Alibaba Cloud
Log Service

Product Introduction

Issue: 20190111

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.
5. By law, all the content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectu

al property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of the Alibaba Cloud website, product programs, or content shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates).

6. Please contact Alibaba Cloud directly if you discover any errors in this document.

Generic conventions

Table -1: Style conventions

Style	Description	Example
	This warning information indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 **Danger:** Resetting will result in the loss of user configuration data.
	This warning information indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 **Warning:** Restarting will cause business interruption. About 10 minutes are required to restore business.
	This indicates warning information, supplementary instructions, and other content that the user must understand.	 **Notice:** Take the necessary precautions to save exported data containing sensitive information.
	This indicates supplemental instructio ns, best practices, tips, and other content that is good to know for the user.	 **Note:** You can use **Ctrl** + **A** to select all files.
>	Multi-level menu cascade.	**Settings** > **Network** > **Set network type**
Bold	It is used for buttons, menus, page names, and other UI elements.	Click **OK**.
`Courier font`	It is used for commands.	Run the `cd /d C:/windows` command to enter the Windows system folder.
`Italics`	It is used for parameters and variables.	`bae log list --instanceid` *`Instance_ID`*
[] or [a|b]	It indicates that it is a optional value, and only one item can be selected.	`ipconfig` *`[-all|-t]`*
{} or {a|b}	It indicates that it is a required value, and only one item can be selected.	`swich` *`{stand | slave}`*

Contents

1 What is Log Service

As a one-stop service for log data, Log Service (Log for short) experiences massive big data scenarios of Alibaba Group. Log Service allows you to quickly complete the collection, consumption, shipping, query, and analysis of log data without the need for development, which improves the Operation & Maintenance (O&M) efficiency and the operational efficiency, and builds the processing capabilities to handle massive logs in the DT (data technology) era.

Log Service learning path

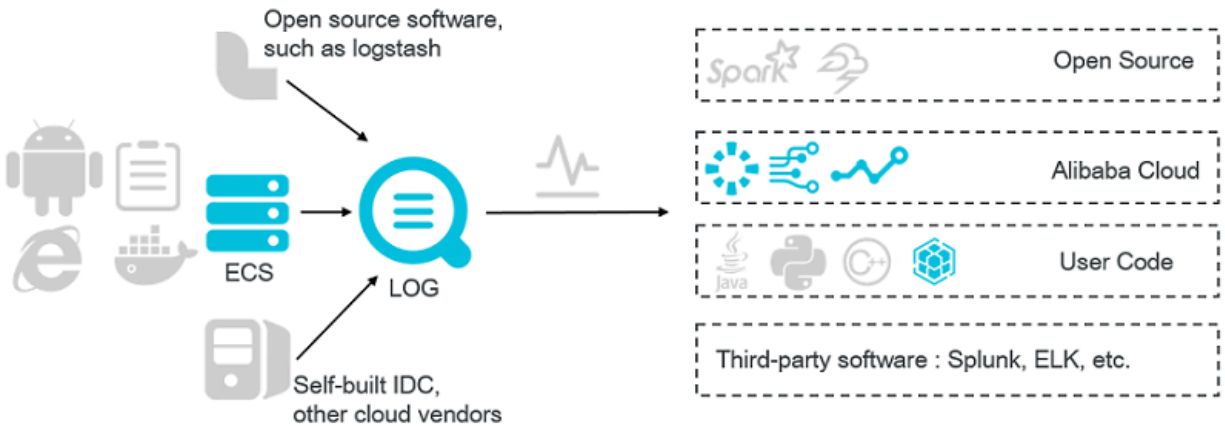
Log Service learning path recommends documents of hot functions, and helps you quickly have a knowlege of Log Service. Combined with video and documents, Log Service learning path optimizes the user experience of user and document reading experience.

Real-time log collection and consumption (LogHub)

Functions:

- Use Elastic Compute Service (ECS), containers, mobile terminals, open-source softwares, and JS to access real-time log data (such as Metric, Event, BinLog, TextLog, and Click data).
- A real-time consumption interface is provided to interconnect with real-time computing and service.

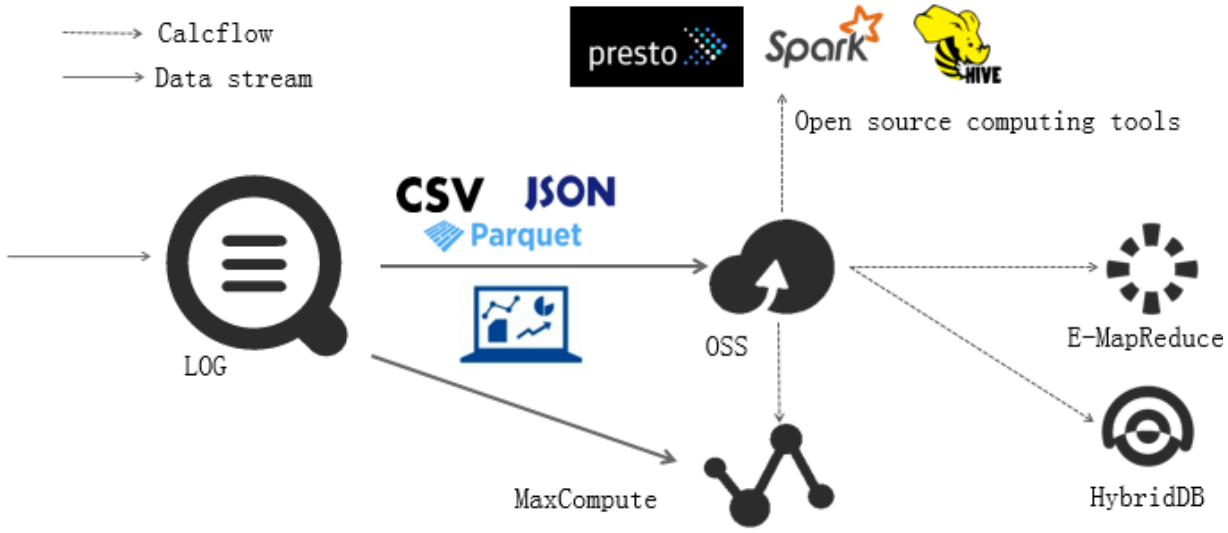
Purposes: ETL, Stream Compute, monitoring and alarm, machine learning, and iterative computing.



LogShipper

Stable and reliable log shipping ships LogHub data to storage services for storage and big data analysis. Supports various storage methods such as compression, user-defined partitions, row storage, and column storage.

Purposes: Data warehouse + data analysis, audit, recommendation system, and user profiling.

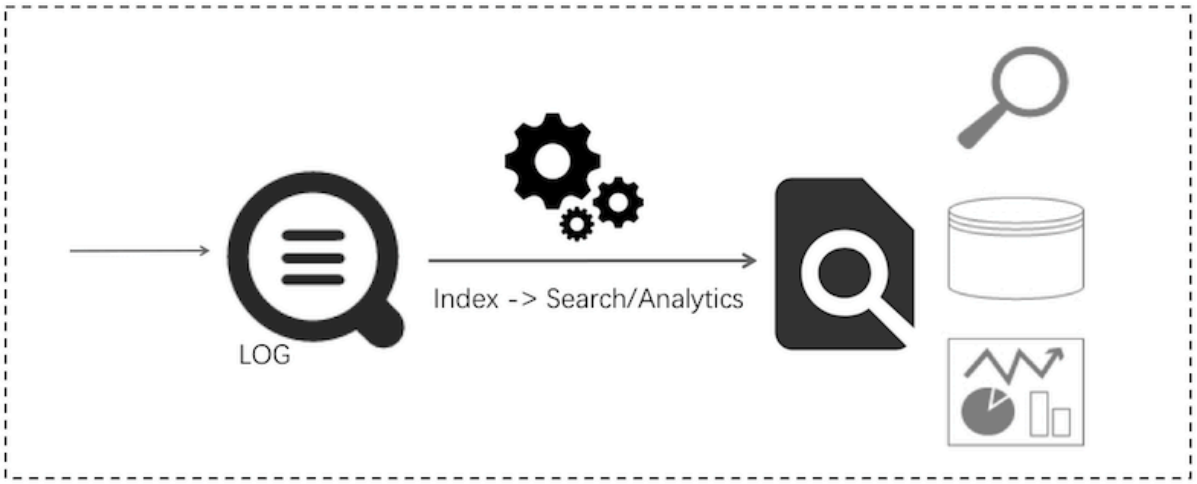


Query and real-time analysis (Search/Analytics)

Index, query, and analyze data in real time.

- Query: Keyword, fuzzy match, context, and range.
- Statistics: Rich query methods such as SQL aggregation.
- Visualization: Dashboard and report functions.
- Interconnection: Grafana and JDBC/SQL92.

