Alibaba Cloud Log Service

Product Introduction

Issue: 20190815

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 intellectual 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 instructions, 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]*

Style	Description	Example
{} 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

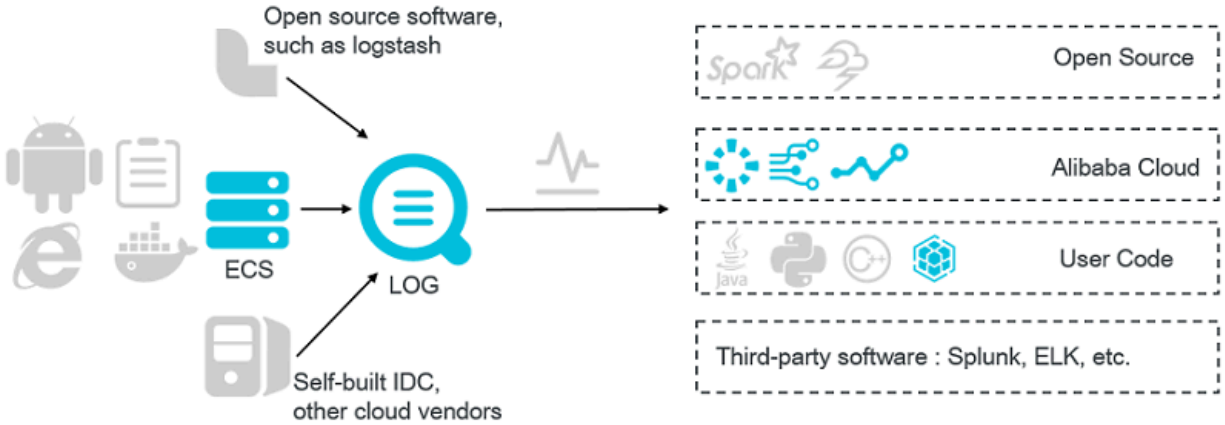
Log Service learning path recommends documents of hot functions, and helps you quickly have a knowlege of Log Service. Combined with video and documents, Log Service learning path optimizes the user experience of user and document reading experience.

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.

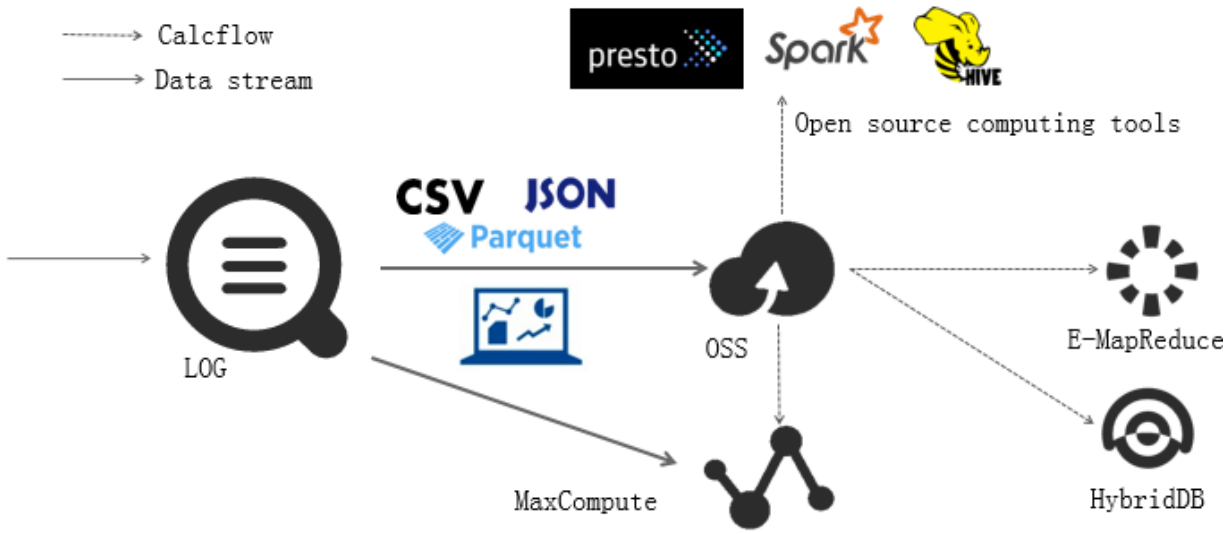
Purposes: ETL, Stream Compute, monitoring and alarm, machine learning, and iterative computing.



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.

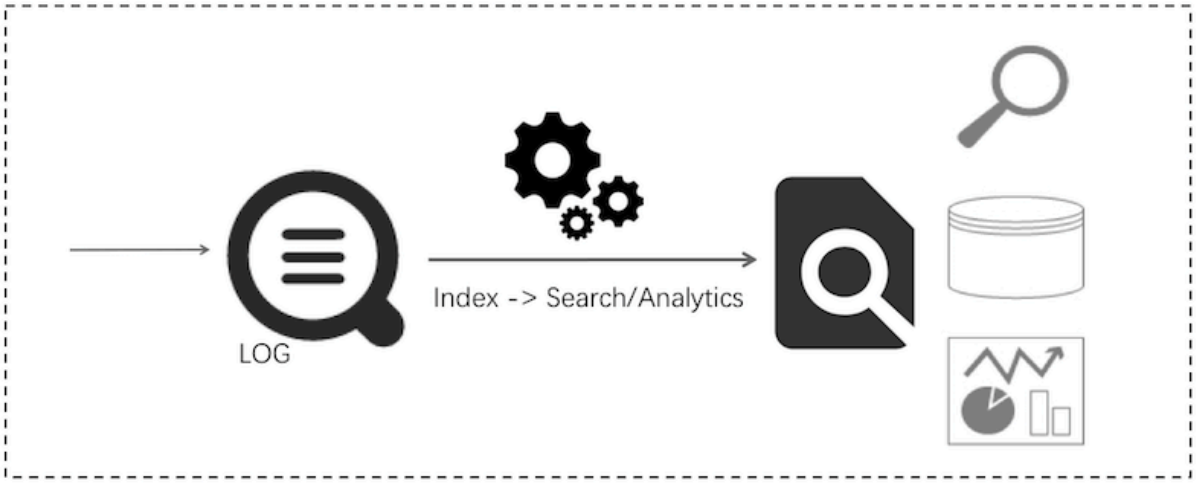


Query and real-time analysis (Search/Analytics)

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.

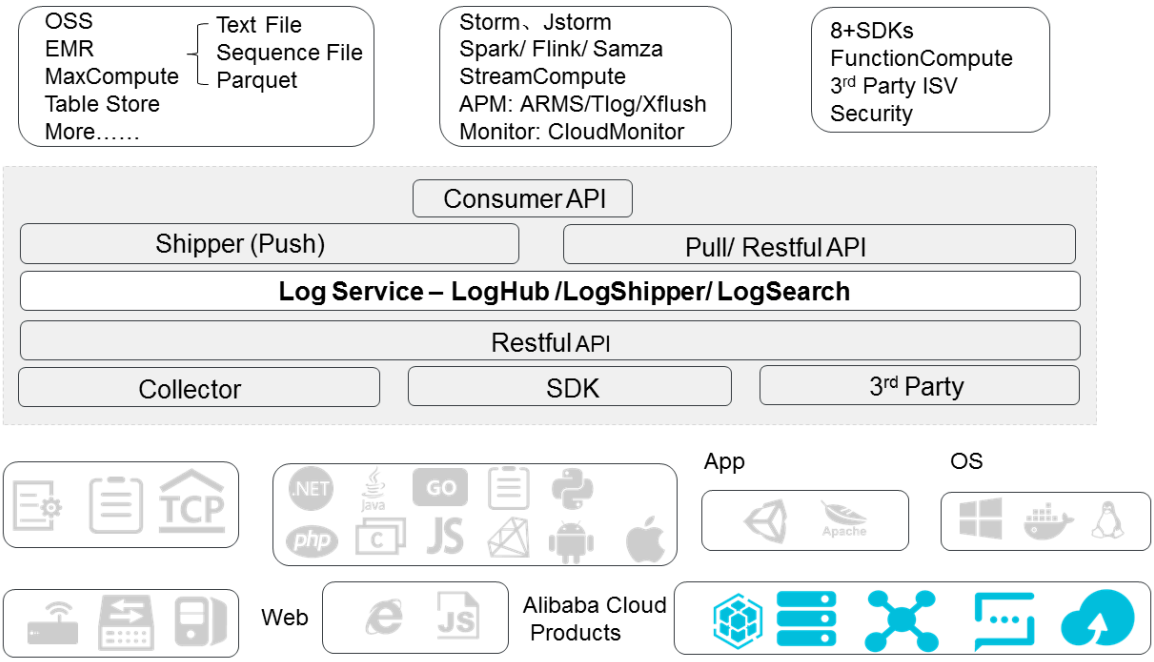
Purposes: DevOps/online O&M, real-time log data analysis, security diagnosis and analysis, and operation and customer service systems.


