

Data Lake Insight

Service Overview

Issue 01
Date 2025-01-09



Copyright © Huawei Technologies Co., Ltd. 2025. All rights reserved.

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

Trademarks and Permissions



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

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

Notice

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

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

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process*. For details about this process, visit the following web page:

<https://www.huawei.com/en/psirt/vul-response-process>

For vulnerability information, enterprise customers can visit the following web page:

<https://securitybulletin.huawei.com/enterprise/en/security-advisory>

Contents

1 Infographics.....	1
2 What Is Data Lake Insight.....	3
3 Advantages.....	8
4 Application Scenarios.....	11
5 Notes and Constraints.....	17
6 Specifications.....	23
7 Security.....	25
7.1 Shared Responsibilities.....	25
7.2 Asset Identification and Management.....	26
7.3 Identity Authentication and Access Control.....	27
7.4 Data Protection Technologies.....	28
7.5 Audit and Logging.....	32
7.6 Service Resilience.....	32
7.7 Security Risk Monitoring.....	33
7.8 Recovery from Failures.....	33
7.9 Update Management.....	33
7.10 Certificates.....	34
8 Permissions Management.....	36
9 Quotas.....	44
10 Related Services.....	46
11 Basic Concepts.....	49

1 Infographics

Huawei Cloud Data Lake Insight

01 Data Lake Challenges

It is difficult to query data from numerous data sources.

Professional analysts with expertise in cluster management and operations are needed.

Compute resources cannot be quickly expanded to keep up with rapid data growth.

The query rate cannot be guaranteed for large amounts of data.

The query and analysis service must be ready for use.

02 Data Lake Insight Makes Cloud Data Processing and Analysis More Convenient

DLI is a fully hosted big data processing and analysis service that frees you from ETL. You can use SQL or Spark programs to gain insight into heterogeneous data on Huawei Cloud.

- 1. Ready-for-Use >>**
You don't need to handle any infrastructure.
- 2. Easy-to-Use >>**
DLI supports mainstream data formats and multiple access modes, allowing you to run SQL and Spark applications.
- 3. Cross-source Data Query >>**
You can directly use SQL to query large-scale detailed data, cached data, and high-order processing data.
- 4. Enterprise Multi-Tenant >>**
DLI helps you isolate resources and control data permission.
- 5. Query Within Seconds >>**
The CarbinData storage technology accelerates local storage, ensuring queries are responded to in seconds.

03 Application Scenarios

- Historical data analysis across sources**
Scenario Description: Directly query a large amount of raw data across a wide time span in various query dimensions.
Typical Applications: transaction auditing, driving behavior analysis, and track playback.
- Joint analysis of heterogeneous data across sources**
Scenario Description: Analyze data association in different storage systems with complex service logic.
Typical Applications: parallel comparison analysis of cold and hot data, and joint analysis of data on OBS, CloudTable, GaussDB(DWS), and RDS.
- Interactive analysis from multiple dimensions**
Scenario Description: Query data with low latency in fixed dimensions to support operational decision-making.
Typical Applications: BI analysis, user retention rate analysis, user classification preference analysis, and applause rate analysis.
- Large-scale log analysis**
Scenario Description: Directly query a large amount of raw data across a wide time span in various query dimensions.
Typical Applications: driving behavior analysis, data analysis of ad operations, and financial market analysis.

04 Success Story — Transaction Data Analysis

TB of transaction data is generated each day by over 3000 listed companies and approximately 120 million registered shareholders. This data is stored in various formats and devices and managed by different systems. A financial company uses DLI to analyze and process data in different storage and formats, improving the analysis efficiency. The company reduced development costs by 70%.

Data sources	Data ingestion	Storage, processing, and analysis	Applications
General information	Periodical collection of general data	DLI Flink, CloudTable	Offline query, Offline analytics, Offline correlation chart connection
Transaction data	Real-time data	DLI Flume, OBS, DLI	Real-time stock transaction and precise data analysis
User data	Real-time data	RDS for MySQL	Real-time stock transaction and precise data analysis

Customer Benefits:

- Reduced O&M costs
- Reduced development costs for heterogeneous data query and analysis and quicker monetization

Lower O&M costs Reduced development costs

2 What Is Data Lake Insight

DLI Introduction

Data Lake Insight (DLI) is a serverless data processing and analysis service fully compatible with [Apache Spark](#), HetuEngine, and [Apache Flink](#) ecosystems. It frees you from managing any servers.

DLI supports standard SQL and is compatible with Spark SQL and Flink SQL. It also supports multiple access modes, and is compatible with mainstream data formats. You can use standard SQL or Spark and Flink applications to query mainstream data formats without data ETL. DLI supports SQL statements and Spark applications for heterogeneous data sources, including [RDS](#), [GaussDB\(DWS\)](#), [CSS](#), [OBS](#), custom databases on [EC2s](#), and offline databases.

Functions

You can query and analyze heterogeneous data sources such as CloudTable, RDS, and GaussDB(DWS) on the cloud using access methods, such as visualized interface, RESTful API, JDBC, and Beeline. The data format is compatible with five mainstream data formats: CSV, JSON, Parquet, and ORC.

- Basic functions
 - You can use standard SQL statements to query in SQL jobs. For details, see [Spark SQL Syntax Reference](#).
 - Flink jobs support Flink SQL online analysis capabilities: supporting aggregation functions such as Window and Join, using SQL to express service logic, and achieving service implementation conveniently and quickly. For details, see [Flink OpenSource SQL Syntax Reference](#).
 - For spark jobs, fully-managed Spark computing can be performed. You can submit computing tasks through interactive sessions or in batch to analyze data in the fully managed Spark queues. For details, see [SQL Syntax Constraints and Definitions](#).
- Federated analysis of heterogeneous data sources
 - Spark datasource connection: Data sources such as CloudTable, GaussDB(DWS), RDS, and CSS can be accessed through DLI. For details, see [Enhanced Datasource Connections](#).

- Interconnection with multiple cloud services is supported in Flink jobs to form a rich stream ecosystem. The DLI stream ecosystem consists of cloud service ecosystems and open source ecosystems.
 - Cloud service ecosystem: DLI can interconnect with other services in Flink SQL. You can directly use SQL to read and write data from cloud services, such as DIS, OBS, CloudTable, MRS, RDS, SMN and DCS.
 - Open-source ecosystem: By establishing network connections with other VPCs through enhanced datasource connections, you can access all Flink and Spark-supported data sources and output sources, such as Kafka, Hbase, Elasticsearch, in the tenant-authorized DLI queues.

For details, see [Flink Jobs](#).

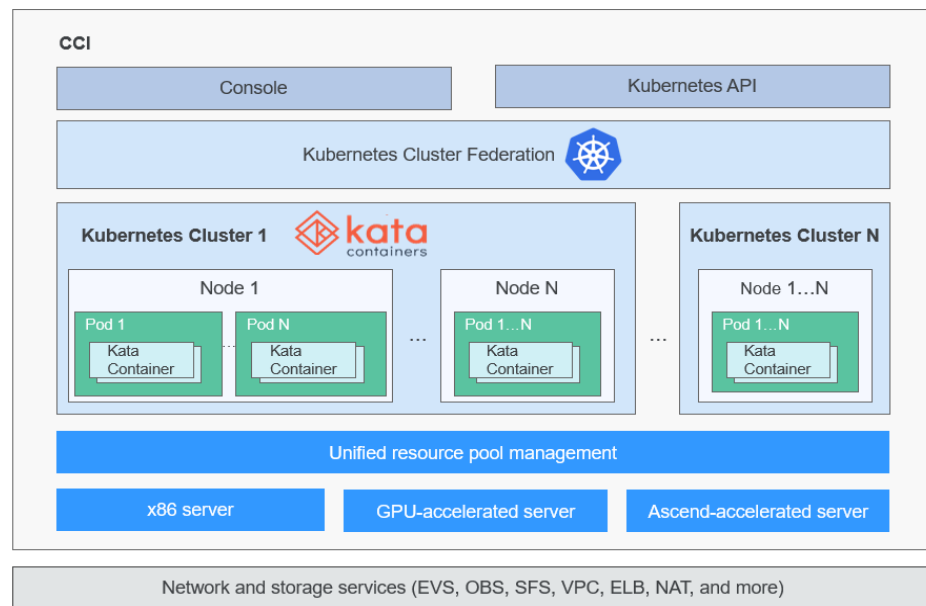
- Storage-compute decoupling

DLI is interconnected with OBS for data analysis. In this architecture where storage and compute are decoupled, resources of these two types are charged separately, helping you reduce costs and improving resource utilization.

You can choose single-AZ or multi-AZ storage when you create an OBS bucket for storing redundant data on the DLI console. The differences between the two storage policies are as follows:

 - Multi-AZ storage means data is stored in multiple AZs, improving data reliability. If the multi-AZ storage is enabled for a bucket, data is stored in multiple AZs in the same region. If one AZ becomes unavailable, data can still be properly accessed from the other AZs. The multi-AZ storage is ideal for scenarios that demand high reliability. You are advised to use this policy.
 - Single-AZ storage means that data is stored in a single AZ, with lower costs.
- Elastic resource pool

Elastic resource pools support the CCE cluster architecture for heterogeneous resources so you can centrally manage and allocate them. For details, see [Elastic Resource Pool](#).

Figure 2-1 Elastic resource pool architecture

Elastic resource pools have the following advantages:

- **Unified management**
 - You can manage multiple internal clusters and schedule jobs. You can manage millions of cores for compute resources.
 - Elastic resource pools can be deployed across multiple AZs to support high availability.
 - **Tenant resource isolation**

Resources of different queues are isolated to reduce the impact on each other.
 - **Shared access and flexibility**
 - Minute-level scaling helps you to handle request peaks.
 - Queue priorities and CU quotas can be set at different time to improve resource utilization.
 - **Job-level isolation (supported in later versions)**

SQL jobs can run on independent Spark instances, reducing mutual impacts between jobs.
 - **Automatic scaling (supported in later versions)**

The queue quota is updated in real time based on workload and priority.
- Using elastic resource pools has the following advantages.

