Elastic MapReduce

Product Introduction

Product Documentation

Contents

Product Introduction
- Overview
- Strengths
- Architecture
- Features
- Use Cases
- Cluster Types
- Version Overview
- Constraints and Limits
- Product Releases and Component Versions

Product Introduction

Overview

Last updated：2022-08-17 16:26:16

Elastic MapReduce (EMR) is a secure, low-cost, and highly reliable open-source big data platform based on the cloud native technology and pan-Hadoop ecosystem. It provides diverse open-source big data components, such as Hive, Spark, HBase, Flink, StarRocks, Iceberg, and Alluxio, which are easy to deploy and manage, helping you efficiently build a cloud-based enterprise-grade data lake technology architecture.

Features

Completely derived from the open-source community, EMR enables you to seamlessly and smoothly migrate your existing big data clusters to Tencent Cloud. It is integrated with commonly used community components such as Hadoop, Hive, HBase, Spark, Presto, Flink, Sqoop, Hue, Iceberg, Druid, and StarRocks, fully meeting your various needs like online big data business, offline/online data warehousing, cloud-native data lake formation, and real-time stream computing.

EMR seamlessly incorporates the COS service, allowing you to migrate files stored in HDFS to COS with unlimited scalability, low storage costs, and high reliability for easy computing-storage separation. With COS, a cluster can be created whenever needed and terminated after the task is completed without the concerns over data loss. In addition, on-demand clusters can greatly reduce your big data processing costs.

EMR has five node types: master, core, task, router, and common. For the purposes of each type, see Cluster Types.

Currently, EMR supports multiple resource specifications, including Standard, Standard Network Optimized, MEM Optimized, High I/O, Computing, Computing Network Enhanced, and Big Data models. If you wan to deploy a cluster on CPM, contact us for assistance.

Strengths

Last updated：2020-11-23 10:09:12

Compared to self-created Hadoop clusters, Tencent Cloud EMR provides simpler, more stable, and more reliable Hadoop services.

> ⚠ **Note：**
>
> In addition to the Hadoop cluster, Druid and ClickHouse big data clusters are also supported to provide more choices of big data architectures.

Flexibility

- A secure and reliable Hadoop cluster can be created in just a few minutes to run mainstream open-source big data computing frameworks such as Hive, Spark, Presto, Impala, ClickHouse, Druid, and Flink, meeting your needs in scenarios such as **interactive BI, data warehousing, and real-time computation.**
- Existing EMR clusters can be elastically and quickly scaled, and in-cloud computing resources can be scheduled in real time to respond to fast changes in your business data, reducing the high costs for reserving IT hardware.

Reliability

- The master node is designed with disaster recovery in mind, and if it fails, a slave node will be started in seconds to ensure the availability of big data services.
- A comprehensive monitoring system is in place, which can send SMS messages for exceptions in cluster components and tasks in a matter of seconds.
- Hive metadata can be stored in MetaDB with a metadata reliability of 99.9996%.
- Petabytes of high-persistence data stored in COS can be analyzed.
- The recycle bin feature is enabled for clusters by default for you to restore devices that are deleted by mistake.

Security

- The network policy for managed Hadoop clusters can be well planned through the convenient network isolation enabled by VPCs. Network ACLs and security groups can be created to filter traffic at the subnet and server levels, helping meet your network security needs in all aspects.

- Tencent Cloud security reinforcement service provides an integrated security solution for EMR clusters, ranging from network protection and intrusion detection to vulnerability protection.
- Kerberos authentication can be enabled for clusters to ensure secure access.

Ease of Use

- Different clusters versions can be created to analyze the same data in COS in response to the actual business needs.
- Petabytes of data stored in data nodes or COS can be analyzed with the aid of out-of-the-box community components such as Hue and Oozie, eliminating your concerns over any knowledge migration costs.
- A full-featured, intuitive, and easy-to-use monitoring system is provided to present nearly 1,000 cluster-level and component-level monitoring metrics on the monitoring overview page.
- **In-Cloud clusters consisting of multiple models are supported in a flexible manner, so that you can easily scale out or distribute configurations to heterogeneously configured clusters, enabling you to cope with business analysis challenges with higher-specced hardware.**

