

数据湖探索

用户指南

文档版本 01
发布日期 2025-01-26



版权所有 © 华为技术有限公司 2025。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编： 518129

网址： <https://www.huawei.com>

客户服务邮箱： support@huawei.com

客户服务电话： 4008302118

安全声明

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：

<https://www.huawei.com/cn/psirt/vul-response-process>

如企业客户须获取漏洞信息，请参见如下网址：

<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>

目录

1 DLI 作业开发流程	1
2 准备工作	5
2.1 配置 DLI 云服务委托权限	5
2.2 创建 IAM 用户并授权使用 DLI	10
2.3 配置 DLI 作业桶	12
3 创建弹性资源池和队列	14
3.1 DLI 弹性资源池与队列简介	14
3.2 创建弹性资源池并添加队列	18
3.3 管理弹性资源池	25
3.3.1 查看弹性资源池的基本信息	25
3.3.2 弹性资源池权限管理	26
3.3.3 弹性资源池关联队列	29
3.3.4 弹性资源池 CU 设置	29
3.3.5 弹性资源池规格变更	32
3.3.6 弹性资源池标签管理	34
3.3.7 调整弹性资源池中队列的扩缩容策略	36
3.3.8 查看弹性资源池扩缩容历史	41
3.3.9 分配弹性资源池至项目	42
3.4 管理队列	43
3.4.1 查看队列的基本信息	43
3.4.2 队列权限管理	44
3.4.3 分配队列至项目	46
3.4.4 创建消息通知主题	46
3.4.5 队列标签管理	48
3.4.6 队列属性设置	50
3.4.7 测试地址连通性	51
3.4.8 删除队列	52
3.4.9 变更普通队列规格	52
3.4.10 普通队列弹性扩缩容	54
3.4.11 设置普通队列的弹性扩缩容定时任务	57
3.4.12 修改普通队列的网段	60
3.5 典型场景示例：创建弹性资源池并运行作业	61

3.6 典型场景示例：配置弹性资源池队列扩缩容策略.....	67
3.7 创建非弹性资源池队列（废弃，不推荐使用）.....	72
4 创建数据库和表.....	77
4.1 了解数据目录、数据库和表.....	77
4.2 在 DLI 控制台创建数据库和表.....	79
4.3 查看表元数据.....	86
4.4 在 DLI 控制台管理数据库资源.....	87
4.4.1 在 DLI 控制台配置数据库权限.....	87
4.4.2 在 DLI 控制台删除数据库.....	91
4.4.3 在 DLI 控制台修改数据库所有者.....	91
4.4.4 库表管理标签管理.....	92
4.5 在 DLI 控制台管理表资源.....	95
4.5.1 在 DLI 控制台配置表权限.....	95
4.5.2 在 DLI 控制台删除表.....	103
4.5.3 在 DLI 控制台修改表所有者.....	103
4.5.4 将 OBS 数据导入至 DLI.....	104
4.5.5 导出 DLI 表数据至 OBS 中.....	107
4.5.6 在 DLI 控制台预览表数据.....	109
4.6 创建并使用 LakeFormation 元数据.....	110
4.6.1 DLI 对接 LakeFormation.....	110
4.6.2 LakeFormation 资源权限支持列表与策略项.....	131
5 数据迁移与数据传输.....	136
5.1 数据迁移与传输方式概述.....	136
5.2 迁移外部数据源数据至 DLI.....	137
5.2.1 迁移数据场景概述.....	137
5.2.2 使用 CDM 迁移数据至 DLI.....	139
5.2.3 典型场景示例：迁移 Hive 数据至 DLI.....	146
5.2.4 典型场景示例：迁移 Kafka 数据至 DLI.....	154
5.2.5 典型场景示例：迁移 Elasticsearch 数据至 DLI.....	161
5.2.6 典型场景示例：迁移 RDS 数据至 DLI.....	167
5.2.7 典型场景示例：迁移 DWS 数据至 DLI.....	173
5.3 配置 DLI 读写外部数据源数据.....	180
5.3.1 配置 DLI 读写外部数据源数据的操作流程.....	180
5.3.2 配置 DLI 与数据源网络连通（增强型跨源连接）.....	180
5.3.2.1 增强型跨源连接概述.....	180
5.3.2.2 创建增强型跨源连接.....	182
5.3.2.3 建立 DLI 与共享 VPC 中资源的网络连接.....	187
5.3.2.4 DLI 常用跨源分析开发方式.....	189
5.3.3 使用 DEW 管理数据源访问凭证.....	191
5.3.4 使用 DLI 的跨源认证管理数据源访问凭证.....	193
5.3.4.1 跨源认证概述.....	193
5.3.4.2 创建 CSS 类型跨源认证.....	195

5.3.4.3 创建 Kerberos 跨源认证.....	197
5.3.4.4 创建 Kafka_SSL 类型跨源认证.....	200
5.3.4.5 创建 Password 类型跨源认证.....	203
5.3.4.6 跨源认证权限管理.....	205
5.3.5 管理增强型跨源连接.....	206
5.3.5.1 查看增强型跨源连接的基本信息.....	206
5.3.5.2 增强型跨源连接权限管理.....	207
5.3.5.3 增强型跨源连接绑定弹性资源池.....	208
5.3.5.4 增强型跨源连接与弹性资源池解绑.....	209
5.3.5.5 添加增强型跨源连接的路由信息.....	209
5.3.5.6 删除增强型跨源连接的路由信息.....	210
5.3.5.7 修改弹性资源池的主机信息.....	211
5.3.5.8 增强型跨源连接标签管理.....	213
5.3.5.9 删除增强型跨源连接.....	214
5.3.6 典型场景示例：配置 DLI 与内网数据源的网络联通.....	214
5.3.7 典型场景示例：配置 DLI 与公网网络连通.....	219
6 配置 DLI 访问其他云服务的委托权限.....	226
6.1 DLI 委托概述.....	226
6.2 创建 DLI 自定义委托权限.....	228
6.3 常见场景的委托权限策略.....	234
6.4 典型场景 DLI 委托权限配置示例.....	237
7 使用 DLI 提交 SQL 作业.....	238
7.1 创建并提交 SQL 作业.....	238
7.2 导出 SQL 作业结果.....	242
7.3 配置 SQL 防御规则.....	245
7.4 设置 SQL 作业优先级.....	250
7.5 查询 SQL 作业日志.....	250
7.6 管理 SQL 作业.....	252
7.7 创建并管理 SQL 作业模板.....	255
7.7.1 创建 SQL 作业模板.....	255
7.7.2 使用 SQL 作业模板开发并提交 SQL 作业.....	257
7.7.3 DLI 预置的 SQL 模板中 TPC-H 样例数据说明.....	259
8 使用 DLI 提交 Flink 作业.....	262
8.1 Flink 作业概述.....	262
8.2 创建 Flink OpenSource SQL 作业.....	262
8.3 创建 Flink Jar 作业.....	271
8.4 配置 Flink 作业权限.....	279
8.5 管理 Flink 作业.....	282
8.5.1 查看 Flink 作业详情.....	282
8.5.2 设置 Flink 作业优先级.....	288
8.5.3 开启 Flink 作业动态扩缩容.....	290

8.5.4 查询 Flink 作业日志.....	291
8.5.5 Flink 作业常用操作.....	294
8.6 管理 Flink 作业模板.....	299
8.7 添加 Flink 作业标签.....	305
9 使用 DLI 提交 Spark 作业.....	308
9.1 创建 Spark 作业.....	308
9.2 设置 Spark 作业优先级.....	317
9.3 查询 Spark 作业日志.....	318
9.4 管理 Spark 作业.....	320
9.5 管理 Spark 作业模板.....	321
10 使用 Notebook 实例提交 DLI 作业.....	322
11 使用 CES 监控 DLI 服务.....	335
12 使用 AOM 监控 DLI 服务.....	340
12.1 配置 DLI 对接 AOM Prometheus 监控.....	340
12.2 DLI 对接 AOM Prometheus 监控的配置项.....	344
12.3 DLI 支持的 Prometheus 基础监控指标.....	345
13 使用 CTS 审计 DLI 服务.....	351
14 权限管理.....	354
14.1 权限管理概述.....	354
14.2 DLI 自定义策略.....	357
14.3 DLI 资源.....	363
14.4 DLI 请求条件.....	364
14.5 常用操作与系统权限关系.....	365
15 DLI 常用管理操作.....	369
15.1 使用自定义镜像增强作业运行环境.....	369
15.2 管理 DLI 全局变量.....	373
15.3 管理 Jar 作业程序包.....	375
15.3.1 程序包管理概述.....	375
15.3.2 创建 DLI 程序包.....	377
15.3.3 配置 DLI 程序包权限.....	379
15.3.4 修改 DLI 程序包所有者.....	382
15.3.5 DLI 程序包标签管理.....	383
15.3.6 DLI 内置依赖包.....	384
15.4 管理 DLI 资源配额.....	412

1 DLI 作业开发流程

本节内容为您介绍DLI作业开发流程。

创建 IAM 用户并授权使用 DLI

- 如果您是企业用户，并计划使用IAM对您所拥有的DLI资源进行精细的权限管理，请创建IAM用户并授权使用DLI。具体操作请参考[创建IAM用户并授权使用DLI](#)。
- 首次使用DLI您需要根据控制台的引导更新DLI委托，用于将操作权限委托给DLI服务，让DLI服务以您的身份使用其他云服务，代替您进行一些资源运维工作。该委托包含获取IAM用户相关信息、跨源场景访问和使用VPC、子网、路由、对等连接的权限、作业执行失败需要通过SMN发送通知消息的权限。

详细委托包含的权限请参考[配置DLI云服务委托权限](#)。

创建执行作业所需的计算资源和元数据

- 使用DLI提交作业前，您需要先创建弹性资源池，并在弹性资源池中创建队列，为提交作业准备所需的计算资源。请参考[DLI弹性资源池与队列简介](#)创建弹性资源池并添加队列。

您还可以通过自定义镜像增强DLI的计算环境，通过下载DLI提供的基础镜像再按需制作自定义镜像，将作业运行需要的依赖（文件、jar包或者软件）、私有能力等内置到自定义镜像中，可以改变Spark作业和Flink作业的容器运行环境，增强作业的功能、性能。

例如，在自定义镜像中加入机器学习相关的Python包或者C库，可以通过这种方式帮助用户实现功能扩展。创建自定义镜像请参考[使用自定义镜像增强作业运行环境](#)。

- DLI元数据是SQL作业、Spark作业场景开发的基础。在执行作业前您需要根据业务场景定义数据库和表。

📖 说明

- Flink支持动态数据类型，可以在运行时定义数据结构，不需要事先定义元数据。
- 定义您的数据结构，包括数据目录、数据库、表。请参考[创建数据库和表](#)。
- 创建必要的存储桶来存储作业运行过程中产生的临时数据：作业日志、作业结果等。请参考[配置DLI作业桶](#)。
- 配置元数据的访问权限。请参考[在DLI控制台配置数据库权限](#)、[在DLI控制台配置表权限](#)。

DLI 数据导入指引

- DLI支持在不迁移数据的情况下，直接对OBS中存储的数据进行查询分析，您只需要将数据上传OBS即可使用DLI进行数据分析。
上传数据至OBS请参考《[对象存储用户指南](#)》。
- 当需要将来自不同源的数据进行集中存储和处理时，迁移数据至DLI可以提供统一的数据平台。
您可以参考[使用CDM迁移数据至DLI](#)迁移数据至DLI后再提交作业。
- 如果业务需求需要实时访问和处理来自不同数据源的数据，跨源访问可以减少数据的复制和延迟。
跨源访问的必要条件包括“DLI与数据源网络连通”、“DLI可获取数据源的访问凭证”：
 - DLI与数据源网络连通：您可以参考[配置DLI与数据源网络连通（增强型跨源连接）](#)配置DLI与数据源的网络连通。
 - 管理数据源的凭证：
 - 您可以使用DLI提供的跨源认证功能管理访问指定数据源的认证信息。
适用范围：SQL作业、Flink 1.12作业场景。具体操作请参考[使用DLI的跨源认证管理数据源访问凭证](#)。
 - 您还可以使用DEW管理数据源的访问凭证，并通过“自定义委托”方式授予DLI访问DEW服务的权限。
适用范围：Spark 3.3.1及以上版本、Flink 1.15及以上版本。具体操作请参考[使用DEW管理数据源访问凭证](#)和[配置DLI访问其他云服务的委托权限](#)。

使用 DLI 提交作业

- DLI提供一站式的流处理、批处理、交互式分析的Serverless融合处理分析服务，支持多种作业类型以满足不同的数据处理需求。

表 1-1 DLI 支持的作业类型

作业类型	说明	适用场景
SQL作业	适用于使用标准SQL语句进行查询的场景。通常用于结构化数据的查询和分析。 详细操作请参考 创建并提交SQL作业 。	适用于数据仓库查询、报表生成、OLAP（在线分析处理）等场景。

作业类型	说明	适用场景
Flink作业	<p>专为实时数据流处理设计，适用于低时延、需要快速响应的场景。适用于实时监控、在线分析等场景。</p> <ul style="list-style-type: none"> Flink OpenSource作业：DLI提供了标准的连接器（connectors）和丰富的API，便于快速与其他数据系统的集成。详细操作请参考创建Flink OpenSource SQL作业。 Flink Jar作业：允许用户提交编译为Jar包的Flink作业，提供了更大的灵活性和自定义能力。适合需要自定义函数、UDF（用户定义函数）或特定库集成的复杂数据处理场景。可以利用Flink的生态系统，实现高级流处理逻辑和状态管理。详细操作请参考创建Flink Jar作业。 	<p>适用于实时数据监控、实时推荐系统等需要快速响应的场景。</p> <p>Flink Jar作业适用于需要自定义流处理逻辑、复杂的状态管理或特定库集成的数据分析场景。</p>
Spark作业	<p>可通过交互式会话（session）和批处理（batch）方式提交计算任务。通过在DLI提供的弹性资源池队列上提交作业，简化了资源管理和作业调度。</p> <p>支持多种数据源和格式，提供了丰富的数据处理能力，包括但不限于SQL查询、机器学习等。详细操作请参考创建Spark作业。</p>	<p>适用于大规模数据处理和分析，如机器学习训练、日志分析、大规模数据挖掘等场景。</p>

- 管理Jar作业的程序包

DLI允许用户提交编译为Jar包的Flink或Spark作业，Jar包中包含了Jar作业执行所需的代码和依赖信息，用于在数据查询、数据分析、机器学习等特定的数据处理任务中使用。通过DLI管理控制台可以管理作业所需的呈现包。

在提交Spark Jar和Flink Jar类型的作业前，需要将程序包上传至OBS，然后在DLI服务中创建程序包，并将程序包与数据和作业参数一起提交以运行作业。[管理Jar作业程序包](#)。

说明

Spark3.3.1及以上版本、Flink1.15及以上版本在创建Jar作业时支持直接配置OBS中的程序包，不支持读取DLI程序包。

使用 CES 监控 DLI 服务

您可以通过云监控服务提供的管理控制台或API接口来检索数据湖探索服务产生的监控指标和告警信息。

例如监控DLI队列资源使用量和作业的运行情况。了解更多DLI支持的监控指标请参考[使用CES监控DLI服务](#)。

使用 CTS 审计 DLI 服务

通过云审计服务，您可以记录与DLI服务相关的操作事件，便于日后的查询、审计和回溯。了解更多审计支持列表请参考[使用CTS审计DLI服务](#)。

2 准备工作

2.1 配置 DLI 云服务委托权限

使用DLI服务前请先配置DLI云服务权限。

本节操作适用于以下场景：

- **首次使用DLI服务，请参考本节操作按需配置DLI云服务委托权限。**
使用DLI的过程中需要与其他云服务协同工作，因此需要您将部分服务的操作权限委托给DLI服务，确保DLI具备基本使用的权限，让DLI服务以您的身份使用其他云服务，代替您进行一些资源运维工作。
- **仍在使用的DLI上一代委托dli_admin_agency，请参考本节操作更新DLI委托。**
为了解决在满足实际业务使用的同时，避免委托权限过大的风险，DLI升级了系统委托，做到更细粒度的委托权限控制，将原有的dli_admin_agency升级为dli_management_agency，新的委托包含获取IAM用户信息、跨源操作、消息通知所需的权限。有效避免DLI相关联服务权限不受控制的问题。升级后的DLI委托灵活性更强，更适合中大型企业场景化定制委托的需求。


配置DLI云服务的委托权限后会在IAM委托页面生成dli_management_agency的委托。请勿删除系统默认创建的dli_management_agency委托，否则会导致委托包含的权限自动取消，系统将无法正常获取IAM用户相关信息、或影响访问跨源所需的网络资源、无法访问SMN服务发送通知消息。

约束限制

- 服务授权需要主账号或者用户组admin中的子账号进行操作。
- DLI服务授权需要区分项目，请在每个需要新委托的项目分别执行更新委托操作，即切换至对应项目后，再按照本节的操作更新委托权限。

更新 DLI 委托权限（dli_management_agency）

1. 在DLI控制台左侧导航栏中单击“全局配置 > 服务授权”。
2. 在委托设置页面，按需选择以下场景的权限。

单击权限卡片上的 可以查看包含的详细的权限策略。

委托说明如表2-1所示。

表 2-1 dli_management_agency 委托包含的权限

适用场景	委托名	权限说明
基础使用	IAM ReadOnlyAccess	DLI对未登录过DLI的用户进行授权时，需获取IAM用户相关信息。因此需要IAM ReadOnlyAccess权限。 IAM ReadOnlyAccess是全局权限，请务必勾选该权限，否则IAM ReadOnlyAccess权限将在所有区域失效，系统将无法正常获取IAM用户相关信息。
跨源场景	DLI Datasource Connections Agency Access	访问和使用VPC、子网、路由、对等连接的权限
运维场景	DLI Notification Agency Access	作业执行失败需要通过SMN发送通知消息的权限

📖 说明

dli_management_agency包含的三个权限中：

- IAM ReadOnlyAccess授权范围是全局服务资源，授权范围不区分区域：
 - 任意区域在更新DLI委托时选择了该权限，那么所有区域的项目都将生效。
 - 任意项目在更新委托时未勾选该权限，代表回收该权限，那么所有区域的项目都将回收该权限，即所有项目无法正常获取IAM用户相关信息。
- DLI Datasource Connections Agency Access、DLI Notification Agency Access授权范围是指定区域项目资源：

仅在勾选该权限且更新DLI委托权限后的项目生效。未勾选该权限的项目不具备跨源场景所需权限、和SMN发送通知消息的权限。

示例1：在项目A配置DLI的基础使用、跨源场景、运维场景的权限和示例2：在项目B配置DLI的基础使用、跨源场景、运维场景的权限给出了同一个区域的不同项目更新DLI委托带来的委托权限差异。

3. 单击选择dli_management_agency需要包含的权限，并单击“更新委托权限”。

图 2-1 更新委托权限



- 查看并了解更新委托的提示信息，单击“确定”。完成DLI委托权限的更新。
 - 更新委托权限后，系统将升级您的dli_admin_agency为dli_management_agency。
 - 为兼容存量的作业委托权限需求，dli_admin_agency仍为您保留在IAM委托中。
 - 请勿删除系统默认创建的委托。

后续操作

除dli_management_agency提供的委托权限外，一些场景需要用户自行在IAM页面创建相关委托，并在作业配置中添加新建的委托信息。例如允许DLI读写OBS将日志转储、允许DLI在访问DEW获取数据访问凭证场景的委托需求等，具体操作请参考[创建DLI自定义委托权限](#)和[常见场景的委托权限策略](#)。

- 使用Flink 1.15和Spark 3.3.1(Spark通用队列场景)及以上版本的引擎执行作业时，需自行在IAM页面创建相关委托。
- 引擎版本低于Flink1.15，执行作业时默认使用dli_admin_agency；引擎版本低于Spark 3.3.1，执行作业时使用用户认证信息（AKSK、SecurityToken）。即引擎版本低于Flink1.15和Spark 3.3.1版本的作业不受更新委托权限的影响，无需自定义委托。

常见的需要自建委托的业务场景：

- DLI表生命周期清理数据及Lakehouse表数据清理所需的数据清理委托。需用户自行在IAM创建名为dli_data_clean_agency的DLI云服务委托并授权。该委托需新建后自定义权限，但委托名称固定为dli_data_clean_agency。
- DLI Flink作业访问和使用OBS、日志转储（包括桶授权）、开启checkpoint、作业导入导出等，需要获得访问和使用OBS（对象存储服务）的Tenant Administrator权限。
- DLI Flink作业所需的AKSK存储在数据加密服务DEW中，如需允许DLI在执行作业时访问DEW数据，需要新建委托将DEW数据操作权限委托给DLI，允许DLI服务以您的身份访问DEW服务。

- 允许DLI在执行作业时访问DLI Catalog元数据，需要新建委托将DLI Catalog数据操作权限委托给DLI，允许DLI服务以您的身份访问DLI Catalog元数据。
- DLI Flink作业所需的云数据存储于LakeFormation中，如需允许DLI在执行作业时访问Catalog获取元数据，需要新建委托将Catalog数据操作权限委托给DLI，允许DLI服务以您的身份访问Catalog元数据。

新建委托时，请注意委托名称不可与系统默认委托重复，即不可以是 dli_admin_agency、dli_management_agency、dli_data_clean_agency。

更多自定义委托的操作请参考[创建DLI自定义委托权限](#)和[常见场景的委托权限策略](#)。

示例 1：在项目 A 配置 DLI 的基础使用、跨源场景、运维场景的权限

- **操作说明：**某DLI用户按系统的指引在华北-北京四的项目A将原有的 dli_admin_agency 升级为 dli_management_agency：
 - 在DLI管理控制台，切换至华北-北京四区域下的项目A，选择“全局配置 > 服务授权”。
 - 勾选基础使用、跨源场景、运维场景的权限。

图 2-2 华北-北京四的项目 A 更新委托权限示意图

管理相关委托设置 (委托名: dli_management_agency)



- 单击“更新委托权限”。
- **权限说明：**华北-北京四的项目A按上述操作步骤更新委托权限后，
 - 由于IAM ReadOnlyAccess授权范围是全局服务资源，所以所有区域和项目均具备该权限。
 - 而跨源场景的权限“DLI Datasource Connections Agency Access”和消息通知的权限“DLI Notification Agency Access”是区域级权限，所以仅在华北-北京四的项目A生效。

华北-北京四的项目A委托权限示例	华北-北京四的项目B委托权限示例
dli_management_agency 包含以下权限： <ul style="list-style-type: none"> • IAM ReadOnlyAccess • DLI Datasource Connections Agency Access • DLI Notification Agency Access 	dli_management_agency 包含以下权限： <ul style="list-style-type: none"> • IAM ReadOnlyAccess

示例 2：在项目 B 配置 DLI 的基础使用、跨源场景、运维场景的权限

操作说明：如需在华北-北京四的项目B授权跨源场景的权限和消息通知的权限，需按如下操作更新华北-北京四的项目B的委托权限。

1. 在DLI管理控制台，切换至华北-北京四的项目B，选择“全局配置 > 服务授权”。
2. 勾选基础使用、跨源场景、运维场景的权限。

⚠ 注意

项目B更新时请务必勾选IAM ReadOnlyAccess，因IAM ReadOnlyAccess授权范围是全局服务资源，如果取消勾选该权限后更新委托，则所有区域和项目的IAM ReadOnlyAccess都将失效。

图 2-3 华北-北京四的项目 B 更新委托权限示意图

管理相关委托设置 (委托名: dli_management_agency)



3. 单击“更新委托权限”。

权限说明：

由于跨源场景的权限“DLI Datasource Connections Agency Access”和消息通知的权限“DLI Notification Agency Access”的授权范围是指定区域项目资源，因此按上述操作步骤更新委托权限后，华北-北京四的项目B将同时具备获取IAM用户相关信息的权限、跨源场景的权限和消息通知的权限。

区域A委托权限示例	区域B委托权限示例
dli_management_agency 包含以下权限： <ul style="list-style-type: none">• IAM ReadOnlyAccess• DLI Datasource Connections Agency Access• DLI Notification Agency Access	dli_management_agency 包含以下权限： <ul style="list-style-type: none">• IAM ReadOnlyAccess• DLI Datasource Connections Agency Access• DLI Notification Agency Access

2.2 创建 IAM 用户并授权使用 DLI

如果您需要对您所拥有的DLI资源进行精细的权限管理，您可以使用统一身份认证服务（Identity and Access Management，简称IAM），具体IAM使用场景可以参考[权限管理概述](#)。

如果华为云账号已经能满足您的要求，不需要创建独立的IAM用户，您可以跳过本章节，不影响您使用DLI服务的其它功能。

本章节介绍创建IAM用户并授权使用DLI的方法，操作流程如[图2-4](#)所示。

前提条件

给用户组授权之前，请您先了解用户组可以添加的DLI权限，并结合实际需求进行选择。DLI支持的系统权限，请参见：[DLI系统权限](#)。

示例流程

图 2-4 给用户授权 DLI 权限流程



1. 创建用户组并授权

在IAM控制台创建用户组，并授予DLI服务普通用户权限“DLI ReadOnlyAccess”。

2. 创建用户并加入用户组

在IAM控制台创建用户，并将其加入1中创建的用户组。

3. 用户登录并验证权限

使用新创建的用户登录控制台，切换至授权区域，验证权限：

- 在“服务列表”中选择数据湖探索，进入DLI主界面。如果在“队列管理”页面可以查看队列列表，但是单击右上角“购买队列”，无法购买DLI队列（假设当前权限仅包含DLI ReadOnlyAccess），表示“DLI ReadOnlyAccess”已生效。



- 在“服务列表”中选择除数据湖探索外（假设当前策略仅包含DLI ReadOnlyAccess）的任一服务，如果提示权限不足，表示“DLI ReadOnlyAccess”已生效。

更多操作

- 创建子用户请参考《[如何创建子用户](#)》。
- 创建自定义策略请参考[DLI自定义策略](#)。
- 修改用户策略请参考《[如何修改用户策略](#)》。

2.3 配置 DLI 作业桶

使用DLI服务前需配置DLI作业桶，该桶用于存储DLI作业运行过程中产生的临时数据，例如：作业日志、作业结果。

本节操作指导您在DLI管理控制台的“全局配置 > 工程配置”页面配置DLI作业桶。

操作前准备

配置前，请先购买**OBS桶**或**并行文件系统**。

大数据场景推荐使用并行文件系统，并行文件系统（Parallel File System）是对象存储服务（Object Storage Service, OBS）提供的一种经过优化的高性能文件系统，提供毫秒级别访问时延，以及TB/s级别带宽和百万级别的IOPS，能够快速处理高性能计算（HPC）工作负载。

并行文件的详细介绍和使用说明，请参见《[并行文件系统特性指南](#)》。

使用须知

- 请勿将该OBS桶用作其它用途，避免出现作业结果混乱等问题。
- OBS桶需要由用户主账户统一设置及修改，子用户无权限。
- 不配置DLI作业桶无法查看作业日志。
- 您可以通过配置桶的生命周期规则，定时删除桶中的对象或者定时转换对象的存储类别。
- DLI的作业桶设置后请谨慎修改，否则可能会造成历史数据无法查找。

操作步骤


1. 在DLI控制台左侧导航栏中单击“全局配置 > 工程配置”。
2. 在“工程配置”页面，选择“DLI作业桶”，单击配置桶信息。

图 2-5 工程配置




3. 单击  打开桶列表。
4. 选择用于存放DLI作业临时数据的桶，并单击“确定”。
完成设置后DLI作业运行过程中产生的临时数据将会存储在该OBS桶中。

图 2-6 设置 DLI 作业桶



3 创建弹性资源池和队列

3.1 DLI 弹性资源池与队列简介

DLI的计算资源是执行作业的基础，本节内容介绍DLI计算资源的模式和队列类型。

什么是弹性资源池和队列？

在了解DLI计算资源模式前首先了解弹性资源池和队列的基本概念。

- **弹性资源池**是DLI计算资源的一种池化管理模式，可以看做DLI计算资源的集合。DLI支持在弹性资源池中创建多个队列，且这些队列可以共享弹性资源池中的资源。
- **队列**是DLI中被实际使用和分配的基本单元，即队列是执行作业所需的具体的计算资源。您可以为不同的作业或数据处理任务创建不同的队列，并按需对这些队列分配和调整资源。了解DLI的队列类型请参考[DLI队列类型](#)。

📖 说明

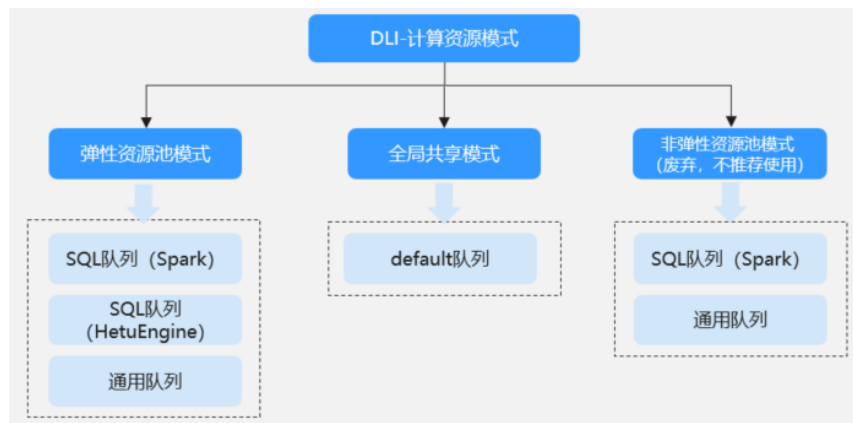
DLI弹性资源池之间为物理集群隔离，同一个弹性资源池中的队列之间为逻辑隔离。

建议您对测试业务场景和生产业务场景分别创建弹性资源池，通过资源物理隔离的方式，保障资源管理的独立性和安全性。

DLI 计算资源模式

DLI提供了三种计算资源的管理模式，每一种模式都有独特的优势和适用场景。

图 3-1 DLI 计算资源模式



- **弹性资源池模式：** 计算资源的池化管理模式，提供计算资源的动态扩缩容能力，同一弹性资源池中的队列共享计算资源。通过合理设置队列的计算资源分配策略，可以提高计算资源利用率，应对业务高峰期的资源需求。
 - 适用场景：适合业务量有明显波动的场合，如周期性的数据批处理任务或实时数据处理需求。
 - 支持的队列类型：SQL队列（Spark）、SQL队列（HetuEngine）、通用队列。了解DLI的队列类型请参考[DLI队列类型](#)。

📖 说明

弹性资源池模式的通用队列和SQL队列不支持跨可用区。

- 使用方法：先创建弹性资源池，然后在弹性资源池中创建队列并分配计算资源，队列关联到具体的作业和数据处理任务。
购买弹性资源池并在弹性资源池中添加队列的具体操作步骤请参考[创建弹性资源池并添加队列](#)。
- **全局共享模式：**
全局共享模式是一种根据SQL查询中实际扫描的数据量来分配计算资源的模式，不支持指定或预留计算资源。
DLI服务预置的“default”队列即为全局共享模式的计算资源，资源的大小是按需分配的。在不确定数据量大小或偶尔需要进行数据处理的用户，可以使用default队列执行作业。
 - 适用场景：适用于测试作业或资源消耗不高的情况。
 - 支持的队列类型：仅DLI预置的default队列为全局共享模式的计算资源。
“default”队列只用于用户体验DLI，是所有人共享的公共资源，使用时可能会出现用户间抢占资源的情况，不能保证每次都可以得到资源执行相关操作。建议使用自建队列执行生产作业。
 - 使用方法：default队列仅适用于提交SQL作业，在DLI管理控制台提交SQL作业时选择“default队列”即可。
- **非弹性资源池模式（废弃，不推荐使用）：**
DLI的上一代计算资源管理方式，因缺乏灵活性，目前已不推荐使用。
非弹性资源池模式提供固定规格的计算资源，购买后独占资源，无法根据需求动态调整，可能会导致资源浪费或在需求高峰期资源不足。

表 3-1 DLI 计算资源模式与支持的队列类型

DLI计算资源模式	支持的队列类型	资源特点	适用场景
弹性资源池模式	SQL队列 (Spark) SQL队列 (HetuEngine) 通用队列	单用户多队列共享资源 资源动态分配, 灵活调整	适合业务需求波动较大, 需要灵活调整资源以应对波峰波谷的业务场景。
全局共享模式	default队列	多用户多队列共享资源 按量付费, 不支持预留资源	适合不确定数据量大小或仅需要偶尔进行数据处理的临时或测试项目场景。
非弹性资源池模式 (废弃, 不推荐使用)	SQL队列 通用队列	单用户单队列独享资源 无法动态调整, 资源可能会闲置	废弃, 不推荐使用

为了方便您理解DLI不同计算资源模式的适用场景, 我们把购买DLI计算资源比作用车服务:

- 弹性资源池模式可以比作“租车”, 您可以根据实际需求动态调整资源的规模。这种模式适合于业务需求波动较大的场景, 灵活地根据业务峰谷来调整资源, 优化成本。
- 全局共享模式可以比作“打车”, 您只需为实际使用的数据量付费。这种模式适合于不确定数据量大小或仅需要偶尔进行数据处理的场景, 按需使用资源, 无需预先购买或预留资源。

弹性资源池扩缩容

当弹性资源池中添加队列、删除队列时, 会触发弹性资源扩缩容。

而弹性资源池缩容可能会触发缩容含有shuffle数据的节点, 会导致Spark Task重算, 引起Spark作业和SQL作业内部自动重试, 当作业重试超过限制会导致作业执行失败, 需用户重新执行作业。

📖 说明

- Spark2.3版本作业需要升级作业版本后才能支持运行中动态缩容功能。
- Spark Steaming作业、Flink作业在运行过程中所在节点无法缩容, 需要暂停作业或迁移作业至其他弹性资源池后才能完成缩容。

DLI 队列类型

DLI分为三种队列类型: default队列、SQL队列、通用队列, 您可以根据业务场景和作业特性选择最合适的队列类型。

- default队列:**
 DLI服务预置的队列，所有用户共享。
 不支持指定default队列资源大小，资源在执行作业时按需分配，并按实际扫描的数据量计费。
 由于default队列是共享资源，在使用时可能会出现资源抢占的情况，不能保证每次都能获得资源来执行作业。
 default队列适用小规模或临时的数据处理需求。对于重要的或需要保证资源的作业，建议购买弹性资源池并在弹性资源池中创建队列来执行作业。
- SQL队列:**
 SQL队列是用于执行SQL作业的队列，支持指定引擎类型包括Spark和HetuEngine。
 SQL队列适用于需要快速数据查询和分析，以及需要定期清理缓存或重置环境的业务。
- 通用队列:**
 通用队列用于执行Spark作业、Flink OpenSource SQL作业和Flink Jar作业的队列。
 适合适用于复杂数据处理、实时数据流处理或批量数据处理的场景。

弹性资源池使用场景

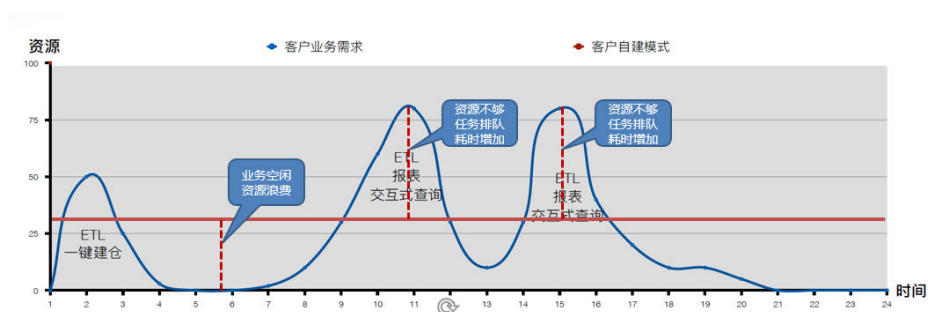
推荐使用弹性资源池队列，提高资源使用的灵活性和资源利用效率。本节介绍常见的弹性资源池的使用场景。

场景一：固定资源造成资源浪费和资源不足的场景

在每天的不同时段，作业任务对资源的请求量也会发生变化，如果采用固定资源规格则会导致资源浪费或者资源不足的问题。例如，如下图图3-2示例可以看出：

- 大约在凌晨4点到7点这个数据段，ETL作业任务结束后没有其他作业，因为资源固定一直占用，导致严重的资源浪费。
- 上午9点到12点以及下午14点16点的两个时段，ETL报表和作业查询的请求量很高，因为当前固定资源不够，导致作业任务排队，任务一直排队。

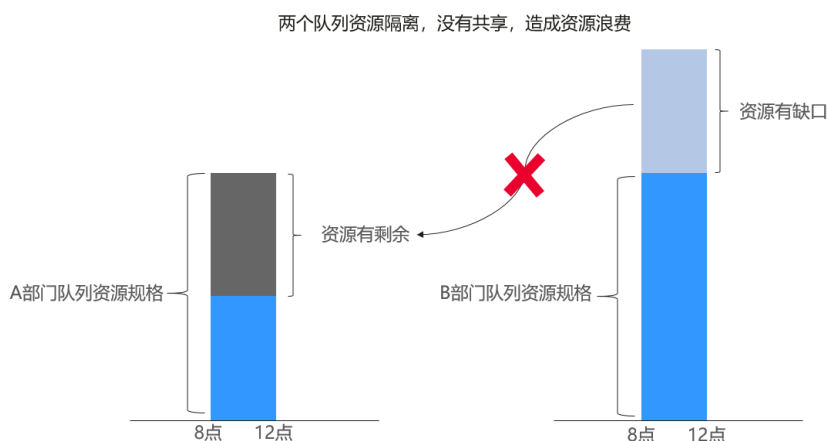
图 3-2 固定资源场景



场景二：资源相互隔离，没有共享，造成资源浪费的场景

某公司下有两个部门，两个部门的不同作业运行在DLI的两个队列上。部门A在上午8点到12点业务比较空闲，资源有剩余，部门B在这个时间段业务请求量大，原有资源规格满足不了，需要扩容时，请求不了部门A的队列资源，造成资源浪费。

图 3-3 资源隔离造成的资源浪费



弹性资源池通过“分时按需弹性”功能，支持按照不同时间段对资源进行动态的扩缩容，保证资源的利用率和应对资源洪峰等诉求。

弹性资源池对后端资源统一进行管理和调度，多个队列绑定弹性资源池后，资源池内资源共享，资源利用率高，解决了场景二的问题。

3.2 创建弹性资源池并添加队列

弹性资源池为DLI作业运行提供所需的计算资源（CPU和内存），用于灵活应对业务对计算资源变化的需求。

创建弹性资源池后，您可以在弹性资源池中创建多个队列，队列关联到具体的作业和数据处理任务，是资源池中资源被实际使用和分配的基本单元，即队列是执行作业所需的具体的计算资源。

同一弹性资源池中，队列之间的计算资源支持共享。通过合理设置队列的计算资源分配策略，可以提高计算资源利用率。本章节介绍创建弹性资源池并添加队列的操作步骤。

📖 说明

DLI弹性资源池之间为物理集群隔离，同一个弹性资源池中的队列之间为逻辑隔离。

建议您对测试业务场景和生产业务场景分别创建弹性资源池，通过资源物理隔离的方式，保障资源管理的独立性和安全性。

弹性资源池约束与限制

表 3-2 弹性资源池约束限制

限制项	说明
资源规格	<ul style="list-style-type: none"> 当前弹性资源池最大的计算资源 32000CUs。 弹性资源池中可创建队列的最小CU： <ul style="list-style-type: none"> 通用队列：4CUs SQL队列：Spark SQL队列：8CUs；HetuEngine SQL队列：16CUs
弹性资源池计费模式	<ul style="list-style-type: none"> 弹性资源池支持按需和包年包月的购买方式。 不支持切换弹性资源池的计费模式。 当前仅支持包年包月计费模式的弹性资源池进行规格变更。 按需计费的弹性资源池默认勾选专属资源模式，自创建起按自然小时收费。
管理弹性资源池	<ul style="list-style-type: none"> 弹性资源池不支持切换区域。 Flink 1.10及其以上版本的作业支持在弹性资源池运行。 弹性资源池网段设置后不支持更改。 仅支持查看30天以内的弹性资源池扩缩容历史。 弹性资源池无法直接访问公网。
弹性资源池关联队列	<ul style="list-style-type: none"> 弹性资源池关联队列： <ul style="list-style-type: none"> 仅支持关联按需计费模式的队列（包括专属队列）。 队列和弹性资源池状态正常，资源未被冻结。
弹性资源池扩缩容	<ul style="list-style-type: none"> 弹性资源池CU设置、弹性资源池中添加/删除队列、修改弹性资源池中队列的扩缩容策略、系统自动触发弹性资源池扩缩容时都会引起弹性资源池CU的变化，部分情况下系统无法保证按计划扩容/缩容至目标CUs： <ul style="list-style-type: none"> 弹性资源池扩容时，可能会由于物理资源不足导致弹性资源池无法扩容到设定的目标大小。 弹性资源池缩容时，系统不保证将队列资源完全缩容到设定的目标大小。 在执行缩容任务时，系统会先检查资源使用情况，判断是否存在缩容空间，如果现有资源无法按照最小缩容步长执行缩容任务，则弹性资源池可能缩容不成功，或缩容一部分规格的情况。 因资源规格不同可能有不同的缩容步长，通常是16CUs、32CUs、48CUs、64CUs等。 示例：弹性资源池规格为192CUs，资源池中的队列执行作业占用了68CUs，计划缩容至64CUs。 执行缩容任务时，系统判断剩余124CUs，按64CUs的缩容步长执行缩容任务，剩余60CUs资源无法继续缩容，因此弹性资源池执行缩容任务后规格为128CUs。

创建弹性资源池

步骤1 在左侧导航栏单击“资源管理 > 弹性资源池”，可进入弹性资源池管理页面。

步骤2 在弹性资源池管理界面，单击界面右上角的“购买弹性资源池”。

步骤3 在“购买弹性资源池”界面，填写具体的弹性资源池参数，具体参数填写参考如下。

表 3-3 参数说明

参数名称	描述
计费模式	<ul style="list-style-type: none">包年/包月：预付费模式，按订单的购买周期计费。拥有专属的计算资源，空闲（无作业运行）时不会释放，使用体验更佳，价格比按需计费模式更优惠。按需计费：后付费模式，默认勾选专属资源模式，空闲时资源不被释放。
区域	选择所在的区域。不同区域的云服务之间内网互不相通；请就近选择靠近您业务的区域，可减少网络时延，提高访问速度。
项目	每个区域默认对应一个项目，这个项目由系统预置。
名称	弹性资源池的具体名称。 <ul style="list-style-type: none">名称只能包含数字、英文字母和下划线，但不能是纯数字，且不能以下划线或数字开头。输入长度不能超过128个字符。 说明 弹性资源池名称不区分大小写，系统会自动转换为小写。
类型	<ul style="list-style-type: none">基础版：提供16-64CUs规格的资源<ul style="list-style-type: none">适用于对资源消耗不高、对资源高可靠性和高可用性要求不高的测试场景。不支持高可靠与高可用。不支持设置队列属性和作业优先级。不支持对接Notebook实例。标准版：提供64CUs及以上规格的资源 具备强大的计算能力、高可用性、及灵活的资源管理能力，适用于大规模计算任务场景和有长期资源规划需求的业务场景。

参数名称	描述
CU范围	<p>弹性资源池最大最小CU范围。</p> <p>CU设置主要是为了控制弹性资源池扩缩容的最大最小CU范围，避免无限制的资源扩容风险。</p> <p>“CU范围”参数中，左边为最小CU，右边为最大CU，根据情况分别设置。</p> <ul style="list-style-type: none"> 弹性资源池中所有队列的最小CU数之和需要小于等于弹性资源池的最小CU数。 弹性资源池中任意一个队列的最大CU必须小于等于弹性资源池的最大CU。 <p>弹性资源池至少可以满足弹性资源池中所有队列按最小CU运行，尽量满足队列按最大CU运行。</p> <p>弹性资源池的规格（“规格”即“包周期CU”）等于创建时的最小CU，是首次创建时分配的资源数。即弹性资源池首次创建时，实际CU_s=规格=最小CU。</p>
描述	创建的弹性资源池的描述信息。
网段	<p>规划弹性资源池所属的网段。如需使用DLI增强型跨源，弹性资源池网段与数据源网段不能重合。弹性资源池网段设置后不支持更改。</p> <p>建议使用网段：</p> <p>10.0.0.0~10.255.0.0/16~19</p> <p>172.16.0.0~172.31.0.0/16~19</p> <p>192.168.0.0~192.168.0.0/16~19</p>
企业项目	<p>如果所建弹性资源池属于企业项目，可选择对应的企业项目。</p> <p>企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理。</p> <p>关于如何设置企业项目请参考《企业管理用户指南》。</p> <p>说明</p> <p>只有开通了企业管理服务的用户才显示该参数。</p>
购买时长	<p>选择“包年/包月”计费模式时，需要选择“购买时长”。购买时长越长，优惠越多。可勾选“自动续费”，按月购买，自动续费周期为1个月。按年购买，自动续费周期为1年。</p>

参数名称	描述
标签	<p>使用标签标识云资源。包括标签键和标签值。如果您需要使用同一标签标识多种云资源，即所有服务均可在标签输入框下拉选择同一标签，建议在标签管理服务（TMS）中创建预定义标签。</p> <p>如您的组织已经设定DLI的相关标签策略，则需按照标签策略规则为资源添加标签。标签不符合标签策略的规则，则可能会导致资源创建失败，请联系组织管理员了解标签策略详情。</p> <p>具体请参考《标签管理服务用户指南》。</p> <p>说明</p> <ul style="list-style-type: none"> • 最多支持20个标签。 • 一个“键”只能添加一个“值”。 • 每个资源中的键名不能重复。 • 标签键：在输入框中输入标签键名称。 <p>说明 标签的键的最大长度为128个字符，标签的键可以包含任意语种字母、数字、空格和_ . : + - @ ,但首尾不能含有空格，不能以_sys_开头。</p> <ul style="list-style-type: none"> • 标签值：在输入框中输入标签值。 <p>说明 标签值的最大长度为255个字符，标签的值可以包含任意语种字母、数字、空格和_ . : + - @ 。</p>

步骤4 参数填写完成后，单击“立即购买”，在界面上确认当前配置是否正确。

步骤5 单击“提交”完成创建。等待弹性资源池状态变成“可使用”表示当前弹性资源池创建成功。

步骤6 弹性资源池创建成功后，可以根据当前业务场景参考[典型场景示例：创建弹性资源池并运行作业](#)和[典型场景示例：配置弹性资源池队列扩缩容策略](#)完成后续操作。

----结束

在弹性资源池中添加队列

创建完弹性资源池后，弹性资源池需要添加一个或多个队列用于后续作业的运行。本节操作介绍在弹性资源池中添加队列的操作步骤。

添加到弹性资源池中的队列不再单独计费，以弹性资源池为计费项计费。

当弹性资源池添加队列时会引起弹性资源CUs扩缩容变化。

步骤1 在左侧导航栏单击“弹性资源池”，可进入弹性资源池管理页面。

步骤2 选择要操作的弹性资源池，在“操作”列，单击“添加队列”。

步骤3 在“添加队列”界面，首先需要配置队列的基础配置，具体参数信息如下。

表 3-4 弹性资源池添加队列基础配置

参数名	参数描述
名称	弹性资源池添加的队列名称。

参数名	参数描述
类型	<ul style="list-style-type: none"> SQL队列：用于运行SQL作业。 通用队列：用于运行Spark作业、Flink 作业。
执行引擎	如果队列类型选择为“SQL队列”，则可以选择队列引擎是：Spark或者HetuEngine HetuEngine类型的SQL队列最小CU不能小于96CUs。
企业项目	选择队列的企业项目。弹性资源池支持添加不同企业项目的队列资源。 企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理。 关于如何设置企业项目请参考《 企业管理用户指南 》。 说明 只有开通了企业管理服务的用户才显示该参数。
描述	弹性资源池添加队列的描述信息。
标签	使用标签标识云资源。包括标签键和标签值。如果您需要使用同一标签标识多种云资源，即所有服务均可在标签输入框下拉选择同一标签，建议在标签管理服务（TMS）中创建预定义标签。 如您的组织已经设定DLI的相关标签策略，则需按照标签策略规则为资源添加标签。标签不符合标签策略的规则，则可能会导致资源创建失败，请联系组织管理员了解标签策略详情。 具体请参考《 标签管理服务用户指南 》。 说明 <ul style="list-style-type: none"> 最多支持20个标签。 一个“键”只能添加一个“值”。 每个资源中的键名不能重复。 标签键：在输入框中输入标签键名称。 说明 标签的键的最大长度为128个字符，标签的键可以包含任意语种字母、数字、空格和_ . : =+-@ ,但首尾不能含有空格，不能以_sys_开头。 标签值：在输入框中输入标签值。 说明 标签值的最大长度为255个字符，标签的值可以包含任意语种字母、数字、空格和_ . : =+-@ ,但首尾不能含有空格。

步骤4 单击“下一步”，在“扩缩容策略”界面配置当前队列在弹性资源池的扩缩容策略。

图 3-4 添加队列时配置扩缩容策略



单击“新增”，可以添加不同优先级、时间段、“最小CU”和“最大CU”扩缩容策略。每条扩缩容策略的参数说明如下：

表 3-5 扩缩容策略参数说明

参数名	参数描述
优先级	当前弹性资源池中的优先级数字越大表示优先级越高。当前优先级支持的范围为：1到100。
时间段	时间段设置仅支持整点，左侧为开始时间，右侧为结束时间。请注意以下说明： <ul style="list-style-type: none"> 时间区间包括开始时间，不包括结束时间即[开始时间, 结束时间)。例如当前选择的时间段范围为：01--17，则表示当前扩缩容规则生效时间范围为[01,17)。 同一队列不同优先级的时间段区间不能有交集。
最小CU	当前扩缩容策略支持的最小CU数。 <ul style="list-style-type: none"> 在全天的任意一个时间段内，弹性资源池中所有队列的最小CU数之和必须小于等于弹性资源池的最小CU数。 当队列的最小CUs小于16CUs时，在队列属性中设置的“最大spark driver实例数”和“最大预拉起spark driver实例数”不生效。了解队列属性设置。 HetuEngine类型的SQL队列最小CU不能小于96CUs。
最大CU	当前扩缩容策略支持的最大CU数。 在全天的任意一个时间段内，弹性资源池中任意一个队列的最大CU必须小于等于弹性资源池的最大CU。

说明

- 首条扩缩容策略是默认策略，不能删除和修改时间段配置。
- Flink作业不支持触发弹性资源池队列的自动扩缩容。

步骤5 单击“确定”完成添加队列配置。弹性资源池队列添加完成后，可以参考[调整弹性资源池中队列的扩缩容策略](#)查看弹性资源池添加的所有队列配置和策略信息。

----结束

3.3 管理弹性资源池

3.3.1 查看弹性资源池的基本信息

资源池创建完成后您可以通过管理控制台查看和管理您的弹性资源池。

本节操作介绍在管理控制台如何查看弹性资源池基本信息，包括弹性资源池的VPC网段、IPv6网段、创建时间等信息。

查看弹性资源池的基本信息



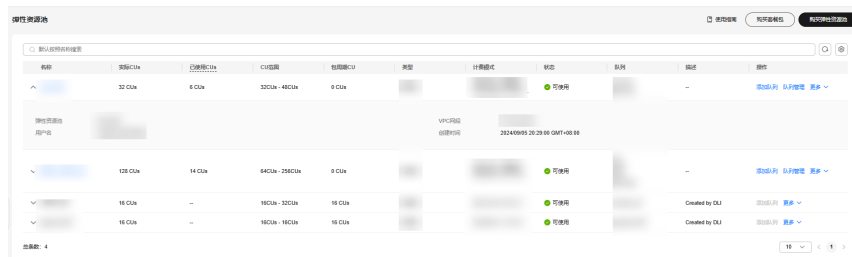
1. 登录DLI管理控制台。
2. 选择“资源管理 > 弹性资源池”。
3. 进入弹性资源池列表页面，选择您需要查看的弹性资源池。
 - 在列表页面的右上方单击可以自定义显示列，并设置表格内容显示规则、操作列显示规则。
 - 在列表页面上方的搜索区域，您可以名称和标签筛选需要的弹性资源池。
4. 单击展开弹性资源池基本信息卡片，查看弹性资源池详细信息。
支持查看以下信息：弹性资源池名称、弹性资源池创建用户、创建时间、弹性资源池VPC网段、弹性资源池是否启用IPv6，如果开启IPv6将显示具体的子网的IPv6网段。
关于弹性资源池的实际CUs、已使用CUs、CU范围、规格（包周期CU）的含义请参考[弹性资源池的实际CUs、已使用CUs、CU范围、规格（包周期CU）](#)。

图 3-5 弹性资源池基本信息



弹性资源池的实际 CUs、已使用 CUs、CU 范围、规格（包周期 CU）

- 实际CUs：弹性资源池当前分配的可用CUs。
 - 实际CUs的计算公式：
 - $\text{实际CUs} = \min\{\text{sum}(\text{队列maxCU}), \text{弹性资源池maxCU}\}$ 。
 - 计算结果需满足为16CUs的倍数，如果不能整除16CUs则向上取整。
 - 实际CUs的分配示例：

如表3-6所示，弹性资源池实际CUs分配的计算过程如下：

 - i. 计算队列maxCU之和： $\text{sum}(\text{队列maxCU}) = 32 + 56 = 88\text{CUs}$ 。
 - ii. 比较队列maxCU之和与弹性资源池maxCU，两者取最小值： $\min\{88\text{CUs}, 112\text{CUs}\} = 88\text{CUs}$ 。

- iii. 检查88CUs是否为16CU的倍数，由于88不能被16整除，故向上取整为96CUs。

表 3-6 弹性资源池实际 CUs 分配示例

场景说明	资源类型	CU范围
新建弹性资源池 64-112CUs 添加了两个队列，分别为 队列A和队列B。两个队列 设置的CU范围如下： <ul style="list-style-type: none"> 队列A的CU范围： 16-32CUs 队列B的CU范围： 16-56CUs 	弹性资源池	64-112CUs
	队列A	16-32CUS
	队列B	16-56CUS

- **已使用CUs:** 已经被作业或任务占用的CU资源。这些资源可能正在执行计算任务，暂时不可用。

📖 说明

HetuEngine已使用CUs和实际CU一致。

- **CU范围:** CU设置主要是为了控制弹性资源池扩缩容的最大最小CU范围，避免无限制的资源扩容风险。
 - 弹性资源池中所有队列的最小CU数之和需要小于等于弹性资源池的最小CU数。
 - 弹性资源池中任意一个队列的最大CU必须小于等于弹性资源池的最大CU。
 - 弹性资源池至少可以满足弹性资源池中所有队列按最小CU运行，尽量满足队列按最大CU运行。
- **规格（包周期CU）:** 购买弹性资源池时选择的CU范围的最小值即弹性资源池规格。规格是包周期弹性资源池特有的。规格部分以包周期的计费，规格之外的部分按需计费。

3.3.2 弹性资源池权限管理

针对不同用户，管理员可以通过权限设置赋予各用户不同的操作权限，控制各用户弹性资源池的操作范围。

注意事项

- 管理员用户和弹性资源池的所有者拥有所有权限，不需要进行权限设置且其他用户无法修改其队列权限。
- 给新用户设置弹性资源池权限时，该用户所在用户组的所属区域需具有Tenant Guest权限。关于Tenant Guest权限的介绍和开通方法，详细参见《[权限策略](#)》和《[统一身份认证服务用户指南](#)》中的[创建用户组](#)。

弹性资源池权限管理操作步骤

步骤1 在DLI管理控制台的左侧，选择“资源管理 > 弹性资源池”。

步骤2 选择待设置的弹性资源池，单击其“操作”列中的“更多 > 权限管理”。“用户权限信息”区域展示了当前具备此弹性资源池权限的用户列表。

权限设置有3种场景：为新用户赋予权限，为已有权限的用户修改权限，回收某用户具备的所有权限。

- 为新用户赋予权限

新用户指之前不具备此弹性资源池权限的用户。

- a. 单击“用户权限信息”右侧的“授权”，弹出“授权”对话框。
- b. “用户名”参数处填写具体要被授权的IAM用户名，并勾选需要赋权给该用户的对应权限。
- c. 单击“确定”，完成新用户的权限的设置。

待设置的参数说明如表3-7所示。

图 3-6 弹性资源池权限授权



表 3-7 参数说明

参数名称	描述
用户名	被授权的用户名称。 说明 该用户名称是已存在的IAM用户名称且该用户登录过DLI管理控制台。

参数名称	描述
权限设置	<ul style="list-style-type: none"> ▪ 更新：当前用户可更新弹性资源池的描述信息。 ▪ 资源管理：当前用户可在弹性资源池上添加队列、删除队列、操作队列的扩缩容策略配置。 ▪ 删除：当前用户可删除此弹性资源池。 ▪ 规格变更：当前用户对于包年包月的弹性资源池可以执行规格变更操作。 ▪ 赋权：当前用户可将弹性资源池的操作权限赋予其他用户。 ▪ 回收：当前用户可回收其他用户具备的该弹性资源池的权限，但不能回收该弹性资源池所有者的权限。 ▪ 查看其他用户具备的权限：当前用户可查看其他用户具备的该弹性资源池的权限。

- 为已有权限的用户赋予权限或回收权限。
 - a. 在对应弹性资源池“权限信息”区域的用户列表中，选择需要修改权限的用户，在“操作”列单击“权限设置”。
 - b. 在队列“权限设置”对话框中，对当前用户具备的权限进行修改。详细权限描述如表3-7所示。

当“权限设置”中的选项为灰色时，表示您不具备修改此队列权限的权限。可以向管理员用户、弹性资源池所有者等具有赋权权限的用户申请弹性资源池的“赋权”和“回收”权限。

图 3-7 弹性资源池权限设置



- c. 单击“确定”完成权限设置。
- 回收某用户具备的所有权限。
在对应弹性资源池“用户权限信息”区域的用户列表中，选择需要删除权限的用户，在“操作”列单击“回收”。在“回收”对话框中单击“确定”后，此用户将不具备该弹性资源池的任意权限。

----结束

3.3.3 弹性资源池关联队列

操作场景

参考[创建弹性资源池并添加队列](#)创建完弹性资源池后，您可以将已有的队列关联至弹性资源池，即可将弹性资源池的资源用于后续作业的运行

您可以在弹性资源池页面通过“关联队列”将队列添加到弹性资源池。还可以在队列管理页面分配队列至弹性资源池。

说明

弹性资源池Flink版本只支持1.10及其以上版本，如果准备分配到弹性资源池的作业使用Flink1.7版本可能会出现兼容性问题，需要提前做好Flink版本适配。

约束与限制

- 弹性资源池和队列均是可用状态。
- 队列是按需专属队列。
- 队列和弹性资源池状态正常，资源未被冻结。
- 弹性资源池仅支持关联同一企业项目的队列资源。

在弹性资源池页面关联队列

步骤1 在左侧导航栏单击“资源管理 > 弹性资源池”，可进入弹性资源池管理页面。

步骤2 选择要操作的弹性资源池，在“操作”列，单击“更多 > 关联队列”。

步骤3 在“关联队列”界面，选择待添加的队列，单击“确定”完成操作。

----结束

在队列管理页面分配队列至弹性资源池

步骤1 在左侧导航栏单击“资源管理 > 队列管理”，可进入队列管理页面。

步骤2 选择要操作的队列，在“操作”列，单击“更多 > 分配至弹性资源池”。

步骤3 选择资源池，单击“确定”完成操作。

----结束

3.3.4 弹性资源池 CU 设置

CU设置主要是为了控制弹性资源池扩缩容的最大最小CU范围，避免无限制的资源扩容风险。

例如，当前弹性资源池CU设置的最大CU为256CU，并且该弹性资源池添加了2个队列，2个队列扩缩容策略最小CU数为64CU，这时如果该弹性资源池再添加一个队列并且该队列最小CU为256CU时，因为受到CU最大设置的控制，该队列不能添加到该弹性资源池。

注意事项

- 在全天的任意一个时间段内，弹性资源池中所有队列的最小CU数之和需要小于等于弹性资源池的最小CU数。

- 在全天的任意一个时间段内，弹性资源池中任意一个队列的最大CU必须小于等于弹性资源池的最大CU。
- 弹性资源池创建后，调整最小CU时，最小CU需小于等于弹性资源池当前的CU值，否则会修改失败。

CU 设置操作

步骤1 在左侧导航栏单击“资源管理 > 弹性资源池”，可进入弹性资源池管理页面。

步骤2 选择要操作的弹性资源池，在“操作”列，单击“更多 > CU设置”。

步骤3 在“CU设置”界面，“CU范围”参数中，左边为最小CU，右边为最大CU，根据情况分别设置。单击“确定”完成设置操作。

----结束

CU 设置时怎样调大 CU 范围最小值？

弹性资源池CU范围的最小值小于等于弹性资源池的实际CUs，如果想要调整的CU范围的最小值大于当前CUs，需要先调大实际CUs。

以下操作适用于目标CU小于等于CU范围的最大值。目标CU大于CU范围最大值时，请先调大弹性资源池的最大CU。

- 包年包月计费模式的弹性资源池，通过调整当前弹性资源池中队列的最大CU或增加队列，以增大弹性资源池的实际CUs，再通过变更弹性资源池规格，将规格调整至大于等于目标CUs，此时实际CUs等于规格，再通过CU设置，调整CU范围的最小值等于目标CUs。
- 按需计费模式的弹性资源池，通过调整当前弹性资源池中队列的最大CU或增加队列，以增大弹性资源池的实际CUs，再通过CU设置，调整CU范围的最小值等于目标CUs。

示例

- 示例1：包年包月弹性资源池，实际CUs：64CUs、CU范围：64CUs - 96CUs、规格64。计划调整CU范围：80CUs - 96CUs。

操作步骤：

- a. 通过调整当前弹性资源池中队列的最大CU或增加队列，以增大弹性资源池的实际CUs。

当弹性资源池队列的最大CU的和大于弹性资源池的实际CUs时，会触发弹性资源池扩容，扩容后的实际CUs= min（队列的最大CU和，弹性资源池CU范围最大值）。

- b. 通过“规格变更”将弹性资源池规格调整为80CUs，规格变更后，CU范围的最小值自动调整为与规格一致大小的CU。

- 示例2：按需计费的弹性资源池，实际CUs：64CUs、CU范围：64CUs - 96CUs。计划调整CU范围：80CUs - 96CUs。

操作步骤：

- a. 通过调整当前弹性资源池中队列的最大CU或增加队列，以增大弹性资源池的实际CUs。

当弹性资源池队列的最大CU的和大于弹性资源池的实际CUs时，会触发弹性资源池扩容，扩容后的实际CUs= min（队列的最大CU和，弹性资源池CU范围最大值）。

- b. 通过“CU设置”调整CU范围：80CUs - 96CUs。

说明

- 调整队列的CU范围、弹性资源池规格变更、弹性资源池的CU设置，均在下一个整点生效。
- 通过增加队列调整弹性资源池的实际CUs，立即生效。

CU 设置时怎样调小 CU 范围最大值？

弹性资源池CU范围的最小值小于等于弹性资源池的实际CUs，如果想要调整的CU范围的最大值小于当前CUs，需要先调小实际CUs。

- 包年包月计费模式的弹性资源池，通过调小当前弹性资源池中队列的最大CU或删除队列，以调小弹性资源池的实际CUs，再通过变更弹性资源池规格，将规格调整至小于等于目标CUs，此时实际CUs等于规格，再通过CU设置，调整CU范围的最小值等于目标CUs。
- 按需计费模式的弹性资源池，通过调整当前弹性资源池中队列的最大CU或删除队列，以调小弹性资源池的实际CUs，再通过CU设置，调整CU范围的最小值等于目标CUs。

示例

- 示例1：包年包月弹性资源池，实际CUs：96CUs、CU范围：64CUs - 128CUs、规格96。计划调整CU范围：64CUs - 80CUs。

操作步骤：

- a. 通过调小当前弹性资源池中队列的最大CU或删除队列，以调小弹性资源池的实际CUs。

当弹性资源池队列的最大CU的和小于弹性资源池的实际CUs时，会触发弹性资源池扩容，扩容后的实际CUs= min（队列的最大CU和，弹性资源池CU范围最大值）。

- b. 通过“规格变更”将弹性资源池规格调整为80CUs，规格变更后，CU范围的最小值自动调整为与规格一致大小的CU。

- 示例2：按需计费的弹性资源池，实际CUs：96CUs、CU范围：64CUs - 128CUs、计划调整CU范围：64CUs - 80CUs。

操作步骤：

- a. 通过调小当前弹性资源池中队列的最大CU或删除队列，以调小弹性资源池的实际CUs。

当弹性资源池队列的最大CU的和小于弹性资源池的实际CUs时，会触发弹性资源池扩容，扩容后的实际CUs= min（队列的最大CU和，弹性资源池CU范围最大值）。

- b. 通过“CU设置”调整CU范围：64CUs - 80CUs。

说明

- 调整队列的CU范围、弹性资源池规格变更、弹性资源池的CU设置，均在下一个整点生效。
- 通过增加队列调整弹性资源池的实际CUs，立即生效。

3.3.5 弹性资源池规格变更

使用场景

包年包月的弹性资源池CU数在规格（包周期CU）的范围内使用包年包月计费，超过规格（包周期CU）的部分则按弹性资源池CU时计费的方式计费，您可以根据实际CU的使用情况通过规格变更来使得计费更优惠。

例如，当前弹性资源池的规格（包周期CU）为64CU，实际使用过程中大部分时间CU数在128CU以上，没有规格变更的场景下64CU部分采用包年包月计费，超出的64CU按弹性资源池CU时计费方式计费。为了满足该场景下更优惠的计费，则可以通过规格变更的方式，将弹性资源池的规格扩大到128CU，则规格变更成功后128CU范围内都使用包年包月计费，整体相比原来更优惠。

弹性资源池规格不满足当前业务需求时，可以通过“包周期CU变更”功能修改当前弹性资源池的规格。

注意事项

当前仅支持包年包月计费模式的弹性资源池进行规格（包周期CU）变更。

弹性资源池扩容

1. 在DLI管理控制台左侧，选择“资源管理 > 弹性资源池”。
2. 选择需要扩容的弹性资源池，单击“操作”列“更多”中的“包周期CU变更”。
3. 在“包周期CU变更”页面，“变更方式”选择“扩容”，变更数量选择要扩容的CU数量。

图 3-8 规格变更扩容

< | 包周期CU变更

名称

计费模式 包年/包月

过期时间 2026/12/03 23:59:59 GMT+08:00

当前规格 64 CUs

变更方式 扩容 缩容

变更数量 - 16 +

最终规格 80 CUs

4. 确定费用后，单击“提交”。
5. 扩容任务提交后，可以选择“作业管理 > SQL作业”，查看“SCALE_POOL”类型SQL作业的状态。

如果作业状态为“规格变更中”，表示弹性资源池规格正在扩容中。等待作业状态变为“已成功”表示当前变更操作完成。

弹性资源池缩容

📖 说明

系统默认最小CU值为16CU，即当弹性资源池的规格为16CU时，不能进行手动缩容。

1. 在DLI管理控制台左侧，选择“资源管理 > 弹性资源池”。
2. 选择需要缩容的弹性资源池，单击“操作”列“更多”中的“包周期CU变更”。
3. 在“包周期CU变更”页面，“变更方式”选择“缩容”，变更数量选择要缩容的CU数量。

图 3-9 弹性资源池规格缩容



4. 确定费用后，单击“提交”。
5. 缩容任务提交后，可以选择“作业管理 > SQL作业”，查看“SCALE_POOL”类型SQL作业的状态。

如果作业状态为“规格变更中”，表示弹性资源池规格正在缩容中。等待作业状态变为“已成功”表示当前变更操作完成。

3.3.6 弹性资源池标签管理

标签管理

标签是用户自定义的、用于标识云资源的键值对，它可以帮助用户对云资源进行分类和搜索。标签由标签“键”和标签“值”组成。

如果用户在其他云服务中使用了标签，建议用户为同一个业务所使用的云资源创建相同的标签键值对以保持一致性。

DLI支持以下两类标签：

- 资源标签：在DLI中创建的非全局的标签。
- 预定义标签：在标签管理服务（简称TMS）中创建的预定义标签，属于全局标签。

有关预定义标签的更多信息，请参见《[标签管理服务用户指南](#)》。

如果您的组织已经设定DLI的相关标签策略，则需按照标签策略规则为资源添加标签。标签不符合标签策略的规则，则可能会导致资源创建失败，请联系组织管理员了解标签策略详情。

以下介绍如何为队列添加标签、修改标签和删除标签。

- 步骤1** 在DLI管理控制台的左侧导航栏中，单击“资源管理 > 弹性资源池”。
- 步骤2** 在对应队列的“操作”列，选择“更多> 标签”。
- 步骤3** 进入标签管理页面，显示当前队列的标签信息。
- 步骤4** 单击“添加/编辑标签”，弹出“添加/编辑标签”对话框，配置参数。配置完成一个标签，单击“添加”将标签添加到输入框中。

图 3-10 添加/编辑标签



表 3-8 标签配置参数

参数	参数说明
标签键	<p>您可以选择：</p> <ul style="list-style-type: none"> ● 在输入框的下拉列表中选择预定义标签键。 如果添加预定义标签，用户需要预先在标签管理服务中创建好预定义标签，然后在“标签键”的下拉框中进行选择。用户可以通过单击“查看预定义标签”进入标签管理服务的“预定义标签”页面，然后单击“创建标签”来创建新的预定义标签。 具体请参见《标签管理服务用户指南》中的“创建预定义标签”章节。 ● 在输入框中输入标签键名称。 <p>说明 标签的键的最大长度为128个字符，标签的键可以包含任意语种字母、数字、空格和_ : +-@ ,但首尾不能含有空格，不能以_sys_开头。</p>

参数	参数说明
标签值	<p>您可以选择：</p> <ul style="list-style-type: none"> 在输入框的下拉列表中选择预定义标签值。 在输入框中输入标签值。 <p>说明 标签值的最大长度为255个字符，标签的值可以包含任意语种字母、数字、空格和_.:+@。</p>

📖 说明

- 最多支持20个标签。
- 一个“键”只能添加一个“值”。
- 每个资源中的键名不能重复。

步骤5 单击“确定”。

步骤6 （可选）在标签列表中，单击“操作”列中“删除”可对选中的标签进行删除。

----结束

3.3.7 调整弹性资源池中队列的扩缩容策略

弹性资源池上可以添加多个不同队列用于作业运行，具体添加弹性资源池添加队列的操作可以参考[创建弹性资源池并添加队列](#)。添加完队列后，可以根据不同队列计算资源使用量的波峰和波谷和优先级来配置要扩缩容的CU数，从而来保障作业的稳定运行。

注意事项

- 建议对流批业务实施资源池的精细化管理，将Flink实时流类型的作业与SQL批处理类型的作业分别置于独立的弹性资源池中。
优势在于：Flink实时流任务具有常驻运行的特质，确保其稳定运行而不会强制缩容，进而避免任务中断和系统不稳定。
而SQL批处理类型的作业在独立的资源池中能够更加灵活地进行扩缩容，显著提升扩缩容的成功率和操作效率。
- 在全天的任意一个时间段内，弹性资源池中所有队列的最小CU数之和必须小于等于弹性资源池的最小CU数。
- 在全天的任意一个时间段内，弹性资源池中任意一个队列的最大CU必须小于等于弹性资源池的最大CU。
- 同一队列不同扩缩容策略的时间段区间不能有交集。
- 弹性资源池队列中的扩缩容策略时间段仅支持整点的时间段设置，并且包含设置的开启时间，不包含设置的结束时间，例如设置时间段00-09，则时间段范围为：[00:00, 09:00)。默认的扩缩容策略不支持时间段配置修改。
- 弹性资源池扩缩容策略生效规则为：在任意一个时间段周期内，优先满足所有队列的最小CU数，剩余的CU（弹性资源池最大CU-所有队列的最小CU数之和）则根据配置的优先级顺序分配，直到剩余的CU数分配完成。

- 队列扩容成功后，系统开始对扩容的CU进行计费，直到缩容成功停止对扩容的CU计费。因此，要注意如果业务没有需求的情况下，要及时清理释放资源，否则不管CU是否真正的使用，都会一直计费。

表 3-9 弹性资源池扩缩容 CU 分配场景说明（无任务场景）

场景	弹性资源池CU数分配说明
<p>弹性资源池当前最大CU为256CU，添加了两个队列，分别为队列A和队列B。两个队列设置的扩缩容策略如下：</p> <ul style="list-style-type: none"> 队列A扩缩容策略：优先级5，时间段：00:00-9:00，最小CU是32，最大CU是64 队列B扩缩容策略：优先级10，时间段：00:00-9:00，最小CU是64，最大CU是128 	<p>到了00:00-9:00时间段：</p> <ol style="list-style-type: none"> 弹性资源池优先满足两个队列的最小CU，队列A先分配32CU，队列B分配64CU，剩余CU数为160CU：弹性资源池的最大CU-两个队列的最小CU之和=256-32-64=160CU。 剩余CU数根据优先级高低来分配，因为队列B的优先级高于队列A，则优先将64CU分配给队列B，再分配32CU给队列A。
<p>弹性资源池当前最大CU为96CU，添加了两个队列，分别为队列A和队列B。两个队列设置的扩缩容策略如下：</p> <ul style="list-style-type: none"> 队列A扩缩容策略：优先级5，时间段：00:00-9:00，最小CU是32，最大CU是64 队列B扩缩容策略：优先级10，时间段：00:00-9:00，最小CU是64，最大CU是128 	<p>到了00:00-9:00时间段：</p> <ol style="list-style-type: none"> 弹性资源池优先满足两个队列的最小CU，队列A先分配32CU，队列B分配64CU，剩余CU数为0CU：弹性资源池的最大CU-两个队列的最小CU之和=96-32-64=0CU。 因为剩余的CU数已经没有，则停止分配。
<p>弹性资源池当前最大CU为128CU，添加了两个队列，分别为队列A和队列B。两个队列设置的扩缩容策略如下：</p> <ul style="list-style-type: none"> 队列A扩缩容策略：优先级5，时间段：00:00-9:00，最小CU是32，最大CU是64 队列B扩缩容策略：优先级10，时间段：00:00-9:00，最小CU是64，最大CU是128 	<p>到了00:00-9:00时间段：</p> <ol style="list-style-type: none"> 弹性资源池优先满足两个队列的最小CU，队列A先分配32CU，队列B分配64CU，剩余CU数为32CU：弹性资源池的最大CU-两个队列的最小CU之和=128-32-64=32CU。 按照优先级，则优先将剩余的32CU分配给B队列后停止分配。
<p>弹性资源池当前最大CU为128CU，添加了两个队列，分别为队列A和队列B。两个队列设置的扩缩容策略如下：</p> <ul style="list-style-type: none"> 队列A扩缩容策略：优先级5，时间段：00:00-9:00，最小CU是32，最大CU是64 队列B扩缩容策略：优先级5，时间段：00:00-9:00，最小CU是64，最大CU是128 	<p>到了00:00-9:00时间段：</p> <ol style="list-style-type: none"> 弹性资源池优先满足两个队列的最小CU，队列A先分配32CU，队列B分配64CU，剩余CU数为32CU：弹性资源池的最大CU-两个队列的最小CU之和=128-32-64=32CU。 因为两个队列的优先级相同，则剩余32CU随机分配给两个队列。

表 3-10 弹性资源池扩缩容 CU 分配场景说明（有任务场景）

场景	弹性资源池实际CUs	队列A资源分配	队列B资源分配	弹性资源池CU数分配说明
弹性资源池添加了两个队列，分别为队列A和队列B。两个队列设置的扩缩容策略如下： <ul style="list-style-type: none"> 队列A扩缩容策略：时间段：00:00-9:00，最小CU是32，最大CU是64 队列B扩缩容策略：00:00-9:00，最小CU是64，最大CU是128 	192CUs	64CUs	128CUs	当弹性资源池实际cu大于等于两个队列最大cu之和，队列都分配最大值
	96CUs	32CUs	64CUs	弹性资源池会优先满足两个队列的最小CU，两个队列分配了最小CU后，无可用资源进行分配
	128CUs	32CUs-64CUs	64CUs-96CUs	弹性资源池会优先满足两个队列的最小CU，即队列A先分配32CUs，队列B分配64CUs，有剩余32CUs可供分配。 剩余部分按照队列的负载以及队列优先级进行分配。队列实际CU会在列出的范围内变化。

弹性资源池队列管理

步骤1 在左侧导航栏单击“资源管理 > 弹性资源池”，可进入弹性资源池管理页面。

步骤2 选择要操作的弹性资源池，在“操作”列，单击“队列管理”，进入弹性资源池队列管理界面。

步骤3 在队列管理界面会显示添加的所有队列列表信息。具体参数说明如下：

表 3-11 弹性资源池队列管理界面参数说明

参数名	参数描述
名称	弹性资源池添加的队列名称。
类型	弹性资源池添加的队列类型。 <ul style="list-style-type: none"> SQL队列。 通用队列。
时间段	弹性资源池队列扩缩容策略的开始和结束时间范围。时间区间包括开始时间，不包括结束时间即[开始时间, 结束时间)
最小CUs	弹性资源池队列扩缩容策略配置的最小CU数。

参数名	参数描述
最大CUs	弹性资源池队列扩缩容策略配置的最大CU数。
优先级	弹性资源池队列扩缩容策略的优先级。优先级范围为1到100，数字越小，优先级越低。
执行引擎	添加的队列类型为“SQL队列”时执行引擎为spark。 添加的队列类型为“通用队列”时执行引擎可以是spark和flink，当前界面显示为--。
创建时间	弹性资源池添加队列的时间。
企业项目	队列所属的企业项目。 弹性资源池支持添加不同企业项目的队列资源。
所有者	弹性资源池添加队列的用户名。
操作	<ul style="list-style-type: none"> 编辑：重新修改或者添加弹性资源池队列的扩缩容策略。 删除：删除当前弹性资源池添加的队列。

图 3-11 弹性资源池队列管理

名称	类型	时间段	最小CUs	最大CUs	优先级	执行引擎	创建时间	企业项目	所有者	操作
demo_queue_1	通用队列	(00:00, 17:00) (00:00, 24:00)	16	64	1	--	2022/05/18 16:15:37 GMT+08:00	default		编辑 删除
flink	SQL队列	(00:00, 24:00)	16	16	1	spark	2022/05/18 16:14:58 GMT+08:00	default		编辑 删除

步骤4 选择要操作的队列，在“操作”列，单击“编辑”，进入到编辑队列界面。

步骤5 在编辑队列界面，根据您当前操作场景，分别对应以下操作：

图 3-12 编辑队列界面



- **新增扩缩容策略：**单击“新增”，添加新的扩缩容策略，分别对“优先级”、“时间段”、“最小CU”和“最大CU”参数设置，单击“确定”完成操作。
- **修改扩缩容策略：**直接修改已有记录的扩缩容策略参数，单击“确定”完成操作。
- **删除扩缩容策略：**在对应扩缩容策略所在行单击“删除”，单击“确定”删除已有的优先级设置。

📖 说明

“优先级”和“时间段”参数说明如下：

- 优先级：默认为1，设置范围为1-100，参数值越大优先级越高。
- 时间段：
 - 时间段设置仅支持整点，时间区间包括开始时间，不包括结束时间即[开始时间, 结束时间)。
 - 例如当前选择的时间段范围为：01--17，则表示当前规则时间范围为[01, 17)。
 - 不同优先级的时间段区间不能有交集。
- 最大最小CU：
 - 在全天的任意一个时间段内，弹性资源池中所有队列的最小CU数之和必须小于等于弹性资源池的最小CU数。
 - 在全天的任意一个时间段内，弹性资源池中任意一个队列的最大CU必须小于等于弹性资源池的最大CU。

步骤6 设置完成后，单击“结果图形化”，查看所有队列的扩缩容策略设置情况。

图 3-13 弹性资源池队列扩缩容策略结果图形化

名称	类型	时间段	最小CUs	最大CUs	优先级	执行策略	创建时间	企业级策略	所有者	操作
period_queue_1	通用队列	[00:00, 17:00] [00:00, 24:00]	16	64	1	--	2022/05/18 16:15:37 GMT+08:00	default		编辑 删除
flink	SQL队列	[00:00, 24:00]	16	16	1	spark	2022/05/18 16:14:50 GMT+08:00	default		编辑 删除

图 3-14 弹性资源池队列扩缩容策略图像化展示



步骤7 后续到了队列扩缩容策略配置的时间，会生成一个扩缩容任务。具体可以在“作业管理 > SQL作业”下查看作业类型为“SCALE_QUEUE”的作业。

----结束

3.3.8 查看弹性资源池扩缩容历史

操作场景

当弹性资源池添加队列、删除队列，或添加的队列扩缩容时，可能会引起弹性资源CUs扩缩容变化。控制台提供的“扩缩容历史”功能，可以查看弹性资源池的CUs变化历史。


约束与限制

当前控制台仅支持查看30天以内的弹性资源池扩缩容历史。

查看弹性资源池扩缩容历史

1. 在DLI管理控制台左侧，选择“资源管理 > 弹性资源池”。
2. 选择需要查看的弹性资源池，单击“操作”列“更多”中的“扩缩容历史”。
3. 在“扩缩容历史”页面，选择日期范围，查看CUs资源使用趋势。

您可以查看弹性资源池扩缩容前CUs，扩缩容后CUs，扩缩容目标CUs。

扩缩容历史支持“图形化展示”和“表格展示”两种形式。可以单击右上方进行切换。

例如：从[查看弹性资源池扩缩容历史](#)可见某一时间扩缩容异常，切换至表格形式后，如[图3-16](#)所示，预期扩容至80CUs，扩容前为64CUs，扩容后64CUs，扩容失败。

图 3-15 图形化展示扩缩容历史



图 3-16 表格形式展示扩缩容历史

最大CUs	最小CUs	扩缩容预期CUs	扩缩容前CUs	扩缩容后CUs	状态	完成时间
128	64	112	80	80	失败	2022/08/05 09:47:35 GMT+08:00
128	64	128	80	80	失败	2022/08/05 09:43:26 GMT+08:00
80	64	80	64	80	成功	2022/08/05 09:35:29 GMT+08:00
80	64	80	64	64	失败	2022/08/05 09:29:53 GMT+08:00
80	64	80	64	64	失败	2022/08/05 09:20:40 GMT+08:00

3.3.9 分配弹性资源池至项目

企业项目是一种云资源管理方式，企业可以根据组织架构规划企业项目，将分布在不同区域的资源按照企业项目进行统一管理，同时可以为每个企业项目设置拥有不同权限的用户组和用户。

DLI支持在创建弹性资源池时选择企业项目，本节操作为您介绍DLI弹性资源池如何绑定、修改企业项目。

说明

修改弹性资源池的企业项目，会同时修改弹性资源池下的队列资源的企业项目。
即弹性资源池下仅支持添加同一企业项目的队列资源。

前提条件

在绑定企业项目前，您已在“企业项目管理控制台”创建[创建企业项目](#)。

绑定企业项目

在创建弹性资源池资源时，可以在“企业项目”绑定已创建的企业项目。

您还可以单击“新建企业项目”，前往企业项目管理控制台，新建企业项目和查看已有的企业项目。

更多创建队列的操作步骤请参考[创建弹性资源池并添加队列](#)。

修改企业项目

针对之前已创建的集群，其绑定的企业项目可根据实际情况进行修改。

1. 登录DLI管理控制台，
2. 在左侧导航栏，选择“资源管理 > 弹性资源池”。
3. 在弹性资源池资源列表中，选择待修改企业项目的资源，并单击操作列下“更多 > 分配至项目”。
4. 在“分配至项目”页面，选择企业项目。
您还可以单击“新建企业项目”，前往企业项目管理控制台，新建企业项目和查看已有的企业项目。
5. 修改完成后，单击“确定”，保存弹性资源池的企业项目信息。

