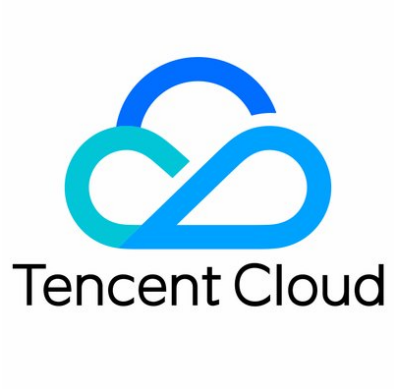
Data Lake Compute

Getting Started



Contact Us

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

Contents

Getting Started
Complete Activation Process for New Users

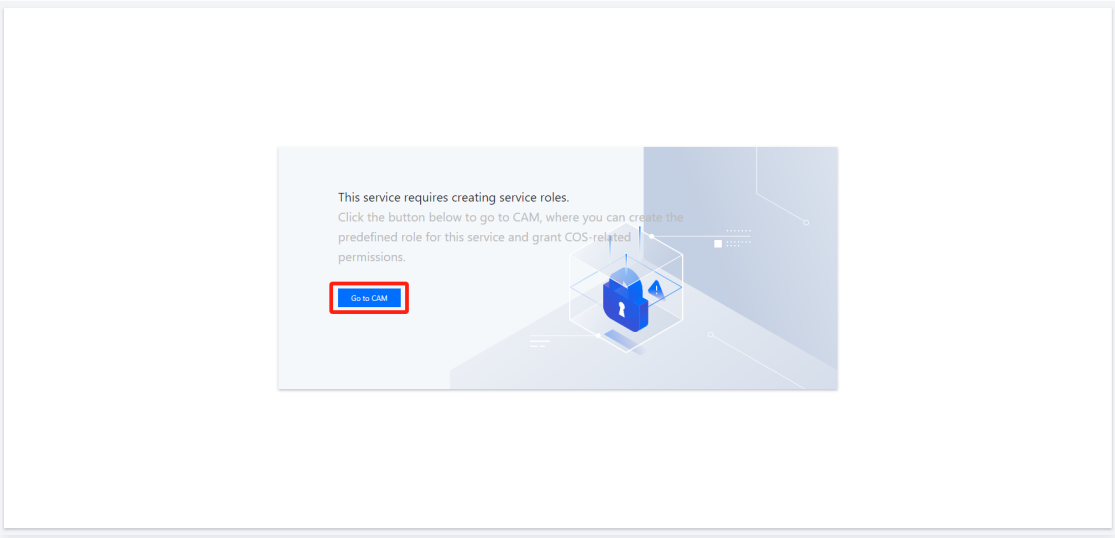
Last updated: 2024-01-10 15:52:25

Preliminary Preparations

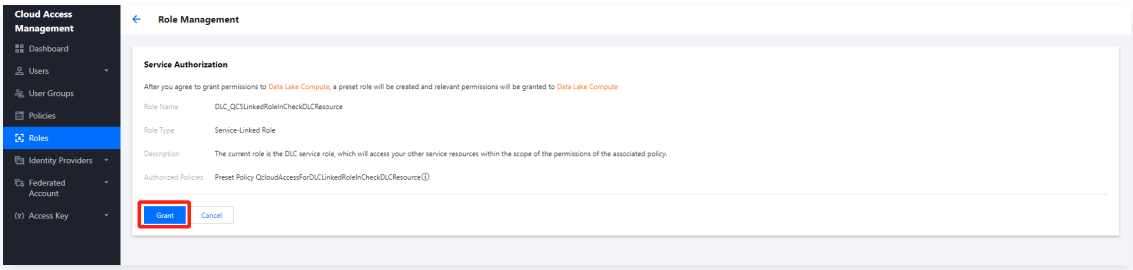
Registering an account

> **Note:**
> This is exclusively for administrators and may involve the following operations.

1. Navigate to the Data Lake Compute (DLC) console and click **Go to CVM** to authorize Data Lake Compute.



2. In Role Management, click **Grant**.

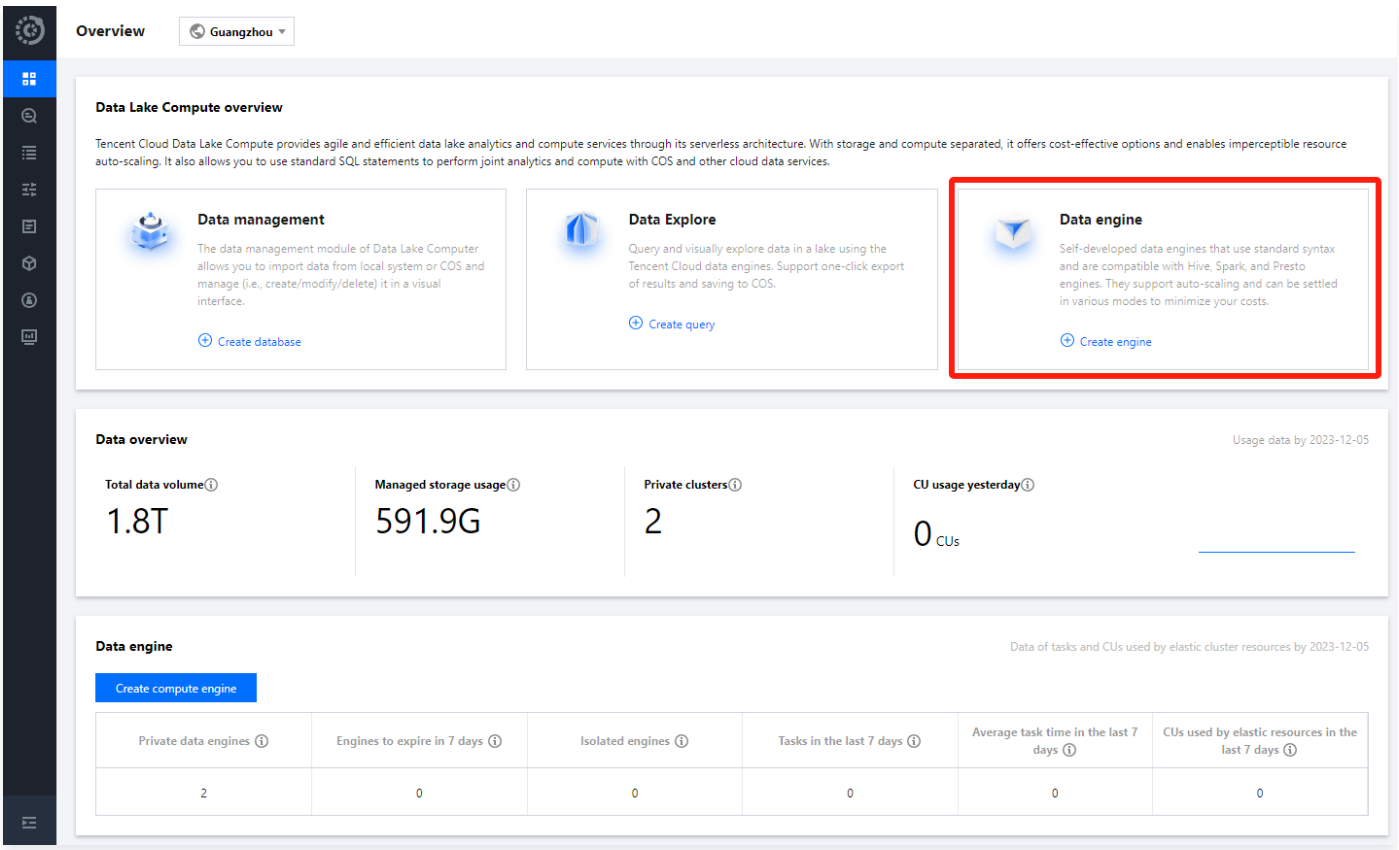


Purchasing an Engine

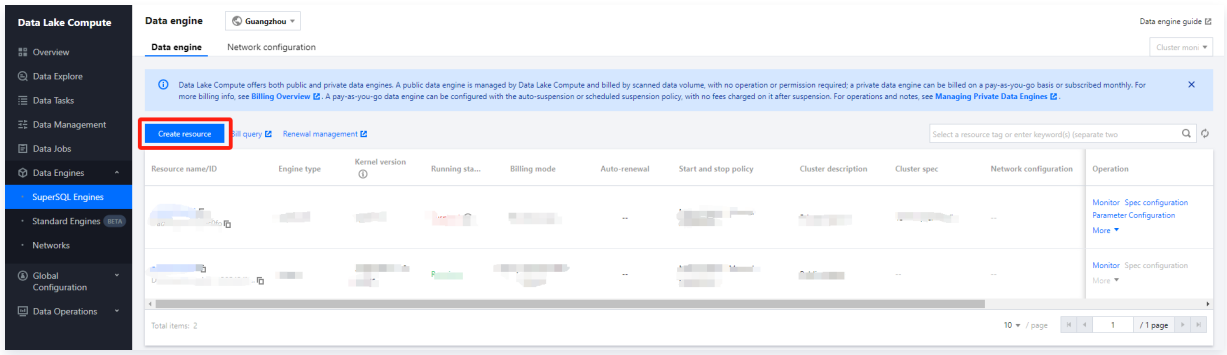
> **Note:**
> **Financial permissions are required in CAM (Cloud Access Management).**

1. On the Data Lake Compute (DLC) console, you can navigate to the engine purchase page through the **Overview** and **Data Engine** pages to purchase an engine.

   Overview Page > Initial Configuration > Purchase Data Engine:

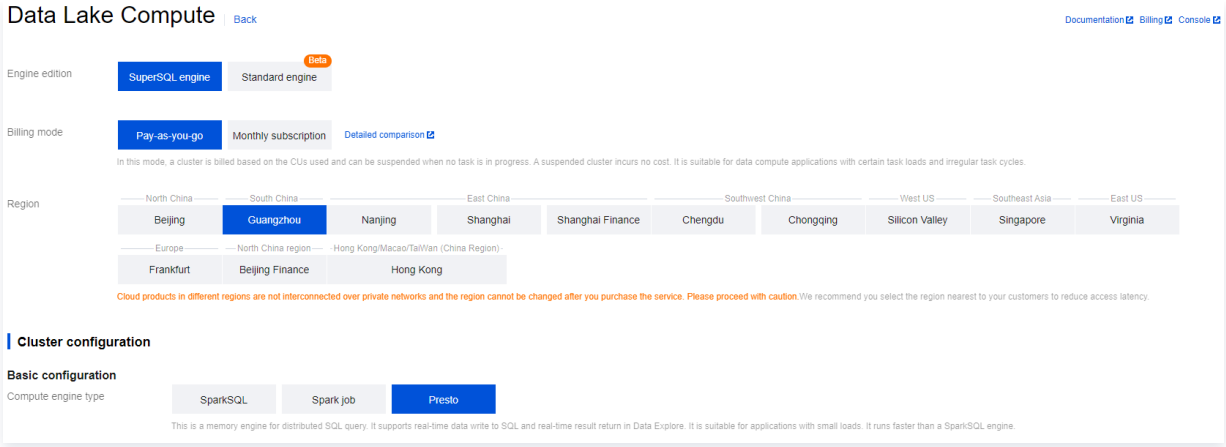


Data Engine > Create Resource:



2. Select the type of engine you wish to purchase on the purchase page:

> **Note:**
> - SparkSQL: Suitable for stable and efficient offline SQL tasks.
> - Spark Job: Suitable for native Spark streaming/batch data job processing.
> - Presto: Suitable for agile and rapid interactive query analysis.



> **Note:**
> The scale of a 6CU cluster is relatively small, it is recommended for testing scenarios only. For real production scenarios, it is advisable to select a cluster specification of 64CUs or above.

Team Account Activation

If you have a demand for multiple accounts to use the product collaboratively, you can follow the suggested operations below to enable it:

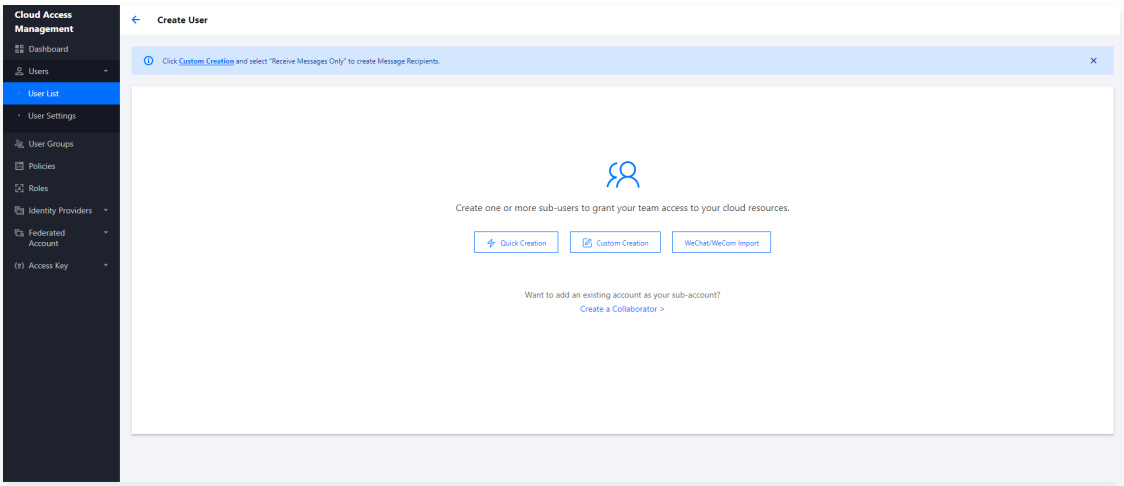
1. Permissions are not universal across each region; different regions require separate configuration of corresponding permissions.
2. Quick access to Data Lake Compute (DLC) permissions:
   - To grant a sub-account access to Data Lake Compute (DLC), please navigate to the CAM Console for configuration.
   - To enable sub-account read and write permissions for data and engines within the Data Lake Compute (DLC) product, please go to the DLC console for configuration.

