# Alibaba Cloud E-MapReduce

## Product Introduction

Issue: 20190919

# Legal disclaimer

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

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

or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.

5. By law, all the content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of the Alibaba Cloud website, product programs, or content shall be used, modified , reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates . The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates).
6. Please contact Alibaba Cloud directly if you discover any errors in this document.

# Generic conventions

Table -1: Style conventions

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

| Style | Description | Example |
|---|---|---|
| {} or {a\|b} | It indicates that it is a required value, and only one item can be selected. | `swich` *{stand \| slave}* |

# Contents

# 1 What is E-MapReduce?

Alibaba Cloud Elastic MapReduce (or E-MapReduce) is a big data processing solution that facilitates the processing and analysis of massive amounts of data.

Built on Alibaba Cloud Elastic Compute Service (ECS) and based on open-source Apache Hadoop and Apache Spark, E-MapReduce flexibly manages your data in a wide range of scenarios, such as trend analysis, data warehousing, and online and offline data processing. It also makes it easy for you to import and export data to and from other cloud storage systems and database systems, such as Alibaba Cloud OSS and Alibaba Cloud RDS.

## Using E-MapReduce

In general, to use a distributed processing system such as Hadoop or Spark, follow these steps:

1. Evaluate the business characteristics.
2. Select a machine type.
3. Purchase a machine.
4. Prepare the hardware environment.
5. Install an operating system.
6. Deploy applications (such as Hadoop and Spark).
7. Start a cluster.
8. Write applications.
9. Run a job.
10. Obtain data or perform another operation.

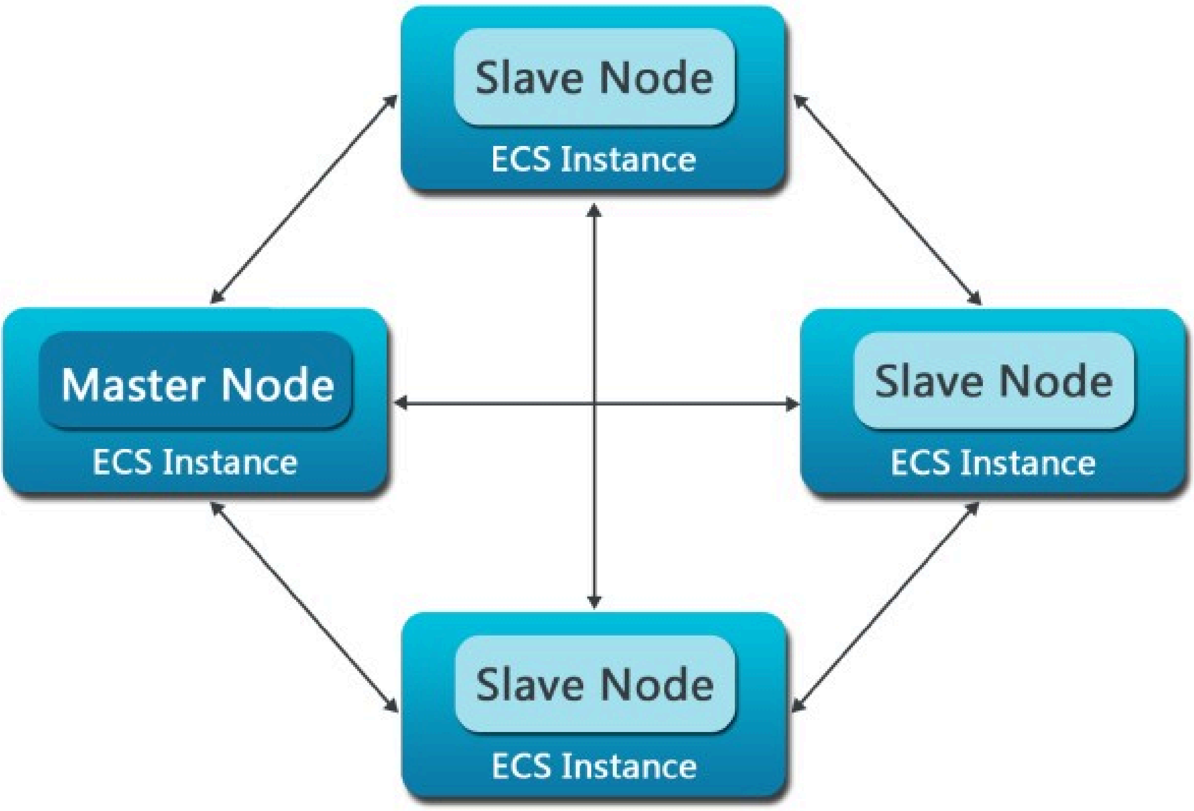
Steps 1-7 are preliminary tasks and may take some time to complete. Steps 8-10, however, concern application logic. E-MapReduce provides an integrated set of cluster management tools, including those used to build, configure, run, and manage clusters, configure and run jobs, as well as select hosts, deploy environments, and monitor performance.

With E-MapReduce, processes such as procurement, preparation, operation, and maintenance are all managed, allowing you to focus on the processing logic of your applications. E-MapReduce also provides flexible combination modes, allowing you to select different cluster services according to your needs. For example, if you want to receive daily statistics or perform simple batch operations, you can choose to only run Hadoop services in E-MapReduce. If you then want to implement stream-oriented and real-time computing at a later stage, you can add in Spark.

### Structure of E-MapReduce

Clusters are the core component of E-MapReduce. An E-MapReduce cluster is essentially a Spark or Hadoop cluster that consists of multiple Alibaba Cloud ECS instances. For example, in Hadoop, the daemons that typically run on each ECS instance (such as NameNode, DataNode, ResourceManager, and NodeManager) form a Hadoop cluster. The nodes that run NameNode and ResourceManager are known as master nodes, while those that run DataNode and NodeManager are called slave nodes.