LOG
Index -> Search/Analytics

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

Focus-		LogSearch	ES (Lucene Based)	NoSQL	Hive
	Query (US $/GB)	0	0	0.2	0.3
	Speed-query	Millisecond level-second level	Millisecond level-second level	Within milliseconds	In minutes
	Speed-statistics	Weak+	Relatively strong	Weak	Strong
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.Compare LogSearch/Analytics with ELK in log query and analysis

3.3 Comparison with the ELK Stack in log query and analysis

Background

To realize real-time log analysis, many people may think of using the popular Elasticsearch, Logstash, and Kibana (known as the ELK Stack) to build a project. The ELK Stack is an open-source solution that has accumulated a large number of topics and use cases in the community.

Alibaba Group provides Alibaba Cloud Log Service as a solution to log scenarios. Its predecessor was a monitoring and diagnosis solution that came into shape in early 2012 when Alibaba Cloud was researching and developing the Apsara system. As the number of users grows and the business evolves, Alibaba Cloud has applied Log Service to log analysis in Ops scenarios, such as DevOps, Market Ops, and DevSecOps. During its development, Log Service has taken on challenges in scenarios such as

the Double 11 Shopping Festival, Ant Financial Double 12 Shopping Festival, Spring Festival red envelopes, and international business. It is now a product that serves both Alibaba Group and external users.

Orientation for log query and analysis

As an open-source search engine software library supported by the Apache Software Foundation, Apache Lucene provides full-text searching and indexing and text analysis capabilities. Lucene was originally written by Doug Cutting, who is also a founder of Apache Hadoop. Lucene joined the Apache Software Foundation in 2001. Founded in 2012, Elastic developed Elasticsearch based on the Lucene library and launched the ELK Stack in 2015 as an integrated solution to log collection, storage, and query. Lucene was designed to retrieve information based on documents. It has only limited log processing capabilities. For example, Lucene provides limited log processing scale, query performance, and support for custom functions such as LogReduce. Alibaba Cloud Log Service uses a self-developed log storage engine. In the past three years, Alibaba Cloud has applied Log Service to tens of thousands of applications. Log Service supports indexing for data in units of PB per day and serves tens of thousands of developers to query and analyze data hundreds of millions of times per day. In Alibaba Group, Log Service serves all Alibaba Cloud products, such as SQL audit, EagleEye, Cloud Map, sTrace, and Ditecting.

Log query is the most basic requirement of DevOps. According to the industry research report 50 Most Frequently Used UNIX/Linux Commands, the tar and grep commands rank the first and second, respectively. This shows the importance of log query to programmers.

Here, we make a comprehensive comparison of the application of Log Service and the ELK Stack in log query and analysis scenarios from the following aspects:

- Ease of use: the cost when you get started and use the solution.
- Features (key point): the query and analysis capabilities.
- Performance (key point): the query and analysis requirements for unit data volume and how is the latency.
- Scale (key point): the data volume that can be processed and the scalability.
- Cost (key point): the cost for using the same feature and performance.

Ease of use

The use of a log analysis system involves the following phases:

1. Collection: writes data in a stable manner.
2. Configuration: configures the data source.
3. Resizing: uses more data sources and machines to extend the storage space and scale out machines.
4. Usage: provides query and analysis features, which are described in the Features section of this topic.
5. Export: exports data to other systems for further processing, such as stream computing in StreamCompute and backup in Object Storage Service (OSS).
6. Multi-tenancy: shares data with other users and uses data securely.

The following table lists the comparison results between Log Service and the ELK Stack in terms of ease of use.

Item	Sub-item	On-premises ELK Stack	Log Service
Collection	Protocol	RESTful API	· RESTful API · Java Database Connectivity (JDBC)
	Client	Various products in the ecosystem, including Logstash, Beats, and Fluentd	· Logtail, which is the major component · Others, such as Logstash
Configuration	Unit	Uses indexes to differentiate logs.	· Project · Logstore Uses a project to function as a namespace and contain multiple Logstores.
	Attribute	API + Kibana	· API + SDK · Console

Item	Sub-item	On-premises ELK Stack	Log Service
Resizing	Storage	· Adds machines. · Purchases cloud disks.	No operation is needed.
	Computing	Adds machines.	No operation is needed.
	Configuration	· Configures machines in the configuration management system. · Logstash has supported centralized configuration in a beta version.	Uses the console or API, without the need for a configuration management system.
	Collection point	Installs configuration and Logstash on machine groups in the configuration management system.	Uses the console or API, without the need for a configuration management system.
	Capacity	Does not support dynamic resizing.	Supports dynamic resizing and auto scaling.
Export	Method	· API · SDK	· API · SDK · Kafka consumer API · Stream computing engines, such as Spark, Storm, and Flink · Stream computing class libraries, such as Python and Java

Item	Sub-item	On-premises ELK Stack	Log Service
Multi-tenancy	Security	Commercial version	· HTTPS · Transmission with a signature · Multi-tenant data segregation · Access control
	Throttling	No throttling	· Project-level throttling · Shard-level throttling
	Multi-tenancy	Kibana support	Native account and permission management

Conclusion:

- The ELK Stack has extensive support from various products in the ecosystem and supports many write, installation, and configuration tools.
- Log Service is a highly integrated hosting service that is easy to access, configure, and use. Common users can access Log Service within 5 minutes.
- Log Service is a software as a service (SaaS)-based service. You do not need to worry about capacity or concurrency. It supports auto scaling and is O&M-free.

Features (query and analysis)

The query feature quickly hits logs that meet query criteria. The analysis feature collects statistics and computes 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 an HTTP status code greater than 200. This analysis requirement can be converted to two operations: querying specified results and conducting statistical analysis for the results. In some cases, you can directly analyze all logs without a prior query.

```
1 .  Status   in  ( 200 , 500 ]  and   Method : Get *
```

```
2 . select count ( 1 ) as c , sum ( inflow ) as sum_inflow
, ip group by Ip
```

- Basic query capability

  The following table lists the comparison results based on Elasticsearch 6.5 Indices.

Type	Sub-type	On-premises ELK Stack	Log Service
Text	Query by index	Supported	Supported
	Word-breaking	Supported	Supported
	Chinese word-breaking	Supported	Supported
	Prefix	Supported	Supported
	Suffix	Supported	Not supported
	Fuzzy	Supported	Supported by SQL
	Wildcast	Supported	Supported by SQL
Numeric	Long	Supported	Supported
	Double	Supported	Supported
Nested	JSON	Supported	Not supported
Geo	Geo	Supported	Supported by SQL
IP address	Query by IP address	Supported	Supported by SQL

  Conclusion:

  - The ELK Stack supports more data types and provides a stronger native query capability than Log Service.
  - Log Service can use SQL statements in replacement of fuzzy string matching and comparison functions such as Geo. However, the query performance is slightly worse than the native query performance. The following examples show how to use SQL statements to query data:

```
Queries data that hits the specified substring :
* | select content where content like '% substring %'
limit 100

Queries data that matches the specified regular
expression :
* | select content where regexp_lik e ( content , '\ d + m
') limit 100
```

```
 Parses   JSON - formatted   data   and   queries   data   that
 matches   the   specified   query   criterion :
* |  select   content   where   json_extra  ct ( content , '$. store
 . book ')=' mybook '  limit   100

 If   a   JSON - type   index   is   created ,  you   can   also
 specify   the   query   criterion   as   follows :
 field . store . book =' mybook '
```

- Extended query capability

  In log analysis scenarios, you may need to perform follow-up operations on queried data. For example:

  1. After finding an error log, check the context to find out the parameter that causes the error.
  2. After identifying an error, check for similar errors. For example, run the tail -f command to view the content of raw log files and run the grep command to query similar data.
  3. After obtaining millions of log entries from a query by keyword, filter out 90% of known issues that distract you.