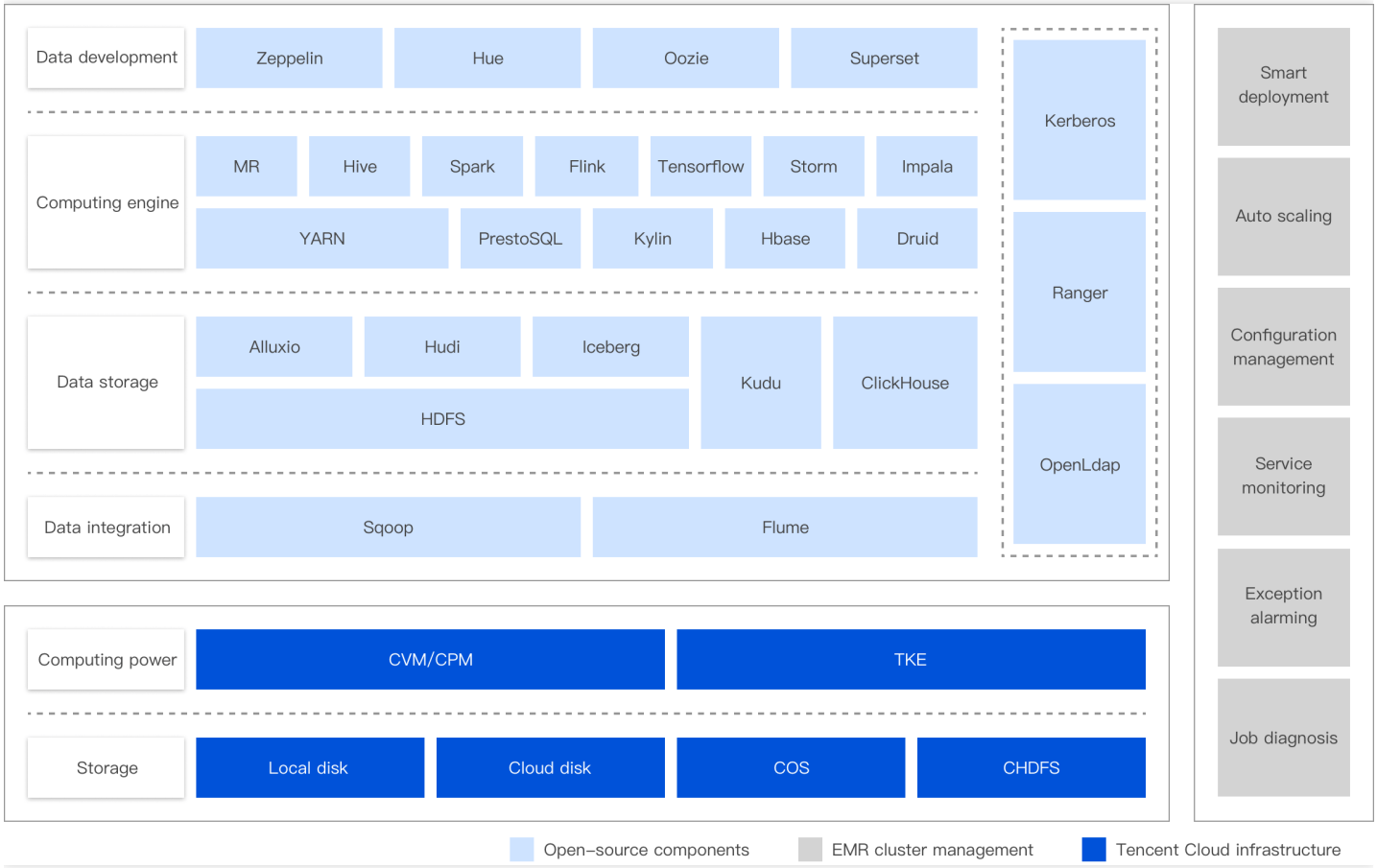
Reduced Costs

- EMR allows elastic scaling of your managed Hadoop cluster based on the business curve to reduce the high hardware costs.
- It comes with a rich set of OPS tools which greatly improve the efficiency and enable you to focus on the business itself without having to worry about repeated construction of infrastructure for monitoring, security, and OPS.
- Warm and cold data can be stored on COS/CHDFS, effectively reducing the costs by **28%–50%**.
- With unified Hive metadatabases and COS buckets, you can implement a cross-cluster architecture for analyzing the same dataset and create or terminate clusters as needed, reducing the cluster scaling costs.

Architecture

Last updated：2022-09-16 11:30:58

The logical architecture of EMR is as shown below:



An EMR cluster consists mainly of three parts: open-source components, Tencent Cloud infrastructure, and cluster management.

- Open-source components
  - EMR integrates dozens of cutting-edge open-source big data components from the Apache community, such as Hadoop, Hive, Spark, HBase, Presto, Flink, Alluxio, and Iceberg. For more information, see Product Releases and Component Versions.
  - Based on optimized Iceberg, Alluxio and other open-source components, EMR offers Iceberg Z-Order algorithm, Alluxio transparent URI and other enhanced features.
- Tencent Cloud infrastructure
  - EMR can be deployed based on multiple types of underlying computing resources, including CVM and CBM. It supports containerized deployment.
  - Data can be stored in a local disk, cloud disk, COS bucket, or CHDFS instance.
  - VPCs, network ACLs, and security groups provide EMR with a securely isolated network environment.

- Cluster Management
  - EMR supports smart cloud deployment management, including fast creation, flexible scaling, and auto scaling.
  - EMR offers a great variety of convenient Ops tools, such as service configuration management, batch node management, and visual service Ops.
  - EMR provides complete cluster monitoring and diagnosis capabilities ranging from multidimensional metric monitoring, event, and inspection to alarming and log search.

Features

Last updated：2020-10-10 16:35:51

Tencent Cloud Elastic MapReduce (EMR) is integrated with open-source frameworks and projects, such as Apache Hadoop, Apache Hive, Apache Spark, and Apache Storm. The cloud-based Hadoop services allow you to process vast amount of data securely, cost-efficiently, and elastically at scale. It has the following advantages:

Auto Scaling

Creating a cluster in minutes

You can create a secure, stable, cloud-managed Hadoop cluster in just a few minutes in the console.

Scaling in minutes

Your EMR cluster can be smoothly scaled up or down just a few minutes as your computing needs change.

API support

EMR clusters can be easily created, scaled, and terminated in programs through APIs.

Separation of Storage and Computation

Intra-cluster separation of storage and computation

At the cluster level, cloud-based Hadoop clusters can be planned in a manner where storage nodes and compute nodes are separated, so that you can scale the compute nodes as needed to lower the hardware costs.

COS-based separation of storage and computation

Massive amounts of data to be analyzed can be stored in COS. While the storage costs is reduced through COS, different versions of EMR clusters can be created to analyze the same data, which brings out extreme architectural flexibility.

OPS Support

Monitoring and multi-channel alarming

A comprehensive monitoring and OPS system is provided, which can detect exceptions in components such as Spark, Hive, and Presto and jobs within seconds after they occur to ensure the robust operations of big data clusters.

Technical support

In addition to comprehensive technical documentation, Tencent Cloud also has a technical service system where

complete technical support is provided through various channels such as ticket.

Security

EMR uses security groups to control inbound and outbound traffic to your CVM instances. Components' web UIs can only be accessed through one specified instance assigned with public IP, and the access requests must be authenticated by username and password. In addition, the security group of this instance only allows SSH ports and proxy access ports.

> ⚠ **Note：**
>
> Changing the project will cause the CVM instance to lose its security group.

