

# Alibaba Cloud E-MapReduce Data Development

**Issue: 20191101**

## 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, consequent**









ial, 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 contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, 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 this document 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

.



## Document conventions

Style	Description	Example
	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 <b>Danger:</b> Resetting will result in the loss of user configuration data.
	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 <b>Warning:</b> Restarting will cause business interruption. About 10 minutes are required to restart an instance.
	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	 <b>Notice:</b> If the weight is set to 0, the server no longer receives new requests.
	A note indicates supplemental instructions, best practices, tips, and other content.	 <b>Note:</b> You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click Settings > Network > Set network type.
<b>Bold</b>	<b>Bold formatting is used for buttons, menus, page names, and other UI elements.</b>	Click <b>OK</b> .
Courier font	<b>Courier font is used for commands.</b>	Run the <code>cd /d C:/window</code> command to enter the Windows system folder.
<i>Italic</i>	<b>Italic formatting is used for parameters and variables.</b>	<code>bae log list --instanceid</code> <code>Instance_ID</code>
[ ] or [a b]	<b>This format is used for an optional value, where only one item can be selected.</b>	<code>ipconfig [-all -t]</code>

Style	Description	Example
<b>{}</b> or <b>{a b}</b>	<b>This format is used for a required value, where only one item can be selected.</b>	switch { <i>active</i>   <i>stand</i> }



# Contents

---

<b>Legal disclaimer</b> .....	<b>I</b>
<b>Document conventions</b> .....	<b>I</b>
<b>1 Manage a workflow project</b> .....	<b>1</b>
<b>2 Edit a job</b> .....	<b>3</b>
<b>3 Ad hoc queries</b> .....	<b>9</b>
<b>4 Edit a workflow</b> .....	<b>12</b>
<b>5 Cluster template</b> .....	<b>15</b>
<b>6 Event codes for CloudMonitor</b> .....	<b>16</b>
<b>7 Jobs</b> .....	<b>18</b>
7.1 Job operations.....	18
7.2 Time and date variables.....	19
7.3 Configure a Hive SQL job.....	20
7.4 Configure a Hadoop MapReduce job.....	22
7.5 Configure a Hive job.....	24
7.6 Configure a Pig job.....	25
7.7 Configure a Spark job.....	27
7.8 Configure a Spark SQL job.....	29
7.9 Configure a Shell job.....	30
7.10 Configure a Sqoop job.....	31
7.11 Configure a Flink job.....	32
7.12 Configure a Spark Streaming job.....	33
7.13 Configure Streaming SQL jobs.....	35
<b>8 Old EMR Scheduling (Soon will be unavailable)</b> .....	<b>39</b>
8.1 Notebooks.....	39
8.1.1 Introduction.....	39
8.1.2 Operations.....	45
8.1.3 Examples.....	54
8.1.3.1 Query bank employee information.....	54
8.1.3.2 Video playback data.....	54
8.2 Execution plans.....	57
8.2.1 Create an execution plan.....	57
8.2.2 Manage an execution plan.....	60
8.2.3 Execution plan list.....	61
8.2.4 View job results and logs.....	62
8.2.5 Parallel execution of multiple execution plans.....	64
8.3 Create a job.....	65



# 1 Manage a workflow project

---

After creating an E-MapReduce cluster, you can create workflow projects so that multiple jobs can be run simultaneously or sequentially.

Create a project

1. At the top of the page, click the Data Platform tab to enter the Projects page.

Under the master account, you can view all of its projects and RAM user accounts. RAM users can only view projects if they have development permissions. The granting of project development permissions must be configured in the master account. For more information about authorization, see [User management](#) below.

2. In the upper-right corner, click New Project. The New Project dialog box is displayed.
3. Enter the project name and description and click Create.



**Note:**

You can only create a project with the master account. New Projects are only visible to the administrator of the master account.

User management

After creating a new project, you can grant operational permissions for the project to RAM user accounts.

1. In the Project List page, click View Details in the Actions column.
2. Click the User Management tab.
3. Click Add User to add RAM users to the project under the master account.

The added RAM users become members of the project and are able to view and develop the jobs and workflows under the project. If you remove a RAM user from a project, click Delete in the Actions column.



**Note:**

You can only add project members with the master account. The User Management tab is only visible to the administrator of the master account.

## Associate clusters

After creating a new project, you need to associate it with a cluster so that the workflow in the project can run on it.

1. In the Projects page, click **View Details** in the Actions column.
2. Click the **Cluster Settings** tab.
3. Click **Add Cluster**. From the drop-down menu, you can select a **Subscription** or **Pay-As-You-Go** cluster. (Clusters created by temporary jobs are not listed here.)
4. Click **OK**.

To disassociate the cluster, click **Delete** in the Operation column.



### Note:

You can only associate cluster with the master account. The Cluster Settings tab is only visible to the administrator of the master account.

To set both the queue and user to submit jobs to the cluster, click **Modify Configuration** in the Operation column. The configuration items are as follows:

- **Default Submit Job User:** Sets the default Hadoop user who submits the job to the selected cluster in the project. The default value is `hadoop`. There can only be one default user.
- **Default Submit Job Queue:** Sets the default queue that the jobs are submitted to in the project. If you leave this blank, the job will be submitted to the default queue.
- **Submit Job User Whitelist:** Sets Hadoop users who can submit jobs to the selected cluster in the project. If there is more than one user, they can be separated by a comma (,).
- **Submit Job Queue Whitelist:** Sets the queue of the selected cluster that jobs in the project can run in. If there is more than one queue, they can be separated by a comma (,).
- **Client whitelist:** Configures the client that can submit jobs. You can select either the E-MapReduce master node or the E-MapReduce gateway. Gateways that you have built are not currently listed here.

## 2 Edit a job

---

In a project, you can create Shell, Hive, Hive SQL, Spark, SparkSQL, MapReduce, Sqoop, Pig, Spark Streaming, and Flink jobs.

Create a job

1. Log on to the [Alibaba Cloud E-MapReduce console](#).
2. Click the Data Platform tab to enter the Projects page.
3. Click Workflows in the Actions column. Click Edit Job in the left-side navigation pane to go to the Edit Job page.
4. In the left-side navigation pane, right-click a folder as required and select Create Job from the drop-down list.
5. In the Create Job dialog box, enter a name and description for the job and select a job type.

Once selected, the job type cannot be modified.

6. Click OK.



Note:

You can also right-click the folder to create a subfolder, rename the folder, and delete the folder.

Develop a job

For more information, see the [Configure a Hadoop MapReduce job](#) section in *EMR User Guide*

.



Note:

If you choose `ossref://` as the OSS path prefix, the file that is stored in the OSS path is downloaded to the local machine and the local path is added to the value of the Classpath parameter.

- **Basic job settings**

In the upper-right corner, click **Job Settings**. The **Job Settings** dialog box appears.

- **Retries:** sets the number of retries when this job fails during the workflow running. This option will not take effect when you run the job on the **Edit Job** page.
- **Actions on Failures:** sets whether to continue running the next job or suspend the current workflow when this job fails during the workflow running.
- **Resources:** If you want to add resources such as jar packages or UDF that a job execution depends on, you must upload these files to OSS. When you select a resource, you can use this resource in a job directly.
- **Configuration Parameters:** specifies the values of variables used in a job. You can reference variables in your code. The format is: `${variable name}`. Click the plus (+) icon on the right side to add key-value pairs. Key is the name of a variable and value is the value of a variable. In addition, you can follow these rules to customize time variables according to the start time of a schedule.
  - **yyyy** represents a 4-digit year.
  - **MM** represents a 2-digit month.
  - **dd** represents a 2-digit day.
  - **HH** indicates that the 24-hour clock is used. **hh** indicates that the 12-hour clock is used.
  - **mm** represents a 2-digit minute.
  - **ss** represents a 2-digit second.

A time variable consists of the combination of a 4-digit year and one or more other time formats. In addition, you can use **plus (+)** and **minus (-)**

to add or reduce a period of time for the current time. For example, the `${yyyy-MM-dd}` variable represents the current date.

- One year after the current date can be represented as `${yyyy+Ny}` or `${yyyy-MM-dd hh:mm:ss+1y}`.
- Three months after the current date can be represented as: `${yyyyMM+Nm}` or `${hh:mm:ss yyyy-MM-dd+3m}`.
- Five days before the current date can be represented as: `${yyyyMMdd-Nd}` or `${hh:mm:ss yyyy-MM-dd-5d}`.



#### Notice:

The parameter of a time variable is required to start with yyyy. For example, `${yyyy-MM}`. If you want to obtain the values based on a specific period such as a month, you can use the following functions in a job.

- `parseDate(<parameter name>, <time format>)`: you can use this function to convert a specified parameter to a Date object. A parameter name represents the variable (key) name set in the Parameter Configuration area. A time format represents the time format used by the variable name. For example, if the parameter name is `${yyyyMMddHHmmss-1d}`, the time format is `yyyyMMddHHmmss`.
- `formatDate(<Date object>, <time format>)`: you can use this function to convert a specified Date object to a time format string.

#### Examples:

- To retrieve the hour literal value of the `current_time` variable: `${formatDate(parseDate(current_time, 'yyyyMMddHHmmss'), 'HH')}`
- To retrieve the year literal value of the `current_time` variable: `${formatDate(parseDate(current_time, 'yyyyMMddHHmmss'), 'yyyy')}`

- **Advanced job settings**

In the Job Settings dialog box, click the Advanced Settings tab.

- **Mode: Worker Node and Header/Gateway Node.**

- In the Worker Node mode, jobs are submitted to resources that are allocated by YARN as launchers.

- In the Header/Gateway Node mode, jobs are running on allocated nodes as processes.

- **Environment variables: the environment variables used to run the job. You can also export environment variables in the job script.**

For example, the content in a Shell job is `echo ${ENV_ABC}`. You set `ENV_ABC=12345` for the environment variable. A result of 12345 is returned by using the `echo` command. If the content of the job is `java -jar abc.jar` and the content of the `abc.jar` file is as follows:

```
public static void main(String[] args) {System.out.println(System.getEnv("ENV_ABC"));}
```

A result of 12345 is returned.

Configuring the environment variable has the same effect of executing the following script.

```
export ENV_ABC=12345
java -jar abc.jar
```

- **Scheduling parameters: includes information, such as YARN queues of a job, vCPU, memory, and Hadoop user. If you do not specify these parameters, a job uses the default values of the Hadoop cluster.**

#### Description

**Job submission supports the Worker Node and Header/Gateway Node modes. [DO NOT TRANSLATE]**

- **Header/Gateway: the spark-submit process is run on the header node. The spark-submit process requests much memory. A large number of jobs consume many resources of the header node, which causes an unstable cluster.**
- **Worker: the spark-submit process is run on a worker node and is allocated to a container of YARN, which This can alleviate the resource usage of header nodes.**

The **spark-submit** process (LAUNCHER in the data development module) is a Spark job submission command used to submit Spark jobs, typically occupying more than 600 MB of memory.

The **Memory (MB)** scheduling parameter in Job Settings is used to set the memory size allocated to LAUNCHER.

A complete Spark JOB includes: **spark-submit** (memory consumption: 600 MB) + **driver** (memory consumption: depending on the specific JOB, either JOB or LAUNCHER) + **executor** (memory consumption: depending on the specific JOB implementation, JOB)

- If Spark uses **yarn-client** mode, **spark-submit + driver** is in the same process (memory consumption: 600 MB + driver ). By using the **Header/Gateway Node** mode, the script is run on the header node and is not monitored by YARN. By using the **Worker Node** mode, the script is run on a worker node and monitored by YARN.
- If Spark uses the **YARN cluster** mode, the driver runs a separate process and occupies a container of YARN. The driver and **spark-submit** are not in the same process.

Whether a script is run on the header node or a worker node depends on the job submission mode (**Header/Gateway Node** or **Worker Node**). Whether the driver is launched within the **spark-submit** process depends on the deploy mode (**yarn-client** or **yarn-cluster**).

Execute a job

After the development and configuration of a job are complete, you can click **Run** in the upper-right corner to run the job.

View logs

After you execute a job, you can view the running logs on the **Records** tab page at the bottom of the page. Click **Details** to go to the details page. On this page, you can view details, including the job submission log and **YARN Container** log.

## FAQs

- **Insufficient disk capacity caused by many logs of streaming jobs**

For streaming jobs such as Spark Streaming jobs, we recommend that you enable log rolling to avoid insufficient disk capacity caused by many logs of long-running jobs. Perform the following steps.

1. In the *E-MapReduce console*, choose **Data Development > Project ID > Edit Job > Job Settings > Advanced Settings**.
2. In the **Environment Variables** section, click the plus sign (+) to add the following environment variable.

```
FLOW_ENABLE_LOG_ROLLING = true
```

3. **Save and restart the job.**

**Note:**

If you want to delete the log of a job without restarting the job, you can use the `echo > /path/to/log/dir/stderr` command to delete the job.



## 3 Ad hoc queries

---

You can only select HiveSQL, SparkSQL, and Shell as the type of an ad hoc query.

When you execute an ad hoc query statement, the log and query results show at the bottom of the Query page.

### Create a job

When you execute a job on the Edit Jobs page and click Details, you will be directed to the Details page that shows the operation logs and run logs of this job. Ad hoc queries and jobs are used in different places. Ad hoc queries are usually used by data scientists and data analysts. In addition, you need to use SQL as a tool to implement an ad hoc query.

1. Log on to the [Alibaba Cloud E-MapReduce console](#).
2. Click the Data Platform tab to enter the Projects page.
3. Click Design Workflow on the right side of the associated project to enter the Edit Jobs page.
4. In the left-side navigation pane, click the Query tab to enter the Query page.
5. In the left-side navigation pane, right-click a folder as required and select New Job.
6. In the New Job dialog box, enter the job name and job description and select a job type.

The job type cannot be modified once the job has been created.

7. Click OK.



#### Note:

You can right click on a folder and then select the corresponding option to perform New Subfolders, Rename Folder, and Delete Folder operations.

### Develop a job

For more information about how to develop jobs with HiveSQL, SparkSQL, and Shell types, see the *jobs* section of E-MapReduce user guide.



#### Note:

When you insert an OSS UNI and select OSSREF as a File Prefix, E-MapReduce will download OSS files to your cluster and add these files to the classpath.

- **Basic job settings**

In the top-right corner, click **Configure Jobs**, and then the **Job Settings** dialog box appears.

