filling in the connection information, complete the network configuration to connect the engine with the external data source.



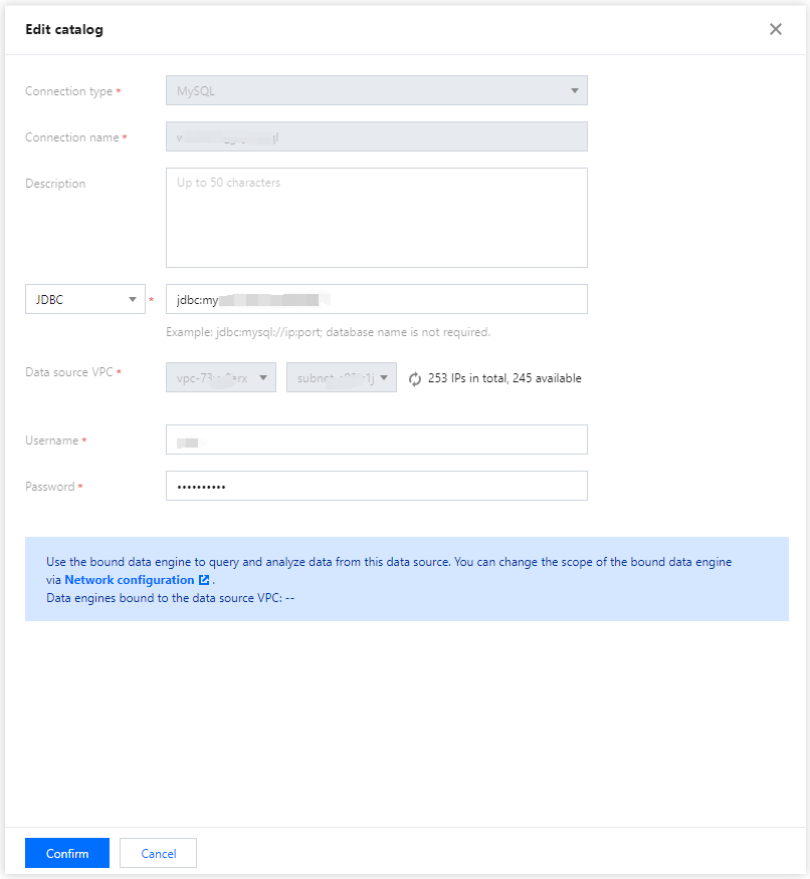


4. After filling in the data source information, click **Confirm** to complete the creation of the data source.

5. In the Data Catalog List, view connection information, status, creator, and other information.

Edit Data Catalog

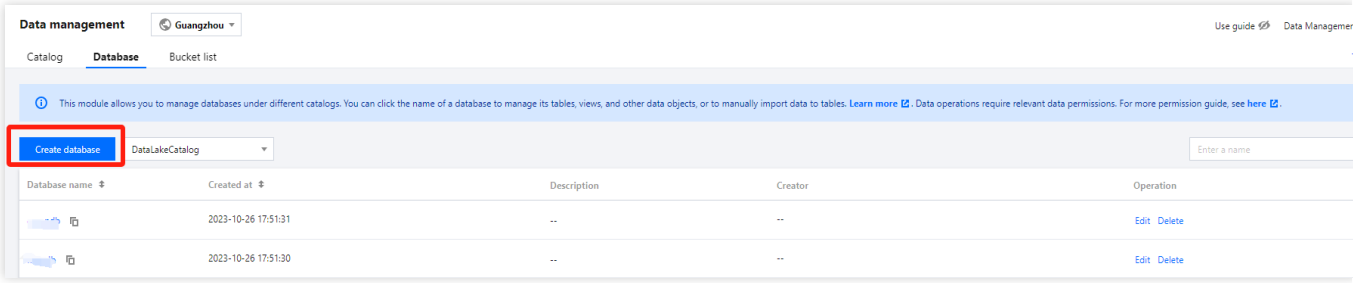
1. Click **Data Catalog List > Operations > Edit** to modify the Data Catalog's description information, network configuration information, username, password, and running cluster, etc.



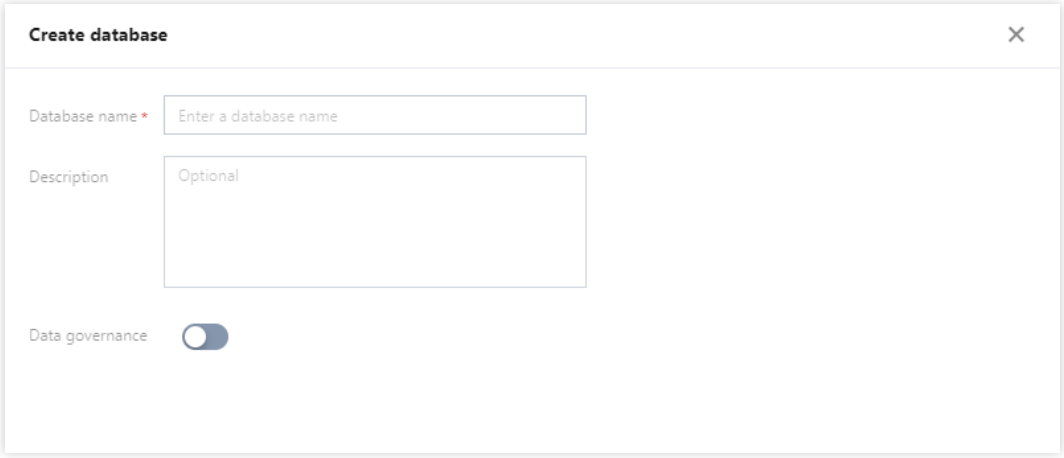
2. After modifications, click **Create** to reconstruct the Data Catalog.

New database

1. Log in to DLC Console, select the service region. The account used to log in must have database creation permissions.
2. Enter Data Management, click on the directory name under the Data Catalog to view the databases within that directory.
3. Click **Create Data Catalog** to enter the Database Creation Visual Interface.



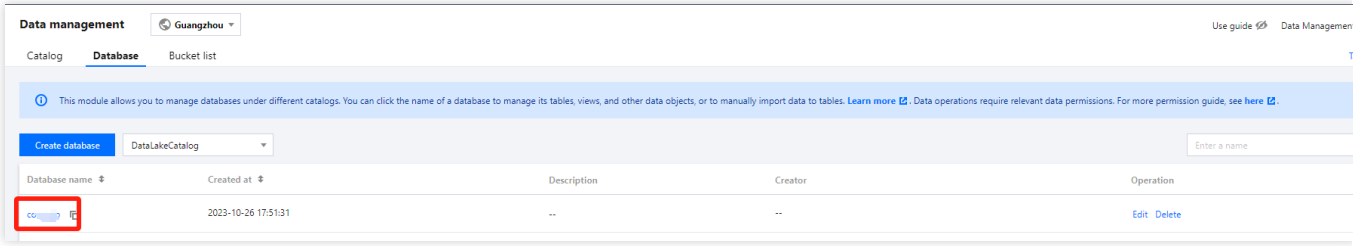
4. After filling in the relevant database information and saving, the database creation is complete. When creating a database, you can enable data optimization for the entire database.



Database Name: Globally unique, supports English case-sensitive letters, numbers, "_", cannot start with a number, up to 128 characters.
Description: Supports both Chinese and English, up to 2,048 characters.
A root account can create up to 100 databases.

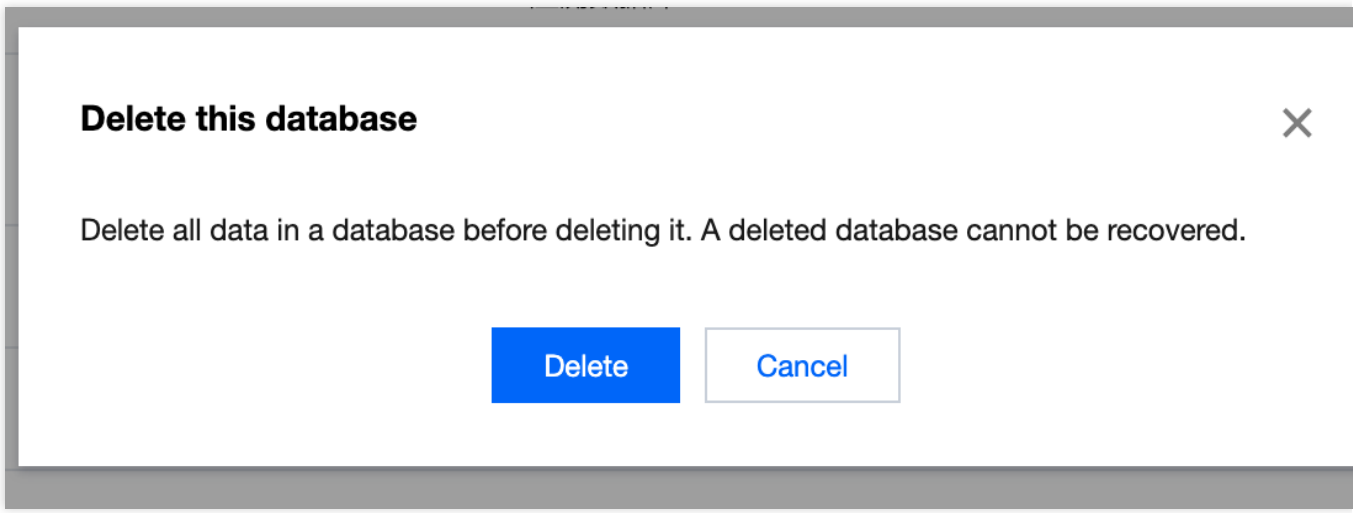
View Database

1. Log in to DLC Console, select the service region. The account used to log in must have database query permissions.
2. Enter Data Management > **Database**, select the data directory, click **Database Name** to access the database details, manage the database's tables. For a detailed operation guide, refer to Data Table Management.



Dropping a Database

1. Log in to DLC Console, select the service region. The account used to log in must have database deletion permissions.
2. Enter Data Management, click **Delete**. After confirming a second time, the database can be deleted.



Data Table Management

Last updated：2024-07-31 17:27:51

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

Creating Table

Approach one: Create in Data Exploration

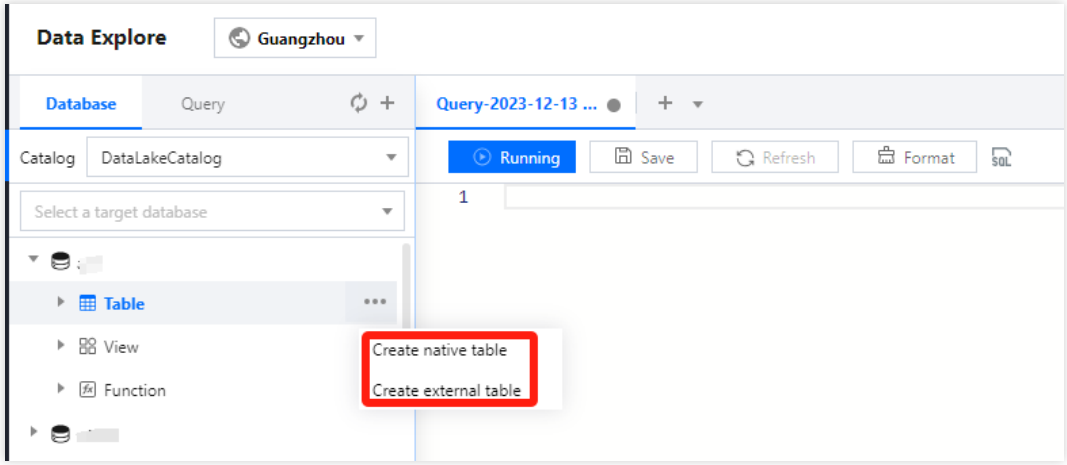
1. Log in to the DLC console, select the service region, log in to users need to have the permission to create tables.
2. Enter the Data Exploration module, in the left list, click on an existing database, hover over the table row, then click the

icon, click **Create Native Table** or **Create External Table.**

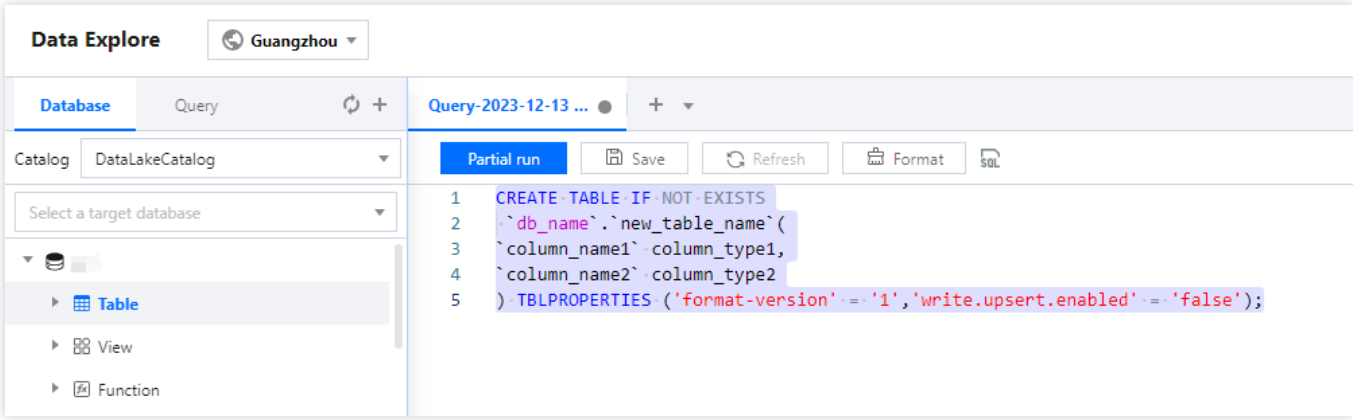
Note:

A native table refers to a table on the DLC managed storage. With a native table, you don't need to worry about the underlying Iceberg storage format, and it has capabilities like data optimization. To use a native table, you need to enable managed storage first, see Managed Storage Configuration for details.

The underlying data of the external table resides on your own COS. Creating an external table requires specifying the data path.



3. After clicking **Create Native Table/Create External Table**, the system will automatically generate an SQL template for creating a data table. Users can modify the SQL template to create a data table. After clicking **Run**, the SQL statement to create the data table is executed, completing the creation.

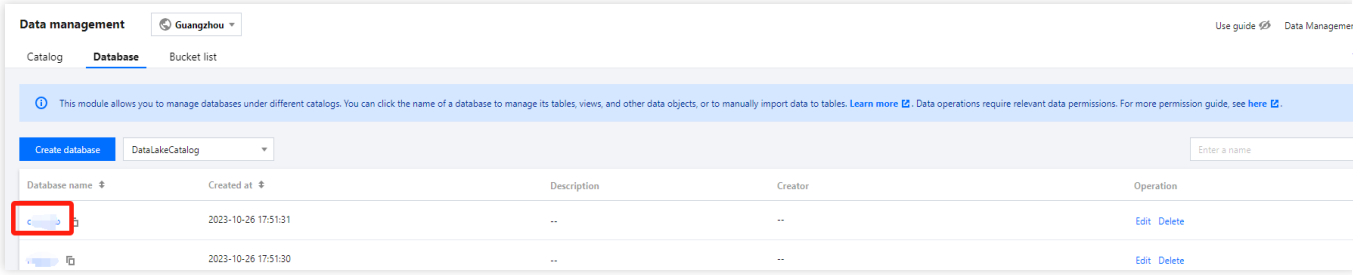


Approach two: Create in Data Management

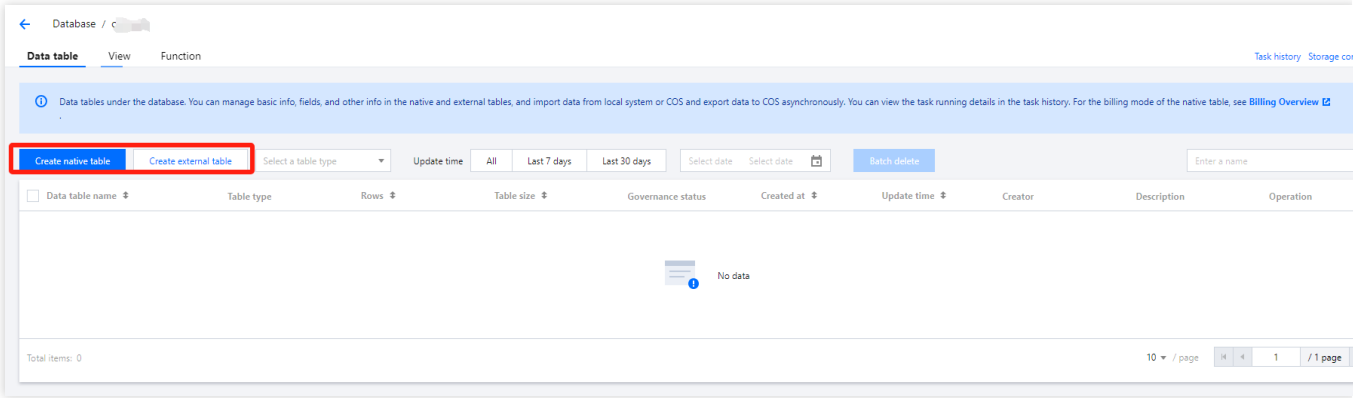
The Data Management module supports managing native tables and external tables stored in DLC.

1. Log in to the DLC console, select the service region, log in to users need to have the permission to create tables.

2. Through the left menu, enter **Data Management**, enter **Database**, click the name of the database where the data table is located, enter the DMC page.



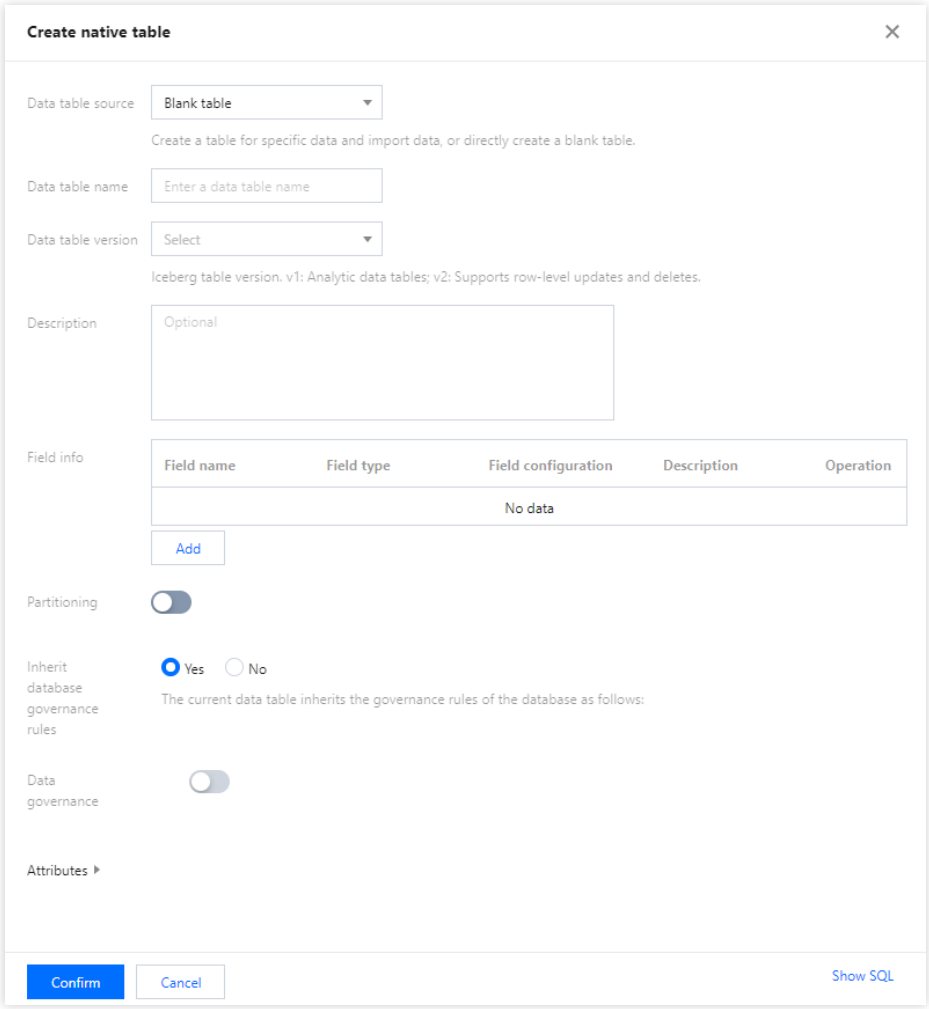
3. Click **Create Native Table** or **Create External Table** button to enter the data table configuration page.



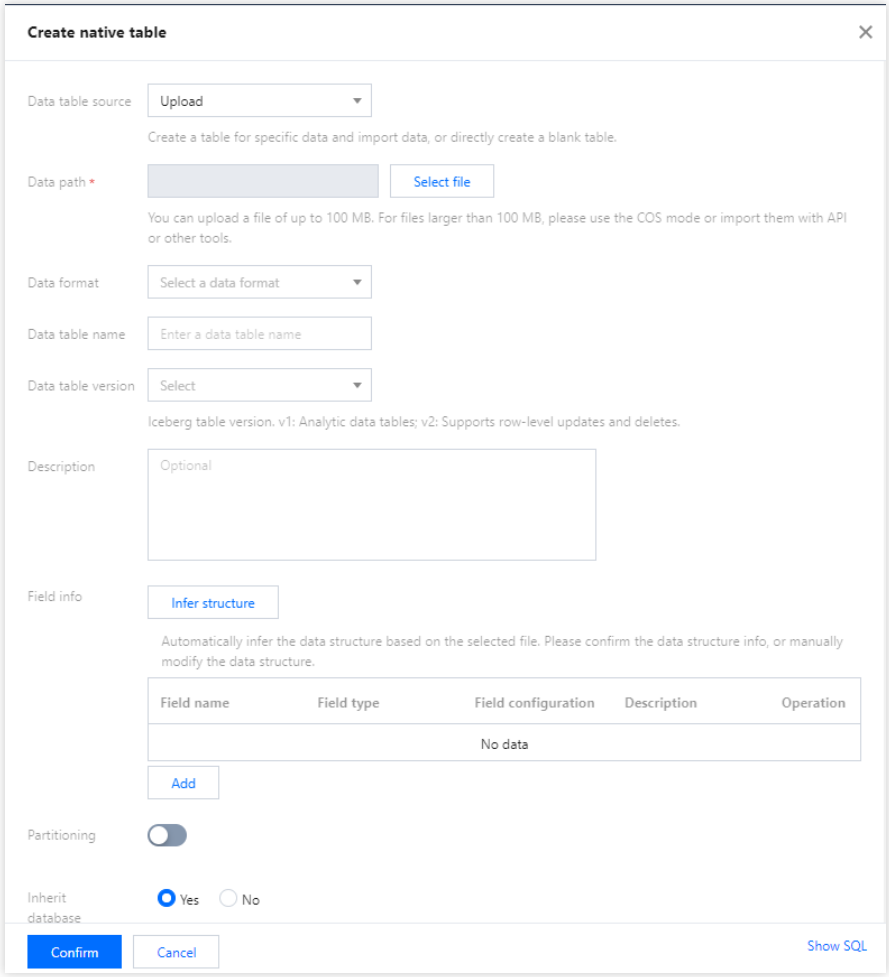
Native table data sources support three different types: empty table, local upload, and COS COS. Choosing different data sources corresponds to different creation processes. Native tables support capabilities such as data optimization and can choose to inherit database governance rules or individually turn them on/off.

3.1 Create Empty Table: Create an empty table with no records.

Data Table Name: Cannot start with a number, supports uppercase and lowercase letters, numbers, and underscores —, with a maximum of 128 characters.
Support for entering data table description information.
Manually add and enter column names and field types. Supports the configuration of three complex type fields: array/map/struct.



3.2 Local Upload: Upload local form files to DLC to create data tables, supports files up to 100MB.
CSV: Supports visual configuration of CSV parsing rules, including Compression Format, Column Splitting Symbol, Field Domain Symbol. Supports automatic inference of the data file's Schema and parsing the first row as Column Names.
Json: DLC only recognizes the first level of Json as columns, supports automatic inference of the Json file's Schema. The system will recognize the first level fields of Json as Column Names.
Supports common Big Data Format files like Parquet, ORC, AVRO, etc.
Manually add and enter Column Names and Field Types.
If the Automatic Structure Inference is selected, DLC will automatically fill in the detected columns, Column Names, and Field Types. If incorrect, please manually modify.



3.3 Create a data table through COS COS.

Create a data table by reading the COS data buckets under the current account.