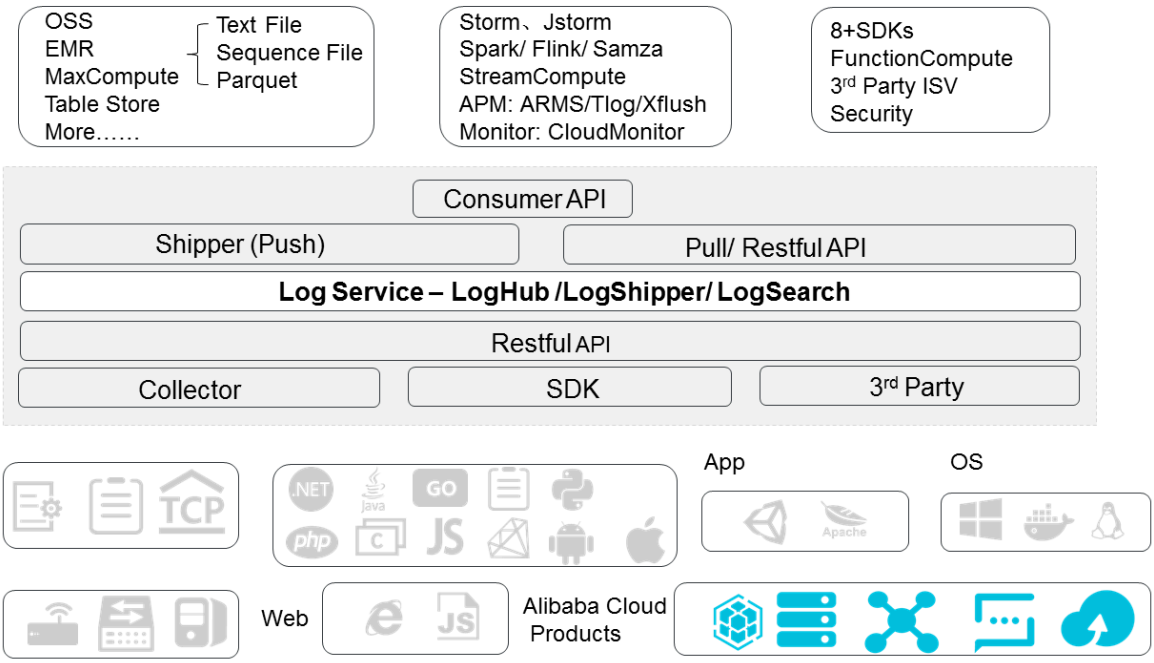
Purposes: DevOps/online O&M, real-time log data analysis, security diagnosis and analysis, and operation and customer service systems.



2 Architecture

The Log Service system architecture is as follows.

Figure 2-1: Architecture



Logtail

Logtail is an agent that helps you quickly collect logs and has the following features:

- Non-invasive log collection based on log files
  - Only read files.
  - Non-invasion during the reading process.
- Secure and reliable
  - Supports file rotation, so no loss of data.
  - Supports local caching.
  - Provides network exception retry mechanism.
- Convenient management
  - Management on Web.
  - Supports visualization configuration.
- Comprehensive self-protection

- Monitors the CPU and memory consumed by the process in real time.
- Restricts the upper limit of memory usage.

Frontend servers

Frontend servers are the frontend machines built with LVS + Nginx and have the following features:

- HTTP and REST protocols
- Horizontal scaling
  - Supports horizontal scaling when traffic increases.
  - Frontend servers can be added to quickly improve processing capabilities.
- High throughput and low latency
  - Pure asynchronous processing. A single request exception does not affect other requests.
  - Adopts the Lz4 compression, which is specially for logs, to increase the processing capabilities of individual machines and reduce network bandwidth.

Backend servers

The backend is a distributed process deployed on multiple machines. It provides real-time Logstore data persistence, index, query, and shipping to MaxCompute. The features of the overall backend service are as follows:

- High data security
  - Each log you write is saved in triplicate.
  - Data is automatically replicated and repaired if a disk is damaged or the machine hardware/software has a system error.
- Stable service
  - Logstores are automatically migrated if the process is crashed or the machine does not have a response for a long time.
  - Automatic Server Load Balancer makes sure that traffic is distributed evenly among different machines.
  - Strict quota limits that prevent abnormal behavior of a single user from affecting other users.
- Horizontal scaling
  - Horizontal scaling is performed by using shards as the unit.
  - You can dynamically add shards as needed to increase throughput.

3 Benefits

3.1 Benefits

Fully managed service

- Easy to use. You can access the service for usage in five minutes and use Agents to collect data in any network environment.
- LogHub has all the functions of Kafka, provides complete functional data, such as monitoring and alarms, and supports auto scaling (by PB/day). The use cost is less than 50% of the self-built cost.
- LogSearch/Analytics provides the functions of saving queries, dashboard, and alarm. The use cost is less than 20% of the self-built cost.
- Log Service has more than 30 Access Methods, and interconnects with cloud products (such as Object Storage Service (OSS), E-MapReduce, MaxCompute, Table Store, MNS, CDN, and ARMS) and open-source softwares (Storm and Spark) seamlessly.

Rich ecosystem

- LogHub supports over 30 collectors, including Logstash and Fluent, and can be easily accessed by using embedded devices, Web pages, servers, and programs. It can also be interconnected with consumption systems such as Spark Streaming, Storm, CloudMonitor, and ARMS.
- LogShipper Supports rich data formats (textfile, sequencefile, parquet, etc.), custom partition , data can be taken directly by presto, hive, spark, hadoop, e-mapreduce, maxcompute, hybridgedb, etc. processing.
- LogSearch/Analytics has complete query and analysis syntaxes and is compatible with SQL-92 . Supports interconnecting with Grafana by using JDBC protocol.

Strong real-timeliness

- LogHub: Data can be used after being written. Logtail (collection agent) can collect and transfer data in real time to the server side within one second (in 99.9% cases).
- LogSearch/Analytics: Data can be queried and analyzed after being written. When multiple query conditions are used, billions of data pieces can be queried within one second. When multiple aggregation conditions are used, hundreds of millions of data pieces can be analyzed within one second.

Complete API/SDK

- Easily supports user-defined management and secondary development.
- All functions can be implemented by using APIs/SDKs. SDKs for multiple languages are provided. Services and millions of devices can be managed in an easy way.
- The query and analysis syntax is simple (compatible with SQL-92). The interfaces can be used to interconnect with the ecological softwares (supports Grafana interconnection solution).

3.2 Cost advantages

Cost advantages

Log Service has the following cost advantages in three log processing scenarios:

- Loghub:
  - A more cost-effective choice for users in 98% scenarios compared to building Kafka with purchased cloud hosts + cloud disks. At less than 30% of the Kafka cost for small websites.
  - Provides RESTful APIs and supports data collection on mobile devices, saving you the cost of the gateway servers for log collection.
  - Operation & Maintenance (O&M) -free and auto scaling anytime and anywhere.
- Logshipper:
  - No code/machine resources required, flexible configuration, and rich monitoring data.
  - Linear scalability (PB grade/day), available for free currently.
- Logsearch/analytics:
  - At less than 15% of the cost of purchasing cloud hosts + self-building ELK, and offers dramatic enhancement in query capability and data processing scale. See Comparison report. A better choice than the above-mentioned log management softwares for its ability to seamlessly integrate with various popular stream computing + offline computing frameworks to allow for unobstructed flow of logs.

Cost Comparison

The following is the comparison of Log Service and self-built solutions in the billing model, for your reference only.

LogHub (LogHub vs Kafka)

-	Focus	LogHub	Self-built middleware (such as Kafka)
Decompress the file by using	New	Imperceptible	O&M required
	Expansion	Imperceptible	O&M required
	Increase backups	Imperceptible	O&M required
	Multitenancy	Quarantine	Might affect each other
Charge	Internet collection (10 GB/day)	USD 2/day	USD 16.1/day
	Internet collection (1 TB/day)	USD 162/day	USD 800/day
	Intranet collection (small data size)	-	-
	Intranet collection (moderate data size)	-	-
	Intranet collection (large data size)	-	-

Log Storage and Query Engine

Focus-		LogSearch	ES (Lucene Based)	NoSQL	Hive
Scale	Scale	PB	TB	PB	PB
Cost	Store (USD/GB per day)	0.0115	3.6	0.02	0.035
	Write (USD/ GB)	0.35	5	0.4	0
	Query (US $/GB)	0	0	0.2	0.3
	Speed-query	Millisecon d level-second level	Millisecon d level-second level	Within milliseconds	In minutes
	Speed-statistics	Weak+	Relatively strong	Weak	Strong

Focus-		LogSearch	ES (Lucene Based)	NoSQL	Hive
Latency	Write-> queryable	Real time	In minutes	Real time	Ten-minute level

Note:

The price comparisons here are calculated basically based on the fact that softwares are deployed on Elastic Compute Service (ECS) and three copies have been configured.

For more information, see Comparisons of log query solutions.*#unique_8*

3.3 Compare LogSearch/Analytics with ELK in log query and analysis

When talking about real-time log analysis, people will think of using ELK (Elastic, Logstash, and Kibana) to implement it. ELK Stack is an open-source solution that has accumulated many contents and use cases in the community.

The new version of Alibaba Cloud Log Service adds enhancements of LogSearch/LogAnalytics to support real-time indexing, query, and analysis of log data and optimize query performance and data volume computing in many aspects. This document conducts a comprehensive comparison and analyzes the aspects that you pay attention.

- Ease of use: The cost when you get started and use the function.
- Functions (important): Query and analysis.
- Performance (important): The query and analysis requirements for unit data volume and how is the latency.
- Scale (important): The data volumes that can be processed and the scalability.
- Cost (important): The cost for using the same function and performance.

Ease of use

A log analysis system is used in the following procedures:

1. Collection: Write data in a stable manner.
2. Configuration: How to configure the data source.
3. Expansion: Access more data sources and machines. Expand the storage space and machines.
4. Usage: Described in the Functions section.

5. Export: Whether data can be conveniently exported to other systems for operations such as streaming computing and storage in OSS for backup purposes
6. Multi-tenant: The way data is shared to others and whether or not data can be used securely.

