Contents

1 What Is MRS?........................................................................................................................... 1
2 Advantages of MRS Compared with Self-Built Hadoop.................................................5
3 Application Scenarios........................................................................................................... 11
4 Components............................................................................................................................14
  4.1 List of MRS Component Versions...................................................................................... 14
  4.2 HDFS: Hadoop Distributed File System.......................................................................... 15
  4.3 YARN: Unified Resource Management and Scheduling Framework............................ 16
  4.4 MapReduce: Distributed Batch Processing Engine......................................................... 17
  4.5 Hive: Data Warehouse Infrastructure............................................................................... 18
  4.6 HBase: Distributed Database............................................................................................ 19
  4.7 Spark: Distributed In-Memory Computing Engine......................................................... 20
  4.8 CarbonData...................................................................................................................... 21
  4.9 Loader: Batch Data Integration....................................................................................... 23
  4.10 Hue: Data Integration Access....................................................................................... 26
  4.11 Presto: Interactive SQL Query Engine.......................................................................... 26
  4.12 Tez: High-speed Computing Engine Supporting DAG Models.................................... 27
  4.13 Flume: Real-Time Data Ingestion................................................................................ 28
  4.14 Storm: Stream Computing Engine................................................................................. 29
  4.15 Flink.............................................................................................................................. 30
  4.16 Kafka: Distributed Message Queue................................................................................ 32
  4.17 KafkaManager............................................................................................................... 33
  4.18 OpenTSDB.................................................................................................................... 33
  4.19 Impala............................................................................................................................ 34
  4.20 Kudu............................................................................................................................... 35
  4.21 Alluxio............................................................................................................................ 36
  4.22 Ranger............................................................................................................................ 36
5 Functions................................................................................................................................. 38
  5.1 New Functions of MRS 2.0............................................................................................ 38
  5.2 Multi-tenant..................................................................................................................... 42
  5.3 Security Hardening......................................................................................................... 43
  5.4 Easy Access to Web UIs of Components........................................................................ 45
  5.5 Reliability Enhancement............................................................................................... 45
5.6 Job Management .................................................................................................................................................................. 47
5.7 Bootstrap Actions .................................................................................................................................................................. 47
5.8 Enterprise Project Management ....................................................................................................................................... 48
5.9 Cluster Management ........................................................................................................................................................... 48
5.9.1 Cluster Lifecycle Management ....................................................................................................................................... 48
5.9.2 Manually Scale Out/In a Cluster ..................................................................................................................................... 51
5.9.3 Auto Scaling ........................................................................................................................................................................ 51
5.9.4 Task Node Creation ........................................................................................................................................................... 53
5.9.5 Scaling Up Master Node Specifications .............................................................................................................................. 53
5.9.6 Isolating a Host ................................................................................................................................................................. 53
5.9.7 Managing Tags ..................................................................................................................................................................... 54
5.10 Cluster O&M ....................................................................................................................................................................... 54
5.11 Message Notification .......................................................................................................................................................... 55
6 Restrictions .............................................................................................................................................................................. 57
7 Billing ....................................................................................................................................................................................... 59
8 Permissions Management ......................................................................................................................................................... 61
9 Related Services ....................................................................................................................................................................... 66
10 Quota Description ................................................................................................................................................................. 68
11 Change History ........................................................................................................................................................................ 69
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. MRS is fully compatible with open source APIs, and incorporates advantages of HUAWEI CLOUD computing and storage and big data industry experience to provide customers with a full-stack big data platform featuring high performance, low cost, flexibility, and ease-of-use. In addition, the platform can be customized based on service requirements to help enterprises quickly build a massive data processing system and discover new value points and business opportunities by analyzing and mining massive amounts of data in real time or in non-real time.

Product Architecture

Figure 1-1 shows the logical architecture of HUAWEI CLOUD MRS.
MRS architecture includes infrastructure and big data processing phases.

- **Infrastructure**
  MRS big data clusters are built based on HUAWEI CLOUD Elastic Cloud Server (ECS), and make full use of the high reliability and security capabilities of the virtualization layer.
  - A Virtual Private Cloud (VPC) is a virtual internal network provided for each tenant. It is isolated from other networks by default.
  - Elastic Volume Service (EVS) provides highly reliable and high-performance storage.
  - ECS provides scalable VMs, and works with VPCs, security groups, and the EVS multi-replica mechanism to build an efficient, reliable, and secure computing environment.

- **Data integration**
  The data integration layer provides data access capabilities of MRS clusters, including components Flume (data ingestion), Loader (relational data import), and Kafka (highly reliable message queue). Data can be imported to MRS clusters from various data sources.

- **Data storage**
  MRS clusters can store structured and unstructured data, and support multiple efficient formats to meet the requirements of different computing engines.
  - HDFS is a general-purpose distributed file system on a big data platform.
  - OBS is an object storage service that features high availability and low cost.
  - HBase supports data storage with indexes, and is applicable to high-performance index-based query scenarios.

- **Data computing**
  MRS provides multiple mainstream computing engines, including MapReduce (batch processing), Tez (DAG model), Spark (in-memory computing),
SparkStreaming (micro-batch stream computing), Storm (stream computing), and Flink (stream computing), to meet the requirements of various big data application scenarios. The engines convert data structures and logic into data models that meet service requirements.

- Data analysis
  Based on the preset data model and easy-to-use SQL data analysis, users can select Hive (data warehouse), SparkSQL, and Presto (interactive query engine).

- Data display and scheduling
  To present data analysis results, MRS is integrated with Data Lake Factory (DLF), which is a one-stop big data collaboration development platform, to help you easily complete multiple tasks, such as data modeling, data integration, script development, job scheduling, and job monitoring. This makes big data more accessible than ever before, helping you quickly build big data processing centers.

- Cluster management
  All components of the Hadoop-based big data ecosystem are deployed in distributed mode, and their deployment, management, and O&M are complex.
  MRS provides a unified O&M management platform for cluster management, supporting one-click cluster deployment, multi-version selection, as well as manual scaling and auto scaling of clusters without service interruption. In addition, MRS provides job management, resource tag management, and O&M of the preceding data processing components at each layer. It also provides one-stop O&M capabilities, covering monitoring, alarm reporting, configuration, and patch upgrade.

Product Advantages

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.

**MRS Learning Paths**

You can quickly understand MRS and learn how to use MRS by referring to Progressive Knowledge.
MRS provides enterprise-level big data clusters on the cloud. Tenants can fully control the clusters and run big data components such as Hadoop, Spark, HBase, Kafka, and Storm with ease. MRS frees you from hardware purchase and maintenance. MRS is built based on Huawei FusionInsight big data enterprise-class platform, and has been deployed on tens of thousands of nodes in the industry, providing multi-level SLA assurance with professional Hadoop kernel service support. Compared with self-built Hadoop clusters, MRS has the following advantages:

1. **MRS supports one-click cluster creation, deletion, and scaling. You can use an elastic IP address (EIP) to access MRS Manager, making big data clusters easier to use.**
   - Self-built big data clusters pose problems such as high costs, long periods, difficult and inflexible O&M. To solve these problems, MRS provides one-click cluster creation, deletion, scale-out, and scale-in, allowing you to customize the cluster type, component range, number of nodes of each type, VM specifications, availability zones (AZs), VPC network, and authentication information. MRS can automatically create a cluster that meets the configuration requirements. In addition, you can quickly create multi-application clusters, for example, Hadoop analysis cluster, HBase cluster, and Kafka cluster. MRS supports heterogeneous cluster deployment. That is, VMs of different specifications can be combined in a cluster based on CPU types, disk capacities, disk types, and memory sizes.
   - MRS provides an EIP-based secure channel for you to easily access the web UIs of components. This is more convenient than binding an EIP by yourself, and you can access the web UIs with a few clicks, avoiding the steps for logging in to a VPC, adding security group rules, and obtaining a public IP address.
   - MRS provides custom bootstrap actions to flexibly configure your dedicated clusters. Third-party software that is not supported by MRS can be automatically installed, allowing you to perform custom operations such as modifying the cluster running environment.
   - MRS supports the WrapperFS feature, provides the OBS translation capability (that is, access to OBS through address mapping) and can
smoothly migrate data from HDFS to OBS. After migration, you can access the data stored in OBS from clients without modifying service code logic.

2. **MRS supports auto scaling, which is more cost-effective than the self-built Hadoop cluster.**

MRS supports auto scaling to address peak and off-peak service loads. It applies for extra resources during peak hours and releases idle resources during off-peak hours, helping you save idle resources on the big data platform during off-peak hours, minimize costs, and focus on core services. In big data applications, especially in periodic data analysis and processing, cluster computing resources need to be dynamically adjusted based on service data changes to meet service requirements. The auto scaling function of MRS enables clusters to be elastically scaled out or in based on cluster loads. In addition, if the data volume changes regularly and you want to scale out or in a cluster before the data volume changes, you can use the MRS resource plan feature. MRS supports two types of auto scaling policies: auto scaling rules and resource plans.

   - **Auto scaling rules:** You can increase or decrease Task nodes based on real-time cluster loads. Auto scaling will be triggered when the data volume changes but there may be some delay.
   - **Resource plans:** If the data volume changes periodically, you can create resource plans to resize the cluster before the data volume changes, thereby avoiding a delay in increasing or decreasing resources.

Both auto scaling rules and resource plans can trigger auto scaling. You can configure both of them or configure one of them. Configuring both resource plans and auto scaling rules improves the cluster node scalability to cope with occasionally unexpected data volume peaks.

3. **MRS supports storage-compute decoupling, greatly improving the resource utilization of big data clusters.**

In the traditional big data architecture where storage and compute resources are integrated, scaling-out is difficult and resources are not well-utilized. To solve these problems, MRS adopts a compute-storage separation architecture. Based on OBS, the storage achieves 99.999999999% reliability and unlimited capacity, supporting continuous growth of enterprise data. Computing resources can be elastically scaled in or out from 0 to \(N\) nodes. Hundreds of nodes can be quickly provisioned. With the new architecture, compute nodes can be elastically scaled. OBS-based cross-AZ data storage ensures higher reliability, frees you from worrying about emergencies such as earthquakes and fiber cuts. Storage and compute resources can be flexibly configured and elastically scaled as required. This makes resource allocation more accurate and reasonable, greatly improving the resource utilization of big data clusters and reducing the comprehensive analysis cost by 50%.

