# 数据治理中心（DataArts Studio）

## 故障排除

天翼云科技有限公司

# 目 录

# 1 产品咨询类

## 1.1 区域

### 什么是区域？

我们用区域来描述数据中心的位置，您可以在特定的区域创建资源。

- 区域（Region）指物理的数据中心。每个区域完全独立，这样可以实现最大程度的容错能力和稳定性。资源创建成功后不能更换区域。

### 如何选择区域？

建议就近选择靠近您或者您的目标用户的区域，这样可以减少网络时延，提高访问速度。

### 实例可以转移到另一个区域吗？

- 实例创建成功后，无法转移到另一个区域。

### 区域和终端节点

终端节点（Endpoint）即调用 API 的**请求地址**，不同服务不同区域的终端节点不同。Endpoint 可从企业管理员处获取。

## 1.2 用户已添加权限，还是无法查看已有的工作空间？

请查看该工作空间下是否已添加用户，如果没有，请参考以下步骤添加该用户。

### 添加成员和角色

1. 登录 DataArts Studio 控制台，进入工作空间列表页面。
2. 单击相应工作空间列表后的“编辑”，进入成员空间页面。

3. 单击空间成员下的“添加”，在弹出的“添加成员”对话框中选择“按用户添加”或“按用户组添加”，然后从“成员帐号”的下拉选项中选择用户或用户组，并设置角色。
4. 单击“确定”即可添加成功。添加完成后，您可以在空间成员列表中查看或修改已有的成员和对应角色，也可将空间成员从工作空间中删除。

## 1.3 DataArts Studio 的工作空间可以删除吗？

工作空间创建成功后，暂不支持删除空间的操作，您可以将不需要的工作空间禁用，以后仍可以重新启用工作空间。

## 1.4 实例试用/购买成功后，可以转移到其他帐号下吗？

不可以，实例试用/购买后不能转移到另一个账户。

## 1.5 DataArts Studio 是否支持版本降级？

已购买 DataArts Studio 实例后，不支持降级版本。

## 1.6 如何查看 DataArts Studio 的版本？

# 2 管理中心

## 2.1 创建数据连接需要注意哪些事项？

创建 DWS/MRS Hive/RDS/SparkSQL 类型的数据连接时，需要绑定由 CDM 集群提供的代理服务，目前不支持低于 1.8.6 版本的 CDM 集群。

## 2.2 为什么 DWS/Hive/HBase 数据连接突然无法获取数据库或表的信息？

可能是由于 CDM 集群被关闭或者并发冲突导致，您可以通过切换 agent 代理来临时规避此问题。

建议您通过以下措施解决此问题：

步骤 1 检查 CDM 集群是否被关机。

- 是，将 CDM 集群开机后，确认管理中心的数据连接恢复正常。
- 否，跳转至步骤 2。

步骤 2 检查该 CDM 集群是否同时被用于数据迁移作业和管理中心连接代理。

- 是，您可以错开数据迁移作业和管理中心连接代理的使用时间，或再创建 CDM 集群，与原有 CDM 集群分开使用。、
- 否，跳转至步骤 3。

步骤 3 直接重启该 CDM 集群，释放连接池资源。确认管理中心的数据连接恢复正常。

**----结束**

## 2.3 为什么在创建数据连接的界面上 MRS Hive/HBase 集群不显示？

出现该问题的可能原因有：

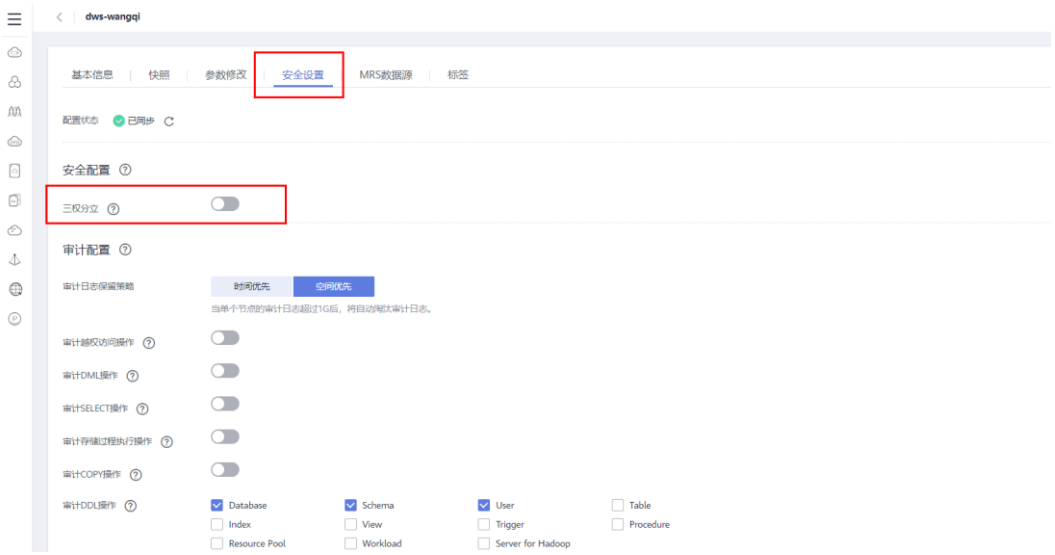
- 创建 MRS 集群时未选择 Hive/HBase 组件。
- 创建 MRS 数据连接时所选择的 CDM 集群和 MRS 集群网络不互通。

  CDM 集群作为网络代理，与 MRS 集群需网络互通才可以成功创建基于 MRS 的数据连接。

## 2.4 创建 DWS 数据连接，开启 SSL 连接时测试连接失败？

可能是由于 DWS 集群的三权分立功能导致的。请在 DWS 控制台，点击进入对应的 DWS 集群后，选择“安全设置”，然后关闭三权分立功能。

图2-1 关闭 DWS 集群三权分立功能




## 2.5 通过代理方式创建数据连接，一个空间可以创建多个连接吗？

同一个工作空间可以创建多个不同类型或相同类型的连接，但是连接的名字不能相同。

## 2.6 创建 DWS 连接的时候，连接方式是直接连还是通过代理连比较好？

连接方式一般选择代理连接即可。

## 2.7 如何将一个空间的数据开发作业和数据连接迁移到另一空间？

您可以在数据开发中将作业导出，随后在新空间数据开发中再导入作业。

您可以在管理中心中资源迁移进行数据连接的导入导出。

## 2.8 空间管理下创建的工作空间是否可以删除？

DataArts Studio 目前不支持删除工作空间，可以对工作空间名称进行编辑、更改。

# 3 数据集成

## 3.1 通用类

### 3.1.1 CDM 有哪些优势？

云数据迁移（Cloud Data Migration，简称 CDM）服务基于分布式计算框架，利用并行化处理技术，使用 CDM 迁移数据的优势如表 3-1 所示。

表3-1 CDM 优势

| 优势项 | 用户自行开发 | CDM |
| --- | --- | --- |
| 易使用 | 自行准备服务器资源，安装配置必要的软件并进行配置，等待时间长。<br>程序在读写两端会根据数据源类型，使用不同的访问接口，一般是数据源提供的对外接口，例如 JDBC、原生 API 等，因此在开发脚本时需要依赖大量的库、SDK 等，开发管理成本较高。 | CDM 提供了 Web 化的管理控制台，通过 Web 页实时开通服务。<br>用户只需要通过可视化界面对数据源和迁移任务进行配置，服务会对数据源和任务进行全面的管理和维护，用户只需关注数据迁移的具体逻辑，而不用关心环境等问题，极大降低了开发维护成本。<br>CDM 还提供了 REST API，支持第三方系统调用和集成。 |
| 实时监控 | 需要自行选型开发。 | 您可以使用云监控服务监控您的 CDM 集群，执行自动实时监控、告警和通知操作，帮助您更好地了解 CDM 集群的各项性能指标。 |
| 免运维 | 需要自行开发完善运维功能，自行保证系统可用性，尤其是告警及通知功能，否则只能人工值守。 | 使用 CDM 服务，用户不需要维护服务器、虚拟机等资源。CDM 的日志，监控和告警功能，有异常可以及时通知相关人员，避免 7*24 小时人工值守。 |
| 高效率 | 在迁移过程中，数据读写过程都是由一个单一任务完成的，受限于资 | CDM 任务基于分布式计算框架，自动将任务切分为独立的子任务并 |

| 优势项 | 用户自行开发 | CDM |
| --- | --- | --- |
| | 源，整体性能较低，对于海量数据场景往往不能满足要求。 | 行执行，能够极大提高数据迁移的效率。针对 Hive、HBase、MySQL、DWS（数据仓库服务）数据源，使用高效的数据导入接口导入数据。 |
| 多种数据源支持 | 数据源类型繁杂，针对不同数据源开发不同的任务，脚本数量成千上万。 | 支持数据库、Hadoop、NoSQL、数据仓库、文件等多种类型的数据源。 |
| 多种网络环境支持 | 随着云计算技术的发展，用户数据可能存在于各种环境中，例如公有云、自建/托管 IDC、混合场景等。在异构环境中进行数据迁移需要考虑网络连通性等因素，给开发和维护都带来较大难度。 | 无论数据是在用户本地自建的 IDC 中（Internet Data Center，互联网数据中心）、云服务中、第三方云中，或者使用 ECS 自建的数据库或文件系统中，CDM 均可帮助用户轻松应对各种数据迁移场景，包括数据上云，云上数据交换，以及云上数据回流本地业务系统。 |

## 3.1.2 CDM 有哪些安全防护？

CDM 是一个完全托管的服务，提供了以下安全防护能力保护用户数据安全。

- 实例隔离：CDM 服务的用户只能使用自己创建的实例，实例和实例之间是相互隔离的，不可相互访问。
- 系统加固：CDM 实例的操作系统进行了特别的安全加固，攻击者无法从 Internet 访问 CDM 实例的操作系统。
- 密钥加密：用户在 CDM 上创建连接输入的各种数据源的密钥，CDM 均采用高强度加密算法保存在 CDM 数据库。
- 无中间存储：数据在迁移的过程中，CDM 只处理数据映射和转换，而不会存储任何用户数据或片段。

## 3.1.3 如何降低 CDM 使用成本？

如果是迁移公网的数据上云，可以使用 NAT 网关服务，实现 CDM 服务与子网中的其他弹性云主机共享弹性 IP，可以更经济、更方便的通过 Internet 迁移本地数据中心或第三方云上的数据。

具体操作如下：

1. 假设已经创建好了 CDM 集群（无需为 CDM 集群绑定专用弹性 IP），记录下 CDM 集群所在的 VPC 和子网。
2. 创建 NAT 网关，注意选择和 CDM 集群相同的 VPC、子网。
3. 创建完 NAT 网关后，回到 NAT 网关控制台列表，单击创建好的网关名称，然后选择“添加 SNAT 规则”。
4. 选择子网和弹性 IP，如果没有弹性 IP，需要先申请一个。

完成之后，就可以到 CDM 控制台，通过 Internet 迁移公网的数据上云了。例如：迁移本地数据中心 FTP 服务器上的文件到 OBS、迁移第三方云上关系型数据库到云服务 RDS。

**说明**

如果用户对本地数据源的访问通道做了 SSL 加密，则 CDM 无法通过弹性 IP 连接数据源。

## 3.1.4 CDM 集群是否支持升级操作？

CDM 集群目前不支持升级操作，如果需要使用高版本集群则需要重新创建。

## 3.1.5 CDM 迁移性能如何？

单个 cdm.large 规格实例理论上可以支持 1TB～8TB/天的数据迁移，实际传输速率受公网带宽、集群规格、文件读写速度、作业并发数设置、磁盘读写性能等因素影响。

## 3.1.6 CDM 不同集群规格对应并发的作业数是多少？

CDM 不同集群规格对应并发的作业数如表 3-2 所示。

表3-2 并发任务数

| 产品规格 | cdm.large | cdm.xlarge | cdm.4xlarge |
| --- | --- | --- | --- |
| 规格 | 节点数量：1 个<br>vCPUs/内存：8 核 16GB<br>基准/最大带宽：0.8/3Gbit/s | 节点数量：1 个<br>vCPUs/内存：16 核 32GB<br>基准/最大带宽：4/10Gbit/s | 节点数量：1 个<br>vCPUs/内存：64 核 128GB<br>基准/最大带宽：36/40Gbit/s |
| 并发执行的作业数 | 30 | 100 | 300 |