Granting Sub-account Access and DLC Permissions

The primary account inherently possesses all operational permissions for Data Lake Compute (DLC). The primary account can grant DLC access permissions to sub-users through Cloud Access Management (CAM), enabling the sub-users to have corresponding DLC operational permissions: **QcloudDLCFullAccess (Full operational permissions for DLC).**

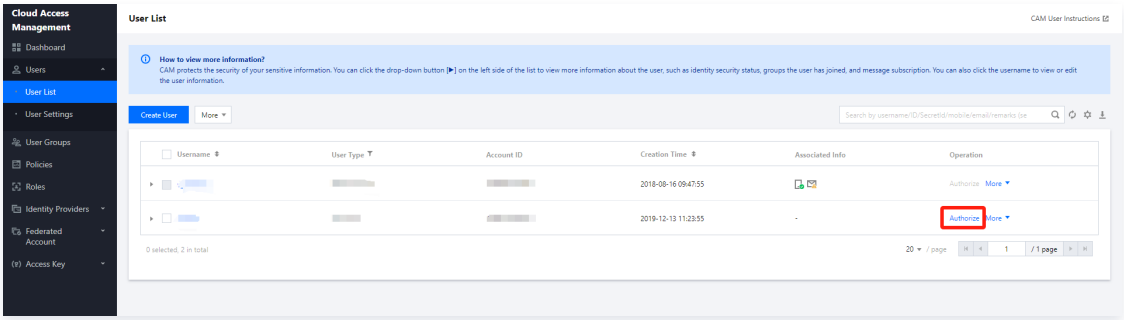
Instructions

1. Log in to the CAM Console to create a sub-user. For detailed operations, please refer to Creating and Authorizing a Sub-account.

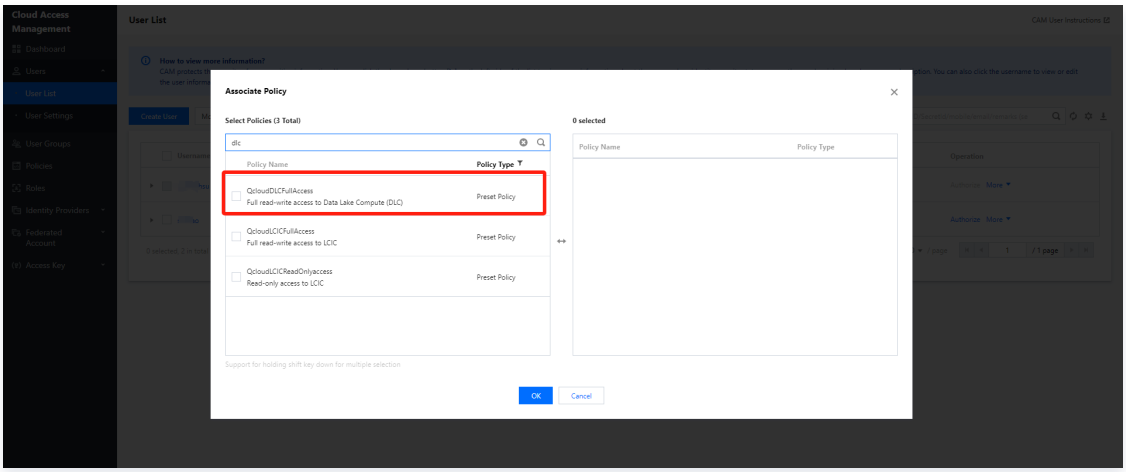


2. Add the preset policy **QcloudDLCFullAccess** (full operational permissions for DLC) to the sub-account.

   You can search for the user to be authorized in the user list and click **Authorize**.



   In the policy list, select **QcloudDLCFullAccess (full operation permissions for DLC).**



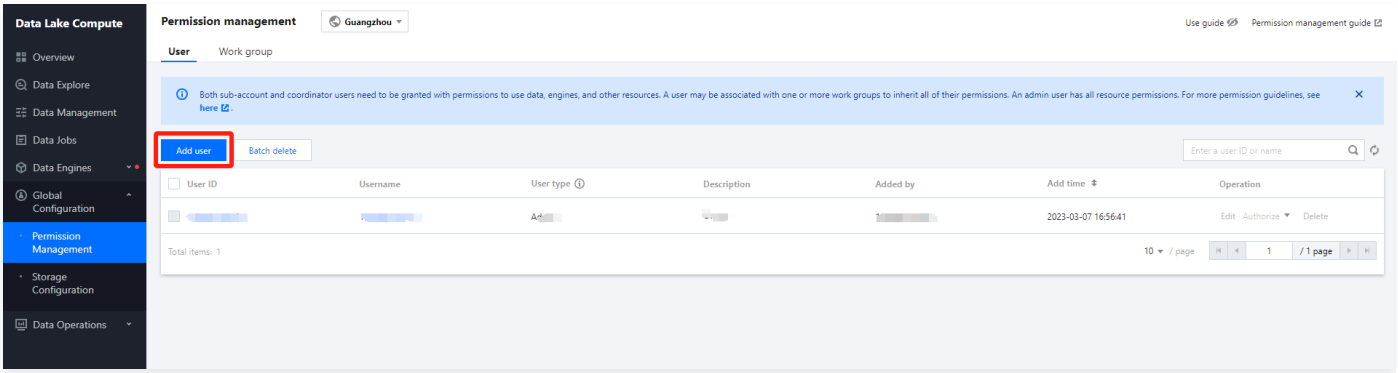
Granting sub-account permissions for data and engines in DLC

Adding Users to DLC Permission Management

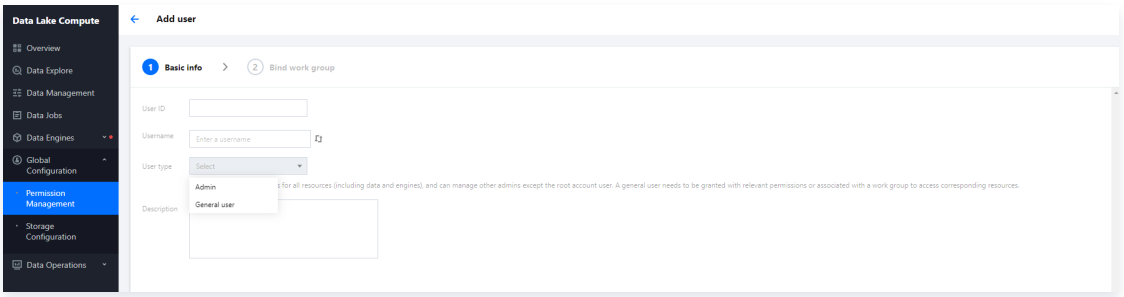
Note:

1. Please confirm the region where the user permissions take effect.
2. Regarding User Segmentation:
   - Administrator: Possesses permissions to all resources.
   - Regular users: Specific permissions need to be granted, or they can be associated with a workgroup to obtain permissions.

1. Log in to the Data Lake Computing DLC Console, navigate to the Permission Management page, select the **corresponding service region**, and proceed to the Permission Management page. Click on **Add User.**

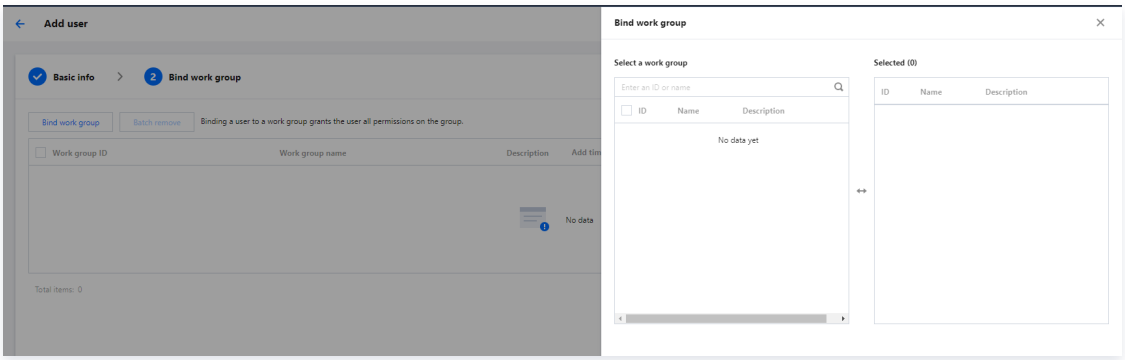


2. Add the account to DLC for management using the sub-user's **CAM ID**. Please select the user type as per your requirements.



3. Associating users with a workgroup (This step is optional).

> **Note:**
> If you need to manage the usage permissions of multiple users, you can do so by associating them with a work group. After the work group is created, you can proceed with the addition. For the specific creation process, please refer to Getting Started with DLC Permission Management in One Minute.

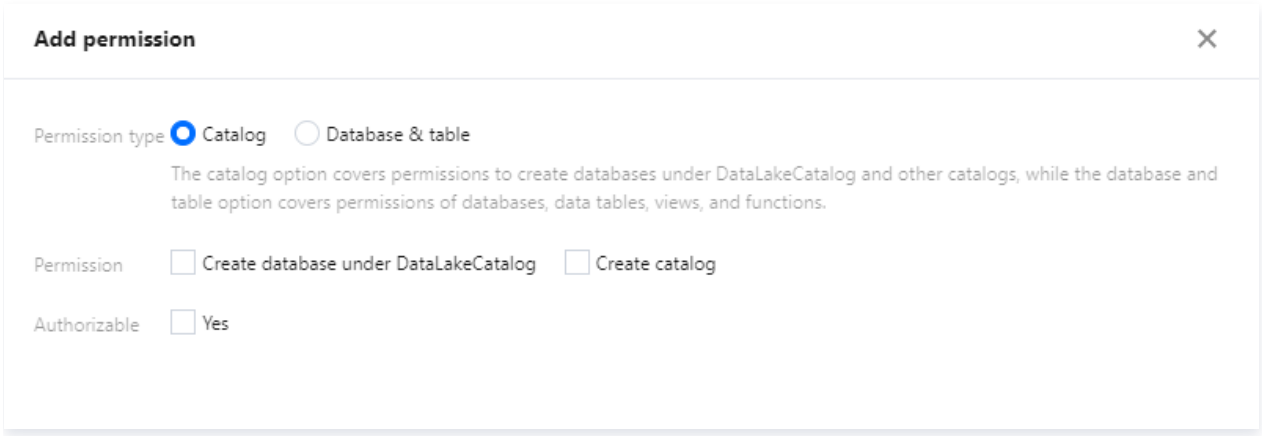


Add Engine and Data Permissions

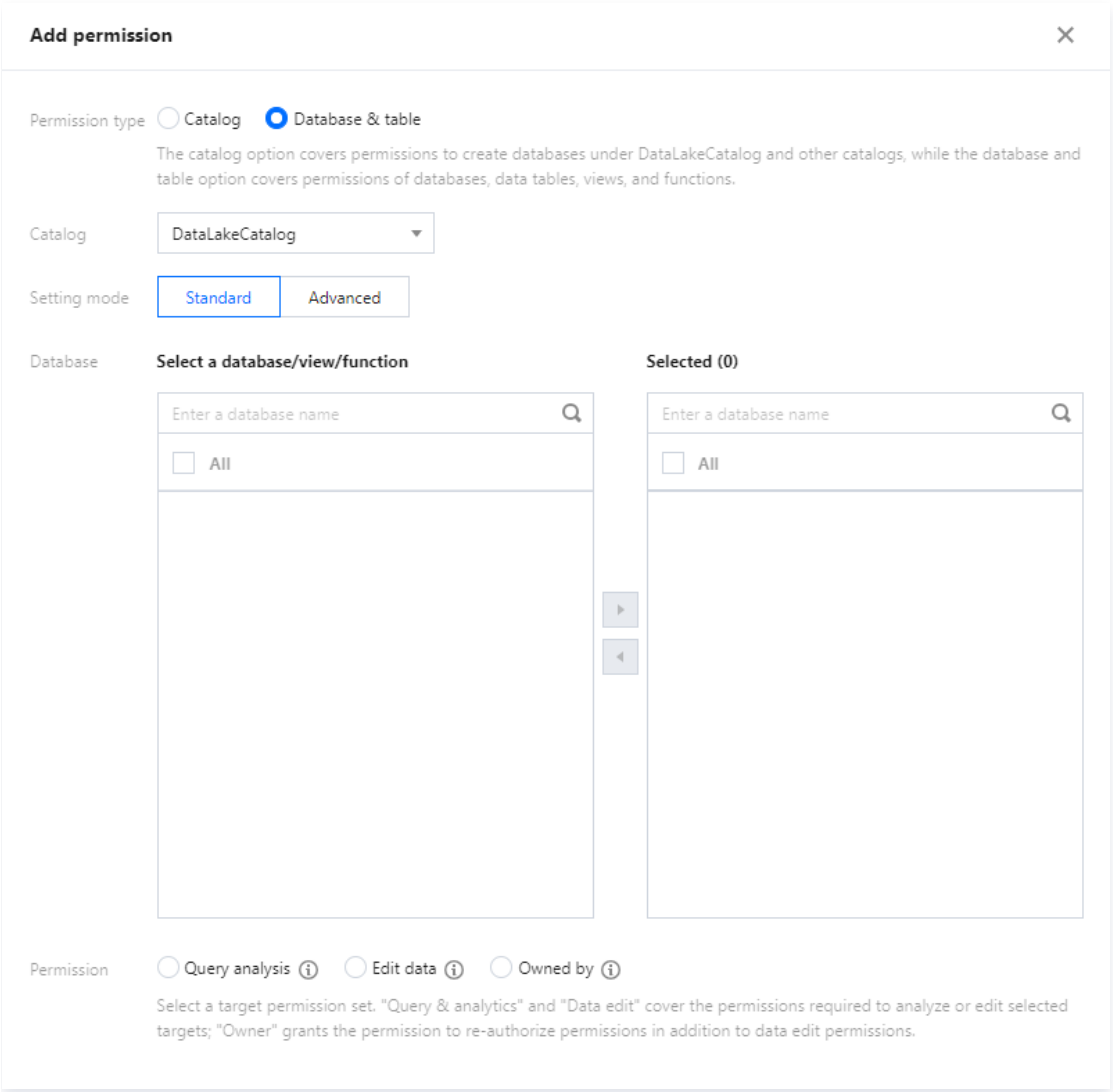
After creating a user or workgroup, click on the authorization operation in the list to add permissions to the workgroup, including data permissions and engine permissions.

