

阿里云 E-MapReduce

数据开发

文档版本：20190905

法律声明

阿里云提醒您 在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的”现状“、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含”阿里云”、Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

| 格式 | 说明 | 样例 |
|---|-----------------------------------|--|
|  | 该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。 |  禁止： 重置操作将丢失用户配置数据。 |
|  | 该类警示信息可能导致系统重大变更甚至故障，或者导致人身伤害等结果。 |  警告： 重启操作将导致业务中断，恢复业务所需时间约10分钟。 |
|  | 用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。 |  说明： 您也可以通过按Ctrl + A选中全部文件。 |
| > | 多级菜单递进。 | 设置 > 网络 > 设置网络类型 |
| 粗体 | 表示按键、菜单、页面名称等UI元素。 | 单击 确定 。 |
| <code>courier</code> 字体 | 命令。 | 执行 <code>cd /d C:/windows</code> 命令，进入Windows系统文件夹。 |
| <code>##</code> | 表示参数、变量。 | <code>bae log list --instanceid</code> <code>Instance_ID</code> |
| <code>[]</code> 或者 <code>[a b]</code> | 表示可选项，至多选择一个。 | <code>ipconfig [-all -t]</code> |
| <code>{ }</code> 或者 <code>{a b}</code> | 表示必选项，至多选择一个。 | <code>swich {stand slave}</code> |

目录

| | |
|--------------------------------|----|
| 法律声明..... | I |
| 通用约定..... | I |
| 1 项目管理..... | 1 |
| 2 作业编辑..... | 3 |
| 3 临时查询..... | 7 |
| 4 workflow 编辑..... | 10 |
| 5 集群模板..... | 12 |
| 6 云监控事件编码..... | 13 |
| 7 作业..... | 14 |
| 7.1 作业的可执行操作..... | 14 |
| 7.2 作业日期设置..... | 14 |
| 7.3 Hive SQL 作业配置..... | 15 |
| 7.4 Hadoop MapReduce 作业配置..... | 17 |
| 7.5 Hive 作业配置..... | 19 |
| 7.6 Pig 作业配置..... | 20 |
| 7.7 Spark 作业配置..... | 22 |
| 7.8 Spark SQL 作业配置..... | 24 |
| 7.9 Shell 作业配置..... | 25 |
| 7.10 Sqoop 作业配置..... | 26 |
| 7.11 Flink 作业配置..... | 27 |
| 7.12 Spark Streaming 作业配置..... | 28 |
| 7.13 Streaming SQL 作业配置..... | 29 |
| 8 老版作业调度（即将下线）..... | 33 |
| 8.1 交互式工作台..... | 33 |
| 8.1.1 交互式工作台简介..... | 33 |
| 8.1.2 交互式工作台操作说明..... | 34 |
| 8.1.3 交互式工作台示例..... | 37 |
| 8.1.3.1 银行员工信息查询示例..... | 37 |
| 8.1.3.2 视频播放数据示例..... | 38 |
| 8.2 执行计划..... | 40 |
| 8.2.1 创建执行计划..... | 40 |
| 8.2.2 管理执行计划..... | 42 |
| 8.2.3 执行计划列表..... | 43 |
| 8.2.4 作业结果和日志查看..... | 43 |
| 8.2.5 多执行计划并行执行..... | 45 |
| 8.3 创建作业..... | 45 |

1 项目管理

创建E-MapReduce集群后，用户可以创建工作流项目，使多个作业可以同时或者按照先后顺序运行，以便更好的管理作业的运行。

创建项目

1. 通过主账号登录[阿里云 E-MapReduce 控制台](#)。
2. 单击上方的数据开发页签，进入项目列表页面。

主账号下可以查看该账号下的所有项目（包括所有子账号），子账号仅可以查看具有开发权限的项目。如需添加项目开发权限，需要通过主账号来配置，请参见[用户管理](#)。

3. 单击右上角的新建项目按钮，弹出新建项目对话框。
4. 输入项目名称和项目描述，单击创建。



说明：

只有主账号才能创建项目，即新建项目按钮只对主账号管理员可见。

用户管理

创建新的项目后，您可以为RAM子账号添加该项目的操作权限。

1. 在项目列表页面，单击项目右侧的详情。
2. 单击用户管理页签。
3. 单击添加用户，添加该主账号下的RAM子账号到该项目。

被添加的子账号将成为该项目的成员，并能查看、开发该项目下的作业和工作流。如果不想将子账号继续设置为所选项目成员，单击用户右侧的删除即可。



说明：

只有主账号才能添加项目成员，即项目列表页面中的用户管理功能只对主账号管理员可见。

关联集群资源

创建新的项目后，您需要为项目关联集群，使得该项目中的工作流可以运行在关联的集群上。

1. 在项目列表页面，单击项目右侧的详情。
2. 单击集群设置页签。
3. 单击添加集群，从下拉菜单中可以选择已购买的包年包月和按量付费集群（执行临时作业创建的集群此处不会列示）。

4. 单击确定。

单击集群右侧的删除，可以取消关联该集群资源。



说明：

只有主账号才能添加集群资源，即项目列表页面中的集群设置功能只对主账号管理员可见。

单击集群右侧的修改配置，可以设置提交作业到该集群的队列和用户。具体配置项说明如下：

- 提交作业默认用户：设置项目使用所选集群提交作业时的默认Hadoop用户，默认值是hadoop，默认用户只能有一个。
- 提交作业默认队列：设置项目使用所选集群提交作业时的默认队列，如果此处不填写，则作业会提交到default队列。
- 提交作业用户白名单：设置可以提交作业的Hadoop用户，如果有多个用户，可以通过英文半角逗号（,）分隔。
- 提交作业队列白名单：用于设置项目中的作业可以运行在所选集群的队列，若果有多个队列，可以通过英文半角逗号（,）分隔。
- 配置客户端白名单：配置可以提交作业的客户端，用户可以使用EMR的Master节点或EMR购买的Gateway，ECS自建Gateway暂不支持在此处配置。

2 作业编辑

在项目中，您可以创建Shell、Hive、Hive SQL、Spark、SparkSQL、MapReduce、Sqoop、Pig、Spark Streaming、Flink 等类型的作业。

新建作业

1. 通过主账号登录[阿里云 E-MapReduce 控制台](#)。
2. 单击上方的数据开发页签，进入项目列表页面。
3. 单击项目右侧的工作流设计，单击左侧导航栏中的作业编辑进入作业编辑页面。
4. 在页面左侧，在需要操作的文件夹上单击右键，选择新建作业。
5. 在新建作业对话框中，输入作业名称、作业描述，选择作业类型。

创建作业时作业类型一经确定，不能修改。

6. 单击确定。



说明：

您可以通过在文件夹上单击右键，进行创建子文件夹、重命名文件夹和删除文件夹操作。

开发作业

关于各类作业的具体开发，请参见《EMR 用户指南》的[作业](#)部分。



说明：

插入 OSS 路径时，如果选择 OSSREF 文件前缀，系统会把 OSS 文件下载到集群本地，并添加到 classpath 中。

· 作业基础设置

单击页面右上角的作业设置，弹出作业设置页面。

- 失败重试次数：设置在工作流运行到该作业失败时，重试的次数。直接在作业编辑页面运行作业，该选项不会生效。
- 失败策略：设置在工作流运行到该作业失败时，继续执行下一个节点还是暂停当前工作流。
- 添加运行资源：如添加作业执行需依赖的 jar 包或UDF等资源，需将资源先上传至 OSS。在作业的运行资源中选中该资源后，可以直接在作业中引用该资源。
- 配置参数：指定作业代码中所引用变量的值。用户可以在代码中引用变量，格式为：`${## #}`。单击右侧的加号图标添加 key 和 value，key 为变量名，value 为变量的值。另外，您还可以根据调度启动时间自定义时间变量，规则如下：

- yyyy 表示 4 位的年份。
- MM 表示月份。
- dd 表示天。
- HH 表示 24 小时制，12 小时制使用 hh。
- mm 表示分钟。
- ss 表示秒。

时间变量可以是包含 yyyy 年份的任意时间组合，同时支持用+和-方式来分别表示提前和延后。例如，变量 `${yyyy-MM-dd}` 表示当前日期，则：

- 后 1 年的表示方式：`${yyyy+1y}` 或者 `${yyyy-MM-dd hh:mm:ss+1y}`。
- 后 3 月的表示方式：`${yyyyMM+3m}` 或者 `${hh:mm:ss yyyy-MM-dd+3m}`。
- 前 5 天的表示方式：`${yyyyMMdd-5d}` 或者 `${hh:mm:ss yyyy-MM-dd-5d}`。



注意：

时间变量参数必须以 'yyyy' 开始，如 `${yyyy-MM}`。如果希望单独获取月份等特定时间区域的值，可以在作业内容中使用如下两个函数提取：

- `parseDate(<参数名称>, <时间格式>)`: 将给定参数转换为 Date 对象。其中，参数名称为上述配置参数中设置的一个变量名，时间格式为设置该变量时所使用的时间格式。如设置一个变量 `current_time = ${yyyyMMddHHmmss-1d}`，则此处时间格式应设置为 'yyyyMMddHHmmss'；
- `formatDate(<Date 对象>, <时间格式>)`: 将给定 Date 对象转换为给定格式的时间字符串。