- **Resource File:** If you want to add resources such as jar packages or UDF that a job execution depends on, you must upload these files to OSS. When you select a resource, you can use this resource in a job directly.
- **Parameter Configuration:** specifies the values of variables used in a job. You can use variables in your code. The format is:  $\${variable\ name}$ . Click the plus (+) icon on the right side to add key-value pairs. Key is the name of a variable and value is the value of a variable. In addition, you can follow these rules to customize time variables according to the start time of a schedule.

- **yyyy** represents a 4-digit year.

- **MM** represents a 2-digit month.

- **dd** represents a 2-digit day.

- **HH** indicates that the 24-hour clock is used. **hh** indicates that the 12-hour clock is used.

- **mm** represent a 2-digit minute.

- **ss** represents a 2-digit second.

A time variable consists of the combination of a 4-digit year and one or more other time formats. In addition, you can use plus (+) and minus (-) to add or reduce a period of time for the current time. For example, the  $\${yyyy-MM-dd}$  variable represents the current date.

- One year after the current date can be represented as  $\${yyyy+Ny}$  or  $\${yyyy-MM-dd\ hh:mm:ss+1y}$ .

- Three months after the current date can be represented as:  $\${yyyyMM+Nm}$  or  $\${hh:mm:ss\ yyyy-MM-dd+3m}$ .

- Five days before the current date can be represented as:  $\${yyyyMMdd-Nd}$  or  $\${hh:mm:ss\ yyyy-MM-dd-5d}$ .



**Notice:**

The parameter of a time variable is required to start with yyyy. For example, `#{yyyy-MM}`. If you want to obtain the values based on a specific period such as a month, you can use the following functions in a job.

■ `parseDate(<parameter name>, <time format>)`: You can use this function to convert a specified parameter to a Date object. A parameter name represents the variable (key) name set in the Parameter Configuration area. A time format represents the time format used by the variable name. For example, if the parameter name is `#{yyyyMMddHHmmss-1d}`, the time format is `yyyyMMddHHmmss`.

■ `formatDate(<Date object>, <time format>)`: You can use this function to convert a specified Date object to a time format string.

**Examples:**

■ To retrieve the hour literal value of the `current_time` variable: `#{formatDate(parseDate(current_time, 'yyyyMMddHHmmss'), 'HH')}`

■ To retrieve the year literal value of the `current_time` variable: `#{formatDate(parseDate(current_time, 'yyyyMMddHHmmss'), 'yyyy')}`

- Advanced job settings

In the Job Settings dialog box, click the Advanced tab.

- **Mode: Submit on Worker Node and Submit on Header/Gateway Node.** In the Submit on Worker Node mode, jobs are submitted to resources that are allocated by YARN as launchers. In the Submit on Header/Gateway Node mode, jobs are running on allocated nodes as processes.
- **Scheduling Parameters:** YARN scheduler, CPU and memory specifications, and Hadoop users. Default values of a Hadoop cluster are applied if you do not specify the parameter values.

Execute a job

After the development and configuration of a job are complete, you can click Run in the top-right corner to run the job.

View logs

After you execute a job, you can view run logs on the Log tab at the bottom of the Query page.

## 4 Edit a workflow

---

An E-MapReduce workflow can be represented as a directed acyclic graph (DAG). You can pause, stop, and resume workflows. You can also view the running status of workflows in the web UI.

### Create a workflow

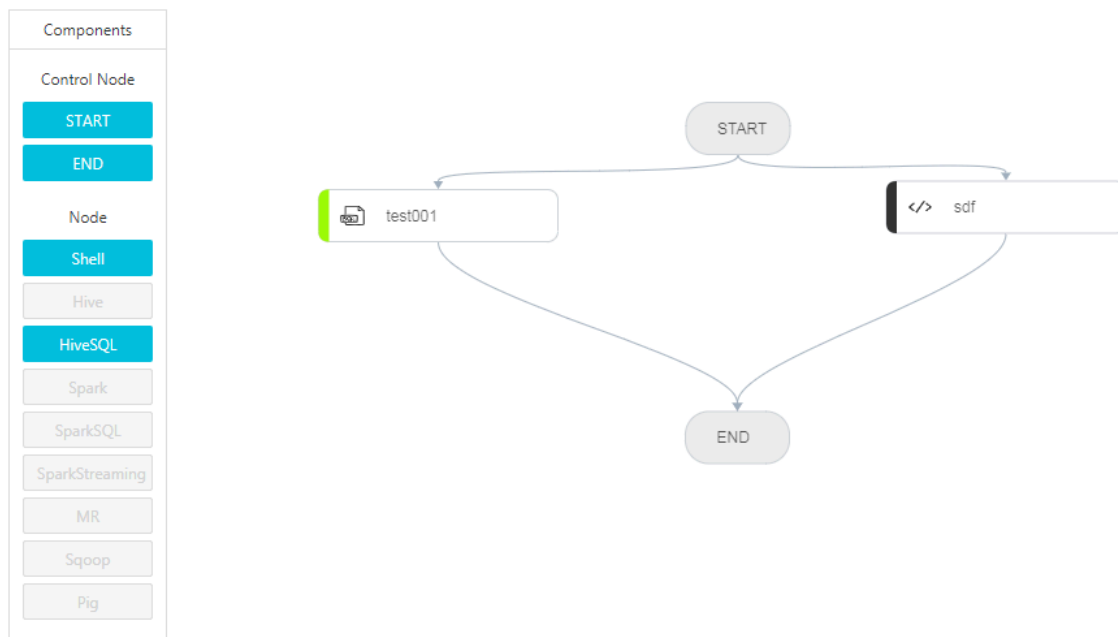
1. Log on to the [Alibaba Cloud E-MapReduce console](#).
2. Click the Data Platform tab to go to the Projects page.
3. Click Workflows in the Actions column for a project. Click the Workflows tab and go to the Workflows page.
4. Right-click the target folder and click Create Workflow.
5. In the Create Workflow dialog box, enter a name and description for the workflow. Select a cluster from the Target Cluster drop-down list.

You can select an existing cluster (Subscription or Pay-As-You-Go) that is associated with the project to run the workflow, or you can use a cluster template to create a temporary cluster for running the workflow.

6. Click OK.

### Edit a workflow

You can drag and drop multiple types of job nodes on the Workflows canvas and connect the nodes to schedule jobs. After the jobs are scheduled, drag the END controller node, drop it on the canvas, and connect it to a job node to complete the workflow design.



Configure a workflow

**In the upper-right corner of the Workflows canvas, click Configure to schedule a workflow.**

- **Target Cluster**

**Select a cluster for running the jobs in the workflow. From the Target Cluster drop-down list, select Select Existing Cluster or Create Cluster from Template.**

- **Select Existing Cluster:** When the workflow is executed, the jobs run on the existing cluster that you have selected.
- **Create Cluster from Template:** When the workflow is executed, the jobs run on the temporary cluster created by using the *Cluster template* that you have selected. The cluster is released when the workflow finishes.
- **Scheduling:** When scheduling is enabled, time-based scheduling is applied by default. You can also configure rules for dependency-based scheduling.
  - **Time-based Scheduling:** Sets a start time, an end time, and a cycle for scheduling the workflow. During this period, the workflow is executed according to the cycle you have set.
  - **Dependency-based Scheduling:** Selects a project and selects a dependent workflow. The workflow is scheduled only when the dependent workflow finishes. You can select a maximum of one dependent workflow.

- **Configure alerts**

Alerts can be sent by SMS, email, and DingTalk group. You can configure the following alerting rules:

- **Execution Failed:** alerts that are sent when a workflow fails.
- **Actions on Failures:** alerts that are sent when job nodes of a workflow fail.
- **Executed:** notifications that are sent when a workflow succeeds.
- **Action on Startup Timeout:** When a job node fails to start 30 minutes after it is assigned to a cluster, an alert is sent and the job is canceled.

Run a workflow

After the design and configuration are complete, click Run to run the workflow.

View and operate a workflow instance

After you run the workflow, click the Records tab page to view the running status of the workflow instance. Click Details for the workflow instance to view the running status of the workflow instance. You can also pause, resume, stop, or rerun the workflow instance.

Home Page > Data Platform > Project (FP-19DF27ECF61E5AF1) > View Records > Workflow Instance (FI-EC6641ECF29D5E2F) > View Details

Cluster Information Chart

ID: FI-EC6641ECF29D5E2F	Name: test
Workflow ID: F-798FE400F5485142	Run In: C-555D28C90826D188
Status: <b>FAILED</b>	Run Time: 11 Seconds
Start Time: 2018-11-12 17:08:42	End Time: 2018-11-12 17:08:53

Name  Enter

Job Instance ID	Name	Run In	Type	Host	Start Time	Completed At	Run Time	Execution State	Actions
FNI-7C04F5769E6802FB	test001	C-555D28C90826D188	HIVE_SQL	emr-header-1.cluster-84010	2018-11-12 17:08:43	2018-11-12 17:08:53	10 Seconds	<b>FAILED</b>	<a href="#">View Details</a>
FNI-05CD626FF3B17CC9	sdf	C-555D28C90826D188	SHELL	emr-header-1.cluster-84010	2018-11-12 17:08:44	2018-11-12 17:08:53	9 Seconds	<b>OK</b>	<a href="#">View Details</a>

- **Pause Workflow:** The job node that is running continues. The subsequent job nodes do not start. You can click Resume Workflow to start the subsequent job nodes again.
- **Stop Workflow:** All running job nodes stop running.
- **Rerun Workflow:** The workflow runs from the START node.

## 5 Cluster template

---

A cluster template is a saved configuration for quickly creating a cluster. Currently, cluster templates are used to create test clusters for workflows in Data Development. More scenarios will be supported.

Choose [Alibaba Cloud E-MapReduce console](#) > Data Platform > Cluster Template.

### Scenarios

Creating a cluster template is similar to creating a cluster. On the Basic Settings page, you need to specify a cluster template name. In the template list, click Edit to modify a cluster template. After the modification is saved, it is immediately applied to related workflows. You can click Delete to delete a cluster template.



#### Notice:

Whether the cluster template is in use is not checked. After the cluster template is deleted, the workflows that use the cluster template fail.

For more information, see [Edit a workflow](#).

### Restrictions

- Cluster templates may not support the latest cluster type. Submit a ticket to Alibaba Cloud if you want to use cluster templates to create clusters of latest types.
- Currently, setting a cluster password is not supported.

## 6 Event codes for CloudMonitor

---

In the Event Monitoring module of CloudMonitor, you can subscribe to system events related to data mining of E-MapReduce (EMR). This allows you to send alerts such as mobile phone alerts when an event occurs.

The following table shows event codes and the description of each event code.

Code	Description	Type
EMR-110401002	Indicates that the workflow is complete.	FLOW
EMR-110401003	Indicates that the workflow is submitted.	FLOW
EMR-110401004	Indicates that the job is submitted.	FLOW
EMR-110401005	Indicates that the workflow node is started.	FLOW
EMR-110401006	Indicates that the status of the workflow node is checked.	FLOW
EMR-110401007	Indicates that the workflow node is complete.	FLOW
EMR-110401008	Indicates that the workflow node is stopped.	FLOW
EMR-110401009	Indicates that the workflow node is canceled.	FLOW
EMR-110401010	Indicates that the workflow is canceled.	FLOW
EMR-110401011	Indicates that the workflow is restarted.	FLOW
EMR-110401012	Indicates that the workflow is resumed.	FLOW
EMR-110401013	Indicates that the workflow is paused.	FLOW



<b>Code</b>	<b>Description</b>	<b>Type</b>
<b>EMR-110401014</b>	<b>Indicates that the workflow is stopped.</b>	<b>FLOW</b>
<b>EMR-110401015</b>	<b>Indicates that the workflow node failed.</b>	<b>FLOW</b>
<b>EMR-110401016</b>	<b>Indicates that the job failed.</b>	<b>FLOW</b>
<b>EMR-210401001</b>	<b>Indicates that the workflow failed.</b>	<b>FLOW</b>
<b>EMR-210401003</b>	<b>Indicates that the startup timeout for the workflow node has been exceeded.</b>	<b>FLOW</b>
<b>EMR-210401004</b>	<b>Indicates that the startup timeout for the job has been exceeded.</b>	<b>FLOW</b>

## 7 Jobs

---

### 7.1 Job operations

**You can create, clone, modify, and delete jobs.**

#### Job creation

**A new job can be created at any time. Currently, a job can only be used in the region where it is created.**

#### Job cloning

**Configurations that already exist for a job can be cloned. A cloned job can also only be used in the region where it is created.**

#### Job modification

**Before you can modify a job that needs to be added to an execution plan, you must first ensure that the execution plan is not running and that its periodic scheduling is not in progress.**

**Before you can modify a job that needs to be added to several execution plans, you must first ensure that none of the execution plans are running and that none of their periodic scheduling is in progress. Modifying a job may result in changes to all of the execution plans that use this job.**

**If you need to debug, we recommend that you perform cloning instead. After you debug, the original jobs in the execution plan are replaced.**

#### Job deletion

**As with modification, a job can only be deleted when the execution plan where the job is located is not running and its periodic scheduling is not in progress.**

## 7.2 Time and date variables

When you are creating a job, variable wildcards are supported in the job parameters for both time and date.

Variable wildcard format

The format of the variable wildcards supported by E-MapReduce is either `${dateexpr-1d}` or `${dateexpr-1h}`. For example, assuming the current date and time is `/04/27 12:08:01`:

- If `${yyyyMMdd HH:mm:ss-1d}` is displayed, the parameter wildcard is replaced with `20160426 12:08:01` when executed, which is the current date minus one day, and time accurate to the second.
- If `${yyyyMMdd-1d}` is displayed, the parameter wildcard is replaced with `20160426` when executed, which is the current date minus one day.
- If `${yyyyMMdd}` is displayed, the parameter wildcard is replaced with `20160427`, which is the current date.

`dateexpr` represents the standard format of expressing time. Time is therefore formatted according to this expression and is followed by the amount of time that you want to add or deduct, which can be written as N. For example, `${yyyyMMdd-5d}`, `${yyyyMMdd+5d}`, `${yyyyMMdd+5h}`, or `${yyyyMMdd-5h}`.