The following figure shows an E-MapReduce cluster that consists of one master node and three slave nodes:



# 2 Release notes

## 2.1 Version overview

This topic describes the version number format, version update records, and version release notes of E-MapReduce.

### Version number format

E-MapReduce version numbers follow the a.b.c format. The details are as follows:

- Letter a indicates that major changes have been made in the new version.
- Letter b indicates that minor changes have been made to some components in the new version.
- Letter c indicates that the new version has fixed several bugs and is forward compatible.

The following section describes the version number changes in version upgrades:

- Upgrade from version 1.0.0 to 2.0.0: The value of letter a changes, indicating that major changes have been made in the new version. You must test that all existing jobs can run normally after the upgrade.
- Upgrade from version 1.0.0 to 1.1.0: The value of letter b changes, indicating that minor changes have been made to some components in the new version. Most features remain unchanged after the upgrade. However, we recommend that you conduct a test to verify the upgrade.
- Upgrade from version 1.0.0 to 1.0.1: The value of letter c changes, indicating that the new version is fully compatible with the previous version.

The bundled software, cluster creation, and cluster upgrade of each E-MapReduce release are described as follows:

- Bundled software: The software and software version are fixed in each release of E-MapReduce. Currently, E-MapReduce does not support choosing different versions of software. We recommend that you do not change the software version.
- Cluster creation: After you select a version of E-MapReduce and create a cluster with it, the cluster version is not upgraded automatically. Subsequent upgrades

to the image used by this version will not affect clusters that have been created. However, the latest image will be used to create new clusters.

- Cluster upgrade: When you upgrade the version of your cluster (for example, from version 1.0.x to 1.1.x), you must test that the existing jobs in the cluster can run normally in the new software environment.

## Version update records

- EMR-3.19.0 to EMR-3.22.0

| Version | EMR-3.19.0 | EMR-3.19.1 | EMR-3.20.0 | EMR-3.21.0 | EMR-3.22.0 |
|---|---|---|---|---|---|
| Release time | 2019.3 | 2019.4 | 2019.5 | 2019.6 | 2019.7 |
| Hadoop | 2.8.5 | 2.8.5 | 2.8.5 | 2.8.5 | 2.8.5 |
| Knox | 1.1.0 | 1.1.0 | 1.1.0 | 1.1.0 | 1.1.0 |
| ApacheDS | 2.0.0 | 2.0.0 | 2.0.0 | 2.0.0 | 2.0.0 |
| Spark | 2.4.1 | 2.4.1 | 2.4.2 | 2.4.3 | 2.4.3 |
| Hive | 3.1.1 | 3.1.1 | 3.1.1 | 3.1.1 | 3.1.1 |
| Tez | 0.9.1 | 0.9.1 | 0.9.1 | 0.9.1 | 0.9.1 |
| Pig | 0.14.0 | 0.14.0 | 0.14.0 | 0.14.0 | 0.14.0 |
| Sqoop | 1.4.7 | 1.4.7 | 1.4.7 | 1.4.7 | 1.4.7 |
| YARN | 2.8.5 | 2.8.5 | 2.8.5 | 2.8.5 | 2.8.5 |
| HDFS | 2.8.5 | 2.8.5 | 2.8.5 | 2.8.5 | 2.8.5 |
| Flink | 1.7.2 | 1.7.2 | 1.7.2 | 1.7.2 | 1.7.2 |
| Druid | 0.13.0 | 0.13.0 | 0.13.0 | 0.14.2 | 0.14.2 |
| HBase | 1.4.9 | 1.4.9 | 1.4.9 | 1.4.9 | 1.4.9 |
| Phoenix | 4.14.1 | 4.14.1 | 4.14.1 | 4.14.1 | 4.14.1 |
| Zookeeper | 3.4.13 | 3.4.13 | 3.4.13 | 3.4.13 | 3.5.5 |
| Livy | 0.60 | 0.60 | 0.60 | 0.60 | 0.60 |
| Presto | 0.213 | 0.213 | 0.213 | 0.213 | 0.221 |
| Storm | 1.2.2 | 1.2.2 | 1.2.2 | 1.2.2 | 1.2.2 |
| Impala | 2.12.2 | 2.12.2 | 2.12.2 | 2.12.2 | 2.12.2 |
| Flume | 1.8.0 | 1.8.0 | 1.8.0 | 1.8.0 | 1.8.0 |
| Hue | 4.1.0 | 4.1.0 | 4.1.0 | 4.4.0 | 4.4.0 |

| Version | EMR-3.19.0 | EMR-3.19.1 | EMR-3.20.0 | EMR-3.21.0 | EMR-3.22.0 |
|---|---|---|---|---|---|
| Oozie | 4.2.0 | 4.2.0 | 5.1.0 | 5.1.0 | 5.1.0 |
| Zeppelin | 0.8.0 | 0.8.0 | 0.8.1 | 0.8.1 | 0.8.1 |
| Ranger | 1.2.0 | 1.2.0 | 1.2.0 | 1.2.0 | 1.2.0 |
| Ganglia | 3.7.2 | 3.7.2 | 3.7.2 | 3.7.2 | 3.7.2 |
| OS | CentOS 7.4 | CentOS 7.4 | CentOS 7.4 | CentOS 7.4 | CentOS 7.4 |
| Tensorflow | - | - | 1.8.0 | 1.8.0 | 1.8.0 |
| Kafka | 2.11-1.1.1 | 1.1.1 | 2.11 | 1.1.1 | 1.1.1 |
| Superset | 0.28.1 | 0.28.1 | 0.28.1 | 0.28.1 | 0.28.1 |
| Jupyter | - | - | 4.4.0 | 4.4.0 | 4.4.0 |
| Analytics Zoo | - | - | 0.2.0 | 0.5.0 | 0.5.0 |
| Bigboot | - | - | 1.0.0 | 1.0.0 | 2.0.0 |
| OpenLDAP | - | - | - | - | 2.4.44 |
| Kudu | - | - | - | - | 1.10.0 |