函数使用示例：

- 获取 `current_time` 变量的小时字面值: `${formatDate(parseDate(current_time, 'yyyyMMddHHmmss'), 'HH')}`
- 获取 `current_time` 变量的年字面值: `${formatDate(parseDate(current_time, 'yyyyMMddHHmmss'), 'yyyy')}`

· 作业高级设置

在作业设置页面，单击高级设置页签。

- 模式：包括从 Worker 节点提交和从 Header/Gateway 节点提交两种模式。

- Worker 节点提交模式下，作业通过 Launcher 在 YARN 上分配资源进行提交。

- 从 Header/Gateway 节点提交模式下，作业在分配的机器上直接运行。

- 环境变量：添加作业执行的环境变量，也可以在作业脚本中直接 `export` 环境变量。

例如您有一个 shell 类型的任务，内容是 `echo ${ENV_ABC}` 您在这里设置了一个环境变量，`ENV_ABC=12345` 那么，上面的 `echo` 命令您会得到输出结果：12345；进一步，如果您有一个 shell 类型的作业，内容是 `java -jar abc.jar`，其中 `abc.jar` 的内容是：

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

```

那么您会得到结果：12345

这里的环境变量的设置相当于执行了这样的脚本：

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

```

- 调度参数：设置作业运行 YARN 队列、CPU、内存和 Hadoop 用户等信息，可以不设置，作业会直接采用 Hadoop 集群的默认值。

配置说明

新版作业提交支持两种模式：

- 从 Header/Gateway 节点提交：`spark-submit` 这个进程在 header 节点运行，不受 YARN 监控。`spark-submit` 比较耗内存，过多的作业会造成 header 节点资源紧张，进而导致整个集群不稳定。
- 从 Worker 节点提交：`spark-submit` 这个进程在 worker 节点运行，占用 YARN 的一个 container，受 YARN 监控。可以缓解 header 节点的资源使用。

`spark-submit` 进程（在数据开发模块里为 LAUNCHER）是 Spark 的作业提交命令，用来提交 Spark 作业，一般占用 600MB+。

作业配置面板中的内存设置，用于设置 LAUNCHER 的内存配额。

一个完整的 Spark 作业包括：spark-submit（消耗内存：600 MB）+ driver（消耗内存：看具体作业，可能是JOB，也可能是LAUNCHER）+ executor（消耗内存：看具体作业实现，JOB）

- 如果 Spark 使用 yarn-client 模式， spark-submit + driver 是在同一个进程中（消耗内存：600MB + driver的内存消耗）。这个进程在作业提交中使用 LOCAL 模式的话，是 header 节点上的一个进程，不受 YARN 监控。如果用 YARN 模式的话，是 worker 上的一个进程，占用一个YARN container，受 YARN 监控。
- 如果 Spark 使用 YARN cluster 模式， driver 独立一个进程，占用 YARN 的一个 container，和 spark-submit 不在一个进程。

综上，从Header/Gateway节点提交决定 spark-submit 进程在 header 节点还是在 worker 节点，受不受 YARN 的监控。Spark 的 yarn-client/yarn-cluster 模式，决定 driver 是否和 spark-submit 一个进程。

作业执行

作业开发和配置完成后，您可以单击右上角的运行按钮执行作业。

查看日志

作业运行后，您可以在页面下方的运行记录页签中查看作业的运行日志。单击详情跳转到运行记录中该作业的详细日志页面，可以看到作业的提交日志、YARN Container 日志。

常见问题

- 流式作业的日志过多，导致磁盘空间不足的问题

Spark Streaming 等流式作业的场景，建议用户开启日志 Rolling，防止因为运行的时间过长而导致日志过大，磁盘空间不足的问题，具体开启方法如下：

1. 在E-MapReduce 控制台依次单击数据开发 > 项目ID > 作业编辑 > 作业设置 > 高级设置。
2. 在环境变量部分单击 '+' 添加环境变量：

```
FLOW_ENABLE_LOG_ROLLING = true
```

3. 保存，重启作业。



说明：

如果已经发现作业日志过多，而又不想重启作业，可以先使用 `echo > /path/to/log/dir/stderr` 的方式，将作业的日志清空。

3 临时查询

临时查询是 adhoc 即席查询的场景，只支持 HiveSQL SparkSQL 和 Shell 三种类型，运行临时查询的语句，在页面下方显示日志和查询结果。

新建作业

作业编辑页中运行作业，单击对应作业详情会跳转到详情页面显示提交日志和运行日志。作业与两者的区别主要是运行场景不同，临时查询针对数据科学家和数据分析师，主要用SQL为工具。

1. 通过主账号登录[阿里云 E-MapReduce 控制台](#)。
2. 单击上方的数据开发页签，进入项目列表页面。
3. 单击对应项目右侧的工作流设计，进入作业编辑页面。
4. 单击页面左侧的临时查询页签，进入临时查询页面。
5. 在页面左侧，在需要操作的文件夹上单击右键，选择新建作业。
6. 在新建作业对话框中，输入作业名称、作业描述，选择作业类型。

创建作业时作业类型一经确定，不能修改。

7. 单击确定。



说明:

您可以通过在文件夹上单击右键，进行创建子文件夹、重命名文件夹和删除文件夹操作。

开发作业

关于 HiveSQL SparkSQL 和 shell 作业的具体开发，请参见 EMR 用户指南-数据开发-《作业》部分。



说明:

插入 OSS 路径时，如果选择 OSSREF 文件前缀，系统会把OSS文件下载到集群本地，并添加到 classpath 中。

· 作业基础设置

单击页面右上角的作业设置，弹出作业设置页面。

- 添加运行资源：如添加作业执行需依赖的 jar 包或 UDF 等资源，需将资源先上传至 OSS。在作业的运行资源中选中该资源后，可以直接在作业中引用该资源。
- 配置参数：指定作业代码中所引用变量的值。用户可以在代码中引用变量，格式为：`${## #}`。单击右侧的加号图标添加 key 和 value，key 为变量名，value 为变量的值。另外，您还可以根据调度启动时间自定义时间变量，规则如下：

- yyyy 表示 4 位的年份。
- MM 表示月份。
- dd 表示天。
- HH 表示 24 小时制，12 小时制使用 hh。
- mm 表示分钟。
- ss 表示秒。

时间变量可以是包含 yyyy 年份的任意时间组合，同时支持用+和-方式来分别表示提前和延后。例如，变量 `${yyyy-MM-dd}` 表示当前日期，则：

- 后 1 年的表示方式：`${yyyy+1y}` 或者 `${yyyy-MM-dd hh:mm:ss+1y}`。
- 后 3 月的表示方式：`${yyyyMM+3m}` 或者 `${hh:mm:ss yyyy-MM-dd+3m}`。
- 前 5 天的表示方式：`${yyyyMMdd-5d}` 或者 `${hh:mm:ss yyyy-MM-dd-5d}`。



注意：

时间变量参数必须以 'yyyy' 开始，如 `${yyyy-MM}`。如果希望单独获取月份等特定时间区域的值，可以在作业内容中使用如下两个函数提取：

- `parseDate(<参数名称>, <时间格式>)`：将给定参数转换为 Date 对象。其中，参数名称为上述配置参数中设置的一个变量名，时间格式为设置该变量时所使用的时间格式。如设置一个变量 `current_time = ${yyyyMMddHHmmss-1d}`，则此处时间格式应设置为 'yyyyMMddHHmmss'；
- `formatDate(<Date对象>, <时间格式>)`：将给定 Date 对象转换为给定格式的时间字符串。

函数使用示例：

- 获取 current_time 变量的小时字面值：`${formatDate(parseDate(current_time, 'yyyyMMddHHmmss'), 'HH')}`

```
■ 获取 current_time 变量的年字面值: ${formatDate(parseDate(current_time, 'yyyyMMddHHmmss'), 'yyyy')}
```

· 作业高级设置

在作业设置页面，单击高级设置页签。

- 模式：包括从 Worker 节点提交和从 Header/Gateway 节点提交两种模式。
 - Worker 节点提交模式下，作业通过 Launcher 在 YARN 上分配资源进行提交。
 - 从 Header/Gateway 节点提交模式下，作业在分配的机器上直接运行。
- 调度参数：设置作业运行 YARN 队列、CPU、内存和 Hadoop 用户等信息，可以不设置，作业会直接采用 Hadoop 集群的默认值。

作业执行

作业开发和配置完成后，您可以单击右上角的运行按钮执行作业。

查看日志

作业运行后，您可以在页面下方的日志页签中查看作业的运行日志。

4 workflow 编辑

E-MapReduce workflow 支持通过 DAG 的方式并行执行大数据作业，用户可以暂停、停止、重新运行 workflow，还可以在 Web UI 查看 workflow 的执行状态。

新建 workflow

1. 通过主账号登录[阿里云 E-MapReduce 控制台](#)。
2. 单击上方的数据开发页签，进入项目列表页面。
3. 单击对应项目右侧的 workflow 设计，然后单击左侧的 workflow 设计页签，进入 workflow 设计页面。
4. 在页面左侧，在需要操作的文件夹上单击右键，选择新建 workflow。
5. 在新建 workflow 对话框中，输入 workflow 名称、workflow 描述，选择执行集群。

