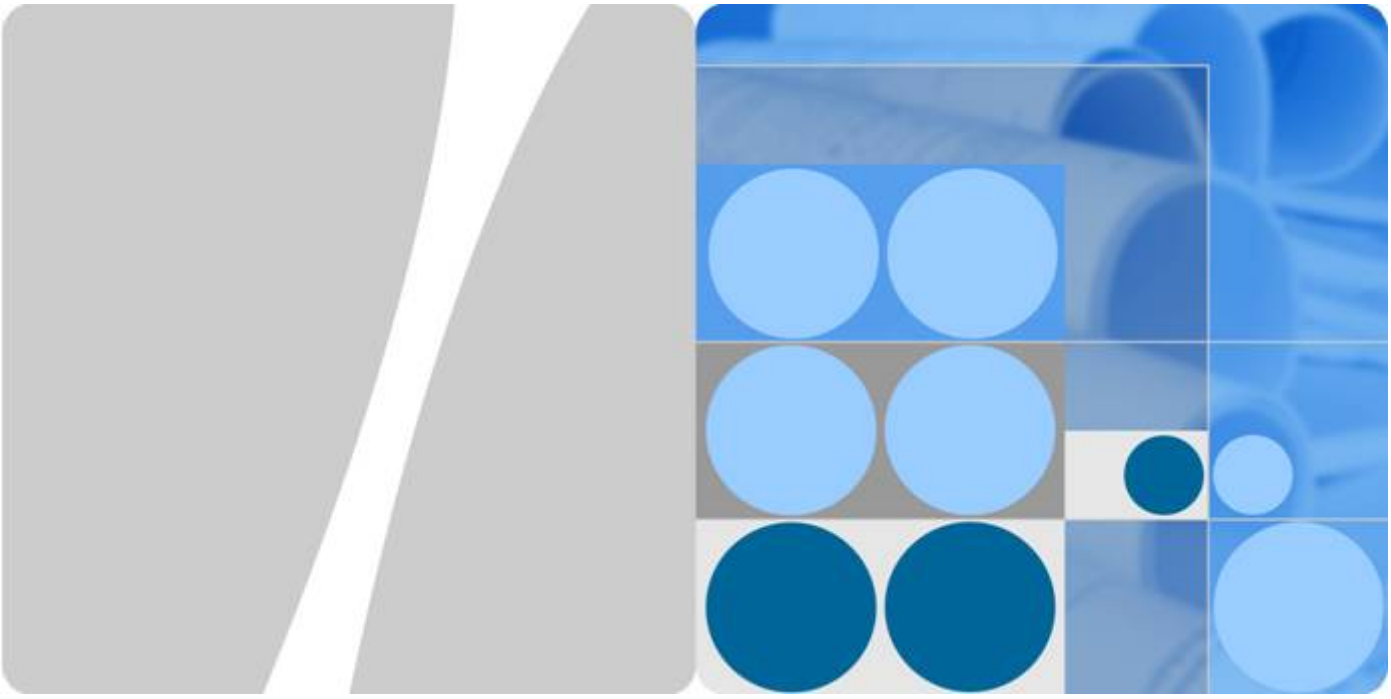


MapReduce Service

Product Introduction

Date **2019-10-10**

Contents

1 What Is MRS?

Big data is a huge challenge facing the Internet era as the data volume and types increase rapidly. Conventional data processing technologies, such as single-node storage and relational databases, are unable to solve the emerging big data problems. In this case, the Apache Software Foundation (ASF) has launched an open source Hadoop big data processing solution. Hadoop is an open source distributed computing platform that can fully utilize computing and storage capabilities of clusters to process massive amounts of data. If enterprises deploy Hadoop systems by themselves, the disadvantages include high costs, long deployment period, difficult maintenance, and inflexible use.

To solve the preceding problems, HUAWEI CLOUD provides MapReduce Service (MRS) for managing the Hadoop system. With MRS, you can deploy a Hadoop cluster in just one click. MRS provides enterprise-level big data clusters on the cloud. Tenants can fully control clusters and easily run big data components such as Hadoop, Spark, HBase, Kafka, and Storm. In addition, MRS can be customized and developed based on service requirements.

2 Advantages of MRS

MRS has a powerful Hadoop kernel team and is deployed based on Huawei's enterprise-level FusionInsight big data platform. MRS has been deployed on tens of thousands of nodes and can ensure Service Level Agreements (SLAs) for multi-level users. MRS has the following advantages:

- High performance

  MRS supports self-developed CarbonData storage technology. CarbonData is a high-performance big data storage solution. It allows one data set to apply to multiple scenarios and supports features, such as multi-level indexing, dictionary encoding, pre-aggregation, dynamic partitioning, and quasi-real-time data query. This improves I/O scanning and computing performance and returns analysis results of tens of billions of data records in seconds. In addition, MRS supports self-developed enhanced scheduler Superior, which breaks the scale bottleneck of a single cluster and is capable of scheduling over 10,000 nodes in a cluster.

- Low cost

  Based on diversified cloud infrastructure, MRS provides various computing and storage choices and separates computing from storage, delivering low-cost massive data storage solutions. MRS supports auto scaling to address peak and off-peak service loads, releasing idle resources on the big data platform for customers. MRS clusters can be created and scaled out when you need them, and can be terminated or scaled in after you use them, minimizing cost.

- High security

  With Kerberos authentication, MRS provides role-based access control (RBAC) and sound audit functions. MRS is a one-stop big data platform that allows different physical isolation modes to be set up for customers in the public resource area and dedicated resource area of HUAWEI CLOUD as well as HCS Online in the customer's equipment room. A cluster supports multiple logical tenants. Permission isolation enables the computing, storage, and table resources of the cluster to be divided based on tenants.

- Easy O&M

  MRS provides a visualized big data cluster management platform, improving O&M efficiency. MRS supports rolling patch upgrade and provides visualized patch release information and one-click patch installation without manual intervention, ensuring long-term stability of user clusters.

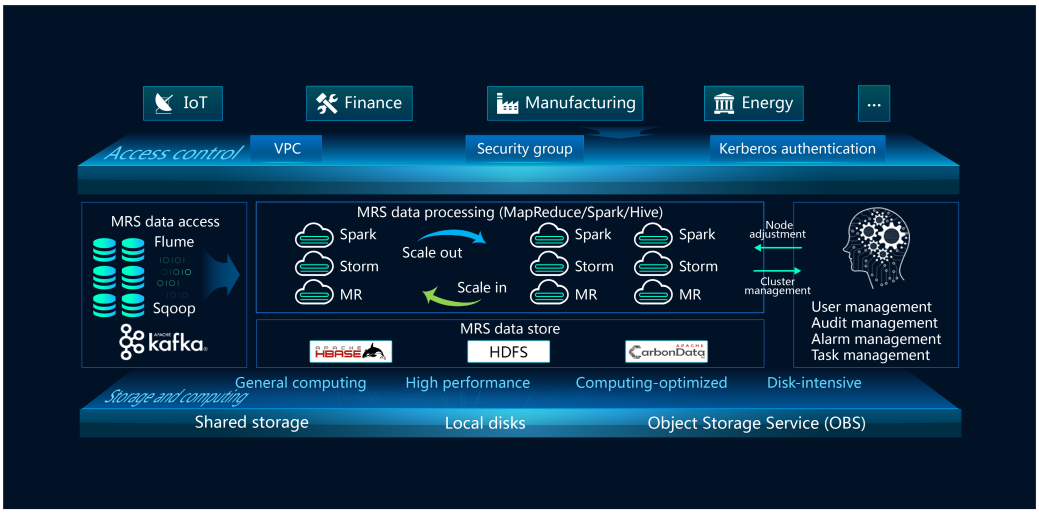
- High reliability

MRS delivers high availability (HA) and real-time SMS and email notification on all nodes.