Comparison results:

Item	Sub item	Self-built ELK	LogSearch/Analytics
Collection	Protocol	Restful API	Restful API
	Agent	Logstash/Beats/ Fluentd, with rich ecosystem	Logtail (main) + Others (for example, Logstash)
Configuration	Unit	Use index to differenti ate logs	Project + Logstore. Provide a concept of two levels. A project is considered as a namespace, and multiple Logstores can be created in a project
	Attribute	API + kikana	API + SDK + Console
Expansion	Storage	Add machines and purchase cloud disks	No operation is needed
	machine	Add machines	No operation is needed
	Configuration	Configure Logstash and apply Logstash to machines by using the configuration management system	Perform operations in the console or by using APIs, without using the configuration management system
	Collection point	Install configuration and Logstash on a machine group by using the configuration management system	Perform operations in the console or by using APIs, without using the configuration management system
Export	Method	API/SDK	API/SDK + Stream computing engines (Spark, Storm, Flink, and CloudMonitor) + Storage (OSS)

Item	Sub item	Self-built ELK	LogSearch/Analytics
Multi-tenant	Safety	None (non-commercial version)	HTTPS + Transmission signature + Multi-tenant isolation + Access control
	Decompress the file by using	Same account	Sub-account, role, product, and temporary authorization

Conclusion:

The ELK has many ecosystem and write tools and supports many installation and configuration tools. LogSearch/Analytics is a hosting service with a high degree of integration in terms of access, configuration, and usage. Normal users can access LogSearch/Analytics in five minutes, without worrying about the capacity and concurrency issues. The billing method is Pay-As-You-Go and auto scaling is supported.

Functions (query and analysis)

The query function enables quick hitting of logs that comply with search criteria. The analysis function performs statistics and computing of data.

For example, you have an analysis requirement that intends to collect statistics by IP address on the number and traffic of all read requests with a status code greater than 200. This analysis requirement can be converted to two operations: query specified results and perform statistical analysis of the results. In some cases, you can directly analyze all the logs without query.

```
1. Status in (200,500] and Method:Get*
 2. select count(1) as c, sum(inflow) as sum_inflow, ip group by Ip
```

Comparison of query capability

Type	Sub item	Self-built ELK	LogSearch/Analytics
Text	Index query	Supported	Supported
	Word segmentation	Supported	Supported
	Chinese word segmentation	Supported	Not supported
	Prefix	Supported	Supported
	TLD	Supported	
	Fuzzy	Supported	Supported

Type	Sub item	Self-built ELK	LogSearch/Analytics
	Wildcast	Supported	Not supported
Numeric value	long	Supported	Supported
	double	Supported	Supported
Nested	JSON query	Supported	
Geo	Geo query	Supported	Not directly supported . You can use the range query to have the same effect
Ip	IP address query	Supported	Not directly supported . You can use the string query to have the same effect
Context	Contextual Query		Supported
	Context filter		Supported

Elasticsearch supports more data types and more advanced query methods. LogSearch/Analytics supports most of the common queries with unique features (for example, context query and expansion of program logs).

Comparison of analysis capability

- *ES 5.5 aggregation*
- *#unique_10*

Type	Sub item	Self-built ELK	LogSearch/Analytics
Interface	Method	API/SDK	API/SDK + SQL92
	Other protocols		JDBC
Agg	Bucketing	Supported	Supported
	Metric	Supported	Supported
	Matrix	Supported	Supported
	Pipeline	Limited support	Full support
Arithmetic operation	Numeric value		Supported
	String		Supported
	Estimation		Supported
	Mathematical statistics		Supported

Type	Sub item	Self-built ELK	LogSearch/Analytics
	Date Conversion		Supported
GroupBy	Agg	Supported	Supported
	Having condition		Supported
Sort	Sort		Supported
Join	Join of multiple tables		Supported

LogSearch/LogAnalytics provides a superset of functions compared with Elasticsearch and fully supports SQL-92. LogSearch/LogAnalytics can be directly used in SQL writing scenarios.

Performance

By using the same data set, compare the self-built ELK and LogSearch/Analytics in terms of data writing, data query, and aggregation.

Hands-on Environment

1. Test configuration

Type	Self-built ELK	LogSearch/Analytics
Environment	Elastic Compute Service (ECS) instance (4 core and 16 GB) x 4 + Efficient SSD cloud disk	-
Shard	10	10
Number of copies	2	3 (configured by default and invisible to users)

2. Test data

   - Five columns of double-type data.
   - Five columns of long-type data.
   - Five columns of text-type data, with the dictionary sizes 256, 512, 768, 1024, and 1280, respectively.

   The preceding fields are random. The following is a sample test log:

```
timestamp:August 27th 2017, 21:50:19.000
  long_1:756,444 double_1:0 text_1:value_136
  long_2:-3,839,872,295 double_2:-11.13 text_2:value_475
 long_3:-73,775,372,011,896 double_3:-70,220.163 text_3:value_3
 long_4:173,468,492,344,196 double_4:35,123.978 text_4:value_124
```

```
long_5:389,467,512,234,496 double_5:-20,10.312 text_5:value_1125
```

3. Size of the data set

   - Size of raw data: 50 GB
   - Size of raw data with the key removed: 27 GB (LogSearch/Analytics uses this size as the unit of storage and billing.)
   - Number of log lines: 162,640,232 (about 160 million logs)

Write test results

Elasticsearch writes data in batches using the Bulk API, whereas LogSearch/LogAnalytics performs batch write using the PostLogstoreLogs API. The results are shown as follows:

Type	Item	Self-built ELK	LogSearch/Analytics
Latency	Average write latency	40 ms	14 ms
Storage	Data volume copied at a time	86 GB	58 GB
	Expansion rate: Data volume/Raw data size	172%	121%

Note:
The storage size of LogSearch/Analytics that generates bills includes the volume of compressed raw data that has been written (23 GB) and the indexing traffic (27 GB), amounting to 50 GB in total.

According to the test results:

- LogSearch/Analytics has a lower write latency (14 ms) than Elasticsearch (40 ms).
- Space: The size of raw data is 50 GB. The storage space expands because the test data is random. (In most real scenarios, the storage space after the compression is smaller than the size of raw data.) The storage space occupied by Elasticsearch expands to 86 GB, with an expansion rate of 172%, which is 58% more than the storage space occupied by LogSearch/Analytics.

Read (query + analysis) test

Test scenario

Use two common scenarios as an example: log query and aggregation. The average latency in the two cases is counted when concurrency is 1, 5, and 10, respectively.

1. Perform GROUP BY calculation on any text column of full data. Calculate the avg, min, max, sum, and count values of five columns and sort the values by count. The first 1,000 results are obtained. Example:

```
select count(long_1) as pv,sum(long_2),min(long_3),max(long_4),sum(
long_5)
group by text_1 order by pv desc limit 1000
```

2. For full data, randomly query logs by using a keyword, such as value_126. Obtain the number of logs that meet the query condition and the first 100 log lines. Example:

```
value_126
```

Test results

Type	Number of concurrencies	Latency (unit : seconds) of Elasticsearch	Latency (unit : seconds) of LogSearch/Analytics
Case1: Analysis class	1	3.76	3.4
	5	3.9	4.7
	10	6.6	7.2
Case 2: Query	1	0.097	0.086
	5	0.171	0.083
	10	0.2	0.082

Results Analysis

- According to the test results, for the scale of 150 million data, both Elasticsearch and LogSearch/Analytics can query and analyze data within seconds.
- In Case 1 (statistics), Elasticsearch and Log Service are at the same performance level in terms of latency. Elasticsearch with SSD cloud disks has I/O advantage over Log Service when reading large amounts of data.
- In Case 2 (query), LogSearch/Analytics has much lower latency than Elasticsearch. As concurrency increases, the latency of the ELK increases, while that of LogSearch/Analytics remains stable and even decreases.

Scale

1. LogSearch/Analytics can index petabytes of data in one day and query dozens of terabytes of data within seconds at a time, and supports auto scaling and horizontal scaling of data volume.

2. Elasticsearch is applicable to writing gigabytes to terabytes of data in one day and storing terabytes of data. The main limits are as follows:
   - Single cluster scale: The ideal condition is that one cluster contains about 20 machines. In the industry, one cluster can contain up to 100 nodes.
   - Expansion of write capability: The write capability cannot be modified after shards are created. Nodes are dynamically expanded when the throughput rate is increased. The maximum number of nodes that can be used is the number of shards.
   - Storage expansion: When the primary shard reaches the upper limit of disk capacity, it must be migrated to another disk with larger capacity, or more shards must be allocated. Generally, you can create an index, specify more shards, and rebuild existing data.

     LogSearch/Analytics does not have expansion issues because each shard is in distribute d storage. When the throughput rate is increased, shards can be dynamically split for horizontal scaling of the processing capability.

Cost

Based on the preceding test data, this section calculates the average monthly cost in the case that 50 GB data is written on a daily basis and stored for 90 days (the actual data size is 27 GB).

1. 1.The billing method of Log Service LogSearch/Analytics includes read and write traffic, indexing traffic, and storage space. The query function is free of charge. For more information, see *#unique_11*

Billing item	Value	Unit price	Cost (USD)
Read and write traffic	23 GB x 30	USD 0.2/GB	138
Storage space (data stored for 90 days)	50 GB x 90	USD 0.3/GB x Month	1350
Indexing traffic	27 GB x 30	USD 0.0875/GB	283
In total	-	-	1771

2. The Elasticsearch costs include the machine costs and the costs of SSD cloud disks used for data storage
   - Generally, cloud disks provide high reliability. Therefore, the storage of copies is not billed.
   - Generally, for storage disks, 15% available space must be reserved to avoid full space occupation by written data. Therefore, a factor of 1.15 is multiplied.