相关操作

如需修改队列企业项目请参考[分配队列至项目](#)。

3.4 管理队列

3.4.1 查看队列的基本信息

本节操作介绍在管理控制台如何查看队列的基本信息，包括队列的引擎类型和引擎版本。

查看队列的基本信息



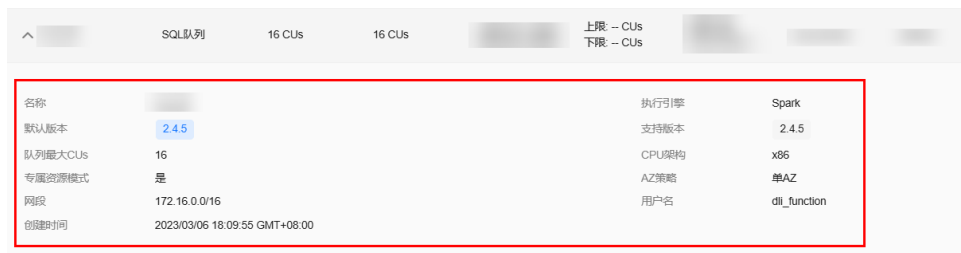
1. 登录DLI管理控制台。
2. 选择“资源管理 > 队列管理”。
3. 进入队列列表页面，选择您需要查看的队列。
 - 在列表页面的右上方单击可以自定义显示列，并设置表格内容显示规则、操作列显示规则。
 - 在列表页面上方的搜索区域，您可以名称和标签筛选需要的队列资源。
4. 单击查看队列的详细信息。
关于队列引擎相关字段的含义：
 - 执行引擎：负责执行队列中任务的引擎类型。
 - 默认版本：执行引擎的默认配置版本，或者是在没有指定特定版本时系统将使用的版本。
 - 支持版本：执行引擎支持的所有版本列表。通过查看队列的支持版本，您可以了解哪些版本的执行引擎可以用于处理队列中的任务。

图 3-17 队列基本信息



3.4.2 队列权限管理

管理员用户和队列的所有者拥有队列的所有操作权限，且根据业务需求对其他用户分配队列的操作权限，确保用户之间的作业互不影响，保障作业的执行性能。本节操作介绍队列权限管理的相关操作。

操作须知

- 管理员用户和队列的所有者拥有所有权限，不需要进行权限设置且其他用户无法修改其队列权限。
- 给新用户设置队列权限时，该用户所在用户组的所属区域需具有Tenant Guest权限。

关于Tenant Guest权限的介绍和开通方法，详细参见《[权限策略](#)》和《[统一身份认证服务用户指南](#)》中的[创建用户组](#)。

队列权限相关操作步骤

步骤1 在DLI管理控制台的左侧，选择“资源管理 > 队列管理”。

步骤2 选择待设置的队列，单击其“操作”列中的“权限管理”。“用户权限信息”区域展示了当前具备此队列权限的用户列表。

常见权限设置的场景：为新用户赋予权限，为已有权限的用户修改权限，回收某用户具备的所有权限。

- **为新用户赋予权限**

新用户指之前不具备此队列权限的用户。

- 单击“权限信息”右侧的“授权”，弹出“授权”对话框。
- 填写“用户名”，并勾选对应权限。
- 单击“确定”，完成新用户的添加。

待设置的参数说明如[表3-12](#)所示。

图 3-18 队列权限授权



表 3-12 参数说明

参数名称	描述
用户名	被授权的用户名称。 说明 该用户名称是已存在的IAM用户名称且该用户登录过DLI管理控制台。
权限设置	<ul style="list-style-type: none"> ▪ 删除队列：删除此队列。 ▪ 提交作业：向此队列提交作业。 ▪ 终止作业：终止提交到此队列的作业。 ▪ 赋权：当前用户可将队列的权限赋予其他用户。 ▪ 回收：当前用户可回收其他用户具备的该队列的权限，但不能回收该队列所有者的权限。 ▪ 查看其他用户具备的权限：当前用户可查看其他用户具备的该队列的权限。 ▪ 重启队列权限：重启此队列的权限。 ▪ 规格变更：修改队列规格的权限。

- 为已有权限的用户赋予权限或回收权限。
 - a. 在对应队列“权限信息”区域的用户列表中，选择需要修改权限的用户，在“操作”列单击“权限设置”。
 - b. 在队列“权限设置”对话框中，对当前用户具备的权限进行修改。详细权限描述如表3-12所示。

当“权限设置”中的选项为灰色时，表示您不具备修改此队列权限的权限。可以向管理员用户、队列所有者等具有赋权权限的用户申请“队列的赋权”和“队列权限的回收”权限。

图 3-19 队列权限设置



- c. 单击“确定”完成权限设置。
- 回收某用户具备的所有权限。

在对应队列“权限信息”区域的用户列表中，选择需要删除权限的用户，在“操作”列单击“回收用户权限”。在“回收用户权限”对话框中单击“是”后，此用户将不具备该队列的任意权限。

---结束

3.4.3 分配队列至项目

企业项目是一种云资源管理方式，企业可以根据组织架构规划企业项目，将分布在不同区域的资源按照企业项目进行统一管理，同时可以为每个企业项目设置拥有不同权限的用户组和用户。

DLI支持在创建队列时选择企业项目，本节操作为您介绍DLI队列资源如何绑定、修改企业项目。

说明

当前仅支持对未加入弹性资源池的队列资源修改企业项目。

前提条件

在绑定企业项目前，您已在“企业项目管理控制台”创建[创建企业项目](#)。

修改企业项目

针对之前已创建的队列，其绑定的企业项目可根据实际情况进行修改。

1. 登录DLI管理控制台。
2. 在左侧导航栏，选择“资源管理 > 队列管理”。
3. 在队列资源列表中，选择待修改企业项目的队列，并单击操作列下“更多 > 分配至项目”。
4. 在“分配至项目”页面，选择企业项目。

您还可以单击“新建企业项目”，前往企业项目管理控制台，新建企业项目和查看已有的企业项目。

说明

弹性资源池中的队列不计费，弹性资源池下队列切换的企业项目和计费无关。即不支持按企业项目查看弹性资源池中的队列资源计费信息。

5. 修改完成后，单击“确定”，保存队列的企业项目信息。

相关操作

如需修改弹性资源池企业项目请参考[分配弹性资源池至项目](#)。

3.4.4 创建消息通知主题

操作场景

确定创建消息通知主题后，您可在消息通知服务的“主题管理”页面中，对相应的主题添加订阅，选择不同方式（例如短信或者邮件等）进行订阅。订阅成功后，如果作业失败，则系统将会自动发送消息到您指定的订阅终端。

- 如果作业提交1分钟内立即失败，通常不会触发消息通知。
- 如果作业提交1分钟后失败，则系统将会自动发送消息到您指定的订阅终端。

操作步骤

1. 在“资源管理 > 队列管理”页面，单击左上角“创建消息通知主题”。

图 3-20 创建消息通知主题

创建消息通知主题

创建消息通知主题后，您可选择不同方式进行订阅；订阅成功后，作业失败将会自动发送消息到您指定的订阅终端。

请选择队列

2. 选择队列，单击“确定”。

说明

- 选择队列时，可以选择单个队列，也可以选择所有队列。
- 如果单个队列和所有队列的终端不一致，当选择了单个队列，同时选择了所有队列进行订阅时，在所有队列的消息通知中将不包含该队列的消息。
- 创建消息通知主题后，只有在订阅队列上创建的Spark作业失败时才会收到消息通知。

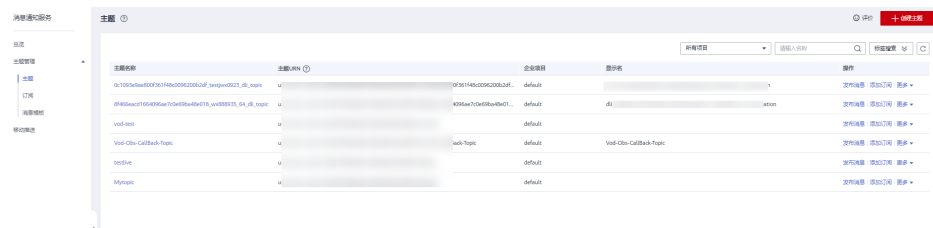
图 3-21 创建主题成功

创建成功

主题名称为
011c99a26ae84a1bb963a75e7637d3fd_all_dli_topic,
您可在消息通知服务主题管理添加订阅。

3. 单击图3-21中“主题管理”，跳转至消息通知服务“主题管理”页面。

图 3-22 主题管理



4. 在对应主题的“操作”列中，单击“添加订阅”，选择“协议”，确定订阅方式。
 - 如果选择“短信”协议，需要在“订阅终端”中填写接收确认短信的手机号码。
 - 如果选择“邮件”协议，需要在“订阅终端”中填写接收确认邮件的邮箱地址。
 - 更多信息，请参考《消息通知服务用户指南》中《[添加订阅](#)》章节。

图 3-23 添加订阅

添加订阅

* 主题名称 +

* 协议

* 订阅终端 + 添加订阅终端

备注

确定 取消

5. 通过单击短信或者邮件中的链接确认后，将收到“订阅成功”的信息。
6. 在消息通知服务的“订阅”页面，对应的订阅状态为“已确认”，表示订阅成功。

3.4.5 队列标签管理

标签管理

标签是用户自定义的、用于标识云资源的键值对，它可以帮助用户对云资源进行分类和搜索。标签由标签“键”和标签“值”组成。

如果用户在其他云服务中使用了标签，建议用户为同一个业务所使用的云资源创建相同的标签键值对以保持一致性。

DLI支持以下两类标签：

- 资源标签：在DLI中创建的非全局的标签。
- 预定义标签：在标签管理服务（简称TMS）中创建的预定义标签，属于全局标签。

有关预定义标签的更多信息，请参见《[标签管理服务用户指南](#)》。

如您的组织已经设定DLI的相关标签策略，则需按照标签策略规则为资源添加标签。标签不符合标签策略的规则，则可能会导致资源创建失败，请联系组织管理员了解标签策略详情。

以下介绍如何为队列添加标签、修改标签和删除标签。

步骤1 在DLI管理控制台的左侧导航栏中，单击“资源管理 > 队列管理”。

步骤2 在对应队列的“操作”列，选择“更多”>“标签”。

步骤3 进入标签管理页面，显示当前队列的标签信息。

步骤4 单击“添加/编辑标签”，弹出“添加/编辑标签”对话框，配置参数。配置完成一个标签，单击“添加”将标签添加到输入框中。

图 3-24 添加/编辑标签



表 3-13 标签配置参数

参数	参数说明
标签键	<p>您可以选择：</p> <ul style="list-style-type: none"> 在输入框的下拉列表中选择预定义标签键。 如果添加预定义标签，用户需要预先在标签管理服务中创建好预定义标签，然后在“标签键”的下拉框中进行选择。用户可以通过单击“查看预定义标签”进入标签管理服务的“预定义标签”页面，然后单击“创建标签”来创建新的预定义标签。 具体请参见《标签管理服务用户指南》中的“创建预定义标签”章节。 在输入框中输入标签键名称。 <p>说明 标签的键的最大长度为128个字符，标签的键可以包含任意语种字母、数字、空格和_ . : + - @ ,但首尾不能含有空格，不能以_sys_开头。</p>
标签值	<p>您可以选择：</p> <ul style="list-style-type: none"> 在输入框的下拉列表中选择预定义标签值。 在输入框中输入标签值。 <p>说明 标签值的最大长度为255个字符，标签的值可以包含任意语种字母、数字、空格和_ . : + - @ 。</p>

📖 说明

- 最多支持20个标签。
- 一个“键”只能添加一个“值”。
- 每个资源中的键名不能重复。

步骤5 单击“确定”。

步骤6 （可选）在标签列表中，单击“操作”列中“删除”可对选中的标签进行删除。

----结束

3.4.6 队列属性设置

操作场景

DLI支持在队列创建完成后设置队列的属性。

当前支持设置队列的Spark driver的相关参数：通过设置队列的Spark driver，以提升队列资源的调度效率。

本节操作介绍在管理控制台设置队列属性的操作步骤。

约束与限制

- 仅Spark引擎的SQL队列支持配置队列属性。
- 仅在队列创建完成后支持设置队列属性。
- 当前仅支持设置Spark driver相关队列属性。
- 不支持批量设置队列属性。
- 弹性资源池中的队列，当队列的最小CUs小于16CUs时，在队列属性中设置的“最大spark driver实例数”和“最大预拉起spark driver实例数”不生效。

设置队列属性操作步骤

1. 在DLI管理控制台的左侧导航栏中，单击“资源管理 > 队列管理”。
2. 在对应队列的“操作”列，选择“更多 > 属性设置”。
3. 进入队列属性设置页面，设置对应的属性值。属性值相关参数说明请参考[表3-14](#)

表 3-14 队列属性

属性名称	说明	取值范围
最大spark driver实例数	队列能启动的最大spark driver数量。包含预先启动的spark driver和运行作业的spark driver。	<ul style="list-style-type: none">• 当队列为16CUs时范围：2• 当队列大于16CUs时范围：2-（CU数/16）• 队列最小CUs小于16CUs时，该配置项不生效。

属性名称	说明	取值范围
最大预拉起spark driver实例数	队列预先启动的最大spark driver数量。当运行作业的spark driver任务数超过“单spark driver实例最大并发数”的值时，作业将会分配到预先启动的spark driver上面。	<ul style="list-style-type: none"> 当队列为16CUs时范围：0-1 当队列大于16CUs时范围：2-（CU数/16） 队列最小CUs小于16CUs时，该配置项不生效。
单spark driver实例最大并发数	单个spark driver能同时运行的最大任务数量。当任务超过此值时，作业将会分配给其它spark driver运行。	1-32

4. 单击“确定”完成队列属性的设置。

3.4.7 测试地址连通性

DLI提供的“测试地址连通性”用于验证DLI队列与目标地址之间的网络连通性。

常用于读写外部数据源场景，在配置了跨源连接后，检验DLI队列与绑定的跨源对端地址之间的通信能力。

测试队列与数据源地址连通性

1. 登录DLI管理控制台，选择“资源管理 > 队列管理”。
2. 在“队列管理”页面，选择需要测试地址连通性的队列，单击操作列下的“更多 > 测试地址连通性”。
3. 在“测试地址连通性”页面填写需要测试的地址。支持域名和IP，可指定端口。数据源地址支持以下输入格式：IPv4地址、IPv4+端口号、域名、域名+端口号。
 - IPv4地址：192.168.x.x
 - IPv4+端口号：192.168.x.x:8080
 - 域名：domain-xxxxxx.com
 - 域名+端口号：domain-xxxxxx.com:8080

图 3-25 测试地址连通性



4. 单击“测试”。
 - 如果测试地址可连通，页面上将提示地址可达。
 - 如果测试地址不可连通，页面上将提示地址不可达，请检查网络配置后重试。检查网络配置即检查所测试的VPC对等连接或跨源连接是否处于已激活状态。

相关操作

[创建跨源成功但测试网络连通性失败怎么办？](#)

3.4.8 删除队列

根据实际使用情况，您可以通过删除操作释放队列。

📖 说明

- 如果待删除的队列中有正在提交或正在运行的作业，将不支持删除操作。
- 删除队列不会导致您数据库中的表数据丢失。

删除队列步骤

步骤1 在DLI管理控制台左侧，选择“资源管理 > 队列管理”。

步骤2 选择待删除的队列，单击“操作”列的“删除”删除。

图 3-26 删除队列



📖 说明

如果“操作”列的“删除”为灰色，表示当前用户没有删除队列的权限。您可以向管理员申请删除队列的权限。

步骤3 在弹出的确认对话框中，单击“是”。

----结束

3.4.9 变更普通队列规格

前提条件

新创建的包年包月计费队列需要运行作业后才可进行规格变更。

📖 说明

本节操作仅适用于普通队列，不适用于弹性资源池队列。

注意事项

- 目前只支持64CUs以上规格包年包月队列进行规格变更。
- 如果在“规格变更”页面提示“Status of queue xxx is assigning, which is not available”，表示需要等待队列资源分配完毕才可进行扩缩容。

扩容

当前队列规格不满足业务需要时，可以通过手动变更队列规格来扩容当前队列。

📖 说明

扩容属于耗时操作，在DLI“规格变更”页面执行扩容操作后，需要等待大约10分钟，具体时长和扩容的CU值有关，等待一段时间后，可以通过刷新“队列管理”页面，对比“规格”和“实际CUs”大小是否一致来判断是否扩容成功。或者在“作业管理”页面，查看“SCALE_QUEUE”类型SQL作业的状态，如果作业状态为“规格变更中”，表示队列正在扩容中。

操作步骤如下：

1. 在DLI管理控制台左侧，选择“资源管理 > 队列管理”。
2. 选择需要扩容的队列，单击“操作”列“更多”中的“规格变更”。
3. 在“规格变更”页面，“变更方式”选择“扩容”，设置扩容的CU值。

图 3-27 扩容

名称	_____
计费模式	包年/包月
过期时间	_____
当前规格	64 CUs
变更方式	<input checked="" type="button" value="扩容"/> <input type="button" value="缩容"/>
变更数量	<input type="button" value="-"/> <input type="text" value="16"/> <input type="button" value="+"/>
最终规格	80 CUs

4. 确定费用后，单击“提交”。

缩容

当计算业务较小，不需要那么大的队列规格时，可以通过手动变更队列规格来缩容当前队列。

说明

- 缩容属于耗时操作，在DLI“规格变更”页面执行缩容操作后，需要等待大约10分钟，具体时长和缩容的CU值有关，等待一段时间后，可以通过刷新“队列管理”页面，对比“规格”和“实际CUs”大小是否一致来判断是否缩容成功。或者在“作业管理”页面，查看“SCALE_QUEUE”类型SQL作业的状态，如果作业状态为“规格变更中”，表示队列正在缩容中。
- 系统不保证完全缩容到设定的目标大小。如果当前队列正在使用或者队列业务量比较大，会出现缩容不成功，或者缩容一部分规格的情况。
- 系统默认最小CU值为16CU。即当队列规格为16CUs时，不能进行手动缩容。

操作步骤如下：

1. 在DLI管理控制台左侧，选择“资源管理 > 队列管理”。
2. 选择需要缩容的队列，单击“操作”列“更多”中的“规格变更”。
3. 在“规格变更”页面，“变更方式”选择“缩容”，设置缩容的CU值。

图 3-28 手动缩容

名称

计费模式 包年/包月

过期时间

当前规格 80 CUs

变更方式 扩容 缩容

变更数量 - 4 +

最终规格 76 CUs

4. 确定费用后，单击“提交”。

3.4.10 普通队列弹性扩缩容

前提条件

新创建的按需计费队列需要运行作业后才可进行弹性扩缩容。

📖 说明

本节操作仅适用于普通队列，不适用于弹性资源池队列。

约束与限制

- 16CUs队列不支持扩容和缩容。
- 64CUs队列不支持缩容。
- 目前只支持计费模式为“按需/CU时”和“按需/专属资源模式”的队列进行弹性扩缩容。
- 如果在“弹性扩缩容”页面提示“Status of queue xxx is assigning, which is not available”，表示需要等待队列资源分配完毕才可进行扩缩容。
- 队列资源扩容时，可能会由于物理资源不足导致队列资源无法扩容到设定的目标大小。
- 队列资源缩容时，系统不保证将队列资源完全缩容到设定的目标大小。通常队列资源缩容时，系统会先检查资源使用情况，判断是否存在缩容空间，如果现有资源无法按照最小缩容步长执行缩容任务，则队列可能缩容不成功，或缩容一部分规格的情况。

因资源规格不同可能有不同的缩容步长，通常是16CUs、32CUs、48CUs、64CUs等。

示例：队列大小为48CUs，执行作业占用了18CUs，剩余30CUs不满足该32CUs步长缩容的要求，如果执行缩容任务，则缩容失败。

弹性扩容

当前队列规格不满足业务需要时，可以通过手动变更队列规格来扩容当前队列。

📖 说明

扩容属于耗时操作，在DLI“弹性扩缩容”页面执行扩容操作后，需要等待大约10分钟，具体时长和扩容的CU值有关，等待一段时间后，可以通过刷新“队列管理”页面，对比“规格”和“实际CUs”大小是否一致来判断是否扩容成功。或者在“作业管理”页面，查看“SCALE_QUEUE”类型SQL作业的状态，如果作业状态为“弹性扩缩容中”，表示队列正在扩容中。

操作步骤如下：

1. 在DLI管理控制台左侧，选择“资源管理 > 队列管理”。
2. 选择需要扩容的队列，单击“操作”列“更多”中的“弹性扩缩容”。
3. 在“弹性扩缩容”页面，“变更方式”选择“扩容”，设置扩容的CU值。

图 3-29 弹性扩容

名称	sparktest
计费模式	按需计费
实际CUs	16 CUs
变更方式	<input checked="" type="radio"/> 扩容 <input type="radio"/> 缩容
变更数量	<input type="button" value="-"/> <input type="text" value="16"/> <input type="button" value="+"/>
最终CUs	32 CUs

4. 确认费用无误后，单击“确定”。

弹性缩容

当计算业务较小，不需要那么大的队列规格时，可以通过手动变更队列规格来缩容当前队列。

说明

- 缩容属于耗时操作，在DLI“弹性扩缩容”页面执行缩容操作后，需要等待大约10分钟，具体时长和缩容的CU值有关，等待一段时间后，可以通过刷新“队列管理”页面，对比“规格”和“实际CUs”大小是否一致来判断是否缩容成功。或者在“作业管理”页面，查看“SCALE_QUEUE”类型SQL作业的状态，如果作业状态为“弹性扩缩容中”，表示队列正在缩容中。
- 系统默认最小CU值为16CU，即当队列规格为16CUs时，不能进行手动缩容。

操作步骤如下：

1. 在DLI管理控制台左侧，选择“资源管理 > 队列管理”。
2. 选择需要缩容的队列，单击“操作”列“更多”中的“弹性扩缩容”。
3. 在“弹性扩缩容”页面，“变更方式”选择“缩容”，设置缩容的CU值。

图 3-30 手动扩容

名称	createoutcuqueue
计费模式	按需计费
实际CUs	128 CUs
变更方式	<input type="radio"/> 扩容 <input checked="" type="radio"/> 缩容
变更数量	<input type="button" value="-"/> <input type="text" value="16"/> <input type="button" value="+"/>
最终CUs	112 CUs

4. 确认费用无误后，单击“确定”。

3.4.11 设置普通队列的弹性扩缩容定时任务

弹性扩缩容定时任务使用场景

通常，用户业务繁忙的场景是有周期性的，在某个周期内，用户需要更多的计算资源来处理业务，过了这个周期，则不需要那么多资源。如果用户购买的队列规格比较小，在业务繁忙时会存在资源不足的情况；而如果购买的队列规格比较大，又可能会存在资源浪费的情况。

基于以上场景，DLI提供了队列弹性扩缩容定时任务功能。用户可以根据自己的业务周期或者使用情况，基于现有队列规格，在不同的时间或者周期内设置不同的队列大小，以满足自己的业务需求，节约成本。

📖 说明

本节操作仅适用于普通队列，不适用于弹性资源池队列。

使用弹性扩缩容定时任务注意事项

- 新创建的队列需要运行作业后才可进行扩缩容。
- 目前只支持规格为64CUs以上的队列进行定时弹性扩缩容任务，即队列最小规格为64CUs。
- 对于每个队列，最多支持创建12个定时任务。
- 每个定时任务开始时，弹性扩缩容的实际开始的时间有5分钟误差。建议扩容时间定时至少比实际使用队列的时间提前20分钟。
- 每个定时任务之间需要至少有2小时的间隔。
- 队列的定时弹性扩缩容属于耗时操作，变更所消耗的时间取决于扩缩容目标规格与当前规格的差值大小，用户在“队列管理”页面中可以查看当前队列的规格。
- 如果当前队列有作业正在运行时，可能无法缩容到目标CU值，而是缩容到当前队列规格和目标规格中间的某个值，系统将在1小时后继续尝试进行缩容，直至下一个定时任务开始。

- 当一个定时任务没有扩容或者缩容到目标CU值时，系统会在约15分钟后再次触发扩缩计划，直到下一个定时任务开始。

创建弹性扩缩容定时任务

- 如果只设定扩容或者缩容，只需创建一个弹性扩缩容定时任务。设定“任务名称”、“最终CUs”和“执行时间”即可，具体请参考表3-15。
- 如果需要同时设定扩容和缩容，则需要创建两个弹性扩缩容定时任务，分别设定扩容和缩容的“任务名称”、“最终CUs”和“执行时间”，具体请参考表3-15。

操作步骤如下：

1. 在DLI管理控制台左侧，选择“资源管理 > 队列管理”。
2. 选择需要设置弹性扩缩容定时任务的队列，单击“操作”列“更多”中的“弹性扩缩容定时任务”。
3. 在“弹性扩缩容定时任务”页面，单击右上角的“创建定时任务”。
4. 在“创建定时任务”页面，设置参数。单击“确定”。

图 3-31 创建定时任务

创建定时任务

* 任务名称

激活任务

有效期 -

实际CUs 32 CUs

* 最终CUs 16 CUs

重复规律 全选
 周一 周二 周三 周四 周五 周六 周日

* 执行时间

表 3-15 参数说明

参数名称	描述
任务名称	输入定时任务的名称。 <ul style="list-style-type: none"> • 只能包含数字、英文字母和下划线，但不能是纯数字，不能以下划线开头，且不能为空。 • 输入长度不能超过128个字符。

参数名称	描述
激活任务	激活队列扩缩容定时任务。默认开启。如果关闭，则系统不会触发执行当前设置的定时规格变更任务。
有效期	<p>设置执行定时任务的时间段。包括“日期”和“时间”。</p> <p>说明</p> <ul style="list-style-type: none"> “有效期”中的“开始时间”需要晚于当前的系统时间。 如果只设置了扩容，在“有效期”结束之后，系统不会自动缩容，需要手动修改或设置缩容定时任务。反之亦然。即为单次执行定时扩缩容。 如果同时设置了扩容和缩容，在有效期内会按照设定扩缩容，在“有效期”结束之后，将保持最后一次设定的队列规格。
实际 CUs	队列扩容或缩容前的规格。
最终 CUs	<p>队列扩容或缩容后的规格。</p> <p>说明</p> <ul style="list-style-type: none"> 系统默认队列最大规格为512CUs。 进行定时扩缩容操作的队列最小规格为64CUs，即当“实际CUs”小于64CUs时，不能进行定时扩缩容。 最终规格只能为16的倍数。
重复规律	<p>选择执行定时扩缩容的周期。定时任务的“重复规律”支持按周为周期进行调度。</p> <ul style="list-style-type: none"> 默认不选，表示“不重复”，即只在“执行时间”执行一次； 如果全选，表示该计划每天都会执行； 如果选择部分，则选择规律的计划在对应的时间每周都会被执行一次。 <p>说明</p> <ul style="list-style-type: none"> 如果只是单次执行扩容或者缩容，无需选择“执行周期”。 如果同时设置了扩缩容，可根据需要选择“执行周期”，还可与“有效期”进行配合使用。
执行时间	<p>执行定时扩容或者缩容的时间。</p> <ul style="list-style-type: none"> 每个定时任务开始时，弹性扩缩容的实际开始的时间有5分钟误差。建议扩容时间定时至少比实际使用队列的时间提前20分钟。 每个定时任务之间需要至少有2小时的间隔。

定时任务创建后，可以在“弹性扩缩容定时任务”页面查看当前队列的规格变化情况，以及计划最近一次的执行时间。

或者在“队列管理”页面，查看“规格”大小是否改变来判断是否扩缩容成功。

或者在“作业管理”页面，查看“SCALE_QUEUE”类型作业的状态，如果作业状态为“规格变更中”，表示队列正在扩缩容中。

修改弹性扩缩容定时任务

如果设定的定时任务不再满足业务需求，可以在“弹性扩缩容定时任务”页面修改弹性扩缩容定时任务。

1. 在DLI管理控制台左侧，选择“资源管理 > 队列管理”。
2. 选择需要设置弹性扩缩容定时任务的队列，单击“操作”列“更多”中的“弹性扩缩容定时任务”。
3. 在“弹性扩缩容定时任务”页面，单击操作列的“修改”，根据提示修改弹性扩缩容定时任务。

删除弹性扩缩容定时任务

如果不再需要定时修改队列规格，可以在“弹性扩缩容定时任务”页面删除弹性扩缩容定时任务。

1. 在DLI管理控制台左侧，选择“资源管理 > 队列管理”。
2. 选择需要设置弹性扩缩容定时任务的队列，单击“操作”列“更多”中的“弹性扩缩容定时任务”。
3. 在“弹性扩缩容定时任务”页面，单击操作列的“删除”，根据提示删除弹性扩缩容定时任务。

3.4.12 修改普通队列的网段

使用增强型跨源时，如果DLI队列的网段和用户数据源的网段发生冲突，您可以通过修改网段操作更改包年包月队列的网段。

如果待修改网段的队列中有正在提交或正在运行的作业，或者该队列已经绑定了增强型跨源，将不支持修改网段操作。

说明

本节操作仅适用于普通队列，不适用于弹性资源池队列。

修改队列网段步骤

说明

目前只支持计费模式为“包年包月”和“按需/专属资源模式”的队列修改网段。

1. 在DLI管理控制台左侧，选择“资源管理 > 队列管理”。
2. 选择待修改的队列，单击“操作”列“更多”中的“修改网段”。

图 3-32 修改网段



修改网段

名称 testaaa

网段 172 . 16 . 0 . 0 / 18

如需使用DLI增强型跨源, DLI队列网段与数据源网段不能重合

建议使用网段:
10.0.0.0~10.255.0.0/8~24,172.16.0.0~172.31.0.0/12~24,192.168.0.0~192.168.0.0/16~24

确定 取消

3. 填写需要的网段后, 单击“确定”。队列修改网段成功后, 需要等待5~10分钟, 待队列所属集群资源重新拉起后再运行作业。

建议使用网段:

10.0.0.0~10.255.0.0/8~24

172.16.0.0~172.31.0.0/12~24

192.168.0.0~192.168.0.0/16~24

3.5 典型场景示例：创建弹性资源池并运行作业

本章节主要介绍从创建弹性资源池、创建增强型跨源、添加队列到弹性资源池并运行作业的一个完整流程, 帮助您更好、更方便的使用弹性资源池。

图 3-33 创建弹性资源池运行作业流程图

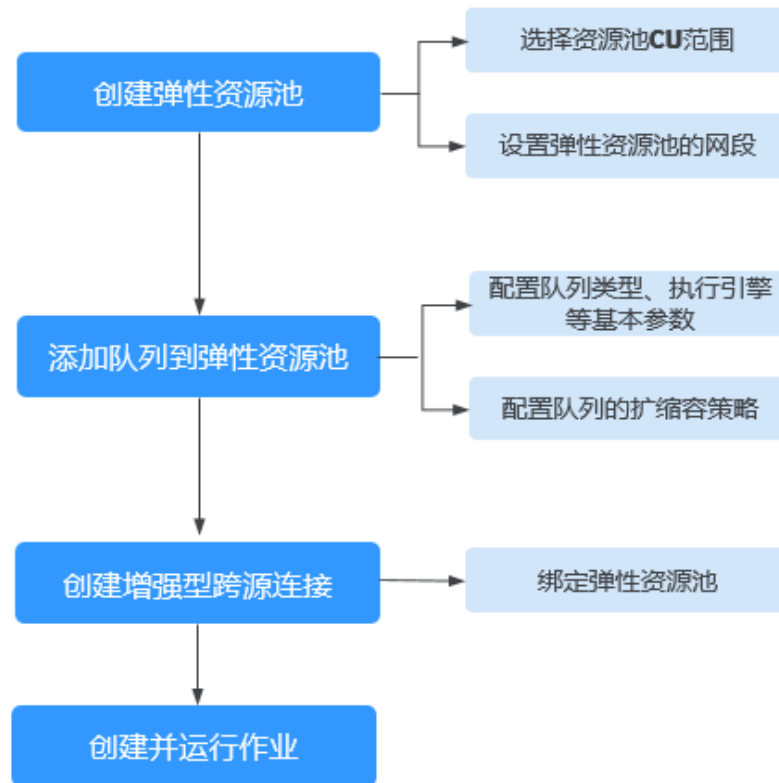


表 3-16 创建新队列时绑定弹性资源池流程说明

阶段	说明	参考文档
步骤一：创建弹性资源池	创建弹性资源池，配置弹性资源池的基本信息，如：计费模式、CU范围、网段等配置。	创建弹性资源池并添加队列
步骤二：添加队列到弹性资源池	添加作业运行的队列到弹性资源池。具体内容包括： 1. 设置弹性资源池队列的名称、队列类型等基本信息。 2. 配置当前队列的扩缩容策略，包括队列策略的优先级、时间段、最大最小CU范围等配置。	创建弹性资源池并添加队列 调整弹性资源池中队列的扩缩容策略

阶段	说明	参考文档
(可选) 步骤三：创建增强型跨源连接	如果运行的作业需要跨源访问其他外部数据源数据，如作业需要访问DWS、RDS等数据时，需要创建跨源连接。 创建的跨源连接需要和弹性资源池进行绑定。	创建增强型跨源连接
步骤四：运行作业	根据业务需求创建和提交作业。	管理SQL作业 Flink作业概述 创建Spark作业

步骤一：创建弹性资源池

1. 登录DLI管理控制台，在左侧导航栏单击“资源管理 > 弹性资源池”，可进入弹性资源池管理页面。
2. 在弹性资源池管理界面，单击界面右上角的“购买弹性资源池”。
3. 在“购买弹性资源池”界面，填写具体的弹性资源池参数，具体参数填写参考如下。
 - 名称：填写具体的弹性资源池名称。例如设置为：pool_test。
 - CU范围：弹性资源池扩缩容的CU范围。
 - 网段：配置弹性资源池网段。例如当前配置为172.16.0.0/18。
 - 其他参数根据需要选择和配置。

图 3-34 创建弹性资源池

The screenshot shows the 'Purchase Elastic Resource Pool' configuration interface. Key elements include:

- 计费模式 (Billing Mode):** Options for '包年/包月' (Annual/Monthly) and '按需计费' (Pay-as-you-go).
- 区域 (Region):** A dropdown menu with a note: '不同区域的云服务之间内网互不相通；请就近选择靠近您业务的区域，可减少网络时延，提高访问速度。' (Internal networks of cloud services in different regions are not connected; please choose the region closest to your business to reduce network latency and improve access speed.)
- 项目 (Project):** A dropdown menu.
- * 名称 (Name):** Input field containing 'pool_test'.
- CU范围 (CU Range):** Input fields showing a range from 64 to 64, with a '专属资源模式' (Dedicated Resource Mode) checkbox checked.
- 描述 (Description):** An empty text area with a character limit of 0/256.
- 网段 (Network Segment):** Input field showing '172.16.0.0 / 18'. Below it, a note states: '预估可管理 4096 CUs' (Estimated 4096 CUs manageable) and '如需使用DLI增强型跨源，弹性资源池网段与数据源网段不能重合。弹性资源池网段设置后不支持更改' (If using DLI enhanced cross-source, the elastic resource pool network segment cannot overlap with the data source network segment. After setting the elastic resource pool network segment, it is not supported to be changed). A recommended network segment is provided: '建议使用网段：10.0.0.0~10.255.0.0/8-19, 172.16.0.0~172.31.0.0/12~19, 192.168.0.0~192.168.0.0/16-19'.
- * 企业项目 (Enterprise Project):** Input field containing 'default', with a '新建企业项目' (New Enterprise Project) button.

详细的弹性资源池创建流程可以参考[创建弹性资源池并添加队列](#)。

4. 参数填写完成后，单击“立即购买”，确认配置信息无误后，单击“提交”完成弹性资源池创建。
5. 弹性资源池创建任务提交后，会在弹性资源池管理界面的“状态”列显示当前资源池的创建状态，当状态显示为“可使用”时表示资源池可以正常使用。

步骤二：添加队列到弹性资源池

1. 在已创建的弹性资源池的“操作”列，单击“添加队列”进入弹性资源池添加的队列的操作界面。
2. 首先配置弹性资源池队列的基本信息，具体参数参考如下。
 - 名称：添加的队列的名称。
 - 类型：根据作业需要选择队列类型。本示例选择为：通用队列。
SQL队列类型：用于运行Spark SQL和HetuEngine作业。
通用队列类型：用于运行Flink和Spark Jar作业。
 - 其他参数请根据需要配置。

图 3-35 添加队列



3. 配置完基本参数后，单击“下一步”，在队列的扩缩容策略配置界面，修改扩缩容策略配置：最小CU：64、最大CU：64。

图 3-36 队列扩缩容策略配置



- 单击“确定”完成添加队列操作。

(可选) 步骤三：创建增强型跨源连接

本示例演示的操作需要跨源连接RDS外部数据源，所以需要创建跨源连接。如果作业不需要连接外部数据源，则该步骤可以跳过。

- 登录RDS控制台，创建RDS数据库实例。
具体操作请参见[购买RDS for MySQL实例](#)。
- 登录RDS实例后，单击“新建数据库”，创建名称为“test2”的数据库。
- 在“test2”的数据库所在行，操作列，单击“SQL查询”，输入以下创建表语句，单击“执行SQL”创建表“tabletest2”。建表语句参考如下：

```
CREATE TABLE `tabletest2` (  
  `id` int(11) unsigned,  
  `name` VARCHAR(32)  
) ENGINE = InnoDB DEFAULT CHARACTER SET = utf8mb4;
```
- 在RDS管理控制台，单击“实例管理”，单击已创建的RDS具体实例名称，查看该RDS实例的“基本信息”。
- 在“基本信息”的“连接信息”中获取该实例的“内网地址”、“数据库端口”、“虚拟私有云”和“子网”信息，方便后续操作步骤使用。
- 单击“连接信息”中的安全组名称，在“入方向规则”中添加放通弹性资源池网段的规则。例如本示例为3弹性资源池网段为“172.16.0.0/18”，数据库端口为3306，则规则添加为：优先级选择：1，策略选择：允许，协议级别和端口选择：TCP和3306，类型：IPv4，源地址为：172.16.0.0/18
单击“确定”完成安全组规则添加。
- 登录DLI管理控制台，在左侧导航栏单击“跨源管理”，在跨源管理界面，单击“增强型跨源”，单击“创建”。
- 在增强型跨源创建界面，配置具体的跨源连接参数。具体参考如下。
 - 连接名称：设置具体的增强型跨源名称。
 - 弹性安全策略资源池：选择[步骤一：创建弹性资源池](#)中已经创建的好的弹性资源池。

说明

如果该步骤不选择弹性资源池，可以创建跨源完后，在增强型跨源界面，在对应跨源连接所在行的“操作”列，单击“更多 > 绑定弹性资源池”进行绑定。

- 虚拟私有云：选择5中获取的RDS的虚拟私有云。
- 子网：选择5中获取的RDS的子网。
- 其他参数可以根据需要选择配置。

参数配置完成后，单击“确定”完成增强型跨源配置。单击创建的跨源连接名称，查看跨源连接的连接状态，等待连接状态为：“已激活”后可以进行后续步骤。

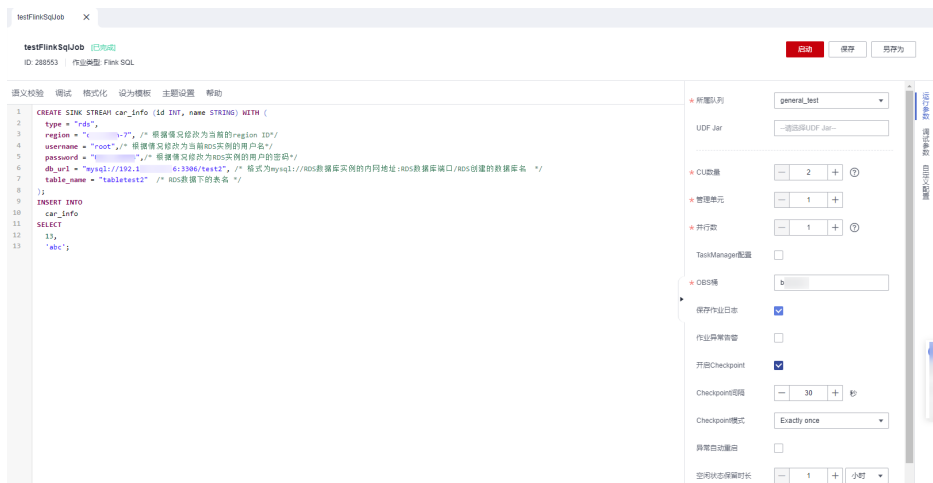
- 单击“资源管理 > 队列管理”，选择操作的队列，如本示例的“general_test”，在操作列，单击“更多 > 测试地址连通性”。
- 在“测试连通性”界面，根据5中获取的RDS连接信息，地址栏输入“RDS内网地址:RDS数据库端口”，单击“测试”测试到RDS网络是否可达。

步骤四：运行作业

本示例通过在弹性资源池队列上运行一个Flink SQL举例演示。

1. 在DLI管理控制台，单击“作业管理 > Flink作业”，在Flink作业管理界面，单击“创建作业”。
2. 在创建作业界面，类型选择“Flink SQL”，名称填写为：testFlinkSqlJob。单击“确定”，跳转到Flink作业编辑界面。
3. 在Flink SQL作业编辑界面，配置如下参数。

图 3-37 创建 Flink SQL 作业

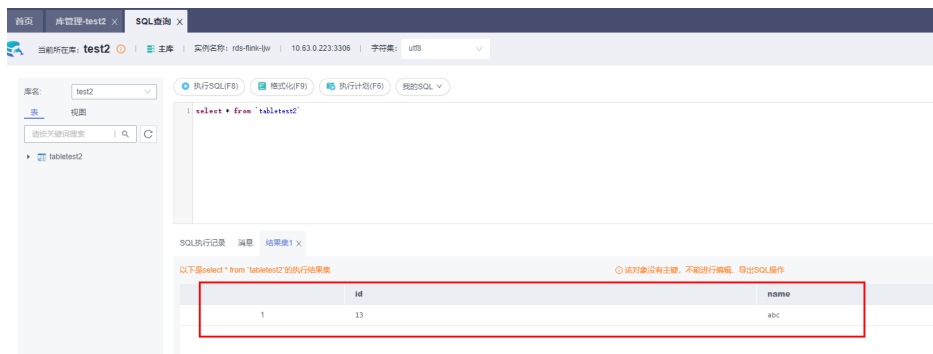


- 所属队列：选择**步骤二：添加队列到弹性资源池**中弹性资源池添加的队列“general_test”。
- 保存作业日志：勾选。
- OBS桶：选择保存作业日志的OBS桶，根据提示进行OBS桶权限授权。
- 开启Checkpoint：勾选。
- Flink作业编辑框中输入具体的作业SQL，本示例作业参考如下。具体加粗的参数需要根据实际情况修改。

```
CREATE SINK STREAM car_info (id INT, name STRING) WITH (
  type = "rds",
  region = "", /* 根据实际情况修改为当前的region ID*/
  'pwd_auth_name'="xxxxx", // DLI侧创建的Password类型的跨源认证名称。使用跨源认证则无需在
  作业中配置账号和密码。
  db_url = "mysql://192.168.x.x:3306/test2", /* 格式为mysql://RDS数据库实例的内网地址:RDS数
  据库端口/RDS创建的数据库名 */
  table_name = "tabletest2" /* RDS数据下的表名 */
);
INSERT INTO
  car_info
SELECT
  13,
  'abc';
```

4. 单击“语义校验”确保SQL语义校验成功。单击“保存”，保存作业。单击“启动”，启动作业，确认作业参数信息，单击“立即启动”开始执行作业。
5. 等待作业运行完成，作业状态显示为“已完成”。
6. 登录RDS控制台，单击RDS数据库实例，单击创建的数据库名，如“test2”，在创建的表“tabletest2”所在行的“操作”列，单击“SQL查询”。
7. 在“SQL查询”界面，单击“执行SQL”，查看RDS表数据已写入成功。

图 3-38 RDS 表查询结果



3.6 典型场景示例：配置弹性资源池队列扩缩容策略

场景介绍

一个企业有多个部门，多个部门不同业务数据分析的时间段可能有所差异，具体场景如下：

- A部门：在00:00-09:00时间段内资源请求量大，其他时间段有短时间的资源请求量不大的任务运行。
- B部门：在10:00-22:00时间段内资源请求量大，其他时间段内也有固定周期的作业请求也需要保障。

针对上述场景，弹性资源池上可以添加两个队列，队列test_a用于运行A部门的作业任务，队列test_b运行B部门的作业任务。两个部门请求量大的任务时间段固定，则可以在test_a和test_b队列上分别添加两个时间段00:00-09:00和10:00-23:00的扩缩容策略，其他时间段的作业任务通过配置队列的默认扩缩容策略进行保障。

表 3-17 队列扩缩容策略

队列名	新增的扩缩容时间段	新增的扩缩容时间段优先级	新增的扩缩容时间段最小和最大CU	默认扩缩容时间段	默认时间段优先级	默认扩缩容时间段最小和最大CU	备注
test_a	[00:00, 09:00)	20	最小CU: 64 最大CU: 128	新增的扩缩容时间段 [00:00, 09:00)以外的时间段范围	5	最小CU: 16 最大CU: 32	运行A部门作业

队列名	新增的扩缩容时间段	新增的扩缩容时间段优先级	新增的扩缩容时间段最小和最大CU	默认扩缩容时间段	默认时间段优先级	默认扩缩容时间段最小和最大CU	备注
test_b	[10:00, 23:00)	20	最小CU: 64 最大CU: 128	新增的扩缩容时间段 [10:00, 23:00)以外的时间段范围	5	最小CU: 32 最大CU: 64	运行B部门作业

注意事项

- 建议对流批业务实施资源池的精细化管理，将Flink实时流类型的作业与SQL批处理类型的作业分别置于独立的弹性资源池中。

优势在于：Flink实时流任务具有常驻运行的特质，确保其稳定运行而不会强制扩容，进而避免任务中断和系统不稳定。

而SQL批处理类型的作业在独立的资源池中能够更加灵活地进行扩缩容，显著提升扩缩容的成功率和操作效率。
- 在全天的任意一个时间段内，弹性资源池中所有队列的最小CU数之和需要小于等于弹性资源池的最小CU数。
- 在全天的任意一个时间段内，弹性资源池中任意一个队列的最大CU必须小于等于弹性资源池的最大CU。
- 同一队列不同扩缩容策略的时间段区间不能有交集。
- 弹性资源池队列中的扩缩容策略时间段仅支持整点的时间段设置，并且包含设置的开启时间，不包含设置的结束时间，例如设置时间段00-09，则时间段范围为：[00:00, 09:00)。默认的扩缩容策略不支持时间段配置修改。
- 弹性资源池扩缩容策略生效规则为：在任意一个时间段周期内，优先满足所有队列的最小CU数。剩余的CU（弹性资源池最大CU-所有队列的最小CU数之和）则根据配置的优先级顺序分配：

 - 如果队列的优先级不同，根据配置的优先级顺序分配，直到剩余的CU数分配完成。
 - 如果队列的优先级相同，资源会被随机分配到某一队列，如果分配后资源还有剩余会随机分配到剩下的某一队列中，直到剩余的CU数分配完成。

表 3-18 弹性资源池扩缩容 CU 分配场景说明

场景	弹性资源池CU数分配说明
<p>弹性资源池当前最大CU为256CU，添加了两个队列，分别为队列A和队列B。两个队列设置的扩缩容策略如下：</p> <ul style="list-style-type: none"> • 队列A扩缩容策略：优先级5，时间段：00:00-9:00，最小CU是32，最大CU是128 • 队列B扩缩容策略：优先级10，时间段：00:00-9:00，最小CU是64，最大CU是128 	<p>到了00:00-9:00时间段：</p> <ol style="list-style-type: none"> 1. 弹性资源池优先满足两个队列的最小CU，队列A先分配32CU，队列B分配64CU，剩余CU数为160CU：弹性资源池的最大CU-两个队列的最小CU之和=256-32-64=160CU。 2. 剩余CU数根据优先级高低来分配，因为队列B的优先级高于队列A，则优先将64CU分配给队列B，剩余的96CU全部分配给队列A。
<p>弹性资源池当前最大CU为96CU，添加了两个队列，分别为队列A和队列B。两个队列设置的扩缩容策略如下：</p> <ul style="list-style-type: none"> • 队列A扩缩容策略：优先级5，时间段：00:00-9:00，最小CU是32，最大CU是64 • 队列B扩缩容策略：优先级10，时间段：00:00-9:00，最小CU是64，最大CU是128 	<p>到了00:00-9:00时间段：</p> <ol style="list-style-type: none"> 1. 弹性资源池优先满足两个队列的最小CU，队列A先分配32CU，队列B分配64CU，剩余CU数为0CU：弹性资源池的最大CU-两个队列的最小CU之和=96-32-64=0CU。 2. 因为剩余的CU数已经没有，则停止分配。
<p>弹性资源池当前最大CU为128CU，添加了两个队列，分别为队列A和队列B。两个队列设置的扩缩容策略如下：</p> <ul style="list-style-type: none"> • 队列A扩缩容策略：优先级5，时间段：00:00-9:00，最小CU是32，最大CU是64 • 队列B扩缩容策略：优先级10，时间段：00:00-9:00，最小CU是64，最大CU是128 	<p>到了00:00-9:00时间段：</p> <ol style="list-style-type: none"> 1. 弹性资源池优先满足两个队列的最小CU，队列A先分配32CU，队列B分配64CU，剩余CU数为32CU：弹性资源池的最大CU-两个队列的最小CU之和=128-32-64=32CU。 2. 按照优先级，则优先将剩余的32CU分配给B队列后停止分配。
<p>弹性资源池当前最大CU为128CU，添加了两个队列，分别为队列A和队列B。两个队列设置的扩缩容策略如下：</p> <ul style="list-style-type: none"> • 队列A扩缩容策略：优先级5，时间段：00:00-9:00，最小CU是32，最大CU是64 • 队列B扩缩容策略：优先级5，时间段：00:00-9:00，最小CU是64，最大CU是128 	<p>到了00:00-9:00时间段：</p> <ol style="list-style-type: none"> 1. 弹性资源池优先满足两个队列的最小CU，队列A先分配32CU，队列B分配64CU，剩余CU数为32CU：弹性资源池的最大CU-两个队列的最小CU之和=128-32-64=32CU。 2. 因为两个队列的优先级相同，则剩余32CU随机分配给两个队列。

弹性资源池队列扩缩容策略配置

- 步骤1** 登录DLI控制台，参考[创建弹性资源池并添加队列](#)创建一个最小CU数为128CU和最大CU数为256CU的弹性资源池。
- 步骤2** 单击“资源管理 > 弹性资源池”，在已创建的弹性资源池所在行的“操作”列单击“队列管理”。
- 步骤3** 参考[创建弹性资源池并添加队列](#)添加队列test_a，在添加队列扩缩容配置步骤里面添加扩缩容策略。
1. 设置默认的时间段优先级为5，最小CU为16，最大CU为32。
 2. 单击“新增”，添加一个优先级为20，时间段为：00--09，最小CU为64，最大CU为128。

图 3-39 添加队列 test_a 的扩缩容策略



- 步骤4** 添加完成后，可以在队列管理的界面看到队列test_a的扩缩容策略配置。

图 3-40 队列 test_a 扩缩容策略配置

名称	类型	时间段	最小CU	最大CU	优先级	执行引擎	创建时间	全量数据	所有者	操作
test_a	SQL引擎	[00:00, 09:00]	64	128	20	spark	2022-05-18 17:38:13 GMT+08:00	default		编辑 删除


单击结果图形化  按钮，可以看到队列test_a的优先级和所有时间段的CU设置。

图 3-41 队列 test_a 扩缩容策略结果图形化



步骤5 参考[创建弹性资源池并添加队列](#)添加队列test_b，在添加队列扩缩容配置步骤里面添加扩缩容策略。

1. 设置默认的时间段优先级为5，最小CU为32，最大CU为64。
2. 单击“新增”，添加一个优先级为20，时间段为：10--23，最小CU为64，最大CU为128。

图 3-42 添加队列 test_b 的扩缩容策略



步骤6 添加完成后，可以在队列管理的界面看到队列test_b的扩缩容策略配置。

图 3-43 队列 test_b 扩缩容策略配置

名称	类型	时间段	最小CU	最大CU	优先级	队列扩容	创建时间	创建用户	所有者	操作
test_a	SQL队列	(00:00, 09:00) (09:00, 24:00)	64	128	20	spark	2022/05/18 17:47:40	DBAT-00100	default	编辑 删除
test_b	SQL队列	(00:00, 24:00) (10:00, 23:00)	32	64	5	spark	2022/05/18 17:48:59	DBAT-00100	default	编辑 删除

单击结果图形化按钮，可以看到队列test_b和test_a所有时间段的优先级和CU设置。

图 3-44 test_b 和 test_a 所有时间段的优先级和 CU 设置



----结束

3.7 创建非弹性资源池队列（废弃，不推荐使用）

非弹性资源池模式的队列是DLI的上一代计算资源管理方式，按使用需求购买和释放资源，需要预先估计资源使用需求再进行购买。

优先推荐使用弹性资源池队列，提高资源使用的灵活性和资源利用效率。购买弹性资源池并在弹性资源池中添加队列请参考[创建弹性资源池并添加队列](#)。

说明

- 用户首次使用子账号创建队列时，需要先使用主账号登录控制台，在DLI的数据库中保持记录，才能创建队列。
- 新队列第一次运行作业时，需要一定的时间，通常为6~10分钟。
- 按需队列创建完成后，如果在1小时内未运行作业，系统将进行释放。
- 按需队列与包年/包月队列不能互相转换，如需使用包年/包月队列，直接购买即可。
- 16CUs队列不支持扩容和缩容。
- 64CUs队列不支持缩容。

约束限制

表 3-19 队列使用约束限制

限制项	说明
资源类型	<ul style="list-style-type: none"> • 队列类型： <ul style="list-style-type: none"> - default队列：DLI服务预置了名为“default”的队列供用户体验，资源的大小按需分配。运行作业时按照用户每个作业的数据扫描量（单位为“GB”）收取计算费用。 - SQL类型队列：SQL队列支持提交Spark SQL作业。 - 通用队列：支持Spark程序、Flink SQL、Flink Jar作业。 • 不支持队列类型切换，如需使用其他队列类型，请重新购买新的队列。
管理队列	<ul style="list-style-type: none"> • 不支持切换队列的计费模式。 • 队列不支持切换区域。 • 创建队列时（非弹性资源池模式的队列），仅支持包年包月队列和按需专属队列选择跨AZ双活，且跨AZ的队列价格为单AZ模式下的2倍。 • DLI队列不支持访问公网。
队列扩缩容	<ul style="list-style-type: none"> • 16CUs队列不支持扩容和缩容。 • 64CUs队列不支持缩容。 • 新创建的队列需要运行作业后才可进行扩缩容。

创建队列步骤


1. 创建队列的操作入口有三个，分别在“总览”页面、“SQL编辑器”页面和“队列管理”页面。
 - 单击总览页面右上角“购买队列”进行创建队列。
 - 在“队列管理”页面创建队列。
 - i. 在DLI管理控制台的左侧导航栏中，选择“资源管理 > 队列管理”。
 - ii. 单击“队列管理”页面右上角“购买队列”进行创建队列。
 - 在“SQL编辑器”页面创建队列。
 - i. 在DLI管理控制台的左侧导航栏中，选择“SQL编辑器”。
 - ii. 单击“队列”切换到该页签，单击右侧的  创建队列。
2. 在“购买队列”页面，参见表3-20设置相关参数。

表 3-20 参数说明

参数名称	描述
计费模式	<ul style="list-style-type: none"> ● 包年/包月 该计费模式的队列为专属队列。 ● 按需计费：建议购买cu时套餐包享受优惠。
区域	选择所在的区域。不同区域的云服务之间内网互不相通；请就近选择靠近您业务的区域，可减少网络时延，提高访问速度。
项目	每个区域默认对应一个项目，这个项目由系统预置。
名称	队列的名称。 <ul style="list-style-type: none"> ● 只能包含数字、英文字母和下划线，但不能是纯数字，不能以下划线开头，且不能为空。 ● 输入长度不能超过128个字符。 说明 队列名称不区分大小写，系统会自动转换为小写。
类型	<ul style="list-style-type: none"> ● SQL队列：SQL作业的计算资源。 ● 通用队列：Spark作业、Flink作业的计算资源。 说明 专属队列是指队列对应的资源为专属资源，空闲时不释放，即无论是否使用均保留资源的队列类型。专属队列可以保证提交作业时资源一定存在。专属队列支持创建增强型跨源。 购买按需队列时可以选择专属队列。专属队列无论是否使用，24小时均收费。包年包月队列为专属队列。
AZ策略	采用双AZ策略创建的队列，当某个AZ不可用时，仍然能够从其他AZ正常访问数据，适用于对可用性要求较高的场景。 说明 <ul style="list-style-type: none"> ● 当前包年/包月计费模式和按需计费专属资源模式支持双AZ策略。 ● 双AZ能提高您的数据可用性。 ● 双AZ属性一旦启用，后续无法修改。

参数名称	描述
CPU架构	<ul style="list-style-type: none"> • X86 • 鲲鹏
规格	<p>队列规格指的是计算节点所有CU数的总和，1CU=1核4GB。DLI系统会自动分配各计算节点的内存和CPU大小，具体计算节点个数客户端不感知。</p> <p>选择“包年/包月”计费模式时，可选择“固定规格”，也可以“自定义规格”。“按需计费”只支持选择固定规格。</p> <p>请按需选择队列规格。队列规格指的是计算节点所有CU数的总和，DLI系统会自动分配各计算节点的内存和CPU大小，具体计算节点个数客户端不感知。</p> <ul style="list-style-type: none"> • 固定规格包括“16CUs”、“64CUs”、“256CUs”、“512CUs”。 • 自定义规格可以选择“128CUs”、“192CUs”、“256CUs”、“320CUs”、“384CUs”等“64CUs”的倍数。 <p>说明</p> <ul style="list-style-type: none"> • 当剩余CU配额小于上述规格时，则不能创建队列。 • 如果需要购买其他规格的队列，则可以先创建上述某一规格队列后，通过如下操作实现： “包年/包月”队列：则可以通过变更普通队列规格操作来实现。 “按需计费”队列：则可以通过普通队列弹性扩缩容操作来实现。
购买时长	<p>选择“包年/包月”计费模式时，需要选择“购买时长”。购买时长越长，优惠越多。可勾选“自动续费”，按月购买，自动续费周期为1个月。按年购买，自动续费周期为1年。</p>
企业项目	<p>如果所建队列属于企业项目，可选择对应的企业项目。</p> <p>企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理。</p> <p>关于如何设置企业项目请参考《企业管理用户指南》。</p> <p>说明 只有开通了企业管理服务的用户才显示该参数。</p>
描述	<p>所创建队列的相应描述。输入长度不能超过128个字符。</p>

参数名称	描述
高级选项	<p>仅在选择“包年/包月”计费模式，或在“按需计费”模式中，勾选“专属资源模式”时，支持配置“高级配置”。</p> <p>在“队列类型”中，勾选了“专属资源模式”后，需要选择“高级选项”。</p> <ul style="list-style-type: none"> 默认配置：由系统自动配置。 自定义配置： <ul style="list-style-type: none"> “网段”：支持指定使用的网段范围。如需使用DLI增强型跨源，DLI队列网段与数据源网段不能重合。 建议使用网段： <ul style="list-style-type: none"> 10.0.0.0~10.255.0.0/8~24 172.16.0.0~172.31.0.0/12~24 192.168.0.0~192.168.0.0/16~24 “队列特性”：运行AI相关SQL作业时选择“AI增强型”队列，运行其他作业时选择“基础型”队列。
标签	<p>使用标签标识云资源。包括标签键和标签值。如果您需要使用同一标签标识多种云资源，即所有服务均可在标签输入框下拉选择同一标签，建议在标签管理服务（TMS）中创建预定义标签。</p> <p>如您的组织已经设定DLI的相关标签策略，则需按照标签策略规则为资源添加标签。标签不符合标签策略的规则，则可能会导致资源创建失败，请联系组织管理员了解标签策略详情。</p> <p>具体请参考《标签管理服务用户指南》。</p> <p>说明</p> <ul style="list-style-type: none"> 最多支持20个标签。 一个“键”只能添加一个“值”。 每个资源中的键名不能重复。 标签键：在输入框中输入标签键名称。 <p>说明</p> <p>标签的键的最大长度为128个字符，标签的键可以包含任意语种字母、数字、空格和_ . : + - @ ,但首尾不能含有空格，不能以_sys_开头。</p> <ul style="list-style-type: none"> 标签值：在输入框中输入标签值。 <p>说明</p> <p>标签值的最大长度为255个字符，标签的值可以包含任意语种字母、数字、空格和_ . : + - @ 。</p>

- 单击“立即购买”，确认配置。

说明

- 第一次创建队列时，需要勾选“同意以上隐私协议”并确定。
- 配置确认无误，单击“提交”完成队列创建。
如果队列名称已存在，单击“提交”时，系统会提示“Queue xxx already exists”错误，可返回“上一步”进行修改。
- 队列创建成功后，您可以在“队列管理”页面查看和选择使用对应的队列。

 **说明**

新队列第一次运行作业时，需要一定的时间，通常为6~10分钟。

4 创建数据库和表

4.1 了解数据目录、数据库和表

数据库和表是SQL作业、Spark作业场景开发的基础，在执行作业前您需要根据业务场景定义数据库和表。

说明

Flink支持动态数据类型，可以在运行时定义数据结构，不需要事先定义元数据。

数据目录

数据目录（Catalog）是元数据管理对象，它可以包含多个数据库。

在DLI数据目录库下创建数据库和表请参考[在DLI控制台创建数据库和表](#)。

数据库

数据库是按照数据结构来组织、存储和管理数据的建立在计算机存储设备上的仓库。数据库通常用于存储、检索和管理结构化数据，由多个数据表组成，这些数据表通过键和索引相互关联。

表

表是数据库最重要的组成部分之一，它由行和列组成。每一行代表一个数据项，每一列代表数据的一个属性或特征。表用于组织和存储特定类型的数据，使得数据可以被有效地查询和分析。

数据库是一个框架，表是其实质内容。一个数据库包含一个或者多个表。

用户可通过管理控制台或SQL语句创建数据库和表，其中SQL语句的操作方法请参见[创建数据库](#)、[创建OBS表](#)和[创建DLI表](#)等。本章节介绍在管理控制台创建数据库和表的操作步骤。

说明

创建数据库和表时，有权限控制，需要对其他用户授权，其他用户才可查看该用户新建的数据库和表。

表的元数据

元数据（Metadata）是用来定义数据类型的数据。主要是描述数据自身信息，包含源、大小、格式或其它数据特征。数据库字段中，元数据用于诠释数据仓库的内容。

创建表时，会定义元数据，由列名、类型、列描述三列组成。

DLI 支持创建的表类型

- **DLI表**

DLI表是存储在DLI数据湖中的数据表。支持多种数据格式，可以存储结构化、半结构化和非结构化数据。

DLI表的数据存储在DLI服务内部，查询性能更好，适用于对时延敏感类的业务，如交互类的查询等。

库表管理中表的列表页面，表类型为**Managed**的即代表**DLI表**。

- **OBS表**

OBS表的数据存储在OBS上，适用于对时延不敏感的业务，如历史数据统计分析等。

OBS表通常以对象的形式存储数据，每个对象包含数据和相关的元数据。

库表管理中表的列表页面，表类型为**External**，**存储位置为OBS路径**的即代表**OBS表**。

- **视图表**

视图表（View）是一种虚拟表，它不存储实际的数据，而是根据定义的查询逻辑动态生成数据。视图通常用于简化复杂的查询，或者为不同的用户或应用提供定制化的数据视图。

视图表可以基于一个或多个表创建，提供了一种灵活的方式来展示数据，而不影响底层数据的存储和组织。

库表管理中表的列表页面，表类型为**View**的即代表**视图表**。

说明

View只能通过SQL语句进行创建，不能通过“创建表”页面进行创建。视图中包含的表或视图信息不可被更改，如有更改可能会造成查询失败。

- **跨源表**

跨源表是指能够跨越多个数据源进行查询和分析的数据表。这种表可以整合来自不同数据源的数据，提供统一的数据视图。

跨源表常用于数据仓库和数据湖架构中，允许用户执行跨多个数据源的复杂查询。

库表管理中表的列表页面，表类型为**External**，**存储位置非OBS路径**的即代表**跨源表**。

数据库和表的约束与限制

表 4-1 DLI 资源相关约束限制

限制项	说明
数据库	<ul style="list-style-type: none">“default”为内置数据库，不能创建名为“default”的数据库。DLI支持创建的数据库的最大数量为50个。
数据表	<ul style="list-style-type: none">DLI支持创建的表的最大数量为5000个。DLI支持创建表类型：<ul style="list-style-type: none">Managed：数据存储位置为DLI的表。External：数据存储位置为OBS的表。View：视图，视图只能通过SQL语句创建。跨源表：表类型同样为External。创建DLI表时不支持指定存储路径。
数据导入	<ul style="list-style-type: none">仅支持将OBS上的数据导入DLI或OBS中。支持将OBS中CSV, Parquet, ORC, JSON和Avro格式的数据导入到在DLI中创建的表。将CSV格式数据导入分区表，需在数据源中将分区列放在最后一列。导入数据的编码格式仅支持UTF-8。
数据导出	<ul style="list-style-type: none">只支持将DLI表（表类型为“Managed”）中的数据导出到OBS桶中，且导出的路径必须指定到文件夹级别。导出文件格式为json格式，且文本格式仅支持UTF-8。支持跨账号导出数据，即B账户对A账户授权后，A账户拥有B账户OBS桶的元数据信息和权限信息的读取权限，以及路径的读写权限，则A账户可将数据导出至B账户的OBS路径中。

表管理页面

在“数据管理”页面中，单击对应数据库名称或“操作”列中的“表管理”，可进入其表管理页面。

表管理页面显示用户在当前数据库中创建所有的表，您可以查看表类型，数据存储位置等信息。表列表默认按创建时间排列，创建时间最近的表显示在最前端。

4.2 在 DLI 控制台创建数据库和表

数据库是按照数据结构来组织、存储和管理数据的建立在计算机存储设备上的仓库。

表是数据库最重要的组成部分之一。表是由行与列组合成的。每一列被当作是一个字段。每个字段中的值代表一种类型的数据。

数据库是一个框架，表是其实质内容。一个数据库包含一个或者多个表。

用户可通过管理控制台或SQL语句创建数据库和表，其中SQL语句的操作方法请参见[创建数据库](#)、[创建OBS表](#)和[创建DLI表](#)等。本章节介绍在管理控制台创建数据库和表的操作步骤。

说明


View只能通过SQL语句进行创建，不能通过“创建表”页面进行创建。

注意事项

- 当OBS的目录下有同名文件夹和文件时，创建OBS表指向该路径会优先指向文件而非文件夹。
- 创建数据库和表时，有权限控制，需要对其他用户授权，其他用户才可查看该用户新建的数据库和表。具体请参考[常用操作与系统权限关系](#)。

创建数据库

步骤1 创建数据库的入口有两个，分别在“数据管理”和“SQL编辑器”页面。

- 在“数据管理”页面创建数据库。
 - a. 在管理控制台左侧，单击“数据管理” > “库表管理”。
 - b. 在库表管理页面右上角，单击“创建数据库”可创建数据库。
- 在“SQL编辑器”页面创建数据库。
 - a. 在管理控制台左侧，单击“SQL编辑器”。
 - b. 在左侧导航栏单击“数据库”页签右侧  可创建数据库。

步骤2 在“创建数据库”页面，参见[表4-2](#)输入数据库名称和描述信息。

图 4-1 库表管理-创建数据库

创建数据库
×

您还可以创建1个数据库。 [申请扩大配额。](#)

* 数据库名称

描述

0/128

* 企业项目 C ? 新建企业项目

如果您需要使用同一标签识别多种云资源，即所有服务均可在标签输入框下拉选择同一标签，建议在TMS中创建预定义标签。 [查看预定义标签](#) C

在下方键/值输入框输入内容后单击'添加'，即可将标签加入此处

标签

您还可以添加10个标签。

表 4-2 参数说明

参数名称	描述
数据库名称	<ul style="list-style-type: none"> ● 数据库名称只能包含数字、英文字母和下划线，但不能是纯数字，且不能以下划线开头。 ● 数据库名称大小写不敏感且不能为空。 ● 输入长度不能超过128个字符。 <p>说明 “default”为内置数据库，不能创建名为“default”的数据库。</p>
企业项目	<p>如果所建队列属于企业项目，可选择对应的企业项目。</p> <p>企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理。</p> <p>关于如何设置企业项目请参考《企业管理用户指南》。</p> <p>说明 只有开通了企业管理服务的用户才显示该参数。</p>
描述	该数据库的描述。

参数名称	描述
标签	<p>使用标签标识云资源。包括标签键和标签值。如果您需要使用同一标签标识多种云资源，即所有服务均可在标签输入框下拉选择同一标签，建议在标签管理服务（TMS）中创建预定义标签。</p> <p>如您的组织已经设定DLI的相关标签策略，则需按照标签策略规则为资源添加标签。标签不符合标签策略的规则，则可能会导致资源创建失败，请联系组织管理员了解标签策略详情。</p> <p>具体请参考《标签管理服务用户指南》。</p> <p>说明</p> <ul style="list-style-type: none"> • 最多支持20个标签。 • 一个“键”只能添加一个“值”。 • 每个资源中的键名不能重复。 • 标签键：在输入框中输入标签键名称。 <p>说明</p> <p>标签的键的最大长度为128个字符，标签的键可以包含任意语种字母、数字、空格和_ . : + - @ , 但首尾不能含有空格，不能以_sys_开头。</p> <ul style="list-style-type: none"> • 标签值：在输入框中输入标签值。 <p>说明</p> <p>标签值的最大长度为255个字符，标签的值可以包含任意语种字母、数字、空格和_ . : + - @ 。</p>

步骤3 单击“确定”，完成数据库创建。

数据库创建成功后，您可以在“库表管理”页面或者“SQL编辑器”页面查看和选择使用对应的数据库。

----结束

创建表



创建表前，请确保已创建数据库。

步骤1 创建表的入口有两个，分别在“数据管理”和“SQL编辑器”页面。

📖 说明

此处创建表的方式不支持创建View，HBase（CloudTable/MRS）表、OpenTSDB（CloudTable/MRS）表、DWS表、RDS表和CSS表等跨源连接表。可通过SQL方式创建View和跨源连接表，具体请参考《[数据湖探索SQL语法参考](#)》。

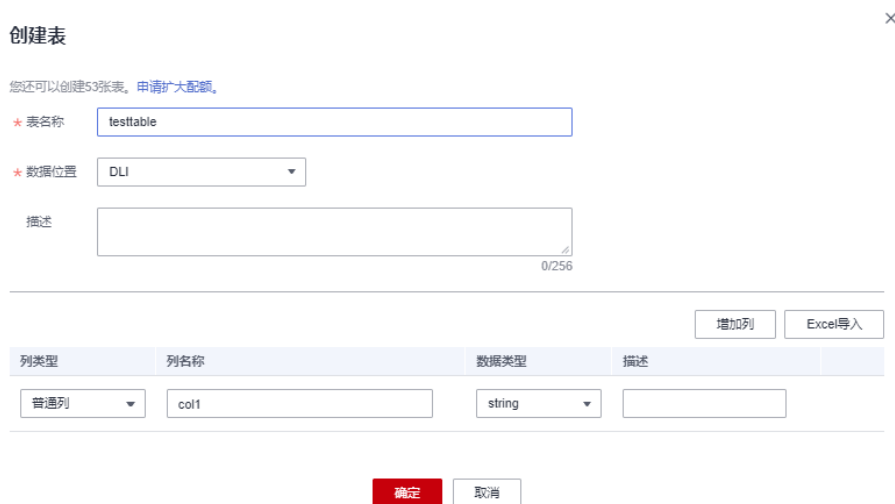
- 在“数据管理”页面创建表。
 - a. 在管理控制台左侧，单击“数据管理”>“库表管理”。
 - b. 在库表管理页面中，选择需要建表的数据库。在其“操作”栏中，单击“更多”>“创建表”，可创建当前数据库下的表。
- 在“SQL编辑器”页面创建表。
 - a. 在管理控制台左侧，单击“SQL编辑器”。
 - b. 在“SQL编辑器”页面的左侧导航栏单击“数据库”页签。有两种方式创建表。

- 鼠标左键单击数据库名，进入“表”区域，单击右侧 ，创建当前数据库下的表。
- 鼠标左键单击对应数据库右侧的 ，在列表菜单中选择“创建表”，创建当前数据库下的表。

步骤2 在“创建表”页面，填写参数。

- 当数据位置为DLI时，请参见表4-3填写相关参数；

图 4-2 创建表-DLI



创建表 ×

您还可以创建53张表。申请扩大配额。

* 表名称

* 数据位置

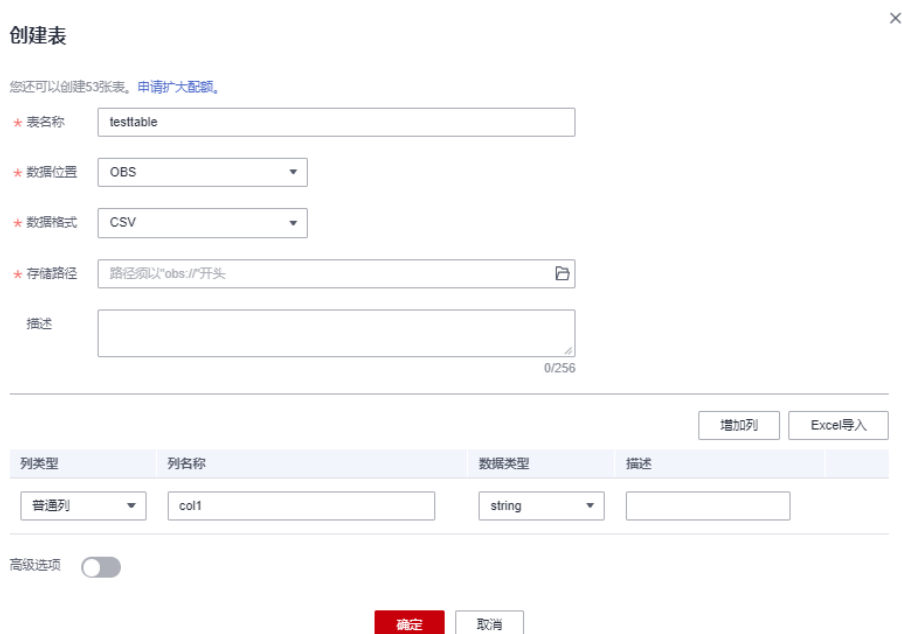
描述

0/256

列类型	列名称	数据类型	描述
普通列	col1	string	

- 当数据位置为OBS时，请参见表4-3和表4-4填写相关参数。

图 4-3 创建表-OBS



创建表 ×

您还可以创建53张表。申请扩大配额。

* 表名称

* 数据位置

* 数据格式

* 存储路径

描述

0/256

列类型	列名称	数据类型	描述
普通列	col1	string	

高级选项

表 4-3 通用参数说明

参数名称	描述	示例
表名称	<ul style="list-style-type: none"> - 表名称只能包含数字、英文字母和下划线，但不能是纯数字，且不能以下划线开头。 - 表名称大小写不敏感且不能为空。 - 表名称支持包含“\$”符号。例如：\$test。 - 输入长度不能超过128个字符。 	table 01
数据位置	数据存储位置，当前支持DLI和OBS。	DLI
描述	该表的描述。	-
列类型	选择为“普通列”或“分区列”。	普通列
列名称	<p>表的列名。列名应至少包含一个字母，并允许下划线（_），但不支持纯数字。</p> <p>可选择“普通列”或“分区列”。“分区列”是分区表专用的，对用户数据进行分区，可提高查询效率。</p> <p>说明 列名不区分大小写，不能相同。</p>	name
数据类型	<p>与“列名”对应，表示该列的数据类型。</p> <ul style="list-style-type: none"> - 字符串（string）：字符串类型。 - 有符号整数（int）：存储空间为4字节。 - 日期类型（date）：所表示日期的范围为0000-01-01 to 9999-12-31。 - 双精度浮点型（double）：存储空间为8字节。 - 布尔类型（boolean）：存储空间为1字节。 - 固定有效位数和小数位数的数据类型（decimal）：有效位数为1~38之间的正整数，包含1和38；小数位数为小于10的整数。 - 有符号整数（smallint/short）：存储空间为2字节。 - 有符号整数（bigint/long）：存储空间为8字节。 - 时间戳（timestamp）：表示日期和时间，可达到小数点后6位。 - 单精度浮点型（float）：存储空间为4字节。 - 有符号整数（tinyint）：存储空间为1字节。仅OBS表支持。 	string
列描述	该列的描述。	-
操作	<ul style="list-style-type: none"> - 增加列 - 删除列 <p>说明 当列数较多时，建议您使用SQL语句创建表，或直接从本地Excel导入列信息。</p>	-

表 4-4 数据位置为 OBS 的参数说明

参数名称	描述	示例
数据格式	支持以下数据格式。 <ul style="list-style-type: none"> - Parquet: DLI支持读取不压缩、snappy压缩、gzip压缩的parquet数据。 - CSV: DLI支持读取不压缩、gzip压缩的csv数据。 - ORC: DLI支持读取不压缩、snappy压缩的orc数据。 - JSON: DLI支持读取不压缩、gzip压缩的json数据。 - Avro: DLI支持读取不压缩的avro数据。 	CSV
存储路径	输入或选择OBS路径。路径可以是文件夹，也可以是路径。 说明 如果需要将OBS上的数据导入OBS表，则建表路径必须是文件夹。建表路径是文件时，会导致数据导入失败。	obs://obs1/ sampledata.csv
表头:无/有	当“数据格式”为“CSV”时，该参数有效。设置导入数据源是否含表头。 选中“高级选项”，勾选“表头:无”前的方框，“表头:无”显示为“表头:有”，表示有表头；取消勾选即为“表头:无”，表示无表头。	-
自定义分隔符	当“数据格式”为“CSV”，并在自定义分隔符前的方框打勾时，该参数有效。 选中高级选项，支持选择如下分隔符。 <ul style="list-style-type: none"> - 逗号(,) - 竖线() - 制表符(\t) - 其他: 输入自定义分隔符 	逗号(,)
自定义引用字符	当“数据格式”为“CSV”，并在自定义引用字符前的方框打勾时，该参数有效。 选中高级选项，支持选择如下引用字符。 <ul style="list-style-type: none"> - 单引号(') - 双引号(") - 其他: 输入自定义引用字符 	单引号(')
自定义转义字符	当“数据格式”为“CSV”，并在自定义转义字符前的方框打勾时，该参数有效。 选中高级选项，支持选择如下转义字符。 <ul style="list-style-type: none"> - 反斜杠(\) - 其他: 输入自定义转义字符 	反斜杠(\)

参数名称	描述	示例
日期格式	当“数据格式”为“CSV”和“JSON”时此参数有效。 选中“高级选项”，该参数表示表中日期的格式，默认格式为“yyyy-MM-dd”。日期格式字符定义详见 加载数据 中的“表3 日期及时间模式字符定义”。	2000-01-01
时间戳格式	当“数据格式”为“CSV”和“JSON”时此参数有效。 选中“高级选项”，该参数表示表中时间戳的格式，默认格式为“yyyy-MM-dd HH:mm:ss”。时间戳格式字符定义详见 加载数据 中的“表3 日期及时间模式字符定义”。	2000-01-01 09:00:00

步骤3 单击“确定”，完成表创建。

表创建成功后，您可以在“表管理”页面或者“SQL编辑器”页面查看和选择使用对应的表。

步骤4 （可选）DLI表创建完成后，您可以选择是否直接向该表导入数据。

----结束

4.3 查看表元数据


元数据说明

- 元数据（Metadata）是用来定义数据类型的数据。主要是描述数据自身信息，包含源、大小、格式或其它数据特征。数据库字段中，元数据用于诠释数据仓库的内容。
- 创建表时，会定义元数据，由列名、类型、列描述三列组成。
- “元数据”页面将显示目标表的列名、列类型、类型和描述。

查看元数据步骤

查看元数据的入口有两个，分别在“数据管理”和“SQL编辑器”页面。

- 在“数据管理”页面查看元数据。
 - 在管理控制台左侧，单击“数据管理”>“库表管理”。
 - 单击需导出数据对应数据库名称，进入该数据库“表管理”页面。
 - 单击目标表“操作”栏中的“更多”，选择“表属性”，即可在“元数据”页签查看该表的元数据信息。
- 在“SQL编辑器”页面查看元数据。
 - 在管理控制台左侧，单击“SQL编辑器”。
 - 在“SQL编辑器”页面的左侧导航栏中，选择“数据库”页签。
 - 单击对应数据库名，进入该数据库的表列表。

- d. 鼠标左键单击对应表右侧的 ，在列表菜单中选择“表属性”，即可在“元数据”页签查看该表的元数据信息。

4.4 在 DLI 控制台管理数据库资源

4.4.1 在 DLI 控制台配置数据库权限

数据库权限操作场景

- 针对不同用户，可以通过权限设置分配不同的数据库权限。
- 管理员用户和数据库的所有者拥有所有权限，不需要进行权限设置且其他用户无法修改其数据库权限。
- 给新用户设置数据库权限时，该用户所在用户组的所属区域需具有Tenant Guest 权限。关于Tenant Guest 权限的介绍和开通方法，详细参见《[权限策略](#)》和《[统一身份认证服务用户指南](#)》中的[创建用户组](#)。

注意事项

- 如果需要查看管理员或者其他用户账号下的数据库，需要对当前用户授权（显示权限），具体请参考[常用操作与系统权限关系](#)。
- 数据库和表赋权对象具有层级关系，用户赋予上一层级的权限会自动继承到下一层级对象上，层级关系为：数据库>表>列。
- 数据库所有者、表所有者、被赋予“赋权权限”的用户都可以对数据库和表赋权。
- 列只能继承查询权限。“可继承权限”详细信息请参见[在DLI控制台配置数据库权限](#)。
- 回收权限时，只能在初始赋权的层级上回收。在哪一层赋权的，在哪一层进行权限回收。赋予权限和回收权限需要在同一层级操作。例如：在数据库上给用户赋予插入权限，那么在数据库下面的表就有了插入权限，回收这个插入权限，只能在数据库上回收，不能在表上回收。
- 如果数据库被删除后，再重新创建同名的数据库，数据库权限不会继承，需要对操作该数据库的用户或项目重新进行赋权。
例如，testdb数据库给用户A赋予了删除数据库的权限，后续执行了删除testdb数据库，并重新创建了testdb数据库。如果希望A用户继续保留删除testdb数据库的权限，则需要重新对A用户赋予该权限。

查看数据库权限

1. 在管理控制台左侧，单击“数据管理”>“库表管理”。
2. 单击所选数据库“操作”栏中的“权限管理”，将显示该数据库对应的权限信息。
权限设置有3种场景：为新用户或项目赋予权限、为已有权限的用户或项目修改权限、回收某用户或项目具备的所有权限。

为新用户或项目赋予权限

为新用户或新项目赋予权限，新用户或新项目指之前不具备此数据库权限的用户或项目。

1. 在数据库权限管理页面右上角单击“授权”。
2. 在“授权”弹出框中，选择“用户授权”或“项目授权”，填写需要授权的用户名或选择需要授权的项目，选择相应的权限。具体权限说明请参考表4-5。

图 4-4 数据库用户授权

The screenshot shows a dialog box titled "授权" (Authorization) with a close button (X) in the top right corner. It features two tabs: "用户授权" (User Authorization) and "项目授权" (Project Authorization), with "用户授权" selected. Below the tabs is a text input field labeled "* 用户名" (Username) with the placeholder text "请输入要被授权的IAM用户名称" (Please enter the IAM user name to be authorized). Underneath, there is a section "请勾选需要赋予的权限" (Please check the permissions to be granted) divided into "不可继承权限" (Non-inheritable permissions) and "可继承权限" (Inheritable permissions). Each section contains a list of permissions with checkboxes, such as "全选" (Select all), "显示所有表" (Show all tables), "创建表" (Create table), etc. At the bottom right, there are two buttons: "确定" (Confirm) and "取消" (Cancel).

图 4-5 数据库项目授权

The screenshot shows a dialog box titled "授权" (Authorization) with a close button (X) in the top right corner. It features two tabs: "用户授权" (User Authorization) and "项目授权" (Project Authorization), with "项目授权" selected. Below the tabs is a dropdown menu labeled "* 项目" (Project). Underneath, there is a section "请勾选需要赋予的权限" (Please check the permissions to be granted) divided into "不可继承权限" (Non-inheritable permissions) and "可继承权限" (Inheritable permissions). Each section contains a list of permissions with checkboxes, such as "全选" (Select all), "删除数据库" (Delete database), "创建表" (Create table), etc. At the bottom right, there are two buttons: "确定" (Confirm) and "取消" (Cancel).