3 MRS Architecture

Figure 3-1 shows the MRS architecture.

Figure 3-1 MRS architecture



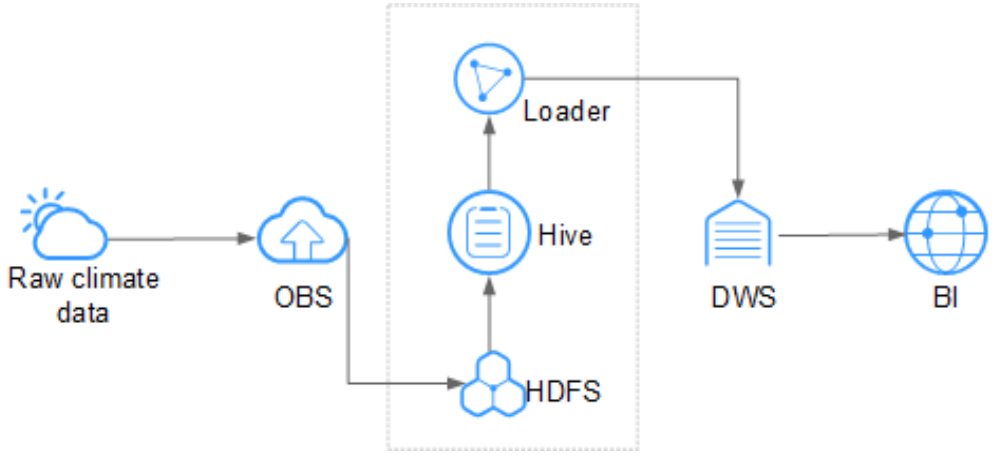
4 Application Scenarios

Big data is ubiquitous in people's lives. HUAWEI CLOUD MRS is suitable to process big data in the industries such as the Internet of things (IoT), e-commerce, finance, manufacturing, healthcare, energy, and government departments.

- Large-scale data analysis

  In the environmental protection industry, climate data is stored on OBS and periodically dumped into HDFS for batch analysis. 10 TB of climate data can be analyzed in 1 hour.

  Figure 4-1 Large-scale data analysis in the environmental protection industry

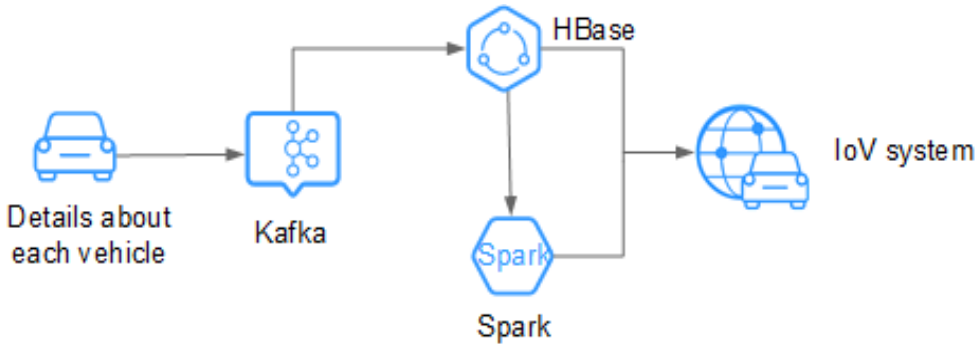
  MRS has the following advantages in this scenario.

  - Low cost: OBS offers cost-effective storage.
  - Massive data analysis: TB/PB-level data is analyzed by Hive.
  - Visualized data import and export tool: Loader exports data to Data Warehouse Service (DWS) for business intelligence (BI) analysis.

- Large-scale data storage

  For example, in the Internet of vehicles (IoV) industry, an automobile company stores data on HBase, which supports PB-level storage and CDR queries in milliseconds.

Figure 4-2 Large-scale data storage in the IoV industry



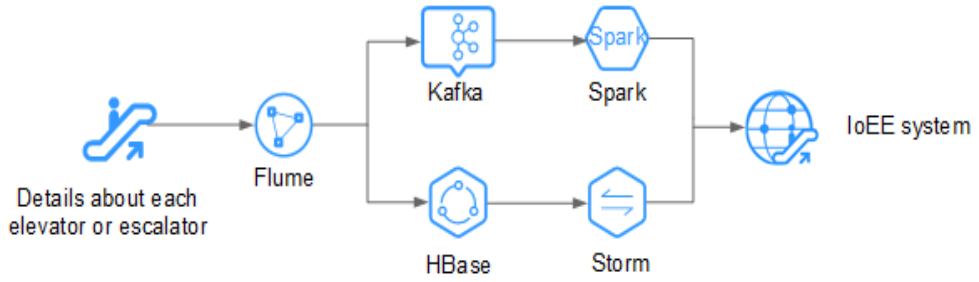
MRS has the following advantages in this scenario.

- Real time: Kafka accesses massive numbers of vehicle messages in real time.
- Massive data storage: HBase stores massive volumes of data and supports data queries in milliseconds.
- Distributed data query: Spark analyzes and queries massive volumes of data.

- Low-latency streaming processing

  For example, in the Internet of elevators & escalators (IoEE) industry, data of smart elevators and escalators is imported to MRS streaming clusters in real time for real-time alarming.

Figure 4-3 Low-latency streaming processing in the IoEE industry



MRS has the following advantages in this scenario.

- Real-time data ingestion: Flume implements real-time data ingestion and provides various data collection and storage access methods.
- Data source access: Kafka accesses data of tens of thousands of elevators and escalators in real time.

5 List of MRS Component Versions

Table 5-1 lists component versions of MRS clusters of each version.

Table 5-1 MRS component versions

Components Supported by MRS	MRS 1.8.7	MRS 2.0.1	MRS 2.0.3
Presto	0.215	308	308
Hadoop	2.8.3	3.1.1	3.1.1
Spark	2.2.1	2.3.2	2.3.2
HBase	1.3.1	2.1.1	2.1.1
OpenTSDB	2.3.0	-	-
Hive	1.2.1	3.1.0	3.1.0
Tez	-	0.9.1	0.9.1
Hue	3.11.0	3.11.0	3.11.0
Loader	2.0.0	2.0.0	2.0.0
Flink	1.7.0	-	-
Kafka	1.1.0	1.1.0	1.1.0
KafkaManager	1.3.3.1	-	-
Storm	1.2.1	1.2.1	1.2.1
Flume	1.6.0	1.6.0	1.6.0

6 MRS Functions

6.1 New Functions of MRS 2.0

In MRS 2.0, the Hadoop, Hive, Spark, and HBase components are upgraded and Tez is supported. **Table 6-1** provides details about the new functions.