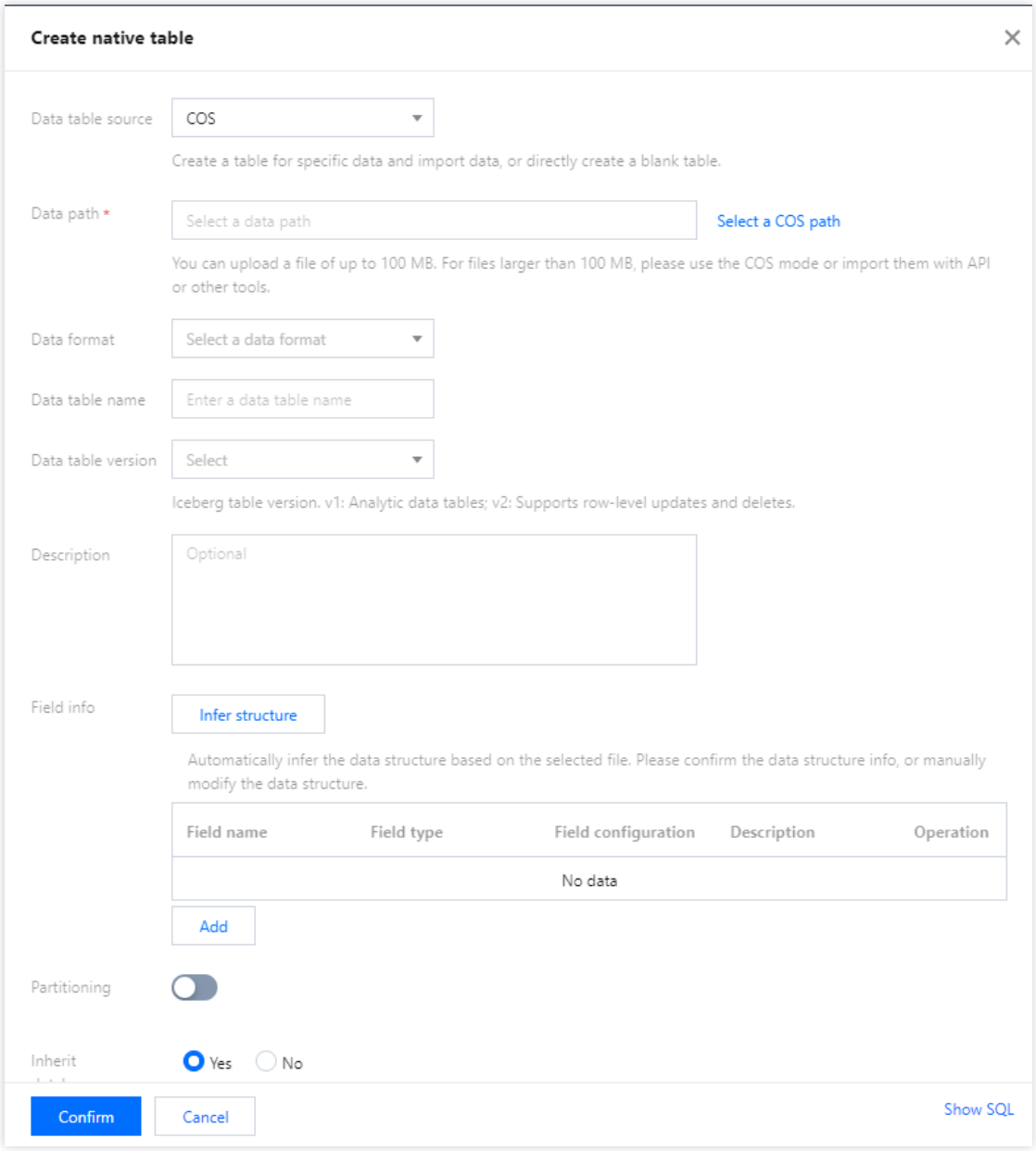
CSV: Supports visual configuration of CSV parsing rules, including Compression Format, Column Splitting Symbol, Field Domain Symbol. Supports automatic inference of the data file's Schema and parsing the first row as Column Names.

Json: DLC only recognizes the first level of Json as columns, supports automatic inference of the Json file's Schema. The system will recognize the first level fields of Json as Column Names.

Supports common Big Data Format files like Parquet, ORC, AVRO, etc.

Manually add and enter Column Names and Field Types.

If the Automatic Structure Inference is selected, DLC will automatically fill in the detected columns, Column Names, and Field Types. If incorrect, please manually modify.



4. Data Partitioning is often used to enhance Query Performance and is applied to large volume tables. DLC supports data querying by Data Partitioning. Users need to add partition information at this step. By partitioning your data, you can limit the amount of data scanned with each query, thereby improving Query Performance and reducing usage costs. DLC adheres to Apache Hive's partitioning rules.

The partition column corresponds to a subdirectory under the COS path of the table, with the directory naming convention being **Partition Column Name=Partition Column Value**.

Example:

```
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, they need to be nested in the order specified in the create table statement.

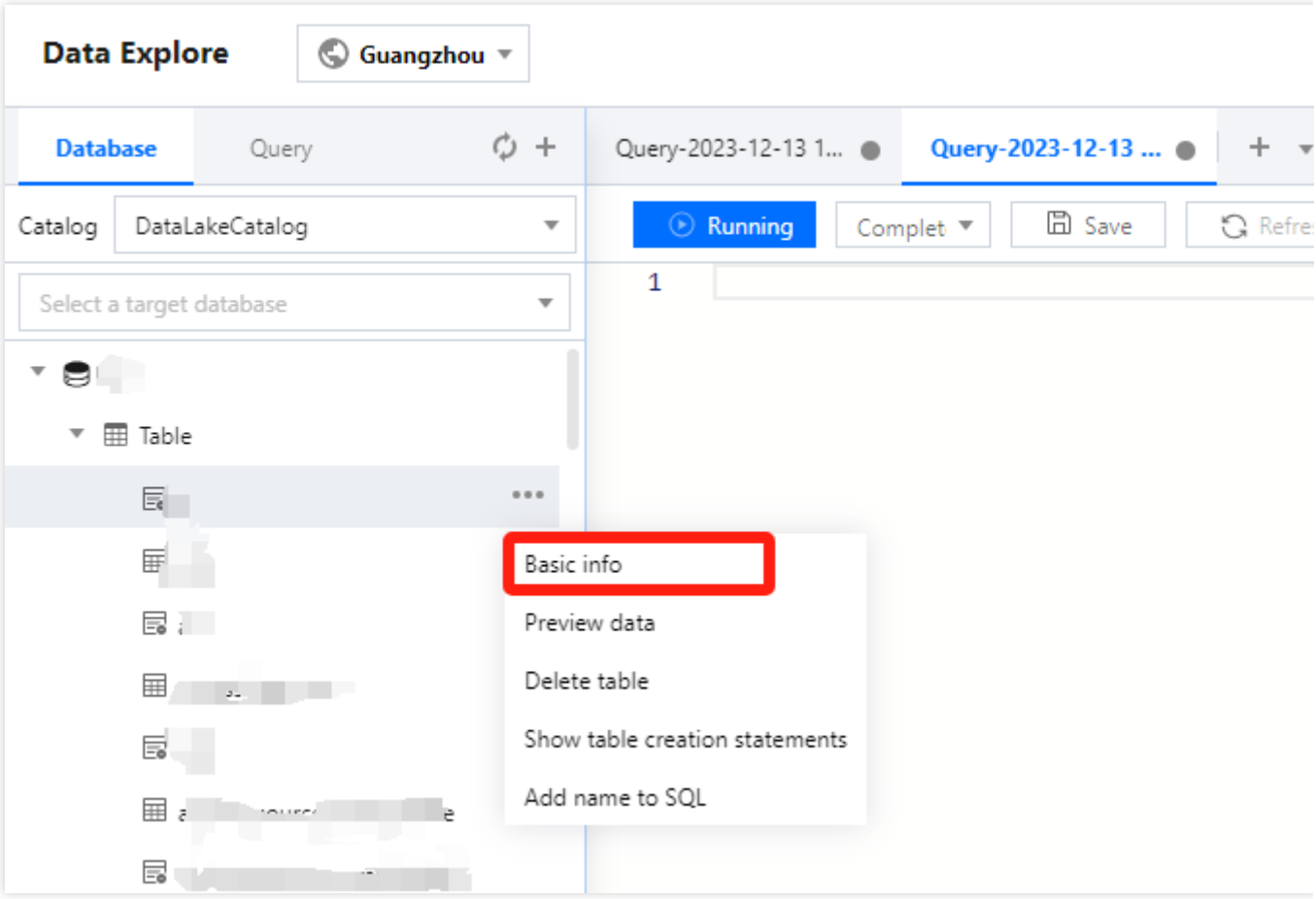
```
CREATE EXTERNAL TABLE IF NOT EXISTS `COSDataCatalog`.`dlc_demo`.`table_demo` (
    `_c0` string,
    `_c1` string,
    `_c2` string,
    `_c3` string
) PARTITIONED BY (`year` string, `month` string, `day` string)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES ('separatorChar'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

Query basic information of the data table

Approach one: Query in Data Exploration

In the Data Table Item, mouse hover over the **Data Table Name** row, then click the

icon, in the Dropdown Menu click **Basic info** to view the basic information of the created data table.

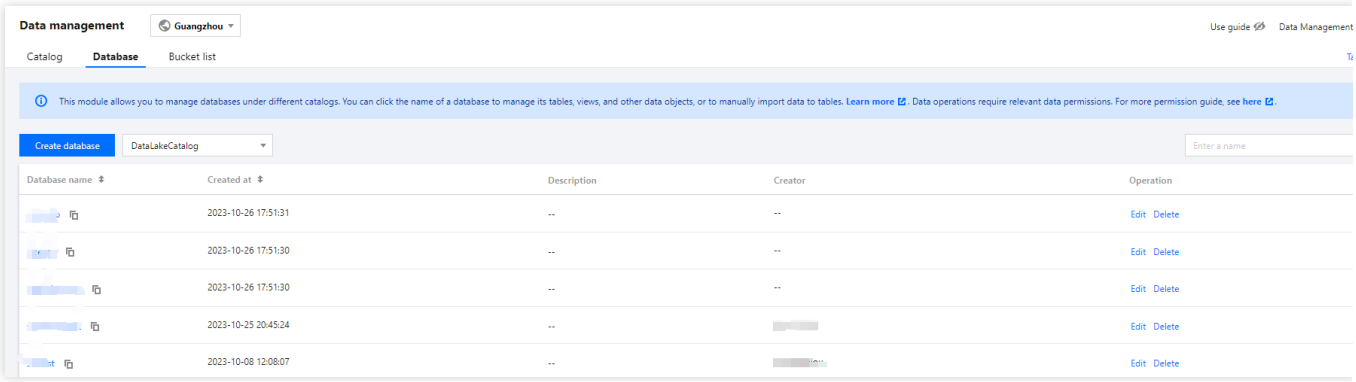


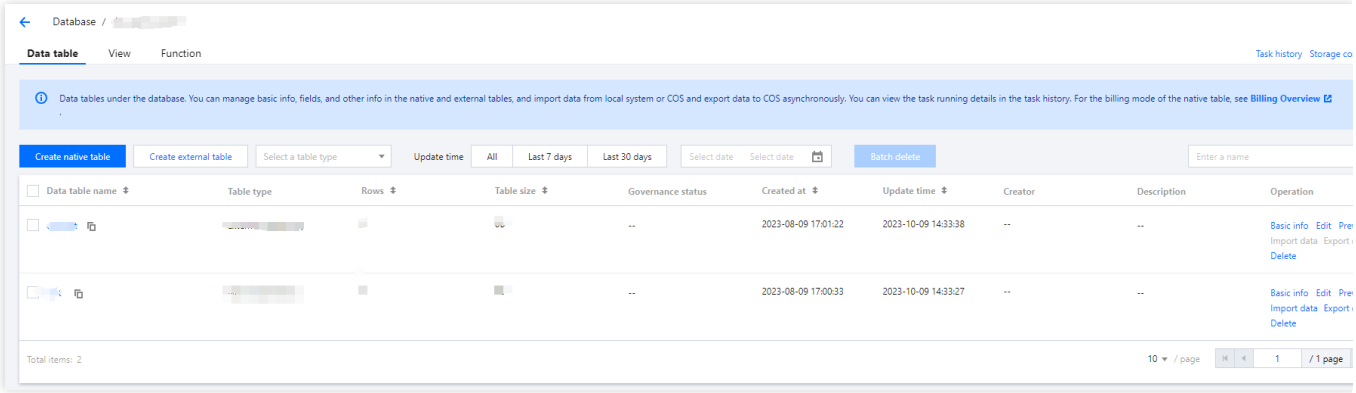
The basic information of the data table is as follows:

Approach two: View in Data Management

1. Log in to the DLC Console, select the service region. Users need to have the permission to view data tables.

2. Through the left menu, enter the **Data Management** page, click the name of the database where the data table is located, enter the DMC page. It supports querying information such as the number of rows, storage space, creator, fields, partitions, etc.

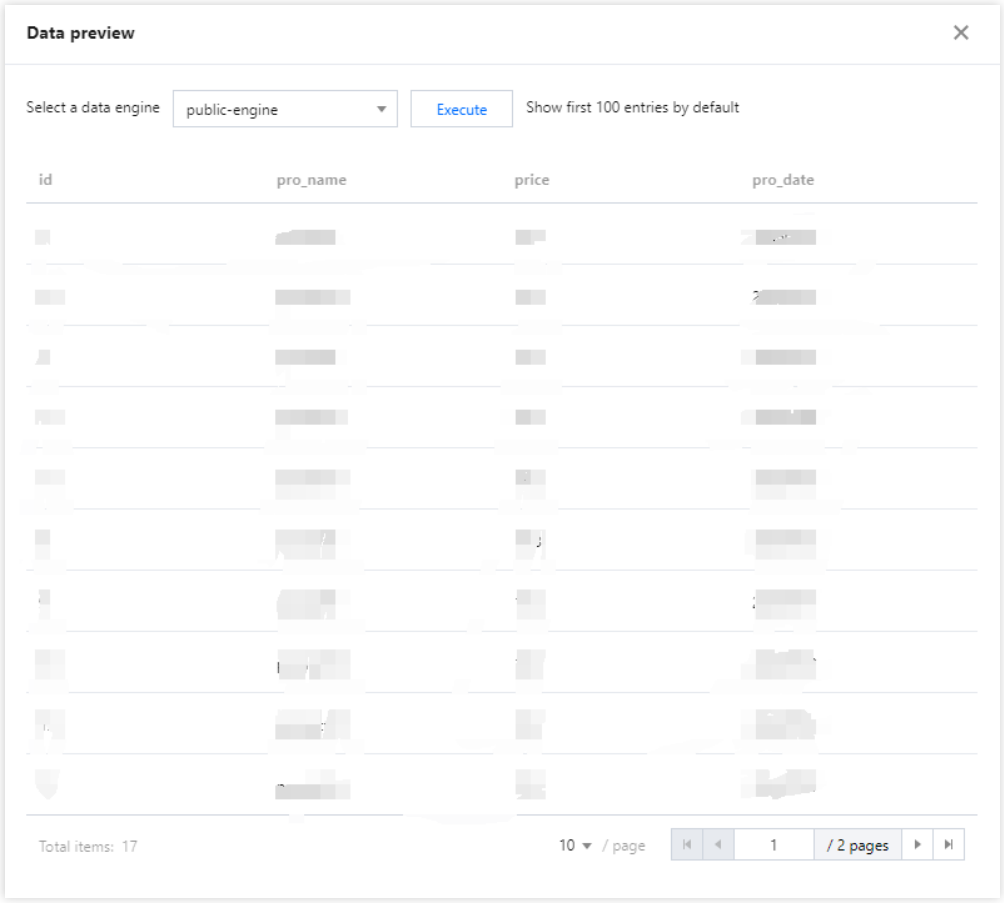




Preview Data Table Data

In the Data Table Item, hover the mouse over the **Data Table Name** row, then click the

icon, in the Dropdown Menu click **Preview Data**. DLC will automatically generate a SQL Statements to preview the first 10 rows of data, executing the SQL Statements to query the top 10 rows of the data table.



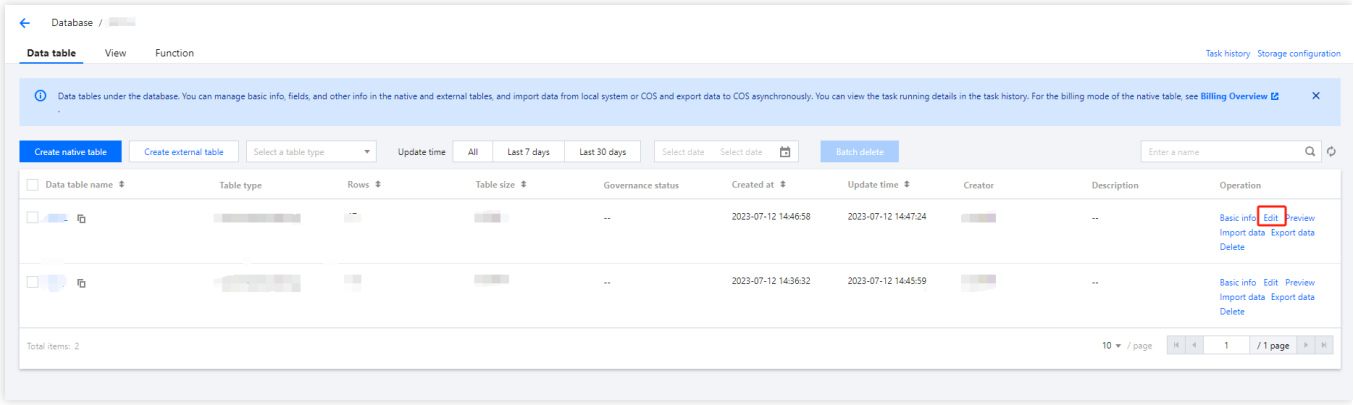
Support for previewing data in **Data Management > Database > Data Table > Data Table List**.

The Data Preview Function by default displays the first 100 rows of data.

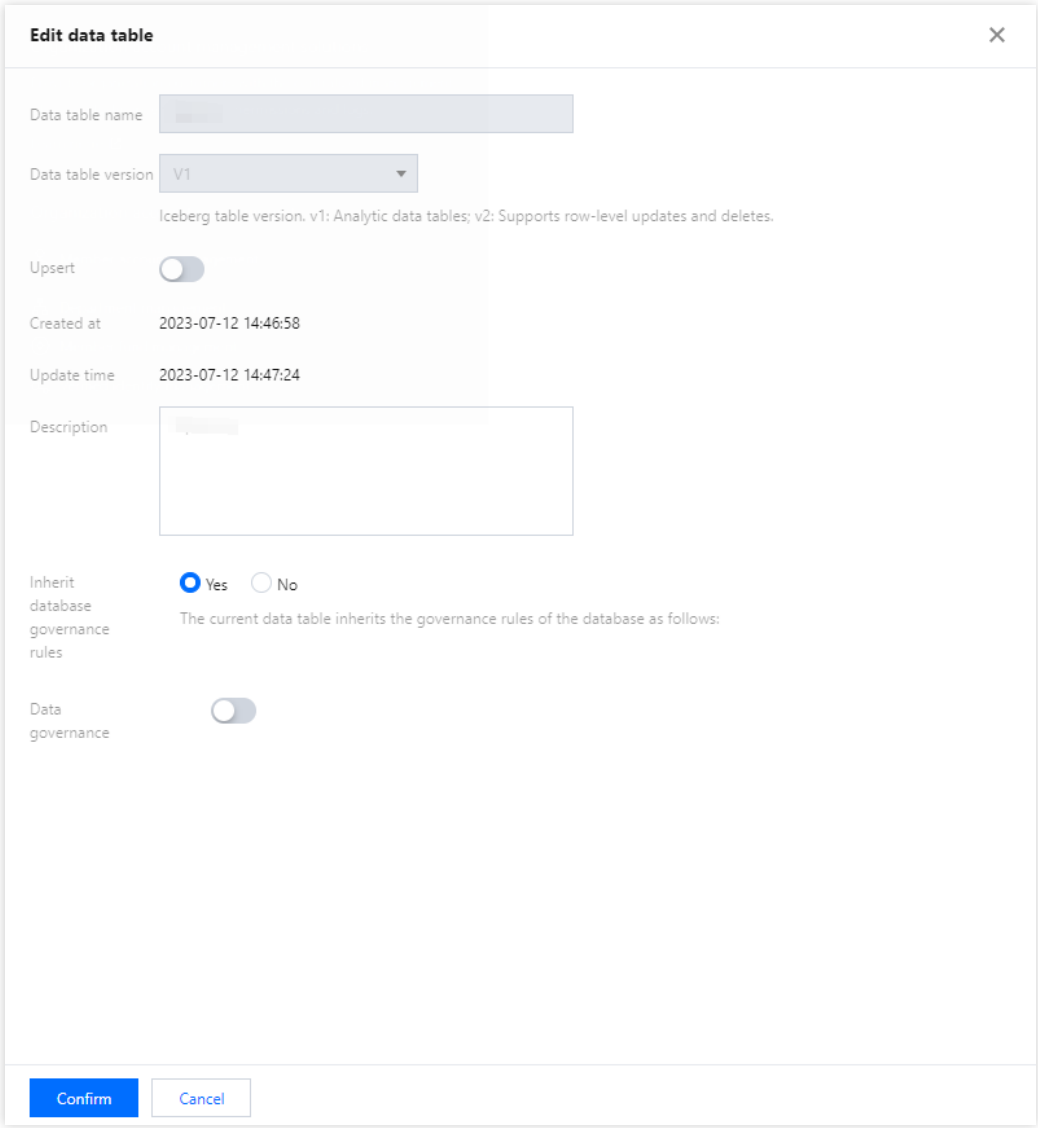
Editing Data Table Information

Support editing the Description information of the data table in the Data Management module.

1. Log in to the DLC Console, select the Service Region. Users need to have the permission to edit data tables.
2. Through the left menu, enter the **Data Management > Database** page, click the name of the database where the data table is located, enter the DMC page.
3. Find the data you need to edit, click the **Edit** button on the right to edit.



4. After modification, click the **Confirm** button to complete the editing.



Dropping a Table

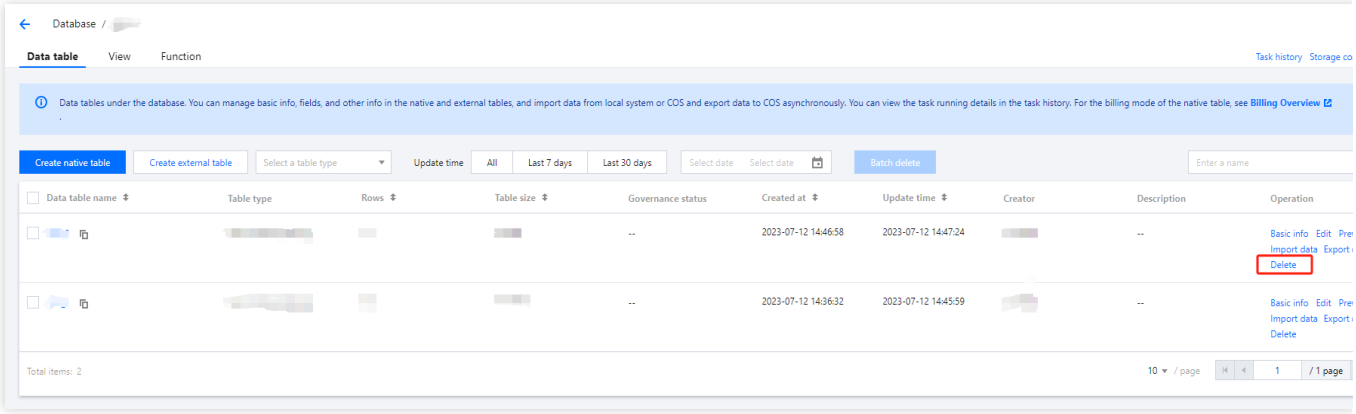
Approach one: Delete in Data Exploration

In the Data Table Items, hover the mouse over the **Data Table Name** row, then click the

icon, in the dropdown menu click **Delete**. DLC will automatically generate the SQL statement to drop the data table, execute the SQL statement to drop the table.

Dropping an external table, dropping a data table only removes the metadata stored in DLC, it does not affect the data source file.

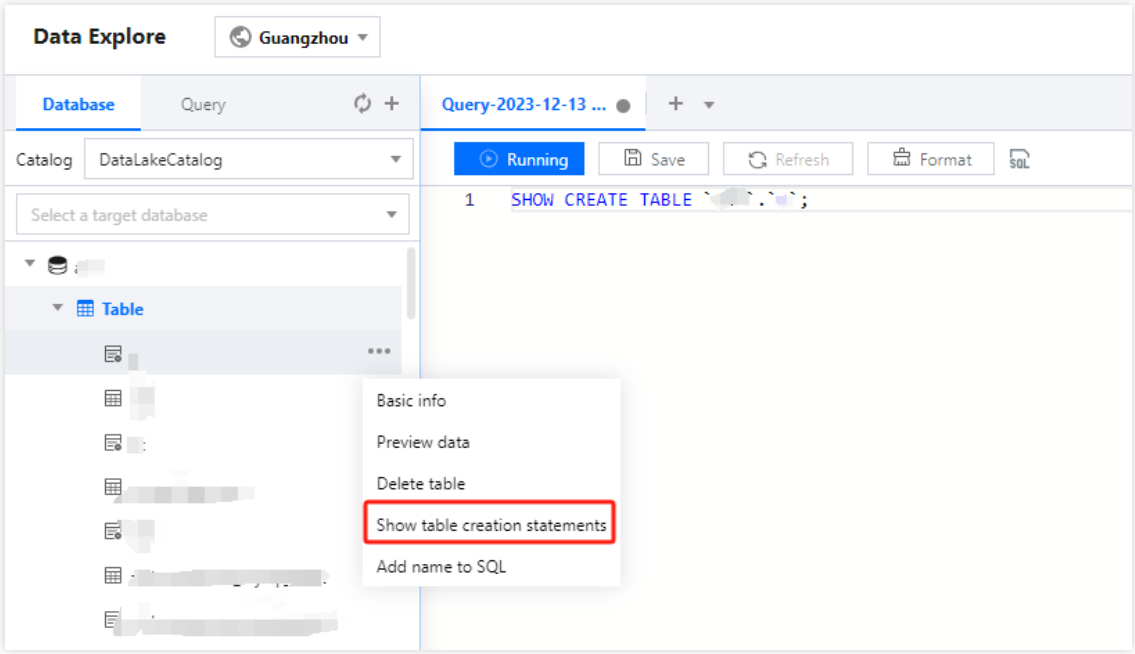
Deleting tables under the DataLakerCatalog directory will clear all data of that table, proceed with caution.



Approach two: Delete in Data Management

Currently, Data Management only supports the management of databases and tables hosted in DLC. For external tables, please use approach one for deletion.