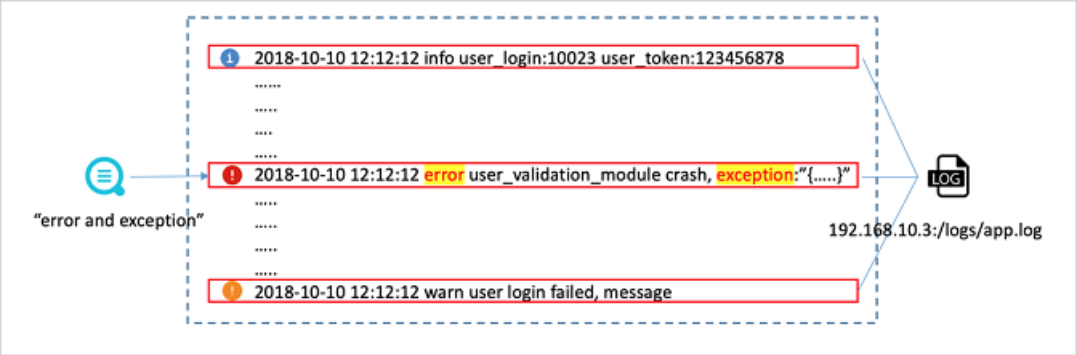
  To resolve the preceding issues, Log Service provides the following closed-loop solutions:

  - Context Lookup: allows you to look up the context of a log entry in raw log files by page, without the need to log on to the server.
  - LiveTail: runs the tail -f command in the cloud and allows you to view the content of raw log files in real time.
  - LogReduce: dynamically classifies logs based on their similar patterns to detect anomalies.

  1. LiveTail (tail -f command in the cloud)

     According to the traditional O&M method, you must run the `tail -f` command on log files on the server to monitor the log files in real time. If a large amount of log data is obtained in real time, you can run the `grep` or `grep -v` command to filter data by keyword. Log Service provides LiveTail in the

console for you to monitor and analyze online log data in real time. This makes O&M easier.



LiveTail has the following features:

- Supports various data sources, including Docker, Kubernetes, servers, and Log4j Appender.
- Monitors log data in real time, and allows you to specify keywords and filter data by keyword.
- Performs word-breaking for log fields to query the context of log entries that contain the specified field after word-breaking.

2. LogReduce

With the rapid development of business, you need more stable systems. Considering that each system generates a large number of logs every day, you may have the following concerns:

- Your system has potential anomalies, which, however, cannot be effectively detected in large amounts of log data.
- Intruders have logged on to your machine without permission, but you have no idea until it is too late.
- After a new version is published, the system behavior changes, but you are unaware of the changes. The root cause is that the system records too much information but fails to classify it. Logs that record the information have no schema and vary in format, and therefore cannot be well classified. Log

Service provides LogReduce to classify logs based on their similarity to help you quickly have a full view of log data. LogReduce has the following features:

- Supports logs in all formats, such as Log4j, JSON, and syslog.
- Filters logs by any criterion, uses the Reduce and Pattern functions to classify the filtered log data, and then queries the raw data based on a signature.
- Compares the patterns of different time periods.
- Dynamically adjusts the precision of the Reduce function.
- Obtains the query result from hundreds of millions of data entries in seconds.

Analysis capability

Elasticsearch provides the aggregation syntax based on Lucene DocValues and supports the SQL syntax in version 6.x to compute data by group and aggregation. Log Service supports the complete SQL-92 standard syntax in compliance with RESTful API and JDBC. In addition to the basic aggregate functions, Log Service also supports the complete SQL computing functions and provides self-developed functions, such as those for union query that joins internal and external data sources, machine learning, and pattern analysis.

Note:

The following figure shows the comparison results based on Elasticsearch 6.5 Aggregations and Log Service analysis syntax.

In addition to the SQL-92 standard syntax, Log Service also develops a series of useful functions based on the actual log analysis requirements.

1. Interval-valued comparison and periodicity-valued comparison functions

   You can use interval-valued comparison and periodicity-valued comparison functions in SQL statements to compare calculations of different time windows to gain insight into the growth trend. The result of each calculation can be a single value, multiple values, or a curve.

```
* | select compare ( pv , 86400 ) from ( select count ( 1
 ) as pv from log )
```

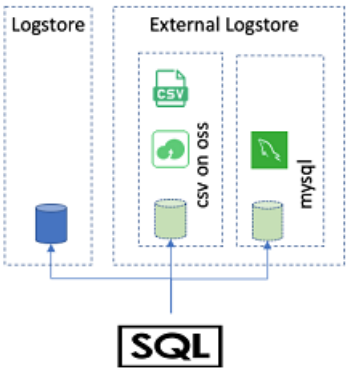
```
*| select t , diff [ 1 ] as current , diff [ 2 ] as
 yesterday , diff [ 3 ] as percentage from ( select t ,
 compare ( pv , 86400 ) as diff from ( select count ( 1
```

```
) as pv , date_forma t ( from_unixt ime ( __time__ ), '% H
:% i ') as t from log group by t ) group by t
order by t ) s
```

2. Union query that joins internal and external data sources

   You can join internal and external data sources in a union query, which has the following features:

   - Supports data sources such as Logstore, MySQL, and OSS (CSV files).
   - Supports LEFT JOIN, RIGHT JOIN, FULL JOIN, and INNER JOIN.
   - Uses SQL statements to query data in external tables and join internal and external tables.



   The following example shows how to join internal and external tables to query data.

```
SQL statements
Creates an external table :
* | create table user_meta ( userid bigint , nick
varchar , gender varchar , province varchar , gender
varchar , age bigint ) with ( endpoint =' oss - cn - hangzhou
. aliyuncs . com ', accessid =' LTA288 ', accesskey =' EjsowA ',
bucket =' testosscon nector ', objects = ARRAY [' user . csv '],
type =' oss ')

Uses the external table :
```

```
* | select u . gender , count ( 1 ) from chiji_acce sslog
 l join user_meta1 u on l . userid = u . userid group
  by u . gender
```

3. Geo location functions

   You can use geo location functions to analyze user sources based on IP addresses, mobile numbers, and geographic data. Geo location functions include:

   - ip: obtains the country, province, city, longitude and latitude, and carrier of an IP address.
   - mobile: obtains the carrier, province, and city of a mobile number.
   - geohash: converts geographic data to coordinates.

   The following example shows how to analyze the query results:

```
 SQL statements
* | SELECT count ( 1 ) as pv , ip_to_prov ince ( ip ) as
 province WHERE ip_to_doma in ( ip ) ! = ' intranet ' GROUP
 BY province ORDER BY pv desc limit 10

* | SELECT mobile_cit y ( try_cast (" mobile " as bigint
 )) as " city ", mobile_pro vince ( try_cast (" mobile " as
  bigint )) as " province ", count ( 1 ) as " number of
  requests " group by " province ", " city " order by "
 number of requests " desc limit 100
```

   - Geographic functions
   - Phone number functions
   - IP address functions

4. Security detection functions

   Based on the globally shared white hat security asset library, Log Service provides security detection functions. You can use security detection functions to check whether an IP address, domain name, or URL in logs is secure.

   - security_check_ip
   - security_check_domain
   - security_check_url

5. Machine learning and time series-based detection functions

   Log Service provides new machine learning and intelligent diagnostic functions to:

   - Automatically learn the rules of historical data and predict the future trend.
   - Detect imperceptible abnormal changes in real time, and analyze the characteristics that cause anomalies in combination with analysis functions.
   - Intelligently detect exceptions and inspect the system based on the period-on-period comparison and alerting features. This applies to intelligent O&M, security, and operations to gain insight into data in a faster, more effective, and more intelligent manner.