Advantage	No Elastic Resource Pool	Use Elastic Resource Pool
Efficiency	You need to set scaling tasks repeatedly to improve the resource utilization.	Dynamic scaling can be done in seconds.
Resource utilization	Resources cannot be shared among different queues. For example, a queue has idle CUs and another queue is heavily loaded. Resources cannot be shared. You can only scale up the second queue.	Queues added to the same elastic resource pool can share compute resources.
	When you set a data source, you must allocate different network segments to each queue, which requires a large number of VPC network segments.	You can add multiple general-purpose queues in the same elastic resource pool to one network segment, simplifying the data source configuration.
Resource allocation	If resources are insufficient for scale-out tasks of multiple queues, some queues will fail to be scaled out.	You can set the priority for each queue in the elastic resource pool based on the peak hours to ensure proper resource allocation.

- BI tool
Interconnection with Yonghong BI for data analysis.

DLI Core Engine: Spark+Flink+Trino+HetuEngine

- Spark is a unified analysis engine that is ideal for large-scale data processing. It focuses on query, compute, and analysis. DLI optimizes performance and reconstructs services based on open-source Spark. It is compatible with the Apache Spark ecosystem and interfaces, and improves performance by 2.5x when compared with open-source Spark. In this way, DLI enables you to perform query and analysis of EB's of data within hours.
- Flink is a distributed compute engine that is ideal for batch processing, that is, for processing static data sets and historical data sets. You can also use it for stream processing, that is, processing real-time data streams and generating data results in real time. DLI enhances features and security based on the open-source Flink and provides the Stream SQL feature required for data processing.

- HetuEngine is an open source SQL query engine that allows for interactive query and analysis. It excels in quickly and efficiently processing large-scale data queries and analyses with low latency.

Serverless Architecture

DLI is a serverless big data query and analysis service. It has the following advantages:

- Pay-per-use: You pay only for what you use (scanned data volume/CUH packages). When no jobs are running, you will not be billed.
- Auto scaling: DLI ensures you always have enough capacity on hand to deal with any traffic spikes.

Accessing DLI

A web-based service management platform is provided. You can access DLI using the management console or HTTPS-based APIs, or connect to the DLI server through the JDBC client.

- Using the management console
You can submit SQL, Spark, or Flink jobs on the DLI management console. Log in to the management console. Choose **EI Enterprise Intelligence > Data Lake Insight** from the service list.
- Using APIs
If you need to integrate DLI into a third-party system for secondary development, you can call DLI APIs to use the service. For details, see [Data Lake Insight API Reference](#).
- JDBC
You can use JDBC to connect to the server for data query. For details, see [Obtaining the Server Connection Address](#).
- Spark-submit
Jobs can be submitted using Spark-submit. For details, see [Using Spark-submit to Submit a Spark Jar Job](#).

3 Advantages

Full SQL Compatibility

You do not need a background in big data to use DLI for data analysis. You only need to know SQL, and you are good to go. The SQL syntax is fully compatible with the standard ANSI SQL 2003.

Decoupled Storage and Compute

DLI compute and storage loads are decoupled. This architecture allows you to flexibly configure storage and compute resources on demand, improving resource utilization and reducing costs.

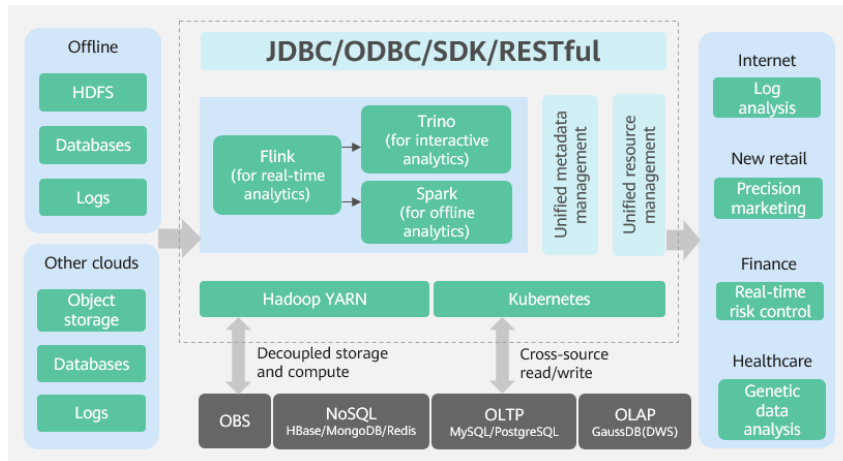
Enterprise Multi-Tenancy

You can manage compute or resource related permissions by project or user, and implement fine-grained control to isolate data for each task.

Serverless DLI

DLI is fully compatible with [Apache Spark](#) and [Apache Flink](#) ecosystems and APIs. It is a serverless big data computing and analysis service that integrates real-time, offline, and interactive analysis. Offline applications can be seamlessly migrated to the cloud, reducing the migration workload. DLI provides a highly-scalable framework integrating batch and stream processing, allowing you to handle data analysis requests with ease. With a deeply optimized kernel and architecture, DLI delivers 100-fold performance improvement compared with the MapReduce model. Your analysis is backed by an industry-vetted 99.95% SLA.

Figure 3-1 DLI serverless architecture



DLI has the following advantages over self-built Hadoop clusters:

Table 3-1 Advantages comparison

Advantage	Dimension	Data Lake Insight	Self-built Hadoop
Low cost	Capital cost	Billing is based on the actual amount of data scanned or used CUH. Saving up to 50% costs.	Long-term resource occupation, causing severe resource waste and high costs
	Elastic scalability	Container-based Kubernetes, intelligent elastic scaling	Not supported.
O&M free	O&M cost	Out-of-the-box, serverless architecture	Strong technical capabilities are required for configuration and O&M
	High availability	Cross-AZ DR	N/A
Easy to use	Learning cost	Low. The optimization parameters are standardized based on 10 years' experience in thousands of projects. In addition, DLI provides a GUI for intelligent optimization.	High. Hundreds of tuning parameters need to be learned.

Advantage	Dimension	Data Lake Insight	Self-built Hadoop
	Supported data sources	<ul style="list-style-type: none"> Cloud: OBS, RDS, GaussDB(DWS), CSS, MongoDB, and Redis On-premises: self-built databases, MongoDB, and Redis 	<ul style="list-style-type: none"> Cloud: OBS On-premises: HDFS
	Ecosystem compatibility	DLV, Yonghong BI, and Fanruan BI	Big data ecosystem tool
	Custom image	Supported. Dependencies can be added as required to meet service diversity requirements.	Not supported.
	Workflow scheduling	Scheduling between Data Lake Factory (DLF) and DataArts Studio	Self-built scheduling tools, such as Airflow
	Multiple enterprise-level tenants	Table-based permission management, providing column level permission granularity.	File-based permission management
High performance	Performance	Higher performance with in-depth software and hardware optimization	Performance is the same as that of Hadoop open-source versions

Cross-Source Analysis

Analyze your data across databases. No migration required. A unified view of your data gives you a comprehensive understanding of your data and helps you innovate faster. There are no restrictions on data formats, cloud data sources, or whether the database is created online or off.