包含但不限于以下情况，建议使用多个 CDM 集群进行业务分流：

- 作为不同的用途，例如用于数据迁移作业，或作为 DataArts Studio 管理中心连接代理。
- 给不同的业务部门使用，例如财务、网上商城等。

# 3.2 功能类

## 3.2.1 是否支持增量迁移？

CDM 支持增量数据迁移。利用定时任务配置和时间宏变量函数等参数，可支持以下场景的增量数据迁移：

- 文件增量迁移
- 关系数据库增量迁移

- 使用时间宏变量完成增量同步
- HBase/CloudTable 增量迁移

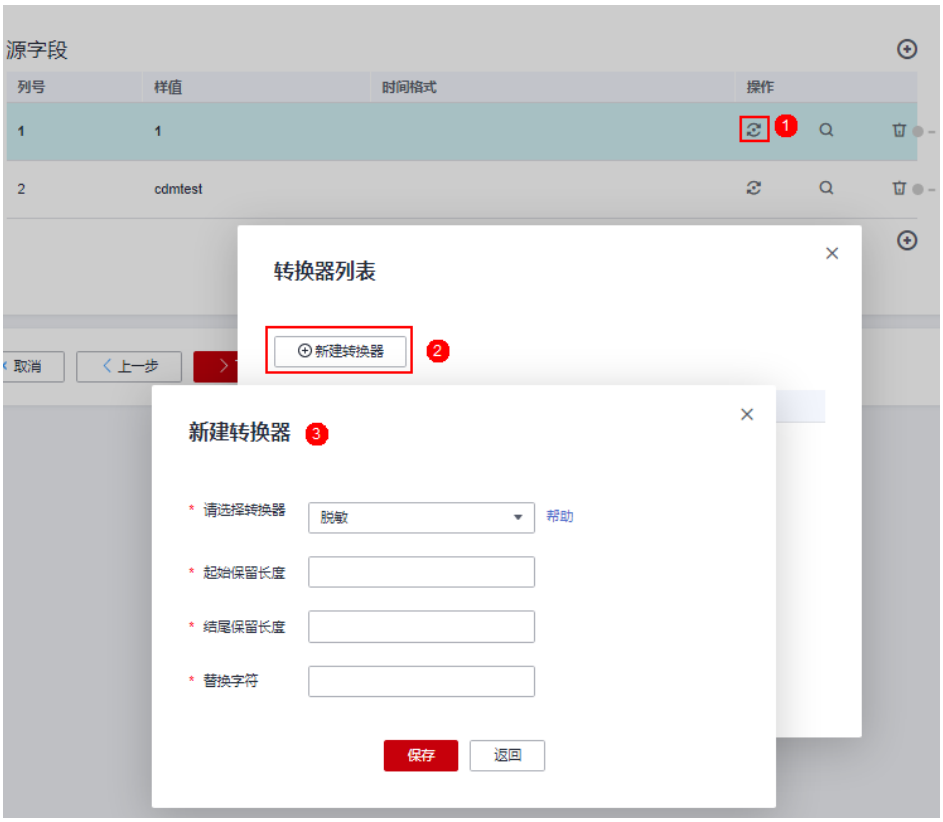
## 3.2.2 是否支持字段转换？

支持，CDM 支持以下字段转换器：

- 脱敏
- 去前后空格
- 字符串反转
- 字符串替换
- 表达式转换

在创建表/文件迁移作业的字段映射界面，可新建字段转换器，如图 3-1 所示。

图3-1 新建字段转换器




### 脱敏

隐藏字符串中的关键信息，例如要将“12345678910”转换为“123****8910”，则配置如下：

- “起始保留长度”为“3”。
- “结尾保留长度”为“4”。

- “替换字符”为“*”。

图3-2 字段脱敏


## 去前后空格

自动去字符串前后的空值，不需要配置参数。

## 字符串反转

自动反转字符串，例如将“ABC”转换为“CBA”，不需要配置参数。

## 字符串替换

替换字符串，需要用户配置被替换的对象，以及替换后的值。

## 表达式转换

使用 JSP 表达式语言（Expression Language）对当前字段或整行数据进行转换。JSP 表达式语言可以用来创建算术和逻辑表达式。在表达式内可以使用整型数，浮点数，字符串，常量 true、false 和 null。

表达式支持以下两个环境变量：

- value：当前字段值。
- row：当前行，数组类型。

表达式支持以下工具类：

- StringUtils：字符串处理类，参考 Java SDK 代码的包结构“org.apache.commons.lang.StringUtils”。
- DateUtils：日期工具类。
- CommonUtils：公共工具类。
- NumberUtils：字符串转数值类。
- HttpsUtils：读取网络文件类。

应用举例：

1. 如果当前字段为字符串类型，将字符串全部转换为小写，例如将“aBC”转换为“abc”。

   表达式：StringUtils.lowerCase(value)
2. 将当前字段的字符串全部转为大写。

   表达式：StringUtils.upperCase(value)
3. 如果当前字段值为“yyyy-MM-dd”格式的日期字符串，需要截取年，例如字段值为“2017-12-01”，转换后为“2017”。

   表达式：StringUtils.substringBefore(value,"-")
4. 如果当前字段值为数值类型，转换后值为当前值的两倍。

   表达式：value*2
5. 如果当前字段值为“true”，转换后为“Y”，其它值则转换后为“N”。

   表达式：value=="true"?"Y":"N"
6. 如果当前字段值为字符串类型，当为空时，转换为“Default”，否则不转换。

   表达式：empty value? "Default":value
7. 如果想将日期字段格式从“2018/01/05 15:15:05”转换为“2018-01-05 15:15:05”。

   表达式：DateUtils.format(DateUtils.parseDate(value,"yyyy/MM/dd HH:mm:ss"),"yyyy-MM-dd HH:mm:ss")
8. 获取一个 36 位的 UUID（Universally Unique Identifier，通用唯一识别码）。

   表达式：CommonUtils.randomUUID()
9. 如果当前字段值为字符串类型，将首字母转换为大写，例如将“cat”转换为“Cat”。

   表达式：StringUtils.capitalize(value)
10. 如果当前字段值为字符串类型，将首字母转换为小写，例如将“Cat”转换为“cat”。

    表达式：StringUtils.uncapitalize(value)
11. 如果当前字段值为字符串类型，使用空格填充为指定长度，并且将字符串居中，当字符串长度不小于指定长度时不转换，例如将“ab”转换为长度为 4 的“ab”。

    表达式：StringUtils.center(value,*4*)
12. 删除字符串末尾的一个换行符（包括“\n”、“\r”或者“\r\n”），例如将“abc\r\n\r\n”转换为“abc\r\n”。

    表达式：StringUtils.chomp(value)

13. 如果字符串中包含指定的字符串，则返回布尔值 true，否则返回 false。例如“abc”中包含“a”，则返回 true。

    表达式：StringUtils.contains(value,"*a*")

14. 如果字符串中包含指定字符串的任一字符，则返回布尔值 true，否则返回 false。例如“zzabyycdxx”中包含“z”或“a”任意一个，则返回 true。

    表达式：StringUtils.containsAny("value","*za*")

15. 如果字符串中不包含指定的所有字符，则返回布尔值 true，包含任意一个字符则返回 false。例如“abz”中包含“xyz”里的任意一个字符，则返回 false。

    表达式：StringUtils.containsNone(value,"*xyz*")

16. 如果当前字符串只包含指定字符串中的字符，则返回布尔值 true，包含任意一个其它字符则返回 false。例如“abab”只包含“abc”中的字符，则返回 true。

    表达式：StringUtils.containsOnly(value,"*abc*")

17. 如果字符串为空或 null，则转换为指定的字符串，否则不转换。例如将空字符转换为 null。

    表达式：StringUtils.defaultIfEmpty(value,*null*)

18. 如果字符串以指定的后缀结尾（包括大小写），则返回布尔值 true，否则返回 false。例如“abcdef”后缀不为 null，则返回 false。

    表达式：StringUtils.endsWith(value,*null*)

19. 如果字符串和指定的字符串完全一样（包括大小写），则返回布尔值 true，否则返回 false。例如比较字符串“abc”和“ABC”，则返回 false。

    表达式：StringUtils.equals(value,"*ABC*")

20. 从字符串中获取指定字符串的第一个索引，没有则返回整数-1。例如从“aabaabaa”中获取“ab”的第一个索引 1。

    表达式：StringUtils.indexOf(value,"*ab*")

21. 从字符串中获取指定字符串的最后一个索引，没有则返回整数-1。例如从“aFkyk”中获取“k”的最后一个索引 4。

    表达式：StringUtils.lastIndexOf(value,"*k*")

22. 从字符串中指定的位置往后查找，获取指定字符串的第一个索引，没有则转换为“-1”。例如“aabaabaa”中索引 3 的后面，第一个“b”的索引是 5。

    表达式：StringUtils.indexOf(value,"*b*",*3*)

23. 从字符串获取指定字符串中任一字符的第一个索引，没有则返回整数-1。例如从“zzabyycdxx”中获取“z”或“a”的第一个索引 0。

    表达式：StringUtils.indexOfAny(value,"*za*")

24. 如果字符串仅包含 Unicode 字符，返回布尔值 true，否则返回 false。例如“ab2c”中包含非 Unicode 字符，返回 false。

    表达式：StringUtils.isAlpha(value)

25. 如果字符串仅包含 Unicode 字符或数字，返回布尔值 true，否则返回 false。例如“ab2c”中仅包含 Unicode 字符和数字，返回 true。

    表达式：StringUtils.isAlphanumeric(value)

26. 如果字符串仅包含 Unicode 字符、数字或空格，返回布尔值 true，否则返回 false。例如“ab2c”中仅包含 Unicode 字符和数字，返回 true。

表达式：StringUtils.isAlphanumericSpace(value)

27. 如果字符串仅包含 Unicode 字符或空格，返回布尔值 true，否则返回 false。例如“ab2c”中包含 Unicode 字符和数字，返回 false。

    表达式：StringUtils.isAlphaSpace(value)

28. 如果字符串仅包含 ASCII 可打印字符，返回布尔值 true，否则返回 false。例如“!ab-c~”返回 true。

    表达式：StringUtils.isAsciiPrintable(value)

29. 如果字符串为空或 null，返回布尔值 true，否则返回 false。

    表达式：StringUtils.isEmpty(value)

30. 如果字符串中仅包含 Unicode 数字，返回布尔值 true，否则返回 false。

    表达式：StringUtils.isNumeric(value)

31. 获取字符串最左端的指定长度的字符，例如获取“abc”最左端的 2 位字符“ab”。

    表达式：StringUtils.left(value,*2*)

32. 获取字符串最右端的指定长度的字符，例如获取“abc”最右端的 2 位字符“bc”。

    表达式：StringUtils.right(value,*2*)

33. 将指定字符串拼接至当前字符串的左侧，需同时指定拼接后的字符串长度，如果当前字符串长度不小于指定长度，则不转换。例如将“yz”拼接到“bat”左侧，拼接后长度为 8，则转换后为“yzyzybat”。

    表达式：StringUtils.leftPad(value,*8*,"*yz*")

34. 将指定字符串拼接至当前字符串的右侧，需同时指定拼接后的字符串长度，如果当前字符串长度不小于指定长度，则不转换。例如将“yz”拼接到“bat”右侧，拼接后长度为 8，则转换后为“batyzyzy”。

    表达式：StringUtils.rightPad(value,*8*,"*yz*")

35. 如果当前字段为字符串类型，获取当前字符串的长度，如果该字符串为 null，则返回 0。

    表达式：StringUtils.length(value)

36. 如果当前字段为字符串类型，删除其中所有的指定字符串，例如从“queued”中删除“ue”，转换后为“qd”。

    表达式：StringUtils.remove(value,"*ue*")

37. 如果当前字段为字符串类型，移除当前字段末尾指定的子字符串。指定的子字符串若不在当前字段的末尾，则不转换，例如移除当前字段“www.domain.com”后的“.com”。

    表达式：StringUtils.removeEnd(value,"*.com*")

38. 如果当前字段为字符串类型，移除当前字段开头指定的子字符串。指定的子字符串若不在当前字段的开头，则不转换，例如移除当前字段“www.domain.com”前的“www.”。

    表达式：StringUtils.removeStart(value,"*www.*")