Billing item	Value	Unit price	Cost (USD)
Server	Server of 4 cores and 16 GB x 4 (three months) (ecs.mn4.xlarge)	Cost of monthly or yearly subscription: USD 675/month	2021
Storage	86*1.15*90 (only one copy is calculated here)	SSD: USD 1/GB x Month	8901
	-	SATA: USD 0.35/GB x Month	3115
In total			12943 (SSD)
			5135 (SATA)

With the same performance, the cost ratio of LogSearch/Analytics to the ELK (SSD) is 13.6%. During the test process, SSD is replaced with SATA to lower costs (the cost ratio of LogSearch/Analytics to the ELK with SATA is 34%). However, latency increases from 40 ms to 150 ms. After a long period of reading and writing, the query and read/write latency increases greatly and the query and analysis functions become abnormal.

concluding remarks

Compared with the open-source ELK, LogSearch/Analytics provides the same query speed but higher throughput, more robust analysis capability, and a 87% cost reduction, with a support for the Pay-As-You-Go billing method and zero O&M, allowing you to focus on business analysis.

In addition to LogSearch/Analytics, Log Service also provides the LogHub and LogShipper functions and supports real-time data collection and interconnection with stream computing systems (Spark, Storm, and Flink) and offline analysis systems (E-MapReduce, Presto, and Hive) to provide a one-stop real-time data solution.

3.4 Compare log query solutions

Compare Log Service against ELK (search class) and Hadoop/Hive in DevOps scenario

To handle the accelerating demand for software and service delivery, startup teams and big IT companies have switched or are switching to the DevOps mode. With the effective collaboration between developers and Operation & Maintenance (O&M) personnel, they implement the collaboration across departments, respond to customer requirements quickly, and conduct continuous delivery.

In the DeveOps mode, logs play an important support role in aspects such as problem investigat ion, security audit, and operation support. An appropriate log solution is important to DevOps.

Compare LogSearch against ELK and Hadoop/Hive solutions in the following aspects:

- When the user can perform query after the log is generated
- Query capability: The data volume scanned in unit time.
- Query function: The keyword query, condition combination query, fuzzy query, numerical comparison, and context query.
- Rapid response to rise of hundred times of traffic
- Cost: The cost per GB.
- Reliability: The log data is secure and will not be lost.

Common solutions and comparison

- Self-built ELK: Use Elastic, Logstash, and Kibana for comparison.
- Offline Hadoop + Hive: The data is stored in Hadoop, and Hive or Presto is used for query (not analysis).
- Use Log Service (LogSearch).

Compare these solutions by using application logs and Nginx access logs as an example (10 GB per day).

Function	ELK system	Hadoop + Hive	Log Service
Latency that can be queried	1–60 seconds (controlled by refresh_interval)	Several minutes to several hours	Real time
Query latency	Less than 1 second	In minutes	Less than 1 second
Super large query	Tens of seconds to several minutes	In minutes	In seconds (query one billion logs)
Keyword query	Supported	Supported	Supported
Fuzzy search	Supported	Supported	Supported
#unique_13	Not supported	Not supported	Supported
Context query	Supported	Supported	Supported
Consecutive string query	Supported	Supported	Not supported
Elasticity	Prepare machines in advance	Prepare machines in advance	10 times of expansion in seconds

Function	ELK system	Hadoop + Hive	Log Service
Write cost	USD 5/GB for write. No charge for query	No charge for write. USD 0.3/GB for one query	USD 0.5/GB for write. No charge for query
Storage cost	Less than or equal to USD 3.36/GB * day	Less than or equal to USD 0.035/GB * day	Less than or equal to USD 0.016/GB * day
Reliability	Set the number of copies	Set the number of copies	SLA > 99.9%. Data > 99.99999999%

4 Scenarios

Typical scenarios of Log Service include data collection, real-time computing, data warehousing and offline analysis, product operation and analysis, and Operation & Maintenance (O& M) and management. This document introduces some typical scenarios. For more scenarios, see Best practices.

Data collection and consumption

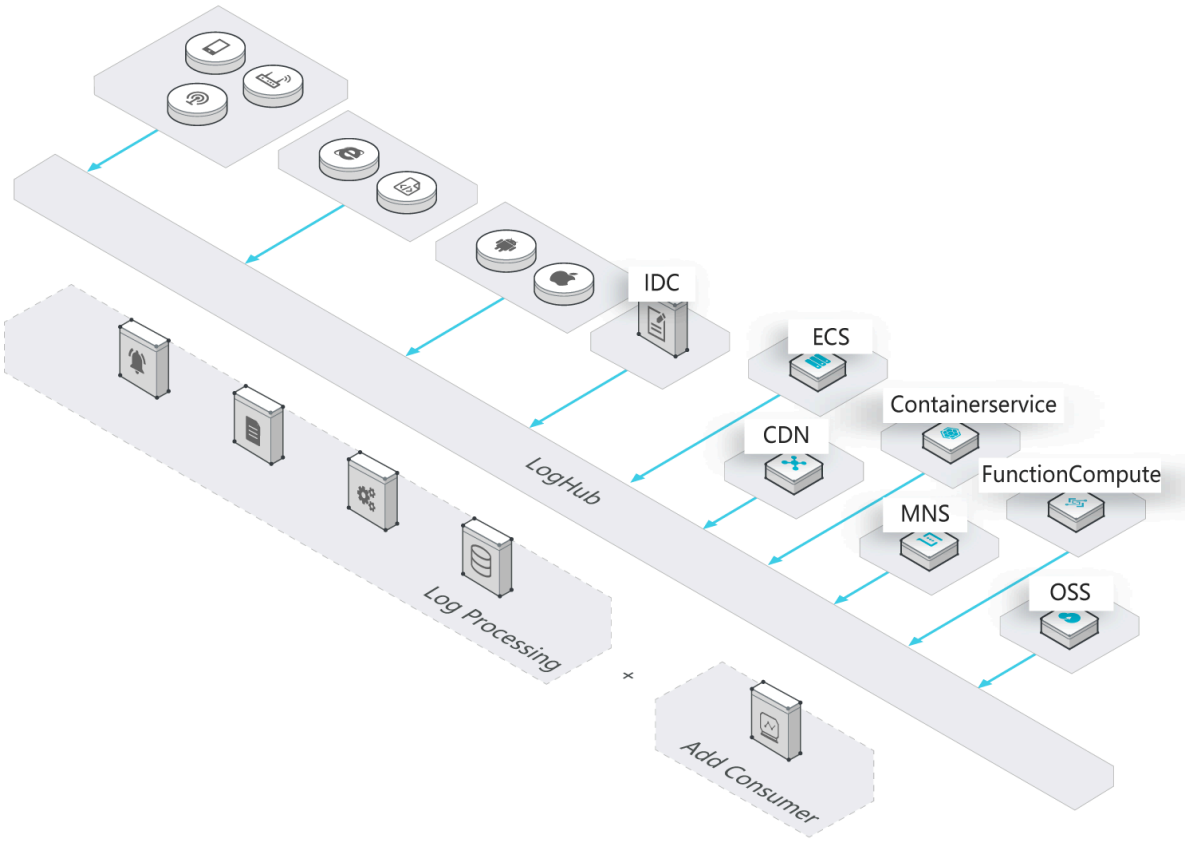
The LogHub function of Log Service enables access to massive real-time log data (including Metric, Event, BinLog, TextLog, and Click data) at the lower costs.

Advantages of the solution:

- Easy to use: Over 30 real-time data collection methods are provided for you to quickly build your platform. The powerful configuration and management capabilities can ease O&M workload. Nodes are available across China and the rest of the world.

- Auto scaling: It helps easily cope with traffic peaks and business growth.