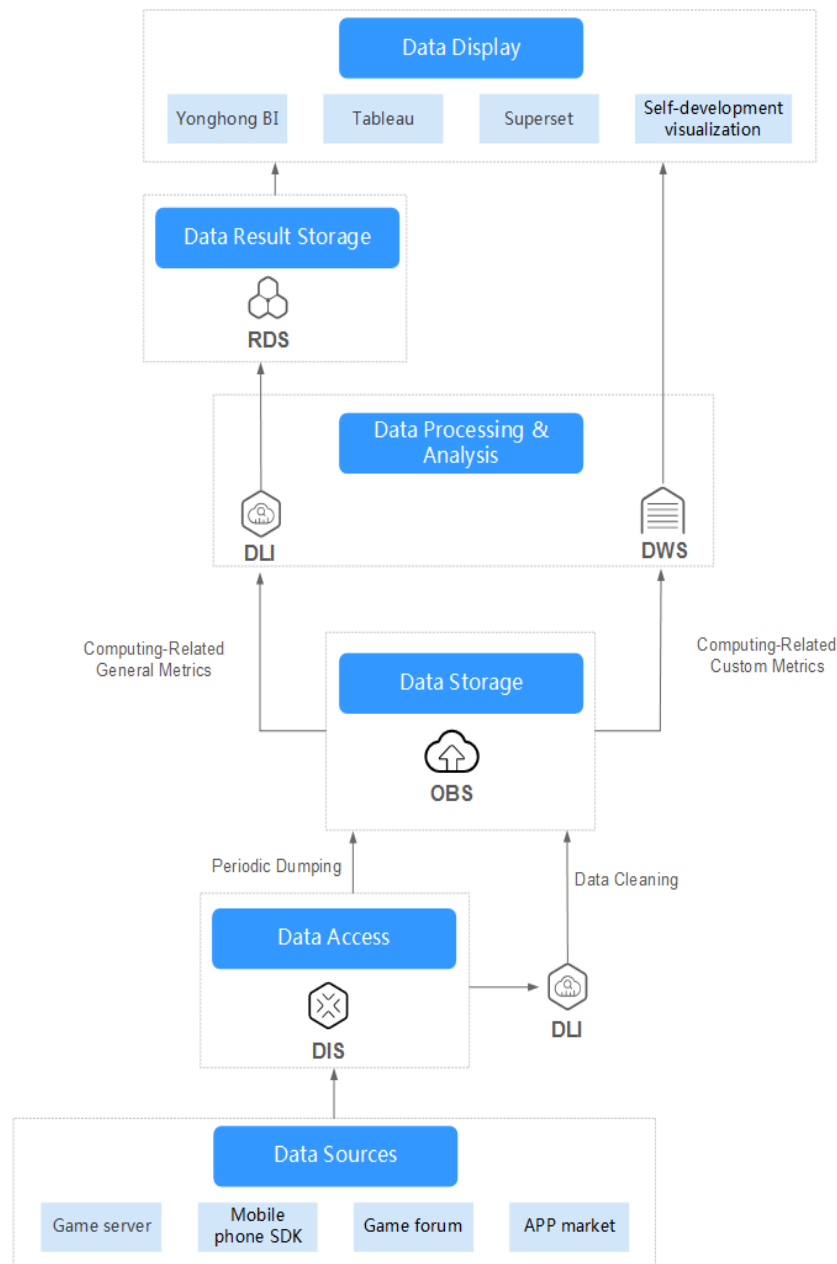
4 Application Scenarios

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

Large-scale Log Analysis

- Gaming operations data analysis
Different departments of a game company analyze daily new logs via the game data analysis platform to obtain required metrics and make decision based on the obtained metric data. For example, the operation department obtains required metric data, such as new players, active players, retention rate, churn rate, and payment rate, to learn the current game status and determine follow-up actions. The placement department obtains the channel sources of new players and active players to determine the platforms for placement in the next cycle.
- Advantages
 - Efficient Spark programming model: DLI directly ingests data from DIS and performs preprocessing such as data cleaning. You only need to edit the processing logic, without paying attention to the multi-thread model.
 - Ease of use: You can use standard SQL statements to compile metric analysis logic without paying attention to the complex distributed computing platform.
 - Pay-per-use: Log analysis is scheduled periodically based on time-critical requirements. There is a long idle period between every two scheduling operations. DLI adopts the pay-per-use billing mode, which saves the cost by more than 50% compared with the dedicated queue mode. DLI only bills you for the resources used for scheduling.
- It is recommended that you use the following related services: OBS, DIS, GaussDB(DWS), and RDS

Figure 4-1 Gaming operations data analysis



Federated Analysis of Heterogeneous Data Sources

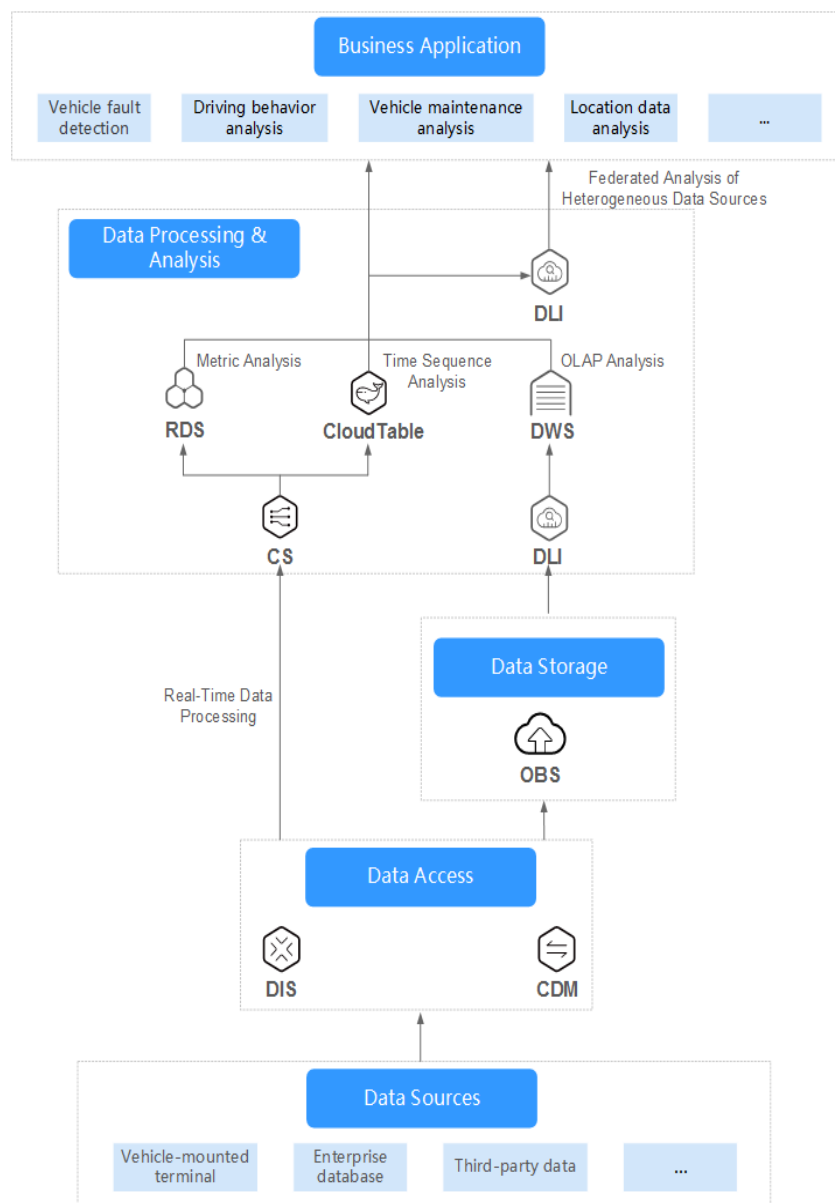
- Digital service transformation for car companies

In the face of new competition pressures and changes in travel services, car companies build the IoV cloud platform and IVI OS to streamline Internet applications and vehicle use scenarios, completing digital service transformation for car companies. This delivers better travel experience for vehicle owners, increases the competitiveness of car companies, and promotes sales growth. For example, DLI can be used to collect and analyze daily vehicle metric data (such as batteries, engines, tire pressure, and airbags), and give maintenance suggestions to vehicle owners in time.

- Advantages

- No need for migration in multi-source data analysis: RDS stores the basic information about vehicles and vehicle owners, table store CloudTable saves real-time vehicle location and health status, and GaussDB(DWS) stores periodic metric statistics. DLI allows federated analysis on data from multiple sources without data migration.
- Tiered data storage: Car companies need to retain all historical data to support auditing and other services that require infrequent data access. Warm and cold data is stored in OBS and frequently accessed data is stored in CloudTable and GaussDB(DWS), reducing the overall storage cost.
- Rapid and agile alarm triggering: There are no special requirements for the CPU, memory, hard disk space, and bandwidth.
- It is recommended that you use the following related services: DIS, CDM, OBS, GaussDB(DWS), RDS, and CloudTable

Figure 4-2 Digital service transformation for car companies

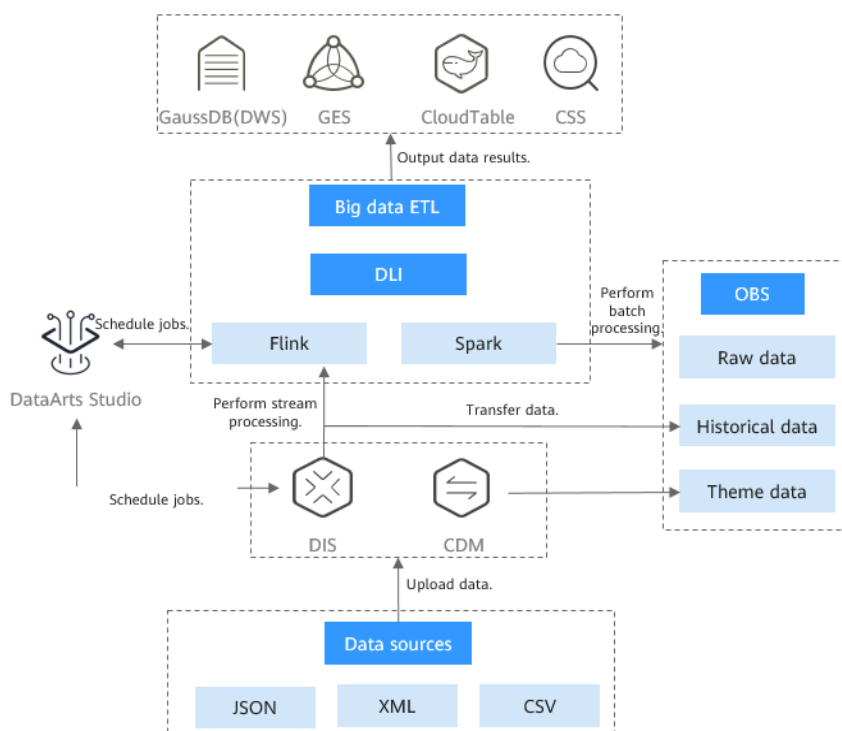


Big Data ETL Processing

- Carrier big data analysis

Carriers typically require petabytes, or even exabytes of data storage, for both structured (base station details) and unstructured (messages and communications) data. They need to be able to access the data with extremely low data latency. It is a major challenge to extract value from this data efficiently. DLI provides multi-mode engines such as batch processing and stream processing to break down data silos and perform unified data analysis.
- Advantages
 - Big data ETL: You can enjoy TB to EB-level data governance capabilities to quickly perform ETL processing on massive carrier data. Distributed datasets are provided for batch processing.
 - High Throughput, Low Latency: DLI uses the Dataflow model of Apache Flink, a real-time computing framework. High-performance computing resources are provided to consume data from your created Kafka, DMS Kafka, and MRS Kafka clusters. A single CU processes 1,000 to 20,000 messages per second.
 - Fine-grained permissions management: Your company may have numerous departments, where data needs to be shared and isolated. Using DLI, you can apply for resource queues by tenant to isolate computing resources (CPUs and memory), ensuring job SLA. DLI supports table- or column-level data permission control, allowing for secure access for different departments.
- It is recommended that you use the following related services: OBS, DIS, and DataArts Studio