用户可以选择已经创建的且被关联到该项目的预付费和后付费 EMR 集群用于执行 workflow，也可以通过集群模板的方式新建一个临时集群用于执行该 workflow。

6. 单击确定。

编辑 workflow

用户可以通过拖拽方式将不同类型的作业拉到 workflow 编辑画布，将不同作业节点通过连线的方式指定 workflow 的流转。作业拖拽完成后，从控制节点处拖拽 END 组件到画布中，表示整个 workflow 设计完成。

配置 workflow

在 workflow 设计页面的右侧，单击配置按钮，可以进行 workflow 调度配置。

· 执行集群

选择当前 workflow 中各个节点默认的执行集群，有以下两种模式：

- 常驻集群：选择当前已存在的集群，workflow 执行时，相关任务会下发到该集群中。
- 按需集群：选择 `#unique_8`，调度系统在 workflow 启动时先按模版创建一个集群，然后将作业下发到该集群上执行。在 workflow 结束后，调度系统会自动释放该集群。

- 调度策略：在开启 workflow 调度后，时间依赖是默认必须使用的，同时您可以添加 workflow 依赖调度。
 - 时间调度：设置 workflow 调度的开始时间和结束时间，在此时间范围内，系统会根据您设置的周期执行 workflow。
 - 依赖调度：从所选项目中，选择当前 workflow 的前续 workflow。当前续 workflow 执行完成后，当前 workflow 才会被调度执行。目前依赖调度只能选择一个 workflow。
- 告警配置

目前支持通过短信、邮件和钉钉群的方式发送告警，相关告警事件包括：

- 执行失败： workflow 执行失败时告警
- 节点失败： workflow 中有节点执行失败时告警
- 执行成功： workflow 执行成功时发送通知
- 启动超时：如果 workflow 中有节点在下发到集群后 30 分钟内还没有启动，将发送告警信息并取消该节点任务

执行 workflow

workflow 设计和配置完成后，您可以单击右上角的运行按钮执行 workflow。

查看并操作 workflow 实例

workflow 运行后，单击左侧的运行记录页签，可以查看 workflow 实例的运行状态。单击 workflow 实例对应的详情，可以查看作业实例的运行情况，也可以暂停、恢复、停止和重跑 workflow 实例。

- 暂停 workflow 后：正在运行的作业节点会继续执行，但后续的作业节点不再执行，可以单击恢复 workflow，系统将继续执行暂停作业节点之后的作业。
- 取消 workflow：所有正在运行的作业节点立即停止。
- 重跑 workflow 实例：系统将从 workflow 的 start 节点从头开始执行 workflow。

5 集群模板

集群模版是为快速创建集群而保存的配置。目前，集群模版用于数据开发工作流自动创建测试集群，后续会支持其他使用场景。

入口：[阿里云 E-MapReduce 控制台](#) -> 数据开发 -> 集群模版

使用

集群模版的创建过程与集群创建基本一致，在基础配置中，您需要指定模版的名字。您可以在集群模版列表对应条目后单击编辑按钮修改集群模版。修改之后，会立即生效到引用此模版的工作流。您也可以在此列表单击删除按钮删除对应集群模版。在 EMR 工作流中使用集群模板创建的集群，工作流执行结束后，集群会自动释放。



注意:

系统不会检查此模版是否被引用，删除之后，通过集群模版自动创建集群的工作流会失败。

关于集群模版在工作流中的使用，请参见[#unique_10](#)。

限制

- 集群模版可能不支持新的集群类型，如有需求，您可提交工单
- 暂不支持集群密码设置

6 云监控事件编码

在云监控的事件监控模块中，您可以订阅EMR数据开发相关的系统事件，实现电话告警之类的需求。

云监控系统事件编码及其含义如下：

| 事件编码 | 事件描述 | 事件类型 |
|---------------|------------------|------|
| EMR-110401002 | workflow已成功。 | FLOW |
| EMR-110401003 | workflow已提交。 | FLOW |
| EMR-110401004 | 作业已提交。 | FLOW |
| EMR-110401005 | workflow节点已启动。 | FLOW |
| EMR-110401006 | workflow节点状态已检查。 | FLOW |
| EMR-110401007 | workflow节点已完成。 | FLOW |
| EMR-110401008 | workflow节点已结束。 | FLOW |
| EMR-110401009 | workflow节点已取消。 | FLOW |
| EMR-110401010 | workflow已取消。 | FLOW |
| EMR-110401011 | workflow已重跑。 | FLOW |
| EMR-110401012 | workflow已恢复。 | FLOW |
| EMR-110401013 | workflow已暂停。 | FLOW |
| EMR-110401014 | workflow已结束。 | FLOW |
| EMR-110401015 | workflow节点已失败。 | FLOW |
| EMR-110401016 | 作业已失败。 | FLOW |
| EMR-210401001 | workflow已失败。 | FLOW |
| EMR-210401003 | workflow节点启动超时。 | FLOW |
| EMR-210401004 | 作业启动超时。 | FLOW |

7 作业

7.1 作业的可执行操作

您可以对作业进行创建、克隆、修改、删除操作。

作业的创建

一个新作业可以在任何时候被创建。被创建的作业目前只可以在所创建的 Region 内被使用。

作业的克隆

完全的克隆一个已经存在作业的配置。同样也只限定在同一个 Region 内。

作业的修改

如果要将作业加入到一个执行计划中，需要保证该执行计划当前没有在运行中，同时也需要保证执行计划的周期调度没有在调度中，这个时候才可以修改该作业。

如果要将这个作业加入到多个执行计划中，需停止要加入的所有执行计划的运行和周期调度后才可以修改。因为修改作业会导致所有使用该作业的执行计划也发生变化，可能会导致正在执行的或者周期调度的执行计划的错误。

如果想要进行调试，推荐使用克隆功能，完成调试后，替换执行计划中的原作业。

作业的删除

和修改一样，只有在作业加入的执行计划当前没有在运行中，同时周期调度也没有在调度中的情况下，才能被删除。

7.2 作业日期设置

在创建作业过程中，支持在作业参数中设置时间变量通配符。

变量通配符格式

E-MapReduce 所支持的变量通配符的格式为 `${dateexpr-1d}` 或者 `${dateexpr-1h}` 的格式。例如，假设当前时间为 20160427 12:08:01:

- 如果在作业参数中写成 `${yyyyMMdd HH:mm:ss-1d}`，那么这个参数通配符在真正执行的时候会被替换成 20160426 12:08:01，即在当前日期上减了一天并精确到了秒。
- 如果写成 `${yyyyMMdd-1d}`，则执行时会替换成 20160426，表示当前日期的前一天。
- 如果写成 `${yyyyMMdd}`，则会被替换成 20160427，直接表示当前的日期。

`dateexpr` 表示标准的时间格式表达式，对应的时间会按照该表达式指定的格式进行格式化，后面可以再跟上对应加减的时间。支持表达式后面的加减 1d（加减1天），也可以写成加减 N 天或者加减 N 小时，例如 `${yyyyMMdd-5d}`、`${yyyyMMdd+5d}`、`${yyyyMMdd+5h}`、`${yyyyMMdd-5h}` 都可以支持，对应的替换方式和前面描述的一致。



说明:

目前 E-MapReduce 仅支持小时和天维度的加减，即只支持在 `dateexpr` 后面 `+Nd`、`-Nd`、`+Nh`、`-Nh` 的形式（`dateexpr` 为时间格式表达式，N 为整数）。

示例

1. 在作业编辑页面单击右上角作业设置。
2. 在配置参数部分单击添加图标新增参数，并按照上文介绍的变量通配符格式填写参数，如下图所示：
3. 配置完成后就可以在作业中引用配置的参数的key了。

7.3 Hive SQL 作业配置

本文介绍 Hive SQL 作业配置的操作步骤。

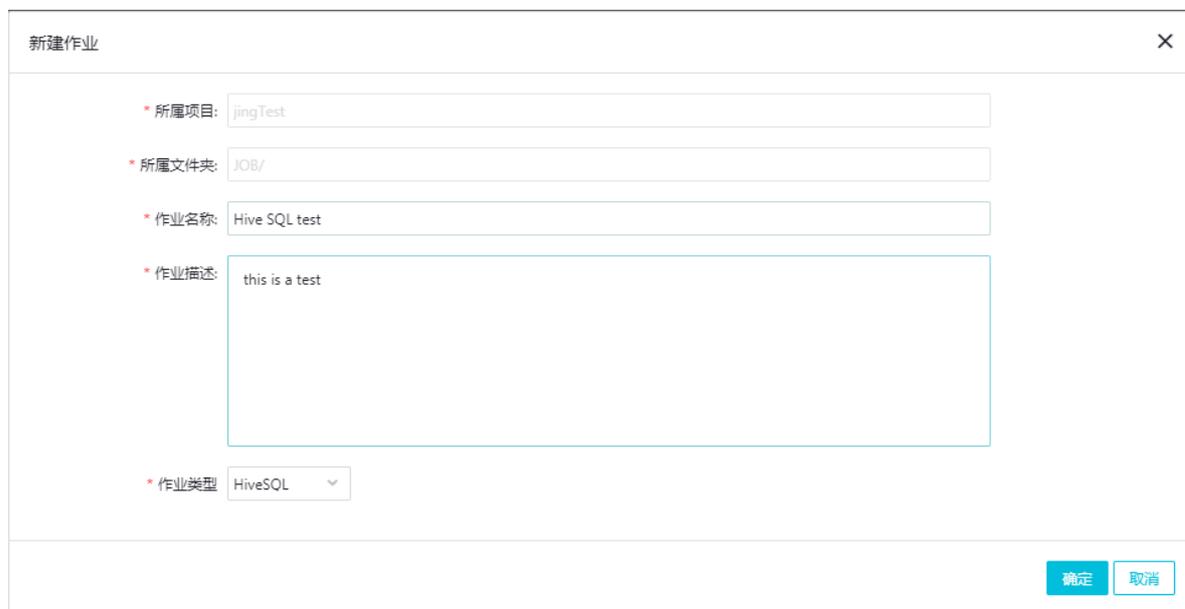
操作步骤

1. 通过主账号登录[阿里云 E-MapReduce 控制台](#)。
2. 单击上方的数据开发页签，进入项目列表页面。
3. 单击对应项目右侧的工作流设计，在左侧导航栏中单击作业编辑进入作业编辑页面。
4. 在页面左侧，在需要操作的文件夹上单击右键，选择新建作业