39. 如果当前字段为字符串类型，替换当前字段中所有的指定字符串，例如将“aba”中的“a”用“z”替换，转换后为“zbz”。

表达式：StringUtils.replace(value,"*a*","z")

40. 如果当前字段为字符串类型，一次替换字符串中的多个字符，例如将字符串“hello”中的“h”用“j”替换，“o”用“y”替换，转换后为“jelly”。

    表达式：StringUtils.replaceChars(value,"*ho*","*jy*")

41. 如果字符串以指定的前缀开头（区分大小写），则返回布尔值 true，否则返回 false，例如当前字符串“abcdef”以“abc”开头，则返回 true。

    表达式：StringUtils.startsWith(value,"*abc*")

42. 如果当前字段为字符串类型，去除字段中所有指定的字符，例如去除“abcyx”中所有的“x”、“y”和“z”，转换后为“abc”。

    表达式：StringUtils.strip(value,"*xyz*")

43. 如果当前字段为字符串类型，去除字段末尾所有指定的字符，例如去除当前字段末尾的所有空格。

    表达式：StringUtils.stripEnd(value,*null*)

44. 如果当前字段为字符串类型，去除字段开头所有指定的字符，例如去除当前字段开头的所有空格。

    表达式：StringUtils.stripStart(value,*null*)

45. 如果当前字段为字符串类型，获取字符串指定位置后（不包括指定位置的字符）的子字符串，指定位置如果为负数，则从末尾往前计算位置。例如获取“abcde”第 2 个字符后的字符串，则转换后为“cde”。

    表达式：StringUtils.substring(value,*2*)

46. 如果当前字段为字符串类型，获取字符串指定区间的子字符串，区间位置如果为负数，则从末尾往前计算位置。例如获取“abcde”第 2 个字符后、第 5 个字符前的字符串，则转换后为“cd”。

    表达式：StringUtils.substring(value,*2*,*5*)

47. 如果当前字段为字符串类型，获取当前字段里第一个指定字符后的子字符串。例如获取“abcba”中第一个“b”之后的子字符串，转换后为“cba”。

    表达式：StringUtils.substringAfter(value,"*b*")

48. 如果当前字段为字符串类型，获取当前字段里最后一个指定字符后的子字符串。例如获取“abcba”中最后一个“b”之后的子字符串，转换后为“a”。

    表达式：StringUtils.substringAfterLast(value,"*b*")

49. 如果当前字段为字符串类型，获取当前字段里第一个指定字符前的子字符串。例如获取“abcba”中第一个“b”之前的子字符串，转换后为“a”。

    表达式：StringUtils.substringBefore(value,"*b*")

50. 如果当前字段为字符串类型，获取当前字段里最后一个指定字符前的子字符串。例如获取“abcba”中最后一个“b”之前的子字符串，转换后为“abc”。

    表达式：StringUtils.substringBeforeLast(value,"*b*")

51. 如果当前字段为字符串类型，获取嵌套在指定字符串之间的子字符串，没有匹配的则返回 null。例如获取“tagabctag”中“tag”之间的子字符串，转换后为“abc”。

    表达式：StringUtils.substringBetween(value,"*tag*")

52. 如果当前字段为字符串类型，删除当前字符串两端的控制字符（char≤32），例如删除字符串前后的空格。

    表达式：StringUtils.trim(value)

53. 将当前字符串转换为字节，如果转换失败，则返回 0。

    表达式：NumberUtils.toByte(value)

54. 将当前字符串转换为字节，如果转换失败，则返回指定值，例如指定值配置为 1。

    表达式：NumberUtils.toByte(value,*1*)

55. 将当前字符串转换为 Double 数值，如果转换失败，则返回 0.0d。

    表达式：NumberUtils.toDouble(value)

56. 将当前字符串转换为 Double 数值，如果转换失败，则返回指定值，例如指定值配置为 1.1d。

    表达式：NumberUtils.toDouble(value,*1.1d*)

57. 将当前字符串转换为 Float 数值，如果转换失败，则返回 0.0f。

    表达式：NumberUtils.toFloat(value)

58. 将当前字符串转换为 Float 数值，如果转换失败，则返回指定值，例如配置指定值为 1.1f。

    表达式：NumberUtils.toFloat(value,*1.1f*)

59. 将当前字符串转换为 Int 数值，如果转换失败，则返回 0。

    表达式：NumberUtils.toInt(value)

60. 将当前字符串转换为 Int 数值，如果转换失败，则返回指定值，例如配置指定值为 1。

    表达式：NumberUtils.toInt(value,*1*)

61. 将字符串转换为 Long 数值，如果转换失败，则返回 0。

    表达式：NumberUtils.toLong(value)

62. 将当前字符串转换为 Long 数值，如果转换失败，则返回指定值，例如配置指定值为 1L。

    表达式：NumberUtils.toLong(value,*1L*)

63. 将字符串转换为 Short 数值，如果转换失败，则返回 0。

    表达式：NumberUtils.toShort(value)

64. 将当前字符串转换为 Short 数值，如果转换失败，则返回指定值，例如配置指定值为 1。

    表达式：NumberUtils.toShort(value,*1*)

65. 将当前 IP 字符串转换为 Long 数值，例如将“10.78.124.0”转换为 LONG 数值是“172915712”。

    表达式：CommonUtils.ipToLong(value)

66. 从网络读取一个 IP 与物理地址映射文件，并存放到 Map 集合，这里的 URL 是 IP 与地址映射文件存放地址，例如“http://10.114.205.45:21203/sqoop/IpList.csv”。

    表达式：HttpsUtils.downloadMap("*url*")

67. 将 IP 与地址映射对象缓存起来并指定一个 key 值用于检索，例如“ipList”。

表达式：CommonUtils.setCache("*ipList*",HttpsUtils.downloadMap("*url*"))

68. 取出缓存的 IP 与地址映射对象。

    表达式：CommonUtils.getCache("*ipList*")

69. 判断是否有 IP 与地址映射缓存。

    表达式：CommonUtils.cacheExists("*ipList*")

70. 根据指定的偏移类型（month/day/hour/minute/second）及偏移量（正数表示增加，负数表示减少），将指定格式的时间转换为一个新时间，例如将“2019-05-21 12:00:00”增加 8 个小时。

    表达式：DateUtils.getCurrentTimeByZone("*yyyy-MM-dd HH:mm:ss*",value, "*hour*", *8*)

## 3.2.3 Hadoop 类型的数据源进行数据迁移时，建议使用的组件版本有哪些？

建议使用的组件版本既可以作为目的端使用，也可以作为源端使用。

表3-3 建议使用的组件版本

| Hadoop 类型 | 组件 | 说明 |
| --- | --- | --- |
| MRS/Apache/FusionInsight HD | Hive | 暂不支持 2.x 版本，建议使用的版本：<br>• 1.2.X<br>• 3.1.X |
| | HDFS | 建议使用的版本：<br>• 2.8.X<br>• 3.1.X |
| | Hbase | 建议使用的版本：<br>• 2.1.X<br>• 1.3.X |

## 3.2.4 数据源为 Hive 时支持哪些数据格式？

云数据迁移服务支持从 Hive 数据源读写的数据格式包括 SequenceFile、TextFile、ORC、Parquet。

## 3.2.5 是否支持同步作业到其他集群？

CDM 虽然不支持直接在不同集群间迁移作业，但是通过批量导出、批量导入作业的功能，可以间接实现集群间的作业迁移，方法如下：

1. 将 CDM 集群 1 中的所有作业批量导出，将作业的 JSON 文件保存到本地。

   由于安全原因，CDM 导出作业时没有导出连接密码，连接密码全部使用“Add password here”替换。

2. 在本地编辑 JSON 文件，将“Add password here”替换为对应连接的正确密码。
3. 将编辑好的 JSON 文件批量导入到 CDM 集群 2，实现集群 1 和集群 2 之间的作业同步。

## 3.2.6 是否支持批量创建作业？

CDM 可以通过批量导入的功能，实现批量创建作业，方法如下：

1. 手动创建一个作业。
2. 导出作业，将作业的 JSON 文件保存到本地。
3. 编辑 JSON 文件，参考该作业的配置，在 JSON 文件中批量复制出更多作业。
4. 将 JSON 文件导入 CDM 集群，实现批量创建作业。

## 3.2.7 是否支持批量调度作业？

支持。

1. 访问 DataArts Studio 服务的数据开发模块。
2. 在数据开发主界面的左侧导航栏，选择“数据开发 > 作业开发”，新建作业。
3. 拖动多个 CDM Job 节点至画布，然后再编排作业。

## 3.2.8 如何备份 CDM 作业？

可以，如果用户长时间不需要使用 CDM 集群，可以将 CDM 集群停掉或删除来降低成本。

删除前，用户可以先通过 CDM 的批量导出功能，把所有作业脚本保存到本地，仅在需要的时候再重新创建集群、重新导入作业，实现作业备份。

## 3.2.9 如果 HANA 集群只有部分节点和 CDM 集群网络互通，应该如何配置连接？

如果 HANA 集群只有部分节点和 CDM 网络互通，为确保 CDM 正常连接 HANA 集群，则需要进行如下配置：

1. 关闭 HANA 集群的 Statement Routing 开关。但须注意，关闭 Statement Routing，会增加配置节点的压力。
2. 新建 HANA 连接时，在高级属性中添加属性“distribution”，并将值置为“off”。

完成配置后，CDM 即可正常连接 HANA 集群。

## 3.2.10 如何使用 Java 调用 CDM 的 Rest API 创建数据迁移作业？

CDM 提供了 Rest API，可以通过程序调用实现自动化的作业创建或执行控制。

这里以 CDM 迁移 MySQL 数据库的表 city1 的数据到 DWS 的表 city2 为例，介绍如何使用 Java 调用 CDM 服务的 REST API 创建、启动、查询、删除该 CDM 作业。

需要提前准备以下数据：

1. 云帐号的用户名、帐号名和项目 ID。
2. 创建一个 CDM 集群，并获取集群 ID。

   获取方法：在集群管理界面，单击 CDM 集群名称可查看集群 ID，例如“c110beff-0f11-4e75-8b10-da7cd882b0ef”。
3. 创建一个 MySQL 数据库和一个 DWS 数据库，并创建好表 city1 和表 city2，创表语句如下：

```
MySQL:
create table city1(code varchar(10),name varchar(32));
insert into city1 values('NY','New York');
DWS:
create table city2(code varchar(10),name varchar(32));
```

4. 在 CDM 集群下，创建连接到 MySQL 的连接，例如连接名称为“mysqltestlink”。创建连接到 DWS 的连接，例如连接名称为“dwstestlink”。
5. 运行下述代码，依赖 HttpClient 包，建议使用 4.5 版本。Maven 配置如下：

```
<project>
<modelVersion>4.0.0</modelVersion>
<groupId>cdm</groupId>
<artifactId>cdm-client</artifactId>
<version>1</version>
<dependencies>
<dependency>
<groupId>org.apache.httpcomponents</groupId>
<artifactId>httpclient</artifactId>
<version>4.5</version>
</dependency>
</dependencies>
</project>
```