Table 6-1 New functions of MRS 2.0

Component Version	New Function	Description
Hadoop 3.1.1	Erasure coding (EC)	It is a data persistency method that saves storage space. The storage space of cold data can be reduced. As a new function added to HDFS, the erasure coding is a data persistency method that saves more storage space than replicas. For example, the Reed-Solomon(10,4) standard encoding technology requires only 1.4 times of the space overhead, while the standard HDFS replica technology requires 3 times of the space overhead.
	DataNode multi-disk balancer	This function is used to solve the problem of unbalanced data storage among multiple disks in the DataNode when disks are added or replaced.
	Opportunistic containers	This function improves the cluster resource usage and increases task throughput.
	User-defined resource model	User-defined resource models are supported. In MRS 2.0 or later, Yarn supports user-defined countable resource types. In addition to CPUs and memory, cluster administrators can customize resources such as GPUs and software licenses. Yarn tasks can be scheduled based on the availability of these resources.

Component Version	New Function	Description
Hive 3.1.0	Hive web UI	Hive web UI makes O&M easier. HiveServer provides a web UI for O&M personnel to view the running SQL statements and how long an SQL statement has been executed.
	HPL/SQL	Hive provides HPL/SQL to implement procedural SQL for Hive. This makes data migration from conventional data warehouses and relational databases such as Oracle to Hive more convenient.
	Cost-based optimizer (CBO)	CBO performs optimization based on costs. It selects the syntax tree of the minimum cost from multiple possible syntax trees for execution. The core of CBO is to evaluate the actual cost of a given syntax tree, significantly optimizing multi-table JOIN performance.
Tez 0.9.1	New execution engine: Tez	Tez is a distributed computing framework that supports directed acyclic graphs. As the default execution engine of Hive, Tez remarkably surpasses the original MapReduce computing engine in terms of execution efficiency.
Spark 2.3.2	Continuous Processing	The delay of microbatch processing is reduced from 100 ms to 3 ms. Both the early Spark Streaming and Structured Streaming launched in Spark 2.0 use the scheduled triggering mode to generate microbatches for streaming processing. The microbatch processing has the minimum delay limit (about 100 ms). The Continuous Processing mode newly added to Structured Streaming can implement millisecond-level low-latency processing (about 3 to 5 milliseconds).
	Stream-Stream join	Support for stream-stream joins Structured Streaming is used to replace the original Spark Streaming. In earlier versions, Structured Streaming supports only joins between streams and static data sets. Spark 2.3 provides stream-stream joins and supports internal and external connections, which can be used in a large number of real-time scenarios, for example, in a common join of click log streams.

Component Version	New Function	Description
	PySpark performance optimization	Processing duration is reduced by 60% to 90%. The **pandas_udf** is implemented based on Apache Arrow and Pandas. Pandas is used for vectorizing data and Apache Arrow is used to reduce the overhead of communications between Python and Spark. The **pandas_udf** replaces the original user-defined functions (UDFs) in PySpark to process data, which reduces the processing duration by 60% to 90% (affected by specific operations).
	MLlib optimization	It makes secondary development more convenient. In Spark 2.3, many improvements have been made in MLlib. For example, MLlib models and pipelines can be used in Structured Streaming. DataFrame of image data can be created. APIs for using Python to compile custom machine learning algorithms have been simplified.
HBase 2.1.1	AssignmentManager V2	Region state transition is optimized. AssignmentManager V2 is implemented based on Procedure V2 and can quickly allocate regions. The maintained region state machine storage does not depend on ZooKeeper. The region state information in ZooKeeper is removed. Region states are maintained only in HMaster memory and a Meta table. This solves the problems that occur during the region state transition.
	In-Memory Compaction	The implementation of HFile is optimized. After data in a MemStore reaches a certain size, the data is flushed to an immutable segment in the memory. Multiple segments in the memory can be merged in advance and flushed to HFiles in HDFS after the segments reach a certain size. This effectively reduces the write I/O amplification caused by Memory Compaction.
	Offheaping of Read/Write	The read/write performance is optimized. HBase 2.x changes the data read/write mode and directly reads and writes data in the "L2" second caching tier. Offheap memory is used to replace the previous heap memory to reduce the usage of the heap memory, reducing the GC pressure.

Component Version	New Function	Description
	NettyBaseRpc framework	The throughput is improved and the delay is reduced. By default, HBase 2.x uses NettyRpcServer to replace the native RPC server of HBase, greatly improving the throughput of HBaseRPC and reducing the delay.
	RegionServer Group	Physical isolation of multiple tenants is supported. As a new multi-tenant solution, RegionServer Group can group multiple RegionServers to form different RGSs. Different tables can be distributed in various RGSs, and tables in different RGSs do not affect each other. In this way, a multi-tenant solution is implemented by physically isolating the tables in the RegionSevers.
	HBase on OBS	OBS can be interconnected. Data can be decoupled from MRS clusters. HBase 2.x in MRS 2.0 can be interconnected with OBS and store the final data to OBS. It applies to scenarios where a large amount of data needs to be archived and stored. Data can be decoupled from MRS clusters and a flexible switch is allowed.

6.2 Hadoop

MRS deploys and hosts Apache Hadoop clusters in the cloud to provide services featuring high availability and enhanced reliability for big data processing and analysis. MRS uses the FusionInsight Hadoop commercial release. Hadoop is a distributed system architecture that consists of HDFS, MapReduce, and Yarn. The following describes the functions of each component:

- HDFS: HDFS provides high-throughput data access and is applicable to the processing of large data sets. MRS cluster data is stored in HDFS.
- MapReduce: As a programming model that simplifies parallel computing, MapReduce gets its name from two key operations: Map and Reduce. Map divides one task into multiple tasks, and Reduce summarizes the processing results of these tasks and produces the final analysis result. MRS clusters allow you to submit self-developed MapReduce programs, execute the programs, and obtain the result.
- Yarn: As the resource management system of Hadoop, Yarn manages and schedules resources for applications. MRS uses Yarn to schedule and manage cluster resources.

For details about Hadoop architecture and principles, see **https://hadoop.apache.org/docs/stable/index.html**.

For details about how to visit the websites of Hadoop components, see **Websites of Open Source Components**.

6.3 Spark

MRS deploys and hosts Apache Spark clusters in the cloud, and Spark is a distributed and parallel data processing framework. MRS uses the FusionInsight Spark commercial release, which has been officially certified by Databricks (a company founded by the creators of Apache Spark).

Fault-tolerant Spark is a distributed computing framework based on memory, which ensures that data can be quickly restored and recalculated. It is more efficient than MapReduce in terms of iterative data computing.

In the Hadoop ecosystem, Spark and Hadoop are seamlessly interconnected. By using HDFS for data storage and Yarn for resource management and scheduling, you can switch from MapReduce to Spark quickly.