5. 填写作业名称，作业描述；选择 Hive SQL 作业类型，表示创建的作业是一个 Hive SQL 作业。Hive SQL 作业在 E-MapReduce 后台使用以下的方式提交：

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

其中 SQL_CONTENT 为作业编辑器中填写的 SQL 语句。



新建作业

* 所属项目: jingTest

* 所属文件夹: JOB/

* 作业名称: Hive SQL test

* 作业描述: this is a test

* 作业类型: HiveSQL

确定 取消

6. 单击确定。



说明:

您可以通过在文件夹上单击右键，进行创建子文件夹、重命名文件夹和删除文件夹操作。

7. 在作业内容输入框中填入 Hive SQL 语句，例如：

```
-- SQL语句示例  
-- SQL语句最大不能超过64KB  
show databases;  
show tables;  
-- 系统会自动为SELECT语句加上'limit 2000'的限制
```

```
select * from test1;
```



The screenshot shows a web-based code editor for HIVE_SQL. The main editor area contains the following text:

```
1 -- SQL语句示例
2 -- SQL语句最大不能超过64KB
3 show databases;
4 show tables;
5 -- 系统会自动为SELECT语句加上'limit 2000'的限制
6 select * from test1;
```

Below the editor is a terminal window titled "实际运行(仅供参考)" (Actual execution, for reference only). It shows the execution of the same SQL commands in a shell environment:

```
hive -e "-- SQL语句示例
-- SQL语句最大不能超过64KB
show databases;
show tables;
-- 系统会自动为SELECT语句加上'limit 2000'的限制
select * from test1;"
```

8. 单击保存，作业配置即定义完成。

7.4 Hadoop MapReduce 作业配置

本文介绍 Hadoop MapReduce 作业配置的操作步骤。

1. 通过主账号登录[阿里云 E-MapReduce 控制台](#)。
2. 单击上方的数据开发页签，进入项目列表页面。
3. 单击对应项目右侧的工作流设计，在左侧导航栏中单击作业编辑进入作业编辑页面。
4. 在页面左侧，在需要操作的文件夹上单击右键，选择新建作业。
5. 填写作业名称，作业描述；选择 Hadoop 作业类型。表示创建的作业是一个 Hadoop Mapreduce 作业。这种类型的作业，其后台实际上是通过以下的方式提交的 Hadoop 作业。

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

6. 单击确定。

 **说明：**
您还可以通过在文件夹上单击右键，进行创建子文件夹、重命名文件夹和删除文件夹操作。

7. 在作业内容输入框中填写提交该作业需要提供的命令行参数。所填写的命令行参数需要自jadoop jar 命令后的第一个参数开始填写，即在输入框中首先填写运行该作业所需的 jar 包所在路径，再填写 [MainClass] 和其它您想要设置的命令行参数。

例如，您想要提交一个 Hadoop 的 sleep 作业，该作业不读写任何数据、只提交一些 mapper 和 reducer task 到集群中，且每个 task 执行时需要 sleep 一段时间。在 Hadoop (以 hadoop-2.6.0版本为例) 中，该作业处于 Hadoop 发行版的 *hadoop-mapreduce-client-*

`jobclient-2.6.0-tests.jar` 包文件中。这种情况下，如果您通过命令行的方式提交该作业，需要执行以下命令：

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

而在 E-MapReduce 中配置这个作业，则应在作业内容输入框中填写以下内容：

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



说明：

这里用的 jar 包路径是 E-MapReduce 宿主机上的一个绝对路径，这种方式有一个问题，就是用户可能会将这些 jar 包放在任何位置，而且随着集群的创建和释放，这些 jar 包也会跟着释放而变得不可用。所以，请使用以下方法上传 jar 包：

- a. 用户将自己的 jar 包上传到 OSS 的 bucket 中进行存储，当配置 Hadoop 的参数时，单击选择 OSS 路径，从 OSS 目录中进行选择要执行的 jar 包。系统会为用户自动补齐 jar 包所在的 OSS 地址。请务必将代码的 jar 的前缀切换为 `ossref`（单击切换资源类型），以保证这个 jar 包会被 E-MapReduce 正确下载。
- b. 单击确定，该 jar 包所在的 OSS 路径地址就会自动填充到应用参数选项框中。作业提交的时候，系统能够根据这个路径地址自动从 OSS 找到相应的 jar 包。
- c. 在该 OSS 的 jar 包路径后面，即可进一步填写作业运行的其他命令行参数。

8. 单击保存，作业配置即定义完成。

上面的例子中，`sleep` 作业并没有数据的输入输出，如果作业要读取数据，并输出处理结果（比如 `wordcount`），则需要指定数据的 `input` 路径和 `output` 路径。用户可以读写 E-MapReduce 集群 HDFS 上的数据，同样也可以读写 OSS 上的数据。如果需要读写 OSS 上的数据，只需要在填写 `input` 路径和 `output` 路径时，数据路径写成 OSS 上的路径地址即可，例如：

```
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 Hive 作业配置

E-MapReduce 默认为提供了 Hive 环境，用户可以直接使用 Hive 来创建和操作自己的表和数据。

操作步骤

1. 用户需要提前准备好 Hive SQL 的脚本，例如：

```
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. 保存该脚本文件（例如 `uservisits_aggre_hdfs.hive`），然后上传到 OSS 的某个目录中（例如 `oss://path/to/uservisits_aggre_hdfs.hive`）。
3. 通过主账号登录[阿里云 E-MapReduce 控制台](#)。
4. 单击上方的数据开发页签，进入项目列表页面。
5. 单击对应项目右侧的工作流设计，在左侧导航栏中单击作业编辑进入作业编辑页面。
6. 在页面左侧，在需要操作的文件夹上单击右键，选择新建作业。
7. 填写作业名称，作业描述。
8. 选择 Hive 作业类型，表示创建的作业是一个 Hive 作业。这种类型的作业，其后台实际上是通过以下的方式提交。

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

9. 单击确定。



说明：

您还可以通过在文件夹上单击右键，进行创建子文件夹、重命名文件夹和删除文件夹操作。

10.在作业内容输入框中填入 Hive 命令后续的参数。例如，如果需要使用刚刚上传到 OSS 的 Hive 脚本，则填写的内容如下：

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

您也可以单击选择 OSS 路径，从 OSS 中进行浏览和选择，系统会自动补齐 OSS 上 Hive 脚本的绝对路径。请务必将 Hive 脚本的前缀修改为 ossref（单击切换资源类型），以保证 E-MapReduce 可以正确下载该文件。

11.单击保存，Shell 作业即定义完成。

7.6 Pig 作业配置

E-MapReduce 中，用户申请集群的时候，默认为用户提供了 Pig 环境，用户可以直接使用 Pig 来创建和操作自己的表和数据。

操作步骤

1. 用户需要提前准备好 Pig 的脚本，例如：

```
```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 YMMDDHHMMSS.
-- Call the ExtractHour UDF to extract the hour (HH) from the time
field.
houred = 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 houred 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. 将该脚本保存到一个脚本文件中，例如叫 `script1-hadoop-oss.pig`，然后将该脚本上传到 OSS 的某个目录中（例如：`oss://path/to/script1-hadoop-oss.pig`）。
3. 通过主账号登录[阿里云 E-MapReduce 控制台](#)。
4. 单击上方的数据开发页签，进入项目列表页面。
5. 单击对应项目右侧的工作流设计，在左侧导航栏中单击作业编辑进入作业编辑页面。
6. 在页面左侧，在需要操作的文件夹上单击右键，选择新建作业。

7. 填写作业名称，作业描述。
8. 选择 Pig 作业类型，表示创建的作业是一个 Pig 作业。这种类型的作业，其后台实际上是通过以下的方式提交。

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

9. 单击确定。



说明:

您还可以通过在文件夹上单击右键，进行创建子文件夹、重命名文件夹和删除文件夹操作。

10. 在作业内容输入框中填入 Pig 命令后续的参数。例如，如果需要使用刚刚上传到 OSS 的 Pig 脚本，则填写如下：

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

您也可以单击选择 OSS 路径，从 OSS 中进行浏览和选择，系统会自动补齐 OSS 上 Pig 脚本的绝对路径。请务必将 Pig 脚本的前缀修改为 ossref（单击切换资源类型），以保证 E-MapReduce 可以正确下载该文件。

11. 单击保存，Shell 作业即定义完成。

## 7.7 Spark 作业配置

本文介绍 Spark 作业配置的操作步骤。

### 操作步骤

1. 通过主账号登录[阿里云 E-MapReduce 控制台](#)，进入集群列表页面。
2. 单击上方的数据开发页签，进入项目列表页面。
3. 单击对应项目右侧的工作流设计，在左侧导航栏中单击作业编辑进入作业编辑页面。
4. 在页面左侧，在需要操作的文件夹上单击右键，选择新建作业。
5. 填写作业名称，作业描述。
6. 选择 Spark 作业类型，表示创建的作业是一个 Spark 作业。Spark 作业在 E-MapReduce 后台使用以下的方式提交：

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

7. 单击确定。



说明:

您还可以通过在文件夹上单击右键，进行创建子文件夹、重命名文件夹和删除文件夹操作。

8. 在作业内容输入框中填写提交该 Spark 作业需要的命令行参数。请注意，应用参数框中只需要填写spark-submit之后的参数即可。以下分别示例如何填写创建 Spark 作业和 pyspark 作业的参数。

· 创建 Spark 作业

新建一个 Spark WordCount 作业。

- 作业名称：Wordcount
- 类型：选择 Spark
- 应用参数：

■ 在命令行下完整的提交命令是：

```
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
```

■ 在 E-MapReduce 作业的作业内容输入框中只需要填写：

```
--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
```



注意：

作业 jar 包保存在 OSS 中，引用这个 jar 包的方式是 `ossref://emr/checklist/jars/emr-checklist_2.10-0.1.0.jar`。您可以单击选择OSS路径，从 OSS 中进

行浏览和选择，系统会自动补齐 OSS 上 Spark 脚本的绝对路径。请务必将默认的 OSS 协议切换到 ossref 协议。

- 创建 pyspark 作业

E-MapReduce 除了支持 Scala 或者 Java 类型作业外，还支持 python 类型 Spark 作业。以下新建一个 python 脚本的 Spark Kmeans 作业。

- 作业名称: Python-Kmeans
- 类型: Spark
- 应用参数:

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

- 支持 Python 脚本资源的引用，同样使用 ossref 协议。
- pyspark 目前不支持在线安装 Python 工具包。

9. 单击保存，Spark 作业即定义完成。

## 7.8 Spark SQL 作业配置

本文介绍 Spark SQL 作业配置的操作步骤。



说明:

Spark SQL 提交作业的模式默认是 yarn-client 模式。

### 操作步骤

1. 通过主账号登录[阿里云 E-MapReduce 控制台](#)，进入集群列表页面。
2. 单击上方的数据开发页签，进入项目列表页面。
3. 单击对应项目右侧的工作流设计，在左侧导航栏中单击作业编辑进入作业编辑页面。
4. 在页面左侧，在需要操作的文件夹上单击右键，选择新建作业。
5. 填写作业名称，作业描述。
6. 选择 Spark SQL 作业类型，表示创建的作业是一个 Spark SQL 作业。Spark SQL 作业在 E-MapReduce 后台使用以下的方式提交：

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

- options: 通过在作业配置->高级配置->添加环境变量 SPARK\_CLI\_PARAMS 来设置，如 SPARK\_CLI\_PARAMS="--executor-memory 1g --executor-cores
- SQL\_CONTENT: 作业编辑器中填写的 SQL 语句。

7. 单击确定。



说明:

您还可以通过在文件夹上单击右键，进行创建子文件夹、重命名文件夹和删除文件夹操作。

8. 在作业内容输入框中填入 Spark SQL 语句，如：

```
-- SQL语句示例
-- SQL语句最大不能超过64KB
show databases;
show tables;
-- 系统会自动为SELECT语句加上'limit 2000'的限制
select * from test1;
```

9. 单击保存，Spark SQL 作业即定义完成。

## 7.9 Shell 作业配置

本文介绍 Shell 作业配置的操作步骤。



注意:

目前 Shell 脚本默认是使用 Hadoop 用户执行的，如果需要使用 root 用户，可以使用 sudo 命令。请谨慎使用 Shell 脚本作业。

### 操作步骤

1. 通过主账号登录[阿里云 E-MapReduce 控制台](#)，进入集群列表页面。
2. 单击上方的数据开发页签。
3. 单击对应项目右侧的工作流设计，在左侧导航栏中单击作业编辑进入作业编辑页面。
4. 在页面左侧，在需要操作的文件夹上单击右键，选择新建作业。
5. 填写作业名称，作业描述。
6. 选择 Shell 作业类型，表示创建的作业是一个 Bash Shell 作业。
7. 单击确定。



说明:

您还可以通过在文件夹上单击右键，进行创建子文件夹、重命名文件夹和删除文件夹操作。

8. 在作业内容输入框中填入 Shell 命令后续的参数。

- -c 选项

-c 选项可以直接设置要运行的 Shell 脚本，在作业内容输入框中直接输入，如下所示：

```
-c "echo 1; sleep 2; echo 2; sleep 4; echo 3; sleep 8; echo 4;
sleep 16; echo 5; sleep 32; echo 6; sleep 64; echo 8; sleep 128;
echo finished"
```

- -f 选项

-f 选项可以直接运行 Shell 脚本文件。通过将 Shell 脚本文件上传到 OSS 上，在 job 参数里面可以直接制定 OSS 上的 Shell 脚本，比使用 -c 选项更加灵活，如下所示：

```
-f ossref://mxbucket/sample/sample-shell-job.sh
```

9. 单击保存，Shell 作业即定义完成。

## 7.10 Sqoop 作业配置

本文介绍 Sqoop 作业配置的操作步骤。



说明：

只有 E-MapReduce 产品版本 V1.3.0（包括）以上支持 Sqoop 作业类型。在低版本集群上运行 Sqoop 作业会失败，errlog 会报不支持的错误。参数细节请参见[数据传输 Sqoop](#)。

### 操作步骤

1. 通过主账号登录[阿里云 E-MapReduce 控制台](#)，进入集群列表页面。
2. 单击上方的数据开发页签，进入项目列表页面。
3. 单击对应项目右侧的工作流设计，在左侧导航栏中单击作业编辑进入作业编辑页面。
4. 在页面左侧，在需要操作的文件夹上单击右键，选择新建作业。
5. 填写作业名称，作业描述；选择 Sqoop 作业类型，表示创建的作业是一个 Sqoop 作业。Sqoop 作业在 E-MapReduce 后台使用以下的方式提交：

```
sqoop [args]
```

6. 在作业内容输入框中填入 Sqoop 命令后续的参数。
7. 单击确定。



说明：

您还可以通过在文件夹上单击右键，进行创建子文件夹、重命名文件夹和删除文件夹操作。

8. 单击保存，Sqoop 作业即定义完成。

## 7.11 Flink作业配置

本文介绍 Flink 作业配置的操作步骤。

### 前提条件

- 已创建好项目，详情请参见[项目管理](#)。
- 已获取作业所需的资源，以及作业要处理的数据文件，例如，JAR包、数据文件名称，以及两者的保存路径。

### 操作步骤

1. 通过主账号登录[阿里云 E-MapReduce 控制台](#)，进入集群列表页面。
2. 单击上方的数据开发页签，进入项目列表页面。
3. 单击对应项目右侧的工作流设计，然后在左侧导航栏中单击作业编辑。
4. 在作业编辑页面左侧，右键单击作业所属的文件夹并选择新建作业。



说明：

通过右键单击文件夹，您还可以进行创建子文件夹、重命名文件夹和删除文件夹操作。

5. 在弹出的新建作业对话框中，输入作业名称和作业描述，并从作业类型列表中选择Flink。
6. 完成上述参数配置后，单击确定，创建一个作业。
7. 创建作业完成后，您需要给作业配置内容。



Flink作业的作业内容示例如下：

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



注意：

如果作业JAR包保存在OSS中，则引用这个JAR包的方式是`ossref://xxx/.../xxx.jar`。您可以单击选择OSS路径，从OSS中进行浏览和选择，系统会自动补齐OSS上Flink脚本的绝对路径。请务必将默认的OSS协议切换成`ossref`协议。

如果是在E-MapReduce后台的命令行中，Flink作业提交命令的格式和示例如下：

- Flink作业提交命令的格式：

```
flink run [options] xxx.jar args
```

- 本例中Flink作业的提交命令：

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

8. 完成上述参数配置后，单击保存，Flink作业即定义完成。

## 7.12 Spark Streaming作业配置

本文介绍 Spark Streaming作业配置的操作步骤。

### 前提条件

- 已创建好项目，详情请参见[项目管理](#)。
- 已准备好作业所需的资源，以及作业要处理的数据。

### 操作步骤

1. 通过主账号登录[阿里云 E-MapReduce 控制台](#)，进入集群列表页面。
2. 单击上方的数据开发页签，进入项目列表页面。
3. 单击对应项目右侧的工作流设计，然后在左侧导航栏中单击作业编辑。
4. 在作业编辑页面左侧，右键单击作业所属的文件夹并选择新建作业。