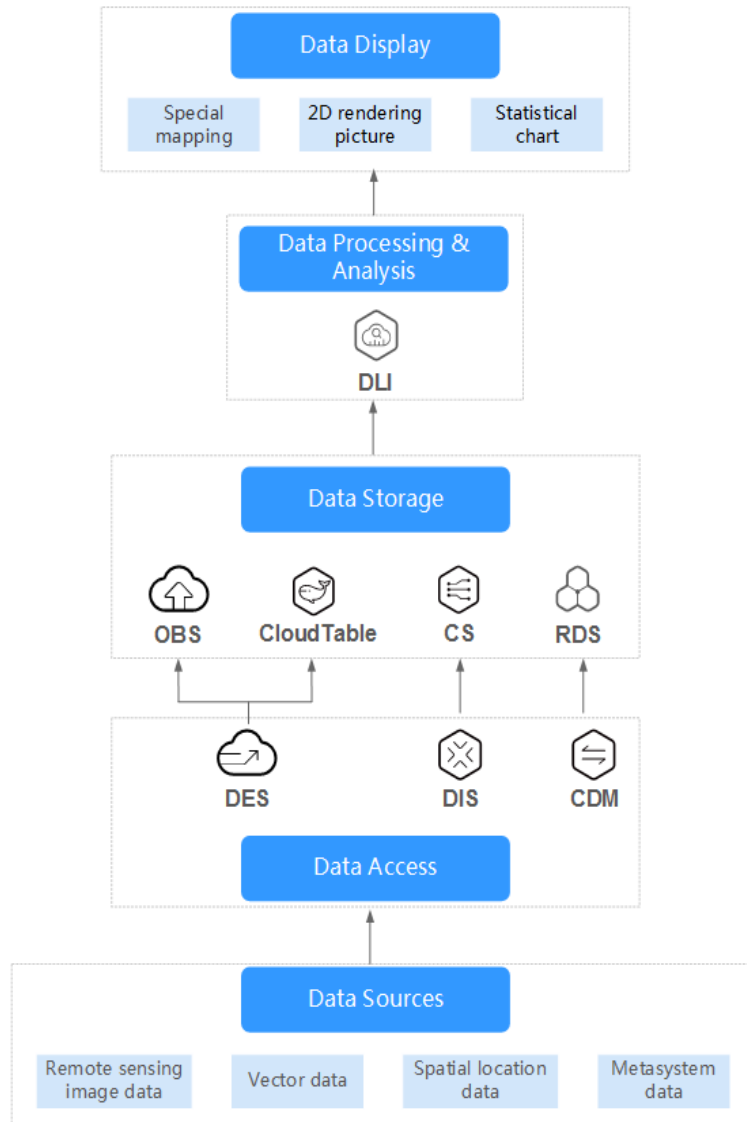
Figure 4-3 Carrier big data analysis



Geographic Big Data Analysis

- **Geographic Big Data Analysis**
Geographic big data usually has a large data volume. For example, global satellite remote sensing images might take up to petabytes of data. Besides, there are various types of data, including structured remote sensing image grid data, vector data, unstructured spatial location data, and 3D modeling data. For this scenario, efficient mining tools or methods are essential.
- **Advantages**
 - **Spatial Data Analysis Operators:** With full-stack Spark capabilities and rich Spark spatial data analysis Spatial Data Analysis Operators With full-stack Spark capabilities and rich Spark spatial data analysis algorithm operators, DLI delivers comprehensive support for real-time processing of dynamic streaming data with location attributes and offline batch processing. DLI can handle massive data, including structured remote sensing image data, unstructured 3D modeling, and laser point cloud data.
 - **CEP SQL:** DLI delivers geographical location analysis functions to analyze geospatial data in real time. You can fulfill yaw detection and geofencing through SQL statements.
 - **Big Data Processing:** DLI allows you to quickly migrate remote sensing image data at the TB to EB scale to the cloud and perform image data slicing to offer resilient distributed datasets (RDDs) for distributed batch computing.
- It is recommended that you use the following related services:
DIS, CDM, DES, OBS, RDS, and CloudTable

Figure 4-4 Geographic Big Data Analysis



5 Notes and Constraints

On Jobs

- DLI supports the following types of jobs: Spark SQL, Spark Jar, Flink SQL, and Flink Jar.
- DLI supports the following Spark versions: Spark 3.3.1, Spark 3.1.1 (EOM), Spark 2.4.5 (EOM), and Spark 2.3 (EOS).
- DLI supports the following Flink versions: Flink Jar 1.15, Flink 1.12 (EOM), Flink 1.10 (EOS), and Flink 1.7 (EOS).
- SQL jobs support the Spark and HetuEngine engines.
 - **Spark**: displays jobs whose execution engine is Spark.
 - **HetuEngine**: displays jobs whose execution engine is HetuEngine.
- SparkUI can only display the latest 100 jobs.
- A maximum of 1,000 job results can be displayed on the console. To view more or all jobs, export the job data to OBS.
- To export job run logs, you must have the permission to access OBS buckets. You need to configure a DLI job bucket on the **Global Configuration > Project** page in advance.
- The **View Log** button is not available for synchronization jobs and jobs running on the default queue.
- Only Spark jobs support custom images.
- An elastic resource pool supports a maximum of 32,000 CUs.
- Minimum CUs of a queue that can be created in an elastic resource pool:
 - General purpose queue: 4 CUs
 - SQL queue: Spark SQL queue: 8 CUs; HetuEngine SQL queue: 16 CUs

On DLI Packages

- After a package is purchased, the region cannot be changed. A purchased package can be used only in the region where it is purchased.
- Packages cannot be refunded once purchased.
- When billing, resources from the package are used first. Once the package's resources are exhausted, any excess usage is billed on a pay-per-use basis.

- Resources used before a package is purchased cannot be deducted from the purchased package.
- If your package expires, you can still use DLI resources and you will be billed on a pay-per-use basis.
- The quota of a storage package is reset every hour. The quota of other types of packages is reset every month.

On Queues

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

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

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

For more notes and constraints on using a DLI queue, see [Notes and Constraints on Using a Queue](#).

On Elastic Resource Pools

- The billing mode of an elastic resource pool cannot be changed.
- The region of an elastic resource pool cannot be changed.
- For a pay-per-use resource pool, the dedicated resource mode is selected by default. The resource pool is billed by calendar hour since it is created.
- Jobs of Flink 1.10 or later can run in elastic resource pools.
- The CIDR block of an elastic resource pool cannot be changed once set.
- Associating an elastic resource pool with a queue:
 - Only pay-per-use queues (including dedicated queues) can be associated with elastic resource pools.
 - No resources are frozen.
- Currently, only yearly/monthly elastic resource pools can be scaled.
- You can only view the scaling history of resource pools in the last 30 days.
- Elastic resource pools cannot access the Internet.

- Changes to elastic resource pool CUs can occur when setting the CU, adding or deleting queues in an elastic resource pool, or modifying the scaling policies of queues in an elastic resource pool, or when the system automatically triggers elastic resource pool scaling. However, in some cases, the system cannot guarantee that the scaling will reach the target CUs as planned.
 - If there are not enough physical resources, an elastic resource pool may not be able to scale out to the desired target size.
 - The system does not guarantee that an elastic resource pool will be scaled in to the desired target size.

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

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

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

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

For more notes and constraints on elastic resource pools, see [Notes and Constraints on Using an Elastic Resource Pool](#).

On DLI Storage Resources

DLI can store databases and tables. DLI storage is billed based on the amount of stored data.

On Resources

- **Database**
 - **default** is the database built in DLI. You cannot create a database named **default**.
 - DLI supports a maximum of 50 databases.
- **Table**
 - DLI supports a maximum of 5,000 tables.
 - DLI supports the following table types:
 - **MANAGED**: Data is stored in a DLI table.
 - **EXTERNAL**: Data is stored in an OBS table.
 - **View**: A view can only be created using SQL statements.
 - **Datasource table**: The table type is also **EXTERNAL**.

- You cannot specify a storage path when creating a DLI table.
- **Data import**
 - Only OBS data can be imported to DLI or OBS.
 - You can import data in CSV, Parquet, ORC, JSON, or Avro format from OBS to tables created on DLI.
 - To import data in CSV format to a partitioned table, place the partition column in the last column of the data source.
 - The encoding format of imported data can only be UTF-8.
- **Data export**
 - Data in DLI tables (whose table type is **MANAGED**) can only be exported to OBS buckets, and the export path must contain a folder.
 - The exported file is in JSON format, and the text format can only be UTF-8.
 - Data can be exported across accounts. That is, after account B authorizes account A, account A has the permission to read the metadata and permission information of account B's OBS bucket as well as the read and write permissions on the path. Account A can export data to the OBS path of account B.
- **Package**
 - A package can be deleted, but a package group cannot be deleted.
 - The following types of packages can be uploaded:
 - **JAR**: JAR file
 - **PyFile**: User Python file
 - **File**: User file
 - **ModelFile**: User AI model file

For more notes and constraints on resources, see [Data Management](#).