## 代码示例

使用 Java 调用 CDM 服务的 REST API 创建、启动、查询、删除 CDM 作业的代码示例如下：

```
package cdmclient;
import java.io.IOException;
import org.apache.http.Header;
import org.apache.http.HttpEntity;
import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.client.config.RequestConfig;
import org.apache.http.client.methods.CloseableHttpResponse;
import org.apache.http.client.methods.HttpDelete;
```

```
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.client.methods.HttpPut;
import org.apache.http.entity.StringEntity;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.util.EntityUtils;
public class CdmClient {
private final static String DOMAIN_NAME="云帐号名";
private final static String USER_NAME="云用户名";
private final static String USER_PASSWORD="云用户密码";
private final static String PROJECT_ID="项目ID";
private final static String CLUSTER_ID="CDM集群ID";
private final static String JOB_NAME="作业名称";
private final static String FROM_LINKNAME="源连接名称";
private final static String TO_LINKNAME="目的连接名称";
private final static String IAM_ENDPOINT="IAM的Endpoint";
private final static String CDM_ENDPOINT="CDM的Endpoint";
private CloseableHttpClient httpclient;
private String token;

public CdmClient() {
this.httpclient = createHttpClient();
this.token = login();
}

private CloseableHttpClient createHttpClient() {
CloseableHttpClient httpclient =HttpClients.createDefault();
return httpclient;
}

private String login(){
HttpPost httpPost = new
HttpPost("https://"+IAM_ENDPOINT+"/v3/auth/tokens");
String json =
"{\r\n"+
"\"auth\": {\r\n"+
"\"identity\": {\r\n"+
"\"methods\": [\"password\"],\r\n"+
"\"password\": {\r\n"+
"\"user\": {\r\n"+
```

```
"\"name\": \""+USER_NAME+"\",\r\n"+
"\"password\": \""+USER_PASSWORD+"\",\r\n"+
"\"domain\": {\r\n"+
"\"name\": \""+DOMAIN_NAME+"\"\r\n"+
"}\r\n"+
"}\r\n"+
"}\r\n"+
"},\r\n"+
"\"scope\": {\r\n"+
"\"project\": {\r\n"+
"\"name\": \"PROJECT_NAME\"\r\n"+
"}\r\n"+
"}\r\n"+
"}\r\n"+
"}\r\n";
try {
StringEntity s = new StringEntity(json);
s.setContentEncoding("UTF-8");
s.setContentType("application/json");
httpPost.setEntity(s);
CloseableHttpResponse response = httpclient.execute(httpPost);
Header tokenHeader = response.getFirstHeader("X-Subject-Token");
String token = tokenHeader.getValue();
System.out.println("Login successful");
return token;
} catch (Exception e) {
throw new RuntimeException("login failed.", e);
}
}
/*创建作业*/

public void createJob(){
HttpPost httpPost = new
HttpPost("https://"+CDM_ENDPOINT+"/cdm/v1.0/"+PROJECT_ID+"/clusters/"+CLU
STER_ID+"/cdm/job");

/**此处JSON信息比较复杂，可以先在作业管理界面上创建一个作业，然后单击作业后的“作业
JSON定义”，复制其中的JSON内容，格式化为Java字符串语法，然后粘贴到此处。
*JSON消息体中一般只需要替换连接名、导入和导出的表名、导入导出表的字段列表、源表中用于分
区的字段。**/

String json =
"{\r\n"+
```

```
"\"jobs\": [\r\n"+
"{\r\n"+
"\"from-connector-name\": \"generic-jdbc-connector\",\r\n"+
"\"name\": \""+JOB_NAME+"\",\r\n"+
"\"to-connector-name\": \"generic-jdbc-connector\",\r\n"+
"\"driver-config-values\": {\r\n"+
"\"configs\": [\r\n"+
"{\r\n"+
"\"inputs\": [\r\n"+
"{\r\n"+
"\"name\": \"throttlingConfig.numExtractors\",\r\n"+
"\"value\": \"1\"\r\n"+
"}\r\n"+
"],\r\n"+
"\"validators\": [],\r\n"+
"\"type\": \"JOB\",\r\n"+
"\"id\": 30,\r\n"+
"\"name\": \"throttlingConfig\"\r\n"+
"}\r\n"+
"]\r\n"+
"},\r\n"+
"\"from-link-name\": \""+FROM_LINKNAME+"\",\r\n"+
"\"from-config-values\": {\r\n"+
"\"configs\": [\r\n"+
"{\r\n"+
"\"inputs\": [\r\n"+
"{\r\n"+
"\"name\": \"fromJobConfig.schemaName\",\r\n"+
"\"value\": \"sqoop\"\r\n"+
"},\r\n"+
"{\r\n"+
"\"name\": \"fromJobConfig.tableName\",\r\n"+
"\"value\": \"city1\"\r\n"+
"},\r\n"+
"{\r\n"+
"\"name\": \"fromJobConfig.columnList\",\r\n"+
"\"value\": \"code&name\"\r\n"+
"},\r\n"+
"{\r\n"+
"\"name\": \"fromJobConfig.partitionColumn\",\r\n"+
"\"value\": \"code\"\r\n"+
"}\r\n"+
"],\r\n"+
"\"validators\": [],\r\n"+
```

```
"\"type\": \"JOB\",\r\n"+
"\"id\": 7,\r\n"+
"\"name\": \"fromJobConfig\"\r\n"+
"}\r\n"+
"]\r\n"+
"},\r\n"+
"\"to-link-name\": \""+TO_LINKNAME+"\",\r\n"+
"\"to-config-values\": {\r\n"+
"\"configs\": [\r\n"+
"{\r\n"+
"\"inputs\": [\r\n"+
"{\r\n"+
"\"name\": \"toJobConfig.schemaName\",\r\n"+
"\"value\": \"sqoop\"\r\n"+
"},\r\n"+
"{\r\n"+
"\"name\": \"toJobConfig.tableName\",\r\n"+
"\"value\": \"city2\"\r\n"+
"},\r\n"+
"{\r\n"+
"\"name\": \"toJobConfig.columnList\",\r\n"+
"\"value\": \"code&name\"\r\n"+
"}, \r\n"+
"{\r\n"+
"\"name\": \"toJobConfig.shouldClearTable\",\r\n"+
"\"value\": \"true\"\r\n"+
"}\r\n"+
"],\r\n"+
"\"validators\": [],\r\n"+
"\"type\": \"JOB\",\r\n"+
"\"id\": 9,\r\n"+
"\"name\": \"toJobConfig\"\r\n"+
"}\r\n"+
"]\r\n"+
"}\r\n"+
"}\r\n"+
"]\r\n"+
"}\r\n";
try {
StringEntity s = new StringEntity(json);
s.setContentEncoding("UTF-8");
s.setContentType("application/json");
httpPost.setEntity(s);
httpPost.addHeader("X-Auth-Token", this.token);
```

```
httpPost.addHeader("X-Language", "en-us");
CloseableHttpResponse response = httpclient.execute(httpPost);
int status = response.getStatusLine().getStatusCode();
if(status == 200){
System.out.println("Create job successful.");
}else{
System.out.println("Create job failed.");
HttpEntity entity = response.getEntity();
System.out.println(EntityUtils.toString(entity));
}
} catch (Exception e) {
e.printStackTrace();
throw new RuntimeException("Create job failed.", e);
}
}
/*启动作业*/

public void startJob(){
HttpPut httpPut = new
HttpPut("https://"+CDM_ENDPOINT+"/cdm/v1.0/"+PROJECT_ID+"/clusters/"+CLUS
TER_ID+"/cdm/job/"+JOB_NAME+"/start");
String json = "";
try {
StringEntity s = new StringEntity(json);
s.setContentEncoding("UTF-8");
s.setContentType("application/json");
httpPut.setEntity(s);
httpPut.addHeader("X-Auth-Token", this.token);
httpPut.addHeader("X-Language", "en-us");
CloseableHttpResponse response = httpclient.execute(httpPut);
int status = response.getStatusLine().getStatusCode();
if(status == 200){
System.out.println("Start job successful.");
}else{
System.out.println("Start job failed.");
HttpEntity entity = response.getEntity();
System.out.println(EntityUtils.toString(entity));
}
} catch (Exception e) {
e.printStackTrace();
throw new RuntimeException("Start job failed.", e);
}
}
```

```
/*循环查询作业运行状态，直到作业运行结束。*/

public void getJobStatus(){
HttpGet httpGet = new
HttpGet("https://"+CDM_ENDPOINT+"/cdm/v1.0/"+PROJECT_ID+"/clusters/"+CLUS
TER_ID+"/cdm/job/"+JOB_NAME+"/status");
try {
httpGet.addHeader("X-Auth-Token", this.token);
httpGet.addHeader("X-Language", "en-us");
boolean flag = true;
while(flag){
CloseableHttpResponse response = httpclient.execute(httpGet);
int status = response.getStatusLine().getStatusCode();
if(status == 200){
HttpEntity entity = response.getEntity();
String msg = EntityUtils.toString(entity);
if(msg.contains("\"status\":\"SUCCEEDED\"")){
System.out.println("Job succeeded");
break;
}else if (msg.contains("\"status\":\"FAILED\"")){
System.out.println("Job failed.");
break;
}else{
Thread.sleep(1000);
}

}else{
System.out.println("Get job status failed.");
HttpEntity entity = response.getEntity();
System.out.println(EntityUtils.toString(entity));
break;
}
}
} catch (Exception e) {
e.printStackTrace();
throw new RuntimeException("Get job status failed.", e);
}
}

/*删除作业*/

public void deleteJob(){
HttpDelete httpDelte = new
```

```
HttpDelete("https://"+CDM_ENDPOINT+"/cdm/v1.0/"+PROJECT_ID+"/clusters/"+C
LUSTER_ID+"/cdm/job/"+JOB_NAME);
try {
httpDelte.addHeader("X-Auth-Token", this.token);
httpDelte.addHeader("X-Language", "en-us");
CloseableHttpResponse response = httpclient.execute(httpDelte);
int status = response.getStatusLine().getStatusCode();
if(status == 200){
System.out.println("Delete job successful.");
}else{
System.out.println("Delete job failed.");
HttpEntity entity = response.getEntity();
System.out.println(EntityUtils.toString(entity));
}
} catch (Exception e) {
e.printStackTrace();
throw new RuntimeException("Delete job failed.", e);
}
}
/*关闭*/

public void close(){
try {
httpclient.close();
} catch (IOException e) {
throw new RuntimeException("Close failed.", e);
}
}

public static void main(String[] args){
CdmClient cdmClient = new CdmClient();
cdmClient.createJob();
cdmClient.startJob();
cdmClient.getJobStatus();
cdmClient.deleteJob();
cdmClient.close();
}
}
```

## 3.2.11 如何将云下内网或第三方云上的私网与 CDM 连通？

很多企业会把关键数据源建设在内网，例如数据库、文件服务器等。由于 CDM 运行在云上，如果要通过 CDM 迁移内网数据到云上的话，可以通过以下几种方式连通内网和 CDM 的网络：

- 如果目标数据源为云下的数据库，则需要通过公网或者专线打通网络。通过公网互通时，需确保 CDM 集群已绑定 EIP、CDM 云上安全组出方向放通云下数据源所在的主机、数据源所在的主机可以访问公网且防火墙规则已开放连接端口。
- 在本地数据中心和云服务 VPC 之间建立 VPN 通道。
- 通过 NAT（网络地址转换，Network Address Translation）或端口转发，以代理的方式访问。

这里重点介绍如何通过端口转发工具来实现访问内部数据，流程如下：

1. 找一台 windows 机器作为网关，该机器必须可以直接访问 Internet，同时可以访问内网。
2. 在该机器上安装端口映射工具（IPOP）。
3. 通过端口映射工具（IPOP）配置端口映射。

**须知**

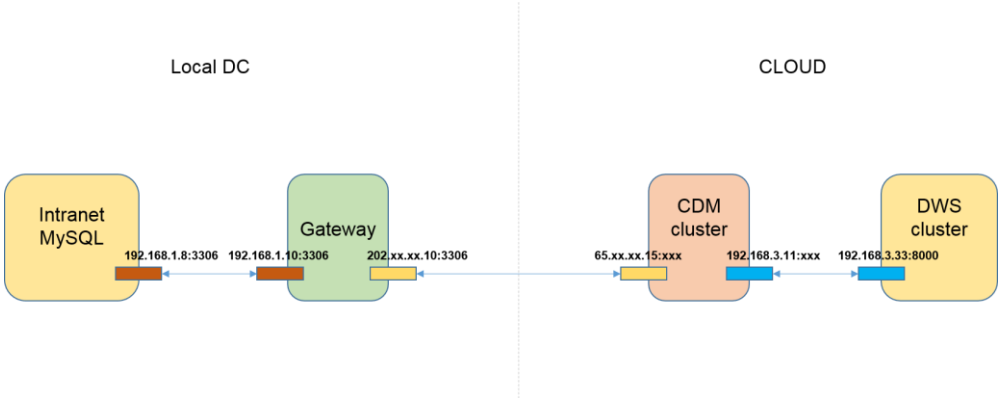
**长时间将内网数据库暴露在公网会有安全风险，迁移数据完成后，请及时停止端口映射。**