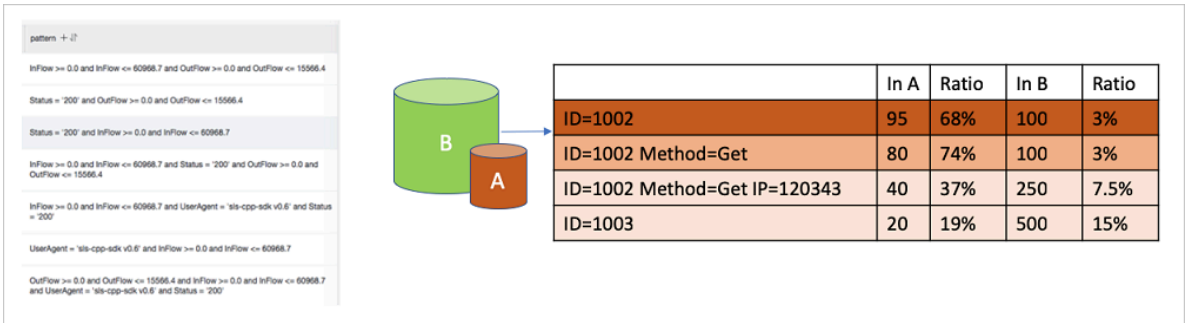
   These functions have the following features:

   - Prediction: fits the baseline based on historical data.
   - Anomaly detection, change point detection, and inflexion point detection: finds anomalies.
   - Time series forecasting: finds the periodic rules of data access.
   - Time series clustering: finds time series data with different curve shapes.

6. Pattern analysis functions

   You can use pattern analysis functions to detect the characteristics and rules in data to help you quickly and accurately identify problems. These functions have the following features:

   a. Finds the frequent pattern in statistical patterns. For example, 90% of invalid requests are sent by a user ID.
   b. Finds the pattern that causes differences between two collections under specified conditions. For example:
      - In requests with a latency greater than 10 seconds, the ratio of combined dimensions that contain an ID is much higher than that of other combined dimensions.
      - The ratio of this ID in collection B is lower than that in collection A.
      - Collections A and B are significantly different.



	In A	Ratio	In B	Ratio
ID=1002	95	68%	100	3%
ID=1002 Method=Get	80	74%	100	3%
ID=1002 Method=Get IP=120343	40	37%	250	7.5%
ID=1003	20	19%	500	15%

Performance

By using the same dataset, this section compares the performance of Log Service and the ELK Stack in terms of data write, data query, and aggregation.

- Test environment

  1. Test configuration

Item	On-premises ELK Stack	Log Service
Runtime environment	Elastic Compute Service (ECS) instance (4 vCPUs and 16 GB memory) × 4 + Ultra disks or SSDs	N/A
Shard	10	10

Item	On-premises ELK Stack	Log Service
Number of replicas	2	3 (configured by default and invisible to users)

2. Test data

   - Five columns of the double type, five columns of the long type, and five columns of the text type, with the values in alphabetical order based on the initial letter of field names as follows: 256, 512, 768, 1024, and 1280
   - Random fields
   - Size of raw data: 50 GB
   - Number of log lines: 162,640,232 (about 160 million log entries)

   The sample test log is as follows:

```
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
```

· Write test results

To write multiple data entries at a time, the ELK Stack uses a bulk API, whereas Log Service calls the PostLogstoreLogs operation. The following table lists the test results.

Item	Sub-item	On-premises ELK Stack	Log Service
Latency	Average write latency	40 ms	14 ms
Storage	Data volume of a replica	86 GB	58 GB
	Expansion rate: Data volume/Raw data size	172%	116%

 Note:

Log Service charges a total storage fee for 50 GB, where 23 GB is occupied by the written and compressed raw data and 27 GB is occupied by indexing.

Conclusion:

- Log Service has a shorter data write latency than the ELK Stack.
- The data stored in Log Service occupies a smaller space than that in the on-premises ELK Stack. The size of raw data is 50 GB. The storage space expands because the test data is random. In most live-network scenarios, the storage space after compression is smaller than the size of raw data. The data stored in the on-premises ELK Stack expands to 86 GB, with an expansion rate of 172%, which is 58% higher than that in Log Service. For the ELK Stack, this expansion rate is approximate to the recommended value, where the storage space is 2.2 times the size of raw data.

· Read (query and analysis) test

- Test scenarios

  This test uses two common scenarios as an example: log query and aggregation. It calculates the average latency in the two scenarios when the number of concurrent requests is 1, 5, and 10, respectively. The two scenarios are as follows:

  1. Uses a GROUP BY clause on any text column of full data, calculates the AVG, MIN, MAX, SUM, and COUNT values of five numeric columns, and then sorts the calculated values by the COUNT value to obtain the first 1,000 results:

     ```
     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. Queries a random keyword, for example, value_126, in logs of full data, and then obtains the first 100 lines of log entries that hit the keyword:

     ```
     value_126
     ```

- Test results

Scenario	Number of concurrent requests	Latency of the on-premises ELK Stack (Unit: seconds)	Latency of Log Service (Unit: seconds)
Analysis	1	3.76	3.4

Scenario	Number of concurrent requests	Latency of the on-premises ELK Stack (Unit: seconds)	Latency of Log Service (Unit: seconds)
	5	3.9	4.7
	10	6.6	7.2
Query	1	0.097	0.086
	5	0.171	0.083
	10	0.2	0.082

- Result analysis

  ■ According to the test results, both Log Service and the ELK Stack can query and analyze 150 million data entries within seconds.

  ■ In the analysis scenario, the latency is similar between Log Service and the ELK Stack. The ELK Stack uses SSD and delivers better I/O performance than Log Service when a large amount of data is read.

  ■ In the query scenario, Log Service has a much shorter latency than the ELK Stack. As the number of concurrent requests increases, the latency of the ELK Stack increases, whereas that of Log Service remains stable and even decreases.

Scale and cost

· Processing scale

1. Log Service can perform indexing for data in units of PB per day and query data among dozens of TB within seconds at a time. It supports auto scaling and scale-out for the processing scale.
2. The ELK Stack is suitable for scenarios where data is written in units of GB to TB per day and stored in units of TB. The processing scale is constrained by the following factors:

   - Single cluster scale: An ideal cluster consists of about 20 nodes. In the industry, a cluster can contain up to 100 nodes and is often split into multiple clusters to process business data in a convenient manner.
   - Write capacity: After shards are created, the write capacity cannot be modified. As the throughput increases, more shards are required to

dynamically scale up nodes. The maximum number of available nodes equals the number of shards.

- Storage capacity: When the primary shard reaches the maximum disk capacity, it must be migrated to another disk with larger capacity, or more shards must be allocated. The typical solution is to create an index, specify more shards, and rebuild existing data.

Example about the processing scale

Customer A is one of the major information websites in China. It has thousands of machines and hundreds of developers. The O&M team used to build a cluster based on the ELK Stack to process Nginx logs. However, the cluster always failed when processing a large scale of data. For example:

1. The cluster stops responding when processing a large amount of data for a user query. This makes the cluster unavailable to process data for other user queries.
2. During peak hours, the cluster is fully occupied and busy collecting and processing data. As a result, the cluster fails to guarantee data integrity and the accuracy of query results.
3. When the business grows to a certain scale, out of memory (OOM) often occurs due to memory settings and heartbeat synchronization. The cluster cannot guarantee its availability and accuracy. In this case, this cluster does not work properly and is abandoned by the development team.

In June 2018, the O&M team began to use Log Service, which provides the following features:

1. Uses Logtail to collect online logs, and uses API operations to integrate collection configuration and machine management into the customer's own O&M and control systems.
2. Embeds the Log Service query page into the unified logon and O&M platform to separate business permissions from account permissions.
3. Embeds Log Service console pages into the customer's own platform to enable the development team to query logs. Uses Grafana plug-ins to interconnect with Log Service and monitor business. Uses DataV to interconnect with Log Service and build a dashboard.

 Note:

For more information, see the following topics:

- Console sharing embedment
- Interconnection with Grafana
- Interconnect with DataV big screen
- OpenTracing implementation of Jaeger

After two months, the customer's O&M has been improved as follows:

1. The number of queries per day has increased significantly. The development team is gradually getting used to log query and analysis on the O&M platform, which improves their R&D efficiency. The O&M team also revokes online logon permissions.
2. In addition to Nginx logs, the O&M platform also imports app logs, mobile logs , and container logs. The processing scale is 10 times that before Log Service is used.
3. More applications are developed. For example, after Jaeger plug-ins are used to interconnect with Log Service, a tracing system is built based on logs to generate daily alerts and reports for online errors and accordingly carry out inspection.
4. Various platforms are interconnected with the unified O&M platform to collect data in a uniform manner and avoid repeated data collection. For example, the platforms of the big data department, such as Spark and Flink, can directly subscribe to and process real-time log data.

· Cost

Based on the preceding test data, this section calculates the average monthly cost in the scenario where 50 GB of data is written on a daily basis and stored for 90 days . The actual data size is 23 GB.

- The billing items of Log Service include read and write operations, indexing, and storage space. The query feature is free of charge. For more information about the billing standards, see Billing method.