On Enhanced Datasource Connections

- Datasource connections cannot be created for the **default** queue.
- Flink jobs can directly access DIS, OBS, and SMN data sources without using datasource connections.
- Enhanced connections can only be created for yearly/monthly and pay-per-use queues.
- **VPC Administrator** permissions are required for enhanced connections to use VPCs, subnets, routes, VPC peering connections.
You can set these permissions by referring to [Service Authorization](#).
- If you use an enhanced datasource connection, the CIDR block of the elastic resource pool or queue cannot overlap with that of the data source.
- Only queues bound with datasource connections can access datasource tables.
- Datasource tables do not support the preview function.
- When checking the connectivity of datasource connections, the notes and constraints on IP addresses are:

- The IP address must be valid, which consists of four decimal numbers separated by periods (.). The value ranges from 0 to 255.
- During the test, you can add a port after the IP address and separate them with colons (:). The port can contain a maximum of five digits. The value ranges from 0 to 65535.
For example, **192.168.xx.xx** or **192.168.xx.xx:8181**.
- When checking the connectivity of datasource connections, the notes and constraints on domain names are:
 - The domain name can contain 1 to 255 characters. Only letters, numbers, underscores (_), and hyphens (-) are allowed.
 - The top-level domain name must contain at least two letters, for example, **.com**, **.net**, and **.cn**.
 - During the test, you can add a port after the domain name and separate them with colons (:). The port can contain a maximum of five digits. The value ranges from 0 to 65535.
For example, **example.com:8080**.

For more notes and constraints on enhanced datasource connections, see [Enhanced Datasource Connection Overview](#).

On Datasource Authentication

- Only Spark SQL and Flink OpenSource SQL 1.12 jobs support datasource authentication.
- Flink jobs can use datasource authentication only on queues created after May 1, 2023.
- DLI supports four types of datasource authentication. Select an authentication type specific to each data source.
 - CSS: applies to 6.5.4 or later CSS clusters with the security mode enabled.
 - Kerberos: applies to MRS security clusters with Kerberos authentication enabled.
 - Kafka_SSL: applies to Kafka with SSL enabled.
 - Password: applies to GaussDB(DWS), RDS, DDS, and DCS.

For more notes and constraints on datasource authentication, see [Datasource Authentication Introduction](#).

On SQL Syntax

- Constraints on the SQL syntax:
 - You are not allowed to specify a storage path when creating a DLI table using SQL statements.
- Constraints on the size of SQL statements:
 - Each SQL statement should contain less than 500,000 characters.
 - The size of each SQL statement must be less than 1 MB.

Other

- For quota notes and constraints, see [Quotas](#).

- Recommended browsers for logging in to DLI:
 - Google Chrome 43.0 or later
 - Mozilla Firefox 38.0 or later
 - Internet Explorer 9.0 or later

For details about the compatibility list of more browsers, see [Which Browsers Are Supported?](#)

6 Specifications

Elastic Resource Pool Specifications

An elastic resource pool provides compute resources (CPU and memory) for running DLI jobs. The unit is CU. One CU contains one CPU and 4 GB memory.

You can create multiple queues in an elastic resource pool. Compute resources can be shared among queues. You can properly set the resource pool allocation policy for queues to improve compute resource utilization.

DLI offers compute resources in the specifications listed in [Table 6-1](#).

Table 6-1 Elastic resource pool specifications

Edition	Specifications	Notes and Constraints	Use Case
Basic	16–64 CUs	<ul style="list-style-type: none">• High reliability and availability are not supported.• Queue properties and job priorities cannot be set.• Notebook instances cannot be interconnected with. <p>For other notes and constraints on using an elastic resource pool, see Notes and Constraints on Using an Elastic Resource Pool.</p>	This edition is suitable for testing scenarios with low resource consumption and low requirements for resource reliability and availability.

Edition	Specifications	Notes and Constraints	Use Case
Standard	64 CUs or higher	For notes and constraints on using an elastic resource pool, see Notes and Constraints on Using an Elastic Resource Pool .	This edition offers powerful computing capabilities, high availability, and flexible resource management. It is suitable for large-scale computing tasks and business scenarios with long-term resource planning needs.

7 Security

7.1 Shared Responsibilities

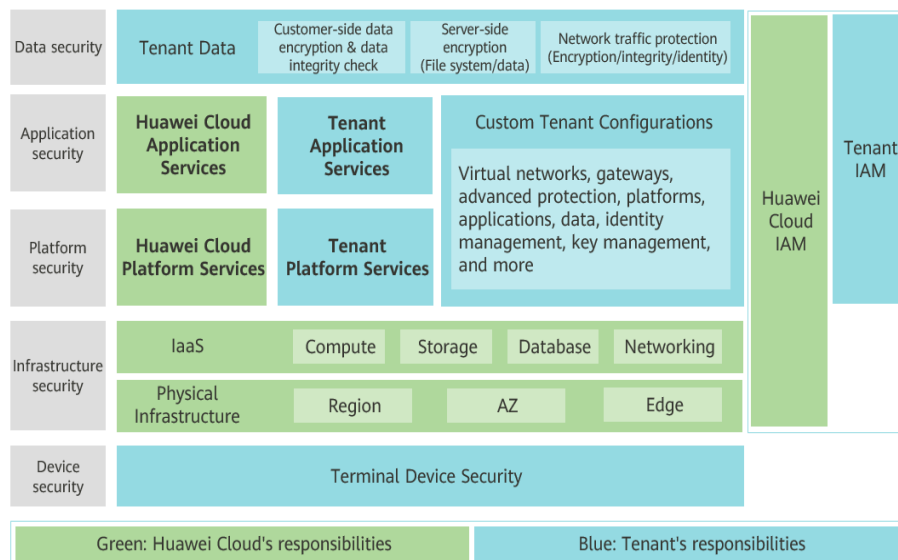
Huawei guarantees that its commitment to cyber security will never be outweighed by the consideration of commercial interests. To cope with emerging cloud security challenges and pervasive cloud security threats and attacks, Huawei Cloud builds a comprehensive cloud service security assurance system for different regions and industries based on Huawei's unique software and hardware advantages, laws, regulations, industry standards, and security ecosystem.

Figure 7-1 illustrates the responsibilities shared by Huawei Cloud and users.

- **Huawei Cloud:** Ensure the security of cloud services and provide secure clouds. Huawei Cloud's security responsibilities include ensuring the security of our IaaS, PaaS, and SaaS services, as well as the physical environments of the Huawei Cloud data centers where our IaaS, PaaS, and SaaS services operate. Huawei Cloud is responsible for not only the security functions and performance of our infrastructure, cloud services, and technologies, but also for the overall cloud O&M security and, in the broader sense, the security and compliance of our infrastructure and services.
- **Tenant:** Use the cloud securely. Tenants of Huawei Cloud are responsible for the secure and effective management of the tenant-customized configurations of cloud services including IaaS, PaaS, and SaaS. This includes but is not limited to virtual networks, the OS of virtual machine hosts and guests, virtual firewalls, API Gateway, advanced security services, all types of cloud services, tenant data, identity accounts, and key management.

Huawei Cloud Security White Paper elaborates on the ideas and measures for building Huawei Cloud security, including cloud security strategies, the shared responsibility model, compliance and privacy, security organizations and personnel, infrastructure security, tenant service and security, engineering security, O&M security, and ecosystem security.

Figure 7-1 Huawei Cloud shared security responsibility model



7.2 Asset Identification and Management

DLI allows you to use tags to identify and manage resources.

Scenario

Generally, your service system may use multiple Huawei Cloud services. You can set tags for different resource instances of these cloud services. These tags are also displayed in the billing records of each service. If your system is composed of multiple applications, setting the same tag for all resource instances used for each application helps you easily analyze resource usage and costs.

For DLI, tags are used to identify and classify purchased queues and created databases. If you add tags to a queue or database, CDRs generated by the requests for the queue or database will contain these tags. Using the tags, you can classify CDRs for detailed cost analysis.

For example, if a department uses a queue, you can add the department name to the queue as a tag. You can use tags to analyze the CDRs for development and usage costs of the department.

A tag is a key-value pair. By default, you can add 20 tags for a queue. Each tag has only one key and one value. The key and value can exist in either sequence in a tag. Each key is unique among all tags of a cluster, whereas values can be repetitive or blank.

How to Use

You can create queues and database tags on the DLI management console. For details, see [Managing Queue Tags](#).

7.3 Identity Authentication and Access Control

Identity Authentication

You can access DLI through the DLI console or open APIs. In either way, access requests are sent through the RESTful APIs provided by DLI.

DLI APIs can be accessed upon successful authentication. Requests sent through the DLI console and requests for calling APIs can both be authenticated using tokens.

Access Control

You can use Identity and Access Management (IAM) to implement fine-grained permissions management. IAM provides identity authentication, permissions management, and access control, helping you secure access to your Huawei Cloud resources.

For more information about IAM, see [IAM Service Overview](#).

You can grant users permissions by using roles and policies.

- **Roles:** A type of coarse-grained authorization mechanism that defines permissions related to user responsibilities. Only a limited number of service-level roles are available. When using roles to grant permissions, you need to also assign other roles on which the permissions depend to take effect. However, roles are not an ideal choice for fine-grained authorization and secure access control.
- **Policies:** A type of fine-grained authorization mechanism that defines permissions required to perform operations on specific cloud resources under certain conditions. This mechanism allows for more flexible policy-based authorization, meeting requirements for secure access control. For example, a specific user group is not allowed to delete a cluster. Only basic DLI operations (such as creating and querying jobs) are allowed.

For details about DLI permissions, see [Permission Management Overview](#).

The following table lists all the system permissions of DLI.

Role/Policy Name	Description	Category	Authorization Method
DLI FullAccess	Full permissions for DLI.	System - defined policy	For details about the authorization mode, see Creating an IAM User and Granting Permissions, Creating an IAM User , and Policies .
DLI ReadOnlyAccess	Read-only permissions for DLI.	System - defined policy	

Role/Policy Name	Description	Category	Authorization Method
Tenant Administrator	Tenant administrator <ul style="list-style-type: none"> Administer permissions for managing and accessing all cloud services. After a database or a queue is created, the user can use the ACL to assign rights to other users. Scope: project-level service 	System - defined role	
DLI Service Administrator	DLI administrator <ul style="list-style-type: none"> Administer permissions for managing and accessing the queues and data of DLI. After a database or a queue is created, the user can use the ACL to assign rights to other users. Scope: project-level service 	System - defined role	

7.4 Data Protection Technologies

Data Storage Security

To prevent your personal data (such as username, password, and mobile number) from being obtained by unauthorized or unauthenticated entities or individuals, DLI encrypts your data during storage and transmission to prevent data leakage.