Spark applies to the following scenarios:

- Data processing and ETL (extract, transform, and load)
- Machine learning
- Interactive analysis
- Iterative computing and data reuse. You benefit more from Spark when you perform operations frequently and the volume of the required data is large.
- On-demand capacity expansion. This is due to Spark's ease-of-use and low cost in the cloud.

For details about Spark architecture and principles, see **https://spark.apache.org/docs/2.3.2/quick-start.html**.

6.4 Spark SQL

Spark SQL is an important component of Apache Spark. It helps engineers familiar with traditional databases but unfamiliar with Spark technology to get started quickly. You can enter SQL statements directly to analyze, process, and query data.

Spark SQL has the following highlights:

- Is compatible with most Hive syntax, which enables seamless switchovers.
- Is compatible with standard SQL syntax.
- Resolves data skew problems. Spark SQL can join and convert skew data. It evenly distributes data that does not contain skewed keys to different tasks for processing. For data that contains skewed keys, Spark SQL broadcasts the smaller amount of data and uses the Map-Side Join to evenly distribute the data to different tasks for processing. This fully utilizes CPU resources and improves performance.
- Optimizes small files. Spark SQL employs the coalesce operator to process small files and combines partitions generated by small files in tables. This reduces the number of hash buckets during a shuffle operation and improves performance.

For details about Spark SQL architecture and principles, see **https://spark.apache.org/docs/2.3.2/rdd-programming-guide.html**.

6.5 HBase

Data storage is implemented by HBase. HBase is a column-oriented distributed cloud storage system that features enhanced reliability, excellent performance, and elastic scalability. It applies to the storage of massive amounts of data and distributed computing. You can use HBase to build a storage system capable of storing TB- or even PB-level data. With HBase, you can filter and analyze data with ease and get responses in milliseconds, rapidly mining data value.

HBase applies to the following scenarios:

- Storage of massive amounts of data

  You can use HBase to build a storage system capable of storing TB or PB of data. It also provides dynamic scaling capabilities so that you can adjust cluster resources to meet specific performance or capacity requirements.

- Real-time query

  The columnar and key-value storage models apply to the ad-hoc querying of enterprise user details. The low-latency point query, based on the master key, reduces the response latency to seconds or milliseconds, facilitating real-time data analysis.

HBase has the following highlights:

- Provides automatic Region recovery from an exception, ensuring reliability of data access.
- Enables data imported to the active cluster using BulkLoad to be automatically synchronized to the disaster recovery backup cluster. HBase also enhances the Replication feature, for example, supporting table structure synchronization, data synchronization between tables with system permissions, and the cluster readonly function.
- Improves performance of the BulkLoad feature, accelerating data import.
- The HBase secondary index enables HBase to query data based on specific column values. HBase can quickly locate the data that you want to read, improving data obtaining efficiency.

For details about HBase architecture and principles, see **https://hbase.apache.org/book.html**.

6.6 Hive

Hive is a database warehouse infrastructure built on top of Hadoop. It provides a series of tools that can be used to extract, transform, and load (ETL) data. Hive is a mechanism that can store, query, and analyze mass data stored on Hadoop. Hive defines simple SQL-like query language, which is known as HQL. It allows a user familiar with SQL to query data. In addition, this language allows a developer familiar with MapReduce to develop customized mapper and reducer for finishing complex analysis work that cannot be finished by the built-in mapper and reducer.

Hive system structure:

- User interface

Three user interfaces are available, that is, CLI, Client, and WUI. CLI is the most frequently-used user interface. A Hive transcript is started when CLI is started. Client refers to a Hive client, and a client user connects to the Hive Server. When entering the Client mode, specify the node where the Hive Server resides and start the Hive Server on this node. WUI is used to access Hive through a browser.

- Metadata storage

  Hive stores metadata into databases, for example, MySQL and Derby. The metadata in Hive includes the table name, table column and partition and their properties, table property (indicating whether the table is an external table), and directory where the data of the table is stored.

6.7 Tez

Tez is Apache's latest open source computing framework that supports Directed Acyclic Graph (DAG) jobs. It can convert multiple dependent jobs into one job, greatly improving the performance of DAG jobs. If projects like Hive and Pig use Tez instead of MapReduce as the backbone of data processing, response time will be significantly reduced. Tez is built on Yarn and can run MR jobs without any modification.

MRS uses Tez as the default execution engine of Hive. Tez remarkably surpasses the original MapReduce computing engine in terms of execution efficiency.

For details about Tez, see **https://tez.apache.org/**.

6.8 Hue

Hue is a web application developed based on the open source Django Python Web framework. It provides graphical user interfaces (GUIs) for users to configure, use, and view MRS clusters. Hue supports HDFS, Hive, MapReduce, and ZooKeeper in MRS clusters, including the following application scenarios:

- HDFS: You can create, view, modify, upload, and download files as well as create directories and modify directory permission.
- Hive: You can edit and execute HiveQL and add, delete, modify, and query databases, tables, and views through MetaStore.
- MapReduce: You can check MapReduce tasks that are being executed or have been finished in the clusters, including their status, start and end time, and run logs.
- ZooKeeper: You can check ZooKeeper status in the clusters.
- Sqoop: You can create, configure, run, and check Sqoop jobs.

For details about Hue, visit **http://gethue.com/**.

6.9 Kafka

MRS deploys and hosts Kafka clusters in the cloud based on the open source Apache Kafka. Kafka is a distributed, partitioned, replicated message publishing and subscription system. It provides features similar to Java Message Service (JMS) and has the following enhancements:

- Message persistency

Messages are stored in the storage space of clusters in persistence mode and can be used for batch consumption and real-time application programs. Data persistence prevents data loss.

- High throughput

  High throughput is provided for message publishing and subscription.

- Reliability

  Message processing methods such as At-Least Once, At-Most Once, and Exactly Once are provided.

- Distribution

  A distributed system is easy to be expanded. When new Core nodes are added for capacity expansion, the MRS cluster detects the nodes on which Kafka is installed and adds them to the cluster without interrupting services.

Kafka applies to online and offline message consumption. It is ideal for network service data collection scenarios, such as conventional data collection, website active tracing, data monitoring, and log collection.

For details about Kafka architecture and principles, see **https://kafka.apache.org/0100/documentation.html**.

6.10 Storm

MRS deploys and hosts Storm clusters in the cloud based on the open-source Apache Storm. Storm is a distributed, reliable, fault-tolerant computing system that processes large-volume streaming data in real time. It is applicable to real-time analysis, continuous computing, and distributed extract, transform, and load (ETL). It has the following features:

- Distributed real-time computing

  In a Storm cluster, each node runs multiple work processes; each work process creates multiple threads; each thread executes multiple tasks; and each task processes data concurrently.

- Fault tolerance

  During message processing, if a node or a process is faulty, the message processing unit can be redeployed.

- Reliable messages

  Data processing methods At-Least Once, At-Most Once, and Exactly Once are supported.

- Flexible topology defining and deployment

  The Flux framework is used to define and deploy service topologies. If the service DAG is changed, users only need to modify YAML domain specific language (DSL), but do not need to recompile or package service code.

- Integration with external components

  Storm supports integration with external components such as Kafka, HDFS, and HBase. This facilitates implementation of services that involve multiple data sources.