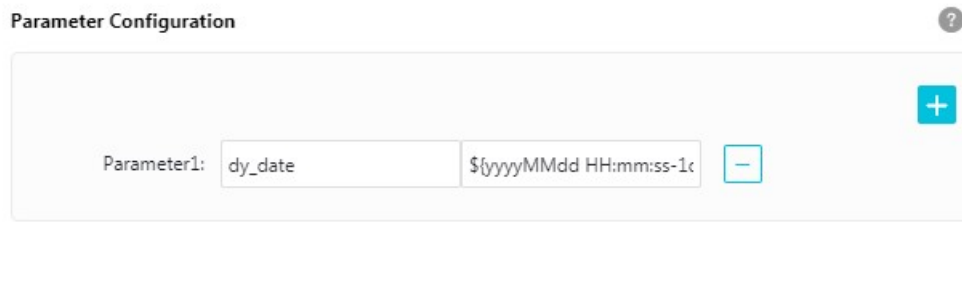
**Note:**

E-MapReduce currently supports the addition and deduction of hours and days only.

Example

1. Click Job Settings on the top right of the Edit Jobs page.

2. Click the add icon to add new parameters on the Parameter Configuration part, and fill in the parameter according to the Variable wildcard format that mentioned above.



3. You can now use the reference of the parameter key in the job editing.

## 7.3 Configure a Hive SQL job

This topic describes how to configure a Hive SQL job.

### Procedure

1. Log on to the [Alibaba Cloud E-MapReduce console](#) by using an Alibaba Cloud account.
2. Click the Data Platform tab to go to the Projects page.
3. Click Workflows in the Actions column of the project. Click Edit Jobs in the left-side navigation pane to go to the Edit Jobs page.
4. Right-click the folder based on which you want to create a job and click Create Job.

5. Enter a name and description. Select HiveSQL from the Job Type drop-down list. Hive SQL jobs are submitted by using the following command in the background of EMR.

```
hive -e {SQL CONTENT}
```

SQL CONTENT refers to the SQL statements that you enter in the job editor.

Create Job ×

---

\* Project:

\* Folder:

\* Name:

\* Description:

\* Job Type:

---

6. Click OK.

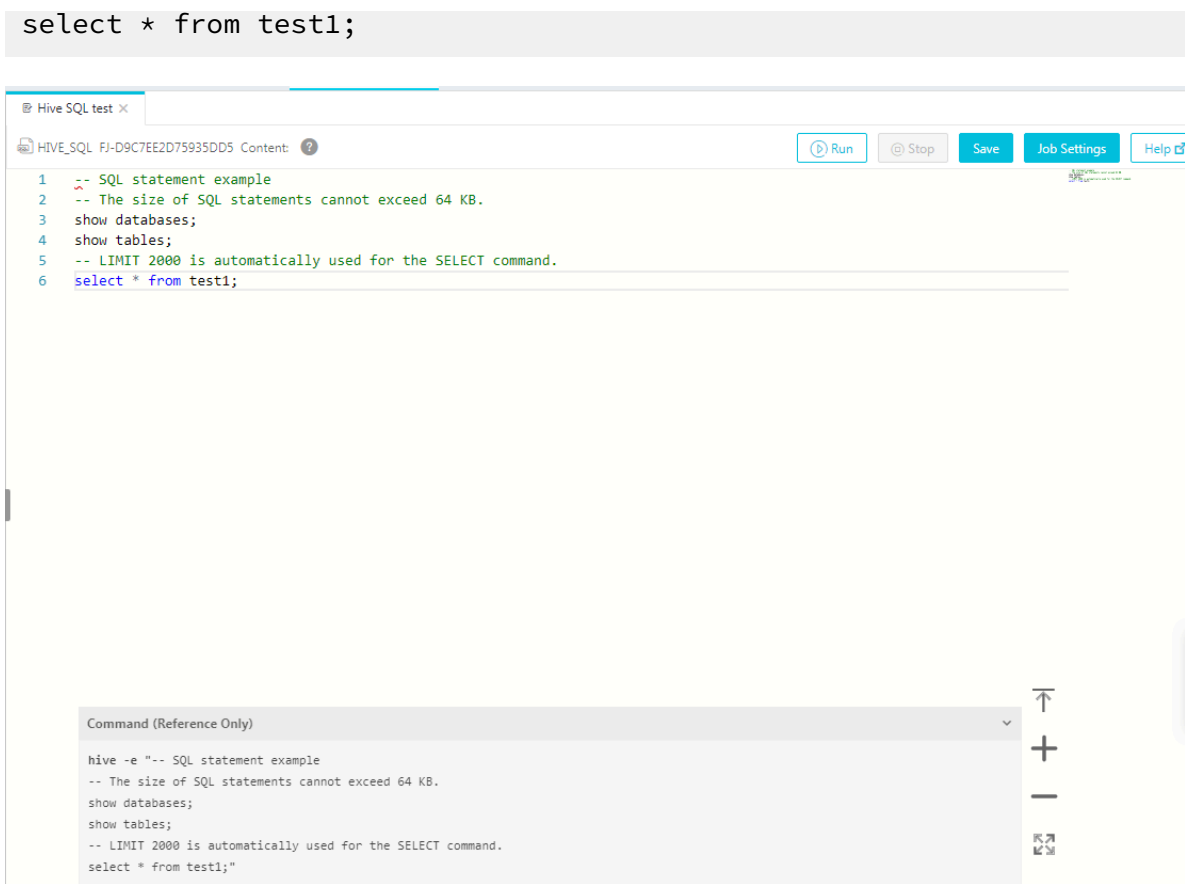


**Note:**

You can also right-click the folder to create a subfolder, rename the folder, and delete the folder.

7. In the Content field, enter Hive SQL statements. For example:

```
-- SQL statement example
-- The size of SQL statements cannot exceed 64 KB.
show databases;
show tables;
-- LIMIT 2000 is automatically used for the SELECT command.
```



8. Click Save to complete the configuration.

## 7.4 Configure a Hadoop MapReduce job

### Procedure

1. Log on to the [Alibaba Cloud E-MapReduce console](#).
2. At the top of the navigation bar, click Data Platform.
3. In the Actions column, click Design Workflow next to the specified project.
4. On the left of the Job Editing page, right-click the folder you want to operate and select New Job.
5. In the New Job dialog box, enter the job name and description.
6. Select a Hadoop job type to create a Hadoop MapReduce job. This type of job is submitted in the background using the following process.

```
hadoop jar xxx.jar [MainClass] -Dxxx ....
```

7. Click OK.

 **Note:**

You can also create subfolders, rename folders, and delete folders by right-clicking on them.

8. Enter the parameters in the Content field that are required to submit this job. Enter the parameters after the Hadoop jar, followed by other command line parameters.

For instance, if you want to submit a Hadoop sleep job that does not read or write any data, this will only succeed if you submit Mapper/Reducer tasks to the cluster and wait for each task to sleep for a while. In Hadoop, this job is packaged in the Hadoop release version's `hadoop-mapreduce-client-jobclient-2.6.0-tests.jar`. If this job is submitted from the command line, the command should read as follows.

```
hadoop jar /path/to/hadoop-mapreduce-client-jobclient-2.6.0-tests.jar sleep -m 3 -r 3 -mt 100 -rt 100
```

To configure this job in E-MapReduce, enter the following content in the Content field.

```
/path/to/hadoop-mapreduce-client-jobclient-2.6.0-tests.jar sleep -m 3 -r 3 -mt 100 -rt 100
```



**Note:**

The jar package path used here is an absolute path on the E-MapReduce host. However, the user may put these jar packages anywhere, and as clusters are created and released, the packages become unavailable. Therefore, upload the jar package by performing the following steps:

- a. Users send their own jar packages to the bucket of OSS for storage. When you configure the parameters for Hadoop, click Select OSS path to select and execute the jar package you want from the OSS directory. The system will then auto-complete the OSS address for jar packages. Be sure to switch the prefix of the jar to `ossref` by clicking Switch resource type. This ensures that the jar package is downloaded correctly by MapReduce.
- b. Click OK. The OSS path for this package will be auto-completed in the Content field. When a job is submitted, the system will find the corresponding jar packages automatically based on this path.
- c. Behind the jar package path for this OSS, other command line parameters for running jobs will be filled in further.

## 9. Click Save.

In the example above, the sleep job has no data input/output. If you want the job to read data and process input results, such as wordcounts, the data input and output paths need to be specified. You can read/write data on the HDFS of the E-MapReduce cluster as well as on OSS. To read/write data on OSS, write the data path as the OSS path when specifying the input and output paths. For instance:

```
jar ossref://emr/checklist/jars/chengtao/hadoop/hadoop-mapreduce-examples-2.6.0.jar randomtextwriter -D mapreduce.randomtextwriter.totalbytes=320000 oss://emr/checklist/data/chengtao/hadoop/Wordcount/Input
```

## 7.5 Configure a Hive job

When you apply for clusters in E-MapReduce, you are provided with a Hive environment by default. Using Hive, you can create and operate tables and data.

### Procedure

#### 1. Prepare the Hive script in advance. For example:

```
USE DEFAULT;
DROP TABLE uservisits;
CREATE EXTERNAL TABLE IF NOT EXISTS uservisits (sourceIP STRING,
destURL STRING,visitDate STRING,adRevenue DOUBLE,userAgent STRING
,countryCode STRING,languageCode STRING,searchWord STRING,duration
INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS
SEQUENCEFILE LOCATION '/HiBench/Aggregation/Input/uservisits';
DROP TABLE uservisits_aggre;
CREATE EXTERNAL TABLE IF NOT EXISTS uservisits_aggre (sourceIP
STRING, sumAdRevenue DOUBLE) STORED AS SEQUENCEFILE LOCATION '/
HiBench/Aggregation/Output/uservisits_aggre';
INSERT OVERWRITE TABLE uservisits_aggre SELECT sourceIP, SUM(
adRevenue) FROM uservisits GROUP BY sourceIP;
```

2. Save this script into a script file, such as `uservisits_aggre_hdfs.hive`, and upload it to an OSS directory (for example, `oss://path/to/uservisits_aggre_hdfs.hive`).
3. Log on to the [Alibaba Cloud E-MapReduce console](#).
4. At the top of the navigation bar, click Data Platform.
5. In the Actions column, click Design Workflow next to the specified project.
6. On the left of the Job Editing page, right-click the folder you want to operate and select New Job.
7. In the New Job dialog box, enter the job name and description.



8. **Select the Hive job type to create a Hive job. This type of job is submitted in the background using the following method.**

```
hive [user provided parameters]
```

9. **Click OK.**



**Note:**

**You can also create subfolders, rename folders, and delete folders by right-clicking on them.**

10. **Enter the parameters in the Content field after the Hive commands. For example, if you want to use a Hive script uploaded to OSS, enter the following.**

```
-f ossref://path/to/uservisits_aggre_hdfs.hive
```

**You can also click Select OSS path to view and select from OSS. The system will automatically complete the path of the Hive script on OSS. Switch the Hive script prefix to ossref by clicking Switch resource type. This ensures that the file is correctly downloaded by E-MapReduce.**

11. **Click Save to complete the Hive job configuration.**

## 7.6 Configure a Pig job

**When you apply for clusters in E-MapReduce, a Pig environment is provided by default. Using Pig, you can create and operate tables and data.**

Procedure

1. **Prepare the Pig script in advance. For example:**

```
```shell
/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements.  See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership.  The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License.  You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing,
software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
 * See the License for the specific language governing permissions
and
```

```

* limitations under the License.
*/
-- Query Phrase Popularity (Hadoop cluster)
-- This script processes a search query log file from the Excite
search engine and finds search phrases that occur with particular
high frequency during certain times of the day.
-- Register the tutorial JAR file so that the included UDFs can be
called in the script.
REGISTER oss://emr/checklist/jars/chengtao/pig/tutorial.jar;
-- Use the PigStorage function to load the excite log file into
the "raw" bag as an array of records.
-- Input: (user,time,query)
raw = LOAD 'oss://emr/checklist/data/chengtao/pig/excite.log.bz2'
USING PigStorage('\t') AS (user, time, query);
-- Call the NonURLDetector UDF to remove records if the query field
is empty or a URL.
clean1 = FILTER raw BY org.apache.pig.tutorial.NonURLDetector(query
);
-- Call the ToLower UDF to change the query field to lowercase.
clean2 = FOREACH clean1 GENERATE user, time, org.apache.pig.
tutorial.ToLower(query) as query;
-- Because the log file only contains queries for a single day, we
are only interested in the hour.
-- The excite query log timestamp format is YYMMDDHHMMSS.
-- Call the ExtractHour UDF to extract the hour (HH) from the time
field.
hoored = FOREACH clean2 GENERATE user, org.apache.pig.tutorial.
ExtractHour(time) as hour, query;
-- Call the NGramGenerator UDF to compose the n-grams of the query.
ngramed1 = FOREACH hoored GENERATE user, hour, flatten(org.apache.
pig.tutorial.NGramGenerator(query)) as ngram;
-- Use the DISTINCT command to get the unique n-grams for all
records.
ngramed2 = DISTINCT ngramed1;
-- Use the GROUP command to group records by n-gram and hour.
hour_frequency1 = GROUP ngramed2 BY (ngram, hour);
-- Use the COUNT function to get the count (occurrences) of each n
-gram.
hour_frequency2 = FOREACH hour_frequency1 GENERATE flatten($0),
COUNT($1) as count;
-- Use the GROUP command to group records by n-gram only.
-- Each group now corresponds to a distinct n-gram and has the
count for each hour.
uniq_frequency1 = GROUP hour_frequency2 BY group::ngram;
-- For each group, identify the hour in which this n-gram is used
with a particularly high frequency.
-- Call the ScoreGenerator UDF to calculate a "popularity" score
for the n-gram.
uniq_frequency2 = FOREACH uniq_frequency1 GENERATE flatten($0),
flatten(org.apache.pig.tutorial.ScoreGenerator($1));
-- Use the FOREACH-GENERATE command to assign names to the fields
.
uniq_frequency3 = FOREACH uniq_frequency2 GENERATE $1 as hour, $0
as ngram, $2 as score, $3 as count, $4 as mean;
-- Use the FILTER command to move all records with a score less
than or equal to 2.0.
filtered_uniq_frequency = FILTER uniq_frequency3 BY score > 2.0;
-- Use the ORDER command to sort the remaining records by hour and
score.
ordered_uniq_frequency = ORDER filtered_uniq_frequency BY hour,
score;
-- Use the PigStorage function to store the results.
-- Output: (hour, n-gram, score, count, average_counts_among
_all_hours)

```

```
STORE ordered_uniq_frequency INTO 'oss://emr/checklist/data/
chengtao/pig/script1-hadoop-results' USING PigStorage();
\`\`\`
```

2. Save this script into a script file, such as `script1-hadoop-oss.pig`, and upload it to an OSS directory (for example, `oss://path/to/script1-hadoop-oss.pig`).
3. Log on to the [Alibaba Cloud E-MapReduce console](#).
4. At the top of the navigation bar, click Data Platform.
5. In the Actions column, click Design Workflow next to the specified project.
6. On the left of the Job Editing page, right-click the folder you want to operate and select New Job.
7. In the New Job dialog box, enter the job name and description.
8. Select the Pig job type to create a Pig job. This type of job is submitted in the background using the following method.