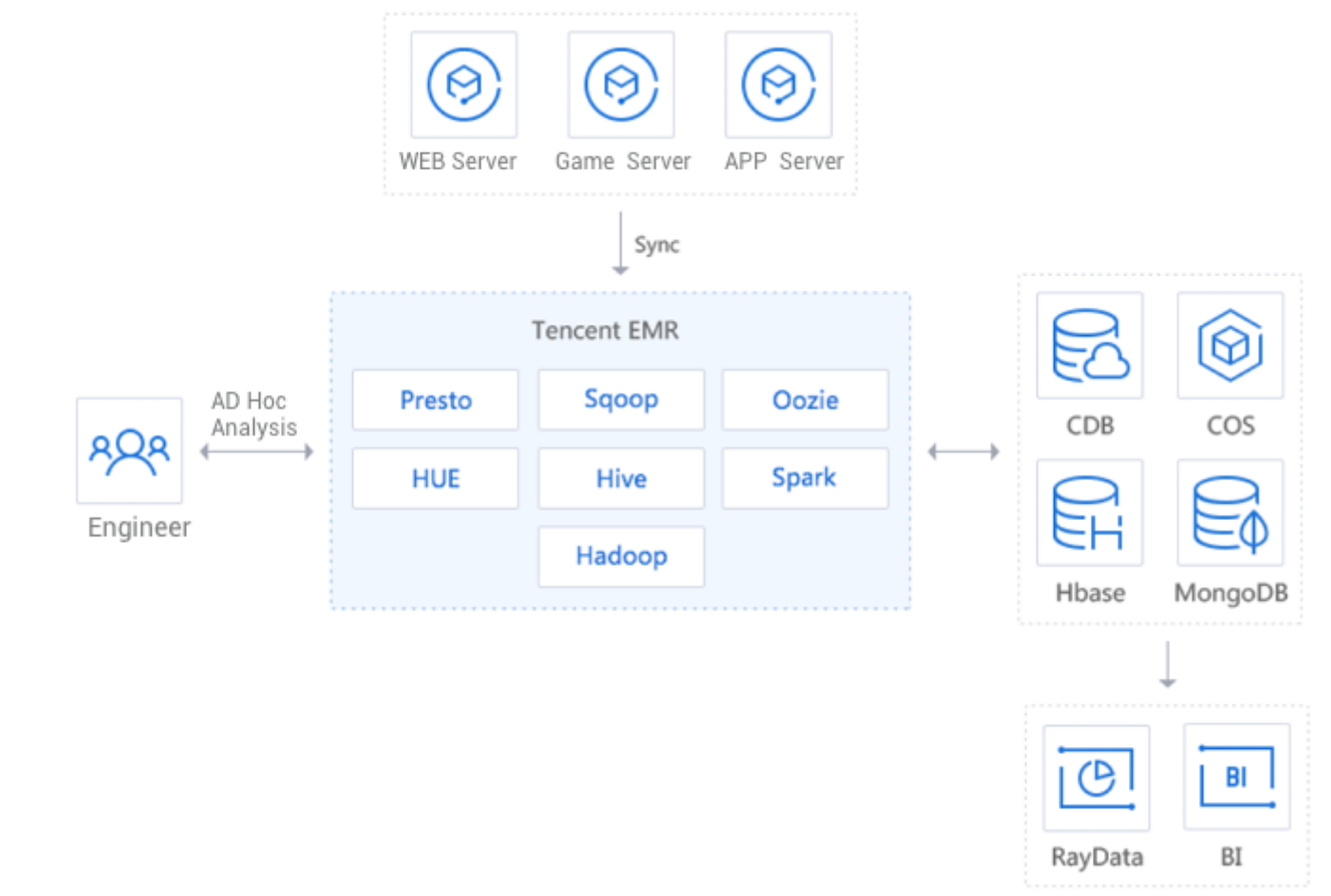
Use Cases

Last updated：2020-03-25 10:44:42

Elastic MapReduce (EMR) clusters supports many application scenarios by running big data frameworks Hadoop and Spark. Below are some typical EMR application scenarios:

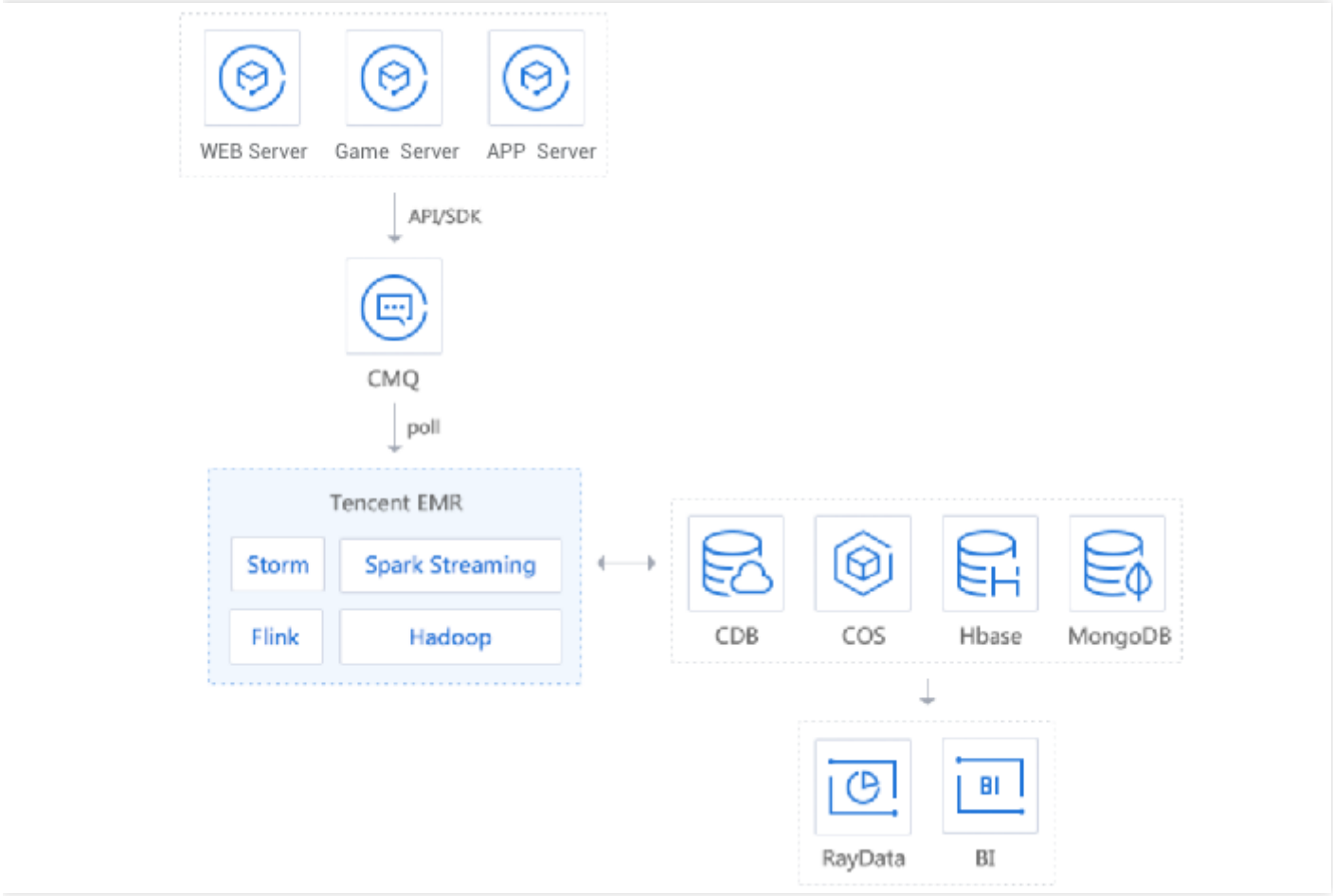
Offline Data Analysis

EMR synchronizes vast amounts of log data from business applications (e.g. games, webs, and mobile Apps) servers to nodes or COS. With Hue, an open source SQL Workbench for Data Warehouses, you can use big data frameworks such as Hive, Spark, and Presto to analyze data to derive business insights. In addition, you can use Sqoop to integrate and analyze the data scattered across TencentDB and other storage engines. Sqoop synchronizes the analyzed data back to TenentDB to support data visualization services like RayData.



Streaming Data Processing

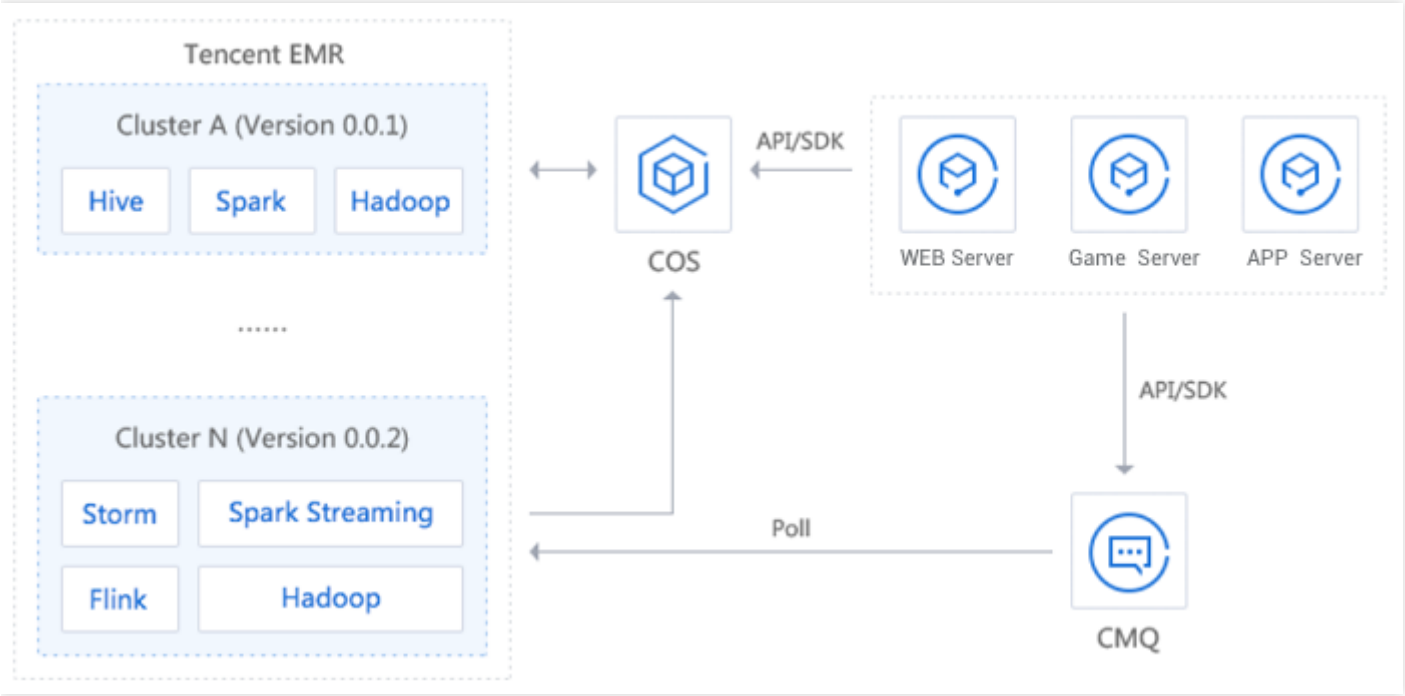
After pushing the real-time data generated from business servers to the CMQ messaging middleware through APIs and SDKs in programs/tools, you can select an appropriate streaming data processing engine in EMR to analyze the data for real-time alarming in respond to business changes. The analysis results can be synced to storage engines such as TencentDB in real time, which helps you monitor your business by using data visualization services like RayData.



COS Data Analysis

You can use EMR to process and analyze vast amounts of data stored in COS so that computation and storage can be completely separated. This architecture allows you to use various data synchronization tools in COS. At the same time, you can use multiple Hadoop cluster versions to analyze the same data to achieve data consistency and resolve

legacy issues caused by the coexistence of multi-version Hadoop clusters.



Cluster Types

Last updated：2022-08-12 14:59:18

EMR supports six cluster types and their respective use cases and defines five node types. Different cluster types and their respective use cases support different node types, number of deployed nodes, and deployed services. You can select the most appropriate cluster type and use case based on your business needs when creating a cluster.

Note：
ClickHouse, Doris, and Kafka cluster types are not available by default. To use them, submit a ticket for application.