For details about Storm architecture and principles, see **https://storm.apache.org/**.

6.11 CarbonData

CarbonData is a new Apache Hadoop file format. It adopts the advanced column-oriented storage, index, compression, and encoding technologies and stores data in HDFS to improve computing efficiency. It helps accelerate the PB-level data query and is applicable to quicker interactive queries. CarbonData is also a high-performance analysis engine that integrates data sources with Spark. Users can execute Spark SQL statements to query and analyze data.

CarbonData has the following features:

- SQL

  CarbonData is compatible with Spark SQL and supports SQL query operations performed on Spark SQL.

- Simple definition of table data sets

  CarbonData supports defining and creating data sets by using user-friendly Data Definition Language (DDL) statements. CarbonData DDL is flexible and easy to use, and can define complex tables.

- Convenient data management

  CarbonData provides various data management functions for data loading and maintenance. It can load historical data and incrementally load new data. The loaded data can be deleted according to the loading time and specific data loading operations can be canceled.

- Quick query response

  CarbonData features high-performance query. It uses dedicated data formats and applies multiple index technologies, global dictionary code, and multiple push-down optimizations. The query speed is 10 times that of Spark SQL.

- Efficient data compression

  CarbonData compresses data by combining the lightweight and heavyweight compression algorithms. This saves 60% to 80% data storage space and the hardware storage cost.

- Table pre-aggregation

  CarbonData supports the pre-aggregation feature. You do not need to modify any SQL statement to accelerate the **group by** statistics performance and query detailed data. In this way, one copy of data meets multiple application scenarios.

- Real-time storage and query

  You can use Structured Streaming to import data to CarbonData tables in real time and immediately query the data.

- Partition table creation

  CarbonData enables you to create partition tables. You can use any column to create partitions to accelerate query.

- Table permission control

  CarbonData supports table permission control. You need permissions to operate databases and tables.

For details about CarbonData architecture and principles, see **https://carbondata.apache.org/**.

6.12 Flume

Flume is a distributed and highly available system for massive log aggregation. Users can customize data transmitters in Flume to collect data. Flume can also roughly process the data it receives.

Flume provides the following features:

- Collects and aggregates event stream data in a distributed approach.
- Collects log data.
- Supports dynamic configuration update.
- Provides the context-based routing function.
- Supports load balancing and failover.
- Provides comprehensive scalability.

For details about the Flume architecture and principles, see **https://flume.apache.org/releases/1.6.0.html**.

6.13 Loader

Loader is a data migration component developed based on Apache Sqoop. It quickens and simplifies the migration of structured, semi-structured, and unstructured data by Hadoop. Loader can both import and export data into and out of MRS clusters.

Loader provides the following features:

- Uses a high-available service architecture.
- Supports data migration using a client.
- Manages data migration jobs.
- Supports data processing during migration.
- Runs migration jobs using MapReduce components.

For details about the Loader architecture and principles, see **https://sqoop.apache.org/docs/1.99.7/index.html**.

6.14 Presto

The Presto component is available in MRS 1.8.5 or later.

Presto is an open source distributed SQL query engine for running interactive analytic queries against data sources of all sizes. It applies to massive structured/semi-structured data analysis, massive multi-dimensional data aggregation/report, ETL, ad-hoc queries, and more scenarios.

Presto allows querying data where it lives, including HDFS, Hive, HBase, Cassandra, relational databases or even proprietary data stores.

For details about Presto, visit **https://prestodb.github.io/**.

6.15 KafkaManager

KafkaManager is a tool for managing Apache Kafka and provides GUI-based metric monitoring and management of Kafka clusters.

KafkaManager supports the following functions:

- Manage multiple Kafka clusters.
- Easy inspection of cluster states (topics, consumers, offsets, partitions, replicas, and nodes)
- Run preferred replica election.
- Generate partition assignments with option to select brokers to use.
- Run reassignment of partition (based on generated assignments).
- Create a topic with optional topic configurations (Multiple Kafka cluster versions are supported).
- Delete a topic (only supported on 0.8.2+ and **delete.topic.enable=true** is set in broker configuration).
- Batch generate partition assignments for multiple topics with option to select brokers to use.
- Batch run reassignment of partitions for multiple topics.
- Add partitions to an existing topic.
- Update configurations for an existing topic.
- Optionally enable JMX polling for broker-level and topic-level metrics.
- Optionally filter out consumers that do not have ids/ owner / & offsets/ directories in ZooKeeper.

6.16 OpenTSDB

OpenTSDB is a distributed, scalable time series database based on HBase. OpenTSDB is designed to collect monitoring information of a large-scale cluster and implement second-level data query, eliminating the limitations of querying and storing massive amounts of monitoring data in common databases.

Application scenarios of OpenTSDB have the following features:

- The collected metrics have a unique value at a time point and do not have a complex structure or relationship.
- Monitoring metrics change with time.
- Like HBase, OpenTSDB features high throughput and good scalability.

OpenTSDB provides an HTTP based application programming interface to enable integration with external systems. Almost all OpenTSDB features are accessible via the API such as querying time series data, managing metadata, and storing data points. For details, visit **http://opentsdb.net/docs/build/html/api_http/index.html**.

6.17 Flink

Flink is a unified computing framework that supports both batch processing and stream processing. It provides a stream data processing engine that supports data distribution and parallel computing. Flink features stream processing and is a top open source stream processing engine in the industry.

Flink provides high-concurrency pipeline data processing, millisecond-level latency, and high reliability, making it extremely suitable for low-latency data processing.

The entire Flink system consists of three parts:

- Client

  Flink client is used to submit jobs (streaming jobs) to Flink.
- TaskManager

  TaskManager is a service execution node of Flink. It executes specific tasks. A Flink system can have multiple TaskManagers. These TaskManagers are equivalent to each other.
- JobManager

  JobManager is a management node of Flink. It manages all TaskManagers and schedules tasks submitted by users to specific TaskManagers. In high-availability (HA) mode, multiple JobManagers are deployed. Among these JobManagers, one is selected as the active JobManager, and the others are standby.

Flink provides the following features:

- Low latency

  Millisecond-level processing capability
- Exactly Once

  Asynchronous snapshot mechanism, ensuring that all data is processed only once
- HA

  Active/standby JobManagers, preventing single points of failure (SPOFs)
- Scale-out

  Manual scale-out supported by TaskManagers

For details about Flink, visit **https://flink.apache.org/**.

6.18 Cluster Management

MRS is a basic service on the public cloud and can be used to process, analyze, and compute massive amounts of data. MRS provides a web interface, on which you can perform the following operations.