Figure 4-1: Data collection and consumption



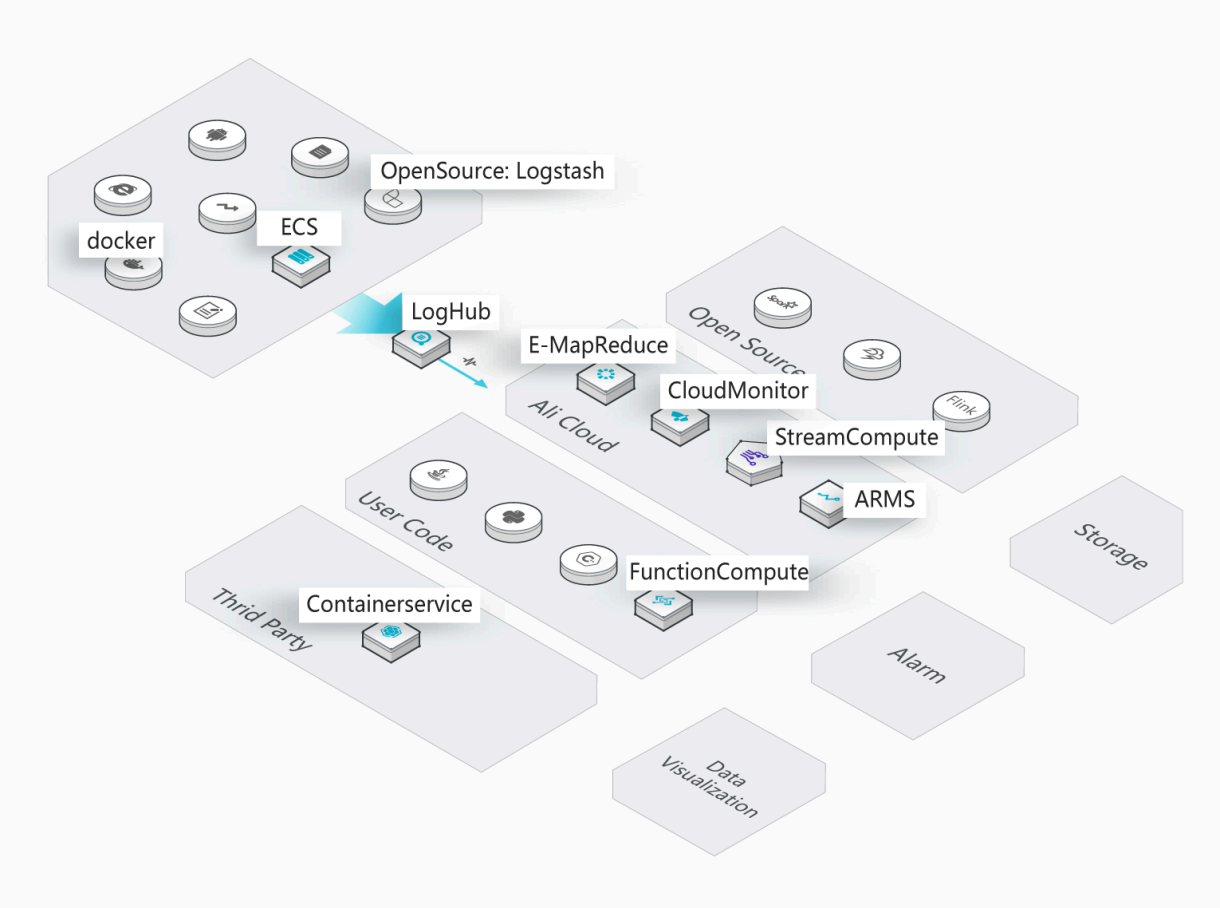
ETL/Stream Processing

LogHub can interconnect with various real-time computing and services, provides complete progress monitoring and alarm notification functions, and supports SDK/API-based custom consumption.

- Easy to operate: It provides various SDKs and programming frameworks and can interconnect with various stream computing engines seamlessly.
- Comprehensive functions: Rich monitoring data and delay alarm functions are provided.

- Auto scaling: PB-grade elasticity and zero latency.

Figure 4-2: Data cleaning and Flow Calculation



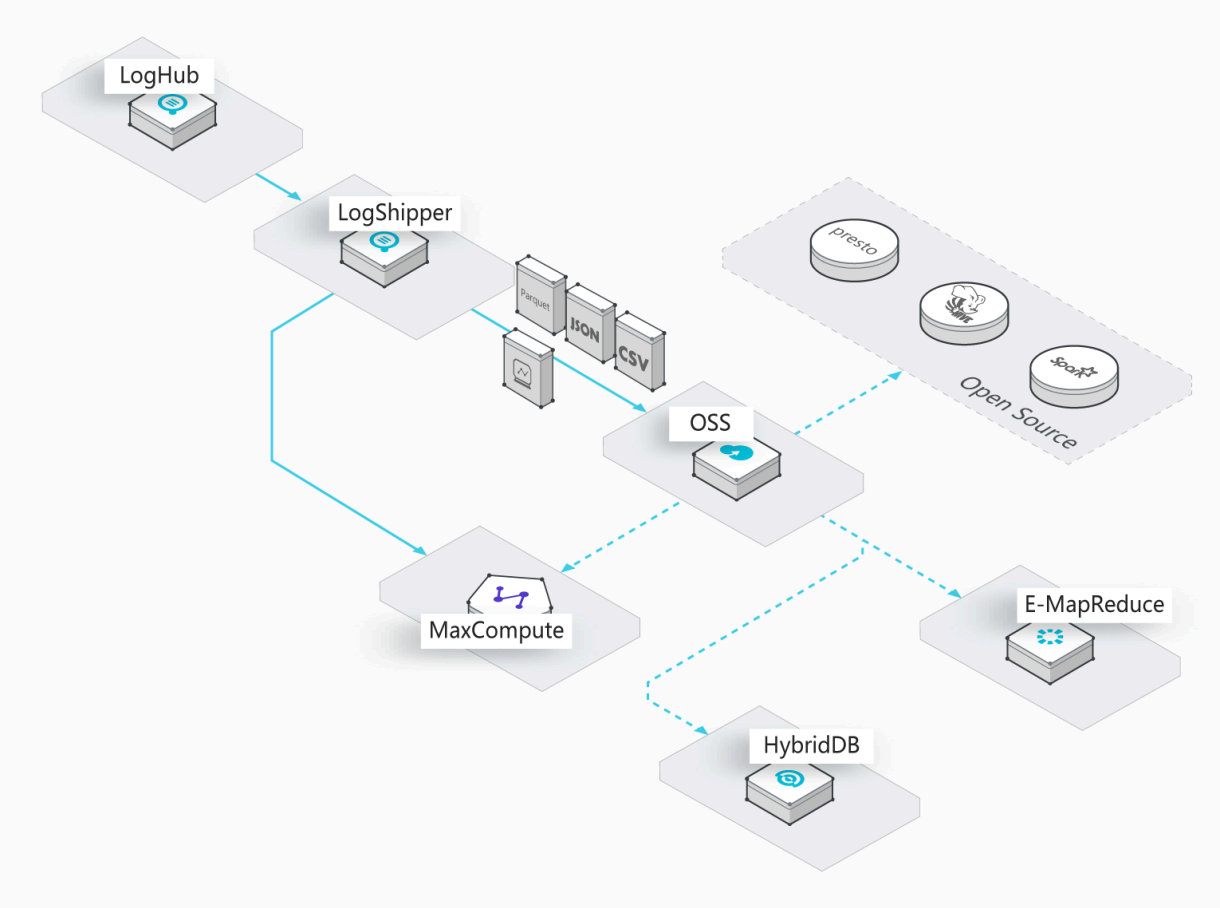
Data warehouse

LogShipper ships LogHub data to storage services and supports various storage formats such as compression, user-defined partitions, row storage, and column storage.

- Massive data: No upper limit is configured for the amount of data.
- Rich storage formats: Various storage formats are supported, such as row storage, column storage, and TextFile.

- Flexible configuration: Configurations such as user-defined partitions are supported.

Figure 4-3: Data Warehouse docking



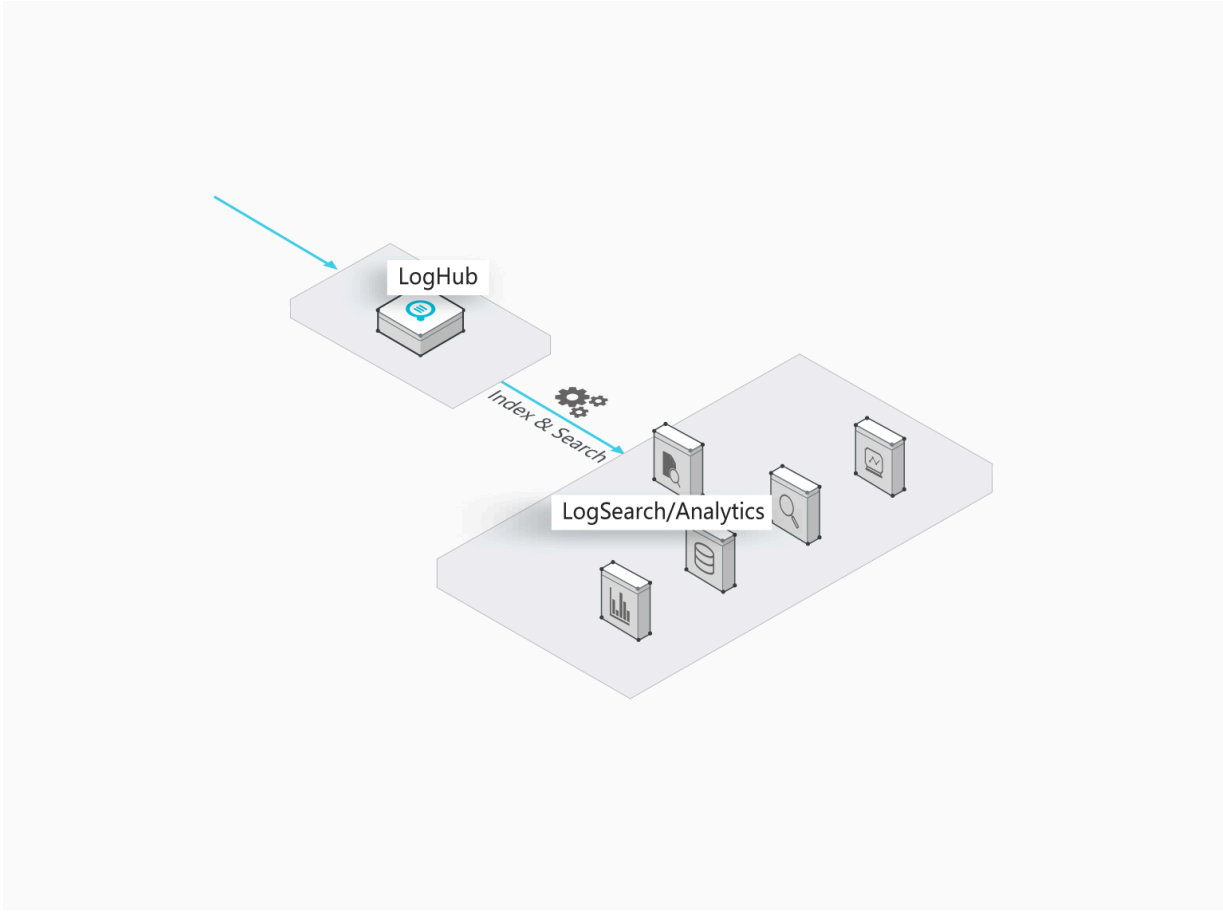
Real-time query and analysis of logs

LogAnalytics supports indexing LogHub data in real time and provides rich query methods such as keywords, fuzzy match, context, range, and SQL aggregation.