Data Permissions

Data Directory Permissions: This includes the authority to create databases and data directories within the data directory.



Database Table Permissions: Fine-grained permissions at the database table level can be granted, including query and edit permissions for databases, tables, views, and functions.

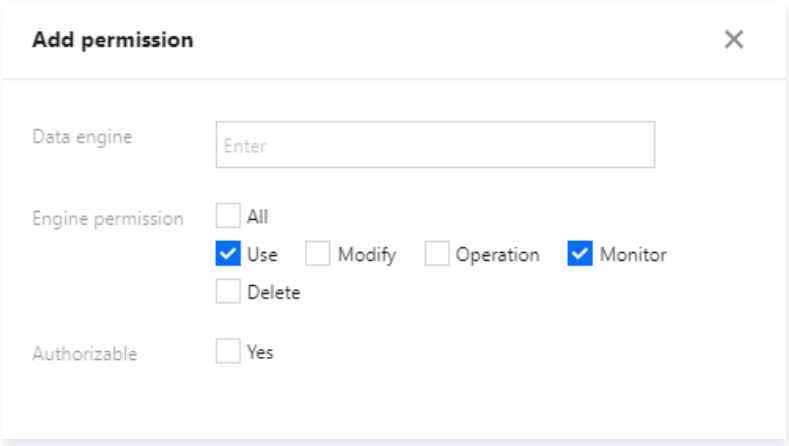


Engine permission

Based on the usage scenarios of the user or workgroup, select the engine's permission policies.

Note:

- Utilization: Employ this engine for task execution.
- Modification: Alter the configuration parameters of the engine, such as adjusting the engine's specifications.
- Operation: Pause and suspend the engine.
- Monitoring: Operational oversight of engine usage.
- Deletion: Proceed to remove the engine.
- Grantable: If checked, all members of this sub-user or workgroup will have the authority to grant permissions to the engine.

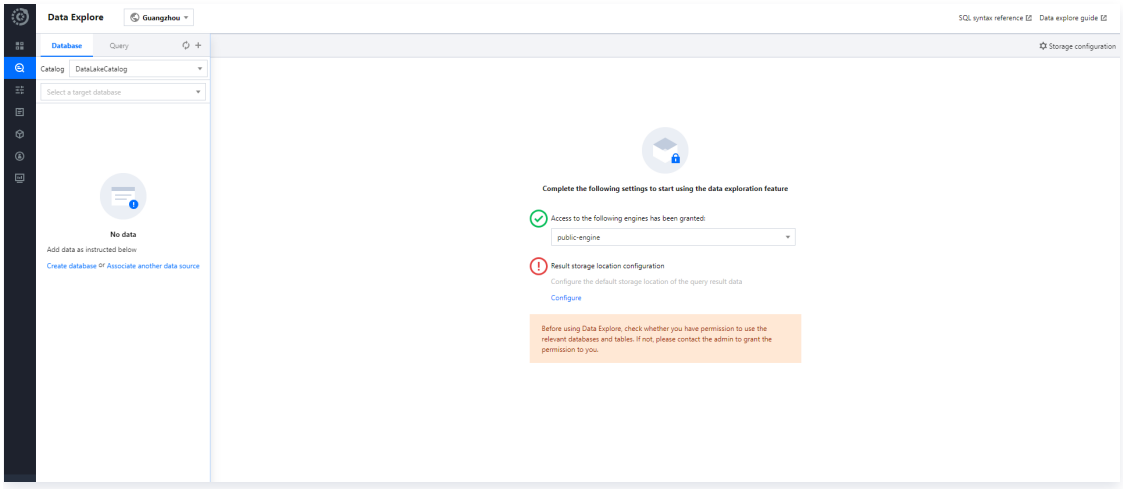


Configuring the Result Storage Location

Before using the data exploration feature, you need to configure the query result path. Once configured, the query results will be saved to the specified COS path or DLC's managed storage. For detailed operations, please refer to the Guide to Configuring Query Result Paths.

Configuring the Result Storage Location

Navigate to the Data Lake Compute (DLC) console, select the data exploration feature, choose the configuration for the result storage location, and click **Configure**.

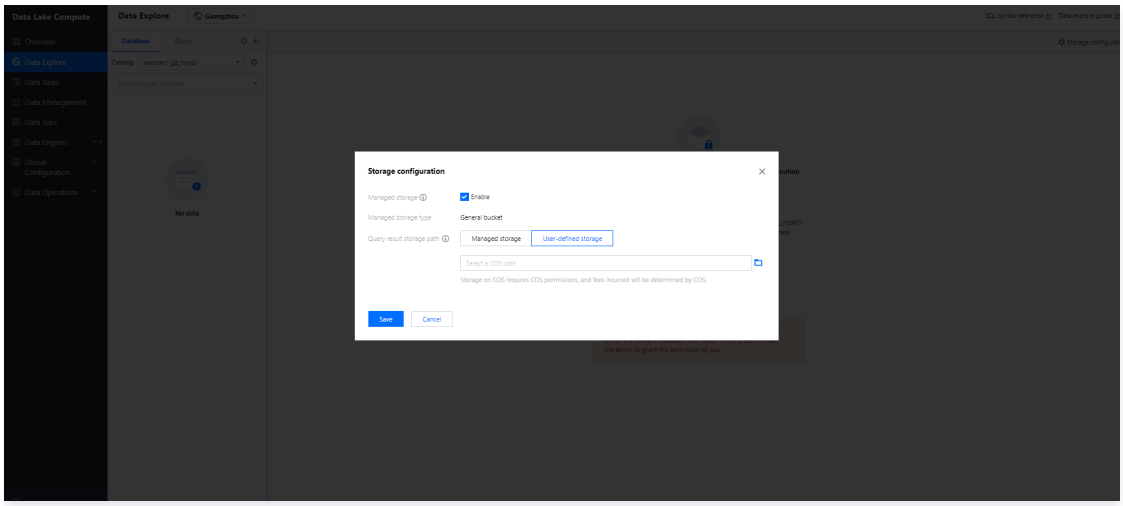


Select the location and method of storage

Note:

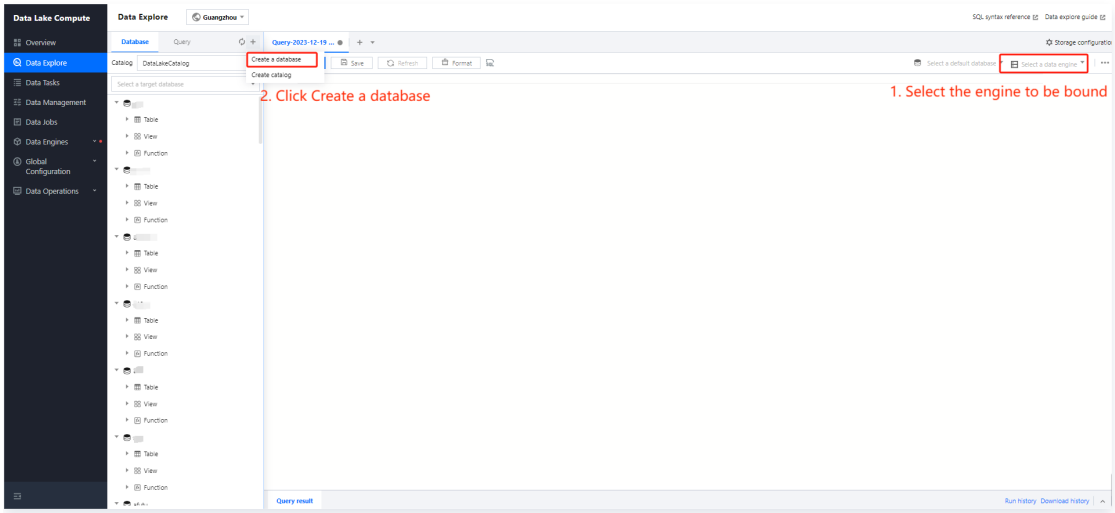
- Metadata Acceleration Bucket: In the current region, it can significantly enhance query analysis performance. Internal tables can be directly enabled, while external tables require confirmation if the engine permission allows enabling.

  Please note: Shared engines cannot bind to metadata acceleration buckets. When a user selects a user storage path, the exclusive engine needs to bind to the metadata acceleration bucket first, and then the query can take effect.
- User Storage: User storage refers to your bucket path on COS.

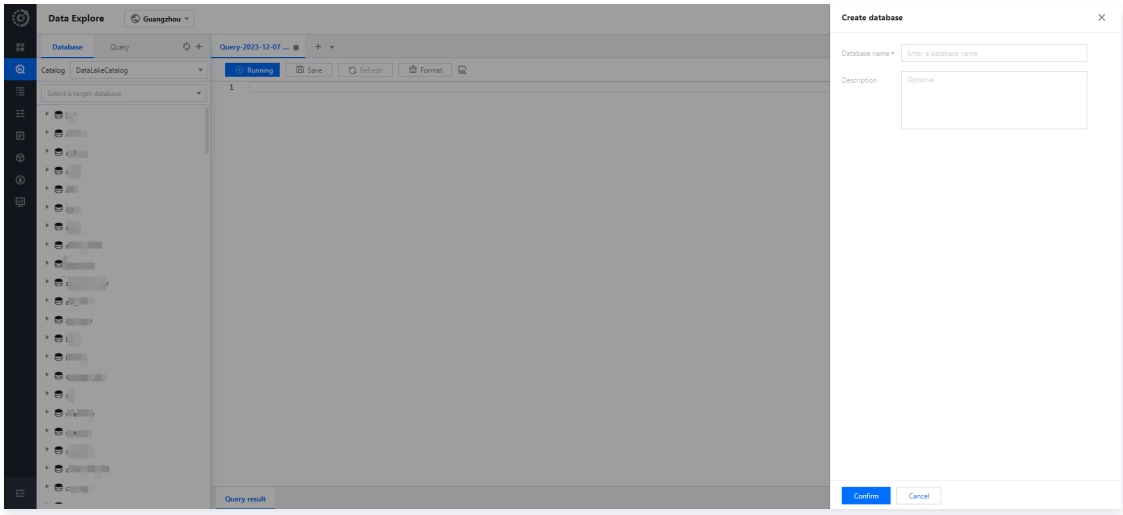


Establishing a Database

Before creating a database, choose the engine to be utilized.

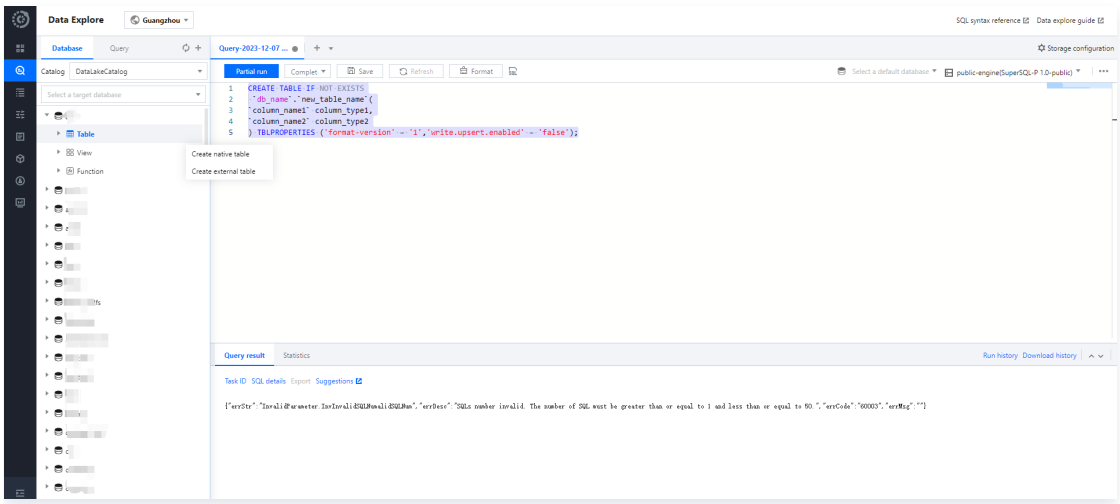


Enter the database name and click **Confirm**.