Cluster Type Description

Hadoop cluster

Use Case	Description	Node Deployment Description

Use Case	Description	Node Deployment Description
Default use case	Based on open-source Hadoop and the components that form a Hadoop ecosystem, it provides big data solutions for massive data storage, offline/real-time data analysis, streaming data compute, and machine learning.	• **Master node:** It is a management node that ensures the scheduling of the cluster works properly. Processes such as NameNode, ResourceManager, and HMaster are deployed here. The number of master nodes is 1 in non-HA mode or 2 in HA mode. **Note: If Kudu is included in the deployed components, the cluster supports only the HA mode, and the number of master nodes is 3.** • **Core node:** It is a compute and storage node. All your data in HDFS is stored in core nodes. Therefore, in order to ensure data security, once core nodes are scaled out, they cannot be scaled in. Processes such as DataNode, NodeManager, and RegionServer are deployed here. The number of core nodes is ≥2 in non-HA mode or ≥3 in HA mode. • **Task node:** It is a pure compute node and does not store any data. The computed data comes from a core node or COS. Therefore, it is often used as an elastic node and can be scaled in or out at any time. Processes such as NodeManager and PrestoWork are deployed here. The number of task nodes can be changed at any time to scale the cluster. The minimum value is 0. • **Common node:** It provides data sharing and syncing and HA fault tolerance services for the master nodes in an HA cluster. Distributed coordinator components such as ZooKeeper and JournalNode are deployed here. The number of common nodes is 0 in non-HA mode or ≥3 in HA mode. • **Router node:** It is used to share the load of a master node or as the task submitter of the cluster. It can be scaled in or out at any time. Hadoop packages, including software programs and processes such as Hive, Hue, and Spark, are deployed here. The number of router nodes can be changed at any time. The minimum value is 0.
ZooKeeper	It is suitable for creating a distributed, high-availability coordination service for large clusters.	• **Common node:** Distributed coordinator components such as ZooKeeper are deployed here. The number of deployed nodes must be odd and at least three common nodes. Common nodes support only the HA mode.

Use Case	Description	Node Deployment Description
HBase	It is suitable for storing massive amounts of unstructured or semi-structured data. It provides a high-reliability, high-performance, column-oriented, scalable distributed storage system that supports real-time data read/write.	• **Master node:** It is a management node that ensures the scheduling of the cluster works properly. Processes such as NameNode, ResourceManager, and HMaster are deployed here. The number of master nodes is 1 in non-HA mode or 2 in HA mode. • **Core node:** It is a compute and storage node. All your data in HDFS is stored in core nodes. Therefore, in order to ensure data security, once core nodes are scaled out, they cannot be scaled in. Processes such as DataNode, NodeManager, and RegionServer are deployed here. The number of core nodes is ≥2 in non-HA mode or ≥3 in HA mode. • **Task node:** It is a pure compute node and does not store any data. The computed data comes from a core node or COS. Therefore, it is often used as an elastic node and can be scaled in or out at any time. Processes such as NodeManager are deployed here. The number of task nodes can be changed at any time to scale the cluster. The minimum value is 0. • **Common node:** It provides data sharing and syncing and HA fault tolerance services for the master nodes in an HA cluster. Distributed coordinator components such as ZooKeeper and JournalNode are deployed here. The number of common nodes is 0 in non-HA mode or ≥3 in HA mode. • **Router node:** It is used to share the load of a master node or as the task submitter of the cluster. It can be scaled in or out at any time. The number of router nodes can be changed at any time. The minimum value is 0.

Use Case	Description	Node Deployment Description
Presto	It provides an open-source distributed SQL query engine for quick query and analysis of massive amounts of data. It is suitable for interactive analytical queries.	• **Master node:** It is a management node that ensures the scheduling of the cluster works properly. Processes such as NameNode and ResourceManager are deployed here. The number of master nodes is 1 in non-HA mode or 2 in HA mode. • **Core node:** It is a compute and storage node. All your data in HDFS is stored in core nodes. Therefore, in order to ensure data security, once core nodes are scaled out, they cannot be scaled in. Processes such as DataNode and NodeManager are deployed here. The number of core nodes is ≥2 in non-HA mode or ≥3 in HA mode. • **Task node:** It is a pure compute node and does not store any data. The computed data comes from a core node or COS. Therefore, it is often used as an elastic node and can be scaled in or out at any time. Processes such as NodeManager and PrestoWork are deployed here. The number of task nodes can be changed at any time to scale the cluster. The minimum value is 0. • **Common node:** It provides data sharing and syncing and HA fault tolerance services for the master nodes in an HA cluster. Distributed coordinator components such as ZooKeeper and JournalNode are deployed here. The number of common nodes is 0 in non-HA mode or ≥3 in HA mode. • **Router node:** It is used to share the load of a master node or as the task submitter of the cluster. It can be scaled in or out at any time. The number of router nodes can be changed at any time. The minimum value is 0.

Use Case	Description	Node Deployment Description
Kudu	It provides a distributed and scalable columnar storage manager and supports random reads/writes and OLAP analysis to process frequently updated data.	• **Master node:** It is a management node that ensures the scheduling of the cluster works properly. Processes such as NameNode and ResourceManager are deployed here. The number of master nodes is 1 in non-HA mode or 2 in HA mode. • **Core node:** It is a compute and storage node. All your data in HDFS is stored in core nodes. Therefore, in order to ensure data security, once core nodes are scaled out, they cannot be scaled in. The number of core nodes is ≥2 in non-HA mode or ≥3 in HA mode. • **Task node:** It is a pure compute node and does not store any data. The computed data comes from a core node or COS. Therefore, it is often used as an elastic node and can be scaled in or out at any time. The number of task nodes can be changed at any time to scale the cluster. The minimum value is 0. • **Common node:** It provides data sharing and syncing and HA fault tolerance services for the master nodes in an HA cluster. Distributed coordinator components such as ZooKeeper and JournalNode are deployed here. The number of common nodes is 0 in non-HA mode or ≥3 in HA mode. • **Router node:** It is used to share the load of a master node or as the task submitter of the cluster. It can be scaled in or out at any time. The number of router nodes can be changed at any time. The minimum value is 0.