- Strong real-timeliness: Data can be queried after being written.
- Massive amount and low cost: Supports PB/day indexing capabilities, and the cost is 15% of the self-built solution.

- Strong analysis capabilities: Supports multiple query methods. Supports SQL aggregation and analysis. Visualization and alarm notification functions are provided.

Figure 4-4: Real-time query and analysis of logs



5 Basic concepts

5.1 Overview

Logs

Log is an abstraction of system changes during the running process. The log content is a time-ordered collection of some operations and the corresponding operation results of specified objects . LogFile, Event, BinLog, and Metric data are different carriers of logs. In LogFile, every log file is composed of one or more logs, and every log describes a single system event. A log is the minimum data unit processed in Log Service.

Log group

A log group is a collection of logs and the basic unit for writing and reading.

Log topic

Logs in a Logstore can be classified by log topics. Users can specify the topic when writing a log, and must specify the log topic when querying logs.

Project

A project is the Log Service's resource management unit, used to isolate and control resources . You can manage all the logs and the related log sources of an application by using projects. It manages all the Logstores of a user and configurations of log-collecting machines. It also serves as the portal by which users access the Log Service resources.

Logstore

The Logstore is a unit in Log Service for the collection, storage, and query of log data. Each Logstore belongs to a project, and each project can create multiple Logstores.

Partition

Each Logstore is divided into several shards and each shard is composed of an MD5 left-closed, right-open interval. These intervals do not overlap and the range of all intervals is the entire MD5 value range.

5.2 Log

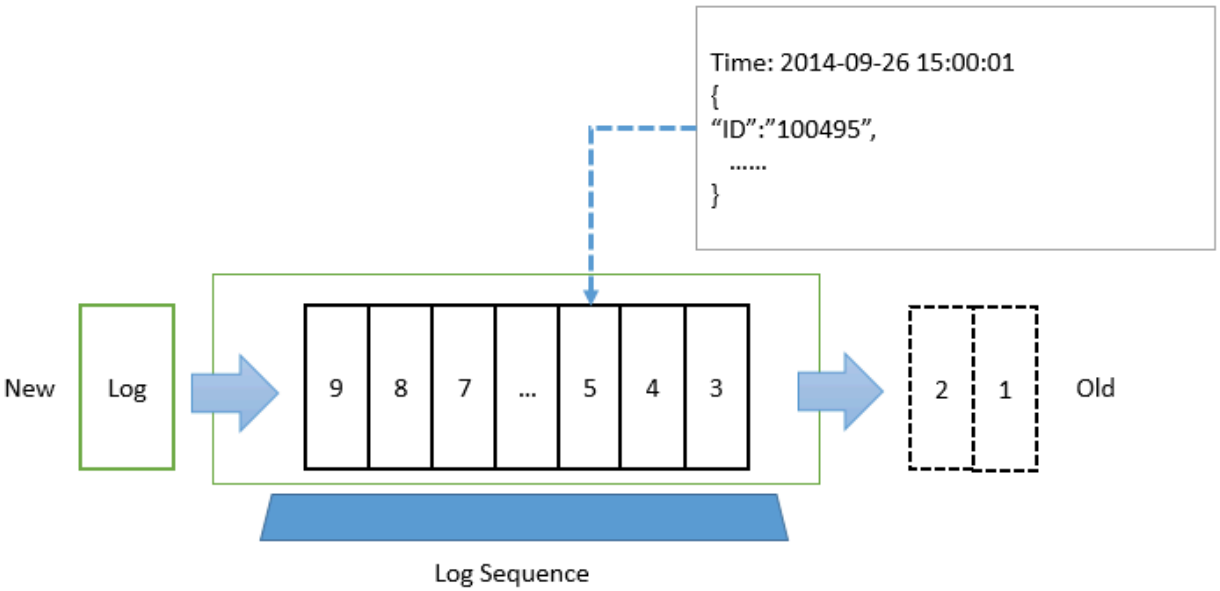
Half a century ago, the term “log” was associated with a thick notebook written by a ship captain or operator. Nowadays, with the advent of computers, logs are generated and used everywhere. Servers, routers, sensors, GPS devices, orders, and various IoT devices describe the world we live from different angles by generating and using logs. With the computing power, we continuously update our recognition to the whole world and system by collecting, processing, and using logs.

What is a log?

Consider an example of a ship captain’s log. In addition to a recorded timestamp, a log can contain almost all sorts of information, such as a text record, an image, weather conditions, and the sailing course. After centuries passed, now the “ship captain’s log” has been expanded to various areas such as orders, payment records, user accesses, and database operations.

The reason why logs are widely used and enduring is that logs are the simplest storage abstraction. Logs are a collection of chronological records that can only be added. The following figure is what logs (time-series data) look like.

Figure 5-1: Log



We can add a record to the end of a log and read the log records from left to right. Each record has a unique log record number with a sequence.

The log sequence is determined by “time”. From the preceding figure, we can see that the log time sequence is from right to left. The new event is recorded, and the old event is gradually out

of sight. But a log is a record of events. This is the foundation of recognition and reasoning to computers, humans, and the whole world.

Logs in Log Service

A log is an abstraction of system changes during the running process. The log content is a time-ordered collection of some operations and the corresponding operation results of specified objects . LogFile, Event, BinLog, and Metric data are different carriers of logs. In LogFile, every log file is composed of one or more logs, and every log describes a single system event. A log is the minimum data unit processed in Log Service

Log Service defines a log by using the semi-structured data mode. This mode includes the following four data fields: Topic, Time, Content, and Source.

Meanwhile, Log Service has different format requirements for different log fields. For more information, see the following table.

Data field	Meaning	Format
Topic	A custom field used to mark multiple logs. For example, access logs can be marked according to sites.	Any string up this field is a
Time	A reserved field in the log used to indicate the log generation time. Generally this field is generated directly based on the time in the log.	An integer in seconds. Th 1970-1-1 00:
Content	A field used to record the specific log content. The log content is composed of one or more content items, and each content item is a key-value pair.	The key is a contain letter a number or • `__time_` • `__source` • `__topic_` • `__parti` • `_extrac` • `__extra` The value ca
Source	A field used to indicate the source of the log. For example, the IP address of the machine where the log is generated.	Any string up

Various log formats are used in actual usage scenarios. For better understanding, the following example describes how to map an original Nginx access log to the Log Service log data model.

Assume that the IP address of your Nginx server is 10.249.201.117 . The following is an original log of this server.

```
10.1.168.193 - - [01/Mar/2012:16:12:07 +0800] "GET /Send? AccessKeyId=
8225105404 HTTP/1.1" 200 5 "-" "Mozilla/5.0 (X11; Linux i686 on x86_64
; rv:10.0.2) Gecko/20100101 Firefox/10.0.2"
```

Map the original log to the Log Service log data model as follows:

Data field	Content	Description
Topic	“”	Use the default value (null string).
Time	1330589527	The precise log generation time, indicating the number of seconds since 1970-1-1 00:00:00 UTC. The time is converted from the timestamp of the original log.
Content	Key-value pair	Specific log content.
Source	“10.249.201.117”	Use the IP address of the server as the log source.

You can decide how to extract the original log contents and combine them into key-value pairs. The following table is shown as an example.

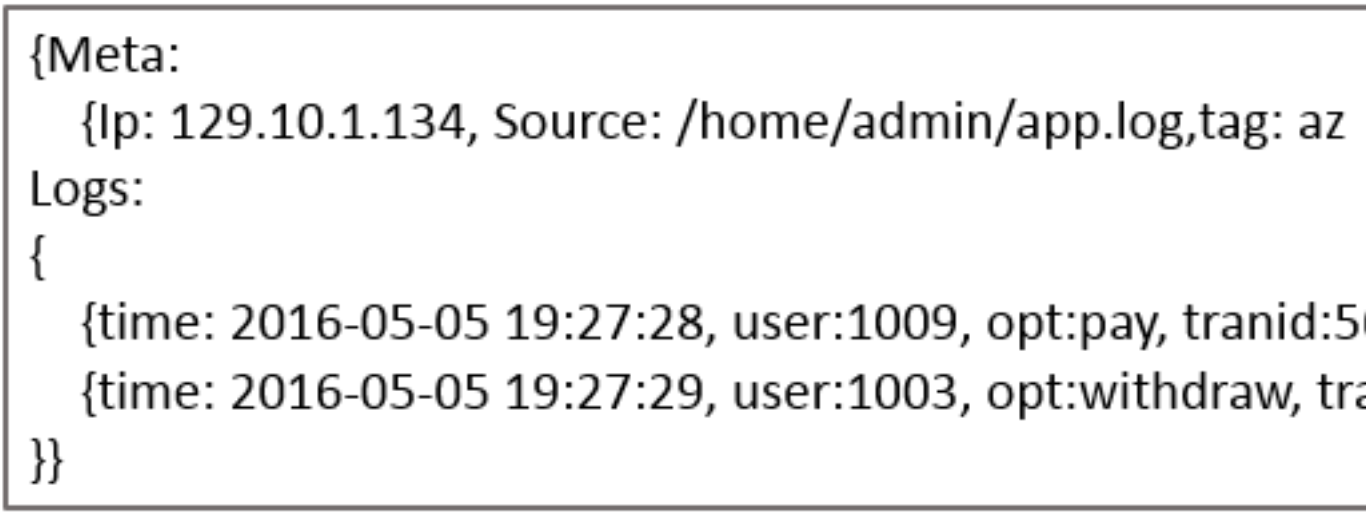
Key	Value
ip	“10.1.168.193”
method	“GET”
Status	“200”
length	“5”
ref_url	“-“
browser	“Mozilla/5.0 (X11; Linux i686 on x86_64; rv:10.0.2) Gecko/20100101 Firefox/10.0.2”

Log Group

A log group is a collection of logs and is the basic unit for writing and reading.