Data Destruction Mechanism

When you delete a DLI cluster, your personal data stored in the cluster will also be deleted.

When you delete your mobile number and email address on the console and disable message notifications, the mobile number and email address will also be deleted from the database.

Data Transmission Security

Your personal data is encrypted using TLS 1.2 or TLS 1.3 during transmission. All the calls made to DLI APIs use HTTPS to encrypt the transmitted data.

Communication Encryption During Spark Job Transmission

For Spark jobs, you can enable communication encryption by setting the parameters listed in [Table 7-1](#).

NOTE

Make sure that you have uploaded the keys and certificates to the specified OBS path and included them in the other dependencies in the job configuration.

Table 7-1 Parameters used to configure communication encryption for Spark job transmission

Parameter	Description	Example Value
spark.network.crypto.enabled	This parameter lets you enable or disable communication encryption for data transmission between nodes. When set to true , Spark encrypts all communications between Executors and Drivers and between Executors. This configuration is critical to ensure data transmission security.	true
spark.network.sasl.serverAlwaysEncrypt	This parameter is used to configure whether the server uses encryption to communicate with clients. When set to true , the server requires all clients to use encrypted connections, which improves communication security.	true
spark.authenticate	This parameter is used to configure whether to authenticate the accesses to the components of a Spark application. Enabling authentication blocks unauthorized accesses. Setting to true means enabling authentication.	true

Communication Encryption During Flink Job Transmission

For Flink jobs, you can enable SSL transmission by setting the parameters listed in [Table 7-2](#).

NOTE

- Enabling SSL for the data transmission streams between Task Managers significantly affects system performance. Thoroughly evaluate both security and performance before you enable it.
- Configure certificate file information in **Other Dependencies** of the job editing page.
- The default path for storing dependency files is `/opt/flink/usrlib/userData/` in OBS.
- Make sure that you have uploaded the keys and certificates to the specified OBS path and included them in the other dependencies in the job configuration.

Table 7-2 Parameters used to configure communication encryption for Flink job transmission

Parameter	Description	Mandatory	Example Value
security.ssl.enabled	Whether to enable SSL.	Yes	true
akka.ssl.enabled	Whether to enable Akka SSL.	No	true
blob.service.ssl.enabled	Whether to enable SSL for the Blob channel.	No	true
taskmanager.data.ssl.enabled	Whether to enable SSL for communications between TaskManagers.	No	true
security.ssl.algorithms	Whether to configure SSL encryption algorithms.	No	TLS_DHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_DHE_RSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

Parameter	Description	Mandatory	Example Value
security.ssl.keystore	Path used to store keystore files. flink.keystore indicates the keystore file generated by generate_keystore.sh* .	Yes	/opt/flink/usrlib/userData/flink.keystore
security.ssl.keystore-password	Password for the keystore file, which can be customized	Yes	-
security.ssl.key-password	Password for the SSL key, which can be customized	Yes	-
security.ssl.truststore	Path used to store truststore files. flink.truststore indicates the truststore file generated by generate_keystore.sh* .	Yes	/opt/flink/usrlib/userData/flink.truststore
security.ssl.truststore-password	Password for the truststore file, which can be customized	Yes	-
security.ssl.rest.enabled	Whether to enable SSL/TLS encryption for REST APIs	Yes	false
security.ssl.verify-hostname	Controls whether to check if the host name (hostname) of the peer matches the information in the certificate when establishing an SSL/TLS connection.	No	false
security.ssl.protocol	Protocol version used when SSL/TLS connections are specified	No	TLS 1.2, TLS 1.3

Parameter	Description	Mandatory	Example Value
security.ssl.encrypt.enabled	Whether to enable data encryption for communications within a Flink cluster and between Flink and other components	No	false

Example of enabling communication encryption for Flink job transmission:

```
security.ssl.enabled: true
security.ssl.encrypt.enabled: false
security.ssl.key-password: ***
security.ssl.keystore-password: Admin12!
security.ssl.keystore: /opt/flink/usrlib/userData/*.keystore
security.ssl.protocol: TLSv1.2
security.ssl.rest.enabled: false
security.ssl.truststore-password: ***
security.ssl.truststore: /opt/flink/usrlib/userData/*.truststore
security.ssl.verify-hostname: false
```

7.5 Audit and Logging

- **Interconnecting DLI with CTS**

Cloud Trace Service (CTS) records operations on the cloud resources in your account. You can use the logs generated by CTS to perform security analysis, track resource changes, audit compliance, and locate faults.

For details about the DLI operations that can be recorded, see [DLI Operations That Can Be Recorded by CTS](#). After you enable CTS and create and configure a tracker, CTS starts recording operations for auditing. For details about how to enable CTS and view real-time traces, see [Getting Started with CTS](#).

You can [configure key event notification](#) on CTS. With CTS, you can monitor high-risk and sensitive operations related to IAM in real time. If you perform such an operation when using DLI, CTS sends a notification to subscribers.

- **DLI Job Logs**

When you create a DLI job, you can enable the job log saving function to store job logs to OBS.

For details about how to query job logs, see [How Do I View DLI SQL Logs?](#)

Job logs are important for routine service O&M and are used for tracking resource usage, detecting job security threats, and detecting errors.

7.6 Service Resilience

DLI ensures the durability and reliability of data through various technical methods, including traffic throttling, cross-AZ disaster recovery, and backup and restoration.

- Traffic throttling: DLI implements a traffic control mechanism to prevent service overload and maintain service stability.
- Cross-AZ disaster recovery: DLI is deployed across multiple availability zones to reduce the risk of single point failures and improve system availability and elasticity.
- Backup and restoration: DLI has automated backup policies and restoration plans in place to ensure quick service and data restoration in the event of a failure.

7.7 Security Risk Monitoring

Cloud Eye provides a multi-dimensional monitoring platform for cloud resources. With Cloud Eye, you can view the resource usage and service running status, and respond to exceptions in a timely manner to ensure smooth running of services.

DLI resources can be monitored by Cloud Eye.

You can connect DLI to Cloud Eye to monitor your queues in real time, report alarms, and be notified of operations on DLI resources. You can learn about the inbound rate, outbound rate, CPU usage, memory usage, disk usage, job failure rate, and number of jobs waiting in a queue in real time. You can also use the management console or APIs provided by Cloud Eye to query the monitoring metrics and alarms generated for DLI.

For details about DLI metrics, see [Viewing Monitoring Metrics](#).

7.8 Recovery from Failures

- **System-Level**
DLI uses an architecture with separated storage and compute resources. A compute cluster can be automatically recovered if a system fault occurs, thanks to the Kubernetes resource scheduling and failover mechanism.
- **Job-Level**
You can enable automatic restart and recovery for Flink and Spark jobs. After this function is enabled, jobs will be automatically restarted and recovered if exceptions occur.

7.9 Update Management

Vulnerability Patching

DLI closely monitors vulnerabilities through Huawei Cloud security notices, such as the Apache Log4j2 remote code execution vulnerability (CVE-2021-44228) and the Fastjson deserialization vulnerability (CNVD-2022-40233).

Once vulnerabilities affecting service modules are identified, official solutions are promptly applied to upgrade and patch the live network.

Configuration Updates

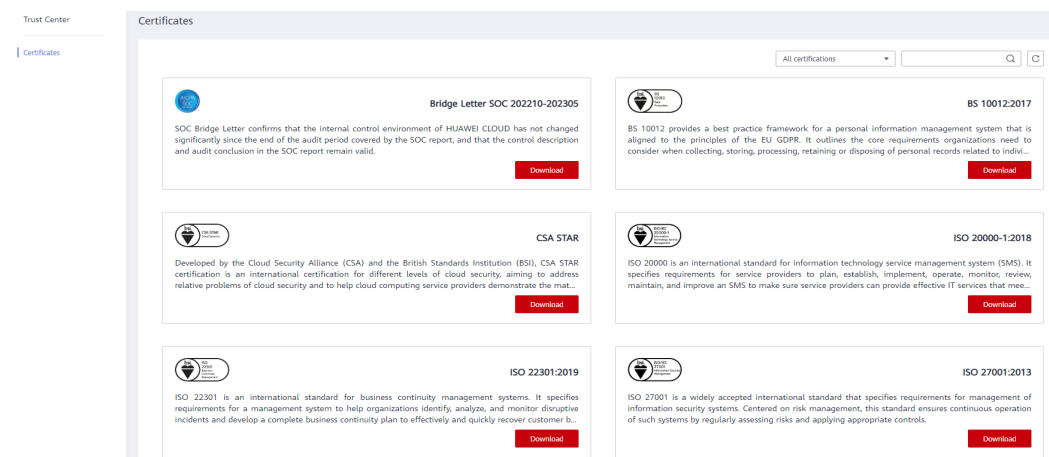
DLI ensures its security and stability by regularly updating and upgrading configurations.

7.10 Certificates

Compliance Certificates

Huawei Cloud services and platform have passed security and compliance certifications from various authoritative organizations both domestically and internationally, such as International Organization for Standardization (ISO), system and organization controls (SOC), and Payment card industry (PCI). You can [apply and download](#) compliance certificates on your own.