## 场景描述

这里假设是将内网 MySQL 迁移到云服务 DWS

图中的内网既可以是企业自己的数据中心，也可以是在第三方云的虚拟数据中心私网。

图3-3 网络拓扑样例




## 操作步骤

步骤 1 找一台 Windows 机器作为网关机，该机器同时配置内网和外网 IP。通过以下测试来确保网关机器的服务要求：

1. 在该机器上 **ping** 内网 MySQL 地址可以 **ping** 通，例如：**ping 192.168.1.8**。

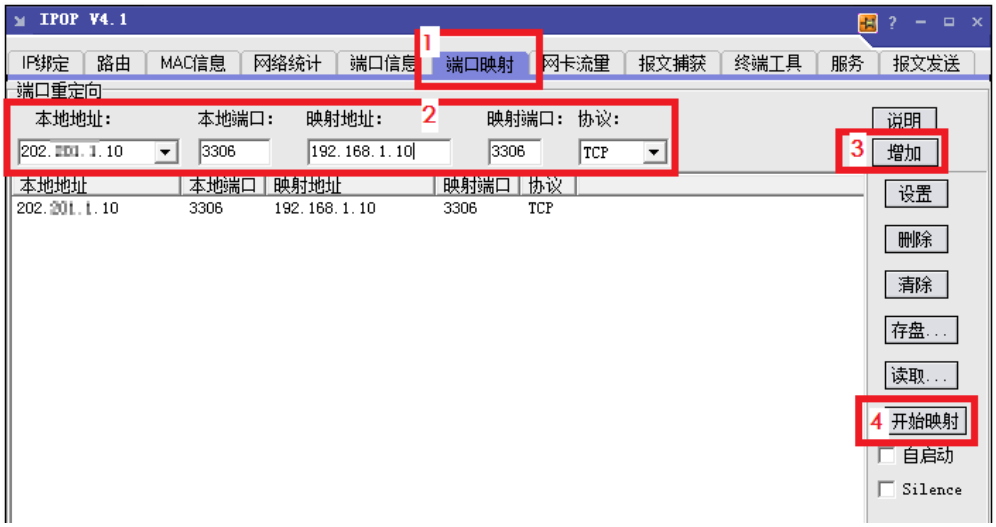
2. 在另外一台可上网的机器上 **ping** 网关机的公网地址可以 **ping** 通，例如 **ping 202.*xx*.*xx*.10**。

步骤 2 下载端口映射工具 IPOP，在网关机上安装 IPOP。

步骤 3 运行端口映射工具，选择“端口映射”，如图 3-4 所示。

- 本地地址、本地端口：配置为网关机的公网地址和端口（后续在 CDM 上创建 MySQL 连接时输入这个地址和端口）。
- 映射地址、映射端口：配置为内网 MySQL 的地址和端口。

图3-4 配置端口映射




步骤 4 单击“增加”，添加端口映射关系。

步骤 5 单击“开始映射”，这时才会真正开始映射，接收数据包。

至此，就可以在 CDM 上通过弹性 IP 读取本地内网 MySQL 的数据，然后导入到云服务 DWS 中。

**说明**

1. CDM 要访问本地数据源，也必须给 CDM 集群配置 EIP。
2. 一般云服务 DWS 默认也是只允许 VPC 内部访问，创建 CDM 集群时，必须将 CDM 的 VPC 与 DWS 配置一致，且推荐在同一个内网和安全组，如果不同，还需要配置允许两个安全组之间的数据访问。
3. 端口映射不仅可以用于迁移内网数据库的数据，还可以迁移例如 SFTP 服务器上的数据。
4. Linux 机器也可以通过 IPTABLE 实现端口映射。
5. 内网中的 FTP 通过端口映射到公网时，需要检查是否启用了 PASV 模式。这种情况下客户端和服务端建立连接的时候是走的随机端口，所以除了配置 21 端口映射外，还需要配置 PASV 模式的端口范围映射，例如 vsftp 通过配置 pasv_min_port 和 pasv_max_port 指定端口范围。

----结束

## 3.2.12 CDM 迁移作业的抽取并发数应该如何设置？

CDM 迁移作业的抽取并发数，与集群规格和表大小有关。并发抽取数取值范围为 1-300，若配置过大，则以队列的形式进行排队。

建议每 1CUs（1CUs=1 核 4G）配置为 4，如表 3-4 所示，您也可以根据实际情况进行调整。另外，每行数据大小为 1MB 以下的可以多并发抽取，超过 1MB 的建议单线程抽取数据。

**说明**

- 迁移的目的端为文件时，CDM 不支持多并发，此时应配置为单进程抽取数据。
- 单作业的抽取并发数，受到作业“配置管理”中所配置的“最大抽取并发数”影响。“最大抽取并发数”配置的是抽取并发总数。

表3-4 抽取并发数参考配置

| CDM 集群规格 | vCPUs/内存 | 抽取并发数参考配置 |
| --- | --- | --- |
| cdm.large | 8 核 16GB | 16 |
| cdm.xlarge | 16 核 32GB | 32 |
| cdm.4xlarge | 64 核 128GB | 128 |

## 3.2.13 CDM 是否支持动态数据实时迁移功能？

不支持。如果源端在迁移过程中写数据，可能会出现报错。

# 3.3 故障处理类

## 3.3.1 OBS 导入数据到 SQL Server 时出现 Unable to execute the SQL statement 怎么处理？

### 问题描述

使用 CDM 从 OBS 导入数据到 SQL Server 时，作业运行失败，错误提示为：Unable to execute the SQL statement. Cause : 将截断字符串或二进制数据。

### 原因分析

用户 OBS 中的数据超出了 SQL Server 数据库的字段长度限制。

### 解决方法

在 SQL Server 数据库中建表时，将数据库字段改大，长度不能小于源端 OBS 中的数据长度。

## 3.3.2 Oracle 迁移到 DWS 报错 ORA-01555

### 问题现象

使用 CDM 迁移 Oracle 数据至 DWS，报错图 3-5 所示。

图3-5 报错现象

```
2020-09-21 22:51:02,991 ERROR LocalJobRunner Map Task #3 [org.apache.sqoop.common.SqoopException:111] SqoopException
java.sql.SQLException: ORA-01555: snapshot too old: rollback segment number 3 with name "_SYSSMU3_20976775318$" too small

    at oracle.jdbc.driver.T4CTTIoer11.processERROR(T4CTTIoer11.java:494)
    at oracle.jdbc.driver.T4CTTIoer11.processERROR(T4CTTIoer11.java:446)
    at oracle.jdbc.driver.T4C8Oall.processERROR(T4C8Oall.java:1054)
    at oracle.jdbc.driver.T4CTTIfun.receive(T4CTTIfun.java:623)
    at oracle.jdbc.driver.T4CTTIfun.doRPC(T4CTTIfun.java:252)
    at oracle.jdbc.driver.T4C8Oall.doOALL(T4C8Oall.java:612)
    at oracle.jdbc.driver.T4CPreparedStatement.doOall8(T4CPreparedStatement.java:226)
    at oracle.jdbc.driver.T4CPreparedStatement.fetch(T4CPreparedStatement.java:1023)
    at oracle.jdbc.driver.OracleStatement.fetchMoreRows(OracleStatement.java:3353)
    at oracle.jdbc.driver.InsensitiveScrollableResultSet.fetchMoreRows(InsensitiveScrollableResultSet.java:736)
    at oracle.jdbc.driver.InsensitiveScrollableResultSet.absoluteInternal(InsensitiveScrollableResultSet.java:692)
    at oracle.jdbc.driver.InsensitiveScrollableResultSet.next(InsensitiveScrollableResultSet.java:406)
    at org.apache.sqoop.connector.jdbc.sql.impl.WrapResultSet.next(WrapResultSet.java:36)
    at org.apache.sqoop.connector.jdbc.GenericJdbcExtractor.extractObjectRecord(GenericJdbcExtractor.java:151)
    at org.apache.sqoop.connector.jdbc.GenericJdbcExtractor.extract(GenericJdbcExtractor.java:129)
    at org.apache.sqoop.connector.jdbc.GenericJdbcExtractor.extract(GenericJdbcExtractor.java:59)
    at org.apache.sqoop.job.mr.SqoopMapper.runInternal(SqoopMapper.java:184)
    at org.apache.sqoop.job.mr.SqoopMapper.run(SqoopMapper.java:81)
    at org.apache.hadoop.mapred.MapTask.runNewMapper(MapTask.java:799)
    at org.apache.hadoop.mapred.MapTask.run(MapTask.java)
    at org.apache.hadoop.mapred.LocalJobRunner$Job$MapTaskRunnable.run(LocalJobRunner.java:271)
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
    at java.util.concurrent.FutureTask.run(FutureTask.java:266)
    at org.apache.sqoop.submission.mapreduce.MapperExecutorGroup$1.lambda$execute$0(MapperExecutorGroup.java:222)
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
    at java.util.concurrent.FutureTask.run(FutureTask.java:266)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at java.lang.Thread.run(Thread.java:748)
Caused by: oracle.jdbc.OracleDatabaseException: ORA-01555: snapshot too old: rollback segment number 3 with name "_SYSSMU3_20976775318$" too small

    at oracle.jdbc.driver.T4CTTIoer11.processERROR(T4CTTIoer11.java:498)
    ... 28 common frames omitted
```

### 原因分析

1. 数据迁移，整表查询且该表数据量大，那么查询时间较长。
2. 查询过程中，其他用户频繁进行 commit 操作。
3. Oracel 的 RBS(rollbackspace 回滚时使用的表空间 )较小，造成迁移任务没有完成，源库已更新，回滚超时。

### 建议与总结

1. 调小每次查询的数据量。
2. 通过修改数据库配置调大 Oracle 的 RBS。

## 3.3.3 MongoDB 连接迁移失败时如何处理？

在默认情况下，userAdmin 角色只具备对角色和用户的管理，不具备对库的读和写权限。

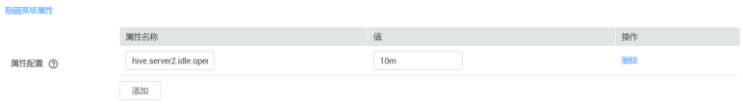
当用户选择 MongoDB 连接迁移失败时，用户需查看 MongoDB 连接中用户的权限信息，确保对指定库具备 ReadWrite 权限。

## 3.3.4 Hive 迁移作业长时间卡住怎么办？

为避免 Hive 迁移作业长时间卡住，可手动停止迁移作业后，通过编辑 Hive 连接增加如下属性设置：

- 属性名称：hive.server2.idle.operation.timeout
- 值：10m

如图所示：




## 3.3.5 使用 CDM 迁移数据由于字段类型映射不匹配导致报错怎么处理？

### 问题描述

在使用 CDM 迁移数据到数据仓库服务（DWS）时，迁移作业失败，且执行日志中出现“value too long for type character varying”错误提示。

### 原因分析

这种情况一般是源表与目标表类型不匹配导致，例如源端 dli 字段为 string 类型，目标端 dws 字段为 varchar(50)类型，导致精度缺省，就会报：value too long for type character varying。类似的问题还有 string 转 bigint，bigint 转 int。

### 解决方案

- 根据报错信息找到哪个字段映射有问题，找 DBA 修改表结构。
- 如果只有极少数据有问题，可以配置脏数据策略解决。

## 3.3.6 MySQL 迁移时报错“JDBC 连接超时”怎么办？

### 问题描述

MySQL 迁移时报错：Unable to connect to the database server. Cause: connect timed out.