表 4-5 参数说明

参数	描述
授权对象	选择“用户授权”或“项目授权”。
用户名或项目名	<ul style="list-style-type: none"> 选择“用户授权”时，输入数据库新增用户对应的IAM用户名称。 说明 该用户名称是已存在的IAM用户名称且该用户登录过DLI管理控制台。 选择“项目授权”时，选择当前区域下需要授权的项目。 说明 选择“项目授权”时： <ul style="list-style-type: none"> 如果赋权选择不可继承权限，则在该项目中无法查看对应数据库中表的信息。 如果赋权选择可继承权限，则在该项目中可查看该数据库内所有表的信息。
非继承权限	<p>选中权限即对用户或项目进行赋权，取消勾选即对用户权限或项目权限进行回收。</p> <p>非继承权限只作用于当前数据库。</p> <ul style="list-style-type: none"> 同时适用于“用户授权”和“项目授权”的权限包括： <ul style="list-style-type: none"> 删除数据库：删除当前数据库。 创建表：在当前数据库创建表。 创建视图：在当前数据库创建视图。 将SQL语句解释为执行计划：执行explain语句。 创建角色：在当前数据库创建角色。 删除角色：删除当前数据库中的角色。 显示角色：显示当前用户的角色。 绑定角色：在当前数据库绑定角色。 解除角色绑定：在当前数据库解除角色绑定。 显示所有角色和用户的绑定关系：显示所有角色和用户的绑定关系。 创建函数：在当前数据库创建函数。 删除函数：删除当前数据库中的函数。 显示所有函数：显示当前数据库中的所有函数。 显示函数详情：显示当前函数详情。 只适用于“用户授权”的权限包括： <ul style="list-style-type: none"> 显示所有表：显示当前数据库下的所有表。 说明 没有授权“显示所有表”权限，则该数据库在库表管理中不显示该数据库下的所有表。 显示数据库：显示当前数据库的信息。 说明 没有授权“显示数据库”权限，则该数据库在库表管理中不显示。

参数	描述
继承权限	<p>选中权限即对用户进行赋权，取消勾选即对用户权限进行回收。</p> <p>继承权限可作用到当前数据库及其所有的表上，但是表中的列只能继承其中的查询权限。</p> <p>以下权限同时适用于“用户授权”和“项目授权”。</p> <ul style="list-style-type: none"> ● 删除表：删除数据库下的表。 ● 查询表：在当前表内查询。 ● 显示表信息：显示当前表的信息。 ● 插入：在当前表内插入数据。 ● 添加列：在当前表中增加列。 ● 重写：在当前表内插入覆盖数据。 ● 赋权：用户可将数据库的权限赋予其他用户或项目。 ● 回收：用户可回收其他用户或项目具备的此数据库的权限，但是不能回收数据库所有者的权限。 ● 添加分区：在分区表中添加新的分区。 ● 删除分区：删除分区表中已有的分区。 ● 设置分区的路径：将分区表中的某个分区路径设置为用户指定的OBS路径。 ● 重命名表分区：对分区表中的分区重新命名。 ● 重命名表：对表重新命名。 ● 恢复表分区：从文件系统中导出分区信息保存到元数据中。 ● 显示所有分区：显示分区表中的所有分区。 ● 查看其他用户具备的权限：查看其他用户或项目具备的当前数据库的权限。

3. 单击“确定”，完成授权。

为已有权限的用户或项目修改权限

某用户或项目已具备此数据库的一些权限时，可为此用户或项目赋予或取消权限。

说明

当“权限设置”中的选项为灰色时，表示对应账号不具备修改此数据库的权限。可以向管理员用户、数据库所有者等具有赋权权限的用户申请数据库的“赋权”和数据库权限的“回收”权限。

1. 在“用户权限信息”列表中找到需要设置权限的用户：
 - 如果用户为子用户，可进行“权限设置”。
 - 如果用户为管理员用户，只能查看“权限信息”。

在“项目权限信息”列表中找到需要设置权限的项目，进行“权限设置”。
2. 在子用户或项目的“操作”栏中单击“权限设置”，可弹出数据库“权限设置”对话框。
数据库用户或项目详细的权限描述请参考[表4-5](#)。

3. 单击“确定”，完成权限设置。

回收某用户或项目具备的所有权限

回收某用户具备的所有权限，或回收某项目具备的所有权限。

- 在“用户权限信息”区域的用户列表中，选择需要回收权限的子用户，在“操作”栏中单击“回收”，在“回收用户权限”对话框中单击“确定”后，此用户将不具备数据库的任意权限。

📖 说明

用户为管理员用户时，“回收”为灰色，表示不可回收该用户的权限。

- 在“项目权限信息”区域的项目列表中，选择需要回收权限的项目，在“操作”栏中单击“回收”，在“回收项目权限”对话框中单击“确定”后，此项目将不具备数据库的任意权限。

4.4.2 在 DLI 控制台删除数据库

根据实际使用情况，您可以通过删除数据库。

注意事项

- 具有正在运行中的作业的数据库或者表不能删除。
- 管理员用户、数据库的所有者和具有删除数据库权限的用户可以删除数据库。

📖 说明

数据库和表删除后，将不可恢复，请谨慎操作。

删除数据库

1. 在管理控制台左侧，单击“数据管理”>“库表管理”。
2. 单击需要删除的数据库“操作”栏中的“更多 > 删除数据库”。

📖 说明

需要删除的数据库中含有表时，不能执行删除操作。需要先删除其中的表。

3. 在弹出的确认对话框中，单击“是”。


4.4.3 在 DLI 控制台修改数据库所有者

在实际使用过程中，开发人员创建了数据库和表，交给测试人员进行测试，测试人员测试完成后，再交给运维人员进行体验，在这种情况下，可以通过修改数据库的所有者，将数据转移给其他所有者。

修改数据库所有者

修改数据库所有者的入口有两个，分别在“数据管理”和“SQL编辑器”页面。

- 在“数据管理”页面修改数据库所有者。
 - a. 在管理控制台左侧，单击“数据管理”>“库表管理”。
 - b. 在“库表管理”页面选中需要修改的数据库，单击“操作”栏中的“更多 > 修改数据库”。

- c. 在弹出的对话框中，输入新的所有者用户名（已存在的用户名），单击“确定”。
- 在“SQL编辑器”页面修改数据库所有者。
 - a. 在管理控制台左侧，单击“SQL编辑器”。
 - b. 在左侧导航栏单击选择“数据库”页签，鼠标左键单击对应数据库右侧的，在列表菜单中选择“修改数据库”。
 - c. 在弹出的确认对话框中，输入新的所有者用户名（已存在的用户名），单击“确定”。

4.4.4 库表管理标签管理

标签管理

标签是用户自定义的、用于标识云资源的键值对，它可以帮助用户对云资源进行分类和搜索。标签由标签“键”和标签“值”组成。如果用户在其他云服务中使用了标签，建议用户为同一个业务所使用的云资源创建相同的标签键值对以保持一致性。

如您的组织已经设定DLI的相关标签策略，则需按照标签策略规则为资源添加标签。标签不符合标签策略的规则，则可能会导致资源创建失败，请联系组织管理员了解标签策略详情。

DLI支持以下两类标签：

- 资源标签：在DLI中创建的非全局的标签。
- 预定义标签：在标签管理服务（简称TMS）中创建的预定义标签，属于全局标签。

有关预定义标签的更多信息，请参见《[标签管理服务用户指南](#)》。

本节操作介绍如何为数据库和数据表添加标签、修改标签和删除标签。

数据库标签管理

- 步骤1** 在DLI管理控制台的左侧导航栏中，单击“数据管理> 库表管理”。
- 步骤2** 在对应数据库的操作列，选择“更多> 标签”。
- 步骤3** 进入标签管理页面，显示当前数据库的标签信息。
- 步骤4** 单击“添加/编辑标签”，弹出“添加/编辑标签”对话框，配置参数。
输入框输入内容后单击'添加'，将标签添加到输入框中。

图 4-6 数据库添加/编辑标签



表 4-6 标签配置参数

参数	参数说明
标签键	<p>您可以选择：</p> <ul style="list-style-type: none"> 在输入框的下拉列表中选择预定义标签键。 如果添加预定义标签，用户需要预先在标签管理服务中创建好预定义标签，然后在“标签键”的下拉框中进行选择。用户可以通过单击“查看预定义标签”进入标签管理服务的“预定义标签”页面，然后单击“创建标签”来创建新的预定义标签。 具体请参见《标签管理服务用户指南》中的“创建预定义标签”章节。 在输入框中输入标签键名称。 <p>说明 标签的键的最大长度为128个字符，标签的键可以包含任意语种字母、数字、空格和_ . : + - @ ,但首尾不能含有空格，不能以_sys_开头。</p>
标签值	<p>您可以选择：</p> <ul style="list-style-type: none"> 在输入框的下拉列表中选择预定义标签值。 在输入框中输入标签值。 <p>说明 标签值的最大长度为255个字符，标签的值可以包含任意语种字母、数字、空格和_ . : + - @ 。</p>

📖 说明

- 最多支持20个标签。
- 一个“键”只能添加一个“值”。
- 每个资源中的键名不能重复。

步骤5 单击“确定”，完成数据库标签的添加。

如需删除标签，在标签列表中，单击操作列中“删除”可对选中的标签进行删除。

----结束

数据表标签管理

步骤1 在DLI管理控制台的左侧导航栏中，单击“数据管理> 库表管理”。

步骤2 单击数据库名称，查看数据库下的数据表。

步骤3 在数据表的操作列，选择“更多> 标签”。

步骤4 进入标签管理页面，显示当前数据表的标签信息。

步骤5 单击“添加/编辑标签”，弹出“添加/编辑标签”对话框，配置参数。

输入框输入内容后单击'添加'，将标签添加到输入框中。

图 4-7 数据表添加/编辑标签



表 4-7 标签配置参数

参数	参数说明
标签键	<p>您可以选择：</p> <ul style="list-style-type: none"> 在输入框的下拉列表中选择预定义标签键。 如果添加预定义标签，用户需要预先在标签管理服务中创建好预定义标签，然后在“标签键”的下拉框中进行选择。用户可以通过单击“查看预定义标签”进入标签管理服务的“预定义标签”页面，然后单击“创建标签”来创建新的预定义标签。 具体请参见《标签管理服务用户指南》中的“创建预定义标签”章节。 在输入框中输入标签键名称。 <p>说明 标签的键的最大长度为128个字符，标签的键可以包含任意语种字母、数字、空格和_ . : + - @ ,但首尾不能含有空格，不能以_sys_开头。</p>
标签值	<p>您可以选择：</p> <ul style="list-style-type: none"> 在输入框的下拉列表中选择预定义标签值。 在输入框中输入标签值。 <p>说明 标签值的最大长度为255个字符，标签的值可以包含任意语种字母、数字、空格和_ . : + - @ 。</p>

📖 说明

- 最多支持20个标签。
- 一个“键”只能添加一个“值”。
- 每个资源中的键名不能重复。

步骤6 单击“确定”，完成数据表标签的添加。

如需删除标签，在标签列表中，单击操作列中“删除”可对选中的标签进行删除。

----结束

4.5 在 DLI 控制台管理表资源

4.5.1 在 DLI 控制台配置表权限

表权限操作场景

- 针对不同用户，可以通过权限设置分配不同的表权限。
- 管理员用户和表的所有者拥有所有权限，不需要进行权限设置且其他用户无法修改其表权限。
- 给新用户设置表权限时，该用户所在用户组的所属区域需具有Tenant Guest权限。关于Tenant Guest权限的介绍和开通方法，详细参见《[权限策略](#)》和《[统一身份认证服务用户指南](#)》中的[创建用户组](#)。

注意事项

- 如果需要查看主账号下数据库中的表，需要对当前子账号用户授权（显示权限），具体请参考[常用操作与系统权限关系](#)。
- 如果表被删除后，再重新创建同名的表，表权限不会继承，需要对操作该表的用户和项目重新进行权限赋予。

例如，testTable表给用户A赋予了删除表的权限，后续执行了删除testTable表，并重新创建了testTable表。如果希望A用户继续保留删除testTable表的权限，则需要重新对A用户赋予该权限。

查看表权限

1. 在管理控制台左侧，单击“数据管理”>“库表管理”。
2. 单击需要设置权限的表所在的数据库名，进入该数据库的“表管理”页面。
3. 单击所选表“操作”栏中的“权限管理”，将显示该表对应的权限信息。

图 4-8 表权限管理



表权限设置有3种场景：为新用户或项目赋予权限，为已有权限的用户或项目修改权限，回收某用户或项目具备的所有权限。

为新用户或项目赋予权限

为新用户或项目赋予权限，新用户或项目指之前不具备此表任何权限的用户或项目。

1. 单击表权限管理页面右上角的“授权”按钮。
2. 在弹出的“授权”对话框中选择相应的权限。
 - DLI表具体权限说明请参考[表4-8](#)。

图 4-9 DLI 表用户授权

授权

* 授权对象 **用户授权** 项目授权

* 用户名

请勾选需要赋予的权限

非继承权限

全选

显示表 查询表 显示表信息

显示创建表语句 删除表 重命名表

插入 重写 添加列

访问元数据 查看其他用户具备的权限 赋权

回收

确定 取消

图 4-10 DLI 表项目授权

授权

* 授权对象 用户授权 **项目授权**

* 项目

请勾选需要赋予的权限

非继承权限

全选

查询表 显示表信息 显示创建表语句

删除表 重命名表 插入

重写 添加列 访问元数据

查看其他用户具备的权限 赋权 回收

确定 取消

表 4-8 参数配置

参数	描述
授权对象	选择“用户授权”或“项目授权”。
用户名/项目	<ul style="list-style-type: none"> 选择“用户授权”时，输入表新增用户对应IAM用户名称。 说明 该用户名称是已存在的IAM用户名称且该用户登录过DLI管理控制台。 选择“项目授权”时，选择当前区域下需要授权的项目。 说明 选择“项目授权”时，只能查看被授权的表及其所在数据库的信息。

参数	描述
非继承权限	<p>选中权限即对用户或项目进行赋权，取消勾选即对用户权限或项目权限进行收回。</p> <ul style="list-style-type: none"> 同时适用于“用户授权”和“项目授权”的权限包括： <ul style="list-style-type: none"> 查询表：在当前表内查询数据。 显示表信息：显示当前表的信息。 显示创建表语句：显示当前表的创建语句。 删除表：删除当前表。 重命名表：对当前表重新命名。 插入：在当前表内插入数据。 重写：在当前表内插入覆盖数据。 添加列：在当前表中增加列。 赋权：当前用户可将表的权限赋予其他用户。 回收：当前用户可回收其他用户具备的此表的权限，并且不能回收表所有者的权限。 查看其他用户具备的权限：查看其他用户具备的当前表的权限。 <p>分区表还具有以下权限：</p> <ul style="list-style-type: none"> 删除分区：删除分区表中的分区。 显示所有分区：显示分区表中的所有分区。 <ul style="list-style-type: none"> 只适用于“用户授权”的权限包括： <ul style="list-style-type: none"> 显示表：显示当前表。

- OBS表具体权限说明请参考[表4-9](#)。

图 4-11 OBS 表用户授权



图 4-12 OBS 表项目授权



表 4-9 参数配置

参数	描述
授权对象	选择“用户授权”或“项目授权”。
用户名/项目	<ul style="list-style-type: none"> 选择“用户授权”时，输入表新增用户对应IAM用户名称。 说明 该用户名称是已存在的IAM用户名称且该用户登录过DLI管理控制台。 选择“项目授权”时，选择当前区域下需要授权的项目。 说明 选择“项目授权”时，只能查看被授权的表及其所在数据库的信息。

参数	描述
非继承权限	<p>选中权限即对用户或项目进行赋权，取消勾选即对用户权限或项目权限进行收回。</p> <ul style="list-style-type: none"> ● 同时适用于“用户授权”和“项目授权”的权限包括： <ul style="list-style-type: none"> - 显示创建表语句：显示当前表的创建语句。 - 显示表信息：显示当前表的信息。 - 查询表：在当前表内查询数据。 - 删除表：删除当前表。 - 重命名表：对当前表重新命名。 - 插入：在当前表内插入数据。 - 重写：在当前表内插入覆盖数据。 - 添加列：在当前表中增加列。 - 赋权：当前用户可将表的权限赋予其他用户或项目。 - 回收：当前用户或项目可回收其他用户或项目具备的此表的权限，并且不能回收表所有者的权限。 - 查看其他用户具备的权限：查看其他用户具备的当前表的权限。 <p>分区表还具有以下权限：</p> <ul style="list-style-type: none"> - 添加分区：在分区表中添加新的分区。 - 删除分区：删除分区表中的任意分区。 - 设置分区的路径：将分区表中的某个分区路径设置为用户指定的OBS路径。 - 重命名表分区：对分区表中的分区重新命名。 - 恢复表分区：从文件系统中导出分区信息保存到元数据中。 - 显示所有分区：显示分区表中的所有分区。 <ul style="list-style-type: none"> ● 只适用于“用户授权”的权限包括： <ul style="list-style-type: none"> - 显示表：显示当前表。

- View具体权限说明请参考[表4-10](#)。

说明

View只能通过SQL语句进行创建，不能通过“创建表”页面进行创建。

图 4-13 View 用户授权

授权

* 授权选择 用户授权 项目授权

* 用户名

请勾选需要授予的权限

非继承权限

全选

显示表 显示表信息 显示创建表语句

删除表 查询表 重命名表

查看其他用户具备的权限 赋权 回收

确定 取消

图 4-14 View 项目授权

授权

* 授权选择 用户授权 项目授权

* 项目名

请勾选需要授予的权限

非继承权限

全选

显示表信息 显示创建表语句 删除表

查询表 重命名表 查看其他用户具备的权限

赋权 回收

确定 取消

表 4-10 参数配置

参数	描述
授权选择	选择“用户授权”或“项目授权”。
用户名/项目	<ul style="list-style-type: none"> 选择“用户授权”时，输入表新增用户对应IAM用户名称。 说明 该用户名称是已存在的IAM用户名称且该用户登录过DLI管理控制台。 选择“项目授权”时，选择当前区域下需要授权的项目。 说明 选择“项目授权”时，只能查看被授权的表及其所在数据库的信息。

参数	描述
非继承权限	<p>选中权限即对用户或项目进行赋权，取消勾选即对用户权限或项目权限进行收回。</p> <ul style="list-style-type: none"> ● 同时适用于“用户授权”和“项目授权”的权限包括： <ul style="list-style-type: none"> - 显示表信息：显示当前表的信息。 - 显示创建表语句：显示当前表的创建语句。 - 删除表：删除当前表。 - 查询表：在当前表内查询数据。 - 重命名表：对当前表重新进行命名。 - 赋权：当前用户或项目可将表的权限赋予其他用户或项目。 - 回收：当前用户或项目可回收其他用户或项目具备的此表的权限，并且不能回收表所有者的权限。 - 查看其他用户具备的权限：查看其他用户具备的当前表的权限。 ● 只适用于 <ul style="list-style-type: none"> - 显示表：显示当前表。

3. 单击“确定”，完成表权限设置。

为已有权限的用户或项目修改权限

某用户或项目已具备此表的一些权限时，可为此用户或项目赋予或回收权限。

说明

当“权限设置”中的选项为灰色时，表示您不具备修改此表的权限。可以向管理员用户、表所有者等具有赋权权限的用户申请表的“赋权”和表权限的“回收”权限。

1. 在“用户权限信息”列表中找到需要设置权限的用户：
 - 如果用户为子用户且不是表的所有者，可进行“权限设置”。
 - 若用户为管理员用户或表的所有者，只能查看“权限信息”。

在“项目权限信息”列表中找到需要设置权限的项目，进行“权限设置”。
2. 在子用户或项目的“操作”栏中单击“权限设置”，可弹出表“权限设置”对话框。
 - DLI表用户或项目权限说明请参考[表4-8](#)。
 - OBS表用户或项目权限说明请参考[表4-9](#)。
 - View用户或项目权限说明请参考[表4-10](#)。
3. 单击“确定”，完成表权限设置。

回收某用户或项目具备的所有权限

回收某用户具备的所有权限，或回收某项目具备的所有权限。

- 在“用户权限信息”区域的用户列表中，选择需要回收权限的子用户，在“操作”栏中单击“回收”，确定后，此用户将不具备表的任意权限。

📖 说明

以下情况中，“回收”为灰色，表示不可回收该用户的权限。

- 用户为管理员用户
- 子用户是表的所有者
- 子用户只有可继承权限
- 在“项目权限信息”区域的项目列表中，选择需要回收权限的项目，在“操作”栏中单击“回收”，确定后，此项目将不具备表的任意权限。

📖 说明

当项目只有可继承权限时，“回收”为灰色，表示不可回收该项目的权限。

4.5.2 在 DLI 控制台删除表

根据实际使用情况，您可以通过删除表。

注意事项


- 具有正在运行中的作业的数据库或者表不能删除。
- 管理员用户、表的所有者和具有删除表权限的用户可以删除表。

📖 说明

数据表删除后，将不可恢复，请谨慎操作。

删除表

删除表的入口有两个，分别在“数据管理”和“SQL编辑器”页面。

- 在“数据管理”页面删除表。
 - a. 在管理控制台左侧，单击“数据管理”>“库表管理”。
 - b. 单击需删除表的数据库名，进入该数据库的“表管理”页面。
 - c. 选中目标表，单击“操作”栏中的“更多 > 删除表”。
 - d. 在弹出的确认对话框中，单击“是”。
- 在“SQL编辑器”页面删除表。
 - a. 在SQL作业管理控制台的顶部菜单栏中，选择“SQL编辑器”。
 - b. 在左侧导航栏选择“数据库”页签，鼠标左键单击需要删除表的数据库名，进入“表”区域。
 - c. 鼠标左键单击对应表右侧的 ，在列表菜单中选择“删除”。
 - d. 在弹出的确认对话框中，单击“确定”。

4.5.3 在 DLI 控制台修改表所有者

在实际使用过程中，开发人员创建了数据库和表，交给测试人员进行测试，测试人员测试完成后，再交给运维人员进行体验，在这种情况下，可以通过修改表的所有者，将数据转移给其他所有者。

修改表所有者

1. 在管理控制台左侧，单击“数据管理”>“库表管理”。
2. 单击需要修改的表对应数据库名，进入该数据库的“表管理”页面。
3. 单击目标表“操作”栏中的“更多”>“修改所有者”。
4. 在弹出的对话框中，输入新的所有者用户名（已存在的用户名），单击“确定”。

4.5.4 将 OBS 数据导入至 DLI

本节操作介绍将OBS上的数据导入到DLI控制台的表中。

注意事项

- 导入数据时只能指定一个路径，路径中不能包含逗号。
- 如果将CSV格式数据导入分区表，需在数据源中将分区列放在最后一列。
- 不建议对同一张表并发导入数据，因为有一定概率发生并发冲突，导致导入失败。
- 导入文件支持CSV，Parquet，ORC，JSON和Avro格式，且编码格式仅支持UTF-8。

前提条件

待导入的数据已存储到OBS上。

导入数据步骤

步骤1 导入数据的入口有两个，分别在“数据管理”和“SQL编辑器”页面。

- 在“数据管理”页面导入数据。
 - a. 在管理控制台的左侧，选择“数据管理”>“库表管理”。
 - b. 单击需导入数据的表对应的数据库名称，进入该数据库的“表管理”页面。
 - c. 在目标表“操作”栏中选择“更多”中的“导入”，弹出“导入数据”页面。

图 4-15 导入数据




- 在“SQL编辑器”页面导入数据。
 - a. 在管理控制台的左侧，单击“SQL编辑器”。
 - b. 在“SQL编辑器”页面左侧导航栏选择“数据库”页签，鼠标左键单击需要导入数据的表对应的数据库名，进入“表”区域。
 - c. 鼠标左键单击对应表右侧的☰，在列表菜单中选择“导入”，弹出“导入数据”页面。

图 4-16 SQL 编辑器-导入数据



步骤2 在“导入数据”页面，参见表4-11填写相关信息。

表 4-11 参数说明

参数名称	描述	示例
数据库	当前表所在的数据库。	-
表名称	当前表名称。	-
队列	选择队列。	-
文件格式	导入数据源的文件格式。导入支持CSV, Parquet, ORC, JSON, Avro格式。编码格式仅支持UTF-8。	CSV
数据源路径	<p>直接输入路径或单击  选择OBS的路径，如果没有合适的桶可直接跳转OBS创建。</p> <ul style="list-style-type: none"> 创建OBS表时指定的路径必须是文件夹，如果建表路径是文件将导致导入数据失败。 当OBS的目录下有同名文件夹和文件时，数据导入指向该路径会优先指向文件而非文件夹。 <p>说明 路径同时支持文件和文件夹。</p>	obs://DLI/sampled ata.csv
表头:无/有	<p>当“文件格式”为“CSV”时该参数有效。设置导入数据源是否含表头。</p> <p>选中“高级选项”，勾选“表头:无”前的方框，“表头:无”显示为“表头:有”，表示有表头；取消勾选即为“表头:无”，表示无表头。</p>	-


参数名称	描述	示例
自定义分隔符	<p>当“文件格式”为“CSV”，勾选自定义分隔符前的方框时，该参数有效。</p> <p>支持选择如下分隔符。</p> <ul style="list-style-type: none"> • 逗号(,) • 竖线() • 制表符(\t) • 其他：输入自定义分隔符 	默认值： 逗号(,)
自定义引用字符	<p>当“文件格式”为“CSV”，勾选自定义引用字符前的方框时，该参数有效。</p> <p>支持选择如下引用字符。</p> <ul style="list-style-type: none"> • 单引号(') • 双引号(") • 其他：输入自定义引用字符 	默认值： 单引号(')
自定义转义字符	<p>当“文件格式”为“CSV”，并在自定义转义字符前的方框打勾时，该参数有效。</p> <p>选中高级选项，支持选择如下转义字符。</p> <ul style="list-style-type: none"> • 反斜杠(\) • 其他：输入自定义转义字符 	默认值： 反斜杠(\)
日期格式	<p>当“文件格式”为“CSV”和“JSON”时此参数有效。</p> <p>选中“高级选项”，该参数表示表中日期的格式，默认格式为“yyyy-MM-dd”。日期格式字符定义详见加载数据中的“表3 日期及时间模式字符定义”。</p>	2000-01-01
时间戳格式	<p>当“文件格式”为“CSV”和“JSON”时此参数有效。</p> <p>选中“高级选项”，该参数表示表中时间戳的格式，默认格式为“yyyy-MM-dd HH:mm:ss”。时间戳格式字符定义详见加载数据中的“表3 日期及时间模式字符定义”。</p>	2000-01-01 09:00:00
错误数据存储路径	<p>当“文件格式”为“CSV”和“JSON”时此参数有效。</p> <p>选中“高级选项”，该参数表示可将错误数据保存到对应的OBS路径中。</p>	obs://DLI/

步骤3 单击“确定”，系统开始导入数据。

步骤4 有两种方式可查看导入的数据。

说明

目前预览只显示导入的前十条数据。

- 在“数据管理” > “库表管理”页面，单击数据库名，在表管理界面对应表的“操作”栏选择“更多”中的“表属性”，在弹框的“预览”页签中，可查看导入的数据
- 在“SQL编辑器”的“数据库”页签中，单击数据库名称，进入对应的表列表，鼠标左键单击对应表右侧的 ，在列表菜单中选择“表属性”，在弹框的“预览”页签中，可查看导入的数据。

步骤5（可选）可以在“作业管理 > SQL作业”页面，查看该导入作业的状态以及执行结果。

----结束

4.5.5 导出 DLI 表数据至 OBS 中

支持将数据从DLI表中导出到OBS服务中，导出操作将在OBS服务新建文件夹，或覆盖已有文件夹中的内容。

注意事项

- 支持导出json格式的文件，且文本格式仅支持UTF-8。
- 只支持将DLI表（表类型为“Managed”）中的数据导出到OBS桶中，且导出的路径必须指定到文件夹级别。
- 支持跨账号导出数据，即，如果B账户对A账户授权后，A账户拥有B账户OBS桶的元数据信息和权限信息的读取权限，以及路径的读写权限，则A账户可将数据导出至B账户的OBS路径中。

导出数据步骤

步骤1 导出数据的入口有两个，分别在“数据管理”和“SQL编辑器”页面。


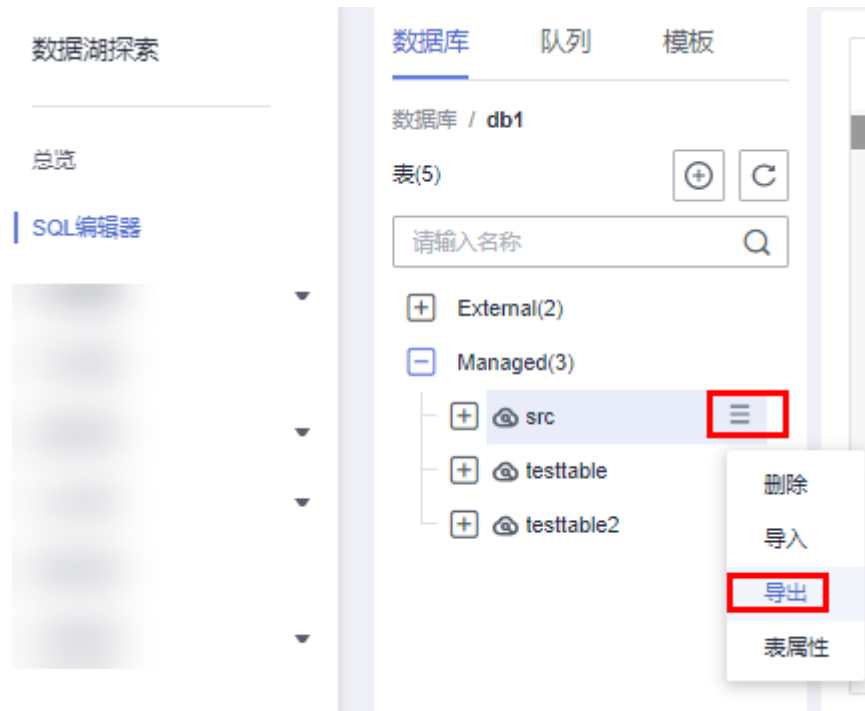
- 在“数据管理”页面导出数据。
 - a. 在管理控制台左侧，单击“数据管理” > “库表管理”。
 - b. 单击需导出数据的表对应的数据库，进入该数据的“表管理”页面。
 - c. 在对应表（DLI表）的“操作”栏中选择“更多”中的“导出”，弹出“导出数据”页面。
- 在“SQL编辑器”页面导出数据。
 - a. 在管理控制台左侧，单击“SQL编辑器”。
 - b. 在左侧导航栏选择“数据库”页签，鼠标左键单击需要导出数据的表对应的数据库名，进入“表”区域。
 - c. 鼠标左键单击需要导出数据的表（Managed表，即DLI表）右侧的 ，在列表菜单中选择“导出”，选择弹出“导出数据”页面。

图 4-17 Managed 表导出



步骤2 在“导出数据”对话框，参考表4-12填写导出数据相关信息。

图 4-18 导出数据




表 4-12 参数说明

参数名称	描述
数据库	当前表所在的数据库。
表名称	当前表名称。
数据格式	导出数据的文件格式。当前只支持json格式。
队列	选择队列。
压缩格式	导出数据的压缩方式，选择如下压缩方式。 <ul style="list-style-type: none">• none• bzip2• deflate• gzip
存储路径	<ul style="list-style-type: none">• 输入或选择OBS路径。• 导出路径必须为OBS桶中不存在的文件夹，即用户需在OBS目标路径后创建一个新文件夹。• 文件夹名称不能包含下列特殊字符：\ / : * ? " < > ，并且不能以“.”开头和结尾。
导出方式	导出数据的保存方式。 <ul style="list-style-type: none">• 随导出创建指定路径：指定的导出目录必须不存在，如果指定目录已经存在，系统将返回错误信息，无法执行导出操作。• 覆盖指定路径：在指定目录下新建文件，会删除已有文件。
表头:无/ 有	设置导出数据是否含表头。

步骤3 单击“确定”即可导出数据。

步骤4 （可选）您可以在“作业管理”>“SQL作业”页面查看导出作业的“状态”、“执行语句”等信息。

1. 在“作业类型”中选择“EXPORT”，输入导出数据的时间段，即可查询出对应条件下的作业列表。
2. 单击导出作业名称前的 ，可查看导出作业的详细信息。

----结束


4.5.6 在 DLI 控制台预览表数据

“预览页面”将显示对应表的前10条数据。

预览数据步骤

预览数据的入口有两个，分别在“数据管理”和“SQL编辑器”页面。

- 在“数据管理”页面预览数据。

- a. 在管理控制台左侧，单击“数据管理” > “库表管理”。
 - b. 单击需导出数据对应数据库名称，进入该数据库“表管理”页面。
 - c. 单击目标表“操作”栏中的“更多”，选择“表属性”。
 - d. 单击“预览”页签，即可预览该表数据。
- 在“SQL编辑器”页面预览数据。
 - a. 在管理控制台左侧，单击“SQL编辑器”。
 - b. 在“SQL编辑器”页面的左侧导航栏中，选择“数据库”页签。
 - c. 鼠标左键单击对应数据库名，进入该数据库的表列表。
 - d. 鼠标左键单击对应表右侧的 ，在列表菜单中选择“表属性”，单击“预览”页签，即可预览该表数据。

4.6 创建并使用 LakeFormation 元数据

4.6.1 DLI 对接 LakeFormation

操作场景

LakeFormation是企业级一站式湖仓构建服务，提供元数据统一管理能力，支持无缝对接多种计算引擎及大数据云服务，便捷高效地构建数据湖和运营相关业务，加速释放业务数据价值。

在DLI的Spark作业和SQL作业场景，支持对接LakeFormation实现元数据的统一管理，本节操作介绍配置DLI与LakeFormation的数据连接的操作步骤。

LakeFormation Spark语法请参考[Spark语法参考](#)。

LakeFormation Flink语法请参考[Flink语法参考](#)。

HetuEngine SQL语法请参考[HetuEngine SQL语法参考](#)。

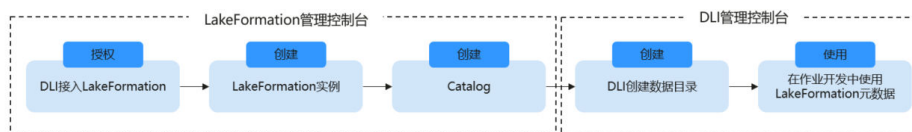
使用须知

该功能为**白名单功能**，如需使用，请在管理控制台右上角，选择“工单 > 新建工单”，提交申请。

DLI对接LakeFormation功能的使用依赖于“湖仓构建”服务的上线状态，如需了解“湖仓构建”服务的上线范围请参考[全球产品和服务](#)。

操作流程

图 4-19 操作流程



约束限制

- 在表4-13中提供了支持对接LakeFormation获取元数据的队列和引擎类型。查看队列的引擎类型和版本请参考[查看队列的基本信息](#)。

表 4-13 LakeFormation 获取元数据的队列和引擎类型

队列类型	引擎类型和支持的版本
default队列	<ul style="list-style-type: none">Spark 3.3.x: 支持对接LakeFormation获取元数据的队列和引擎。HetuEngine 2.1.0: 支持对接LakeFormation获取元数据的队列和引擎。
SQL队列	<ul style="list-style-type: none">Spark 3.3.x: 支持对接LakeFormation获取元数据的队列和引擎。HetuEngine 2.1.0: 支持对接LakeFormation获取元数据的队列和引擎。
通用队列	Flink作业场景: Flink 1.15及以上版本且使用弹性资源池队列时支持对接LakeFormation获取元数据。

- DLI仅支持对接LakeFormation默认实例，请在LakeFormation设置实例为默认实例。
- DLI支持读取Lakeformation的中Avro、Json、Parquet、Csv、Orc、Text、Hudi格式的数据。
- LakeFormation数据目录中的库、表权限统一由LakeFormation管理。
- DLI支持对接LakeFormation后，DLI原始库表下移至dli的数据目录下。

步骤 1：创建 LakeFormation 实例用于元数据存储

LakeFormation实例为元数据的管理提供基础资源，DLI仅支持对接LakeFormation的默认实例。

1. 创建实例

- 登录LakeFormation管理控制台。
- 单击页面右上角“立即购买”或“购买实例”，进入实例购买页面。
首次创建实例时界面显示“立即购买”，如果界面已有LakeFormation实例则显示为“购买实例”。
- 按需配置LakeFormation实例参数，完成实例创建。
本例创建按需计费的共享型实例。
更多参数配置及说明，请参考[创建LakeFormation实例](#)。

2. 设置实例为默认实例

- 查看实例“基本信息”中“是否为默认实例”的参数值。
 - “true”表示当前实例为默认实例。
 - “false”表示当前实例不为默认实例。

- b. 如果需要设置当前实例为默认实例，请单击页面右上角“设为默认实例”。
- c. 勾选操作影响后单击“确定”，将当前实例设置为默认实例。

说明

当前DLI仅对接LakeFormation默认实例，变更默认实例后，可能对使用LakeFormation的周边服务产生影响，请谨慎操作。

步骤 2：在 LakeFormation 管理控制台创建 Catalog

数据目录（Catalog）是元数据管理对象，它可以包含多个数据库。您可以在LakeFormation中创建并管理多个Catalog，用于不同外部集群的元数据隔离。

1. 登录LakeFormation管理控制台。
 2. 选择“元数据 > Catalog”。
 3. 单击“创建Catalog”。
- 按需配置Catalog实例参数。
更多参数配置及说明，请参考[创建Catalog](#)。
4. 创建完成后，即可在“Catalog”页面查看Catalog相关信息。

步骤 3：在 DLI 管理控制台创建数据目录

在DLI管理控制台需要创建到Catalog的连接，才可以访问LakeFormation实例中存储的Catalog。

说明

- DLI仅支持对接LakeFormation默认实例，请在LakeFormation设置实例为默认实例。
- LakeFormation中每一个数据目录只能创建一个映射，不能创建多个。
例如用户在DLI创建了映射名catalogMapping1对应LakeFormation数据目录：catalogA。创建成功后，在同一个项目空间下，不能再创建到catalogA的映射。


1. 登录DLI管理控制台。
2. 选择“SQL编辑器”。
3. 在SQL编辑器页面，选择“数据目录”。
4. 单击  创建数据目录。
5. 配置数据目录相关信息。

表 4-14 数据目录配置信息

参数名称	是否必填	说明
外部数据目录名称	是	LakeFormation默认实例下的Catalog名称。
类型	是	当前只支持LakeFormation。 该选项已固定，无需填写。

参数名称	是否必填	说明
数据目录映射名称	是	在DLI使用的Catalog映射名，用户在执行SQL语句的时候需要指定Catalog映射，以此来标识访问的外部的元数据。建议与外部数据目录名称保持一致。 当前仅支持连接LakeFormation默认实例的数据目录。
描述	否	自定义数据目录的描述信息。

6. 单击“确定”创建数据目录。

步骤 4：授权使用 LakeFormation 资源

- **SQL作业场景**

在进行SQL作业提交之前，需完成LakeFormation元数据、数据库、表、列和函数等资源授权，确保作业在执行过程中能够顺利访问所需的数据和资源。

[LakeFormation SQL资源权限支持列表](#)提供了LakeFormation权限支持列表。

使用LakeFormation资源需要分别完成LakeFormation的IAM细粒度授权和LakeFormation SQL资源授权。

- **LakeFormation的IAM细粒度授权：授权使用LakeFormation API。**

IAM服务通常提供了管理用户、组和角色的访问权限的方式。您可以在IAM控制台中创建策略（Policy），定义哪些用户或角色可以调用LakeFormation的API。然后，将这些策略附加到相应的用户或角色上。

- **方法1：基于角色授权：**

即IAM最初提供的一种根据用户的工作职能定义权限的粗粒度授权机制。该机制以服务为粒度，提供有限的服务相关角色用于授权。

例如参考[LakeFormation权限管理](#)授予用户只读权限，允许查询LakeFormation相关元数据资源的权限。

或如下示例授予LakeFormation相关元数据资源的所有操作权限。

示例：

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lakeformation:table:*",
        "lakeformation:database:*",
        "lakeformation:catalog:*",
        "lakeformation:function:*",
        "lakeformation:transaction:*",
        "lakeformation:policy:describe",
        "lakeformation:credential:describe"
      ]
    }
  ]
}
```

- **方法2：基于策略的精细化授权：**

IAM提供的细粒度授权的能力，可以精确到具体服务的操作、资源以及请求条件等。

LakeFormation权限策略请参考[LakeFormation权限和授权项](#)。

IAM授权的具体操作请参考[创建用户并授权使用LakeFormation](#)。

- **LakeFormation SQL资源授权：授权使用LakeFormation具体资源（元数据、数据库、表、列和函数等）。**

LakeFormation资源授权是指允许用户对特定资源的访问的权限，以此来控制对LakeFormation的数据和元数据的访问。

LakeFormation资源授权有两种方式：

- 方式一：在LakeFormation管理控制台对资源授权。
具体操作请参考LakeFormation用户指南中的[新增授权](#)。
了解LakeFormation SQL资源权限请参考[数据权限概述](#)。
- 方式二：在DLI管理控制台使用GRANT SQL语句授权
GRANT语句是SQL语言中用于授权的一种方式。
您可以使用GRANT语句来授予用户或角色对数据库、表、列、函数等的访问权限。

[LakeFormation SQL资源权限支持列表](#)提供了LakeFormation资源授权的策略。

说明

Catalog资源暂时不支持在DLI SQL授权，请参考[方式一：在LakeFormation管理控制台...](#)在LakeFormation 管理控制台完成授权。

- **Spark Jar、Flink OpenSource SQL、Flink Jar作业场景：**
 - 方式1：使用委托授权：使用Spark 3.3.1及以上版本、Flink 1.15版本的引擎执行作业时，需要您先在IAM页面创建相关委托，并在配置作业时添加新建的委托信息。
委托权限示例请参考[创建DLI自定义委托权限](#)和[常见场景的委托权限策略](#)。
 - 方式2：使用DEW授权：
 - 已为授予IAM用户所需的IAM和Lakeformation权限，具体请参考[SQL作业场景的IAM授权的操作步骤](#)。
 - 已在DEW服务创建通用凭证，并存入凭据值。具体操作请参考[创建通用凭据](#)。
 - 已创建DLI访问DEW的委托并完成委托授权。该委托需具备以下权限：
 - DEW中的查询凭据的版本与凭据值ShowSecretVersion接口权限，csms:secretVersion:get。
 - DEW中的查询凭据的版本列表ListSecretVersions接口权限，csms:secretVersion:list。
 - DEW解密凭据的权限，kms:dek:decrypt。

委托权限示例请参考[创建DLI自定义委托权限](#)和[常见场景的委托权限策略](#)。

步骤 5：在 DLI 作业开发时使用 LakeFormation 元数据

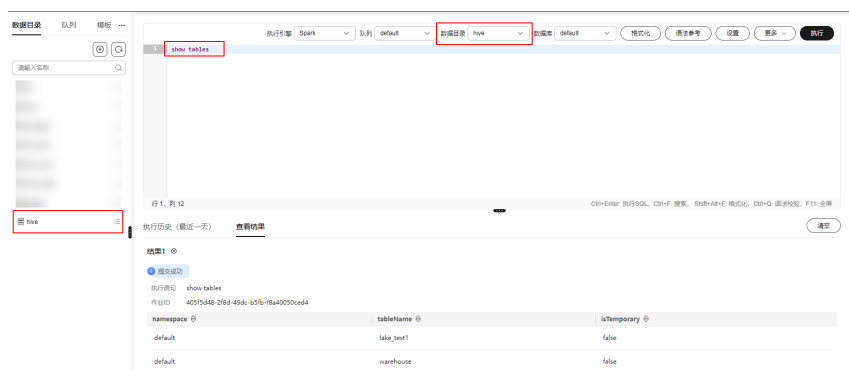
DLI对接LakeFormation默认实例且完成LakeFormation的资源授权后，即可以在作业开发时使用LakeFormation元数据。

- **DLI SQL:**

LakeFormation SQL语法说明请参考[DLI Spark SQL语法参考](#)。

在执行SQL作业时，您可以在控制台选择执行SQL所在的catalog，如[图4-20](#)所示，或在SQL命令中指定catalogName。catalogName是DLI控制台的数据目录映射名。

图 4-20 在 SQL 编辑器页面选择数据目录



说明

- 对接LakeFormation实例场景，在创建数据库时需要指定数据库存储的OBS路径。
- 对接LakeFormation实例场景，在创建表时不支持设置表生命周期和多版本。
- 对接LakeFormation实例场景，LOAD DATA语句不支持datasource表，且LOAD DATA分区表必须指定分区。
- 在LakeFormation控制台创建的数据库和表中包含中文字符时，不支持在DLI执行相关数据库和表的操作。
- 对接LakeFormation实例场景，不支持指定筛选条件删除分区。
- 对接LakeFormation实例场景，不支持创建Truncate Datasource/Hive外表。
- DLI暂不支持使用LakeFormation行过滤条件功能。
- DLI读取binary类型的数据进行console展示时，会对binary数据进行Base64转换。
- 在DLI暂不支持LakeFormation的路径授权。

- **DLI Spark Jar:**

本节介绍在DLI管理控制台提交Spark Jar作业时使用LakeFormation元数据的配置操作。

- **Spark Jar 示例**

```
SparkSession spark = SparkSession.builder()
    .enableHiveSupport()
    .appName("java_spark_demo")
    .getOrCreate();

spark.sql("show databases").show();
```

- **DLI管理控制台Spark Jar作业配置说明**

- **(推荐) 方式一：使用控制台提供的参数项（委托、元数据来源等）配置Spark Jar作业访问LakeFormation元数据**

新建或编辑Spark Jar作业时，请参考[表4-15](#)Spark Jar作业访问LakeFormation元数据。

表 4-15 配置 Spark Jar 作业访问 LakeFormation 元数据

参数	说明	配置示例
Spark版本	Spark 3.3.x及以上版本支持对接 LakeFormation。	3.3.1
委托	使用Spark 3.3.1及以上版本的引擎执行作业时，需要您先在IAM页面创建相关委托，并在此处添加新建的委托信息。选择该参数后系统将自动为您的作业添加以下配置： spark.dli.job.agency.name= <i>agency</i> 委托权限示例请参考 创建DLI自定义委托权限 和 常见场景的委托权限策略 。	-
访问元数据	配置开启Spark作业访问元数据功能。	是
元数据来源	配置Spark作业访问的元数据类型。本场景下请选择Lakeformation。 选择该参数后系统将自动为您的作业添加以下配置项用于加载lakeformation相关依赖。 spark.sql.catalogImplementation=hive spark.hadoop.hive-ext.dlcatalog.metastore.client.enable=true spark.hadoop.hive-ext.dlcatalog.metastore.session.client.class=com.huawei.cloud.dalf.lakecat.client.hiveclient.LakeCatMetaStoreClient og // lakeformation相关依赖加载 spark.driver.extraClassPath=/usr/share/extension/dli/spark-jar/lakeformation/* spark.executor.extraClassPath=/usr/share/extension/dli/spark-jar/lakeformation/* “元数据来源”还支持在Spark (--conf) 参数中配置，且系统优先以Spark (--conf) 中配置信息为准。 优先推荐您使用控制台提供的“元数据来源”参数项进行配置。	Lakeformation

参数	说明	配置示例
数据目录名称	<p>配置Spark作业访问的数据目录名称。</p> <p>此处选择的是在DLI管理控制台创建的数据目录，即DLI与Lakeformation默认实例下的数据目录的映射，该数据目录连接的是LakeFormation默认实例下的数据目录。如需指定LakeFormation其他实例请参考方式二：使用Spark (--conf) 参数配置...在Spark (--conf) 中配置连接的Lakeformation实例和数据目录。</p> <p>选择该参数后系统将自动为您的作业添加以下配置项用于连接Lakeformation默认实例下的数据目录。</p> <pre>spark.hadoop.lakecat.catalogname.default=lfcatalog</pre> <p>“数据目录名称”还支持在Spark (--conf) 参数中配置，且系统优先以Spark (--conf) 中配置信息为准。</p> <p>优先推荐您使用控制台提供的“数据目录名称”参数项进行配置。</p>	-
Spark参数 (--conf)	<p>“元数据来源”和“数据目录名称”均支持在Spark (--conf) 参数中配置，且系统优先以Spark (--conf) 中配置信息为准。</p> <ul style="list-style-type: none"> 如果您需要配置访问Hudi数据表，可在Spark (--conf) 参数中填加以下配置项。 <pre>spark.sql.extensions=org.apache.spark.sql.hudi.HoodieSparkSessionExtension spark.hadoop.hoodie.write.lock.provider=org.apache.hudi.lakeformation.LakeCatMetastoreBasedLockProvider</pre> 如果您需要配置访问Delta数据表，可在Spark (--conf) 参数中填加以下配置项。 <pre>spark.sql.catalog.spark_catalog=org.apache.spark.sql.delta.catalog.DeltaCatalog spark.sql.extensions=io.delta.sql.DeltaSparkSessionExtension</pre> 	-

▪ **方式二：使用Spark (--conf) 参数配置Spark Jar作业访问LakeFormation元数据**

新建或编辑Spark Jar作业时，请在作业配置页面的Spark (--conf) 参数中按需配置以下信息以访问LakeFormation元数据。

```
spark.sql.catalogImplementation=hive
spark.hadoop.hive-ext.dlcatalog.metastore.client.enable=true
spark.hadoop.hive-ext.dlcatalog.metastore.session.client.class=com.huawei.cloud.dalf.lakecat.client.hiveclient.LakeCatMetaStoreClient
spark.sql.extensions=org.apache.spark.sql.hudi.HoodieSparkSessionExtension //支持hudi，可选
spark.hadoop.hoodie.write.lock.provider=org.apache.hudi.lakeformation.LakeCatMetastoreBasedLockProvider //支持hudi，可选
// 使用有OBS和lakeformation权限的委托访问，建议用户设置最小权限集
spark.dli.job.agency.name=agencyForLakeformation
//需要访问的lakeformation实例ID，在lakeformation console查看。可选，如不填写访问
```

```

Lakeformation的默认实例
spark.hadoop.lakeformation.instance.id=xxx
//需要访问的lakeformation侧的CATALOG名称，在lakeformation console查看。可选，如不
填写则默认值为hive
spark.hadoop.lakecat.catalogname.default=lfcatalog
// lakeformation相关依赖加载
spark.driver.extraClassPath=/usr/share/extension/dli/spark-jar/lakeformation/*
spark.executor.extraClassPath=/usr/share/extension/dli/spark-jar/lakeformation/*
    
```

● **DLI Flink OpenSource SQL**

- 示例1：委托的方式对接Lakeformation

创建Flink OpenSource SQL作业并配置如下参数：

参数	说明	配置示例
Flink版本	Flink 1.15及以上版本支持对接LakeFormation。	1.15
委托	使用Flink 1.15及以上版本的引擎执行作业时，需要您先在IAM页面创建相关委托，并在此处添加新建的委托信息。选择该参数后系统将自动为您的作业添加以下配置： <code>flink.dli.job.agency.name=agency</code> 委托权限示例请参考 创建DLI自定义委托权限和常见场景的委托权限策略 。	-
开启checkpoint	勾选开启checkpoint。	开启
自定义参数	<ul style="list-style-type: none"> 配置Flink作业访问的元数据类型。本场景下请选择Lakeformation。 <code>flink.dli.job.catalog.type=lakeformation</code> 配置Flink作业访问的数据目录名称。 <code>flink.dli.job.catalog.name=[lakeformation中的catalog名称]</code> 此处选择的是在DLI管理控制台创建的数据目录，即DLI与Lakeformation默认实例下的数据目录的映射，该数据目录连接的是LakeFormation默认实例下的数据目录。 	-

示例中关于Catalog的参数说明请参考[表4-16](#)

表 4-16 Flink OpenSource SQL 示例中关于 Catalog 的参数说明

参数	说明	是否必填	参数值
type	catalog类型	是	固定值hive
hive-conf-dir	hive-conf路径，固定值/opt/flink/conf	是	固定值/opt/flink/conf

参数	说明	是否必填	参数值
default-database	默认数据库名称	否	默认default库

```

CREATE CATALOG hive
WITH
(
  'type' = 'hive',
  'hive-conf-dir' = '/opt/flink/conf', -- 固定配置/opt/flink/conf
  'default-database'='default'
);

USE CATALOG hive;

CREATE TABLE IF NOT EXISTS
  dataGenSource612 (user_id string, amount int)
WITH
(
  'connector' = 'datagen',
  'rows-per-second' = '1',
  'fields.user_id.kind' = 'random',
  'fields.user_id.length' = '3'
);

CREATE table IF NOT EXISTS
  printSink612 (user_id string, amount int)
WITH
('connector' = 'print');

INSERT INTO
  printSink612
SELECT
  *
FROM
  dataGenSource612;

```

- 示例2：DEW的方式对接Lakeformation

创建Flink OpenSource SQL作业并配置如下参数：

参数	说明	配置示例
Flink版本	Flink 1.15及以上版本支持对接LakeFormation。	1.15
委托	使用Flink 1.15及以上版本的引擎执行作业时，需要您先在IAM页面创建相关委托，并在此处添加新建的委托信息。选择该参数后系统将自动为您的作业添加以下配置： flink.dli.job.agency.name= <i>agency</i> 委托权限示例请参考 创建DLI自定义委托权限和常见场景的委托权限策略 。	-
开启checkpoint	勾选开启checkpoint。	开启

参数	说明	配置示例
自定义参数	<ul style="list-style-type: none"> 配置Flink作业访问的元数据类型。本场景下请选择Lakeformation。 flink.dli.job.catalog.type=lakeformation 配置Flink作业访问的数据目录名称。 flink.dli.job.catalog.name=[lakeformation中的catalog名称] <p>此处选择的是在DLI管理控制台创建的数据目录，即DLI与Lakeformation默认实例下的数据目录的映射，该数据目录连接的是LakeFormation默认实例下的数据目录。</p>	-

示例中关于Catalog的参数说明请参考[表4-17](#)

需要指定properties.catalog.lakeformation.auth.identity.util.class参数值为com.huawei.flink.provider.lakeformation.FlinkDewIdentityGenerator，并且配置dew相关配置。

表 4-17 Flink OpenSource SQL 示例中关于 Catalog 的参数说明（DEW 方式）

参数	说明	是否必填	参数值
type	catalog类型	是	固定值hive
hive-conf-dir	hive-conf路径，固定值/opt/flink/conf	是	固定值/opt/flink/conf
default-database	默认数据库名称	否	不填默认default库
properties.catalog.lakecatalog.auth.identity.util.class	认证信息获取类	是	dew方式必填，固定配置为com.huawei.flink.provider.lakeformation.FlinkDewIdentityGenerator
properties.catalog.dew.projectId	DEW所在的项目ID，默认是Flink作业所在的项目ID。	是	使用dew方式必填
properties.catalog.dew.endpoint	指定要使用的DEW服务所在的endpoint信息。	是	使用dew方式必填。 配置示例： kms.xxx.com
properties.catalog.dew.credentials.secretName	在DEW服务的凭据管理中新建的通用凭据的名称。	是	使用dew方式必填

参数	说明	是否必填	参数值
properties.catalog.dew.csms.version	在DEW服务的凭据管理中新建的通用凭据的版本号。	是	使用dew方式必填
properties.catalog.dew.access.key	在DEW服务的凭据中配置access.key值对应的key	是	使用dew方式必填
properties.catalog.dew.secret.key	在DEW服务的凭据中配置secret.key值对应的key	是	使用dew方式必填

```
CREATE CATALOG myhive
WITH
(
  'type' = 'hive',
  'hive-conf-dir' = '/opt/flink/conf',
  'default-database'='default',
  --下边是dew相关配置，请根据实际情况修改参数值
  'properties.catalog.lakeformation.auth.identity.util.class' =
  'com.huawei.flink.provider.lakeformation.FlinkDewIdentityGenerator',
  'properties.catalog.dew.endpoint'='kms.xxx.com',
  'properties.catalog.dew.csms.secretName'='obsAksK',
  'properties.catalog.dew.access.key' = 'myak',
  'properties.catalog.dew.secret.key' = 'mysk',
  'properties.catalog.dew.projectId'='330e068af1334c9782f4226xxxxxxxx',
  'properties.catalog.dew.csms.version'='v9'
);

USE CATALOG myhive;

create table IF NOT EXISTS dataGenSource_dew612(
  user_id string,
  amount int
) with (
  'connector' = 'datagen',
  'rows-per-second' = '1',
  'fields.user_id.kind' = 'random',
  'fields.user_id.length' = '3'
);

create table IF NOT EXISTS printSink_dew612(
  user_id string,
  amount int
) with (
  'connector' = 'print'
);

insert into printSink_dew612 select * from dataGenSource_dew612;
```

- 示例3：委托的方式对接Lakeformation写hudi表
创建Flink OpenSource SQL作业并配置如下参数：

参数	说明	配置示例
Flink版本	Flink 1.15及以上版本支持对接LakeFormation。	1.15

参数	说明	配置示例
委托	<p>使用Flink 1.15及以上版本的引擎执行作业时，需要您先在IAM页面创建相关委托，并在此处添加新建的委托信息。选择该参数后系统将自动为您的作业添加以下配置：</p> <p><code>flink.dli.job.agency.name=agency</code></p> <p>委托权限示例请参考创建DLI自定义委托权限和常见场景的委托权限策略。</p>	-
开启checkpoint	勾选开启checkpoint。	开启
自定义参数	<ul style="list-style-type: none"> 配置Flink作业访问的元数据类型。本场景下请选择Lakeformation。 <code>flink.dli.job.catalog.type=lakeformation</code> 配置Flink作业访问的数据目录名称。 <code>flink.dli.job.catalog.name=[lakeformation中的catalog名称]</code> <p>此处选择的是在DLI管理控制台创建的数据目录，即DLI与Lakeformation默认实例下的数据目录的映射，该数据目录连接的是LakeFormation默认实例下的数据目录。</p>	-

示例中关于Catalog的参数说明请参考[表4-18](#)。

表 4-18 hudi 类型 Catalog 参数说明

参数	说明	是否必填	参数值
type	catalog类型	是	hudi表配置为hudi。
hive-conf-dir	hive-conf路径，固定值/opt/flink/conf	是	固定值/opt/flink/conf。
default-database	默认数据库名称	否	默认default库。

参数	说明	是否必填	参数值
mode	取值'hms' 或 'non-hms'。 <ul style="list-style-type: none"> 'hms' 表示创建的 Hudi Catalog 会使用 Hive Metastore 存储元数据信息。 'non-hms'表示不使用Hive Metastore存储元数据信息。 	是	固定值hms。

表 4-19 hudi 类型 sink 表的 connector 参数

参数	说明	是否必填	参数值
connector	flink connector类型。 配置为hudi表示 sink表是hudi表。	是	hudi
path	表的基本路径。如果该路径不存在，则会创建它。	是	请参考示例代码中的配置值。
hoodie.datasource.write.recordkey.field	hoodie表的唯一键字段名	否	这里配置order_id为唯一键。
EXTERNAL	是否外表	是 hudi表必填，且设置为true	true

```
CREATE CATALOG hive_catalog
WITH (
  'type'='hive',
  'hive-conf-dir' = '/opt/flink/conf',
  'default-database'='test'
);
USE CATALOG hive_catalog;
create table if not exists genSource618 (
  order_id STRING,
  order_name STRING,
  price INT,
  weight INT
) with (
  'connector' = 'datagen',
  'rows-per-second' = '1',
  'fields.order_id.kind' = 'random',
```

```
'fields.order_id.length' = '8',
'fields.order_name.kind' = 'random',
'fields.order_name.length' = '5'
);

CREATE CATALOG hoodie_catalog
WITH (
  'type'='hudi',
  'hive.conf.dir' = '/opt/flink/conf',
  'mode'='hms' -- supports 'dfs' mode that uses the DFS backend for table DDLs
persistence
);
CREATE TABLE if not exists hoodie_catalog.`test`.`hudiSink618` (
  `order_id` STRING PRIMARY KEY NOT ENFORCED,
  `order_name` STRING,
  `price` INT,
  `weight` INT,
  `create_time` BIGINT,
  `create_date` String
) PARTITIONED BY (create_date) WITH (
  'connector' = 'hudi',
  'path' = 'obs://xxx/catalog/dbtest3/hudiSink618',
  'hoodie.datasource.write.recordkey.field' = 'order_id',
  'write.precombine.field' = 'create_time',
  'EXTERNAL' = 'true' -- must be set
);

insert into hoodie_catalog.`test`.`hudiSink618`
select
  order_id,
  order_name,
  price,
  weight,
  UNIX_TIMESTAMP() as create_time,
  FROM_UNIXTIME(UNIX_TIMESTAMP(), 'yyyyMMdd') as create_date
from genSource618;
```

- **DLI Flink Jar**

- **示例1：委托方式对接Lakeformation**

- i. 开发Flink jar程序，编译并上传jar包到obs，本例上传到obs://obs-test/dlitest/目录

示例代码如下：

本例通过DataGen表产生随机数据并输出到Print结果表中。

其他connector类型可参考[Flink 1.15支持的connector列表](#)。

```
package com.huawei.test;

import org.apache.flink.api.java.utils.ParameterTool;
import org.apache.flink.contrib.streaming.state.RocksDBStateBackend;
import org.apache.flink.runtime.state.filesystem.FsStateBackend;
import org.apache.flink.streaming.api.CheckpointingMode;
import org.apache.flink.streaming.api.environment.CheckpointConfig;
import org.apache.flink.streaming.api.environment.StreamExecutionEnvironment;
import org.apache.flink.table.api.EnvironmentSettings;
import org.apache.flink.table.api.bridge.java.StreamTableEnvironment;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import java.text.SimpleDateFormat;

@SuppressWarnings({"deprecation", "rawtypes", "unchecked"})
public class GenToPrintTaskAgency {
  private static final Logger LOGGER =
    LoggerFactory.getLogger(GenToPrintTaskAgency.class);
  private static final String datePattern = "yyyy-MM-dd_HH-mm-ss";

  public static void main(String[] args) {
```

```

LOGGER.info("Start task.");
ParameterTool paraTool = ParameterTool.fromArgs(args);
String checkpointInterval = "18000000";

// set up execution environment
StreamExecutionEnvironment env =
StreamExecutionEnvironment.getExecutionEnvironment();
EnvironmentSettings settings = EnvironmentSettings.newInstance()
.inStreamingMode().build();
StreamTableEnvironment tEnv = StreamTableEnvironment.create(env, settings);

env.getCheckpointConfig().setCheckpointingMode(CheckpointingMode.EXACTLY_ONCE);
env.getCheckpointConfig().setCheckpointInterval(Long.valueOf(checkpointInterval));
env.getCheckpointConfig().enableExternalizedCheckpoints(
CheckpointConfig.ExternalizedCheckpointCleanup.RETAIN_ON_CANCELLATION);

SimpleDateFormat dateTimeFormat = new SimpleDateFormat(datePattern);
String time = dateTimeFormat.format(System.currentTimeMillis());
RocksDBStateBackend rocksDbBackend =
new RocksDBStateBackend(
new FsStateBackend("obs://obs/xxx/testcheckpoint/" + time), true);
env.setStateBackend(rocksDbBackend);

String createCatalog = "CREATE CATALOG lf_catalog WITH (\n" +
" 'type' = 'hive',\n" +
" 'hive-conf-dir' = '/opt/hadoop/conf'\n" +
" );";
tEnv.executeSql(createCatalog);

String dataSource = "CREATE TABLE if not exists
lf_catalog.`testdb`.`dataGenSourceJar618_1` (\n" +
" user_id string,\n" +
" amount int\n" +
") WITH (\n" +
" 'connector' = 'datagen',\n" +
" 'rows-per-second' = '1',\n" +
" 'fields.user_id.kind' = 'random',\n" +
" 'fields.user_id.length' = '3'\n" +
");";
/*testdb是用户自定义的数数据库*/

tEnv.executeSql(dataSource);

String printSink = "CREATE TABLE if not exists lf_catalog.`testdb`.`printSinkJar618_1`
(\n" +
" user_id string,\n" +
" amount int\n" +
") WITH ('connector' = 'print');";
tEnv.executeSql(printSink);
/*testdb是用户自定义的数数据库*/

String query = "insert into lf_catalog.`test`.`printSinkJar618_1` " +
"select * from lf_catalog.`test`.`dataGenSourceJar618_1`;";
tEnv.executeSql(query);
}
}

```

ii. 创建Flink jar作业并配置如下参数。

参数	说明	配置示例
Flink版本	Flink 1.15及以上版本支持对接LakeFormation。	1.15

参数	说明	配置示例
委托	<p>使用Flink 1.15及以上版本的引擎执行作业时，需要您先在IAM页面创建相关委托，并在此处添加新建的委托信息。选择该参数后系统将自动为您的作业添加以下配置：</p> <p><code>flink.dli.job.agency.name=agency</code></p> <p>委托权限示例请参考创建DLI自定义委托权限和常见场景的委托权限策略。</p>	-
优化参数	<ul style="list-style-type: none"> 配置Flink作业访问的元数据类型。本场景下请选择Lakeformation。 <code>flink.dli.job.catalog.type=lakeformation</code> 配置Flink作业访问的数据目录名称。 <code>flink.dli.job.catalog.name=[lakeformation中的catalog名称]</code> <p>此处选择的是在DLI管理控制台创建的数据目录，即DLI与Lakeformation默认实例下的数据目录的映射，该数据目录连接的是LakeFormation默认实例下的数据目录。</p>	-

- **示例2：DEW方式对接Lakeformation**

- i. 开发Flink jar程序，编译并上传jar包到obs，本例上传到obs://obs-test/dlitest/目录

示例代码如下：

本例通过DataGen表产生随机数据并输出到Print结果表中。

其他connector类型可参考[Flink 1.15支持的connector列表](#)。

```
package com.huawei.test;

import org.apache.flink.api.java.utils.ParameterTool;
import org.apache.flink.contrib.streaming.state.RocksDBStateBackend;
import org.apache.flink.runtime.state.filesystem.FsStateBackend;
import org.apache.flink.streaming.api.CheckpointingMode;
import org.apache.flink.streaming.api.environment.CheckpointConfig;
import org.apache.flink.streaming.api.environment.StreamExecutionEnvironment;
import org.apache.flink.table.api.EnvironmentSettings;
import org.apache.flink.table.api.bridge.java.StreamTableEnvironment;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import java.text.SimpleDateFormat;

@SuppressWarnings({"deprecation", "rawtypes", "unchecked"})
public class GenToPrintTaskDew {
    private static final Logger LOGGER =
        LoggerFactory.getLogger(GenToPrintTaskAgency.class);
    private static final String datePattern = "yyyy-MM-dd_HH-mm-ss";

    public static void main(String[] args) {
        LOGGER.info("Start task.");
        ParameterTool paraTool = ParameterTool.fromArgs(args);
        String checkpointInterval = "180000000";

        // set up execution environment
        StreamExecutionEnvironment env =
```

```

StreamExecutionEnvironment.getExecutionEnvironment();
EnvironmentSettings settings = EnvironmentSettings.newInstance()
    .inStreamingMode().build();
StreamTableEnvironment tEnv = StreamTableEnvironment.create(env, settings);

env.getCheckpointConfig().setCheckpointingMode(CheckpointingMode.EXACTLY_ONCE);
env.getCheckpointConfig().setCheckpointInterval(Long.valueOf(checkpointInterval));
env.getCheckpointConfig().enableExternalizedCheckpoints(
    CheckpointConfig.ExternalizedCheckpointCleanup.RETAIN_ON_CANCELLATION);

SimpleDateFormat dateTimeFormat = new SimpleDateFormat(datePattern);
String time = dateTimeFormat.format(System.currentTimeMillis());
RocksDBStateBackend rocksDbBackend =
    new RocksDBStateBackend(
        new FsStateBackend("obs://obs/xxx/testcheckpoint/" + time), true);
env.setStateBackend(rocksDbBackend);

String createCatalog = "CREATE CATALOG lf_catalog WITH (\n" +
    " 'type' = 'hive',\n" +
    " 'hive-conf-dir' = '/opt/hadoop/conf',\n" +
    " 'properties.catalog.lakeformation.auth.identity.util.class' =
'com.huawei.flink.provider.lakeformation.FlinkDewIdentityGenerator',\n" +
    " 'properties.catalog.dew.endpoint'='kms.xxx.xxx.com',\n" +
    " 'properties.catalog.dew.csms.secretName'='obsAksK',\n" +
    " 'properties.catalog.dew.access.key' = 'ak',\n" +
    " 'properties.catalog.dew.secret.key' = 'sk',\n" +
    " 'properties.catalog.dew.projectId'='330e068af1334c9782f4226xxxxxxx',\n" +
    " 'properties.catalog.dew.csms.version'='v9'\n" +
    " );";
tEnv.executeSql(createCatalog);

String dataSource = "CREATE TABLE if not exists
lf_catalog.`testdb`.`dataGenSourceJarDew618_1` (\n" +
    " user_id string,\n" +
    " amount int\n" +
    ") WITH (\n" +
    " 'connector' = 'datagen',\n" +
    " 'rows-per-second' = '1',\n" +
    " 'fields.user_id.kind' = 'random',\n" +
    " 'fields.user_id.length' = '3'\n" +
    ")";
tEnv.executeSql(dataSource);
/*testdb是用户自定义的数数据库*/

String printSink = "CREATE TABLE if not exists
lf_catalog.`testdb`.`printSinkJarDew618_1` (\n" +
    " user_id string,\n" +
    " amount int\n" +
    ") WITH ('connector' = 'print');
tEnv.executeSql(printSink);
/*testdb是用户自定义的数数据库*/

String query = "insert into lf_catalog.`test`.`printSinkJarDew618_1` " +
    "select * from lf_catalog.`test`.`dataGenSourceJarDew618_1`;
tEnv.executeSql(query);
}
}

```

ii. 创建Flink jar作业并配置如下参数。

参数	说明	配置示例
Flink版本	Flink 1.15及以上版本支持对接LakeFormation。	1.15

参数	说明	配置示例
委托	<p>使用Flink 1.15及以上版本的引擎执行作业时，需要您先在IAM页面创建相关委托，并在此处添加新建的委托信息。选择该参数后系统将自动为您的作业添加以下配置：</p> <p><code>flink.dli.job.agency.name=agency</code></p> <p>委托权限示例请参考创建DLI自定义委托权限和常见场景的委托权限策略。</p>	-
优化参数	<ul style="list-style-type: none"> 配置Flink作业访问的元数据类型。本场景下请选择Lakeformation。 <code>flink.dli.job.catalog.type=lakeformation</code> 配置Flink作业访问的数据目录名称。 <code>flink.dli.job.catalog.name=[lakeformation中的catalog名称]</code> <p>此处选择的是在DLI管理控制台创建的数据目录，即DLI与Lakeformation默认实例下的数据目录的映射，该数据目录连接的是LakeFormation默认实例下的数据目录。</p>	-

- **示例3：Flink jar支持Hudi表**

- i. 开发Flink jar程序，编译并上传jar包到obs，本例上传到obs://obs-test/dlitest/目录

示例代码如下：

本例通过DataGen表产生随机数据并输出到Hudi结果表中。

其他connector类型可参考[Flink 1.15支持的connector列表](#)。

```
package com.huawei.test;

import org.apache.flink.api.java.utils.ParameterTool;
import org.apache.flink.contrib.streaming.state.RocksDBStateBackend;
import org.apache.flink.runtime.state.filesystem.FsStateBackend;
import org.apache.flink.streaming.api.CheckpointingMode;
import org.apache.flink.streaming.api.environment.CheckpointConfig;
import org.apache.flink.streaming.api.environment.StreamExecutionEnvironment;
import org.apache.flink.table.api.EnvironmentSettings;
import org.apache.flink.table.api.bridge.java.StreamTableEnvironment;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import java.io.IOException;
import java.text.SimpleDateFormat;

public class GenToHudiTask4 {
    private static final Logger LOGGER = LoggerFactory.getLogger(GenToHudiTask4.class);
    private static final String datePattern = "yyyy-MM-dd_HH-mm-ss";

    public static void main(String[] args) throws IOException {
        LOGGER.info("Start task.");
        ParameterTool paraTool = ParameterTool.fromArgs(args);
        String checkpointInterval = "30000";

        // set up execution environment
        StreamExecutionEnvironment env =
            StreamExecutionEnvironment.getExecutionEnvironment();
```

```
EnvironmentSettings settings = EnvironmentSettings.newInstance()
    .inStreamingMode().build();
StreamTableEnvironment tEnv = StreamTableEnvironment.create(env, settings);

env.getCheckpointConfig().setCheckpointingMode(CheckpointingMode.EXACTLY_ONCE);
env.getCheckpointConfig().setCheckpointInterval(Long.valueOf(checkpointInterval));
env.getCheckpointConfig().enableExternalizedCheckpoints(
    CheckpointConfig.ExternalizedCheckpointCleanup.RETAIN_ON_CANCELLATION);

SimpleDateFormat dateTimeFormat = new SimpleDateFormat(datePattern);
String time = dateTimeFormat.format(System.currentTimeMillis());
RocksDBStateBackend rocksDbBackend =
    new RocksDBStateBackend(
        new FsStateBackend("obs://xxx/jobs/testcheckpoint/" + time), true);
env.setStateBackend(rocksDbBackend);