Druid cluster

Use Case	Description	Node Deployment Description

Use Case	Description	Node Deployment Description
Default use case	It supports high-performance real-time analysis, big data queries in milliseconds, and multiple data ingestion methods. It is suitable for real-time big data query scenarios.	• **Master node:** It is a management node that ensures the scheduling of the cluster works properly. Processes such as NameNode and ResourceManager are deployed here. The number of master nodes is 1 in non-HA mode or 2 in HA mode. • **Core node:** It is a compute and storage node. All your data in HDFS is stored in core nodes. Therefore, in order to ensure data security, once core nodes are scaled out, they cannot be scaled in. Processes such as DataNode and NodeManager are deployed here. The number of core nodes is ≥2 in non-HA mode or ≥3 in HA mode. • **Task node:** It is a pure compute node and does not store any data. The computed data comes from a core node or COS. Therefore, it is often used as an elastic node and can be scaled in or out at any time. Processes such as NodeManager are deployed here. The number of task nodes can be changed at any time to scale the cluster. The minimum value is 0. • **Common node:** It provides data sharing and syncing and HA fault tolerance services for the master nodes in an HA cluster. Distributed coordinator components such as ZooKeeper and JournalNode are deployed here. The number of common nodes is 0 in non-HA mode or ≥3 in HA mode. • **Router node:** It is used to share the load of a master node or as the task submitter of the cluster. It can be scaled in or out at any time. The number of router nodes can be changed at any time. The minimum value is 0.

ClickHouse cluster

Use Case	Description	Node Deployment Description
Default use case	It provides a column-oriented database management system. It is suitable for data warehouse analysis scenarios such as real-time wide table analysis, real-time BI report analysis, and user behavior analysis.	• **Core node:** It is a compute and storage node. ClickHouseServer is deployed here. • **Common node:** It provides data sharing and syncing and HA fault tolerance services for the master nodes in an HA cluster. Distributed coordinator components such as ZooKeeper are deployed here. The number of common nodes is 0 in non-HA mode or ≥3 in HA mode.

Doris cluster

Use Case	Description	Node Deployment Description
Default use case	It provides an MPP analytical database product that supports sub-second queries on PB-scale, structured data. It is compatible with MySQL protocol and uses the standard SQL syntax. It is suitable for historical report analysis, real-time data analysis, interactive data analysis, etc.	• **Master node:** It is a frontend module that provides the Web UI feature. Processes such as FE Follower and Broker are deployed here. The number of master nodes is ≥1 in non-HA mode or ≥3 in HA mode. • **Core node:** It is a backend module that provides the data storage feature. Processes such as BE and Broker are deployed here. The number of core nodes is ≥3. • **Router node:** It is a frontend module that helps achieve high read/write availability. Processes such as FE Observer and Broker are deployed here. Router nodes can be scaled out but not in.

Kafka cluster

Use Case	Description	Node Deployment Description
Default use case	It provides a distributed, partitioned, multi-replica, and multi-subscriber message processing system based on ZooKeeper coordination. It is suitable for asynchronous processing, message communication, and streaming data receiving and distribution.	• **Core node:** It provides a distributed, partitioned, multi-replica, and multi-subscriber message processing system based on ZooKeeper coordination. It is suitable for asynchronous processing, message communication, and streaming data receiving and distribution. • **Common node:** It provides data sharing and syncing and HA fault tolerance services for the core nodes in an HA cluster. The number of common nodes is 0 in non-HA mode or ≥3 in HA mode.

StarRocks cluster

Use Case	Description	Node Deployment Description
Default use case	StarRocks adopts full vectorization technology. It supports extremely fast and unified OLAP databases. It is suitable for many data analysis scenarios, such as multidimensional, real-time, and high-concurrency analysis.	• **Master node:** It is a frontend module that provides the Web UI feature. Processes such as FE Follower and Broker are deployed here. The number of master nodes is ≥1 in non-HA mode or ≥3 in HA mode. • **Core node:** It is a backend module that provides the data storage feature. Processes such as BE and Broker are deployed here. The number of core nodes is ≥3. • **Router node:** It is a frontend module that helps achieve high read/write availability. Processes such as FE Observer and Broker are deployed here. Router nodes can be scaled out but not in.

Version Overview

Last updated：2022-08-19 10:55:32

Product Release Overview

EMR consists of open-source applications in a series of big data ecosystems. It offers six cluster types for you to deploy as needed.

Product Release Number Format

1. EMR version numbers are in the format of `EMR va.b.c` as detailed below:

   The meanings of `a` for different clusters are as follows:

   - For Hadoop clusters, `a` indicates the Hadoop versions supported by the current version. When `a` is `1` or `2`, Hadoop v2.x is supported; when `a` is `3`, Hadoop v3.x is supported.
   - For Druid clusters, `a` indicates the Druid versions supported by the current version. When `a` is `1`, Druid v0.17.x is supported.
   - For ClickHouse clusters, `a` indicates the ClickHouse versions supported by the current version. When `a` is `1`, ClickHouse v19.x and v20.x are supported.
   - For Kafka clusters, `a` indicates the Kafka versions supported by the current version. When `a` is `1`, Kafka v1.x is supported.
   - For Doris clusters, `a` indicates the Doris versions supported by the current version. When `a` is `1`, Doris v0.13x is supported.
   - For StarRocks clusters, `a` indicates the StarRocks versions supported by the current version. When `a` is `1`, StarRocks v2.x is supported.
     `b` indicates that the version has new components or supports component version upgrade.
     `c` indicates feature optimization.

Note：

- The components and their versions bundled with each EMR version are fixed. Currently, neither selecting multiple versions of a component nor changing a component version in one EMR version is supported. For example, Hadoop v2.8.5 and Spark v3.2.1 are built into EMR v2.7.0.

- Once a version of EMR is selected for cluster creation, the EMR and component version used by the cluster will not be automatically upgraded. For example, if EMR v2.7.0 is selected, then Hadoop will always be v2.8.5, and Spark will always be v3.2.1. Even if EMR is upgraded to v2.8.0, Hadoop is upgraded to a higher version, and Spark is upgraded to v3.3.0 afterward, the previously created cluster will not be affected, and only new clusters will use the new versions.
- When you upgrade the cluster through data migration, for example, from EMR v2.6.0 to EMR v2.7.0, in order to avoid issues such as version incompatibility or environment changes, be sure to test the tasks to be migrated and ensure that they can work properly in the new software environment.
- EMR v2.4.0 comes with Kona (based on OpenJDK8). We have developed and improved Kona based on the characteristics of cloud scenarios.