### 原因分析

这种情况是由于表数据量较大，并且源端通过 where 语句过滤，但并非索引列，或列值不离散，查询会全表扫描，导致 JDBC 连接超时。例如图 3-6 所示 c_date 字段为非索引列。

图3-6 非索引列


## 解决方案

1. 优先联系 DBA 修改表结构，将需要过滤的列配置为索引列，然后重试。
   如果由于数据不离散，导致还是失败请参考 2~4，通过增大 JDBC 超时时间解决。
2. 根据作业找到对应的 MySQL 连接名称，查找连接信息。

   图3-7 连接信息

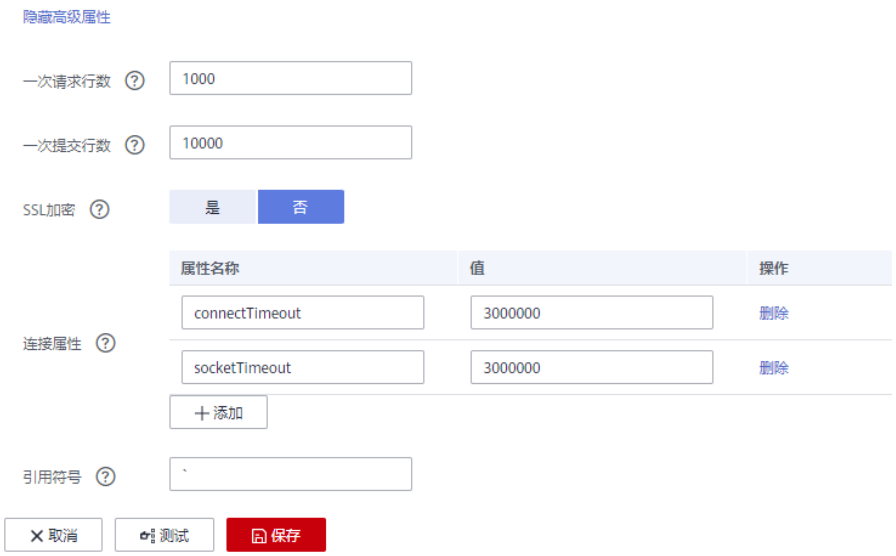
3. 单击“连接管理”，在“操作”列中，单击“连接”进行编辑。

   图3-8 连接

4. 打开高级属性，在“连接属性”中建议新增“connectTimeout”与“socketTimeout”参数及参数值，单击“保存”。

图3-9 编辑高级属性




## 3.3.7 创建了 Hive 到 DWS 类型的连接，进行 CDM 传输任务失败时如何处理？

建议清空历史数据后再次尝试该任务。在使用 CDM 迁移作业的时候需要配置清空历史数据，然后再做迁移，可大大降低任务失败的概率。

## 3.3.8 如何使用 CDM 服务将 MySQL 的数据导出成 SQL 文件，然后上传到 OBS 桶?

CDM 服务暂不支持该操作，建议通过手动导出 MySQL 的数据文件，然后在服务器上开启 SFTP 服务，然后新建 CDM 作业，源端是 SFTP 协议，目的端是 OBS，将文件传过去。

## 3.3.9 如何处理 CDM 从 OBS 迁移数据到 DLI 出现迁移中断失败的问题？

此类作业问题表现为配置了脏数据写入，但并无脏数据。这种情况下需要调低并发任务数，即可避免此类问题。

## 3.3.10 如何处理 CDM 连接器报错“配置项 [linkConfig.iamAuth] 不存在”？

客户证书过期，需要完成更新证书操作，完成后重新配置连接器即可。

## 3.3.11 创建作业时报错“配置项[throttlingConfig.concurrentSubJobs]不存在”怎么办?

当同时存在多个不同版本的集群，先在低版本 CDM 集群创建数据连接或保存作业时后，再进入高版本 CDM 集群时，会偶现此类故障。

需手动清理浏览器缓存，即可避免此类问题。

## 3.3.12 新建 MRS Hive 连接时，提示：CORE_0031:Connect time out.(Cdm.0523) 怎么解决？

新建 MRS Hive 连接时，提示无法下载配置文件，实际是用户权限不足。建议您新建一个业务用户，给对应的权限后重试即可。

如果要创建 MRS 安全集群的数据连接，不能使用 admin 用户。因为 admin 用户是默认的管理页面用户，这个用户无法作为安全集群的认证用户来使用。您可以创建一个新的 MRS 用户，然后在创建 MRS 数据连接时，“用户名”和“密码”填写为新建的 MRS 用户及其密码。

**说明**

- 如果 CDM 集群为 2.9.0 版本及之后版本，且 MRS 集群为 3.1.0 及之后版本，则所创建的用户至少需具备 Manager_viewer 的角色权限才能在 CDM 创建连接；如果需要对 MRS 组件的库、表、列进行操作，还需要参考 MRS 文档添加对应组件的库、表、列操作权限。
- 如果 CDM 集群为 2.9.0 之前的版本，或 MRS 集群为 3.1.0 之前的版本，则所创建的用户需要具备 Manager_administrator 或 System_administrator 权限，才能在 CDM 创建连接。
- 仅具备 Manager_tenant 或 Manager_auditor 权限，无法创建连接。

## 3.3.13 迁移时已选择表不存在时自动创表，提示“CDM not support auto create empty table with no column”怎么处理？

这是由于数据库表名中含有特殊字符导致识别出语法错误，按数据库对象命名规则重新命名后恢复正常。

例如，DWS 数据仓库中的数据表命名需要满足以下约束：长度不超过 63 个字符，以字母或下划线开头，中间字符可以是字母、数字、下划线、$、#。

## 3.3.14 创建 Oracle 关系型数据库迁移作业时，无法获取模式名怎么处理？

这是由于可能上传了暂不支持的最新 ORACLE_8 驱动（如 Oracle Database 21c (21.3) drivers），推荐使用 Oracle Database 12c 中的 ojdbc8.jar 驱动（下载地址：https://www.oracle.com/database/technologies/jdbc-ucp-122-downloads.html）。

# 4 数据架构

## 4.1 码表和数据标准有什么关系？

码表由多条表字段的名称+编码+数据类型组成，码表的表字段可以关联到数据标准上，数据标准会应用到某张模型表的字段上。

## 4.2 关系建模和维度建模的区别？

- 关系建模为事务性模型，对应三范式建模。
- 维度建模为分析性模型，主要包括事实表、维度表的设计，多用于实现多角度、多层次的数据查询和分析。

DataArts Studio 是基于数据湖的数据运营平台，维度建模使用的场景比较多。

## 4.3 数据架构支持哪些数据建模方法？

DataArts Studio 数据架构支持的建模方法有以下两种：

- **关系建模**

  关系建模是用实体关系（Entity Relationship，ER）模型描述企业业务，它在范式理论上符合 3NF，出发点是整合数据，将各个系统中的数据以整个企业角度按主题进行相似性组合和合并，并进行一致性处理，为数据分析决策服务，但是并不能直接用于分析决策。

  用户在关系建模过程中，可以从以下三个层次去设计关系模型，这三个层次是逐层递进的，先设计概念模型，再进一步细化设计出逻辑模型，最后设计物理模型。

  - **概念模型**：是从用户的视角，主要从业务流程、活动中涉及的主要业务数据出发，抽象出关键的业务实体，并描述这些实体间的关系。
  - **逻辑模型**：是概念模型的进一步细化，通过实体、属性和关系勾勒出企业的业务信息蓝图，是 IT 和业务人员沟通的桥梁。逻辑数据模型是一组规范化的逻辑表结构，逻辑数据模型是根据业务规则确定的，关于业务对象、业务对象的数据项及业务对象之间关系的基本蓝图。

- **物理模型**：是在逻辑数据模型的基础上，考虑各种具体的技术实现因素，进行数据库体系结构设计，真正实现数据在数据库中的存放，例如：所选的数据仓库是 DWS。

- **维度建模**

  维度建模是从分析决策的需求出发构建模型，它主要是为分析需求服务，因此它重点关注用户如何更快速地完成需求分析，同时具有较好的大规模复杂查询的响应性能。

  多维模型是由数字型度量值组成的一张事实表连接到一组包含描述属性的多张维度表，事实表与维度表通过主/外键实现关联。

  典型的维度模型有星形模型，以及在一些特殊场景下使用的雪花模型。

  在 DataArts Studio 数据架构中，维度建模是以维度建模理论为基础，构建总线矩阵、抽象出事实和维度，构建维度模型和事实模型，同时对报表需求进行抽象整理出相关指标体系，构建出汇总模型。

## 4.4 规范化的数据如何使用？

规范化的数据可以作为 BI 的基本信息，也可以作为上层应用的源数据，也可以接入各类数据可视化报表等。

## 4.5 数据架构支持逆向数据库吗？

数据架构支持逆向数据库，目前支持基于数据仓库服务（DWS）、MapReduce 服务（MRS Hive）的数据库逆向。

## 4.6 数据架构中的指标与数据质量的指标的区别？

数据架构中指标侧重业务维度，用来衡量目标总体特征的统计数值；数据质量中指标侧重监控维度，用来管理所有业务指标，包括指标的来源、定义等。

注意，数据质量模块的指标与数据架构模块的业务指标、技术指标当前是相互独立的，不支持交互。

## 4.7 为什么数据架构更新表后无变化？

用户在数据架构中更新了表，但实际上表数据并无变化，这是因为在更新前未对数据表更新方式做配置。配置数据表更新方式操作如下：

1. 单击“数据架构 > 配置中心”。
2. 单击“功能配置”页签。
3. “数据表更新方式”选择“重建数据表”。
4. 单击“确定”，完成配置。

# 4.8 表是否可配置生命周期管理？

目前暂不支持表生命周期管理的配置。

# 5 数据开发

## 5.1 数据开发可以创建多少个作业，作业中的节点数是否有限制？

目前默认每个用户最多可以创建 10000 个作业，每个作业建议最多包含 200 个节点。

另外，系统支持用户根据实际需求调整最大配额。如有需求，请进行申请。

## 5.2 作业的计划时间和开始时间相差大，是什么原因？

如图所示，在作业监控页面查看作业运行记录时，发现作业的计划时间和开始时间相差较大。其中计划时间是作业预期开始执行的时间，即用户为作业配置的调度计划。开始时间是作业实际开始执行的时间。

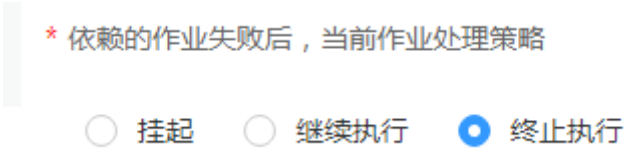
这是因为在数据开发中，单个作业最多允许 5 个实例并行执行，如果作业实际执行时间大于作业配置的调度周期，会导致后面批次的作业实例堆积，从而出现上述问题。

出现上述问题时，请检查作业配置的调度周期是否小于作业实际执行所需要的时间，根据实际情况调整作业的调度计划。

## 5.3 相互依赖的几个作业，调度过程中某个作业执行失败，是否会影响后续作业？这时该如何处理？

这种情况会影响后续作业，后续作业可能会挂起，继续执行或终止执行。

图5-1 作业依赖关系




这时请勿停止作业，您可以将失败的作业实例进行重跑，或者将异常的实例停止再重跑。失败实例成功后，后续作业会继续正常运行。如果不通过数据开发，手动将作业实例中的业务场景处理后，可以强制成功作业实例，后续作业也会继续正常运行。

## 5.4 通过 DataArts Studio 调度大数据服务时需要注意什么？

DLI 和 MRS 作为大数据服务，不具备锁管理的能力。因此如果同时对表进行读和写操作时，会导致数据冲突、操作失败。

如果您需要对大数据服务数据表进行读表和写表操作，建议参考以下方式之一进行串行操作处理：

- 将读表和写表操作拆分为同一作业的不同节点，两个节点通过连线建立先后执行关系，避免同时执行冲突。
- 将读表和写表操作拆分为两个不同的作业，两个作业之间设置依赖关系，避免同时执行冲突。

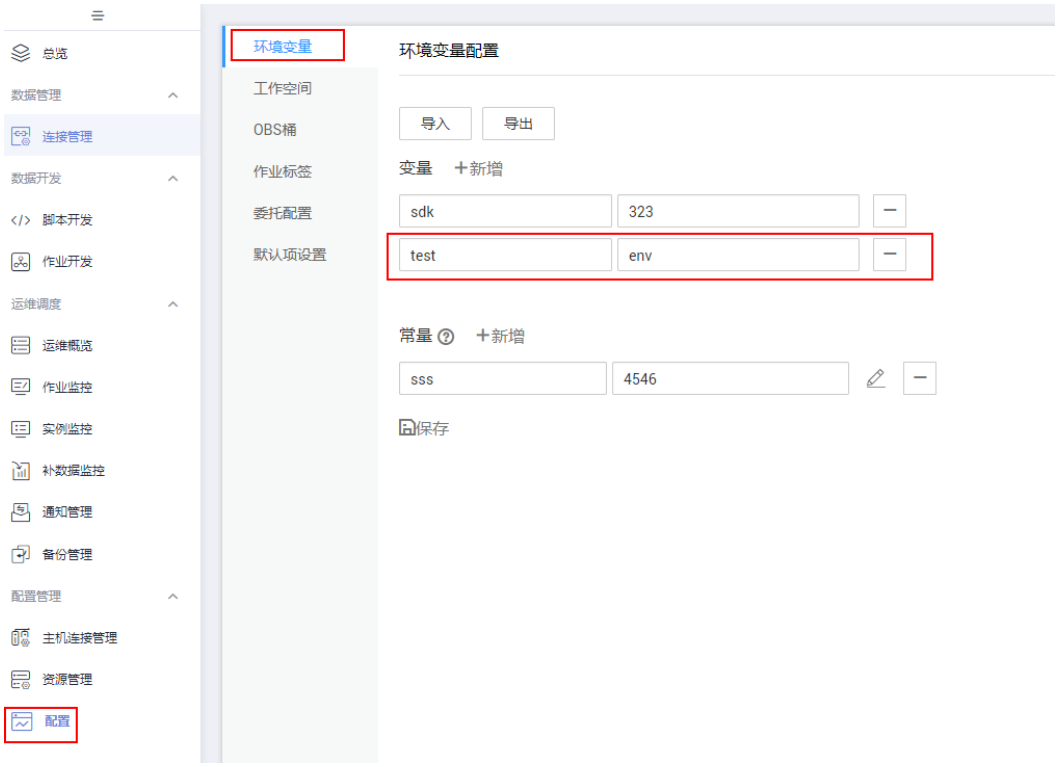
## 5.5 环境变量、作业参数、脚本参数有什么区别和联系？

环境变量、作业参数、脚本参数均可以配置参数，但作用范围不同；另外如果环境变量、作业参数、脚本参数同名冲突，调用的优先级顺序为：**作业参数** > **环境变量参数** > **脚本参数**。