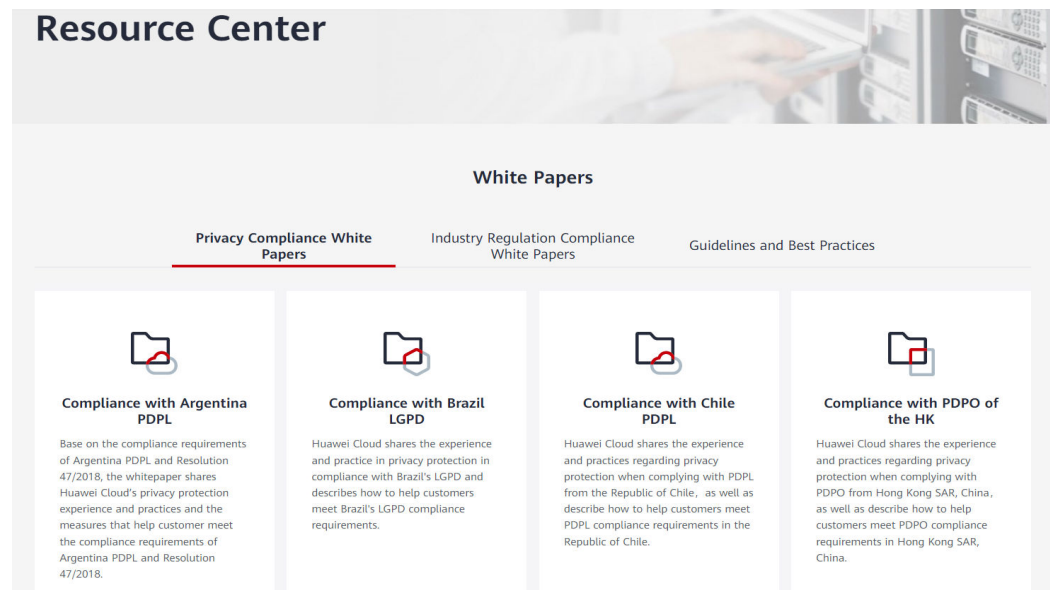
Figure 7-2 Downloading compliance certificates



Resources

Huawei Cloud also provides the following resources to help you meet compliance requirements. For details, see [Resource Center](#).

Figure 7-3 Resources



8 Permissions Management

If you need to assign different permissions to employees in your enterprise to access your DLI resources, IAM is a good choice for fine-grained permissions management. IAM provides identity authentication, permissions management, and access control, helping you securely access to your public cloud resources.

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

If your account does not require individual IAM users for permissions management, you may skip over this section.

IAM can be used free of charge. You pay only for the resources in your account. For more information about IAM, see [IAM Service Overview](#).

DLI Permissions

By default, new IAM users do not have permissions assigned. You need to add the users to one or more groups, and attach permissions policies or roles to these groups. The users then inherit permissions from the groups to which they are added. After authorization, the users can perform specified operations on DLI based on the permissions.

DLI is a project-level service deployed and accessed in specific physical regions. To assign ServiceStage permissions to a user group, specify the scope as region-specific projects and select projects for the permissions to take effect. If **All projects** is selected, the permissions will take effect for the user group in all region-specific projects. When accessing DLI, the users need to switch to a region where they have been authorized to use cloud services.

Type: There are roles and policies.

- **Roles:** A type of coarse-grained authorization mechanism that defines permissions related to user responsibilities. Only a limited number of service-level roles are available. If one role has a dependency role required for accessing SA, assign both roles to the users. However, roles are not an ideal choice for fine-grained authorization and secure access control.

- Policies: A type of fine-grained authorization mechanism that defines permissions required to perform operations on specific cloud resources under certain conditions. This mechanism allows for more flexible policy-based authorization, meeting requirements for secure access control. For example, you can grant DLI users only the permissions for managing a certain type of ECSs. For the actions supported by DLI APIs, [Permissions Policies and Supported Actions](#).

Table 8-1 DLI system permissions

Role/Policy Name	Description	Category	Dependency
DLI FullAccess	Full permissions for DLI.	System-defined policy	This role depends on other roles in the same project. <ul style="list-style-type: none"> • Creating a datasource connection: VPC ReadOnlyAccess • Creating yearly/monthly resources: BSS Administrator • Creating a tag: TMS FullAccess and EPS FullAccess • Using OBS for storage: OBS OperateAccess • Creating an agency: Security Administrator
DLI ReadOnlyAccess	Read-only permissions for DLI. With read-only permissions, you can use DLI resources and perform operations that do not require fine-grained permissions. For example, create global variables, create packages and package groups, submit jobs to the default queue, create tables in the default database, create datasource connections, and delete datasource connections.	System-defined policy	None

Role/Policy Name	Description	Category	Dependency
Tenant Administrator	Tenant administrator <ul style="list-style-type: none"> Job execution permissions for DLI resources. After a database or a queue is created, the user can use the ACL to assign rights to other users. Scope: project-level service 	System-defined role	None
DLI Service Administrator	DLI administrator. <ul style="list-style-type: none"> Job execution permissions for DLI resources. After a database or a queue is created, the user can use the ACL to assign rights to other users. Scope: project-level service 	System-defined role	None

Table 8-2 lists the common operations supported by each system policy. You can choose required system policies according to this table.

For details about how to grant permissions using SQL syntax, see [Data Permissions List](#) in *Data Lake Insight SQL Syntax Reference*.

Table 8-2 Common operations supported by each system permission

Resource	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Administrator
Queue	DROP_QUEUE	Deleting a Queue	√	×	√	√
	SUBMIT_JOB	Submitting a job	√	×	√	√
	CANCEL_JOB	Terminating a Job	√	×	√	√

Resource	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Administrator
	RESTART	Restarting a queue	√	×	√	√
	GRANT_PRIVILEGE	Granting permissions to a queue	√	×	√	√
	REVOKE_PRIVILEGE	Revoking permissions to a queue	√	×	√	√
	SHOW_PRIVILEGES	Viewing the queue permissions of other users	√	×	√	√
Database	DROP_DATABASE	Deleting a database	√	×	√	√
	CREATE_TABLE	Creating a table	√	×	√	√
	CREATE_VIEW	Creating a view	√	×	√	√
	EXPLAIN	Explaining the SQL statement as an execution plan	√	×	√	√
	CREATE_ROLE	Creating a role	√	×	√	√
	DROP_ROLE	Deleting a role	√	×	√	√
	SHOW_ROLES	Displaying a role	√	×	√	√
	GRANT_ROLE	Binding a role	√	×	√	√
	REVOKE_ROLE	Unbinding a role	√	×	√	√

Resource	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Administrator
	SHOW_USERS	Displaying the binding relationships between all roles and users	√	×	√	√
	GRANT_PRIVILEGE	Granting permissions to the database	√	×	√	√
	REVOKE_PRIVILEGE	Revoking permissions to the database	√	×	√	√
	SHOW_PRIVILEGES	Viewing database permissions of other users	√	×	√	√
	DISPLAY_ALL_TABLES	Displaying tables in a database	√	√	√	√
	DISPLAY_DATABASE	Displaying databases	√	√	√	√
	CREATE_FUNCTION	Creating a function	√	×	√	√
	DROP_FUNCTION	Deleting a function	√	×	√	√
	SHOW_FUNCTIONS	Displaying all functions	√	×	√	√
	DESCRIBE_FUNCTION	Displaying function details	√	×	√	√
Table	DROP_TABLE	Deleting tables	√	×	√	√
	SELECT	Querying tables	√	×	√	√
	INSERT INTO TABLE	Inserting table data	√	×	√	√

Resource	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Administrator
	ALTER_TABLE_ADD_COLUMNS	Adding a column	√	×	√	√
	INSERT_OVERWRITE_TABLE	Overwriting a table	√	×	√	√
	ALTER_TABLE_RENAME	Renaming a table	√	×	√	√
	ALTER_TABLE_ADD_PARTITION	Adding partitions to the partition table	√	×	√	√
	ALTER_TABLE_RENAME_PARTITION	Renaming a table partition	√	×	√	√
	ALTER_TABLE_DROP_PARTITION	Deleting partitions from a partition table	√	×	√	√
	SHOW_PARTITIONS	Displaying all partitions	√	×	√	√
	ALTER_TABLE_RECOVER_PARTITION	Restoring table partitions	√	×	√	√
	ALTER_TABLE_SET_LOCATION	Setting the partition path	√	×	√	√
	GRANT_PRIVILEGE	Granting permissions to the table	√	×	√	√
	REVOKE_PRIVILEGE	Revoking permissions to the table	√	×	√	√
	SHOW_PRIVILEGES	Viewing table permissions of other users	√	×	√	√

Resource	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Administrator
	DISPLAY_TABLE	Displaying a table	√	√	√	√
	DESCRIBE_TABLE	Displaying table information	√	×	√	√
Elastic resource pool	DROP	Deleting an elastic resource pool	√	×	√	√
	RESOURCE_MANAGEMENT	Managing an elastic resource pool	√	×	√	√
	SCALE	Scaling an elastic resource pool	√	×	√	√
	UPDATE	Updating an elastic resource pool	√	×	√	√
	CREATE	Creating an elastic resource pool	√	×	√	√
	SHOW_PRIVILEGES	Viewing elastic resource pool permissions of other users	√	×	√	√
	GRANT_PRIVILEGE	Granting elastic resource pool permissions	√	×	√	√

Resource	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Administrator
	REVOKE_PRIVILEGE	Retrieving elastic resource pool permissions	√	×	√	√
Enhanced datasource connection	BIND_QUEUE	Binding an enhanced datasource connection to a queue It is only used to grant permissions across projects.	×	×	×	×

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

9 Quotas

What Is a Quota?

A quota limits the quantity of a resource available to users, thereby preventing spikes in the usage of the resource.

You can also request for an increased quota if your existing quota cannot meet your service requirements.

How Do I View My Quotas?