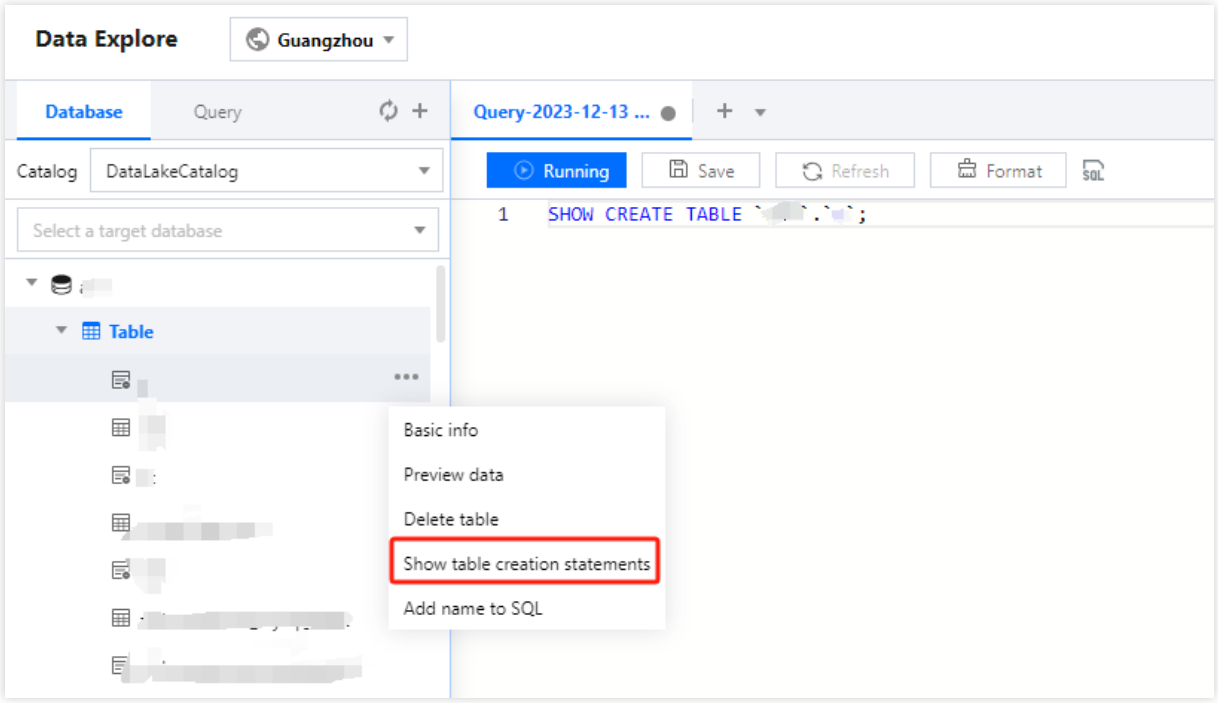
1. log in to the DLC Console, select the service region, users need to have the permission to delete data tables.
2. Through the left menu, enter **Data Management > Database**, click the name of the database where the data table is located, to enter the DMC page.
3. Click the **Delete** button after the data table you wish to delete, after confirmation, the corresponding data table can be deleted and its data cleared.



Show create table statement

In the Data Table Item, hover the mouse over the **Data Table Name** row, then click the

icon, in the dropdown menu click **Show table creation statements**. DLC will automatically generate the SQL statement to view the create table statement for that data table, execute the SQL statement to query the create table statement.



System constraints

DLC allows up to 4096 data tables under each database, supports a maximum of 100,000 partitions per data table, and the maximum number of attribute columns per data table is 4096.

DLC will recognize data files under the same COS path as data from the same table, please ensure data for separate tables is kept in separate folder hierarchies.

DLC does not support multi-version data in COS; it can only query the latest version of data in a COS bucket.

All tables created on DLC are external tables, and the SQL statement to create the table must include the EXTERNAL keyword.

Table names must be unique within the same database.

Table names are case-insensitive and only support letters, numbers, and underscores (_), with a maximum length of 128 characters.

If the table is a partitioned table, you must manually execute the ADD PARTITION statement or the MSCK statement to add partition information before you can query the partition data. For more details, see Query partitioned table.

When creating a table with CSV, DLC will by default convert all field types to string, but this does not affect the computation and querying of raw data fields.

Data View Management

Last updated：2024-07-31 17:28:41

DLC provides data view query capabilities, allowing users to quickly and easily perform data queries and use through the management of data views.

Create View

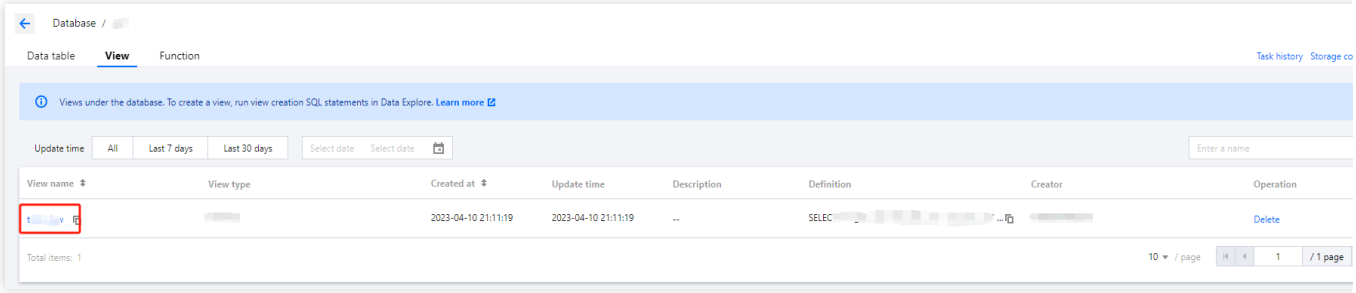
1. log in to DLC console, select the service region, log in users must have the permission to create views.
2. Enter the **Data Exploration page**, you can create views using SQL statements. For details of the statement, see SQL Syntax.
3. Select the computing resource, click the **Running** button to complete view creation.

View Views

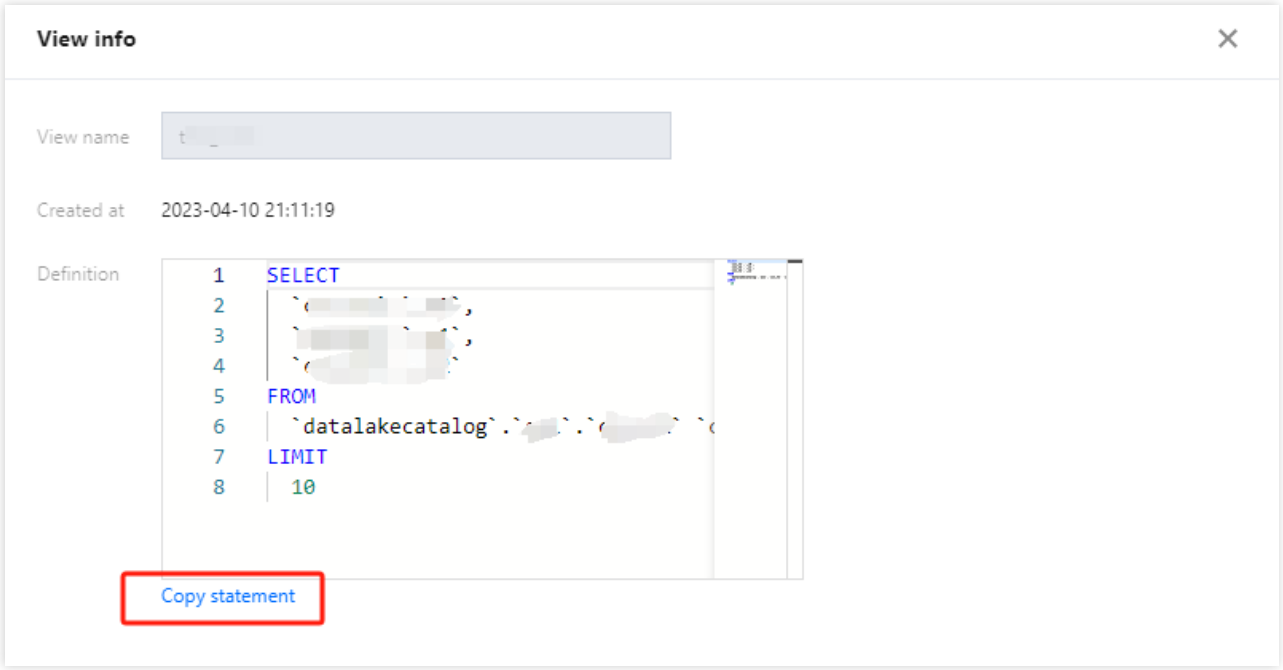
You can view the view using SQL statements through Data Exploration, see SQL Syntax for specific syntax.

Meanwhile, DLC also offers a Visual Interface for managing views, with the following operations.

1. Log in to the DLC console, select the service region, log in users must have the permission to query views.
2. Enter the Data Management page, click on the **Database Name** where the view is located to enter the DMC page.
3. Click **View** to enter View Management.



4. Click the **View Name** you want to inspect to view its information. You can copy the SQL statement.

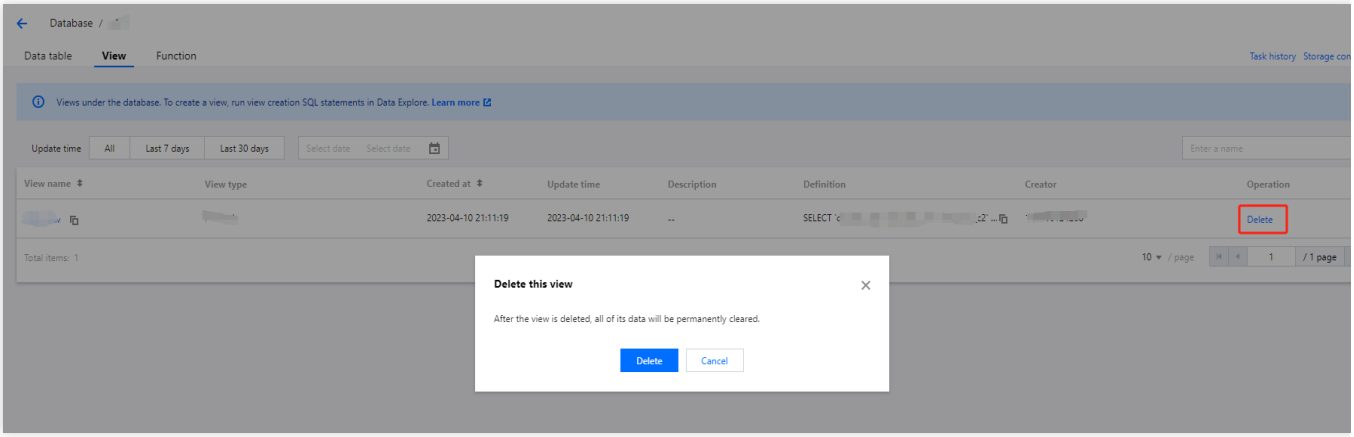


Delete View

You can view the view using SQL statements through Data Exploration, see SQL Syntax for specific syntax.

Meanwhile, DLC also offers a Visual Interface for managing views, with the following operations.

1. Log in to DLC Console, select the service region, users must have view deletion permissions.
2. Enter the Data Management page, click on the **Database Name** where the view is located to enter the DMC page.
3. Click **View** to enter View Management, then click the **Delete** button to delete the view.



Caution

Deleting a view will clear all data under the view and cannot be recovered. Please proceed with caution.

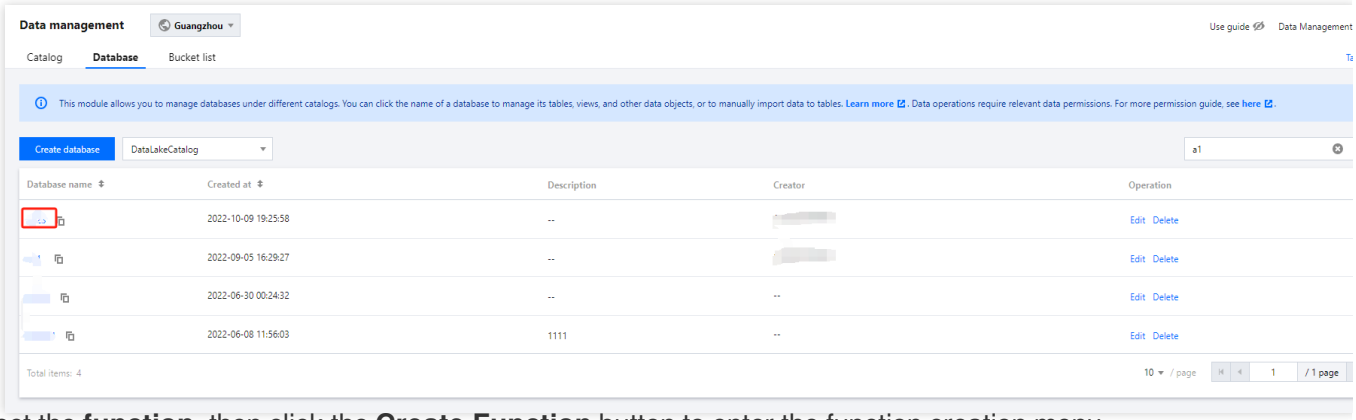
Function Management

Last updated：2025-03-07 15:27:24

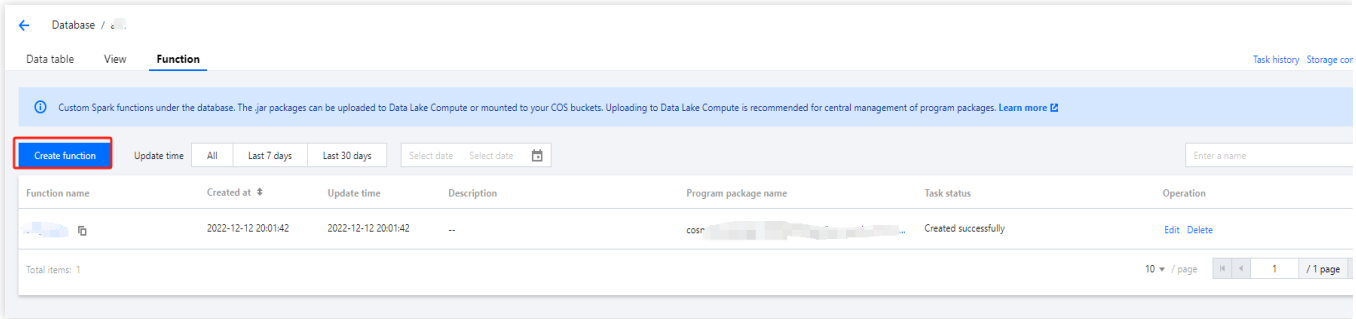
Data Lake Compute (DLC) supports using **user-defined functions** to process and build data, as well as managing functions.

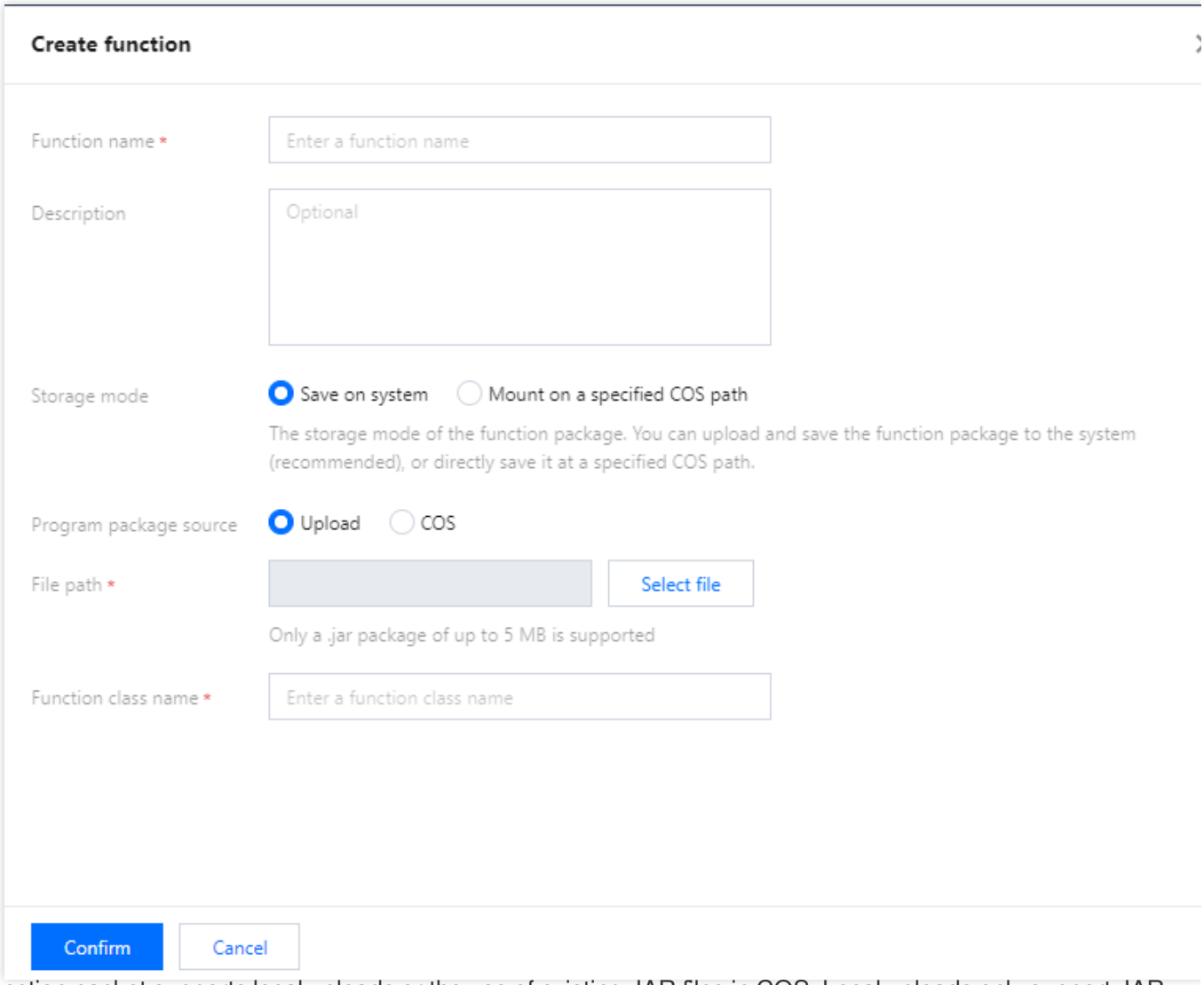
Creating a Function

1. Log in to the DLC Console and select the service region. Ensure the logged-in account has database operation permissions.
2. Go to the **Data Management Page** and click the **database name** where you want to create the function.



3. Select the **function**, then click the **Create Function** button to enter the function creation menu.





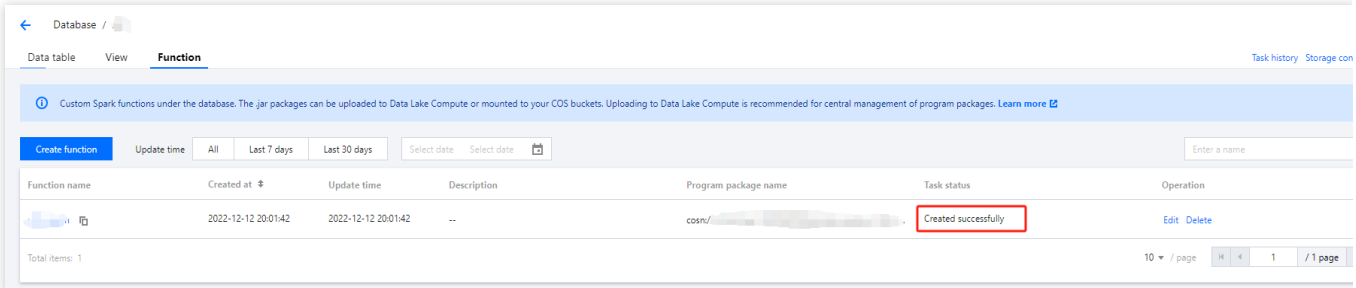
The function packet supports local uploads or the use of existing JAR files in COS. Local uploads only support JAR format, with a maximum size of 5 MB.

Select the Spark cluster to run the function. There will be no fees incurred during the execution.

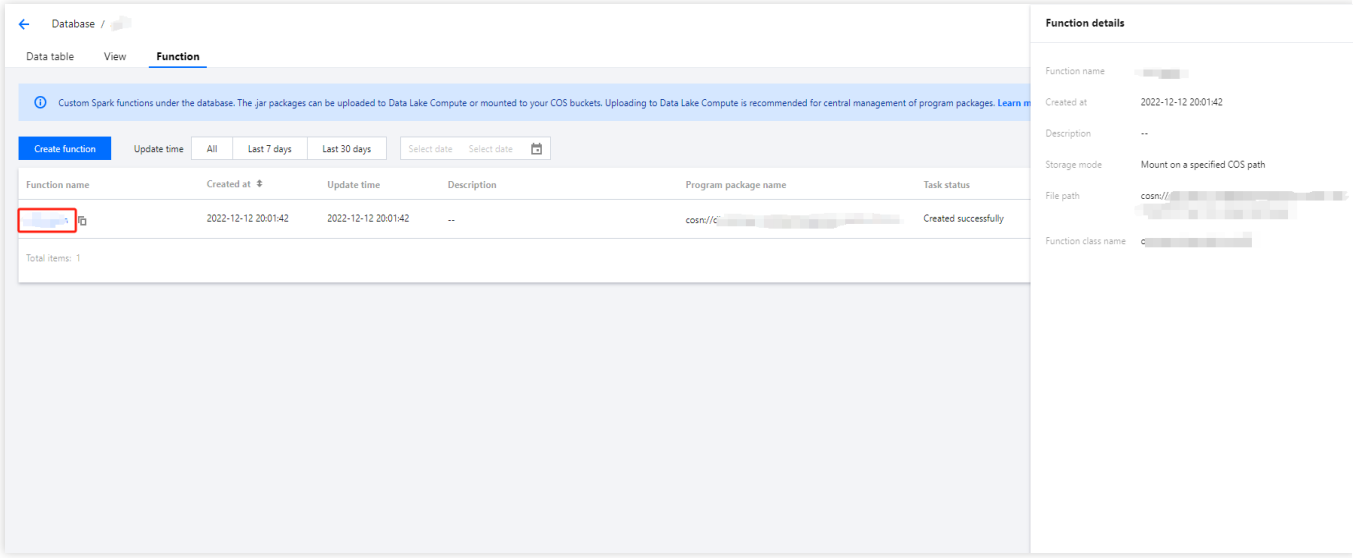
It is recommended to save the function package to the system for easy management and use. You can also mount it to a specified COS path.

Viewing Function Information

1. Log in to the DLC Console and ensure the account has database operation permissions.
2. Go to the **Data Management Page** and click the **database name** where the function is located.
3. Select the function to view its build status. If the build fails, you can **edit** and resubmit it.

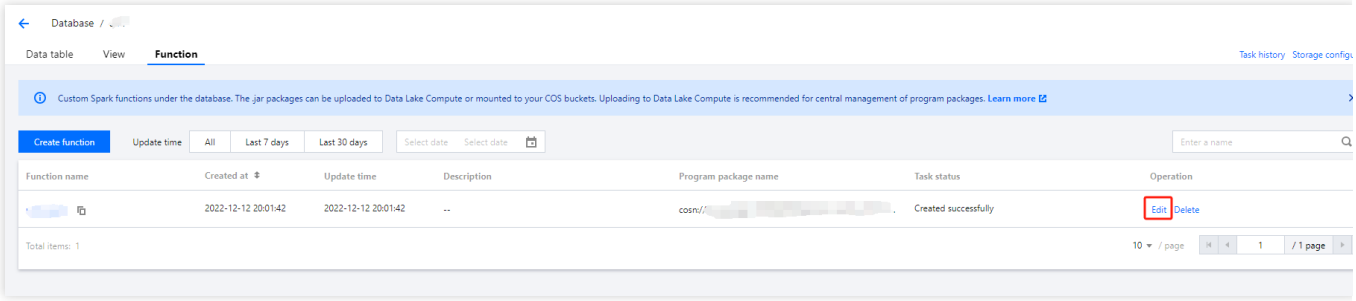


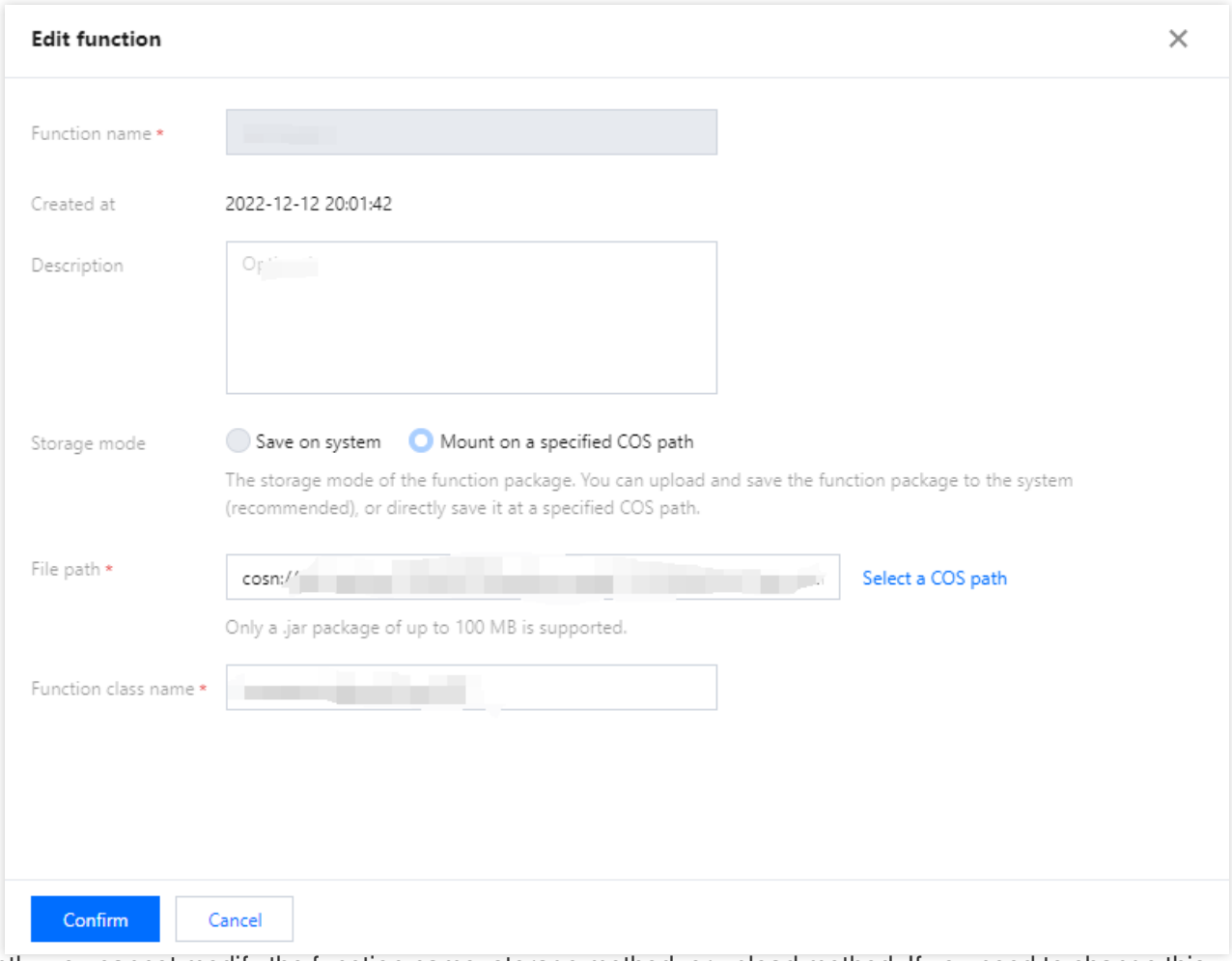
4. Click the **Function Name** to view detailed information about the function.