环境变量、作业参数、脚本参数的介绍和使用方式如下：

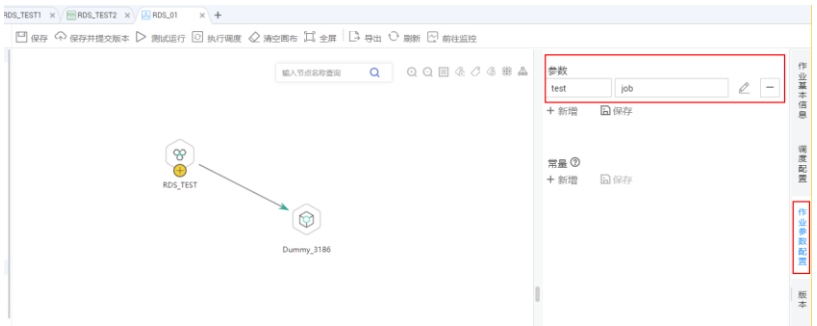
- 环境变量中支持定义变量和常量，环境变量的作用范围为当前工作空间。
  - 变量是指不同的空间下取值不同，需要重新配置值，比如“工作空间名称”变量，这个值在不同的空间下配置不一样，导出导入后需要重新进行配置。
  - 常量是指在不同的空间下都是一样的，导入的时候，不需要重新配置值。

图5-2 环境变量




- 作业参数中支持定义参数和常量，作业参数的作用范围为当前作业。
  - 参数是指不同的作业下取值不同，需要重新配置值，导出导入后需要重新进行配置。
  - 常量是指在不同的作业下都是一样的，导入的时候，不需要重新配置值。

图5-3 作业参数




- 脚本参数支持如下使用方式，脚本参数的作用范围为当前脚本。
  - SQL 脚本支持在脚本编辑器中直接输入参数（Flink SQL 不支持），脚本独立执行时可通过编辑器下方配置，如图 5-4 所示；通过作业调度时可通过节点属性赋值，如图 5-5 所示。
  - Shell 脚本可以在编辑器上方配置参数和交互式参数以实现参数传递功能。

- Python 脚本暂不支持参数传递功能。

图5-4 独立执行时的脚本参数

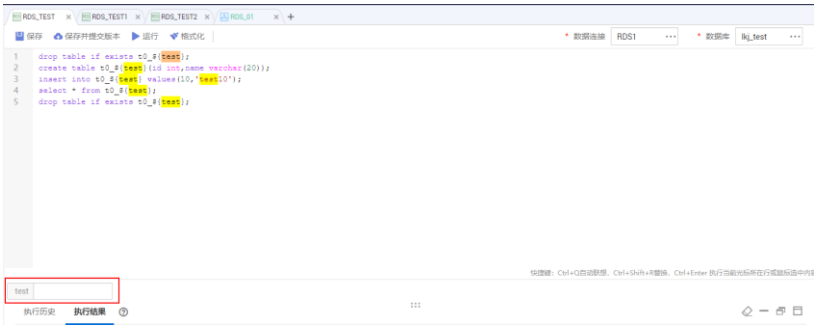


图5-5 作业调度时的脚本参数


## 5.6 作业失败无法查看节点错误日志?

错误日志是在 OBS 中存储，查看日志的当前账户需要具有 OBS 读权限。可以通过检查 IAM 中 OBS 权限、OBS 桶策略来确认。

**说明**

用户在创建作业时，会默认创建 dlf-log-{projectID}命名的桶，此桶若存在，会跳过创建。

## 5.7 配置委托时获取委托列表失败如何处理？

当配置工作空间级或者作业级委托，查看委托列表时，报如下错误：

Policy doesn't allow iam:agencies:listAgencies to be performed.

则需要使用帐号给当前用户添加“查看委托列表”的权限。

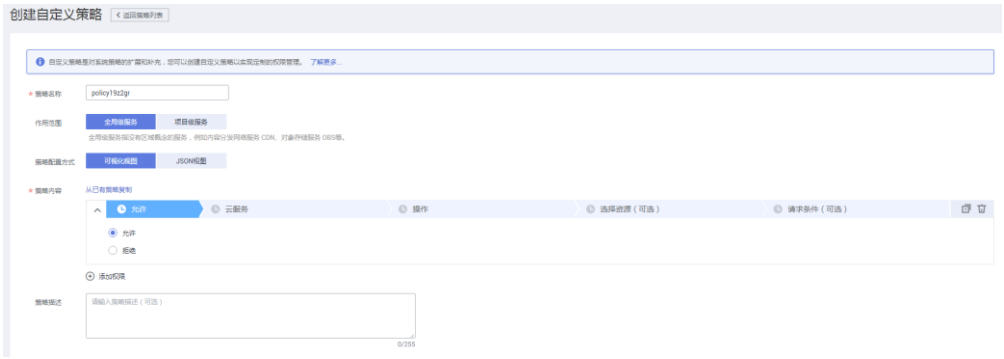
先创建自定义策略（查询指定条件下的委托列表），再通过给用户组授予自定义策略来进行精细的访问控制。

步骤 1 登录控制台。

步骤 2 在控制台页面，鼠标移动至右上方的帐号名，在下拉列表中选择“统一身份认证”。

步骤 3 在左侧导航窗格中，单击“权限”>“创建自定义策略”。

步骤 4 输入“策略名称”。




步骤 5 选择“作用范围”，即自定义策略的生效范围，根据服务的部署区域选择，这里我们要授予的是 IAM 查询指定条件下的委托列表的权限。因 IAM 是全局级服务，所以作用范围选择“全局级服务”。

步骤 6 “策略配置方式”选择“可视化视图”。

步骤 7 在“策略内容”下配置策略。

1. 选择“允许”。
2. 选择“云服务”为“统一身份认证服务”。
3. 选择“操作”，勾选产品权限（iam:agencies:listAgencies）。

步骤 8 单击“确定”，自定义策略创建完成。

步骤 9 参见，给当前用户所在的组添加步骤 7 中定义的策略。

当前用户退出系统，重新登录后，即可正常获取委托列表。

**----结束**

# 5.8 每日执行节点个数超过上限，怎么排查哪些作业调度节点比较多？

每日执行节点个数超过上限，一般是由于作业调度过于频繁导致的。可通过如下方式处理：

1. 在数据开发模块控制台的左侧导航栏，选择 “运维调度 > 实例监控”，日期选择当天，查看哪些作业调度较多。
2. 在数据开发主界面的左侧导航栏，选择“运维调度 > 作业监控”，查看调度较多的作业设置的调度周期是否合理。如果调度周期不合理，建议适当调整这些调度周期或停止调度。一般每日执行节点个数超过上限都是由于分钟级别的作业导致的。

图5-6 查看调度周期

## 5.9 数据开发创建数据连接，为什么选不到指定的周边资源？

请确认当前 DataArts Studio 实例与周边资源在同一个 Region 且在同一个 IAM 项目下。如果账户开通企业项目，则还需在同一个企业项目下。

## 5.10 作业配置了周期调度，但是实例监控没有作业运行调度记录？

1. 在“运维调度 > 作业监控”界面确认作业的调度状态是否是调度中，只有调度中的作业到了调度周期后才会调度。

图5-7 查看作业调度状态

2. 如果作业有依赖于其他作业，在“运维调度 > 实例监控”界面，查看依赖作业的运行状态。如果作业有自依赖，扩大搜索时间窗口，查看是否当前作业历史实例失败，导致作业在等待运行，而没有生成新作业实例。

## 5.11 Hive SQL 和 Spark SQL 脚本脚本执行失败，界面只显示执行失败，没有显示具体的错误原因？

请确认当前 Hive SQL 和 Spark SQL 脚本使用的数据连接为“直接连接”还是“通过代理连接”。

“直接连接”模式下 DataArts Studio 通过 API 把脚本提交给 MRS，然后查询是否执行完成；而 MRS 不会将具体的错误原因反馈到 DataArts Studio，因此导致数据开发脚本执行界面只能显示执行成功还是失败。