说明：

通过右键单击文件夹，您还可以进行创建子文件夹、重命名文件夹和删除文件夹操作。

5. 在弹出的新建作业对话框中，输入作业名称和作业描述，并从作业类型列表中选择Spark Streaming。
6. 完成上述参数配置后，单击确定，创建一个作业。
7. 创建作业完成后，您需要给作业配置内容。

作业名称以SlsStreaming为例，作业的作业内容示例如下：

```
--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>
```



注意:

如果作业JAR包保存在OSS中, 则引用这个JAR包的方式是`ossref://xxx/.../xxx.jar`。您可以单击选择OSS路径, 从OSS中进行浏览和选择, 系统会自动补齐OSS上Spark Streaming脚本的绝对路径。请务必将默认的OSS协议切换到`ossref`协议。

如果是在E-MapReduce后台的命令行中, Spark Streaming作业提交命令的格式和示例如下:

- Spark Streaming作业提交命令的格式:

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

- 本例中Spark Streaming作业的提交命令:

```
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. 完成上述参数配置后, 单击保存, Spark Streaming作业即定义完成。

## 7.13 Streaming SQL作业配置

本文介绍Streaming SQL作业配置的操作步骤。

### 前提条件

- 已创建好项目, 详情请参见[项目管理](#)。
- 已获取Spark Streaming SQL的依赖库, 详情请参见下面的背景信息。

### 背景信息

Streaming SQL的详细信息请参见[Spark Streaming SQL](#)。

在Streaming SQL作业配置过程中, 您需要设置依赖库。以下列出了Spark Streaming SQL提供的数据库依赖包的版本信息和使用说明, 原则上需要使用最新版本。

| 库名称                | 版本    | 发布日期       | 引用字符串                                            | 详细信息                                            |
|--------------------|-------|------------|--------------------------------------------------|-------------------------------------------------|
| datasources-bundle | 1.7.0 | 2019/07/29 | sharedlibs:streamingsql:datasources-bundle:1.7.0 | 支持数据源: Kafka、Loghub、Druid、TableStore、HBase和JDBC |

- 引用字符串在数据开发的作业设置 > 流任务设置 > 依赖库中使用。

- 以上所注明支持的数据源，特指数据源支持了流式读写。
- 如果需要了解更详细的使用方法，请参见[数据源](#)。

#### 步骤一：创建Streaming SQL作业

1. 通过主账号登录[阿里云 E-MapReduce 控制台](#)，进入集群列表页面。
2. 单击上方的数据开发页签，进入项目列表页面。
3. 单击对应项目右侧的工作流设计，然后在左侧导航栏中单击作业编辑。
4. 在作业编辑页面左侧，右键单击作业所属的文件夹并选择新建作业。



说明：

通过右键单击文件夹，您还可以进行创建子文件夹、重命名文件夹和删除文件夹操作。

5. 在弹出的新建作业对话框中，输入作业名称和作业描述，并从作业类型列表中选择Streaming SQL。

新建作业

\* 所属项目: integration\_test\_project

\* 所属文件夹: JOB/

\* 作业名称: Spark Streaming SQL Sample

\* 作业描述: Spark Streaming SQL Sample

\* 作业类型: Streaming SQL

确定 取消

6. 单击确定，完成Streaming SQL的作业创建。

作业创建完成后，自动进入该作业，您可根据实际需要配置作业的代码。

#### 步骤二：配置作业的Streaming SQL语句

在E-MapReduce后台，Streaming SQL作业的提交方式是`streaming-sql -f {SQL_SCRIPT}`，其中SQL\_SCRIPT中保存的即是Streaming SQL作业的代码，即Streaming SQL语句。

创建作业完成后，在作业内容文本框中输入Streaming SQL语句。

Streaming SQL语句示例：

```
--- 创建SLS数据表
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}'
);
--- 将数据导入HDFS
INSERT INTO
 ${hdfsTableName}
SELECT
 col1, col2
FROM ${slsTableName}
WHERE ${condition}

```



### 步骤三：配置依赖库和失败策略

**依赖库：**Streaming SQL作业需要依赖一些数据源相关的库文件。E-MapReduce将这些库以依赖库的形式发布在调度服务的仓库中，在创建作业时需要指定使用哪个版本的依赖库。

**失败策略：**当前语句执行失败时的执行策略。

1. 完成作业代码配置后，单击右上方的作业设置，然后选择流任务设置。
2. 在流任务设置页面配置作业的依赖库和失败策略。

| 区域     | 配置项       | 说明                                                                                                                                                |
|--------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| 失败处理策略 | 当前语句执行失败时 | <p>当前语句执行失败时，支持如下策略：</p> <ul style="list-style-type: none"> <li>· 继续执行下一条语句：如果查询语句执行失败，继续执行下一条语句。</li> <li>· 终止当前作业：如果查询语句执行失败，终止当前作业。</li> </ul> |

| 区域  | 配置项 | 说明                                                                               |
|-----|-----|----------------------------------------------------------------------------------|
| 依赖库 | 库列表 | 您只需设置相应的依赖库版本，例如 <code>sharedlibs:streamingsql:datasources-bundle:1.7.0</code> 。 |

- 单击保存，Streaming SQL作业配置即定义完成。

## 8 老版作业调度（即将下线）

---

### 8.1 交互式工作台

#### 8.1.1 交互式工作台简介

交互式工作台提供在 E-MapReduce 管理控制台直接编写并运行 Spark, SparkSql, HiveSql 任务的能力，您可以在工作台直接看到运行结果。交互式工作台适合处理运行时间较短、想要直接看到数据结果、调试性质的任务，对于运行时间很长，需要定期执行的任务应使用作业和执行计划功能。本节会介绍如何新建演示任务并运行，其他示例和操作说明请参考后面的章节。

##### 创建演示任务

1. 登录[阿里云 E-MapReduce 控制台](#)。
2. 单击上方的老版作业调度。
3. 在左侧导航栏中单击交互式工作台。
4. 单击新建演示任务。
5. 弹出确认框，提示运行需要的集群环境，单击确认创建演示任务。会新建三个示例的交互式任务。

##### 运行Spark演示任务

1. 单击 EMR-Spark-Demo，显示 Spark 的交互式示例。运行之前首先要关联一个已经创建好的集群，单击在可用集群列表中选择。注意关联的集群必须是 EMR-2.3 以上版本，不小于三节点，4 核 8 G 及以上配置。
2. 关联后，单击运行。关联的集群第一次执行 Spark/SparkSQL 交互式任务时会额外花费一些时间构建 Spark 上下文和运行环境，大概要 1 分钟，后续的执行就不需要再耗时构建了。运行结果如下所示：

##### 运行 SparkSQL 演示任务

1. 单击 EMR-Spark-Demo，显示 SparkSQL 的交互式示例。运行之前依然要先关联一个已经创建好的集群，单击右上角在可用集群列表中选择。

2. SparkSQL 的演示任务有好几个演示段落，每个段落可以单独运行，也可以通过运行全部运行。运行后可以看到各段落返回的数据结果。



说明:

创建表的段落如果运行多次会报错提示表已存在。

### 运行Hive演示任务

1. 单击 EMR-Hive-Demo，显示 Hive 的交互式示例。运行之前依然要先关联一个已经创建好的集群，单击右上角在可用集群列表中选择一个。
2. Hive 的演示任务有好几个演示段落，每个段落可以单独运行，也可以通过运行全部运行。运行后可以看到各段落返回的数据结果。



说明:

- 关联的集群第一次执行hive交互式任务时会额外花费一些时间构建 Hive 客户端运行环境，大概要几十秒，后续的执行就不需要再耗时构建了。
- 创建表的段落如果运行多次会报错提示表已存在。

### 取消关联集群

集群运行过交互式任务后，为了再次执行时能够快速响应，会创建进程缓存一些上下文运行环境。如果您暂时不再执行交互式任务，想要释放缓存占用的集群资源，可以把运行过的交互式任务都取消关联，会释放掉原关联集群上占用的内存资源。

## 8.1.2 交互式工作台操作说明

本文向您介绍，如何在 E-MapReduce 控制台上新建交互式任务，并指导您完成任务的创建和运行。

### 新建交互式任务



说明:

要运行交互式任务的集群的配置，必须满足 EMR-2.3 及以上版本，不小于三节点，4 核 8 G 及以上配置。

1. 登录[阿里云 E-MapReduce 控制台](#)。
2. 单击上方的老版作业调度。

3. 在左侧导航栏中单击交互式工作台。
4. 单击右侧新建交互式任务或文件 > 新建交互式任务。
5. 填入名称, 选择默认类型, 关联集群可选, 单击确认新建一个交互式任务。

类型目前支持三类, Spark 可以编写 scala spark 代码, Spark SQL 可以写 Spark 支持的 sql 语句, Hive 可以写 Hive 支持的 sql 语句。

6. 关联集群, 需要是一个创建好的集群, 且必须是 EMR-2.3 及以上版本, 不小于三节点, 4 核 8 G 及以上配置。也可以先不关联, 在运行前再关联。

目前一个账户最多创建 20 个交互式任务。

#### 填写保存段落

段落是运行任务的最小单元, 1 个交互式任务可以填写多个段落。每个段落可以在内容开头写 %spark, %sql, %hive 表明该段落是 Scala Spark 代码段, Spark Sql, 还是 Hive sql。类型前缀以空格或换行和实际内容分割, 不写类型前缀则以交互式任务的默认类型作为该段的运行类型。

一个创建 spark 临时表的示例如下:

将如下代码粘贴进段落内, 会显示一个红\*提醒有修改, 通过保存段落按钮或运行按钮可以保存对段落内容的修改, 单击段落下方的+可以新建一个段落。目前一个交互式任务最多可以创建 30 个段落。