In addition, the high performance compute-storage separation architecture breaks the limit of parallel computing of the integrated storage-compute architecture. It maximizes the high bandwidth and high concurrency of OBS, and optimizes the data access efficiency and in-depth parallel computing (such as metadata operation and write algorithm optimization) to improve higher performance.

4. **MRS supports Huawei-developed CarbonData and Superior Scheduler, delivering better 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 Superior Scheduler, which enhances the scaling capability of a single cluster and is capable of scheduling over 10,000 nodes in a cluster. Superior Scheduler is a scheduling engine designed for the Hadoop Yarn distributed resource management system. It is a high-performance and enterprise-level scheduler designed for converged resource pools and multi-tenant service requirements. Superior Scheduler achieves all functions of open source schedulers, Fair Scheduler, and Capacity Scheduler. Compared with the open source schedulers, Superior Scheduler is enhanced in the enterprise multi-tenant resource scheduling policy, resource isolation and sharing by multiple users in a tenant, scheduling performance, system resource utilization, and cluster scalability, and is designed to replace open source schedulers.

5. **MRS optimizes software and hardware based on Kunpeng processors to fully release hardware computing power and achieve cost-effectiveness.**

MRS supports Huawei-developed Kunpeng servers whose multi-core and high-concurrency capabilities are fully utilized to provide full-stack self-optimized chips, and uses Huawei-developed EulerOS, Huawei JDK, and data acceleration layer to ensure hardware performance, delivering high computing power for big data computing. With the similar performance, the cost of the end-to-end big data solution is reduced by 30%.

6. **MRS supports multiple isolation modes and multi-tenant permission management of enterprise-level big data, ensuring higher security.**

- MRS supports resource deployment and isolation of physical resources in dedicated zones. You can flexibly combine computing and storage resources, such as dedicated computing resources + shared storage resources, shared computing resources + dedicated storage resources, and dedicated computing resources + dedicated storage resources. An MRS cluster supports multiple logical tenants. Permission isolation enables the computing, storage, and table resources of the cluster to be divided based on tenants.

- With Kerberos authentication, MRS provides role-based access control (RBAC) and sound audit functions.

- With Cloud Trace Service (CTS) being interconnected with MRS, you are provided with operation records of MRS resource operation requests and request results for querying, auditing, and backtracking. You can use CTS to audit and trace all cluster operations.

- It is proved that with Host Security Service (HSS) interconnected with MRS, service security is enhanced without deteriorating functions and performance.

- MRS supports unified user login based on web UI. MRS Manager provides user authentication, which grants you permission to access a cluster.
- MRS supports data storage encryption, encrypted storage of all user accounts and passwords, encrypted transmission of data channels, and bidirectional certificate authentication for cross-trusted-zone data access of service modules.

- MRS big data clusters provide a complete multi-tenant solution for enterprise-level big data. Multi-tenant refers to a collection of multiple resources (each resource set is a tenant) in an MRS big data cluster. It can allocate and schedule resources, including computing and storage resources. Multi-tenant isolates the resources of a big data cluster into resource sets. Users can lease desired resource sets to run applications and jobs and store data. In a big data cluster, multiple resource sets can be deployed to meet diverse requirements of multiple users.

- MRS supports fine-grained permission management. With the fine-grained authorization capability provided by HUAWEI CLOUD IAM, MRS can specify the operations, resources, and request conditions of specific services. This mechanism allows for more flexible policy-based authorization, meeting requirements for secure access control. For example, you can grant MRS users only the permissions for performing specified operations on MRS clusters, such as creating a cluster and querying a cluster list rather than deleting a cluster. In addition, MRS supports fine-grained permission management of OBS for multiple tenants. Permissions to access OBS buckets and objects in the buckets are differentiated based on user roles, so that MRS users can each control a different directory in OBS buckets.

- MRS supports enterprise project management. The enterprise project is one way of managing cloud resources. Enterprise Management provides comprehensive management services for enterprise customers, such as cloud resources, personnel, permissions, and financial statuses. Common management consoles are oriented to the control and configuration of individual cloud products. The Enterprise Management console, in contrast, is more focused on resource management. It is designed to help enterprises manage cloud-based resources, personnel, permissions, and finances, in a hierarchical management manner, such as management of companies, departments, and projects. MRS allows users who have enabled Enterprise Project Management Service (EPS) to configure enterprise projects for a cluster during cluster creation and use EPS to manage MRS resources by group. This feature is applicable to scenarios where you need to manage multiple resources by group and perform operations such as permission control and project-based fee query on enterprise projects.

7. MRS implements HA for all management nodes and supports comprehensive reliability mechanism, making the system more reliable.

Based on Apache Hadoop open source software, MRS optimizes and improves the reliability of main service components.

- HA for all management nodes

In the Hadoop open source version, data and compute nodes are managed in a distributed system, in which a single point of failure (SPOF) does not affect the operation of the entire system. However, a SPOF may occur on management nodes running in centralized mode, which becomes the weakness of the overall system reliability.
MRS provides similar double-node mechanisms for all management nodes of the service components, such as MRS Manager, Presto, HDFS NameNodes, Hive Servers, HBase HMasters, Yarn ResourceManagers, Kerberos Servers, and Ldap Servers. All of them are deployed in active/standby mode or configured with load sharing, effectively preventing SPOFs from affecting system reliability.

- Comprehensive reliability mechanism

By reliability analysis, the following measures to handle software and hardware exceptions are provided to improve the system reliability:

- After power supply is restored, services are running properly regardless of a power failure of a single node or the whole cluster, ensuring data reliability in case of unexpected power failures. Key data will not be lost unless the hard disk is damaged.
- Health status checks and fault handling of the hard disk do not affect services.
- The file system faults can be automatically handled, and affected services can be automatically restored.
- The process and node faults can be automatically handled, and affected services can be automatically restored.
- The network faults can be automatically handled, and affected services can be automatically restored.

8. **MRS provides a visualized big data cluster management interface in a unified manner, making O&M easier.**

- On the big data cluster management interface, service startup and stopping, configuration modification, and health check are available. MRS also provides visualized and convenient cluster management, monitoring, and alarm functions. Additionally, you can check and audit the system health status in one click, ensuring normal system running and lowering system O&M costs.
- After Simple Message Notification (SMN) is configured, MRS can send real-time cluster health status information, including cluster changes and component alarms in real time to you through SMS messages or emails, facilitating O&M, real-time monitoring, and real-time alarm sending.
- 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.
- If a problem occurs when you use an MRS cluster, you can initiate O&M authorization on the MRS management console. O&M personnel can help you quickly locate the problem, and you can revoke the authorization at any time. You can also initiate log sharing on the MRS management console to share a specified log scope with O&M personnel, so that O&M personnel can locate faults without accessing the cluster.
- MRS supports to dump logs about cluster creation failures to OBS for O&M personnel to obtain and analyze the logs.

9. **MRS has an open ecosystem and supports seamless interconnection with peripheral services, allowing you to quickly build a unified big data platform.**
Based on MRS, a full-stack big data service, enterprises can build a unified big data platform for data access, data storage, data analysis, and value mining, and interconnect with DAYU and data visualization services to help customers easily resolve difficulties in data channel cloudification, big data job development and scheduling, and data display. Thereby, customers are free from complex big data platform construction and professional big data optimization and maintenance, and they can focus on industry applications and use one piece of data in multiple service scenarios. DAYU is a one-stop operation platform for entire data lifecycle management. It provides data integration, data development, data governance, data service, and data visualization functions. MRS data can be connected to DAYU, which greatly reduces the threshold for using big data. Based on the visualized development GUI, diverse data development types (script development and job development), fully hosted job scheduling and O&M monitoring capabilities, and built-in industry data processing pipelines, it helps you quickly build big data processing centers with visualized development processes and online collaborative operations. In this way, data can be well-managed and scheduled for real profits quicker.

MRS is fully compatible with the open source big data ecosystem. With abundant data and application migration tools, MRS helps you quickly migrate data from your own platforms without code modification and service interruption.
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

Large-scale data analysis is a major scenario in modern big data systems. Generally, an enterprise has multiple data sources. After data is accessed, ETL processing is required to generate modelized data for each service module to analyze and sort out data. This type of service has the following characteristics:
  - The requirements for real-time execution are not high, and job execution time ranges from dozens of minutes to hours.
  - The data volume is large.
  - There are various data sources and diversified formats.
  - Data processing usually consists of multiple tasks, and resources need to be planned in detail.

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 3-1** 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**

A user who has a large amount of structured data usually requires index-based quasi-real-time query capabilities. For example, in an Internet of Vehicles (IoV) scenario, vehicle maintenance information is queried by vehicle number. Therefore, vehicle information is indexed based on vehicle numbers when it is being stored, to implement second-level response in this scenario. Generally, the data volume is large. The user may store data for one to three years.

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

![Figure 3-2 Large-scale data storage in the IoV industry](image)

MRS has the following advantages in this scenario.
- Real time: Kafka accesses massive amounts 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.

**Real-time data processing**

Real-time data processing is usually used in scenarios such as anomaly detection, fraud detection, rule-based alarming, and service process monitoring. Data is processed while it is being inputted to the system.

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.
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.
### 4.1 List of MRS Component Versions

Table 4-1 lists component versions of MRS clusters of each version.