Billing item	Quantity	Unit price	Cost (Unit: RMB)
Read and write operations	23 GB × 30 days	RMB 0.2/GB	138
Storage space (data stored for 90 days)	50 GB × 90 days	RMB 0.3/GB per month	1,350

Billing item	Quantity	Unit price	Cost (Unit: RMB)
Indexing	27 GB × 30 days	RMB 0.35/GB	283
Total	N/A	N/A	1,771

- The billing items of the ELK Stack include machines and data storage on SSD.
  - Generally, cloud disks provide high reliability. Therefore, the storage of replicas is not billed in this topic.
  - Generally, 15% of the available space must be reserved on a storage disk to prevent the space from being fully occupied. Therefore, a coefficient of 1.15 is multiplied.

Billing item	Quantity	Unit price	Cost (Unit: RMB)
Server	Server with 4 vCPUs and 16 GB memory × 4 × Three months (ecs.mn4.xlarge)	Subscription: RMB 675/month	2,021
Storage	86 GB × 1.15 × 90 days (only one replica is calculated)	SSD: RMB 1/GB per month	8,901
	N/A	SATA: RMB 0.35/GB per month	3,115
Total			SSD: 12,943
			SATA: 5,135

With the same performance delivered, the cost ratio of Log Service to the ELK Stack (SSD) is 13.6%. SSD can be replaced with SATA to lower the cost of the on-premises ELK Stack. In this case, the cost ratio of Log Service to the ELK Stack (SATA) is 34%. However, the latency increases from 40 ms (SSD) to 150 ms (SATA). After data is read and written for a long time, the query and read/write latency greatly increases and the on-premises ELK Stack cannot work properly.

- Time to value

  In addition to hardware costs, Log Service is basically free of charge for new data import, new services, maintenance, and resource resizing. It also provides the following features:

  - Seamless interconnection with various log processing systems in the ecosystem, such as Spark, Hadoop, Flink, and Grafana
  - Global deployment in more than 20 regions, which helps you expand business on a global scale
  - More than 30 SDKs that can seamlessly interconnect and integrate Log Service with other Alibaba Cloud products

Summary

Elasticsearch supports more common scenarios such as update, query, and delete. It is widely used in fields such as search, data analysis, and application developmen t. The ELK Stack maximizes the flexibility and performance of Elasticsearch in log analysis scenarios. Log Service is designed for log data analysis scenarios and has customized and developed many applications in this field. The former has a wide service range, whereas the latter is more targeted. Of course, the comparison of data is meaningless if scenarios are not considered. Both of them are useful in suitable scenarios.

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 investigation, 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_26	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
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

Function	ELK system	Hadoop + Hive	Log Service
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

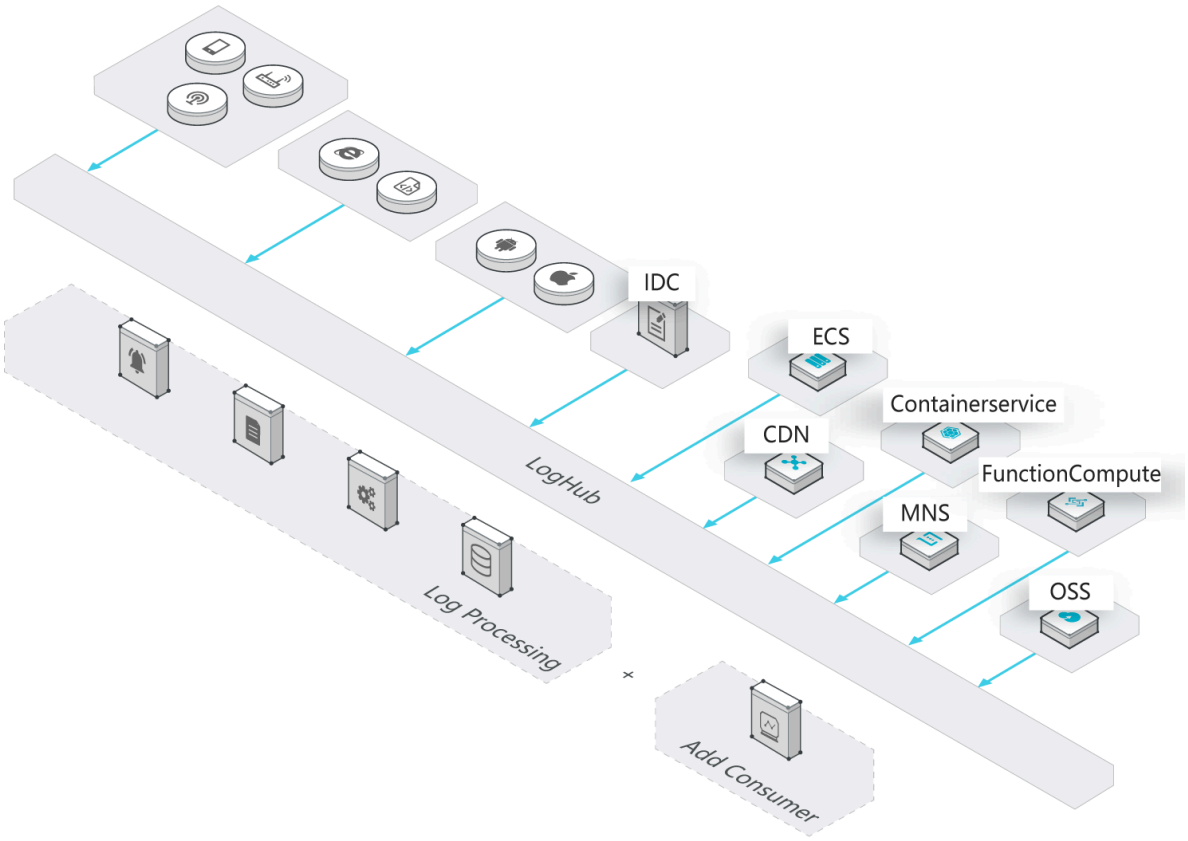
The LogHub function of Log Service enables access to massive real-time log data (including Metric, Event, BinLog, TextLog, and Click data) at the lower costs.