- Creating a cluster: You can create a cluster on the MRS management console. Currently, **Pay-per-use** and **Yearly/Monthly** modes are supported. In **Pay-per-use** mode, nodes are charged by actual duration of use, with a billing cycle of one hour. In **Yearly/Monthly** mode, you can pay for nodes by year or month. The minimum cluster duration is 1 month and the maximum available cluster duration is 1 year. When fees are being deducted, if the account is in arrears, a message will be sent notifying the user to top up

the account. Cluster resources are frozen until the renewal fee has been paid. If no renewal fee is paid, cluster resources are deleted once the freeze period expires. The application scenarios of a cluster are as follows:

  - Data stored on OBS: Data storage and computing are performed separately. Cluster storage costs are low, and storage capacity is not limited. Clusters can be deleted at any time. The computing performance is determined by the OBS access performance and is lower than that of HDFS. OBS is recommended when data computing is infrequent.
  - Data stored in HDFS: Data storage and computing are performed together. Cluster storage costs are high, and storage capacity is limited. The computing performance is high. Data must be exported and stored before the clusters are deleted. HDFS is recommended when data computing is frequent.

- Deploying heterogeneous clusters: VMs of different specifications are supported and different CPU types, disk capacities, disk types, and memory sizes can be combined in a cluster. Various VM specifications can be mixed in a cluster.
- Management interface: MRS Manager functions as a unified management platform for MRS clusters.
  - Cluster monitoring enables you to quickly view the health status of hosts and services.
  - Graphical indicator monitoring and customization enable you to quickly obtain key information about the system.
  - Service property configurations can meet service performance requirements.
  - With cluster, service, and role instance functions, you can start or stop services and clusters in one click.
- Managing clusters: After completing data processing and analysis, you can manage and terminate clusters.
  - Querying alarms: If either the system or a cluster is faulty, MRS will collect fault information and report it to the network management system. Maintenance personnel will then be able to locate the faults.
  - Querying logs: To help locate cluster faults, cluster and job operation information is recorded.
  - Managing files: MRS allows data to be imported from OBS to HDFS and exported from HDFS to OBS after analysis and processing. You can also store data in HDFS.
- Adding a job: A job is an executable program provided by MRS to process and analyze user data. Currently, MRS supports MapReduce jobs, Spark jobs, and Hive jobs, and allows users to submit Spark SQL statements online to query and analyze data.
- Managing jobs: Jobs can be managed, stopped, or deleted. You can also view details of the completed jobs along with detailed configurations. However, Spark SQL jobs cannot be stopped.
- Expanding clusters: Add Core or Task nodes to expand clusters and handle peak service loads.
- Shrinking clusters: Reduce the number of Core or Task nodes to shrink the cluster so that MRS delivers better storage and computing capabilities at lower O&M costs based on service requirements.
- Auto scaling: Automatically adjust computing resources based on service requirements and the preset policies, so that the number of Task nodes can be automatically scaled out or in with service load changes, ensuring stable service running.

- Adding Task nodes: A Task node processes data rather than storing cluster data. When the number of clusters does not change much but the clusters' service processing capabilities need to be remarkably and temporarily improved, add Task nodes to address the following situations:
  - The number of temporary services is increased, for example, report processing at the end of the year.
  - Long-term tasks need to be completed in a short time, for example, some urgent analysis tasks.
- Adding tags: A tag is used to identify a cluster. Adding tags to clusters can help you identify and manage your cluster resources.

  You can add a maximum of 10 tags to a cluster when creating the cluster or add them on the details page of the created cluster.
- Adding bootstrap actions: Bootstrap actions indicate that you can run your scripts on a specified cluster node before or after starting big data components. You can add bootstrap actions when creating a cluster to install third-party software, modify the cluster running environment, and perform other customizations. If you choose to run bootstrap actions when expanding a cluster, the bootstrap actions will be run on the newly added nodes in the same way.

6.19 Expanding Clusters Charged in Yearly/Monthly Mode

If your service growth rate exceeds the expected value after you subscribe to an MRS cluster charged in **Yearly/Monthly** mode, cluster capacity expansion beyond your subscription is required. MRS allows you to expand clusters charged in **Yearly/Monthly** mode while enjoying the subscription discounts.

You can access the MRS management console and add nodes to a cluster with a few clicks. The cluster expansion process does not require manual intervention and takes only a few minutes, which helps ease pressure on growing service data processing needs.

6.20 Multi-disk Attaching

You can attach multiple disks to non-Master nodes when creating a cluster so that data directories of services such as HDFS and Kafka can be distributed to multiple disks to meet service requirements on disk read/write speed and cluster storage capability. This improves cluster running efficiency and provides better user experience.

6.21 Kerberos Authentication

Overview

To ensure data security for users, MRS clusters provide user identity verification and user authentication functions. To enable all verification and authentication functions, you must enable Kerberos authentication when creating the cluster.

Identity Verification

The user identity verification function verifies the identity of a user when the user performs O&M operations or accesses service data in a cluster.

If the user restarts services in an MRS cluster on MRS Manager, the user must enter the password of the current account on MRS Manager, for example, restarting services and synchronizing cluster configurations.

Authentication

Users with different identities may have different permissions to access and use cluster resources. To ensure data security, users must be authenticated after identity verification.

Identity Verification

Clusters that support Kerberos authentication use the Kerberos protocol for identity verification. The Kerberos protocol supports mutual verification between clients and servers. This eliminates the risks incurred by sending user credentials over the network for simulated verification. In MRS clusters, KrbServer provides the Kerberos authentication function.

Kerberos User Object

In the Kerberos protocol, each user object is a principal. A complete principal consists of two parts: username and domain name. In O&M or application development scenarios, the user identity must be verified before a client connects to a server. Users for O&M and service operations in MRS clusters are classified into **Human-Machine** and **Machine-Machine** users. The password of **Human-Machine** users is manually configured, while the password of **Machine-Machine** users is generated by the system randomly.

Kerberos Authentication

Kerberos supports two authentication modes: password and keytab. The default verification validity period is 24 hours.

- Password verification: User identity is verified by inputting the correct password. This mode mainly applies to O&M scenarios where **Human-Machine** users are used. The configuration command is **kinit** *user name*.
- Keytab verification: Keytab files contain users' security information. During keytab verification, the system automatically uses the encrypted credential information for verification. Users do not need to enter the password. This mode mainly applies to component application development scenarios where **Machine-Machine** users are used. Keytab verification can also be configured using the **kinit** command.

Authentication

After identity verification for users, the MRS system also authenticates the users to ensure that they have limited or full permission on cluster resources. If a user does not have the permission for accessing cluster resources, the system administrator must grant the required permission to the user. Otherwise, the user fails to access the resources.

6.22 Task Node Creation

Feature Introduction

Task nodes can be created and used for computing only. They do not store persistent data and are the basis for implementing auto scaling.

Customer Benefits

When MRS is used only as a computing resource, Task nodes can be used to reduce costs and facilitate cluster node scaling, flexibly meeting users' requirements for increasing or decreasing cluster computing capabilities.

Application Scenarios

When the number of clusters does not change but the clusters' service processing capabilities need to be remarkably and temporarily improved, add Task nodes to address the following situations:

- The number of temporary services is increased, for example, report processing at the end of the year.
- Long-term tasks need to be completed in a short time, for example, some urgent analysis tasks.

6.23 Auto Scaling

Feature Introduction

More and more enterprises use technologies such as Spark and Hive to analyze data. Processing a large amount of data consumes huge resources and costs much. Typically, enterprises regularly analyze data in a fixed period of time every day rather than all day long. To meet enterprises' requirements, MRS provides the auto scaling function to apply for extra resources during peak hours and release resources during off-peak hours. This enables users to use resources on demand and focus on core business at lower costs.