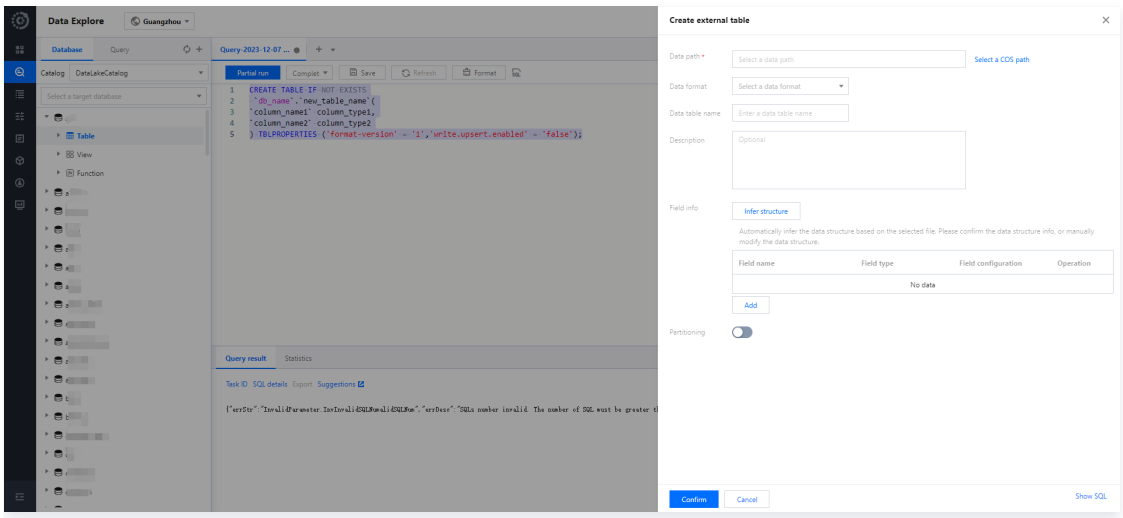


Create table

Native Table: This is your table stored on the managed storage of Data Lake Compute (DLC), by default in Iceberg format. Using a native table eliminates the need to manage underlying Iceberg files and provides data optimization capabilities to help build a data lake.



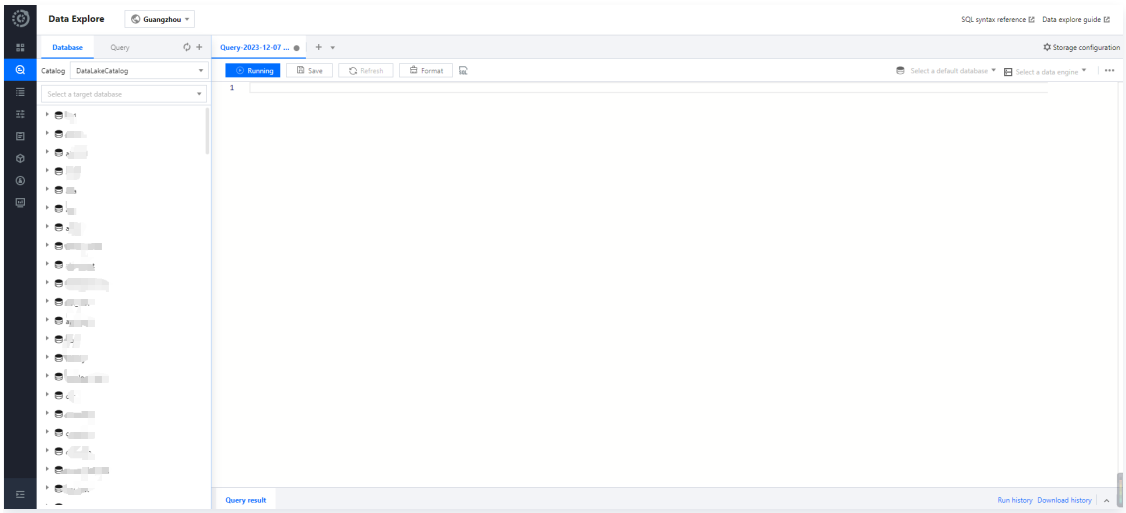
External Table: This refers to tables stored in your own COS bucket or other third-party data storage. DLC can directly create external tables for analysis without the need for additional data loading.



> **Note:**
> **After creating a native table, you need to refresh your browser before you can use it.**

Data Query

Navigate to the Data Lake Compute (DLC) console – Data Exploration. On the analysis page, you can create SQL queries. The features support full execution, partial execution, result download, and materialized views.
For details on SQL syntax, please refer to SQL Syntax.



Data Lake Compute Data Import Guide

Last updated: 2024-01-10 15:52:36

Importing External Table Data via COS

Data Lake Compute (DLC) supports querying and analyzing data directly on COS without migrating data. Therefore, you only need to import the data into COS to start using DLC for seamless data analysis, achieving complete decoupling of data storage and computation. Currently, it supports uploading a variety of formats such as orc, parquet, arvo, json, csv, and text files.
At present, COS offers a wealth of data import methods. You can choose the following methods to import data according to your own situation.

- Log in to COS and directly upload files. For related operation steps, please refer to Uploading Objects.
- Import data using various upload tools provided by the COS service. For a list of supported tools, please refer to Tool Overview.
- Import data using the SDK or API provided by the COS service. For service-related instructions, please refer to Upload Interface Documentation.
- If you need to import logs from CLS for analysis, you can directly deliver the logs to COS according to partitions and then directly analyze and query them using DLC. For related operations, please refer to Using DLC (Hive) to Analyze CLS Logs.
- If you need to import data from other cloud services (such as CDB databases) into COS, you can use DataInLong for the import. When creating a data synchronization link, select the cloud service that needs to be exported as the data source, and select COS as the destination to complete the data import. For more information on how to use the data integration service, please refer to Data Integration.

If you encounter any issues during data import, you can consult us for solutions by submitting a ticket.
After importing data into COS, you can use the DLC console, API, or SDK to perform SQL queries, table creation, analysis, result export, and other operations. For detailed operations, please refer to Getting Started with DLC Data Analysis in One Minute.

Importing Data into Native Tables

To provide better data query performance, Data Lake Compute (DLC) also supports querying and analyzing data after importing it into native tables. DLC native tables arrange data based on the Iceberg table format and optimize the data during the import process. If you have the following use cases, it is recommended to use native tables for data query analysis.

- In the context of data warehouse analysis scenarios, it is desirable to leverage Iceberg indexing for improved analytical performance.
- If there is a need to update data, it can be achieved through the DLC service using SQL or data jobs to perform UPSERT operations.
- Data is updated in real-time through Data Integration DataInLong, Flink, Stream Compute Oceanus, Spark Streaming, with simultaneous reading and writing, suitable for data processing businesses that require transactional guarantees.
- If you wish to utilize features related to Iceberg tables, such as time travel, multi-version snapshots, hidden partitioning, and partition evolution, among other advanced data lake features.

If you need to import data into a native table, you can choose the following methods to import data according to your own situation.

- Import directly through the Data Lake Compute DLC Console. For detailed operation steps, please refer to COS Data Import.

> ⚠ Note

> When importing data via the console, there are certain usage restrictions. It is mainly used for quick testing and is not recommended for production use.

- If your original data is in MySQL, Kafka, etc., and you need to write/update MySQL binlog, middleware data to DLC in real-time every minute, you can achieve this through the real-time import capability of DataInlong. For detailed operation steps, please refer to DLC Real-time Data Import and Small File Merge. Alternatively, you can write through Stream Compute Service or Flink. If you need operation guidance, please contact us through Work Order.
- If the original data is in MySQL, Kafka, MongoDB, etc., you can use the offline synchronization task of DataInLong to transfer the data to the native table (for creation steps, see Creating Offline Synchronization Tasks). During the data warehouse modeling process, the external table is used as the original data source layer. In the process of transferring data to the native table, you can rearrange the data distribution in conjunction with the business by building sparse indexes, etc., to achieve excellent native table query analysis performance. If you need guidance, please Contact Us.
- Use the SELECT INSERT method in SQL syntax to query the data from the external table and write it into the native table. For example, after creating a native table with the same structure as the external table in DLC, complete the transfer by executing SQL syntax through the SparkSQL engine. The syntax example is as follows:

```
--- External table name: outtertable, Native table name: innertable
insert into innertable select * from outtertable
```

If you encounter any issues while importing data, please submit a ticket and we will provide you with a solution.

Federated Query Analysis Across Multiple Data Sources

If you do not wish to export data to COS or the native tables of DLC, Data Lake Compute also provides data federation query analysis capabilities. It allows you to quickly associate and analyze data from multiple data sources using SQL without migrating data. Currently, it supports a variety of data sources including MySQL, SQLServer, ClickHouse, PostgreSQL, EMR on HDFS, and EMR on COS. For instructions on adding a federated analysis data directory, please refer to Data Directory and Database Management.
When using federated analysis, the data source and data engine need to be in the same network. For network connectivity and management, please refer to Engine Network Configuration.

- When analyzing EMR data through Data Lake Compute (DLC), the query performance will be on par with or even exceed that of EMR, making it suitable for production environments. Without migrating EMR services, you can fully utilize the fully managed and elastic capabilities of DLC to reduce costs and increase efficiency.
- Federated analytics can quickly combine data from multiple sources for analysis, providing a convenient way for data insights and rapid analysis. Relying on the fully managed and elastic capabilities of DLC, it can effectively reduce usage costs. It also supports the use of INSERT INTO/INSERT OVERWRITE syntax to write federated data into DLC native tables, completing data import.
- When federating analysis from other data sources, there is a certain performance loss compared to direct queries from the original data source, as the data needs to be synchronized to DLC for analysis during the computation process. If high query performance is required, you can import the data into the native table for analysis. For operation methods, please refer to Importing Data into Native Tables.

Quick Start with Data Analytics in Data Lake Compute

Last updated: 2024-01-10 15:52:43

With Data Lake Compute, you can complete data analysis queries on COS in just a minute. It currently supports multiple formats including CSV, ORC, PARQUET, JSON, ARVO, and text files.

Preliminary Preparations

Before initiating a query, you need to activate the internal permissions of Data Lake Compute and configure the path for query results.

Step 1: Establish the necessary internal permissions for Data Lake Compute.

> **Note**
> If the user already has the necessary permissions, or if they are the root account administrator, this step can be disregarded.

If you are logging in as a sub-account for the first time, in addition to the necessary CAM authorization, you also need to request any Data Lake Compute admin or root account admin to grant you the necessary Data Lake Compute permissions from the **Permission Management** menu on the left side of the Data Lake Compute console (for a detailed explanation of permissions, please refer to DLC Permission Overview).

1. Table Permissions: Grant read and write operation permissions to the corresponding catalog, database, table, and view.
2. Engine Permissions: These can grant usage, monitoring, and modification rights to the computation engine.

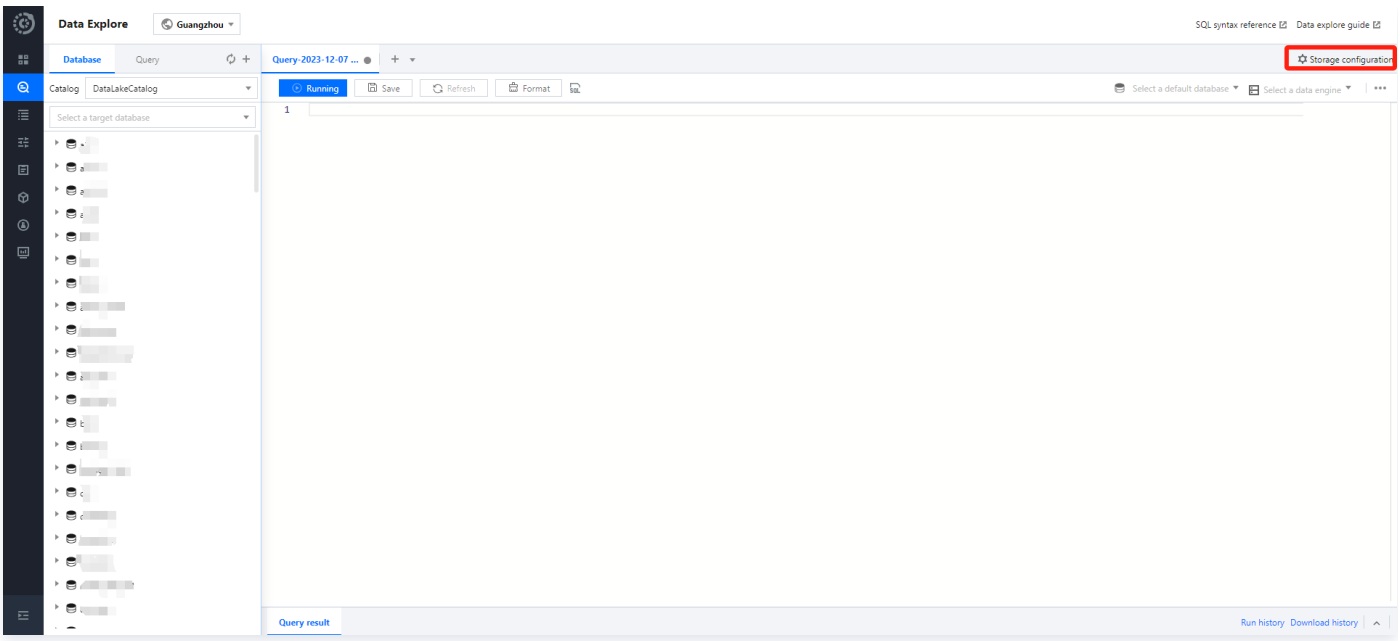
> **Note**
> The system will automatically provide each user with a shared public-engine based on the Presto kernel, allowing you to quickly try it out without the need to purchase a private cluster first.