String catalog = "CREATE CATALOG hoodie_catalog\n" +
    " WITH (\n" +
    "   'type'='hudi',\n" +
    "   'hive.conf.dir' = '/opt/hadoop/conf',\n" +
    "   'mode'='hms'\n" +
    " );";
tEnv.executeSql(catalog);
String dwsSource = "CREATE TABLE if not exists genSourceJarForHudi618_1 (\n" +
    "   order_id STRING,\n" +
    "   order_name STRING,\n" +
    "   price INT,\n" +
    "   weight INT\n" +
    ") WITH (\n" +
    "   'connector' = 'datagen',\n" +
    "   'rows-per-second' = '1',\n" +
    "   'fields.order_id.kind' = 'random',\n" +
    "   'fields.order_id.length' = '8',\n" +
    "   'fields.order_name.kind' = 'random',\n" +
    "   'fields.order_name.length' = '8'\n" +
    ")";
tEnv.executeSql(dwsSource);
/*testdb是用户自定义的数数据库*/
String printSinkdws =
    "CREATE TABLE if not exists hoodie_catalog.`testdb`.`hudiSinkJarHudi618_1`
(\n" +
    "   order_id STRING PRIMARY KEY NOT ENFORCED,\n" +
    "   order_name STRING,\n" +
    "   price INT,\n" +
    "   weight INT,\n" +
    "   create_time BIGINT,\n" +
    "   create_date String\n" +
    ") WITH (" +
    "   'connector' = 'hudi',\n" +
    "   'path' = 'obs://xxx/catalog/dbtest3/hudiSinkJarHudi618_1',\n" +
    "   'hoodie.datasource.write.recordkey.field' = 'order_id',\n" +
    "   'EXTERNAL' = 'true'\n" +
    ")";
tEnv.executeSql(printSinkdws);
/*testdb是用户自定义的数数据库*/
String query = "insert into hoodie_catalog.`testdb`.`hudiSinkJarHudi618_1` select\n" +
    "   order_id,\n" +
    "   order_name,\n" +
    "   price,\n" +
    "   weight,\n" +
    "   UNIX_TIMESTAMP() as create_time,\n" +
    "   FROM_UNIXTIME(UNIX_TIMESTAMP(), 'yyyyMMdd') as create_date\n" +
    " from genSourceJarForHudi618_1";
tEnv.executeSql(query);
}
}
```


表 4-20 hudi 类型 sink 表的 connector 参数

参数	说明	是否必填	参数值
connector	flink connector类型。 配置为hudi表示sink表是hudi表。	是	hudi
path	表的基本路径。 如果该路径不存在，则会创建它。	是	请参考示例代码中的配置值。
hoodie.datasource.write.recordkey.field	hoodie表的唯一键字段名	否	这里配置order_id为唯一键。
EXTERNAL	是否外表	是 hudi表必填，且设置为true	true

ii. 创建Flink jar作业并配置如下参数。

参数	说明	配置示例
Flink版本	Flink 1.15及以上版本支持对接LakeFormation。	1.15
委托	使用Flink 1.15及以上版本的引擎执行作业时，需要您先在IAM页面创建相关委托，并在此处添加新建的委托信息。选择该参数后系统将自动为您的作业添加以下配置： flink.dli.job.agency.name= <i>agency</i> 委托权限示例请参考 创建DLI自定义委托权限和常见场景的委托权限策略 。	-
优化参数	<ul style="list-style-type: none"> 配置Flink作业访问的元数据类型。本场景下请选择Lakeformation。 flink.dli.job.catalog.type=lakeformation 配置Flink作业访问的数据目录名称。 flink.dli.job.catalog.name=[lakeformation中的catalog名称] 此处选择的是在DLI管理控制台创建的数据目录，即DLI与Lakeformation默认实例下的数据目录的映射，该数据目录连接的是LakeFormation默认实例下的数据目录。 	-

4.6.2 LakeFormation 资源权限支持列表与策略项

LakeFormation SQL 资源权限支持列表

DLI支持SQL资源鉴权的操作列表请参考[数据权限列表](#)。

LakeFormation SQL资源权限支持列表请参考[表4-21](#)。

表 4-21 LakeFormation SQL 资源权限支持列表

资源类型	权限类型
Database	ALL
	ALTER
	DROP
	DESCRIBE
	LIST_TABLE
	LIST_FUNC
	CREATE_TABLE
	CREATE_FUNC
Table/View	ALL
	ALTER
	DROP
	DESCRIBE
	UPDATE
	INSERT
	SELECT
	DELETE
Column	SELECT
Function	ALL
	ALTER
	DROP
	DESCRIBE
	EXEC

Lakeformation 权限策略

表 4-22 Lakeformation 权限策略

类型	SQL语句	元数据IAM鉴权权限	SQL资源鉴权权限
DDL语句	ALTER DATABASE	database:describe database:alter	database:DESCRIBE database:ALTER
	ALTER TABLE	database:describe table:describe table:alter database:create	database:DESCRIBE table:DESCRIBE table:ALTER database:CREATE_TABLE column:SELECT或 table:SELECT
	ALTER VIEW	database:describe table:describe table:alter	database:DESCRIBE table:DESCRIBE column:SELECT table:ALTER
	CREATE DATABASE	database:describe database:create	database:DESCRIBE catalog:CREATE_DATABASE
	CREATE OR REPLACE FUNCTION (CREATE)	database:describe function:create	database:DESCRIBE database:CREATE_FUNCTION
	CREATE OR REPLACE FUNCTION (REPLACE)	database:describe function:describe function:alter	database:CREATE_FUNCTION database:DESCRIBE function:DESCRIBE function:ALTER
	CREATE TABLE	database:describe table:describe table:create	database:DESCRIBE database:CREATE_TABLE
	CREATE VIEW	database:describe table:describe table:drop table:create	database:CREATE_TABLE table:DESCRIBE (source\targer) table:DROP(target) column:SELECT

类型	SQL语句	元数据IAM鉴权权限	SQL资源鉴权权限
	DROP DATABASE	database:describe database:drop	database:DESCRIBE database:DROP
	DROP FUNCTION	database:describe function:describe function:drop	database:DESCRIBE function:DESCRIBE function:DROP
	DROP TABLE	database:describe table:describe credential:describe table:drop	database:DESCRIBE table:DESCRIBE table:DROP
	DROP VIEW	database:describe table:describe table:drop	database:DESCRIBE table:DESCRIBE(target \source) table:DROP(target)
	REPAIR TABLE	database:describe table:describe credential:describe table:alter	database:DESCRIBE table:DESCRIBE table:ALTER table:SELECT
	TRUNCATE TABLE	database:describe table:describe table:alter	database:DESCRIBE table:DESCRIBE table:SELECT table:UPDATE
DML语句	INSERT TABLE	database:describe table:describe table:alter credential:describe	database:DESCRIBE table:DESCRIBE table:ALTER table:INSERT column:SELECT或 table:SELECT
	LOAD DATA	database:describe table:describe credential:describe	database:DESCRIBE table:DESCRIBE table:UPDATE table:ALTER table:SELECT
DR语句	SELECT	database:describe table:describe credential:describe	database:DESCRIBE table:DESCRIBE column:SELECT
	EXPLAIN	取决于执行sql	取决于执行sql

类型	SQL语句	元数据IAM鉴权权限	SQL资源鉴权权限
Auxiliary 语句	ANALYZE TABLE	database:describe table:describe credential:describe table:alter	database:DESCRIBE table:DESCRIBE table:SELECT table:ALTER
	DESCRIBE DATABASE	database:describe	database:DESCRIBE
	DESCRIBE FUNCTION	database:describe function:describe	database:DESCRIBE function:DESCRIBE
	DESCRIBE QUERY	database:describe table:describe	database:DESCRIBE table:DESCRIBE table:SELECT
	DESCRIBE TABLE	database:describe table:describe	database:DESCRIBE table:DESCRIBE
	REFRESH TABLE	database:describe table:describe credential:describe	database:DESCRIBE table:DESCRIBE table:SELECT
	REFRESH FUNCTION	database:describe function:describe	database:DESCRIBE function:DESCRIBE
	SHOW COLUMNS	database:describe table:describe	database:DESCRIBE table:DESCRIBE
	SHOW CREATE TABLE	database:describe table:describe	database:DESCRIBE table:DESCRIBE
	SHOW DATABASES	database:describe	catalog:LIST_DATABASE database:DESCRIBE
	SHOW FUNCTIONS	database:describe function:describe	database:DESCRIBE
	SHOW PARTITIONS	database:describe table:describe	database:DESCRIBE table:DESCRIBE
	SHOW TABLE EXTENDED	database:describe table:describe	catalog:LIST_DATABASE database:DESCRIBE table:DESCRIBE database:LIST_TABLE

类型	SQL语句	元数据IAM鉴权权限	SQL资源鉴权权限
	SHOW TABLES	database:describe table:describe	catalog:LIST_DATABASE database:LIST_TABLE database:DESCRIBE
	SHOW TBLPROPERTIES	database:describe table:describe	database:DESCRIBE table:DESCRIBE
	SHOW VIEWS	database:describe table:describe	catalog:LIST_DATABASE database:LIST_TABLE database:DESCRIBE

5 数据迁移与数据传输

5.1 数据迁移与传输方式概述

导入数据至 OBS

DLI支持在不迁移数据的情况下，直接访问OBS中存储的数据进行查询分析。

您只需将本地数据导入OBS即可开始使用DLI进行数据分析。

导入数据的具体操作请参考[上传对象](#)。

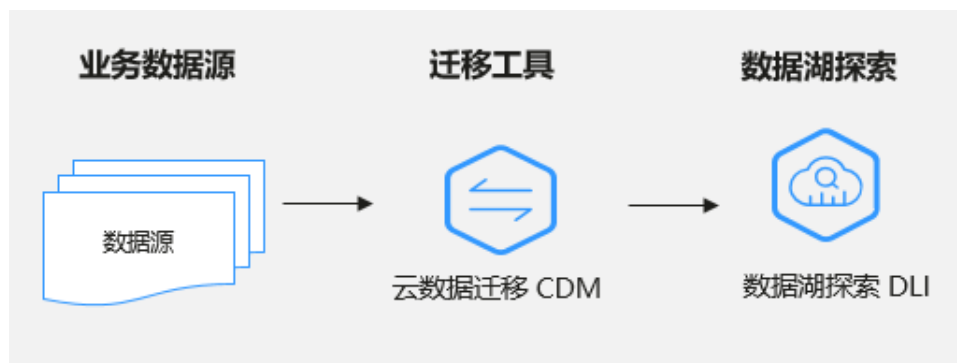
迁移数据至 DLI

为了将分散在不同系统中的数据迁移到DLI，确保数据可以在DLI集中分析和管理，您可以通过云数据迁移服务CDM等迁移工具迁移数据至DLI，再使用DLI提交作业分析数据。

CDM支持数据库、数据仓库、文件等多种类型的数据源，通过可视化界面对数据源迁移任务进行配置，提高数据迁移和集成的效率。

具体操作请参考[迁移外部数据源数据至DLI](#)。

图 5-1 迁移数据至 DLI



配置 DLI 读写外部数据源

如果您不想将数据导入OBS或DLI的数据表中，DLI提供的跨源访问能力，支持您在不迁移数据的情况下，连接数据源获取数据并进行数据分析。

具体操作请参考[配置DLI读写外部数据源数据的操作流程](#)。

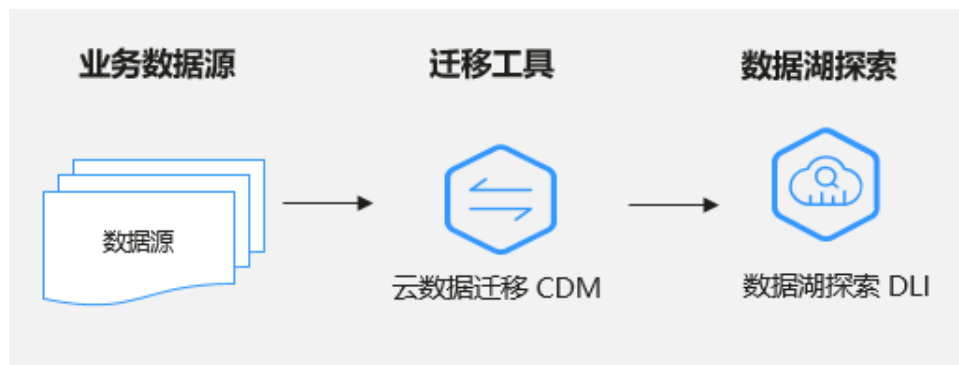
5.2 迁移外部数据源数据至 DLI

5.2.1 迁移数据场景概述

为了将分散在不同系统中的数据迁移到DLI，确保数据可以在DLI集中分析和管理，您可以通过云数据迁移服务CDM等迁移工具迁移数据至DLI，再使用DLI提交作业分析数据。

CDM支持数据库、数据仓库、文件等多种类型的数据源，通过可视化界面对数据源迁移任务进行配置，提高数据迁移和集成的效率。

图 5-2 迁移数据至 DLI



常见迁移场景与迁移方案指导

表 5-1 常见迁移场景与迁移方案指导

数据类型	迁移工具	迁移方案
Hive	CDM	典型场景示例：迁移Hive数据至DLI
Kafka	CDM	典型场景示例：迁移Kafka数据至DLI
Elasticsearch	CDM	典型场景示例：迁移Elasticsearch数据至DLI
RDS	CDM	典型场景示例：迁移RDS数据至DLI
DWS	CDM	典型场景示例：迁移DWS数据至DLI

数据迁移数据类型映射

将其他云服务或业务平台数据迁移到DLI，或者将DLI数据迁移到其他云服务或业务平台时，涉及到源和目的端数据类型的转换和映射，根据表5-2可以获取到源和目的端的数据类型映射关系。

表 5-2 数据类型映射表

MySQL	Hive	DWS	Oracle	PostgreSQL	Hologres	DLI Spark
CHAR	CHAR	CHAR	CHAR	CHAR	CHAR	CHAR
VARCHAR	VARCHAR	VARCHAR	VARCHAR	VARCHAR	VARCHAR	VARCHAR/ STRING
DECIMAL	DECIMAL	NUMERIC	NUMERIC	NUMERIC	DECIMAL	DECIMAL
INT	INT	INTEGER	NUMBER	INTEGER	INTEGER	INT
BIGINT	BIGINT	BIGINT	NUMBER	BIGINT	BIGINT	BIGINT/ LONG
TINYINT	TINYINT	SMALLINT	NUMBER	SMALLINT	SMALLINT	TINYINT
SMALLINT	SMALLINT	SMALLINT	NUMBER	SMALLINT	SMALLINT	SMALLINT/ SHORT
BINARY	BINARY	BYTEA	RAW	BYTEA	BYTEA	BINARY
VARBINARY	BINARY	BYTEA	RAW	BYTEA	BYTEA	BINARY
FLOAT	FLOAT	FLOAT4	FLOAT	DOUBLE	FLOAT4	FLOAT
DOUBLE	DOUBLE	FLOAT8	FLOAT	REAL/ DOUBLE	FLOAT8	DOUBLE
DATE	DATE	TIMESTAMP	DATE	DATE	DATE	DATE
TIME	不支持 (推荐使用: String)	TIME	DATE	TIME	TIME	不支持 (推荐使用: String)
DATETIME	TIMESTAMP	TIMESTAMP	TIME	TIME	TIMESTAMP	TIMESTAMP
TINYINT	TINYINT	BOOLEAN	不支持	TINYINT	BOOLEAN	BOOLEAN

MySQL	Hive	DWS	Oracle	PostgreSQL	Hologres	DLI Spark
不支持 (推荐使用: TEXT)	不支持 (推荐使用: String)	不支持 (推荐使用: TEXT)	不支持 (推荐使用: VARCHAR)	不支持 (推荐使用: TEXT)	不支持 (推荐使用: TEXT)	ARRAY
不支持 (推荐使用: TEXT)	不支持 (推荐使用: String)	不支持 (推荐使用: TEXT)	不支持 (推荐使用: VARCHAR)	不支持 (推荐使用: TEXT)	不支持 (推荐使用: TEXT)	MAP
不支持 (推荐使用: TEXT)	不支持 (推荐使用: String)	不支持 (推荐使用: TEXT)	不支持 (推荐使用: VARCHAR)	不支持 (推荐使用: TEXT)	不支持 (推荐使用: TEXT)	STRUCT

说明

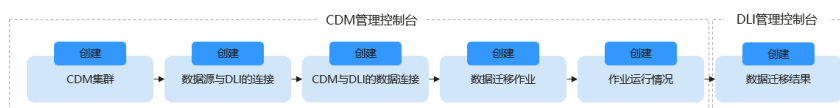
推荐使用：表示当前服务没有支持的标准数据类型，可以使用推荐的数据类型来替换使用。

5.2.2 使用 CDM 迁移数据至 DLI

CDM提供了可视化的迁移任务配置页面，支持多种数据源到数据湖的迁移能力。

本节操作介绍使用CDM迁移工具将数据从数据源迁移至DLI的操作步骤。

图 5-3 使用 CDM 迁移数据至 DLI 操作流程



步骤 1：创建 CDM 集群

CDM集群用于执行数据迁移作业，将数据从数据源迁移至DLI。

1. 登录CDM管理控制台。
2. 单击“购买云数据迁移服务”，进入创建CDM集群的界面，配置集群参数。
 - 其中CDM集群的区域、虚拟私有云、子网、安全组、企业项目建议选择与数据源和DLI一致。
 - 集群创建好以后不支持修改规格，如果需要使用更高规格，需要重新创建。
 更多CDM集群参数配置说明请参考[创建集群](#)。
3. 确认无误后单击“立即购买”进入规格确认界面。
4. 单击“提交”，系统开始自动创建CDM集群，在“集群管理”界面可查看创建进度。

步骤 2: 创建数据源与 CDM 的数据连接

本例以MySQL数据源为例，介绍创建数据源与CDM的数据连接的操作步骤。

- 步骤1** 进入CDM主界面，单击左侧导航上的“集群管理”，找到**步骤1: 创建CDM集群**章节创建的集群“cdm-aff1”。
- 步骤2** 单击CDM集群后的“作业管理”，进入作业管理界面。
- 步骤3** 选择“连接管理 > 新建连接”，进入选择连接器类型的界面，如图5-4所示。

图 5-4 选择连接器类型



- 步骤4** 选择“云数据库 MySQL”后单击“下一步”，配置云数据库 MySQL连接的参数。

单击“显示高级属性”可查看更多可选参数，具体请参见[配置云数据库MySQL/MySQL数据库连接](#)。这里保持默认，必填参数如表5-3所示。

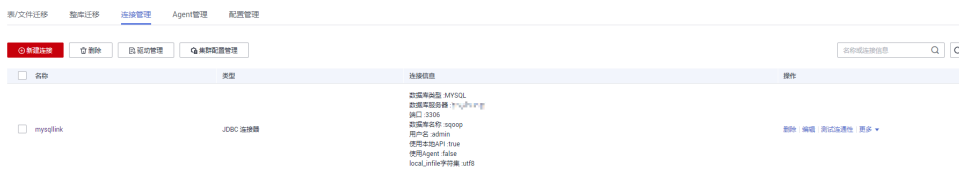
表 5-3 MySQL 连接参数

参数名	说明	取值样例
名称	输入便于记忆和区分的连接名称。	mysqllink
数据库服务器	MySQL数据库的IP地址或域名。	-
端口	MySQL数据库的端口。	3306
数据库名称	MySQL数据库的名称。	sqoop
用户名	拥有MySQL数据库的读、写和删除权限的用户。	admin
密码	用户的密码。	-

参数名	说明	取值样例
使用本地API	使用数据库本地API加速（系统会尝试启用MySQL数据库的local_infile系统变量）。	是
使用Agent	Agent功能待下线，无需配置。	-
local_infile字符集	MySQL通过local_infile导入数据时，可配置编码格式。	utf8
驱动版本	CDM连接关系数据库前，需要先上传所需关系数据库的JDK8版本.jar格式驱动。MySQL的驱动请从 https://downloads.mysql.com/archives/c-j/ 选择5.1.48版本下载，从中获取mysql-connector-java-5.1.48.jar，然后进行上传。	-

步骤5 单击“测试”测试参数是否配置无误，“测试”成功后单击“保存”创建该连接，并回到连接管理界面。

图 5-5 创建 MySQL 连接成功



----结束

步骤 3: 创建 CDM 与 DLI 的数据连接

步骤1 进入CDM主界面，单击左侧导航上的“集群管理”，找到**步骤1: 创建CDM集群**章节创建的集群“cdm-aff1”。

步骤2 单击CDM集群后的“作业管理”，进入作业管理界面。

步骤3 选择“连接管理 > 新建连接”，进入选择连接器类型的界面，如图5-6所示。

图 5-6 选择连接器类型




步骤4 选择“数据湖探索（DLI）”后单击“下一步”，配置DLI连接的参数。

单击“显示高级属性”可查看更多可选参数，具体请参见[配置DLI连接](#)。这里保持默认，必填参数如[表5-4](#)所示。

表 5-4 DLI 连接参数

参数名	说明	取值样例
名称	连接的名称，根据连接的数据源类型，用户可自定义便于记忆、区分的连接名。	dlilink

参数名	说明	取值样例
访问标识(AK)	访问DLI数据库时鉴权所需的AK和SK。	-
密钥(SK)	<p>您需要先创建当前账号的访问密钥，并获得对应的AK和SK。</p> <ol style="list-style-type: none"> 登录控制台，在用户名下拉列表中选择“我的凭证”。 进入“我的凭证”页面，选择“访问密钥 > 新增访问密钥”，如图5-7所示。 <p>图 5-7 单击新增访问密钥</p>  <ol style="list-style-type: none"> 单击“确定”，根据浏览器提示，保存密钥文件。密钥文件会直接保存到浏览器默认的下载文件夹中。打开名称为“credentials.csv”的文件，即可查看访问密钥（Access Key Id和Secret Access Key）。 <p>说明</p> <ul style="list-style-type: none"> 每个用户仅允许新增两个访问密钥。 为保证访问密钥的安全，访问密钥仅在初次生成时自动下载，后续不可再次通过管理控制台界面获取。请在生成后妥善保管。 	-
项目ID	<p>DLI服务所在区域的项目ID。</p> <p>项目ID表示租户的资源，账号ID对应当前账号，IAM用户ID对应当前用户。用户可在对应页面下查看不同Region对应的项目ID、账号ID和用户ID。</p> <ol style="list-style-type: none"> 注册并登录管理控制台。 在用户名的下拉列表中单击“我的凭证”。 在“API凭证”页面，查看账号名和账号ID、IAM用户名和IAM用户ID，在项目列表中查看项目和项目ID。 	-

步骤5 单击“测试”测试参数是否配置无误，“测试”成功后单击“保存”创建该连接，并回到连接管理界面。

----结束

步骤 4：在 CDM 上创建数据迁移作业

建立完成数据源与CDM、CDM与DLI的数据连接后，需要创建数据迁移作业将数据从数据源迁移至DLI。

步骤1 在集群管理界面，找到**步骤1：创建CDM集群**章节创建的集群“cdm-aff1”。

步骤2 单击该CDM集群后的“作业管理”，进入作业管理界面。

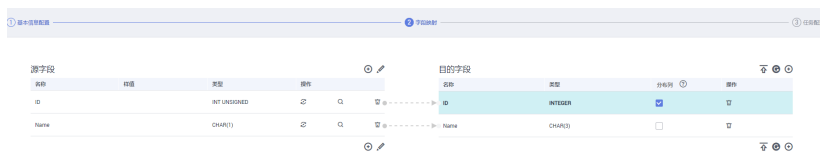
步骤3 选择“表/文件迁移 > 新建作业”，配置作业基本信息。

- 作业名称：输入便于记忆、区分的作业名称，例如：“mysql2dli”。
- 源端作业配置
 - 源连接名称：选择已创建的MySQL连接“mysqllink”。
 - 使用SQL语句：选择“否”。
 - 模式或表空间：选择从MySQL的哪个数据库导出表。
 - 表名：选择导出哪张表。
 - 其他可选参数保持默认即可，详细说明可参见[配置MySQL源端参数](#)。
- 目的端作业配置
 - 目的连接名称：选择已创建的DLI连接“dlilink”。
 - 模式或表空间：选择导入到DLI的哪个模式。
 - 自动创表：这里选择“不存在时创建”，当下面“表名”参数中配置的表不存在时，CDM会自动在DLI中创建该表。
 - 表名：选择导入到DLI的哪张表。
 - 高级属性参数-“扩大字符字段长度”：这里选择“是”。由于MySQL和DLI存储中文时编码不一样，所需的长度也不一样，一个中文字符在UTF-8编码下可能要多占3个字节。该参数选择为“是”后，在DLI中自动创表时，会将字符类型的字段长度设置为原表的3倍，避免出现DLI表的字符字段长度不够的报错。
 - 其他可选参数保持默认即可，详细说明可参见[配置DWS目的端参数](#)。

步骤4 单击“下一步”进入字段映射界面，CDM会自动匹配源端和目的端的数据表字段，需用户检查字段映射关系是否正确。

- 如果字段映射关系不正确，用户单击字段所在行选中后，按住鼠标左键可拖拽字段来调整映射关系。
- 导入到DLI时需要手动选择DLI的分布列，建议按如下顺序选取：
 - a. 有主键可以使用主键作为分布列。
 - b. 多个数据段联合做主键的场景，建议设置所有主键作为分布列。
 - c. 在没有主键的场景下，如果没有选择分布列，DWS会默认第一列作为分布列，可能会有数据倾斜风险。
- 如果需要转换源端字段内容，可在该步骤配置，详细请参见[字段转换](#)，这里选择不进行字段转换。

图 5-8 字段映射



步骤5 单击“下一步”配置任务参数，一般情况下全部保持默认即可。

该步骤用户可以配置如下可选功能：

- 作业失败重试：如果作业执行失败，可选择是否自动重试，这里保持默认值“不重试”。

- 作业分组：选择作业所属的分组，默认分组为“DEFAULT”。在CDM“作业管理”界面，支持作业分组显示、按组批量启动作业、按分组导出作业等操作。
- 是否定时执行：如果需要配置作业定时自动执行可开启。这里保持默认值“否”。
- 抽取并发数：设置同时执行的抽取任务数，适当的抽取并发数可以提升迁移效率，配置原则请参见[性能调优](#)。这里保持默认值“1”。
- 是否写入脏数据：如果需要将作业执行过程中处理失败的数据、或者被清洗过滤掉的数据写入OBS中，以便后面查看，可通过该参数配置，写入脏数据前需要在CDM先配置好OBS连接。这里保持默认值“否”即可，不记录脏数据。

图 5-9 任务配置

任务配置

作业失败重试 ?

作业分组 ? + 添加 ✎ 编辑 🗑️ 删除

是否定时执行

[隐藏高级属性](#)

抽取并发数 ?

分片重试次数 ?

是否写入脏数据 ?

开启限速 ?

步骤6 单击“保存并运行”，CDM立即开始执行作业。

图 5-10 作业执行

名称	源	目标	创建时间	状态	组名	操作
mysql2des	mysql2des	dgc_desc	2022/09/23 20:59:58 GMT+08:00	Running	DEFAULT	运行 历史记录 编辑 更多

----结束

步骤 5：查看数据迁移结果

作业完成后，可以查看作业执行结果及最近90天内的历史信息，包括写入行数、读取行数、写入字节、写入文件数和日志等信息。

- 在CDM查看迁移作业运行情况
 - a. 在集群管理界面，找到[步骤1：创建CDM集群](#)章节创建的集群“cdm-aff1”。

- b. 单击该CDM集群后的“作业管理”，进入作业管理界面。
 - c. 找到**步骤4：在CDM上创建数据迁移作业**创建的作业“mysql2dli”，查看该作业的执行状态。作业状态为Succeeded即迁移成功。
- **在DLI查看数据迁移结果**
 - a. 确认CDM迁移作业运行完成后，登录到DLI管理控制台。
 - b. 单击“SQL编辑器”。

在SQL编辑器中“执行引擎”选择“spark”，“队列”选择已创建的SQL队列，数据库选择已创建的数据库，执行DLI表查询语句，查询MySQL数据是否已成功迁移到DLI的表中。

```
select * from tablename;
```

5.2.3 典型场景示例：迁移 Hive 数据至 DLI

本文为您介绍如何通过CDM数据同步功能，迁移MRS Hive数据至DLI。其他MRS Hadoop组件数据，均可以通过CDM与DLI进行双向同步。

前提条件

- 已创建DLI的SQL队列。创建DLI队列的操作可以参考[创建DLI队列](#)。

注意

创建DLI队列时**队列类型**需要选择为“SQL队列”。

- 已创建包含Hive组件的MRS安全集群。创建MRS集群的操作详细可以参考[创建MRS集群](#)。
 - 本示例创建的MRS集群和各组件版本如下：
 - MRS集群版本：MRS 3.1.0
 - Hive版本：3.1.0
 - Hadoop版本：3.1.1
 - 本示例创建MRS集群时开启了Kerberos认证。
- 已创建CDM迁移集群。创建CDM集群的操作可以参考[创建CDM集群](#)。

说明

- 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。
- 数据源为云上的MRS、DWS等服务时，网络互通需满足如下条件：
 - i. CDM集群与云上服务处于不同区域的情况下，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，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。
 - ii. CDM集群与云上服务同区域情况下，同虚拟私有云、同子网、同安全组的不同实例默认网络互通；如果同虚拟私有云但是子网或安全组不同，还需配置路由规则及安全组规则。

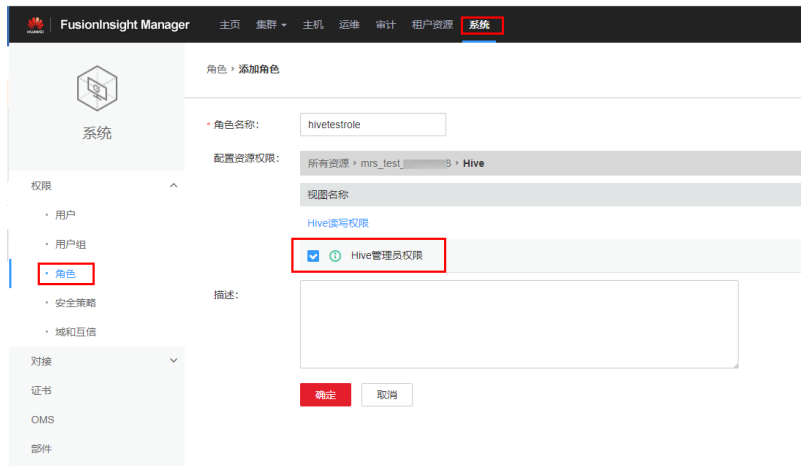
配置路由规则请参见[如何配置路由规则](#)章节，配置安全组规则请参见[如何配置安全组规则](#)章节。
 - iii. 此外，您还必须确保该云服务的实例与CDM集群所属的企业项目必须相同，如果不同，需要修改工作空间的企业项目。

本示例CDM集群的虚拟私有云、子网以及安全组和MRS集群保持一致。

步骤一：数据准备

- MRS集群上创建Hive表和插入表数据。
 - a. 参考[访问MRS Manager](#)登录MRS Manager。
 - b. 在MRS Manager上，选择“系统 > 权限 > 角色”，单击“添加角色”，在添加角色页面分别配置参数。
 - 角色名称：输入自定义的“角色名称”，例如当前输入为：hivetestrole。
 - 配置资源权限：选择“当前MRS集群的名称 > hive”，勾选“Hive管理员权限”。

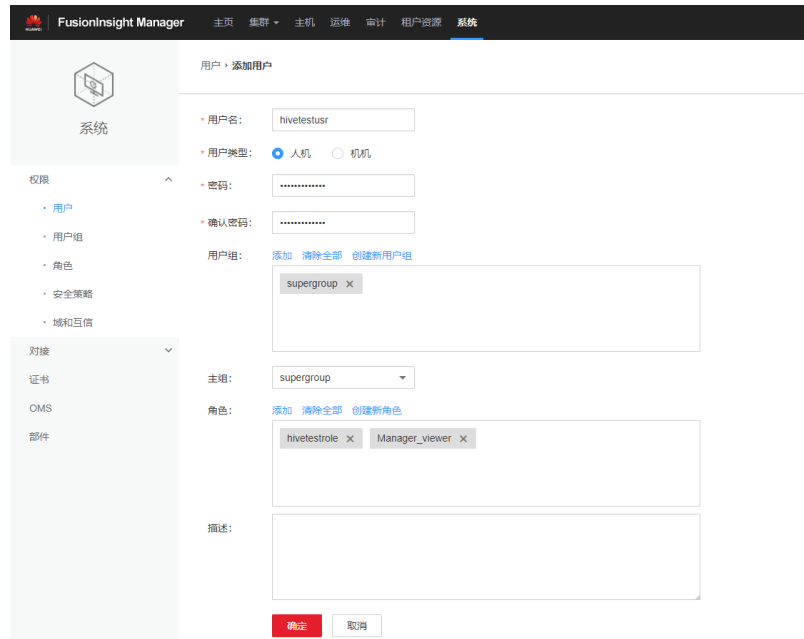
图 5-11 Manager 创建 Hive 的角色



更多MRS创建角色的操作说明可以参考：[创建Hive管理员角色](#)。

- c. 在MRS Manager上，选择“系统 > 权限 > 用户”，单击“添加用户”，在添加用户页面分别配置如下参数。
 - i. 用户名：自定义的用户名。当前示例输入为：hivetestusr。
 - ii. 用户类型：当前选择为“人机”。
 - iii. 密码和确认密码：输入当前用户名对应的密码。
 - iv. 用户组和主组：选择supergroup
 - v. 角色：同时选择**b**中创建的角色和Manager_viewer角色。

图 5-12 MRS Manager 上创建 Hive 用户



- d. 参考[安装MRS客户端](#)下载并安装Hive客户端。例如，当前Hive客户端安装在MRS主机节点的“/opt/hiveclient”目录上。
- e. 以root用户进入客户端安装目录下。
例如：`cd /opt/hiveclient`
- f. 执行以下命令配置环境变量。
source bigdata_env
- g. 因为当前集群启用了Kerberos认证，则需要执行以下命令进行安全认证。认证用户为c中创建的用户。
kinit c中创建的用户名
例如，**kinit hivetestusr**
- h. 执行以下命令连接Hive。
beeline
- i. 创建表和插入表数据。

创建表：

```
create table user_info(id string,name string,gender string,age int,addr string);
```

插入表数据：

```
insert into table user_info(id,name,gender,age,addr) values("12005000201","A","男",19,"A城市");
insert into table user_info(id,name,gender,age,addr) values("12005000202","B","男",20,"B城市");
insert into table user_info(id,name,gender,age,addr) values("12005000202","B","男",20,"B城市");
```

说明

上述示例是通过创建表和插入表数据构造迁移示例数据。如果是迁移已有的Hive数据库和表数据，则可以通过以下命令获取Hive的数据库和表信息。

- 在Hive客户端执行如下命令获取数据库信息
show databases
- 切换到需要迁移的Hive数据库
use Hive数据库名
- 显示当前数据库下所有的表信息
show tables
- 查询Hive表的建表语句
show create table Hive表名
查询出来的建表语句需要做一些处理，建表语句要符合DLI的建表语法，再到具体的DLI上执行。
- 在DLI上创建数据库和表。
 - a. 登录DLI管理控制台，选择“SQL编辑器”，在SQL编辑器中“执行引擎”选择“spark”，“队列”选择已创建的SQL队列。
在编辑器中输入以下语句创建数据库，例如当前创建迁移后的DLI数据库testdb。详细的DLI创建数据库的语法可以参考[创建DLI数据库](#)。

```
create database testdb;
```
 - b. 在数据库下创建表。

说明

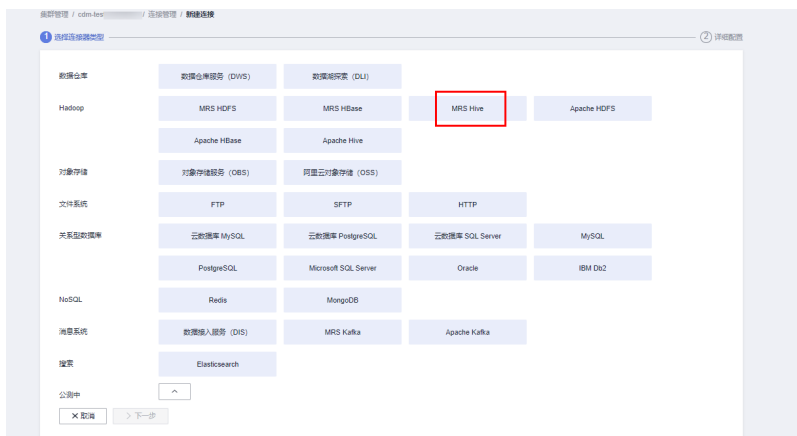
如果是通过在MRS Hive中的“show create table *hive表名*”获取的建表语句，则需要修改该建表语句以符合DLI的建表语法。具体DLI的建表语法可以参考[创建DLI表](#)。

```
create table user_info(id string,name string,gender string,age int,addr string);
```

步骤二：数据迁移

1. 配置CDM数据源连接。
 - a. 配置源端MRS Hive的数据源连接。
 - i. 登录CDM控制台，选择“集群管理”，选择已创建的CDM集群，在操作列选择“作业管理”。
 - ii. 在作业管理界面，选择“连接管理”，单击“新建连接”，连接器类型选择“MRS Hive”，单击“下一步”。

图 5-13 创建 MRS Hive 数据源连接



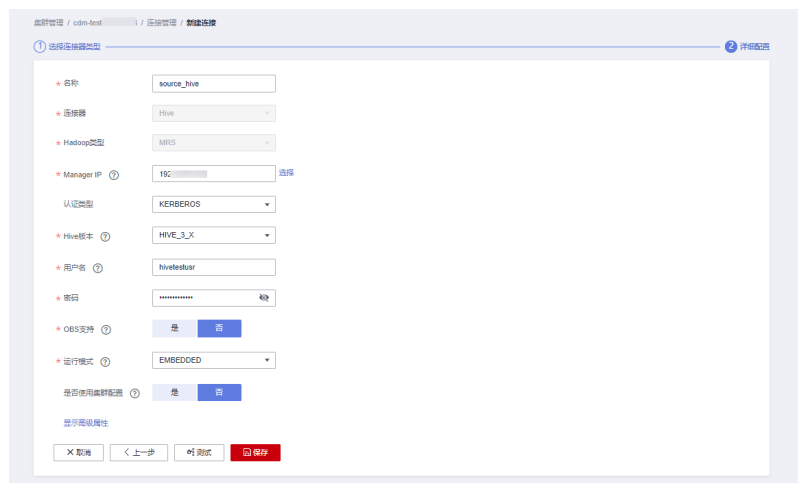
- iii. 配置源端MRS Hive的数据源连接，具体参数配置如下。

表 5-5 MRS Hive 数据源配置

参数	值
名称	自定义MRS Hive数据源名称。例如当前配置为：source_hive
Manager IP	单击输入框旁边的“选择”按钮，选择当前MRS Hive集群即可自动关联出来Manager IP。
认证类型	如果当前MRS集群为普通集群则选择为SIMPLE，如果是MRS集群启用了Kerberos安全认证则选择为KERBEROS。 本示例选择为：KERBEROS。
Hive版本	根据当前创建MRS集群时候的Hive版本确定。当前Hive版本为3.1.0，则选择为：HIVE_3_X。
用户名	在c中创建的MRS Hive用户名。
密码	对应的MRS Hive用户名的密码。

其他参数保持默认即可。更多参数的详细说明可以参考[CDM上配置Hive连接](#)。

图 5-14 CDM 配置 MRS Hive 数据源



- iv. 单击“保存”完成MRS Hive数据源配置。
- b. 配置目的端DLI的数据源连接。
- i. 登录CDM控制台，选择“集群管理”，选择已创建的CDM集群，在操作列选择“作业管理”。
 - ii. 在作业管理界面，选择“连接管理”，单击“新建连接”，连接器类型选择“数据湖探索（DLI）”，单击“下一步”。

图 5-15 创建 DLI 数据源连接



- iii. 配置目的端DLI数据源连接连接参数。具体参数配置可以参考[在CDM上配置DLI连接](#)。

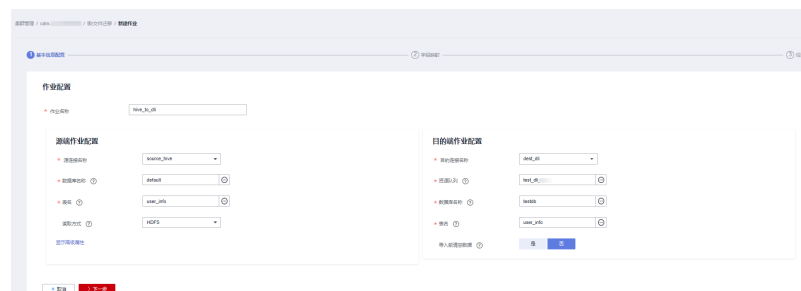
图 5-16 配置 DLI 数据源连接参数



配置完成后，单击“保存”完成DLI数据源配置。

2. 创建CDM迁移作业。
 - a. 登录CDM控制台，选择“集群管理”，选择已创建的CDM集群，在操作列选择“作业管理”。
 - b. 在“作业管理”界面，选择“表/文件迁移”，单击“新建作业”。
 - c. 在新建作业界面，配置当前作业配置信息，具体参数参考如下：

图 5-17 新建 CDM 作业配置



- i. 作业名称：自定义数据迁移的作业名称。例如，当前定义为：hive_to_dli。

- ii. 源端作业配置，具体参考如下：

表 5-6 源端作业配置

参数名	参数值
源连接名称	选择1.a中已创建的数据源名称。
数据库名称	选择MRS Hive待迁移的数据库名称。例如当前待迁移的表数据数据库为“default”。
表名	待建议Hive数据表名。当前示例为在DLI上创建数据库和表中的“user_info”表。
读取方式	当前示例选择为：HDFS。具体参数含义如下： 包括HDFS和JDBC两种读取方式。默认为HDFS方式，如果没有使用WHERE条件做数据过滤及在字段映射页面添加新字段的需求，选择HDFS方式即可。 HDFS文件方式读取数据时，性能较好，但不支持使用WHERE条件做数据过滤及在字段映射页面添加新字段。 JDBC方式读取数据时，支持使用WHERE条件做数据过滤及在字段映射页面添加新字段。

更多参数的详细配置可以参考：[CDM配置Hive源端参数](#)。

- iii. 目的端作业配置，具体参考如下：

表 5-7 目的端作业配置

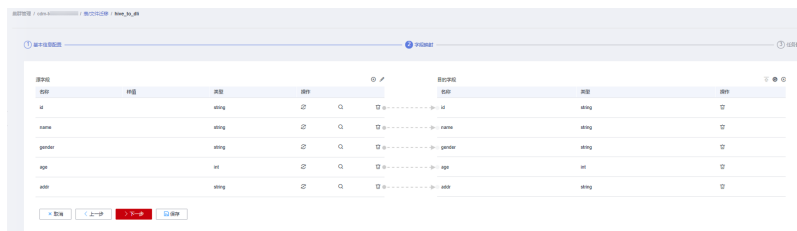
参数名	参数值
目的连接名称	选择1.b已创建的DLI数据源连接。
资源队列	选择已创建的DLI SQL类型的队列。
数据库名称	选择DLI下已创建的数据库。当前示例为在DLI上创建数据库和表中创建的数据库名，即为“testdb”。
表名	选择DLI下已创建的表名。当前示例为在DLI上创建数据库和表中创建的表名，即为“user_info”。
导入前清空数据	选择导入前是否清空目的表的数据。当前示例选择为“否”。 如果设置为是，任务启动前会清除目标表中数据。

更多参数的详细配置可以参考：[CDM配置DLI目的端参数](#)。

3. 单击“下一步”，进入到字段映射界面，CDM会自动匹配源和目的字段。
 - 如果字段映射顺序不匹配，可通过拖拽字段调整。
 - 如果选择在目的端自动创建类型，这里还需要配置每个类型的字段类型、字段名称。

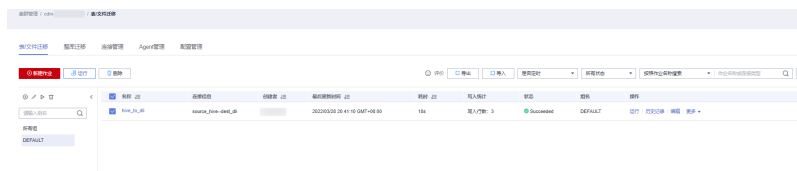
- CDM支持迁移过程中转换字段内容

图 5-18 字段映射



4. 单击“下一步”配置任务参数，一般情况下全部保持默认即可。该步骤用户可以配置如下可选功能：
 - 作业失败重试：如果作业执行失败，可选择是否自动重试，这里保持默认值“不重试”。
 - 作业分组：选择作业所属的分组，默认分组为“DEFAULT”。在CDM“作业管理”界面，支持作业分组显示、按组批量启动作业、按分组导出作业等操作。
 - 是否定时执行：如果需要配置作业定时自动执行，请参见[配置定时任务](#)。这里保持默认值“否”。
 - 抽取并发数：设置同时执行的抽取任务数。这里保持默认值“1”。
 - 是否写入脏数据：如果需要将作业执行过程中处理失败的数据、或者被清洗过滤掉的数据写入OBS中，以便后面查看，可通过该参数配置，写入脏数据前需要先配置好OBS连接。这里保持默认值“否”即可，不记录脏数据。
5. 单击“保存并运行”，回到作业管理界面，在作业管理界面可查看作业执行进度和结果。

图 5-19 迁移作业进度和结果查询

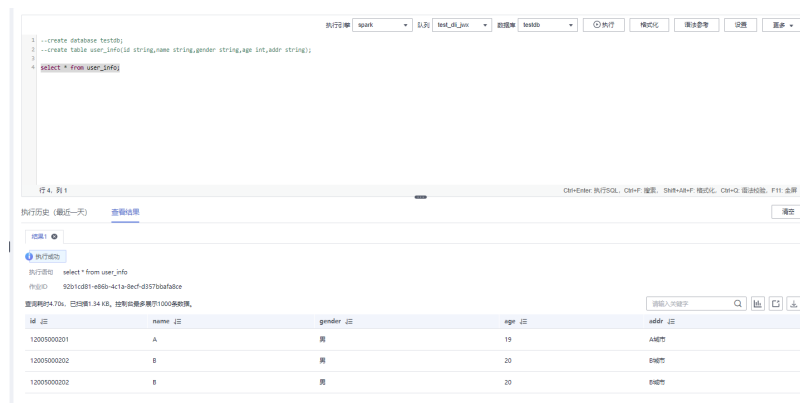


步骤三：结果查询

CDM迁移作业运行完成后，再登录到DLI管理控制台，选择“SQL编辑器”，在SQL编辑器中“执行引擎”选择“spark”，“队列”选择已创建的SQL队列，数据库选择已a已创建的数据库，执行DLI表查询语句，查询Hive表数据是否已成功迁移到DLI的“user_info”表中。

```
select * from user_info;
```


图 5-20 迁移后查询 DLI 的表数据



5.2.4 典型场景示例：迁移 Kafka 数据至 DLI

本文为您介绍如何通过CDM数据同步功能，迁移MRS Kafka数据至DLI。

前提条件

- 已创建DLI的SQL队列。创建DLI队列的操作可以参考[创建DLI队列](#)。

注意

创建DLI队列时**队列类型**需要选择为“SQL队列”。

- 已创建包含Kafka组件的MRS安全集群。具体创建MRS集群的操作可以参考[创建MRS集群](#)。
 - 本示例创建的MRS集群版本为：MRS 3.1.0。
 - 本示例创建的MRS集群开启了Kerberos认证。
- 已创建CDM迁移集群。创建CDM集群的操作可以参考[创建CDM集群](#)。

说明

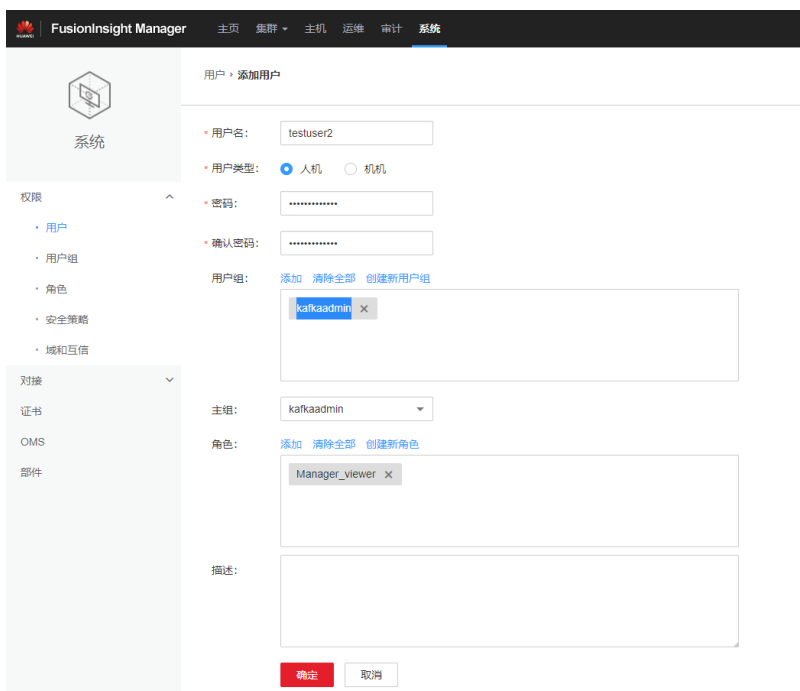
- 如果目标数据源为云下的数据库，则需要通过公网或者专线打通网络。通过公网互通时，需确保CDM集群已绑定EIP、CDM云上安全组出方向放通云下数据源所在的主机、数据源所在的主机可以访问公网且防火墙规则已开放连接端口。
- 数据源为云上的MRS、DWS时，网络互通需满足如下条件：
 - CDM集群与云上服务处于不同区域的情况下，需要通过公网或者专线打通网络。通过公网互通时，需确保CDM集群已绑定EIP，数据源所在的主机可以访问公网且防火墙规则已开放连接端口。
 - CDM集群与云上服务同区域情况下，同虚拟私有云、同子网、同安全组的不同实例默认网络互通；如果同虚拟私有云但是子网或安全组不同，还需配置路由规则及安全组规则。
配置路由规则请参见[如何配置路由规则](#)章节，配置安全组规则请参见[如何配置安全组规则](#)章节。
 - 此外，您还必须确保该云服务的实例与CDM集群所属的企业项目必须相同，如果不同，需要修改工作空间的企业项目。

本示例CDM集群的虚拟私有云、子网以及安全组和创建的MRS集群保持一致。

步骤一：数据准备

- MRS集群上创建Kafka的Topic并且向Topic发送消息。
 - a. 参考[访问MRS Manager](#)登录MRS Manager。
 - b. 在MRS Manager上，选择“系统 > 权限 > 用户”，单击“添加用户”，在添加用户页面分别配置如下参数。
 - i. 用户名：自定义的用户名。当前示例输入为：testuser2。
 - ii. 用户类型：当前选择为“人机”。
 - iii. 密码和确认密码：输入当前用户名对应的密码。
 - iv. 用户组和主组：选择kafkaadmin。
 - v. 角色：选择Manager_viewer角色。

图 5-21 MRS Manager 上创建 Kafka 用户



- c. 在MRS Manager上，选择“集群 > 待操作的集群名称 > 服务 > ZooKeeper > 实例”，获取ZooKeeper角色实例的IP地址，为后续步骤做准备。
- d. 在MRS Manager上，选择“集群 > 待操作的集群名称 > 服务 > kafka > 实例”，获取kafka角色实例的IP地址，为后续步骤做准备。
- e. 参考[安装MRS客户端](#)下载并安装Kafka客户端。例如，当前Kafka客户端安装在MRS主机节点的“/opt/kafkaclient”目录上。
- f. 以root用户进入客户端安装目录下。
例如：`cd /opt/kafkaclient`
- g. 执行以下命令配置环境变量。
`source bigdata_env`
- h. 因为当前集群启用了Kerberos认证，则需要执行以下命令进行安全认证。认证用户为**b**中创建的用户。
`kinit b中创建的用户名`
例如，`kinit testuser2`

- i. 执行以下命令创建名字为kafkatopic的Kafka Topic。

```
kafka-topics.sh --create --zookeeper ZooKeeper角色实例所在节点IP地址1:2181,ZooKeeper角色实例所在节点IP地址2:2181,ZooKeeper角色实例所在节点IP地址3:2181/kafka --replication-factor 1 --partitions 1 --topic kafkatopic
```

上述命令中的“ZooKeeper角色实例所在节点IP地址”即为c中获取的ZooKeeper实例IP。
 - j. 执行以下命令向kafkatopic发送消息。

```
kafka-console-producer.sh --broker-list Kafka角色实例所在节点的IP地址1:21007,Kafka角色实例所在节点的IP地址2:21007,Kafka角色实例所在节点的IP地址3:21007 --topic kafkatopic --producer.config /opt/kafkaclient/Kafka/kafka/config/producer.properties
```

上述命令中的“Kafka角色实例所在节点的IP地址”即为d中获取的Kafka实例IP。

发送测试消息内容如下：

```
{"PageViews":5, "UserID":"4324182021466249494", "Duration":146,"Sign":-1}
```
- 在DLI上创建数据库和表。
 - a. 登录DLI管理控制台，选择“SQL编辑器”，在SQL编辑器中“执行引擎”选择“spark”，“队列”选择已创建的SQL队列。

在编辑器中输入以下语句创建数据库，例如当前创建迁移后的DLI数据库testdb。详细的DLI创建数据库的语法可以参考[创建DLI数据库](#)。

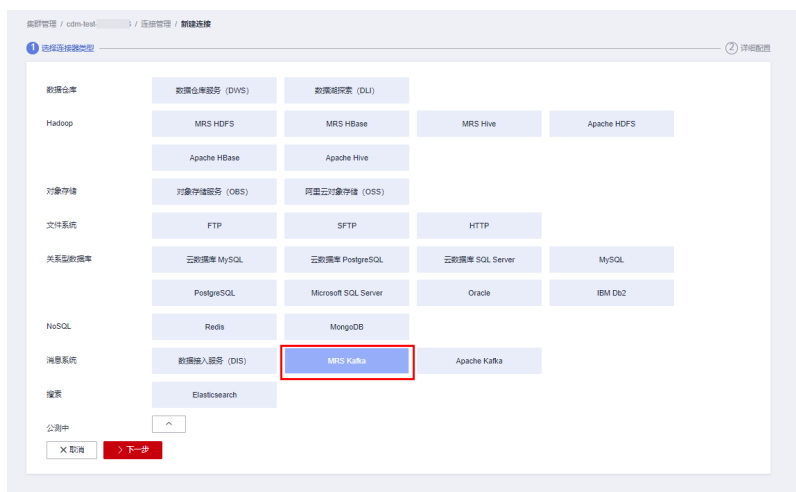
```
create database testdb;
```
 - b. 创建数据库下的表。详细的DLI建表语法可以参考[创建DLI表](#)。

```
CREATE TABLE testditable(value STRING);
```

步骤二：数据迁移

1. 配置CDM数据源连接。
 - a. 配置源端MRS Kafka的数据源连接。
 - i. 登录CDM控制台，选择“集群管理”，选择已创建的CDM集群，在操作列选择“作业管理”。
 - ii. 在作业管理界面，选择“连接管理”，单击“新建连接”，连接器类型选择“MRS Kafka”，单击“下一步”。

图 5-22 创建 MRS Kafka 数据源



- iii. 配置源端MRS Kafka的数据源连接，具体参数配置如下。

表 5-8 MRS Kafka 数据源配置

参数	值
名称	自定义MRS Kafka数据源名称。例如当前配置为“source_kafka”。
Manager IP	单击输入框旁边的“选择”按钮，选择当前MRS Kafka集群即可自动关联出来Manager IP。
用户名	在 b 中创建的MRS Kafka用户名。
密码	对应MRS Kafka用户名的密码。
认证类型	如果当前MRS集群为普通集群则选择为SIMPLE，如果是MRS集群启用了Kerberos安全认证则选择为KERBEROS。 本示例选择为：KERBEROS。

更多参数的详细说明可以参考[CDM上配置Kafka连接](#)。

图 5-23 CDM 配置 MRS Kafka 数据源连接

The screenshot shows a web-based configuration form for a new connection. The breadcrumb path is '集群管理 / cdm-test / 连接管理 / 新建连接'. The form includes the following fields and controls:

- 名称**: Text input field containing 'source_kafka'.
- 连接器**: Dropdown menu set to 'Kafka'.
- Kafka类型**: Dropdown menu set to 'MRS'.
- Manager IP**: Text input field with a '选择' button to the right.
- 用户名**: Text input field containing 'testuser2'.
- 密码**: Password input field with a masked view icon.
- 认证类型**: Dropdown menu set to 'KERBEROS'.
- A link labeled '显示高级属性' is located below the authentication type dropdown.
- At the bottom, there are four buttons: '取消', '< 上一步', '测试', and '保存'.

- iv. 单击“保存”完成MRS Kafka数据源配置。
- b. 配置目的端DLI的数据源连接。
 - i. 登录CDM控制台，选择“集群管理”，选择已创建的CDM集群，在操作列选择“作业管理”。
 - ii. 在作业管理界面，选择“连接管理”，单击“新建连接”，连接器类型选择“数据湖探索（DLI）”，单击“下一步”。

图 5-24 创建 DLI 数据源连接



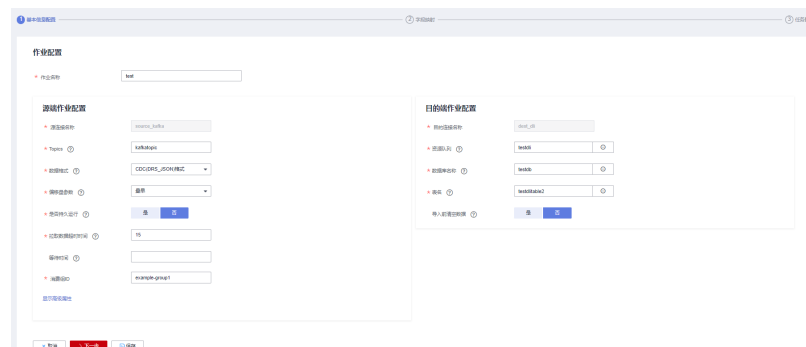
- iii. 配置目的端DLI数据源连接连接参数。具体参数配置可以参考在CDM上配置DLI连接。

图 5-25 配置 DLI 数据源连接参数



- iv. 配置完成后，单击“保存”完成DLI数据源配置。
2. 创建CDM迁移作业。
 - a. 登录CDM控制台，选择“集群管理”，选择已创建的CDM集群，在操作列选择“作业管理”。
 - b. 在“作业管理”界面，选择“表/文件迁移”，单击“新建作业”。
 - c. 在新建作业界面，配置当前作业配置信息，具体参数参考如下：

图 5-26 新建 CDM 作业配置



- i. 作业名称：自定义数据迁移的作业名称。例如，当前定义为：test。

- ii. 源端作业配置，具体参考如下：

表 5-9 源端作业配置

参数名	参数值
源连接名称	选择 1.a 中已创建的数据源名称。
Topics	选择MRS Kafka待迁移的Topic名称，支持单个或多个Topic。当前示例为：kafkatopic。
数据格式	根据实际情况选择当前消息格式。本示例选择为：CDC（DRS_JSON），以DRS_JSON格式解析源数据。
偏移量参数	从Kafka拉取数据时的初始偏移量。本示例当前选择为：最新。 <ul style="list-style-type: none"> 最新：最大偏移量，即拉取最新的数据。 最早：最小偏移量，即拉取最早的数据。 已提交：拉取已提交的数据。 时间范围：拉取时间范围内的数据。
是否持久运行	用户自定义是否永久运行。当前示例选择为：否。
拉取数据超时时间	持续拉取数据多长时间超时，单位分钟。当前示例配置为：15。
等待时间	可选参数，超出等待时间还是无法读取到数据，则不再读取数据，单位秒。当前示例不配置该参数。
消费组ID	用户指定消费组ID。当前使用MRS Kafka默认的消息组ID：“example-group1”。

其他参数的详细配置说明可以参考：[CDM配置Kafka源端参数](#)。

- iii. 目的端作业配置，具体参考如下：

表 5-10 目的端作业配置

参数名	参数值
目的连接名称	选择 1.b 已创建的DLI数据源连接。
资源队列	选择已创建的DLI SQL类型的队列。
数据库名称	选择DLI下已创建的数据库。当前示例为 在DLI上创建数据库和表 中创建的数据库名，即为“testdb”。
表名	选择DLI下已创建的表名。当前示例为 在DLI上创建数据库和表 中创建的表名，即为“testdlitable”。

参数名	参数值
导入前清空数据	选择导入前是否清空目的表的数据。当前示例选择为“否”。 如果设置为是，任务启动前会清除目标表中数据。

详细的参数配置可以参考：[CDM配置DLI目的端参数](#)。

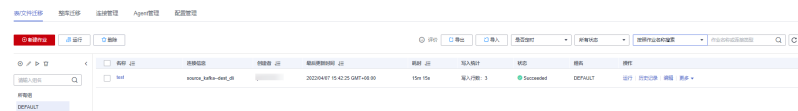
- 单击“下一步”，进入到字段映射界面，CDM会自动匹配源和目的字段。
 - 如果字段映射顺序不匹配，可通过拖拽字段调整。
 - 如果选择在目的端自动创建类型，这里还需要配置每个类型的字段类型、字段名称。
 - CDM支持迁移过程中转换字段内容，详细请参见[字段转换](#)。

图 5-27 字段映射



- 单击“下一步”配置任务参数，一般情况下全部保持默认即可。该步骤用户可以配置如下可选功能：
 - 作业失败重试：如果作业执行失败，可选择是否自动重试，这里保持默认值“不重试”。
 - 作业分组：选择作业所属的分组，默认分组为“DEFAULT”。在CDM“作业管理”界面，支持作业分组显示、按组批量启动作业、按分组导出作业等操作。
 - 是否定时执行：如果需要配置作业定时自动执行，请参见[配置定时任务](#)。这里保持默认值“否”。
 - 抽取并发数：设置同时执行的抽取任务数。这里保持默认值“1”。
 - 是否写入脏数据：如果需要将作业执行过程中处理失败的数据、或者被清洗过滤掉的数据写入OBS中，以便后面查看，可通过该参数配置，写入脏数据前需要先配置好OBS连接。这里保持默认值“否”即可，不记录脏数据。
- 单击“保存并运行”，回到作业管理界面，在作业管理界面可查看作业执行进度和结果。

图 5-28 迁移作业进度和结果查询



步骤三：结果查询

CDM迁移作业运行完成后，再登录到DLI管理控制台，选择“SQL编辑器”，在SQL编辑器中“执行引擎”选择“spark”，“队列”选择已创建的SQL队列，数据库选择已[a](#)已创建的数据库，执行DLI表查询语句，查询Kafka数据是否已成功迁移到DLI的“testdlitable”表中。

```
select * from testdlitable;
```

5.2.5 典型场景示例：迁移 Elasticsearch 数据至 DLI

本文为您介绍如何通过CDM数据同步功能，迁移Elasticsearch类型的CSS集群数据至DLI。其他自建的Elasticsearch等服务数据，均可以通过CDM与DLI进行双向同步。

前提条件

- 已创建DLI的SQL队列。创建DLI队列的操作可以参考[创建DLI队列](#)。

注意

创建DLI队列时**队列类型**需要选择为“SQL队列”。

- 已创建Elasticsearch类型的CSS集群。具体创建CSS集群的操作可以参考[创建CSS集群](#)。
本示例创建的CSS集群版本为：7.6.2，集群为非安全集群。
- 已创建CDM迁移集群。创建CDM集群的操作可以参考[创建CDM集群](#)。

说明

- 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。
- 数据源为云上的CSS服务时，网络互通需满足如下条件：
 - CDM集群与云上服务处于不同区域的情况下，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，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。
 - CDM集群与云上服务同区域情况下，同虚拟私有云、同子网、同安全组的不同实例默认网络互通；如果同虚拟私有云但是子网或安全组不同，还需配置路由规则及安全组规则。
配置路由规则请参见[如何配置路由规则](#)章节，配置安全组规则请参见[如何配置安全组规则](#)章节。
 - 此外，您还必须确保该云服务的实例与CDM集群所属的企业项目必须相同，如果不同，需要修改工作空间的企业项目。

本示例CDM集群的虚拟私有云、子网以及安全组和创建的CSS集群保持一致。

步骤一：数据准备

- 在CSS集群上创建索引并导入数据。
 - 登录CSS管理控制台，选择“集群管理 > Elasticsearch”。
 - 在集群管理界面，在已创建的CSS集群的“操作”列，单击“Kibana”访问集群。
 - 在Kibana的左侧导航中选择“Dev Tools”，进入到Console界面。
 - 在Console界面，执行如下命令创建索引“my_test”。

```
PUT /my_test
{
  "settings": {
    "number_of_shards": 1
  },
  "mappings": {
    "properties": {
```



```
"productName": {
  "type": "text",
  "analyzer": "ik_smart"
},
"size": {
  "type": "keyword"
}
}
}
```

- e. 在Console界面，执行如下命令，将数据导入到“my_test”索引中。

```
POST /my_test/_doc/_bulk
{"index":{}}
{"productName":"2017秋装新款文艺衬衫女装","size":"L"}
{"index":{}}
{"productName":"2017秋装新款文艺衬衫女装","size":"M"}
{"index":{}}
{"productName":"2017秋装新款文艺衬衫女装","size":"S"}
{"index":{}}
{"productName":"2018春装新款牛仔裤女装","size":"M"}
{"index":{}}
{"productName":"2018春装新款牛仔裤女装","size":"S"}
{"index":{}}
{"productName":"2017春装新款休闲裤女装","size":"L"}
{"index":{}}
{"productName":"2017春装新款休闲裤女装","size":"S"}
```

当返回结果信息中“errors”字段的值为“false”时，表示导入数据成功。

- 在DLI上创建数据库和表。
 - a. 登录DLI管理控制台，选择“SQL编辑器”，在SQL编辑器中“执行引擎”选择“spark”，“队列”选择已创建的SQL队列。
在编辑器中输入以下语句创建数据库，例如当前创建迁移后的DLI数据库testdb。详细的DLI创建数据库的语法可以参考[创建DLI数据库](#)。

```
create database testdb;
```
 - b. 创建数据库下的表。详细的DLI建表语法可以参考[创建DLI表](#)。

```
create table tablecss(size string, productname string);
```

步骤二：数据迁移

1. 配置CDM数据源连接。
 - a. 配置源端CSS的数据源连接。
 - i. 登录CDM控制台，选择“集群管理”，选择已创建的CDM集群，在操作列选择“作业管理”。
 - ii. 在作业管理界面，选择“连接管理”，单击“新建连接”，连接器类型选择“云搜索服务”，单击“下一步”。

图 5-29 创建 CSS 数据源



- iii. 配置源端CSS的数据源连接，具体参数配置如下。详细参数配置可以参考 [CDM上配置CSS连接](#)。

表 5-11 CSS 数据源配置

参数	值
名称	自定义CSS数据源名称。例如当前配置为“source_css”。
Elasticsearch 服务器列表	单击输入框旁边的“选择”按钮，选择当前CSS集群即可自动关联出来Elasticsearch服务器列表。
安全模式认证	如果所需连接的CSS集群在创建时开启了“安全模式”，该参数需设置为“是”，否则设置为“否”。本示例选择为“否”。

图 5-30 CDM 配置 CSS 数据源



- iv. 单击“保存”完成CSS数据源配置。
- b. 配置目的端DLI的数据源连接。
 - i. 登录CDM控制台，选择“集群管理”，选择已创建的CDM集群，在操作列选择“作业管理”。
 - ii. 在作业管理界面，选择“连接管理”，单击“新建连接”，连接器类型选择“数据湖探索（DLI）”，单击“下一步”。

图 5-31 创建 DLI 数据源连接



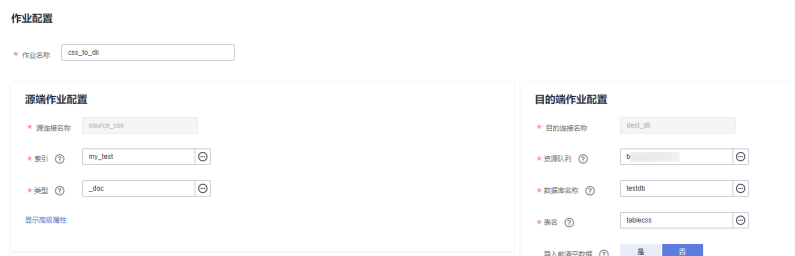
- iii. 配置目的端DLI数据源连接连接参数。具体参数配置可以参考[在CDM上配置DLI连接](#)。

图 5-32 配置 DLI 数据源连接参数



- iv. 配置完成后，单击“保存”完成DLI数据源配置。
- 2. 创建CDM迁移作业。
 - a. 登录CDM控制台，选择“集群管理”，选择已创建的CDM集群，在操作列选择“作业管理”。
 - b. 在“作业管理”界面，选择“表/文件迁移”，单击“新建作业”。
 - c. 在新建作业界面，配置当前作业配置信息，具体参数参考如下：

图 5-33 新建 CDM 作业配置



- i. 作业名称：自定义数据迁移的作业名称。例如，当前定义为：css_to_dli。
- ii. 源端作业配置，具体参考如下：

表 5-12 源端作业配置

参数名	参数值
源连接名称	选择1.a中已创建的数据源名称。
索引	选择CSS集群中创建的Elasticsearch索引名。当前示例为CSS集群上创建索引并导入数据中创建的索引“my_test”。 索引名称只能全部小写，不能有大写。
类型	Elasticsearch的类型，类似关系数据库中的表名称。类型名称只能全部小写，不能有大写。当前示例为：“_doc”。

更多其他参数说明可以参考：[CDM配置CSS源端参数](#)。

- iii. 目的端作业配置，具体参考如下：

表 5-13 目的端作业配置

参数名	参数值
目的连接名称	选择 1.b 已创建的DLI数据源连接。
资源队列	选择已创建的DLI SQL类型的队列。
数据库名称	选择DLI下已创建的数据库。当前示例为 在DLI上创建数据库和表 中创建的数据库名，即为“testdb”。
表名	选择DLI下已创建的表名。当前示例为 在DLI上创建数据库和表 中创建的表名，即为“tablecss”。
导入前清空数据	选择导入前是否清空目的表的数据。当前示例选择为“否”。 如果设置为是，任务启动前会清除目标表中数据。

详细的参数配置可以参考：[CDM配置DLI目的端参数](#)。

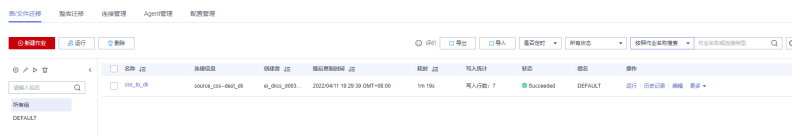
- 单击“下一步”，进入到字段映射界面，CDM会自动匹配源和目的字段。
 - 如果字段映射顺序不匹配，可通过拖拽字段调整。
 - 如果选择在目的端自动创建类型，这里还需要配置每个类型的字段类型、字段名称。
 - CDM支持迁移过程中转换字段内容，详细请参见[字段转换](#)。

图 5-34 字段映射



- 单击“下一步”配置任务参数，一般情况下全部保持默认即可。
该步骤用户可以配置如下可选功能：
 - 作业失败重试：如果作业执行失败，可选择是否自动重试，这里保持默认值“不重试”。
 - 作业分组：选择作业所属的分组，默认分组为“DEFAULT”。在CDM“作业管理”界面，支持作业分组显示、按组批量启动作业、按分组导出作业等操作。
 - 是否定时执行：如果需要配置作业定时自动执行，请参见[配置定时任务](#)。这里保持默认值“否”。
 - 抽取并发数：设置同时执行的抽取任务数。这里保持默认值“1”。
 - 是否写入脏数据：如果需要将作业执行过程中处理失败的数据、或者被清洗过滤掉的数据写入OBS中，以便后面查看，可通过该参数配置，写入脏数据前需要先配置好OBS连接。这里保持默认值“否”即可，不记录脏数据。
- 单击“保存并运行”，回到作业管理界面，在作业管理界面可查看作业执行进度和结果。

图 5-35 迁移作业进度和结果查询

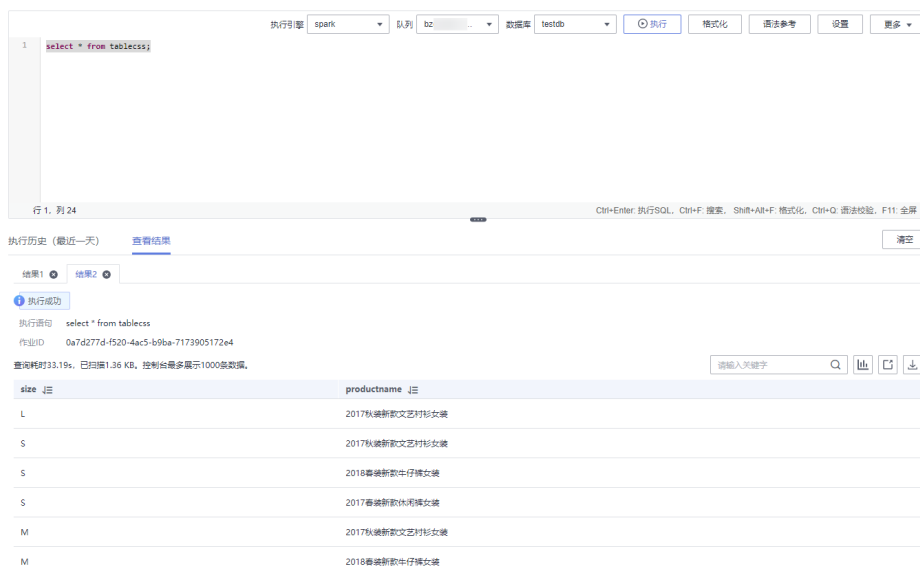


步骤三：结果查询

CDM迁移作业运行完成后，再登录到DLI管理控制台，选择“SQL编辑器”，在SQL编辑器中“执行引擎”选择“spark”，“队列”选择已创建的SQL队列，数据库选择已a中已创建的数据库，执行DLI表查询语句，查询CSS的数据是否已成功迁移到DLI的“tablecss”表中。

```
select * from tablecss;
```

图 5-36 迁移后查询 DLI 的表数据



5.2.6 典型场景示例：迁移 RDS 数据至 DLI

本文为您介绍如何通过CDM数据同步功能，迁移关系型数据库RDS数据至DLI。其他关系型数据库数据都可以通过CDM与DLI进行双向同步。

前提条件

- 已创建DLI的SQL队列。创建DLI队列的操作可以参考[创建DLI队列](#)。



创建DLI队列时队列类型需要选择为“SQL队列”。

- 已创建云数据库RDS的MySQL的数据库实例。具体创建RDS集群的操作可以参考[创建RDS MySQL数据库实例](#)。
 - 本示例RDS数据库引擎：MySQL
 - 本示例RDS MySQL数据库版本：5.7。

- 已创建CDM迁移集群。创建CDM集群的操作可以参考[创建CDM集群](#)。

📖 说明

- 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。
- 数据源为云上服务RDS、MRS时，网络互通需满足如下条件：
 - i. CDM集群与云上服务处于不同区域的情况下，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，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。
 - ii. CDM集群与云上服务同区域情况下，同虚拟私有云、同子网、同安全组的不同实例默认网络互通；如果同虚拟私有云但是子网或安全组不同，还需配置路由规则及安全组规则。
配置路由规则请参见[如何配置路由规则](#)章节，配置安全组规则请参见[如何配置安全组规则](#)章节。
 - iii. 此外，您还必须确保该云服务的实例与CDM集群所属的企业项目必须相同，如果不同，需要修改工作空间的企业项目。

本示例CDM集群的虚拟私有云、子网以及安全组和RDS MySQL实例保持一致。

步骤一：数据准备

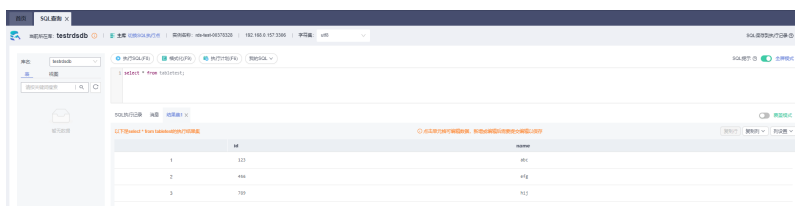
- RDS的MySQL的数据库实例上创建数据库和表。
 - a. 登录RDS管理控制台，在“实例管理”界面，选择已创建的MySQL实例，选择操作列的“更多 > 登录”，进入数据管理服务实例登录界面。
 - b. 输入实例登录的用户名和密码。单击“登录”，即可进入MySQL数据库并进行管理。
 - c. 在数据库实例界面，单击“新建数据库”，数据库名定义为：testrdsdb，字符集保持默认即可。
 - d. 在已创建的数据库的操作列，单击“SQL查询”，输入以下创建表语句，创建RDS MySQL表。

```
CREATE TABLE tabletest (
  `id` VARCHAR(32) NOT NULL,
  `name` VARCHAR(32) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4;
```
 - e. 插入表数据。

```
insert into tabletest VALUES ('123','abc');
insert into tabletest VALUES ('456','efg');
insert into tabletest VALUES ('789','hij');
```
 - f. 查询测试的表数据。

```
select * from tabletest;
```

图 5-37 查询 RDS 表数据



- 在DLI上创建数据库和表。

- a. 登录DLI管理控制台，选择“SQL编辑器”，在SQL编辑器中“执行引擎”选择“spark”，“队列”选择已创建的SQL队列。

在编辑器中输入以下语句创建数据库，例如当前创建迁移后的DLI数据库testdb。详细的DLI创建数据库的语法可以参考[创建DLI数据库](#)。

```
create database testdb;
```

- b. 在“SQL编辑器”中，数据库选择“testdb”，执行以下建表语句创建数据库下的表。详细的DLI建表语法可以参考[创建DLI表](#)。

```
create table tabletest(id string,name string);
```

步骤二：数据迁移

1. 配置CDM数据源连接。

- a. 创建源端RDS数据库的连接。
 - i. 登录CDM控制台，选择“集群管理”，选择已创建的CDM集群，在操作列选择“作业管理”。
 - ii. 首次创建RDS MySQL数据库连接时需要上传MySQL的驱动，单击“连接管理 > 驱动管理”，进入驱动管理界面。
 - iii. 参考[CDM管理驱动](#)下载MySQL的驱动包到本地，将下载后驱动包本地解压，获取驱动的jar包文件。
例如，当前下载MySQL驱动包压缩文件为“mysql-connector-java-5.1.48.zip”，解压后获取驱动文件“mysql-connector-java-5.1.48.jar”。
 - iv. 返回到驱动管理界面，在驱动名称为MYSQL的操作列，单击“上传”，在“导入驱动文件”界面单击“添加文件”，将[1.a.iii](#)获取的驱动文件上传。
 - v. 在驱动管理界面单击“返回”按钮回到连接管理界面，单击“新建连接”，连接器类型选择“云数据库 MySQL”，单击“下一步”。
 - vi. 配置连接RDS的数据源连接参数，具体参数配置如下。

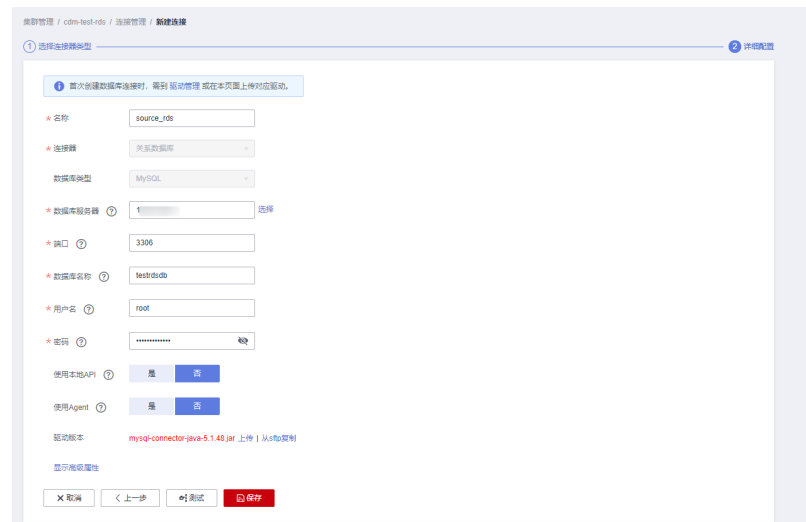
表 5-14 RDS MySQL 数据源配置

参数	值
名称	自定义RDS数据源名称。例如当前配置为：source_rds。
数据库服务	单击输入框旁边的“选择”按钮，选择当前已创建的RDS实例名即可自动关联出来数据库服务器地址。
端口	RDS实例的端口。选择数据库服务器后自动自动关联。
数据库名称	当前需要迁移的RDS MySQL数据库名称。当前示例为c中创建的数据库“testrdsdb”。
用户名	待连接数据库的用户。该数据库用户需要有数据表的读写权限，以及对元数据的读取权限。 本示例使用创建RDS MySQL数据库实例的默认用户“root”。

参数	值
密码	对应的RDS MySQL数据库用户的密码。

其他更多参数保持默认即可，如果需要了解详细参数说明，可以参考[配置关系数据库连接](#)。单击“保存”完成RDS MySQL数据源连接配置。

图 5-38 CDM 配置 RDS MySQL 数据源



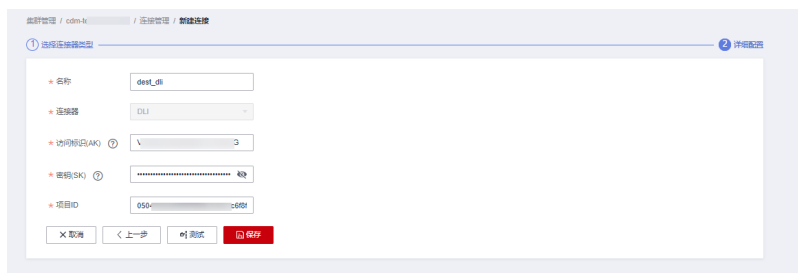
- b. 创建目的端DLI数据源的连接。
 - i. 登录CDM控制台，选择“集群管理”，选择已创建的CDM集群，在操作列选择“作业管理”。
 - ii. 在作业管理界面，选择“连接管理”，单击“新建连接”，连接器类型选择“数据湖探索（DLI）”，单击“下一步”。

图 5-39 创建 DLI 数据源连接



- i. 配置目的端DLI数据源连接。具体参数配置可以参考在[CDM上配置DLI连接](#)。

图 5-40 创建 DLI 数据源连接

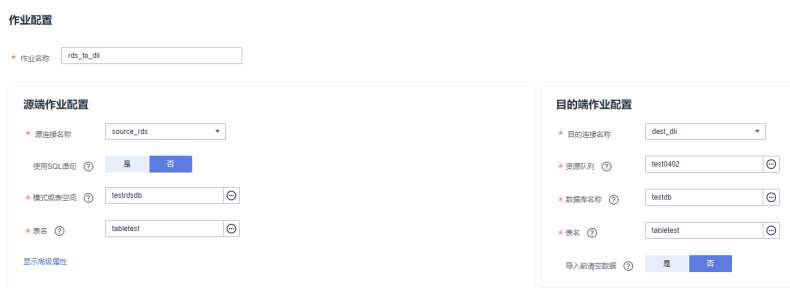


配置完成后，单击“保存”完成DLI数据源配置。

2. 创建CDM迁移作业。

- a. 登录CDM控制台，选择“集群管理”，选择已创建的CDM集群，在操作列选择“作业管理”。
- b. 在“作业管理”界面，选择“表/文件迁移”，单击“新建作业”。
- c. 在新建作业界面，配置当前作业配置信息，具体参数参考如下：

图 5-41 CDM 数据迁移作业配置



- i. 作业名称：自定义数据迁移的作业名称。例如，当前定义为：rds_to_dli。
- ii. 源端作业配置，具体参考如下：

表 5-15 源端作业配置

参数名	参数值
源连接名称	选择1.a中已创建的数据源名称。
使用SQL语句	“使用SQL语句”选择“是”时，您可以在这里输入自定义的SQL语句，CDM将根据该语句导出数据。本示例当前选择为“否”。
模式或表空间	选择RDS MySQL待迁移的数据库名称。例如当前待迁移的表数据数据库为“testrdsdb”。
表名	待迁移的RDS MySQL数据表名。当前为d中的“tabletest”表。

更多详细参数配置请参考[配置关系数据库源端参数](#)。

iii. 目的端参数配置，具体参考如下：

表 5-16 目的端作业配置

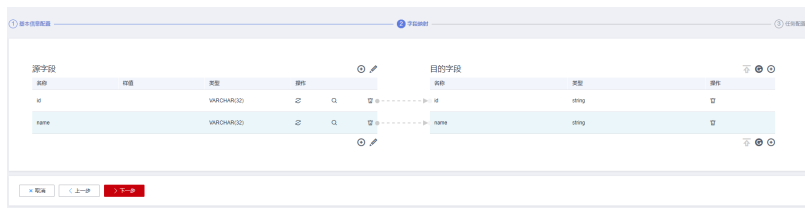
参数名	参数值
目的连接名称	选择已创建的DLI数据源连接。
资源队列	选择已创建的DLI SQL类型的队列。
数据库名称	选择DLI下已创建的数据库。当前示例为在DLI上创建数据库和表创建的数据库名，即为“testdb”。
表名	选择DLI下已创建的表名。当前示例为在DLI上创建数据库和表创建的表名，即为“tabletest”。
导入前清空数据	选择导入前是否清空目的表的数据。当前示例选择为“否”。 如果设置为是，任务启动前会清除目标表中数据。

详细的参数配置可以参考：[CDM配置DLI目的端参数](#)。

iv. 单击“下一步”，进入到字段映射界面，CDM会自动匹配源和目的字段。

- 如果字段映射顺序不匹配，可通过拖拽字段调整。
- 如果选择在目的端自动创建类型，这里还需要配置每个类型的字段类型、字段名称。
- CDM支持迁移过程中转换字段内容，详细请参见[字段转换](#)。

图 5-42 字段映射



v. 单击“下一步”配置任务参数，一般情况下全部保持默认即可。

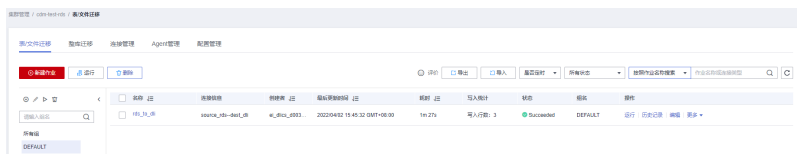
该步骤用户可以配置如下可选功能：

- 作业失败重试：如果作业执行失败，可选择是否自动重试，这里保持默认值“不重试”。
- 作业分组：选择作业所属的分组，默认分组为“DEFAULT”。在CDM“作业管理”界面，支持作业分组显示、按组批量启动作业、按分组导出作业等操作。
- 是否定时执行：如果需要配置作业定时自动执行，请参见[配置定时任务](#)。这里保持默认值“否”。
- 抽取并发数：设置同时执行的抽取任务数。这里保持默认值“1”。
- 是否写入脏数据：如果需要将作业执行过程中处理失败的数据、或者被清洗过滤掉的数据写入OBS中，以便后面查看，可通过该参数

配置，写入脏数据前需要先配置好OBS连接。这里保持默认值“否”即可，不记录脏数据。

- vi. 单击“保存并运行”，回到作业管理界面，在作业管理界面可查看作业执行进度和结果。

图 5-43 迁移作业进度和结果查询

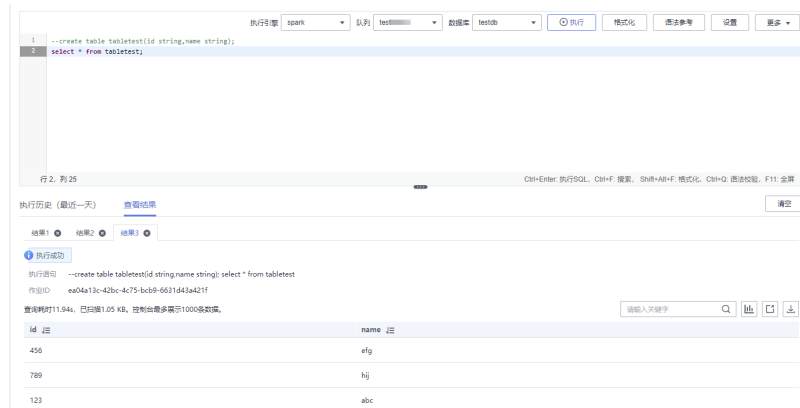


步骤三：结果查询

CDM迁移作业运行完成后，再登录到DLI管理控制台，选择“SQL编辑器”，在SQL编辑器中“执行引擎”选择“spark”，“队列”选择已创建的SQL队列，数据库选择在DLI上创建数据库和表已创建的数据库，执行DLI表查询语句，查询RDS MySQL表数据是否已成功迁移到DLI的“tabletest”表中。

```
select * from tabletest;
```

图 5-44 查询 DLI 表数据



5.2.7 典型场景示例：迁移 DWS 数据至 DLI

本文为您介绍如何通过CDM数据同步功能，迁移数据仓库服务DWS数据至DLI。

前提条件

- 已创建DLI的SQL队列。创建DLI队列的操作可以参考[创建DLI队列](#)。



创建DLI队列时队列类型需要选择为“SQL队列”。

- 已创建数据仓库服务DWS集群。具体创建DWS集群的操作可以参考[创建DWS集群](#)。
- 已创建CDM迁移集群。创建CDM集群的操作可以参考[创建CDM集群](#)。

📖 说明

- 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。
- 数据源为云上的DWS、MRS等服务时，网络互通需满足如下条件：
 - i. CDM集群与云上服务处于不同区域的情况下，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，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。
 - ii. CDM集群与云上服务同区域情况下，同虚拟私有云、同子网、同安全组的不同实例默认网络互通；如果同虚拟私有云但是子网或安全组不同，还需配置路由规则及安全组规则。
配置路由规则请参见[如何配置路由规则](#)章节，配置安全组规则请参见[如何配置安全组规则](#)章节。
 - iii. 此外，您还必须确保该云服务的实例与CDM集群所属的企业项目必须相同，如果不同，需要修改工作空间的企业项目。

本示例CDM集群的虚拟私有云、子网以及安全组和DWS集群保持一致。

步骤一：数据准备

- DWS集群上创建数据库和表。
 - a. 参考[使用gsql命令行客户端连接DWS集群](#)连接已创建的DWS集群。
 - b. 执行以下命令连接DWS集群的默认数据库“gaussdb”：

```
gsql -d gaussdb -h DWS集群连接地址 -U dbadmin -p 8000 -W password -r
```

 - gaussdb：DWS集群默认数据库。
 - DWS集群连接地址：请参见[获取集群连接地址](#)进行获取。如果通过公网地址连接，请指定为集群“公网访问地址”或“公网访问域名”，如果通过内网地址连接，请指定为集群“内网访问地址”或“内网访问域名”。如果通过弹性负载均衡连接，请指定为“弹性负载均衡地址”。
 - dbadmin：创建集群时设置的默认管理员用户名。
 - -W：默认管理员用户的密码。
 - c. 在命令行窗口输入以下命令创建数据库“testdwsdb”。

```
CREATE DATABASE testdwsdb;
```
 - d. 执行以下命令，退出gaussdb数据库，连接新创建的数据库“testdwsdb”。

```
\q  
gsql -d testdwsdb -h DWS集群连接地址 -U dbadmin -p 8000 -W password -r
```
 - e. 执行以下命令创建表并插入数据。
创建表：

```
CREATE TABLE table1(id int, a char(6), b varchar(6),c varchar(6)) ;
```


插入表数据：

```
INSERT INTO table1 VALUES(1,'123','456','789');  
INSERT INTO table1 VALUES(2,'abc','efg','hif');
```
 - f. 查询表数据确认数据插入成功。

```
select * from table1;
```

图 5-45 查询表数据

```
testdwsdb=> select * from table1;
id | a | b | c
---+---+---+---
 1 | 123 | 456 | 789
 2 | abc | efg | hif
(2 rows)
```

- 在DLI上创建数据库和表。
 - a. 登录DLI管理控制台，选择“SQL编辑器”，在SQL编辑器中“执行引擎”选择“spark”，“队列”选择已创建的SQL队列。
在编辑器中输入以下语句创建数据库，例如当前创建迁移后的DLI数据库testdb。详细的DLI创建数据库的语法可以参考[创建DLI数据库](#)。

```
create database testdb;
```
 - b. 在“SQL编辑器”中，数据库选择“testdb”，执行以下建表语句创建数据库下的表。详细的DLI建表语法可以参考[创建DLI表](#)。