Editing Function Information

1. Log in to the DLC Console and select the service region, and ensure the logged-in account has database operation permissions.
2. Go to the **Data Management Page** and click the **database name** where the function is located.
3. Select the **function**, then click the **Edit** button to enter the function information editing page.





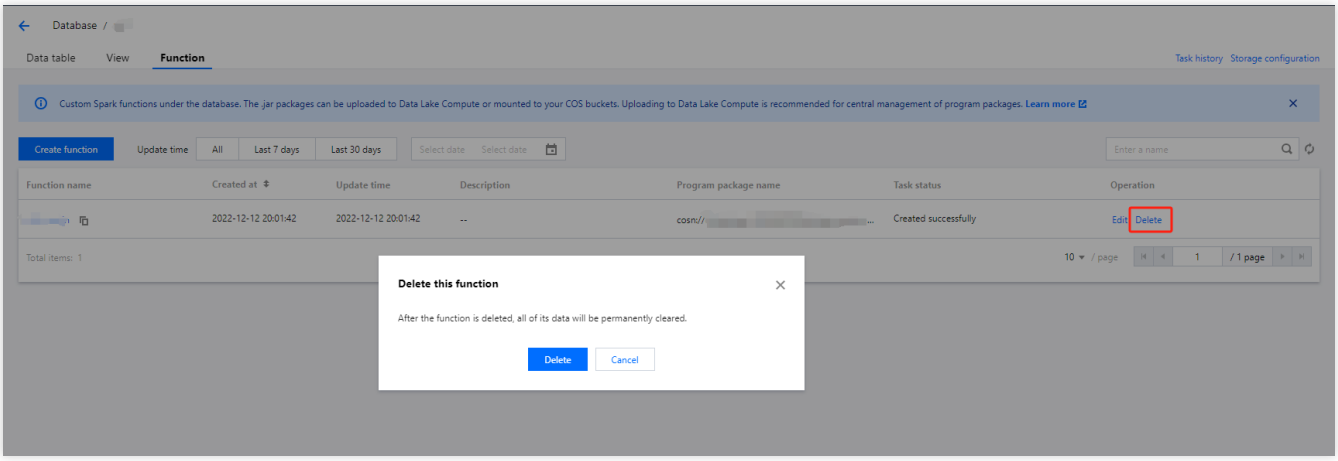
Currently, you cannot modify the function name, storage method, or upload method. If you need to change this information, you must recreate the function.

After the function information is modified, it will be rebuilt. Please operate with caution.

Deleting a Function

For functions that are no longer needed, you can delete them.

1. Log in to the DLC Console and select the service region. Ensure the logged-in account has database operation permissions.
2. Go to the **Data Management Page** and click the **database name** where the function is located.
3. Select the **function**, then click the **delete** button to remove the function that is no longer needed.



Note

After deletion, the data under this function will be cleared and cannot be recovered. Please operate with caution.

Partition Field Policy

Last updated：2024-07-31 17:29:14

In Hive, partition information appears in the form of directories. In Iceberg, partition information is recorded in the underlying data files, making Iceberg's partitions more flexible and allowing the partitioning strategy to evolve with changes in data volume. In DLC, you can create Iceberg tables to utilize features such as hidden partitions.

Note:

By default, native tables are Iceberg tables. External tables, depending on the file format, can choose between Hive or Iceberg tables. For detailed syntax, refer to the document CREATE TABLE.

With hidden partitions, when inserting and querying data, you do not need to specify partition information additionally as required in Hive.

Iceberg partition strategy supports the use of the following functions, with different fields and corresponding partition transformation strategies as shown in the table:

Partitioning Strategy	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	date
hour	timestamp, timestamptz	int

Ops Management
Historical Task Instances

Last updated：2025-06-12 12:01:53

Historical Task Instances focus on recording and managing various types of tasks performed by users in DLC for subsequent tracking, review, and optimization. Through the Historical Task Instances feature, users can quickly view the execution status of tasks, including start and end times, execution status (such as successful or failed), input and output details, and generated logs or error information. It provides users with the convenience of auditing and retrieval, helping users identify task health status, potential issues, and optimize resource configuration, etc.

Operation Steps

1. Log in to Data Lake Compute (DLC) Console > Ops Management > Historical Task Instances and choose service region.
2. Enter the historical task instances page. Administrators can view all historical operation tasks in the past 45 days, and general users can query tasks related to themselves in the past 45 days.
3. Support filtering and viewing by task type, task status, creator, task time range, task name, ID, content, sub-channel, and other methods.
4. Click the task ID/name. Support view task details, including modules such as basic information, running result, task insights, and task logs.
5. Support user click to modify task configuration, quickly enter job details to adjust configuration for optimization.

Historical Task Instances List

Note:
The *field supports after enabling the insight feature. For enablement method, please see How to Enable Insight Feature.

Field Name	Description
Task ID	Unique identifier of the task.
Task name	Prefix_yyyymmddhhmmss_eight-digit uuid, where yyyymmddhhmmss is the task execution time. Prefix rule: 1. The job task submitted by the console is prefixed with the job name. For example, if the user-created job is customer_segmentation_job and it is executed at 21:25:10 on November

	26, 2024, the task id will be customer_segmentation_job_20241126212510_f2a65wk1. According to the current data format restriction, the job name should be <= 100 characters. 2. SQL type submitted on the data exploration page, prefixed with sql_query. Example: sql_query_20241126212510_f2a65wk1. 3. Data optimization tasks, according to the prefixes of different sub-types of optimization tasks, among them: 3.1 The prefix of the optimizer is only optimizer. 3.2 The SQL type of the optimized instance is optimizer_sql. 3.3 The batch type of the optimized instance is optimizer_batch. 3.4 Configuration task created when configuring the data optimization policy: optimizer_config. 4. Import data task, prefixed with import, for example: import_20241126212510_f2a65wk1. 5. Export data task, prefixed with export, for example: export_20241126212510_f2a65wk1. 6. Wedata submission, prefixed with wd, for example: wd_20241126212510_f2a65wk1. 7. Other API submissions, prefixed with customized, for example: customized_20241126212510_f2a65wk1. 8. Tasks created for metadata operations on the metadata management page, prefixed with metadata, for example: metadata_20241126212510_f2a65wk1.
Task status	Starting Executing Queuing up Successful Failed Canceled Expired Task run timeout
Task content	Detailed content of the task. For job type tasks, it is a hyperlink to job details; for SQL type tasks, it is the complete sql statement.
Task type	Be divided into Job type, SQL type.
Task source	The origin of this task. Support data exploration tasks, data job tasks, data optimization tasks, import tasks, export tasks, metadata management, Wedata tasks, and API submission tasks.
Sub-channel	Users can customize sub-channels when submitting tasks via the API.
Compute resource	The computing engine/resource group used to run the task.
Consumed CU*H	During task execution, CU*H consumption occurs. Please note that the final CU consumption is subject to the bill, and the final result may vary. In the Spark scenario, it is approximately equal to the sum of Spark task execution durations divided by 3600.
Compute time	1. If the task supports insight feature, it is the execution time within the engine.

	2. If the task does not support insight feature: 2.1 For a Spark SQL task, it is the platform scheduling time + consumed queuing time within the engine + execution time within the engine. 2.2 For a Spark job task, it is the platform scheduling time + engine startup duration + queuing time within the engine + execution time within the engine. The execution time within the engine is the duration from the start execution of the first task of a Spark task to the task completion.
Scanned data volume	The physical data volume read from storage by this task is approximately equal to the sum of Stage Input Size in Spark UI in the Spark scenario.
*Scanned data records	The number of physical data entries read from storage by this task is, in the Spark scenario, approximately equal to the sum of Stage Input Records in Spark UI.
Creator	If it is a job type task, it refers to the creator of the job.
Executor	The user running the task.
Submitted at	The time when the user submits tasks.
*Engine execution time	The time when the first preemption of the CPU starts execution of the task, the start execution time of the first task within the Spark engine.
*Number of output files	The collection of this metric requires upgrading the Spark engine kernel to a version later than 2024.11.16. Total number of files written by tasks through statements such as Insert. Case-insensitive to task type.
*Output small-sized files	The collection of this metric requires upgrading the Spark engine kernel to a version later than 2024.11.16. Small File Definition: An individual file size of the output that is less than 4 MB is defined as a small file (controlled by the parameter spark.dlc.monitorFileSizeThreshold, with a default value of 4 MB, which can be configured globally or at the task level for the engine). This metric definition: Total number of small files written by tasks through statements such as insert. Case-insensitive to task type.
*Total output lines	The number of records output after this task processes data is, in the Spark scenario, approximately equal to the sum of Stage Output Records in Spark UI.
*Total output size	The Size of the record output after this task processes data is, in the Spark scenario, approximately equal to the sum of Stage Output Size in Spark UI.
*Data shuffle lines	Approximately equal to the sum of Stage Shuffle Read Records in Spark UI in the Spark scenario.
*Data shuffle size	Approximately equal to the sum of Stage Shuffle Read Size in Spark UI in the Spark scenario.

*Health status	Analyze the task to judge the health status of the task and determine whether optimization is required. Please see Task Insight for details.

Historical Task Instances Details

Basic Info

1. Users can view specific task content in **execution content**. For SQL tasks, view the complete SQL statement; for job tasks, view job details and job parameters.
2. Users can view relevant content about task resources in **resource consumption**, including consumed CU*H, computational overhead, scanned data volume, compute resource, kernel version, Driver resource, Executor resource, and count of Executors.
3. Users can view basic information of tasks in **basic info**, including task name, task ID, task type, task source, creator, executor, submission time, and engine execution time.
4. For tasks running on the SuperSQL SparkSQL or SuperSQL Presto engine, users can view the task running progress bar in **query statistics**, which includes the time taken for stages such as creating tasks, scheduling tasks, executing tasks, and obtaining results.

Running Result

After task completion, users can query the task result on the execution result page. There are two types of task results:
1. Write file information: For file writing tasks running on SuperSQL, standard engine, or Spark kernel engine, support user viewing of write file information.
Average file size
minimum file size
maximum file size
Total file size
2. Execution result: SQL task query statement, which can display the query result of the current task and support users to download query results.

Task Insight

After task completion, users can view task insight results on the task insight page. It supports analyzing the aggregate metrics that each task has executed and insights into optimizable issues. Based on the actual execution situation of the current task, DLC task insight will combine data analysis and algorithm rules to provide corresponding optimization suggestions. For details, please see Task Insight.

Task Log

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

Note：

Only the job type supports task log viewing.

1. Support switching logs of nodes in different clusters through Pod Name, including Driver, Executor, etc.
2. Support three log level filters: All, Error, Warning.
3. This page only displays the last 1000 logs. If you need to view all log entries, you can export logs.
4. Support viewing log export records and the status of export tasks. In log export records, users can save log files locally.

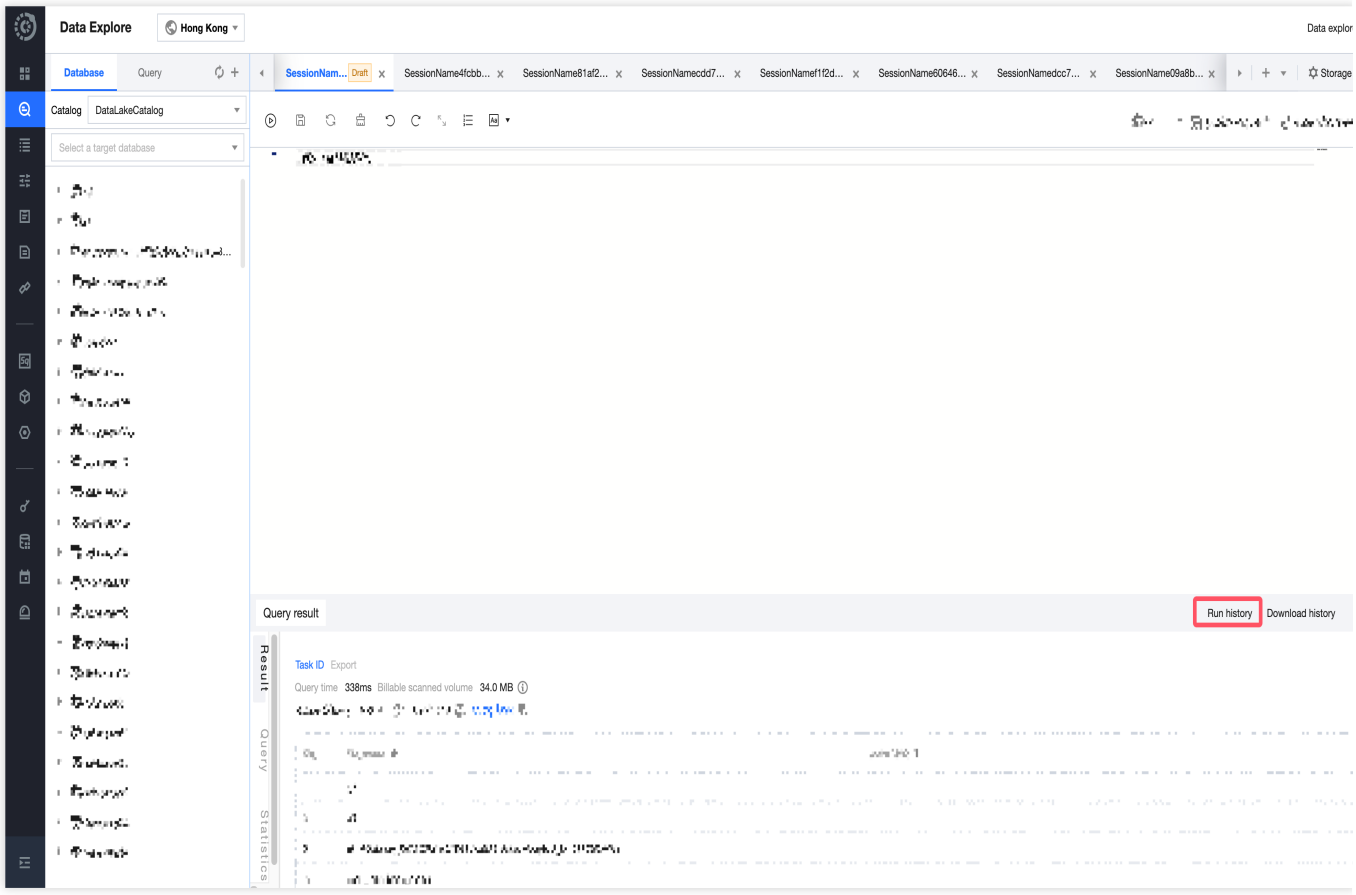
Historical task(Old version)

Last updated：2025-03-21 12:22:27

To facilitate users in querying historical task records, DLC provides three methods to search and process historical tasks.

View historical tasks run in the Query Editor

1. Log in to DLC console, select the service region.
2. Enter the **Data Exploration Page**, click on **Run History** within a single Session to view the task run history for that Session.
3. Click on the history record **Batch ID** to view the corresponding execution results on the left



Each Session's run history is independent, and a maximum of 45 days of run history is kept.
Historical task result data is saved for 24 hours. To view task results beyond 24 hours, the task must be rerun.

View data import history in the Data Management feature

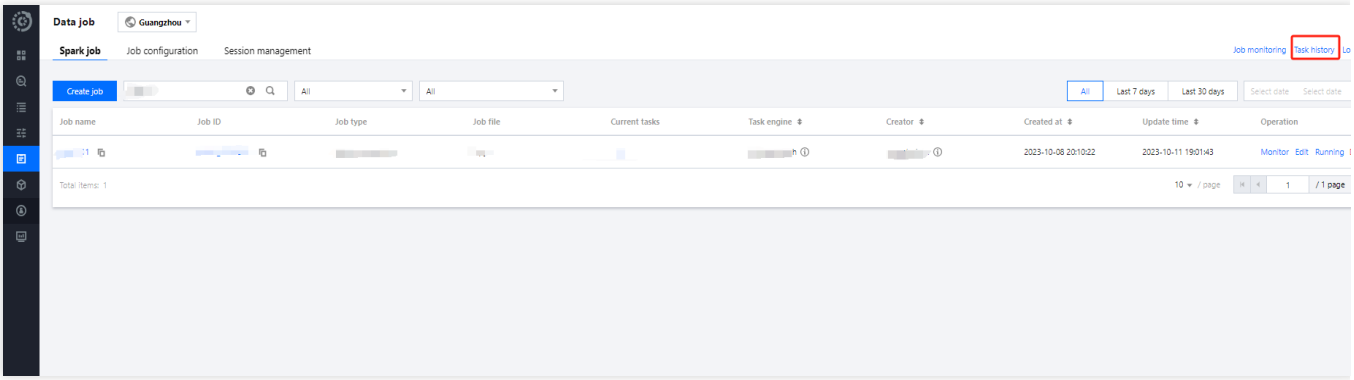
1. log in to DLC Console > Data Management, select the service region.

Note:

Log in to the account requires database-related permissions.

2. Click on **Task History** in the top right corner to query data import history tasks.

3. Supports viewing historical tasks from the past 45 days

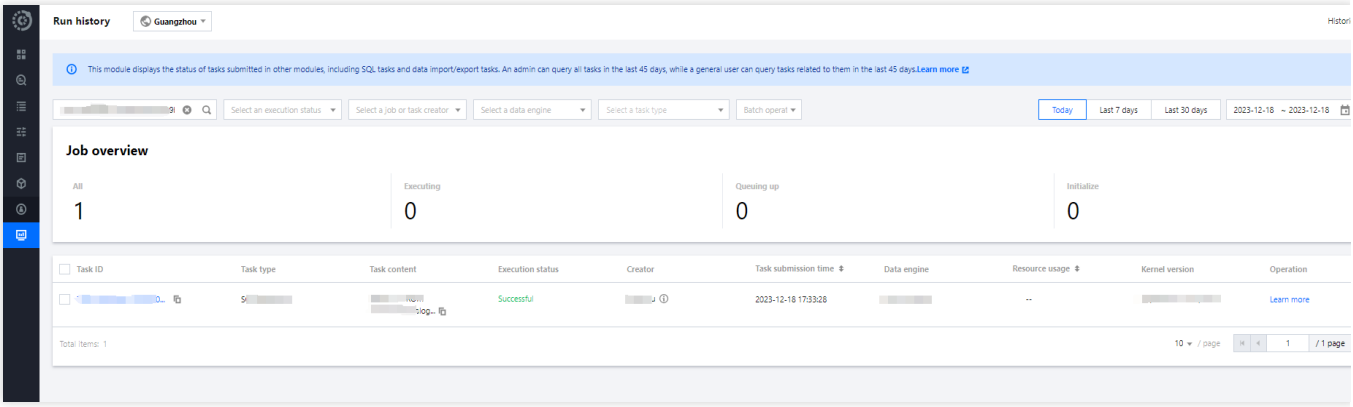


View historical tasks in the Historical Operation feature

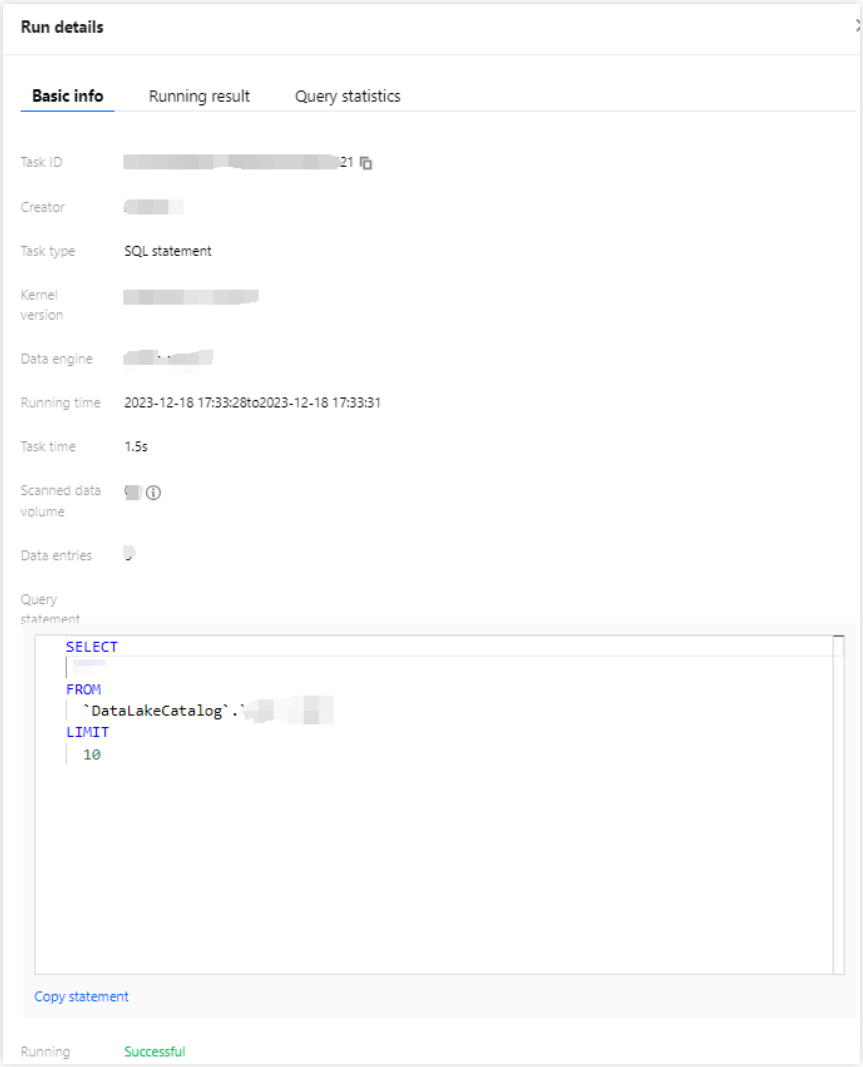
1. log in to DLC Console > Historical Operation, select the service region.

2. Enter the Historical Operation page, where administrators can view all historical operation tasks from the past 45 days, and ordinary users can query tasks related to themselves from the past 45 days.

3. Supports filtering by task type, execution status, creator, data type, etc.



4. click **Run Details** to see the task execution details and results.



Historical task result data is saved for 24 hours. To view task results beyond 24 hours, the task must be rerun. You can directly **Copy Statement** to Data Exploration to execute the task.

You can directly click **Task ID** to quickly switch and view the task execution details.

For tasks that are running, you can **Cancel** them.

Session Management

Last updated：2025-03-21 12:22:27

The session management feature is used to record and trace notebook interactive sessions submitted to the DLC engine through the API or Wedata. Users can perform operations such as SQL queries, data processing, and model training through sessions.

Prerequisites

Environment preparation for Data Lake Compute (DLC).
Enable Tencent Cloud DLC engine service.
Creating a session requires purchasing a job type engine.
SuperSQL job engine.
Standard engine: Spark engine or machine learning resource group.

Operation Steps

1. Log in to DLC Console > Ops Management > Session Management and choose service region.
2. Enter the session management page, and users can view all the historical session records.
3. Support filtering and viewing by engine type, status, Kind, engine name, Session ID, and Session Name.
4. Click Session Name/ID. View session details is supported.
5. Support users to click kill to close the session on the console.
6. Support user viewing of the Spark UI of the session.

Session List

Field Name	Description
Session Name/ID	Unique identifier for the session. Sessions created by the SuperSQL job engine only have a Session ID. Session ID rule: livy-session-uuid. Sessions created by the standard engine or Spark engine User-submitted Notebook, prefixed with session_test User-submitted batch SQL, prefixed with temporary-rg
Status	State of the current session, can be divided into

	not_started: The session has not been started. This status indicates that the session request has been accepted, but the session has not yet started for some reason (for example, insufficient resources or configuration problems). Users need to check related configurations or resource status to start the session. Starting: The session is starting. This status means Livy is allocating resources and initializing the environment for a new Spark session. idle: The session has started successfully and is in idle state. At this point, you can submit Spark jobs. The Livy session is ready to process requests. 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: The session is deactivating. This status means the user has requested to stop the session, and Livy is performing clearing and resource release operations. The session may stay in this status for a period of time until all running jobs are completed and resources are released. error: The session encounters an error during startup or execution. This status usually means the session is unable to function normally, possibly due to insufficient resources, configuration errors, or other problems. dead: The session has died and cannot be recovered. killed: The session is forcefully terminated. This status means the user has actively terminated the session, possibly because the session is no longer needed or there are problems with the ongoing jobs. A killed session cannot be recovered. success: The session has been successfully completed. This status usually indicates that all jobs in the session have been successfully executed and completed. The session can be considered successful in this status, and users can view the results or output.
Engine	Computing engine.
Kind	Session type Spark Pyspark SQL Machine Learning Python MLlib
Creator	The user who creates a session.
Validity period	The running time of the session.

Insight Management
Task Insights

Last updated：2025-04-17 15:22:36

Task insights are made from the task perspective, helping you quickly identify the completed tasks for analysis and providing optimization suggestions.

Prerequisites

1. SuperSQL SparkSQL and Spark job engines:
1. For engines purchased after July 18, 2024, task insights are enabled by default.
2. For Spark kernel versions prior to July 18, 2024, the engine kernel should be upgraded to enable task insights. For details on upgrading, see How to Enable Insights.
3. Standard Spark engine:
1. For engines purchased after December 20, 2024, task insights are supported by default.
2. For engines purchased before December 20, 2024, manual activation of task insights is not supported. Submit a ticket to contact after-sales service for activation.

Other types of engines do not support task insights currently.

Directions

Log in to the DLC Console, select the Insight Management feature, and switch to the task insights page.

Insights Overview

Daily-level statistics offer insights into the distribution and trend of tasks requiring optimization, providing a more intuitive understanding of daily tasks.

Task Insights

The task insights feature supports analyzing the summary metrics of each executed task and identifying the possible optimization issues.
After a task is completed, users only need to select the task to be analyzed and click **Task Insights** in the operation column to view the details.
Based on the actual execution of the current task, DLC task insights leverage data analysis and algorithmic rules to provide the corresponding optimization recommendations.

How to Enable the Insights Feature?

Upgrading Kernel Image for Existing SuperSQL Engines

Note：
For engines purchased after July 18, 2024, or existing engines upgraded to kernel versions after July 18, 2024, Insights are automatically enabled. You can skip this step.