```
pig [user provided parameters]
```

9. Click OK.

**Note:**

You can also create subfolders, rename folders, and delete folders by right-clicking on them.

10. Enter the parameters in the Content field after the Pig commands. For example, if you want to use a Pig script uploaded to OSS, enter the following.

```
-x mapreduce ossref://emr/checklist/jars/chengtao/pig/script1-hadoop
-oss.pig
```

You can click Select OSS path to view and select from OSS. The system will automatically complete the path of Pig script on OSS. Switch the Pig script prefix to `ossref` by clicking Switch resource type. This ensures that the file is correctly downloaded by E-MapReduce.

11. Click Save to complete the Pig job configuration.

## 7.7 Configure a Spark job

In this tutorial, you will learn how to configure a Spark job.

### Procedure

1. Log on to the [Alibaba Cloud E-MapReduce console](#).

2. At the top of the navigation bar, click Data Platform.
3. In the Actions column, click Design Workflow next to the specified project.
4. On the left of the Job Editing page, right-click the folder you want to operate and select New Job.
5. In the New Job dialog box, enter the job name and description.
6. Click OK.

**Note:**

You can also create subfolders, rename folders, and delete folders by right-clicking on them.

7. Select the Spark job type to create a Spark job. This type of job is submitted in the background using the following method.

```
spark-submit [options] --class [MainClass] xxx.jar args
```

8. Enter the parameters in the Content field that are required to submit this job. Only the parameters after `spark-submit` can be entered. The following example shows how to enter the parameters for creating a Spark job and a PySpark job.

- Create a Spark job

**Create a Spark WordCount job:**

- Job name: WordCount
- Type: Select Spark
- Parameters:

**■ Enter the following command:**

```
spark-submit --master yarn-client --driver-memory 7G --
executor-memory 5G --executor-cores 1 --num-executors 32 --
class com.aliyun.emr.checklist.benchmark.SparkWordCount emr
-checklist_2.10-0.1.0.jar oss://emr/checklist/data/wc oss://
emr/checklist/data/wc-counts 32
```

**■ Enter the following in the E-MapReduce job Content field:**

```
--master yarn-client --driver-memory 7G --executor-memory 5G
--executor-cores 1 --num-executors 32 --class com.aliyun.emr
.checklist.benchmark.SparkWordCount ossref://emr/checklist/
jars/emr-checklist_2.10-0.1.0.jar oss://emr/checklist/data/wc
oss://emr/checklist/data/wc-counts 32
```

**Notice:**

**Job jar packages are saved in OSS. In the example above, the way to reference the Jar package is `ossref://emr/checklist/jars/emr-checklist_2.10-0.1.0.jar`. Click **Select OSS path to view** and select one from OSS. The system will automatically complete the absolute path of the Spark script on OSS. Switch the default OSS protocol to the `ossref` protocol.**

- **Create a PySpark job**

In addition to Scala and Java job types, E-MapReduce also supports Python job types in Spark. Create a Spark K-means job for the Python script:

- **Job name:** Python-Kmeans
- **Type:** Spark
- **Parameters:**

```
--master yarn-client --driver-memory 7g --num-executors 10
--executor-memory 5g --executor-cores 1 --jars ossref://emr/
checklist/jars/emr-core-0.1.0.jar ossref://emr/checklist/python
/wordcount.py oss://emr/checklist/data/kddb 5 32
```

- **References of Python script resources are supported, and the `ossref` protocol is used.**
- **For PySpark, the online Python installation kit is not supported.**

9. Click **Save** to complete the Spark job configuration.

## 7.8 Configure a Spark SQL job

This topic describes how to configure a Spark SQL job.



### Note:

**Spark SQL uses `yarn-client` mode to submit jobs by default.**

### Procedure

1. Log on to the [Alibaba Cloud E-MapReduce console](#) and go to the **Cluster Management** page.
2. Click the **Data Platform** tab to go to the **Projects** page.
3. Click **Workflows** in the **Actions** column of the project. Click **Edit Job** in the left-side navigation pane to go to the **Edit Job** page.
4. Right-click the folder based on which you want to create a job and click **Create Job**.
5. Enter a name and description for the job.

6. Select SparkSQL from the Jop Type drop-down list. Spark SQL jobs are submitted in the background of EMR by using the following command.

```
spark-sql [options] [cli option]spark-sql [options] -e {SQL_CONTENT}
```

- **options:** click Job Settings, click the Advanced Settings tab, and then click the plus (+) icon in the Environment Variables section to add the SPARK\_CLI\_PARAMS parameter. For example, set the value of SPARK\_CLI\_PARAMS to --executor-memory 1g --executor-cores.
- **SQL\_CONTENT:** the SQL statements that you enter in the job editor.

7. Click OK.

**Note:**

You can also right-click a folder and create a subfolder, rename the folder, and delete the folder.

8. In the Contents field, enter Spark SQL statements. For example:

```
-- SQL statement example
-- The size of SQL statements cannot exceed 64 KB.
show databases;
show tables;
-- LIMIT 2000 is automatically used for the SELECT command.
select * from test1;
```

9. Click Save to complete the configuration of the Spark SQL job.

## 7.9 Configure a Shell job

In this tutorial, you will learn how to configure a Shell job.

**Notice:**

By default, Shell scripts are currently run by Hadoop. If you need to use the root user, the `sudo` command can be used. Use Shell script jobs with caution.

### Procedure

1. Log on to the [Alibaba Cloud E-MapReduce console](#).
2. At the top of the navigation bar, click Data Platform.
3. In the Projects area, select a target project ID to go to the Project Management tab page.
4. In the left-side navigation bar, click Edit Jobs next to the specified project.

5. On the left of the Edit Jobs tab page, right-click the folder you want to operate and select New Job.
6. In the New Job dialog box, enter the job name and description.
7. Select the Shell job type to create a Bash Shell job.
8. Click OK.

**Note:**

You can also create subfolders, rename folders, and delete folders by right-clicking on them.

9. Enter the parameters in the Content field after the Shell commands.
10. Click Save to complete Shell job configurations.

## 7.10 Configure a Sqoop job

In this tutorial, you will learn how to configure a Sqoop job.

**Note:**

Only E-MapReduce products with version V1.3.0 or later support the Sqoop job type. Running a Sqoop job on lower versions will fail and errlog will report "Not supported" errors. For more information on parameters, see [Sqoop](#).

### Procedure

1. Log on to the [Alibaba Cloud E-MapReduce console](#).
2. At the top of the navigation bar, click Data Platform.
3. In the Actions column, click Design Workflow next to the specified project.
4. On the left of the Job Editing page, right-click the folder you want to operate and select New Job.
5. In the New Job dialog box, enter the job name and description.
6. Select the Sqoop job type to create a Sqoop job. This type of job is submitted in the background using the following method.

```
sqoop [args]
```

7. Click OK.

**Note:**

You can also create subfolders, rename folders, and delete folders by right-clicking on them.

8. Enter the parameters in the Content field after the Sqoop commands.
9. Click Save to complete Sqoop job configuration.

## 7.11 Configure a Flink job

This topic describes how to configure a Flink job.

### Prerequisites

- You have created a project. For more information, see [Manage a workflow project](#).
- You have prepared the required resources and data to be processed, for example ,JAR files, data files, and storage paths for these resources.

### Procedure

1. Log on to the [E-MapReduce \(EMR\) console](#) by using an Alibaba Cloud account and open the Cluster Management tab.
2. Select the Data Platform tab to view the Projects section.
3. Click Workflows next to the target job. In the left-side navigation pane, select Edit Job.
4. On the left side of the Edit Job tab, right-click a folder and select Create Job.



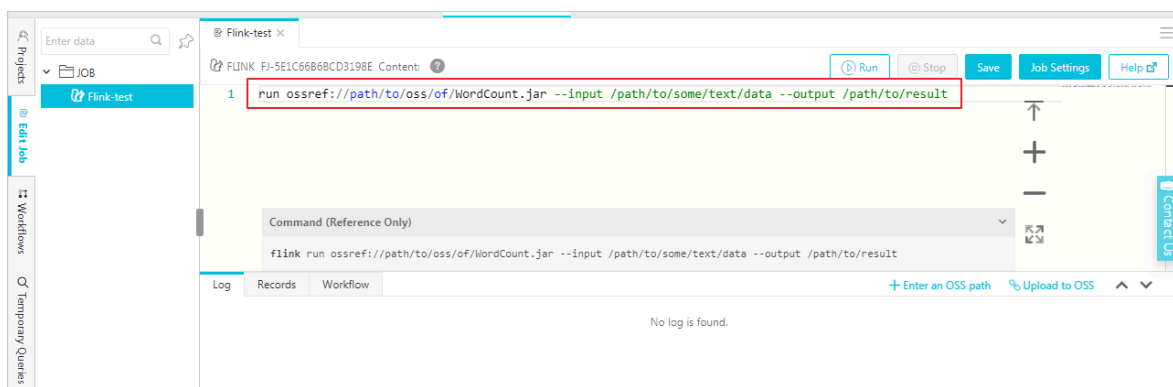
#### Note:

You can also select Create Subfolder, Rename Folder, or Delete Folder from the list.

5. In the Create Job dialog box, enter the Name and Description, and select Flink from the Job Type drop-down list.
6. After the configuration is complete, click OK to create a new job.



## 7. After you create a job, you need to specify content for the job.



The following is an example of job content.

```
run ossref://path/to/oss/of/WordCount.jar --input /path/to/some/text
/path/to/result
```



### Notice:

If a job is stored in OSS as a JAR file, you can use the following endpoint as an example to reference the JAR file: `ossref://<a directory name>/.../<a file name>.jar`. You can click **OSS Path** to select the JAR file from OSS. The full path of the JAR file that includes the Spark Streaming job will be added to the job content. You must change the default OSS protocol to the `ossref` protocol.

If you submit a Flink job in the EMR console, use a command with the following format.

- The following command is an example of how to submit a Flink job.

```
flink run [options] <a file name>.jar args
```

- In this example, use the following command to submit a Flink job.

```
flink run WordCount.jar --input /path/to/some/text/data --output /
path/to/result
```

## 8. After configuring the preceding parameters, click Save to finish the configuration of a Flink job.

## 7.12 Configure a Spark Streaming job

This topic describes how to configure a Spark Streaming job.

### Prerequisites

- You have created a project. For more information, see [Manage a workflow project](#).
- You have prepared all required resources and data to be processed.

## Procedure

1. Log on to the [E-MapReduce \(EMR\) console](#) by using an Alibaba Cloud account and open the Cluster Management tab.
2. Select the Data Platform tab in the top navigation bar and view the Projects section.
3. Click Workflows next to the target project. In the left-side navigation pane, select Edit Job.
4. On the left side of the Edit Job tab, right-click a folder and select Create Job.



### Note:

You can also select Create Subfolder, Rename Folder, or Delete Folder from the list.

5. In the Create Job dialog box, enter the Name and Description, and select Spark Streaming from the Job Type drop-down list.
6. After the configuration is complete, click OK to create a new job.
7. After creating a job, you must configure the job.

Take a job named SlsStreaming as an example. The job content is shown as follows.

```
--master yarn-client --driver-memory 7G --executor-memory 5G --  
executor-cores 1 --num-executors 32 --class com.aliyun.emr.checklist  
.benchmark.SlsStreaming emr-checklist_2.10-0.1.0.jar <project> <  
logstore> <accessKey> <secretKey>
```



### Notice:

If a job is stored in OSS as a JAR file, you can use the following endpoint as an example to reference the JAR file: `ossref://<a directory name>/.../<a file name>.jar`. You can also click OSS Path to select the JAR file from OSS. The full

path of the JAR file that includes the Spark Streaming job will be added to the job content. You must change the default OSS protocol to the ossref protocol.

If you submit a Spark Streaming job in the EMR console, use a command with the following format.

- The following command is an example of how to submit a Spark Streaming job.

```
spark-submit [options] --class [MainClass] <a file name>.jar args
```

- In this example, use the following command to submit a Spark Streaming job.

```
spark-submit --master yarn-client --driver-memory 7G --executor-memory 5G --executor-cores 1 --num-executors 32 --class com.aliyun.emr.checklist.benchmark.SlsStreaming emr-checklist_2.10-0.1.0.jar <project> <logstore> <accessKey> <secretKey>
```

8. After configuring the preceding parameters, click Save to finish the configuration of a Spark Streaming job.

## 7.13 Configure Streaming SQL jobs

This topic describes how to configure Streaming SQL jobs.

### Prerequisites

- You have created a project. For more information, see [Manage a workflow project](#).
- You have downloaded the Spark Streaming SQL dependent libraries. For more information, see [Background information](#).

### Context

For more information about Streaming SQL, see [Spark Streaming SQL](#).

When you configure a Streaming SQL job, you must specify dependent libraries. The following table lists the version and other detailed information about the dependent library provided by Spark Streaming SQL to support multiple data sources. In principle, you need to use the latest version of the dependent library.

Name	Version	Release date	Reference string	Details
<b>datasources-bundle</b>	<b>1.7.0</b>	<b>July 29, 2019</b>	<b>sharedlibs:streaming-sql:datasources-bundle:1.7.0</b>	<b>Supported data sources: Kafka, Log Service, Druid, Table Store, HBase, and JDBC.</b>

- **Copy the reference string to Job Settings > Streaming Task Settings > Dependent Libraries in Data Platform of the E-MapReduce console.**
- **The data sources specified in the preceding table all support reading and writing streams.**
- **For more information, see [Data sources](#).**

Step 1: Create a Streaming SQL job.

- 1. Log on to the [E-MapReduce console](#) by using an Alibaba Cloud account.**
- 2. Click the Data Platform tab. The Projects list is displayed.**
- 3. Click Workflows next to the target job. In the left-side navigation pane, select Edit Job.**
- 4. In the left-side Edit Job pane, right-click a folder and then select Create Job.**



**Note:**

**You can right-click a folder and then choose to create a sub-folder, rename the current folder, or delete the current folder.**

5. In the Create Job dialog box, enter the Name and Description, and select Streaming SQL from the Job Type drop-down list.

6. Click OK to create the Streaming SQL job.

The system then automatically opens the job for you. You can enter code into the job.

Step 2: Configure Streaming SQL statements for the job.

In the E-MapReduce background, Streaming SQL jobs are submitted in the form of `streaming-sql -f {SQL_SCRIPT}`. Streaming SQL statements are saved in `SQL_SCRIPT`.

After you create the job, enter Streaming SQL statements into the text box of the job.

Streaming SQL statement example:

```
--- Create a Log Service table
CREATE TABLE IF NOT EXISTS ${slsTableName}
  USING loghub
  OPTIONS (
    sls.project = '${logProjectName}',
    sls.store = '${logStoreName}',
    access.key.id = '${accessKeyId}',
    access.key.secret = '${accessKeySecret}',
    endpoint = '${endpoint}'
  );
--- Import data to HDFS
INSERT INTO
  ${hdfsTableName}
SELECT
  col1, col2
FROM  ${slsTableName}
```

```

WHERE ${condition}
    
```

```

1 --- Create an SLS data table
2 CREATE TABLE IF NOT EXISTS ${s1sTableName}
3 USING loghub
4 OPTIONS (
5   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
6   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
7   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
8   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
9   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
10  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
11 );
12 --- Import data into HDFS
13 INSERT INTO
14   ${hdfsTableName}
15 SELECT
16   col1, col2
17 FROM ${s1sTableName}
18 WHERE ${condition}
    
```

Step 3: Specify dependent libraries and actions on failures.

**Dependent libraries:** Streaming SQL jobs depend on data source libraries. E-MapReduce publishes these libraries to the repository of the scheduling center as dependent libraries. When you create a job, you need to specify dependent libraries for the job.

**Actions on failures:** Specify the action to be performed when E-MapReduce failed to execute the current statement.

1. After you enter statements into the job, click Job Settings in the upper-right corner, and then click the Streaming Task Settings tab.
2. On the Streaming Task Settings tab, specify the dependent libraries and actions on failures.

Section	Parameter	Description
Actions on Failures	Action on Current Statement Failure	When E-MapReduce failed to execute the current statement, you can choose to perform one of the following actions: <ul style="list-style-type: none"> <li>• <b>Execute Next Statement:</b> Execute the next statement.</li> <li>• <b>Terminate Job:</b> Terminate the job.</li> </ul>
Dependent Libraries	Libraries	Enter the reference strings of dependent libraries, for example, <i>sharedlibs:streamingsql:datasources-bundle:1.7.0</i> .

3. Click Save to complete configuring the Streaming SQL job.

## 8 Old EMR Scheduling (Soon will be unavailable)

---

### 8.1 Notebooks

#### 8.1.1 Introduction

Notebooks allow you to compile and run Spark, Spark SQL, and Hive SQL tasks directly on the E-MapReduce console. You can then view the running results in the notebook. Notebooks are ideal for processing debugging tasks that require a shorter runtime and whose results need to be viewed directly. For tasks that have a longer runtime and require regular execution, the job and execution plan function must be used. This section describes how to create and run a notebook demo task.

Create a demo task

1. Log on to the [Alibaba Cloud E-MapReduce console](#).
2. At the top of the navigation bar, click Old EMR Scheduling.
3. In the navigation bar on the left, click Notebook.
4. Click New notebook demo.

notebook list

EMR-Hive-Demo

EMR-SparkSQL-Demo

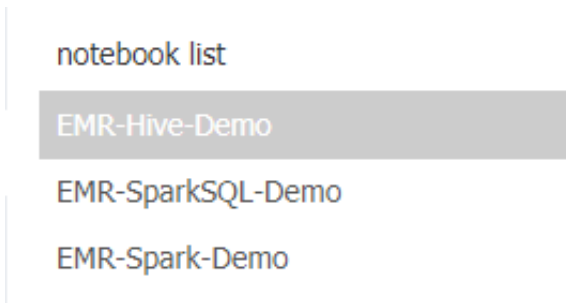
EMR-Spark-Demo

EMR-SparkSQL-Demo

EMR-Spark-Demo

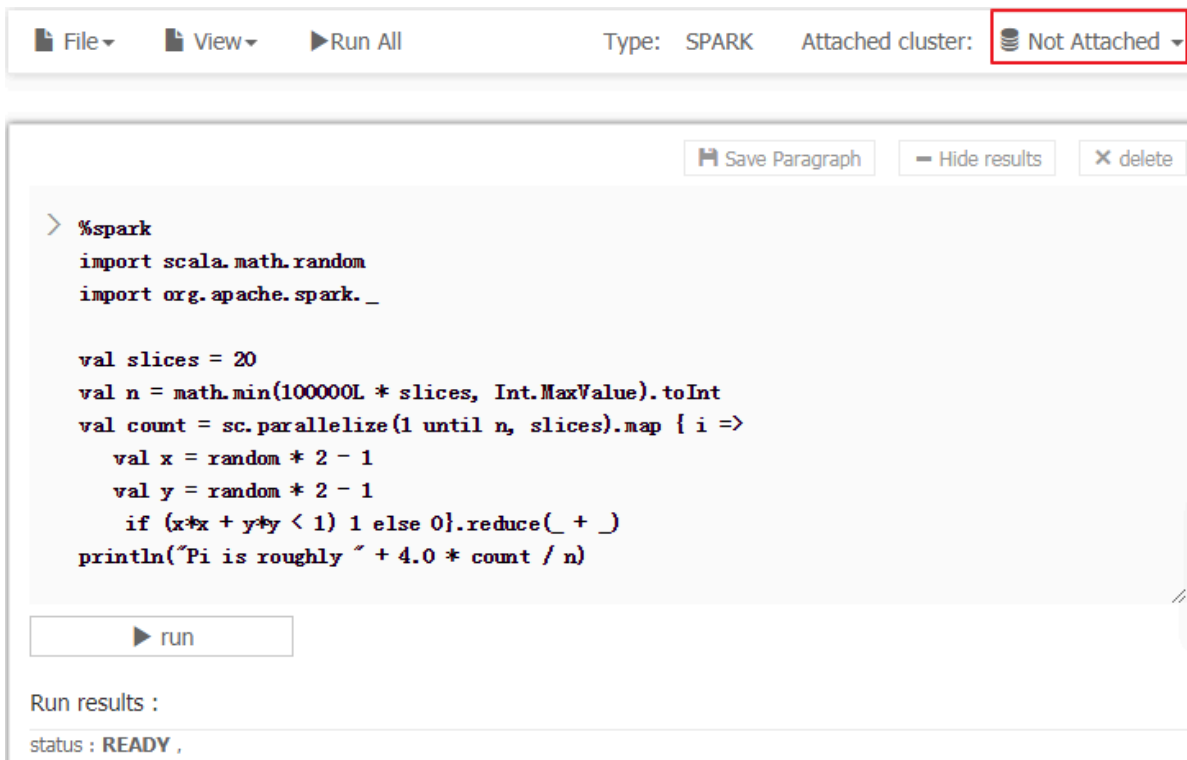
EMR-Hive-Demo

- 5. A confirmation box is displayed, indicating the required cluster environment. Click OK to create a demo task. Three examples of interactive tasks are created.



Run a Spark demo task

- 1. Click EMR-Spark-Demo to display the example of a Spark notebook. Before running the notebook, you need to associate the task to a created cluster. Select a created cluster in the list of available clusters. Note that the associated cluster must be E-MapReduce 2.3 or later and have no less than three nodes, each with at least 4 cores and 8 GB of memory.



- 2. After a cluster is associated, click Run. When the associated cluster executes the Spark or Spark SQL notebook for the first time, it takes about one minute to



build the Spark context and running environment. It does not need to be built in subsequent executions. The running result is displayed under the Run button.

[Save Paragraph](#)[Hide results](#)[delete](#)

```
> %spark
import scala.math.random
import org.apache.spark._

val slices = 20
val n = math.min(100000L * slices, Int.MaxValue).toInt
val count = sc.parallelize(1 until n, slices).map { i =>
  val x = random * 2 - 1
  val y = random * 2 - 1
  if (x*x + y*y < 1) 1 else 0}.reduce(_ + _)
println("Pi is roughly " + 4.0 * count / n)
```

A rectangular button with a red border, containing a right-pointing triangle icon followed by the text "run".

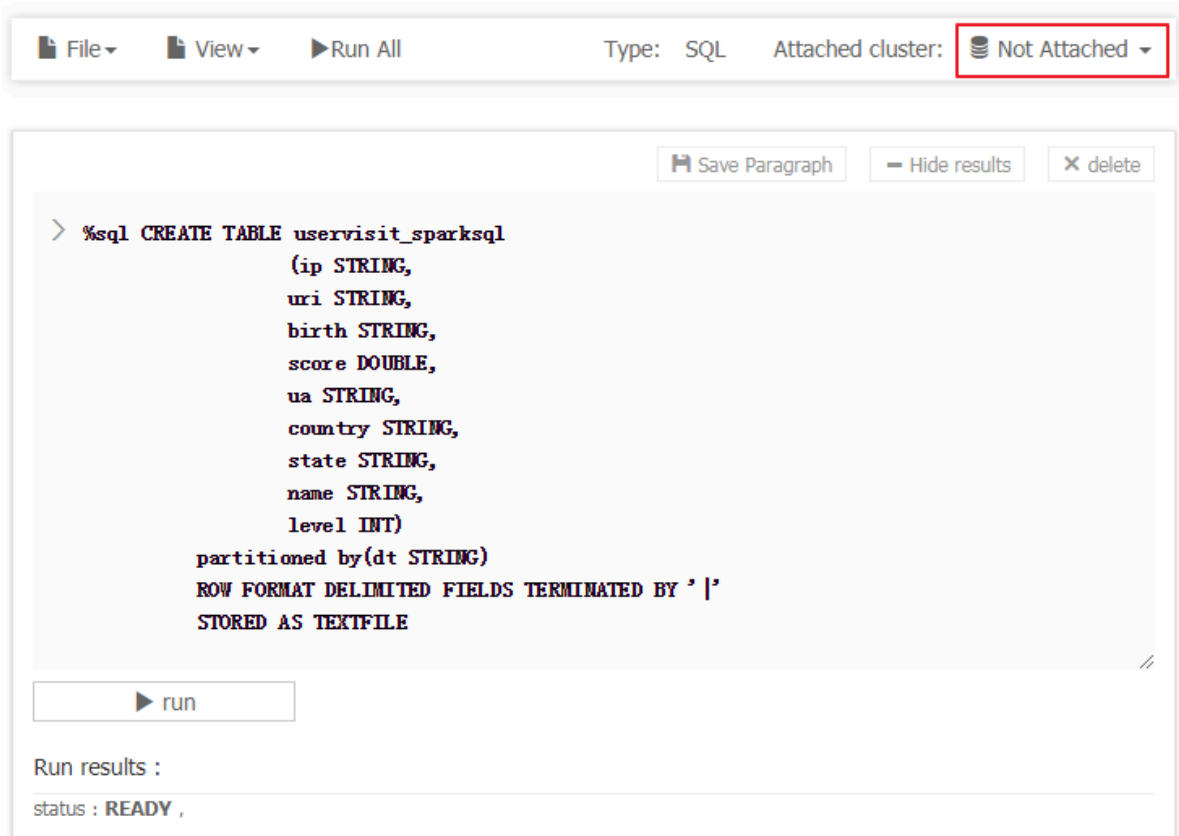
Run results :

```
import scala.math.random
import org.apache.spark._
slices: Int = 20
n: Int = 2000000
count: Int = 1570075
Pi is roughly 3.14015
```

status : **FINISHED** , run 0second(s) , finish Time : Nov 9, 2018 11:25:52 AM

Run a SparkSQL demo task

1. Click EMR-Spark-Demo to display the SparkSQL notebook example. Before running the notebook, you need to associate it to a created cluster. In the upper-right corner, select a created cluster from the list of available clusters.



The screenshot shows the SparkSQL notebook interface. At the top, there is a toolbar with 'File', 'View', and 'Run All' buttons. To the right, it indicates 'Type: SQL' and 'Attached cluster: Not Attached'. The main area contains a SQL query:

```
> %sql CREATE TABLE uservisit_sparksql
      (ip STRING,
       uri STRING,
       birth STRING,
       score DOUBLE,
       ua STRING,
       country STRING,
       state STRING,
       name STRING,
       level INT)
      partitioned by(dt STRING)
      ROW FORMAT DELIMITED FIELDS TERMINATED BY '|'
      STORED AS TEXTFILE
```

Below the query is a 'run' button. Underneath, the 'Run results' section shows 'status : READY'.

2. The SparkSQL demo contains several demo sections that can be run individually or together by clicking Run All. After running, you can see the returned data results of each section.



Note:

**If the section for creating a table is run multiple times, an error is reported indicating that the table already exists.**

Save Paragraph Hide results delete

```
> %sql
  -- get data
  select * from uservisit_sparksql limit 10
```

run

Run results :

ip	uri	birth	score	ua	country	state	name	level	dt
170.131.2.2	13rdgckzlcblruc.html	1984-8-7	336.869186722	NuSearch Spider	HUN	HUN-NL	remnants	3	2016-01-01
162.114.4.2	6xpirezjeytxdjsmwtmyeugkesratmpvamliekrijlgmvyysrlqwgw.html	1978-1-9	331.791153595	Superdown loads Spide rma	AUT	AUT-ZR	MHD	8	2016-01-01
177.110.4.5.18	11zvmoamsyaameokoeylbkivgquksibqbalnmailbiyfxitbhfdroyxesixbjndkyqzl.html	1986-9-25	411.968497603	Mozilla/4.0	FLK	FLK-GB	apj@as.arizona.edu.	7	2016-01-01
157.111.1.2.37	44mvdnl.html	2002-7-3	486.660926201	PHP/4.0.	FIN	FIN-CZ	diffuse	3	2016-01-01

status : FINISHED , run 0second(s) , finish Time : Nov 9, 2018 11:18:38 AM

Run a Hive demo task

1. Click EMR-Hive-Demo to display the Hive notebook example. Before running the notebook, you need to associate it to a created cluster. In the upper-right corner,select a created cluster from the list of available clusters.
2. The Hive demo task contains several demo sections that can be run individually or together by clicking Run All. After running, you can see the returned data results of each section.



Note:

- When the associated cluster executes the Hive notebook for the first time, it takes a few seconds to build the Hive client running environment. It does not need to be built in subsequent executions.

• If the section for creating a table is run multiple times, an error is reported indicating that the table already exists.

Save Paragraph Hide results delete