```
create table tabletest(id INT, name1 string, name2 string, name3 string);
```

步骤二：数据迁移

1. 配置CDM数据源连接。
 - a. 创建源端DWS数据库的连接。
 - i. 登录CDM控制台，选择“集群管理”，选择已创建的CDM集群，在操作列选择“作业管理”。
 - ii. 在作业管理界面，选择“连接管理”，单击“新建连接”，连接器类型选择“数据仓库服务（DWS）”，单击“下一步”。
 - iii. 配置连接DWS的数据源连接参数，具体参数配置如下。

表 5-17 DWS 数据源配置

参数	值
名称	自定义DWS数据源名称。例如当前配置为：source_dws。
数据库服务器	单击输入框旁边的“选择”按钮，选择当前已创建的DWS集群名称。
端口	DWS数据库的端口，默认为：8000。
数据库名称	当前需要迁移的DWS数据库名称。当前示例为 DWS集群上创建数据库和表 中创建的数据库“testdwsdb”。
用户名	待连接数据库的用户。该数据库用户需要有数据表的读写权限，以及对元数据的读取权限。 本示例使用创建DWS数据库实例的默认管理员用户“dbadmin”。
密码	对应的DWS数据库用户的密码。

图 5-46 CDM 配置 DWS 数据源

其他更多参数保持默认即可，如果需要了解更多参数说明，可以参考[配置关系数据库连接](#)。单击“保存”完成DWS数据源连接配置。

- b. 创建目的端DLI数据源的连接。
 - i. 登录CDM控制台，选择“集群管理”，选择已创建的CDM集群，在操作列选择“作业管理”。
 - ii. 在作业管理界面，选择“连接管理”，单击“新建连接”，连接器类型选择“数据湖探索（DLI）”，单击“下一步”。

图 5-47 创建 DLI 数据源连接

- i. 配置目的端DLI数据源连接。具体参数配置可以参考[在CDM上配置DLI连接](#)。

图 5-48 创建 DLI 数据源连接

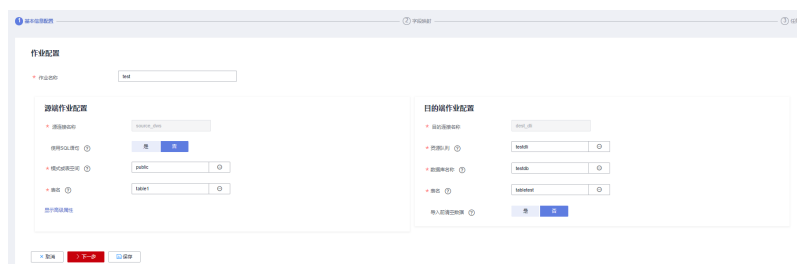


配置完成后，单击“保存”完成DLI数据源配置。

2. 创建CDM迁移作业。

- a. 登录CDM控制台，选择“集群管理”，选择已创建的CDM集群，在操作列选择“作业管理”。
- b. 在“作业管理”界面，选择“表/文件迁移”，单击“新建作业”。
- c. 在新建作业界面，配置当前作业配置信息，具体参数参考如下：

图 5-49 CDM 数据迁移作业配置



- i. 作业名称：自定义数据迁移的作业名称。例如，当前定义为：test。
- ii. 源端作业配置，具体参考如下：

表 5-18 源端作业配置

参数名	参数值
源连接名称	选择1.a中已创建的数据源名称。
使用SQL语句	“使用SQL语句”选择“是”时，您可以在这里输入自定义的SQL语句，CDM将根据该语句导出数据。本示例当前选择为“否”。

参数名	参数值
模式或表空间	<p>“使用SQL语句”选择“否”时，显示该参数，表示待抽取数据的模式或表空间名称。单击输入框后面的按钮可进入模式选择界面，用户也可以直接输入模式或表空间名称。</p> <p>本示例因为DWS集群上创建数据库和表中没有创建SCHEMA，则本参数为默认的“public”。</p> <p>如果选择界面没有待选择的模式或表空间，请确认对应连接里的账号是否有元数据查询的权限。</p> <p>说明 该参数支持配置通配符（*），实现导出以某一前缀开头或者以某一后缀结尾的所有数据库。例如： SCHEMA*表示导出所有以“SCHEMA”开头的数据库。 *SCHEMA表示导出所有以“SCHEMA”结尾的数据库。 *SCHEMA*表示数据库名称中只要有“SCHEMA”字符串，就全部导出。</p>
表名	待迁移的DWS数据表名。当前为 DWS集群上创建数据库和表 中的“table1”表。

更多详细参数配置请参考[配置关系数据库源端参数](#)。

iii. 目的端作业参数配置，具体参考如下：

表 5-19 目的端作业配置

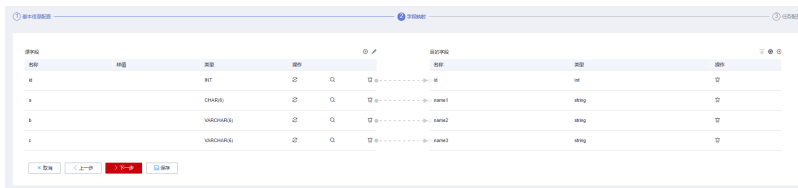
参数名	参数值
目的连接名称	选择已创建的DLI数据源连接。
资源队列	选择已创建的DLI SQL类型的队列。
数据库名称	选择DLI下已创建的数据库。当前示例为 在DLI上创建数据库和表 创建的数据库名，即为“testdb”。
表名	选择DLI下已创建的表名。当前示例为 在DLI上创建数据库和表 创建的表名，即为“tabletest”。
导入前清空数据	<p>选择导入前是否清空目的表的数据。当前示例选择为“否”。</p> <p>如果设置为是，任务启动前会清除目标表中数据。</p>

详细的参数配置可以参考：[CDM配置DLI目的端参数](#)。

- iv. 单击“下一步”，进入到字段映射界面，CDM会自动匹配源和目的字段。
- 如果字段映射顺序不匹配，可通过拖拽字段调整。
 - 如果选择在目的端自动创建类型，这里还需要配置每个类型的字段类型、字段名称。

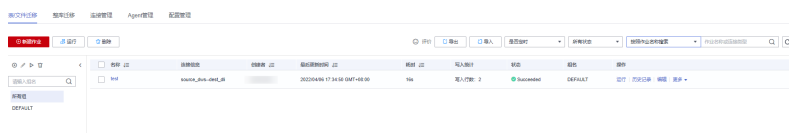
- CDM支持迁移过程中转换字段内容，详细请参见[字段转换](#)。

图 5-50 字段映射



- 单击“下一步”配置任务参数，一般情况下全部保持默认即可。
该步骤用户可以配置如下可选功能：
 - 作业失败重试：如果作业执行失败，可选择是否自动重试，这里保持默认值“不重试”。
 - 作业分组：选择作业所属的分组，默认分组为“DEFAULT”。在CDM“作业管理”界面，支持作业分组显示、按组批量启动作业、按分组导出作业等操作。
 - 是否定时执行：如果需要配置作业定时自动执行，请参见[配置定时任务](#)。这里保持默认值“否”。
 - 抽取并发数：设置同时执行的抽取任务数。这里保持默认值“1”。
 - 是否写入脏数据：如果需要将作业执行过程中处理失败的数据、或者被清洗过滤掉的数据写入OBS中，以便后面查看，可通过该参数配置，写入脏数据前需要先配置好OBS连接。这里保持默认值“否”即可，不记录脏数据。
- 单击“保存并运行”，回到作业管理界面，在作业管理界面可查看作业执行进度和结果。

图 5-51 迁移作业进度和结果查询

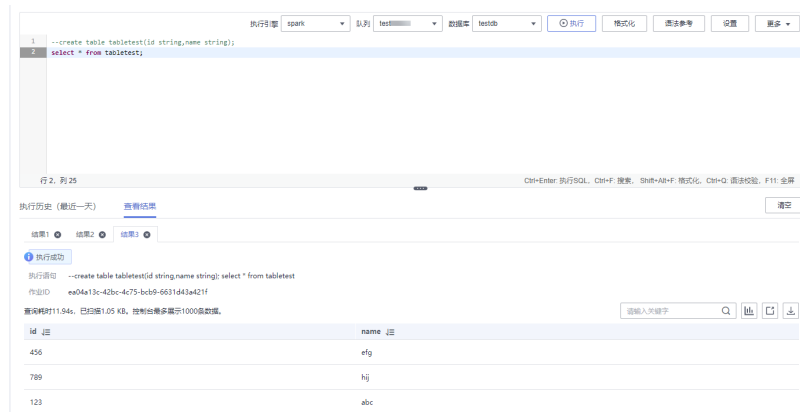


步骤三：结果查询

CDM迁移作业运行完成后，再登录到DLI管理控制台，选择“SQL编辑器”，在SQL编辑器中“执行引擎”选择“spark”，“队列”选择已创建的SQL队列，数据库选择在[DLI上创建数据库和表](#)中已创建的数据库，执行DLI表查询语句，查询DWS表数据是否已成功迁移到DLI的“tabletest”表中。

```
select * from tabletest;
```

图 5-52 查询 DLI 表数据



5.3 配置 DLI 读写外部数据源数据

5.3.1 配置 DLI 读写外部数据源数据的操作流程

DLI执行作业需要读写外部数据源时需要具备两个条件：

- 打通DLI和外部数据源之间的网络，确保DLI队列与数据源的网络连通。
- 妥善保存数据源的访问凭证确保数据源认证的安全性，便于DLI安全访问数据源。

本节操作介绍配置DLI读写外部数据源数据操作流程。

- 配置DLI与数据源网络连通：您可以参考[配置DLI与数据源网络连通（增强型跨源连接）](#)配置DLI与数据源的网络连通。
- 管理DLI具数据源的访问凭证：
 - Spark 3.3.1及以上版本、Flink 1.15及以上版本的跨源访问场景
 - 推荐使用数据加密服务DEW来存储数据源的认证信息，为您解决数据安全、密钥安全、密钥管理复杂等问题。
具体操作请参考[使用DEW管理数据源访问凭证](#)。
 - 使用DEW管理数据源访问凭证时，您还需要创建DLI云服务委托授予DLI访问其他服务（DEW）读取访问凭证。
 - SQL作业、Flink 1.12版本的跨源访问场景，使用DLI提供的“跨源认证”管理数据源的访问凭证，具体操作请参考[使用DLI的跨源认证管理数据源访问凭证](#)。

5.3.2 配置 DLI 与数据源网络连通（增强型跨源连接）

5.3.2.1 增强型跨源连接概述

为什么要创建增强型跨源连接？

DLI跨源分析场景连接外部数据源时，由于数据源的VPC与DLI VPC不同，网络无法连通，导致DLI无法读取数据源数据。DLI提供的增强型跨源连接功能可以实现DLI与数据源的网络连通。

本节操作为您介绍跨VPC的数据源网络连通方案：

- 创建增强型跨源连接：采用对等连接的方式打通DLI与数据源的VPC网络。
- 测试网络连通性：验证队列与数据源网络连通性。

目前DLI支持跨源访问的数据源请参考[DLI常用跨源分析开发方式](#)。

注意

在跨源开发场景中直接配置跨源认证信息存在密码泄露的风险，Spark 3.3.1及以上版本、Flink 1.15及以上版本的跨源访问场景推荐使用数据加密服务DEW来存储数据源的认证信息，为您解决数据安全、密钥安全、密钥管理复杂等问题。具体操作请参考[使用DEW管理数据源访问凭证](#)。

约束和限制

表 5-20 增强型跨源连接约束限制

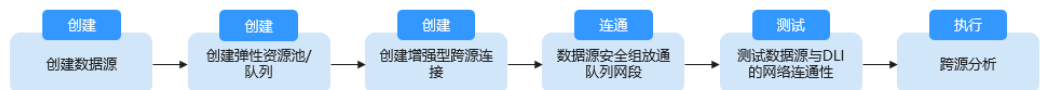
限制项	说明
适用场景约束限制	<ul style="list-style-type: none">● 在同一队列中，如果同时使用了经典型跨源连接和增强型跨源连接，则经典型跨源连接优先于增强型跨源连接。推荐使用增强型跨源连接。● DLI提供的default队列不支持创建跨源连接。● Flink作业访问DIS，OBS和SMN数据源，无需创建跨源连接，可以直接访问。● 增强型跨源仅支持包年包月队列、按需计费模式下的专属队列。
权限要求	<ul style="list-style-type: none">● 增强型跨源连接需要使用VPC、子网、路由、对等连接功能，因此需要获得VPC（虚拟私有云）的VPC Administrator权限。 可在服务授权中进行设置。
使用约束限制	<ul style="list-style-type: none">● 使用DLI增强型跨源时，弹性资源池/队列的网段与数据源网段不能重合。● 访问跨源表需要使用已经创建跨源连接的队列。● 跨源表不支持Preview预览功能。

限制项	说明
检测连通性要求	<ul style="list-style-type: none"> 检测跨源连接的连通性时对IP约束限制如下： <ul style="list-style-type: none"> IP必须为合法的IP地址，用“.”分隔的4个十进制数，范围是0-255。 测试时IP地址后可选择添加端口，用“:”隔开，端口最大限制5位，端口范围：0~65535。 例如192.168.xx.xx或者192.168.xx.xx:8181。 检测跨源连接的连通性时对域名约束限制如下： <ul style="list-style-type: none"> 域名的限制长度为1到255的字符串，并且组成必须是字母、数字、下划线或者短横线。 域名的顶级域名至少包含两个及以上的字母，例如.com，.net，.cn等。 测试时域名后可选择添加端口，用“:”隔开，端口最大限制为5位，端口范围：0~65535。 例如example.com:8080。

DLI 跨源分析流程

使用DLI进行跨源分析，需要先建立跨源连接，打通DLI和数据源的网络，再开发不同的作业访问数据源。

图 5-53 跨源分析流程



相关链接

创建方式	增强型跨源连接
管理控制台操作	创建增强型跨源连接
API接口	《增强型跨源连接相关API》

5.3.2.2 创建增强型跨源连接

操作场景

使用DLI访问其他数据源的数据前，首先要通过建立增强型跨源连接打通DLI和数据源之间的网络，DLI才能够访问、导入、查询、分析其他数据源的数据。

例如：DLI连接MRS、RDS、CSS、Kafka、DWS时，需要打通DLI和对应数据源VPC之间的网络，才能实现数据互通。

本节操作介绍在控制台创建增强型跨源连接的操作步骤。

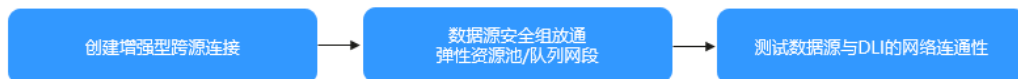
约束和限制

表 5-21 增强型跨源连接约束限制

限制项	说明
适用场景约束限制	<ul style="list-style-type: none"> 在同一队列中，如果同时使用了经典型跨源连接和增强型跨源连接，则经典型跨源连接优先于增强型跨源连接。推荐使用增强型跨源连接。 DLI提供的default队列不支持创建跨源连接。 Flink作业访问DIS，OBS和SMN数据源，无需创建跨源连接，可以直接访问。 增强型跨源仅支持包年包月队列、按需计费模式下的专属队列。
权限要求	<ul style="list-style-type: none"> 增强型跨源连接需要使用VPC、子网、路由、对等连接功能，因此需要获得VPC（虚拟私有云）的VPC Administrator权限。可在服务授权中进行设置。
使用约束限制	<ul style="list-style-type: none"> 使用DLI增强型跨源时，弹性资源池/队列的网段与数据源网段不能重合。 访问跨源表需要使用已经创建跨源连接的队列。 跨源表不支持Preview预览功能。
检测连通性要求	<ul style="list-style-type: none"> 检测跨源连接的连通性时对IP约束限制如下： <ul style="list-style-type: none"> IP必须为合法的IP地址，用“.”分隔的4个十进制数，范围是0-255。 测试时IP地址后可选择添加端口，用“:”隔开，端口最大限制5位，端口范围：0~65535。例如192.168.xx.xx或者192.168.xx.xx:8181。 检测跨源连接的连通性时对域名约束限制如下： <ul style="list-style-type: none"> 域名的限制长度为1到255的字符串，并且组成必须是字母、数字、下划线或者短横线。 域名的顶级域名至少包含两个及以上的字母，例如.com，.net，.cn等。 测试时域名后可选择添加端口，用“:”隔开，端口最大限制为5位，端口范围：0~65535。例如example.com:8080。

操作流程

图 5-54 创建增强型跨源连接操作流程



前提条件

- 已创建弹性资源池/队列用于绑定跨源连接。
- 已获取外部数据源的虚拟私有云、子网、内网IP、端口和安全组信息。
- 外部数据源的安全组已放通弹性资源池/队列的网段。

操作步骤

步骤1 创建增强型跨源连接

1. 登录DLI管理控制台。
2. 在左侧导航栏中，选择“跨源管理 > 增强型跨源”。
3. 选择“增强型跨源”，单击“创建”。

配置增强型跨源连接信息，详细参数介绍请参见[表5-22](#)。

表 5-22 参数说明

参数	参数说明
连接名称	所创建的跨源连接名称。 - 名称只能包含数字、英文字母、下划线。不能为空。 - 输入长度不能超过64个字符。
弹性资源池	可选参数，用于绑定使用跨源连接的弹性资源池或队列。 仅包周期和按需计费模式的专属队列支持绑定弹性资源池。 已上线弹性资源池的region，在队列管理中创建的包年包月或按需专属队列会默认创建同名的资源池。 说明 使用增强型跨源连接之前必须绑定队列且对等连接的状态是“active”。
绑定队列	可选参数，用于绑定需要使用跨源的队列。 仅包周期和按需计费模式的专属队列支持绑定弹性资源池。 说明 使用增强型跨源连接之前必须绑定队列且对等连接的状态是“active”。
虚拟私有云	数据源所使用的虚拟私有云。
子网	数据源所使用的子网。
路由表	显示子网实际绑定的路由表。 说明 - 此处的路由表为目的数据源子网关联的路由表，不同于“路由信息”中的路由。“路由信息”中的路由为所绑定的队列下子网关联的路由表中的路由。 - 目的数据源子网与队列所在子网为不同的子网，否则会造成网段冲突。

参数	参数说明
主机信息	<p>可选参数，用于配置主机的IP与域名的映射关系，在作业配置时只需使用配置的域名即可访问对应的主机。</p> <p>例如：访问MRS的HBase集群时需要配置Zookeeper实例的主机名（即域名）与对应的IP地址。每行填写一条记录，填写格式为：“IP 主机名/域名”。</p> <p>示例： 192.168.0.22 node-masterxxx1.com 192.168.0.23 node-masterxxx2.com</p> <p>获取主机信息的方法请参考怎样获取MRS主机信息?。</p>
标签	<p>使用标签标识云资源。包括标签键和标签值。如果您需要使用同一标签标识多种云资源，即所有服务均可在标签输入框下拉选择同一标签，建议在标签管理服务（TMS）中创建预定义标签。</p> <p>如您的组织已经设定DLI的相关标签策略，则需按照标签策略规则为资源添加标签。标签不符合标签策略的规则，则可能会导致资源创建失败，请联系组织管理员了解标签策略详情。</p> <p>具体请参考《标签管理服务用户指南》。</p> <p>说明</p> <ul style="list-style-type: none"> - 最多支持20个标签。 - 一个“键”只能添加一个“值”。 - 每个资源中的键名不能重复。 <p>- 标签键：在输入框中输入标签键名称。</p> <p>说明 标签的键的最大长度为128个字符，标签的键可以包含任意语种字母、数字、空格和_ . : + - @ ,但首尾不能含有空格，不能以_sys_开头。</p> <p>- 标签值：在输入框中输入标签值。</p> <p>说明 标签值的最大长度为255个字符，标签的值可以包含任意语种字母、数字、空格和_ . : + - @ 。</p>

4. 单击“确定”，创建增强型跨源连接。

创建完成后，增强型跨源连接的链接状态显示“已激活”，代表该链接创建成功。

步骤2 数据源所在安全组放通弹性资源池的网段

1. 在DLI管理控制台，获取弹性资源池/队列的网段。

单击“资源管理 > 队列管理”，选择运行作业的队列，单击队列名称旁的按钮，获取队列的网段信息。

2. 登录VPC控制台。找到数据源所在的VPC。

3. 查找安全组名称，在“弹性网卡 > 更多 > 更改安全组”中可以查到所属安全组。

4. 在左侧导航树选择“访问控制 > 安全组”。

5. 单击外部数据源所属的安全组名称，进入安全组详情界面。

- 在“入方向规则”页签中添加放通队列网段的规则。如图5-55所示。详细的入方向规则参数说明请参考表5-23。

图 5-55 添加入方向规则



表 5-23 入方向规则参数说明

参数	说明	取值样例
优先级	安全组规则优先级。 优先级可选范围为1-100，默认值为1，即最高优先级。优先级数字越小，规则优先级级别越高。	1
策略	安全组规则策略。	允许
协议端口	<ul style="list-style-type: none"> 网络协议。目前支持“All”、“TCP”、“UDP”、“ICMP”和“GRE”等协议。 端口：允许远端地址访问指定端口，取值范围为：1~65535。 	本例中选择TCP协议，端口值不填或者填写为数据源的端口。
类型	IP地址类型。	IPv4
源地址	源地址用于放通来自IP地址或另一安全组内的实例的访问。	本例填写获取的队列网段。
描述	安全组规则的描述信息，非必填项。	-

步骤3 测试DLI队列与数据源的连通性

- 获取数据源的私有内网IP和端口。
以RDS数据源为例：在RDS控制台“实例管理”页面，单击对应实例名称，查看“连接信息”>“内网地址”，即可获取RDS内网地址。查看“连接信息”>“数据库端口”，获取RDS数据库实例端口。

2. 在DLI管理控制台，单击“资源管理 > 队列管理”。
3. 选择增强型跨源所绑定的队列，单击操作列“更多 > 测试地址连通性”。
4. 输入数据源连接地址和端口，测试网络连通性。

格式：IP:端口

⚠ 注意

测试连接前，请确保外部数据源的安全组已放通队列的网段。

图 5-56 测试队列与数据源的网络连通性



----结束

相关操作

- [创建跨源成功但测试网络连通性失败怎么办？](#)

5.3.2.3 建立 DLI 与共享 VPC 中资源的网络连接

共享 VPC 简介

共享VPC是通过资源访问管理服务（RAM）将本账号的VPC资源共享给其他账号使用。例如，账号A可以将自己账号下创建的VPC和子网共享给账号B。在账号B接受共享以后，账号B可以查看到共享的VPC和子网，并可以使用该共享VPC和子网创建资源。

有关共享VPC的更多信息，请参见《虚拟私有云用户指南》的“[共享VPC](#)”相关内容。

DLI 使用场景

企业IT管理账号创建VPC和子网，并将该VPC和子网共享给其他企业业务账号，便于企业集中配置VPC安全策略，有利于资源有序集中管理。

企业业务账号使用共享的VPC和子网创建资源，并想要使用DLI提交作业访问共享VPC中的资源。此时需要建立DLI与共享VPC中资源的网络连接。

例如：账号A为企业IT管理账号，是VPC资源的所有者，创建VPC、子网。并将VPC、子网共享给企业业务账号B。

账号B为企业业务账号，使用共享的VPC和子网创建资源，并使用DLI访问资源。

前提条件

- 账号A已配置DLI云服务委托，且委托需包含DLI Datasource Connections Agency Access，具备访问和使用VPC、子网、路由、对等连接的权限。详细操作请参考[配置DLI云服务委托权限](#)。
- 作为资源所有者的账号A已创建共享VPC和子网，并指定资源使用者为账号B。创建共享的详细操作，请参见[创建共享](#)。

建立 DLI 与共享 VPC 下资源的网络连通

步骤1 账号A创建增强型跨源连接。

1. 账号A登录DLI管理控制台。
2. 在左侧导航栏中，选择“跨源管理 > 增强型跨源”。
3. 选择“增强型跨源”，单击“创建”。

配置增强型跨源连接信息，详细参数介绍请参见[表5-24](#)。

表 5-24 账号 A 创建的增强型跨源连接参数说明

参数	参数说明
连接名称	所创建的跨源连接名称。
弹性资源池	本场景下无需配置。
虚拟私有云	账号A共享给账号B的VPC。
子网	账号A共享给账号B的子网。
路由表	本场景下无需配置。
主机信息	本场景下无需配置。
标签	使用标签标识云资源。包括标签键和标签值。

4. 单击“确定”，创建增强型跨源连接。

步骤2 账号A将[步骤1](#)创建的增强型跨源连接授权给账号B使用。

1. 账号A在增强型跨源连接的列表页面，单击操作列下的“更多 > 权限管理”。
2. 选择赋权，输入账号B所在的项目ID，将该连接共享给账号B，授予账号B使用连接访问共享VPC资源的权限。
获取项目ID请参考[获取项目ID](#)。

步骤3 账号B在共享的增强型跨源连接上绑定DLI弹性资源池。

1. 账号B登录DLI管理控制台，
2. 在左侧导航栏中，选择“跨源管理 > 增强型跨源”。
3. 选择账号A共享的增强型跨源连接，单击操作列下的“更多 > 绑定弹性资源池”。
4. 选择已创建的弹性资源池，单击“确定”完成资源的绑定。
若无可选弹性资源池，可参考[创建弹性资源池并添加队列](#)创建新的弹性资源池。

步骤4 账号B测试弹性资源池与VPC中资源的网络连通性。

说明

若共享VPC下已有资源，请确保该资源所在的安全组已放通弹性资源池的网段。

1. 获取共享VPC下数据源的私有内网IP和端口。
以RDS数据源为例：在RDS控制台“实例管理”页面，单击对应实例名称，查看“连接信息”>“内网地址”，即可获得RDS内网地址。查看“连接信息”>“数据库端口”，获取RDS数据库实例端口。
2. 在DLI管理控制台，单击“资源管理 > 队列管理”。
3. 选择增强型跨源所绑定的资源池下的队列，单击操作列“更多 > 测试地址连通性”。
4. 输入数据源连接地址和端口，测试网络连通性。
若地址可达，说明账号B已建立DLI资源与共享VPC中的资源的网络连接，账号B可以使用DLI弹性资源池的队列提交作业访问共享VPC中的资源。

----结束

5.3.2.4 DLI 常用跨源分析开发方式

跨源分析

当DLI有访问外部数据源的业务需求时，首先需要通过建立增强型跨源连接，打通DLI与数据源之间的网络，再开发不同的作业访问数据源以实现DLI跨源分析。

本节操作介绍DLI支持的数据源对应的开发方式。

使用须知

- Flink作业访问DIS，OBS和SMN数据源，无需创建跨源连接，可以直接访问。
- 推荐使用增强型跨源连接打通DLI与数据源之间的网络。

跨源分析开发方式

表5-25提供DLI支持的数据源对应的开发方式。

表 5-25 跨源分析语法参考

服务名称	开发SQL作业	开发Spark jar作业	开发Flink OpenSource SQL作业	开发Flink Jar作业
CloudTable HBase	<ul style="list-style-type: none"> • 创建HBase关联表 • 插入数据 • 查询数据 	<ul style="list-style-type: none"> • scala样例代码 • pyspark 样例代码 • java样例代码 	<ul style="list-style-type: none"> • Hbase源表 • Hbase结果表 • Hbase维表 	-

服务名称	开发SQL作业	开发Spark jar作业	开发Flink OpenSource SQL作业	开发Flink Jar作业
CloudTable OpenTSDB	<ul style="list-style-type: none"> • 创建OpenTSDB关联表 • 插入数据 • 查询数据 	<ul style="list-style-type: none"> • scala样例代码 • pyspark样例代码 • java样例代码 	-	-
CSS	<ul style="list-style-type: none"> • 创建CSS关联表 • 插入数据 • 查询数据 	<ul style="list-style-type: none"> • scala样例代码 • pyspark样例代码 • java样例代码 	<ul style="list-style-type: none"> • Elasticsearch结果表 	-
DCS Redis	<ul style="list-style-type: none"> • 创建DCS关联表 • 插入数据 • 查询数据 	<ul style="list-style-type: none"> • scala样例代码 • pyspark样例代码 • java样例代码 	<ul style="list-style-type: none"> • Redis源表 • Redis结果表 • Redis维表 	<ul style="list-style-type: none"> • Flink作业样例
DDS	<ul style="list-style-type: none"> • 创建DDS关联表 • 插入数据 • 查询数据 	<ul style="list-style-type: none"> • scala样例代码 • pyspark样例代码 • java样例代码 	-	-
DMS	-	-	<ul style="list-style-type: none"> • Kafka源表 • Kafka结果表 	-
DWS	<ul style="list-style-type: none"> • 创建DWS关联表 • 插入数据 • 查询数据 	<ul style="list-style-type: none"> • scala样例代码 • pyspark样例代码 • java样例代码 	<ul style="list-style-type: none"> • DWS源表 • DWS结果表 • DWS维表 	<ul style="list-style-type: none"> • Flink作业样例
MRS HBase	<ul style="list-style-type: none"> • 创建HBase关联表 • 插入数据 • 查询数据 	<ul style="list-style-type: none"> • scala样例代码 • pyspark样例代码 • java样例代码 	<ul style="list-style-type: none"> • Hbase源表 • Hbase结果表 • Hbase维表 	<ul style="list-style-type: none"> • Flink作业样例

服务名称	开发SQL作业	开发Spark jar作业	开发Flink OpenSource SQL作业	开发Flink Jar作业
MRS Kafka	-	-	<ul style="list-style-type: none"> • Kafka源表 • Kafka结果表 	<ul style="list-style-type: none"> • Flink作业样例
MRS OpenTSDB	<ul style="list-style-type: none"> • 创建OpenTSDB关联表 • 插入数据 • 查询数据 	<ul style="list-style-type: none"> • scala样例代码 • pyspark样例代码 • java样例代码 	-	-
RDS MySQL	<ul style="list-style-type: none"> • 创建RDS关联表 • 插入数据 • 查询数据 	<ul style="list-style-type: none"> • scala样例代码 • pyspark样例代码 • java样例代码 	<ul style="list-style-type: none"> • MySQL CDC源表 	-
RDS PostGre	<ul style="list-style-type: none"> • 创建RDS关联表 • 插入数据 • 查询数据 	<ul style="list-style-type: none"> • scala样例代码 • pyspark样例代码 • java样例代码 	<ul style="list-style-type: none"> • Postgres CDC源表 	-

5.3.3 使用 DEW 管理数据源访问凭证

使用DLI提交作业读写外部数据源数据时，需要妥善保存数据源的访问凭证确保数据源认证的安全性，便于DLI安全访问数据源。数据加密服务（Data Encryption Workshop, DEW）是一个综合的云上数据加密服务，为您解决数据安全、密钥安全、密钥管理复杂等问题。本节操作介绍使用数据加密服务DEW存储数据源的认证信息的操作步骤。

了解[数据加密服务](#)。

在 DEW 创建通用凭据

本例以配置RDS实例访问凭据为例，介绍在DEW保存凭据，并在DLI作业中的配置示例。

1. 登录DEW管理控制台
2. 选择“凭据管理”，进入“凭据管理”页面。
3. 单击“创建凭据”，配置凭据基本信息
 - 凭据名称：待创建凭据的名称。本例名称为secretInfo。
 - 凭据值：配置RDS实例的用户名和密码。

- 第一行凭据值的键为MySQLUsername，值为RDS实例的用户名。
- 第二行凭据值的键为MySQLPassword，值为RDS实例的密码。

图 5-57 设置凭据值



4. 按需完成其他参数的配置后，单击“确定”保存凭据。
了解更多请参考[创建通用凭据](#)。

在 DLI 作业中使用 DEW 中创建的凭据

以Flink作业为例介绍使用DEW凭据的方法。

参考[Flink Opensource SQL使用DEW管理访问凭据](#)的语法格式，在作业中配置凭据信息。

```
WITH (
  'connector' = 'jdbc',
  'url' = 'jdbc:mysql://MySQLAddress:MySQLPort/flink', --其中url中的flink表示MySQL中orders表所在的数据库名
  'table-name' = 'orders',
  'username' = 'MySQLUsername', -- DEW服务中，名称为secretInfo，且版本号v1的通用凭证，定义凭证值的键MySQLUsername，它的值为用户的敏感信息。
  'password' = 'MySQLPassword', -- DEW服务中，名称为secretInfo，且版本号v1的通用凭证，定义凭证值的键MySQLPassword，它的值为用户的敏感信息。
  'sink.buffer-flush.max-rows' = '1',
  'dew.endpoint'='kms.xxx.com', --使用的DEW服务所在的endpoint信息
  'dew.csms.secretName'='secretInfo', --DEW服务通用凭据的凭据名称
  'dew.csms.decrypt.fields'='username,password', --其中username,password字段值，需要利用DEW凭证管理,进行解密替换。
  'dew.csms.version'='v1'
);
```

相关操作

了解更多使用DLI委托获取访问凭据的开发指南请参考[表5-26](#)。

表 5-26 DLI 委托权限配置场景开发指南

类型	操作指导	说明
Flink作业场景	Flink Opensource SQL使用DEW管理访问凭据	Flink Opensource SQL场景使用DEW管理和访问凭据的操作指导，将Flink作业的输出数据写入到Mysql或DWS时，在connector中设置账号、密码等属性。

类型	操作指导	说明
	Flink Jar 使用DEW获取访问凭证读写OBS	访问OBS的AKSK为例介绍Flink Jar使用DEW获取访问凭证读写OBS的操作指导。
	用户获取Flink作业委托临时凭证	DLI提供了一个通用接口，可用于获取用户在启动Flink作业时设置的委托的临时凭证。该接口将获取到的该作业委托的临时凭证封装到com.huaweicloud.sdk.core.auth.BasicCredentials类中。 本操作介绍获取Flink作业委托临时凭证的操作方法。
Spark作业场景	Spark Jar 使用DEW获取访问凭证读写OBS	访问OBS的AKSK为例介绍Spark Jar使用DEW获取访问凭证读写OBS的操作指导。
	用户获取Spark作业委托临时凭证	本操作介绍获取Spark Jar作业委托临时凭证的操作方法。

5.3.4 使用 DLI 的跨源认证管理数据源访问凭证

5.3.4.1 跨源认证概述

什么是跨源认证？

跨源分析场景中，如果在作业中直接配置认证信息会触发密码泄露的风险，因此推荐您使用“数据加密服务DEW”或“DLI提供的跨源认证方式”来存储数据源的认证信息。

- 数据加密服务（Data Encryption Workshop, DEW）是一个综合的云上数据加密服务，为您解决数据安全、密钥安全、密钥管理复杂等问题。推荐使用数据加密服务DEW来存储数据源的认证信息。

Spark 3.3.1及以上版本、Flink 1.15及以上版本的跨源访问场景推荐使用数据加密服务DEW来存储数据源的认证信息，为您解决数据安全、密钥安全、密钥管理复杂等问题。具体操作请参考[使用DEW管理数据源访问凭证](#)。

- 跨源认证用于管理访问指定数据源的认证信息。配置跨源认证后，无需在作业中重复配置数据源认证信息，提高数据源认证的安全性，便于DLI安全访问数据源。SQL作业、Flink 1.12版本的跨源访问场景，使用DLI提供的“跨源认证”管理数据源的访问凭证。

本节操作为您介绍DLI提供的跨源认证的使用方法。

约束与限制

表 5-27 跨源认证约束限制

限制项	说明
适用场景约束限制	<ul style="list-style-type: none">仅Spark SQL、和Flink OpenSource SQL 1.12版本的作业支持使用跨源认证。仅在2023年5月1日后创建的队列，支持Flink作业使用跨源认证。
跨源认证类型	<ul style="list-style-type: none">DLI支持四种类型的跨源认证，不同的数据源按需选择相应的认证类型。<ul style="list-style-type: none">CSS类型跨源认证：适用于“6.5.4”及以上版本的CSS集群且集群已开启安全模式。Kerberos类型的跨源认证：适用于开启Kerberos认证的MRS安全集群。Kafka_SSL类型的跨源认证：适用于开启SSL的Kafka。Password类型的跨源认证：适用于DWS、RDS、DDS、DCS数据源。

跨源认证类型

DLI支持四种类型的跨源认证，不同的数据源按需选择相应的认证类型。

- CSS类型跨源认证：适用于“6.5.4”及以上版本的CSS集群且集群已开启安全模式。配置时需指定集群的用户名、密码、认证证书，通过跨源认证将以上信息存储到DLI服务中，便于DLI安全访问CSS数据源。详细操作请参考[创建CSS类型跨源认证](#)。
- Kerberos类型的跨源认证：适用于开启Kerberos认证的MRS安全集群。配置时需指定MRS集群认证凭证，包括“krb5.conf”和“user.keytab”文件。详细操作请参考[创建Kerberos跨源认证](#)。
- Kafka_SSL类型的跨源认证：适用于开启SSL的Kafka，配置时需指定KafkaTruststore路径和密码。详细操作请参考[创建Kafka_SSL类型跨源认证](#)。
- Password类型的跨源认证：适用于DWS、RDS、DDS、DCS数据源，配置时将数据源的密码信息存储到DLI。详细操作请参考[创建Password类型跨源认证](#)。

支持跨源认证的数据源与作业类型

不同类型的作业支持跨源认证的数据源与认证方式不同。

- Spark SQL支持跨源认证的数据源与约束限制请参考[表5-28](#)。
- Flink OpenSource SQL 1.12支持跨源认证的数据源与约束限制请参考[表5-29](#)。

表 5-28 Spark SQL 支持跨源认证的数据源

跨源认证类型	数据源	约束与限制
CSS	CSS	CSS集群版本选择“6.5.4”或“6.5.4”以上版本。 CSS集群已开启“安全模式”。
Password	DWS、RDS、DDS、Redis	-

表 5-29 Flink OpenSource SQL 1.12 支持跨源认证的数据源

表类型	跨源认证类型	数据源	约束与限制
源表	Kerberos	HBase	MRS安全集群已开启Kerberos认证。
		Kafka	MRS Kafka开启Kerberos认证。
	Kafka_SSL	Kafka	DMS Kafka开启SASL_SSL认证。 MRS Kafka开启SASL认证。 MRS Kafka开启SSL认证。
	Password	DWS、RDS、Redis	-
结果表	Kerberos	HBase	MRS安全集群已开启Kerberos认证。
		Kafka	MRS Kafka开启Kerberos认证。
	Kafka_SSL	Kafka	DMS Kafka开启SASL_SSL认证。 MRS Kafka开启SASL认证。 MRS Kafka开启SSL认证。
	Password	DWS、RDS、CSS、Redis	-
维表	Kerberos	HBase	MRS安全集群已开启Kerberos认证。
	Password	DWS、RDS、Redis	-

5.3.4.2 创建 CSS 类型跨源认证

操作场景

通过在DLI控制台创建的CSS类型的跨源认证，将CSS安全集群的认证信息存储到DLI，无需在SQL作业中配置账号密码，安全访问CSS安全集群。

本节操作介绍在DLI控制台创建CSS安全集群的跨源认证的操作步骤。

操作须知

已创建CSS安全集群，且集群满足以下条件：

- CSS集群版本选择“6.5.4”或“6.5.4”以上版本。
- CSS集群已开启“安全模式”。

创建CSS安全集群请参考[创建Elasticsearch类型集群（安全模式）](#)。

操作步骤

1. 下载CSS安全集群的认证凭证。
 - a. 登录CSS服务管理控制台，单击“集群管理”。
 - b. 在“集群管理”页面中，单击对应的集群名称，进入“基本信息”页面。
 - c. 单击“安全模式”后的下载证书，下载CSS安全集群的证书。
2. 将认证凭证上传到OBS桶。
关于如何创建OBS桶并上传数据，请参考《[对象存储服务快速入门](#)》。
3. 创建跨源认证。
 - a. 登录DLI管理控制台。
 - b. 选择“跨源管理 > 跨源认证”。
 - c. 单击“创建”。

填写CSS认证信息，详细参数说明请参考[表5-30](#)。

表 5-30 参数说明

参数	参数说明
认证信息名称	所创建的跨源认证信息名称。 <ul style="list-style-type: none"> • 名称只能包含数字、英文字母和下划线，但不能是纯数字，且不能以下划线开头。 • 输入长度不能超过128个字符。 • 建议名称中包含CSS安全集群的名称，便于区分不同集群的安全认证信息。
类型	选择CSS。
用户名	安全集群的登录用户名。
用户密码	安全集群的登录密码。
Certificate 路径	上传“安全证书”的OBS路径。即 步骤2 的OBS桶地址。

图 5-58 创建认证信息-CSS

创建认证信息

类型 CSS

* 认证信息名称 请输入认证信息名称

* 用户名 请输入用户名 ?

* 用户密码 请输入用户密码 ?

* Certificate路径 obs://

确定 取消

4. 访问CSS的表。

跨源认证创建成功后，在创建访问CSS的表时只需关联跨源认证即可安全访问数据源。

例如在使用Spark SQL来创建访问CSS的表时使用es.certificate.name字段配置跨源认证信息名称，配置连接安全CSS集群。

创建完跨源认证，可以参考[创建DLI表关联CSS](#)使用Spark SQL来创建访问CSS的表。

5.3.4.3 创建 Kerberos 跨源认证

操作场景

通过在DLI控制台创建的Kerberos类型的跨源认证，将数据源的认证信息存储到DLI，无需在SQL作业中配置账号密码，安全访问数据源。

说明

- MRS Kafka开启Kerberos认证，未开启SSL认证时，创建Kerberos类型的认证。建表时通过krb_auth_name关联跨源认证。
- MRS Kafka开启Kerberos认证，同时开启了SSL认证时，需要同时创建Kerberos和Kafka_SSL类型的认证。建表时分别通过krb_auth_name和ssl_auth_name关联跨源认证。
- MRS Kafka未开启Kerberos认证，仅开启了SASL认证时（例如使用账号密码认证PlainLoginModule场景），无需使用跨源认证。
- MRS Kafka未开启Kerberos认证，仅开启了SSL认证时，创建Kafka_SSL类型的认证。建表时通过ssl_auth_name关联跨源认证。
- MRS Kafka未开启Kerberos认证，开启了SASL认证和SSL认证时，创建Kafka_SSL类型的认证。建表时通过ssl_auth_name关联跨源认证。

Kerberos 类型跨源认证支持连接的数据源

Kerberos类型跨源认证支持连接的数据源如[表5-31](#)所示。

表 5-31 Kerberos 类型跨源认证支持连接的数据源

作业类型	表类型	数据源	约束与限制
Flink OpenSource SQL	源表	HBase	MRS安全集群已开启Kerberos认证。
		Kafka	MRS Kafka开启Kerberos认证。
	结果表	HBase	MRS安全集群已开启Kerberos认证。
		Kafka	MRS Kafka开启Kerberos认证。
	维表	HBase	MRS安全集群已开启Kerberos认证。

操作步骤

1. 下载数据源的认证凭证。
 - a. 登录MRS Manager界面。
 - b. 选择“系统 > 权限 > 用户”。
 - c. 单击“更多 > 下载认证凭据”，保存后解压得到用户的keytab文件与krb5.conf文件。
2. 上传认证凭证到OBS桶。
关于如何创建OBS桶并上传数据，请参考《[对象存储服务快速入门](#)》。
3. 创建跨源认证。
 - a. 登录DLI管理控制台。
 - b. 选择“跨源管理 > 跨源认证”。
 - c. 单击“创建”。
填写Kerberos认证信息，详细参数说明请参考[表5-32](#)。

表 5-32 参数说明

参数	参数说明
类型	选择kerberos。
认证信息名称	所创建的跨源认证信息名称。 <ul style="list-style-type: none"> ● 名称只能包含数字、英文字母和下划线，但不能是纯数字，且不能以下划线开头。 ● 输入长度不能超过128个字符。 ● 建议名称中包含MRS安全集群的名称，便于区分不同集群的安全认证信息。
用户名	安全集群的登录用户名。

参数	参数说明
krb5_conf路径	上传“krb5.conf”文件的OBS路径。 说明 “krb5.conf”中需移除[libdefaults]下的“renew_lifetime”配置项，否则可能会遇到“Message stream modified (41)”问题。
keytab路径	上传“user.keytab”文件的OBS路径。

图 5-59 创建认证信息-Kerberos

创建认证信息
✕

类型 ▼
Kerberos

★ 认证信息名称

★ 用户名 ?

★ krb5_conf路径 🗑️

★ keytab路径 🗑️

确定
取消

4. 访问MRS的表。

跨源认证创建成功后，在创建访问数据源时只需关联跨源认证即可安全访问数据源。

建表时关联跨源认证的字段请参考[表5-33](#)。

表 5-33 建表时与 Kerberos 类型跨源认证关联的字段

作业类型	数据源	参数	是否必选	数据类型	说明
Flink OpenS ource SQL	HBa se	krb_a uth_n ame	否	String	创建源表、结果表、维表时均使用该字段关联跨源认证。

作业类型	数据源	参数	是否必选	数据类型	说明
	Kafka	krb_auth_name	否	String	<p>创建源表、结果表时均使用该字段关联跨源认证。</p> <p>创建的Kerberos类型的跨源认证名称。</p> <p>如果使用SASL_PLAINTEXT类型，且使用Kerberos认证，则需要同时配置以下参数：</p> <ul style="list-style-type: none"> 'properties.sasl.mechanism' = 'GSSAPI' 'properties.security.protocol' = 'SASL_PLAINTEXT'

具体的建表操作指导请参考DLI 语法参考。

- Flink OpenSource SQL语法参考：[创建HBase源表](#)

5.3.4.4 创建 Kafka_SSL 类型跨源认证

操作场景

通过在DLI控制台创建的Kafka_SSL类型的跨源认证，将Kafka的认证信息存储到DLI，无需在SQL作业中配置账号密码，安全访问Kafka实例。

说明

- MRS Kafka开启Kerberos认证，未开启SSL认证时，创建Kerberos类型的认证。建表时通过krb_auth_name关联跨源认证。
- MRS Kafka开启Kerberos认证，同时开启了SSL认证时，需要同时创建Kerberos和Kafka_SSL类型的认证。建表时分别通过krb_auth_name和ssl_auth_name关联跨源认证。
- MRS Kafka未开启Kerberos认证，仅开启了SASL认证时（例如使用账号密码认证PlainLoginModule场景），无需使用跨源认证。
- MRS Kafka未开启Kerberos认证，仅开启了SSL认证时，创建Kafka_SSL类型的认证。建表时通过ssl_auth_name关联跨源认证。
- MRS Kafka未开启Kerberos认证，开启了SASL认证和SSL认证时，创建Kafka_SSL类型的认证。建表时通过ssl_auth_name关联跨源认证。

Kafka_SSL 类型跨源认证支持连接的数据源

Kafka_SSL类型跨源认证支持连接的数据源如[表5-34](#)所示。

表 5-34 Kafka_SSL 类型跨源认证支持连接的数据源

作业类型	表类型	数据源	约束与限制
Flink OpenSource SQL	源表、结果表	Kafka	DMS Kafka开启SASL_SSL认证。 MRS Kafka开启SASL认证。 MRS Kafka开启SSL认证。

操作步骤

1. 下载认证凭证。
 - **DMS Kafka**
 - i. 登录DMS Kafka控制台，单击实例名称进入详情页面。
 - ii. 在连接信息中，找到SSL证书，单击“下载”。
解压下载的kafka-certs压缩包，获取client.jks和phy_ca.crt文件。
 - **MRS Kafka**
 - i. 登录MRS Manager界面。
 - ii. 选择“系统 > 权限 > 用户”。
 - iii. 单击“更多 > 下载认证凭据”，保存后解压得到Truststore文件。
2. 上传认证凭证到OBS桶。
关于如何创建OBS桶并上传数据，请参考《[对象存储服务快速入门](#)》。
3. 创建跨源认证。
 - a. 登录DLI管理控制台。
 - b. 选择“跨源管理 > 跨源认证”。
 - c. 单击“创建”。
填写Kafka认证信息，详细参数说明请参考[表5-35](#)。

表 5-35 参数说明

参数	参数说明
类型	选择Kafka_SSL。
认证信息名称	所创建的跨源认证信息名称。 <ul style="list-style-type: none"> ● 名称只能包含数字、英文字母和下划线，但不能是纯数字，且不能以下划线开头。 ● 输入长度不能超过128个字符。
Truststore路径	上传SSL Truststore文件的OBS路径。 <ul style="list-style-type: none"> ● MRS Kafka请填写Truststore.jks文件的OBS路径。 ● DMS Kafka请填写client.jks文件的OBS路径。
Truststore密码	truststore密码。

参数	参数说明
Keystore路径	上传SSL KEYSTORE(密钥和证书)文件的OBS路径。
Keystore密码	keystore(密钥和证书)密码。
Key密码	keystore中的私钥密码。

图 5-60 创建认证信息-Kafka_SSL

×

创建认证信息

类型	<input type="text" value="Kafka_SSL"/>
* 认证信息名称	<input type="text" value="请输入认证信息名称"/>
* Truststore路径	<input type="text" value="obs://"/>
Truststore密码	<input type="password" value="请输入密码"/>
Keystore路径	<input type="text" value="obs://"/>
Keystore密码	<input type="password" value="请输入密码"/>
Key密码	<input type="password" value="请输入密码"/>

确定 取消

4. 访问开启SASL_SSL认证的Kafka。

跨源认证创建成功后，在创建访问数据源时只需关联跨源认证即可安全访问数据源。

建表时关联跨源认证的字段请参考[表5-36](#)。

表 5-36 建表时与 Kafka_SSL 类型跨源认证关联的字段

参数	是否必选	数据类型	说明
ssl_auth_name	否	String	<p>创建源表、结果表、维表时均使用该字段关联跨源认证。</p> <p>创建的Kafka_SSL类型的跨源认证名称。Kafka配置SSL时使用该配置。</p> <ul style="list-style-type: none"> • 如果仅使用SSL类型，则需要同时配置以下参数： 'properties.security.protocol' = 'SSL'; • 如果使用SASL_SSL类型，则需要同时配置以下参数： <ul style="list-style-type: none"> - 'properties.security.protocol' = 'SASL_SSL'、 - 'properties.sasl.mechanism' = 'GSSAPI或者PLAIN'、 - 'properties.sasl.jaas.config' = 'org.apache.kafka.common.security.plain.PlainLoginModule required username=\"xxx\" password=\"xxx\";'

具体的建表操作指导请参考DLI 语法参考。

- Flink OpenSource SQL语法参考：[创建Kafka源表](#)

5.3.4.5 创建 Password 类型跨源认证

操作场景

通过在DLI控制台创建的Password类型的跨源认证，将DWS、RDS、DCS和DDS数据源的密码信息存储到DLI，无需在SQL作业中配置账号密码，安全访问DWS、RDS、DDS、DCS数据源。

Password 类型跨源认证支持连接的数据源

Password类型跨源认证支持连接的数据源如表5-37所示。

表 5-37 Password 类型跨源认证支持连接的数据源

作业类型	表类型	数据源
Spark SQL	-	DWS、RDS、DDS、Redis
Flink OpenSource SQL	源表	DWS、RDS、Redis
	结果表	DWS、RDS、CSS、Redis
	维表	DWS、RDS、Redis

操作步骤

1. 创建跨源认证。
 - a. 登录DLI管理控制台。
 - b. 选择“跨源管理 > 跨源认证”。
 - c. 单击“创建”。
填写认证信息，详细参数说明请参考[表5-38](#)。

表 5-38 参数说明

参数	参数说明
类型	选择Password。
认证信息名称	所创建的跨源认证信息名称。 <ul style="list-style-type: none"> ● 名称只能包含数字、英文字母和下划线，但不能是纯数字，且不能以下划线开头。 ● 输入长度不能超过128个字符。
用户名称	访问数据源的用户名。
用户密码	访问数据源的密码。

图 5-61 创建认证信息-Password

2. 访问数据源。
跨源认证创建成功后，在创建访问数据源时只需关联跨源认证即可安全访问数据源。
建表时关联跨源认证的字段请参考[表5-39](#)。

表 5-39 建表时与 Password 类型跨源认证关联的字段

作业类型	参数	是否必选	数据类型	说明
Spark SQL	passwd_auth	否	String	跨源认证名称。适用于DWS、RDS、DDS、Redis数据源。
Flink OpenSource SQL	pwd_auth_name	否	String	创建源表、结果表、维表时均使用该字段关联跨源认证。 通过配置pwd_auth_name字段写入创建的Password类型的跨源认证名称。 如果配置该参数则不需要在SQL中配置数据源的账号密码。

具体的建表操作指导请参考DLI 语法参考。

- Flink OpenSource SQL语法参考：[创建DWS源表](#)

5.3.4.6 跨源认证权限管理

操作场景

通过跨源认证的用户授权，可设置分配不同的跨源认证，且不同用户的作业不影响跨源认证的使用。

使用须知

- 管理员用户和跨源认证的所有者拥有所有权限，不需要进行权限设置且其他用户无法修改其跨源认证权限。
- 给新用户设置跨源认证权限时，该用户所在用户组具有Tenant Guest权限。
关于Tenant Guest权限的介绍和开通方法，详细参见《[权限策略](#)》和《[统一身份认证服务用户指南](#)》中的[创建用户组](#)。

跨源认证用户授权

1. 登录DLI管理控制台。
2. 单击“跨源管理 > 跨源认证”。
3. 选择要进行授权的跨源认证，单击操作列“权限管理”进入跨源认证的用户权限信息页面。
4. 单击页面右上角“授权”可对当前的跨源认证新增用户授权。

图 5-62 跨源认证用户授权

授权

* 用户名

请勾选需要赋予的权限

全选
 使用
 更新
 删除
 查看其他用户具备的权限

赋权
 回收

表 5-40 用户授权参数说明

参数名称	描述
用户名	被授权的IAM用户的名称。 说明 该用户名称是已存在的IAM用户名称。
权限设置	<ul style="list-style-type: none">● 使用：使用该跨源认证。● 更新：更新该跨源认证。● 删除：删除该跨源认证。● 赋权：当前用户可将跨源认证的权限赋予其他用户。● 回收：当前用户可回收其他用户具备的该跨源认证的权限，但不能回收该跨源认证所有者的权限。● 查看其他用户具备的权限：当前用户可查看其他用户具备的该跨源认证的权限。

修改当前用户的权限

1. 登录DLI管理控制台。
2. 单击“跨源管理 > 跨源认证”。
3. 选择要进行授权的跨源认证，单击操作列“权限管理”进入跨源认证的用户权限信息页面。
4. 单击操作列的“权限设置”，修改当前用户的权限。详细权限描述如表5-40所示。

说明

- 当“权限设置”中的选项为灰色时，表示您不具备修改此跨源认证权限的权限。可以向管理员用户、组所有者等具有赋权权限的用户申请“跨源认证的赋权”和“跨源认证权限的回收”权限。
- 如需回收当前用户的全部权限，可单击操作列的“回收”，该子用户将不具备该跨源认证的任意权限。

5.3.5 管理增强型跨源连接



5.3.5.1 查看增强型跨源连接的基本信息

增强型跨源连接创建完成后您可以通过管理控制台查看和管理您的增强型跨源连接。

本节操作介绍在管理控制台如何查看增强型跨源连接基本信息，包括增强型跨源连接的是否支持IPv6、主机信息等。

查看增强型跨源连接的基本信息

1. 登录DLI管理控制台。
2. 选择“跨源管理 > 增强型跨源”。
3. 进入增强型跨源连接列表页面，选择您需要查看的增强型跨源连接。

- 在列表页面的右上方单击可以自定义显示列，并设置表格内容显示规则、操作列显示规则。
 - 在列表页面上方的搜索区域，您可以名称和标签筛选需要的增强型跨源连接。
4. 单击查看增强型跨源连接详细信息。
- 支持查看以下信息：
- a. 是否支持IPv6：如果创建增强型跨源连接时您选择的子网是开启IPv6的，则您创建的增强型跨源连接也是支持IPv6的。
 - b. 主机信息：访问MRS的HBase集群时需要配置实例的主机名（即域名）与主机对应的IP地址。详细信息请参考[修改弹性资源池的主机信息](#)。

5.3.5.2 增强型跨源连接权限管理

操作场景

增强型跨源支持项目级授权，授权后，项目内的用户具备该增强型跨源连接的操作权。可查看该增强型跨源连接、可将创建的弹性资源池与该增强型跨源连接绑定、可自定义路由等操作。以此实现增强型跨源连接的跨项目应用。本节操作介绍对增强型跨源连接授权或回收权限的操作步骤。

说明

- 如果被授权的项目属于相同区域（region）的不同用户，则需使用被授权项目所属的用户账号进行登录。
- 如果被授权的项目属于相同区域（region）的同一用户，则需使用当前账号切换到对应的项目下。

应用示例

项目B需要访问项目A上的数据源，对应操作如下。

- 对于项目A：
 - a. 使用项目A对应的账号登录DLI服务。
 - b. 通过对应数据源的VPC信息在DLI服务中创建增强型跨源连接“ds”。
 - c. 将增强型跨源连接“ds”授权给项目B。
- 对于项目B：
 - a. 使用项目B对应的账号登录DLI服务。
 - b. 对增强型跨源连接“ds”进行绑定队列操作。
 - c. （可选）设置主机信息，创建路由。

通过上述操作项目A的增强型跨源连接与项目B的队列创建了对等连接和路由，即可在项目B的队列上创建作业访问项目A的数据源。

操作步骤

1. 登录DLI管理控制台。
2. 在左侧导航栏中，选择“跨源管理 > 增强型跨源”。

3. 选择待操作的增强型跨源连接，单击操作列的“更多 > 权限管理”。
 - **授权：**
 - i. 在权限管理页面，权限设置选择“授权”。
 - ii. 输入项目ID。
 - iii. 单击“确定”，授予该项目弹性资源池的操作权限。
 - **回收权限：**
 - i. 在权限管理页面，权限设置选择“回收”。
 - ii. 输入项目ID。
 - iii. 单击“确定”，回收指定项目的弹性资源池操作权。

5.3.5.3 增强型跨源连接绑定弹性资源池

操作场景

如果其他弹性资源池想要通过已创建的增强型跨源连接来连接数据源，可以在增强型跨源连接页面绑定弹性资源池。本节的操作指导介绍增强型跨源连接绑定弹性资源池的操作指导。

约束限制

- 增强型跨源仅支持包年包月队列和按需专属的弹性资源池/队列。
- 绑定跨源的DLI队列网段和数据源网段不能重合。
- 不支持绑定系统预置的default队列。

操作步骤

1. 登录DLI管理控制台。
2. 在左侧导航栏中，选择“跨源管理 > 增强型跨源”。
3. 绑定弹性资源池。
 - a. 选择待绑定的增强型跨源连接，单击操作列的“更多 > 绑定弹性资源池”。
 - b. 在绑定弹性资源池的对话框中，勾选待绑定的弹性资源池。
 - c. 单击“确定”，弹性资源池的绑定。
4. 绑定完成后，在增强型跨源的列表页面可以查看连接状态。
 - 增强型跨源创建后状态为“已激活”，但不能说明队列和数据源已连通。建议前往队列管理页面测试数据源网络是否打通。操作步骤如下：
 - i. 在队列管理页面选择队列。
 - ii. 单击“操作”列中的“更多 > 测试地址连通性”。
 - iii. 输入数据源的“IP:端口”测试网络连通性。
 - 在增强型跨源连接的详情页可以查看对等连接的相关信息。
 - 对等连接ID：增强型跨源在该队列所在集群中创建的对等连接ID。
每一个增强型跨源对每一个绑定的队列都会创建一个对等连接。该对等连接用于实现跨VPC通信，请确保数据源使用的安全组开放了DLI队列网段的访问，并且在使用跨源过程中不要删除该对等连接。


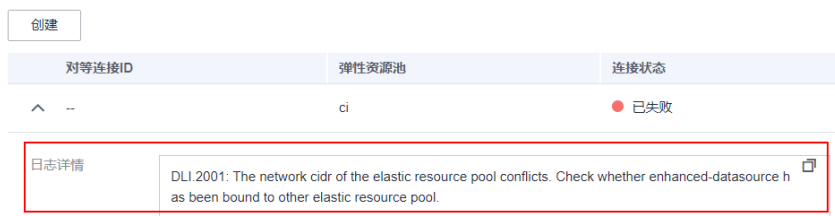
- 对等连接的连接状态：
跨源连接的状态信息，包括以下三种状态：创建中、已激活、已失败。
当连接状态显示为“已失败”时，单击左边对应的 ，可查看详细的错误信息。

图 5-63 查看增强型跨源连接详情



5.3.5.4 增强型跨源连接与弹性资源池解绑

操作场景

当弹性资源池不需要使用增强型跨源连接访问数据源时，可将增强型跨源连接与弹性资源池解绑。

约束限制

增强型跨源绑定弹性资源池所创建的对等连接状态为“已失败”时，不支持解绑该弹性资源池。

操作步骤

- 登录DLI管理控制台。
- 在左侧导航栏中，选择“跨源管理 > 增强型跨源”。
- 解绑弹性资源池。
 - 方法一：
 - 选择待删除的增强型跨源连接，单击操作列的“更多 > 解绑弹性资源池”。
 - 在解绑弹性资源池的对话框中，勾选弹性资源池。
 - 单击“确定”，解除弹性资源池与增强型跨源连接的绑定关系。
 - 方法二：
 - 选择待删除的增强型跨源连接，单击列表中的连接名称，进入连接”。
 - 选择待解绑的弹性资源池，单击操作列“解绑弹性资源池”。
 - 单击“是”，解除弹性资源池与增强型跨源连接的绑定关系。

5.3.5.5 添加增强型跨源连接的路由信息

操作场景

路由即路由规则，在路由中通过配置目的地址、下一跳类型、下一跳地址等信息，来决定网络流量的走向。路由分为系统路由和自定义路由。

增强型跨源连接创建后，子网会自动关系统默认路由。除了系统默认路由，您可以根据需要添加自定义路由规则，将指向目的地址的流量转发到指定的下一跳地址。

了解更多路由相关信息请参考[路由表](#)。

📖 说明

- 创建增强型跨源时的路由表是数据源子网关联的路由表。
- 添加路由信息页的路由是弹性资源池子网关联的路由表中的路由。
- 数据源子网与弹性资源池所在子网为不同的子网，否则会造成网段冲突。

操作步骤

1. 登录DLI管理控制台。
2. 在左侧导航栏中，选择“跨源管理 > 增强型跨源”。
3. 选择待添加路由的增强型跨源连接，并添加路由。
 - 方法一：
 - i. 选择待添加的增强型跨源连接，单击操作列的“路由信息”。
 - ii. 单击“添加路由”。
 - iii. 在添加路由的对话框中，填写路由信息。参数说明请参考[表5-41](#)。
 - iv. 单击“确定”。
 - 方法二：
 - i. 选择待添加的增强型跨源连接，单击操作列的“更多 > 添加路由”。
 - ii. 在添加路由的对话框中，填写路由信息。参数说明请参考[表5-41](#)。
 - iii. 单击“确定”。

表 5-41 自定义路由详情列表参数

参数	参数说明
路由名称	自定义路由的名称，在同一个增强型跨源中唯一。名称规则为：长度1~64字节，数字、字母、下划线（"_"）、中划线（"-"）组成。
路由IP	自定义路由网段，允许不同路由的网段之间有交集，但不允许完全相同。 禁止添加100.125.xx.xx、100.64.xx.xx网段，避免与SWR等服务的内网网段重复，导致增强型跨源连接失败。

4. 添加路由信息后，您可以在路由详情页查看添加的路由信息。

5.3.5.6 删除增强型跨源连接的路由信息

操作场景

本节操作指导用户删除不再使用的路由信息。

约束限制

当自定义路由表被关联至子网时，则无法删除。

请先通过[更换子网关联的路由表](#)将子网关联到其他的路由表，然后尝试删除。

操作步骤

1. 登录DLI管理控制台。
2. 在左侧导航栏中，选择“跨源管理 > 增强型跨源”。
3. 选择待添加路由的增强型跨源连接，并删除路由。
 - 方法一：
 - i. 选择待删除的增强型跨源连接，单击操作列的“路由信息”。
 - ii. 选择待删除的路由信息，单击操作列的“删除”。
 - iii. 单击“确定”。
 - 方法二：
 - i. 选择待删除的增强型跨源连接，单击操作列的“更多 > 删除路由”。
 - ii. 选择待删除的路由信息，
 - iii. 单击“是”。

5.3.5.7 修改弹性资源池的主机信息

操作场景

主机信息用于配置主机的IP与域名的映射关系，在作业配置时只需使用配置的域名即可访问对应的主机。在跨源连接创建完成后，支持修改主机信息。

常见的访问MRS的HBase集群时需要配置实例的主机名（即域名）与主机对应的IP地址。

约束限制

已获取MRS主机信息。请参考[怎样获取MRS主机信息?](#)

修改主机信息

1. 登录DLI管理控制台。
2. 在左侧导航栏中，选择“跨源管理 > 增强型跨源”。
3. 选择待修改的增强型跨源连接，单击操作列的“更多 > 修改主机信息”。
4. 在修改主机信息对话框中，填写已获取的主机信息。

主机信息格式：hostIP hostName。多个主机信息以换行分隔。

样例：

```
192.168.0.22 node-masterxxx1.com
192.168.0.23 node-masterxxx2.com
```

获取MRS主机信息请参考[怎样获取MRS主机信息?](#)
5. 单击“确定”，完成主机信息的修改。

怎样获取 MRS 主机信息？

- **方法一：在管理控制台查看MRS主机信息**

获取MRS集群主机名与IP地址，以MRS3.x集群为例，步骤如下：

- 登录MRS管理控制台页面。
- 单击“集群列表 > 现有集群”，在集群列表中单击指定的集群名称，进入集群信息页面。
- 选择“组件管理”页签；
- 单击进入“Zookeeper”服务；
- 选择“实例”页签，可以查看对应业务IP，可选择任意一个业务IP。
- 参考[修改主机信息](#)修改主机信息。

📖 说明

如果MapReduce服务集群存在多个IP，创建跨源连接时填写其中任意一个业务IP即可。

- **方法二：通过MRS节点的“/etc/hosts”信息获取MRS主机信息**

- 以root用户登录MRS的任意一个主机节点。
- 执行以下命令获取MRS对应主机节点的hosts信息，复制保存。

```
cat /etc/hosts
```

图 5-64 获取 hosts 信息

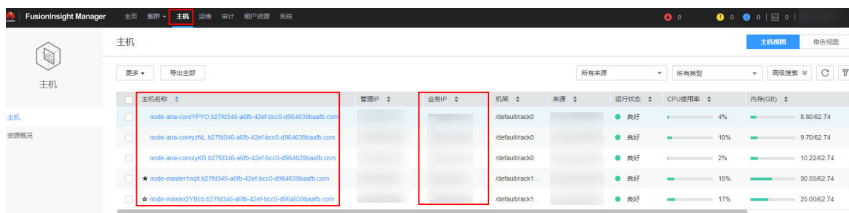
```
[root@node-master1nqtl ~]# cat /etc/hosts
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
10.10.10.10 hadoop.hadoop.com
10.10.10.10 manager
192.168.0.187 node-master1nqtl.b27fd346-a6fb-42ef-bcc0-d964639baafb.com node-master1nqtl.b27fd346-a6fb-42ef-bcc0-d964639baafb.com
192.168.0.198 node-master2Y8cb.b27fd346-a6fb-42ef-bcc0-d964639baafb.com node-master2Y8cb.b27fd346-a6fb-42ef-bcc0-d964639baafb.com
192.168.0.188 node-ana-coreznl.b27fd346-a6fb-42ef-bcc0-d964639baafb.com node-ana-coreznl.b27fd346-a6fb-42ef-bcc0-d964639baafb.com
192.168.0.136 node-ana-corezykb.b27fd346-a6fb-42ef-bcc0-d964639baafb.com node-ana-corezykb.b27fd346-a6fb-42ef-bcc0-d964639baafb.com
192.168.0.174 node-ana-coreypv0.b27fd346-a6fb-42ef-bcc0-d964639baafb.com node-ana-coreypv0.b27fd346-a6fb-42ef-bcc0-d964639baafb.com
[root@node-master1nqtl ~]#
[root@node-master1nqtl ~]#
[root@node-master1nqtl ~]#
```

- 参考[修改主机信息](#)修改主机信息。

- **方法三：登录MRS的FusionInsight Manager获取主机信息**

- 登录MRS的FusionInsight Manager界面。
- 在FusionInsight Manager界面，单击“主机”。在主机页面，分别获取MRS的“主机名称”和“业务IP”。

图 5-65 FusionInsight Manager



- 参考[修改主机信息](#)修改主机信息。

5.3.5.8 增强型跨源连接标签管理

操作场景

标签是用户自定义的、用于标识云资源的键值对，它可以帮助用户对云资源进行分类和搜索。标签由标签“键”和标签“值”组成。

如果用户在其他云服务中使用了标签，建议用户为同一个业务所使用的云资源创建相同的标签键值对以保持一致性。

如您的组织已经设定DLI的相关标签策略，则需按照标签策略规则为资源添加标签。标签不符合标签策略的规则，则可能会导致资源创建失败，请联系组织管理员了解标签策略详情。

DLI支持以下两类标签：

- 资源标签：在DLI中创建的非全局的标签。
- 预定义标签：在标签管理服务（简称TMS）中创建的预定义标签，属于全局标签。

有关预定义标签的更多信息，请参见《[标签管理服务用户指南](#)》。

以下介绍如何为跨源连接添加标签、修改标签和删除标签。

操作步骤

1. 在DLI管理控制台的左侧导航栏中，单击“跨源管理”，选择“增强型跨源”页签。
2. 在对应连接的“操作”列，选择“更多”>“标签”。
3. 进入标签管理页面，显示当前连接的标签信息。
4. 单击“添加/编辑标签”，弹出“添加/编辑标签”对话框，配置参数。标签键和标签值设置完成后，单击“添加”，将标签加入到输入框中。

图 5-66 添加标签

表 5-42 标签配置参数

参数	参数说明
标签键	<p>您可以选择：</p> <ul style="list-style-type: none">在输入框的下拉列表中选择预定义标签键。 如果添加预定义标签，用户需要预先在标签管理服务中创建好预定义标签，然后在“标签键”的下拉框中进行选择。用户可以通过单击“查看预定义标签”进入标签管理服务的“预定义标签”页面，然后单击“创建标签”来创建新的预定义标签。 具体请参见《标签管理服务用户指南》中的“创建预定义标签”章节。在输入框中输入标签键名称。 <p>说明 标签的键的最大长度为128个字符，标签的键可以包含任意语种字母、数字、空格和_ . : + - @ ,但首尾不能含有空格，不能以_sys_开头。</p>
标签值	<p>您可以选择：</p> <ul style="list-style-type: none">在输入框的下拉列表中选择预定义标签值。在输入框中输入标签值。 <p>说明 标签值的最大长度为255个字符，标签的值可以包含任意语种字母、数字、空格和_ . : + - @ 。</p>

- 单击“确定”。
- （可选）在标签列表中，单击“操作”列中“删除”可对选中的标签进行删除。

5.3.5.9 删除增强型跨源连接

操作场景

本节操作介绍在控制台删除不再使用的增强型跨源连接的操作步骤。

操作步骤

- 登录DLI管理控制台。
- 在左侧导航栏中，选择“跨源管理 > 增强型跨源”。
- 选择待删除的增强型跨源连接，单击操作列的“删除”。
- 单击“是”，删除增强型跨源连接。

5.3.6 典型场景示例：配置 DLI 与内网数据源的网络联通

背景信息

DLI与内网数据源的网络联通通常指的是华为云内部服务的场景，例如DLI连接MRS、RDS、CSS、Kafka、DWS时，需要打通DLI和外部数据源之间的网络。使用DLI提供的增强型跨源连接，采用对等连接的方式打通DLI与目的数据源的VPC网络，实现数据互通。

本节操作介绍适用增强型跨源连接配置DLI与内网数据源的网络联通的操作指导。

创建增强型跨源连接网络不通的问题，可以根据本指导的整体流程和步骤进行排查验证。

整体流程

图 5-67 增强型跨源连接配置流程



前提条件

- 已创建DLI队列。创建队列详见[创建DLI队列操作指导](#)。

⚠ 注意

队列的计费类型必须为：“包年/包月”，“按需计费”（按需计费需勾选“专属资源模式”。）

仅“包年/包月”资源、“专属资源模式”的“按需计费”资源才能创建增强型跨源连接。

- 已创建对应的外部数据源集群。具体对接的外部数据源根据业务自行选择。

表 5-43 创建各外部数据源参考

服务名	参考文档链接
RDS	购买RDS for MySQL实例
DWS	创建DWS集群
DMS Kafka	创建Kafka实例 注意 创建DMS Kafka实例时，不能开启Kafka SASL_SSL。
CSS	创建CSS集群
MRS	创建MRS集群

⚠ 注意

- 绑定跨源的DLI队列网段和其他数据源子网网段不能重合。
- 系统default队列不支持创建跨源连接。


步骤 1：获取外部数据源的内网 IP、端口和安全组

表 5-44 各数据源信息获取

数据源	参数获取
DMS Kafka	<ol style="list-style-type: none"> 1. 在Kafka管理控制台，选择“Kafka专享版”，单击对应的Kafka名称，进入到Kafka的基本信息页面。 2. 在“连接信息”中获取该Kafka的“内网连接地址”，在“网络”中获取该实例的“虚拟私有云”和“子网”信息。 3. Kafka的基本信息页面，“网络 > 安全组”参数下获取Kafka的安全组。
RDS	在RDS控制台“实例管理”页面，单击对应实例名称，查看“连接信息”，获取“内网地址”、“虚拟私有云”、“子网”、“数据库端口”和“安全组”信息。
CSS	<ol style="list-style-type: none"> 1. 在CSS管理控制台，选择“Elasticsearch > 集群管理”，单击已创建的CSS集群名称，进入到CSS的基本信息页面。 2. 在“基本信息”中获取CSS的“内网访问地址”、“虚拟私有云”、“子网”和“安全组”信息，方便后续操作步骤使用。
DWS	<ol style="list-style-type: none"> 1. 在DWS管理控制台，选择“集群管理”，单击已创建的DWS集群名称，进入到DWS的基本信息页面。 2. 在“基本信息”的“数据库属性”中获取该实例的“内网IP”、“端口”，在“网络”中获取“虚拟私有云”、“子网”和“安全组”信息，方便后续操作步骤使用。

数据源	参数获取
MRS HBase	<p>以MRS 3.x版本集群为例。</p> <ol style="list-style-type: none"> 1. 登录MRS管理控制台，单击“集群列表 > 现有集群”，单击对应的集群名称，进入到集群概览页面。 2. 在集群概览页面“基本信息”中获取“虚拟私有云”、“子网”和“安全组”。 3. 因为在创建连接MRS HBase的作业时，需要用到MRS集群的ZooKeeper实例和端口，则还需要获取MRS集群主机节点信息。 <ol style="list-style-type: none"> a. 参考访问MRS Manager登录MRS Manager，在MRS Manager上，选择“集群 > 待操作的集群名称 > 服务 > ZooKeeper > 实例”，根据“主机名称”和“业务IP”获取ZooKeeper的主机信息。 b. 在MRS Manager上，选择“集群 > 待操作的集群名称 > 服务 > ZooKeeper > 配置 > 全部配置”，搜索参数“clientPort”，获取“clientPort”的参数值即为ZooKeeper的端口。 c. 使用root用户ssh登录任意一个MRS主机节点。具体请参考登录MRS集群节点。 d. 执行以下命令获取MRS对应主机节点的hosts信息，复制保存。 cat /etc/hosts 例如，查询结果参考如下，将内容复制保存，以备后续步骤使用。 <pre> [root@node-master1k0no ~]# cat /etc/hosts ::1 localhost localhost.localdomain localhost6 localhost6.localdomain6 127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4 10.10.10.10 hadoop.hadoop.com 10.10.10.10 manager 192.168.0.22 node-master3tVbG.mrs-v08w.com node-master3tVbG.mrs-v08w.com. 192.168.0.238 node-group-1ySw0.mrs-v08w.com node-group-1ySw0.mrs-v08w.com. 192.168.0.123 node-master1k0no.mrs-v08w.com node-master1k0no.mrs-v08w.com. 192.168.0.154 node-group-1zKgA.mrs-v08w.com node-group-1zKgA.mrs-v08w.com. 192.168.0.71 node-master2qLhC.mrs-v08w.com node-master2qLhC.mrs-v08w.com. 192.168.0.7 node-group-1yRpv.mrs-v08w.com node-group-1yRpv.mrs-v08w.com. [root@node-master1k0no ~]# </pre>

步骤 2：获取 DLI 队列网段

在DLI管理控制台，单击“资源管理 > 队列管理”，选择运行作业的队列，单击队列名称旁的  按钮，获取队列的网段信息。

步骤 3：外部数据源的安全组添加放通 DLI 队列网段的规则

1. 登录VPC控制台。
2. 在左侧导航树选择“访问控制 > 安全组”。
3. 单击外部数据源所属的安全组名称，进入安全组详情界面。
您可以在对应数据源的管理控制台，参考[步骤1：获取外部数据源的内网IP、端口和安全组](#)获取对应数据源的安全组名称。
4. 在“入方向规则”页签中添加放通队列网段的规则。如[图5-68](#)所示。
详细的入方向规则参数说明请参考[表5-45](#)。

图 5-68 添加入方向规则



表 5-45 入方向规则参数说明

参数	说明	取值样例
优先级	安全组规则优先级。 优先级可选范围为1-100，默认值为1，即最高优先级。优先级数字越小，规则优先级级别越高。	1
策略	安全组规则策略。	允许
协议端口	<ul style="list-style-type: none"> 网络协议。目前支持“All”、“TCP”、“UDP”、“ICMP”和“GRE”等协议。 端口：允许远端地址访问指定端口，取值范围为：1~65535。 	本例中选择TCP协议，端口值不填或者填写为 步骤1：获取外部数据源的内网IP、端口和安全组获取的数据源的端口 。
类型	IP地址类型。	IPv4
源地址	源地址用于放通来自IP地址或另一安全组内的实例的访问。	本例填写 步骤2：获取DLI队列网段 获取的队列网段。
描述	安全组规则的描述信息，非必填项。	-

步骤 4：创建增强型跨源连接

1. 登录DLI管理控制台，在左侧导航栏单击“跨源管理”，在跨源管理界面，单击“增强型跨源”，单击“创建”。
2. 在增强型跨源创建界面，配置具体的跨源连接参数。具体参考如下。
 - 连接名称：设置具体的增强型跨源名称。
 - 弹性资源池：选择DLI的队列。（未添加至资源池的队列，请直接选择队列名称。）

- 虚拟私有云：选择**步骤1：获取外部数据源的内网IP、端口和安全组**获取的外部数据源的虚拟私有云。
 - 子网：选择**步骤1：获取外部数据源的内网IP、端口和安全组**获取的外部数据源子网。
 - 其他参数可以根据需要选择配置。
3. 参数配置完成后，单击“确定”完成增强型跨源配置。单击创建的跨源连接名称，查看跨源连接的状态，等待连接状态为：“已激活”后可以继续进行后续步骤。
 4. 如果是连接MRS HBase，则还需要添加MRS的主机节点信息，具体步骤如下：
 - a. 在“跨源管理 > 增强型跨源”中，在已创建的增强型跨源连接的“操作”列，单击“更多 > 修改主机信息”。
 - b. 在“主机信息”参数中，将**步骤1：获取外部数据源的内网IP、端口和安全组**中获取到的MRS HBase主机节点信息复制追加进去。

图 5-69 修改主机信息



- c. 单击“确定”完成主机信息添加。

步骤 5：测试网络连通性

1. 单击“队列管理”，选择操作的队列，在操作列，单击“更多 > 测试地址连通性”。
2. 在“测试连通性”界面，根据**步骤1：获取外部数据源的内网IP、端口和安全组**中获取的数据源的IP和端口，地址栏输入“数据源内网IP:数据源端口”，单击“测试”测试DLI到外部数据源网络是否可达。

说明

MRS HBase在测试网络连通性的时候，使用：**ZooKeeperIP地址:ZooKeeper端口**，或者，**ZooKeeper的主机信息:ZooKeeper端口**。

5.3.7 典型场景示例：配置 DLI 与公网网络连通

操作场景

公网数据源指的是可以通过互联网访问的数据源。这些数据源资源有一个公网IP地址，配置DLI与公网网络联通可以实现对这些数据源的访问。

本节提供了详细的操作指导，介绍如何通过设置SNAT规则和配置路由信息，实现DLI服务与公网的网络连接。

操作流程

图 5-70 配置 DLI 队列访问公网流程



步骤 1：创建 VPC

登录虚拟私有云控制台，创建虚拟私有云。创建的VPC供NAT访问公网使用。

创建VPC的具体操作请参考[创建虚拟私有云](#)。

图 5-71 创建 VPC

基本信息

区域

不同区域的云服务产品之间内网互不相通；请就近选择靠近您业务的区域，可减少网络时延，提高访问速度。

名称

IPv4网段 · · · /

建议使用网段:10.0.0.0/8-24 (选择) 172.16.0.0/12-24 (选择) 192.168.0.0/16-24 (选择)

步骤 2：创建弹性资源池和队列

1. 登录DLI管理控制台。
2. 在左侧导航栏单击“资源管理 > 弹性资源池”，可进入弹性资源池管理页面。
3. 在弹性资源池管理界面，单击界面右上角的“购买弹性资源池”。
4. 在“购买弹性资源池”界面，填写具体的弹性资源池参数。

表 5-46 参数说明

参数名称	参数说明
计费模式	选择弹性资源池计费模式。
区域	选择弹性资源池所在区域。
项目	每个区域默认对应一个项目，由系统预置。
名称	弹性资源池名称。
规格	选择弹性资源池规格。
CU范围	弹性资源池最大最小CU范围。

参数名称	参数说明
网段	规划弹性资源池所属的网段。如需使用DLI增强型跨源，弹性资源池网段与数据源网段不能重合。 弹性资源池网段设置后不支持更改。
企业项目	选择对应的企业项目。

5. 参数填写完成后，单击“立即购买”，在界面上确认当前配置是否正确。
6. 单击“提交”完成弹性资源池的创建。
7. 在弹性资源池的列表页，选择要操作的弹性资源池，单击操作列的“添加队列”。
8. 配置队列的基础配置，具体参数信息如下。

表 5-47 弹性资源池添加队列基础配置

参数名称	参数说明
名称	弹性资源池添加的队列名称。
类型	选择创建的队列类型。 <ul style="list-style-type: none"> • 执行SQL作业请选择SQL队列。 • 执行Flink或Spark作业请选择通用队列。
执行引擎	SQL队列可以选择队列引擎为Spark或者HetuEngine。
企业项目	选择对应的企业项目。

9. 单击“下一步”，配置队列的扩缩容策略。
单击“新增”，可以添加不同优先级、时间段、“最小CU”和“最大CU”扩缩容策略。
本例配置的扩缩容策略如图5-72所示。

图 5-72 添加队列时配置扩缩容策略



表 5-48 扩缩容策略参数说明

参数名称	参数说明	配置样例
优先级	当前弹性资源池中的优先级数字越大表示优先级越高。本例设置一条扩缩容策略，默认优先级为1。	1
时间段	首条扩缩容策略是默认策略，不能删除和修改时间段配置。 即设置00-24点的扩缩容策略。	00-24
最小CU	设置扩缩容策略支持的最小CU数。	16
最大CU	当前扩缩容策略支持的最大CU数。	64

10. 单击“确定”完成添加队列配置。

步骤 3：创建专属队列和 VPC 的增强型跨源连接

1. 在DLI管理控制台左侧导航栏中，选择“跨源管理”。
2. 选择“增强型跨源”页签，单击左上角的“创建”按钮。

输入连接名称，选择创建的弹性资源池/队列，虚拟私有云，子网，输入主机信息（可选）。

图 5-73 创建增强型跨源连接

创建连接

增强型跨源会在用户网络中创建对等连接，并配置对等连接需要的路由

★ 连接名称

弹性资源池

★ 虚拟私有云

★ 子网

路由表

主机信息

步骤 4：购买弹性公网 IP

1. 在“弹性公网IP”界面，单击“购买弹性公网IP”。

2. 根据界面提示配置参数。
参数填写说明请参考“[购买弹性公网IP](#)”。

步骤 5：配置 NAT 网关

步骤1 创建NAT网关。

1. 登录控制台，在“服务列表”搜索“NAT网关”，进入网络控制台页面。
2. 单击“购买公网NAT网关”，配置NAT网关的相关信息。
详细请参考《NAT网关用户指南》中“[购买公网NAT网关](#)”。

图 5-74 购买 NAT 网关

高级配置 ▾ 描述 | 标签

3. 配置完成后，单击“立即购买”。

📖 说明

“虚拟私有云”为[步骤1：创建VPC](#)创建的VPC。

步骤2 添加路由。

进入VPC的路由表，配置路由规则。通常NAT创建成功会自动创建到NAT网关的路由。
目的地址为访问的公网IP地址，下一跳为NAT网关。

图 5-75 添加路由

路由

删除 添加路由 复制路由 帮我配置

目的地址	下一跳类型	下一跳
Local	Local	Local
172.16.192.0/18	对等连接	DLI_DATACONNECTION_9d05e866-...
14.38/32	NAT网关	nat-32c8

步骤3 添加SNAT规则。

为新建的NAT网关添加SNAT规则，才能实现该子网下的主机与Internet互相访问。

1. NAT网关购买成功后，在NAT控制台，单击购买成功的NAT网关“名称”，进入NAT网关详情页面。
2. 选择“SNAT规则”页签，单击“添加SNAT规则”。
详细请参考《NAT网关用户指南》中“[添加SNAT规则](#)”。
3. 使用场景选择云专线/云连接。
4. 添加专属队列所在的网段。
5. 绑定对应的弹性公网IP。

图 5-76 添加 SNAT 规则



6. 添加完成后，单击“确定”。

----结束

步骤 6：添加自定义路由

在增强型跨源连接页面添加自定义路由。此处添加的是访问的IP地址的路由信息。
详细操作请参考[自定义路由信息](#)。

图 5-77 增强型跨源链接添加测试路由信息

添加路由

* 路由名称

* 路由IP 14 . . 0 / 24

确定 取消

步骤 7：测试公网连通性

测试队列到公网的连通性。单击队列操作列下方的“更多 > 测试地址连通性”，输入访问的公网IP地址。

图 5-78 测试地址连通性

测试地址连通性

测试队列到指定地址是否可达，支持域名或ip，两种方式均需指定端口。

* 地址

地址14. .38:80可达。

6 配置 DLI 访问其他云服务的委托权限

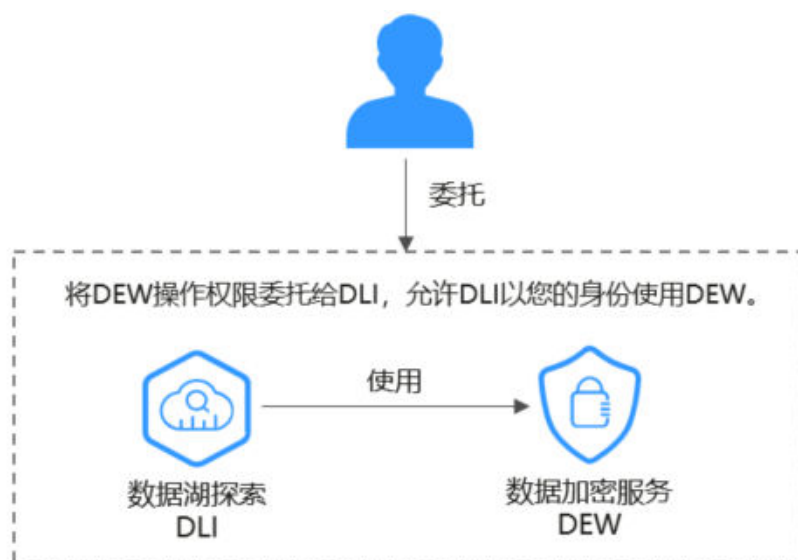
6.1 DLI 委托概述

什么是委托？

各云服务之间存在业务交互关系，一些云服务需要与其他云服务协同工作，需要您创建云服务委托，将操作权限委托给DLI服务，让DLI服务以您的身份使用其他云服务，代替您进行一些资源运维工作。

例如：在DLI新建Flink作业所需的AKSK存储在数据加密服务DEW中，如需允许DLI在执行作业时访问DEW数据，需要提供IAM委托将DEW数据操作权限委托给DLI，允许DLI服务以您的身份访问DEW服务。

图 6-1 DLI 云服务委托



DLI 委托

在使用DLI前，为了确保正常使用DLI的功能，建议先进行DLI委托权限设置。

- DLI默认提供以下类型的委托：`dli_admin_agency`、`dli_management_agency`、`dli_data_clean_agency`（名称固定，权限需自定义）。其他场景需用户自定义委托。委托的详细说明请参考[表6-1](#)。
- DLI为了满足细粒度的委托权限需求，DLI升级了系统委托，将原有的`dli_admin_agency`升级为`dli_management_agency`，新的委托包含跨源操作、消息通知、用户授权操作所需的权限。[配置DLI云服务委托权限](#)。
- 使用Flink 1.15和Spark 3.3.1（Spark通用队列场景）及以上版本的引擎执行作业时，需完成以下配置操作：
需用户自行在IAM页面创建相关委托，并在作业配置中添加新建的委托信息。具体操作请参考[创建DLI自定义委托权限](#)。
 - 常见新建委托场景：允许DLI读写OBS数据、日志转储、Flink checkpoint；允许DLI在访问DEW获取数据访问凭证、允许访问Catalog获取元数据等场景。
 - 委托名称不可与系统默认委托重复，即不可以是`dli_admin_agency`、`dli_management_agency`、`dli_data_clean_agency`。
- 引擎版本低于Flink1.15，执行作业时默认使用`dli_admin_agency`；引擎版本低于Spark 3.3.1，执行作业时使用用户认证信息（AKSK、SecurityToken）。
即引擎版本低于Flink1.15和Spark 3.3.1版本的作业不受更新委托权限的影响，无需自定义委托。
- 为兼容存量的作业委托权限需求，`dli_admin_agency`仍为您保留在IAM委托中。

说明

- 服务授权需要主账号或者用户组admin中的子账号进行操作。
- 请勿删除系统默认创建的委托。

表 6-1 DLI 委托

权限名	类型	权限说明
<code>dli_admin_agency</code>	系统默认委托	该委托已废弃，不推荐使用，请尽快更新委托升级至 <code>dli_management_agency</code> 。 更新委托请参考 配置DLI云服务委托权限 。
<code>dli_management_agency</code>	系统默认委托	DLI系统委托，用于将操作权限委托给DLI服务，让DLI服务以您的身份使用其他云服务，代替您进行一些资源运维工作。该委托包含跨源操作、消息通知、用户授权操作所需的权限。详细委托包含的权限请参考 表 6-2
<code>dli_data_clean_agency</code>	系统默认委托，需用户自行授权	数据清理委托，表生命周期清理数据及lakehouse表数据清理使用。需用户自行在IAM创建名为 <code>dli_data_clean_agency</code> 的DLI云服务委托并授权。 该委托需新建后自定义权限，但委托名称固定为 <code>dli_data_clean_agency</code> 。 委托的权限策略示例请参考 常见场景的委托权限策略 。

权限名	类型	权限说明
其他自定义委托	自定义委托	<p>使用Flink 1.15和Spark 3.3及以上版本的引擎执行作业时，请自行在IAM页面创建相关委托，并在作业配置中添加新建的委托信息。创建DLI自定义委托权限</p> <p>常见新建委托场景：允许DLI读写OBS将日志转储、允许DLI在访问DEW获取数据访问凭证、允许访问Catalog获取元数据等场景。</p> <p>委托名称不可与系统默认委托重复，即不可以是dli_admin_agency、dli_management_agency、dli_data_clean_agency。</p> <p>委托的权限策略示例请参考常见场景的委托权限策略。</p>

表 6-2 dli_management_agency 委托包含的权限

权限名	权限说明
IAM ReadOnlyAccess	DLI对未登录过DLI的用户进行授权时，需获取IAM用户相关信息。因此需要IAM ReadOnlyAccess权限。
DLI Datasource Connections Agency Access	访问和使用VPC、子网、路由、对等连接的权限
DLI Notification Agency Access	作业执行失败需要通过SMN发送通知消息的权限

6.2 创建 DLI 自定义委托权限

使用Flink 1.15和Spark 3.3及以上版本的引擎执行作业时，当您所需的委托没有包含在DLI系统委托dli_management_agency时，您需要在IAM页面创建相关委托，并在作业配置中添加新建的委托信息。dli_management_agency包含跨源操作、消息通知、用户授权操作所需的权限，除此之外的其他委托权限需求，都需自定义DLI委托。了解dli_management_agency请参考[DLI委托概述](#)。

本节操作介绍自定义委托，并完成服务授权，以及在作业配置中添加新建的委托信息的操作步骤。

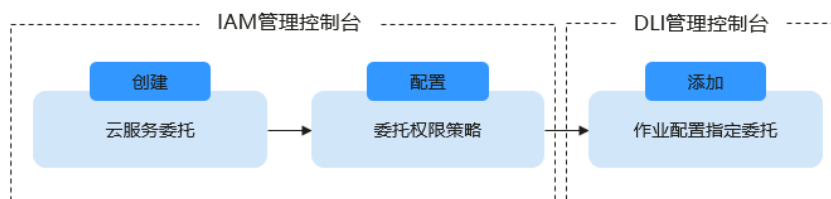
DLI 自定义委托场景

表 6-3 DLI 自定义委托场景

场景	委托名称	适用场景	权限策略
允许DLI按表生命周期清理数据	dli_data_clean_agency	数据清理委托，表生命周期清理数据、Lakehouse表数据清理使用。 该委托需新建后自定义权限，但委托名称固定为dli_data_clean_agency。	数据清理委托权限配置
允许DLI读写OBS将日志转储	自定义	DLI Flink作业下载OBS对象、OBS/DWS数据源（外表）、日志转储、使用savepoint、开启checkpoint，DLI Spark作业下载OBS对象、读写OBS外表。	访问和使用OBS的权限策略
允许DLI在访问DEW获取数据访问凭证	自定义	DLI 作业使用DEW-CSMS凭证管理能力。	使用DEW加密功能的权限
允许访问DLI Catalog元数据	自定义	DLI 访问DLI元数据。	访问DLI Catalog元数据的权限
允许访问LakeFormation Catalog元数据	自定义	DLI 访问LakeFormation元数据。	访问LakeFormation Catalog元数据的权限

操作流程

图 6-2 自定义委托操作流程



约束与限制

- 自定义委托名称不可与系统默认委托重复，即不可以是dli_admin_agency、dli_management_agency、dli_data_clean_agency。
- 允许DLI按表生命周期清理数据的委托名称必须为dli_data_clean_agency。
- 仅Flink 1.15和Spark 3.3.1（Spark通用队列场景）及以上版本的引擎执行作业支持配置自定义委托。

- 更新委托权限后，系统将升级您的dli_admin_agency为dli_management_agency，新的委托包含跨源操作、消息通知、用户授权操作所需的权限。除此之外的其他委托权限需求都需要用户自定义委托。了解dli_management_agency请参考[DLI委托概述](#)。
- 常见新建委托场景：允许DLI读写OBS数据、日志转储、Flink checkpoint；允许DLI在访问DEW获取数据访问凭证、允许访问Catalog获取元数据等场景。以上场景的委托权限请参考[常见场景的委托权限策略](#)。

步骤 1：在 IAM 控制台创建云服务委托并授权

1. 登录管理控制台。
2. 单击右上方登录的用户名，在下拉列表中选择“统一身份认证”。
3. 在左侧导航栏中，单击“委托”。
4. 在“委托”页面，单击“创建委托”。
5. 在“创建委托”页面，设置如下参数：
 - 委托名称：按需填写，例如“dli_obs_agency_access”。
 - 委托类型：选择“云服务”。
 - 云服务：（“委托类型”选择“云服务”时出现此参数项。）在下拉列表中选择“DLI”。
 - 持续时间：选择“永久”。
 - 描述：非必选，可以填写“拥有OBS OperateAccess权限的委托”。

图 6-3 创建委托

* 委托名称

* 委托类型 普通账号
 将账号内资源的操作权限委托给其他华为云账号。
 云服务
 将账号内资源的操作权限委托给华为云服务。

* 云服务

* 持续时间

描述

24/255

6. 配置完委托的基本信息后，单击“下一步”。

7. 授予当前委托所需的权限策略，单击“新建策略”。
8. 配置策略信息。

- a. 输入策略名称，本例：dli-obs-agency。
- b. 选择“JSON视图”。
- c. 在策略内容中粘贴自定义策略。

本例权限包含访问和使用OBS的权限，适用于以下场景：DLI Flink作业下载OBS对象、OBS/DWS数据源（外表）、日志转储、使用savepoint、开启Checkpoint。DLI Spark作业下载OBS对象、读写OBS外表。

更多Flink作业常见委托权限配置请参考[常见场景的委托权限策略](#)。

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "obs:bucket:GetBucketPolicy",
        "obs:bucket:GetLifecycleConfiguration",
        "obs:bucket:GetBucketLocation",
        "obs:bucket:ListBucketMultipartUploads",
        "obs:bucket:GetBucketLogging",
        "obs:object:GetObjectVersion",
        "obs:bucket:GetBucketStorage",
        "obs:bucket:GetBucketVersioning",
        "obs:object:GetObject",
        "obs:object:GetObjectVersionAcl",
        "obs:object:DeleteObject",
        "obs:object:ListMultipartUploadParts",
        "obs:bucket:HeadBucket",
        "obs:bucket:GetBucketAcl",
        "obs:bucket:GetBucketStoragePolicy",
        "obs:object:AbortMultipartUpload",
        "obs:object:DeleteObjectVersion",
        "obs:object:GetObjectAcl",
        "obs:bucket:ListBucketVersions",
        "obs:bucket:ListBucket",
        "obs:object:PutObject"
      ],
      "Resource": [
        "OBS:*:bucket:bucketName", //请替换bucketName为对应的桶名称
        "OBS:*:object:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "obs:bucket:ListAllMyBuckets"
      ]
    }
  ]
}
```