Directions

1. Go to the SuperSQL Engine list page and select the engine for which you want to enable the insights feature.
2. On the engine details page, click **Kernel version management > Version upgrade** (default upgrade to the latest kernel version).

Overview of Key Insight Metrics

Metric Name	Metric Definition
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).
Execution time within the engine	Reflects the time actually required for computing, namely, the time taken from the start of the first task execution in a Spark task to the completion of the Spark task. 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 such as 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.
Queuing time (time spent	Specifies the time taken from task submission to the start execution of the first

waiting for execution)	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.
Consumed CU*H	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 approximately equals 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 the data. In the Spark scenario, it approximately equals 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	(This metric requires the Spark engine kernel to be upgraded to a version after November 16, 2024)The total number of files written by tasks through statements such as insert.
Number of output small files	(This metric requires the Spark engine kernel to be upgraded to a version after November 16, 2024)Small files are defined as output files with a size less than 4 MB (controlled by the parameter spark.dlc.monitorFileSizeThreshold, default 4 MB, configurable at the engine or task level). This metric represents the total number of small files written by tasks through statements such as insert.
Parallel task	Displays the parallel execution of tasks, making it easier to analyze affected tasks (up to 200 entries).

Overview of Insight Algorithms

Insight Type	Algorithm Description (Continuously Improving and Adding New Algorithms)
Resource preemption	SQL execution task delay time is greater than 1 minute after stage submission, or delay exceeds 20% of the total runtime (the threshold formula dynamically adjusts based on task runtime and data volume).

Shuffle exception	Stage execution encounters shuffle-related error stack information.
Slow task	Task duration in a stage is greater than twice the average duration of other tasks in the same stage (the threshold formula dynamically adjusts 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 dynamically adjusts based on task runtime and data volume).
Disk or memory insufficiency	Error stack information during stage execution includes OOM, insufficient disk space, or COS bandwidth limitation errors related to disk or memory insufficiency.
Excessive small file output	(This insights type requires the Spark engine kernel to be upgraded to a version after November 16, 2024)See the metric number of output small files in the list, and the presence of excessive small file output is determined if any of the following conditions are 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. If partitioned or non-partitioned tables output more than 3,000 files with an average file size less than 4 MB.

System Management
User and Permission Management
CAM Service

Last updated：2025-01-03 15:27:28

Data Lake Compute has a complete data access control mechanism and divides permissions into operation permissions and data permissions. The former is managed by CAM, while the latter is managed by the permission module of Data Lake Compute.
A root account has all the operation and data permissions of Data Lake Compute by default.
If a sub-user is granted the operation permissions of Data Lake Compute, the sub-user can grant the data permissions to other sub-users and can be regarded as an "admin" of this type of sub-users.
If a sub-user is granted the data read/write permissions, the sub-user can query data as permitted. The data permissions are granted by an "admin".
The data permissions of all sub-users other than root accounts are granted by an "admin". They cannot query data which they don't have permissions on.
A root account has all the operation permissions of Data Lake Compute by default and can grant sub-users the access permissions of Data Lake Compute through CAM, so that the sub-users can have corresponding operation permissions of Data Lake Compute.

Directions

1. Create and authorize a sub-user.
In the CAM console, create a sub-user and grant permissions as instructed in Sub-user authorization.
Preset policy `QcloudDLCFullAccess` : All the operation permissions in Data Lake Compute.
Custom policy: Specified operation permissions of Data Lake Compute.
2. Log in to the Data Lake Compute console with a sub-user account and verify the permissions.
If the operation succeeds, the authorization has taken effect.

Operation permission category

Data Lake Compute operation permissions are categorized by API as follows.

Permission Type	Description
Metadata	Manipulate the metadata information of databases and data tables managed in Data

management	Lake Compute.
Task management	Submit and view tasks in Data Lake Compute.
Permission management	Manage users' data access permissions.
System configuration	Perform basic configurations of the Data Lake Compute service.

Sub-user authorization

If you access Data Lake Compute as a root account, skip this step.

1. Create a sub-account as instructed in Creating and Authorizing Sub-account.

2. Create a custom policy.

On the Policies page in the CAM console, click **Create Custom Policy**.

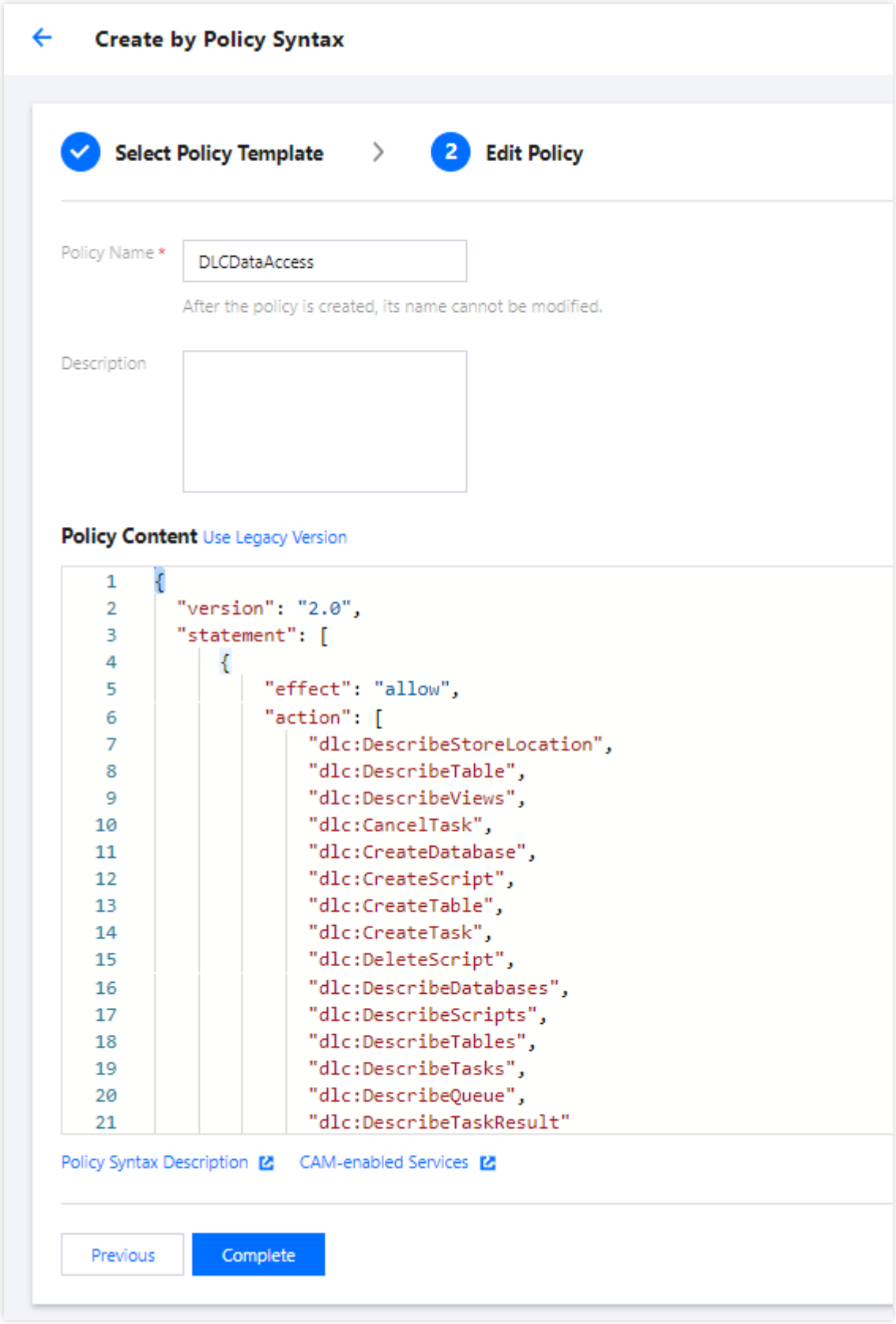
In the pop-up window, click **Create by Policy Syntax**.

On the **Create by Policy Syntax** page, select **Blank Template** and click **Next**.

In the template, enter the **Policy Name** (e.g., `DLCDataAccess`) and **Description**, copy the following policy, paste it into **Policy Content**, and click **Complete**. A sub-user bound to the custom policy can log in to the Data Lake Compute console to run SQL tasks but cannot manage data permissions. For more information, see Sub-Account Permission Management.

```
{
  "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"
          ],
```

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



3. Bind the preset or custom policy to a sub-account, and the sub-account can log in to and access Data Lake Compute. For more information, see Setting Sub-user Permissions.

Preset policy: `QcloudDLCFullAccess` .

Custom policy: The policy customized in the above steps for accessing Data Lake Compute.

Permission Overview

Last updated：2024-07-17 15:42:58

Data Lake Compute permissions include data permissions and data engine permissions. If you have the admin permission, you can log in to the Data Lake Compute console or use an API to grant a sub-user data and data engine permissions. Sub-users cannot use, modify, or delete data or data engines before they are authorized.

User and work group

Data Lake Compute provides the user mode and work group mode for personnel permission management.
User: You can select users in CAM, including sub-accounts and collaborator accounts.
Work group: It is a group of users with the same permissions managed in the product.
Note:
If users are granted different permissions from those granted in their work groups, all the granted permissions will take effect.
A work group allows you to quickly grant permissions to a batch of users, so it is recommended for batch user authorization. For detailed directions, see User and User Group.

User type

In Data Lake Compute, **User type** can be **Admin** or **General user**.
Admin: An admin have all the data, engine, and task permissions and can add, authorize, and remove users and work groups in Data Lake Compute.
General user: A general user is added by an admin, has no Data Lake Compute permissions by default, and needs to be authorized. Only data and engine permissions that can be **regranted** can be granted to general users.

Permission and Operation	Admin	General User
Data permissions	All	None by default (to be authorized by an admin)
Data engine permissions	All	None by default (to be authorized by an admin)
User management	Yes	No
Work group management	Yes	No
Authorization scope	All	Permissions that **can be regranted**

Note:
The above permissions only include those defined in Data Lake Compute. To perform purchase, configuration adjustment, and refund operations that involve billing, log in to the CAM console and get the financial collaborator permission `QCloudFinanceFullAccess` (for detailed directions, see Creating and Authorizing Sub-account).

Data permissions

Data Lake Compute data permissions allow operations on data catalogs, databases, and data tables. To facilitate your management and configuration, permissions can be granted in the standard or advanced mode.
In standard mode, you can grant roles while ignoring the specific permission configuration (for more information on roles and permissions, see Sub-Account Permission Management). The authorization granularity can be data catalog, database, or data table. This mode is suitable for quick authorization with no complex permission management involved.
In advanced mode, you can grant permissions at the database, data table, view, or function level. It is suitable for refined permission management.
SQL statements for permission operations are as follows:

Action	CREATE	ALTER	DROP	SELECT	INSERT	DELETE	Target
CREATE DATABASE	✓	-	-	-	-	-	Cataglog
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

Data engine permissions

Data Lake Compute data engine permissions allow using, modifying, manipulating, monitoring, and deleting data engines as detailed below:

Use: The permission to use engines to perform tasks.
Modify: The permission to modify the basic information and configuration information of engines (modifying the configuration information requires the CAM financial collaborator permission).
Manipulate: The permission to suspend and restart engines.
Monitor: The permission to view the running tasks and monitoring information of engines.
Delete: The permission to return engines.

Permission granting

A single user can be granted multiple permissions. For detailed directions, see Sub-Account Permission Management.

User and Work Group

Last updated：2024-07-17 15:44:57

Data Lake Compute provides the user mode and work group mode for personnel permission management. For more information on permissions, see Permission Overview.

Description

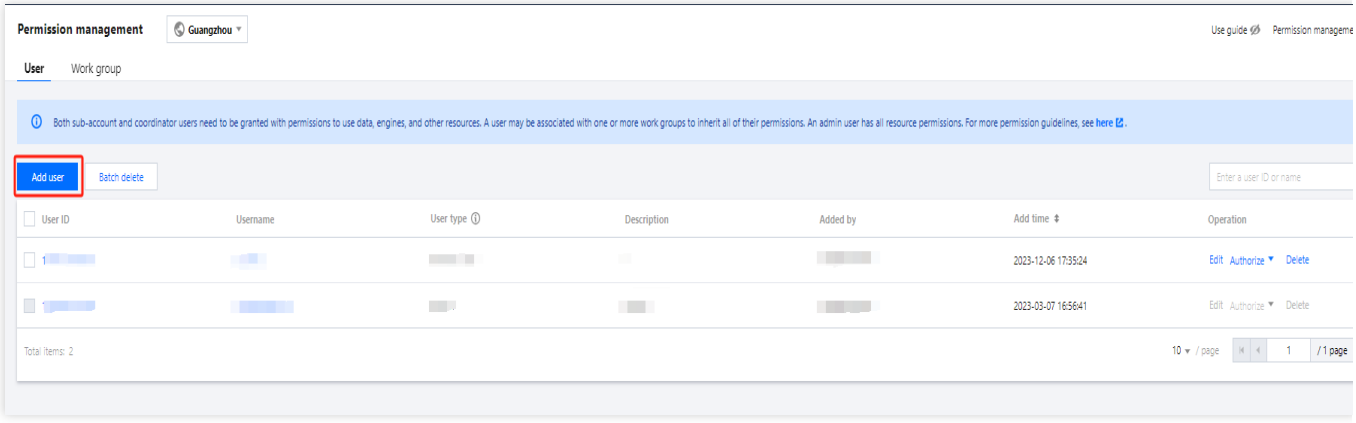
User: You can select users in CAM, including sub-accounts and collaborator accounts.
Work group: It is a group of users with the same permissions managed in the product.
Note:
If users are granted different permissions from those granted in their work groups, all the granted permissions will take effect.
A work group allows you to quickly grant permissions to a batch of users, so it is recommended for batch user authorization.

User Management

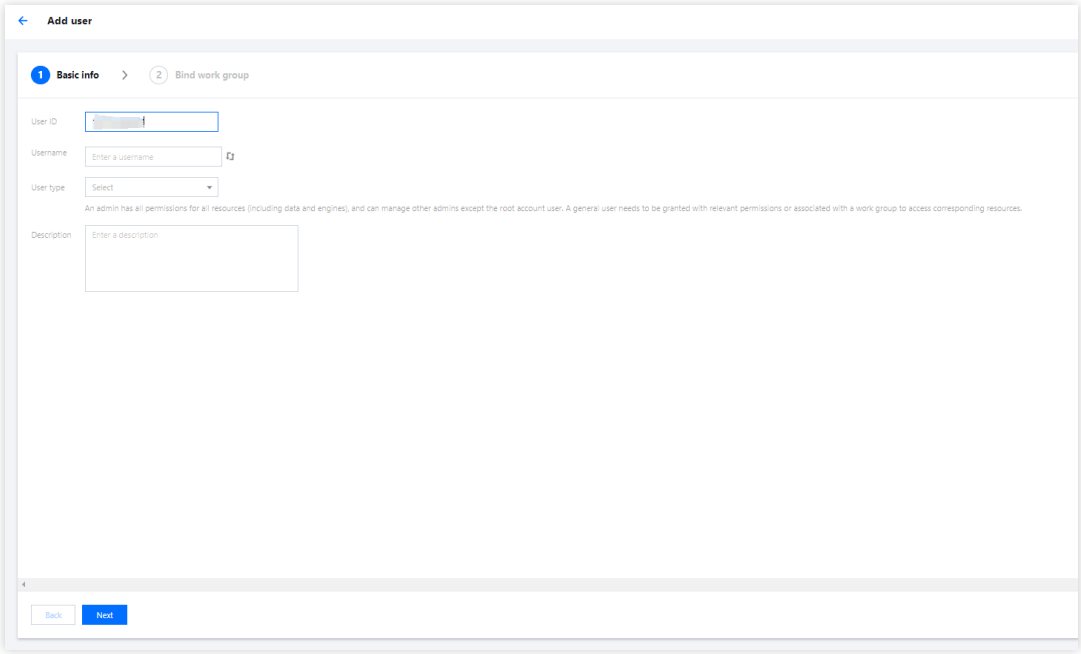
User management requires Data Lake Compute operation permissions. For more information, see CAM Service.

Adding a user

1. Log in to the Data Lake Compute console, select the service region, and go to the **Permission management** page.
2. Click **Add user** to add an account with a specified user ID to Data Lake Compute for management.



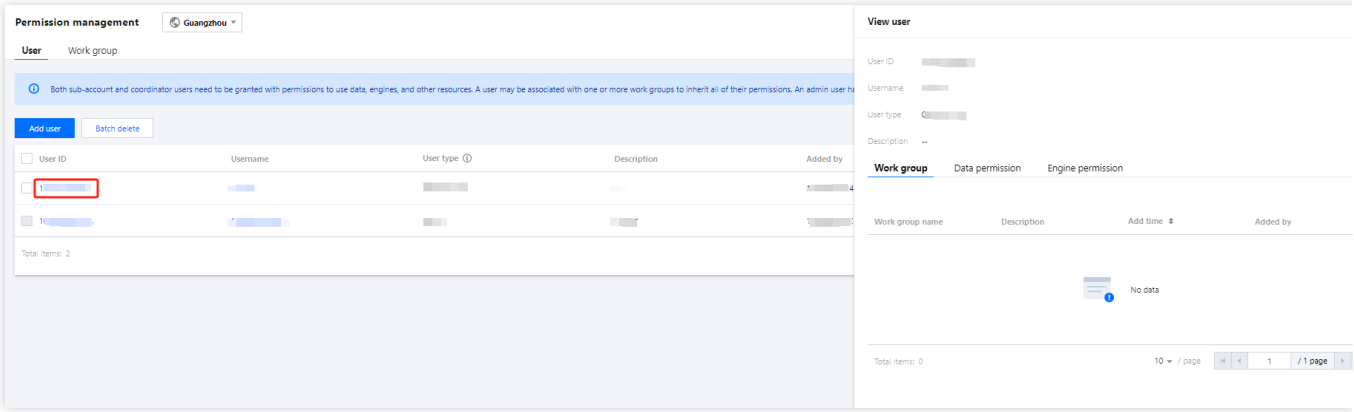
3. After entering the **User ID**, bind the user to a work group (which requires the admin permission). If binding is not needed, directly click **Complete**.



Viewing user information

A Data Lake Compute admin can modify the basic information and permissions of a user.

1. Log in to the Data Lake Compute console, select the service region, and go to the **Permission management** page.
2. Search for the target **User ID** and click the **Username** to view the user information and permissions.



Editing user information

You can edit the description and work group of a user. For detailed directions, see Sub-Account Data Authorization.

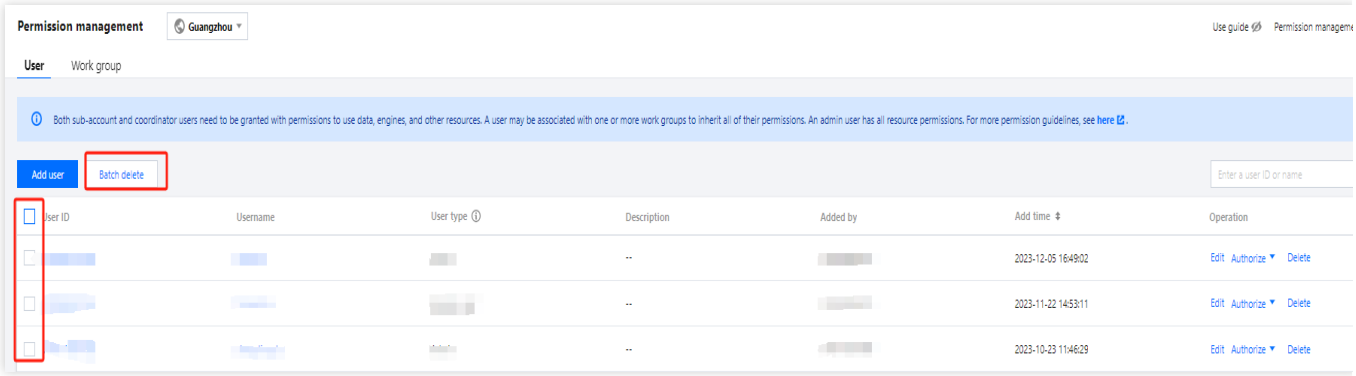
1. Log in to the Data Lake Compute console, select the service region, and go to the **Permission management** page.

2. Search for the target user account ID and click **Edit** in the **Operation** column to enter the edit page.

Removing a user

If you don't want a user to use Data Lake Compute any more, you can use an admin account to remove the user. Then, the Data Lake Compute permission granted to the user will be revoked.

1. Log in to the Data Lake Compute console, select the service region, and go to the **Permission management** page.
2. Search for and select one or multiple target user account IDs and click **Batch remove** to remove them from Data Lake Compute.



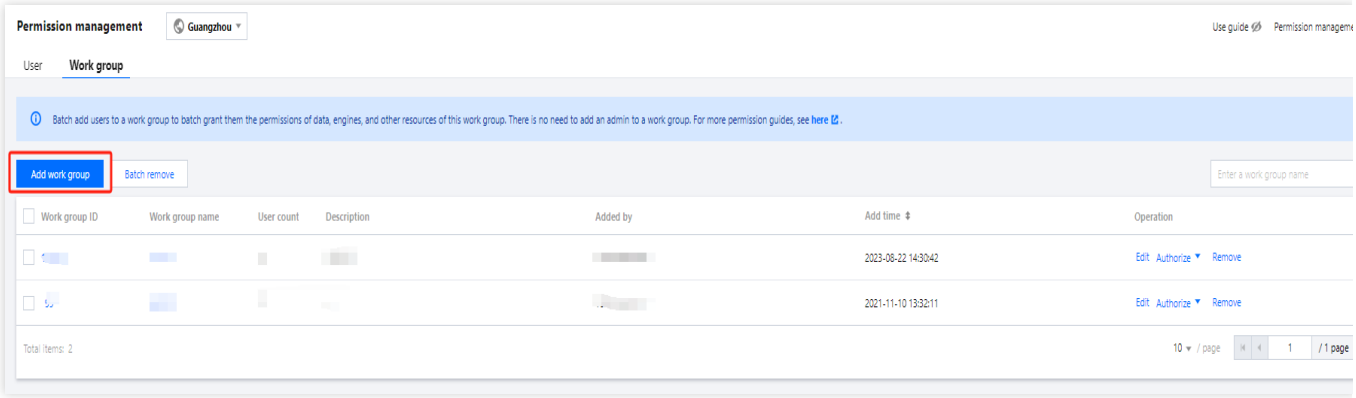
Work Group Management

Work group management requires Data Lake Compute operation permissions. For more information, see CAM Service.

Adding a work group

You can manage permissions that need to be repeatedly granted to users through a work group. The following describes how to add a work group.

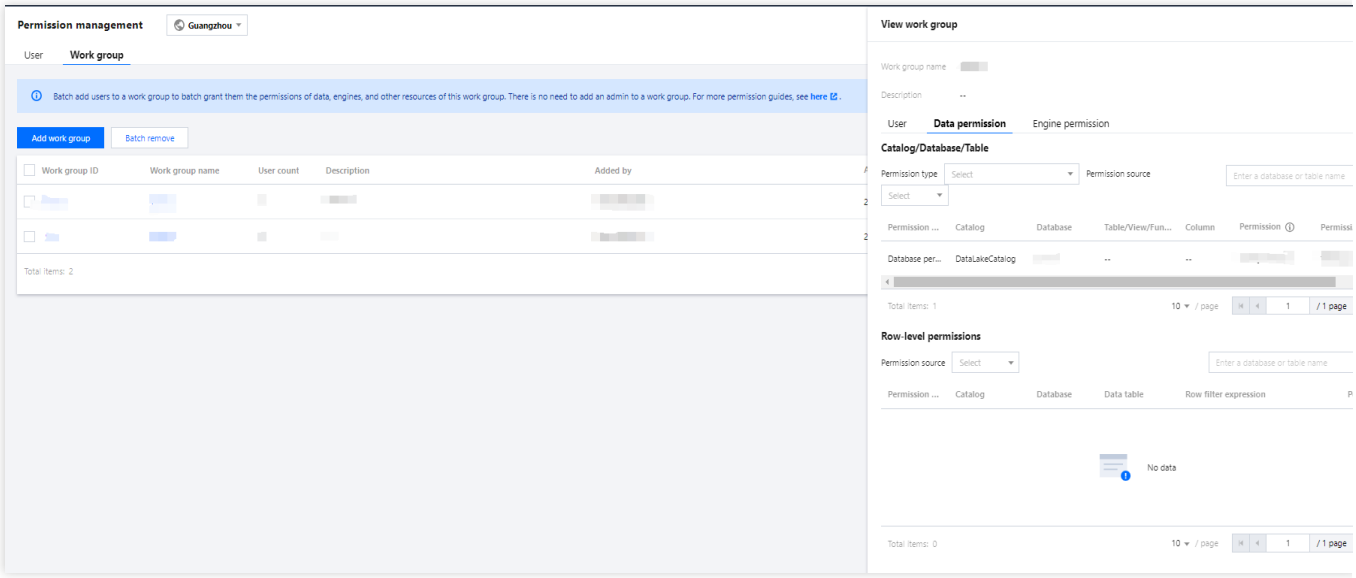
1. Log in to the Data Lake Compute console, select the service region, and go to the **Permission management** page.
2. Click **Work group** to enter the work group management page.
3. Click **Add work group**, enter relevant information, and click **Confirm**.



Viewing work group information

You can view the information of a work group in the following steps:

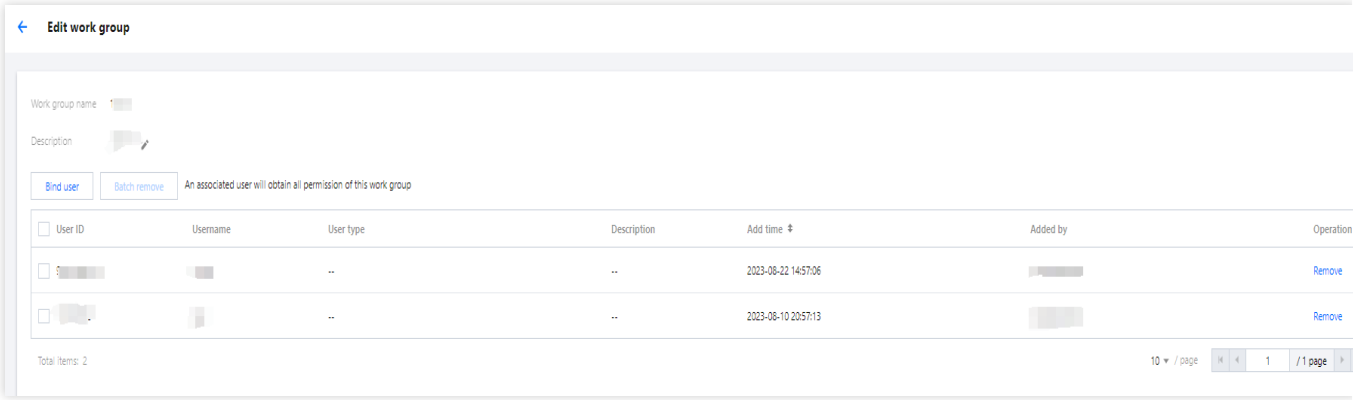
1. Log in to the Data Lake Compute console, select the service region, and go to the **Permission management** page.
2. Click **Work group** to enter the work group management page.
3. Search for the target work group and click **Work group ID** or **Work group name** to view the work group information.