For detailed steps on granting permissions, please refer to Sub-account Permission Management.

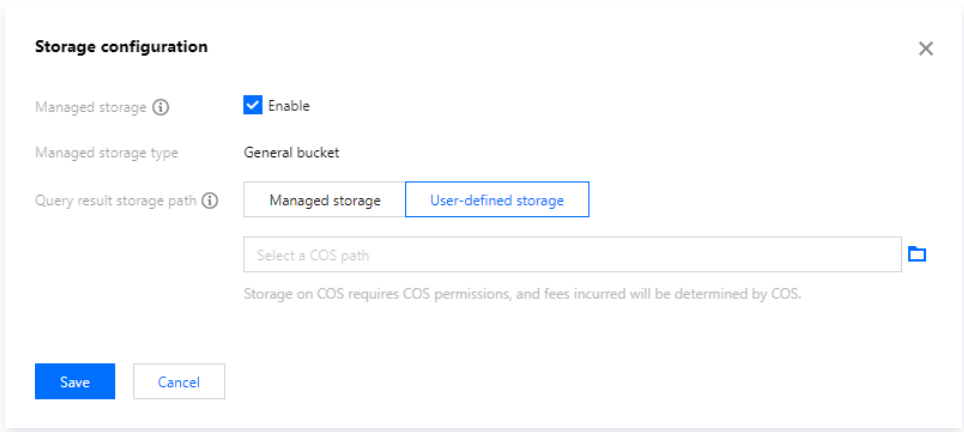
Step 2: Configure the path for query results.

Upon initial use of Data Lake Compute, you must first configure the path for query results. Once configured, the query results will be saved to this COS path.

1. Log in to the Data Lake Compute DLC console and select the **service region**.
2. Navigate to **Data Exploration** via the left sidebar menu.
3. Under the **Database and Tables** page, click on **Storage Configuration** to set the path for query results.



Specify the COS path for storage. If there are no available COS buckets in your account, you can create one through the Object Storage Console.

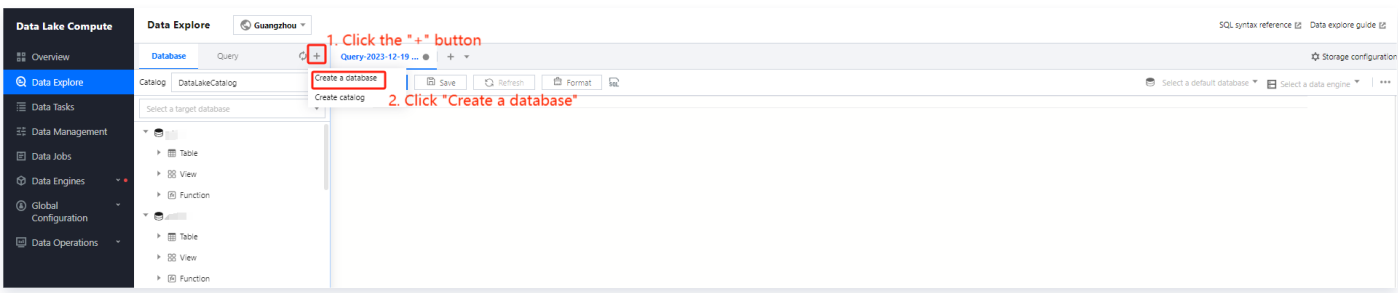


Analysis Steps

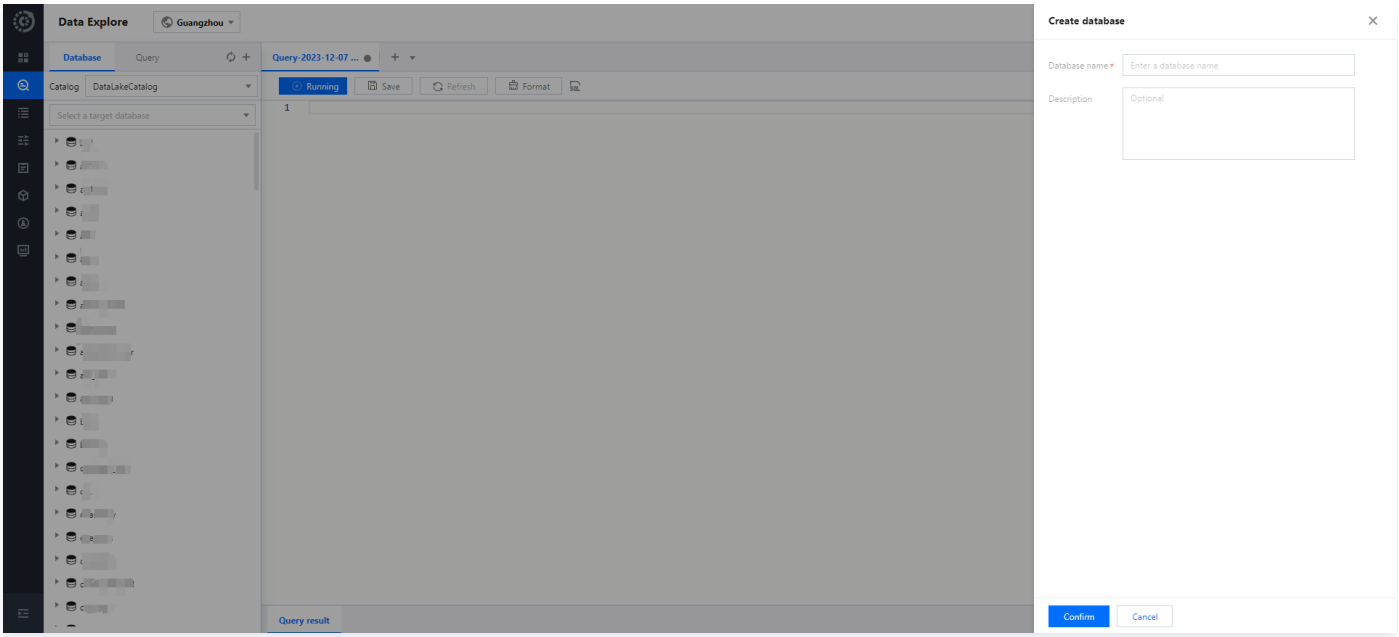
Step 1: Create a Database

If you are familiar with SQL statements, write the `CREATE DATABASE` statement in the query and skip the creation wizard.

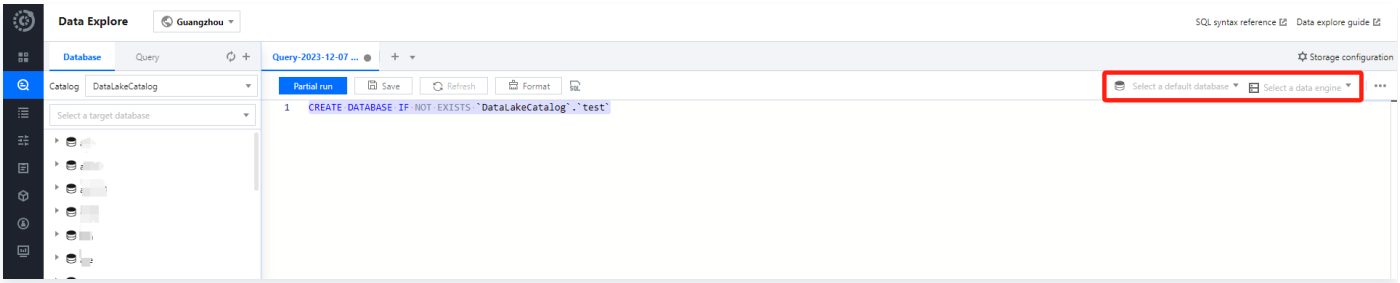
1. Log in to the Data Lake Compute DLC console and select the **service region**.
2. Navigate to **Data Exploration** via the left sidebar menu.
3. Select **Database**, click +, choose **Create Database** to establish a new database. As shown below:



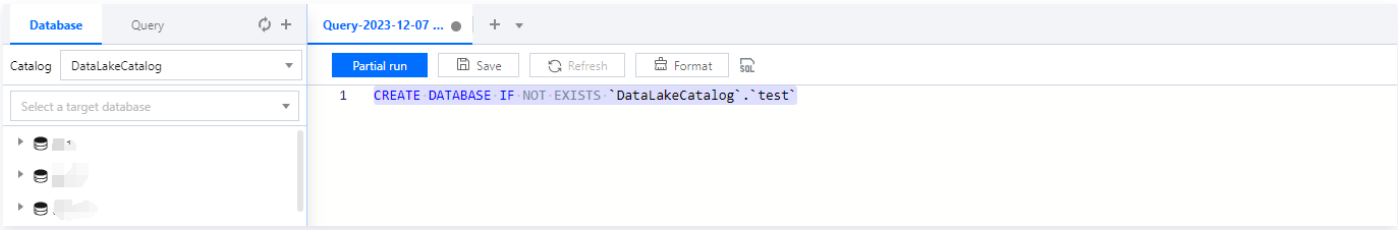
Enter the database name and its descriptive information.



4. After selecting the execution engine in the upper right corner, execute the generated 'create database' statement to complete the database creation.



The details are as shown below:



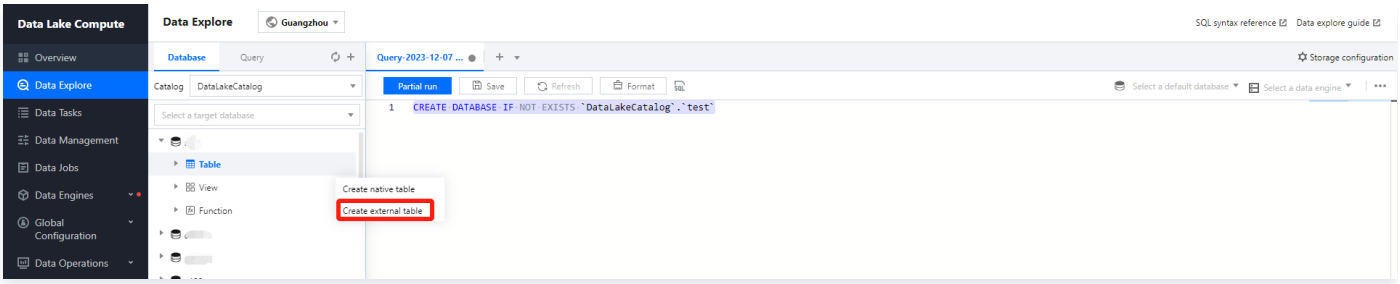
For detailed operation steps and configuration methods, please refer to Database Management.

Step 2: Create an External Table

If you are familiar with SQL statements, write the `CREATE TABLE` statement in the query and skip the creation wizard.

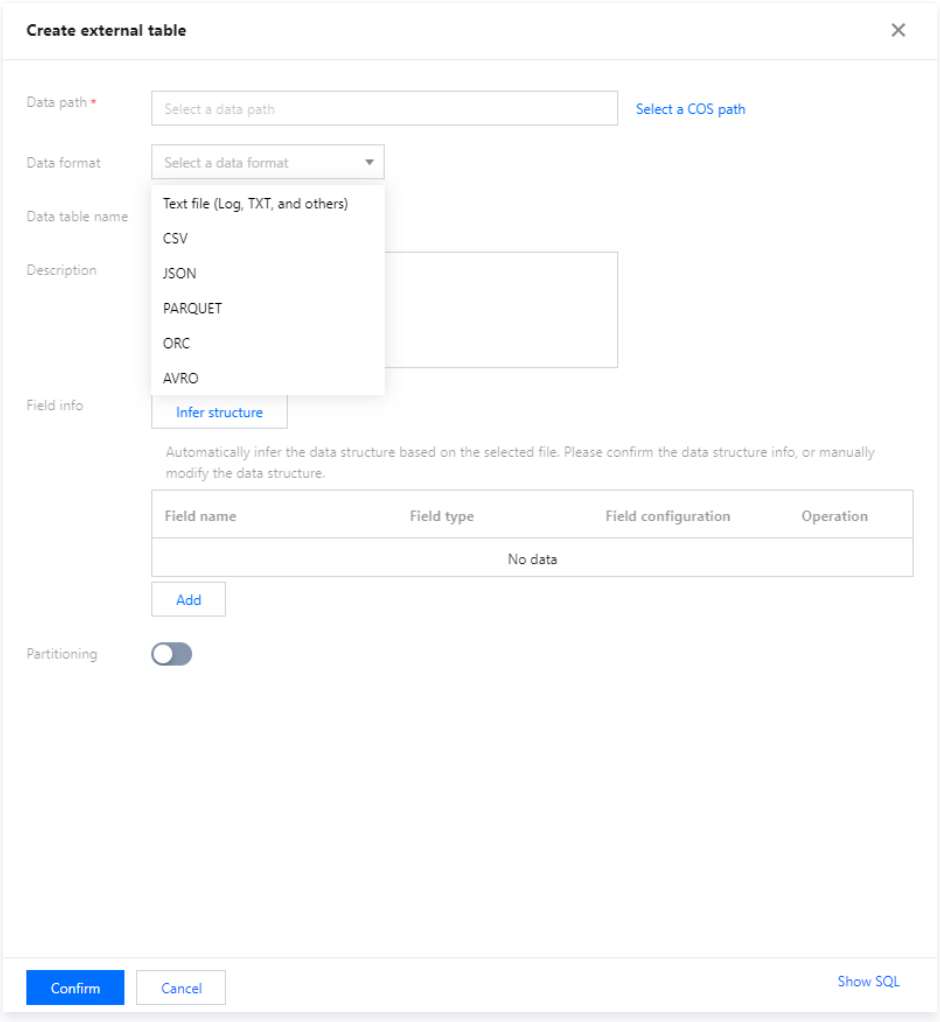
1. Log in to the Data Lake Compute DLC console and select the service region.
2. Navigate to **Data Exploration** via the left sidebar menu.
3. Select the database/table, right-click on the newly created table, and choose **Create External Table**.