```

> %hive
  -- get data
  select * from uservisit_hive limit 10
    
```

run

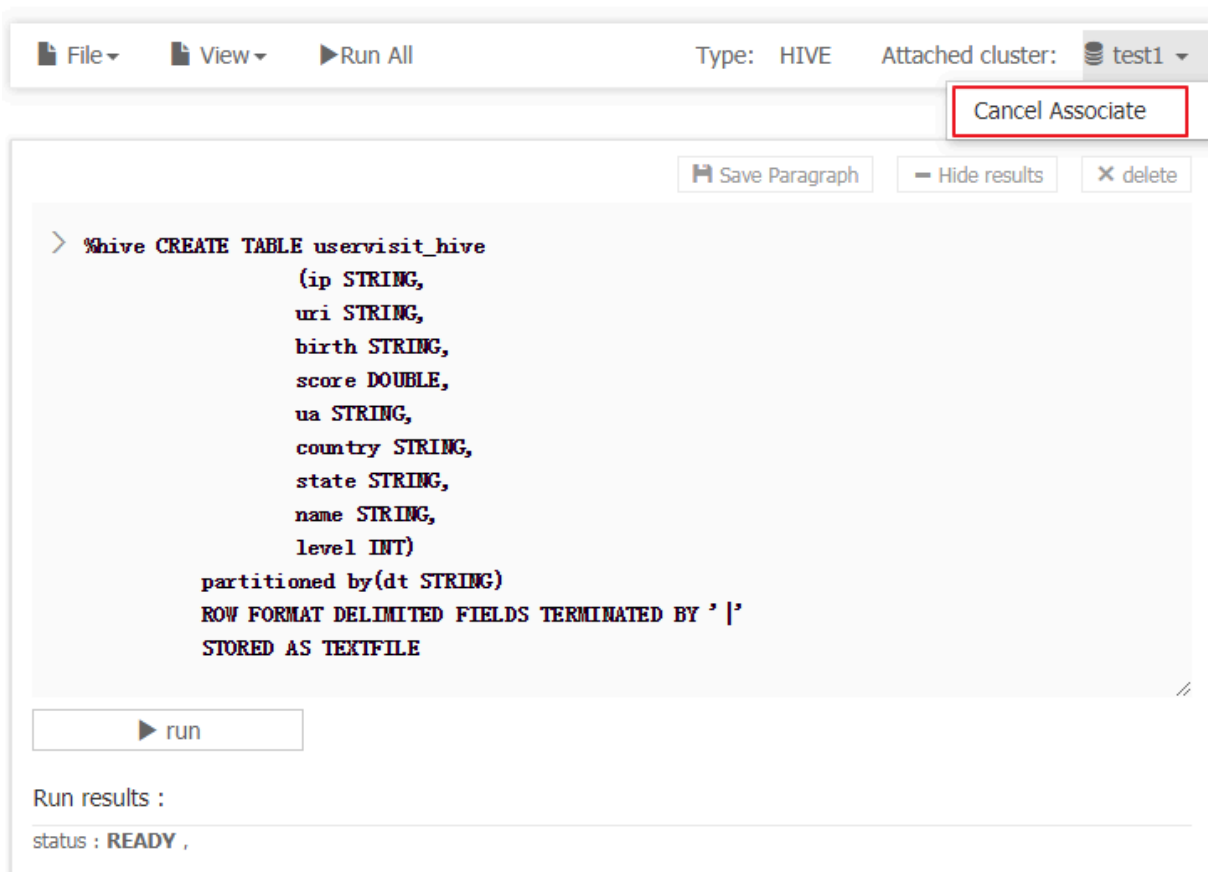
Run results :

uservisit_hive.ip	uservisit_hive.uri	uservisit_hive.birth	uservisit_hive.score	uservisit_hive.ua	uservisit_hive.country	uservisit_hive.state	uservisit_hive.name	uservisit_hive.level	uservisit_hive.dt
170.131.22.2	13rdgckzlcblruc.html	1984-8-7	336.869186722	NuSearch Spider	HUN	HUN-NL	remnants	3	2016-01-01
162.114.42	6xpizrzejtxdjsmwtnyeugkesratmpvamliekrijlgmvyysrlqwgw.html	1978-1-9	331.791153595	Superdown loads Spiderma	AUT	AUT-ZR	MHD	8	2016-01-01
177.110.45.18	11zvmoamsyaameokoeylbkivgquksibqbalnmailbiyfixitbhfdroyxesixbjndkyqz.html	1986-9-25	411.968497603	Mozilla/4.0	FLK	FLK-GB	apj@as.arizona.edu.	7	2016-01-01

status : FINISHED , run 0second(s) , finish Time : Nov 9, 2018 11:21:08 AM

Cancel the association with clusters

**After a notebook is run in a cluster, the cluster creates a process for caching some context running environments in order to ensure a quick response upon re-execution. If you do not need to execute other notebooks, and you want to release the cluster resources occupied by caching, you can disassociate all interactive tasks that have been run from the associated clusters. In this way, you can release the memory resources occupied on the original associated clusters.**



### 8.1.2 Operations

This section details how to perform a number of notebook operations, including how to create a new notebook task on the E-MapReduce console.

Create a new notebook task

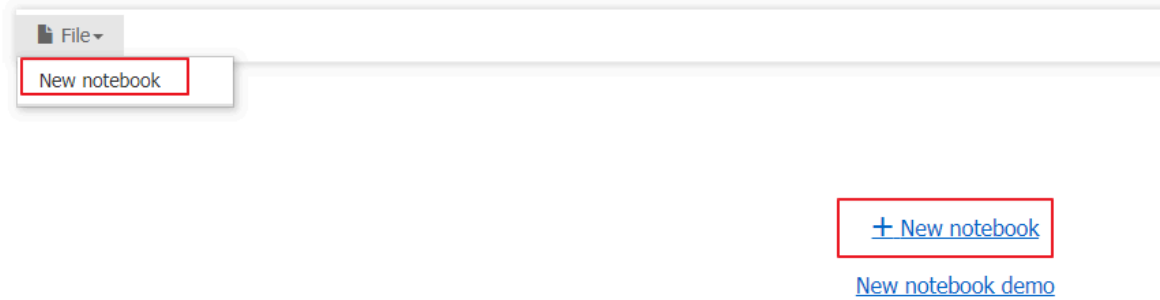


**Note:**

The cluster on which an interactive task is run must be E-MapReduce2.3 or later and have no less than three nodes, each with at least 4 cores and 8 GB of memory.

1. Log on to the [Alibaba Cloud E-MapReduce console](#).
2. At the top of the navigation bar, click Old EMR Scheduling.
3. In the navigation bar on the left, click Notebook.

4. Click **New notebook** or **File > -New notebook**.



5. Enter a name and select the default type. Associating a cluster is optional. Click **OK** to create a notebook.

A screenshot of a 'Notebook' configuration dialog box. The title bar says 'Notebook' with a close button (X) on the right. The form contains three main sections: 1. 'Name': A text input field containing 'test|'. Below it, a note says 'Length: 1 to 64 characters. Only Chinese characters, English letters, numbers '-', and '\_' are allowed'. 2. 'defaultType': Three radio buttons labeled 'Spark', 'Spark SQL', and 'Hive'. 'Spark' is selected. Below it, a note says 'when you run notebook without specify any type , notebook will use this as default type'. 3. 'Attached cluster': An empty dropdown menu. At the bottom right, there are two buttons: a blue 'OK' button and a grey 'Cancel' button.

**Three types of notebook task are supported. Spark can be used to write Scala code, Spark SQL can be used to write SQL statements supported by Spark, and Hive can be used to write SQL statements supported by Hive.**

6. An associated cluster must be E-MapReduce 2.3 or later and have no less than three nodes, each with at least 4 cores and 8 GB of memory. You can also associate the cluster before running the task.

Up to 20 interactive tasks can be created in one account.

Enter and save a section

A paragraph is the smallest unit for running a notebook. Multiple paragraphs can be entered into a notebook. Each paragraph starts with either `%spark`, `%sql`, or `%hive`, indicating whether it is a Scala code paragraph, Spark SQL paragraph, or Hive SQL paragraph. The type prefix is separated by a blank space or by line feed and actual content. If the type prefix is not specified, the default type of the interactive task is used as the run type of this paragraph.

The following example shows how to create a temporary Spark table:

Paste the following code into the section and a red \* symbol is displayed, indicating that this notebook has been changed. Click Save Paragraph or run to save the modifications to the paragraph. Click + under the paragraph to create a new paragraph. Up to 30 paragraphs can be created in one notebook.

```
%spark
import org.apache.commons.io.IOUtils
import java.net.URL
import java.nio.charset.Charset
// load bank data
val bankText = sc.parallelize(
  IOUtils.toString(
    new URL("http://emr-sample-projects.oss-cn-hangzhou.aliyuncs.
com/bank.csv"),
    Charset.forName("utf8")).split("\n"))
case class Bank(age: Integer, job: String, marital: String, education
: String, balance: Integer)
val bank = bankText.map(s => s.split(";")).filter(s => s(0) != "\"age
\").map(
  s => Bank(s(0).toInt,
    s(1).replaceAll("\"", ""),
    s(2).replaceAll("\"", ""),
    s(3).replaceAll("\"", ""),
    s(5).replaceAll("\"", "").toInt
  )
).toDF()
```

```
bank.registerTempTable("bank")
```

The screenshot shows a notebook interface with a code editor and a 'run' button. The code in the editor is as follows:

```
> %spark import org.apache.commons.io.IOUtils
import java.net.URL
import java.nio.charset.Charset
// load bank data
val bankText = sc.parallelize(
  IOUtils.toString(
    new URL("http://emr-sample-projects.oss-cn-hangzhou.aliyuncs.com/bank.csv"),
    Charset.forName("utf8")).split("\n")
)
case class Bank(age: Integer, job: String, marital: String, education: String, balance: Integer)
val bank = bankText.map(s => s.split(";")).filter(s => s(0) != "\age").map(
  s => Bank(s(0).toInt,
    s(1).replaceAll("\\", ""),
    s(2).replaceAll("\\", ""),
    s(3).replaceAll("\\", ""),
    s(5).replaceAll("\\", "").toInt ) ).toDF()
bank.registerTempTable("bank")
```

Below the code editor is a 'run' button. Underneath the 'run' button, the text 'Run results : status : READY ,' is visible. The interface also includes a 'Save Paragraph' button, a 'Hide results' button, and a 'delete' button. A red asterisk icon is in the top-left corner, and a red plus sign icon is at the bottom center.

### Run a paragraph

Before running a notebook, you must first associate it to a created cluster. If a created notebook is not associated with a cluster, Not Attached is displayed in the upper-right corner of the page. Click it to select a cluster from the list of available clusters. Note that the associated cluster must be E-MapReduce 2.3 or later and have no less than three nodes, each with at least 4 cores and 8 GB of memory.



File View Run All
Type: SPARK Attached cluster: Not Attached

\* Save Paragraph
Hide results
delete

```

> %spark import org.apache.commons.io.IOUtils
import java.net.URL
import java.nio.charset.Charset
// load bank data
val bankText = sc.parallelize(
  IOUtils.toString(
    new URL("http://emr-sample-projects.oss-cn-hangzhou.aliyuncs.com/bank.csv"),
    Charset.forName("utf8")).split("\n")
)
case class Bank(age: Integer, job: String, marital: String, education: String, balance: Integer)
val bank = bankText.map(s => s.split(";")).filter(s => s(0) != "\age").map(
s => Bank(s(0).toInt,
s(1).replaceAll("\\", ""),
s(2).replaceAll("\\", ""),
s(3).replaceAll("\\", ""),
s(5).replaceAll("\\", "").toInt ) ).toDF()
bank.registerTempTable("bank")

```

Run results :

status : READY ,

Click Run to save the current paragraph and run the content. If this is the last paragraph, a new paragraph is created automatically.

PENDING indicates that the paragraph has not run yet, RUNNING indicates that the paragraph is running, FINISHED indicates that the running has finished, and ERROR indicates that an error has occurred. The running result is displayed beneath the Run button. During running, you can click Cancel beneath the Run button to cancel running. ABORT is displayed after running has been canceled.

Save Paragraph

Hide results

delete

```

> %spark
import org.apache.commons.io.IOUtils
import java.net.URL
import java.nio.charset.Charset // load bank data
val bankText = sc.parallelize(
  IOUtils.toString(
    new URL("http://emr-sample-projects.oss-cn-hangzhou.aliyuncs.com/bank.csv"),
    Charset.forName("utf8")).split("\n")
)
case class Bank(age: Integer, job: String, marital: String, education: String, balance: Integer)
val bank = bankText.map(s => s.split(";")).filter(s => s(0) != "\age\").map(
  s => Bank(s(0).toInt,
    s(1).replaceAll("\", ""),
    s(2).replaceAll("\", ""),
    s(3).replaceAll("\", ""),
    s(5).replaceAll("\", ").toInt
  )
).toDF()
bank.registerTempTable("bank")

```

run

Run results :