Editing work group information

You can modify the description and users of a work group in the following steps:

1. Log in to the Data Lake Compute console, select the service region, and go to the **Permission management** page.
2. Click **Work group** to enter the work group management page.
3. Find the target **Work group name** and click **Edit** in the **Operation** column.



To edit the description, click

.

You can click **Bind user** to add Data Lake Compute users to the work group.

Select multiple target users and click **Batch remove**, or click **Remove** in the **Operation** column of a specific target user. Removed users will no longer have the permissions of the work group, which does not affect other permissions granted to them though.

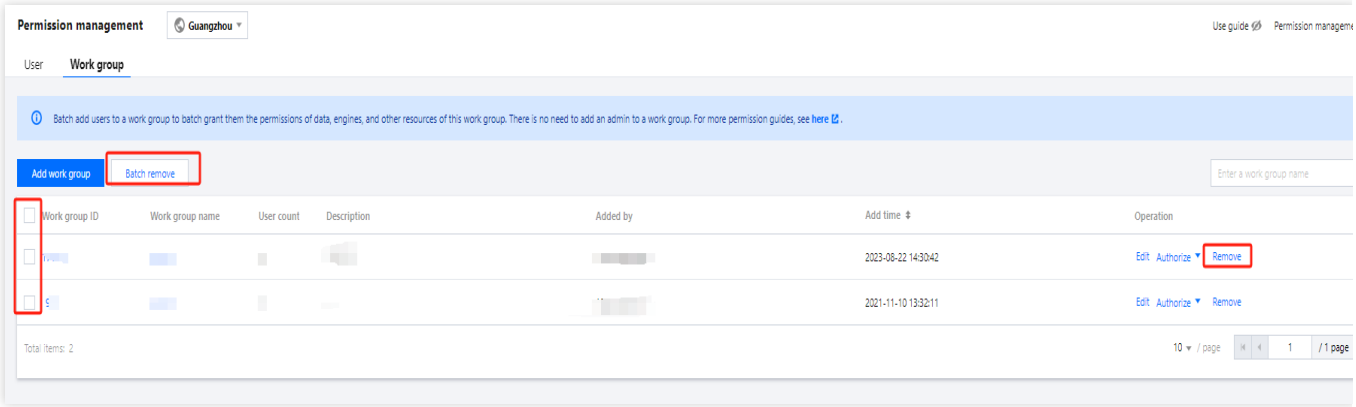
Deleting a work group

A Data Lake Compute admin can remove work groups.

Note:

After a work group is removed, all its permissions granted to users in it will be revoked. Note that a removed work group cannot be recovered. Proceed with caution.

1. Log in to the Data Lake Compute console, select the service region, and go to the **Permission management** page.
2. Click **Work group** to enter the work group management page.
3. Select multiple target work groups and click **Batch remove**, or click **Remove** in the **Operation** column of a specific target work group.


Permission management
Guangzhou
Use guide
User
Work group
Batch add users to a work group to batch grant them the permissions of data, engines, and other resources of this work group. There is no need to add an admin to a work group. For more permission guides, see here .
Add work group
Batch remove
Enter a work group name
Work group ID
Work group name
User count
Description
Added by
Add time
Operation
2023-08-22 14:30:42
Edit
Authorize
Remove
2021-11-10 13:32:11
Edit
Authorize
Remove
Total items: 2
10 / page
1
/ 1 page

Sub-Account Permission Management

Last updated：2024-07-17 15:46:12

User permission

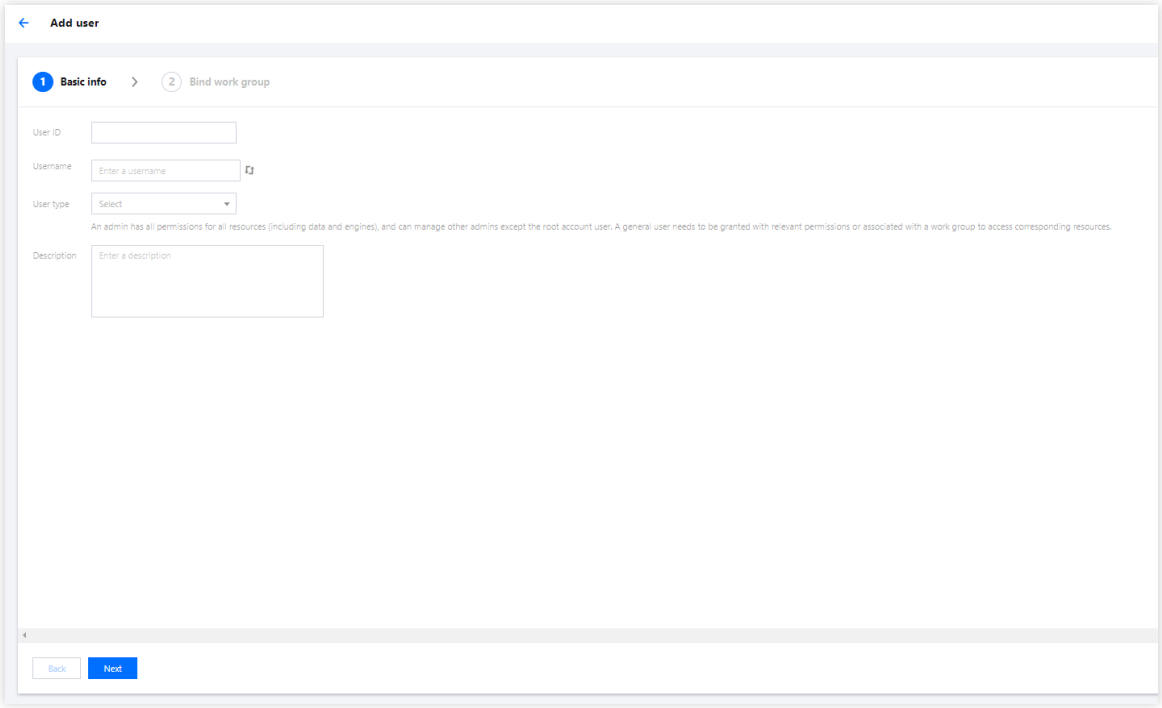
User permissions include data permissions and engine permissions (for more information on permissions, see Permission Overview). The former is required to access data in Data Lake Compute, while the latter is used for resource management. Data Lake Compute enables permission management at the database, table, and column levels, so that you can authorize a user or work group for refined data permission management in different use cases.

User and work group

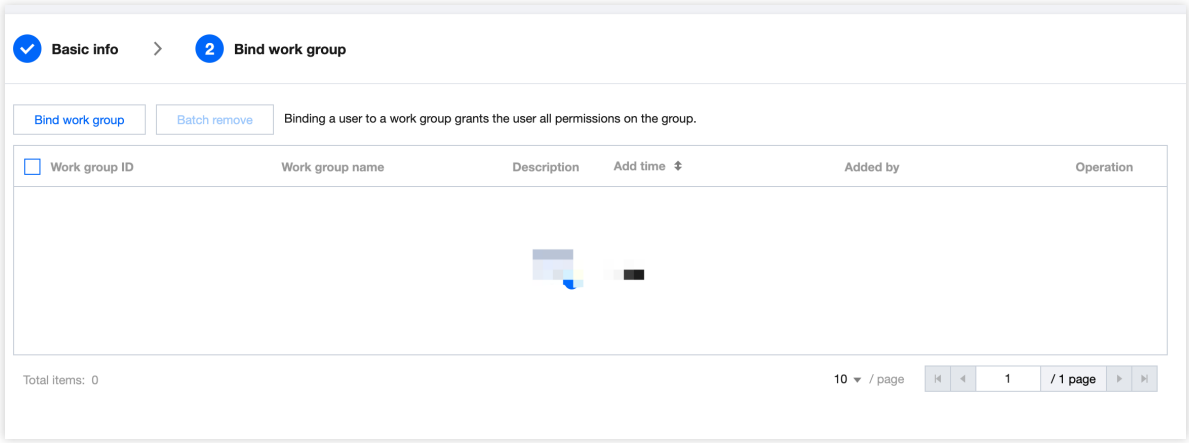
You can authorize a user or create and authorize a work group of users. For detailed directions, see User and Work Group.
User: You can select users in CAM, including sub-accounts and collaborator accounts.
Work group: It is a group of users with the same permissions managed in the product.
Note:
If users are granted different permissions from those granted in their work groups, all the granted permissions will take effect.
A work group allows you to quickly grant permissions to a batch of users, so it is recommended for batch user authorization.

Granting a user a permission

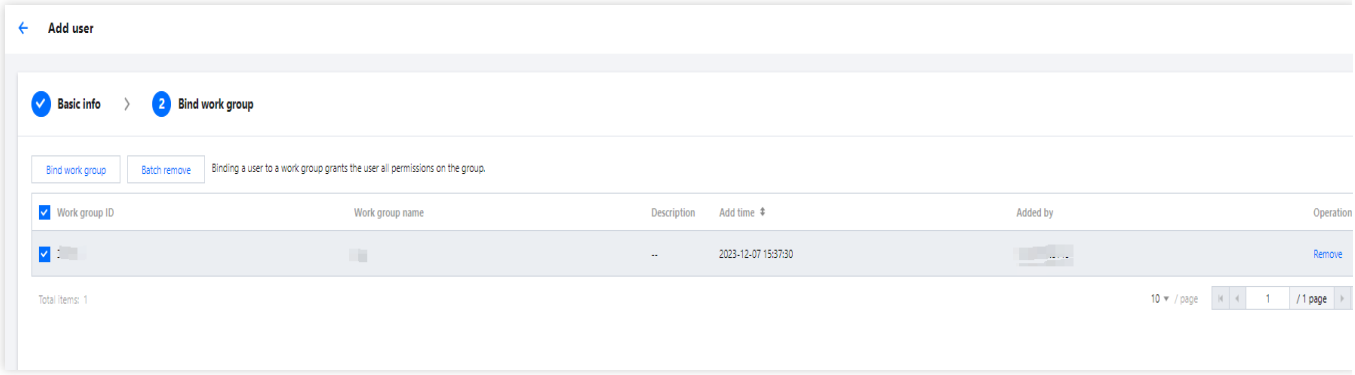
Grant permissions to the specified user.
1. Set a user to **Admin** or **General user**. Admins have the permissions of all the data and engines by default with no need to be bound to a work group. They can also manage admin users other than the root account. **Set an admin with caution.**



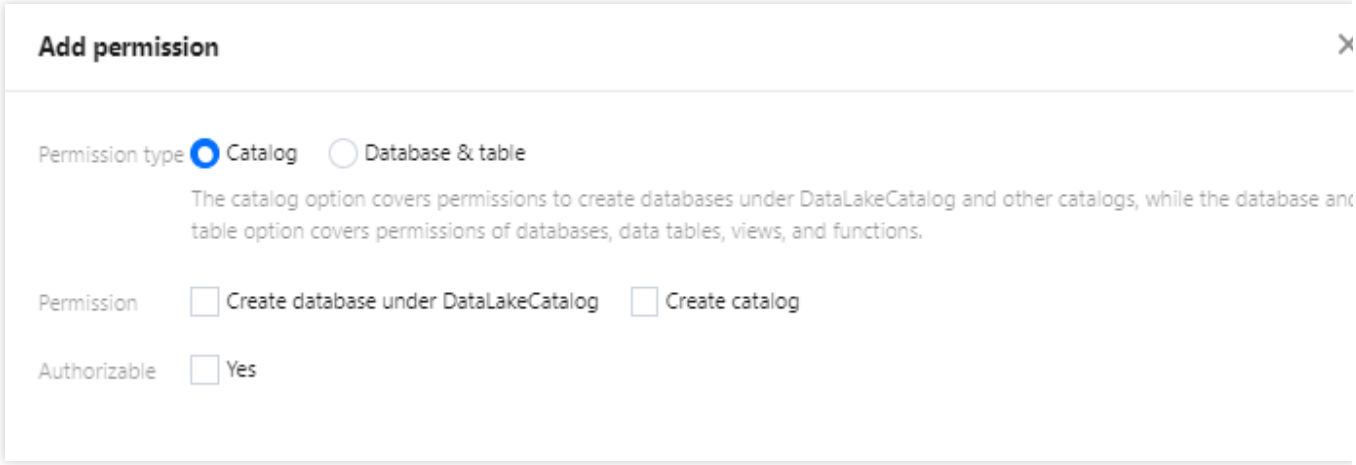
2. Bind a work group: General users need to be granted permissions or bound to a work group before they can access resources.



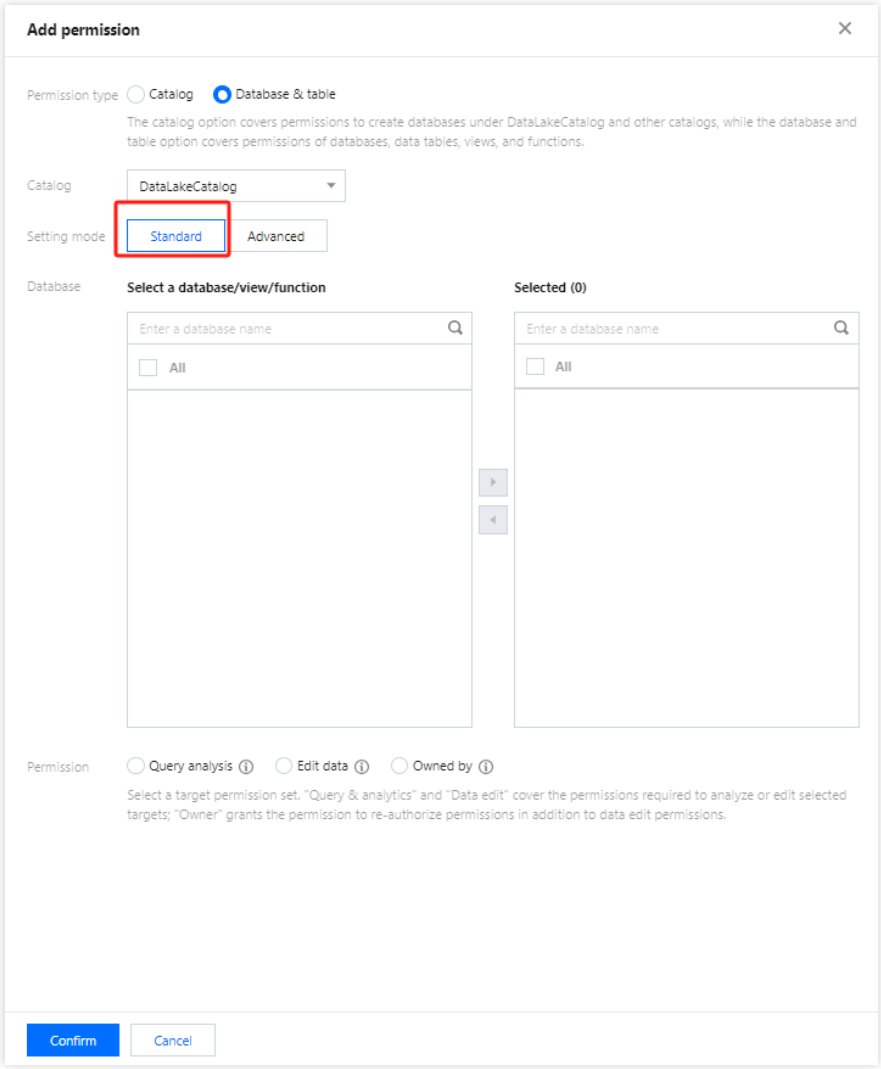
3. Add a data permission: In the **User list**, click **Authorize** in the **Operation** column and select **Data permission** to grant permissions at the data catalog or database/table level.



Add a data catalog permission. You can grant permissions to create databases under DataLakeCatalog and create other data catalogs.



Add a database/table permission: You can grant permissions in **Standard** or **Advanced** mode. In standard mode, you can grant database/table permissions in the specified catalog and set **Query & analytics**, **Data edit**, and **Owner** permissions.

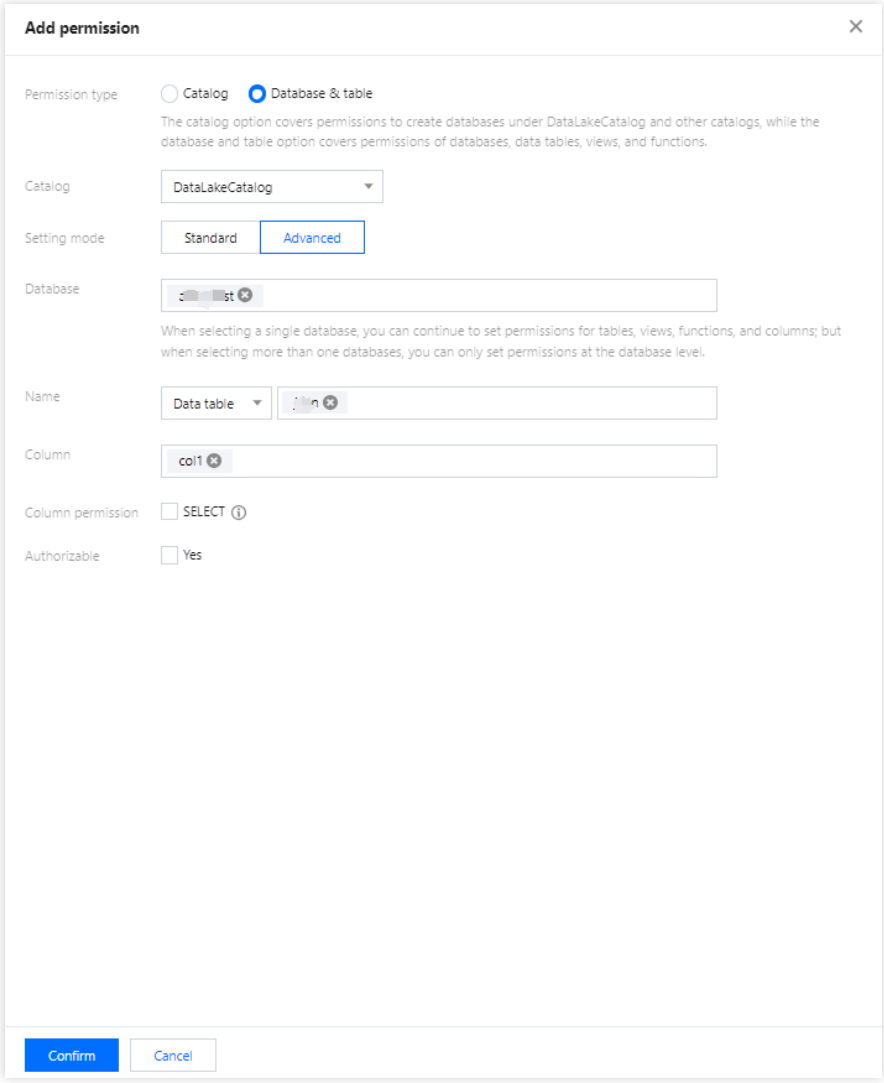


Specific permissions are as follows:

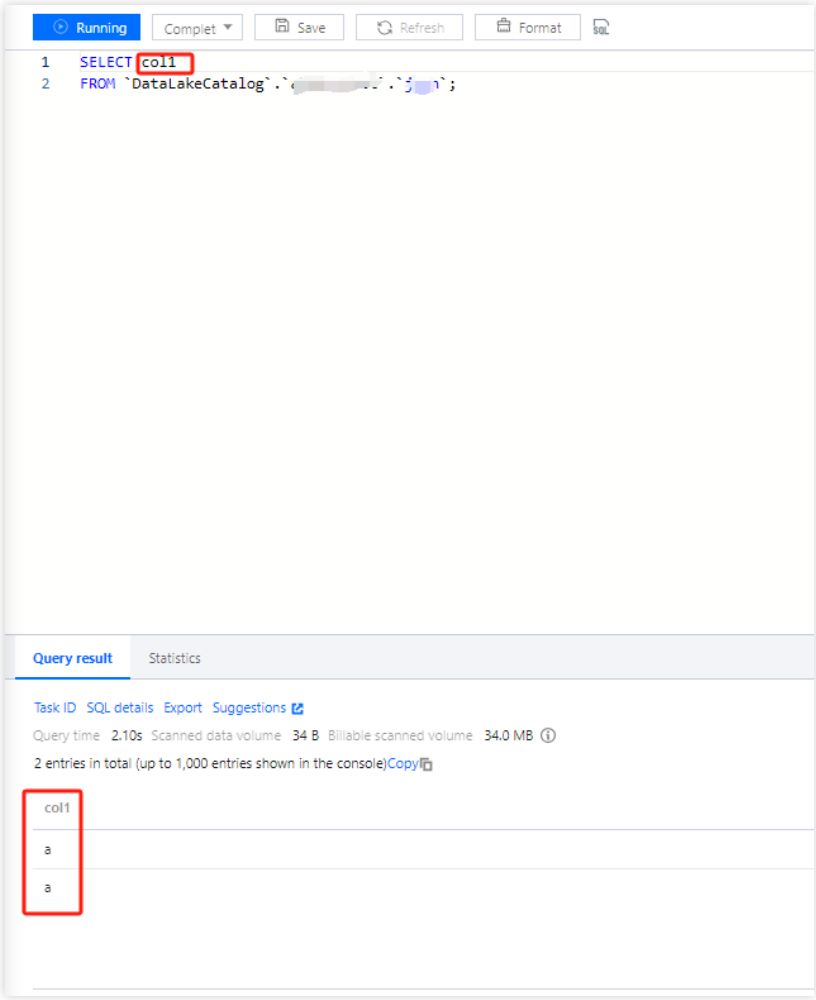
Permission Type	Database	Data Table	View and Function
Query & analytics	• Query all the tables, views, and functions in databases.• Create data tables.	Query	Query
Data edit	• Modify and delete databases and create tables.• Permissions of all the tables, views, and functions.	• Query, insert, update, and delete data.• Modify and delete tables.	Query, create, modify, and delete.
Owner (grants the permission to re-authorize permissions in addition to data edit permissions)	• Modify and delete databases and create tables.• Permissions of all the tables, views, and functions.	• Query, insert, update, and delete data.• Modify and delete tables.	Query, create, modify,

			and delete.

Advanced permission settings: When selecting a single database, you can further set the permissions to query, insert, update, and delete tables, views, and functions; when selecting multiple databases, you can only set permissions at the database level.
In advanced mode, you can set permissions at the column level. When selecting a single data table, you can add the permission to query columns. You can select one or more columns or all of them for authorization.

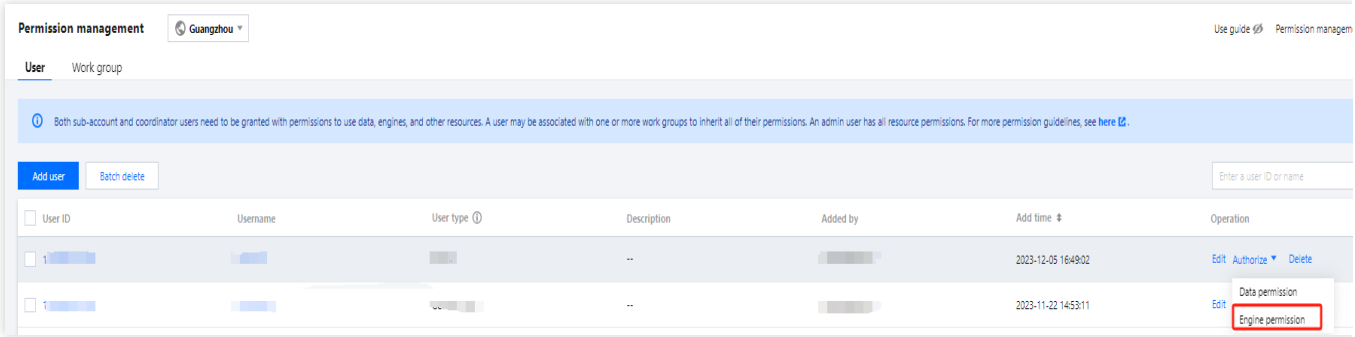


Click **Confirm** and perform queries in the **Data Explore** module. Enter the following SQL statement to preview the information of **col1** and run the statement to view the preview result of the column.



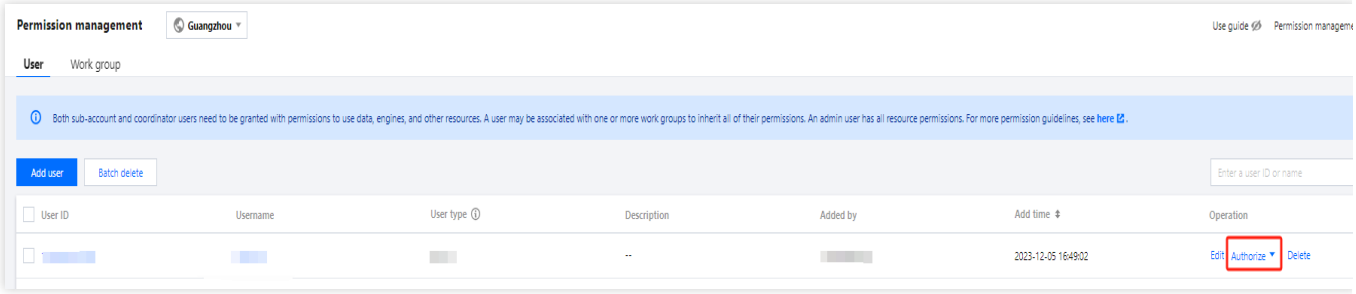
The permission is not granted for data column **b** in the data table. If you enter the SQL statement to view the information of **b**, the query cannot be performed due to lack of permission.