> **Note**
> External tables typically refer to data files stored in your own COS bucket. Data Lake Compute can directly create external tables for analysis without the need for additional data loading. Given the characteristics of external tables, actions such as executing 'drop table' will not delete your original data in Data Lake Compute, but only the metadata of the table.

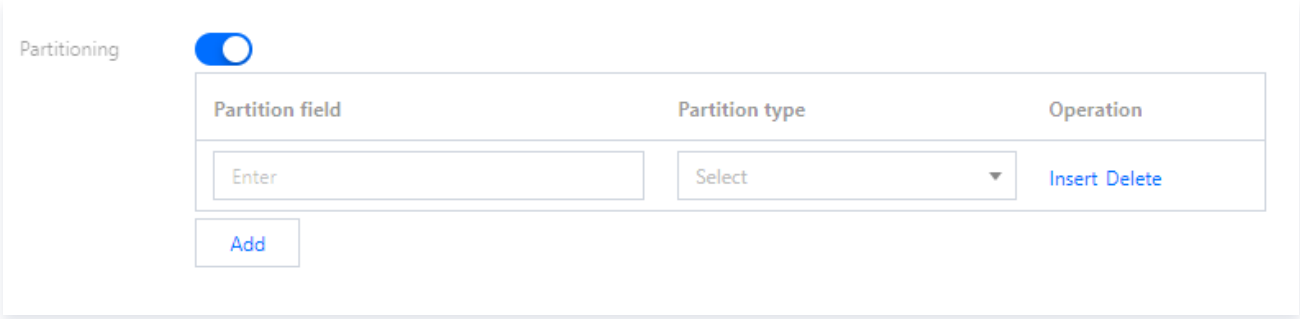


4. Follow the guide to generate the table creation statement, completing each step in the following order: **Data Path > Data Format > Data Format Configuration > Edit Partition**.
   - Step 1: Select the COS path where the data files are stored (the path must be a directory under the COS bucket, not directly to the COS bucket). A shortcut for quickly uploading files to COS is also provided here. This operation requires relevant COS permissions.
   - Currently, Data Lake Compute supports the creation of: **File, CSV, JSON, PARQUET, ORC, AVRD**

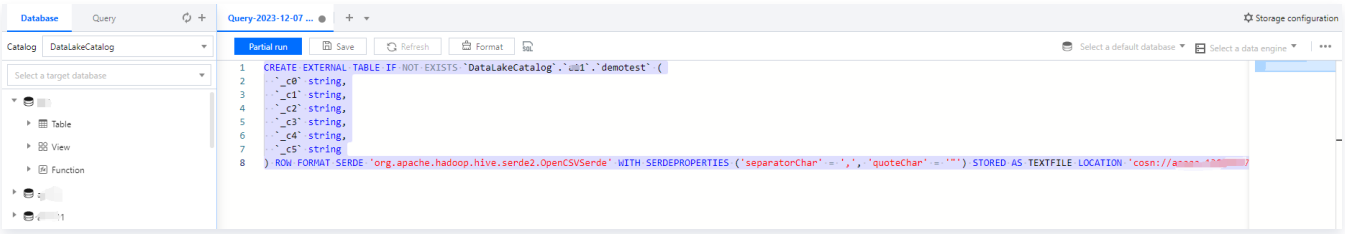
> **Note**
> Structure inference is an auxiliary tool for table creation and cannot guarantee 100% accuracy. You still need to review and verify whether the field names and types meet your expectations, and edit them to the correct information based on the actual situation.



   - Step 3: If there are no partitions, you can skip this step. Enabling partition use can reasonably enhance analysis performance. For detailed partition information, refer to Query Partition Table.

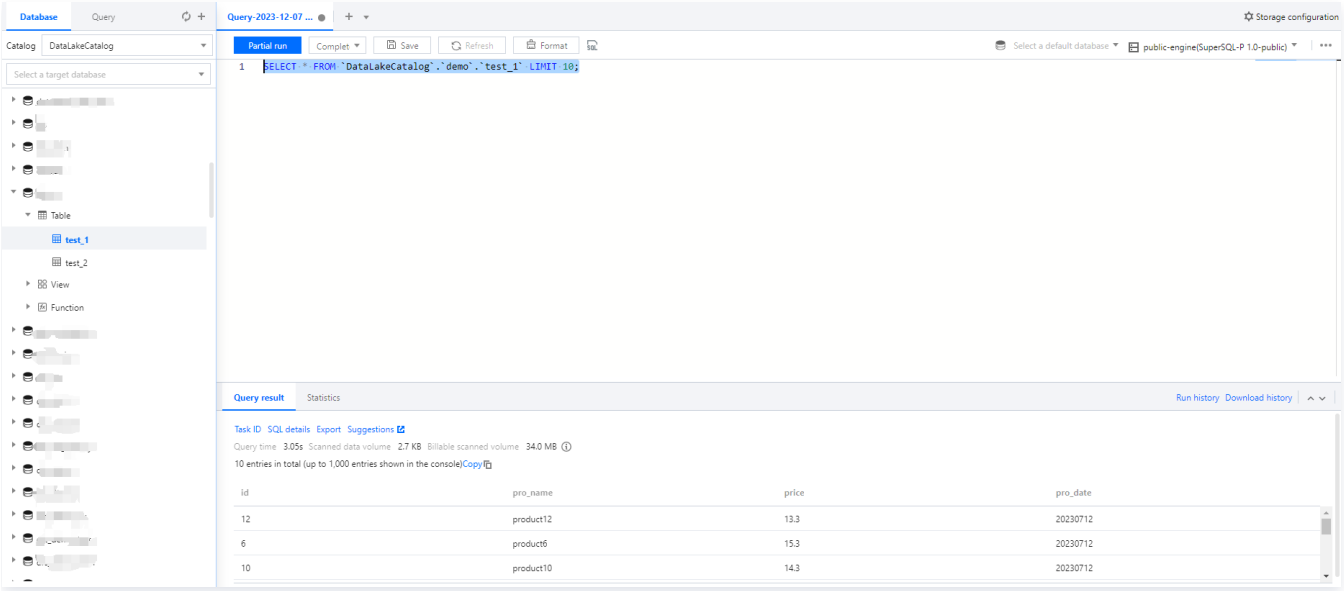


5. Click **Complete** to generate the SQL table creation statement. Execute the generated statement after selecting the data engine to complete the table creation.



Step 3: Execute SQL Analysis

After the data is prepared, write the SQL analysis statement, select an appropriate compute engine, and start data analysis.



Sample

Write a SQL statement with all data query results being `SUCCESS` and run the statement after selecting a compute engine.

```
select * from DataLakeCatalog.demo2.demo_audit_table where _c5 = 'SUCCESS'
```

The execution result is as follows:


Query result Statistics Run history Download history

Task ID SQL details Export Suggestions

Query time 1.91s Scanned data volume 2.0 KB Billable scanned volume 34.0 MB

9 entries in total (up to 1,000 entries shown in the console)Copy

id	pro_name	price	pro_date
5	product5	18.3	20230712
14	product14	13.3	20230712
12	product12	13.3	20230712
8	product8	16.3	20230712
2	product2	13.3	20230712
6	product6	15.3	20230712
10	product10	14.3	20230712
1	product1	12.3	20230712
4	product4	14.3	20230712

Quick Start with Permission Management in Data Lake Compute

Last updated: 2024-01-10 15:52:50

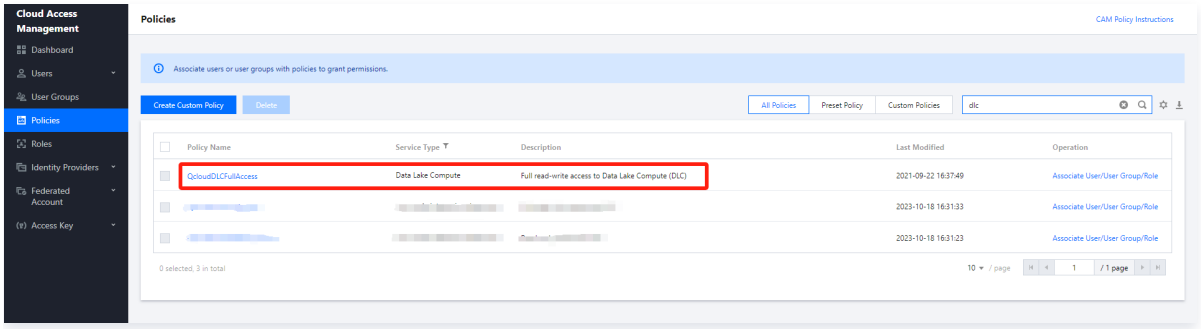
During the utilization of Data Lake Compute (DLC), if you need to establish varying access permissions for employees within your organization to achieve isolation of authority among them, you can employ the permissions management feature for meticulous management of user and workgroup permissions.

Note:

1. The policy of permissions is highly correlated with the usage of the product. It is recommended that administrators configure the policies for roles such as workgroups and sub-users in advance before officially utilizing the product features.
2. **In different regions, administrators are required to reconfigure the member management and permissions management for DLC in that specific region.**

CAM Authorization

Data Lake Compute (DLC) possesses a comprehensive data access permission mechanism. If you have sub-account management requirements, please grant the corresponding sub-account with the **QcloudDLCFullAccess (Full read-write access to Data Lake Compute (DLC))** policy in the Access Management Console. For specific steps on creating sub-accounts and authorizing policies, please refer to Creating and Authorizing Sub-accounts.
Data Lake Compute (DLC) offers permissions refined to the granularity of row and column levels in data tables, ensuring that you need not worry about overstepping authority with this operation.



Users and Workgroups

Data Lake Compute (DLC) manages user permissions through two methods: granting permissions to users and binding workgroup authorizations.

- **User**: Users in CAM, encompassing administrators, sub-accounts, and collaborator accounts.
- **Workgroup**: Data Lake Compute (DLC) allows for the binding of a group of users to a workgroup, granting the group permissions for data, engines, and other resources. This facilitates bulk management of user permissions, with users within the same workgroup possessing identical permissions.

Note

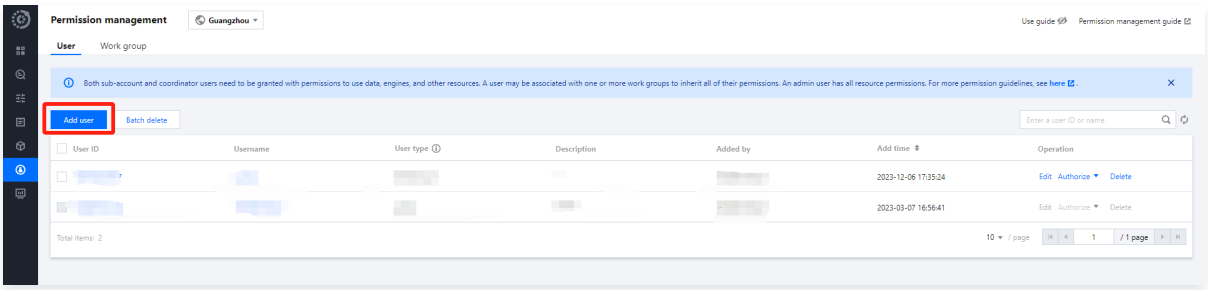
- When the permissions assigned to a user differ from those of their work group, the union of both sets of permissions is taken.
- By default, ordinary users created by the administrator do not have any permissions. They need to be added to a work group and the corresponding permission policies need to be granted to the work group in order for the users within the group to obtain the respective permissions.

Adding a User

Data Lake Compute utilizes the Tencent Cloud account ID as the default user ID. It distinguishes between two user types: administrators and ordinary users. Administrators inherently possess all resource permissions, while ordinary users must be granted specific permissions or be associated with a work group to acquire permissions.