- EMR-3.15.0 to EMR-3.18.1

| Version | EMR-3.15.0 | EMR-3.16.0 | EMR-3.17.0 | EMR-3.18.1 |
|---|---|---|---|---|
| Release time | 2018.11 | 2018.12 | 2019.1 | 2019.2 |
| Hadoop | 2.7.2 | 2.7.2-1.3.2 | 2.7.2 | 2.8.5 |
| Knox | 0.13.0 | 0.13.0 | 1.1.0 | 1.1.0 |
| ApacheDS | 2.0.0 | 2.0.0 | 2.0.0 | 2.0.0 |
| Spark | 2.3.2 | 2.3.2-1.0.1 | 2.3.2 | 2.3.2 |
| Hive | 2.3.3 | 2.3.3-1.0.3 | 2.3.3 | 3.1.1 |
| Tez | 0.9.1 | 0.9.1-1.0.2 | 0.9.1 | 0.9.1 |
| Pig | 0.14.0 | 0.14.0 | 0.14.0 | 0.14.0 |
| Sqoop | 1.4.7 | 1.4.7-1.0.0 | 1.4.7 | 1.4.7 |
| YARN | 2.7.2 | 2.7.2 | 2.7.2 | 2.8.5 |
| HDFS | 2.7.2 | 2.7.2 | 2.7.2 | 2.8.5 |
| Flink | 1.4.0 | 1.6.2-1.0.0 | 1.6.2 | 1.6.2 |
| Druid | 0.12.3 | 0.12.3-1.0.1 | 0.12.3 | 0.13.0 |
| HBase | 1.1.1 | 1.1.1-1.0.2 | 1.1.1 | 1.4.9 |

| Version | EMR-3.15.0 | EMR-3.16.0 | EMR-3.17.0 | EMR-3.18.1 |
|---|---|---|---|---|
| Phoenix | 4.10.0 | 4.10.0-1.0.0 | 4.10.0 | 4.14.1 |
| Zookeeper | 3.4.13 | 3.4.13 | 3.4.13 | 3.4.13 |
| Livy | 0.50. | 0.50. | 0.50. | 0.60. |
| Presto | 0.208 | 0.208 | 0.213 | 0.213 |
| Storm | 1.1.2 | 1.2.2 | 1.2.2 | 1.2.2 |
| Impala | 2.10.0 | 2.10.0-1.0.0 | 2.12.2 | 2.12.2 |
| Flume |  | 1.8.0 | 1.8.0 | 1.8.0 |
| Hue | 4.1.0 | 4.1.0 | 4.1.0 | 4.1.0 |
| Oozie | 4.2.0 | 4.2.0 | 4.2.0 | 4.2.0 |
| Zeppelin | 0.8.0 | 0.8.0 | 0.8.0 | 0.8.0 |
| Ranger | 1.0.0 | 1.0.0 | 1.0.0 | 1.0.0 |
| Ganglia | 3.7.2 | 3.7.2 | 3.7.2 | 3.7.2 |
| OS | CentOS 7.4 | CentOS 7.4 | CentOS 7.4 | CentOS 7.4 |
| Tensorflow | 1.8.0 | 1.8.0 | 1.8.0 | - |
| Kafka | 2.11-1.0.1 | 2.11-1.1.0 | 2.11-1.1.1 | 2.11-1.1.1 |
| Superset | 0.27.0 | 0.28.1 | 0.28.1 | 0.28.1 |
| Jupyter | 4.4.0 | 4.4.0 | 4.4.0 | - |
| Analytics Zoo | 0.2.0 | 0.2.0 | 0.2.0 | - |

- EMR3.11.0 to EMR-3.14.0

| Version | EMR-3.11.0 | EMR-3.12.0 | EMR-3.13.0 | EMR-3.14.0 |
|---|---|---|---|---|
| Release time | 2018.6 | 2018.7 | 2018.8 | 2018.10 |
| Hadoop | 2.7.2-emr-1.2.14 | 2.7.2-emr-1.2.14 | 2.7.2 | 2.7.2 |
| Knox | 0.13.0 | 0.13.0 | 0.13.0 | 0.13.0 |
| ApacheDS | 2.0.0 | 2.0.0 | 2.0.0 | 2.0.0 |
| Spark | 2.2.1 | 2.3.1 | 2.3.1 | 2.3.1 |
| Hive | 2.3.3 | 2.3.3 | 2.3.3 | 2.3.3 |
| Tez | 0.9.1 | 0.9.1 | 0.9.1 | 0.9.1 |
| Pig | 0.14.0 | 0.14.0 | 0.14.0 | 0.14.0 |
| Sqoop | 1.4.6 | 1.4.7 | 1.4.7 | 1.4.7 |