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



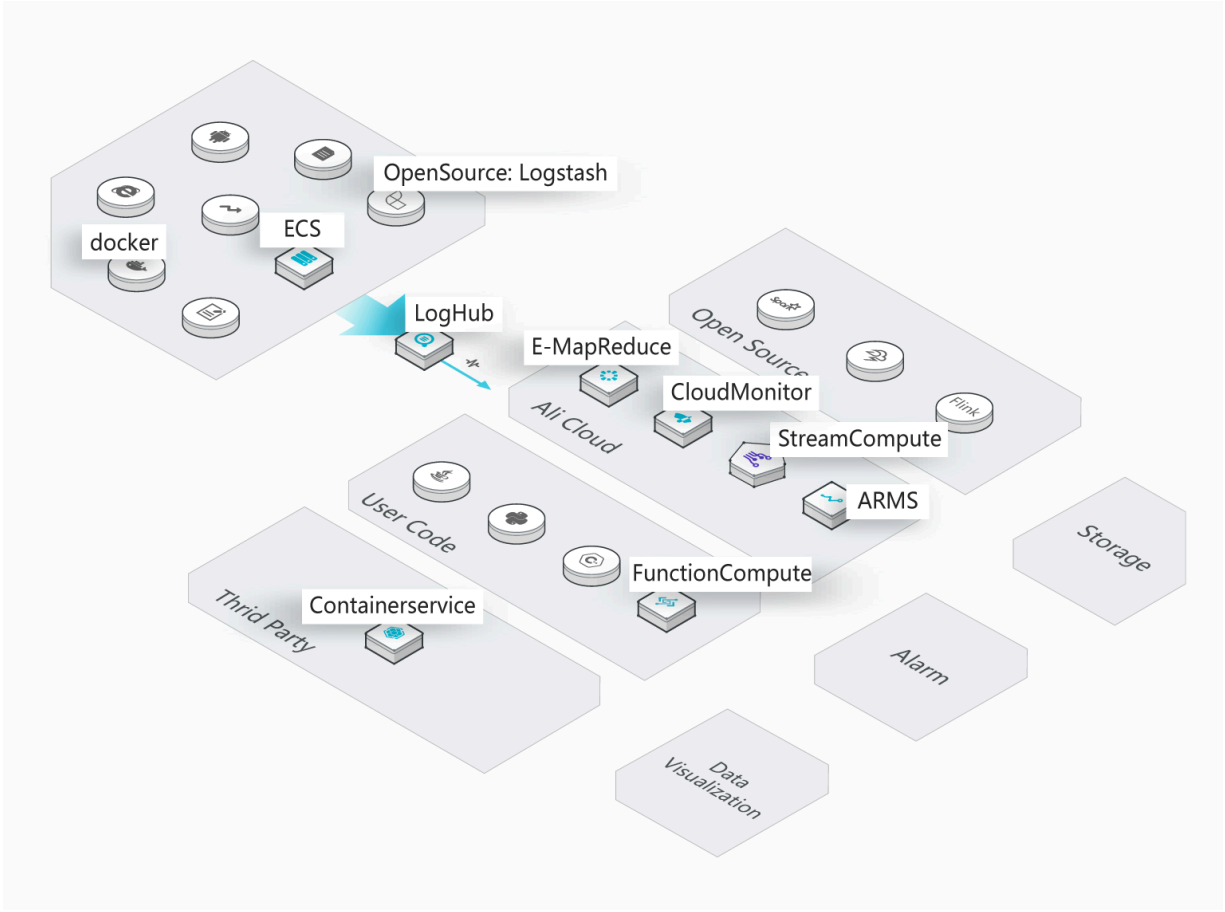
ETL/Stream Processing

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



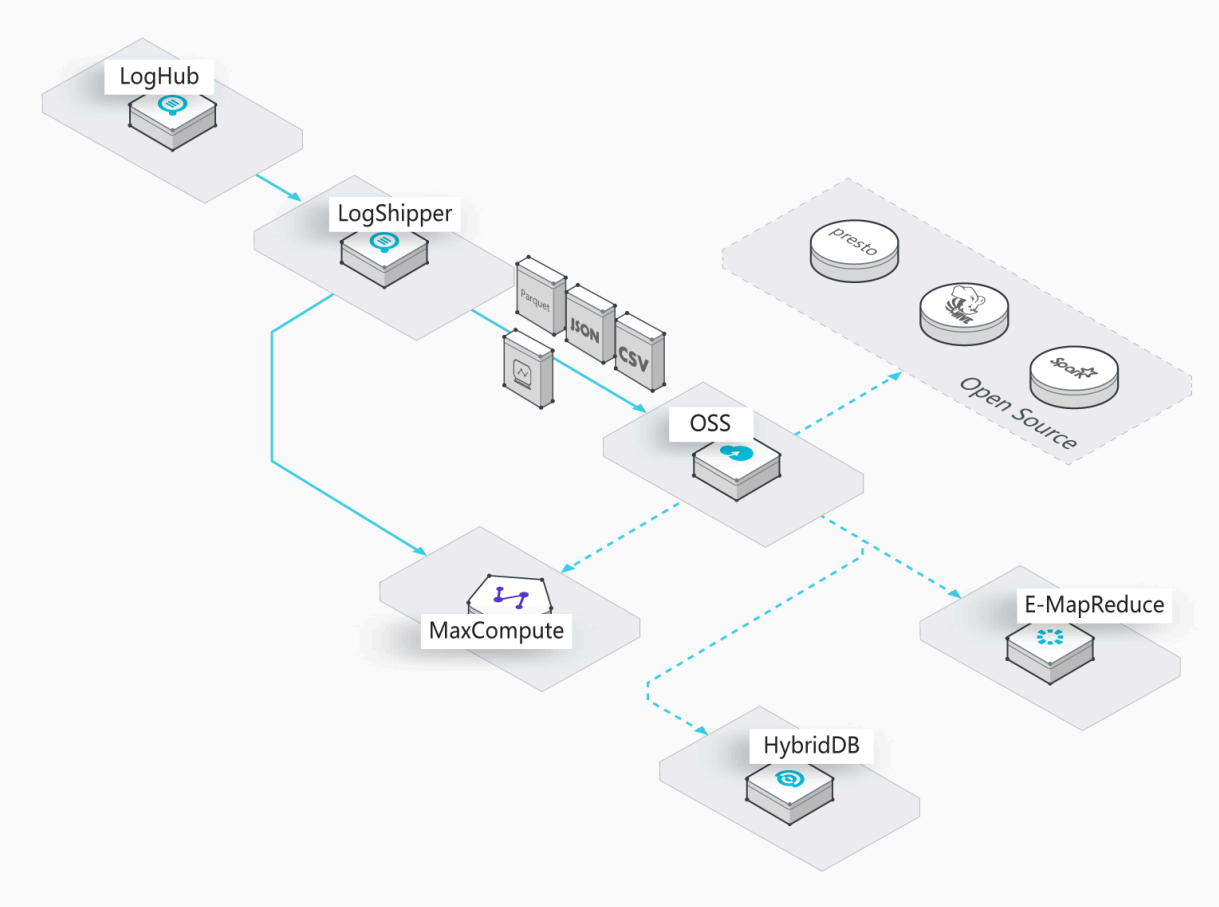
Data warehouse

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



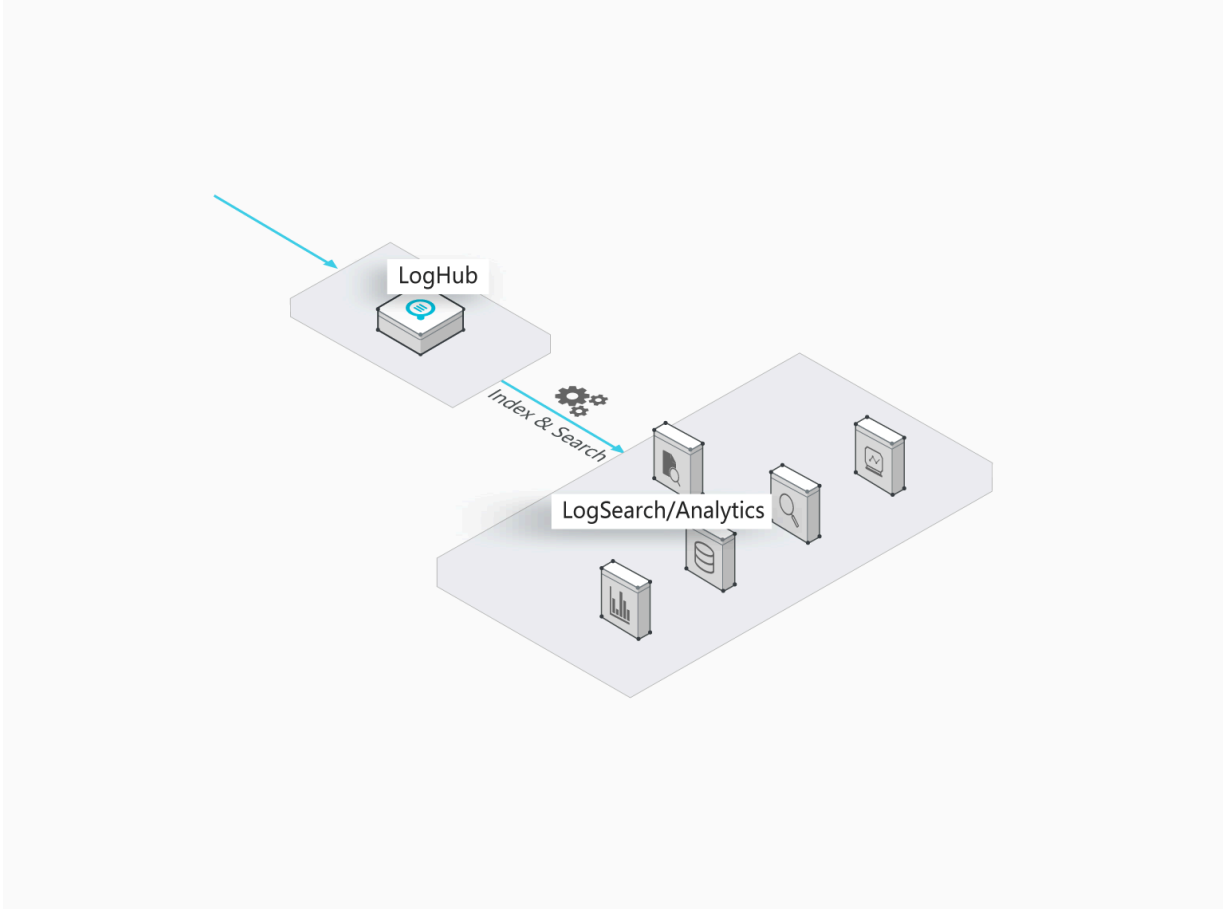
Real-time query and analysis of logs

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 applicatio n 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

A log is an abstraction of changes that happen in a system. A log is a sequence of records ordered by time, and contains information about operations and results of specific objects. Log files, events, binary logs, and metrics are different types of log carriers. A 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 uses a semi-structured data model to define a log. This model includes these data fields: Topic, Time, Content, Source, and Tags.