```

import org.apache.commons.io.IOUtils
import java.net.URL
import java.nio.charset.Charset
bankText: org.apache.spark.rdd.RDD[String] = ParallelCollectionRDD[29] at parallelize at <console>:36
defined class Bank
bank: org.apache.spark.sql.DataFrame = [age: int, job: string, marital: string, education: string, balance: int]

```

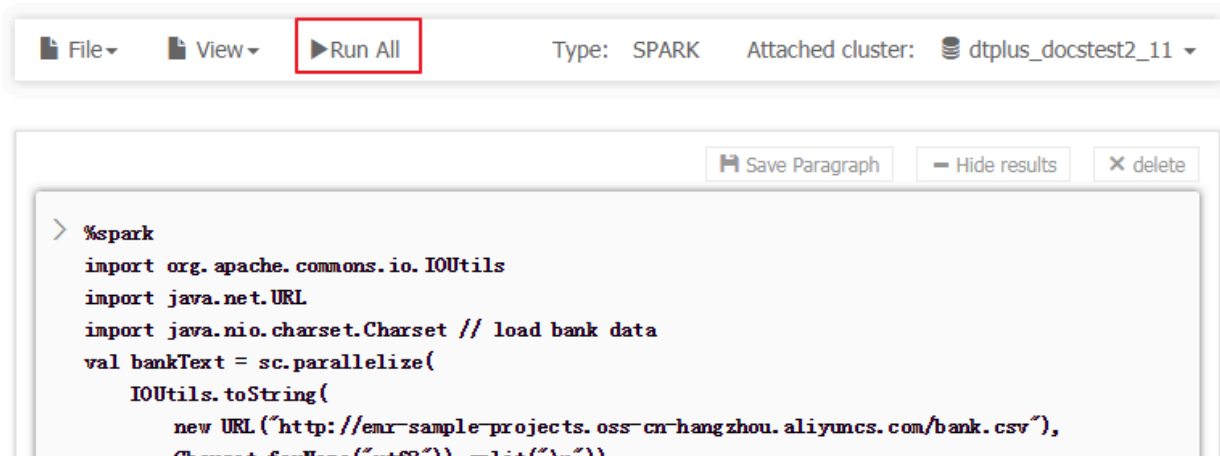
status : FINISHED , run 0second(s) , finish Time : Nov 9, 2018 11:43:32 AM

The paragraph can be run multiple times, but only the result of the last running is retained. You cannot modify the content of a paragraph while it is running. It can only be modified after the running has finished.

Run all

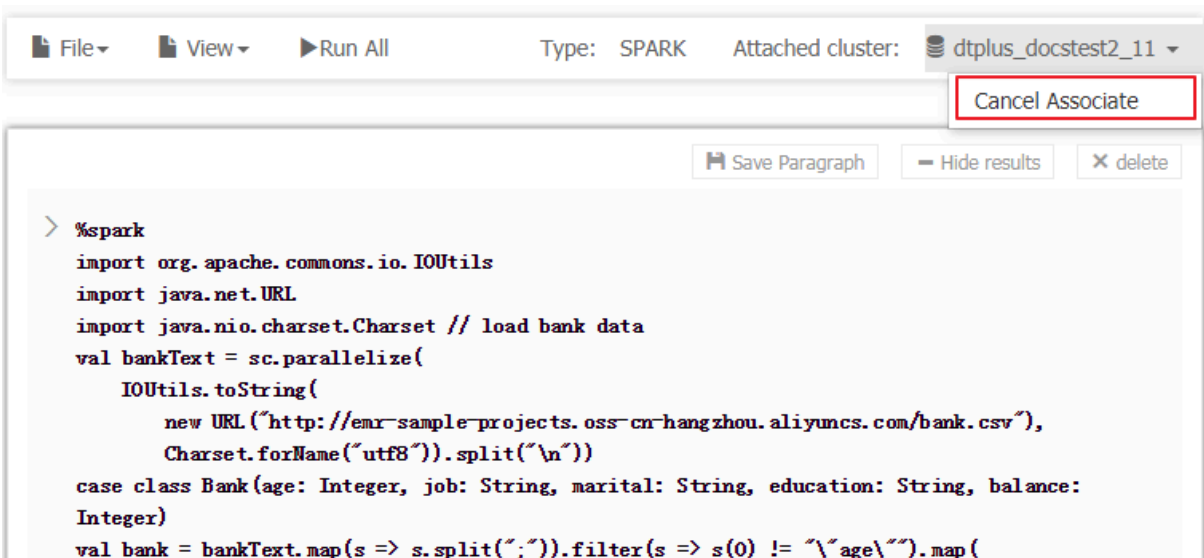
For a notebook, you can click Run All on the menu bar to run all paragraphs. The paragraphs are then submitted sequentially for running. Different types have independent execution queues. If a notebook contains multiple paragraph types, the order for executing them on the cluster is decided based on type after they have been submitted sequentially. Spark and Spark SQL support one-by-one execution. Hive supports concurrent execution, with the maximum number of concurrently executed interactive paragraphs on the same cluster is 10. Note that

all concurrently executed paragraphs are restricted by cluster resources. If the cluster size is small and many paragraphs need to be executed concurrently, the paragraphs still need to queue in YARN.



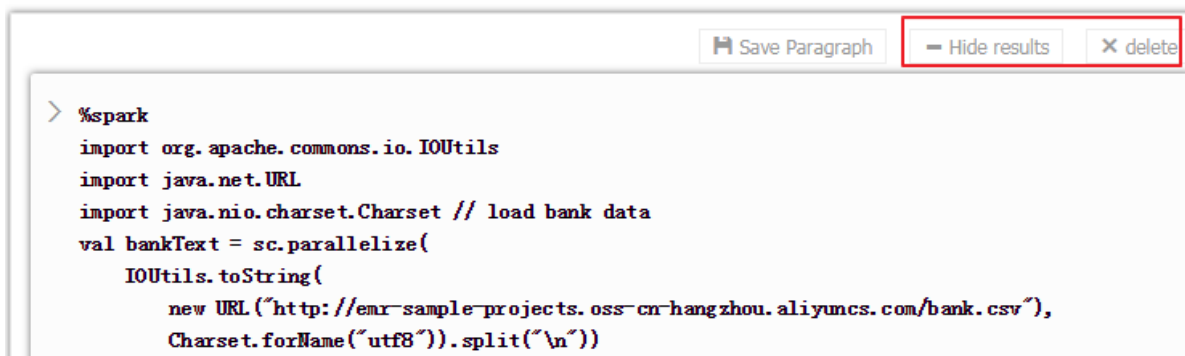
Cancel the association with clusters

After a notebook is run in a cluster, the cluster creates a process for caching some context running environments to ensure a quick response upon re-execution. If you do not need to run other notebooks, and you want to release the cluster resources occupied by caching, you can disassociate all notebooks that have been run from the associated clusters. In this way, you can release the memory resources occupied on the original associated clusters.



## Other operations

## • Paragraph operations



The screenshot shows a code cell in a notebook interface. At the top right of the cell, there are three buttons: "Save Paragraph", "Hide results", and "delete". The "Hide results" button is highlighted with a red border. Below the buttons, the code cell contains the following Spark code:

```
> %spark
import org.apache.commons.io.IOUtils
import java.net.URL
import java.nio.charset.Charset // load bank data
val bankText = sc.parallelize(
  IOUtils.toString(
    new URL("http://emr-sample-projects.oss-cn-hangzhou.aliyuncs.com/bank.csv"),
    Charset.forName("utf8")).split("\n"))
```

## - Hide and display the results

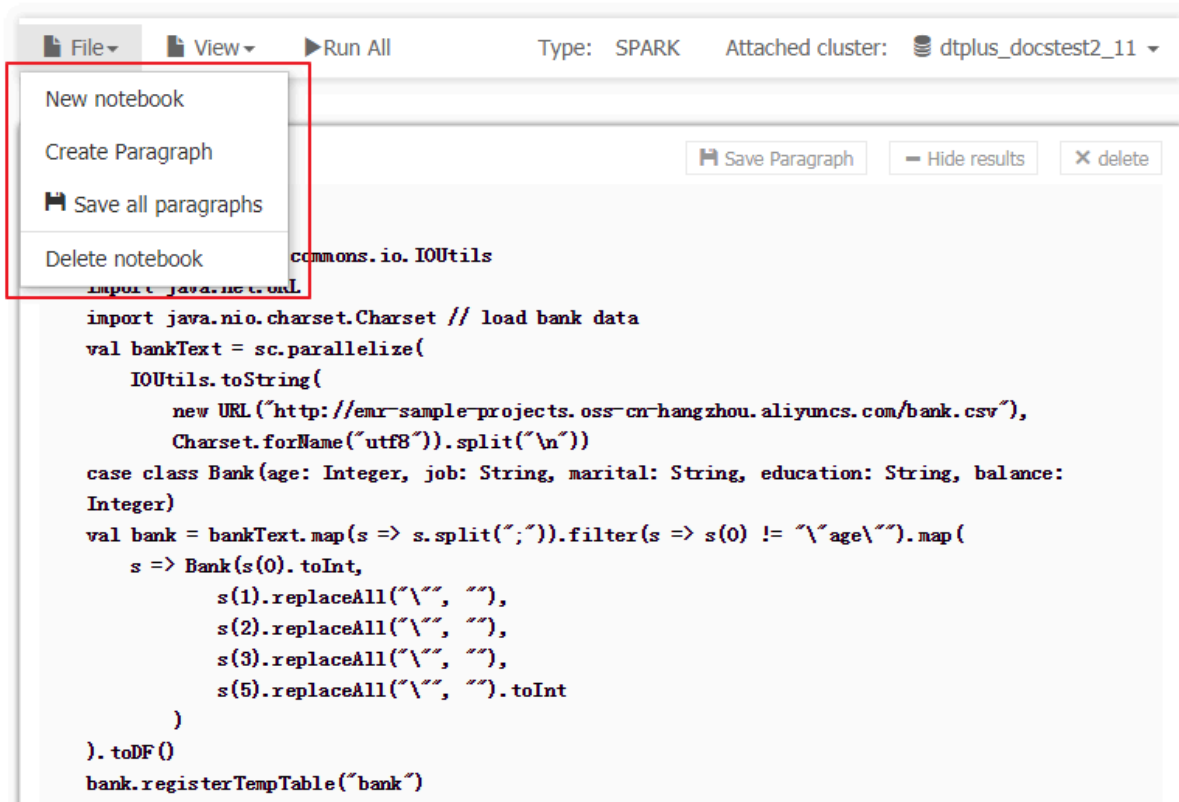
Hide the paragraph results and only display the entered content of the paragraph.

## - Delete a paragraph

Delete the current paragraph. Paragraphs that are running can also be deleted

.

- **File menu**



- **New notebook**

Create a notebook and switch to the created notebook interface.

- **Create Paragraph**

Add a new paragraph to the end of a notebook. A notebook can have up to 30 paragraphs.

- **Save all paragraphs**

Save all modified paragraphs.

- **Delete notebook**

Delete the current notebook. If a cluster has been associated, it will be disassociated.

- **View**

Only display codes or display codes and results.

## 8.1.3 Examples

### 8.1.3.1 Query bank employee information

#### 1. Create a temporary table

```
%spark
import org.apache.commons.io.IOUtils
import java.net.URL
import java.nio.charset.Charset
// Zeppelin creates and injects sc (SparkContext) and sqlContext (
HiveContext or SqlContext)
// So you don't need create them manually
// load bank data
val bankText = sc.parallelize(
  IOUtils.toString(
    new URL("http://emr-sample-projects.oss-cn-hangzhou.aliyuncs.
com/bank.csv"),
    Charset.forName("utf8")).split("\n"))
case class Bank(age: Integer, job: String, marital: String, education
: String, balance: Integer)
val bank = bankText.map(s => s.split(";")).filter(s => s(0) != "\"age
\").map(
  s => Bank(s(0).toInt,
    s(1).replaceAll("\"", ""),
    s(2).replaceAll("\"", ""),
    s(3).replaceAll("\"", ""),
    s(5).replaceAll("\"", "").toInt
  )
).toDF()
bank.registerTempTable("bank")
```

#### 2. Query the table structure

```
%sql
desc bank
```

#### 3. Query the number of employees in each age group under 30

```
%sql select age, count(1) value from bank where age < 30 group by age
order by age
```

#### 4. Query the information of employees younger than or equal to 20

```
%sql select * from bank where age <= 20
```

### 8.1.3.2 Video playback data

#### Preparations

**In this example, you need to download data from OSS and upload it to your OSS bucket. This data includes:**

- [User table sample data](#)
- [Video table sample data](#)

- [Video playback table sample data](#)

**Upload this sample data respectively to the specified UserInfo, Videoinfo, and Playvideo on your OSS bucket. For example, upload the data to the Demo or UserInfo directory under Bucket Example.**

**In the following table, replace the SQL [bucketname] with your bucket name, replace [region] with your OSS region name, and replace [bucketpath] with your specified OSS path prefix, such as Demo.**

1. Create a user table

```
%hive
CREATE EXTERNAL TABLE user_info(id int,sex int,age int, marital_status int) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LOCATION 'oss://[bucketname].oss-cn-[region]-internal.aliyuncs.com/[bucketpath]/userinfo'
```

2. Create a video table

```
%hive
CREATE EXTERNAL TABLE video_info(id int,title string,type string) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LOCATION 'oss://[bucketname].oss-cn-[region]-internal.aliyuncs.com/[bucketpath]/videoinfo'
```

3. Create a video playback table

```
%hive
CREATE EXTERNAL TABLE play_video(user_id int,video_id int, play_time bigint) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LOCATION 'oss
```

```
://[bucketname].oss-cn-[region]-internal.aliyuncs.com/[bucketpath]/playvideo'
```

#### 4. Count the user tables

```
%sql select count(*) from user_info
```

#### 5. Count the video tables

```
%sql select count(*) from video_info
```

#### 6. Count the video playback tables

```
%sql select count(*) from play_video
```

#### 7. Count the video playbacks for each video type

```
%sql select video.type, count(video.type) as count from play_video play join video_info video on (play.video_id = video.id) group by video.type order by count desc
```

#### 8. Display the video information for the top 10 video playbacks

```
%sql select video.id, video.title, video.type, video_count.count from (select video_id, count(video_id) as count from play_video group by video_id order by count desc limit 10) video_count join video_info video on (video_count.video_id = video.id) order by count desc
```

#### 9. Display the age of the viewers watching the video with the most video playbacks

```
%sql select age , count(*) as count from (select distinct(user_id) from play_video where video_id =49 ) play join user_info userinfo on (play.user_id = userinfo.id) group by userinfo.age
```

#### 10. Display the gender, age, and marital status of the viewers watching the video with the most video playbacks

```
%sql select if(sex=0,'Female','Male') as title, count(*) as count, 'Gender' as type from (select distinct(user_id) from play_video where video_id =49 ) play join user_info userinfo on (play.user_id = userinfo.id) group by userinfo.sex union all select case when userinfo.age<15 then 'Less than 15' when age<25 then '15-25' when age<35 then '25-35' else 'More than 35' end , count (*) as count, 'Age Group' as type from (select distinct(user_id) from play_video where video_id =49) play join user_info userinfo on (play.user_id = userinfo.id) group by case when userinfo.age<15 then 'Less than 15' when age<25 then '15-25' when age<35 then '25-35' else 'More than 35' end union all select if(marital_status=0,'Unmarried','Married') as title, count(*) as count, 'Marital Status' as type from (select distinct(user_id) from
```