| Version | EMR-3.11.0 | EMR-3.12.0 | EMR-3.13.0 | EMR-3.14.0 |
|---|---|---|---|---|
| YARN | 2.7.2 | 2.7.2 | 2.7.2 | 2.7.2 |
| HDFS | 2.7.2 | 2.7.2 | 2.7.2 | 2.7.2 |
| Flink | 1.4.0 | 1.4.0 | 1.4.0 | 1.4.0 |
| Druid | 0.11.0 | 0.12.0 | 0.12.2 | 0.12.3 |
| HBase | 1.1.1 | 1.1.1 | 1.1.1 | 1.1.1 |
| Phoenix | 4.10.0 | 4.10.0 | 4.10.0 | 4.10.0 |
| Zookeeper | 3.4.11 | 3.4.12 | 3.4.12 | 3.4.13 |
| Livy | - | - | - | - |
| Presto | 0.188 | 0.188 | 0.208 | 0.208 |
| Storm | 1.1.2 | 1.1.2 | 1.1.2 | 1.1.2 |
| Impala | 2.10.0 | 2.10.0 | 2.10.0 | 2.10.0 |
| Flume | - | - | - | - |
| Hue | 4.1.0 | 4.1.0 | 4.1.0 | 4.1.0 |
| Oozie | 4.2.0 | 4.2.0 | 4.2.0 | 4.2.0 |
| Zeppelin | 0.7.3 | 0.7.3 | 0.8.0 | 0.8.0 |
| Ranger | 0.7.3 | 1.0.0 | 1.0.0 | 1.0.0 |
| Ganglia | 3.7.2 | 3.7.2 | 3.7.2 | 3.7.2 |
| OS | CentOS 7.4 | CentOS 7.4 | CentOS 7.4 | CentOS 7.4 |
| Tensorflow | - | - | 1.8.0 | 1.8.0 |
| Kafka | - | - | 2.11-1.0.1 | 2.11-1.0.1 |
| Superset | - | - | 0.25.6 | 0.25.6 |
| Jupyter | - | - | - | - |
| Analytics Zoo | - | - | - | - |

- EMR-3.7.1 to EMR-3.10.1

| Version | EMR-3.7.1 | EMR-3.8.1 | EMR-3.9.1 | EMR-3.10.1 |
|---|---|---|---|---|
| Release time | 2018.1 | 2018.1 | 2018.2 | 2018.4 |
| Hadoop | 2.7.2-emr-1.2.10 | 2.7.2-emr-1.2.12 | 2.7.2-emr-1.2.13 | 2.7.2-emr-1.2.14 |
| Knox | 0.13.0 | 0.13.0 | 0.13.0 | 0.13.0 |
| ApacheDS | 2.0.0 | 2.0.0 | 2.0.0 | 2.0.0 |

| Version | EMR-3.7.1 | EMR-3.8.1 | EMR-3.9.1 | EMR-3.10.1 |
|---|---|---|---|---|
| Spark | 2.2.1 | 2.2.1 | 2.2.1 | 2.2.1 |
| Hive | 2.3.2 | 2.3.2 | 2.3.2 | 2.3.2 |
| Tez | 0.8.4 | 0.8.4 | 0.8.4 | 0.9.1 |
| Pig | 0.14.0 | 0.14.0 | 0.14.0 | 0.14.0 |
| Sqoop | 1.4.6 | 1.4.6 | 1.4.6 | 1.4.6 |
| YARN | 2.7.2 | 2.7.2 | 2.7.2 | 2.7.2 |
| HDFS | 2.7.2 | 2.7.2 | 2.7.2 | 2.7.2 |
| Flink | - | 1.4.0 | 1.4.0 | 1.4.0 |
| Druid | - | - | 0.11.0 | 0.11.0 |
| HBase | 1.1.1 | 1.1.1 | 1.1.1 | 1.1.1 |
| Phoenix | 4.10.0 | 4.10.0 | 4.10.0 | 4.10.0 |
| Zookeeper | 3.4.11 | 3.4.11 | 3.4.11 | 3.4.11 |
| Livy | - | - | - | - |
| Presto | 0.188 | 0.188 | 0.188 | 0.188 |
| Storm | 1.0.1 | 1.0.1 | 1.0.1 | 1.1.2 |
| Impala | 2.10.0 | 2.10.0 | 2.10.0 | 2.10.0 |
| Flume | - | - | - | - |
| Hue | 3.12.0 | 3.12.0 | 3.12.0 | 4.1.0 |
| Oozie | 4.2.0 | 4.2.0 | 4.2.0 | 4.2.0 |
| Zeppelin | 0.7.1 | 0.7.1 | 0.7.1 | 0.7.1 |
| Ranger | - | - | 0.7.1 | 0.7.1 |
| Ganglia | 3.7.2 | 3.7.2 | 3.7.2 | 3.7.2 |
| OS | CentOS 7.4 | CentOS 7.4 | CentOS 7.4 | CentOS 7.4 |
| Tensorflow | - | - | - | - |
| Kafka | - | - | - | - |
| Superset | - | - | - | - |
| Jupyter | - | - | - | - |
| Analytics Zoo | - | - | - | - |

- EMR-2.9.2 to EMR-2.11.0

| Version | EMR-2.9.2 | EMR-2.10.0 | EMR-2.11.0 |
|---|---|---|---|
| Release time | 2018.2 | 2018.4 | 2018.7 |
| Hadoop | 2.7.2-emr-1.2.12 | 2.7.2-emr-1.2.12 | 2.7.2-emr-1.2.12 |
| Spark | 1.6.3 | 1.6.3 | 1.6.3 |
| Hive | 2.3.2 | 2.3.2 | 2.3.3 |
| Tez | 0.8.4 | 0.9.1 | 0.9.1 |
| Pig | 0.14.0 | 0.14.0 | 0.14.0 |
| Sqoop | 1.4.6 | 1.4.6 | 1.4.6 |
| Hue | 3.12.0 | 4.1.0 | 4.1.0 |
| Zeppelin | 0.7.1 | 0.7.1 | 0.7.3 |
| HBase | 1.1.1 | 1.1.1 | 1.1.1 |
| Phoenix | 4.10.0 | 4.10.0 | 4.10.0 |
| Storm | 1.0.1 | 1.1.2 | 1.1.2 |
| Presto | 0.188 | 0.188 | 0.188.0 |
| Impala | 2.10.0 | 2.10.0 | 2.10.0 |
| Zookeeper | 3.4.6 | 3.4.11 | 3.4.11 |
| Oozie | 4.2.0 | 4.2.0 | 4.2.0 |
| Ranger | 0.7.1 | 0.7.1 | 0.7.3 |
| Ganglia | 3.7.2 | 3.7.2 | 3.7.2 |
| OS | CentOS 7.4 | CentOS 7.4 | CentOS 7.4 |