<table>
<thead>
<tr>
<th>Components Supported by MRS</th>
<th>MRS 1.8.10 (Applicable to MRS 1.8.x)</th>
<th>MRS 1.9.2 (Applicable to MRS 1.9.x)</th>
<th>MRS 2.1.x (Applicable to MRS 2.1.x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Presto</td>
<td>0.215</td>
<td>0.216</td>
<td>308</td>
</tr>
<tr>
<td>Hadoop</td>
<td>2.8.3</td>
<td>2.8.3</td>
<td>3.1.1</td>
</tr>
<tr>
<td>Spark</td>
<td>2.2.1</td>
<td>2.2.2</td>
<td>2.3.2</td>
</tr>
<tr>
<td>HBase</td>
<td>1.3.1</td>
<td>1.3.1</td>
<td>2.1.1</td>
</tr>
<tr>
<td>OpenTSDB</td>
<td>2.3.0</td>
<td>2.3.0</td>
<td>-</td>
</tr>
<tr>
<td>Hive</td>
<td>1.2.1</td>
<td>2.3.3</td>
<td>3.1.0</td>
</tr>
<tr>
<td>Hue</td>
<td>3.11.0</td>
<td>3.11.0</td>
<td>3.11.0</td>
</tr>
<tr>
<td>Loader</td>
<td>2.0.0</td>
<td>2.0.0</td>
<td>2.0.0</td>
</tr>
<tr>
<td>Tez</td>
<td>-</td>
<td>0.9.1</td>
<td>0.9.1</td>
</tr>
<tr>
<td>Flink</td>
<td>1.7.0</td>
<td>1.7.0</td>
<td>1.7.0</td>
</tr>
<tr>
<td>Alluxio</td>
<td>-</td>
<td>2.0.1</td>
<td>-</td>
</tr>
<tr>
<td>Ranger</td>
<td>-</td>
<td>1.0.1</td>
<td>-</td>
</tr>
<tr>
<td>Impala</td>
<td>-</td>
<td>-</td>
<td>3.2.0</td>
</tr>
<tr>
<td>Kudu</td>
<td>-</td>
<td>-</td>
<td>1.9.0</td>
</tr>
<tr>
<td>Kafka</td>
<td>1.1.0</td>
<td>1.1.0</td>
<td>1.1.0</td>
</tr>
</tbody>
</table>
### 4.2 HDFS: Hadoop Distributed File System

The Hadoop Distributed File System (HDFS) implements reliable and distributed read/write of massive amounts of data. HDFS is applicable to the scenario where data read/write features "write once and read multiple times". However, the write operation is performed in sequence, that is, it is a write operation performed during file creation or an add operation performed behind the existing file. HDFS ensures that only one caller can perform write operation on a file but multiple callers can perform read operation on the file at the same time.

The HDFS component of MRS supports the following features:

- Supports erasure code, reducing data redundancy to 50% and improving reliability. In addition, the striped block storage structure is introduced to maximize the use of the capability of a single node and multiple disks in an existing cluster. After the coding process is introduced, the data write performance is improved, and the performance is close to that with the multi-copy redundancy.
• Supports balanced node scheduling on HDFS and balanced disk scheduling on a single node, improving HDFS storage performance after node or disk scale-out.


4.3 YARN: Unified Resource Management and Scheduling Framework

The Apache open source community introduces the unified resource management framework YARN to share Hadoop clusters, improve their scalability and reliability, and eliminate a performance bottleneck of JobTracker in the early MapReduce framework.

The fundamental idea of YARN is to split up the two major functionalities of the JobTracker, resource management and job scheduling/monitoring, into separate daemons. The idea is to have a global ResourceManager (RM) and per-application ApplicationMaster (AM).

NOTE

An application is either a single job in the classical sense of Map-Reduce jobs or a Directed Acyclic Graph (DAG) of jobs.

The essence of the layered structure of YARN is the ResourceManager. This entity controls an entire cluster and manages the allocation of applications to underlying compute resources. The ResourceManager carefully allocates various resources (compute, memory, bandwidth, and so on) to underlying NodeManagers (YARN's per-node agents). The ResourceManager also works with ApplicationMasters to allocate resources, and works with the NodeManagers to start and monitor their underlying applications. In this context, the ApplicationMaster has taken some of the role of the prior TaskTracker, and the ResourceManager has taken the role of the JobTracker.

ApplicationMaster manages each instance of an application running in YARN. The ApplicationMaster negotiates resources from the ResourceManager and works with the NodeManagers to monitor container execution and resource usage (CPU and memory resource allocation).

The NodeManager manages each node in a YARN cluster. The NodeManager provides per-node services in a cluster, from overseeing the management of a container over its lifecycle to monitoring resources and tracking the health of its nodes. MRv1 manages execution of the Map and Reduce tasks through slots, whereas the NodeManager manages abstract containers, which represent per-node resources available for a particular application.
4.4 MapReduce: Distributed Batch Processing Engine

MapReduce is the core of Hadoop. As a software architecture proposed by Google, MapReduce is used for parallel computing of large-scale datasets (larger than 1 TB). The concepts "Map" and "Reduce" and their main thoughts are borrowed from functional programming language and also borrowed from the features of vector programming language.

Current software implementation is as follows: Specify a Map function to map a series of key-value pairs into a new series of key-value pairs, and specify a Reduce function to ensure that all values in the mapped key-value pairs share the same key.
MapReduce is a software framework for processing large datasets in parallel. The root of MapReduce is the Map and Reduce functions in functional programming. The Map function accepts a group of data and transforms it into a key-value pair list. Each element in the input domain corresponds to a key-value pair. The Reduce function accepts the list generated by the Map function, and then shrinks the key-value pair list based on the keys. MapReduce divides a task into multiple parts and allocates them to different devices for processing. In this way, the task can be finished in a distributed environment instead of a single powerful server.

4.5 Hive: Data Warehouse Infrastructure

Hive is a data 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 HiveQL. It allows a user familiar with SQL to query data.

The new execution engine Tez is used to replace the original MapReduce, greatly improving performance. Tez can convert multiple dependent jobs into one job, so only once HDFS write is required and fewer transit nodes are needed, greatly improving the performance of DAG jobs.

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, you need to 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. MRS can access Hive only in client mode. For details, see Using Hive from Scratch. For details about how to develop Hive applications, see Hive Application Development.

- **Metadata storage:** Hive stores metadata into databases, for example, MySQL and Derby. Metadata in Hive includes a table name, table columns and
partitions and their properties, table properties (indicating whether a table is an external table), and the directory where table data is stored.

### 4.6 HBase: Distributed Database

HBase undertakes data storage. HBase is an open source, column-oriented, distributed storage system that is suitable for storing massive amounts of unstructured or semi-structured data. It features high reliability, high performance, and flexible scalability, and supports real-time data read/write.

![HBase diagram](image)

**Figure 4-4** Distributed database HBase

Typical features of a table stored in HBase are as follows:

- **Big table (BigTable):** One table contains hundred millions of lines and millions of columns.
- **Column-oriented:** Column-oriented storage, retrieval, and permission control
- **Sparse:** Null columns in the table do not occupy any storage space.

The HBase component of MRS separates computing from storage. Data can be stored in cloud storage services at low cost, for example, Object Storage Service (OBS), and can be backed up across AZs. MRS supports secondary indexes for HBase and allows adding indexes for column values to filter data by column through native HBase APIs.

For details about HBase architecture and principles, see [https://hbase.apache.org/book.html](https://hbase.apache.org/book.html).
4.7 Spark: Distributed In-Memory Computing Engine

Spark is an open source parallel data processing framework. It helps users to easily develop unified big data applications and perform cooperative processing, stream processing, and interactive analysis on data.

Spark provides a framework featuring fast computing, write, and interactive query. Spark has obvious advantages over Hadoop in terms of performance. Spark uses the in-memory computing mode to avoid I/O bottlenecks in scenarios where multiple tasks in a MapReduce workflow process the same dataset. Spark is implemented by using Scala programming language. Scala enables distributed datasets to be processed in a method the same as that of processing local data. In addition to interactive data analysis, Spark supports interactive data mining. Spark adopts in-memory computing, which facilitates iterative computing. By coincidence, iterative computing of the same data is a general problem facing data mining. In addition, Spark can run in Yarn clusters where Hadoop 2.0 is installed. The reason why Spark cannot only retain various features like MapReduce fault tolerance, data localization, and scalability but also ensure high performance and avoid busy disk I/Os is that a memory abstraction structure called Resilient Distributed Dataset (RDD) is created for Spark.

Original distributed memory abstraction, for example, key-value store and databases, supports small-granularity update of variable status. This requires backup of data or log updates to ensure fault tolerance. Consequently, a large amount of I/O consumption is brought about to data-intensive workflows. For the RDD, it has only one set of restricted APIs and only supports large-granularity update, for example, map and join. In this way, Spark only needs to record the transformation operation log generated during data establishment to ensure fault tolerance without recording a complete dataset. This data transformation link record is a source for tracing data set. Generally, parallel applications apply the same computing process for a large dataset. Therefore, the limit to the aforementioned large-granularity update is not large. In fact, as described in Spark theses, the RDD can function as multiple different computing frameworks, for example, programming models of MapReduce and Pregel. In addition, Spark allows a user to explicitly make a data transformation process be persistent on hard disks. Data localization is implemented by allowing a user to control data partitions based on the key value of each record. (An obvious advantage of this method is that two copies of data to be associated will be hashed in the same mode.) If memory usage exceeds the physical limit, Spark writes relatively large partitions into hard disks, thereby ensuring scalability.

Spark has the following features:

- **Fast**: The data processing speed of Spark is 10 to 100 times higher than that of MapReduce.
- **Easy-to-use**: Java, Scala, and Python can be used to simply and quickly compile parallel applications for processing massive amounts of data. Spark provides over 80 operators to help users compile parallel applications.
- **Universal**: Spark provides many tools, for example, Spark SQL, MLlib, GraphX, and Spark Streaming. These tools can be flexibly combined within one application.
Integration with Hadoop: Spark can directly run in a Hadoop cluster and read existing Hadoop data.

The Spark component of MRS has the following advantages:

- The SparkStreaming component of MRS supports real-time data processing rather than triggering as scheduled.
- The Spark component of MRS provides Structured Streaming and allows users to build streaming applications using the Dataset API. Spark supports exactly-once semantics and inner and outer JOIN operations for streams.
- The Spark component of MRS uses `pandas_udf` to replace the original user-defined functions (UDFs) in PySpark to process data, which reduces the processing duration by 60% to 90% (affected by specific operations).
- The Spark component of MRS also supports graph data processing and allows modeling using graphs during graph computing.
- Spark SQL of MRS is compatible with some Hive syntax (based on the 64 SQL statements of the Hive-Test-benchmark test set) and standard SQL syntax (based on the 99 SQL statements of the tpc-ds test set).

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

### 4.8 CarbonData

CarbonData is a new Apache Hadoop native data-store format. CarbonData allows faster interactive queries over PetaBytes of data using advanced columnar storage, index, compression, and encoding techniques to improve computing efficiency. In addition, CarbonData is also a high-performance analysis engine that integrates data sources with Spark.
The purpose of using CarbonData is to provide quick response to ad hoc queries of big data. Essentially, CarbonData is an Online Analytical Processing (OLAP) engine, which stores data using tables similar to those in Relational Database Management System (RDBMS). You can import more than 10 TB data to tables created in CarbonData format, and CarbonData automatically organizes and stores data using the compressed multi-dimensional indexes. After data is loaded to CarbonData, CarbonData responds to ad hoc queries in seconds.