```
play_video where video_id =49 ) play join user_info userinfo on (
play.user_id = userinfo.id) group by marital_status
```

## 8.2 Execution plans

### 8.2.1 Create an execution plan

An execution plan is a set of jobs that can be executed either at one time or periodically by means of scheduling. It can be executed on an existing E-MapReduce cluster and can also create a temporary cluster to execute the jobs dynamically. Its biggest advantage is that it only uses the resources it needs during execution.

#### Procedure

To create an execution plan, follow these steps:

1. Log on to the [Alibaba Cloud E-MapReduce console](#) .
  2. Select a region.
  3. In the upper-right corner, click Old MER Scheduling to go to the Jobs page.
  4. In the navigation panel on the left, click Execution plan.
  5. In the upper-right corner, click Create an execution plan.
  6. In the Create an execution plan page, select between Create as needed and Existing clusters.
    - a. Create as needed: Create a new cluster to run jobs.
      - Execution plan for one-time scheduling: Clusters with corresponding configurations are created when the execution starts and are then released upon completion of the operation. For more information about creation parameters, see [#unique\\_34](#).
      - Execution plan for periodic scheduling: A new cluster is created based on the scheduling settings you define and is then released upon completion of the operation.
    - b. Existing clusters: Use an existing cluster that complies with the following requirement:
      - Execution plans can only be added to clusters that are Running or Idle.
- Select Existing clusters and then enter the Select Cluster page. Here, you can select a cluster to associate with the execution plan.

7. Click Next to enter the job configuration page. All user jobs are listed in the table on the left. You can select jobs for execution from this table. By clicking the right-facing button, the checked jobs are added to the job queue. Jobs in the queue are then submitted to the cluster for execution in order. The same job can be added and executed several times. If you have not created any jobs, see [Jobs](#).
8. Click Next to enter the scheduling mode configuration page. The configuration items are as follows:
  - a. Name: Must be between 1-64 characters and may only consist of Chinese characters, English letters, numbers, hyphens (-), and underscores (\_).
  - b. Scheduling policy
    - Manual execution: The execution plan is not executed automatically after it is created. Instead, it must be executed manually. Once the execution is in progress, it cannot be executed again.
    - Periodic scheduling: If you select this function, it is enabled immediately after the execution plan is created. The execution then begins from the configured scheduling time. Periodic scheduling can be disabled in the list page. If a scheduling execution starts, but its last execution is not completed, the scheduling is ignored.
  - c. Set the scheduling cycle: There are two scheduling periods: days and hours. The day cycle is one day by default and cannot be changed. However, you can set a specific time interval for hours. The range must be 1-23.
  - d. First execution time: The effective start-time of the scheduling. From this point onwards, periodic scheduling is conducted according to the intervals specified.
9. Click OK to complete the creation of the execution plan.

## Other information

- **Example of periodic scheduling**

\* Set the scheduling cycle :

\* Set the scheduling cycle : per    day(s)

\* first execute time :     :

First run time 2018-11-1 17:35

Subsequent intervals 1 day(s) run 1 Times

**These configurations indicate that the scheduling started on 11/01/2018 at 17:35 with an interval of one day. This means that the second scheduling was conducted on 11/02/2018, at 17:35.**

- **Sequence of jobs**

**Jobs in the execution plan are executed from first to last according to the sequence that you defined in the job list.**

- **Sequence of multiple execution plans**

**When multiple execution plans are submitted to the same cluster, each one submits jobs from its own job sequence. This means that jobs run parallel with each other.**

- **Example of early job debugging**

**During the debugging of a job, it may take some time to create and start a cluster on demand. We recommend that you create a cluster manually first, select Associate the cluster in the execution plan to run jobs, and then set the scheduling mode to Execute immediately. During debugging, you can view the results by clicking Run now on the execution plan list page. Once the debugging is finished, modify the execution plan, modify the way you associate an existing cluster to create a new cluster on demand, and then modify the scheduling mode to periodic scheduling as required. Jobs are then executed automatically on demand.**

## 8.2.2 Manage an execution plan

You can view, manage, and modify your execution plans as follows.

1. Log on to the [Alibaba Cloud E-MapReduce console](#) .
2. Select a region.
3. In the upper-right corner, click Old MER Scheduling to go to the Jobs page.
4. In the navigation panel on the left, click Execution plan.
5. Click Manage next to a plan to go to the execution plan detail page. Here, you can perform the following operations:

- View details of the execution plan

You can view the basic information of the execution plan, such as its name , associated clusters, job configurations, scheduling mode and status, and alarm information.

- Modify the execution plan



**Notice:**

**Jobs can only be modified if they are not currently running or being scheduled. For an execution plan to be executed immediately, it can only be modified when it is not currently running. If the execution plan is scheduled periodically, wait for the completion of its current operation and verify**

whether it is in periodical scheduling. If it is, click Stop scheduling before modifying it.

Each separate module can be modified independently. Click the pen icon to modify information.

- Configure alarm notifications

There are three types of alarm notifications:

- **Booting timeout:** If the periodical scheduling has not been conducted correctly at the specified time and is not executed within 10 minutes of timeout, an alarm is sent.
- **Failed execution:** If any job in the execution plan fails, an alarm is sent.
- **Successful execution:** If all jobs in the execution plan are executed successfully, a notification is sent.

- Run and view results

If the execution plan can be run, in Basic Information, there will be a Run now button to the right of Scheduling status. If you click this button, a schedule will be executed.

At the bottom of the page, there are running records displaying the execution plan instances executed each time, making it easy to view the corresponding job list and logs.

### 8.2.3 Execution plan list

An execution plan list displays basic information about all of your execution plans, as shown in the following figure.

ID/Name	Last run cluster	Last run	Scheduling status	Operation
WF-4330E1VCF3K0TDE plan-luolang	↳ luolang-test	Start Time : 2019/02/20 16:24:32 Running time : 26second(s) Running Status : Complete		Manage   Run now   More ▾
WF-39F4C3A7F311DF zive8ba	↳ ESS_test	Start Time : 2019/02/18 11:00:56 Running time : 5second(s) Running Status : Complete		Manage   Run now   More ▾
WF-4C2D9W112170B4B aa				Manage   Run now   More ▾

- **ID/Name:** The ID and name of the execution plan.
- **Last run cluster:** The last cluster to execute this execution plan. This can either be a cluster created on demand or an existing associated cluster. If a cluster is created automatically on demand, (Automatically created) is displayed beneath it, indicating that the cluster was created on demand by E-MapReduce and will be released automatically after running.

- **Last run:** The running status of the last execution plan.
  - **Start time:** The time at which the last execution plan started.
  - **Running time:** The duration for which the last plan ran.
  - **Running status:** The running status of the last execution plan.
- **Scheduling status:** This indicates whether scheduling is in progress or has been stopped. Only periodic jobs have a scheduling status.
- **Operation**
  - **Manage:** View and modify execution plans.
  - **Run now:** A job can only be run manually when it is neither running nor being scheduled. Click Run now to run the execution plan immediately.
  - **More**
    - **Start/Stop scheduling:** If the scheduling is stopped, Enable scheduling is displayed, which you can click to start the scheduling. If Stop scheduling is displayed during scheduling, you can click it to stop the scheduling. This button is only available for periodic execution plans.
    - **Running log:** Click to enter the job log viewing page.
    - **Delete:** Deletes an execution plan. A running execution plan or one in the process of scheduling cannot be deleted.

## 8.2.4 View job results and logs

In this tutorial, you will learn how to view job results and logs.

View execution records

1. Log on to the [Alibaba Cloud E-MapReduce console](#) .
2. Select a region.
3. In the upper-right corner, click Old MER Scheduling to go to the Jobs page.
4. In the navigation panel on the left, click Execution plan.

5. To the right of the execution plan, click More > Running log.

Running log [Back to execution plan list](#) [Refresh](#)

ID/Name:WF-A388AA222DA06359/test1

Execution order ID	Running Status	Start Time	Running time	Execute cluster	Operation
7	Running	2018/10/31 17:32:00	211day(s)22hour(s)13minute(s)12second(s)		<a href="#">Stop all jobs</a>   <a href="#">View job list</a>
6	Complete	2018/10/30 17:32:00	7second(s)		<a href="#">View job list</a>
5	Complete	2018/10/29 17:32:00	11second(s)		<a href="#">View job list</a>
4	Complete	2018/10/28 17:32:00	5second(s)		<a href="#">View job list</a>
3	Complete	2018/10/27 17:32:00	9second(s)		<a href="#">View job list</a>
2	Complete	2018/10/26 17:31:58	5second(s)		<a href="#">View job list</a>
1	Complete	2018/10/26 17:23:29	10second(s)		<a href="#">View job list</a>

- **Execution order ID:** The sequence of execution for the execution record, which indicates its position in the execution queue. For example, 1 stands for the first position.
- **Running status:** The running status of each execution record.
- **Start time** The time at which the execution plan starts.
- **Running time:** The total running time until the page is viewed.
- **Execute cluster:** The cluster run by the execution plan can either be created on demand or it can be an existing associated cluster. Click to view the cluster details page.
- **Operation**

**View job list:** Click to enter the job list page.

View job records

On the Job list page, you can view the job list in the execution records of a single execution plan as well as the details of each job, as shown in the following figure.

Job execution order ID	Name	Status	Type	Start Time	Running time	Operation
WNE-92693C5408D039E03	test	Failed	Spark	2018/10/26 17:23:39	1second(s)	<a href="#">Stop Job</a>   <a href="#">stdout</a>   <a href="#">stderr</a>   <a href="#">workers log</a>

- **Job execution order ID:** After a job is executed, a corresponding ID is created, which is different from the job ID. The job execution ID is the unique identifier for viewing logs on OSS.
- **Name:** The name of the job.
- **Status:** The running status of the job.
- **Type:** The type of job.
- **Start time:** The time at which the job starts. This is converted into local time.

- **Running time:** The total running time of the job, in seconds.
- **Operation**
  - **Stop job:** You can stop a job if it is in the process of submission or running. If a job is in submission, stopping it will cancel execution. If the job is running, it will be killed.
  - **stdout:** Records all output content from the standard output (Channel 1) of the master process. If log saving is not enabled for the cluster where jobs are run, this function cannot be executed.
  - **stderr:** Records all output content from the diagnostic output (Channel 2) of the master process. If log saving is not enabled for the cluster where jobs are run, this function cannot be executed.
  - **Workers log:** Views the logs of all job worker nodes. If log saving is not enabled for the cluster where jobs are run, this function cannot be executed.

View job worker logs

- **Cloud server instance IP:** The ECS instance ID of a running job and the corresponding intranet IP address.
- **Container ID:** The container ID that runs YARN.
- **Type:** Different log types. stdout and stderr come from different outputs.
- **Operation**

**View the log:** Click different types to view the corresponding logs.

## 8.2.5 Parallel execution of multiple execution plans

To maximize the use of a cluster's available computing resources, multiple execution plans can be associated to the same cluster and executed in parallel.

The main points are summarized as follows:

- **Jobs in the same execution plan are executed in sequence.** By default, preceding jobs are executed before new jobs can be submitted and executed.
- **If you have enough cluster resources, you can create multiple different execution plans and associate them to the same cluster to run and execute jobs in parallel.** Clusters support a maximum of 20 execution plans by default.
- **The management and control system currently supports the submission to YARN of multiple execution plans associated to the same cluster.** However, if



**the cluster itself has insufficient resources, it may take some time for jobs in the YARN queue to wait for scheduling.**

**For more information on how to create execution plans and associate them to a cluster, see [Create an execution plan](#).**

## 8.3 Create a job

**This section describes how to create a job for scheduling in an early E-MapReduce version.**

**To run a computing task, you need to follow these steps to define a job:**

- 1. Log on to the [Alibaba Cloud E-MapReduce console](#).**
- 2. Select a region where you want to create the job.**
- 3. Click the Old EMR Scheduling tab to go to the jobs list page.**

4. Click **Create job** in the upper-right corner to go to the job creation page, as shown in the following figure:

Create job ×

---

\* Name :   
Length: 1 to 64 characters. Only Chinese characters, English letters, numbers '-', and '\_' are allowed

\* Type :  Spark     Hadoop     Hive     Pig  
 Sqoop     Spark SQL     Shell

\* Parameter :   
[+ Select OSS path](#)    [oss console Upload](#)

\* Actual execution :

Fail retry     Yes     No

\* Failure policy :  Pause current execution plan  
 Continue execution of next job

---

5. Enter the job name.

6. Select a job type.

7. Enter parameters of the job. The application parameters must include the following information: the JAR package run by the job, data input and output addresses of the job, and certain command line parameters. You must enter all parameters in the command line in this field. If you need to use an OSS path,

click **Select OSS Path** to select the OSS resource path. For more information about the parameters of all job types, see the *Job* section in *User Guide* .

8. **Actual execution.** The job command that has been executed on ECS will be displayed in this field. You can copy the displayed command and directly run it in the command line environment on an E-MapReduce cluster.
9. **Fail Retry.** You can set the number of retries and their intervals. This feature is disabled by default.
10. **Select a failure policy.** **Pause the current execution plan:** Indicates that the current execution plan will be paused if this job fails and will wait for you to process. **Continue execution of the next job:** Indicates that errors will be ignored and the next job will be executed after this job fails.
11. **Click OK** to complete the creation.

#### Example

This is a Spark job, which sets parameters such as the input and output paths in the application parameters.



**Notice:**

**This example is for reference only.**

Create job
✕

---

**\* Name :**

Length: 1 to 64 characters. Only Chinese characters, English letters, numbers '-', and '\_' are allowed

**\* Type :**

Spark   
  Hadoop   
  Hive   
  Pig  
 Sqoop   
  Spark SQL   
  Shell

**\* Parameter :**

```
spark-submit --class org.apache.spark.examples.SparkPi --master yarn-client --driver-memory 512m --num-executors 1 --executor-memory 1g --executor-cores 2 /opt/apps/spark-1.6.1-bin-hadoop2.7/lib/spark-examples-1.6.1-hadoop2.7.2.jar 100
```

+ Select OSS path

**\* Actual execution :**

```
spark-submit spark-submit --class org.apache.spark.examples.SparkPi --master yarn-client --driver-memory 512m --num-executors 1 --executor-memory 1g --executor-cores 2 /opt/apps/spark-1.6.1-bin-hadoop2.7/lib/spark-examples-1.6.1-hadoop2.7.2.jar 100
```

**\* Failure policy :**

Pause current execution plan  
 Continue execution of next job

## OSS and OSSREF

**The oss:// prefix indicates that the data path points to an OSS path, which specifies the operation path when reading/writing the data. This is similar to hdfs://.**

**The ossref:// prefix also indicates that the data path points to an OSS path. However, it is used to download the corresponding code to a local disk, and then replace the path in the command line with this local path. It is easier for you to run native code. You do not need to log on to the computer to upload the code and the dependent resource packages.**

**In this example, the ossref://xxxxxx/xxx.jar parameter represents the JAR package of job resources. This JAR package is stored on OSS. When this path is executed in the code, the JAR package will automatically download to the cluster and be executed. The two oss://xxxx and the two values following the JAR package are processed by the main class in the JAR package as parameters.**



### Notice:

**The ossref cannot be used to download large amounts of data. Otherwise, an error may occur.**