- EMR-1.0.0 to EMR-1.3.0

| Version | EMR-1.0.0 | EMR-1.1.0 | EMR-1.2.0 | EMR-1.3.0 |
|---|---|---|---|---|
| Release time | 2015.11 | 2016.3 | 2016.4 | 2016.5 |
| Hadoop | 2.6.0 | 2.6.0 | 2.6.0 | 2.6.0-emr-1.1.1 |
| Spark | 1.4.1 | 1.6.0 | 1.6.1 | 1.6.1 |
| Hive | 1.0.1 | 1.0.1 | 2.0.0 | 2.0.0 |
| Pig | 0.14.0 | 0.14.0 | 0.14.0 | 0.14.0 |
| Sqoop | - | - | - | 1.4.6 |
| Hue | - | - | - | 3.9.0 |

| Version | EMR-1.0.0 | EMR-1.1.0 | EMR-1.2.0 | EMR-1.3.0 |
|---|---|---|---|---|
| Zeppelin | - | - | - | 0.5.6 |
| HBase | - | - | 1.1.1 | 1.1.1 |
| Phoenix | - | - | - | - |
| Zookeeper | - | - | 3.4.6 | 3.4.6 |
| Ganglia | 3.7.2 | 3.7.2 | 3.7.2 | 3.7.2 |

Note:

About Hadoop version:

To ensure full compatibility with Alibaba Cloud Object Storage Service (OSS), we add the emr-core component based on the open-source Hadoop version, without making any changes to the existing APIs. The version number of the emr-core component is appended to the end of the Hadoop version.

Release notes

For more information about the description of each release of E-MapReduce, see:

- #unique_6

## 2.2 Release notes

## 2.2.1 Release notes of EMR 3.x

This topic provides an overview of each EMR 3.x series version that has been released, including the release dates, component upgrades, new features, and updates.

EMR-3.22.0

For information about EMR-3.22.0, see #unique_9.

EMR-3.1.1

- Upgrades the operating system to CentOS 7.2.
- Upgrades Spark to version 2.1.
- Upgrades emr-core to version 1.2.6.
- Fixes several bugs related to AccessKey-free OSS operations.

EMR-3.0.2

- Upgrades emr-core to version 1.2.5.
- Supports AccessKey-free OSS operations in more regions.
- Adjusts the AccessKey replacement strategy of RAM roles.
- Fixes several bugs related to Hive and Hadoop.

EMR-3.0.1

- Supports table management in an interactive and unified manner, and stores Hive metadata in an external unified database. All clusters that use external Hive metadata share the same metadata.
- Upgrades emr-core to version 1.2.4 to improve the read and write performance of OSS.
- Upgrades Spark to version 2.0.2.

EMR-3.0.0

We recommend that you use EMR-3.0.1. EMR-3.0.1 is fully compatible with EMR-3.0.0 .

## 2.2.2 Release notes of EMR 3.22.0

This topic provides an overview of EMR 3.22.0 version that has been released, including the release dates, component upgrades, new features, and updates.

Release date

July 28, 2019

New features

- Kudu
  - The new component, Kudu, is added. Kudu fills in the gaps in the Hadoop ecosystem. It provides HBase-like fast data insertion and random access, and allows users to modify data. It also provides ultra-large-scale data analysis and query capabilities similar to querying Parquet files on HDFS.
    - Kudu provides C++ and Java APIs for secondary development.
    - Kudu supports integration of Impala, Spark, and Hive Metastore.
  - The Kudu component of EMR-3.22.0 is based on the open-source Apache Kudu 1 .10.0.

- OpenLDAP
  - The new component, OpenLDAP, is added to replace the previous component ApacheDS.
  - High availability is provided.

Component upgrades

- ZooKeeper

  Upgraded to version 3.5.5.
- Presto

  Upgraded to version 0.221.
- Bigboot

  Upgraded to version 2.0.0

Updates

- JindoFileSystem
  - Multiple storage modes
    - Block mode: Data is stored in blocks in the backend Object Storage Service (OSS). Local namespaces are used to maintain metadata. The block mode provides better metadata and data performance than the cache mode. The block mode provides multiple storage policies, including WARM (one local replica plus one replica in OSS), COLD (only one replica in OSS), HOT ( multiple local replicas plus one replica in OSS), TEMP (only one local replica ), and ALL_HDD (multiple local replicas). The default storage policy is WARM . You can set different storage policies for directories in different scenarios.
    - Cache mode: This mode is compatible with the existing data storage pattern of OSS. In the cache mode, files are stored as objects in OSS. The data and metadata of each file are cached locally based on the actual situations to improve data and metadata access performance. The cache mode provides different metadata synchronization policies to meet the requirements of different scenarios.
  - External clients
    - The client SDK provides access to E-MapReduce JindoFileSystem from outside E-MapReduce clusters. You can use the external client to access namespaces

in the block mode. However, you cannot use the external client to access the data cached by JindoFileSystem in E-MapReduce clusters. In addition, the performance of accessing data through the external client is worse than that of data access within E-MapReduce clusters.