CarbonData integrates data sources into the Spark ecosystem. You can use Spark SQL to query and analyze data, or use the third-party tool ThriftServer provided by Spark to connect to Spark SQL.

**CarbonData features**

- **SQL**: CarbonData is compatible with Spark SQL and supports SQL query operations performed on Spark SQL.
- **Simple Table dataset definition**: CarbonData allows you to define and create datasets by using user-friendly Data Definition Language (DDL) statements. CarbonData DDL is flexible and easy to use, and can define complex tables.
- **Easy 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.
- **CarbonData file format**: CarbonData file format is a columnar store in HDFS. It has many features that a modern columnar format has, such as splittable and compression schema.

**Unique features of CarbonData**
Stores data along with index: Significantly accelerates query performance and reduces the I/O scans and CPU resources, when there are filters in the query. CarbonData index consists of multiple levels of indices. A processing framework can leverage this index to reduce the task it needs to schedule and process, and it can also do skip scan in more finer grain unit (called blocklet) in task side scanning instead of scanning the whole file.

Operable encoded data: Through supporting efficient compression and global encoding schemes, CarbonData can query on compressed/encoded data. The data can be converted just before returning the results to the users, which is "late materialized".

Supports various use cases with one single data format: like interactive OLAP-style query, Sequential Access (big scan), and Random Access (narrow scan).

**Key technologies and advantages of CarbonData**

- Quick query response: CarbonData features high-performance query. The query speed of CarbonData is 10 times of that of Spark SQL. It uses dedicated data formats and applies multiple index technologies, global dictionary code, and multiple push-down optimizations, providing quick response to TB-level data queries.
- Efficient data compression: CarbonData compresses data by combining the lightweight and heavyweight compression algorithms. This significantly saves 60% to 80% data storage space and the hardware storage cost.

For details about CarbonData architecture and principles, see [https://carbondata.apache.org/](https://carbondata.apache.org/).

### 4.9 Loader: Batch Data Integration

Loader is developed based on the open source Sqoop component. It is used to exchange data and files between MRS and relational databases and file systems. Loader can import data from relational databases or file servers to the HDFS and HBase components of MRS, or export data from HDFS and HBase to relational databases or file servers.

A Loader model consists of a Loader Client and a Loader Server, as shown in [Figure 4-6](#).
Table 4-2 describes the functions of each module shown in the preceding figure.

Table 4-2 Components of the Loader model

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loader Client</td>
<td>Loader client. It provides two interfaces: web UI and CLI.</td>
</tr>
<tr>
<td>Loader Server</td>
<td>Loader server. It processes operation requests sent from the client, manages connectors and metadata, submits MapReduce jobs, and monitors MapReduce job status.</td>
</tr>
<tr>
<td>REST API</td>
<td>It provides a Representational State Transfer (RESTful) APIs (HTTP + JSON) to process the operation requests sent from the client.</td>
</tr>
<tr>
<td>Job Scheduler</td>
<td>Simple job scheduler. It periodically executes Loader jobs.</td>
</tr>
<tr>
<td>Transform Engine</td>
<td>Data transformation engine. It supports field combination, string cutting, and string reverse.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Execution Engine</td>
<td>Loader job execution engine. It executes Loader jobs in MapReduce manner.</td>
</tr>
<tr>
<td>Submission Engine</td>
<td>Loader job submission engine. It submits Loader jobs to MapReduce.</td>
</tr>
<tr>
<td>Job Manager</td>
<td>It manages Loader jobs, including creating, querying, updating, deleting, activating, deactivating, starting, and stopping jobs.</td>
</tr>
<tr>
<td>Metadata Repository</td>
<td>Metadata repository. It stores and manages data about Loader connectors, transformation procedures, and jobs.</td>
</tr>
<tr>
<td>HA Manager</td>
<td>It manages the active/standby status of Loader Server processes. The Loader Server has two nodes that are deployed in active/standby mode.</td>
</tr>
</tbody>
</table>

Loader imports or exports jobs in parallel using MapReduce jobs. Some job import or export may involve only the Map operations, while some may involve both Map and Reduce operations.

Loader implements fault tolerance using MapReduce. Jobs can be rescheduled upon a job execution failure.

- **Importing data to HBase**
  When the Map operation is performed for MapReduce jobs, Loader obtains data from an external data source.
  When a Reduce operation is performed for a MapReduce job, Loader enables the same number of Reduce tasks based on the number of Regions. The Reduce tasks receive data from Map tasks, generate HFiles by Region, and stores the HFiles in a temporary directory of HDFS.
  When a MapReduce job is submitted, Loader migrates HFiles from the temporary directory to the HBase directory.

- **Importing data to HDFS**
  When a Map operation is performed for a MapReduce job, Loader obtains data from an external data source and exports the data to a temporary directory (named `export directory-ldtmp`).
  When a MapReduce job is submitted, Loader migrates data from the temporary directory to the export directory.

- **Exporting data to a relational database**
  When a Map operation is performed for a MapReduce job, Loader obtains data from HDFS or HBase and inserts the data to a temporary table (Staging Table) through the Java DataBase Connectivity (JDBC) API.
When a MapReduce job is submitted, Loader migrates data from the
temporary table to a formal table.

- **Exporting data to a file system**
  When a Map operation is performed for a MapReduce job, Loader obtains
data from HDFS or HBase and writes the data to a temporary directory of the
file server.

  When a MapReduce job is submitted, Loader migrates data from the
temporary directory to a formal directory.

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

### 4.10 Hue: Data Integration Access

Hue is a group of web applications that interact with MRS big data components. It helps you browse HDFS, perform Hive query, and start MapReduce jobs. Hue bears applications that interact with all MRS big data components.

Hue involves the following components and functions:

- **File browser**
  This application allows you to directly browse and operate different HDFS
directories on the GUI. It has the following functions:
  - Creating a file or directory, uploading and downloading a file, renaming,
migrating, deleting a file or directory Modifying the owner of a file or
directory and permissions
  - Searching a file, a directory, a file owner, or a user group to which a user
  belongs
  - Viewing and editing a file

- **Query editor**
  You can use the query editor to compile simple SQL and query data stored on
Hadoop, for example, HDFS, HBase, and Hive. With the query editor, you can
easily create, manage, and execute SQL and download the execution result as
an Excel file. It has the following functions:
  - Editing and executing SQL, storing an SQL template, and copying and
  editing a template SQL explanation, query, and history recording
  - Database presentation and data table presentation
  - Supporting different types of Hadoop storage

For details about Hue, visit [http://gethue.com/](http://gethue.com/).

### 4.11 Presto: Interactive SQL Query Engine

Presto is an open source 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. A Presto query
can combine different data sources to perform data analysis across the data sources.

**Figure 4-7** Presto architecture

Presto runs in a cluster in distributed mode and contains one coordinator and multiple worker processes. Query requests are submitted from clients (for example, CLI) to the coordinator. The coordinator parses SQL statements, generates execution plans, and distributes the plans to multiple worker processes for execution.

For details about Presto, visit [https://prestodb.github.io/](https://prestodb.github.io/).

**Multiple Presto Instances**

MRS supports the installation of multiple Presto instances for a large-scale cluster by default. That is, multiple Worker instances, such as Worker1, Worker2, and Worker3, are installed on a Core/Task node. Multiple Worker instances interact with the Coordinator to execute computing tasks, greatly improving node resource utilization and computing efficiency.

Presto multi-instance applies only to the Arm architecture. Currently, a single node supports a maximum of four instances.

**4.12 Tez: High-speed Computing Engine Supporting DAG Models**

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/](https://tez.apache.org/).

### 4.13 Flume: Real-Time Data Ingestion

Flume is a distributed, reliable, and available system for efficiently collecting, aggregating and moving large amounts of log data. Flume supports customization of various data senders in the log system for data collection. In addition, Flume can roughly process data and write data to various data receivers (customizable).

A Flume-NG is a branch of Flume. It is simple, small, and easy to deploy. The following figure shows the basic architecture of the Flume-NG.

**Figure 4-8 Flume-NG architecture**

![Flume-NG architecture diagram](image)

A Flume-NG consists of agents. Each agent consists of three components (source, channel, and sink). A source is used for receiving data. A channel is used for transmitting data. A sink is used for sending data to the next end.

- **Source**: collects log data, divides data into transactions and events, and imports them into a channel.
- **Channel**: provides a queue to implement simple cache of the data provided by the source.
- **Sink**: obtains data in the channel and imports the data to a storage file system or database, or submits the data to a remote server.

The reliability of Flume depends on transaction switchovers between agents. If the next agent breaks down, the channel stores data persistently and transmits data until the agent recovers. The availability of Flume depends on the built-in load balancing and failover mechanisms. Both the channel and agent can be configured with multiple entities between which they can use load balancing policies. Each agent is a Java Virtual Machine (JVM) process. A server can have multiple agents. Collection nodes (for example, agent1, 2, 3) process logs. Aggregation nodes (for example, agent4) write the logs into HDFS. The agent of each collection node can select multiple aggregation nodes for load balancing.
For details about the Flume architecture and principles, see https://flume.apache.org/releases/1.6.0.html.

4.14 Storm: Stream Computing Engine

Apache Storm is a distributed, reliable, and fault-tolerant real-time stream data processing system. In Storm, a graph-shaped data structure called topology needs to be designed first for real-time computing. The topology will be submitted to a cluster. Then a master node in the cluster distributes codes and assigns tasks to worker nodes. A topology contains two roles: spout and bolt. A spout sends messages and sends data streams in tuples. A bolt converts the data streams and performs computing and filtering operations. The bolt can randomly send data to other bolts. Tuples sent by a spout are unchangeable arrays and map to fixed key-value pairs.
Figure 4-10 System architecture of Storm

Service processing logic is encapsulated in the topology of Storm. A topology is a set of spout (data sources) and bolt (logical processing) components that are connected using Stream Groupings in DAG mode. All components (spout and bolt) in a topology are working in parallel. In a topology, you can specify the parallelism for each node. Then, Storm allocates tasks in the cluster for computing to improve system processing capabilities.

Figure 4-11 Topology

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

4.15 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.