Log Service has different format requirements for different data fields, as described in the following table.

Data field	Description	Format
Topic	This is a user-defined field used to mark multiple logs. For example, access logs can be marked based on sites.	Any string of up to 128 bytes in length, including a null string. By default, this field is a null string.
Time	This is a reserved field in a log and is used to indicate the time when a log is generated. In most cases, it is generated directly based on the time in a log.	An integer in the UNIX timestamp format. The unit is in seconds. This field indicates the number of seconds that have elapsed since 1970-1-1 00:00:00 UTC.
Content	This field is used to record the specific content of a log. The content consists of one or more content items, and each content item is a key-value pair.	A key is a UTF-8 encoded string of up to 128 bytes in length. It can contain letters, underscores, and digits. It cannot start with a digit or use any of the following keywords: · `__time__` · `__source__` · `__topic__` · `__partitio n_time__` · `_extract_o thers_` · `__extract_ others__` The value can be any string of up to 1,024 bytes.

Data field	Description	Format
Source	This field indicates the source of a log. For example, the IP address of the server where a log is generated.	Any string of up to 128 bytes in length. This field is null by default.
Tags	Log tags include: · User-defined tags: the tags that you add when you use the PutLogs API operation to write data. · System tags: the tags added by Log Service, including `__client_i p__` and `__receive_ time__` .	Dictionary format. Both keys and values are strings. When you query logs in the console, the system displays the tags with the `__tag__ :` prefix.

Logs are used in various formats in actual scenarios. The following example shows you how to map an original NGINX access log to the log data model of Log Service. For example, the IP address of your NGINX server is `10 . 249 . 201 . 117` . An original log of this server is as follows:

```
10 . 1 . 168 . 193  - - [ 01 / Mar / 2012 : 16 : 12 : 07  + 0800 ] "
GET  / Send ?  AccessKeyI  d = 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 "
```

The following example shows how to map the original log to the log data model of Log Service.

Data field	Content	Description
Topic	""	The default null string is used.
Time	1330589527	The exact timestamp when the log is generated . This timestamp is the number of seconds that have elapsed since 1970 -1-1 00:00:00 UTC. The time is converted from the timestamp of the original log.

Data field	Content	Description
Content	Key-value pair	The specific content of a log.
Source	"10.249.201.117"	The IP address of the server is used as the log source.
Tags	None	You or Log Service add the tags.

You can decide how to extract the original content of a log and combine the extracted content into key-value pairs, as shown in the following table.

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"

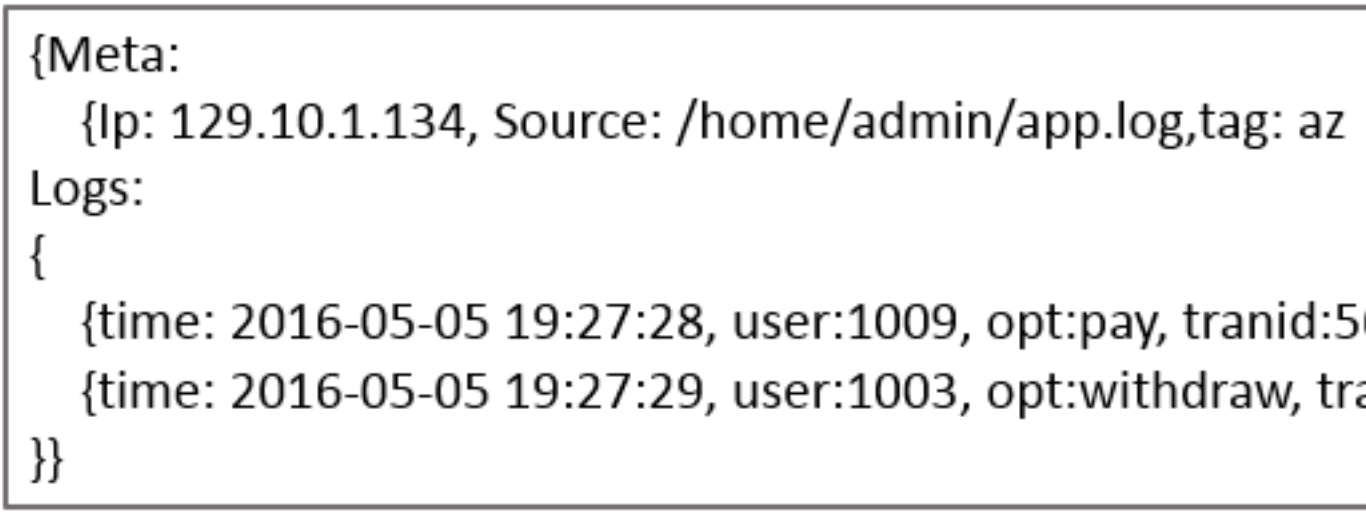
5.3 Log group

A log group is a collection of logs. It is the basic unit for writing and reading data through the API or SDK. Logs in a log group have the same metadata such as the IP address and source.

When data is written through the Log Service API or SDK, multiple logs are packaged into a log group and sent to Log Service. Compared with reading and writing data by log, reading and writing data by log group can minimize the number of read and write operations and improve business efficiency.

A log group has a maximum size of 5 MB.

Figure 5-1: Log group



5.4 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.

5.5 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

5.6 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

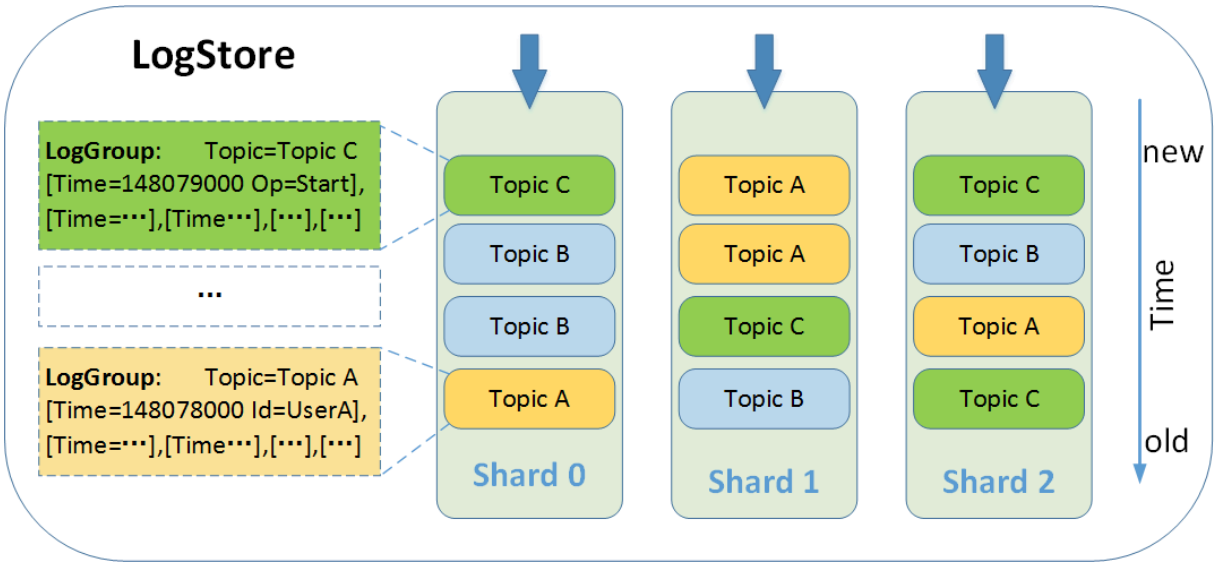
finds the adjacent shard at the right of the specified shard and merges these two shards. After the merge, the specified shard and the adjacent shard on the right are in readonly status. Data can still be consumed, while new data cannot be written. A new shard in readwrite status is generated and its MD5 range covers the total range of the original two shards.