The maximum capacity of a log group is up to 4096 logs or 5 MB.

Figure 5-2: Log Group



5.3 Project

The project is the resource management unit in Log Service and is used to isolate and control resources. You can manage all the logs and the related log sources of an application by using projects. Projects manage the information of all your Logstores and the log collection machine configuration, and serve as the portals where you can access the Log Service resources.

Specifically, projects provide the following functions:

- Projects help you organize and manage different Logstores. In actual use, you might use Log Service to centrally collect and store the logs of the different projects, products, or environments. You can classify different logs for management in different projects to facilitate subsequent usage, export, or index of logs. In addition, projects are the carriers of the log access permission management.
- Projects serve as the portals where you can access the Log Service resources. Log Service allocates a unique access point for each created project. The access point supports writing, reading, and managing logs by using the network.

5.4 Logstore

The Logstore is a unit in Log Service to collect, store, and query the log data. Each Logstore belongs to a project, and each project can create multiple Logstores. You can create multiple Logstores for a project according to your actual needs. Typically, an independent Logstore is

created for each type of logs in an application. For example, you have a game application “big-game”, and three types of logs are on the server: operation_log, application_log, and access_log . You can first create a project named “big-game”, and then create three Logstores under this project for these three types of logs to collect, store, and query logs respectively.

You must specify the Logstore for writing and querying logs. If you want to deliver log data to maxcompute for offline analysis, its data delivery is also based on the logstore as a unit for data synchronization, that is, The log data in the logstore is delivered to a maxcompute table.

Specifically, Logstores provide the following functions:

- Log collection, supports real-time logging.
- Log storage, supports real-time consumption.
- Index creation, supports real-time log query.
- Provides data channels delivered to maxcompute

5.5 Shard

Logstore read/write logs must be stored in a certain shard. Each Logstore is divided into several shards and each shard is composed of MD5 left-closed and right-open intervals. Each interval range does not overlap with others and the total range of all the intervals is the entire MD5 value range.

Range

You must specify the number of shards when creating a Logstore. Then, the entire MD5 range is automatically divided evenly according to the specified number of shards. Each shard has a range, which can be expressed in the MD5 mode and must be within the following value range: [00000000000000000000000000000000,ffffffffffffffffffffffffffffffff).

All of the shard ranges are left-closed and right-open intervals, and composed of the following keys:

- BeginKey: Indicates the start of the shard. This key is included in the shard range.
- EndKey: Indicates the end of the shard. This key is excluded from the shard range.

With the shard range, you can write logs by specifying Hash Key, split shards, and merge shards . To read data from a shard, you must specify the corresponding shard. To write data to a shard, you can use Server Load Balancer or specify the Hash Key. By using Server Load Balancer, each data packet is written to an available shard at random. By specifying the Hash Key, data is written to the shard whose range includes the specified key. To read data from a shard, you must specify

the corresponding shard. To write data to a shard, you can use Server Load Balancer or specify the Hash Key. By using Server Load Balancer, each data packet is written to an available shard at random. By specifying the Hash Key, data is written to the shard whose range includes the specified key.

For example, a Logstore has four shards and the MD5 value range of this Logstore is [00,FF). Each shard range is as follows.

Shard No.	Range
Shard0	[00,40)
Shard1	[40,80)
Shard2	[80,C0)
Shard3	[C0,FF)

If you specify the MD5 key as 5F by specifying the Hash If you specify the MD5 key as 5F by specifying the Hash Key when writing logs, the log data is written to Shard1 that contains the MD5 key 5F. If you specify the MD5 key as 8C, the log data is written to Shard2 that contains the MD5 key 8C.

Read/write capacities

Each shard has certain service capacities:

- Writing: 5 MB/s, 500 times/s
- Read: 10 MB/s, 100 times/s

We recommend that you plan the number of shards according to the actual data traffic. If the traffic exceeds the read/write capacities, split the shard in time to increase the number of shards so as to achieve greater read/write capacities. If the traffic is far less than the maximum read/write capacities of shards, we recommend that you merge the shards to reduce the number of shards so as to save the rental costs of shards.

For example, assume that you have two shards in readwrite status and can write data at 10 MB/s at maximum. If you write data at 14 MB/s in real time, we recommend that you split a shard to make the number of shards in readwrite status reach three. If you write data at only 3 MB/s in real time, we recommend that you merge these two shards because one shard can meet the needs.

 **Note:**

- If the API consistently reports error 403 or 500 during the writing, see Log Service monitoring metrics to determine whether to increase the number of shards.
- For read/write operations that exceed the service capacities of shards, the system attempts to provide the needed services, but the service quality cannot be guaranteed.

Status

The shard status includes:

- readwrite: Supports reading and writing data.
- readonly: Only supports reading data.

When a shard is created, all the shards are in readwrite status. Split or merge operations change the shard status to readonly and generate a new shard in readwrite status. The shard status does not affect the performance of reading data. Shards in readwrite status maintain normal data writing performance, while shards in readonly status do not support writing data.

When splitting a shard, you must specify a ShardId in readwrite status and an MD5. The MD5 must be greater than the shard BeginKey and less than the shard EndKey. Split operations can split two other shards from one, that is, the number of shards is increased by 2 after the split. After the split, the status of the original shard specified to be split is changed from readwrite to readonly. Data can still be consumed, while new data cannot be written. The two newly generated shards are in readwrite status and arranged behind the original shard. The MD5 range of these two shards covers the range of the original shard.

When merging shards, you must specify a shard in readwrite status. Make sure the specified shard is not the last shard in readwrite status. The server automatically finds the adjacent shard at the right of the specified shard and merges these two shards. After the merge, the specified shard and the adjacent shard on the right are in readonly status. Data can still be consumed, while new data cannot be written. A new shard in readwrite status is generated and its MD5 range covers the total range of the original two shards.

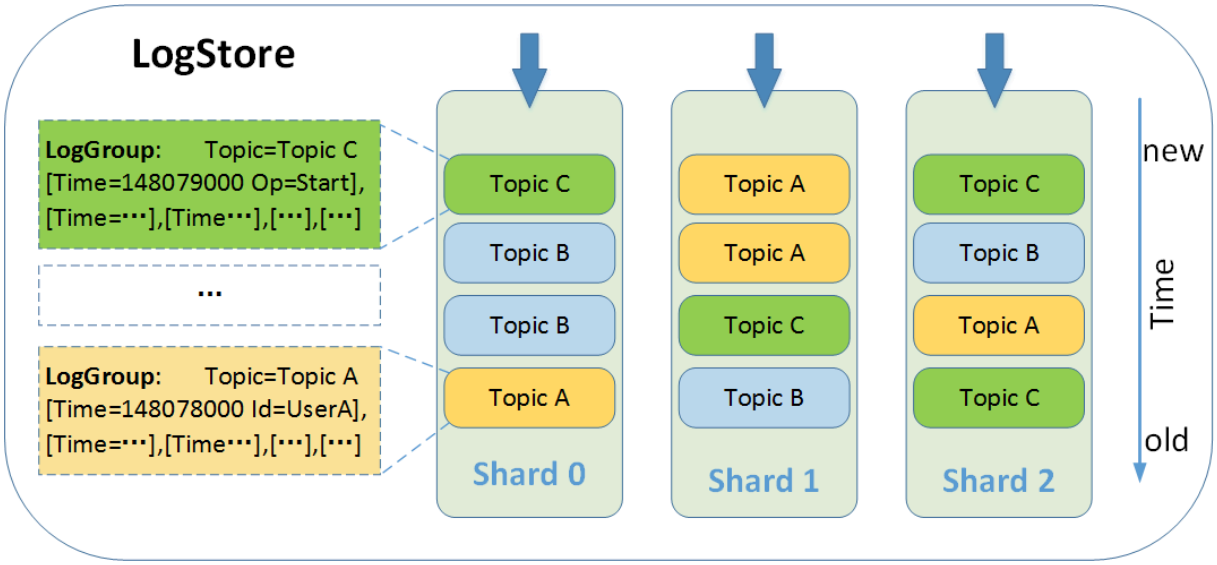
5.6 Log topic

Logs in a Logstore can be classified by log topics. You can specify the topic when writing and querying logs. For example, as a platform user, you can use your user ID as the log topic when writing logs. In this way, you can select to only view your own logs based on the log topic when querying logs. If you do not need to classify the logs in a Logstore, use the same topic for all of the logs.

Note:

A null string is a valid log topic and is the default log topic when writing and querying logs. So if you do not need to use the log topic, the easiest way is to use the default log topic, the null string, when writing and querying logs.

The relationship among Logstores, log topics, and logs is as follows.