**Figure 4-12** shows the technology stack of Flink.

**Figure 4-12 Technology stack of Flink**

As shown in the above figure, 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 point of failure (SPOF)

- **Scale-out**
  Manual scale-out supported by TaskManagers

For details about Flink, visit [https://flink.apache.org/](https://flink.apache.org/).
4.16 Kafka: Distributed Message Queue

Kafka is an open source, distributed, partitioned, and replicated commit log service. Kafka is publish-subscribe messaging, rethought as a distributed commit log. It provides features similar to Java Message Service (JMS) but another design. It features message endurance, high throughput, distributed methods, multi-client support, and real time. It applies to both online and offline message consumption, such as regular message collection, website activeness tracking, aggregation of statistical system operation data (monitoring data), and log collection. These scenarios engage large amounts of data collection for Internet services.

Figure 4-13 Kafka architecture

Table 4-3 Kafka components

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Broker</td>
<td>A broker is a server in a Kafka cluster.</td>
</tr>
<tr>
<td>Topic</td>
<td>A topic is a category or feed name to which messages are published. A topic can be divided into multiple partitions, which can act as a parallel unit.</td>
</tr>
<tr>
<td>Partition</td>
<td>A partition is an ordered, immutable sequence of messages that is continually appended to - a commit log. The messages in the partitions are each assigned a sequential ID number called the offset that uniquely identifies each message within the partition.</td>
</tr>
<tr>
<td>Producer</td>
<td>Producers publish messages to a Kafka topic.</td>
</tr>
<tr>
<td>Consumer</td>
<td>Consumers subscribe to topics and process the feed of published messages.</td>
</tr>
</tbody>
</table>

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

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

KafkaManager supports the following operations:

- 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.

4.18 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.

OpenTSDB consists of a Time Series Daemon (TSD) as well as a set of command line utilities. Interaction with OpenTSDB is primarily implemented by running one or more TSDs. Each TSD is independent. There is no master server and no shared state, so you can run as many TSDs as required to handle any load you throw at it. Each TSD uses HBase in a CloudTable cluster to store and retrieve time series data. The data schema is highly optimized for fast aggregations of similar time series to minimize storage space. TSD users never need to directly access the underlying storage. You can communicate with the TSD through an HTTP API. All communications happen on the same port (the TSD figures out the protocol of the client by looking at the first few bytes it receives).
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 [https://opentsdb.net/docs/build/html/api_http/index.html](https://opentsdb.net/docs/build/html/api_http/index.html).

### 4.19 Impala

Impala provides fast, interactive SQL queries directly on your Apache Hadoop data stored in HDFS, HBase, or the Object Storage Service (OBS). In addition to using the same unified storage platform, Impala also uses the same metadata, SQL syntax (Hive SQL), ODBC driver, and user interface (Impala query UI in Hue) as Apache Hive. This provides a familiar and unified platform for real-time or batch-oriented queries. Impala is an addition to tools available for querying big data. Impala does not replace the batch processing frameworks built on MapReduce such as Hive. Hive and other frameworks built on MapReduce are best suited for long running batch jobs.

Impala provides the following features:
Most common SQL-92 features of Hive Query Language (HiveQL) including SELECT, JOIN, and aggregate functions

HDFS, HBase, and OBS storage, including:
- HDFS file formats: delimited text files, Parquet, Avro, SequenceFile, and RCFile
- Compression codecs: Snappy, GZIP, Deflate, BZIP

Common data access interfaces including:
- JDBC driver
- ODBC driver
- Hue Beeswax and the Impala query UI

impala-shell command line interface

Kerberos authentication

Impala applies to offline analysis (such as log and cluster status analysis) of real-time data queries, large-scale data mining (such as user behavior analysis, interest region analysis, and region display), and other scenarios.

4.20 Kudu

Kudu is a columnar storage manager developed for the Apache Hadoop platform. Kudu shares the common technical properties of Hadoop ecosystem applications: it runs on commodity hardware, is horizontally scalable, and supports highly available operation.

Kudu's design has the following benefits:

- Fast processing of OLAP workloads
- Integration with MapReduce, Spark and other Hadoop ecosystem components
- Tight integration with Apache Impala, making it a good, mutable alternative to using HDFS with Apache Parquet
- Strong but flexible consistency model, allowing you to choose consistency requirements on a per-request basis, including the option for strict-serializable consistency
- Strong performance for running sequential and random workloads simultaneously
- Easy to manage
- High availability Tablet Servers and Masters use the Raft Consensus Algorithm, which ensures that as long as more than half the total number of replicas is available, the tablet is available for reads and writes. For example, if 2 out of 3 replicas or 3 out of 5 replicas are available, the tablet is available. Reads can be serviced by read-only follower tablets, even in the event of a leader tablet failure.
- Structured data model

By combining all of these properties, Kudu targets support for families of applications that are difficult or impossible to implement on current generation Hadoop storage technologies.

A few examples of applications for which Kudu is a great solution are:
- Reporting applications where newly-arrived data needs to be immediately available for end users
- Time-series applications that must simultaneously support queries across large amounts of historic data and granular queries about an individual entity that must return very quickly
- Applications that use predictive models to make real-time decisions with periodic refreshes of the predictive model based on all historic data

### 4.21 Alluxio

Alluxio is data orchestration technology for analytics and AI for the cloud. In the MRS big data ecosystem, Alluxio lies between computing and storage. It provides a data abstraction layer for computing frameworks including Apache Spark, Presto, MapReduce, and Apache Hive, so that upper-layer computing applications can access persistent storage systems including HDFS and OBS through unified client APIs and a global namespace. In this way, computing and storage are separated.

![Alluxio architecture](image)

**Figure 4-15** Alluxio architecture

Advantages:
- Provides in-memory I/O throughput, and makes elastically scale data-driven applications cost effective.
- Simplified cloud and object storage access
- Simplified data management and a single point of access to multiple data sources
- Easy application deployment


### 4.22 Ranger

Apache Ranger offers a centralized security framework and supports fine grained authorization and auditing. It manages fine grained access control over Hadoop and related components, such as HDFS, Hive, HBase, Kafka, and Storm. You can use the front-end web UI console provided by Ranger to configure policies to control users' access to these components.
Currently, Ranger does not support Kerberos authentication.

The entire Ranger system consists of three parts:

- Ranger Admin
  Provides a web UI for users to manage the entire Ranger.
- Ranger UserSync
  Synchronizes Unix OS users or LADP users to Ranger for unified management.
- Ranger Plugin
  Provides plug-ins for each component to control the access permissions of each component.
5 Functions

5.1 New Functions of MRS 2.0

In MRS 2.0, the Hadoop, Hive, Spark, and HBase components are upgraded and Tez is supported. Table 5-1 provides details about the new functions.

Table 5-1 New functions of MRS 2.0

<table>
<thead>
<tr>
<th>Component Version</th>
<th>New Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hadoop 3.1.1</td>
<td>Erasure coding (EC)</td>
<td>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 code 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.</td>
</tr>
<tr>
<td></td>
<td>DataNode multi-disk balancer</td>
<td>This function is used to solve the problem of unbalanced data storage among multiple disks in the DataNode when disks are added or replaced.</td>
</tr>
<tr>
<td></td>
<td>Opportunistic containers</td>
<td>This function improves the cluster resource usage and increases task throughput.</td>
</tr>
<tr>
<td>Component Version</td>
<td>New Function</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>User-defined resource model</td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Hive 3.1.0</td>
<td>Hive web UI</td>
<td>Hive web UI makes O&amp;M easier. HiveServer provides a web UI for O&amp;M personnel to view the running SQL statements and how long an SQL statement has been executed.</td>
</tr>
<tr>
<td>HPL/SQL</td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Cost-based optimizer (CBO)</td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Tez 0.9.1</td>
<td>New execution engine: Tez</td>
<td>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.</td>
</tr>
<tr>
<td>Spark 2.3.2</td>
<td>Continuous Processing</td>
<td>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).</td>
</tr>
<tr>
<td>Component Version</td>
<td>New Function</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Stream-Stream join</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>PySpark performance optimization</td>
<td>Processing duration is reduced by 60% to 90%. The <code>pandas_udf</code> 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 <code>pandas_udf</code> 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).</td>
<td></td>
</tr>
<tr>
<td>MLlib optimization</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>HBase 2.1.1</td>
<td>AssignmentManager V2</td>
<td>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.</td>
</tr>
<tr>
<td>Component Version</td>
<td>New Function</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>In-Memory Compaction</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Offheaping of Read/Write</td>
<td>The read/write performance is optimized. HBase 2.x changes the data read/write mode and directly reads and writes data in the &quot;L2&quot; 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.</td>
</tr>
<tr>
<td></td>
<td>NettyBaseRpc framework</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>RegionServer Group</td>
<td>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 RegionServers.</td>
</tr>
<tr>
<td></td>
<td>HBase on OBS</td>
<td>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.</td>
</tr>
</tbody>
</table>
5.2 Multi-tenant

Feature Introduction

Modern enterprises' data clusters are developing towards centralization and cloudification. Enterprise-class big data clusters must meet the following requirements:

- Carry data of different types and formats and run jobs and applications of different types (analysis, query, and stream processing).
- Isolate data of a user from that of another user who has demanding requirements on data security, such as a bank or government institute.

The preceding requirements bring the following challenges to the big data cluster:

- Proper allocation and scheduling of resources to ensure stable operating of applications and jobs
- Strict access control to ensure data and service security

Multi-tenant isolates the resources of a big data cluster into resource sets. Users can lease desired resource sets to run applications and jobs and store data. In a big data cluster, multiple resource sets can be deployed to meet diverse requirements of multiple users.

The MRS big data cluster provides a complete enterprise-class big data multi-tenant solution. Multi-tenant is a collection of multiple resources (each resource set is a tenant) in an MRS big data cluster. It can allocate and schedule resources, including computing and storage resources.

Advantages

- Proper resource configuration and isolation
  The resources of a tenant are isolated from those of another tenant. The resource use of a tenant does not affect other tenants. This mechanism ensures that each tenant can configure resources based on service requirements, improving resource utilization.

- Resource consumption measurement and statistics
  Tenants are system resource applicants and consumers. System resources are planned and allocated based on tenants. Resource consumption by tenants can be measured and recorded.

- Ensured data security and access security
  In multi-tenant scenarios, the data of each tenant is stored separately to ensure data security. The access to tenants' resources is controlled to ensure access security.

Enhanced Schedulers

Schedulers are divided into the open source Capacity scheduler and Huawei proprietary Superior scheduler.