    - The cache mode is compatible with the existing semantics of OSS storage. JindoFileSystem accelerates data caching in E-MapReduce clusters. Therefore , you can use the OSS client to directly access data from outside E-MapReduce clusters. For example, you can use the OSS SDK or OSSFileSystem of E-MapReduce to gain external access to the data.
  - Ecosystem components
    - Currently, JindoFileSystem supports various compute engines on E-MapReduce, such as Spark, Flink, Hive, MapReduce, Impala, and Presto.
    - If you need to separate data computing from data storage, you can store job logs, such as YARN container logs and Spark event logs, on JindoFileSystem.
    - JindoFileSystem can be used as the HFile backend storage of HBase to enhance the storage capabilities of HBase.
- OSSFileSystem
  - Adds the feature of automatically detecting bad disks. This feature can fix cache write failures caused by bad disks when you write data to OSS.
  - Completes the configurations of OSSFileSystem.
- Bigboot
  - Supports multiple namespaces, local data storage in blocks, multiple storage modes, and access from external clients.
  - Fixes the issue where the Bigboot monitor status is abnormal during server restart.
  - Improves the service specifications of Kudu.
  - Supports a validity check for the specification of each service.

- Hadoop
  - HDFS
    - Adapts to HDFS Federation. You can create an HDFS Federation cluster through custom configurations or APIs to avoid formatting the cluster a second time.
    - Optimizes the bad disk detection logic. If you are using a local disk, the system can perform bad disk detection when DataNode BR is triggered by the dfsadmin command.
  - YARN
    - Fixes the issue where the MapReduce job history list is not updated when MapReduce job container logs are stored on JindoFileSystem or OSS.
- Spark
  - Relational cache
    - Supports using a relational cache to accelerate data queries through pre-computing. You can create a relational cache to pre-compute data. During a data query, Spark Optimizer automatically discovers an appropriate relational cache, optimizes the SQL execution plan, and continues data computing based on the relational cache. This improves the query speed

and is suitable for various scenarios, such as reports, dashboards, data synchronization, and multidimensional analysis.

        - Supports using the data definition language (DDL) to perform operations such as CACHE, UNCACHE, ALTER, and SHOW. A relational cache supports all data sources and data formats of Spark.
        - Supports automatic data updates, manual data updates by the REFRESH command, and partition-based incremental updates of a relational cache.
        - Supports optimizing the SQL execution plan based on a relational cache.
- Streaming SQL
    - Normalizes the parameter settings of Stream Query Writer.
    - Optimizes the schema compatibility check of Kafka data tables.
    - Automatically registers a schema with SchemaRegistry for a Kafka data table that does not have a schema.
    - Optimizes log information recorded when a Kafka schema is incompatible.
    - Fixes the issue where the column name must be explicitly specified when the query result is written to a Kafka data table.
    - Removes the restriction that streaming SQL queries only support the Kafka and LogHub data sources.
- Delta
    - Adds the Delta component. For more information about Delta, click here. You can use Spark to create a Delta data source to perform streaming data writing, transactional reading and writing, data verification, and data backtracking.
        - You can call the dataframe API operation to read data from or write data to Delta.
        - You can call the structure streaming API operation to read or write data by using Delta as the data source or sink.
        - You can call Delta API operations to update, delete, merge, vacuum, and optimize data.
        - You can use SQL statements to create Delta tables, import data to Delta, and read data from Delta tables.
- Others
    - (Constraint feature) Supports primary keys and foreign keys.
    - Resolves JAR conflicts such as the servlet conflict

- Flink
  - Supports Log4j log rotation.
- Kafka
  - Supports Log4j log rotation.
  - Upgrades Fastjson.
- Zeppelin
  - Upgrades the dependent commons-lang3 package to version 3.7 to fix the issue where PySpark cannot write data to OSS. For more information, click here.
- Ranger
  - Supports the SHOW GRANTS command.
- Analytics-zoo
  - Fixes the Numpy installation error.
- Impala
  - Compatible with Apache Kudu 1.10.0.

# 3 Benefits

E-MapReduce has some practical strength over the self-built clusters. For example, it provides some convenient and controllable means to manage its clusters.

In addition, it also has the following strengths:

- Easy to use

  The user can select the required ECS types and disks and select the required software for automatic deployment.

  The user can apply for cluster resources at the corresponding position according to the geographical location where the user or the data source is located. Alibaba Cloud ECS currently supports the following regions: China (Hangzhou), China ( Shanghai), China (Qingdao), China (Beijing), China (Zhangjiakou), China (Hohhot ), China (Shenzhen), Singapore, Hong Kong, Australia (Sydney), Malaysia (Kuala Lumpur), Indonesia (Jakarta), Japan (Tokyo), Germany (Frankfurt), US (Silicon Valley), US (Virginia), India, and UAE (Dubai). E-MapReduce supports all of the regions supported by Alibaba Cloud ECS.
- Low price

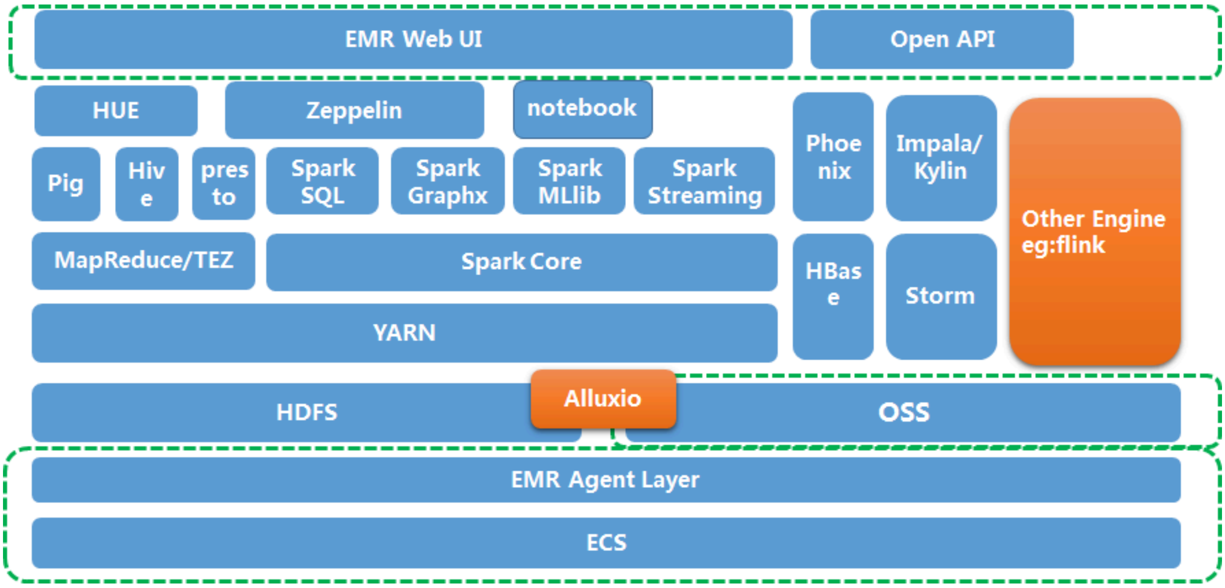
  The user can create a cluster as needed, that is, it can release the cluster after an offline task running is completed and add a node dynamically when needed.
- Deep integration