Constraints and Limits

Last updated：2022-05-24 09:31:13

Before using EMR, carefully read and understand the following use limits:

- To ensure the cluster network security, new clusters will be placed in the same VPC. Do not change the VPC of an existing cluster or node; otherwise, the cluster network may fail.
- When you create a cluster, EMR can help you create a new security group, or you can also manually select an existing EMR security group. Make sure that the selected security group has the necessary inbound/outbound rules for EMR. Do not delete or change the security group in use after creating the cluster; otherwise, cluster communication may fail, thus affecting the service.
- Plan the storage space of nodes in advance based on your business needs, and promptly add storage nodes, as an insufficient storage space may cause data and node risks. Currently, core, task, and router nodes in the EMR cluster support mounting multiple cloud disks, and a node can have up to 15 cloud disks. Clusters in BM 2.0 and local disk models (IO and D series) don't support mounting multiple cloud disks.
- When using EMR, do not perform operations in the CVM console, such as shutdown, restart, VPC switch, security group rule adjustment, so as to avoid cluster exceptions. OS reinstallation, instance termination, configuration adjustment, renewal, and billing mode change are also not recommended. You can perform necessary cluster maintenance operations in the EMR console.
- Public IPs can increase the possibility of master nodes being attacked, so you need to manage and monitor the risks. EIPs (including the IPs on the secondary ENI) will be retained after the cluster is terminated, and the idle IPs will continue to incur fees. If you don't need to retain them, release them on the corresponding resource management page.
- When you create a cluster, EMR provides component initialization parameters for general scenarios. Before you use components such as HDFS and HBase, we recommend you check whether the component parameters meet the needs of your business scenarios. To get the component initialization guide, contact us.
- Keep the host login password of your EMR cluster secure. If you configure passwordless login for cross-node access, Tencent Cloud security services may detect vulnerability risks and prompt you.
- Even if an exception occurs in your cluster, the cluster will continue to be billed. In this case, we recommend you promptly contact us for assistance. If we need to log in to your cluster for troubleshooting, we will gain your consent to request your account and password.

When you use or maintain your EMR cluster, some unexpected operations may render it unavailable or unstable, and you will receive a risk warning before performing such operations in the console. This document lists some of the prohibited and risky operations:

Prohibited Operations

Operation	Risk
Change the private IP of an EMR node in CVM	Node communication exception or cluster unavailability
Modify the security group of a CVM node while the cluster is running	Node communication exception or component service unavailability
Delete an existing process, application, or file from a node	Cluster/Component service unavailability
Delete or modify the `hosts` file in the `/etc` directory	Service exception, as the cluster cannot be associated with the service on the node
Delete or modify the HDFS metadata file `edit log`	HDFS cluster unavailability
Manually modify the data in the Hive metadatabase	Service exception caused by a Hive data parsing error
Delete a ZooKeeper data directory	Failure of dependent components

Risky Operations

Operation	Risk	Suggestion
Shut down or restart an EMR cluster node in CVM	Service unavailability	Check whether the operation is necessary and read the CVM operation limits
Mount a disk to an EMR node in the CVM console	Disk unavailability caused by EMR's failure to recognize and initialize the disk	Do this through core node scaling or under technical guidance
Unmount a disk from an EMR node in the CVM console	Data loss or cluster unavailability	Do this through core node scaling or under technical guidance
Modify the component configuration file parameters in CVM	Modified parameter overwrite after service restart	Modify the parameter configuration in the EMR console and seek technical guidance in special circumstances
Delete or modify the `resolv.conf`	Service exception, as the cluster cannot be associated with the service on the node	Check whether the operation is necessary and perform it under

file in the `/etc` directory		technical guidance
Modify the MetaDB password	EMR depends on the password configured in MetaDB, and after the password is changed, services such as Hive and Ranger will become unavailable	Modify the configuration in the EMR console under technical guidance
Modify the MetaDB floating IP	Hive and Ranger service unavailability, as EMR depends on the IP configured in MetaDB	Modify the configuration in the EMR console under technical guidance
Modify the MetaDB security group	Interruption of the communication between MetaDB and the cluster, or Hive and Ranger service unavailability	Perform the operation under technical guidance

Product Releases and Component Versions

Last updated：2022-11-07 15:21:50

Disused EMR Versions

Some earlier EMR versions are disused as they can’t use the new features from the community due to the low versions of open-source components. You can't create new clusters on those disused versions, but you can still scale in and out the existing ones.

- Disused versions of the Hadoop cluster type: EMR v1.3.1, EMR vV2.0.1, EMR v2.1.0, EMR v2.2.0, EMR v2.4.0, EMR v2.5.1, EMR v3.0.0, EMR v3.2.0, and EMR-TianQiong v1.0.0.
- Disused version of the Druid cluster type: Druid v1.0.0.

We recommend that you create clusters with the latest stable versions of cluster types to enjoy more features and more stable services.

EMR Standard Edition Changelog

Currently, EMR Standard supports the Hadoop, Druid, ClickHouse, Kafka, Doris, and StarRocks clusters.

Show All

Hadoop v2.X Standard supports the following component versions:

展开&收起

Component	EMR v 2.7.0	EMR v2.6.0	EMR v2.5.0	EMR v2.3.0
Release Date	July 2022	July 2021	September 2020	May 2020
HDFS (required)	2.8.5	2.8.5	2.8.5	2.8.5
Yarn (required)	2.8.5	2.8.5	2.8.5	2.8.5
ZooKeeper (required)	3.6.3	3.6.1	3.6.1	3.5.5
OpenLDAP (required)	2.4.44	2.4.44	-	-
Knox (required)	1.6.1	1.2.0	1.2.0	1.2.0