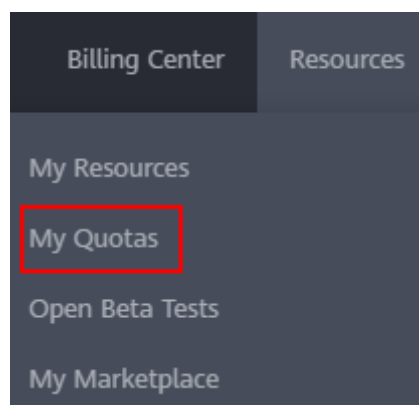
1. Log in to the management console.
2. Click  in the upper left corner and select a region and a project.
3. In the upper right corner of the page, choose **Resources > My Quotas**.
The **Service Quota** page is displayed.

Figure 9-1 My quotas



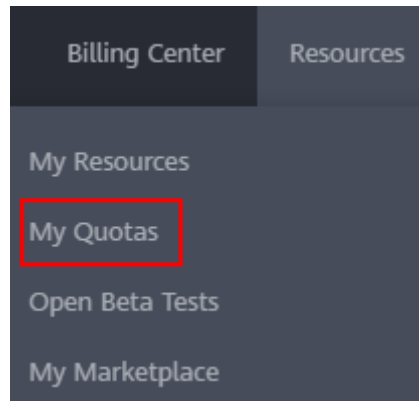
4. View the used and total quota of each type of resources on the displayed page.
If a quota cannot meet service requirements, increase a quota.

How Do I Apply for a Higher Quota?

1. Log in to the management console.

2. In the upper right corner of the page, choose **Resources > My Quotas**.
The **Service Quota** page is displayed.

Figure 9-2 My quotas



3. Click **Increase Quota**.
4. On the **Create Service Ticket** page, configure parameters as required.
In the **Problem Description** area, fill in the content and reason for adjustment.
5. Select the agreement and click **Submit**.

10 Related Services

OBS

OBS works as the data source and data storage system for DLI, and delivers the following capabilities:

- Data source: DLI provides an API for you to import data from corresponding OBS paths to DLI tables.

For details about the API, see [Importing Data](#).

- Data storage: You can create OBS tables on DLI. However, such tables only store metadata while data content is stored in corresponding OBS paths.

- Data backup: DLI provides an API for you to export the data in DLI to OBS for backup.

For details about the API, see [Exporting Data](#).

- Query result storage: DLI provides an API for you to save routine query result data on OBS.

For details about the API, see [Exporting Query Result](#).

IAM

Identity and Access Management (IAM) authenticates access to DLI.

For details about related operations, see [Creating an IAM User and Granting Permissions](#) and [Creating a Custom Policy](#).

CTS

Cloud Trace Service (CTS) audits performed DLI operations.

For details about DLI operations that can be recorded by CTS, see [DLI Operations That Can Be Recorded by CTS](#).

Cloud Eye

Cloud Eye helps monitor job metrics for DLI, delivering status information in a concise and efficient manner.

For details about the metrics, see [Viewing Monitoring Metrics](#).

SMN

Simple Message Notification (SMN) can send notifications to users when a job running exception occurs on DLI.

For details, see [Creating an SMN Topic](#).

CloudTable

CloudTable works as the data source and data storage system for DLI, and delivers the following capabilities:

- Data source: DLI allows you to import CloudTable data using DataFrame or SQL.
- Query result storage: DLI uses the SQL INSERT syntax to store query result data to CloudTable tables.

For how to use a DLI datasource connection to access CloudTable data, see [Cross-Source Analysis Development Methods](#).

RDS

Relational Database Service (RDS) works as the data source and data storage system for DLI, and delivers the following capabilities:

- Data source: DLI allows you to import RDS data using DataFrame or SQL.
- Query result storage: DLI uses the SQL INSERT syntax to store query result data to RDS tables.

For how to use a DLI datasource connection to access RDS data, see [Cross-Source Analysis Development Methods](#).

GaussDB(DWS)

GaussDB(DWS) works as the data source and data storage system for DLI, and delivers the following capabilities:

- Data source: DLI allows you to import GaussDB(DWS) data using DataFrame or SQL.
- Query result storage: DLI uses the SQL INSERT syntax to store query result data to GaussDB(DWS) tables.

For how to use a DLI datasource connection to access GaussDB(DWS) data, see [Cross-Source Analysis Development Methods](#).

CSS

CSS works as the data source and data storage system for DLI, and delivers the following capabilities:

- Data source: DLI allows you to import CSS data using DataFrame or SQL.
- Query result storage: DLI uses the SQL INSERT syntax to store query result data to CSS tables.

For how to use a DLI datasource connection to access GaussDB(DWS) data, see [Cross-Source Analysis Development Methods](#).

DCS

Distributed Cache Service (DCS) works as the data source and data storage system for DLI, and delivers the following capabilities:

- Data source: DLI allows you to import DCS data using DataFrame or SQL.
- Query result storage: DLI uses the SQL INSERT syntax to store query result data to DCS tables.

For how to use a DLI datasource connection to access GaussDB(DWS) data, see [Cross-Source Analysis Development Methods](#).

DDS

Document Database Service (DDS) works as the data source and data storage system for DLI, and delivers the following capabilities:

- Data source: DLI allows you to import DDS data using DataFrame or SQL.
- Query result storage: DLI uses the SQL INSERT syntax to store query result data to DDS tables.

For how to use a DLI datasource connection to access GaussDB(DWS) data, see [Cross-Source Analysis Development Methods](#).

MRS

MapReduce Service (MRS) works as the data source and data storage system for DLI, and delivers the following capabilities:

- Data source: DLI allows you to import MRS data using DataFrame or SQL.
- Query result storage: DLI uses the SQL INSERT syntax to store query result data to MRS tables.

For how to use a DLI datasource connection to access GaussDB(DWS) data, see [Cross-Source Analysis Development Methods](#).

11 Basic Concepts

Use Elastic Resource Pool

Dedicated computing resources. They are isolated by resource pools and can only be shared by queues in the same elastic resource pool. You can set scaling policies of different priorities for these queues to adjust compute resources according to queue workload in different periods of a day.

DLI Storage Resource

DLI storage resources are the internal storage capacities of the DLI service. They are utilized for storing databases and DLI tables, playing a crucial role in data import into DLI. These resources also indicate the volume of data that users have stored within DLI.

Actual CUs, Used CUs, CU Range, and Specifications (Yearly/Monthly CUs) of an Elastic Resource Pool

- **Actual CUs:** number of CUs that can be allocated in the elastic resource pool.
- **Used CUs:** CUs that have been used by jobs or tasks. These CU resources may be currently engaged in computing tasks and therefore temporarily unavailable.

NOTE

The CUs used by HetuEngine match the actual CUs.

- **CU range:** CU settings are used to control the maximum and minimum CU ranges for elastic resource pools to avoid unlimited resource scaling.
 - The total minimum CUs of all queues in an elastic resource pool must be no more than the minimum CUs of the pool.
 - The maximum CUs of any queue in an elastic resource pool must be no more than the maximum CUs of the pool.
 - An elastic resource pool should at least ensure that all queues in it can run with the minimum CUs and should try to ensure that all queues in it can run with the maximum CUs.
- **Specifications (Yearly/Monthly CUs):** The minimum CUs selected during elastic resource pool purchase are elastic resource pool specifications. Specifications are exclusive to yearly/monthly elastic resource pools. Billing for

these specifications is yearly/monthly, while any usage exceeding the specifications is billed on a pay-per-use basis.

Database

A database is a warehouse where data is organized, stored, and managed based on the data structure. DLI management permissions are granted on a per database basis.

In DLI, tables and databases are metadata containers that define underlying data. The metadata in the table shows the location of the data and specifies the data structure, such as the column name, data type, and table name. A database is a collection of tables.

OBS Table, DLI Table, and CloudTable Table

The table type indicates the storage location of data.

- OBS table indicates that data is stored in the OBS bucket.
- DLI table indicates that data is stored in the internal table of DLI.
- CloudTable table indicates that data is stored in CloudTable.

You can create a table on DLI and associate the table with other services to achieve querying data from multiple data sources.

Metadata

Metadata is used to define data types. It describes information about the data, including the source, size, format, and other data features. In database fields, metadata interprets data content in the data warehouse.

SQL Job

SQL job refers to the SQL statement executed in the SQL job editor. It serves as the execution entity used for performing operations, such as importing and exporting data, in the SQL job editor.

This type is suitable for scenarios where standard SQL statements are used for querying. It is typically used for querying and analyzing structured data.

Flink Job

This type is specifically designed for real-time data stream processing, making it ideal for scenarios that require low latency and quick response. It is well-suited for real-time monitoring and online analysis.

- Flink OpenSource job: When submitting jobs, you can quickly integrate with other data systems using DLI's standard connectors and various APIs.
- Flink Jar job: allows you to submit Flink jobs compiled into JAR files, providing greater flexibility and customization capabilities. It is suitable for complex data processing scenarios that require user-defined functions (UDFs) or specific library integration. The Flink ecosystem can be utilized to implement advanced stream processing logic and status management.

Spark Job

Spark jobs are those submitted by users through visualized interfaces and RESTful APIs. Full-stack Spark jobs are allowed, such as Spark Core, DataSet, MLib, and GraphX jobs.

CU

Compute unit (CU) is the unit of compute resources in DLI, where 1 CU equals 1 vCPU and 4 GB of memory. The higher the specifications of compute resources, the better its computing power.

Constants and Variables

The differences between constants and variables are as follows:

- During the running of a program, the value of a constant cannot be changed.
- Variables are readable and writable, whereas constants are read-only. A variable is a memory address that contains a segment of data that can be changed during program running. For example, in **int a = 123**, **a** is an integer variable.

Table Lifecycle

The table lifecycle management feature in DLI refers to the automatic recycling of tables or partitions that have not been updated for a specified period of time since their last update. This specified period is known as the lifecycle. This feature simplifies the process of recycling data and frees up storage space. Additionally, it provides data backup and recovery functions to prevent data loss due to accidental operations.