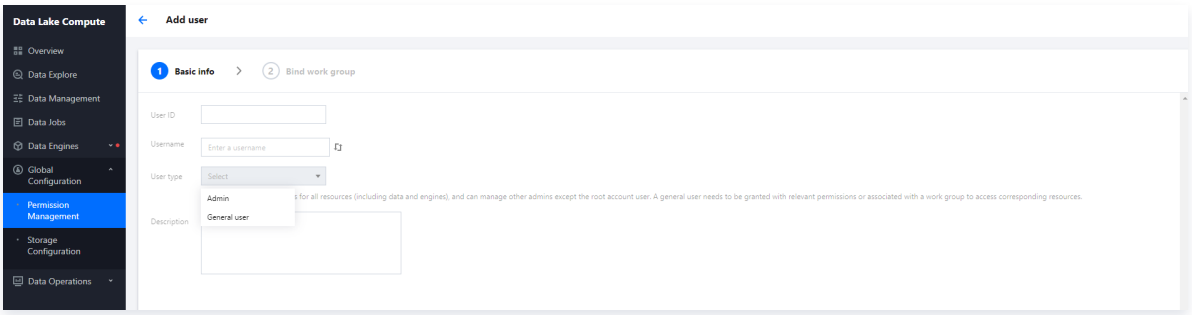
1. Incorporate a user and associate them with a work group.

   Log into the DLC console, select Permission Management, and click on **Users > Add User** to incorporate a new user.

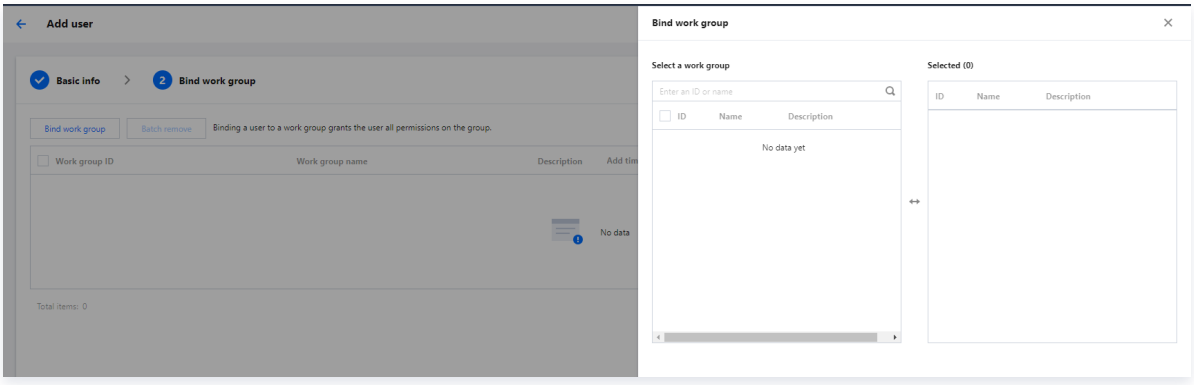


2. Enter the basic information: Provide the user ID, user name, and description, and select the user type.

> **Note:**
> When selecting the user type as "**Ordinary User**", permissions can be obtained through individual authorization or by acquiring all permissions of a specified work group. When selecting "**Administrator**" as the user type, there is no need to associate with a work group to gain all permissions.

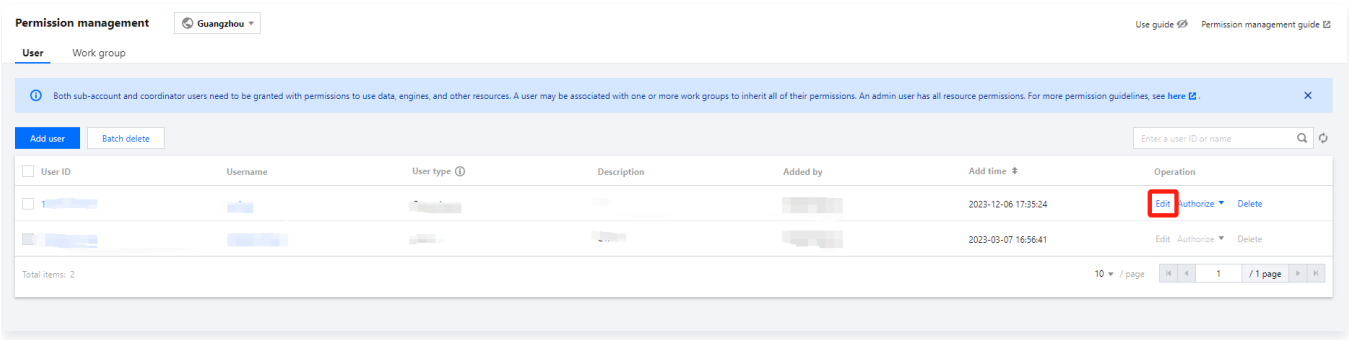


3. Associate with a work group: Select a work group for association (optional).



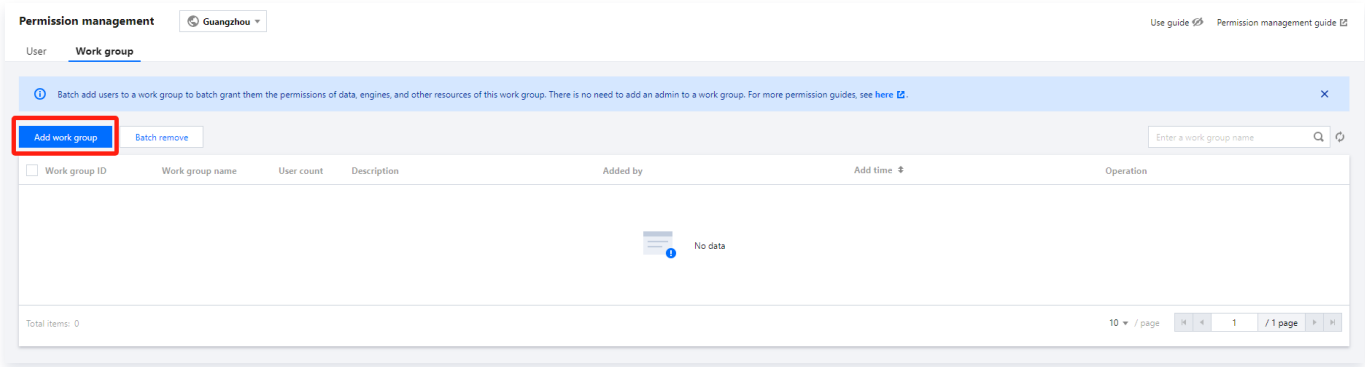
User authorization

In the user list, authorize each user individually. The authorization includes "Data Permissions" and "Engine Permissions", and the permission policy is consistent with the work group's permission policy. For more detailed operations, refer to Sub-account Permission Management.

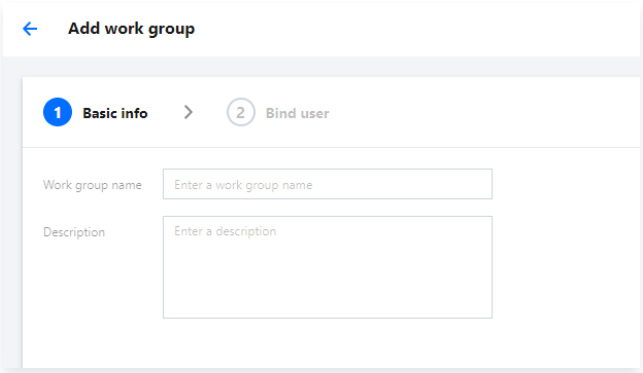


Add Work Group

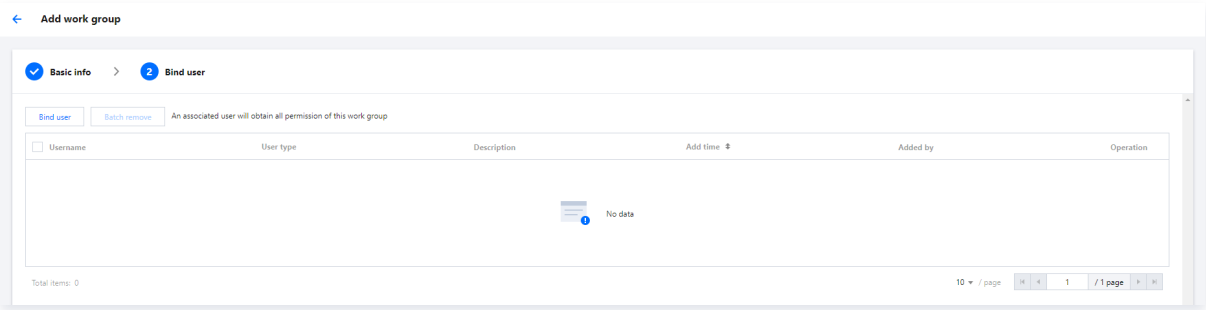
1. In the Data Lake Compute DLC, select Permission Management from the left sidebar, and click on **Work Group > Add Work Group** to create a work group for the user. When creating a work group, you can choose to bind it to a user or create an empty work group. For detailed operations, refer to Users and User Groups.



2. Enter the basic information: Provide the work group name and description.

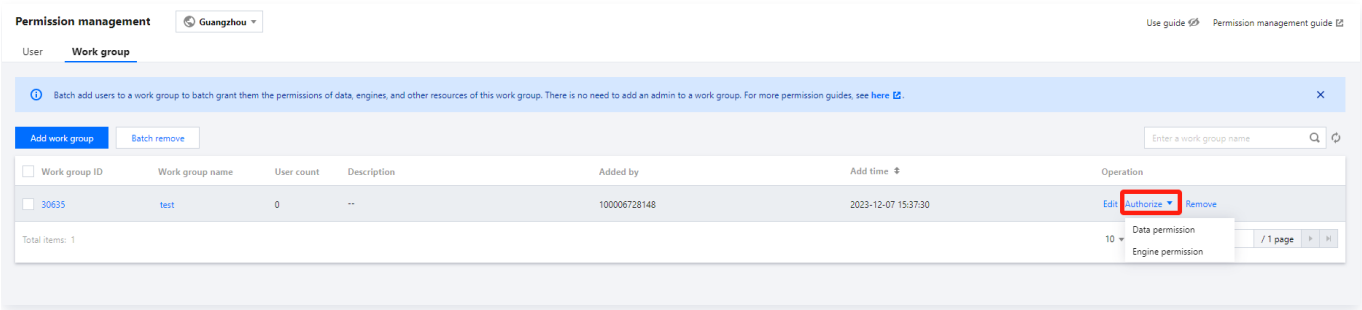


3. Associate a user: The associated user will acquire all permissions under the respective work group.



Granting permissions to a work group.

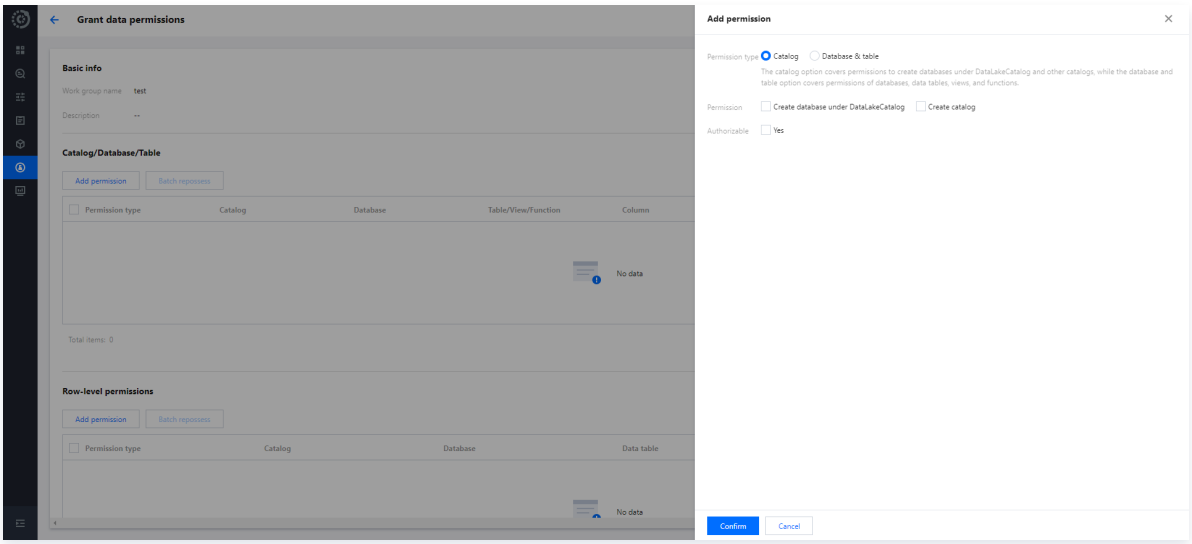
After creating the work group, click on the **Authorize** operation in the list to add permissions to the work group, including **Data Permissions** and **Engine Permissions**.