Component	EMR v 2.7.0	EMR v2.6.0	EMR v2.5.0	EMR v2.3.0
Tez	0.10.1	0.9.2	0.9.2	0.9.2
Hive	2.3.9	2.3.7	2.3.7	2.3.5
Spark	3.2.1	3.0.2	3.0.0	2.4.3
Livy	0.8.0	0.8.0	0.7.0	0.7.0
Kyuubi	1.4.1	1.4.1	-	-
Kylin	4.0.1	2.5.2	2.5.2	2.5.2
Presto	-	-	-	0.228
Trino (PrestoSQL)	385	332	332	-
Kudu	1.15.0	1.12.0	1.12.0	-
Impala	3.4.0	3.4.0	2.10.0	2.10.0
Storm	1.2.3	1.2.3	1.2.3	1.2.3
Flink	1.14.3	1.12.1	1.10.0	1.9.2
HBase	2.4.5	1.4.9	1.4.9	1.4.9
Phoenix (integrated in HBase)	5.1.2	4.14.0	4.13.0	4.13.0
Alluxio	2.8.0	2.5.0	2.3.0	1.8.1
Iceberg	0.13.0	0.11.0	-	-
Hudi	0.11.0	0.7.0	-	0.5.1
Hue	4.10.0	4.6.0	4.6.0	4.6.0
Oozie	5.2.1	5.1.0	5.1.0	5.1.0
Zeppelin	0.10.1	0.9.1	0.8.2	0.8.2
Superset	1.4.1	0.35.2	0.35.2	0.35.2
TensorFlowSpark	1.4.4	1.4.4	1.4.4	1.4.4
Jupyter (installed with TensorFlow)	4.6.3	4.6.3	4.6.3	4.6.3

Component	EMR v 2.7.0	EMR v2.6.0	EMR v2.5.0	EMR v2.3.0
Sqoop	1.4.7	1.4.7	1.4.7	1.4.7
Flume	1.9.0	1.9.0	1.9.0	1.9.0
Ranger	2.1.0	1.2.0	1.2.0	1.2.0
Kerberos (only available for selection in the creation process)	1.15.0	1.15.0	1.15.0	1.15.0
Ganglia	3.7.2	3.7.2	3.7.2	3.7.2
goosefs	1.2.0	-	-	-

Hadoop v3.X Standard supports the following component versions:

展开&收起

Component	EMR-V3.5.0	EMR-V3.4.0	EMR-V3.3.0	EMR v3.2.1	EMR v3.1.0
Release Date	October 2022	April 2022	September 2021	July 2021	December 2020
HDFS (required)	3.2.2	3.2.2	3.2.2	3.2.2	3.1.2
Yarn (required)	3.2.2	3.2.2	3.2.2	3.2.2	3.1.2
ZooKeeper (required)	3.6.3	3.6.3	3.6.1	3.6.1	3.6.1
OpenLDAP (required) ）	2.4.44	2.4.44	2.4.44	2.4.44	-
Knox (required)	1.6.1	1.6.1	1.2.0	1.2.0	1.2.0
Tez	0.10.2	0.10.1	0.10.1	0.10.0	0.9.2
Hive	3.1.3	3.1.2	3.1.2	3.1.2	3.1.1
Spark	3.2.2	3.2.1	3.0.2	3.0.2	2.4.3
Livy	0.8.0	0.8.0	0.8.0	-	-
Kyuubi	1.6.0	1.4.1	1.1.0	-	-
Kylin	4.0.1	4.0.1	4.0.1	-	-

Component	EMR-V3.5.0	EMR-V3.4.0	EMR-V3.3.0	EMR v3.2.1	EMR v3.1.0
Presto	-	-	-	-	-
Trino (PrestoSQL)	389	372 (renamed Trino)	350	350	332
Impala	4.1.0	4.0.0	3.4.0	3.4.0	3.4.0
Kudu	1.16.0	1.15.0	1.15.0	1.13.0	1.13.0
HBase	2.4.5	2.4.5	2.3.5	2.3.3	2.3.3
Phoenix (integrated in HBase)	5.1.2	5.1.2	5.1.2	5.0.0	5.0.0
Flink	1.14.5	1.14.3	1.12.1	1.12.1	1.10.0
Hue	4.10.0	4.10.0	4.10.0	4.4.0	4.4.0
Oozie	5.2.1	5.1.0	5.1.0	5.1.0	5.1.0
Zeppelin	0.10.1	0.10.1	0.9.1	0.9.1	0.8.2
Superset	1.5.1	1.4.1	1.4.1	-	-
Alluxio	2.8.0	2.8.0	2.5.0	2.5.0	2.3.0
Iceberg	0.13.1	0.13.1	0.11.0	0.11.0	-
Hudi	0.12.0	0.10.1	0.8.0	-	-
Flume	1.10.0	1.9.0	1.9.0	1.9.0	1.9.0
Sqoop	1.4.7	1.4.7	1.4.7	1.4.7	1.4.7
Ranger	2.3.0	2.1.0	2.1.0	2.1.0	2.0.0
Kerberos (only available for selection in the creation process)	1.15.1	1.15.1	1.15.1	1.51.1	1.15.1
Ganglia	3.7.2	3.7.2	3.7.2	-	-
deltalake	2.0.0	-	-	-	-
goosefs	1.3.0	1.2.0	-	-	-

Druid clusters support the following component versions:

展开&收起

Component	Druid v1.1.0
Release Date	August 2022
HDFS (required)	2.8.5
YARN (required)	2.8.5
Druid (required)	0.23.0
ZooKeeper (required)	3.6.3
Knox (required)	1.2.0
Superset	1.4.1
Ganglia	3.7.2

ClickHouse clusters support the following component versions:

展开&收起

Component	ClickHouse v1.2.0	ClickHouse v1.1.0	ClickHouse v1.0.0
Release Date	September 2020	May 2020	April 2020
ClickHouse (required)	20.7.2.30	20.3.10.75	19.16.12.49
ZooKeeper (required)	3.4.9	3.4.9	3.4.9
Superset	0.35.2	0.35.2	-

Kafka clusters support the following component versions:

展开&收起

Component	Kafka v1.0.0
Release Date	May 2021
Kafka (required)	1.1.1
KafkaManager (required)	2.0.0.2
Knox (required)	1.2.0

Component	Kafka v1.0.0
ZooKeeper (required)	3.6.1

Doris clusters support the following component versions:

展开&收起

Component	Doris v1.2.0	Doris v1.1.0	Doris v1.0.0
Release Date	January 2022	September 2021	May 2021
Doris (required)	0.15.0	0.14.0	0.13.0
Knox (required)	1.2.0	1.2.0	1.2.0

StarRocks clusters support the following component versions:

展开&收起

Component	StarRocks v1.1.0	StarRocks v1.0.0
Release Date	August 2022	March 2022
StarRocks (required)	2.2.2	2.0.0
Knox (required)	1.2.0	1.2.0