4. Add an engine permission: In the **User list**, click **Authorize** in the **Operation** column and select **Engine permission** to grant permissions to use, modify, manipulate, monitor, and delete specified resources.

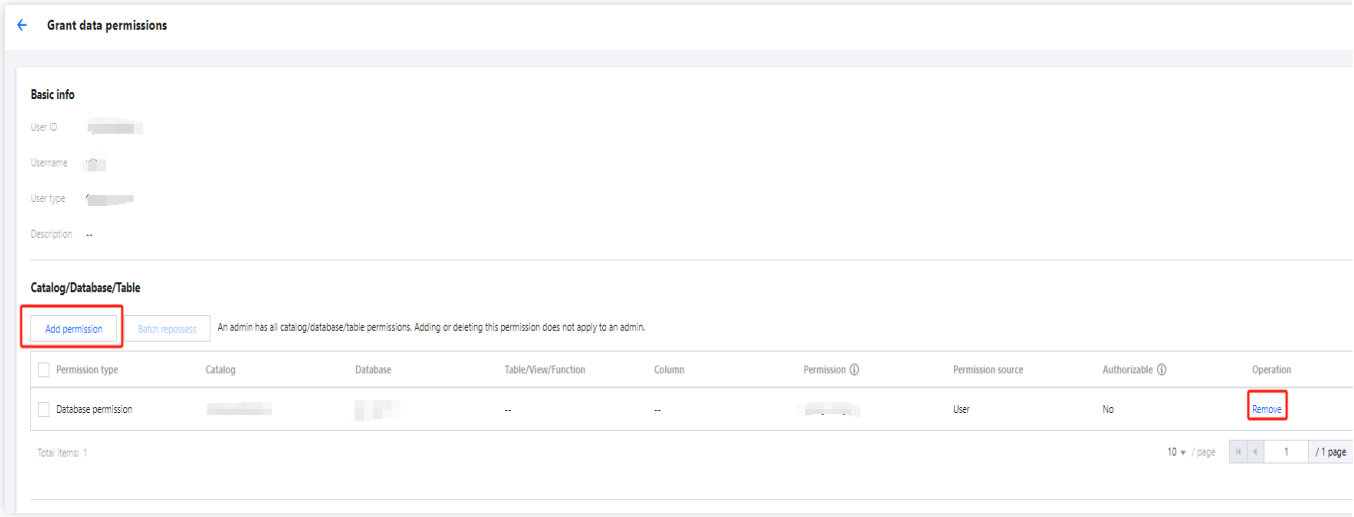


Modifying a user permission

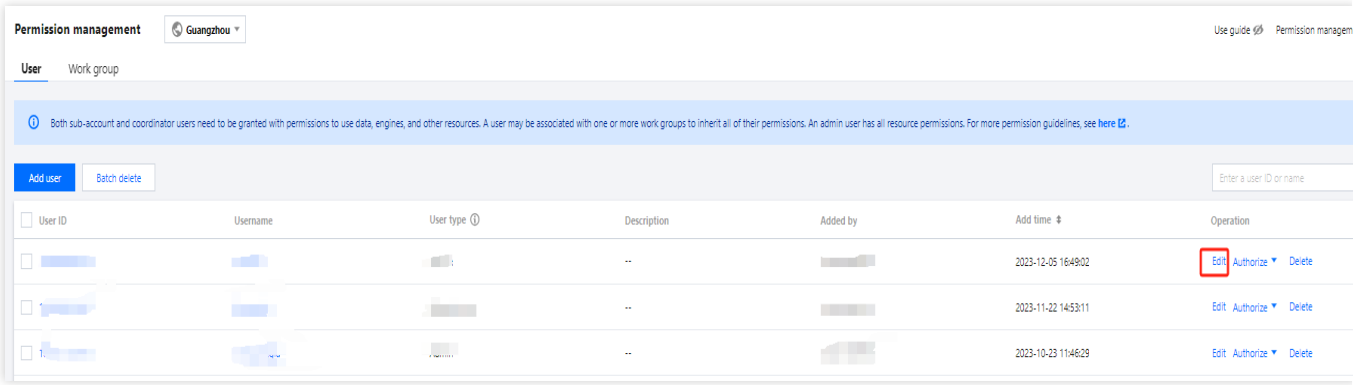
1. In the **User list**, click **Authorize** and select **Data permission** or **Engine permission**.



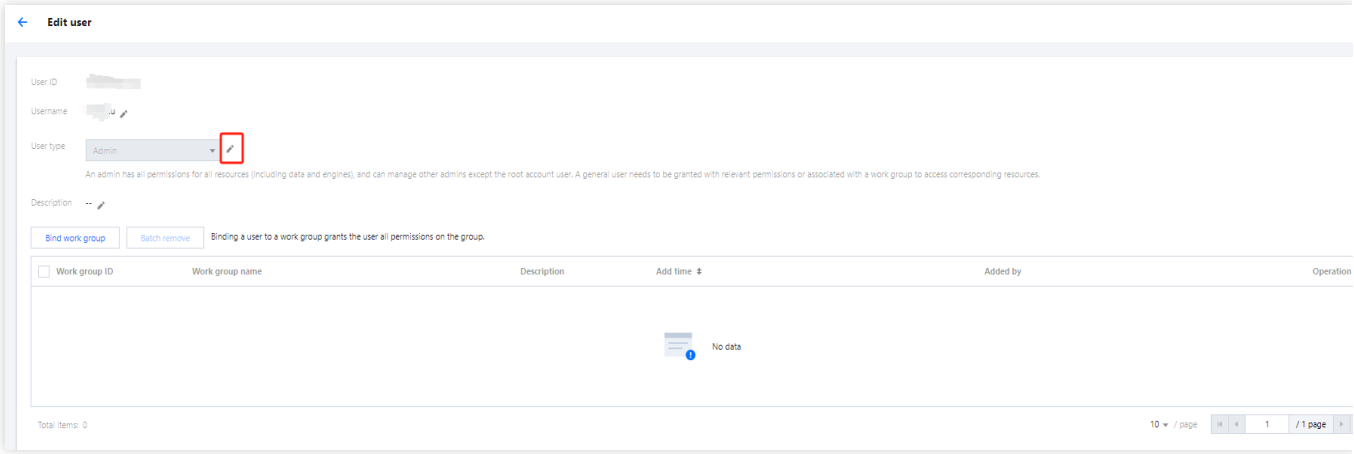
The following takes data permission as an example. On the **Data permission authorization** page, click **Add permission** or **Remove** to modify a permission. The steps for engine permission modification are similar.



2. Modify **Work group** or **User type**. Click **Operation** > **Edit** to enter the **Edit user** page, where you can modify the **Username**, **User type**, and **Description**. You can also add/remove general users to/from a work group.



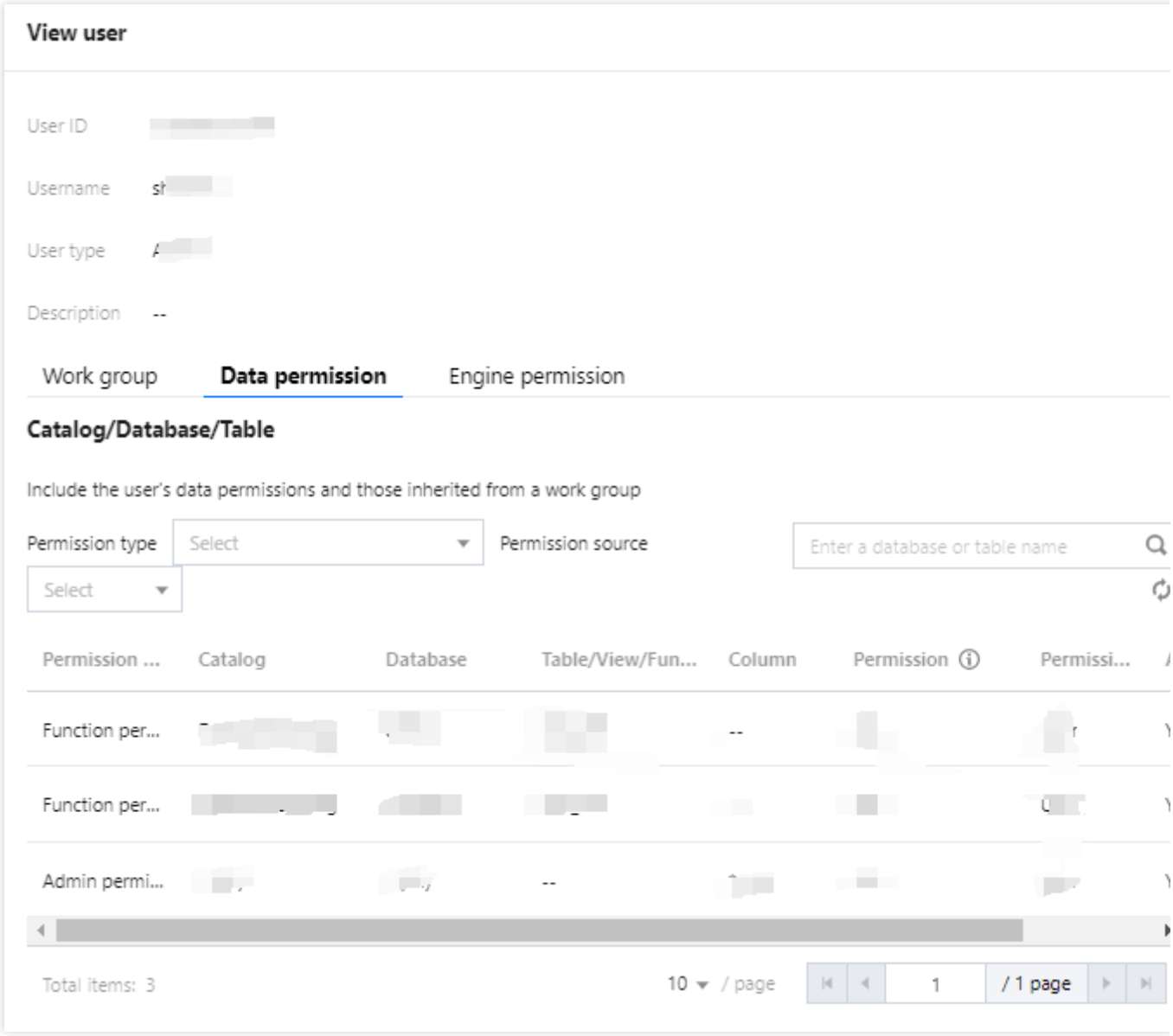
Click **Edit** to modify **User type**.



Viewing a user's permissions

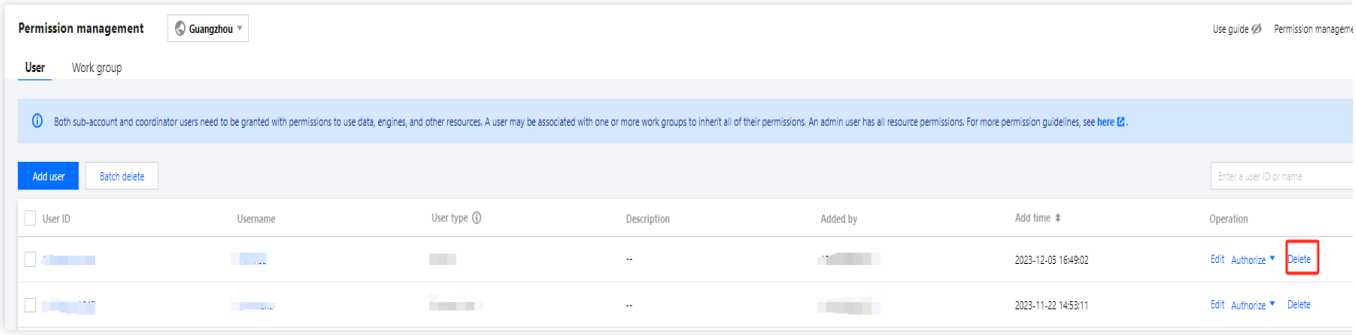
1. Click a user ID in the user list to enter the user details page.

2. View the user's work group, data permission, and engine permission information



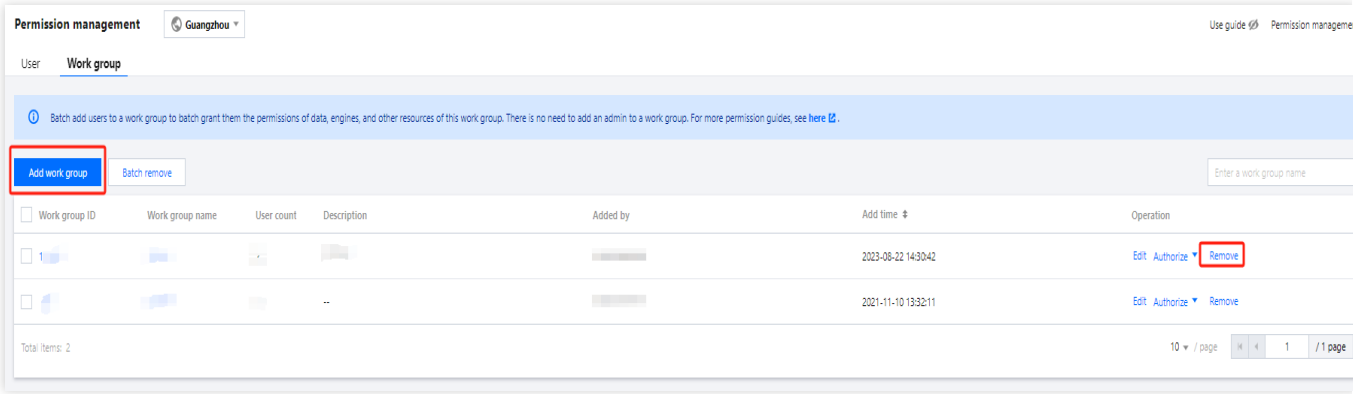
Revoking a user's permissions

Remove permissions to be revoked from the permission list of a user. This operation requires the admin permission.



Adding and removing a work group permission

Only admins can add or remove work group permissions in a similar way to manipulate data permissions. Users in a work group have all the permissions of the group, so you can bind users to a work group to grant them the data and engine permissions of the work group. Admins don't need to be bound to a work group.



Monitoring and Alarms
Data Engine Monitoring

Last updated：2024-07-31 17:31:18

Data Lake Compute (DLC) provides monitoring services for data engines based on the Tencent Cloud Observability Platform (TCOP), ensuring you can understand the real-time status of data engines and configure data alarms. For alarm configuration methods, see Monitoring Alarm Configuration.

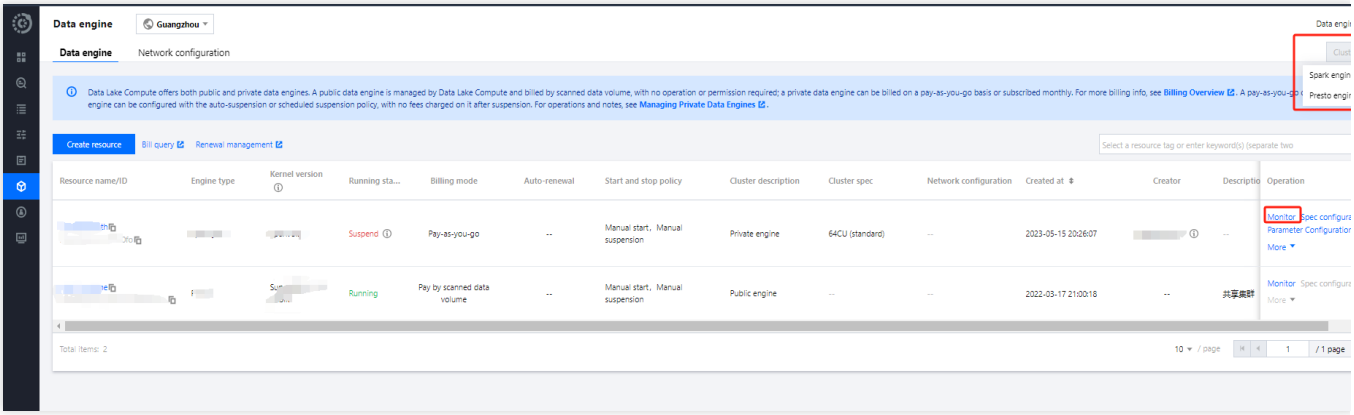
Usage Notice

Before using the Data Lake Compute (DLC) monitoring service, you need to activate the TCOP service. If this service is not yet activated, you can use the root account to activate it.
The use of the TCOP service may incur related charges. For detailed pricing information, see Billing Overview.

Monitoring Access

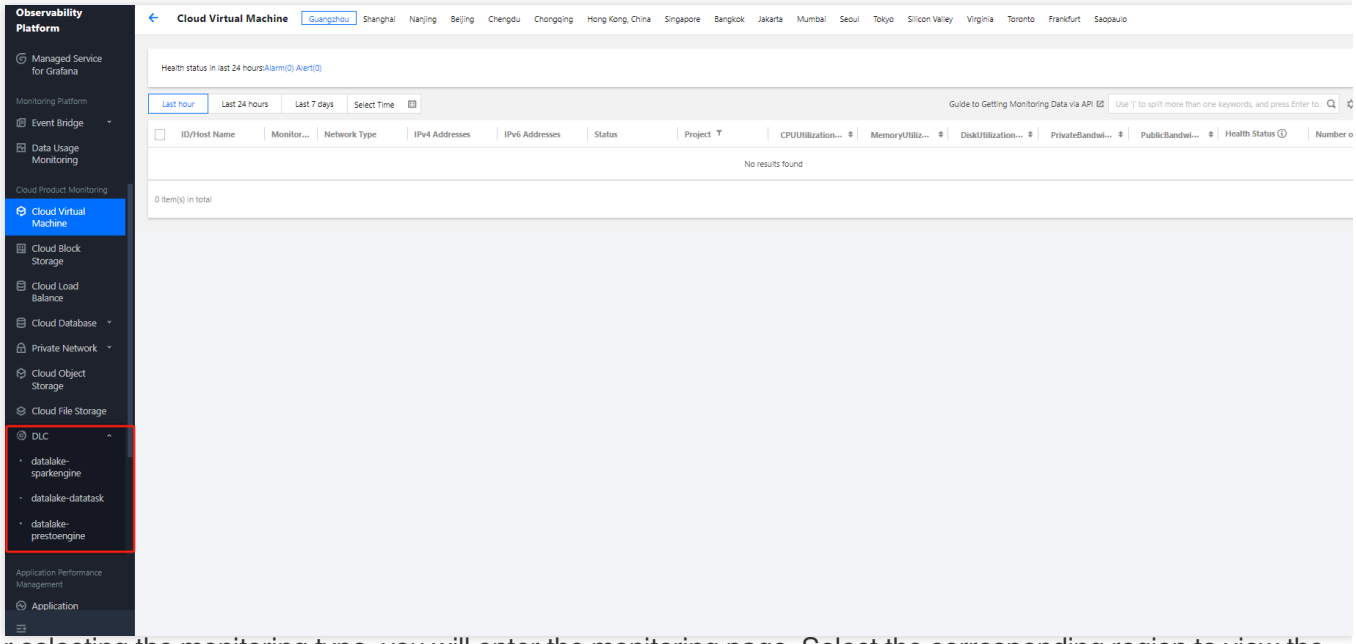
Access Point I: Data Lake Compute (DLC) Console

Note:
The account must have monitoring permissions for the data engine.
1. Log in to the DLC console and select the service region.
2. Navigate to the **SuperSQL engine** page from the left menu.
3. Viewing methods supported:
Method 1: Select the engine type to enter the matching engine monitoring list.
Method 2: Select the target engine from the engine list and click **Monitoring** to view the target engine monitoring.

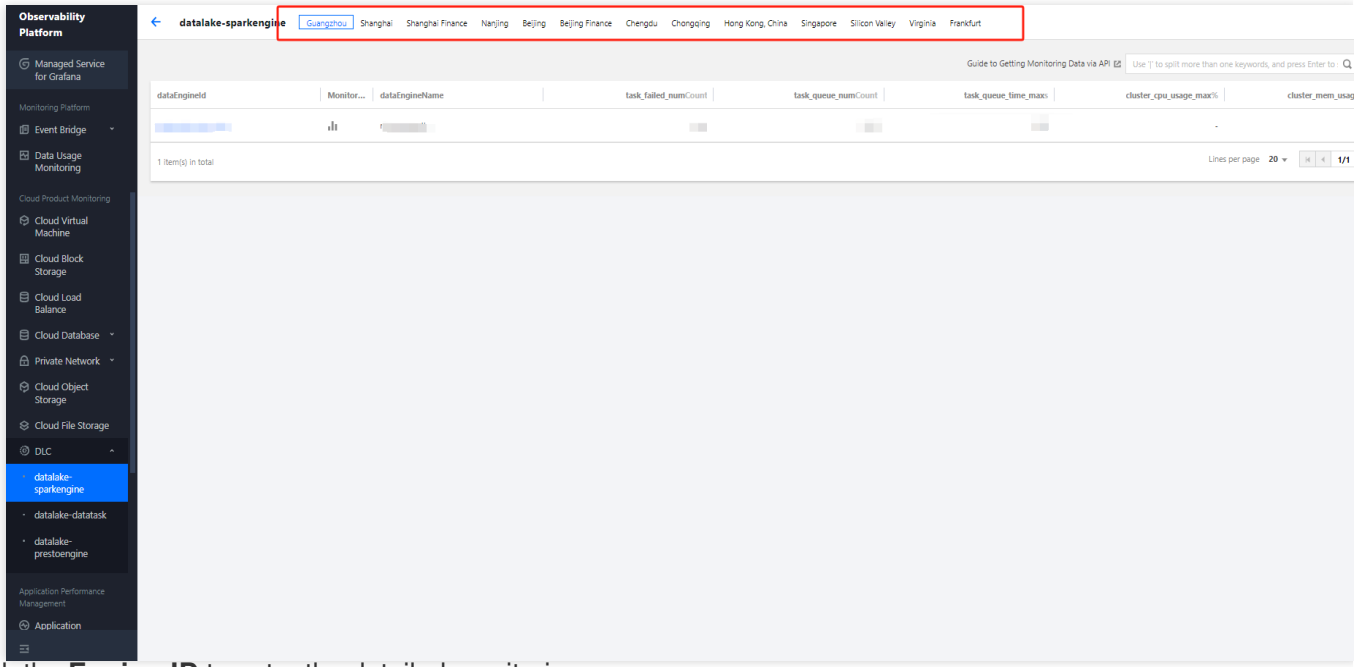


Access Point Two: TCOP

1. Log in to the TCOP with an account that has the necessary permissions.
2. Select **Cloud Product Monitoring** from the left menu, find Data Lake Compute DLC, and choose the type of monitoring you need to view.



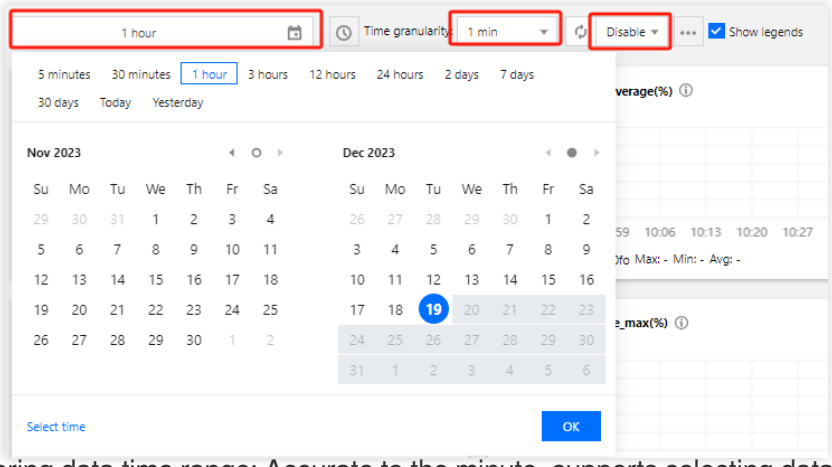
3. After selecting the monitoring type, you will enter the monitoring page. Select the corresponding region to view the monitoring resource information for that region.



4. Click the **Engine ID** to enter the detailed monitoring page.

Monitoring Granularity Configuration

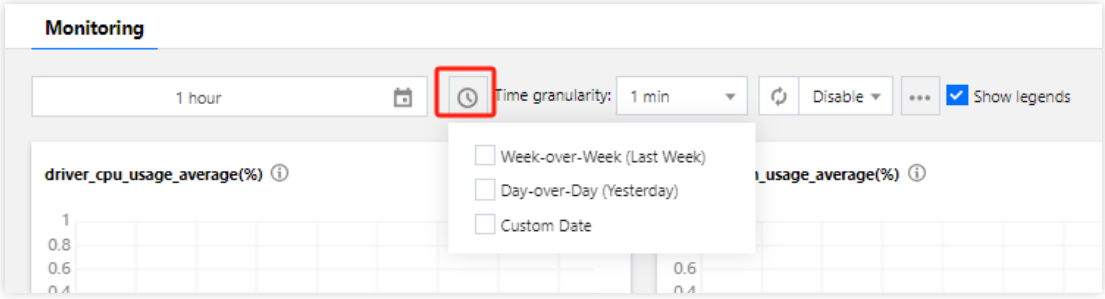
You can configure the monitoring data time range, time granularity, and auto-update interval at the top of the monitoring page.



Monitoring data time range: Accurate to the minute, supports selecting data for a specific time period.
Time granularity: Interval between monitoring points, configurable to 1 minute or 5 minutes.
Auto-update data: Configures the automatic refresh interval for page data, with options to set it to off, 30 seconds, 5 minutes, 30 minutes, or 1 hour.

Monitoring Data Comparison

You can select a time period for data comparison. After selecting the comparison time range through one click, you can view the comparison data in the data compass below.