- d. 按需输入策略描述。
9. 新建策略完成后，单击“下一步”，返回委托授权页面。
10. 选择步骤8新建的自定义策略。

图 6-4 选择自定义策略



11. 单击“下一步”，选择委托的授权范围。

了解更多授权操作说明请参考[创建用户组并授权](#)。

- 所有资源：授权后，IAM用户可以根据权限使用账号中所有资源，包括企业项目、区域项目和全局服务资源。
- 全局服务资源：全局服务部署时不区分区域，访问全局级服务，不需要切换区域，全局服务不支持基于区域项目授权。如对象存储服务（OBS）、内容分发网络（CDN）等。授权后，用户根据权限使用全局服务的资源。
- 指定区域项目资源：授权后，IAM用户根据权限使用所选区域项目中的资源，未选择的区域项目中的资源，该IAM用户将无权访问。
- 指定企业项目资源：授权后，IAM用户根据权限使用所选企业项目中的资源。如企业项目A包含资源B，资源B部署在北京四和上海二，IAM用户所在用户组关联企业项目A后，北京四和上海二的资源B用户都可访问，不在企业项目A内的其他资源，该IAM用户将无权访问。

本例自定义策略中是OBS权限，因此选择全局服务资源。如果使用的是DLI权限，推荐选择“指定区域项目资源”。

12. 单击“确定”，完成授权。

授权后需等待15-30分钟才可生效。

步骤 2：在作业中设置委托权限

使用Flink 1.15和Spark 3.3及以上版本的引擎执行作业时，需要在作业配置中添加新建的委托信息。

否则Spark3.3.1作业不指定委托时，无法使用OBS；Flink1.15作业不指定委托时，无法开启checkpoint、savepoint，作业提交日志无法转储，无法使用OBS、DWS等数据源。

注意

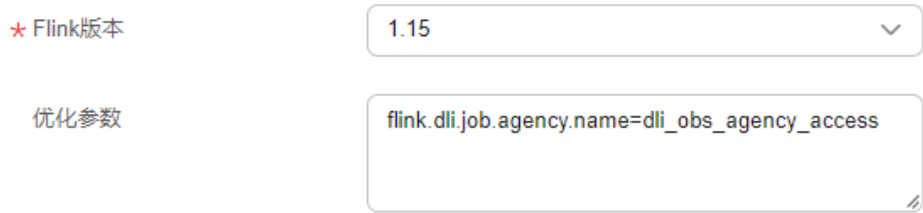
- 仅有运行在弹性资源池队列上的Flink 1.15和Spark3.3.1作业支持指定委托。
- 作业指定委托后，授予委托的权限要谨慎修改，委托权限变动可能会影响作业的正常运行。

• Flink Jar作业指定委托

- a. 登录DLI管理控制台，选择“作业管理 > Flink作业”。
- b. 选择待编辑的Flink Jar作业，单击操作列的“编辑”。
- c. 在作业配置区域配置委托信息：

- Flink版本：选择1.15。
- 优化参数：配置新建的委托key-value信息，key固定为“flink.dli.job.agency.name”，value为自定义的委托名。
本例配置为：flink.dli.job.agency.name=dli_obs_agency_access

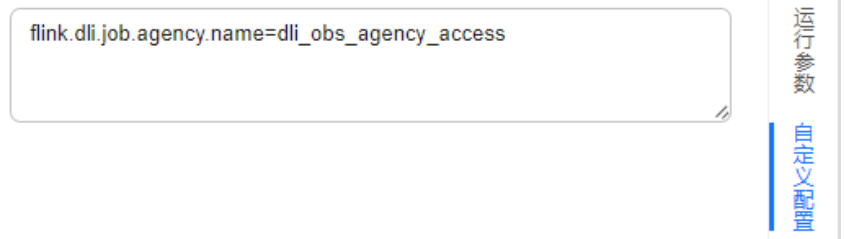
图 6-5 Flink Jar 作业指定委托



- **Flink OpenSource SQL作业指定委托**

- 登录DLI管理控制台，选择“作业管理 > Flink作业”。
- 选择待编辑的Flink OpenSource SQL作业，单击操作列的“编辑”。
- 在作业配置区域配置委托信息：
 - 在“运行参数”页签，确保所选的Flink版本为1.15。
 - 单击“自定义配置”，配置新建的委托key-value信息，key固定为“flink.dli.job.agency.name”，value为自定义的委托名。
本例配置为：flink.dli.job.agency.name=dli_obs_agency_access

图 6-6 Flink OpenSource SQL 作业指定委托



- **Spark作业指定委托**

- 登录DLI管理控制台，选择“作业管理 > Spark作业”。
- 选择待编辑的Spark作业，单击操作列的“编辑”。
- 在作业配置区域配置委托信息：
 - 确保所选的Spark版本为3.3.1。
 - 在“Spark参数（--conf）”区域配置新建的委托key-value信息，key固定为“spark.dli.job.agency.name”，value为自定义的委托名。
本例配置为：spark.dli.job.agency.name=dli_obs_agency_access

图 6-7 Spark 作业指定委托

选择运行队列

* 队列

* Spark版本 3.3.1

作业配置

作业名称 (--name)

* 应用程序

主类 (--class)

应用程序参数

Spark参数 (--conf) spark.dli.job.agency.name=dli_obs_agency_access

6.3 常见场景的委托权限策略

本节操作提供了DLI常见场景的委托权限策略，用于用户自定义权限时配置委托的权限策略。委托策略中的“Resource”根据需要进行替换。

数据清理委托权限配置

适用场景：数据清理委托，表生命周期清理数据及lakehouse表数据清理使用。该委托需新建后自定义权限，但委托名称固定为dli_data_clean_agency。

说明

请在设置委托的授权范围时分别对OBS权限和DLI权限授权范围：

- OBS权限请选择“全局服务资源”
- DLI权限选择“指定区域项目资源”

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "obs:object:GetObject",
        "obs:object:DeleteObject",
        "obs:bucket:HeadBucket",
        "obs:bucket:ListBucket",
        "obs:object:PutObject"
      ]
    }
  ]
}
```

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dli:table:showPartitions",

```

```
        "dli:table:select",
        "dli:table:dropTable",
        "dli:table:alterTableDropPartition"
    ]
}
]
```

访问和使用 OBS 的权限策略

适用场景：DLI Flink作业下载OBS对象、OBS/DWS数据源（外表）、日志转储、使用savepoint、开启checkpoint，DLI Spark作业下载OBS对象、读写OBS外表。

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "obs:bucket:GetBucketPolicy",
        "obs:bucket:GetLifecycleConfiguration",
        "obs:bucket:GetBucketLocation",
        "obs:bucket:ListBucketMultipartUploads",
        "obs:bucket:GetBucketLogging",
        "obs:object:GetObjectVersion",
        "obs:bucket:GetBucketStorage",
        "obs:bucket:GetBucketVersioning",
        "obs:object:GetObject",
        "obs:object:GetObjectVersionAcl",
        "obs:object:DeleteObject",
        "obs:object:ListMultipartUploadParts",
        "obs:bucket:HeadBucket",
        "obs:bucket:GetBucketAcl",
        "obs:bucket:GetBucketStoragePolicy",
        "obs:object:AbortMultipartUpload",
        "obs:object:DeleteObjectVersion",
        "obs:object:GetObjectAcl",
        "obs:bucket:ListBucketVersions",
        "obs:bucket:ListBucket",
        "obs:object:PutObject"
      ],
      "Resource": [
        "OBS:*:bucket:bucketName",//请替换bucketName为对应的桶名称
        "OBS:*:object:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "obs:bucket:ListAllMyBuckets"
      ]
    }
  ]
}
```

使用 DEW 加密功能的权限

适用场景：DLI Flink、Spark作业场景使用DEW-CSMS凭证管理能力。

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "csms:secretVersion:get",
        "csms:secretVersion:list",

```

```
    "kms:dek:decrypt"  
  ]  
}  
]  
}
```

访问 DLI Catalog 元数据的权限

适用场景：DLI Flink、Spark作业场景，授权DLI访问DLI元数据。

```
{  
  "Version": "1.1",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "dli:table:showPartitions",  
        "dli:table:alterTableAddPartition",  
        "dli:table:alterTableAddColumns",  
        "dli:table:alterTableRenamePartition",  
        "dli:table:delete",  
        "dli:column:select",  
        "dli:database:dropFunction",  
        "dli:table:insertOverwriteTable",  
        "dli:table:describeTable",  
        "dli:database:explain",  
        "dli:table:insertIntoTable",  
        "dli:database:createDatabase",  
        "dli:table:alterView",  
        "dli:table:showCreateTable",  
        "dli:table:alterTableRename",  
        "dli:table:compaction",  
        "dli:database:displayAllDatabases",  
        "dli:database:dropDatabase",  
        "dli:table:truncateTable",  
        "dli:table:select",  
        "dli:table:alterTableDropColumns",  
        "dli:table:alterTableSetProperties",  
        "dli:database:displayAllTables",  
        "dli:database:createFunction",  
        "dli:table:alterTableChangeColumn",  
        "dli:database:describeFunction",  
        "dli:table:showSegments",  
        "dli:database:createView",  
        "dli:database:createTable",  
        "dli:table:showTableProperties",  
        "dli:database:showFunctions",  
        "dli:database:displayDatabase",  
        "dli:table:alterTableRecoverPartition",  
        "dli:table:dropTable",  
        "dli:table:update",  
        "dli:table:alterTableDropPartition"  
      ]  
    }  
  ]  
}
```

访问 LakeFormation Catalog 元数据的权限

适用场景：DLI Spark作业场景，授权DLI访问LakeFormation Catalog元数据。

```
{  
  "Version": "1.1",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "lakeformation:table:drop",  
      ]  
    }  
  ]  
}
```

```

"lakeformation:table:create",
"lakeformation:policy:create",
"lakeformation:database:create",
"lakeformation:database:drop",
"lakeformation:database:describe",
"lakeformation:catalog:alter",
"lakeformation:table:alter",
"lakeformation:database:alter",
"lakeformation:catalog:create",
"lakeformation:function:describe",
"lakeformation:catalog:describe",
"lakeformation:function:create",
"lakeformation:table:describe",
"lakeformation:function:drop",
"lakeformation:transaction:operate"
    ]
  }
}
}

```

6.4 典型场景 DLI 委托权限配置示例

表 6-4 DLI 委托权限配置场景开发指南

类型	操作指导	说明
Flink作业场景	Flink Opensource SQL使用DEW管理访问凭据	Flink Opensource SQL场景使用DEW管理和访问凭据的操作指导，将Flink作业的输出数据写入到Mysql或DWS时，在connector中设置账号、密码等属性。
	Flink Jar 使用DEW获取访问凭证读写OBS	访问OBS的AKSK为例介绍Flink Jar使用DEW获取访问凭证读写OBS的操作指导。
	用户获取Flink作业委托临时凭证	DLI提供了一个通用接口，可用于获取用户在启动Flink作业时设置的委托的临时凭证。该接口将获取到的该作业委托的临时凭证封装到com.huaweicloud.sdk.core.auth.BasicCredentials类中。 本操作介绍获取Flink作业委托临时凭证的操作方法。
Spark作业场景	Spark Jar 使用DEW获取访问凭证读写OBS	访问OBS的AKSK为例介绍Spark Jar使用DEW获取访问凭证读写OBS的操作指导。
	用户获取Spark作业委托临时凭证	本操作介绍获取Spark Jar作业委托临时凭证的操作方法。

7 使用 DLI 提交 SQL 作业

7.1 创建并提交 SQL 作业

SQL 编辑器简介

SQL作业编辑器支持使用SQL语句执行数据查询操作。

支持SQL2003，兼容SparkSQL，详细语法描述请参见《[数据湖探索SQL语法参考](#)》。

在总览页面，单击左侧导航栏中的“SQL编辑器”或SQL作业简介右上角的“创建作业”，可进入“SQL编辑器”页面。

本节内容介绍使用DLI的SQL编辑器创建并提交SQL作业。

使用须知

- 首次进入SQL编辑器，页面会提示设置DLI作业桶。该桶用于存储使用DLI服务产生的临时数据，例如：作业日志等。
如果不创建该桶，将无法查看作业日志。桶名称为系统默认。
在OBS管理控制台页面通过配置桶的生命周期规则，可以实现定时删除OBS桶中的对象或者定时转换对象的存储类别。
具体操作请参考通过[配置生命周期规则](#)。
- SQL编辑器页面可以批量执行SQL语句。
- 作业编辑窗口常用语法采用不同颜色突出显示。
- 支持单行注释和多行注释。以“--”开头，后续内容即为注释。

使用 SQL 编辑器创建并提交 SQL 作业

1. 登录DLI管理控制台，选择“SQL编辑器”页面。

说明

进入“SQL编辑器”页面后，系统会提示“创建DLI临时数据桶”，用于存储使用DLI服务产生的临时数据。在“设置DLI作业桶”对话框中，单击“去设置”。在现实页面上单击DLI作业桶卡片右上角单击编辑符号。在弹出的“设置DLI作业桶”对话框，输入作业桶路径，并单击“确定”。

- 在SQL作业编辑窗口右上方的依次选择执行SQL作业所需的队列、数据库等信息，详细参数说明请参考[表7-1](#)。

表 7-1 配置 SQL 作业信息

按键&下拉列	描述
执行引擎	SQL作业支持Spark和HetuEngine两种引擎： <ul style="list-style-type: none"> • Spark引擎适用于离线分析。 • HetuEngine引擎适用于交互式分析。
队列	下拉选择需要使用的队列。如果没有可用队列，此处显示“default”队列，default队列为体验使用，建议重新创建队列。具体队列创建可以参考 创建弹性资源池并添加队列 。 SQL作业只能在队列类型为“SQL队列”下执行。
数据目录	数据目录（Catalog）是元数据管理对象，它可以包含多个数据库。 您可以在DLI中创建并管理多个Catalog，用于不同的元数据隔离。
数据库	下拉选择需要使用的数据库。如果没有可用数据库，此处显示“default”默认数据库。数据库创建操作详见在 DLI控制台创建数据库和表 。 说明 如果SQL语句中指定了表所在的数据库，则此处选择的数据库无效。
设置	包括设置“参数设置”和“标签”。 参数设置：以“key/value”的形式设置提交SQL作业的配置项。详细内容请参见《 数据湖探索SQL语法参考 》。 标签：以“key/value”的形式设置SQL作业的标签。




- 创建数据库和表。
您可以参考在[DLI控制台创建数据库和表](#)提前创建数据库和表。例如本例创建表，表名为“qw”。
- 在SQL作业编辑窗口输入表“qw”的SQL查询语句：

```
SELECT * FROM qw.qw LIMIT 10;
```

 或者双击左侧表名“qw”，上述查询语句会自动在作业编辑窗口中输入。
DLI还为您提供了丰富的SQL模板，每种模板都为您提供使用了使用场景、代码示例和使用指导。您也可以直接使用SQL作业模板快速实现您的业务逻辑。了解模板更多信息请参考[创建SQL作业模板](#)。
- 单击“更多”中的“语法校验”，确认SQL语句书写是否正确。
 - 如果语法校验失败，请参考《[数据湖探索SQL语法参考](#)》检查SQL语句准确性。
 - 如果语法校验通过，单击“执行”，阅读并同意隐私协议，单击“确定”后执行SQL语句。

SQL语句执行成功后，在SQL作业编辑窗口下方会显示执行结果。

6. 查看作业执行结果。

在查看结果页签，单击  以图形形式呈现查询结果。再单击  切换回表格形式。当前控制台界面查询结果最多显示1000条数据，如果需要查看更多或者全量数据，则可以单击  将数据导出到OBS获取。

 说明

- 如果执行结果中无数值列，则无法进行图形化。
- 图形类型包括柱状图、折线图、扇形图。
- 柱状图和折线图的X轴可为任意一列，Y轴仅支持数值类型的列，扇形图对应图例和指标。

SQL 编辑器功能概览

- **SQL作业参数设置**

单击SQL编辑器页面右上方的“设置”按钮。可以设置SQL作业运行参数和作业标签。

- 参数设置：以“key/value”的形式设置提交SQL作业的配置项。
详细内容请参见《[数据湖探索SQL语法参考](#)》。
- 标签：以“key/value”的形式设置SQL作业的标签。

表 7-2 SQL 作业运行参数配置说明

参数名称	默认值	描述
spark.sql.files.maxRecordsPerFile	0	要写入单个文件的最大记录数。如果该值为零或为负，则没有限制。
spark.sql.autoBroadcastJoinThreshold	209715200	配置执行连接时显示所有工作节点的表的最大字节大小。通过将此值设置为“-1”，可以禁用显示。 说明 当前仅支持运行命令ANALYZE TABLE COMPUTE statistics noscan的配置单元存储表，和直接根据数据文件计算统计信息的基于文件的数据源表。
spark.sql.shuffle.partitions	200	为连接或聚合过滤数据时使用的默认分区数。
spark.sql.dynamicPartitionOverwrite.enabled	false	当前配置设置为“false”时，DLI在覆盖写之前，会删除所有符合条件的分区。例如，分区表中有一个“2021-01”的分区，当使用INSERT OVERWRITE语句向表中写入“2021-02”这个分区的数据时，会把“2021-01”的分区数据也覆盖掉。 当前配置设置为“true”时，DLI不会提前删除分区，而是在运行时覆盖那些有数据写入的分区。

参数名称	默认值	描述
spark.sql.files.maxPartitionBytes	134217728	读取文件时要打包到单个分区中的最大字节数。
spark.sql.badRecordsPath	-	Bad Records的路径。
dli.sql.sqlasync.enabled	true	DDL和DCL语句是否异步执行，值为“true”时启用异步执行。
dli.sql.job.timeout	-	设置作业运行超时时间，超时取消。单位：秒。

- **跳转至SparkUI查看SQL语句执行进程**

SQL编辑器页面提供了跳转至SparkUI查看SQL语句执行进程的功能。

- 目前DLI配置SparkUI只展示最新的100条作业信息。
- default队列下运行的作业或者该作业为同步作业时不支持跳转至SparkUI查看SQL语句执行进程。

 **说明**

新建队列，运行作业时会重新拉集群，大概需要10分钟左右才能拉好集群，在集群创建好之前单击SparkUI会导致缓存空的projectID，从而导致无法查看SparkUI。建议使用专属队列，集群不会被释放，就不会有该问题，或者提交作业后等一段时间再查看SparkUI，确保集群已经拉好了，不要立即单击SparkUI。

- **归档SQL运行日志**

单击SQL编辑器页面，执行历史中SQL作业操作列下的“更多 > 归档日志”，系统自动跳转至日志存储的OBS路径。您可以按需下载对应的日志使用。

 **说明**

default队列下运行的作业或者该作业为同步作业时不支持归档日志操作。

- **SQL编辑器快捷功能**

表 7-3 快捷键说明

快捷键	描述
Ctrl+Enter	执行SQL。通过按下键盘上的Ctrl+R或Ctrl + Enter，您可以执行SQL语句。
Ctrl+F	搜索SQL。通过按下键盘上的Ctrl + F，您可以搜索需要的SQL语句。
Shift+Alt+F	格式化SQL。通过按下键盘上的Shift+Alt+F，您可以将SQL语句格式化。
Ctrl+Q	语法校验。通过按下键盘上的Ctrl + Q，您可以对SQL语句进行语法校验。
F11	全屏。通过按下键盘上的F11，您可将SQL作业编辑器窗口全屏。再次按下F11，将从全屏复原。

7.2 导出 SQL 作业结果

导出作业结果是将SQL作业分析后的数据结果按指定格式存储到指定位置。

DLI默认将SQL作业结果存储在DLI作业桶中。同时也支持下载作业结果到本地或导出作业结果到指定的OBS桶。

导出作业结果到 DLI 作业桶

DLI在指定了一个默认的OBS桶作为作业结果的存储位置，请在DLI管理控制台的“全局配置 > 工程配置”中配置桶信息。当作业完成后，系统会自动将结果存储到这个默认桶中。

使用DLI作业桶读取查询结果，需具备以下条件：

- 在DLI管理控制台“全局配置 > 工程配置”中完成作业桶的配置。作业桶配置请参考[配置DLI作业桶](#)。
- 提交工单申请开启查询结果写入桶特性的白名单。
- 确保执行作业的用户具备该作业桶的读写权限，或授予作业桶“jobs/result”路径的读写权限。

详细操作请参考[自定义创建桶策略](#)。


获取DLI桶中的作业结果请参考《对象存储用户指南》中“对象管理 > [下载](#)”。

导出作业结果到指定桶地址

除了使用默认桶存储作业结果，用户还可以导出作业结果到指定的桶地址，提高作业结果管理的灵活性，便于作业结果的组织和管理。

控制台界面查询结果最多显示1000条数据，如果需要查看更多或者全量数据，则可以通过该功能将数据导出到OBS获取。具体操作步骤如下：

导出查询结果的操作入口有两个，分别在“SQL作业”和“SQL编辑器”页面。

- 在“作业管理” > “SQL作业”页面，可单击对应作业“操作”列“更多 > 导出结果”，可导出执行查询后的结果。
- 在“SQL编辑器”页面，查询语句执行成功后，在“查看结果”页签右侧，单击“导出结果”，可导出执行查询后的结果。

说明

- 如果查询结果中无数值列，则无法导出查询结果。
- 确保执行导出作业结果的用户具备该OBS桶的读写权限。

表 7-4 参数说明

参数	是否必选	说明
数据格式	是	选择导出结果的数据格式，当前支持json和csv格式。

参数	是否必选	说明
队列	是	选择执行导出作业的队列。SQL作业只能在队列类型为“SQL队列”下执行。
压缩格式	否	<p>导出查询结果数据的压缩方式，选择如下压缩方式。</p> <ul style="list-style-type: none"> • none • bzip2 • deflate • gzip
存储路径	是	<p>选择导出的作业结果的OBS桶路径。</p> <ul style="list-style-type: none"> • 如果导出方式选择的是“随导出创建指定路径” 在选择桶路径后，需手动输入自定义的指定路径的目录名称，且该目录名称不存在，否则系统将返回错误的信息，无法执行导出操作。 说明 文件夹名称不能包含下列特殊字符：\ / : * ? " < > ，并且不能以“.”开头和结尾。 例如选择存储路径obs://bucket/src1/后，需手动补充路径名称为obs://bucket/src1/src2/，且确保src1下不存在src2的目录。 那么导出的作业结果的路径为obs://bucket/src1/src2/test.csv • 如果导出方式选择的是“覆盖指定路径” 在选择桶路径后，将作业结果导出至该路径下，如有重名文件将自动覆盖。 例如选择存储路径obs://bucket/src1/ 那么导出的作业结果的路径为obs://bucket/src1/test.csv

参数	是否必选	说明
导出方式	是	<ul style="list-style-type: none"> ● 随导出创建指定路径： 该方式导出作业结果时，会创建一个新的文件夹路径，并将作业结果保存在这个路径中。适用于当您希望新的路径下保存本次的导出结果的场景，方便作业结果的管理的回溯。 选择“随导出创建指定路径”时，请务必在“存储路径”后手动输入指定的导出目录，且该目录必须不存在，如果指定目录已经存在，系统将返回错误信息，无法执行导出操作。 ● 覆盖指定路径：当计划导出某一个结果时，您可以选择一个已有的文件路径作为输出目录，如果这个路径下已有同名文件，将会自动覆盖这个文件，即原有的作业结果会被新导出的作业结果文件所替代。 覆盖指定路径方式适用于在同一个路径下保存唯一的作业结果文件的场景，即不需要旧的作业结果的场景。
结果条数	否	指定导出的结果条数。 不填写数值或数值为“0”时，导出全部结果。
表头	否	设置导出查询结果数据是否含表头。

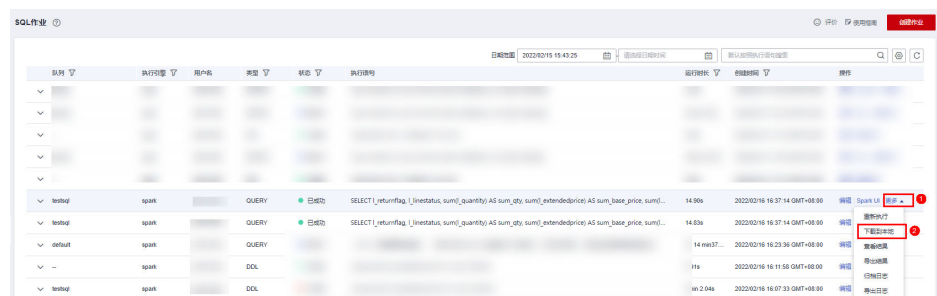
导出作业结果到本地

异步DDL和QUERY语句支持将结果下载到本地。下载到本地默认仅支持下载1000条数据。

具体操作如下：

1. 单击执行成功的异步DDL或QUERY语句“操作”列中的“下载到本地”，在提示窗口单击“确认”。此时，“操作”列中的“下载到本地”将变为“立即下载”。

图 7-1 下载到本地



2. 单击“立即下载”将对应结果下载到本地。

7.3 配置 SQL 防御规则

什么是 SQL 防御

大数据领域的SQL引擎层出不穷，在带给解决方案多样性的同时，也暴露出一定的问题，例如SQL输入语句质量良莠不齐、SQL问题难定位、大SQL语句消耗资源过多等。

低质量的SQL会对数据分析平台系统带来不可预料的冲击，影响系统的性能或者平台稳定性。

DLI在Spark SQL引擎中增加SQL防御能力，基于用户可理解的SQL防御策略，实现对典型大SQL、低质量SQL的主动防御，包括事前提示、拦截和事中熔断，并不强制改变用户的SQL提交方式、SQL语法，对业务零改动且易落地。

- DLI支持可视化配置SQL防御策略，同时可支持防御规则的查询和修改。
- 每个SQL引擎在进行SQL业务响应、执行过程中，基于SQL防御策略进行主动防御行为。
- 管理员可将SQL防御行为在“提示”、“拦截”、“熔断”选项之间进行灵活切换，系统会将发生的SQL防御事件实时写入到防御审计日志中。运维人员可进行日志分析，评估现网SQL质量，提前感知潜在SQL风险，并做出有效预防措施。

本节操作介绍创建SQL防御规则以增加SQL防御能力的配置方法。

DLI SQL 防御规则约束与限制

- 仅Spark 3.3.x及以上版本支持SQL防御功能。
- 同一个队列，同一个动作的防御规则仅支持创建一条。
- 每条规则最多可以关联50个SQL队列。
- 每个项目最多可以创建1000条规则。

创建 SQL 防御规则

您可以在DLI SQL防御界面对指定SQL队列添加SQL防御规则，系统会对触发规则的SQL请求进行提示、拦截或熔断操作。

📖 说明

添加或者修改SQL防御规则时请结合业务场景评估规则的开启、规则阈值是否合理，避免不合理的防御规则对相关SQL请求进行拦截或阻断后，对业务造成影响。

1. 登录DLI管理控制台。
2. 选择“全局配置 > SQL防御”，打开SQL防御页面。
3. 单击“创建规则”，编辑规则信息。

表 7-5 SQL 防御规则参数配置

参数	说明
规则名称	自定义SQL防御规则名称。

参数	说明
系统规则	选择防御规则，DLI支持的系统防御规则请参考 DLI支持的SQL防御系统规则 。
队列	选择绑定规则的队列。
描述信息	输入规则描述信息。
防御规则动作	配置当前SQL防御规则动作的阈值参数。 SQL规则支持的类型： <ul style="list-style-type: none"> ● 提示：配置系统对SQL请求满足防御规则后是否进行日志记录和提示处理。开启按钮时，如果当前规则有变量参数，需同时配置阈值。 ● 拦截：配置系统对SQL请求满足防御规则后是否进行拦截处理。开启按钮时，如果当前规则有变量参数，需同时配置阈值。 ● 熔断：配置系统对SQL请求满足防御规则后是否进行熔断处理。开启按钮时，如果当前规则有变量参数，需同时配置阈值。

4. 单击“确定”完成规则的添加。

规则添加成功后，可以在“SQL防御”界面查看已添加的防御规则。防御规则动态生效。

如需对当前规则进行调整，可单击对应规则所在“操作”列的“修改”，修改SQL防御规则。

DLI 支持的 SQL 防御系统规则

本节操作介绍DLI支持的系统防御规则，详细信息参考[表7-6](#)。

- 系统默认创建的规则是指在队列创建时，DLI自动为您创建的SQL防御规则，该规则与队列绑定，且不支持删除。
- 以下规则为系统默认创建的规则：Scan files number、Scan partitions number、Shuffle data(GB)、Count(distinct) occurrences、Not in<Subquery>
- 同一个队列，同一个动作的防御规则仅支持创建一条。
- 系统默认创建的规则会分别创建每个支持动作的规则。例如：创建队列时，会分别创建“提示”和“拦截”动作的Scan files number规则。
- 不同的引擎版本支持的防御规则不同。

如需查看队列的引擎版本，您可以在队列资源的资源列表页面，通过查看队列基本信息中的“默认版本”获取引擎的版本信息。

图 7-2 查看队列引擎版本

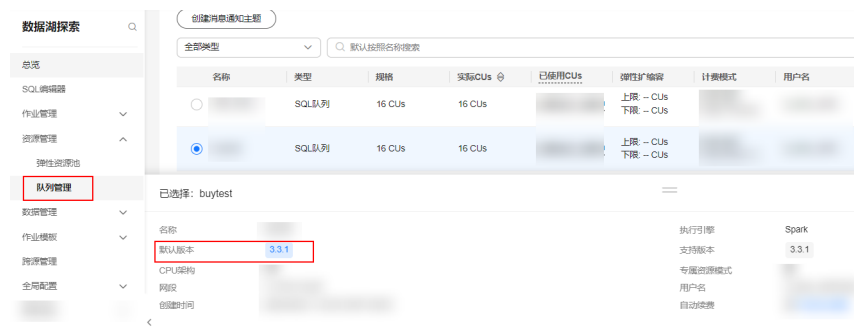


表 7-6 DLI 支持的系统防御规则

规则ID	规则名称	说明	类别	适用引擎	支持的动作	取值说明	系统默认创建规则	SQL语句示例
dynamic_0001	Scan files number	扫描文件数的限制。	dynamic	spark Hetu Engine	提示拦截	取值范围： 1-200000 默认值： 200000	是	NA
dynamic_0002	Scan partitions number	对单个表操作（select, delete, update, alter）涉及的分区数超限限制。	dynamic	spark	提示拦截	取值范围： 1-500000 默认值： 5000	是	select * from 分区表
running_0002	Memory used(MB)	SQL的占用内存峰值超绝对值限制。	running	spark	熔断	取值范围： 1-8388608	否	NA
running_0003	Run time(S)	SQL已经运行的时长限制。	running	spark	熔断	单位： 秒 取值范围： 1-43200	否	NA
running_0004	Scan data(GB)	扫描数据量的限制。	running	spark	熔断	单位： GB 取值范围： 1-10240	否	NA

规则ID	规则名称	说明	类别	适用引擎	支持的动作	取值说明	系统默认创建规则	SQL语句示例
running_0005	Shuffle data(GB)	Shuffle数据量的限制。	running	spark	熔断	单位: GB 取值范围: 1-10240 默认值: 2048	是	NA
static_0001	Count(distinct) occurrences	SQL中count(distinct)出现次数限制。	static	spark	提示拦截	取值范围: 1-100 默认值: 10	是	SELECT COUNT(DISTINCT deviceId), COUNT(DISTINCT collDeviceId) FROM table GROUP BY deviceName, collDeviceName, collCurrentVersion ;
static_0002	Not in<Subquery>	SQL中是否使用了not in <subquery>语句。	static	spark	提示拦截	取值范围: 是/否 默认值: 是	是	SELECT * FROM Orders o WHERE Orders.Order_ID not in (Select Order_ID FROM HeldOrders h where h.order_id = o.order_id);

规则ID	规则名称	说明	类别	适用引擎	支持的动作	取值说明	系统默认创建规则	SQL语句示例
static_0003	Join occurrences	SQL中的join次数的限制。	static	spark	提示拦截	取值范围：1-50	否	SELECT name, text FROM table_1 JOIN table_2 ON table_1.Id = table_2.Id
static_0004	Union occurrences	SQL中的union all次数的限制。	static	spark	提示拦截	取值范围：1-100	否	select * from tables t1 union all select * from tables t2 union all select * from tables t3
static_0005	Subquery nesting layers	子查询嵌套层数的限制。	static	spark	提示拦截	取值范围：1-20	否	select * from (with temp1 as (select * from tables) select * from temp1);
static_0006	Sql size(KB)	SQL文件大小的限制。	static	spark	提示拦截	单位：KB 取值范围：1-1024	否	NA
static_0007	Cartesian product	多表关联时笛卡尔积的限制	static	spark	提示拦截	取值范围：0-1	否	select * from A,B;

7.4 设置 SQL 作业优先级

操作场景

在实际作业运行中，由于作业的重要程度以及紧急程度不同，需要重点保障重要和紧急的作业正常运行，因此需要满足它们正常运行所需的计算资源。

DLI提供的设置作业优先级功能，可以对每个SQL设置作业优先级，当资源不充足时，可以优先满足优先级较高的作业的计算资源。

使用须知

- 对于每个作业都允许设置优先级，其取值为1-10，数值越大表示优先级越高。优先满足高优先级作业的计算资源，即如果高优先级作业计算资源不足，则会减少低优先级作业的计算资源
- SQL队列上运行的作业优先级默认为3。
- 调整作业优先级需要停止作业后编辑，并重新提交运行才能生效。

设置 SQL 作业优先级

在“设置 > 参数配置”中配置如下参数，其中x为优先级取值。
`spark.sql.dli.job.priority=x`

1. 登录DLI管理控制台。
2. 单击“作业管理 > SQL作业”。
3. 选择待配置的作业，单击操作列下的编辑。
4. 在“设置 > 参数配置”中配置`spark.sql.dli.job.priority`参数。

图 7-3 SQL 作业配置样例



7.5 查询 SQL 作业日志

操作场景

DLI作业桶用于存储DLI作业运行过程中产生的临时数据，例如：作业日志、作业结果。

本节操作指导您在DLI管理控制台配置DLI作业桶，并获取SQL作业日志的操作方法。

使用须知

- 请勿将该DLI作业桶绑定的OBS桶用作其它用途，避免出现作业结果混乱等问题。
- DLI作业要由用户主账户统一设置及修改，子用户无权限。
- 不配置DLI作业桶无法查看作业日志。
- 您可以通过配置桶的生命周期规则，定时删除桶中的对象或者定时转换对象的存储类别。
- DLI的作业桶设置后请谨慎修改，否则可能会造成历史数据无法查找。

前提条件

配置前，请先购买OBS桶或并行文件系统。大数据场景推荐使用并行文件系统，并行文件系统（Parallel File System）是对象存储服务（Object Storage Service，OBS）提供的一种经过优化的高性能文件系统，提供毫秒级别访问时延，以及TB/s级别带宽和百万级别的IOPS，能够快速处理高性能计算（HPC）工作负载。

并行文件系统的详细介绍和使用说明，请参见《[并行文件系统特性指南](#)》。

配置 DLI 作业桶


1. 在DLI控制台左侧导航栏中单击“全局配置 > 工程配置”。
2. 在“工程配置”页面，选择“DLI作业桶”，单击配置桶信息。

图 7-4 工程配置




3. 单击打开桶列表。
4. 选择用于存放DLI作业临时数据的桶，并单击“确定”。
完成设置后DLI作业运行过程中产生的临时数据将会存储在该OBS桶中。

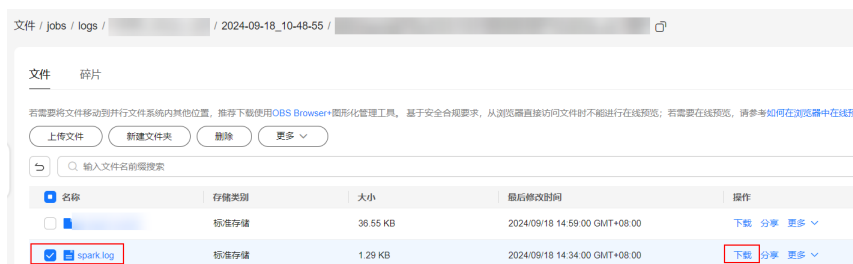
图 7-5 设置 DLI 作业桶



查询 SQL 作业日志

1. 登录DLI管理控制台，单击“作业管理 > SQL作业”。
2. 选择待查询的SQL作业，单击操作列的“更多 > 归档日志”。
系统自动跳转至DLI作业桶日志路径下。
3. 选择需要查看的日期，单击操作列的“下载”，下载SQL作业日志到本地。

图 7-6 下载 SQL 作业日志



7.6 管理 SQL 作业

在 SQL 作业列表页面查看作业的基本信息

DLI SQL作业管理页面显示所有SQL作业，作业数量较多时，系统分页显示，可根据需要跳转至指定页面。您可以查看任何状态下的作业。作业列表默认按创建时间降序排列。


表 7-7 作业管理参数

参数	参数说明
队列	作业所属队列的名称。
执行引擎	SQL作业支持Spark和HetuEngine两种引擎。 <ul style="list-style-type: none"> • Spark：显示执行引擎为“Spark”的作业。 • HetuEngine：显示执行引擎为“HetuEngine”的作业。
用户名	执行该作业的用户名。

参数	参数说明
类型	<p>作业的类型，包括如下。</p> <ul style="list-style-type: none"> ● IMPORT：导入数据到DLI的作业。 ● EXPORT：从DLI导出数据的作业。 ● DCL：包括传统DCL，以及队列权限相关的操作。 ● DDL：与传统DDL操作一致，即创建和删除数据库，创建和删除表的作业。 ● QUERY：执行SQL查询数据的作业。 ● INSERT：执行SQL插入数据的作业。 ● UPDATE：更新数据。 ● DELETE：删除SQL作业。 ● DATA_MIGRATION：数据迁移。 ● RESTART_QUEUE：重启队列。 ● SCALE_QUEUE：队列规格变更（扩容/缩容）。
状态	<p>作业的状态信息，包括如下。</p> <ul style="list-style-type: none"> ● 提交中 ● 运行中 ● 已成功 ● 已取消 ● 已失败 ● 规格变更中
执行语句	<p>作业的具体SQL语句以及导出、建表的操作，此处展示操作的描述。</p> <p>单击  可复制对应的语句。</p>
运行时长	作业的运行时长。
创建时间	每个作业的创建时间，可按创建时间顺序或倒序显示作业列表。

参数	参数说明
操作	<ul style="list-style-type: none"> ● 编辑：重新编辑修改该作业。 ● 终止： <ul style="list-style-type: none"> - 当作业状态在“提交中”和“运行中”时，“终止”按钮才生效。 - 当作业状态为“已成功”、“已失败”、“已取消”的作业不能终止。 - 当“终止”按钮为灰色时，表示无法执行终止操作。 ● 重新执行：重新执行该作业。 ● SparkUI：单击后，将跳转至Spark任务运行情况界面。 <p>说明</p> <ul style="list-style-type: none"> ● 新建队列，运行作业时会重新拉集群，大概需要10分钟左右才能拉好集群，在集群创建好之前单击SparkUI会导致缓存空的projectID，从而导致无法查看SparkUI。建议使用专属队列，集群不会被释放，就不会有该问题，或者提交作业后等一段时间再查看SparkUI，确保集群已经拉好了，不要立即单击SparkUI。 ● 目前DLI配置SparkUI只展示最新的100条作业信息。 ● QUERY作业和异步DDL作业除上述操作外，还包括： <ul style="list-style-type: none"> - 下载到本地：异步DDL和QUERY语句支持将结果下载到本地。具体操作请见导出作业结果到本地。 - 查看结果：查看作业运行结果。 - 导出结果：将作业运行结果导出至用户创建的OBS桶中。 ● EXPORT作业除上述操作外，还包括： <ul style="list-style-type: none"> - 立即下载 ● 归档日志：将作业日志保存到系统创建的DLI临时OBS数据桶中。 <p>说明 default队列下运行的作业或者该作业为同步作业时不支持归档日志操作。</p>

查看作业详情

在“SQL作业”页面，选中一条作业，单击该作业对应的 ，可查看该条作业的详细信息。

不同类型的作业，显示的作业详情不同。作业详情根据作业类型、状态和配置选项不同显示可能存在差异，具体以实际界面显示为准。以导入数据作业，建表作业和查询作业为例说明。其他作业类型支持查看的详细信息请以控制台信息为准。

- 导入数据（load data）作业（作业类型：IMPORT），包括以下信息：队列，作业ID，用户名，类型，状态，执行语句，运行时长，创建时间，结束时间，参数设置，标签，结果条数，已扫描数据，扫描数据条数，错误记录条数，存储路径，数据格式，数据库，表，表头，分隔符，引用字符，转义字符，日期格式，时间戳格式，CPU累计使用量，输出字节。
- 建表（create table）作业（作业类型：DDL），包括以下信息：队列，作业ID，用户名，类型，状态，执行语句，运行时长，创建时间，结束时间，参数设置，标签，结果条数，已扫描数据，数据库。

- 查询 (select) 作业 (作业类型: QUERY), 包括以下信息: 队列, 作业ID, 用户名, 类型, 状态, 执行语句, 运行时长, 创建时间, 结束时间, 参数设置, 标签, 结果条数 (运行成功, 可导出结果), 已扫描数据, 执行用户, 结果状态 (运行成功, 可查看结果; 运行失败, 显示失败原因), 数据库, CPU累计使用量, 输出字节。

说明

- CPU累计使用量: 作业执行过程的CPU消耗总和, 单位: Core*ms
- 输出字节: 作业执行完成后输出的字节数。

查找作业

在“SQL作业”页面, 可以通过以下方式对作业进行过滤筛选, 在页面中显示符合对应条件的作业。

- 选择队列名称
- 选择执行引擎
- 设置日期范围
- 输入用户名/执行语句/作业ID/标签
- 选择创建时间顺序/倒序排列
- 选择作业类型
- 选择作业状态
- 选择运行时长顺序/倒序排列

终止 SQL 作业

在“SQL作业”页面, 可单击“操作”列的“终止”, 终止“提交中”或“运行中”的作业。

7.7 创建并管理 SQL 作业模板

7.7.1 创建 SQL 作业模板

为了便捷快速的执行SQL操作, DLI支持定制模板或将正在使用的SQL语句保存为模板。保存模板后, 不需编写SQL语句, 可通过模板直接执行SQL操作。

SQL模板包括样例模板和自定义模板。当前系统默认的样例模板包括22条标准的TPC-H查询语句, 可以满足用户大部分的TPC-H需求场景测试, TPC-H样例说明请参考[DLI 预置的SQL模板中TPC-H样例数据说明](#)。

说明

在“SQL模板”页面右上角, 单击“设置”可以选择是否按照分组展示模板。

如果选择“按分组展示”, 有以下三种展示方式: 展开第一个分组、全部展开、全部收起。

创建 SQL 作业模板

创建模板的操作入口有两个, 分别在“作业模板”和“SQL编辑器”页面。

- 在“作业模板”页面创建模板。
 - a. 在管理控制台左侧，单击“作业模板” > “SQL模板”。
 - b. 在“SQL模板”页面，单击右上角“创建模板”。
输入模板名称、语句和描述信息，详细参数介绍请参见表7-8。

图 7-7 创建模板

创建模板

您还可以创建89个模板。如需申请更多配额请点击[申请扩大配额](#)。

* 名称

* 语句
0/10,000

描述
0/256

分组设置 已有分组 创建新分组 不分组

* 分组名称

确定 取消

表 7-8 参数说明

参数名称	描述
名称	模板名称。 <ul style="list-style-type: none"> • 模板名称只能包含数字、英文字母和下划线，但不能是纯数字，不能以下划线开头，且不能为空。 • 输入长度不能超过50个字符。
语句	需要保存为模板的SQL语句。
描述	该模板的相应描述。
分组设置	<ul style="list-style-type: none"> • 已有分组 • 创建新分组 • 不分组
分组名称	“分组设置”选择“已有分组”或者“创建新分组”时，需要填写分组名称。

- c. 单击“确定”，完成模板创建。
- 在“SQL编辑器”页面创建模板。
 - a. 在管理控制台左侧，单击“SQL编辑器”。
 - b. 单击SQL作业编辑窗口右上方的“更多”，选择“设为模板”，可将编辑窗口中的SQL语句设置为模板。
输入模板名称、语句和描述信息，详细介绍请参见表7-8。
 - c. 单击“确定”，完成模板创建。

使用模板提交 SQL 作业

执行模板操作步骤如下：

1. 在管理控制台左侧，单击“作业模板”>“SQL模板”。
2. 在“SQL模板”页面，勾选相应的模板，单击“操作”列的“执行”，将跳转至“SQL编辑器”页面，并在SQL作业编辑窗口中自动输入对应的SQL语句。
3. 在SQL作业编辑窗口右上方，单击“执行”运行SQL语句，执行结束后，可以在SQL作业编辑窗口下方区域中查看执行结果。

查找 SQL 作业模板

在“SQL模板”页面，可在右上方搜索框中输入模板名称关键字，查找与之匹配的模板。

修改 SQL 作业模板

修改模板仅支持对自定义模板进行操作，具体步骤如下：

- 步骤1** 在“SQL模板”页面，单击“自定义模板”，选中需修改的模板，单击“操作”列的“修改”。
- 步骤2** 在弹出的“修改模板”对话框中，根据需要修改模板的名称、语句和描述。
- 步骤3** 单击“确定”，保存修改结果。

----结束

删除模板

在“SQL模板”页面，单击“自定义模板”，勾选一个或多个待删除的模板，单击“删除”，可删除选中的模板。

7.7.2 使用 SQL 作业模板开发并提交 SQL 作业

为了便捷快速地执行SQL操作，DLI支持定制模板或将正在使用的SQL语句保存为模板。保存模板后，不需编写SQL语句，可通过模板直接执行SQL操作。

当前系统提供了多条标准的TPC-H查询语句模板，您可以按需选择自定义模板或系统模板创建SQL作业。

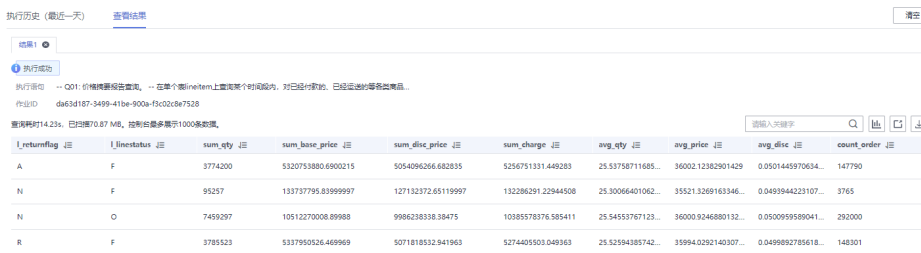
本样例演示通过一个TPC-H样例模板开发并提交SQL作业的基本流程：

操作步骤

1. 登录DLI管理控制台。
2. 在DLI管理控制台，选择“作业模板 > SQL模板”。
3. 在“tpchQuery”下找到适合您业务场景的样例模板，单击操作列的“执行”进入SQL编辑器页面。
4. 在SQL编辑器页面右侧的编辑窗口上方，“执行引擎”选择“spark”，“队列”选择“default”，“数据库”选择“default”，单击“执行”。



5. SQL作业编辑窗口下方“查看结果”页签查看查询结果。



本示例使用系统预置的“default”队列和数据库进行演示，也可以在自建的队列和数据库下执行。

创建队列请参考[创建队列](#)。创建数据库请参考[创建数据库](#)。

后续指引

完成TPC-H样例模板开发并提交SQL作业操作后，如果您想了解更多关于SQL作业相关操作，建议您参考以下指引阅读。

分类	文档	说明
界面操作	SQL编辑器	提供执行SQL语句操作的界面指导，包含SQL编辑器界面基本功能介绍、快捷键以及使用技巧等说明。
	Spark SQL作业管理	提供SQL作业管理界面功能介绍。
	Spark SQL模板管理	DLI支持定制模板或将正在使用的SQL语句保存为模板，便捷快速的执行SQL操作。
开发指导	Spark SQL语法参考	提供SQL数据库、表、分区、导入及导出数据、自定义函数、内置函数等语法说明和样例指导。
	使用Spark作业访问DLI元数据	提供SQL作业开发的操作指引和样例代码参考。

分类	文档	说明
	Spark SQL 相关 API	提供SQL相关API的使用说明。

7.7.3 DLI 预置的 SQL 模板中 TPC-H 样例数据说明

TPC-H 样例数据简介

TPC-H（商业智能计算测试）是交易处理效能委员会（TPC, Transaction Processing Performance Council）组织制定的用来模拟决策支持类应用的一个测试集。目前，在学术界和工业界普遍用来评价决策支持技术方面应用的性能。这种商业测试可以全方位评测系统的整体商业计算综合能力，对厂商的要求更高，同时也具有普遍的商业实用意义，目前在银行信贷分析和信用卡分析、电信运营分析、税收分析、烟草行业决策分析中都有广泛的应用。

TPC-H 基准测试是由 TPC-D（由 TPC 组织于 1994 年制定的标准，用于决策支持系统方面的测试基准）发展而来的。TPC-H 用 3NF 实现了一个数据仓库，共包含 8 个基本关系，其数据量可以设定从 1G~3T 不等。TPC-H 基准测试包括 22 个查询(Q1~Q22)，其主要评价指标是各个查询的响应时间，即从提交查询到结果返回所需时间。TPC-H 基准测试的度量单位是每小时执行的查询数(QphH@size)，其中“H”表示每小时系统执行复杂查询的平均次数，“size”表示数据库规模的大小，能够反映出系统在处理查询时的能力。TPC-H 是根据真实的生产运行环境来建模的，这使得它可以评估一些其他测试所不能评估的关键性能参数。总而言之，TPC 组织颁布的 TPC-H 标准满足了数据仓库领域的测试需求，并且促使各个厂商以及研究机构将该项技术推向极限。

本示例将演示 DLI 直接对存储在 OBS 中的 TPC-H 数据集进行查询的操作，DLI 已经预先生成了 100M 的 TPC-H-2.18 的标准数据集，已将数据集上传到了 OBS 的 tpch 文件夹中，并且赋予了只读访问权限，方便用户进行查询操作。

TPC-H 的测试和度量指标

TPC-H 测试分解为 3 个子测试：数据装载测试、Power 测试和 Throughput 测试。建立测试数据库的过程被称为装载数据，装载测试是为测试 DBMS 装载数据的能力。装载测试是第一项测试，测试装载数据的时间，这项操作非常耗时。Power 测试是在数据装载测试完成后，数据库处于初始状态，未进行其它任何操作，特别是缓冲区还没有被测试数据库的数据，被称为 raw 查询。Power 测试要求 22 个查询顺序执行 1 遍，同时执行一对 RF1 和 RF2 操作。最后进行 Throughput 测试，也是最核心和最复杂的测试，更接近于实际应用环境，与 Power 测试比对 SUT 系统的压力有非常大的增加，有多个查询语句组，同时有一对 RF1 和 RF2 更新流。

测试中测量的基础数据都与执行时间有关，这些时间又可分为：装载数据的每一步操作时间、每个查询执行时间和每个更新操作执行时间，由这些时间可计算出：数据装载时间、Power@Size、Throughput@Size、QphH@Size 和 \$/QphH@Size。

Power@Size 是 Power 测试的结果，被定义为查询时间和更改时间的几何平均值的倒数，公式如下：

$$\text{TPC-H Power@Size} = \frac{3600 * SF}{\sqrt[24]{\prod_{i=1}^{i=22} QI(i,0) * \prod_{j=1}^{j=2} RI(j,0)}}$$

其中：Size 为数据规模；SF 为数据规模的比例因子；QI (i, 0) 为第 i 个查询的时间，以秒为单位；R (1j, 0) 为 RFj更新的时间，以秒为单位。

Throughput@Size 是Throughput 测试的结果，被定义为所有查询执行时间平均值的倒数，公式如下：

$$QphH@Size = \sqrt{\text{Power @ Size} * \text{Throughput @ Size}}$$

业务场景

用户可以通过DLI内置的TPC-H测试套件进行简单高效的交互式查询，无需用户上传数据，即可以体验DLI的核心功能。

DLI 内置 TPC-H 的优势

- 用户只需要登录DLI，完成授予权限，即可操作SQL语句，无需用户自己创建表和导入数据。
- 预置22条TPC-H SQL查询模板，功能丰富，可满足大部分的商业场景，无需用户自行下载TPC-H的查询语句，省时省力。
- 用最小的时间代价体验serverless化的DLI产品，领略数据湖带给我们的全新体验。

注意事项

子账号使用TPC-H测试套件时，需要主账号为子账号赋权OBS访问权限和查看主账号表的权限；如果主账号未登录过DLI服务，子账号除上述权限外，还需要创建数据库和创建表的权限。

使用 TPC-H 样例模板开发并提交 SQL 作业

为了便捷快速地执行SQL操作，DLI支持定制模板或将正在使用的SQL语句保存为模板。保存模板后，不需编写SQL语句，可通过模板直接执行SQL操作。

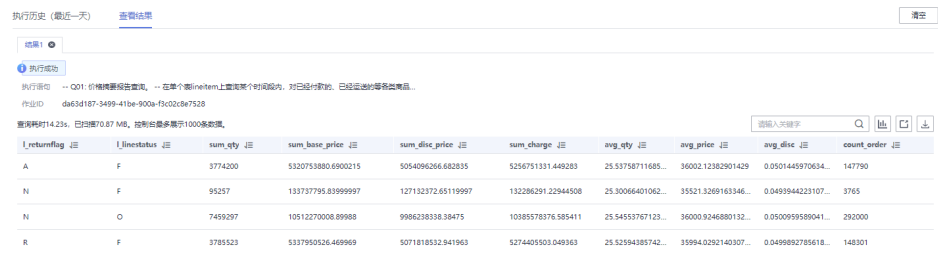
1. 登录DLI管理控制台。
2. 在DLI管理控制台，选择“作业模板”>“SQL模板”>“样例模板”，在“tpchQuery”下找到“Q1_价格摘要报告查询”样例模板，单击操作列的“执行”进入“SQL编辑器”。



3. 在“SQL编辑器”页面右侧的编辑窗口上方，“执行引擎”选择“spark”，“队列”选择“default”，“数据库”选择“default”，单击“执行”。



4. SQL作业编辑窗口下方“查看结果”页签查看查询结果。



本示例使用系统预置的“default”队列和数据库进行演示，也可以在自建的队列和数据库下执行。

8 使用 DLI 提交 Flink 作业

8.1 Flink 作业概述

DLI支持的两种类型的Flink作业:

- **Flink OpenSource SQL类型作业:**

- 完全兼容社区版的Flink，确保了作业可以在这些Flink版本上无缝运行。
- 在社区版Flink的基础上，DLI扩展了Connector的支持，新增了Redis、DWS作为数据源类型。为用户提供了更多的数据源选择，使得数据集成更加灵活和方便。
- Flink OpenSource SQL作业适合通过SQL语句来定义和执行流处理逻辑的场景，简化了流处理的复杂性，使得开发者可以更加专注于业务逻辑的实现。

创建Flink OpenSource SQL请参考[创建Flink OpenSource SQL作业](#)。

- **Flink Jar作业:**

- DLI允许用户提交编译为Jar包的Flink作业，提供了更高的灵活性和自定义能力，适合需要进行复杂数据处理的场景。
- 当社区版Flink提供的Connector不能满足特定需求时，用户可以通过Jar作业来实现自定义的Connector或数据处理逻辑。
- 适合需要实现UDF（用户定义函数）或特定库集成的场景，用户可以利用Flink的生态系统来实现高级的流处理逻辑和状态管理。

创建Flink Jar作业请参考[创建Flink Jar作业](#)。

8.2 创建 Flink OpenSource SQL 作业

本章节介绍如何新建Flink OpenSource SQL作业。

DLI Flink OpenSource SQL类型作业完全兼容社区Flink版本，并在社区connector基础之上，新增了Redis、DWS(GaussDB)数据源类型。社区Flink SQL DDL/DML/函数等语法说明及限制可参考[Table API & SQL](#)。

- Flink OpenSource SQL1.15语法请参考[Flink OpenSource SQL1.15语法](#)。
- Flink OpenSource SQL1.12语法请参考[Flink OpenSource SQL1.12语法](#)。

前提条件

- 创建Flink OpenSource SQL作业时，需要事先准备数据源以及数据输出通道。
- 创建Flink OpenSource SQL作业，访问其他外部数据源时，需要先创建跨源连接，打通作业运行队列到外部数据源之间的网络。
 - 当前Flink作业支持访问的外部数据源详情请参考[DLI常用跨源分析开发方式](#)。
 - 创建跨源连接操作请参见[配置DLI与数据源网络连通（增强型跨源连接）](#)。
创建完跨源连接后，可以通过“资源管理 > 队列管理”页面，单击“操作”列“更多”中的“测试地址连通性”，验证队列到外部数据源之间的网络连通是否正常。详细操作可以参考[测试地址连通性](#)。

注意事项

创建作业提交任务前，建议先开通云审计服务，用于记录与DLI服务相关的操作事件，便于日后的查询、审计和回溯。云审计服务支持的DLI操作列表详见[使用CTS审计DLI服务](#)。

关于如何开通云审计服务以及如何查看追踪事件，请参考《[云审计服务快速入门](#)》。

创建 Flink OpenSource SQL 作业

步骤1 在DLI管理控制台的左侧导航栏中，单击“作业管理” > “Flink作业”，进入“Flink作业”页面。

步骤2 在“Flink作业”页面右上角单击“创建作业”，弹出“创建作业”对话框。

图 8-1 创建 Flink OpenSource SQL 作业

创建作业 ×

类型: Flink OpenSource SQL

★ 名称: testFlinkJob

描述: 请输入描述

模板名称: -请选择模板名称-

标签: 如果您需要使用同一标签识别多种云资源，即所有服务均可在标签输入框下拉选择同一标签。建议在TMS中创建预定义标签。 [查看预定义标签](#)

在下方键/值输入框输入内容后单击添加，即可将标签加入此处

请输入标签键 | 请输入标签值 | 添加

您还可以添加10个标签。

确定 | 取消

步骤3 配置作业信息。

表 8-1 作业配置信息

参数	参数说明
类型	选择“Flink OpenSource SQL”：用户通过编辑SQL语句来启动作业。
名称	作业名称，只能由字母、中文、数字、中划线和下划线组成，并且长度为1~57字节。 说明 作业名称必须是唯一的。
描述	作业的相关描述，长度为0~512字节。
模板名称	用户可以选择样例模板或自定义的作业模板。关于模板的详细信息，请参见 管理Flink作业模板 。
标签	使用标签标识云资源。包括标签键和标签值。如果您需要使用同一标签标识多种云资源，即所有服务均可在标签输入框下拉选择同一标签，建议在标签管理服务（TMS）中创建预定义标签。 如您的组织已经设定DLI的相关标签策略，则需按照标签策略规则为资源添加标签。标签不符合标签策略的规则，则可能会导致资源创建失败，请联系组织管理员了解标签策略详情。 具体请参考《 标签管理服务用户指南 》。 说明 <ul style="list-style-type: none"> 最多支持20个标签。 一个“键”只能添加一个“值”。 每个资源中的键名不能重复。 标签键：在输入框中输入标签键名称。 说明 标签的键的最大长度为128个字符，标签的键可以包含任意语种字母、数字、空格和_ . : + - @ ,但首尾不能含有空格，不能以_sys_开头。 标签值：在输入框中输入标签值。 说明 标签值的最大长度为255个字符，标签的值可以包含任意语种字母、数字、空格和_ . : + - @ 。

步骤4 单击“确定”，进入作业“编辑”页面。

步骤5 编辑OpenSource SQL作业。

在SQL语句编辑区域，输入详细的SQL语句。相关SQL语句请参考《[数据湖探索Flink OpenSource SQL语法参考](#)》。

步骤6 单击“语义校验”，确保语义校验成功。

- 只有语义校验成功后，才可以执行“启动”作业的操作。
- 如果校验成功，提示“SQL语义校验成功”。
- 如果校验失败，会在错误的SQL语句前面显示红色的“X”记号，鼠标移动到“X”号上可查看详细错误，请根据错误提示修改SQL语句。

说明

Flink 1.15不支持语法校验功能。

步骤7 设置作业运行参数。

图 8-2 设置 Flink OpenSource SQL 作业运行参数

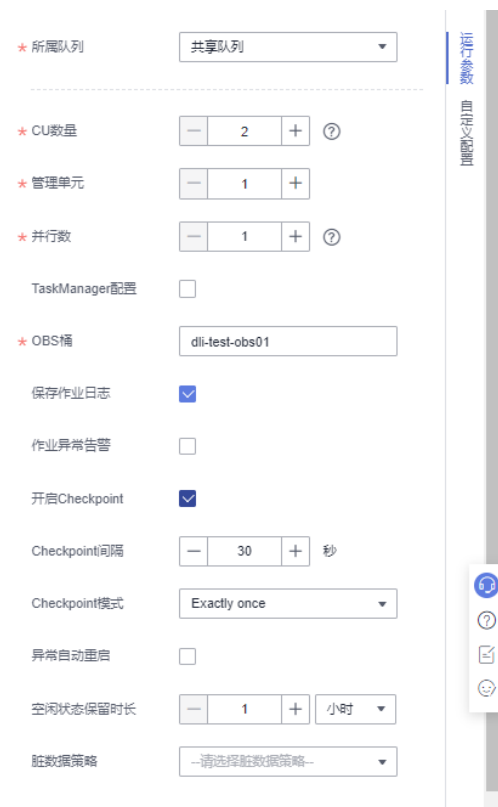


表 8-2 作业运行参数说明

参数	参数说明
所属队列	选择执行作业所需的队列资源。
UDF Jar	<p>用户自定义UDF文件，在后续作业中可以调用插入Jar包中的自定义函数。</p> <p>UDF Jar包的管理方式：</p> <ul style="list-style-type: none"> 上传OBS管理程序包：提前将对应的jar包上传至OBS桶中。并在此处选择对应的OBS路径。 上传DLI管理程序包：提前将对应的jar包上传至OBS桶中，并在DLI管理控制台的“数据管理>程序包管理”中创建程序包，具体操作请参考创建DLI程序包。 <p>Flink1.15及以上版本在创建作业时仅支持配置OBS中的程序包，不支持读取DLI程序包。</p>

参数	参数说明
Flink版本	<p>Flink版本是选择作业运行时所使用的Flink的版本。不同版本的Flink支持不同的特性。</p> <p>了解更多Flink版本的信息请参考DLI Flink版本说明。</p> <p>选择使用Flink1.15版本时请在作业中配置允许DLI访问的云服务的委托信息。</p> <p>Flink 1.15版本语法请参考Flink OpenSource SQL1.15版本使用说明、Flink OpenSource SQL1.15语法。</p> <p>Flink 1.12版本语法请参考Flink OpenSource SQL1.12语法。</p> <p>说明</p> <p>不建议长期混用不同版本的Flink引擎。</p> <ul style="list-style-type: none"> 长期混用不同版本的Flink引擎会导致代码在新旧版本之间不兼容，影响作业的执行效率。 当作业依赖于特定版本的库或组件，长期混用不同版本的Flink引擎可能会导致作业因依赖冲突而执行失败。
委托	<p>使用Flink 1.15版本执行作业时，按需配置自定义委托用于授予DLI访问其他服务的操作权限。</p>
CU数量	<p>CU数量为DLI的计算单元数量和管理单元数量总和，CU也是DLI的计费单位，1CU=1核4G。</p> <p>当前配置的CU数量为运行作业时所需的CU数，不能超过其绑定队列的CU数量。</p> <p>说明</p> <p>当开启TaskManager配置时，为了优化弹性资源池队列的管理，在您设置“单TM Slot”后，为您自动调整CU数量与实际CU数量一致。</p> <p>CU数量=实际CU数量=max[管理单元和TaskManager的CPU总和，(管理单元和TaskManager的内存总和/4)]</p> <ul style="list-style-type: none"> 管理单元和TaskManager的CPU总和=实际TM数 * 单TM所占CU数 + 管理单元。 管理单元和TaskManager的内存总和= 实际TM数 * 设置的单个TM的内存 + 管理单元内存 如果配置了单 TM Slot数，实际TM数 = 并行数 / 单 TM Slot数。 如果没配置了单 TM Slot数，实际TM数 = (CU数量 - 管理单元) / 单TM所占CU数。 如果没在优化参数配置单个TM的内存和管理单元内存，默认单个TM的内存 = 单TM所占CU数 * 4。管理单元内存 = 管理单元 * 4。 Spark资源并行度由Executor数量和Executor CPU核数共同决定。
管理单元	<p>管理单元CU数量。</p>
并行数	<p>作业的并行数是指作业中各个算子的并行执行的子任务的数量，即算子的子任务数就是其对应算子的并行度。</p> <p>说明</p> <p>最大并行数不能大于计算单元（CU数量-管理单元）的4倍。</p>

参数	参数说明
TaskManager配置	<p>用于设置TaskManager资源参数。</p> <ul style="list-style-type: none"> 勾选后需配置下列参数： <ul style="list-style-type: none"> “单TM所占CU数”：每个TaskManager占用的资源数量。 “单TM Slot”：每个TaskManager包含的Slot数量。 不勾选该参数，，系统自动按照默认值为您配置。 <ul style="list-style-type: none"> “单TM所占CU数”：默认值为1。 “单TM Slot”：默认值为“(并行数 * 单TM所占CU数) / (CU数量 - 管理单元)”。
OBS桶	<p>选择OBS桶用于保存用户作业日志信息、checkpoint等信息。如果选择的OBS桶是未授权状态，需要单击“OBS授权”。</p>
保存作业日志	<p>设置是否将作业运行时的日志信息保存到OBS。日志信息的保存路径为：“桶名/jobs/logs/作业id开头的目录”。</p> <p>注意 该参数建议勾选，否则作业运行完成后不会生成运行日志，后续如果作业运行异常则无法获取运行日志进行定位。</p> <p>勾选后需配置下列参数： “OBS桶”：选择OBS桶用于保存用户作业日志信息。如果选择的OBS桶是未授权状态，需要单击“OBS授权”。</p> <p>说明 如果同时勾选了“开启Checkpoint”和“保存作业日志”，OBS授权一次即可。</p>
作业异常告警	<p>设置是否将作业异常告警信息，如作业出现运行异常或者欠费情况，以SMN的方式通知用户。</p> <p>勾选后需配置下列参数： “SMN主题”： 选择一个自定义的SMN主题。如何自定义SMN主题，请参见《消息通知服务用户指南》中“创建主题”章节。</p>

参数	参数说明
开启Checkpoint	<p>设置是否开启作业快照，开启后可基于Checkpoint（一致性检查点）恢复作业。</p> <p>勾选后需配置下列参数：</p> <ul style="list-style-type: none"> “Checkpoint间隔”：Checkpoint的时间间隔，单位为秒，输入范围 1~999999，默认值为30s。 “Checkpoint模式”：支持如下两种模式： <ul style="list-style-type: none"> - At least once：事件至少被处理一次。 - Exactly once：事件仅被处理一次。 <p>勾选开启Checkpoint还需配置OBS桶参数：</p> <p>“OBS桶”：选择OBS桶用于保存用户Checkpoint。如果选择的OBS桶是未授权状态，需要单击“OBS授权”。</p> <p>Checkpoint保存路径为：“桶名/jobs/checkpoint/作业id开头的目录”。</p> <p>说明 如果同时勾选了“开启Checkpoint”和“保存作业日志”，OBS授权一次即可。</p>
异常自动重启	<p>设置是否启动异常自动重启功能，当作业异常时将自动重启并恢复作业。</p> <p>勾选后需配置下列参数：</p> <ul style="list-style-type: none"> “异常重试最大次数”：配置异常重试最大次数。单位为“次/小时”。 <ul style="list-style-type: none"> - 无限：无限次重试。 - 有限：自定义重试次数。 “从Checkpoint恢复”：需要同时勾选“开启Checkpoint”才可配置该参数。
空闲状态保留时长	<p>用于清除GroupBy、RegularJoin、Rank、Depuplicate等算子经过最大保留时间后仍未更新的中间状态，默认设置为1小时。</p>
脏数据策略	<p>选择处理脏数据的策略。支持如下三种策略：“忽略”，“抛出异常”和“保存”。</p> <p>“脏数据策略”选择“保存”时，配置“脏数据转储地址”。单击地址框选择保存脏数据的OBS路径。</p> <p>仅DIS数据源支持配置脏数据策略。</p>

步骤8 （可选）根据需要设置自定义配置。相关参数详情可以参考[Flink作业调优](#)。


图 8-3 自定义配置



步骤9 单击“保存”，保存作业和相关参数。

步骤10 单击“启动”，进入“启动Flink作业”页面，确认作业规格和费用后，单击“立即启动”，启动作业。

启动作业后，系统将自动跳转到Flink作业管理页面，新创建的作业将显示在作业列表中，在“状态”列中可以查看作业状态。作业提交成功后，状态将由“提交中”变为“运行中”。运行完成后显示“已完成”。

如果作业状态为“提交失败”或“运行异常”，表示作业提交或运行失败。用户可以在作业列表中的“状态”列中，将鼠标移动到状态图标上查看错误信息，单击可以复制错误信息。根据错误信息解决故障后，重新提交。

说明

其他功能按钮说明如下：

- 另存为：将新建作业另存为一个新作业。
- 静态流图：提供静态资源预估功能和流图展示功能。如图8-5所示。
- 简化流图：展示source到sink的数据处理流程。如图8-4所示。
- 格式化：对SQL语句进行格式化。
- 设为模板：将新创建的作业设置为作业模板。
- 主题设置：设置页面主题，可以设置字体大小，自动换行和页面风格。
- 帮助：跳转至帮助中心，为用户提供SQL语法参考。

----结束

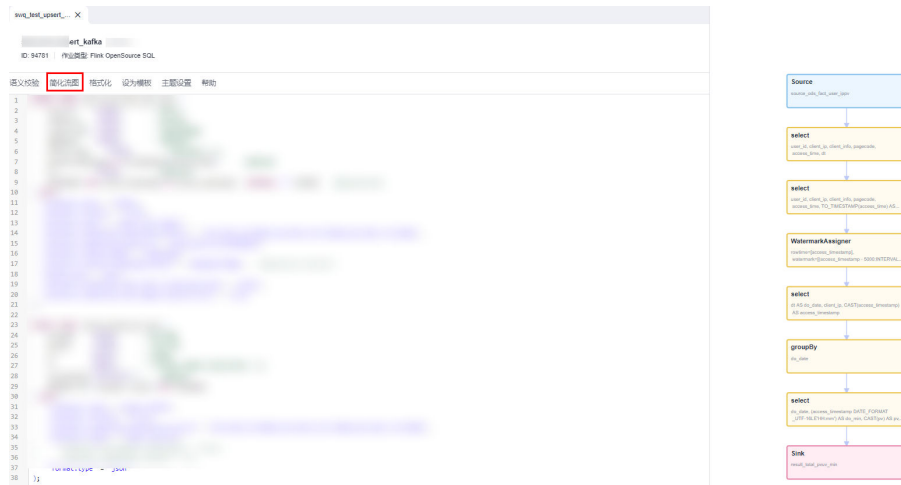
简化流图

在OpenSource SQL作业编辑页面，单击“简化流图”按钮即可展示。

说明

仅Flink 1.12和Flink 1.10版本支持查看简化流图。

图 8-4 简化流程图



静态流图

在OpenSource SQL作业编辑页面，单击“静态流图”按钮即可展示。

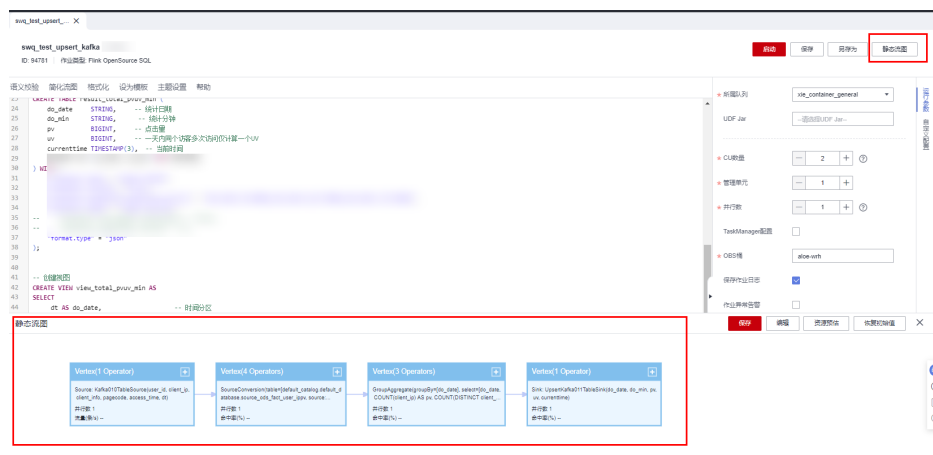
说明

- 仅Flink 1.12和Flink 1.10版本支持查看简化流图。
- Flink OpenSource SQL作业中使用自定义函数时，不支持生成静态流图。

“静态流图”页面还支持以下功能：

- 支持资源预估。通过单击“静态流图”页面中的“资源预估”按钮，可进行资源预估。单击“恢复初始值”按钮，可在资源预估后恢复初始值。
- 支持展示页面缩放。
- 支持根据算子链展开/合并。
- 支持编辑“并行数”，“流量”和“命中率”。
 - 并行数：一个任务的并发数。
 - 流量：算子的数据流量，单位：条/s。
 - 命中率：数据经过算子处理之后的保留率。命中率=算子的数据流出量/流入量，单位：%。

图 8-5 静态流图



8.3 创建 Flink Jar 作业

Flink Jar作业是基于Flink能力进行二次开发的场景，即构建自定义应用Jar包并提交到DLI的队列运行。

Flink Jar作业场景需要用户自行编写并构建应用Jar包，适用于对流计算处理复杂度要求较高的用户场景，且用户可以熟练掌握Flink二次开发能力。

本节操作介绍在DLI管理控制台创建Flink Jar作业的操作步骤。

前提条件

- 创建Flink Jar作业，访问其他外部数据源时，如访问OpenTSDB、HBase、Kafka、DWS、RDS、CSS、CloudTable、DCS Redis、DDS等，需要先创建跨源连接，打通作业运行队列到外部数据源之间的网络。
 - 当前Flink作业支持访问的外部数据源详情请参考[DLI常用跨源分析开发方式](#)。
 - 创建跨源连接操作请参见[配置DLI与数据源网络连通（增强型跨源连接）](#)。创建完跨源连接后，可以通过“资源管理 > 队列管理”页面，单击“操作”列“更多”中的“测试地址连通性”，验证队列到外部数据源之间的网络连通是否正常。详细操作可以参考[测试地址连通性](#)。
- 用户运行Flink Jar作业时，需要将二次开发的应用代码构建为Jar包，上传到已经创建的OBS桶中。
- 由于DLI服务端已经内置了Flink的依赖包，并且基于开源社区版本做了安全加固。为了避免依赖包兼容性问题或日志输出及转储问题，打包时请注意排除以下文件：
 - 系统内置的依赖包，或者在Maven或者Sbt构建工具中将scope设为provided
 - 日志配置文件（例如：“log4j.properties”或者“logback.xml”等）
 - 日志输出实现类JAR包（例如：log4j等）

注意事项

创建作业提交任务前，建议先开通云审计服务，用于记录与DLI服务相关的操作事件，便于日后的查询、审计和回溯。云审计服务支持的DLI操作列表详见[使用CTS审计DLI服务](#)。

关于如何开通云审计服务以及如何查看追踪事件，请参考《[云审计服务快速入门](#)》。

创建 Flink Jar 作业

步骤1 在DLI管理控制台的左侧导航栏中，单击“作业管理” > “Flink作业”，进入“Flink作业”页面。

步骤2 在“Flink作业”页面右上角单击“新建作业”，弹出“新建作业”对话框。

图 8-6 新建 Flink Jar 作业

创建作业
×

类型 Flink Jar ▼

*** 名称**

描述

标签

如果您需要使用同一标签识别多种云资源，即所有服务均可在标签输入框下拉选择同一标签，建议在TMS中创建预定义标签。 [查看预定义标签](#) C

在下方键/值输入框输入内容后单击添加，即可将标签加入此处

您还可以添加10个标签。

步骤3 配置作业信息。

表 8-3 作业配置信息

参数	参数说明
类型	选择Flink Jar。
名称	作业名称，只能由英文、中文、数字、中划线和下划线组成，并且长度为1~57字节。 说明 作业名称必须是唯一的。
描述	作业的相关描述，且长度为0~512字节。

参数	参数说明
标签	<p>使用标签标识云资源。包括标签键和标签值。如果您需要使用同一标签标识多种云资源，即所有服务均可在标签输入框下拉选择同一标签，建议在标签管理服务（TMS）中创建预定义标签。</p> <p>如您的组织已经设定DLI的相关标签策略，则需按照标签策略规则为资源添加标签。标签不符合标签策略的规则，则可能会导致资源创建失败，请联系组织管理员了解标签策略详情。</p> <p>具体请参考《标签管理服务用户指南》。</p> <p>说明</p> <ul style="list-style-type: none"> • 最多支持20个标签。 • 一个“键”只能添加一个“值”。 • 每个资源中的键名不能重复。 <ul style="list-style-type: none"> • 标签键：在输入框中输入标签键名称。 说明 标签的键的最大长度为128个字符，标签的键可以包含任意语种字母、数字、空格和_ . : +-@ ,但首尾不能含有空格，不能以_sys_开头。 • 标签值：在输入框中输入标签值。 说明 标签值的最大长度为255个字符，标签的值可以包含任意语种字母、数字、空格和_ . : +-@ 。

步骤4 单击“确定”，进入编辑页面。

步骤5 选择队列。

步骤6 配置Flink Jar作业参数

图 8-7 配置 Flink Jar 作业参数

The screenshot displays the configuration interface for a Flink Jar job. It includes the following elements:

- 应用程序**: A dropdown menu with the placeholder text "--请选择应用程序--" and a link to "查看内置依赖包".
- 主类**: Two buttons, "默认" (selected) and "指定". Below them is the text "默认类名根据应用程序的Manifest文件指定。".
- 参数**: A text input field with the placeholder "请输入参数 (多个参数用空格分隔)".
- 依赖jar包**: A dropdown menu with the placeholder "--请选择依赖jar包--" and a link to "查看内置依赖包".
- 其他依赖文件**: A dropdown menu with the placeholder "--请选择其他依赖文件--".
- Flink版本**: A dropdown menu with the placeholder "--请选择Flink版本--".
- 优化参数**: A text input field with the placeholder "请输入格式为key=value的参数，多个参数以Enter键分隔".