Customer Benefits

- Reducing costs

  Enterprises do not analyze data all the time but perform a batch data analysis in a specified period of time, for example, 03:00 a.m. The batch analysis may take only two hours.

  The auto scaling function enables enterprises to add nodes for batch analysis and automatically releases the nodes after completion of the analysis, minimizing costs.

- Meeting instant query requirements

  Enterprises usually encounter instant analysis tasks, for example, data reports for enterprise decision-making. As a result, resource consumption increases sharply in a short period of time. With the auto scaling function, computing nodes can be added for emergent big data analysis, avoiding a service breakdown due to insufficient computing resources. You do not need to purchase extra resources. After the emergency is settled down, MRS can automatically release the nodes.

- Focusing on core business

  It is difficult for developers to determine resource consumption on the big data secondary development platform because of complex query analysis conditions (such as global sorting, filtering, and merging) and data complexity, for example, uncertainty of incremental data. As a result, estimating the computing volume is difficult. MRS's auto scaling function enables developers to focus on service development without the need for resource estimation.

6.24 Message Notification

Feature Introduction

MRS clusters often have updates, for example, scale-out and scale-in, auto scaling triggered for an emergent data volume increase, and cluster termination. To immediately notify you of successful operations, cluster unavailability, and node faults, MRS uses Simple Message Notification (SMN) to send notifications to you through SMS and emails, facilitating maintenance.

Customer Benefits

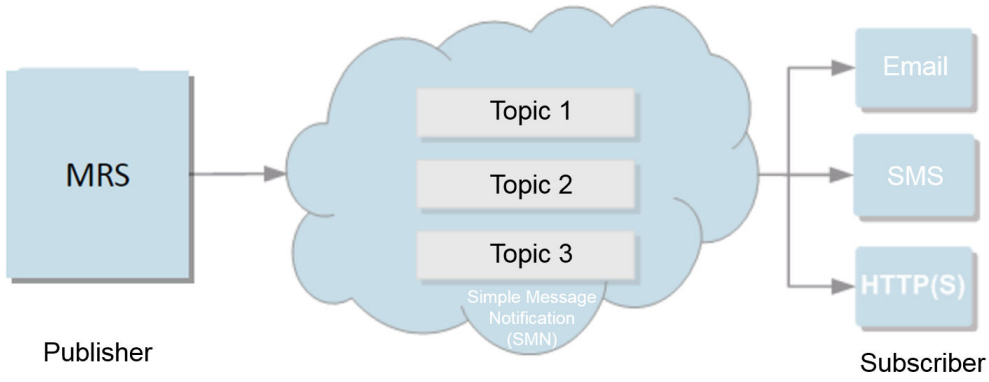
After configuring SMN, you can receive MRS cluster health status, updates, and component alarms through SMS or emails in real time. MRS sends real-time monitoring and alarm notification to help you easily perform O&M and efficiently deploy big data services.

Feature Description

MRS uses SMN to provide one-to-multiple message subscription and notification over a variety of protocols.

You can create a topic and configure topic policies to control publisher and subscriber permissions on the topic. Then, MRS sends cluster messages to a topic on which you have the permission to publish a message. Then all subscribers who subscribe to the topic can receive cluster updates and component alarms through SMS and emails.

Figure 6-1 Implementation process



6.25 Bootstrap Actions

Feature Introduction

MRS provides standard elastic big data clusters on the cloud. Nine big data components, such as Hadoop and Spark, can be installed and deployed. Currently, standard cloud big data clusters cannot meet all user requirements, for example, in the following scenarios:

- Common operating system configurations cannot meet data processing requirements, for example, increasing the maximum number of system connections.
- Software tools or running environments need to be installed, for example, Gradle and dependency R language package.
- Big data component packages need to be modified based on service requirements, for example, modifying the Hadoop or Spark installation package.
- Other big data components that are not supported by MRS need to be installed.

To meet the preceding customization requirements, you can manually perform operations on the existing and newly added nodes. The overall process is complex and error-prone. In addition, manual operations cannot be traced, and data cannot be processed immediately after a pay-per-use cluster is created.

Therefore, MRS supports custom bootstrap actions that enable you to run scripts on a specified node before or after a cluster component is started. You can run bootstrap actions to install third-party software that is not supported by MRS, modify the cluster running environment, and perform other customizations. If you choose to run bootstrap actions when expanding a cluster, the bootstrap actions will be run on the newly added nodes in the same way.

MRS runs the script you specify as user **root**. You can run the **su - *XXX*** command in the script to switch the user.

Customer Benefits

You can use the custom bootstrap actions to flexibly and easily configure your dedicated clusters and customize software installation.

7 Related Services

MRS works with the following services:

- Virtual Private Cloud (VPC)

  MRS clusters are created in the subnets of a VPC. VPCs provide a secure, isolated, and logical network environment for your MRS clusters.

- Object Storage Service (OBS)

  OBS stores the following user data:

  - MRS job input data, such as user programs and data files
  - MRS job output data, such as result files and log files of jobs

  In MRS clusters, HDFS, Hive, MapReduce, Yarn, Spark, Flume, and Loader can import or export data from OBS.

- Relational Database Service (RDS)

  RDS stores MRS system running data, including MRS cluster metadata and user billing information.

- Elastic Cloud Server (ECS)

  Each node in an MRS cluster is an ECS.

- Identity and Access Management (IAM)

  IAM provides authentication for MRS.

- Simple Message Notification (SMN)

  MRS uses SMN to provide one-to-multiple message subscription and notification over a variety of protocols.

- Cloud Trace Service (CTS)

  CTS provides you with operation records of MRS resource operation requests and request results for querying, auditing, and backtracking.

Table 7-1 MRS operations recorded by CTS

Operation	Resource Type	Trace Name
Creating a cluster	cluster	createCluster
Deleting a cluster	cluster	deleteCluster

Operation	Resource Type	Trace Name
Expanding a cluster	cluster	scaleOutCluster
Shrinking a cluster	cluster	scaleInCluster

After you enable CTS, the system starts recording operations on cloud resources. You can view operation records of the last 7 days on the CTS management console. For details, see **Cloud Trace Service** > **Getting Started** > **Querying Real-Time Traces**.

8 Restrictions

Before using MRS, ensure that you have read and understood the following restrictions.