```
%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\n").map(
 s => Bank(s(0).toInt,
 s(1).replaceAll("\"", ""),
 s(2).replaceAll("\"", ""),
 s(3).replaceAll("\"", ""),
 s(5).replaceAll("\"", "").toInt
)
).toDF()
bank.registerTempTable("bank")
```

## 运行段落

运行之前首先要关联一个已经创建好的集群，如果创建交互式任务时未关联，右上角显示未关联，单击在可用集群列表中选择。注意关联的集群必须是 EMR-2.3 以上版本，不小于三节点，4 核 8 G 即以上配置。

单击运行按钮，会自动保存当前段落，运行内容，如果这是最后一个段落会自动新建一个段落。

运行后会显示当前的运行状态，还未实际运行的是 PENDING，运行后是 RUNNING。运行完成是 FINISHED，如果有错误是 ERROR，运行结果会显示在段落的运行按钮下方。运行时可以点运行按钮下方的取消按钮取消运行，取消的状态显示 ABORT。

段落可以反复多次运行，只保留最后一次运行的结果。运行时不能修改段落的输入内容，运行后可以修改。

## 运行全部

交互式任务可以单击菜单栏上的运行全部运行所有的段落，段落会顺序提交运行。不同的类型有独立的执行队列，如果一个交互式任务包含多种段落类型，顺序提交运行后，实际在集群上的执行顺序是按照类型划分的。Spark 和 Spark sql 类型是顺序一个个的执行。Hive 支持并发执行，同一个集群交互式段落最大并发数是 10。注意并发运行的作业同时受集群资源限制，集群规模小并发很多依然要在 YARN 上排队。

## 取消关联集群

集群运行过交互式任务后，为了再次执行时能够快速响应，会创建进程缓存一些上下文运行环境。如果您暂时不再执行交互式任务，想要释放缓存占用的集群资源，可以把运行过的交互式任务都取消关联，会释放掉原关联集群上占用的内存资源。

## 其他操作项

- 段落操作

- 隐藏结果/显示结果

- 可以将段落的结果隐藏掉，只显示段落的输入内容。

- 删除

- 删除当前段落，运行中的段落也可以删除。

- 文件菜单

- 新建交互式任务

- 新建一个交互式任务，并切换界面到新建的交互式任务上。

- 新建段落

- 在交互式任务的尾部添加一个新段落，一个交互式任务最多有30个段落。

- 保存所有段落

- 所有修改过的段落都会保存

- 删除交互式任务

- 删除掉当前的交互式任务。如果关联了集群会同时取消关联。

- 视图

- 只显示代码/显示代码和结果

- 所有段落只显示输入的代码，还是同时显示结果内容。

## 8.1.3 交互式工作台示例

### 8.1.3.1 银行员工信息查询示例

#### 段落 1 创建临时表

```
%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")

```

## 段落 2 查询表结构

```

%sql
desc bank

```

## 段落 3 查询年龄小于 30 各年龄段员工人数

```

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

```

## 段落 4 查询年龄小于等于 20 岁的员工信息

```

%sql select * from bank where age <= 20

```

### 8.1.3.2 视频播放数据示例

#### 数据准备

本示例需要您从 OSS 上下载数据，并上传到您自己的 OSS bucket 上。数据包含

- [用户表示例数据](#)
- [视频表示例数据](#)
- [播放表示例数据](#)

分别上传到您 OSS bucket 指定目录的 `userinfo` 子目录，`videoinfo` 目录，`playvideo` 目录。例如 bucket `example` 下的 `demo/userinfo` 目录。

将下面创建表的 sql 中 `[bucketname]` 替换成您的 bucket 名字例如 `example`，`[region]` 替换成您用的 OSS 地域名如 `hangzhou`，`[bucketpath]` 替换成您 OSS 的指定的路径前缀例如 `demo`。

#### 段落 1 创建用户表

```

%hive
CREATE EXTERNAL TABLE user_info(id int,sex int,age int, marital_st
atus int) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LOCATION 'oss

```

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

## 段落 2 创建视频表

```
%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'
```

## 段落 3 创建播放表

```
%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'
```

## 段落 4 用户表计数

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

## 段落 5 视频表计数

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

## 段落 6 播放表计数

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

## 段落 7 统计各类型视频播放数

```
%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
```

## 段落 8 播放数 top 10 的视频信息

```
%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
```

## 段落 9 播放数最高视频观看者的年龄分布

```
%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
```

## 段落 10 播放数最高视频观看者的性别, 年龄段, 婚姻状态分布汇总

```
%sql select if(sex=0,'女','男') as title, count(*) as count, '性别' 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 '小于15' when age<25 then '15-25'
' when age<35 then '25-35' else '大于35' end , count(*) as count, '年
年龄段' 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 '小于15' when age<25 then
'15-25' when age<35 then '25-35' else '大于35' end
union all
select if(marital_status=0,'未婚','已婚') as title, count(*) as count
, '婚否' 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 执行计划

### 8.2.1 创建执行计划

执行计划是一组作业的集合，他们通过调度上的配置，可以被一次性或者周期性的执行。他可以在一个现有的 E-MapReduce 集群上运行，也可以动态的按需创建一个临时集群来运行作业。它最大的优势就是跑多少就用多少资源，最大化的节省资源的浪费。

#### 操作步骤

1. 登录[阿里云 E-MapReduce 控制台](#)。
2. 选择地域 (Region) 。
3. 单击上方的老版作业调度页签，进入作业列表页面
4. 单击左侧的执行计划页签，进入执行计划页面
5. 单击右上角的创建执行计划，进入创建执行计划页面。
6. 在选择集群方式页面上，有两个选项，分别是按需创建和已有集群。
  - a. 按需创建：创建一个全新的集群，用来运行作业。
    - 一次性调度的执行计划，会在开始执行的时候创建对应配置的集群，并在运行完成以后释放该集群。具体创建参数说明参考[#unique\\_38](#)。
    - 周期调度的执行计划，会在每一个调度周期开始时，按照用户的设置创建出一个新的集群运行作业，并在运行结束后，释放集群。
  - b. 已有集群：使用一个已有的集群，并且该集群要符合以下要求：
    - 目前只有运行中和空闲这 2 个状态的集群可以被提交执行计划。

如果选择已有集群，则进入选择集群页面。用户可选择要将该执行计划关联到的集群。

7. 单击下一步，进入配置作业页面。左边表中会列出用户所有的作业，可以单击选中需要执行的作业，然后单击中央的右向按钮将作业加入已选作业队列。已选作业队列中的作业会被按排列顺序提交到集群中执行。同一个作业可以被添加多次，就会多次执行。如果您还没有创建任何作业，请您先参见创建作业的操作说明创建作业。

8. 单击下一步，进入配置调度方式页面。配置项说明如下：

a. 名称：长度限制为 1-64 个字符，只允许包含中文、字母、数字、' - '、' \_ '。

b. 调度策略

- 手动执行：创建完执行计划以后，并不会自动执行。需要用户手动执行。一旦已经在运行中了，不可以被再次执行。
- 周期调度：创建完执行计划以后，周期调度功能会立刻启动。并在用户设置的调度时间点上开始执行。可以在列表页面关闭周期调度。当调度执行开始的时候，上一周期的执行还未结束，本次调度就会被忽略。

c. 设置调度周期：可以有天或小时两种调度的周期。天默认是一天，且无法更改。若选择小时，则可设置具体间隔时间，范围从 1-23。

d. 首次执行时间：调度有效的开始时间。从这个时间开始，按照调度周期进行周期调度。第一次调度按照实际的时间满足要求的最近一个时间点开始调度。

9. 单击确认，完成执行计划的创建。

其他

· 周期调度示例

这个设置表示，从 2015 年 10 月 31 日 10 点 0 分开始第一次调度，以后每隔一天调度一次。第二次调度是 2015 年 11 月 1 日 10 点 0 分。

· 作业的执行顺序

执行计划中的作业，按照用户选择的作业在作业列表中的顺序，从第一个开始一直执行到最后一个。

· 多个执行计划的执行顺序

每一个执行计划都可以看做是一个整体。当多个执行计划被提交到同一个集群上后，每一个执行计划都会按照自身内部的作业顺序提交作业，和单个执行计划的顺序是一致的。而多个执行计划之间的作业是并行的。