表 8-4 参数说明

名称	描述
所属队列	选择作业运行时使用的队列资源。
应用程序	<p>选择Jar作业程序包。</p> <p>Jar包的管理方式：</p> <ul style="list-style-type: none"> • 上传OBS管理程序包：提前将对应的jar包上传至OBS桶中。并在此处选择对应的OBS路径。 • 上传DLI管理程序包：提前将对应的jar包上传至OBS桶中，并在DLI管理控制台的“数据管理>程序包管理”中创建程序包，具体操作请参考创建DLI程序包。 <p>Flink1.15推荐配置OBS中的程序包，不推荐使用DLI程序包。 Flink1.15以上版本将不再支持读取DLI程序包。</p>
主类	<p>指定加载的Jar包类名，如KafkaMessageStreaming。</p> <ul style="list-style-type: none"> • 默认：根据Jar包文件的Manifest文件指定。 • 指定：必须输入“类名”并确定类参数列表（参数间用空格分隔）。 <p>说明 当类属于某个包时，主类路径需要包含完整包路径，例如： packagePath.KafkaMessageStreaming</p>
参数	<p>指定类的参数列表，参数之间使用空格分隔。</p> <p>Flink参数支持非敏感的全局变量替换。例如，在“全局配置”>“全局变量”中新增全局变量windowSize，Flink Jar作业就可以添加参数-windowSize {{windowSize}}。</p>
依赖jar包	<p>用户自定义的依赖程序包。依赖的相关程序包将会被放置到集群classpath下。</p> <p>Jar包的管理方式：</p> <ul style="list-style-type: none"> • 上传OBS管理程序包：提前将对应的jar包上传至OBS桶中。并在此处选择对应的OBS路径。 • 上传DLI管理程序包：提前将对应的jar包上传至OBS桶中，并在DLI管理控制台的“数据管理>程序包管理”中创建程序包，具体操作请参考创建DLI程序包。 <p>Flink1.15推荐配置OBS中的程序包，不推荐使用DLI程序包。 Flink1.15以上版本将不再支持读取DLI程序包。</p> <p>打包Flink jar作业jar包时，为避免包信息冲突，不需要上传系统已有的内置依赖包。 内置依赖包请参考DLI内置依赖包。</p>

名称	描述
其他依赖文件	<p>用户自定义的依赖文件。其他依赖文件需要自行在代码中引用。</p> <p>依赖文件的管理方式：</p> <ul style="list-style-type: none"> • 上传OBS管理程序包：提前将对应的依赖文件上传至OBS桶中。并在此处选择对应的OBS路径。 • 上传DLI管理程序包：提前将对应的依赖文件上传至OBS桶中，并在DLI管理控制台的“数据管理>程序包管理”中创建程序包，具体操作请参考创建DLI程序包。 <p>Flink1.15推荐配置OBS中的程序包，不推荐使用DLI程序包。 Flink1.15以上版本将不再支持读取DLI程序包。</p> <p>通过在应用程序中添加以下内容可访问对应的依赖文件。其中，“fileName”为需要访问的文件名，“ClassName”为需要访问该文件的类名。</p> <pre>ClassName.class.getClassLoader().getResource("userData/fileName")</pre>
作业特性	<p>队列为CCE队列时，显示该参数。</p> <ul style="list-style-type: none"> • 基础型 • 自定义镜像：选择镜像名称和镜像版本。用户可在“容器镜像服务”设置的镜像。具体操作请参考使用自定义镜像增强作业运行环境。
Flink版本	<p>先选择所属的队列再选择Flink版本。</p> <p>说明 不建议长期混用不同版本的Flink引擎。</p> <ul style="list-style-type: none"> • 长期混用不同版本的Flink引擎会导致代码在新旧版本之间不兼容，影响作业的执行效率。 • 当作业依赖于特定版本的库或组件，长期混用不同版本的Flink引擎可能会导致作业因依赖冲突而执行失败。
委托	<p>使用Flink 1.15版本执行作业时，按需配置自定义委托用于授予DLI访问其他服务的操作权限。</p>

名称	描述
优化参数	<p>用户自定义的优化参数。参数格式为key=value。</p> <p>Flink优化参数支持非敏感的全局变量替换。例如，在“全局配置”>“全局变量”中新增全局变量phase，Flink Jar作业就可以添加优化参数table.optimizer.agg-phase.strategy={{phase}}。</p> <p>Flink 1.15版本支持Flink Jar作业最小化提交，通过在运行优化参数中配置flink.dli.job.jar.minimize-submission.enabled=true可以开启最小化提交功能。</p> <p>说明</p> <p>Flink Jar作业最小化提交是指Flink仅提交作业必须的依赖项，而不是整个Flink环境。通过将非Connector的Flink依赖项（以flink-开头）和第三方库（如Hadoop、Hive、Hudi、Mysql-cdc）的作用域设置为provided，可以确保这些依赖项不会被包含在Jar作业中，从而实现最小化提交，避免依赖包与flink内核中依赖包冲突：</p> <ul style="list-style-type: none"> • 仅Flink 1.15版本支持Flink Jar作业最小化提交。 • Flink相关依赖作用域请使用provided，即在依赖中添加<scope>provided</scope>。主要包含org.apache.flink组下以flink-开头的非Connector依赖。 • Hadoop、Hive、Hudi、Mysql-cdc相关依赖，作用域请使用provided，即在依赖中添加<scope>provided</scope>。 • Flink源代码中只有明确标注了@Public或者@PublicEvolving的才是公开供用户调用的方法，DLI只对这些方法的兼容性做出产品保证。

步骤7 配置计算资源规格参数。

图 8-8 配置参数

* CU数量	<input type="text" value="2"/>	作业占用资源总CU数，需配置与实际占用资
* 管理单元	<input type="text" value="1"/>	
* 并行数	<input type="text" value="1"/>	算子默认最大并行数，优先级比代码中低。；置优先级低于代码中并行数设置。
TaskManager配置	<input type="checkbox"/>	
保存作业日志	<input checked="" type="checkbox"/>	
* OBS桶	<input type="text" value="--请选择OBS路径--"/>	
作业异常告警	<input type="checkbox"/>	
异常自动重启	<input type="checkbox"/>	

表 8-5 参数说明


名称	描述
CU数量	<p>一个CU为1核4G的资源量。CU数量范围为2~10000个。</p> <p>说明 当开启TaskManager配置时，为了优化弹性资源池队列的管理，在您设置“单TM Slot”后，为您自动调整CU数量与实际CU数量一致。</p> <p>CU数量=实际CU数量=max[管理单元和TaskManager的CPU总和，(管理单元和TaskManager的内存总和/4)]</p> <ul style="list-style-type: none"> 管理单元和TaskManager的CPU总和=实际TM数 * 单TM所占CU数 + 管理单元。 管理单元和TaskManager的内存总和= 实际TM数 * 设置的单个TM的内存 + 管理单元内存 如果配置了单 TM Slot数，实际TM数 = 并行数 / 单 TM Slot数。 如果没配置了单 TM Slot数，实际TM数 = (CU数量 - 管理单元) / 单TM所占CU数。 如果没在优化参数配置单个TM的内存和管理单元内存，默认单个TM的内存 = 单TM所占CU数 * 4。管理单元内存 = 管理单元 * 4。 Spark资源并行度由Executor数量和Executor CPU核数共同决定。
管理单元	设置管理单元的CU数。
并行数	<p>作业的并行数是指作业中各个算子的并行执行的子任务的数量，即算子子任务数就是其对应算子的并行度。</p> <p>说明</p> <ul style="list-style-type: none"> 并行数不能大于计算单元（CU数量-管理单元CU数量）的4倍。 并行数应大于用户作业里设置的并发数，否则有可能提交失败。
TaskManager配置	<p>用于设置TaskManager资源参数。</p> <ul style="list-style-type: none"> 勾选后需配置下列参数： <ul style="list-style-type: none"> “单TM所占CU数”：每个TaskManager占用的资源数量。 “单TM Slot”：每个TaskManager包含的Slot数量。 不勾选该参数，，系统自动按照默认值为您配置。 <ul style="list-style-type: none"> “单TM所占CU数”：默认值为1。 “单TM Slot”：默认值为“(并行数 * 单TM所占CU数) / (CU数量 - 管理单元)”。
保存作业日志	<p>设置是否将作业运行时的日志信息保存到OBS桶。</p> <p>勾选“保存作业日志”时，还需要设置OBS桶信息。</p> <p>注意 该参数建议勾选，否则作业运行完成后不会生成运行日志，后续如果作业运行异常则无法获取运行日志进行定位。</p>

名称	描述
开启 Checkpoint	<p>Checkpoint用于定期保存作业状态，开启Checkpoint可以在系统故障时快速恢复指定的作业状态。</p> <p>DLI开启Checkpoint有两种方式：</p> <ul style="list-style-type: none"> 在作业代码中配置Checkpoint相关参数，适用于Flink 1.15及历史Flink版本。 在DLI管理控制台的Jar作业配置界面开启Checkpoint，适用于Flink 1.15及更高的引擎版本。 <p>Flink 1.15版本请勿重复在作业代码和Jar作业配置界面配置Checkpoint相关参数，作业代码中的配置项优先级更高，重复配置可能导致作业在异常重启时使用错误的Checkpoint路径恢复数据，导致恢复失败或数据不一致。</p> <p>勾选“开启Checkpoint”后配置以下参数开启Checkpoint：</p> <ul style="list-style-type: none"> Checkpoint间隔：Checkpoint的时间间隔，单位为秒。 Checkpoint模式：选择Checkpoint的模式： <ul style="list-style-type: none"> At least once：事件至少被处理一次。 Exactly once：事件仅被处理一次。 <p>注意</p> <ul style="list-style-type: none"> 勾选开启Checkpoint后需配置OBS桶参数，用于保存Checkpoint信息，默认Checkpoint保存路径为：“桶名/jobs/checkpoint/作业ID开头的目录”。 开启Checkpoint后，请勿在作业代码中设置Checkpoint参数，作业代码中配置的参数优先级高于界面配置的参数优先级。重复配置可能导致作业在异常重启时使用错误的Checkpoint路径恢复数据，导致恢复失败或数据不一致。 开启Checkpoint后，如果同时勾选了“异常自动重启”，并勾选了“从Checkpoint恢复”，无需再指定“Checkpoint路径”，系统会根据“开启Checkpoint”的配置信息自动指定。
OBS桶	<p>勾选“保存作业日志”或“开启Checkpoint”时，该参数必选。</p> <p>选择OBS桶用于保存用户作业日志信息。如果选择的OBS桶是未授权状态，需要单击“OBS授权”。</p>
作业异常告警	<p>设置是否将作业异常告警信息，如作业出现运行异常或者欠费情况，以SMN的方式通知用户。</p> <p>勾选后需配置下列参数：</p> <p>“SMN主题”：</p> <p>选择一个自定义的SMN主题。如何自定义SMN主题，请参见《消息通知服务用户指南》中“创建主题”章节。</p>

名称	描述
异常自动重启	<p>设置是否启动异常自动重启功能，当作业异常时将自动重启并恢复作业。</p> <p>勾选后需配置下列参数：</p> <ul style="list-style-type: none"> “异常重试最大次数”：配置异常重试最大次数。单位为“次/小时”。 <ul style="list-style-type: none"> 无限：无限次重试。 有限：自定义重试次数。 “从Checkpoint恢复”：从保存的checkpoint恢复作业。勾选该参数后，需要选择“Checkpoint路径”。 <p>“Checkpoint路径”：选择checkpoint保存路径。必须和应用程序中配置的Checkpoint地址相对应。且不同作业的路径不可一致，否则无法获取准确的Checkpoint。</p> <p>说明</p> <ul style="list-style-type: none"> 仅当同时勾选了“开启Checkpoint”时无需再指定“Checkpoint路径”，系统会根据“开启Checkpoint”的配置信息自动指定。 如果未勾选“开启Checkpoint”，需要选择“Checkpoint路径”。

步骤8 单击右上角“保存”，保存作业和相关参数。

步骤9 单击右上角“启动”，进入“启动Flink作业”页面，确认作业规格和费用，单击“立即启动”，启动作业，系统将自动跳转到Flink作业管理页面，新创建的作业将显示在作业列表中，在“状态”列中可以查看作业状态。

- 作业提交成功后，状态将由“提交中”变为“运行中”。运行完成后显示“已完成”。
- 如果作业状态为“提交失败”或“运行异常”，表示作业提交或运行失败。用户可以在作业列表中的“状态”列中，将鼠标移动到状态图标上查看错误信息，单击可以复制错误信息。根据错误信息解决故障后，重新提交。

---结束

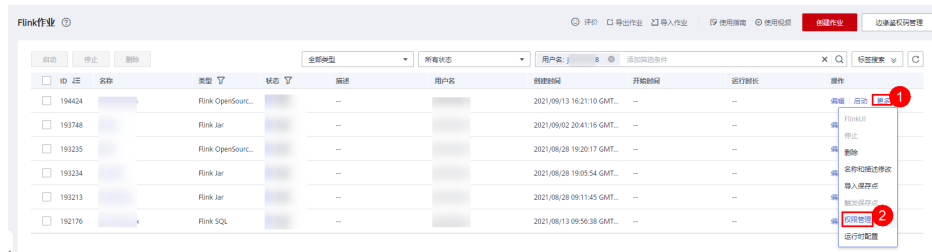
8.4 配置 Flink 作业权限

Flink 作业权限操作场景

- 针对不同用户，可以通过权限设置分配不同的作业，不同用户之间的作业效率互不影响，保障作业性能。
- 管理员用户和作业的所有者拥有所有权限，不需要进行权限设置且其他用户无法修改其作业权限。
- 给新用户设置作业权限时，该用户所在用户组的所属区域需具有Tenant Guest权限。关于Tenant Guest权限的介绍和开通方法，详细参见《[权限策略](#)》和《[统一身份认证服务用户指南](#)》中的[创建用户组](#)。

Flink 作业权限相关操作步骤

1. 在DLI管理控制台的左侧，选择“作业管理”>“Flink作业”。
2. 选择待设置的作业，单击其“操作”列中的“更多”>“权限管理”。“用户权限信息”区域展示了当前具备此作业权限的用户列表。



权限设置有3种场景：为新用户赋予权限，为已有权限的用户修改权限，回收某用户具备的所有权限。

- 为新用户赋予权限
 - 新用户指之前不具备此作业权限的用户。
 - i. 单击“权限信息”右侧的“授权”，弹出“授权”对话框。
 - ii. 填写“用户名”，并勾选对应权限。
 - iii. 单击“确定”，完成新用户的添加。
- 待设置的参数说明如表8-6所示。

图 8-9 Flink 作业授权



表 8-6 Flink 作业授权参数说明

参数名称	描述
用户名	被授权用户的名称。 说明 该用户名称是已存在的IAM用户名称。并且该用户需要登录过华为云，才能进行授权操作。

参数名称	描述
权限设置	<ul style="list-style-type: none"> ● 全选：所有的权限都勾选上。 ● 查看作业详情：查看此作业的作业详情。 ● 更新作业：编辑修改此作业。 ● 删除作业：删除此作业。 ● 启动作业：启动该作业权限。 ● 停止作业：停止该作业。 ● 导出作业：导出该作业。 ● 赋权：当前用户可将作业的权限赋予其他用户。 ● 回收：当前用户可回收其他用户具备的该作业的权限，但不能回收该作业所有者的权限。 ● 查看其他用户具备的权限：当前用户可查看其他用户具备的该作业的权限。

- 为已有权限的用户赋予权限或回收权限。
 - i. 在对应作业“用户权限信息”区域的用户列表中，选择需要修改权限的用户，在“操作”列单击“权限设置”。
 - ii. 在作业“权限设置”对话框中，对当前用户具备的权限进行修改。详细权限描述如表8-6所示。
当“权限设置”中的选项为灰色时，表示您不具备修改此作业权限的权限。可以向管理员用户、作业所有者等具有赋权权限的用户申请“作业的赋权”和“作业权限的回收”权限。
 - iii. 单击“确定”完成权限设置。
- 回收某用户具备的所有权限。
在对应作业“权限信息”区域的用户列表中，选择需要删除权限的用户，在“操作”列单击“回收”。在“回收”对话框中单击“确定”后，此用户将不具备该作业的任意权限。

Flink 作业权限使用说明

- **查看作业详情**
 - 租户以及admin用户可以查看和操作所有作业。
 - 子用户以及拥有只读权限的用户只能查看自己的作业。

说明

他人赋权给该子用户查看权限外的任意权限，则该作业仅显示在作业列表中，但不支持该子用户查看作业详情。

- **启动作业**
用户需要同时拥有队列的提交作业权限以及作业的启动作业权限。
- **停止作业**
用户需要同时拥有队列的停止作业权限以及作业的停止作业权限。
- **删除作业**
 - 如果作业在可删除状态，则用户拥有作业的删除权限即可。

- 如果作业在不可删除状态，用户删除作业时，系统会先停止作业，停止作业权限说明可以参考[停止作业](#)，并且用户还需要拥有作业的删除权限。
- **创建作业**
 - 子用户默认不能创建作业。
 - 创建作业时，用户需要拥有创建作业的权限。目前只有admin用户创建作业的权限，同时用户还需要拥有该作业使用的相关程序包组权限或者程序包权限。
- **编辑作业**

编辑作业时，用户需要拥有更新作业的权限，同时用户还需要拥有该作业使用的相关程序包所属组权限或者程序包权限。

8.5 管理 Flink 作业

8.5.1 查看 Flink 作业详情

创建作业后，您可以在DLI管理控制台查看Flink作业的基本信息、作业详情、任务列表、执行计划等信息。

本节操作介绍怎样查看Flink作业相关信息。

表 8-7 查看 Flink 作业相关信息

类型	说明	操作指导
Flink作业基本信息	包括Flink作业的ID、作业类型、作业执行状态等信息。	查看Flink作业基本信息
Flink作业详情	包括作业的SQL语句和参数设置信息，Jar作业支持查看参数设置信息。	查看Flink作业详情
Flink作业监控	通过云监控服务（CES）查看作业数据输入输出的详细信息。	查看Flink作业监控
Flink作业任务列表	查看作业运行时每个任务的详细信息，例如任务的开始时间、收发字节数和运行时长等。	查看Flink作业任务列表
Flink作业执行计划	了解运行中的作业的算子流向。	查看Flink作业执行计划

查看 Flink 作业基本信息

单击“作业管理 > Flink作业”，进入Flink作业管理页面。Flink作业管理页面显示所有的Flink作业，通过Flink作业列表可以了解Flink作业的基本信息。

表 8-8 Flink 作业基本信息

参数	参数说明
ID	所提交Flink作业ID，由系统默认生成。
名称	所提交Flink作业的名称。
类型	所提交Flink作业的类型。包括： <ul style="list-style-type: none"> ● Flink SQL：Flink SQL作业 ● Flink Jar：Flink Jar作业 ● Flink OpenSource SQL：Flink OpenSource SQL作业
状态	作业的状态信息。具体状态信息以控制台为准。
描述	所提交Flink作业的描述。
用户名	提交作业的用户名称。
创建时间	每个作业的创建时间。
开始时间	Flink作业开始运行的时间。
运行时长	作业运行所消耗的时间。
操作	<ul style="list-style-type: none"> ● 编辑：编辑已经创建好的作业。 ● 启动：启动作业并运行。 ● 更多 <ul style="list-style-type: none"> - FlinkUI：单击后，将跳转至Flink任务运行情况界面。 <p>说明 如果是新建队列，在该队列提交作业后，如果立即单击FlinkUI，因为后台大约需要10分钟创建集群，会导致缓存空的projectID，从而导致无法查看FlinkUI。 建议作业选择使用专属队列，后台集群不会被释放，避免上述问题产生。或者等待作业运行中时再查看FlinkUI，确保集群已经拉好了，不要立即单击FlinkUI。</p> <ul style="list-style-type: none"> - 停止：停止Flink作业。如果该功能置灰，表示当前状态的作业不支持停止。 - 删除：删除作业。 <p>说明 作业删除后不可恢复，请谨慎操作。</p> <ul style="list-style-type: none"> - 名称和描述修改：修改作业名称和描述。 - 导入保存点：导入原实时流计算服务作业导出的数据。 - 触发保存点：“运行中”的作业可以“触发保存点”，保存作业的状态信息。 - 权限管理：查看作业对应的用户权限信息以及对其他用户授权。 - 运行时配置：支持作业在运行时配置作业异常告警和异常自动重启。

查看 Flink 作业详情

用户作业创建完成并保存后，用户可以单击作业名查看作业的详细信息，包括作业的 SQL 语句和参数设置信息，如果是 jar 作业只可以看到参数设置信息。

步骤1 在DLI管理控制台的左侧导航栏中，单击“作业管理”>“Flink作业”，进入Flink作业管理页面。

步骤2 单击需要查看的作业名称，进入“作业详情”页面。

在“作业详情”页签，用户可以查看作业的SQL语句、参数设置信息和总费用。

以某个Flink SQL作业为例进行说明。

表 8-9 参数说明

参数名称	参数说明
类型	作业类型，如Flink SQL作业。
名称	Flink作业名称。
描述	Flink作业描述信息。
状态	Flink作业运行状态。
运行模式	默认独享模式。
Flink 版本	作业所选的Flink作业的版本。
所属队列	Flink作业所属的队列名称。
UDF Jar	配置UDF Jar参数时显示。
优化参数	作业添加了自定义参数配置时显示该参数。
CU数量	作业配置的CU数量。
管理单元	作业配置的管理单元CU数量。
并行数	作业配置的同时运行Flink作业的任务数。
单TM所占CU数	作业配置的每个TaskManager所占CU数量。
单TM Slot数	作业配置的每个TaskManager Slot数量。
OBS桶	作业配置了OBS桶显示。当勾选“开启Checkpoint”和“保存作业日志”参数后，用于保存Checkpoint和作业运行日志。
保存作业日志	开启或关闭。
作业异常告警	开启或关闭。
SMN主题	作业配置的SMN主题名称。当作业勾选了“作业异常告警”参数时显示。
异常自动重启	开启或关闭。
异常重试最大次数	异常重试最大次数为无限时显示为“无限”，其余为设置的具体值。

参数名称	参数说明
从Checkpoint恢复	开启或关闭。
ID	作业ID。
保存点路径	保存点的OBS存储路径。
开启Checkpoint	开启或关闭。
Checkpoint间隔	将作业运行的中间结果保存到OBS的间隔时间，单位为秒。
Checkpoint模式	Checkpoint 模式： <ul style="list-style-type: none"> at_least_once：事件至少被处理一次 exactly_once：事件仅被处理一次
空闲状态保留时长	用于清除GroupBy、RegularJoin、Rank、Depuplicate等算子经过最大保留时间后仍未更新的中间状态，默认设置为1小时。
脏数据策略	作业出现脏数据时的处理策略。作业配置脏数据策略时显示该参数。 忽略。 抛出异常。 保存。
脏数据转储地址	“脏数据策略”选择“保存”时，保存脏数据的OBS路径。
创建时间	作业创建的具体时间。
更新时间	作业最近一次更新的时间。

----结束

查看 Flink 作业监控

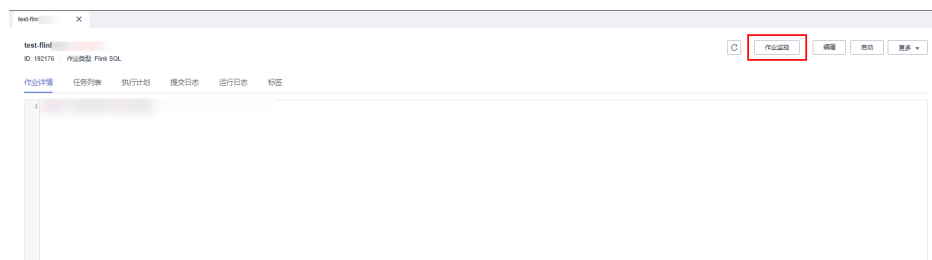
用户可以通过云监控服务（CES）查看作业数据输入输出的详细信息。

步骤1 在DLI管理控制台的左侧导航栏中，单击“作业管理”>“Flink作业”，进入Flink作业管理页面。

步骤2 单击需要查看的作业名称，进入“作业详情”页面。

单击页面右上角的“作业监控”，将跳转至云监控服务（CES）。

图 8-10 作业监控



Flink 作业包含如下监控指标。

表 8-10 Flink 作业监控指标

指标名称	说明
Flink作业数据输入速率	展示用户Flink作业的数据输入速率，供监控和调试使用。单位：条/秒。
Flink作业数据输出速率	展示用户Flink作业的数据输出速率，供监控和调试使用。单位：条/秒。
Flink作业数据输入总数	展示用户Flink作业的数据输入总数，供监控和调试使用。单位：条。
Flink作业数据输出总数	展示用户Flink作业的数据输出总数，供监控和调试使用。单位：条。
Flink作业字节输入速率	展示用户Flink作业每秒输入的字节数。单位：字节/秒。
Flink作业字节输出速率	展示用户Flink作业每秒输出的字节数。单位：字节/秒。
Flink作业字节输入总数	展示用户Flink作业字节的输入总数。单位：字节。
Flink作业字节输出总数	展示用户Flink作业字节的输出总数。单位：字节。
Flink作业CPU使用率	展示用户Flink作业的CPU使用率。单位：%。
Flink作业内存使用率	展示用户Flink作业的内存使用率。单位：%。
Flink作业最大算子延迟	展示用户Flink作业的最大算子延迟时间，单位ms。
Flink作业最大算子反压	展示用户Flink作业的最大算子反压值，数值越大，反压越严重。 0：表示OK 50：表示Low 100：表示High

----结束

查看 Flink 作业任务列表

用户可以查看作业运行时每个任务的详细信息，例如任务的开始时间、收发字节数和运行时长等。

说明

如果数据为零，表示没有从数据源接收到数据。

步骤1 在DLI管理控制台的左侧导航栏中，单击“作业管理”>“Flink作业”，进入Flink作业管理页面。

步骤2 单击需要查看的作业名称，进入“作业详情”页面。

步骤3 在“任务列表”页签，可以查看任务的节点信息。

图 8-11 任务列表



查看算子任务列表，具体参见下表：

表 8-11 算子任务列表参数

参数	说明
名称	算子名称。
持续时间	算子运行的持续时间。
最大并行数	算子中并行的Task的个数。
任务	算子的任务有以下几种： <ul style="list-style-type: none"> ● 红色数字表示已失败的Task个数。 ● 浅灰色数字表示已取消的Task个数。 ● 黄色数字表示取消中的Task个数。 ● 绿色数字表示已完成的Task个数。 ● 蓝色数字表示运行中的Task个数。 ● 天蓝色数字表示部署中的Task个数。 ● 深灰色数字表示排队中的Task个数。
状态	算子任务对应的状态。
反压状态	算子的工作负荷状态。包含如下几种状态： <ul style="list-style-type: none"> ● OK：表示工作负荷正常。 ● LOW：表示工作负荷略高。DLI处理数据的速度比较快。 ● HIGH：表示工作负荷高。源端输入数据的速度比较慢。
时延	指事件从源端算子到达本算子的过程中消耗的时间，单位为毫秒（ms）。
发送的记录数	算子发送数据的记录。
发送的字节数	算子发送的字节数。
接受的字节数	算子接收的字节数。
接受的记录数	算子收到数据的记录。

参数	说明
开始时间	算子运行开始时间。
结束时间	算子运行结束时间。

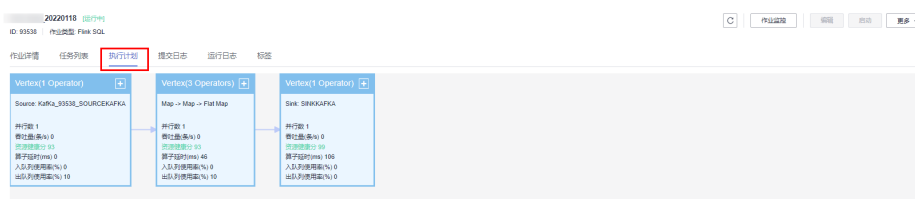
----结束

查看 Flink 作业执行计划

用户通过查看执行计划了解到运行中的作业的算子流向。

- 步骤1** 在DLI管理控制台的左侧导航栏中，单击“作业管理”>“Flink作业”，进入Flink作业管理页面。
- 步骤2** 单击需要查看的作业名称，进入“作业详情”页面。
- 步骤3** 单击“执行计划”，在“执行计划”页签可以查看作业的算子流向。

图 8-12 执行计划



单击对应的节点，在页面右侧显示对应的信息。

- 滚动鼠标滚轮可对流图进行缩放查看。
- 流图展示当前运行作业的实时算子流图信息。

----结束

8.5.2 设置 Flink 作业优先级

操作场景

在实际作业运行中，由于作业的重要程度以及紧急程度不同，需要重点保障重要和紧急的作业正常运行，因此需要满足它们正常运行所需的计算资源。

DLI提供的设置作业优先级功能，可以对每个Flink作业设置作业优先级，当资源不充足时，可以优先满足优先级较高的作业的计算资源。

说明

Flink 1.12及以上版本的作业支持设置作业优先级。

使用须知

- 对于每个作业都允许设置优先级，其取值为1-10，数值越大表示优先级越高。优先满足高优先级作业的计算资源，即如果高优先级作业计算资源不足，则会减少低优先级作业的计算资源

- 通用队列上运行的Flink作业优先级默认为5。
- 作业优先级的调整需要停止作业进行编辑，并提交运行才能生效。
- 对于Flink作业，请参考[开启Flink作业动态扩缩容](#)设置 `flink.dli.job.scale.enable=true`开启动态扩缩容功能，再设置作业优先级。
- 调整作业优先级需要停止作业后编辑，并重新提交运行才能生效。

设置 Flink Opensource SQL 作业优先级

1. 登录DLI管理控制台。
2. 单击“作业管理 > Flink作业”。
3. 选择要待配置的作业，单击操作列下的编辑。
4. 单击“自定义配置”。
5. 在“自定义配置”中输入如下语句，先开启动态扩缩容功能，再设置作业优先级。

说明

对于Flink作业，必须先设置`flink.dli.job.scale.enable=true`开启动态扩缩容功能，再设置作业优先级。

开启动态扩缩容的更多参数设置请参考[开启Flink作业动态扩缩容](#)。

```
flink.dli.job.scale.enable=true  
flink.dli.job.priority=x
```

图 8-13 Flink Opensource SQL 作业配置样例



设置 Flink Jar 作业优先级

在“优化参数”中配置如下参数，其中x为优先级取值。

```
flink.dli.job.priority=x
```

1. 登录DLI管理控制台。
2. 单击“作业管理 > Flink作业”。
3. 选择待配置的作业，单击操作列下的编辑。
4. 在“优化参数”中输入如下语句。先开启动态扩缩容功能，再设置作业优先级。

📖 说明

对于Flink作业，必须先设置`flink.dli.job.scale.enable=true`开启动态扩缩容功能，再设置作业优先级。

开启动态扩缩容的更多参数设置请参考[开启Flink作业动态扩缩容](#)。

```
flink.dli.job.scale.enable=true  
flink.dli.job.priority=x
```

图 8-14 Flink Jar 作业配置样例

优化参数

```
flink.dli.job.scale.enable=true  
flink.dli.job.priority=8
```

8.5.3 开启 Flink 作业动态扩缩容

操作场景

在实际作业运行中，由于作业的数据流量变化，导致所需计算资源不同，造成流量较小时计算资源浪费，流量较大时计算资源不足以满足计算所需。

DLI提供的动态扩缩容功能可以根据当前作业的负载情况，例如：数据输入输出量、数据输入输出速率、反压等情况，动态的调整当前作业所用的计算资源，提升资源利用率。

开启Flink作业动态扩缩容后，系统将根据Flink作业的实际资源需求动态调整资源分配。当弹性资源池中剩余的Pod资源足以支持作业的最小资源需求时，系统将自动减少作业所在节点的数量，确保作业高效运行，同时提高资源的利用效率。

📖 说明

当前仅Flink 1.12版本的作业支持开启动态扩缩容。

使用须知

- 在Flink作业进行动态扩缩容时如果队列资源被抢占，剩余资源不满足作业启动所需资源则可能存在作业无法正常恢复的情况。
- 在Flink作业进行动态扩缩时后台作业需要停止继而从savepoint恢复，因此未恢复成功前，作业无法处理数据。
- 因扩缩容过程中需要触发savepoint，因此必须配置obs桶，并保存日志，同时请注意开启checkpoint。
- 扩缩容检测周期不要设置过小，避免频繁启停作业。
- 扩缩容作业恢复过程中的时间长短受savepoint的大小影响，如果保存点较大，可能恢复时间较慢。
- 如果需要调整动态扩缩容的配置项，则需要停止作业进行编辑，并提交运行才能生效。

操作步骤

Flink作业动态扩缩容适用于Flink Opensource SQL作业和Flink Jar作业。

1. 登录DLI管理控制台。
2. 单击“作业管理 > Flink作业”。
3. 选择要开启动态扩缩容的作业，单击操作列下的编辑。
 - Flink Opensource SQL作业单击“自定义配置”，配置动态扩缩容参数。
 - Flink Jar作业单击“优化参数”框，配置动态扩缩容参数。

表 8-12 动态扩缩容参数说明

名称	默认值	描述
flink.dli.job.scale.enable	false	该配置表示是否开启动态扩缩的功能，即是否允许根据作业的负载调整作业的使用资源量和是否允许DLI根据作业优先级调整作业的使用资源量。 当前配置为false时，表示不允许。 当前配置为true时，表示允许。 默认值为false。
flink.dli.job.scale.interval	30	该配置表示检测当前作业是否需要动态扩缩的时间周期，其单位为分钟，默认值为30。例如30表示每隔30分钟进行一次检测，判断是否需要作业使用资源量进行扩缩。 注意：只有当用户开启动态扩缩时，该配置才有意义。
flink.dli.job.cu.max	用户CU初始值	该配置表示当前作业在进行动态扩缩时允许使用的最大CU数，如果用户未配置则默认值为该作业的初始总CU数。 注意：该配置值不能小于用户配置的总CU数，且只有当用户开启动态扩缩时，该配置才有意义。
flink.dli.job.cu.min	2	该配置表示当前作业在进行动态扩缩时允许使用的最小CU数，其默认值为2。 注意：该配置值不能大于用户配置的总CU数，且只有当用户开启动态扩缩时，该配置才有意义。

8.5.4 查询 Flink 作业日志

操作场景

DLI作业桶用于存储DLI作业运行过程中产生的临时数据，例如：作业日志、作业结果。

本节操作指导您在DLI管理控制台配置DLI作业桶，并查看Flink作业日志的操作方法。

使用须知

- 请勿将该DLI作业桶绑定的OBS桶用作其它用途，避免出现作业结果混乱等问题。
- DLI作业要由用户主账户统一设置及修改，子用户无权限。
- 不配置DLI作业桶无法查看作业日志。
- 您可以通过配置桶的生命周期规则，定时删除桶中的对象或者定时转换对象的存储类别。
- DLI的作业桶设置后请谨慎修改，否则可能会造成历史数据无法查找。

前提条件

配置前，请先购买OBS桶或并行文件系统。大数据场景推荐使用并行文件系统，并行文件系统（Parallel File System）是对象存储服务（Object Storage Service，OBS）提供的一种经过优化的高性能文件系统，提供毫秒级别访问时延，以及TB/s级别带宽和百万级别的IOPS，能够快速处理高性能计算（HPC）工作负载。

并行文件系统的详细介绍和使用说明，请参见《[并行文件系统特性指南](#)》。

配置 DLI 作业桶


1. 在DLI控制台左侧导航栏中单击“全局配置 > 工程配置”。
2. 在“工程配置”页面，选择“DLI作业桶”，单击  配置桶信息。

图 8-15 工程配置




3. 单击  打开桶列表。
4. 选择用于存放DLI作业临时数据的桶，并单击“确定”。完成设置后DLI作业运行过程中产生的临时数据将会存储在该OBS桶中。

图 8-16 设置 DLI 作业桶

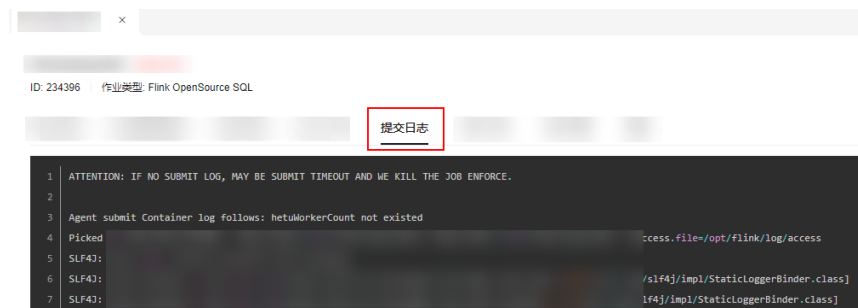


查看 Flink 作业提交日志

用户可以通过查看提交日志排查提交作业异常的故障。

- 步骤1** 在DLI管理控制台的左侧导航栏中，单击“作业管理”>“Flink作业”，进入Flink作业管理页面。
- 步骤2** 单击需要查看的作业名称，进入“作业详情”页面。
- 步骤3** 在“提交日志”页签，可以查看提交作业的过程信息。

图 8-17 提交日志



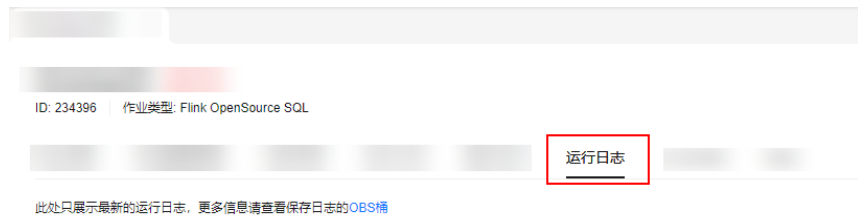
----结束

查看 Flink 作业运行日志

用户可以通过查看运行日志排查作业运行异常的故障。

- 步骤1** 在DLI管理控制台的左侧导航栏中，单击“作业管理”>“Flink作业”，进入Flink作业管理页面。
- 步骤2** 单击需要查看的作业名称，进入“作业详情”页面。
- 步骤3** 在“运行日志”页签，可以查看运行中作业的JobManager和TaskManager信息。

图 8-18 Flink 作业运行日志



JobManager和TaskManager信息每分钟刷新一次，默认展示最近一分钟的运行日志。如果作业配置了保存作业日志的OBS桶，更多历史日志信息可以到保存日志的OBS桶中下载查看。

📖 说明

在OBS中，上传文件的具体方式和要求可以参考《对象存储服务快速入门》>“[上传对象](#)”。如果作业没有运行，则无法查看TaskManager信息。

----结束

8.5.5 Flink 作业常用操作

用户创建了新作业后，需要根据用户的实际需求对作业进行操作，包括编辑作业基本信息，启停作业、导入/导出作业等。

编辑作业

用户可以对已经创建的作业进行编辑，如修改SQL语句、作业名称和描述、作业配置信息等。

步骤1 在DLI管理控制台的左侧导航栏中，单击“作业管理”>“Flink作业”，进入Flink作业管理页面。

步骤2 在需要编辑作业对应的“操作”列中，单击“编辑”，进入作业编辑页面。

步骤3 根据实际需求编辑作业。

具体请参考[创建Flink OpenSource SQL作业](#)，[创建Flink Jar作业](#)。

----结束

启动作业

用户可以启动已创建保存的作业或已经停止的作业。

步骤1 在DLI管理控制台的左侧导航栏中，单击“作业管理”>“Flink作业”，进入Flink作业管理页面。

步骤2 启动作业，有以下两种方式：

- 启动单个作业
选择一个作业，在对应的“操作”列中，单击“启动”。
也可以在作业列表中，勾选一个作业，单击作业列表左上方的“启动”。
- 批量启动作业
勾选多个作业，单击作业列表左上方的“启动”，可以启动多个作业。

单击“启动”后，跳转至“作业配置清单”页面。

步骤3 在“作业配置清单”页面，确认作业信息及价格，如果无误，单击“立即启动”。

作业启动后，可在对应作业的“状态”列中查看运行成功或失败。

----结束

停止作业

当用户不需要运行某个作业时，用户可以将状态为“运行中”和“提交中”的作业停止。

步骤1 在DLI管理控制台的左侧导航栏中，单击“作业管理”>“Flink作业”，进入Flink作业管理页面。


步骤2 停止作业，有以下两种方式：

- 停止单个作业
选择需要停止的作业，在对应的“操作”列中，单击“更多 > 停止”。
也可以在作业列表中，勾选一个作业，单击作业列表左上方的“停止”。

- 批量停止作业
勾选多个需要停止作业，单击作业列表左上方的“停止”。可以停止多个作业。

步骤3 在弹出的“停止作业”窗口中，单击“确认”，停止作业。

图 8-19 停止作业

 确认要停止以下1个作业吗?

名称	状态	描述
	● 提交中	--

触发保存点 

说明

- 在停止作业之前，用户可以触发保存点，保存作业的状态信息。当该作业再次启动时用户可以选择是否从保存点恢复。
- 勾选“触发保存点”表示创建保存点。不勾选“触发保存点”表示不创建保存点。默认不创建保存点。
- 保存点的生命周期从触发保存点并停止作业开始，重启作业后结束。保存点在重启作业后自动删除，不会一直保存。

停止作业过程中，在作业列表的“状态”列中将显示作业状态，说明如下：

- 如果在“状态”中显示“停止中”，表示正在停止作业。
- 如果在“状态”中显示“已停止”，表示停止作业成功。
- 如果在“状态”中显示“停止失败”，表示停止作业失败。

----结束

删除作业

当用户不再需要使用某个作业时，可以参考如下操作删除该作业。作业删除后，将不可恢复，请谨慎操作。

步骤1 在DLI管理控制台的左侧导航栏中，单击“作业管理”>“Flink作业”，进入Flink作业管理页面。

步骤2 删除作业，有以下两种方式：

- 删除单个作业
在需要删除作业对应的“操作”列中，单击“更多 > 删除”，弹出“删除作业”页面。
也可以在作业列表中，勾选一个作业，单击作业列表左上方的“删除”，弹出“删除作业”页面。
- 批量删除作业
勾选多个需要删除作业，单击作业列表左上方的“删除”，弹出“删除作业”页面，可以删除多个作业。

步骤3 单击“确定”，完成作业的删除。

----结束

导出作业

用户可以将所创建的Flink作业导出至OBS桶中。

适用于当用户切换区域、项目或用户时，需要创建相同的作业，而作业比较多的情况。此时，不需要重新创建作业，只需要将原有的作业导出，再在新的区域、项目或者使用新的用户登录后，导入作业即可。

📖 说明

切换项目或用户时，需要对新项目或用户授权，具体请参考[配置Flink作业权限](#)。

步骤1 在DLI管理控制台的左侧导航栏中，单击“作业管理”>“Flink作业”，进入Flink作业管理页面。

步骤2 单击右上角“导出作业”，打开“导出作业”对话框。

图 8-20 导出作业



步骤3 选择保存作业的OBS桶。单击“下一步”。

步骤4 选择待导出的作业。

默认导出所有作业，也可以勾选“自定义导出”选择需要导出的作业。

步骤5 单击“确认导出”，完成导出作业。

----结束

导入作业

用户可以将保存在OBS桶中的Flink作业配置文件导入至DLI的Flink作业管理中。

适用于当用户切换区域、项目或用户时，需要创建相同的作业，而作业比较多的情况。此时，不需要重新创建作业，只需要将原有的作业导出，再在新的区域、项目或者使用新的用户登录后，导入作业即可。

如果需要导入自建的作业，建议使用创建作业的功能。

具体请参考[创建Flink OpenSource SQL作业](#)，[创建Flink Jar作业](#)。

📖 说明

- 切换项目或用户时，需要对新项目或用户授权，具体请参考[配置Flink作业权限](#)。
- 仅支持导入与从DLI导出的Flink作业相同数据格式的作业。

步骤1 在DLI管理控制台的左侧导航栏中，单击“作业管理”>“Flink作业”，进入Flink作业管理页面。

步骤2 单击右上角“导入作业”，打开“导入作业”对话框。

步骤3 选择需导入的作业配置文件的完整OBS路径。单击“下一步”。

步骤4 配置同名作业策略。单击“下一步”。

- 勾选“配置同名替换”，如果待导入的作业名已存在，则覆盖已存在的作业配置，并且作业状态重置为草稿。
- 不勾选“配置同名替换”，如果待导入的作业名已存在，则不导入同名作业的配置。

步骤5 确认“配置文件”和“同名作业策略”配置无误。单击“确认导入”，完成导入作业。

----结束

修改 Flink 作业名称和描述信息

用户可以根据需要修改作业名称和描述。

步骤1 在DLI管理控制台的左侧导航栏中，单击“作业管理”>“Flink作业”，进入Flink作业管理页面。

步骤2 在需要修改名称和描述的作业对应的“操作”列中，单击“更多 > 名称和描述修改”，弹出“属性修改”页面。修改作业名称和描述。

步骤3 单击“确定”完成修改。

----结束

触发保存点

在停止作业前，您可以先触发保存点，保存作业的状态信息。当该作业再次启动时，您可以选择是否从最近的保存点快速恢复作业。

步骤1 在DLI管理控制台的左侧导航栏中，单击“作业管理”>“Flink作业”，进入Flink作业管理页面。

步骤2 选择计划停止的作业，单击“更多 > 触发保存点”，选择保存点的存储路径。

步骤3 单击“确定”完成保存。

----结束

说明

- 状态为“运行中”的作业可以“触发保存点”，保存作业的状态信息。
- 保存点的生命周期从触发保存点并停止作业开始，重启作业后结束。保存点在重启作业后自动删除，不会一直保存。

导入保存点

Flink作业可以根据导入的保存点来恢复作业状态。

步骤1 在DLI管理控制台的左侧导航栏中，单击“作业管理”>“Flink作业”，进入Flink作业管理页面。

步骤2 选择计划停止的作业，单击“更多 > 导入保存点”，选择保存点的存储路径。

步骤3 单击“确定”导入保存点。

----结束

运行时配置

用户可通过选择“运行时配置”配置作业异常告警和重启选项。

说明

支持Flink OpenSource SQL作业和Flinkjar作业。

1. 在对应Flink作业操作列的“更多 > 运行时配置”。
2. 在“运行时配置”页面配置以下参数。

图 8-21 运行时配置

运行时配置

名称 testflinkjar

作业异常告警

* SMN主题 DLI_fink_info
前往消息通知服务进行设置。

异常自动重启

异常重试最大次数 无限 有限

从Checkpoint恢复

* Checkpoint路径 dli-test-obs01/
必须和应用程序中配置的Checkpoint地址相对应。另注意不同作业的路径不可一致，否则无法获取准确的Checkpoint。

确定 取消

表 8-13 作业运行参数说明

参数	参数说明
名称	作业的名称。

参数	参数说明
作业异常告警	<p>设置是否将作业异常告警信息，如作业出现运行异常或者欠费情况，以SMN的方式通知用户。</p> <p>勾选后需配置下列参数：</p> <p>“SMN主题”：</p> <p>选择一个自定义的SMN主题。如何自定义SMN主题，请参见《消息通知服务用户指南》中“创建主题”章节。</p>
异常自动重启	<p>设置是否启动异常自动重启功能，当作业异常时将自动重启并恢复作业。</p> <p>勾选后需配置下列参数：</p> <ul style="list-style-type: none"> “异常重试最大次数”：配置异常重试最大次数。单位为“次/小时”。 <ul style="list-style-type: none"> 无限：无限次重试。 有限：自定义重试次数。 “从Checkpoint恢复”：从已保存的checkpoint恢复作业。 <p>说明</p> <p>Flink SQL作业和Flink OpenSource SQL作业不支持配置该参数。勾选该参数后，Flinkjar作业还需要选择“Checkpoint路径”。</p> <p>“Checkpoint路径”：选择checkpoint保存路径。必须和应用程序中配置的Checkpoint地址相对应。且不同作业的路径不可一致，否则无法获取准确的Checkpoint。</p>

8.6 管理 Flink 作业模板

Flink模板包括样例模板和自定义模板。用户可以在已有的样例模板中进行修改，来实现实际的作业逻辑需求，节约编辑SQL语句的时间。也可以根据自己的习惯和方法自定义作业模板，方便后续可以直接调用或修改。

Flink模板管理主要包括如下功能：

- [Flink SQL样例模板](#)
- [Flink OpenSource SQL样例模板](#)
- [自定义模板](#)
- [新建模板](#)
- [基于模板新建作业](#)
- [修改模板](#)
- [删除模板](#)

Flink SQL 样例模板

Flink SQL样例模板列表显示已有的Flink SQL样例作业模板，Flink SQL样例模板列表参数说明如表 1所示。

已有样例模板的具体场景以控制台为准。

表 8-14 Flink SQL 样例模板列表参数

参数	参数说明
名称	模板名称，只能由英文、中文、数字、中划线和下划线组成，并且长度为1~64个字符。
描述	模板的相关描述，且长度为0~512个字符。
操作	“创建作业”：直接在该模板下创建作业，创建完后，系统跳转到“作业管理”下的作业编辑页面。

Flink OpenSource SQL 样例模板

Flink OpenSource SQL样例模板列表显示已有的Flink OpenSource SQL样例作业模板，Flink OpenSource SQL样例模板列表参数说明如表 1所示。

表 8-15 Flink OpenSource SQL 样例模板列表参数

参数	参数说明
名称	模板名称，只能由英文、中文、数字、中划线和下划线组成，并且长度为1~64个字符。
描述	模板的相关描述，且长度为0~512个字符。
操作	“创建作业”：直接在该模板下创建作业，创建完后，系统跳转到“作业管理”下的作业编辑页面。

当前已有的样例模板包括如下场景：

- 利用地址信息的维表生成订单信息宽表
- 实时统计每天成交额、订单数和支付人数等指标
- 统计实时点击量最高的商品

自定义模板

自定义模板列表显示所有的jar作业模板，自定义模板列表参数说明如表 1所示。

表 8-16 自定义模板列表参数

参数	参数说明
名称	模板名称，只能由英文、中文、数字、中划线和下划线组成，并且长度为1~64个字符。

参数	参数说明
类型	模板类型。 <ul style="list-style-type: none"> • Flink SQL作业模板 • Flink OpenSource SQL作业模板
描述	模板的相关描述，且长度为0~512个字符。
创建时间	创建模板的时间。
更新时间	最后修改模板的时间。
操作	<ul style="list-style-type: none"> • “编辑”：对已经创建好的模板进行修改。 • “创建作业”：直接在该模板下创建作业，创建完后，系统跳转到“作业管理”下的作业编辑页面。 • 更多： <ul style="list-style-type: none"> - “删除”：将已经创建的模板删除。 - “标签”：查看或添加标签。

新建模板

创建作业模板，有以下四种方法。

- 进入“作业模板”页面新建模板。
 - a. 在DLI管理控制台的左侧导航栏中，单击“作业模板”>“Flink模板”。
 - b. 单击页面右上角“创建模板”，弹出“创建模板”页面。
 - c. 输入“名称”和“描述”。

图 8-22 新建 Flink 模板

创建模板
×

类型

* 名称

描述

标签

如果您需要使用同一标签识别多种云资源，即所有服务均可在标签输入框下拉选择同一标签，建议在TMS中创建预定义标签。 [查看预定义标签](#) C

在下方键/值输入框输入内容后单击添加，即可将标签加入此处

您还可以添加10个标签。

表 8-17 模板配置信息

参数	参数说明
类型	模板类型。 <ul style="list-style-type: none"> • Flink SQL作业模板 • Flink OpenSource SQL作业模板
名称	模板名称，只能由字母、中文、数字、中划线和下划线组成，并且长度为1~64个字符。 说明 模板名称必须是唯一的。
描述	模板的相关描述，且长度为0~512字符。

参数	参数说明
标签	<p>使用标签标识云资源。包括标签键和标签值。如果您需要使用同一标签标识多种云资源，即所有服务均可在标签输入框下拉选择同一标签，建议在标签管理服务（TMS）中创建预定义标签。</p> <p>如您的组织已经设定DLI的相关标签策略，则需按照标签策略规则为资源添加标签。标签不符合标签策略的规则，则可能会导致资源创建失败，请联系组织管理员了解标签策略详情。具体请参考《标签管理服务用户指南》。</p> <p>说明</p> <ul style="list-style-type: none"> 最多支持20个标签。 一个“键”只能添加一个“值”。 每个资源中的键名不能重复。 标签键：在输入框中输入标签键名称。 <p>说明</p> <p>标签的键的最大长度为128个字符，标签的键可以包含任意语种字母、数字、空格和_ . : + @ ,但首尾不能含有空格，不能以_sys_开头。</p> <ul style="list-style-type: none"> 标签值：在输入框中输入标签值。 <p>说明</p> <p>标签值的最大长度为255个字符，标签的值可以包含任意语种字母、数字、空格和_ . : + @ 。</p>

- d. 单击“确定”，进入“编辑”页面。
模板编辑页面参数说明参考[表8-18](#)。

表 8-18 编辑模板参数说明

功能	描述
名称	可以修改模板名称。
描述	可以修改模板描述。
保存方式	<ul style="list-style-type: none"> 修改：将修改保存至当前的模板中。 新增：将修改另存为新的模板。
SQL语句编辑区域	输入详细的SQL语句，实现业务逻辑功能。SQL语句的编写请参考《 数据湖探索SQL语法参考 》。
保存	保存修改。
创建作业	使用当前模板创建作业。
格式化	对SQL语句进行格式化，将SQL语句格式化后，需要重新编辑SQL语句。
主题设置	更改字体大小、自动换行、页面风格（黑色底或白色底）等配置。

- e. 在SQL语句编辑区域，输入SQL语句，实现业务逻辑功能。SQL语句的编写请参考《[数据湖探索SQL语法参考](#)》。
 - f. SQL编辑完成后，单击右上角的“保存”，完成创建模板。
 - g. （可选）如果不需要进行修改，也可以单击右上角的“创建作业”基于当前模板创建作业。创建作业请参考[创建Flink Jar作业](#)和[创建Flink OpenSource SQL作业](#)。
- 基于现有作业模板新建模板
 - a. 在DLI管理控制台的左侧导航栏中，单击“作业模板”>“Flink模板”，单击“自定义模板”页签。
 - b. 在自定义模板列表中，单击所需作业模板“操作”列中的“编辑”，进入“模板编辑”页面。
 - c. 修改完成后，“保存方式”选择“新增”。
 - d. 单击右上角“保存”，完成另存一个新模板。
 - 基于新建作业新建模板
 - a. 在DLI管理控制台的左侧导航栏中，单击“作业管理”>“Flink作业”，进入“Flink作业”页面。
 - b. 单击右上角“创建作业”，弹出“创建作业”页面。
 - c. 配置作业信息，输入“名称”和“描述”，选择“模板”。
 - d. 单击“确认”，进入“作业编辑”页面。
 - e. SQL编辑完成后，单击“设为模板”，弹出“设为模板”窗口。
 - f. 输入“名称”和“描述”，单击“确认”，完成另存一个新模板。
 - 基于现有作业新建模板
 - a. 在DLI管理控制台的左侧导航栏中，单击“作业管理”>“Flink作业”，进入“Flink作业”页面。
 - b. 在作业列表中，选择一个需要设置为模板的作业，在“操作”列单击“编辑”，进入“作业编辑”页面。
 - c. SQL编辑完成后，单击“设为模板”，弹出“设为模板”窗口。
 - d. 输入“名称”和“描述”，单击“确认”，完成另存一个新模板。

基于模板新建作业

用户可以基于样例模板或者自定义模板新建作业。

1. 在DLI管理控制台的左侧导航栏中，单击“作业模板”>“Flink模板”。
2. 在样例模板列表中，单击对应模板“操作”列中的“创建作业”。创建作业请参考[创建Flink OpenSource SQL作业](#)和[创建Flink Jar作业](#)。

修改模板

用户创建完自定义模板后，可以根据实际需求修改自定义模板。样例模板不支持修改，但是可以查看。

1. 在DLI管理控制台的左侧导航栏中，单击“作业模板”>“Flink模板”，单击“自定义模板”页签。
2. 在自定义模板列表中，选择一个需要修改的模板，单击模板名称或该模板“操作”列中的“编辑”，进入“编辑”页面。

3. 在SQL语句编辑区，根据需要修改SQL语句。
4. “保存方式”选择“修改”。
5. 单击右上角“保存”，保存当前模板修改的内容。

删除模板

用户可以根据需求删除不需要的自定义模板，不支持删除样例模板。模板删除后无法恢复，请谨慎操作。

1. 在DLI管理控制台的左侧导航栏中，单击“作业模板”>“Flink模板”，单击“自定义模板”页签。
2. 在自定义模板列表中，勾选需要删除的模板，支持多选，单击自定义模板列表左上方的“删除”。
用户也可以在自定义模板列表中，勾选需要删除的模板，单击“操作”栏中“更多 > 删除”，删除对应的模板。
3. 在弹出的删除确认窗口中，单击“是”。

8.7 添加 Flink 作业标签

标签是用户自定义的、用于标识云资源的键值对，它可以帮助用户对云资源进行分类和搜索。标签由标签“键”和标签“值”组成。

DLI支持对Flink作业添加标签。如果想对Flink作业添加如项目名称、业务类别、背景信息等相关信息的标识，用户可以通过添加标签来实现。如果用户在其他云服务中使用了标签，建议用户为同一个业务所使用的云资源创建相同的标签键值对以保持一致性。

DLI支持以下两类标签：

- 资源标签：在DLI中创建的非全局的标签。
- 预定义标签：在标签管理服务（简称TMS）中创建的预定义标签，属于全局标签。

有关预定义标签的更多信息，请参见《[标签管理服务用户指南](#)》。

如您的组织已经设定DLI的相关标签策略，则需按照标签策略规则为资源添加标签。标签如果不符合标签策略的规则，则可能会导致资源创建失败，请联系组织管理员了解标签策略详情。

本章节包含如下内容：

- [管理作业标签](#)
- [根据标签查找作业](#)

管理作业标签

本节介绍如何为作业添加标签、修改标签和删除标签。

- 步骤1** 在DLI管理控制台的左侧导航栏中，单击“作业管理”>“Flink作业”，进入Flink作业管理页面。
- 步骤2** 单击需要查看的作业名称，进入“作业详情”页面。
- 步骤3** 单击“标签”页签，显示当前作业的标签信息。

图 8-23 管理作业标签



步骤4 单击“添加/编辑标签”，弹出“添加/编辑标签”对话框。

步骤5 在“添加/编辑标签”对话框中配置标签参数。

图 8-24 添加标签



表 8-19 标签配置参数

参数	参数说明
标签键	<p>您可以选择：</p> <ul style="list-style-type: none"> 在输入框的下拉列表中选择预定义标签键。 如果添加预定义标签，用户需要预先在标签管理服务中创建好预定义标签，然后在“标签键”的下拉框中进行选择。用户可以通过单击“查看预定义标签”进入标签管理服务的“预定义标签”页面，然后单击“创建标签”来创建新的预定义标签。 具体请参见《标签管理服务用户指南》中的“创建预定义标签”章节。 在输入框中输入标签键名称。 <p>说明 标签的键的最大长度为128个字符，标签的键可以包含任意语种字母、数字、空格和_ . : +-@ ,但首尾不能含有空格，不能以_sys_开头。</p>

参数	参数说明
标签值	<p>您可以选择：</p> <ul style="list-style-type: none"> 在输入框的下拉列表中选择预定义标签值。 在输入框中输入标签值。 <p>说明 标签值的最大长度为255个字符，标签的值可以包含任意语种字母、数字、空格和_ . : + - @ 。</p>

📖 说明

- 最多支持20个标签。
- 一个“键”只能添加一个“值”。
- 每个资源中的键名不能重复。

步骤6 单击“确定”。

步骤7 （可选）在标签列表中，单击“操作”列中“删除”可对选中的标签进行删除。

----结束

根据标签查找作业

对于已经添加过标签的作业，用户可以通过设置标签过滤条件进行搜索，以便快速查找找到作业。

步骤1 在DLI管理控制台的左侧导航栏中，单击“作业管理”>“Flink作业”，进入Flink作业管理页面。

步骤2 单击页面右上角的搜索框，选择“标签”。

图 8-25 标签搜索



步骤3 在标签搜索中，根据提示选择“标签键”和标签值。如果没有可用的标签键和值，请先为作业创建标签，具体参见[管理作业标签](#)。

步骤4 在搜索框中继续选择其他标签，可添加不同标签组合搜索。支持最多20个不同标签的组合搜索，且多个不同标签之间为与的关系。

步骤5 单击搜索按钮，在作业列表中将显示查找到的作业。

----结束

9 使用 DLI 提交 Spark 作业

9.1 创建 Spark 作业

Spark作业编辑页面支持执行Spark作业，为用户提供全托管式的Spark计算服务。

在总览页面，单击Spark作业右上角的“创建作业”，或在Spark作业管理页面，单击右上角的“创建作业”，均可进入Spark作业编辑页面。

进入Spark作业编辑页面，页面会提示系统将创建DLI临时数据桶。该桶用于存储使用DLI服务产生的临时数据，例如：作业日志、作业结果等。如果不创建该桶，将无法查看作业日志。可以通过[配置生命周期规则](#)实现定时删除OBS桶中的对象或者定时转换对象的存储类别。桶名称为系统默认。

如果不需要创建DLI临时数据桶，并且希望不再收到该提示，可以勾选“下次不再提示”并单击“取消”。

前提条件

- 请先将所要依赖的程序包通过“数据管理>程序包管理”页面上传至对应的OBS桶中。具体操作请参考[创建DLI程序包](#)。
- 创建Spark作业，访问其他外部数据源时，如访问OpenTSDB、HBase、Kafka、DWS、RDS、CSS、CloudTable、DCS Redis、DDS等，需要先创建跨源连接，打通作业运行队列到外部数据源之间的网络。
 - 当前Spark作业支持访问的外部数据源详情请参考[DLI常用跨源分析开发方式](#)。
 - 创建跨源连接操作请参见[配置DLI与数据源网络连通（增强型跨源连接）](#)。
创建完跨源连接后，可以通过“资源管理 > 队列管理”页面，单击“操作”列“更多”中的“测试地址连通性”，验证队列到外部数据源之间的网络连通是否正常。详细操作可以参考[测试地址连通性](#)。

操作步骤

1. 在DLI管理控制台的左侧导航栏中，单击“作业管理” > “Spark作业”，进入“Spark作业”页面。
单击右上角的“创建作业”，在Spark作业编辑窗口，可以选择使用“表单模式”或者“API模式”进行参数设置。

以下以“表单模式”页面进行说明，“API模式”即采用API接口模式设置参数及参数值，具体请参考《[数据湖探索API参考](#)》。

2. 选择运行队列。
在下拉列表中选择要使用的队列。
3. 选择Spark版本。在下拉列表中选择支持的Spark版本，推荐使用最新版本。

📖 说明

不建议长期混用不同版本的Spark/Flink引擎。

- 长期混用不同版本的Spark/Flink引擎会导致代码在新旧版本之间不兼容，影响作业的执行效率。
- 当作业依赖于特定版本的库或组件，长期混用不同版本的Spark/Flink引擎可能会导致作业因依赖冲突而执行失败。

4. 作业配置。
参考[表9-1](#)配置作业参数。

图 9-1 Spark 作业配置

The screenshot shows the '作业配置' (Job Configuration) form. It contains the following fields and options:

- 作业名称 (--name):** test-catconfigfile
- * 应用程序:** obs://ai-test/www/Test/catConfigFile.py
- 主类 (--class):** 应用程序为jar包时，主类名不能为空
- 应用程序参数:** 多个参数请以Enter键分隔
- Spark参数 (--conf):** 请输入格式为key=value的参数，多个参数请以Enter键分隔
- 作业特性:** 基础型 (selected), AI增强型, 自定义镜像
- 依赖jar包 (--jars):** 生产业务时建议使用DLI程序包; 开发调试时使用OBS路径，多个参数请以Enter键分隔
- 依赖python文件 (--py-files):** 生产业务时建议使用DLI程序包; 开发调试时使用OBS路径，多个参数请以Enter键分隔
- 其他依赖文件 (--files):** 生产业务时建议使用DLI程序包; 开发调试时使用OBS路径，多个参数请以Enter键分隔
- 依赖分组:** 生产业务时建议使用DLI程序包
- 访问元数据:** 是 (selected), 否
- 是否重试:** 是 (selected), 否
- 高级配置:** 暂不配置, 现在配置

表 9-1 作业配置参数说明

参数名称	参数描述
作业名称 (-- name)	设置作业名称。
应用程序	<p>选择需要执行的程序包。包括 “.jar” 和 “.py” 两种类型。</p> <p>Jar包的管理方式：</p> <ul style="list-style-type: none"> • 上传OBS管理程序包：提前将对应的jar包上传至OBS桶中。并在此处选择对应的OBS路径。 • 上传DLI管理程序包：提前将对应的jar包上传至OBS桶中，并在DLI管理控制台的“数据管理>程序包管理”中创建程序包，具体操作请参考创建DLI程序包。 <p>Spark3.3.x及以上版本只能选择OBS路径下的程序包。</p>
主类 (-- class)	输入主类名称。当应用程序类型为 “.jar” 时，主类名称不能为空。
应用程序参数	<p>用户自定义参数，多个参数请以Enter键分隔。</p> <p>应用程序参数支持全局变量替换。例如，在“全局配置” > “全局变量”中新增全局变量key为batch_num，可以使用 {{batch_num}}，在提交作业之后进行变量替换。</p>
Spark参数 (--conf)	<p>以 “key=value” 的形式设置提交Spark作业的属性，多个参数以Enter键分隔。</p> <p>Spark参数value支持全局变量替换。例如，在“全局配置” > “全局变量”中新增全局变量key为custom_class，可以使用 "spark.sql.catalog"={{custom_class}}，在提交作业之后进行变量替换。</p> <p>说明</p> <ul style="list-style-type: none"> • Spark作业不支持自定义设置jvm垃圾回收算法。 • 如果选择Spark版本为3.1.1时，需在Spark参数 (--conf)配置跨源作业的依赖模块。配置样例请参考表9-2。
作业特性	<p>“所属队列”选择CCE队列时，设置该参数。表示用户作业使用的Spark镜像类型，具体说明如下：</p> <ul style="list-style-type: none"> • 基础型：DLI提供的基础镜像，运行非AI相关作业时选择“基础型”。 • 自定义镜像：自定义的Spark镜像，需要选择“容器镜像服务”中设置的镜像名称及版本。
依赖jar包 (--jars)	运行spark作业依赖的jars。可以输入jar包名称，也可以输入对应jar包文件的OBS路径，格式为：obs://桶名/文件夹路径名/包名。
依赖python文件 (--py-files)	运行spark作业依赖的py-files。可以输入Python文件名称，也可以输入Python文件对应的OBS路径，格式为：obs://桶名/文件夹路径名/文件名。

参数名称	参数描述
其他依赖文件 (--files)	运行spark作业依赖的其他files。可以输入依赖文件名称，也可以输入对应的OBS路径，格式为：obs://桶名/文件夹路径名/文件名。
依赖分组	在创建程序包时，如果选择了分组，在此处选择对应的分组，则可以同时选中该分组中的所有程序包和文件。创建程序包操作请参考 创建DLI程序包 。
访问元数据	是否通过Spark作业访问元数据。具体请参考《 数据湖探索开发指南 》
是否重试	作业失败后是否进行重试。 选择“是”需要配置以下参数： “最大重试次数”：设置作业失败重试次数，最大值为“100”。
高级配置	<ul style="list-style-type: none"> • 暂不配置 • 现在配置：包括以下两项参数 <ul style="list-style-type: none"> - 选择依赖资源：具体参数请参考表9-3。 - 计算资源规格：具体参数请参考表9-4。

表 9-2 Spark 参数 (--conf)配置跨源作业的依赖模块说明

数据源类型	样例
CSS	spark.driver.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/css/* spark.executor.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/css/*
DWS	spark.driver.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/dws/* spark.executor.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/dws/*
HBase	spark.driver.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/hbase/* spark.executor.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/hbase/*
OpenTSDB	park.driver.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/opentsdb/* spark.executor.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/opentsdb/*

数据源类型	样例
RDS	spark.driver.extraClassPath=/usr/share/extension/dli/ spark-jar/datasource/rds/* spark.executor.extraClassPath=/usr/share/extension/dli/ spark-jar/datasource/rds/*
Redis	spark.driver.extraClassPath=/usr/share/extension/dli/ spark-jar/datasource/redis/* spark.executor.extraClassPath=/usr/share/extension/dli/ spark-jar/datasource/redis/*
Mongo	spark.driver.extraClassPath=/usr/share/extension/dli/ spark-jar/datasource/mongo/* spark.executor.extraClassPath=/usr/share/extension/dli/ spark-jar/datasource/mongo/*

图 9-2 创建 Spark 作业-高级配置

选择依赖资源

Module 名称

资源包

计算资源规格

资源规格

系统提供3种资源规格供您选择。资源规格中如下具体配置项支持修改：
Executor内存、Executor CPU核数、Executor个数、Driver CPU核数、Driver内存。
最终配置结果以修改后数据为准。

Executor内存

Executor CPU核数

Executor个数

driver CPU核数

driver内存

5. 高级包括以下两项参数：
- 选择依赖资源：具体参数请参考[表9-3](#)。
 - 计算资源规格：具体参数请参考[表9-4](#)。

📖 说明

Spark资源并行度由Executor数量和Executor CPU核数共同决定。

任务可并行执行的最大Task数量=Executor个数 * Executor CPU核数。

您可以根据购买的队列资源合理规划计算资源规格。

需要注意的是，Spark任务执行需要driver、executor等多个角色共同调度完成，因此“Executor个数*Executor CPU核数”要小于队列的计算资源CU数，避免其他Spark任务角色无法启动。更多Spark任务角色的相关信息请参考[Spark官方](#)。

Spark作业参数计算公式：

- CU数量=实际CU数量=max{ (driver CPU核数+Executor个数*Executor CPU核数) , [(driver CPU内存数+Executor个数*Executor内存) /4]}
- 内存数=driver内存+(Executor个数*Executor内存)

表 9-3 选择依赖资源参数说明

参数名称	参数描述
modules	<p>如果选择Spark版本为3.1.1时，无需选择Module模块，需在Spark参数 (--conf)配置跨源作业的依赖模块。</p> <p>DLI系统提供的用于执行跨源作业的依赖模块访问各个不同的服务，选择不同的模块：</p> <ul style="list-style-type: none"> • CloudTable/MRS HBase: sys.datasource.hbase • DDS: sys.datasource.mongo • CloudTable/MRS OpenTSDB: sys.datasource.opentsdb • DWS: sys.datasource.dws • RDS MySQL: sys.datasource.rds • RDS PostGre: sys.datasource.rds • DCS: sys.datasource.redis • CSS: sys.datasource.css <p>DLI内部相关模块：</p> <ul style="list-style-type: none"> • sys.res.dli-v2 • sys.res.dli • sys.datasource.dli-inner-table
资源包	运行spark作业依赖的jar包。

表 9-4 计算资源规格参数说明

参数名称	参数描述
资源规格	下拉选择所需的资源规格。系统提供3种资源规格供您选择。 资源规格包含以下参数： <ul style="list-style-type: none">• Executor内存• Executor CPU核数• Executor个数• driver CPU核数• driver内存 最终配置结果以修改后数据为准。
Executor内存	在所选资源规格基础上自定义Executor内存规格。 代表每个Executor的内存。通常建议Executor CPU核数： Executor内存=1: 4。
Executor CPU核数	用于设置Spark作业申请的每个Executor的CPU核数，决定每个Executor并行执行Task的能力。
Executor个数	用于设置Spark作业申请的Executor的数量。
driver CPU核数	用于设置driver CPU核数。
driver内存	用于设置driver内存大小，通常建议即driver CPU核数： driver内存=1: 4。

如果选择Spark版本为3.3.1时，支持在Spark参数（--conf）配置计算资源规格参数，且conf的配置优先级高于高级配置指定的值。

参数对应关系请参考[表9-5](#)。

说明

在Spark参数（--conf）配置计算资源规格参数时，可以配置单位 M/G/K，不配置时候默认单位为byte。

表 9-5 控制台计算资源规格参数与 Spark 参数 (--conf)配置计算资源规格参数的对应关系

控制台参数名称	Spark参数 (--conf)配置项参数名称	说明	约束与限制
Executor内存 完整的Executor内存 =spark.executor.memory + spark.executor.memoryOverhead	spark.executor.memory	executor内存，可配置。	-
	spark.executor.memoryOverhead	Spark应用程序中每个执行器（executor）的堆外内存量。该参数不可配置： spark.executor.memoryOverhead=spark.executor.memory * spark.executor.memoryOverheadFactor	最小值为384M， 即当 spark.executor.memory * spark.executor.memoryOverheadFactor的值小于384M时系统自动配置为384M。
	spark.executor.memoryOverheadFactor	该参数定义了堆外内存分配量与堆内内存分配量之比， spark jar时默认0.1， spark python 默认0.4 可配置	spark.executor.memoryOverheadFactor优先级高于 spark.kubernetes.memoryOverheadFactor
Executor CPU核数	spark.executor.cores	对应executor核数 可配置	-
Executor个数	spark.executor.instances	对应executor个数 可配置	-
driver CPU核数	spark.driver.cores	对应driver核数 可配置	-
driver内存 完整的driver内存 =spark.driver.memory + spark.edriver.memoryOverhead	spark.driver.memory	对应driver内存 可配置	-

控制台参数名称	Spark参数 (--conf)配置项参数名称	说明	约束与限制
	spark.driver.memoryOverhead	Spark应用程序中每个driver的堆外内存量。 该参数不可配置： spark.driver.memoryOverhead= spark.driver.memory * spark.driver.memoryOverheadFactor	最小值为384M，即当 spark.driver.memory * spark.driver.memoryOverheadFactor 的值小于384M时系统自动配置为384M。
	spark.driver.memoryOverheadFactor	该参数定义了堆外内存分配量与堆内内存分配量之比，spark jar时默认0.1，spark python 默认0.4 可配置	spark.driver.memoryOverheadFactor优先级高于 spark.kubernetes.memoryOverheadFactor
-	spark.kubernetes.memoryOverheadFactor	用于设置在分配给 Spark Executor的内存之外分配的内存量。spark jar时默认0.1，spark python 默认0.4 可配置	spark.executor.memoryOverheadFactor和 spark.driver.memoryOverheadFactor优先级高于 spark.kubernetes.memoryOverheadFactor。

Spark3.3.及以上版本增加了对作业的计算资源规格的约束限制。详细信息请参考[表9-6](#)。

 **注意**

若计算资源规格配置值设置得过高，超出了集群或项目的资源分配能力，作业可能会因资源申请失败导致运行错误。

表 9-6 计算资源规格取值范围

参数说明	标准版弹性资源池修改后限制	基础版弹性资源池
Executor内存	450MB-64GB	450MB-16GB
Executor CPU核数	0-16	0-4
Executor个数	无限制	无限制
driver CPU核数	0-16	0-4
driver内存	450MB-64GB	450MB-16GB
作业CU配额	无限制	无限制

- 完成作业的参数配置后，单击Spark作业编辑页面右上方“执行”，提交作业。当页面显示“批处理作业提交成功”，可在“Spark作业”管理页面查看提交作业的状态及日志等。

9.2 设置 Spark 作业优先级

操作场景

在实际作业运行中，由于作业的重要程度以及紧急程度不同，需要重点保障重要和紧急的作业正常运行，因此需要满足它们正常运行所需的计算资源。

DLI提供的设置作业优先级功能，可以对每个Spark作业设置作业优先级，当资源不充足时，可以优先满足优先级较高的作业的计算资源。

说明

Spark 2.4.5及以上版本的作业支持设置作业优先级。

使用须知

- 对于每个作业都允许设置优先级，其取值为1-10，数值越大表示优先级越高。优先满足高优先级作业的计算资源，即如果高优先级作业计算资源不足，则会减少低优先级作业的计算资源
- 通用队列上运行的Spark作业的优先级默认为3。
- 调整作业优先级需要停止作业后编辑，并重新提交运行才能生效。

Spark 作业操作步骤

在“Spark参数”中配置如下参数，其中x为优先级取值。

```
spark.dli.job.priority=x
```

- 登录DLI管理控制台。
- 单击“作业管理 > Spark作业”。
- 选择待配置的作业，单击操作列下的编辑。
- 在“Spark参数”中配置spark.dli.job.priority参数。

图 9-3 Spark 作业配置样例

Spark参数 (--conf)

```
spark.dli.job.priority=8
```

9.3 查询 Spark 作业日志

操作场景

DLI作业桶用于存储DLI作业运行过程中产生的临时数据，例如：作业日志、作业结果。

本节操作指导您在DLI管理控制台配置DLI作业桶，并获取Spark作业日志的操作方法。

使用须知

- 请勿将该DLI作业桶绑定的OBS桶用作其它用途，避免出现作业结果混乱等问题。
- DLI作业要由用户主账户统一设置及修改，子用户无权限。
- 不配置DLI作业桶无法查看作业日志。
- 您可以通过配置桶的生命周期规则，定时删除桶中的对象或者定时转换对象的存储类别。
- DLI的作业桶设置后请谨慎修改，否则可能会造成历史数据无法查找。
- Spark日志分割规则：
 - 按大小分割：默认情况下，每个日志文件最大为128MB。
 - 按时间分割：每过一小时自动创建新的日志文件。

前提条件

配置前，请先购买OBS桶或并行文件系统。大数据场景推荐使用并行文件系统，并行文件系统（Parallel File System）是对象存储服务（Object Storage Service，OBS）提供的一种经过优化的高性能文件系统，提供毫秒级别访问时延，以及TB/s级别带宽和百万级别的IOPS，能够快速处理高性能计算（HPC）工作负载。

并行文件系统的详细介绍和使用说明，请参见《[并行文件系统特性指南](#)》。

配置 DLI 作业桶


1. 在DLI控制台左侧导航栏中单击“全局配置 > 工程配置”。
2. 在“工程配置”页面，选择“DLI作业桶”，单击配置桶信息。

图 9-4 工程配置




3. 单击  打开桶列表。
4. 选择用于存放DLI作业临时数据的桶，并单击“确定”。
完成设置后DLI作业运行过程中产生的临时数据将会存储在该OBS桶中。

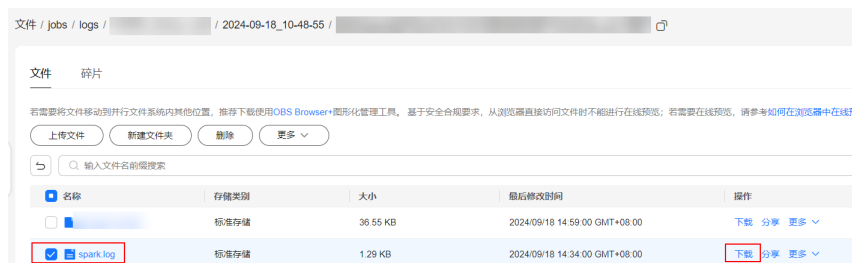
图 9-5 设置 DLI 作业桶



查询 Spark 作业日志

1. 登录DLI管理控制台，单击“作业管理 > Spark作业”。
2. 选择待查询的Spark作业，单击操作列的“更多 > 归档日志”。
系统自动跳转至DLI作业桶日志路径下。
3. 选择需要查看的日期，单击操作列的“下载”下载Spark日志到本地。

图 9-6 下载 Spark 作业日志



9.4 管理 Spark 作业

查看 Spark 作业的基本信息

在总览页面单击“Spark作业”简介，或在左侧导航栏单击“作业管理”>“Spark作业”，可进入Spark作业管理页面。Spark作业管理页面显示所有的Spark作业，作业数量较多时，系统分页显示，您可以查看任何状态下的作业。

表 9-7 作业管理参数

参数	参数说明
作业ID	所提交Spark作业ID，由系统默认生成。
名称	所提交Spark作业的名称。
队列	所提交Spark作业所在的队列。
用户名	执行Spark作业的用户名称。
状态	作业的状态信息，包括如下。 <ul style="list-style-type: none">• 启动中：正在启动• 运行中：正在执行任务• 已失败：session已退出• 已成功：session运行成功• 恢复中：正在恢复任务
创建时间	每个作业的创建时间，可按创建时间顺序或倒序显示作业列表。
最后修改时间	作业运行完成的时间。
操作	<ul style="list-style-type: none">• 编辑：可修改当前作业配置，重新执行作业。• SparkUI：单击后，将跳转至Spark任务运行情况界面。 说明<ul style="list-style-type: none">• 状态为“启动中”的作业不能查看SparkUI界面。• 目前DLI配置SparkUI只展示最新的100条作业信息。• 终止作业：终止启动中和运行中的作业。• 重新执行：重新运行该作业。• 归档日志：将作业日志保存到系统创建的DLI临时数据桶中。• 提交日志：查看提交作业的日志。• Driver日志：查看运行作业的日志。

重新执行作业

在“Spark作业”页面，单击对应作业“操作”列中的“编辑”，跳转至“Spark作业编辑”页面，可根据需要修改参数，执行作业。

查找作业

在“Spark作业”页面，选择“状态”或“队列”。系统将根据设置的过滤条件，在作业列表显示符合对应条件的作业。

终止作业

在“Spark作业”页面，单击对应作业“操作”列中的“更多”>“终止作业”，可停止启动中和运行中的作业。

9.5 管理 Spark 作业模板

操作场景

在创建Spark作业时，您可以在已有的Spark样例模板中进行修改，来实现实际的作业逻辑需求，节约编辑SQL语句的时间。

当前云平台尚未提供预置的Spark模板，但支持用户自定义Spark作业模板，本节操作介绍在Spark管理页面创建Spark模板的操作方法。

新建 Spark 作业模板

Spark作业模板的创建方法是在创建Spark作业时，可直接将配置完成的作业信息设置为模板。

1. 在DLI管理控制台的左侧导航栏中，单击“作业模板”>“Spark模板”，页面跳转至Spark作业页面。
2. 参考[创建Spark作业](#)配置作业参数。
3. 作业编辑完成后，单击“设为模板”。
4. 输入模板名称和描述信息。
5. 设置模板的分组信息。便于模板的统一管理。
6. 单击“确定”，完成Spark模板的创建。