- MRS clusters must be created in VPC subnets.
- You are advised to use any of the following browsers to access MRS:
  - Google Chrome: 36.0 or later
  - Internet Explorer: 9.0 or later

    If you use Internet Explorer 9.0, you may fail to log in to the MRS management console because user **Administrator** is disabled by default in some Windows systems, such as Windows 7 Ultimate. Internet Explorer automatically selects a system user for installation. As a result, Internet Explorer cannot access the management console. Reinstall Internet Explorer 9.0 or later (recommended) or run Internet Explorer 9.0 as user **Administrator**.
- When you create an MRS cluster, you can select **Auto Create** from the drop-down list of **Security Group** to create a security group or select an existing security group. After the MRS cluster is created, do not delete or modify the used security group. Otherwise, a cluster exception may occur.
- To prevent illegal access, only assign access permission for security groups used by MRS where necessary.
- Do not perform the following operations because they will cause cluster exceptions:
  - Shutting down, restarting, or deleting MRS cluster nodes displayed in ECS, changing or reinstalling their OS, or modifying their specifications.
  - Deleting the existing processes, applications, or files on cluster nodes.
  - Deleting MRS cluster nodes. Deleted nodes will still be charged.
- If a cluster exception occurs when no incorrect operations have been performed, contact technical support engineers. The technical support engineers will ask you for your password and then perform troubleshooting.
- Keep the initial password for logging in to the Master node properly because MRS will not save it. Use a complex password to avoid malicious attacks.
- MRS clusters are still charged during exceptions. Contact technical support engineers to handle cluster exceptions.
- Plan disks of cluster nodes based on service requirements. If you want to store a large volume of service data, add EVS disks or storage space to prevent insufficient storage space from affecting node running.

- The cluster nodes store only users' service data. Non-service data can be stored in the OBS or other ECS nodes.
- The cluster nodes only run MRS cluster programs. Other client applications or user service programs are deployed on separate ECS nodes.
- The storage capacity of MRS cluster nodes (including Master, Core, and Task nodes) can be expanded only by attaching new disks instead of expanding capacity of the existing disks.

9 Permissions Management

If you need to assign different permissions to employees in your enterprise to access your MRS resources, IAM is a good choice for fine-grained permissions management. IAM provides identity authentication, permissions management, and access control, helping you secure access to your HUAWEI CLOUD resources.

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

If your HUAWEI CLOUD account does not need individual IAM users for permissions management, then you may skip over this section.

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

Supported System Policies

A policy is a set of permissions defined in JSON format. By default, new IAM users do not have any permissions assigned. You need to add a user to one or more groups, and assign permissions policies to these groups. The user then inherits permissions from the groups it is a member of. This process is called authorization. After authorization, the user can perform specified operations on MRS based on the permissions. IAM provides system policies that define the common permissions for different services, such as administrator and read-only permissions. You can directly use these system policies to assign permissions.

MRS is a project-level service deployed in specific physical regions. Therefore, MRS permissions are assigned to users in specific regions (such as **CN North-Beijing1**) and only take effect for these regions. If you want the permissions to take effect for all regions, you need to assign the permissions to users in each region. When accessing MRS, the users need to switch to a region where they have been authorized to use the MRS service.

Table 9-1 lists all the system policies supported by MRS.

Policy type: There are fine-grained policies and role-based access control (RBAC) policies. Fine-grained policies, as the name suggests, allow for more fine-grained control than RBAC policies. Fine-grained policies are currently available for open beta testing. You can apply to

use the fine-grained access control function free of charge. For more information, see **Fine-grained Policy**.

- RBAC policy: An RBAC policy consists of permissions for an entire service. Users in a group with such a policy assigned are granted all of the permissions required for that service, such as the permissions for accessing and managing that service. RBAC policies do not support operation-specific permission control.
- Fine-grained policy: A fine-grained policy consists of API-based permissions for operations on specific resource types. Fine-grained policies, as the name suggests, allow for more fine-grained control than RBAC policies. Users with such a policy assigned are allowed or not allowed to perform specific operations on the corresponding service. For example, users can only perform basic operations on MRS, such as creating clusters and querying a cluster list, but cannot delete clusters. For the API actions supported by MRS, see **Permissions Policies and Supported Actions**.

Table 9-1 System policy summary

Policy Name	Description	Policy Type
MRS Admin	Administrator permissions for MRS. Users granted these permissions can operate and use all MRS resources.	Fine-grained policy
MRS User	Common user permissions for MRS. Users granted these permissions can use MRS but cannot add or delete resources.	Fine-grained policy
MRS Viewer	Read-only permission for MRS. Users granted these permissions can only view MRS resources.	Fine-grained policy
MRS Administrator	Permissions: ● All operations on MRS ● Users with permissions of this policy must also be granted permissions of the **Tenant Guest**, **Server Administrator**, and **BSS Administrator** policies.	RBAC policy

Table 9-2 lists the common operations supported by each system policy of MRS. Please choose proper system policies according to this table.

Table 9-2 Common operations supported by each system policy

Operation	MRS Admin	MRS User	MRS Viewer	MRS Administrator
Creating a cluster	√	x	x	√

Operation	MRS Admin	MRS User	MRS Viewer	MRS Administrator
Resizing a cluster	√	x	x	√
Upgrading node specifications	√	x	x	√
Deleting a cluster	√	x	x	√
Querying cluster details	√	√	√	√
Querying a cluster list	√	√	√	√
Configuring an auto scaling rule	√	x	x	√
Querying a host list	√	√	√	√
Querying operation logs	√	√	√	√
Creating and executing a job	√	√	x	√
Stopping a job	√	√	x	√
Deleting a single job	√	√	x	√
Deleting jobs in batches	√	√	x	√
Querying job details	√	√	√	√
Querying a job list	√	√	√	√
Creating a folder	√	√	x	√

Operation	MRS Admin	MRS User	MRS Viewer	MRS Administrator
Deleting a file	√	√	x	√
Querying a file list	√	√	√	√
Operating cluster tags in batches	√	√	x	√
Creating a single cluster tag	√	√	x	√
Deleting a single cluster tag	√	√	x	√
Querying a resource list by tag	√	√	√	√
Querying cluster tags	√	√	√	√
Accessing MRS Manager	√	√	x	√
Querying a patch list	√	√	√	√
Installing a patch	√	√	x	√
Uninstalling a patch	√	√	x	√
Authorizing O&M channels	√	√	x	√
Sharing O&M channel logs	√	√	x	√
Querying an alarm list	√	√	√	√
Subscribing to alarm notification	√	√	x	√

Helpful Links

- **IAM Service Overview**
- **Creating User Groups and Users and Granting MRS Permissions**
- **Policy Syntax**
- **Permissions Policies and Supported Actions**

10 Quota Description

MRS uses the following infrastructure resources:

- ECS
- VPC
- Elastic Volume Service (EVS)
- Image Management Service (IMS)
- OBS
- Elastic IP (EIP)
- Simple Message Notification (SMN)
- IAM

For details about how to view and modify quotas, see **Quotas**.

11 Version Description

Table 11-1 Version list

Version	Release Date	Description
MRS 2.0.3	2019-09-23	● Rolled out the optimized MRS UI and migrated content on MRS Manager to the MRS management console. ● Added permission control for multiple MRS users to access OBS. ● Added the function of managing external data source connections.
MRS 1.8.7	2019-08-16	● Added the OpenTSDB component. ● Added the Flink component. ● Added the function of submitting jobs on the page of a security cluster. ● Updated the cluster purchase page.
MRS 2.0.1	2019-06-18	● Upgraded Hadoop to version 3.1.1. ● Upgraded Spark to version 2.3.2. ● Upgraded HBase to version 2.1.1. ● Upgraded Hive to version 3.1.0. ● Added the Tez component. ● Supported the auto scaling function for yearly/monthly subscribed clusters. ● Supported the creation of hybrid clusters.