· 实践示例 —— 前期作业调试

在作业的调试阶段，如果经常用按需自动创建集群的方式会比较慢，每次都需要启动集群会花费不少的时间。推荐的方式是：先手动创建一个集群，然后在执行计划中，选择关联该集群来运行作业，并设置调度方式为立即执行。调试的时候，每次都通过单击执行计划列表页上的“立即运行”来多次运行，查看结果。一旦作业调试完成，修改执行计划。将关联现有集群的方式，修改为按需创建新集群。并将调度方式修改为周期调度（视实际情况而定）。后续就可以按需自动跑任务了。

## 8.2.2 管理执行计划

您可以通过以下步骤管理，查看和修改执行计划。

1. 登录[阿里云 E-MapReduce 控制台](#)。
2. 选择地域（Region）。
3. 单击上方的老版作业调度页签，进入作业列表页面
4. 单击左侧的执行计划页签，进入执行计划页面
5. 找到相应的执行计划条目，单击其操作栏中的管理，进入执行计划管理页面。在这里您可以：

- 查看执行计划详情

您可以查看到该执行计划的名称、关联集群、作业配置等基本信息，还有其调度策略、调度状态、报警信息等。

- 修改执行计划



注意：

一个执行计划当前并没有在运行中且它没有在被周期调度中，才能够被修改。如果是一个立即执行的执行计划，只要它当前没有在运行中就可以被修改。如果是一个周期调度的执行计划，首先要等待它当前的运行结束，然后确认它是否正在被周期调度中，如果是请单击停止调度，然后才可修改执行计划。

### 独立修改

每一个单独的模块，都可以被独立的修改。单击条目右侧的修改图标即可进行修改。

- 配置报警通知

共有三类报警通知：

- 启动超时通知：周期调度任务，在指定时间点，没有正常调度，并在 10 分钟的超时时间内，仍然没有调度执行，发送报警通知。
- 执行失败通知：执行计划内有任何一个作业失败，发送报警通知。
- 执行成功通知：执行计划内的所有作业执行成功，发送通知。

- 运行与查看结果

在基本信息中的调度状态右侧，当执行计划可以被运行的时候，会有立即运行的按钮，单击后即会产生一次调度执行。

在页面的最下方，是运行记录的部分，会显示执行计划每次被执行的实例，可以方便用户查看对应的作业列表和日志。

## 8.2.3 执行计划列表

执行计划列表用来展示您所有的执行计划的基本信息。

- ID/名称：执行计划的 ID 和对应的名称。
- 最近执行集群：最近一次执行该执行计划的集群，是一个按需创建的集群或是一个关联的已有集群。如果是按需的，那么在集群的名字下面会显示（自动创建），表示这个集群是有 E-MapReduce 按需自动创建出来的，运行完成以后会自动释放。
- 最近运行：最近一次执行计划的运行状态。
  - 开始时间：最近一次执行计划开始的时间。
  - 运行时间：最近一次执行计划的运行时长。
  - 运行状态：最近一次执行计划的运行状态。
- 调度状态：是否在调度中还是调度已经停止。只有周期作业才会有调度状态。
- 操作
  - 管理：查看执行计划详情，修改执行计划。
  - 立即运行：非调度中且非运行中，才能手动运行。单击后，会立刻运行一次执行计划。
  - 更多
    - 启动调度/停止调度：当调度停止状态时出现启动调度，单击即会开始调度。当调度运行中显示停止调度，单击即会停止调度。只有周期执行计划才有此按钮。
    - 运行记录：单击会进入执行计划中的作业日志查看页面。
    - 删除：删除执行计划。调度中或者运行中的执行计划不能被删除。

## 8.2.4 作业结果和日志查看

本文将介绍如何查看作业结果和对应作业的运行日志。

执行记录查看

1. 登录[阿里云 E-MapReduce 控制台](#)。
2. 选择地域（Region）。
3. 单击上方的老版作业调度页签，进入作业列表页面
4. 单击左侧的执行计划页签，进入执行计划页面

5. 单击相应执行计划条目右侧操作中的更多 > 运行记录，即可进入执行记录页面。

- 执行序列 ID：本次执行记录的执行次数，表明了它在整个执行队列中的顺序位置。比如第一次执行就是1，第n次就是n。
- 运行状态：每一次执行记录的运行状态。
- 开始时间：执行计划开始运行的时间。
- 运行时间：到查看页面当时为止，一共运行的时间。
- 执行集群：执行计划运行的集群，可以是按需也可以是一个关联的已有集群。点击可以前往集群的详情页查看。
- 操作

查看作业列表：单击该按钮，即可进入单次执行计划的作业列表查看每个作业的执行情况。

#### 作业记录查看

在作业列表中可以查看单次执行计划的执行记录中的作业列表，以及每一个作业的具体信息。

- 作业执行序列ID：作业每一次执行都会产生一个对应的 ID，它和作业本身的 ID 是不同的。这个 ID 可以想象成作业每运行一次的一个记录的唯一标示，您可用其在 OSS 上进行日志查询。
- 名称：作业的名称。
- 状态：作业的运行状态。
- 类型：作业的类型。
- 开始时间：这个作业开始运行的时间，都已经转换为本地时间。
- 运行时间：这个作业一共运行了多久，以秒为单位。
- 操作
  - 停止作业：无论作业在提交中还是在运行中，都可以被停止。如果是提交中，那么停止作业会让这个作业不执行。如果是在运行中，那么这个作业会被 kill 掉。
  - stdout：记录 master 进程的标准输出（即通道 1）的所有输出内容。如果运行作业的集群没有打开日志保存，不会有此查看功能。
  - stderr：记录 master 进程的诊断输出（即通道 2）的所有输出内容。如果运行作业的集群没有打开日志保存，不会有此查看功能。
  - 实例日志：查看作业的所有 worker 的节点的日志。如果运行作业的集群没有打开日志保存，不会有此查看功能。

## 作业worker日志查看

- 云服务器实例 ID/IP：运行作业的 ECS 实例 ID，以及对应的内网 IP。
- 容器 ID：Yarn 运行的容器 ID。
- 类型：日志的不同类型。stdout 与 stderr，来自不同的输出。
- 操作

查看日志：单击对应的类型，查看对应的日志。

## 8.2.5 多执行计划并行执行

为了最大化利用集群的可用计算资源，目前可以将多个执行计划挂载到同一个集群来达到多个执行计划并行执行的效果。

总结为如下几点：

- 同一个执行计划内的作业是串行执行的，默认认为前序作业执行完毕，后序作业才能被提交执行。
- 在集群资源足够的情况下，如果想要让多个作业达到并行执行的效果，需要创建多个不同的执行计划，同时关联到同一个集群提交运行即可（默认一个集群最多支持 20 个执行计划同时执行）。
- 目前管控系统支持将关联到同一个集群的执行计划并行提交到 Yarn，但如果集群本身资源不足，还是可能阻塞在 Yarn 队列中等待调度。

创建执行计划并关联到集群的流程参见：[创建执行计划](#)。

## 8.3 创建作业

本文介绍老版 E-MapReduce 作业调度创建作业流程。

要运行一个计算任务，首先需要定义一个作业，其步骤如下：

1. 登录[阿里云 E-MapReduce 控制台](#)。
2. 选择地域（Region），则作业将会创建在对应的地域内。
3. 单击上方的老版作业调度页签，进入作业列表页面。
4. 单击该页右上角的创建作业，进入创建作业页面，如下图所示：
5. 填写作业名称。
6. 选择作业类型。

7. 填写作业的应用参数。应用参数需要完整填写该作业运行的 jar 包、作业的数据输入输出地址以及一些命令行参数，也就是将用户在命令行的所有参数填写在这里。如果有使用到 OSS 的路径，可以单击下方的选择 OSS 路径选择 OSS 资源路径。关于各作业类型的参数配置，请参见《用户指南》中的《作业》章节。
8. 实际执行命令。这里会显示作业在 ECS 上实际被执行的命令。用户如果把这个命令直接复制下来，就能够在 E-MapReduce 集群的命令行环境中直接运行。
9. 失败重试。可设定重试次数与重试间隔，默认否。
10. 选择执行失败后策略。暂停当前执行计划会在这个作业失败后，暂停当前整个执行计划，等待用户处理。而继续执行下一个作业在这个作业失败以后，会忽略这个错误继续执行后一个作业。
11. 单击确定完成创建。

### 作业示例

这是一个 Spark 类型的作业，应用参数中设置了相关的参数，输入输出路径等。



注意:

本作业仅仅示例，不能实际运行。

### OSS 与 OSSREF

oss:// 的前缀代表数据路径指向一个 OSS 路径，当要读写该数据的时候，这个指明了操作的路径，与 hdfs:// 类似。

ossref:// 同样是指向一个 OSS 的路径，不同的是它会将对应的代码资源下载到本地，然后将命令行中的路径替换为本地路径。它是用于更方便地运行一些本地代码，而不需要登录到机器上去上传代码和依赖的资源包。

上面的例子中，ossref://xxxxxx/xxx.jar 这个参数代表作业资源的 jar，这个 jar 存放在 OSS 上，在运行的时候，E-MapReduce 会自动下载到集群中运行。而跟在 jar 后面的 2 个 oss://xxxx 以及另外两个值则是作为参数出现，他们会被作为参数传递给 jar 中的主类来处理。



注意:

ossref 不可以用来下载过大的数据资源，否则会导致集群作业的失败。