6 Limits

6.1 Basic resources

Resources	Limit	Note
Project	Up to 50 projects can be created for each account.	If you have an extra demand, please open a ticket to apply for support.
Logstore	Up to 200 Logstores can be created in each project.	If you have an extra demand, please open a ticket to apply for support.
Shard	• Up to 200 shards can be created in each project. • Up to 10 shards can be created in each Logstore . You can increase the number of shards by splitting shards.	If you have an extra demand, please open a ticket to apply for support.
LogtailConfig	Up to 100 LogtailConfigs can be created for each project.	If you have an extra demand, please open a ticket to apply for support.
Log storage time	Permanent storage is supported. You can also customize the log storage time in the range of 1 to 3000.	-
Machine group	Up to 100 machine groups can be created for each project.	If you have an extra demand, please open a ticket to apply for support.
Consumer group	Up to 10 consumer groups can be created for eachLogstore.	You can delete unused consumer groups.
Quick query	Up to 100 quick queries can be created for each project.	-

Resources	Limit	Note
Dashboard	• Up to 50 dashboards can be created for each project. • Each dashboard can contain up to 50 analysis charts.	-
LogItem	The maximum length of a LogItem is 1 MB.	1 MB is for the API parameter . If Logtail is used to collect logs, the maximum length for a single LogItem is 512 KB.
LogItem (Key)	The maximum length is 128 bytes.	-
LogItem (Value)	The maximum length is 1 MB.	-
Log group	Each log group contains up to 4096 logs and the maximum length of a log is 10 MB.	-

6.2 Data read and write

Resource	Limit	Description	Note
Project	Write traffic protection	The write traffic is up to 30 GB/min.	If the limit is exceeded , the status code of 403 is returned , prompting Inflow Quota Exceed. If you have an extra demand , please open a ticket to apply for support.
	Number of writes protection	The maximum number of writes is 600000 per minute.	If the limit is exceeded , the status code of 403 is returned, prompting Write QPS Exceed. If you have an extra demand, please open a ticket to apply for support.

Resource	Limit	Description	Note
	Number of reads protection	The maximum number of reads is 600000 per minute.	If the limit is exceeded, the status code of 403 is returned, prompting Read QPS Exceed. If you have an extra demand, please open a ticket to apply for support.
Shard	Write traffic	The maximum write traffic is 5 MB/s.	Not required. When the limit is exceeded, the system serves as much as possible, but does not guarantee the service quality.
	Number of writes.	The maximum number of writes is 500 per second.	Not required. When the limit is exceeded, the system serves as much as possible, but does not guarantee the service quality.
	Read traffic	The maximum read traffic is 10 MB/s.	Not required. When the limit is exceeded, the system serves as much as possible, but does not guarantee the service quality.
	Number of reads	The maximum number of reads is 100 per second.	Not required. When the limit is exceeded, the system serves as much as possible, but does not guarantee the service quality.

6.3 Search, analysis, and visualization

Function	Item	Limit	Note
Query	Number of keywords	The number of conditions specified for querying words besides Boolean logical operators. You can query up to 30 keywords each time.	For example, "a and b or c and d...".
	The length of a single value.	The maximum length of a single value is 10 KB. The excess part of the value is not queried.	If the length of a single value is greater than 10 KB, the log might not be found through keywords, but the data is still complete.
	Single project concurrency	The number of single project concurrency is up to 100.	-
	Number of entries of returned query results	By default, a maximum of 100 entries of query results are returned each time.	You can read the full query results by turning pages.
	Single Log content display	For logs exceeding 10,000 characters , Log service only processes the first 10,000 characters using the DOM word segmentation due to Web browser performance.	-
SQL analysis	Maximum length of a single value	The maximum length of a single value is 2 KB. The excess part of the value is not queried.	Query results might not be accurate when the limit is exceeded , but the data is still complete.
	Single project concurrency	The number of single project concurrency is up to 15.	-

Function	Item	Limit	Note
	Number of entries of results in each analysis	Results returned by each analysis are up to 100 MB or 100000 entries.	-

6.4 Reserved fields

In Log Service, some fields are reserved fields. When you use APIs to write data to logs or add Logtail Configs, the names of the required fields cannot be the same as those of the reserved fields.

Precautions

When collecting logs or delivering data to other cloud products, Log Service can add information, such as log sources and timestamps, in Key-Value format to logs. Fields with fixed names, for example, `__source__`, are reserved fields.

- When using APIs to write data to logs or adding Logtail Configs, do not set the names of the required fields to be the same as those of the reserved fields. Otherwise, your queries may be inaccurate as a result.
- Fields with a prefix of `__tag__` cannot be delivered.

Reserved fields

The following table describes the reserved fields.

Table 6-1: Reserved fields

Reserved field	Type	Index and statistics settings	Description
`__time__`	Integer in standard Unix time format, for example, `__time__: 1523868463`	• Index settings: You do not need to add an index for this field because the field can be set through the **`from`** and **`to`** parameters in APIs. • Statistics settings: By default, statistics for this field are enabled after you enable the statistics function for any other column.	This field specifies the log generation time when you use APIs or SDKs to write data to logs. It can be used for log delivery, query, and analysis.
`__source__`	String	• Index settings: After the index function is enabled, Log Service creates an index for this field by default. The index is of the text type, and no delimiter is specified. If you want to query this field, enter `source:127.0.0.1` or `__source__:127.0.0.1`. • Statistics settings: By default, statistics for this field are enabled after you enable the statistics function for any other column.	This field specifies the device from which logs are collected. It can be used for log delivery , query, analysis, and custom consumption.

Reserved field	Type	Index and statistics settings	Description
`__topic__`	String	• Index settings: After the index function is enabled, Log Service creates an index for this field by default. The index is of the text type, and no delimiter is specified. If you want to query this field, enter `__topic__:xxx`. • Statistics settings: By default, statistics for this field are enabled after you enable the statistics function for any other column.	This field specifies the log topic. If you have set a *log topic*, Log Service automatically adds a field to your log with the Key set to `__topic__` and Value set as the topic content you specified. This field can be used for log delivery, query, analysis, and custom consumption.
`__partition_time__`	String	You do not need to add an index for this field because the field does not exist in any log.	This field specifies the time of a partition for log delivery to MaxCompute. This field is calculated by `__time__` and is used to set the date format partition column when logs are delivered to MaxCompute. For more information, see *#unique_28*.
`__extract_others__`	String that can be deserialized into a JSON map	You do not need to add an index for this field because the field does not exist in any log.	This field specifies the JSON map consisting of the unset fields during log delivery to MaxCompute. It is used to pack the fields that are not separately set during log delivery to MaxCompute. For more information, see *#unique_28*.
`_extract_others_`	String that can be deserialized into a JSON map	You do not need to add an index for this field because the field does not exist in any log.	This field works the same as `__extract_others__`. We recommend that you use `__extract_others__`.

Reserved field	Type	Index and statistics settings	Description
`__tag__:__client_ip__`	String	• Index settings: After the index function is enabled, Log Service creates indexes for all *tags* by default. The index is of the text type, and no delimiter is specified. Both accurate search and fuzzy search are supported. • Statistics settings: By default, the statistics function is disabled for the column indicated by this field. If you want to enable statistics for this field, add an index for the field and then enable the statistics function.	This field is a system tag and specifies the Internet IP address of the device from which logs are collected. After the *recording Internet IP addresses* function is enabled, the server adds this field for a raw log after receiving the log. This field can be used for log query, analysis, and custom consumption.
`__tag__:__receive_time__`	String that can be converted to integer in standard Unix time format	• Index settings: After the index function is enabled, Log Service creates indexes for all *tags* by default. The index is of the text type, and no delimiter is specified. Both accurate search and fuzzy search are supported. • Statistics settings: By default, the statistics function is disabled for this column. If you want to enable statistics for this field, add an index for the field and then enable the statistics function.	This field is a system *tag* and specifies the time when the server receives a log. After the *recording Internet IP addresses* function is enabled, the server adds this field for a raw log after receiving the log. This field can be used for log query, analysis, and custom consumption.

Reserved field	Type	Index and statistics settings	Description
`__tag__:__path__`	String	• Index settings: After the index function is enabled, Log Service creates an index for this field by default. The index type is text, and no delimiter is specified. If you want to query this field, enter `__tag__:__path__:XXX`. • Statistics settings: By default, statistics for this field are enabled after you enable the statistics function for any other column.	This field specifies the log file path collected by Logtail. Logtail automatically adds this field to logs. It can be used for log query, analysis, and custom consumption.
`__tag__:__hostname__`	String	• Index settings: After the index function is enabled, Log Service creates an index for this field by default. The index is of the text type, and no delimiter is specified. If you want to query this field, enter `__tag__:__hostname__:XXX`. • Statistics settings: By default, statistics for this field are enabled after you enable the statistics function for any other column.	This field specifies the name of the host from which Logtail collects data. Logtail automatically adds this field to logs. It can be used for log query, analysis, and custom consumption.

Reserved field	Type	Index and statistics settings	Description
`__raw_log_ _`	String	You need to add and set an index of the text type for this field and enable the statistics function as needed.	This field specifies raw logs with parsing failure. After the *discarding logs with parsing failure* function is disabled, Logtail uploads raw logs once log parsing fails. In this field, Key is `__raw_log__` and Value is the log content. This field can be used for log delivery, query, analysis, and custom consumption.
`__raw__`	String	You need to add and set an index of the text type for this field and enable the statistics function as needed.	This field indicates raw logs that are successfully parsed. After the *uploading raw logs* function is enabled, Logtail regards raw logs as this field and upload the logs with the logs that are successfully parsed. Generally, this field is used for log audit and compliance check. It can also be used for log delivery, query, analysis, and custom consumption.