To meet enterprise requirements and tackle challenges facing the Yarn community in scheduling, Huawei develops the Superior scheduler. In addition to inheriting
the advantages of the Capacity scheduler and Fair scheduler, this scheduler is enhanced in the following aspects:

- **Enhanced resource sharing policy**
  The Superior scheduler supports queue hierarchy. It integrates the functions of open source schedulers and shares resources based on configurable policies. In terms of instances, administrators can use the Superior scheduler to configure an absolute value or percentage policy for queue resources. The resource sharing policy of the Superior scheduler enhances the label scheduling policy of Yarn as a resource pool feature. The nodes in the Yarn cluster can be grouped based on the capacity or service type to ensure that queues can more efficiently utilize resources.

- **Tenant-based resource reservation policy**
  Resources required by tenants must be ensured for running critical tasks. The Superior scheduler builds a mechanism to support the resource reservation policy. By doing so, reserved resources can be allocated to the tasks run by the tenant queues in a timely manner to ensure proper task execution.

- **Fair sharing among tenants and resource pool users**
  The Superior scheduler allows shared resources to be configured for users in a queue. Each tenant may have users with different weights. Heavily weighted users may require more shared resources.

- **Ensured scheduling performance in a big cluster**
  The Superior scheduler receives heartbeats from each NodeManager and saves resource information in memory, which enables the scheduler to control cluster resource usage globally. The Superior scheduler uses the push scheduling model, which makes the scheduling more precise and efficient and remarkably improves cluster resource utilization. Additionally, the Superior scheduler delivers excellent performance when the interval between NodeManager heartbeats is long and prevents heartbeat storms in big clusters.

- **Priority policy**
  If the minimum resource requirement of a service cannot be met after the service obtains all available resources, a preemption occurs. The preemption function is disabled by default.

### 5.3 Security Hardening

MRS is a platform for massive data management and analysis and has high security. MRS protects user data and service running from the following aspects:

- **Network isolation**
  The entire system is deployed in a VPC on the public cloud to provide an isolated network environment and ensure service and management security of the cluster. By combining the subnet division, route control, and security group functions of VPC, MRS provides a secure and reliable isolated network environment.

- **Resource isolation**
  MRS supports resource deployment and isolation of physical resources in dedicated zones. You can flexibly combine computing and storage resources,
such as dedicated computing resources + shared storage resources, shared computing resources + dedicated storage resources, and dedicated computing resources + dedicated storage resources.

- **Host security**
  MRS can be integrated with public cloud security services, including Vulnerability Scan Service (VSS), Host Security Service (HSS), Web Application Firewall (WAF), Cloud Bastion Host (CBH), and Web Tamper Protection (WTP). HUAWEI CLOUD provides the following measures to improve security of the OS and ports:
  - Security hardening of OS kernels
  - OS patch update
  - OS permission control
  - OS port management
  - OS protocol and port attack defense

- **Application security**
  The following measures are used to ensure normal running of big data services:
  - Identification and authentication
  - Web application security
  - Access control
  - Audit security
  - Password security

- **Data security**
  The following measures are provided to ensure the confidentiality, integrity, and availability of massive amounts of user data:
  - Disaster recovery: MRS supports data backup to OBS and cross-region high reliability.
  - Backup: MRS supports backup of DBService, NameNode, and LDAP metadata and backup of HDFS and HBase service data.

- **Data integrity**
  Data is verified to ensure its integrity during storage and transmission.
  - CRC32C is used by default to verify the correctness of user data stored in HDFS.
  - DataNodes of HDFS store the verified data. If the data transmitted from a client is abnormal (incomplete), DataNodes report the abnormality to the client, and the client rewrites the data.
  - The client checks data integrity when reading data from a DataNode. If the data is incomplete, the client will read data from another DataNode.

- **Data confidentiality**
  Based on Apache Hadoop, the distributed file system of MRS supports encrypted storage of files to prevent sensitive data from being stored in plaintext, improving data security. Applications need only to encrypt specified sensitive data. Services are not affected during the encryption process. Based on file system data encryption, Hive provides table-level encryption and HBase provides column family-level encryption. Sensitive data can be
encrypted and stored after you specify an encryption algorithm during table creation.

Encrypted storage and access control of data are used to ensure user data security.

- HBase stores service data to the HDFS after compression. Users can configure the AES and SMS4 encryption algorithm to encrypt data.
- All the components allow access permissions to be set for local data directories. Unauthorized users are not allowed to access data.
- All cluster user information is stored in ciphertext.

- Security authentication
  - Uses a unified user- and role-based authentication system as well as an account- and role-based access control (RBAC) model to centrally control user permissions and batch manage user authorization.
  - Employs Lightweight Directory Access Protocol (LDAP) as an account management system and performs the Kerberos authentication on accounts.
  - Provides the single sign-on (SSO) function that centrally manages and authenticates MRS system and component users.
  - Audits users who have logged in to MRS Manager.

### 5.4 Easy Access to Web UIs of Components

Big data components have their own web UIs to manage their own systems. However, you cannot easily access the web UIs due to network isolation. For example, to access the HDFS web UI, you need to create an ECS to remotely log in to the web UI. This makes the UI access complex and unfriendly.

MRS provides an EIP-based secure channel for you to easily access the web UIs of components. This is more convenient than binding an EIP by yourself, and you can access the web UIs with a few clicks, avoiding the steps for logging in to a VPC, adding security group rules, and obtaining a public IP address. For the Hadoop, Spark, HBase, and Hue components in analysis clusters and the Storm component in streaming clusters, you can quickly access their web UIs from the entries on MRS Manager. In addition, an EIP is used for access in a unified manner, which is easy to use and remember.

### 5.5 Reliability Enhancement

Based on Apache Hadoop open source software, MRS optimizes and improves the reliability and performance of main service components.

#### System Reliability

- HA for all management nodes

  In the Hadoop open source version, data and compute nodes are managed in a distributed system, in which a single point of failure (SPOF) does not affect the operation of the entire system. However, a SPOF may occur on management nodes running in centralized mode, which becomes the weakness of the overall system reliability.
MRS provides similar double-node mechanisms for all management nodes of the service components, such as MRS Manager, HDFS NameNodes, Hive Servers, HBase HMasters, Yarn ResourceManagers, Kerberos Servers, and Ldap Servers. All of them are deployed in active/standby mode or configured with load sharing, effectively preventing SPOFs from affecting system reliability.

- **Reliability guarantee in case of exceptions**
  By reliability analysis, the following measures to handle software and hardware exceptions are provided to improve the system reliability:
  - After power supply is restored, services are running properly regardless of a power failure of a single node or the whole cluster, ensuring data reliability in case of unexpected power failures. Key data will not be lost unless the hard disk is damaged.
  - Health status checks and fault handling of the hard disk do not affect services.
  - The file system faults can be automatically handled, and affected services can be automatically restored.
  - The process and node faults can be automatically handled, and affected services can be automatically restored.
  - The network faults can be automatically handled, and affected services can be automatically restored.

- **Data backup and restoration**
  MRS provides full backup, incremental backup, and restoration functions based on service requirements, preventing the impact of data loss and damages on services and ensuring fast system restoration in case of exceptions.
  - Automatic backup
    MRS provides automatic backup for data on Manager. Based on the customized backup policy, data on clusters, including LDAPServer and DBService data, can be automatically backed up.
  - Manual backup
    You can also manually back up data of the cluster management system before the capacity expansion, upgrade, and patch installation to recover the cluster management system functions upon faults.
    To improve the system reliability, data on Manager and HBase is backed up to a third-party server manually.

**Node Reliability**

- **OS health status monitoring**
  MRS periodically collects OS hardware resource usage data, including usage of CPUs, memory, hard disks, and network resources.

- **Process health status monitoring**
  MRS checks the status of service instances and health indicators of service instance processes, enabling you to know the health status of processes in a timely manner.

- **Automatic disk troubleshooting**
  MRS is enhanced based on the open source version. It can monitor the status of hardware and file systems on all nodes. If an exception occurs, the
corresponding partitions will be removed from the storage pool. If a disk is faulty and replaced, a new hard disk will be added for running services. In this case, maintenance operations are simplified. Replacement of faulty disks can be completed online. In addition, users can set hot backup disks to reduce the faulty disk restoration time and improve the system reliability.

- LVM configuration for node disks
  MRS allows you to configure Logic Volume Management (LVM) to plan multiple disks as a logical volume group. Configuring LVM can avoid uneven usage of disks. It is especially important to ensure even usage of disks on components that can use multiple disk capabilities, such as HDFS and Kafka. In addition, LVM supports disk capacity expansion without re-attaching, preventing service interruption.

Data Reliability

MRS can use the anti-affinity node groups and placement group capabilities provided by ECS and the rack awareness capability of Hadoop to redundantly distribute data to multiple physical host machines, preventing data loss caused by physical hardware failures.

5.6 Job Management

The job management function provides an entry for you to submit jobs in a cluster, including MapReduce, Spark, HiveQL, and SparkSQL jobs. MRS works with HUAWEI CLOUD Data Lake Factory (DLF) to provide a one-stop big data collaboration development environment and fully-managed big data scheduling capabilities, helping you effortlessly build big data processing centers.

DLF allows you to develop and debug MRS HiveQL/SparkSQL scripts online and develop MRS jobs by performing drag-and-drop operations to migrate and integrate data between MRS and more than 20 heterogeneous data sources. Powerful job scheduling and flexible monitoring and alarming help you easily manage data and job O&M.

5.7 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.

### 5.8 Enterprise Project Management

An enterprise project is a cloud resource management mode. Enterprise Management provides users with comprehensive management of cloud-based resources, personnel, permissions, and finances. Common management consoles are oriented to the control and configuration of individual cloud products. The Enterprise Management console, in contrast, is more focused on resource management. It is designed to help enterprises manage cloud-based resources, personnel, permissions, and finances, in a hierarchical management manner, such as management of companies, departments, and projects.

MRS allows users who have enabled Enterprise Project Management Service (EPS) to configure enterprise projects for a cluster during cluster creation and use EPS to manage MRS resources by group.

- The users can manage multiple resources by group.
- The users can view resource information and expenditure details of enterprise projects.
- The users can control access permissions at the enterprise project level.
- The users can view detailed financial information by enterprise project, including orders, expenditure summary, and expenditure details.

### 5.9 Cluster Management

#### 5.9.1 Cluster Lifecycle Management

MRS supports cluster lifecycle management, including creating and terminating clusters.
• Creating a cluster: After you specify a cluster type, components, number of nodes of each type, VM specifications, AZ, VPC, and authentication information, MRS automatically creates a cluster that meets the configuration requirements. You can run customized scripts in the cluster. In addition, you can create clusters of different types for multiple application scenarios, such as Hadoop analysis clusters, HBase clusters, and Kafka clusters. The big data platform supports heterogeneous cluster deployment. That is, VMs of different specifications can be combined in a cluster based on CPU types, disk capacities, disk types, and memory sizes. Various VM specifications can be mixed in a cluster.