5.7 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

Resource	Limit	Remarks
Project	You can create a maximum of 50 projects for an Alibaba Cloud account.	If you request more quotas, open a ticket.
Logstore	You can create a maximum of 200 Logstores for a project.	If you request more quotas, open a ticket.
Shard	· You can create a maximum of 200 shards for a project. · When creating a Logstore in the console, you can create a maximum of 10 shards in the Logstore. When creating a Logstore by using the API, you can create a maximum of 100 shards in the Logstore. However, you can split shards to increase the number of shards no matter how you create a Logstore.	If you request more quotas, open a ticket.
Logtail configuration	You can create a maximum of 100 Logtail configurations for a project.	If you request more quotas, open a ticket.
Log storage time	You can store logs permanently. You can also customize the log storage time in the range of 1-3,000 days.	N/A

Resource	Limit	Remarks
Machine group	You can create a maximum of 100 machine groups for a project.	If you request more quotas, open a ticket.
Consumer group	You can create a maximum of 20 consumer groups for a Logstore.	You can delete consumer groups that are no longer used.
Saved search	You can create a maximum of 100 saved search items for a project.	N/A
Dashboard	· You can create a maximum of 50 dashboards for a project. · Each dashboard can contain a maximum of 50 charts.	N/A
Log item	The maximum length is 1 MB.	This limit applies if you call API operations to collect logs. If you configure Logtail to collect logs, the maximum length for a single log item is 512 KB.
Log item (Key)	The maximum length is 128 bytes.	N/A
Log item (Value)	The maximum length is 1 MB.	N/A
Log group	The maximum length of a log group is 5 MB.	N/A
Alert	You can create a maximum of 100 alerts for a project.	If you request more quotas, open a ticket.

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.
	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.

Resource	Limit	Description	Note
	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...".

Function	Item	Limit	Note
	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.	If a log contains more than 10,000 characters, the system will display a corresponding prompt.
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	· By default, 100 log entries are displayed after an SQL statement is executed. · The returned result of each analysis can be log entries of 100 MB or 100, 000 log entries at most.	You can use the limit parameter to specify the number of returned log entries. You can turn over the page to view the returned results.
	Field size	The size of a field used in a statement can be up to 2 KB.	The system truncates a field larger than 2 KB.
	Double precision	A double type can be shown by using up to 52 digits. If you use more than 52 digits to show a double type, the precision will be affected.	For example, 0.1+ 0.2 is shown as 0. 30000000000000004 . We recommend that you use round (0.1+0.2,2) to take two significant values.

6.4 Reserved fields

In Log Service, some fields are reserved. When you add Logtail configurations or call API operations to write log data, do not set the names of fields to be the same as those of reserved fields.

Important notes

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

- When you add Logtail configurations or call API operations to write log data, do not set the names of fields (that is, keys) to be the same as those of reserved fields. Otherwise, duplicate field names may cause problems such as inaccurate queries.
- Log Service does not deliver any fields prefixed with `__tag__` .

- Log Service charges you in Billing method mode for the new fields that you specify to be recorded in logs. If you enable the indexing feature for the fields, you are also charged a small fee for the indexing and storage traffic.

Reserved fields

The following table describes the reserved fields in Log Service.

Table 6-1: Reserved fields in Log Service

Field	Type	Index and statistics settings	Description
`__time__`	Integer, in UNIX timestamp format. For example, `__time__:1523868463`.	• Index settings: You do not need to create an index on the `__time__` field because this field can be set through the from and to parameters in API operations. • Statistics settings: Statistics for the `__time__` field are enabled after you enable the statistics feature for any column.	The log time that you specify when you use the API or SDK to write log data. This field can be used for log delivery, query, and analysis.

Field	Type	Index and statistics settings	Description
`__source__`	String.	· Index settings: After the indexing feature is enabled for the `__source__` field, Log Service creates an index on this field by default. The index is of the text type, and no delimiter is specified. To query logs based on the index on this field, enter `source : 127 . 0 . 0 . 1` or `__source__ : 127 . 0 . 0 . 1`. · Statistics settings: Statistics for the `__source__` field are enabled after you enable the statistics feature for any column.	The device from which logs are collected. This field can be used for log delivery, query, analysis, and custom consumption.
`__topic__`	String.	· Index settings: After the indexing feature is enabled for the `__topic__` field, Log Service creates an index on this field by default. The index is of the text type, and no delimiter is specified. To query logs based on the index on this field, enter `__topic__ : XXX`. · Statistics settings: Statistics for the `__topic__` field are enabled after you enable the statistics feature for any column.	The topic of logs. If you have set a log topic, Log Service automatically adds the topic field to your logs with the key set to `__topic__` and the value set to the topic content that you specified. This field can be used for log delivery, query, analysis, and custom consumption.

Field	Type	Index and statistics settings	Description
`_extract_others_`	String, which can be deserialized into a JSON map.	You do not need to create an index on this field because this field does not exist in any logs.	This field is the same as the `__extract_others__` field. We recommend that you use the `__extract_others__` field.
`__tag__:__client_ip__`	String.	· Index settings: After the indexing feature is enabled for this field, Log Service creates indexes on all the tag fields by default. The index is of the text type, and no delimiter is specified. When you query logs based on the index on a tag field, both exact match and fuzzy match are supported. · Statistics settings: The statistics feature is disabled for the column indicated by this field. To enable statistics for this field, create an index on the `__tag__:__client_ip__` field and enable the statistics feature.	The public IP address of the device from which logs are collected. This field is a system tag. After the Log Public IP feature is enabled, the server adds this field to each raw log received. This field can be used for log query, analysis, and custom consumption. When conducting SQL analysis on this field, you must enclose this field in double quotation marks (" ").

Field	Type	Index and statistics settings	Description
`__tag__ : __receive_ time__`	String, which can be converted to an integer in UNIX timestamp format.	· Index settings: After the indexing feature is enabled for this field, Log Service creates indexes on all the tag fields by default. The index is of the text type, and no delimiter is specified. When you query logs based on the index on a tag field, both exact match and fuzzy match are supported. · Statistics settings: The statistics feature is disabled for the column indicated by this field. To enable statistics for this field, create an index on the `__tag__ : __receive_ time__` field and enable the statistics feature.	The time when the server receives a log. This field is a system tag. After the Log Public IP feature is enabled, the server adds this field to each raw log received. This field can be used for log query, analysis, and custom consumption.

Field	Type	Index and statistics settings	Description
`__tag__ : __path__`	String.	· Index settings: After the indexing feature is enabled for the `__tag__ : __path__` field, Log Service creates an index on this field by default. The index is of the text type, and no delimiter is specified. To query logs based on the index on this field, enter `__tag__ : __path__ : XXX` . · Statistics settings: The statistics feature is disabled for the column indicated by this field. To enable statistics for this field, create an index on the `__tag__ : __path__` field and enable the statistics feature.	The path to log files collected by Logtail. Logtail automatically adds this field to logs. This field can be used for log query, analysis, and custom consumption. When conducting SQL analysis on this field, you must enclose this field in double quotation marks (" ").

Field	Type	Index and statistics settings	Description
`__tag__:__hostname__`	String.	· Index settings: After the indexing feature is enabled for the `__tag__:__hostname__` field, Log Service creates an index on this field by default. The index is of the text type, and no delimiter is specified. To query logs based on the index on this field, enter `__tag__:__hostname__:XXX`. · Statistics settings: The statistics feature is disabled for the column indicated by this field. To enable statistics for this field, create an index on the `__tag__:__hostname__` field and enable the statistics feature.	The hostname of the device from which Logtail collects data. Logtail automatically adds this field to logs. This field can be used for log query, analysis, and custom consumption. When conducting SQL analysis on this field, you must enclose this field in double quotation marks (" ").
`__raw_log_`	String.	You need to create and configure an index of the text type on this field and enable the statistics feature as needed.	The raw logs that fail to be parsed. After the Drop Failed to Parse Logs feature is disabled, Logtail uploads raw logs if log parsing fails. In this field, the key is `__raw_log_` and the value is the log content. This field can be used for log delivery, query, analysis, and custom consumption.

Field	Type	Index and statistics settings	Description
`__raw__`	String.	You need to create and configure an index of the text type on this field and enable the statistics feature as needed.	The raw logs that are parsed. After the Upload Raw Log feature is enabled, Logtail uploads the raw logs in the `__raw__` field together with the parsed logs. This field is used in log audit and compliance check scenarios. This field can be used for log delivery, query, analysis, and custom consumption.