如果需要查看具体的错误原因，则需要到 MRS 的作业管理界面进行查看。

## 5.12 数据开发节点运行中报 TOKEN 不合法？

请确认当前用户在 IAM 的权限管理中权限是否有变更、是否退出用户组，或者用户所在的用户组权限策略是否有变更？

如果有变更，请重新登录即可解决。

## 5.13 作业开发时，测试运行后如何查看运行日志？

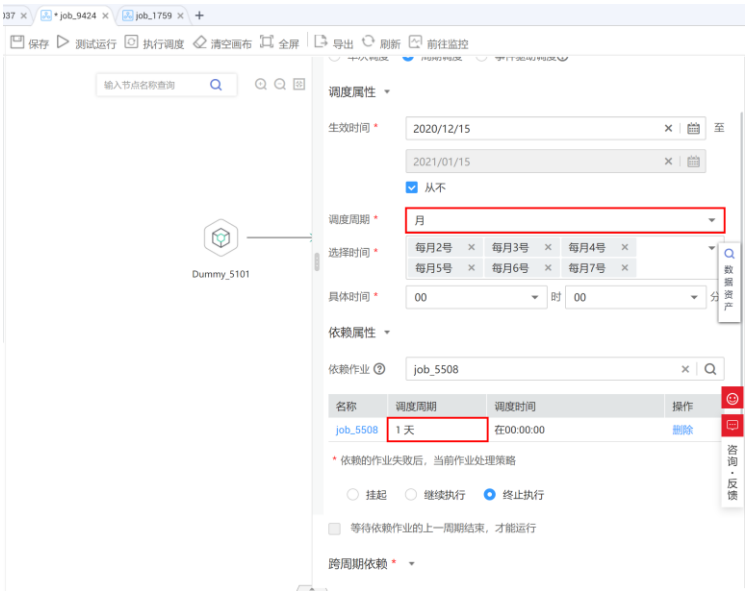
**方式 1：**待节点测试运行完成后，在当前节点鼠标右键选择查看日志。

**方式 2：**通过画布上方的“前往监控”，在实例监控中展开作业实例，查看节点日志。

## 5.14 月周期的作业依赖天周期的作业，为什么天周期作业还未跑完，月周期的作业已经开始运行？

如下图，月周期的作业依赖天周期的作业。为什么在天周期的作业还未跑完，月周期的作业已经开始运行？

图5-8 查看作业调度周期及依赖属性




事实上，月周期的作业依赖天周期作业指的是当月的月周期作业是否运行取决于上月的天周期作业是否全部运行完成，而不是由当月的天周期作业决定。

例如在 11 月中，11 月的月周期作业是否运行取决于 10 月的天周期作业是否全部运行完成。

# 5.15 执行 DLI 脚本，报 Invalid authentication 怎么办？

请确认当前用户在 IAM 中是否具有 DLI Service User 或者 DLI Service Admin 权限。

# 5.16 创建数据连接时，在代理模式下为什么选不到需要的 CDM 集群？

请确认 CDM 集群是否被关机。如果关机，请重新启动。

# 5.17 作业配置了每日调度，但是实例没有作业运行调度记录？

## 问题描述

作业配置了每日调度，但是实例没有作业运行调度记录。

### 原因分析

原因 1：确认作业是否启动调度，如果没有启动，不会进行调度。

原因 2：实例查询时间区间过大，如果配置有依赖作业或者自依赖，查看历史作业实例是否因为依赖失败，导致等待运行，没有生成新作业实例。

### 解决方案

配置作业失败异常告警通知，以及实例超时时间，当等待时间超过实例超时时间，系统将发送告警通知。

## 5.18 查看作业日志，但是日志中没有内容？

### 问题描述

查看作业日志，日志中没有内容。

### 原因分析

确认用户在 IAM 中的 OBS 权限是否具有对象存储服务（OBS）的全局权限，保证用户能够创建桶和操作桶。

### 解决方案

方式 1：用户在对象存储 OBS 中创建以“dlf-log-{projectID}”命名的桶，并将操作权限赋予调度用户。

方式 2：在 IAM 用户权限中增加全局 OBS 管理员权限。

## 5.19 创建了 2 个作业，但是为什么无法建立依赖关系？

### 问题描述

创建 2 个作业，但是无法建立依赖关系。

### 原因分析

查看所创建的 2 个作业的调度周期，确认这 2 个作业是否均为周调度作业或者月调度作业。目前不支持同周期调度，即周依赖周或者月依赖月的作业，不支持建立依赖关系。

### 解决方案

如果这 2 个作业是周依赖周或者月依赖月的作业，可以把这 2 个作业放到同一个画布中再运行。

# 5.20 DataArts Studio 执行调度时报错：提示作业没有可以提交的版本怎么办？

## 问题描述

DataArts Studio 执行调度时报错：作业没有已提交的版本，请先提交作业版本。

## 原因分析

该作业还没有提交版本，就开始执行调度，导致执行调度报错。作业执行调度前必须保证作业存在一个版本。

## 解决方案

1. 提交作业（不是脚本）版本。
2. 执行作业调度。

图5-9 提交版本


# 5.21 DataArts Studio 执行调度时报错：作业中节点 XXX 关联的脚本没有提交的版本？

## 问题描述

DataArts Studio 执行调度时报错：作业中节点 XXX 关联的脚本没有提交的版本。

## 原因分析

该作业内的脚本还没有提交版本，就开始执行调度，导致执行调度报错。作业调度前必须保证作业内脚本都存在一个版本。

## 解决方案

1. 切换到脚本开发，找到对应脚本。
2. 提交脚本版本。
3. 执行作业调度。

## 5.22 提交调度后的作业执行失败，报 depend job [XXX] is not running or pause 怎么办？

### 问题描述

提交调度后的作业执行失败，报 depend job [XXX] is not running or pause。

### 原因分析

该问题是由于上游依赖作业不在运行状态而造成。

### 解决方案

查看上游依赖作业，如果上游依赖的作业不在运行状态中，将这些作业重新执行调度即可。

## 5.23 如何创建数据库和数据表，数据库对应的是不是数据连接?

数据库和数据表可以在 DLI 服务中创建。

数据库对应的不是数据连接，数据连接是创建 DataArts Studio 和其他数据服务的连接通道。

## 5.24 为什么执行完 HIVE 任务什么结果都不显示？

解决方案：清理缓存数据，采用直连方式，数据就可以显示出来了。

## 5.25 在作业监控页面里的 “上次实例状态” 只有运行成功、运行失败，这是为什么？

上次实例状态是作业已经执行完成，只有成功、失败；实例监控里面状态有取消、暂停等好几种，是因为展示了作业的所有状态，另外作业运行异常和错误都会是作业失败的状态。

## 5.26 如何创建通知配置对全量作业都进行结果监控？

1. 在“运维调度->作业监控”中，选择“批作业监控”页签。
2. 勾选需要配置的作业，单击“通知配置”。

图5-10 创建通知配置


3. 设置通知配置参数，单击“确定”完成作业的通知配置。

# 5.27 DataArts Studio 的版本规格与并行执行节点数之间有什么关系？

DataArts Studio 的版本规格与并行执行节点数的关系如下表所示。

表5-1 DataArts Studio 的版本规格与并行执行节点数的关系

| 版本 | 每天执行节点数 | 并行执行节点数 |
| --- | --- | --- |
| 初级版 | 5 千 | 50 |
| 基础版 | 2 万 | 100 |
| 高级版 | 4 万 | 200 |
| 专业版 | 8 万 | 300 |
| 企业版 | 20 万 | 400 |

# 5.28 启动用户、执行用户、工作空间委托、作业委托它们之间的优先级顺序是什么？

系统按照作业委托>工作空间委托>执行用户的优先级顺序来获取权限，然后以该权限来执行作业。

作业执行机制默认以启动作业的用户身份执行该作业。如果作业被低权限的用户启动，也会因为权限不足导致作业执行失败。若需解决该问题，可通过配置委托或者执行用户。

- 当配置了委托后，作业执行过程中，以委托的身份与其他服务交互，可以避免权限问题导致的作业执行失败。委托分两类，工作空间委托和作业委托，作业委托优先级高于工作空间委托。
  - 工作空间委托：工作空间级别的全局委托，适用于该空间内的所有作业。可在数据开发模块的配置>委托配置，配置工作空间委托。
  - 作业委托：适用于单个作业级别。可在作业基本信息，配置作业委托。
- 当配置了执行用户后，会以执行用户的身份来启动作业。可在作业基本信息，配置执行用户。

# 6 数据质量

## 6.1 质量作业和对账作业有什么区别？

- 质量作业可将创建的规则应用到建好的表中进行质量监控。
- 对账作业支持跨源数据对账能力，可将创建的规则应用到两张表中进行质量监控，并输出对账结果。

  数据对账对于数据开发和数据迁移流程中的数据一致性至关重要，而跨源数据对账的能力是检验数据迁移或数据加工前后是否一致的关键指标。

## 6.2 如何确认质量作业或对账作业已经阻塞？

作业运行状态长时间处于运行中时，选择“运维管理”，点击操作栏中的“结果&日志”并选择查看“运行日志”，当“运行日志”不再更新，表示作业已经阻塞。


## 6.3 如何手工重启阻塞的质量作业或对账作业？

阻塞的作业需要进行手工重启，如不重启 1 天内也会因作业超时自动结束该作业。

手工重启需要选择“运维管理”，先点击对应作业操作栏中的“取消”，作业运行状态变更为“失败”，此时然后点击操作栏中的“重跑”即可完成作业重启。

# 6.4 怎样查看质量规则模板关联的作业？

步骤 1　单击待操作规则模板操作列的“发布历史”。

图6-1 发布历史


步骤 2　点击历史版本最右侧的“下线”按钮。则可以查看该规则模板对应的关联作业。

图6-2 查看关联作业


----结束

# 6.5 用户在执行质量作业时提示无 MRS 权限怎么办？

用户在执行质量作业时报错，查看质量作业的日志，提示“ The current user does not exist on MRS Manager. Grant the user sufficient permissions on IAM and then perform IAM user synchronization on the Dashboard tab page. !”

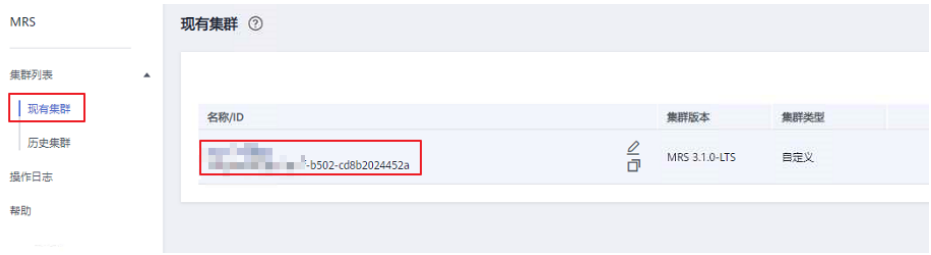
此类问题一般是由于用户不具备 MRS 集群操作权限导致的。

对于租户下新增的用户，需要在 MRS 集群列表的界面找到对应的 MRS 集群实例，手动单击同步。

操作如下：

步骤 1　进入 MRS 控制台，查看现有集群，单击对应的集群名称进入概览页。

图6-3 MRS 集群实例




步骤 2　在“IAM 用户同步”处，单击同步。

图6-4 单击同步


**步骤 3** 在操作日志处查看操作结果。

图6-5 操作日志


步骤 4 如果经过上述步骤，帐号已同步。但还是提示 Mrs 权限不足的话，则需要登录到 Manger 管理页面中创建一个与当前主帐号同名的帐号。

**⚠ 注意**

在步骤 4 中，需要创建一个与当前主帐号同名的帐号。

----结束

# 7 数据目录

## 7.1 数据目录组件有什么用？

数据目录的核心是通过元数据采集任务，采集并展示企业的数据资产地图，包括所有的元数据信息和数据血缘关系。

## 7.2 数据目录支持采集哪些对象的资产？

数据目录目前支持采集的资产有：数据仓库服务（DWS）、MapReduce 服务（MRS HBase）、MapReduce 服务（MRS Hive）、MySQL、云数据库 RDS（DataArts Studio 仅支持 MySQL 和 PostgreSQL 数据库）。

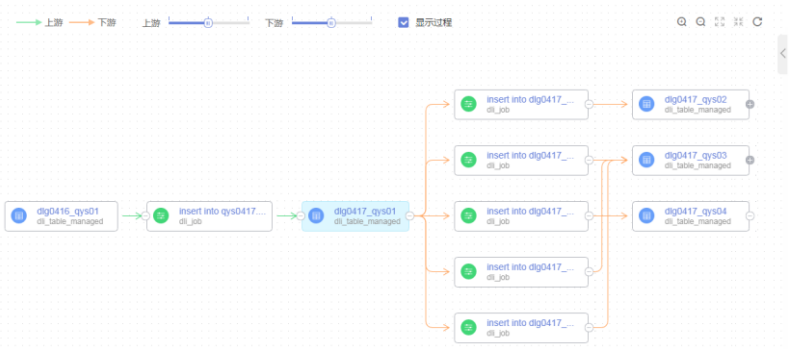
## 7.3 什么是数据血缘关系？

大数据时代，数据爆发性增长，海量的、各种类型的数据在快速产生。这些庞大复杂的数据信息，通过联姻融合、转换变换、流转流通，又生成新的数据，汇聚成数据的海洋。

数据的产生、加工融合、流转流通，到最终消亡，数据之间自然会形成一种关系。我们借鉴人类社会中类似的一种关系来表达数据之间的这种关系，称之为数据的血缘关系。与人类社会中的血缘关系不同，数据的血缘关系还包含了一些特有的特征：

- **归属性**：一般来说，特定的数据归属特定的组织或者个人，数据具有归属性。
- **多源性**：同一个数据可以有多个来源（多个父亲）。一个数据可以是多个数据经过加工而生成的，而且这种加工过程可以是多个。
- **可追溯性**：数据的血缘关系，体现了数据的生命周期，体现了数据从产生到消亡的整个过程，具备可追溯性。
- **层次性**：数据的血缘关系是有层次的。对数据的分类、归纳、总结等对数据进行的描述信息又形成了新的数据，不同程度的描述信息形成了数据的层次。

图7-1 数据血缘关系示例




## 7.4 数据目录如何可视化展示数据血缘？

数据血缘展示，首先要需要有相关的作业调度，其次要进行元数据采集。

# 8 数据服务

## 8.1 创建 API 时提示代理调用失败，怎么办？

需要在空余时间对 CDM 集群进行重启释放内存。

## 8.2 数据服务 API 接口，访问“测试 APP”，填写了相关参数，但是后台报错要怎么处理？

在调用 API 时配置参数 header parameter。

```
header parameter: x-Authorization, nvalid ___ parameter: ___,
```

## 8.3 使用 API 时报错，请问有什么办法可以解决？

使用 API 时需注意，每个子域名每天最多可以访问 1000 次。

## 8.4 API 传参是否支持传递操作符？

不支持传递操作符，传递的只是参数，操作符是固定的，多个参数可使用 in(${})方式。

## 8.5 数据服务专享版提供的 API 配额已满怎么解决？

如果数据服务专享版提供的 API 配额已满，无法创建新的 API 时可修改 API 配额。