Monitoring Metrics

Monitoring Type	Monitoring Metrics
CPU	Maximum CPU utilization of all Driver nodes
	Maximum CPU utilization of all Executor nodes
	Average CPU utilization of all Driver nodes

	Average CPU utilization of all Executor nodes
	Maximum CPU utilization of all clusters
	Average CPU utilization of all clusters
Memory	Maximum memory utilization of all Driver nodes
	Maximum memory utilization of all Executor nodes
	Average memory utilization of all Driver nodes
	Average memory utilization of all Executor nodes
	Maximum memory utilization of all clusters
	Average memory utilization of all clusters
Tasks	Number of canceled tasks
	Number of failed tasks
	Number of initialized tasks
	Average task initialization time
	Maximum task initialization time
	Number of queued tasks
	Average task queue time
	Maximum task queue time
	Number of running tasks
	Number of successful tasks
Network	Maximum inbound bandwidth of all Driver nodes network
	Maximum inbound bandwidth of all Executor nodes network
	Average inbound bandwidth of all Driver nodes network
	Average inbound bandwidth of all Executor nodes network
	Maximum outbound bandwidth of all Driver nodes network
	Maximum outbound bandwidth of all Executor nodes network

	Average outbound bandwidth of all Driver nodes network
	Average outbound bandwidth of all Executor nodes network
Cloud Disk	Maximum cloud disk utilization of all Driver nodes
	Maximum cloud disk utilization of all Executor nodes
	Average cloud disk utilization of all Driver nodes
	Average cloud disk utilization of all Executor nodes
CU	Job Engine CU Count
	CU Utilization

Data Job Monitoring

Last updated：2024-07-31 17:31:39

DLC provides monitoring services for data jobs based on TCOP service, ensuring that you can understand the operation of data jobs in real time and configure data alarms.

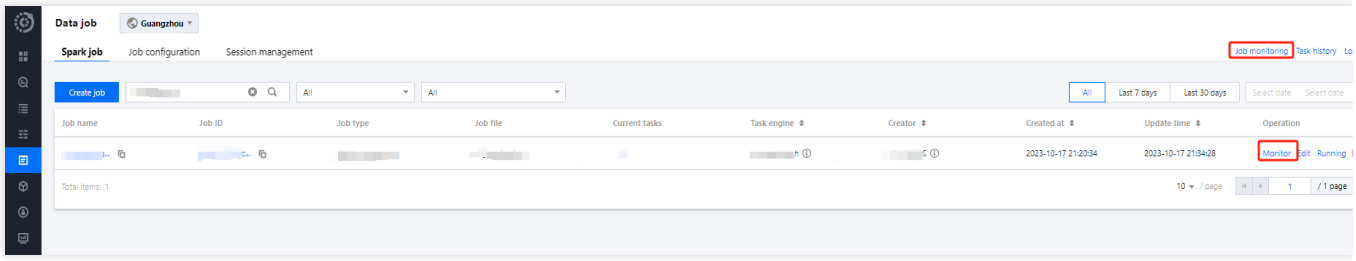
Notes

Before using the monitoring service of DLC, you need to activate the TCOP service (for usage details, refer to TCOP Documentation). If the service has not been activated, it can be done using the root account.
Fees may be incurred during the use of TCOP service; for detailed fee information, refer to TCOP Billing Overview.

Monitoring Entrance

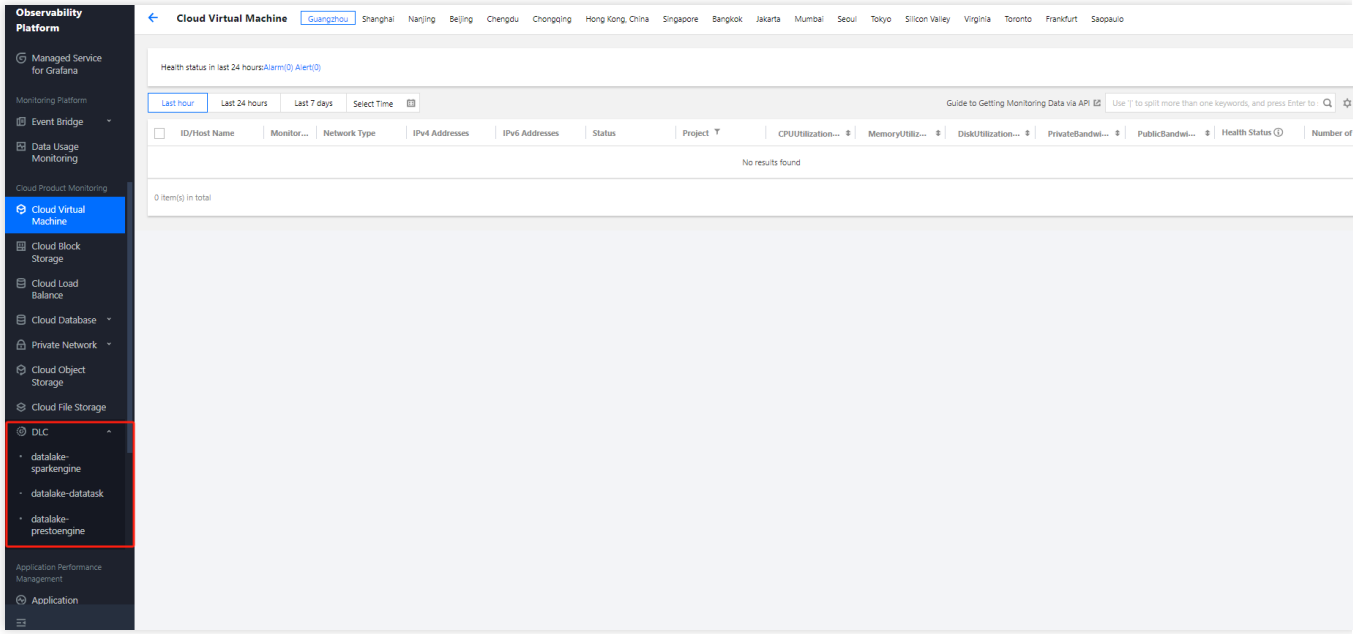
Entrance one: DLC Console

1. Log in to DLC Console > Data Job, and select the service region.
2. Or enter the Data Job page from the left sidebar.
3. In the top right corner, click **Job Monitoring** to go to the monitoring page. Or click the **Monitoring** feature of the target job to enter its monitoring page.

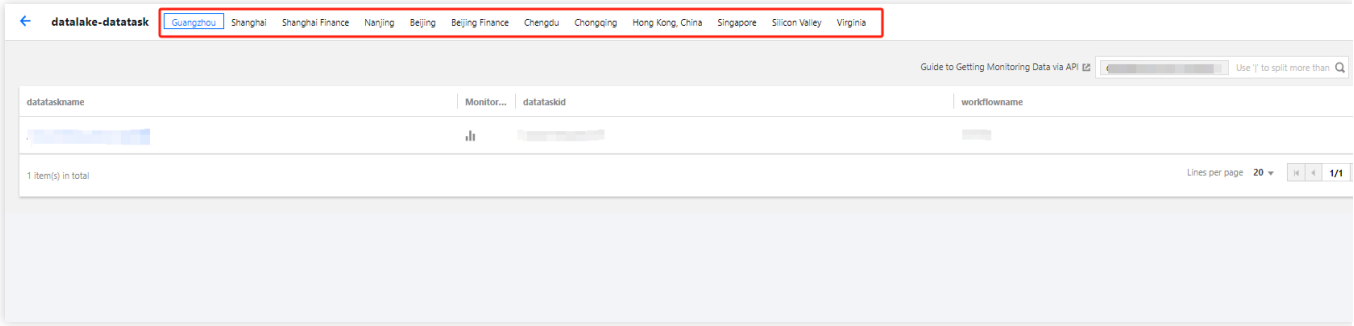


Entrance two: TCOP

1. Log in to TCOP Console. Account must have the required permissions.
2. In the left menu, select Cloud Product Monitoring, find DLC, and choose the type of monitoring you wish to view.



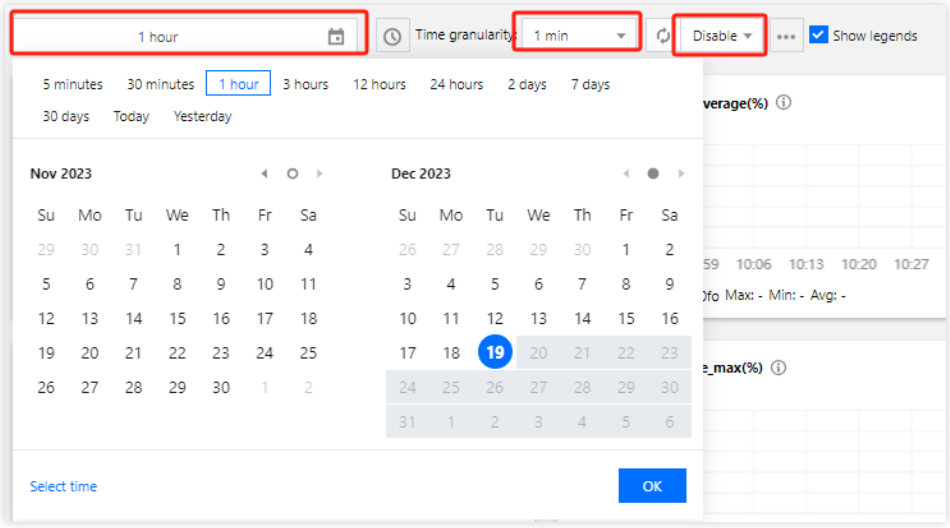
3. After selecting the monitoring type, enter the monitoring page and select the respective region to view the monitoring job information for that region.



4. Click **Job ID** to enter the monitoring details.

Monitoring Granularity Configuration

Supports configuring the monitoring data time period, time granularity, and automatic update time range through the monitoring settings at the top.



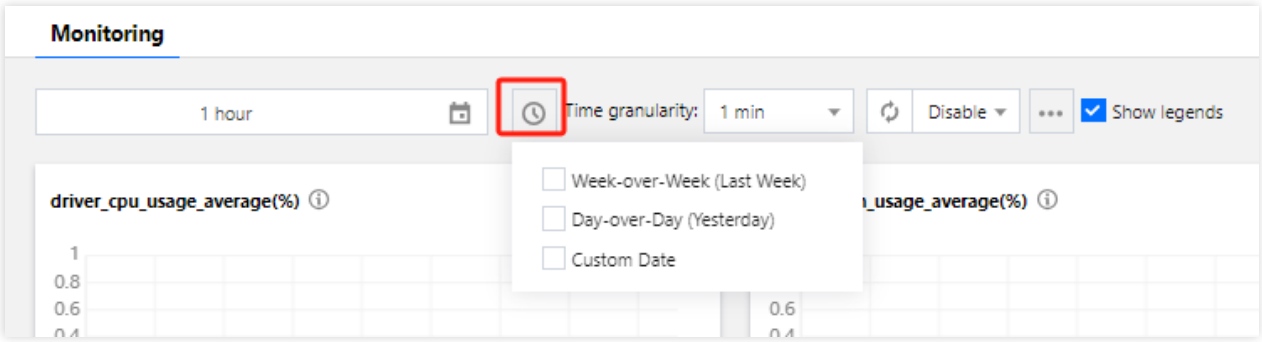
Monitoring Data Time Range: Precise to minutes, supports selecting data for a specific period.

Time Granularity: Monitoring point interval time, supports configuring for 1 minute or 5 minutes.

Automatic Data Update: Page data auto-refresh configuration, supports configuring off, 30s, 5min, 30min, 1h.

Monitoring Data Comparison

Supports selecting data for a specific period to compare monitoring data. After clicking to select the comparison time range, you can view the comparison data in the data compass below.



Monitoring Metric

Monitoring Type	Monitoring Metric
Job	Job error Log Count
	Job warn Log Count

Access Point Gateway Engine Monitoring

Last updated：2024-07-31 17:31:54

DLC provides monitoring services for the access point gateway engine based on TCOP service, ensuring you can understand the gateway status in real time.

Notes

Before using DLC's monitoring service, you need to activate the TCOP service (for usage details, see TCOP Documentation). If the service has not been activated yet, it can be activated using the root account.
TCOP service usage may incur related tariffs, for detailed tariff information, see TCOP Billing Overview.

Monitoring Entrance

Entrance one: DLC Console

1. Log in to the <1>Standard Engine> page, and select the Service Region.
2. Select the Standard Engine, and click on **Monitoring** at the access point to enter the monitoring data display interface.

Configuration Entrance: TCOP

1. Log in to the TCOP Console, the account must have the relevant permissions.
2. From the left menu, select Cloud Product Monitoring, enter the Policy Management page under Alarm Management, select Data Lake Computing, and choose the corresponding Access Point Gateway Engine.

Access Point Gateway Engine Monitoring Configuration Type

Creating alarm policy

1. DLC Access Point Gateway supports alarm capabilities. Log in to TCOP, click **Alarm Management**, and select the Policy Management page.
2. Click **New Policy**, for policy type choose "Data Lake Computing". Access Point Gateway supports alarms for three dimensions, including:
"Gateway" alarm dimension is: appid/gatewayid.
"Gateway (Multi-dimensional)" alarm dimension is: appid/gatewayid/instanceid.
"Gateway Engine (Multi-dimensional)" alarm dimension is: appid/gatewayid/engineid/processid.

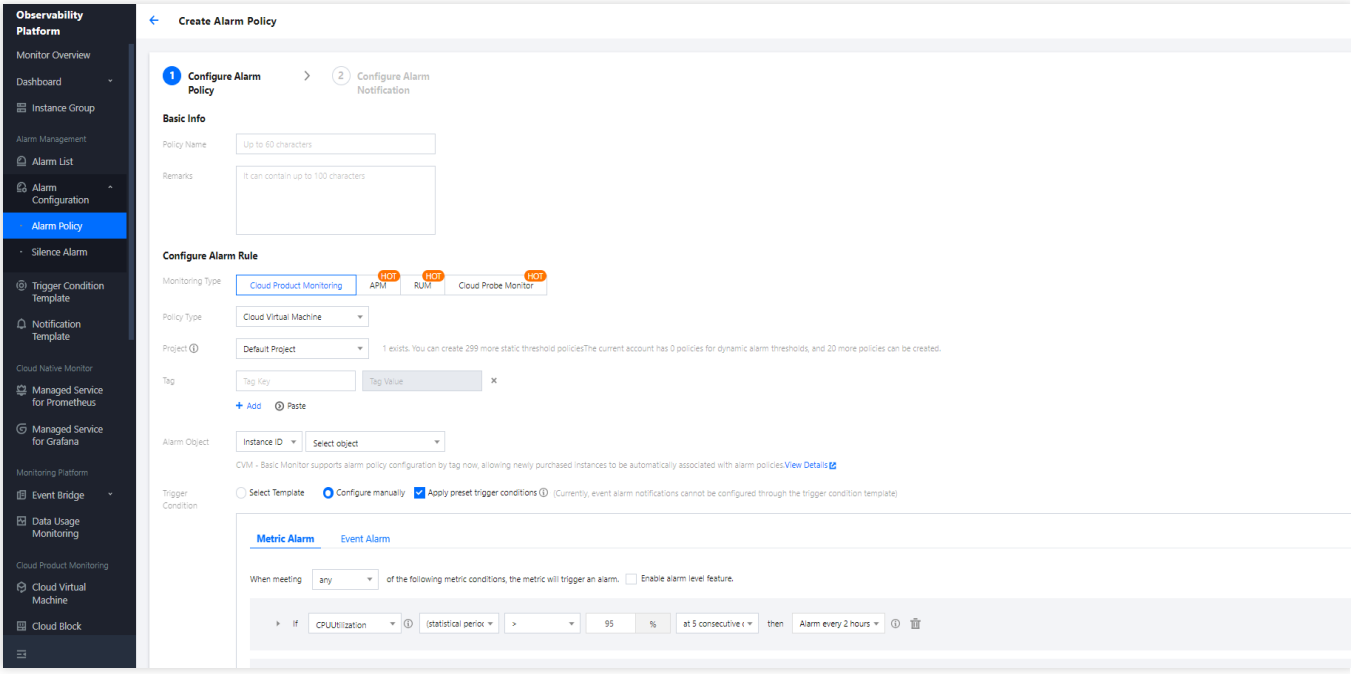
Name	Supported Dimensions	Advantages and Use Cases
Gateway (Multi-dimensional)	Supports: CPU, Memory, Disk, Network Fine-grained Alerting. For example, to configure an alert for the CPU utilization of an Access Point Gateway, you can choose to configure one, several instances under a specific Access Point Gateway, or any instance node triggering the threshold to alert.	Alert supports more dimensions, and the alert method is more flexible. Basic Metrics are recommended to use this approach.
API Gateway	Mainly aimed at monitoring the overall load situation of the current gateway, aggregating basic metrics according to Access Point Gateway Nodes, and supporting Service-level Metric Alerts. For example: execute_statement_num (number of statements executed), opened_operation_num (number of operations opened), launch_engine_num (number of engines started), engine_process_thread_num (number of threads started by the engine).	Supports Dashboard. Suitable for Single-node access point gateway or service metric alert.
Gateway Engine (Multidimensional)	The Gateway Engine refers to the monitoring and alarm of the process of starting the DLC engine by the Access Point Gateway. For example: engine_process_thread_num (number of threads started by the engine), mainly aimed at monitoring the process information of the engine started by the current Access Point Gateway	Supports fine-grained alerting, for example: commonly configure any engine's process count under a specific Access Point Gateway ID to reach the threshold to trigger an alert. Suitable for alerting on process metrics started by the Access Point Gateway.

Monitoring Alarm Configuration

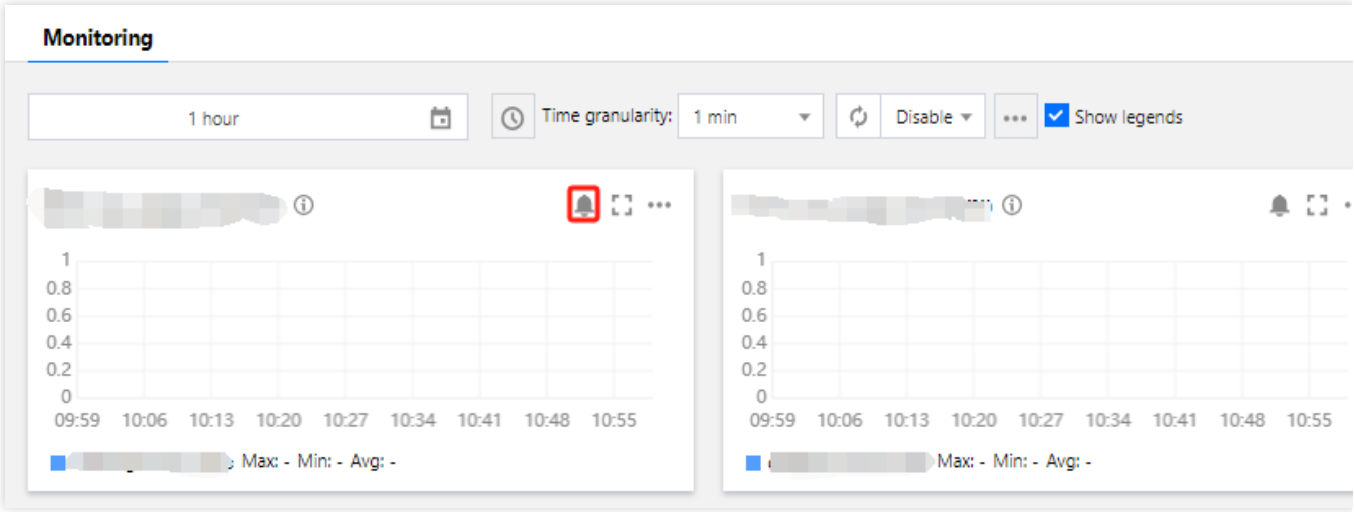
Last updated：2024-07-31 17:32:15

Configuring New Alarm Policy

Supports configuring monitoring alarms for specific metrics. You can go to Creating Alarm Policy to configure the content of the alarm.

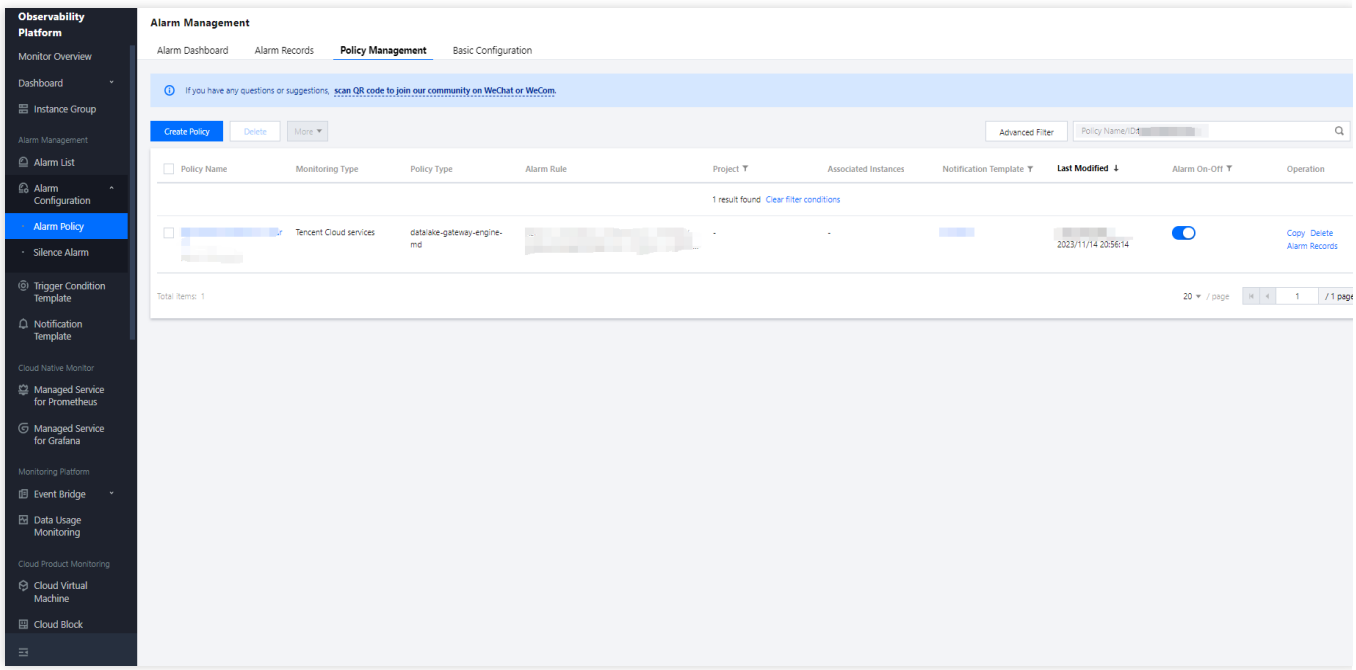


Or click the monitoring content for which you need to configure an alarm to enter the configuration page, where you can configure the content of the alarm.



Managing an alarm policy

To manage configured alarm policies, you can perform configuration management through the Policy Management page.



Configuration Instructions

Configuration Item	Configuration Instructions
Policy name	Name of the alarm policy, up to 60 characters
Remarks	Remarks for the alarm policy, up to 100 characters
Monitoring Type	Please select Cloud Product Monitoring
Policy Type	Please select DLC
Policy Tag	Support for managing policy content via Tag requires relevant permissions to operate
Alarm Object	You can configure alarms for Instance ID (supports multiple selections), grouped instances, and all instances
Alert Configuration Template	You can choose a template or configure manually. Administrators need to create the template in advance, and it supports configuring multiple alert rules
Notification Template	Supports creating or selecting existing notification templates, with support for configuring up to 3 templates

Audit Log

Last updated：2024-07-31 17:30:53

DLC provides an operation log audit service based on Tencent Cloud's CloudAudit service, ensuring you can understand the system operation records in real time and check the operation information.

Notes

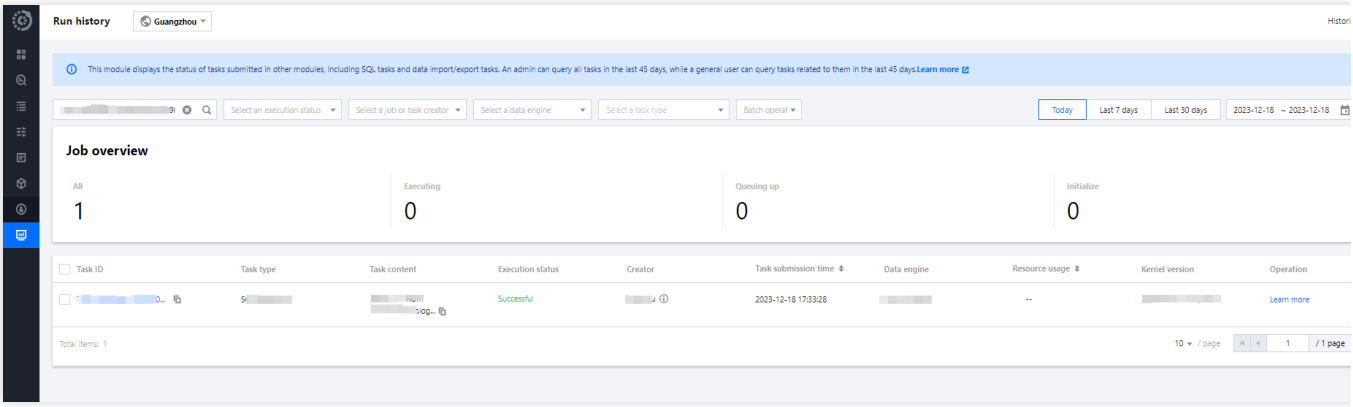
Before using the audit CLS of DLC, you need to activate Tencent Cloud's CloudAudit service. If the service is not yet activated, you can activate it with the primary account.

Use Instructions

The Data Lake Computing Console currently displays up to 3 months of log information. To view older log information, you can go to CloudAudit.
The audit logs contain console operations and API call operations. Currently, it supports viewing log information for engine management, task management, data source management, workgroup management, user management, scheduled task instance management, scheduled task management, and scheduling plan management.

Operation Guide

1. log in to Data Lake Computing Console, select **Service Region**.
2. Through the left menu **Data Operation and Maintenance**, select the Audit Log feature.
3. Supports log queries based on user UIN or request ID.
4. Detailed log information can be viewed by clicking **Query Details**.


Run history
Guangzhou
This module displays the status of tasks submitted in other modules, including SQL tasks and data import/export tasks. An admin can query all tasks in the last 45 days, while a general user can query tasks related to them in the last 45 days.Learn more
Select an execution status
Select a job or task creator
Select a data engine
Select a task type
Batch operat
Today
Last 7 days
Last 30 days
2023-12-18 ~ 2023-12-18
Job overview
All
1
Executing
0
Queuing up
0
Initialize
0
Task ID
Task type
Task content
Execution status
Creator
Task submission time
Data engine
Resource usage
Kernel version
Operation
Successful
2023-12-18 17:33:28
--
Learn more
Total items: 1
10 / page
1
/ 1 page