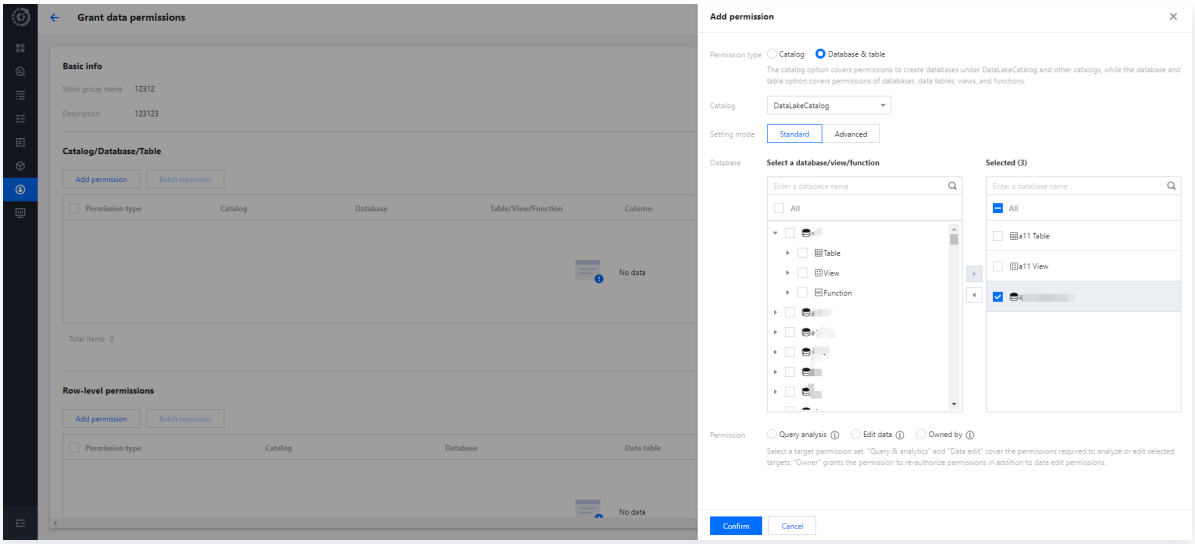
Data permission

Data permissions include:

- Data Catalog Permissions: These include two types of permissions under the data catalog, namely, the ability to **Create Database** and **Create Data Catalog**.

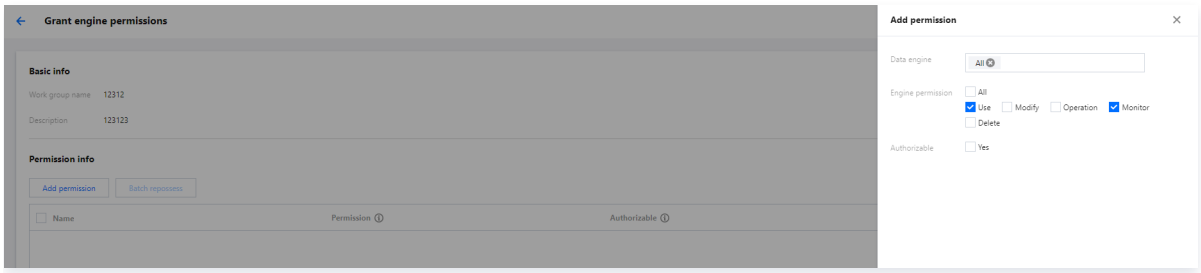


- Database Table Permissions: Fine-grained permissions at the database table level can be granted, including query and edit permissions for databases, tables, views, and functions.



Engine permission

Select a data engine and grant the permissions to use, modify, or delete it.



Quick Start with Partition Table

Last updated: 2024-01-10 15:53:16

Partitioned Table in Data Lake Compute

With the partition catalog feature, you can store data with different characteristics in different catalogs. In this way, when exploring data, you can filter data by partition through the `where` condition. This greatly reduces the scanned data volume and improves the query efficiency.

> **Note**
> - Partitions within the same table should utilize the same data type and format.
> - **Data Lake Compute's native tables** implement implicit partitioning, allowing you to **disregard the partition directory structure.**

Creating a Partitioned Table

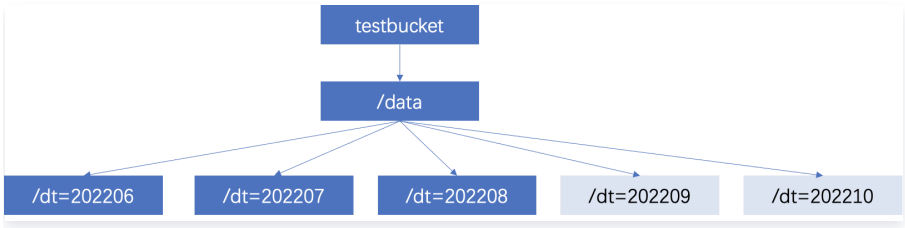
Specify the partition field through the `PARTITIONED BY` parameter in the table creation statement.
Example: Creating the `test_part` partition table

```
CREATE EXTERNAL TABLE IF NOT EXISTS DataLakeCatalog.test_a_db.test_part (
_c0 int,
_c1 int,
_c2 string,
dt string
) USING PARQUET PARTITIONED BY (dt) LOCATION 'cosn://testbucket/data/';
```

Add Partition

Add partitions using the `alter table add partition` command.

If your data partition catalog uses the Hive partitioning rule (partition column name=partition column value), the rule can be used to add partitions. The catalog is organized as follows:




```
ALTER TABLE DataLakeCatalog.test_a_db.test_part add PARTITION (dt = '202206')
ALTER TABLE DataLakeCatalog.test_a_db.test_part add PARTITION (dt = '202207')
ALTER TABLE DataLakeCatalog.test_a_db.test_part add PARTITION (dt = '202208')
ALTER TABLE DataLakeCatalog.test_a_db.test_part add PARTITION (dt = '202209')
ALTER TABLE DataLakeCatalog.test_a_db.test_part add PARTITION (dt = '202210')
```

Add partitions by specifying the location with the `alter table` command.

If your data adopts a general COS catalog (not in the "partition column name=partition column value" format), you can specify a catalog when adding a partition.

Sample SQL:

```
ALTER TABLE DataLakeCatalog.test_a_db.test_part add PARTITION (dt = '202211') LOCATION
'cosn://testbucket/data2/202211'
ALTER TABLE DataLakeCatalog.test_a_db.test_part add PARTITION (dt = '202212') LOCATION
'cosn://testbucket/data2/202212'
```

Utilizing MSCK REPAIR for Automatic Partition Addition

By using the `MSCK REPAIR TABLE` statement, the system scans the data directory specified during table creation. If new partition directories exist, the system will automatically add these partitions to the metadata information of the data table.
SQL Reference:

```
MSCK REPAIR TABLE DataLakeCatalog.test_a_db.test_part
```

We recommend adding partitions primarily through the 'alter table' method. If you choose to use 'msck repair' for automatic partition addition, the following constraints apply:

- The MSCK REPAIR TABLE command only adds partitions to the table metadata, it does not delete partitions.
- When dealing with large volumes of data, it is not recommended to use the MSCK REPAIR TABLE method, as it scans the entire data volume and may lead to timeouts.
- If the partition directory does not follow Hive's partitioning rule: partition column name = partition column value, the MSCK REPAIR TABLE method cannot be used.

Cross-Source Analysis of EMR Hive Data

Last updated: 2024-01-10 15:53:22

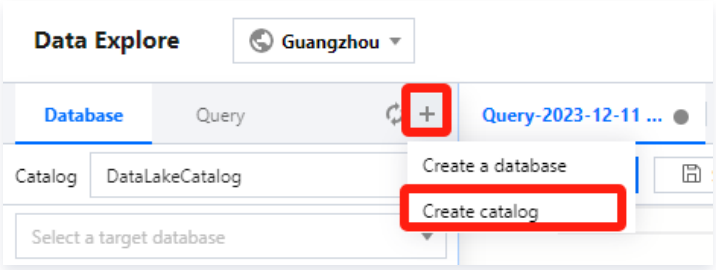
Data Lake Compute allows you to configure an EMR Hive data source for multi-source federated data analysis.

Preparations

- Acquire the EMR Hive address.
- Use an account with the authority to create data directories. For detailed permissions, refer to DLC Permission Overview.

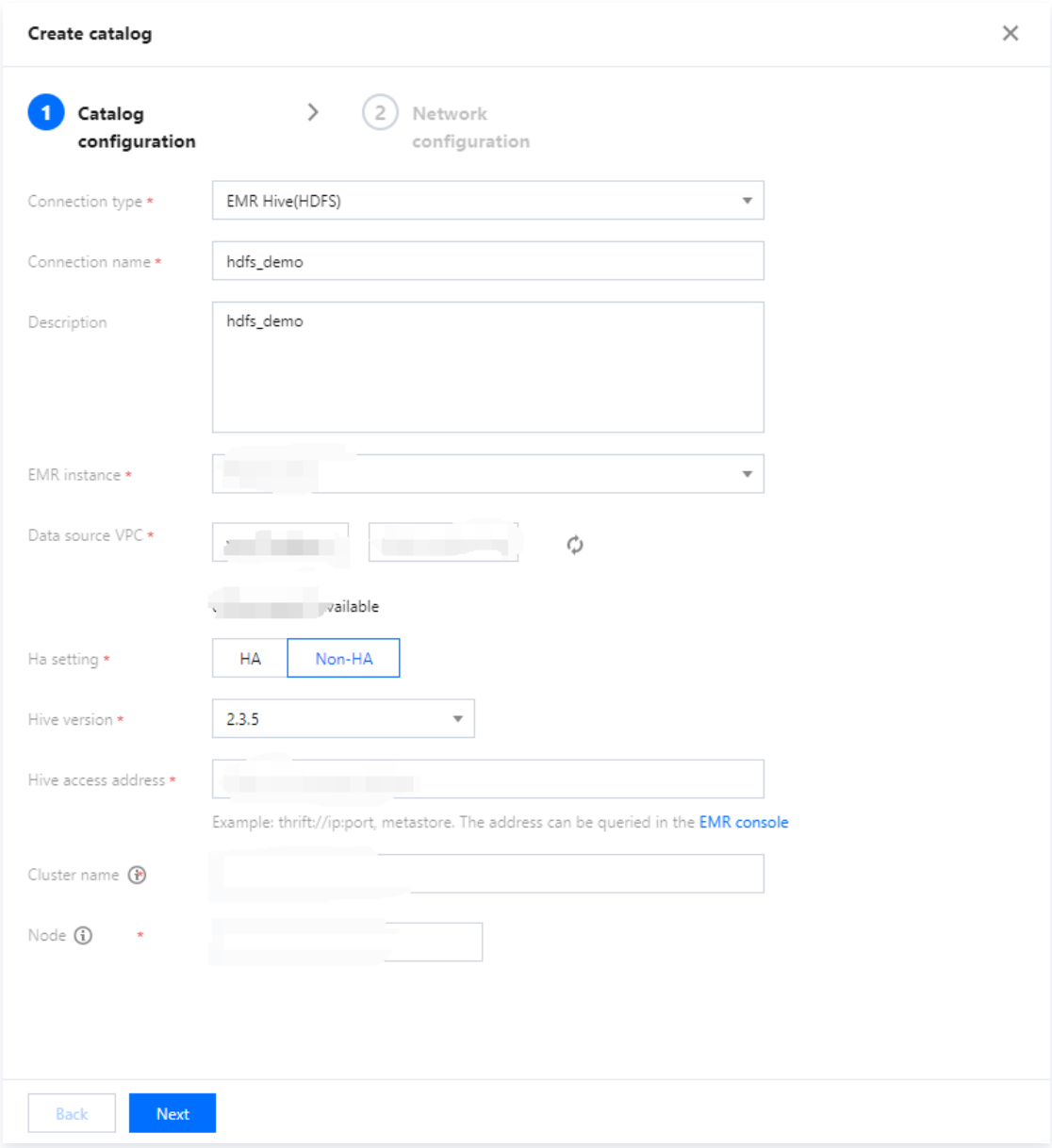
Create an EMR Hive Data Source

1. Log in to the Data Lake Compute DLC console and select the service region.
2. Navigate to **Data Exploration** via the left sidebar, click the +button in the library table column, and select **Create catalog**.



3. Select the connection type as EMR Hive (HDFS), choose the corresponding EMR instance, and the VPC information will be automatically filled in after the instance is selected. **EMR Hive supports the following EMR versions: 2.3.5, 2.3.7, 3.1.1, 3.1.2.**

> **Note**
> You must have the relevant permissions for the EMR Hive instance to make a selection.



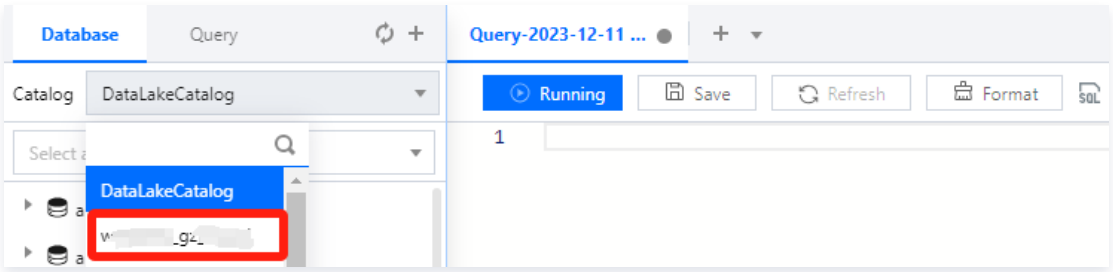
4. Select the running cluster. Currently, only Presto private data engines are available. If there is no corresponding engine, you can create a data engine on the data engine page. For the purchase process, please refer to Purchasing a Private Data Engine.

> **Note**
> The selected data engine IP range must not coincide with the EMR instance IP range, as this would lead to network conflicts and hinder data query analysis.

5. Click **Confirm** to complete the creation of the data directory.

Query EMR Hive Data

Once the data directory is created, you can switch data directories from the data directory menu on the **Data Exploration page**.



At this point, you can use SQL statements to query and analyze the data directory. For SQL syntax, please refer to SQL Syntax Overview.
Select the data engine bound when creating the data directory and click the **Run** button to obtain the query results.

> **Note**
> Only the associated data engine can query this data directory, other data engines will not be able to perform queries. If you need to change the associated engine, you can click the settings button next to the data directory to edit and modify.