  E-MapReduce is deeply integrated with other Alibaba Cloud products (such as OSS, MNS, RDS, and MaxCompute) and can be used as the input source or output destination of the Hadoop/Spark computing engine.
- Security

  E-MapReduce integrates Alibaba Cloud RAM resource permission management system, so that it can isolate the service permissions through the primary account or sub-accounts.

# 4 Architecture

The product architecture of E-MapReduce is detailed in the following figure.



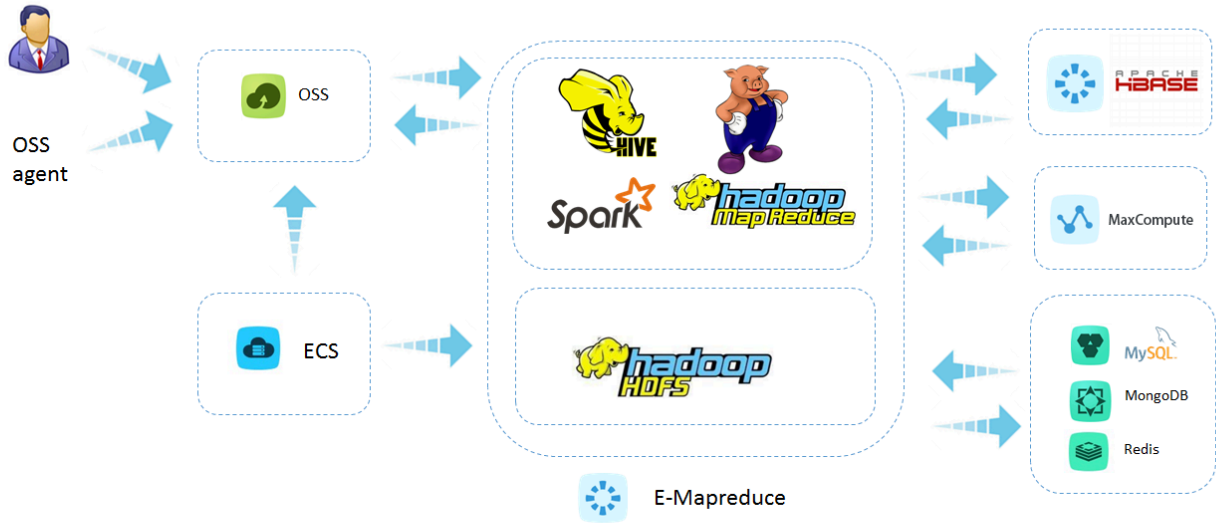
As shown in the figure, an E-MapReduce cluster is built based on the Hadoop ecosystem. This allows data to be exchanged seamlessly with cloud services, such as Alibaba Cloud Object Storage Service (OSS) and ApsaraDB (RDS), and enables you to share and transfer data between multiple systems. This meets the access needs of different types of businesses.

# 5 Scenarios

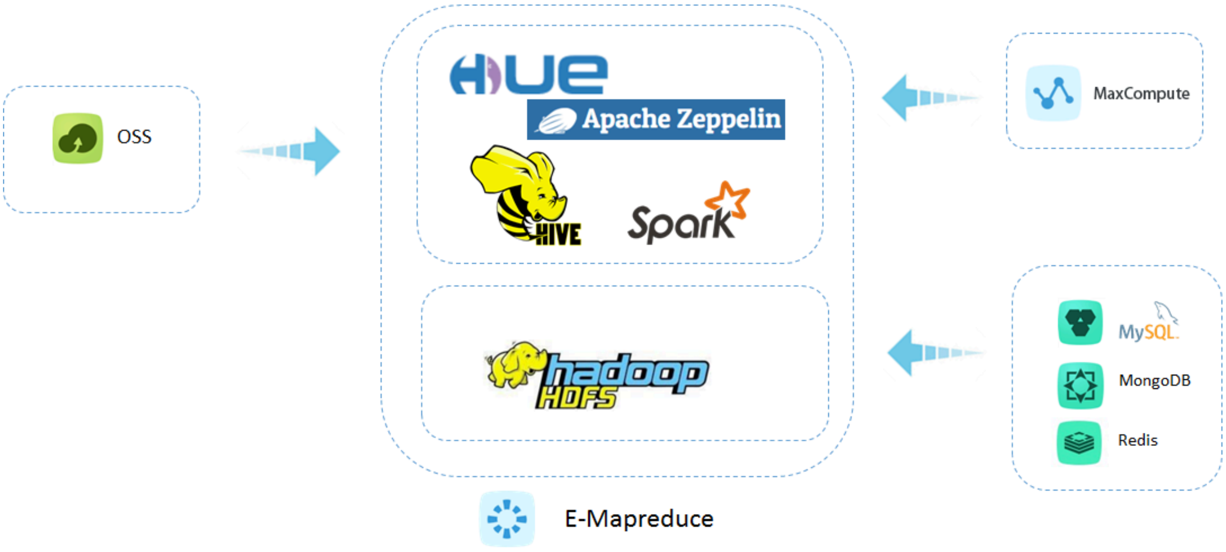
E-MapReduce clusters can be used in various scenarios, as E-MapReduce supports all scenarios that Apache Hadoop and Spark support.