• Terminating a cluster: You can terminate a pay-per-use cluster that is no longer needed (including data and configurations in the cluster). MRS will delete all resources related to the cluster.

• Renewal: MRS provides two billing modes: pay-per-use and yearly/monthly. In pay-per-use mode, fees are deducted every hour and insufficient balance can lead to overdue payments. In yearly/monthly mode, clusters need to be renewed before they expire. If your subscription is not renewed, your services will keep running, but enter into a grace period. If no renewal occurs in the grace period, MRS clusters will stop running but data is retained during a retention period.

• Unsubscription: If you have purchased a yearly/monthly cluster and do not need the cluster resources before the cluster resources expire, you can unsubscribe from the cluster resources on MRS.

Purchasing an MRS Cluster

On the MRS management console, you can purchase an MRS cluster on a pay-per-use or yearly/monthly basis. You can select a region and cloud resource specifications to purchase an MRS cluster that is suitable for enterprise services in one click. MRS automatically installs and deploys the HUAWEI CLOUD enterprise-level big data platform and optimizes parameters based on the selected cluster type, version, and node specifications.

MRS provides you with fully managed big data clusters. When creating a cluster, you can set a VM login mode (password or key pair). You can use all resources of the created MRS cluster. In addition, MRS allows you to deploy a big data cluster on only two ECSs with 4 vCPUs and 8 GB memory, providing more flexible choices for testing and development.

MRS clusters are classified into analysis, streaming, and hybrid clusters.

• Analysis cluster: is used for offline data analysis and provides Hadoop components.

• Streaming cluster: is used for streaming tasks and provides stream processing components.

• Hybrid cluster: is used for not only offline data analysis but also streaming processing, and provides Hadoop components and stream processing components.

MRS cluster nodes are classified into Master, Core, and Task nodes.

• Master node: management node in a cluster. Master processes of a distributed system, MRS Manager, and databases are deployed on Master nodes. Master
nodes cannot be scaled out. The processing capability of Master nodes determines the upper limit of the management capability of the entire cluster. MRS supports scale-up of Master node specifications to provide support for management of a larger cluster.

- Core node: used for both storage and computing and can be scaled in or out. Since Core nodes bear data storage, there are many restrictions on scale-in to prevent data loss and auto scaling cannot be performed.
- Task node: used only for computing only and can be scaled in or out. Task nodes bear only computing tasks. Therefore, auto scaling can be performed.

Two modes for purchasing an MRS cluster are supported: custom config and quick config

- Custom config: On the Custom Config page, you can flexibly configure cluster parameters based on application scenarios, such as the billing mode and ECS specifications to better suit your service requirements.
- Quick config: On the Quick Config page, you can quickly purchase a cluster based on application scenarios, improving cluster configuration efficiency. Currently, Hadoop analysis clusters, HBase clusters, and Kafka clusters are available for your quick purchase.
  - Hadoop analysis cluster: uses components in the open source Hadoop ecosystem to analyze and query vast amounts of data. For example, use Yarn to manage cluster resources, Hive and Spark to provide offline storage and computing of large-scale distributed data, Spark Streaming and Flink to offer streaming data computing, and Presto to enable interactive queries, and Tez to provide a distributed computing framework of directed acyclic graphs (DAGs).
  - HBase cluster: uses Hadoop and HBase components to provide a column-oriented distributed cloud storage system featuring enhanced reliability, excellent performance, and elastic scalability. It applies to the storage and distributed computing of massive amounts of data. 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.
  - Kafka cluster: uses Kafka and Storm to provide an open source message system with high throughput and scalability. It is widely used in scenarios such as log collection and monitoring data aggregation to implement efficient streaming data collection and real-time data processing and storage.

**Terminating a Cluster**

MRS allows you to terminate a cluster when it is no longer needed. After the cluster is terminated, all cloud resources used by the cluster will be released. Before terminating a cluster, you are advised to migrate or back up data. Terminate the cluster only when no service is running in the cluster or the cluster is abnormal and cannot provide services based on O&M analysis. If data is stored on EVS disks or pass-through disks in a big data cluster, the data will be deleted after the cluster is terminated. Therefore, exercise caution when terminating a cluster.
5.9.2 Manually Scale Out/In a Cluster

The processing capability of a big data cluster can be horizontally expanded by adding nodes. If the cluster scale does not meet service requirements, you can manually scale out or scale in the cluster. MRS intelligently selects the node with the least load or the minimum amount of data to be migrated for scale-in. The node to be scaled in will not receive new tasks, and continues to execute the existing tasks. At the same time, MRS copies its data to other nodes and the node is decommissioned. If the tasks on the node cannot be completed after a long time, MRS migrates the tasks to other nodes, minimizing the impact on cluster services.

Scaling Out a Cluster

Currently, you can add Core or Task nodes to scale out a cluster to handle peak service loads. After the MRS cluster scale-out, data stored in HDFS can be dynamically distributed to prevent data skew. Data re-distribution will be automatically completed after scale-out or scale-in.

Scaling Out a Cluster 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 scale-out beyond your subscription is required. MRS allows you to scale out 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 scale-out process does not require manual intervention and takes only a few minutes, which helps ease pressure on growing service data processing needs.

Scaling In a Cluster

You can reduce the number of Core or Task nodes to scale in a cluster so that MRS delivers better storage and computing capabilities at lower O&M costs based on service requirements. After you scale in an MRS cluster, MRS automatically selects nodes that can be scaled in based on the type of services installed on the nodes.

5.9.3 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.

In big data applications, especially in periodic data analysis and processing scenarios, cluster computing resources need to be dynamically adjusted based on service data changes to meet service requirements. The auto scaling function of
MRS enables clusters to be elastically scaled out or in based on cluster loads. In addition, if the data volume changes regularly and you want to scale out or in a cluster before the data volume changes, you can use the MRS resource plan feature.

MRS supports two types of auto scaling policies: auto scaling rules and resource plans

- Auto scaling rules: You can increase or decrease Task nodes based on real-time cluster loads. Auto scaling will be triggered when the data volume changes but there may be some delay.
- Resource plans: If the data volume changes periodically, you can create resource plans to resize the cluster before the data volume changes, thereby avoiding a delay in increasing or decreasing resources.

Both auto scaling rules and resource plans can trigger auto scaling. You can configure both of them or configure one of them. Configuring both resource plans and auto scaling rules improves the cluster node scalability to cope with occasionally unexpected data volume peaks.

In some service scenarios, resources need to be reallocated or service logic needs to be modified after cluster scale-out or scale-in. If you manually scale out or scale in a cluster, you can log in to cluster nodes to reallocate resources or modify service logic. If you use auto scaling, MRS enables you to customize automation scripts for resource reallocation and service logic modification. Automation scripts can be executed before and after auto scaling and automatically adapt to service load changes, all of which eliminates manual operations. In addition, automation scripts can be fully customized and executed at various moments, which can meet your personalized requirements and improve auto scaling flexibility.

Customer Benefits

MRS auto scaling provides the following 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 supporting 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 event ends, 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 enable developers to focus on service development without the need for resource estimation.

5.9.4 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 data volume change is small in a cluster but the cluster's 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.

5.9.5 Scaling Up Master Node Specifications

MRS provides MRS Manager for managing clusters and services in the clusters, such as NameNodes of HDFS, ResourceManagers of Yarn, and Manager management services of MRS, are deployed on the Master node of the clusters.

With the rollout of new services, a cluster scale increases continuously, and Master nodes bear more and more loads. Enterprise users are faced with the problem that CPU loads are too high and memory usage exceeds the threshold. Generally, in an on-premises big data cluster, you need to migrate data and purchase hardware with advanced configurations to scale up the Master node specifications. MRS leverages the advantages of cloud services to enable you to scale up Master node specifications in one click. During the scale-up, the active/standby HA mode of the Master nodes ensures that existing services are not interrupted.

For details about how to scale up the Master node specifications, see Scaling Up Master Node Specifications.

5.9.6 Isolating a Host

When detecting that a host is abnormal or faulty and cannot provide services or affects cluster performance, you can exclude the host from the available nodes in the cluster temporarily so that the client can access other available nodes. In scenarios where patches are to be installed in a cluster, you can also exclude a specified node from patch installation. Only non-management nodes can be isolated.
After a host is isolated, all role instances on the host will be stopped, and you cannot start, stop, or configure the host and all instances on the host. In addition, after a host is isolated, statistics about the monitoring status and metric data of hardware and instances on the host cannot be collected or displayed.

5.9.7 Managing Tags

Tags are cluster identifiers. Adding tags to clusters can help you identify and manage your cluster resources. By associating with Tag Management Service (TMS), MRS allows users with a large number of cloud resources to tag cloud resources, quickly search for cloud resources with the same tag attribute, and perform unified management operations such as review, modification, and deletion, facilitating unified management of big data clusters and other cloud 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.

5.10 Cluster O&M

Alarm Management

MRS can monitor big data clusters in real time and identify system health status based on alarms and events. In addition, MRS allows you to customize monitoring and alarm thresholds to focus on the health status of each metric. When monitoring data reaches the alarm threshold, the system triggers an alarm.

MRS can also interconnect with the message service system of the HUAWEI CLOUD Simple Message Notification (SMN) service to push alarm information to users by SMS message or email. For details, see Message Notification.

Patch Management

MRS supports cluster patching operations and will release patches for open source big data components in a timely manner. On the MRS cluster management page, you can view patch release information related to running clusters, including the detailed description of the resolved issues and impacts. You can determine whether to install a patch based on the service running status. One-click patch installation involves no manual intervention, and will not cause service interruption through rolling installation, ensuring long-term availability of the clusters.

MRS can display the detailed patch installation process. Patch management also supports patch uninstallation and rollback.

O&M Support

Cluster resources provided by MRS belong to users. Generally, when O&M personnel's support is required for troubleshooting of a cluster, O&M personnel cannot directly access the cluster. To better serve customers, MRS provides the following two methods to improve communication efficiency during fault locating:
Log sharing: You can initiate log sharing on the MRS management console to share a specified log scope with O&M personnel, so that O&M personnel can locate faults without accessing the cluster.