Because E-MapReduce is based on Hadoop and Spark clusters, you can use your Alibaba Cloud ECS host as your own physical host. The following figures detail some typical application scenarios of E-MapReduce.

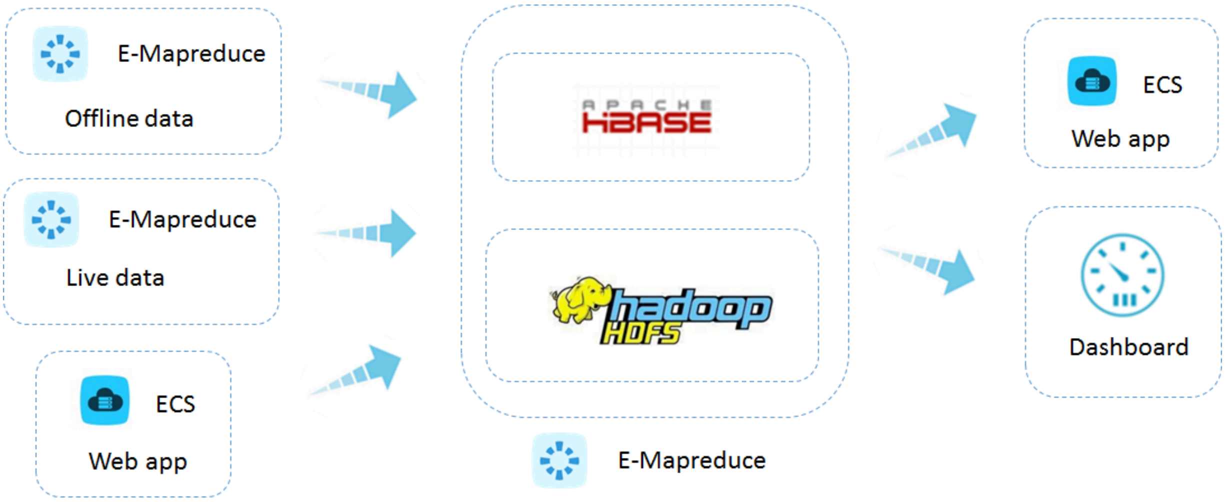
Offline data processing



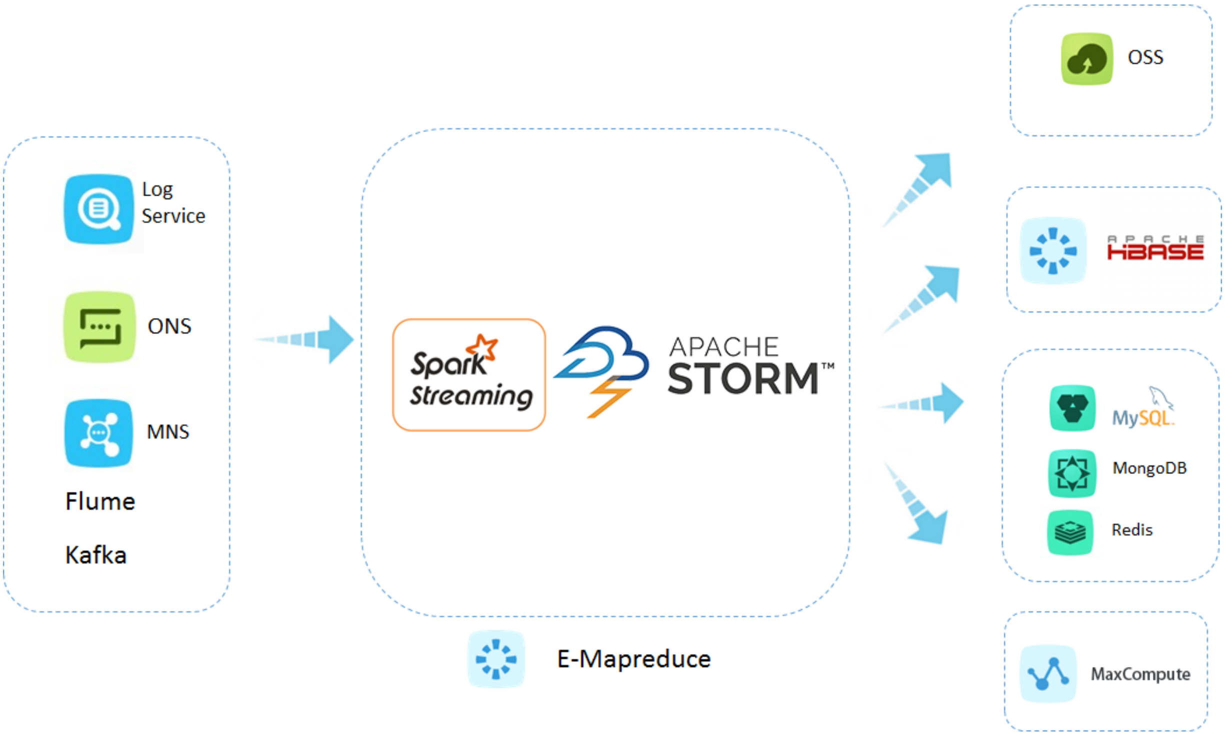
Ad-hoc data analysis



## Online services for massive amounts of data



## Stream data processing


Log Service
ONS
MNS
Flume
Kafka
Spark Streaming
APACHE STORM™
E-Mapreduce
OSS
APACHE HBASE
MySQL
MongoDB
Redis
MaxCompute