O&M authorization: If a problem occurs when you use an MRS cluster, you can initiate O&M authorization on the MRS management console. O&M personnel can help you quickly locate the problem, and you can revoke the authorization at any time.

Health Check

MRS provides automatic inspection on system running environments for you to check and audit system running health status in one click, ensuring proper system running and lowering system operation and maintenance costs. After viewing inspection results, you can export reports for archiving and fault analysis.

5.11 Message Notification

Feature Introduction

The following operations are often performed during the running of a big data cluster:

- Big data clusters often change, for example, cluster scale-out and scale-in.
- When a service data volume changes abruptly, auto scaling will be triggered.
- After related services are stopped, a big data cluster needs to be stopped.

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. MRS sends cluster messages to the topic to which you have permission to publish messages. Then, all subscribers who subscribe to the topic can receive cluster updates and component alarms through SMS and emails.
Figure 5-1 Implementation process

Publisher

MRS

Simple Message Notification (SMN)

Topic

Email

Topic

SMS

Topic

HTTP(S)

Subscriber
Before using MRS, ensure that you have read and understand 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. You are advised to reinstall Internet Explorer 9.0 or later as the administrator (recommended) or alternatively run Internet Explorer 9.0 as the administrator. 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.
MapReduce Service (MRS) pricing is simple and predictable. You can select a pay-per-use billing mode or yearly/monthly subscription depending on what is more economical for you. The total price of an MRS cluster will be automatically calculated so that you can purchase a cluster with one click.

**Billing Items**

The price of an MRS cluster consists of two parts:

- MRS management fee
- Fees of IaaS infrastructure resources, including Elastic Cloud Server (ECS), Elastic Volume Service (EVS), elastic IP (EIP), and bandwidth

For details about the MRS management fee, see Product Pricing Details. You can use the price calculator of MRS to quickly obtain an estimate price of a cluster with the specifications you select.

**Billing Modes**

Before using MRS, you must purchase an MRS cluster. Currently, MRS provides two billing modes:

- Yearly/Monthly: You can pay for clusters by year or month. The minimum duration is 1 month and the maximum duration is 1 year.
- Pay-per-use: Nodes are billed by actual duration of use, with a billing cycle of one hour.

**Changing Billing Mode**

Before subscribing to MRS, choose Master and Core node instances that best fit your needs. MRS provides the following methods for you to change cluster configuration after a cluster is started.

- **Configure Task Node**: Add Task nodes. For details, see Related Operations in Manually Scaling Out a Cluster.
- **Scale Out**: Manually add Core or Task nodes. For details, see Manually Scaling Out a Cluster.
- **Auto Scaling**: The number of nodes in a cluster can be automatically adjusted based on the service data volume to increase or decrease resources. For details, see [Using Auto Scaling in a Cluster](#).

If the configuration change methods provided by MRS do not meet your requirements, you can create a cluster again and migrate data to the cluster to realize cluster configuration change.

### Renewal

MRS provides two billing modes: pay-per-use and yearly/monthly. In pay-per-use mode, fees are deducted every hour and insufficient balance can lead to overdue payments. In yearly/monthly mode, clusters need to be renewed before they expire. If your subscription is not renewed, your services will keep running, but enter into a grace period. If no renewal occurs in the grace period, MRS clusters will stop running but data is retained during a retention period.

The duration of the grace period and retention period depends on the customer level. For details, see [Grace Period and Retention Period](#).

To renew the subscription, go to the Renewals page.

### Overdue Payment

- Overdue payment does not apply to yearly or monthly subscribed clusters.
- In pay-per-use mode, cluster fees are deducted every hour. If your account balance is insufficient to pay for the expense occurred in the last hour, your account will be in arrears, and MRS clusters have a grace period and retention period. If the clusters are renewed within the grace period or the retention period, they will be available and charged from the original expiration date.

### Expiration

- Expiration does not apply to pay-per-use clusters.
- If your yearly or monthly subscription expires, the cluster will enter into a grace period and retention period. During the grace period and retention period, you cannot perform operations on the cluster on the MRS management console, related APIs cannot be called, and O&M operations such as automatic monitoring and alarm reporting will be stopped. If your subscription is not renewed at the end of the retention period, services in the cluster will be terminated and data in the system will be deleted permanently.
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 the IAM Service Overview.

MRS Permissions

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 or roles 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.

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

You can grant users permissions by using roles and policies.

- Roles: A type of coarse-grained authorization mechanism that defines permissions related to user responsibilities. This mechanism provides only a limited number of service-level roles for authorization. When using roles to grant permissions, you need to also assign other roles on which the
permissions depend to take effect. However, roles are not an ideal choice for fine-grained authorization and secure access control.

- Policies: A type of fine-grained authorization mechanism that defines permissions required to perform operations on specific cloud resources under certain conditions. This mechanism allows for more flexible policy-based authorization, meeting requirements for secure access control. For example, you can grant MRS users only the permissions for performing specified operations on MRS clusters, such as creating a cluster and querying a cluster list rather than deleting a cluster. Most policies define permissions based on APIs. For the API actions supported by MRS, see Permissions Policies and Supported Actions.

Table 8-1 lists all the system policies supported by MRS.

<table>
<thead>
<tr>
<th>Policy Name</th>
<th>Description</th>
<th>Policy Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRS FullAccess</td>
<td>Administrator permissions for MRS. Users granted these permissions can operate and use all MRS resources.</td>
<td>Fine-grained policy</td>
</tr>
<tr>
<td>MRS CommonOperations</td>
<td>Common user permissions for MRS. Users granted these permissions can use MRS but cannot add or delete resources.</td>
<td>Fine-grained policy</td>
</tr>
<tr>
<td>MRS ReadOnlyAccess</td>
<td>Read-only permission for MRS. Users granted these permissions can only view MRS resources.</td>
<td>Fine-grained policy</td>
</tr>
</tbody>
</table>
| 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 8-2 lists the common operations supported by each system-defined policy or role of MRS. Select the policies or roles as required.

<table>
<thead>
<tr>
<th>Operation</th>
<th>MRS FullAccess</th>
<th>MRS CommonOperations</th>
<th>MRS ReadOnlyAccess</th>
<th>MRS Administrator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creating a cluster</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Operation</td>
<td>MRS FullAccess</td>
<td>MRS CommonOperations</td>
<td>MRS ReadOnlyAccess</td>
<td>MRS Administrator</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>----------------</td>
<td>----------------------</td>
<td>--------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Resizing a cluster</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Upgrading node specifications</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Deleting a cluster</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Querying cluster details</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Querying a cluster list</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Configuring an auto scaling rule</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Querying a host list</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Querying operation logs</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Creating and executing a job</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Stopping a job</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Deleting a single job</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Deleting jobs in batches</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Querying job details</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Querying a job list</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Creating a folder</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Operation</td>
<td>MRS FullAccess</td>
<td>MRS CommonOperations</td>
<td>MRS ReadOnlyAccess</td>
<td>MRS Administrator</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>----------------</td>
<td>----------------------</td>
<td>--------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Deleting a file</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>√</td>
</tr>
<tr>
<td>Querying a file list</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Operating cluster tags in batches</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>√</td>
</tr>
<tr>
<td>Creating a single cluster tag</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>√</td>
</tr>
<tr>
<td>Deleting a single cluster tag</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>√</td>
</tr>
<tr>
<td>Querying a resource list by tag</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Querying cluster tags</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Accessing MRS Manager</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>√</td>
</tr>
<tr>
<td>Querying a patch list</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Installing a patch</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>√</td>
</tr>
<tr>
<td>Uninstalling a patch</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>√</td>
</tr>
<tr>
<td>Authorizing O&amp;M channels</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>√</td>
</tr>
<tr>
<td>Sharing O&amp;M channel logs</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>√</td>
</tr>
<tr>
<td>Querying an alarm list</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>
### Operation Permissions Management

<table>
<thead>
<tr>
<th>Operation</th>
<th>MRS FullAccess</th>
<th>MRS CommonOperations</th>
<th>MRS ReadOnlyAccess</th>
<th>MRS Administrator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subscribing to alarm notification</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>√</td>
</tr>
<tr>
<td>Submitting an SQL statement</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>√</td>
</tr>
<tr>
<td>Querying SQL results</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>√</td>
</tr>
<tr>
<td>Canceling an SQL execution task</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>√</td>
</tr>
</tbody>
</table>

### Helpful Links
- IAM Service Overview
- Creating User Groups and Users and Granting MRS Permissions
- Permissions Policies and Supported Actions
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>
<thead>
<tr>
<th>Operation</th>
<th>Resource Type</th>
<th>Trace Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creating a cluster</td>
<td>cluster</td>
<td>createCluster</td>
</tr>
<tr>
<td>Deleting a cluster</td>
<td>cluster</td>
<td>deleteCluster</td>
</tr>
<tr>
<td>Operation</td>
<td>Resource Type</td>
<td>Trace Name</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Expanding a cluster</td>
<td>cluster</td>
<td>scaleOutCluster</td>
</tr>
<tr>
<td>Shrinking a cluster</td>
<td>cluster</td>
<td>scaleInCluster</td>
</tr>
</tbody>
</table>

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.
MRS uses the following infrastructure resources:

- ECS
- BMS
- VPC
- EVS
- Image Management Service (IMS)
- OBS
- EIP
- SMN
- IAM

For details about how to view and modify quotas, see Quotas.
## Change History

### Table 11-1 Version list

<table>
<thead>
<tr>
<th>Version</th>
<th>Release Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRS 1.9.2</td>
<td>2020-03-18</td>
<td>• Added the Alluxio component.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Added the Ranger component.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Upgraded Presto to version 0.216.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Upgraded Spark to version 2.2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Upgraded Hive to version 2.3.</td>
</tr>
<tr>
<td>MRS 2.1.0</td>
<td>2019-11-20</td>
<td>• Added the Impala component.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Added the Kudu component.</td>
</tr>
<tr>
<td>MRS 1.8.10</td>
<td>2019-11-13</td>
<td>• Supported interconnection with Bare Metal Server (BMS).</td>
</tr>
<tr>
<td>MRS 2.0.1</td>
<td>2019-06-18</td>
<td>• Upgraded Hadoop to version 3.1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Upgraded Spark to version 2.3.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Upgraded HBase to version 2.1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Upgraded Hive to version 3.1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Added the Tez component.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Supported the auto scaling function for yearly/monthly subscribed clusters.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Supported the creation of hybrid clusters.</td>
</tr>
</tbody>
</table>