10 使用 Notebook 实例提交 DLI 作业

Notebook是基于开源JupyterLab进行了深度优化的交互式数据分析挖掘模块，提供在线的开发和调试能力，用于编写和调测模型训练代码。完成DLI对接Notebook实例后，您可以基于Notebook提供的Web交互的开发环境同时完成代码的编写与作业的开发，使用Notebook灵活的进行数据分析与探索，本节操作介绍使用Notebook作业提交DLI作业的操作步骤。

关于Jupyter Notebook的详细操作指导，请参见[Jupyter Notebook使用文档](#)。

使用Notebook实例提交DLI作业适用于在线开发调试场景下的作业需求，无需准备开发环境，一站式完成数据分析分析与探索。

使用须知

- 该功能为**白名单功能**，如需使用，请在管理控制台右上角，选择“工单 > 新建工单”，提交申请。
- 在DLI管理控制台删除弹性资源池时并不会删除关联的Notebook实例，如果不再使用Notebook实例，请登录ModelArts管理控制台删除对应的Notebook资源。

操作流程

1. 创建弹性资源池并添加通用队列。
在DLI提交Notebook实例需要使用弹性资源池资源，并在弹性资源池中创建通用队列用于后续执行作业所需的计算资源。请参考[步骤1：创建弹性资源池并添加通用队列](#)。
2. 创建Notebook实例所需的VPC和安全组。
配置弹性资源池开启Notebook后，弹性资源池会准备Notebook功能所需的组件。请参考[步骤2：创建虚拟私有云和安全组](#)。
3. 创建增强型跨源连接用于打通DLI弹性资源池和Notebook实例的网络。
请参考[步骤3：创建增强型跨源连接](#)。
4. 准备创建Notebook实例所需的自定义镜像。
请参考[步骤4：注册ModelArts自定义镜像](#)。
5. 创建自定义委托用于访问Notebook实例。
请参考[步骤5：创建DLI自定义委托用于访问Notebook实例](#)。
6. 在DLI的弹性资源池中创建Notebook实例。
请参考[步骤6：在DLI弹性资源池中创建Notebook实例](#)。

7. 配置Notebook访问DLI或LakeFormation元数据。
 - (可选) 配置Notebook访问DLI元数据
 - (可选) 配置Notebook访问LakeFormation元数据
8. 在JupyterLab中编写和调试代码。

进入JupyterLab主页后,可在“Notebook”区域下编辑和调试代码。[步骤8: 使用Notebook实例编写和调试代码。](#)

约束限制

- 使用Notebook实例提交DLI作业必须使用弹性资源池下的通用队列。
- 每一个弹性资源池关联唯一的Notebook实例。
- Notebook作业运行过程中产生的临时数据将会存储在DLI作业桶中,且必须使用并行文件系统。
- 请在ModelArts管理控制台管理Notebook实例。请参考[管理Notebook实例](#)。
- Notebook实例用于代码编辑和开发,关联队列用于执行作业。

如需更换Notebook实例关联的队列请在ModelArts管理控制台进行相关操作。

步骤 1: 创建弹性资源池并添加通用队列

1. 创建弹性资源池。
 - a. 登录DLI管理控制台,在左侧导航栏单击“资源管理 > 弹性资源池”,可进入弹性资源池管理页面。
 - b. 在弹性资源池管理界面,单击界面右上角的“购买弹性资源池”。
 - c. 在“购买弹性资源池”界面,填写具体的弹性资源池参数,具体参数填写参考[创建弹性资源池并添加队列](#)。
 - CU范围:请确保弹性资源池预留资源大于16CUs,用于NoteBook实例资源所需。
 - 网段:请注意弹性资源池网段请勿与下列网段重复:
172.18.0.0/16、172.16.0.0/16、10.247.0.0/16
 - d. 参数填写完成后,单击“立即购买”,在界面上确认当前配置是否正确。
 - e. 单击“提交”完成队列创建。等待弹性资源池状态变成“可使用”表示当前弹性资源池创建成功。
2. 在弹性资源池添加通用队列。
 - a. 选择要操作的弹性资源池,在“操作”列,单击“添加队列”。
 - b. 在“添加队列”界面,配置队列的基础配置,具体参数信息请参考[创建弹性资源池并添加队列](#)。

队列类型选择“通用队列”。
 - c. 单击“下一步”,在“扩缩容策略”界面配置当前队列在弹性资源池的扩缩容策略。
 - d. 单击“确定”完成添加队列配置。

步骤 2: 创建虚拟私有云和安全组

- 创建虚拟私有云

- a. 登录VPC管理控制台，进入创建虚拟私有云页面。
 - b. 在“创建虚拟私有云”页面，根据界面提示配置VPC和子网的参数。
具体参数说明请参考[创建虚拟私有云](#)。
其中配置IPv4网段时，请确保VPC的IPv4网段不要与下列网段重复。
172.18.0.0/16、172.16.0.0/16、10.247.0.0/16
- **创建安全组**
 - a. 登录VPC管理控制台，进入安全组列表页面。
 - b. 在安全组列表右上方，单击“创建安全组”。
进入“创建安全组”页面。根据界面提示，设置安全组参数。
具体参数说明请参考[创建安全组](#)。

请确保安全组需要对DLI弹性资源池网段放通TCP的8998和30000-32767端口。

步骤 3：创建增强型跨源连接

1. 登录DLI管理控制台。
2. 在左侧导航栏中，选择“跨源管理 > 增强型跨源”。
3. 选择“增强型跨源”，单击“创建”。
配置增强型跨源连接信息，详细参数介绍请参见[表5-22](#)。
创建增强型跨源连接时：
 - 弹性资源池：选择[步骤1：创建弹性资源池并添加通用队列](#)创建的弹性资源池。
 - 虚拟私有云：选择[步骤2：创建虚拟私有云和安全组](#)创建的虚拟私有云。

步骤 4：注册 ModelArts 自定义镜像

基于ModelArts提供的MindSpore预置镜像，并借助ModelArts命令行工具，通过加载镜像构建模板并修改Dockerfile，构建出一个新镜像，最后注册后在Notebook使用。

ModelArts命令行工具请参考[ma-cli镜像构建命令介绍](#)。

- 基础镜像地址：swr.{endpoint}/atelier/pyspark_3_1_1:develop-remote-pyspark_3.1.1-py_3.7-cpu-ubuntu_18.04-x86_64-uid1000-20230308194728-68791b4
请按需更换地址中的Region名称后使用
例如，新加坡区域的endpoint为ap-southeast-3.myhuaweicloud.com
拼接后的基础镜像地址为：swr.ap-southeast-3.myhuaweicloud.com/atelier/pyspark_3_1_1:develop-remote-pyspark_3.1.1-py_3.7-cpu-ubuntu_18.04-x86_64-uid1000-20230308194728-68791b4
- 在ModelArts创建并注册自定义镜像的详细操作请参考[在Notebook中通过Dockerfile从0制作自定义镜像](#)。

步骤 5：创建 DLI 自定义委托用于访问 Notebook 实例

参考[创建DLI自定义委托权限](#)创建DLI自定义委托用于访问Notebook实例。

请确保委托中包含以下权限：ModelArts FullAccess、DLI FullAccess、OBS Administratorhe、和IAM的授予向云服务传递委托的权限。

关于IAM的授予向云服务传递委托的权限，如果使用的是IAM角色或策略授权：请授予IAMiam:agencies:*权限。

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:agencies:*"
      ]
    },
    {
      "Effect": "Deny",
      "Action": [
        "iam:agencies:update*",
        "iam:agencies:delete*",
        "iam:agencies:create*"
      ]
    }
  ]
}
```

步骤 6：在 DLI 弹性资源池中创建 Notebook 实例

步骤1 在DLI弹性资源池页面预置创建Notebook实例相关的DLI资源信息。

1. 登录DLI管理控制台，进入弹性资源池列表页面。
2. 选择**步骤1：创建弹性资源池并添加通用队列**中创建的弹性资源池。
3. 单击操作列的“更多 > Notebook(新)”。
4. 单击“创建Notebook”，配置以下参数信息：
 - 自定义镜像：选择**步骤4：注册ModelArts自定义镜像**中注册的镜像。
 - 所属队列：选择**步骤1：创建弹性资源池并添加通用队列**中创建的队列。
 - Spark版本：推荐选择Spark 3.3.1版本。
 - 增强型跨源链接：选择**步骤3：创建增强型跨源连接**中创建的增强型跨源连接。

图 10-1 预置创建 Notebook 实例相关的 DLI 资源信息



5. 单击“确定”创建Notebook实例。系统跳转至Notebook实例创建页面

步骤2 在Notebook实例页面配置Notebook实例相关参数。

1. 创建Notebook实例。

具体参数说明请参考[创建Notebook实例](#)。

配置过程中：

- 镜像：选择自定义镜像，选择[步骤4：注册ModelArts自定义镜像](#)中注册的镜像。
- VPC接入：开启VPC接入接入功能

说明

请联系客户支持开启Notebook实例的VPC接入白名单功能。

安全组请配置为[步骤2：创建虚拟私有云和安全组](#)中创建的安全组，且安全组需要对DLI弹性资源池网段放通TCP的8998和30000-32767端口。

参数配置完成后单击“立即创建”，等待Notebook实例创建完成。

步骤3 配置Notebook实例连接DLI。

1. 在Notebook实例的列表中单击操作类的“打开”跳转至Notebook实例页面。
2. 在Notebook实例页面单击右上角的“connect”连接DLI。

图 10-2 连接 DLI



3. 在Connect Cluster页面中，填写作业运行的相关信息。

图 10-3 Connect Cluster

Connect Cluster ✕

* Service Type ▼
DLI

* Pool Name ▼
[]

* Queue Name ▼
notebook

* Spark Version ▼
[]

[^ Advanced Settings](#)

自定义镜像 ▼
image_org/image_name:tag

Spark参数(--conf) ▼
请输入格式为key=value的参数，多个参数以Enter键分隔

依赖Jar包(--jars) ▼
请填写文件的OBS路径，多个路径以Enter键分隔

依赖python文件(--py-files) ▼
请填写文件的OBS路径，多个路径以Enter键分隔

其他依赖文件(--files) ▼
请填写文件的OBS路径，多个路径以Enter键分隔

资源配置

Driver 内存 ▼
1 GB

表 10-1 Connect Cluster

参数名称	说明	配置样例
Service Type	连接的服务名称。	DLI

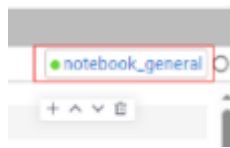
参数名称	说明	配置样例
Pool Name	Notebook作业运行所在队列对应的弹性资源池。	本例配置为 步骤1：创建弹性资源池并添加通用队列 中创建的弹性资源池。
Queue Name	Notebook作业运行所在的队列。	本例配置为 步骤1：创建弹性资源池并添加通用队列 中创建的队列。
Spark Version	Spark引擎版本。	当前仅Spark 3.3.1版本支持使用Notebook实例提交DLI作业。
Spark参数(--conf)	该参数用于配置DLI作业的自定义参数。	请参考 表10-2 。

表 10-2 常用 Spark 参数配置项

参数名称	说明
spark.dli.job.agency.name	用于指定DLI作业的委托权限名称。 在使用Flink 1.15和Spark 3.3及以上版本的引擎执行作业时，需要在作业配置中添加新建的委托信息。 配置样例： 本例配置为用于访问Notebook的DLI委托名称“dli_notebook”。 spark.dli.job.agency.name=dli_notebook
spark.sql.session.state.builder	该参数是指定元数据的配置项。 配置样例：配置访问DLI元数据场景的配置项 spark.sql.session.state.builder=org.apache.spark.sql.hive.DliLakeHouseBuilder
spark.sql.catalog.class	用于指定不同的数据源和元数据管理系统。 配置样例：配置访问DLI元数据场景的配置项 spark.sql.catalog.class=org.apache.spark.sql.hive.DliLakeHouseCatalog
spark.dli.metaAccess.enable	用于开启或关闭对DLI元数据的访问。 spark.dli.metaAccess.enable=true

- 完成后单击连接，等待右上方的connect变成队列名称，名称前面小圆点变绿后代表连接成功，即可执行相关notebook作业。

图 10-4 Notebook 实例完成连接。



5. 单击“Connect”测试连接。

----结束

等待实例初始化完成后即可在Notebook执行在线的数据分析操作。通常实例初始化需要2分钟左右。

在Notebook执行相关sql语句，在DLI就会启动一个Spark作业，同时在Notebook中显示作业结果。

步骤 7：配置 Notebook 访问 DLI 元数据

执行作业前需要配置Notebook访问DLI或LakeFormation元数据。

- [（可选）配置Notebook访问DLI元数据](#)
- [（可选）配置Notebook访问LakeFormation元数据](#)

步骤 8：使用 Notebook 实例编写和调试代码

Notebook与DLI队列连接成功后，即可在“Notebook”区域下编辑和调试代码。

您可以选择使用Notebook提交作业，或在DLI管理控制台的Spark作业操作页面提交作业。

- Notebook相关操作请参考[JupyterLab简介及常用操作](#)。
- Notebook中的数据上传请参考[上传文件至JupyterLab](#)。
- Notebook中的数据下载请参考[下载JupyterLab文件到本地](#)。

（可选）配置 Notebook 访问 DLI 元数据

在完成DLI和Notebook的对接后，您需要配置如需在Notebook提交DLI作业场景下使用元数据的方式，本小节操作介绍配置访问DLI元数据的操作步骤。

如需配置Notebook访问LakeFormation元数据请参考[（可选）配置Notebook访问LakeFormation元数据](#)。

1. **指定Notebook镜像。**
2. **自定义委托授权DLI使用DLI元数据和OBS。**
自定义委托操作步骤请参考[创建DLI自定义委托权限](#)。
请确保自定义委托具备以下权限：

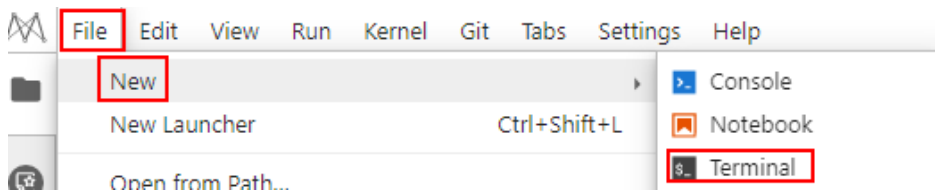
表 10-3 DLI 自定义委托场景

场景	委托名称	适用场景	权限策略
允许DLI读写OBS 将日志转储	自定义	DLI Flink作业下载OBS对象、OBS/DWS数据源（外表）、日志转储、使用savepoint、开启checkpoint，DLI Spark作业下载OBS对象、读写OBS外表。	访问和使用OBS的权限策略
允许访问DLI Catalog元数据	自定义	DLI 访问DLI元数据。	访问DLI Catalog元数据的权限

3. 确认开启访问DLI元数据。

- a. 登录ModelArts管理控制台，选择“开发空间>Notebook”。
- b. 创建Notebook实例，实例处于“运行中”，单击“操作”列的“打开”，进入“JupyterLab”开发页面。
- c. 选择“Files > New > Terminal”，进入到Terminal界面。

图 10-5 进入到 Terminal 界面



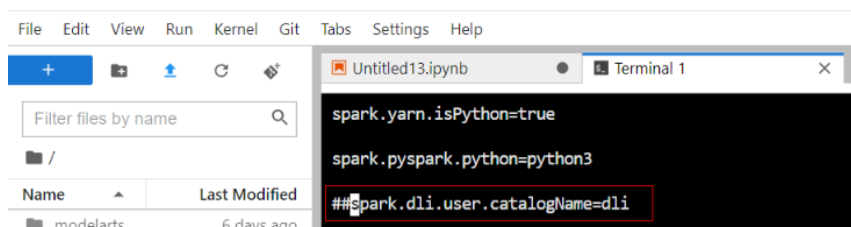
- d. 执行以下命令进入到livy配置目录下，查看spark配置文件。

```
cd /home/ma-user/livy/conf/
vi spark-defaults.conf
```

确认包含spark.dli.user.catalogName=dli配置项，该配置项即访问DLI元数据。

spark.dli.user.catalogName=dli为默认配置项。

图 10-6 关闭默认访问 DLI 元数据



- e. 使用notebook编辑作业。
 - Notebook相关操作请参考[JupyterLab简介及常用操作](#)。
 - Notebook中的数据上传请参考[上传文件至JupyterLab](#)。

- Notebook中的数据下载请参考[下载JupyterLab文件到本地](#)。

(可选) 配置 Notebook 访问 LakeFormation 元数据

在完成DLI和Notebook的对接后，您需要配置如需在Notebook提交DLI作业场景下使用元数据的方式，本小节操作介绍配置访问LakeFormation元数据的操作步骤。

如需配置Notebook访问DLI元数据请参考[\(可选\) 配置Notebook访问DLI元数据](#)。

1. **DLI对接LakeFormation。**
 - a. 具体操作请参考[DLI对接LakeFormation](#)。
2. **指定Notebook镜像。**
3. **自定义委托授权DLI使用LakeFormation和OBS。**
自定义委托操作步骤请参考[创建DLI自定义委托权限](#)。
请确保自定义委托具备以下权限：

表 10-4 DLI 自定义委托场景

场景	委托名称	适用场景	权限策略
允许DLI读写OBS 将日志转储	自定义	DLI Flink作业下载OBS对象、OBS/DWS数据源（外表）、日志转储、使用savepoint、开启checkpoint，DLI Spark作业下载OBS对象、读写OBS外表。	访问和使用OBS的权限策略
允许访问 LakeFormation Catalog元数据	自定义	DLI 访问LakeFormation元数据。	访问LakeFormation Catalog元数据的权限

4. **在Notebook实例页面配置Spark参数。**
 - a. 选择DLI的notebook镜像的队列，并且单击connect，配置spark参数。

```
spark.sql.catalogImplementation=hive
spark.hadoop.hive-ext.dlcatalog.metastore.client.enable=true
spark.hadoop.hive-ext.dlcatalog.metastore.session.client.class=com.huawei.cloud.dalf.lakecat.client.hiveclient.LakeCatMetaStoreClient
spark.hadoop.lakecat.catalogname.default=lfcatalog //需要指定要访问哪个catalog
spark.dli.job.agency.name=agencyForLakeformation //此委托中需要有lf和obs必要的权限，并且需要委托给DLI
spark.driver.extraClassPath=/usr/share/extension/dli/spark-jar/lakeformation/*
spark.executor.extraClassPath=/usr/share/extension/dli/spark-jar/lakeformation/*
spark.sql.extensions=org.apache.spark.sql.hudi.HoodieSparkSessionExtension
spark.hadoop.hoodie.support.write.lock=org.apache.hudi.lakeformation.LakeCatMetastoreBasedLockProvider
```

表 10-5 参数说明

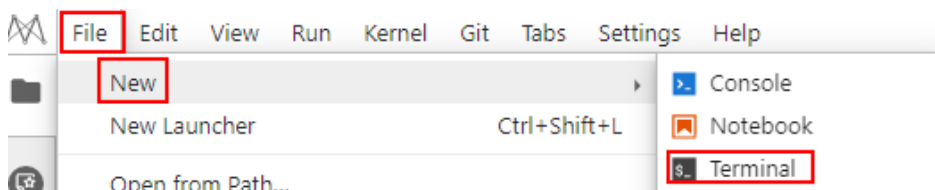
配置项	是否必选	参数值	参数配置场景
spark.sql.catalogImplementation	是	hive	用于指定使用哪种类型的Catalog来存储和管理元数据
spark.hadoop.hive-ext.dlcatalog.metastore.client.enable	是	true	开启访问LakeFormation元数据时需要配置该参数。
spark.hadoop.hive-ext.dlcatalog.metastore.session.client.class	是	com.huawei.cloud.dalf.lakecatalog.client.hiveclient.LakeCatalogMetaStoreClient	开启访问LakeFormation元数据时需要配置该参数。
spark.hadoop.lakecatalogname.default	否	lfcatalog	配置需要访问的LakeFormation数据目录名称。 默认取值hive
spark.dli.job.agency.name	是	用户自定义委托名称	用户自定义委托名。 <ul style="list-style-type: none"> 创建自定义委托请参考创建DLI自定义委托权限 DLI元数据委托权限请参考访问LakeFormation Catalog元数据的权限
spark.driver.extraClassPath	是	/usr/share/extension/dli/spark-jar/lakeformation/*	配置LakeFormation的依赖包加载。
spark.executor.extraClassPath	是	/usr/share/extension/dli/spark-jar/lakeformation/*	配置LakeFormation的依赖包加载。
spark.sql.extensions	否	org.apache.spark.sql.hudi.HoodieSparkSessionExtension	hudi场景需配置该参数。

配置项	是否必选	参数值	参数配置场景
spark.hadoop.hoodie.support.write.lock	否	org.apache.hudi.lakeformati.on.LakeCatMetastoreBasedLockProvider	hudi场景需配置该参数。

5. 关闭默认访问DLI元数据，切换使用Lakeformation元数据。

- 登录ModelArts管理控制台，选择“开发环境>Notebook”。
- 创建Notebook实例，实例处于“运行中”，单击“操作”列的“打开”，进入“JupyterLab”开发页面。
- 选择“Files > New > Terminal”，进入到Terminal界面。

图 10-7 进入到 Terminal 界面

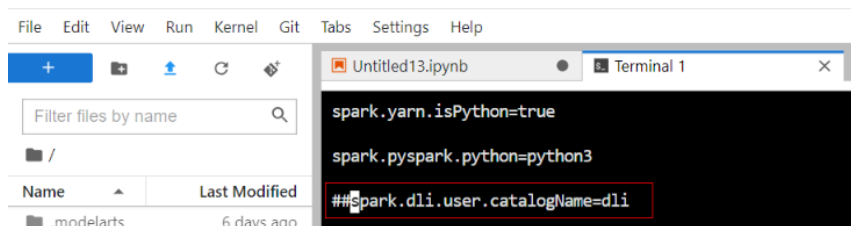


- 执行以下命令进入到livy配置目录下，修改spark配置文件，关闭默认访问DLI元数据。

```
cd /home/ma-user/livy/conf/  
vi spark-defaults.conf
```

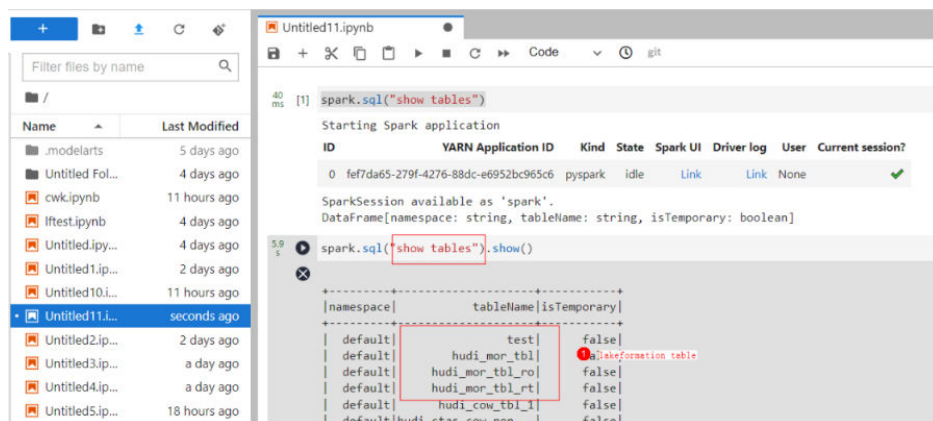
使用#注释掉spark.dli.user.catalogName=dli，关闭默认访问DLI元数据。

图 10-8 关闭默认访问 DLI 元数据



- 使用notebook编辑作业。
执行spark.sql即可访问Lakeformation元数据和hudi相关表。

图 10-9 访问 Lakeformation 元数据



11 使用 CES 监控 DLI 服务

功能说明

本章节定义了数据湖探索服务上报云监控的监控指标的命名空间，监控指标列表和维度定义，用户可以通过云监控服务提供的管理控制台或API接口来检索数据湖探索服务产生的监控指标和告警信息。

命名空间

SYS.DLI

监控指标

表 11-1 数据湖探索服务支持的监控指标

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期（原始指标）
queue_cu_num	队列CU使用量	展示用户队列申请的CU数	≥0	队列	5分钟
queue_job_launching_num	提交中作业数	展示用户队列中状态为提交中的作业数。	≥0	队列	5分钟
queue_job_running_num	运行中作业数	展示用户队列中状态为运行中的作业数。	≥0	队列	5分钟
queue_job_succeed_num	已完成作业数	展示用户队列中状态为已完成的作业数。	≥0	队列	5分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期（原始指标）
queue_job_failed_num	已失败作业数	展示用户队列中状态为已失败的作业数。	≥0	队列	5分钟
queue_job_cancelled_num	已取消作业数	展示用户队列中状态为已取消的作业数。	≥0	队列	5分钟
queue_alloc_cu_num	队列CU分配量	展示用户队列的CU分配情况。	≥0	队列	5分钟
queue_min_cu_num	队列最小CU	展示用户队列中的最小CU。	≥0	队列	5分钟
queue_max_cu_num	队列最大CU	展示用户队列中的最大CU。	≥0	队列	5分钟
queue_priority	队列优先级	展示用户队列的优先级。	1~100	队列	5分钟
queue_cpu_usage	队列CPU使用率	展示用户队列的CPU使用率。 该指标仅适用于普通队列。	0~100	队列	5分钟
queue_disk_usage	队列磁盘使用率	展示用户队列的磁盘使用率。 该指标仅适用于普通队列。	0~100	队列	5分钟
queue_disk_used	队列磁盘使用率最大值	展示用户队列的磁盘使用率的最大值。 该指标仅适用于普通队列。	0~100	队列	5分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期（原始指标）
queue_mem_usage	队列内存使用率	展示用户队列的内存使用率。 该指标仅适用于普通队列。	0~100	队列	5分钟
queue_mem_used	队列内存使用量	展示用户队列的内存使用量。 该指标仅适用于普通队列。	≥0	队列	5分钟
flink_read_records_per_second	Flink作业数据输入速率	展示用户Flink作业的数据输入速率，供监控和调试使用。	≥0	Flink作业	10秒钟
flink_write_records_per_second	Flink作业数据输出速率	展示用户Flink作业的数据输出速率，供监控和调试使用。	≥0	Flink作业	10秒钟
flink_read_records_total	Flink作业数据输入总数	展示用户Flink作业的数据输入总数，供监控和调试使用。	≥0	Flink作业	10秒钟
flink_write_records_total	Flink作业数据输出总数	展示用户Flink作业的数据输出总数，供监控和调试使用。	≥0	Flink作业	10秒钟
flink_read_bytes_per_second	Flink作业字节输入速率	展示用户Flink作业每秒输入的字节数。	≥0	Flink作业	10秒钟
flink_write_bytes_per_second	Flink作业字节输出速率	展示用户Flink作业每秒输出的字节数。	≥0	Flink作业	10秒钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期（原始指标）
flink_read_bytes_total	Flink作业字节输入总数	展示用户Flink作业字节的输入总数。	≥0	Flink作业	10秒钟
flink_write_bytes_total	Flink作业字节输出总数	展示用户Flink作业字节的输出总数。	≥0	Flink作业	10秒钟
flink_cpu_usage	Flink作业CPU使用率	展示用户Flink作业的CPU使用率。	0~100	Flink作业	10秒钟
flink_mem_usage	Flink作业内存使用率	展示用户Flink作业的内存使用率。	0~100	Flink作业	10秒钟
flink_max_op_latency	Flink作业最大算子延迟	展示用户Flink作业的最大算子延迟时间，单位ms。	≥0	Flink作业	10秒钟
flink_max_op_backpressure_level	Flink作业最大算子反压	展示用户Flink作业的最大算子反压值，数值越大，反压越严重。 0: 表示OK 50: 表示Low 100: 表示High	0~100	Flink作业	10秒钟
elastic_resource_pool_cpu_usage	弹性资源池CPU使用率	展示用户弹性资源池的CPU使用率。	0~100	弹性资源池	5分钟
elastic_resource_pool_mem_usage	弹性资源池内存使用率	展示用户弹性资源池的内存使用率。	0~100	弹性资源池	5分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期（原始指标）
elastic_resource_pool_disk_usage	弹性资源池磁盘使用率	展示用户弹性资源池的磁盘使用率。	0~100	弹性资源池	5分钟
elastic_resource_pool_disk_max_usage	弹性资源池磁盘使用率最大值	展示用户弹性资源池的磁盘使用率最大值。	0~100	弹性资源池	5分钟
elastic_resource_pool_cu_num	弹性资源池CU使用量	展示用户弹性资源池的CU使用量。	≥0	弹性资源池	5分钟
elastic_resource_pool_alloc_cu_num	弹性资源池CU分配量	展示用户弹性资源池的CU分配情况。	≥0	弹性资源池	5分钟
elastic_resource_pool_min_cu_num	弹性资源池最小CU	展示用户弹性资源池的最小CU。	≥0	弹性资源池	5分钟
elastic_resource_pool_max_cu_num	弹性资源池最大CU	展示用户弹性资源池的最大CU。	≥0	弹性资源池	5分钟

维度

表 11-2 维度

Key	Value
queue_id	队列
flink_job_id	Flink作业

通过云监控服务 CES 查看 DLI 监控指标

1. 在管理控制台搜索“云监控服务”。
2. 进入云监控服务的控制台后，在左侧列表中，单击“数据湖探索”。
3. 选择队列进行查看相关监控信息。

12 使用 AOM 监控 DLI 服务

12.1 配置 DLI 对接 AOM Prometheus 监控

AOM服务提供的Prometheus监控是一种全面对接开源Prometheus生态的监控解决方案。它支持多种类型的组件监控，提供预置监控大盘和全面托管的Prometheus服务，通过Prometheus监控来统一采集、存储和显示监控对象的数据，适用于时间序列数据库的收集和处理，尤其适用于监控Flink作业场景。

本节操作介绍配置DLI对接AOM Prometheus监控的操作步骤。

使用须知

- 仅Flink 1.15版本支持对接AOM Prometheus监控。
- 需提前创建AOM Prometheus通用集群，创建AOM Prometheus通用集群不收费，AOM的计费项由自定义指标上报量、指标存储时长、数据转储量的费用组成。了解[AOM计费模式与计费项](#)。
- AOM 2.0基于自定义指标上报量进行计费，了解[计费规则](#)。
- 仅支持AOM Prometheus for通用实例。
- 弹性资源池对接Prometheus实例后，当前弹性资源池下所有新提交运行的Flink 1.15作业指标都会上报到绑定的Prometheus。默认只上报基础指标，基础指标AOM Prometheus不收取费用。如需上报所有指标请参考[DLI对接AOM Prometheus监控的配置项](#)章节的metrics.reporter.remote.report-all-metrics参数进行配置。
- DLI Flink指标上报周期默认为30秒，因此指标上报有一定延迟。如需调整上报周期，请参考[DLI对接AOM Prometheus监控的配置项](#)章节metrics.reporter.remote.interval参数进行配置。
不建议将该参数设置过低，否则上报过于频繁，推荐配置为30秒。
- Flink 1.15及以上版本中，弹性资源池与Prometheus实例解绑后，新作业不再上报指标到该Prometheus实例，已提交的作业继续上报至作业运行结束。
- Flink 1.15及以上版本中，修改绑定的Prometheus实例后，新作业上报指标到修改后的Prometheus实例，已提交的作业继续上报至原Prometheus实例直至作业运行结束。

步骤 1：创建 AOM Prometheus 实例

1. 登录AOM 2.0管理控制台。
2. 在左侧导航栏选择“Prometheus监控 > 实例列表”，然后单击“创建 Prometheus实例”。
3. 设置实例名称、企业项目和实例类型信息。

表 12-1 配置 Prometheus 实例

参数名称	说明
实例名称	Prometheus实例的名称。
企业项目	所属的企业项目。 <ul style="list-style-type: none">• 如果在全局页面设置为“ALL”，此处请从下拉列表中选择企业项目。• 如果在全局页面已选择企业项目，则此处灰化不可选。
实例类型	Prometheus实例的类型，此处选择“Prometheus 通用实例”。

步骤 2：弹性资源绑定 AOM Prometheus 集群

1. 登录DLI管理控制台，选择“资源管理 > 弹性资源池”。
2. 选择弹性资源池，单击操作列的“更多 > Prometheus > 绑定Prometheus”。
3. 选择[步骤1：创建AOM Prometheus实例](#)中创建的Prometheus集群。
4. 单击“确定”绑定AOM Prometheus集群。

绑定AOM Prometheus实例后将新提交运行的作业监控指标上报到AOM，并按照AOM计费规则收费。

📖 说明

绑定Prometheus需具备访问AOM Prometheus的权限，否则绑定失败。

具体权限内容包括：

- aom:prometheusInstances:list
- aom:metric:list
- aom:metric:get

步骤 3：创建并提交 Flink 作业

参考[创建Flink OpenSource SQL作业](#)创建Flink作业。

选择Flink版本：1.15。仅Flink 1.15及以上版本支持AOM监控。

在作业运行后约30s后，系统上报作业的监控指标至AOM Prometheus实例。

步骤 4：在 AOM 仪表盘中查看监控指标

DLI支持的Prometheus监控指标请参考[DLI支持的Prometheus基础监控指标](#)

打开AOM仪表盘即可查看监控指标，您可以按需选择以下任一种方法跳转至AOM控制台。

- 方式1：在DLI管理控制台跳转至AOM仪表盘
 - a. 登录DLI管理控制台，选择“作业管理 > Flink作业”。
 - b. 单击作业名称进入作业详情页面。
 - c. 在作业详情页面单击“更多 > Prometheus监控”。
跳转至AOM仪表盘页面。
- 方式2：在AOM预置仪表盘中查看监控仪表盘
 - a. 登录AOM 2.0管理控制台。
 - b. 在左侧导航栏选择“仪表盘”。
在仪表盘页面左侧列表中选择“应用”，并在应用列表中选择类型为“DLI_FLINK”的仪表盘。
 - c. 单击仪表盘名称进入监控指标仪表盘。
 - d. 配置筛选条件查看详细的监控指标
默认情况下会展示当前prometheus下的所有指标数据，若需查看某个弹性资源池、某个作业甚至某个作业某次提交的指标信息，则需要根据实际进行筛选。

表 12-2 监控指标

筛选条件	说明
Prometheus实例	基于Prometheus实例进行过滤，表示查看该实例下的所有指标信息。
弹性资源池	基于弹性资源池名称进行过滤，表示查看该弹性资源池下的所有指标信息。
DLI-flink作业Id	基于DLI Flink的作业ID进行过滤，表示查看当前DLI Flink所有提交的指标信息。 在DLI管理控制台Flink作业的列表页面可获取DLI Flink作业ID。
DLI-flink-jobName	基于DLI Flink的作业名称进行过滤，表示查看当前DLI Flink所有提交的指标信息。 在DLI管理控制台Flink作业的列表页面可获取DLI Flink的作业名称。
jobId	基于Flink作业的job ID进行过滤，表示查看当前Flink作业的指标信息，即仅查看当前job ID的监控指标。 <ul style="list-style-type: none"> • 通过Flink UI查看job ID。 • 通过日志查看，可在Flink jobmanager日志中搜索关键词查看job ID。

步骤 5：配置 Prometheus 监报告警通知（可选）

如需及时了解Prometheus监控状态并做出响应，您还需要配置告警通知，SMN服务为您提供灵活的消息推送能力，可以将Prometheus的告警事件通知发送到不同的终

端，从而实现多通道告警事件通知。本节操作介绍配置Prometheus监控告警通知的操作步骤。了解SMN计费规则请参考[SMN计费说明](#)。

步骤1 创建SMN主题并添加订阅。

1. 创建SMN主题。
 - a. 登录SMN管理控制台。
 - b. 在左侧导航栏，选择“主题管理” > “主题”。进入主题页面。
 - c. 在主题页面，单击“创建主题”。
 - d. 配置主题的相关参数。
输入“主题名称”和“显示名”。更多参数说明请参考[SMN-创建主题](#)。
 - e. 在“主题名称”框中，输入主题名称，在“显示名”框中输入相关描述。
2. 订阅主题。

要接收发布至主题的消息，您必须向该主题添加订阅者。

 - a. 登录SMN管理控制台。
 - b. 在左侧导航栏，选择“主题管理 > 主题”。进入主题页面。
 - c. 在主题列表中，选择您要向其添加订阅者的主题，在右侧“操作”栏单击“添加订阅”。
 - d. 在添加订阅的对话框中，配置协议规则，“协议”下拉框中选择您需要的协议。
 - e. 在“订阅终端”输入框中输入对应的订阅终端。
更多订阅参数说明请参考 [SMN-订阅主题](#)。
添加订阅后，消息通知服务会向订阅终端发送订阅确认信息，信息中包含订阅确认的链接。订阅确认的链接在48小时内有效，用户需要及时在手机端、邮箱或其他协议终端确认订阅。

步骤2 在AOM管理控制台创建告警行动规则。

创建告警行动规则并关联SMN主题与消息模板，当日志、资源或指标数据满足对应的告警条件时，系统根据关联的SMN主题与消息模板来发送告警通知。

请确保已创建SMN主题并已为主题添加订阅。

1. 登录AOM 2.0控制台。
2. 在左侧导航栏中选择“告警管理 > 告警行动规则”。
3. 在右侧区域的“告警行动规则”页签下，单击“创建告警行动规则”。
4. 设置行动规则名称、类型、行动方式等信息。
详细参数说明请参考[AOM-创建告警行动规则](#)。

当资源触发对应的告警条件时，系统根据关联SMN主题根据关联SMN主题与消息模板来发送告警通知。

步骤3 创建指标告警规则。

通过指标告警规则可对资源的指标设置阈值条件。当指标数据满足阈值条件时产生阈值告警，当没有指标数据上报时产生数据不足事件。

AOM创建指标告警规则可分为两种：[按全量指标创建](#)、[按Prometheus命令创建](#)。

本例以按全量指标创建的方式为例。

1. 登录AOM 2.0控制台。
2. 在左侧导航栏中选择“告警管理 > 告警规则”。
3. 单击“创建”。
4. 设置告警规则基本信息，并配置告警规则的详细信息。

详细参数说明请参考[按全量指标创建](#)。

- 配置告警规则时选择的Prometheus实例应是需要配置告警通知的作业所在的弹性资源池绑定的Prometheus 实例。
- 配置高级设置：仅”全量指标创建”的方式支持该配置项，配置时建议开启无数据处理。即配置监控周期内无指标数据产生或指标数据不足时系统的处理方式。
- 告警通知的行动规则：建议开启告警通知的行动规则，确保告警时可以通过邮件或者短信等方式获取通知。配置时选择[步骤2](#)中配置的告警行动规则。

---结束

相关操作

预定义仪表盘不能满足业务需求时，您可以按需自定义仪表盘。具体操作请参考[自定义仪表盘](#)。

12.2 DLI 对接 AOM Prometheus 监控的配置项

在配置DLI对接AOM Prometheus监控时，系统会自动完成[DLI对接AOM Prometheus 监控的配置项](#)中的参数配置。如果这些默认配置不满足您的需求，您可以在Flink作业的“自定义配置”中手动配置以下参数，且优先以您的配置为准。

表 12-3 DLI 对接 AOM Prometheus 监控的配置项

参数	是否必选	默认值	数据类型	默认值	说明
metrics.reporter.remote.class	是	无	String	com.huawei.flink.metrics.prometheus.remote.PrometheusRemoteReporter	metric reporter的类名。
metrics.reporter.remote.factory.class	是	无	String	com.huawei.flink.metrics.prometheus.remote.PrometheusRemoteReporterFactory	metric reporter所对应的Factory类型。
metrics.reporter.remote.interval	是	无	Duration	30 SECONDS	指标上报周期，建议为30秒，若太低会上报过于频繁。

参数	是否必选	默认值	数据类型	默认值	说明
metrics.reporter.remote.write-url	是	无	String	-	AOM Prometheus 通用类型的remote write url。 自定义配置url的时候url里不需要添加https://，系统会自动拼接https。 配置样例： aom-internal-access. {regionId}.xxxx.com:8xx 3/v1/{projectId}/ {prometheusId}/push
metrics.reporter.remote.report-all-metrics	否	false	Boolean	false	是否上报所有指标。默认为false，即只上报基础指标。
metrics.reporter.remote.pool-name	否	None	String	-	为指标加上当前作业所在的弹性资源池名称作为标签。
metrics.reporter.remote.dli-job-id	否	None	String	-	为指标加上当前作业的DLI Flink作业ID作为标签。
metrics.reporter.remote.dli-job-name	否	None	String	-	为指标加上当前作业的DLI Flink作业名称作为标签。

12.3 DLI 支持的 Prometheus 基础监控指标

表12-4提供了DLI支持的Prometheus基础监控指标，AOM Prometheus支持免费存储基础指标。

除基础指标外，AOM Prometheus提供的自定义指标按[计费规则](#)付费使用。

表 12-4 DLI 支持的 Prometheus 监控指标

监控指标	指标含义
flink_jobmanager_Status_JVM_CPU_Load	JobManager CPU的负载
flink_jobmanager_Status_JVM_CPU_Time	JobManager CPU的使用
flink_taskmanager_Status_JVM_CPU_Load	TaskManager CPU的负载
flink_taskmanager_Status_JVM_CPU_Time	TaskManager CPU的使用

监控指标	指标含义
flink_jobmanager_Status_JVM_Memory_Heap_Used	JobManager的堆内存使用量
flink_jobmanager_Status_JVM_Memory_Heap_Committed	保证JobManager的JVM可用的堆内存量
flink_jobmanager_Status_JVM_Memory_Heap_Max	JobManager中可用于内存管理的最大堆内存量
flink_jobmanager_Status_JVM_Memory_NonHeap_Used	JobManager的堆外内存使用量
flink_jobmanager_Status_JVM_Memory_NonHeap_Committed	保证JobManager的JVM可用的堆外内存量
flink_jobmanager_Status_JVM_Memory_NonHeap_Max	JobManager中可用于内存管理的最大堆外内存量
flink_jobmanager_Status_JVM_Memory_Metaspace_Used	JobManager MetaSpace内存池中当前使用的内存量
flink_jobmanager_Status_JVM_Memory_Metaspace_Committed	JobManager MetaSpace内存池中保证可供JVM使用的内存量
flink_jobmanager_Status_JVM_Memory_Metaspace_Max	JobManager MetaSpace内存池中可以使用的最大内存量
flink_jobmanager_Status_JVM_Memory_Direct_Count	JobManager direct缓冲池中的缓冲区数
flink_jobmanager_Status_JVM_Memory_Direct_MemoryUsed	JobManager中JVM用于direct缓冲池的内存量
flink_jobmanager_Status_JVM_Memory_Direct_TotalCapacity	JobManager中direct缓冲池中所有缓冲区的总容量
flink_jobmanager_Status_JVM_Memory_Mapped_Count	JobManager中mapped缓冲池中的缓冲区个数
flink_jobmanager_Status_JVM_Memory_Mapped_MemoryUsed	JobManager中JVM用于mapped缓冲池的内存量
flink_jobmanager_Status_JVM_Memory_Mapped_TotalCapacity	JobManager中mapped缓冲池中所有缓冲区的总容量
flink_jobmanager_Status_Flink_Memory_Managed_Used	JobManager中已使用的托管内存量
flink_jobmanager_Status_Flink_Memory_Managed_Total	JobManager中托管内存总量
flink_taskmanager_Status_JVM_Memory_Heap_Used	TaskManager的堆内存使用量
flink_taskmanager_Status_JVM_Memory_Heap_Committed	保证TaskManager的JVM可用的堆内存量

监控指标	指标含义
flink_taskmanager_Status_JVM_Memory_Heap_Max	TaskManager中可用于内存管理的最大堆内存量
flink_taskmanager_Status_JVM_Memory_NonHeap_Used	TaskManager的堆外内存使用量
flink_taskmanager_Status_JVM_Memory_NonHeap_Committed	保证TaskManager的JVM可用的堆外内存量
flink_taskmanager_Status_JVM_Memory_NonHeap_Max	TaskManager中可用于内存管理的最大堆外内存量
flink_taskmanager_Status_JVM_Memory_Metaspace_Used	TaskManager MetaSpace内存池中当前使用的内存量
flink_taskmanager_Status_JVM_Memory_Metaspace_Committed	TaskManager MetaSpace内存池中保证可供JVM使用的内存量
flink_taskmanager_Status_JVM_Memory_Metaspace_Max	TaskManager MetaSpace内存池中可以使用的最大内存量
flink_taskmanager_Status_JVM_Memory_Direct_Count	TaskManager direct缓冲池中的缓冲区数
flink_taskmanager_Status_JVM_Memory_Direct_MemoryUsed	TaskManager中JVM用于direct缓冲池的内存量
flink_taskmanager_Status_JVM_Memory_Direct_TotalCapacity	TaskManager中direct缓冲池中所有缓冲区的总容量
flink_taskmanager_Status_JVM_Memory_Mapped_Count	TaskManager中mapped缓冲池中的缓冲区个数
flink_taskmanager_Status_JVM_Memory_Mapped_MemoryUsed	TaskManager中JVM用于mapped缓冲池的内存量
flink_taskmanager_Status_JVM_Memory_Mapped_TotalCapacity	TaskManager中mapped缓冲池中所有缓冲区的总容量
flink_taskmanager_Status_Flink_Memory_Managed_Used	TaskManager中已使用的托管内存量
flink_taskmanager_Status_Flink_Memory_Managed_Total	TaskManager中托管内存总量
flink_jobmanager_Status_JVM_Threads_Count	JobManager中活动的线程总数
flink_taskmanager_Status_JVM_Threads_Count	TaskManager中活动中的线程总数
flink_jobmanager_Status_JVM_GarbageCollector_ConcurrentMarkSweep_Count	JobManager CMS垃圾回收器的回收次数
flink_jobmanager_Status_JVM_GarbageCollector_ConcurrentMarkSweep_Time	JobManager CMS执行垃圾回收总耗时

监控指标	指标含义
flink_jobmanager_Status_JVM_GarbageCollector_ParNew_Count	JobManager GC次数
flink_jobmanager_Status_JVM_GarbageCollector_ParNew_Time	JobManager每次GC时间
flink_taskmanager_Status_JVM_GarbageCollector_ConcurrentMarkSweep_Count	TaskManager CMS垃圾回收器的回收次数
flink_taskmanager_Status_JVM_GarbageCollector_ConcurrentMarkSweep_Time	TaskManager CMS执行垃圾回收总耗时
flink_taskmanager_Status_JVM_GarbageCollector_ParNew_Count	TaskManager GC次数
flink_taskmanager_Status_JVM_GarbageCollector_ParNew_Time	TaskManager每次GC时间
flink_jobmanager_Status_JVM_ClassLoader_ClassesLoaded	JobManager自JVM启动以来加载的类的总数
flink_jobmanager_Status_JVM_ClassLoader_ClassesUnloaded	JobManager自JVM启动以来卸载的类的总数
flink_taskmanager_Status_JVM_ClassLoader_ClassesLoaded	TaskManager自JVM启动以来加载的类的总数
flink_taskmanager_Status_JVM_ClassLoader_ClassesUnloaded	TaskManager自JVM启动以来卸载的类的总数
flink_taskmanager_Status_Network_AvailableMemorySegments	TaskManager未使用的内存segments的个数
flink_taskmanager_Status_Network_TotalMemorySegments	TaskManager中分配的内存segments的总数
flink_taskmanager_Status_Shuffle_Netty_AvailableMemorySegments	TM未使用的内存segments的个数
flink_taskmanager_Status_Shuffle_Netty_UsedMemorySegments	TM已使用的内存segments的个数
flink_taskmanager_Status_Shuffle_Netty_TotalMemorySegments	TM分配的内存segments的个数
flink_taskmanager_Status_Shuffle_Netty_AvailableMemory	TM中未使用的内存量
flink_taskmanager_Status_Shuffle_Netty_UsedMemory	TM中已使用的内存量
flink_taskmanager_Status_Shuffle_Netty_TotalMemory	TM中分配的内存量
flink_jobmanager_job_numRestarts	自作业提交以来的重新启动总数

监控指标	指标含义
flink_jobmanager_job_lastCheckpointDuration	完成最新checkpoint所用的时间
flink_jobmanager_job_lastCheckpointSize	最新checkpoint的大小，如果启用了增量检查点或更改日志，则此度量可能与lastCheckpointFullSize不同。
flink_jobmanager_job_numberOfInProgressCheckpoints	正在进行的checkpoint的数量
flink_jobmanager_job_numberOfCompletedCheckpoints	成功完成的checkpoint的数量
flink_jobmanager_job_numberOfFailedCheckpoints	失败的checkpoint的数量
flink_jobmanager_job_totalNumberOfCheckpoints	所有checkpoint的总数
flink_taskmanager_job_task_numBytesOutput	Task输出的字节总数
flink_taskmanager_job_task_numBytesOutputPerSecond	Task每秒输出的字节总数
flink_taskmanager_job_task_isBackPressured	Task是否反压
flink_taskmanager_job_task_numRecordsIn	Task收到的记录总数
flink_taskmanager_job_task_numRecordsInPerSecond	Task每秒收到的记录总数
flink_taskmanager_job_task_numBytesIn	Task收到的字节数
flink_taskmanager_job_task_numBytesInPerSecond	Task每秒收到的字节数
flink_taskmanager_job_task_numRecordsOut	Task发出的记录总数
flink_taskmanager_job_task_numRecordsOutPerSecond	Task每秒发出的记录总数
flink_taskmanager_job_task_operator_numRecordsIn	Operator收到的记录总数
flink_taskmanager_job_task_operator_numRecordsInPerSecond	Operator每秒收到的记录总数
flink_taskmanager_job_task_operator_numRecordsOut	Operator发出的记录总数
flink_taskmanager_job_task_operator_numRecordsOutPerSecond	Operator每秒发出的记录总数

监控指标	指标含义
flink_taskmanager_job_task_operator_sourceIdleTime	Source 闲置时长
flink_taskmanager_job_task_operator_currentEmitEventTimeLag	数据的事件时间与数据离开 Source 时的间隔
flink_taskmanager_job_task_operator_pendingRecords	尚未被 Source 拉取的数据数量

13 使用 CTS 审计 DLI 服务

通过云审计服务，您可以记录与DLI服务相关的操作事件，便于日后的查询、审计和回溯。

表 13-1 云审计服务支持的 DLI 操作列表

操作名称	资源类型	事件名称
创建数据库	database	createDatabase
删除数据库	database	deleteDatabase
修改数据库所有者	database	alterDatabaseOwner
创建表	table	createTable
删除表	table	deleteTable
导出表数据	table	exportData
导入表数据	table	importData
修改表的所有者	table	alterTableOwner
创建队列	queue	createQueue
删除队列	queue	deleteQueue
队列授权	queue	queueAuthorize
修改队列网段	queue	replaceQueue
重启队列	queue	queueActions
扩容/缩容队列	queue	queueActions
提交作业（SQL）	queue	submitJob
取消作业（SQL）	jobs	cancelJob
授权obs桶给DLI服务	obs	authorizeObsBucketsForStream

操作名称	资源类型	事件名称
检查SQL语法	jobs	checkSQL
删除作业	jobs	deleteStreamJob
创建Flink opensource sql 作业	jobs	createStreamSqlJob
更新Flink opensource sql 作业	jobs	updateStreamSqlJob
批量删除Flink作业	jobs	deleteStreamJobs
停止Flink作业	jobs	stopStreamJobs
提交Flink作业	jobs	submitStreamJobs
创建Flink jar作业	jobs	createStreamJarJob
更新Flink jar作业	jobs	updateStreamJarJob
查看flink作业	jobs	checkStreamJob
导入保存点	jobs	dealSavepoint
购买cu时套餐包	order	orderPackage
数据授权	authorization	dataAuthorize
跨项目数据授权	authorization	projectDataAuthorize
导出查询结果	jobs	storeJobResult
保存SQL模板	template	createTemplate
更新SQL模板	template	updateTemplate
删除SQL模板	template	deleteTemplates
新建Flink模板	template	createStreamTemplate
更新Flink模板	template	updateStreamTemplate
查看Flink模板	template	checkStreamTemplate
删除Flink模板	template	deleteStreamTemplate
创建认证信息并上传证书	datasourceauth	uploadAuthInfo
更新跨源认证信息	datasourceauth	updateAuthInfo
删除跨源认证信息	datasourceauth	deleteAuthInfo
上传资源包	resource	uploadResources
删除资源包	resource	deleteResource
创建增强型跨源连接	edsconnection	createConnection
删除增强型跨源连接	edsconnection	deleteConnection

操作名称	资源类型	事件名称
创建经典型跨源连接	edsconnection	createConnection
删除经典型跨源连接	edsconnection	deleteConnection
绑定队列	edsconnection	associateQueueToConnection
解绑队列	edsconnection	disassociateQueueToConnection
修改主机信息	edsconnection	updateHostInfo
添加路由	edsconnection	addRoute
删除路由	edsconnection	deleteRoute
创建批处理作业	jobs	createBatch
取消批处理作业	jobs	cancelBatch
创建全局变量	variable	createGlobalVariable
删除全局变量	variable	deleteGlobalVariable
修改全局变量	variable	updateGlobalVariable

关于如何开通云审计服务以及如何查看追踪事件，请参考《[云审计服务快速入门](#)》中的相关章节。

关于云审计服务事件结构的关键字段详解，请参见《云审计服务用户指南》中的[事件结构](#)和[事件样例](#)。

14 权限管理

14.1 权限管理概述

DLI服务不仅在服务本身有一套完善的权限控制机制，同时还支持通过统一身份认证服务（Identity and Access Management，简称IAM）细粒度鉴权，可以通过在IAM创建策略来管理DLI的权限控制。两种权限控制机制可以共同使用，没有冲突。

IAM 鉴权使用场景

企业用户在华为云上使用DLI服务时，需要对不同部门的员工使用DLI资源（队列）进行管理，包括资源的创建、删除、使用、隔离等。同时，也需要对不同部门的数据进行管理，包括数据的隔离、共享等。

DLI使用IAM进行精细的企业级多租户管理。该服务提供用户身份认证、权限分配、访问控制等功能，可以帮助您安全地控制华为云资源的访问。

通过IAM，您可以在华为云账号中给员工创建IAM用户，并使用策略来控制他们对华为云资源的访问范围。例如您的员工中有负责软件开发的人员，您希望他们拥有DLI的使用权限，但是不希望他们拥有删除DLI等高危操作的权限，那么您可以使用IAM为开发人员创建用户，通过授予仅能使用DLI，但是不允许删除DLI的权限策略，控制他们对DLI资源的使用范围。

说明

对于新建的用户，需要先登录一次DLI，记录元数据，后续才可正常使用。

IAM是华为云提供权限管理的基础服务，无需付费即可使用，您只需要为您账号中的资源进行付费。关于IAM的详细介绍，请参见《[IAM产品介绍](#)》。

如果华为云账号已经能满足您的需求，不需要创建独立的IAM用户进行权限管理，您可以跳过本章节，不影响您使用DLI服务的其他功能。

DLI 系统权限

如[表14-1](#)所示，包括了DLI的所有系统权限。

权限类别：根据授权精程度分为角色和策略。

- 角色：IAM最初提供的一种根据用户的工作职能定义权限的粗粒度授权机制。该机制以服务为粒度，提供有限的服务相关角色用于授权。由于华为云各服务之间

存在业务依赖关系，因此给用户授予角色时，可能需要一并授予依赖的其他角色，才能正确完成业务。角色并不能满足用户对精细化授权的要求，无法完全达到企业对权限最小化的安全管控要求。

- 策略：IAM最新提供的一种细粒度授权的能力，可以精确到具体服务的操作、资源以及请求条件等。基于策略的授权是一种更加灵活的授权方式，能够满足企业对权限最小化的安全管控要求。例如：针对DLI服务，管理员能够控制IAM用户仅能对某一类云服务器资源进行指定的管理操作。

了解DLI SQL常用操作与系统策略的授权关系，请参考[常用操作与系统权限关系](#)。

表 14-1 DLI 系统权限

系统角色/策略名称	描述	类别	依赖关系
DLI FullAccess	数据湖探索所有权限。	系统策略	该角色有依赖，需要在同项目中勾选依赖的角色： <ul style="list-style-type: none"> ● 创建跨源连接：VPC ReadOnlyAccess ● 创建包年/包月资源：BSS Administrator ● 创建标签：TMS FullAccess、EPS EPS FullAccess ● 使用OBS存储：OBS OperateAccess ● 创建委托：Security Administrator
DLI ReadOnlyAccess	数据湖探索只读权限。 只读权限可控制部分开放的、未鉴权的DLI资源和操作。例如创建全局变量、创建程序包以及程序包组、default队列提交作业、default数据库下建表、创建跨源连接、删除跨源连接等操作。	系统策略	无
Tenant Administrator	租户管理员。 <ul style="list-style-type: none"> ● 操作权限：具有数据湖探索服务资源的所有执行权限。创建后，可通过ACL赋权给其他子用户使用。 ● 作用范围：项目级服务。 	系统角色	无

系统角色/策略名称	描述	类别	依赖关系
DLI Service Administrator	数据湖探索管理员。 <ul style="list-style-type: none"> 操作权限：具有数据湖探索服务资源的所有执行权限。创建后，可通过ACL赋权给其他子用户使用。 作用范围：项目级服务。 	系统角色	无

具体的授权方式请参考[创建IAM用户并授权使用DLI](#)以及《[如何创建子用户](#)》和《[如何修改用户策略](#)》。

DLI 权限分类

DLI服务权限分类如[表14-2](#)所示，其可控制的资源请参考[表14-7](#)。

表 14-2 DLI 权限分类

权限大类	权限小类	控制台操作	SQL语法
队列权限	队列管理权限	请参考 队列权限管理	无
	队列使用权限		
数据权限	数据库权限	请参考 在DLI控制台配置数据库权限和在DLI控制台配置表权限	请参考《 权限列表 》。
	表权限		
	列权限		
作业权限	Flink作业	请参考 配置Flink作业权限	无
程序包权限	程序包组权限	请参考 配置DLI程序包权限	无
	程序包权限		
跨源认证权限	跨源认证权限	请参考 跨源认证权限管理	无

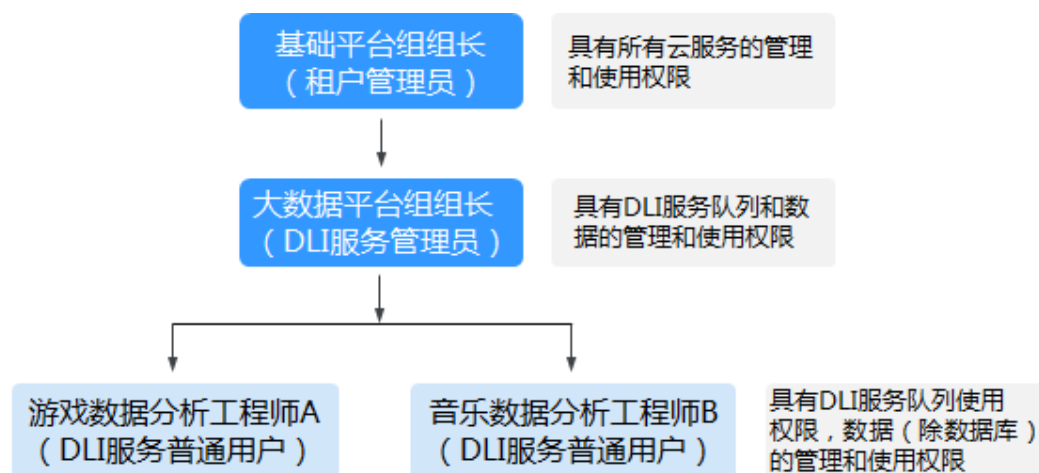
场景举例

某互联网公司，主要有游戏和音乐两大业务，使用DLI服务进行用户行为分析，辅助决策。

如[图14-1](#)所示，“基础平台组组长”在华为云上申请了一个“租户管理员”（Tenant Administrator）账号，用于管理和使用华为云的各个服务。因为“大数据平台组”需要使用DLI进行数据分析，所有“基础平台组组长”增加了一个权限为“DLI服务管理

员”（DLI Service Administrator）的子账号用于管理和使用DLI服务。“基础平台组组长”按照公司两个业务对于数据分析的要求，创建了“队列A”分配给“数据工程师A”运行游戏数据分析业务，“队列B”分配给“数据工程师B”运行音乐数据分析业务，并分别赋予“DLI普通用户”权限，具有队列使用权限，数据（除数据库）的管理和使用权限。

图 14-1 权限分配



“数据工程师A”创建了一个gameTable表用于存放游戏道具相关数据，userTable表用于存放游戏用户相关数据。因为音乐业务是一个新业务，想在存量的游戏用户中挖掘一些潜在的音乐用户，所以“数据工程师A”把userTable表的查询权限赋给了“数据工程师B”。同时，“数据工程师B”创建了一个musicTable用于存放音乐版权相关数据。

“数据工程师A”和“数据工程师B”对于队列和数据的使用权限如表14-3所示。

表 14-3 使用权限说明

用户	数据工程师A（游戏数据分析）	数据工程师B（音乐数据分析）
队列	队列A（队列使用权限）	队列B（队列使用权限）
数据（表）	gameTable（表管理和使用权限）	musicTable（表管理和使用权限）
	userTable（表管理和使用权限）	userTable（表查询权限）

说明

队列的使用权限包括提交作业和终止作业两个权限。

14.2 DLI 自定义策略

如果系统预置的DLI权限，不满足您的授权要求，可以创建自定义策略。自定义策略中可以添加的授权项（Action）请参考[权限策略和授权项](#)。

目前华为云支持以下两种方式创建自定义策略：

- 可视化视图创建自定义策略：无需了解策略语法，按可视化视图导航栏选择云服务、操作、资源、条件等策略内容，可自动生成策略。
- JSON视图创建自定义策略：可以在选择策略模板后，根据具体需求编辑策略内容；也可以直接在编辑框内编写JSON格式的策略内容。

具体创建步骤请参见：[创建自定义策略](#)。本章为您介绍常用的DLI自定义策略样例。

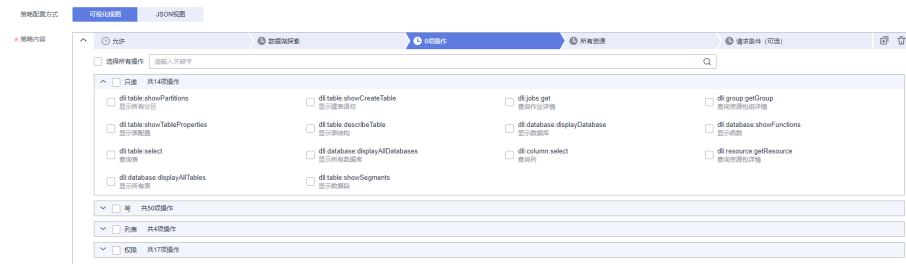
策略字段介绍

以授权用户拥有在所有区域中所有数据库的创建表权限为例进行说明：

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dli:database:createTable"
      ],
      "Resource": [
        "dli:*:*:database:*"
      ]
    }
  ]
}
```

- **Version**
版本信息，1.1：策略。IAM最新提供的一种细粒度授权的能力，可以精确到具体服务的操作、资源以及请求条件等。
- **Effect**
作用。包含两类：允许（Allow）和拒绝（Deny），既有Allow又有Deny的授权语句时，遵循Deny优先的原则。
- **Action**
授权项，指对资源的具体操作权限，不超过100个，如[图14-2](#)所示。

图 14-2 DLI 授权项



说明

- 格式为：服务名:资源类型:操作，例：dli.queue:submit_job。
- 服务名为产品名称，例如dli、evs和vpc等，服务名仅支持小写。资源类型和操作没有大小写，要求支持通配符号*，无需罗列全部授权项。
- 资源类型可以参考[表14-7](#)中的资源类型。
- 操作：操作以IAM服务中已经注册的action为准。

- Condition

限制条件：使策略生效的特定条件，包括条件键和运算符。

条件键表示策略语句的 Condition 元素中的键值，分为全局级条件键和服务级条件键。

- 全局级条件键（前缀为g:）适用于所有操作。详细请参考[策略语法](#)中的条件键说明。
- 服务级条件键，仅适用于对应服务的操作。

运算符与条件键一起使用，构成完整的条件判断语句。具体内容请参考[表14-4](#)。

DLI通过IAM预置了一组条件键。下表显示了适用于DLI服务特定的条件键。

表 14-4 DLI 请求条件

DLI条件键	类型	运算符	描述
g:CurrentTime	全局级	Date and time	接收到鉴权请求的时间。 说明 以“ISO 8601”格式表示，例如： 2012-11-11T23:59:59Z。
g:MFAPresent	全局级	Boolean	用户登录时是否使用了多因素认证。
g:UserId	全局级	String	当前登录的用户ID。
g:UserName	全局级	String	当前登录的用户名。
g:ProjectName	全局级	String	当前登录的Project。
g:DomainName	全局级	String	当前登录的Domain。
g:ResourceTag	全局级	StringEquals	资源标签键值。

- Resource

格式为：服务名:region:domainId:资源类型:资源路径, 通配符号*表示所有，资源类型和资源路径可以参考[表14-7](#)。

示例：

"dli:*:*:queue:*": 表示所有的队列。

创建 DLI 自定义策略

用户可以根据场景设置不同级别的Action和Resource。

1. 定义Action

Action由服务名:资源类型:操作三段组成，通配符为*。例如：

表 14-5 Action

Action	说明
dli:queue:submit_job	DLI队列的提交操作
dli:queue:*	DLI队列的全部操作
dli:*:*	DLI所有资源类型的所有操作

更多操作与系统权限的关系请参考[常用操作与系统权限关系](#)。

2. 定义Resource

Resource由<服务名:region:domainId:资源类型:资源路径>5个字段组成，通配符*表示所有资源。5个字段可以灵活设置，资源路径可以按照场景需要，设置不同级别的权限控制。当需要设置该服务下的所有资源时，可以不指定该字段。Resource定义请参考表14-6。Resource中的资源类型和资源路径请参考表14-7。

表 14-6 Resource

Resource	说明
DLI:*:*:table:databases.dbname.tables.*	DLI服务，任意region，任意账号ID下，数据库名为dbname下的所有表资源。
DLI:*:*:database:databases.dbname	DLI服务，任意region，任意账号ID下，数据库名为dbname的资源。
DLI:xxx:xxx:column:databases.db.tables.tb.columns.col	DLI服务，指定region，账号ID为xxx，数据库名为db，表名为tb，列名为col的资源。
DLI:*:*:queue:queues.*	DLI服务，任意region，任意账号ID下，任意队列资源。
DLI:*:*:jobs:jobs.flink.1	DLI服务，任意region，任意账号ID下，作业Id为1的flink作业。

表 14-7 DLI 的指定资源与对应路径

资源类型	资源名称	资源路径
queue	DLI队列	queues.queueName
database	DLI数据库	databases.dbname

资源类型	资源名称	资源路径
table	DLI表	databases.dbname.tables.tbname
column	DLI列	databases.dbname.tables.tbname.columns.columnname
jobs	DLI Flink作业	jobs.flink.jobid
resource	DLI程序包	resources.resourcename
group	DLI程序包组	groups.groupname
datasourceauth	DLI跨源认证信息	datasourceauth.name
edsconnections	DLI增强跨源	edsconnections.连接ID

- 特定资源：

图 14-3 特定资源



- 所有资源： 指该服务下的所有资源

图 14-4 所有资源



3. 将上述的所有字段拼接为一个json就是一个完整的策略了，其中action和resource均可以设置多个，当然也可以通过IAM提供的可视化界面进行创建，例如：
授权用户拥有DLI服务，任意region，任意账号ID下，任意数据库的创建删除权限，任意队列的提交作业权限，任意表的删除权限。

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dli:database:createDatabase",
        "dli:database:dropDatabase",
        "dli:queue:submitJob",
        "dli:table:dropTable"
      ],
      "Resource": [
        "dli:*:database:*",
        "dli:*:queue:*",
        "dli:*:table:*"
      ]
    }
  ]
}
```



```

    ]
  }
]
}

```

DLI 自定义策略样例

- 示例1：允许

- 授权用户拥有在所有区域中所有数据库的创建表权限。

```

{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dli:database:createTable"
      ],
      "Resource": [
        "dli:*:*:database:*"
      ]
    }
  ]
}

```

- 授权用户拥有在所在区域中数据库db中表tb中列col的查询权限。

```

{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dli:column:select"
      ],
      "Resource": [
        "dli:xxx:*:column:databases.db.tables.tb.columns.col"
      ]
    }
  ]
}

```

- 示例2：拒绝

拒绝策略需要同时配合其他策略使用，即用户需要先被授予部分操作权限策略后，才可以在权限内设置拒绝策略，否则用户无任何权限的情况下，拒绝策略无实际作用。

用户被授予的策略中，一个授权项的作用如果同时存在Allow和Deny，则遵循Deny优先。

- 授权用户不能创建数据库，删除数据库，提交作业（default队列除外），删除表。

```

{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "dli:database:createDatabase",
        "dli:database:dropDatabase",
        "dli:queue:submitJob",
        "dli:table:dropTable"
      ],
      "Resource": [
        "dli:*:*:database:*",
        "dli:*:*:queue:*",
        "dli:*:*:table:*"
      ]
    }
  ]
}

```

- ```
]
 }
- 授权用户不能在队列为demo的队列上提交作业。
```

```
{
 "Version": "1.1",
 "Statement": [
 {
 "Effect": "Deny",
 "Action": [
 "dli:queue:submitJob"
],
 "Resource": [
 "dli:*:*:queue:queues.demo"
]
 }
]
}
```

- 示例3: 标签鉴权, 指定action绑定Condition, 指定g:ResourceTag的key和value。

Condition g:ResourceTag使用表示带有标签key=value的资源, 并且资源操作在策略action中包含的可以鉴权通过。

key不区分大小写, 并且目前不支持value的模糊匹配。

```
{
 "Version": "1.1",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "dli:database:dropDatabase",
 "dli:table:select",
 "dli:database:createTable",
 "dli:table:dropTable"
],
 "Condition": {
 "StringEquals": {
 "g:ResourceTag/key": [
 "value"
]
 }
 }
 }
]
}
```

## 相关链接

- [策略基本概念](#)
- [RBAC策略语法](#)
- [细粒度策略语法](#)
- [创建自定义策略](#)

## 14.3 DLI 资源

资源是服务中存在的对象。在DLI中, 资源如下, 您可以在创建自定义策略时, 通过指定资源路径来选择特定资源。

表 14-8 DLI 的指定资源与对应路径

| 资源类型           | 资源名称        | 资源路径                                           |
|----------------|-------------|------------------------------------------------|
| queue          | DLI队列       | queues.queueName                               |
| database       | DLI数据库      | databases.dbName                               |
| table          | DLI表        | databases.dbName.tables.tbName                 |
| column         | DLI列        | databases.dbName.tables.tbName.columns.colName |
| jobs           | DLI Flink作业 | jobs.flink.jobId                               |
| resource       | DLI程序包      | resources.resourceName                         |
| group          | DLI程序包组     | groups.groupName                               |
| dataSourceAuth | DLI跨源认证信息   | dataSourceAuth.name                            |
| edsconnections | DLI增强跨源     | edsconnections.连接ID                            |

## 14.4 DLI 请求条件

您可以在创建自定义策略时，通过添加“请求条件”（Condition元素）来控制策略何时生效。请求条件包括条件键和运算符，条件键表示策略语句的 Condition 元素，分为全局级条件键和服务级条件键。**全局级条件键**（前缀为g:）适用于所有操作，**服务级条件键**（前缀为服务缩写，如dli）仅适用于对应服务的操作。运算符与条件键一起使用，构成完整的条件判断语句。

DLI通过IAM预置了一组条件键。下表显示了适用于DLI服务特定的条件键。

表 14-9 DLI 请求条件

| DLI条件键        | 类型  | 运算符           | 描述                                                                       |
|---------------|-----|---------------|--------------------------------------------------------------------------|
| g:CurrentTime | 全局级 | Date and time | 接收到鉴权请求的时间。<br><b>说明</b><br>以“ISO 8601”格式表示，例如：<br>2012-11-11T23:59:59Z。 |
| g:MFAPresent  | 全局级 | Boolean       | 用户登录时是否使用了多因素认证。                                                         |
| g:UserId      | 全局级 | String        | 当前登录的用户ID。                                                               |

| DLI条件键        | 类型  | 运算符          | 描述            |
|---------------|-----|--------------|---------------|
| g:UserName    | 全局级 | String       | 当前登录的用户名。     |
| g:ProjectName | 全局级 | String       | 当前登录的Project。 |
| g:DomainName  | 全局级 | String       | 当前登录的Domain。  |
| g:ResourceTag | 全局级 | StringEquals | 资源标签键值。       |

## 14.5 常用操作与系统权限关系

**表14-10**列出了DLI SQL常用操作与系统策略的授权关系，您可以参照该表选择合适的系统策略。更多SQL语法赋权请参考《[权限列表](#)》章节。

**表 14-10** DLI 常用操作与系统权限的关系

| 资源 | 操作               | 说明            | DLI FullAccess | DLI ReadOnlyAccess | Tenant Administrator | DLI Service Administrator |
|----|------------------|---------------|----------------|--------------------|----------------------|---------------------------|
| 队列 | DROP_QUEUE       | 删除队列          | √              | ×                  | √                    | √                         |
|    | SUBMIT_JOB       | 提交作业          | √              | ×                  | √                    | √                         |
|    | CANCEL_JOB       | 终止作业          | √              | ×                  | √                    | √                         |
|    | RESTART          | 重启队列          | √              | ×                  | √                    | √                         |
|    | GRANT_PRIVILEGE  | 队列的赋权         | √              | ×                  | √                    | √                         |
|    | REVOKE_PRIVILEGE | 队列权限的回收       | √              | ×                  | √                    | √                         |
|    | SHOW_PRIVILEGES  | 查看其他用户具备的队列权限 | √              | ×                  | √                    | √                         |

| 资源              | 操作                 | 说明             | DLI FullAccess | DLI ReadOnlyAccess | Tenant Administrator | DLI Service Administrator |
|-----------------|--------------------|----------------|----------------|--------------------|----------------------|---------------------------|
| 数据库             | DROP_DATABASE      | 删除数据库          | √              | ×                  | √                    | √                         |
|                 | CREATE_TABLE       | 创建表            | √              | ×                  | √                    | √                         |
|                 | CREATE_VIEW        | 创建视图           | √              | ×                  | √                    | √                         |
|                 | EXPLAIN            | 将SQL语句解释为执行计划  | √              | ×                  | √                    | √                         |
|                 | CREATE_ROLE        | 创建角色           | √              | ×                  | √                    | √                         |
|                 | DROP_ROLE          | 删除角色           | √              | ×                  | √                    | √                         |
|                 | SHOW_ROLES         | 显示角色           | √              | ×                  | √                    | √                         |
|                 | GRANT_ROLE         | 绑定角色           | √              | ×                  | √                    | √                         |
|                 | REVOKE_ROLE        | 解除角色绑定         | √              | ×                  | √                    | √                         |
|                 | SHOW_USERS         | 显示所有角色和用户的绑定关系 | √              | ×                  | √                    | √                         |
|                 | GRANT_PRIVILEGE    | 数据库的赋权         | √              | ×                  | √                    | √                         |
|                 | REVOKE_PRIVILEGE   | 数据库权限的回收       | √              | ×                  | √                    | √                         |
|                 | SHOW_PRIVILEGES    | 查看其他用户具备的数据库权限 | √              | ×                  | √                    | √                         |
|                 | DISPLAY_ALL_TABLES | 显示数据库中的表       | √              | √                  | √                    | √                         |
|                 | DISPLAY_DATABASE   | 显示数据库          | √              | √                  | √                    | √                         |
| CREATE_FUNCTION | 创建函数               | √              | ×              | √                  | √                    |                           |
| DROP_FUNCTION   | 删除函数               | √              | ×              | √                  | √                    |                           |

| 资源              | 操作                            | 说明        | DLI FullAccess | DLI ReadOnlyAccess | Tenant Administrator | DLI Service Administrator |
|-----------------|-------------------------------|-----------|----------------|--------------------|----------------------|---------------------------|
|                 | SHOW_FUNCTIONS                | 显示所有函数    | √              | ×                  | √                    | √                         |
|                 | DESCRIBE_FUNCTION             | 显示函数详情    | √              | ×                  | √                    | √                         |
| 表               | DROP_TABLE                    | 删除表       | √              | ×                  | √                    | √                         |
|                 | SELECT                        | 查询表       | √              | ×                  | √                    | √                         |
|                 | INSERT INTO TABLE             | 插入        | √              | ×                  | √                    | √                         |
|                 | ALTER_TABLE_ADD_COLUMNS       | 添加列       | √              | ×                  | √                    | √                         |
|                 | INSERT_OVERWRITE_TABLE        | 重写        | √              | ×                  | √                    | √                         |
|                 | ALTER_TABLE_RENAME            | 重命名表      | √              | ×                  | √                    | √                         |
|                 | ALTER_TABLE_ADD_PARTITION     | 在分区表中添加分区 | √              | ×                  | √                    | √                         |
|                 | ALTER_TABLE_RENAME_PARTITION  | 重命名表分区    | √              | ×                  | √                    | √                         |
|                 | ALTER_TABLE_DROP_PARTITION    | 删除分区表的分区  | √              | ×                  | √                    | √                         |
|                 | SHOW_PARTITIONS               | 显示所有分区    | √              | ×                  | √                    | √                         |
|                 | ALTER_TABLE_RECOVER_PARTITION | 恢复表分区     | √              | ×                  | √                    | √                         |
|                 | ALTER_TABLE_SET_LOCATION      | 设置分区路径    | √              | ×                  | √                    | √                         |
| GRANT_PRIVILEGE | 表的赋权                          | √         | ×              | √                  | √                    |                           |

| 资源      | 操作                  | 说明                       | DLI FullAccess | DLI ReadOnlyAccess | Tenant Administrator | DLI Service Administrator |
|---------|---------------------|--------------------------|----------------|--------------------|----------------------|---------------------------|
|         | REVOKE_PRIVILEGE    | 表权限的回收                   | √              | ×                  | √                    | √                         |
|         | SHOW_PRIVILEGES     | 查看其他用户具备的表权限             | √              | ×                  | √                    | √                         |
|         | DISPLAY_TABLE       | 显示表                      | √              | √                  | √                    | √                         |
|         | DESCRIBE_TABLE      | 显示表信息                    | √              | ×                  | √                    | √                         |
| 弹性资源池   | DROP                | 删除弹性资源池                  | √              | ×                  | √                    | √                         |
|         | RESOURCE_MANAGEMENT | 弹性资源池资源管理                | √              | ×                  | √                    | √                         |
|         | SCALE               | 扩缩容弹性资源池                 | √              | ×                  | √                    | √                         |
|         | UPDATE              | 更新弹性资源池                  | √              | ×                  | √                    | √                         |
|         | CREATE              | 创建弹性资源池                  | √              | ×                  | √                    | √                         |
|         | SHOW_PRIVILEGES     | 查看其他用户具备的弹性资源池权限         | √              | ×                  | √                    | √                         |
|         | GRANT_PRIVILEGE     | 赋予指定用户弹性资源池权限            | √              | ×                  | √                    | √                         |
|         | REVOKE_PRIVILEGE    | 移除指定用户弹性资源池权限            | √              | ×                  | √                    | √                         |
| 增强型跨源连接 | BIND_QUEUE          | 增强型跨源连接绑定队列<br>仅用于跨项目授权。 | ×              | ×                  | ×                    | ×                         |

# 15 DLI 常用管理操作

## 15.1 使用自定义镜像增强作业运行环境

### 自定义镜像应用场景

通过下载DLI提供的基础镜像再按需制作自定义镜像，将作业运行需要的依赖（文件、jar包或者软件）、私有能力等内置到自定义镜像中，以此改变Spark作业和Flink作业的容器运行环境，增强作业的功能、性能。

例如，在自定义镜像中加入机器学习相关的Python包或者C库，可以通过这种方式帮助用户实现功能扩展。

#### 📖 说明

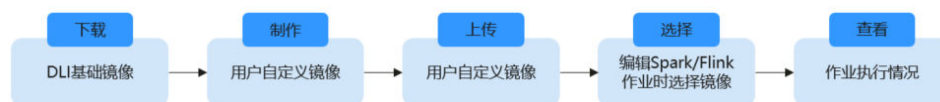
用户使用自定义镜像功能需要具备Docker相关的基础知识。

### 使用限制

- 创建自定义镜像必须使用DLI提供的基础镜像。
- 不能随意修改基础镜像中DLI相关组件及目录。
- 仅支持Spark jar作业、Flink jar作业，即jar包作业。

### 使用流程

图 15-1 自定义镜像使用流程



1. 获取DLI基础镜像。
2. 使用Dockerfile将作业运行需要的依赖（文件、jar包或者软件）打包到镜像中，生成自定义镜像。
3. 将镜像发布到SWR（容器镜像服务）中。



4. 在DLI服务作业编辑页面选择自己生成的镜像，运行作业。
5. 查看作业执行情况。

## 获取 DLI 基础镜像

请根据队列的架构类型选择相同类型的基础镜像。

查看队列的CPU架构类型请参考[查看队列的基本信息](#)。

表 15-1 获取 DLI 基础镜像

| 镜像类型       | 架构  | URL                                                                                                                   |
|------------|-----|-----------------------------------------------------------------------------------------------------------------------|
| general镜像  | X86 | swr.cn-north-4.myhuaweicloud.com/dli-public/spark_general-x86_64:3.3.1-2.3.7.1720240419835647952528832.202404250955   |
| general镜像  | ARM | swr.cn-north-4.myhuaweicloud.com/dli-public/spark_general-aarch64:3.3.1-2.3.7.1720240419835647952528832.202404250955  |
| notebook镜像 | X86 | swr.cn-north-4.myhuaweicloud.com/dli-public/spark_notebook-x86_64:3.3.1-2.3.7.1720240419835647952528832.202404250955  |
| notebook镜像 | ARM | swr.cn-north-4.myhuaweicloud.com/dli-public/spark_notebook-aarch64:3.3.1-2.3.7.1720240419835647952528832.202404250955 |

## 创建自定义镜像

以tensorflow为例，说明如何将tensorflow打包进镜像，生成安装了tensorflow的自定义镜像，在DLI作业中使用该镜像运行作业。

### 步骤1 准备容器环境。

请参考[安装容器引擎](#)文档中的“安装容器引擎”章节。

### 步骤2 使用root用户登录步骤1容器镜像环境，执行以下命令获取DLI的基础镜像。

本示例使用Spark基础镜像为例，使用docker pull方式下载基础镜像到步骤1中的容器镜像环境。


#### docker pull 基础镜像下载地址

基础镜像下载地址参考[使用自定义镜像增强作业运行环境](#)。

示例，Spark基础镜像下载：

```
docker pull swr.xxx/dli-public/spark_general-x86_64:3.3.1-2.3.7.1720240419835647952528832.202404250955
```

### 步骤3 连接容器镜像服务。

1. 登录SWR管理控制台。
2. 选择左侧导航栏的“总览”，单击页面右上角的“登录指令”，在弹出的页面中单击  复制登录指令。
3. 在安装容器引擎的虚拟机中执行上一步复制的登录指令。

**步骤4** 创建容器镜像组织。如果已创建组织则本步骤可以忽略。

1. 登录SWR管理控制台。
2. 选择左侧导航栏的“组织管理”，单击页面右上角的“创建组织”。
3. 填写组织名称，单击“确定”。

**步骤5** 编写Dockerfile文件。

#### vi Dockerfile

具体内容参考如下，将tensorflow打包进镜像：

```
ARG BASE_IMG=swr.xxx/dli-public/spark_general-
x86_64:3.3.1-2.3.7.1720240419835647952528832.202404250955//请替换基础镜像的URL

FROM ${BASE_IMG} as builder
USER omm //注意要使用omm用户执行。
RUN set -ex && \
 mkdir -p /home/omm/.pip && \
 pip3 install tensorflow==2.4.0
内容复制到基础镜像中
USER omm
```

其中，主要包含了以下步骤：

1. 设置pip的可用仓库地址。
2. 使用pip3安装tensorflow算法包。
3. 将安装了算法包的临时镜像builder里的内容复制到基础镜像中（这一步主要是为了减小镜像体积），用于生成最终的自定义镜像。

**步骤6** 利用Dockerfile生成自定义镜像。

镜像打包命令格式：

```
docker build -t [自定义组织名称]/[自定义镜像名称]:[自定义镜像版本] --build-arg BASE_IMG=[DLI基础镜像地址] -f Dockerfile .
```

DLI基础镜像地址为[获取DLI基础镜像](#)中的镜像地址。

示例：

```
docker build -t mydli/spark:2.4 --build-arg BASE_IMG=swr.xxx/dli-public/spark_general-
x86_64:3.3.1-2.3.7.1720240419835647952528832.202404250955 -f Dockerfile .
```

**步骤7** 给自定义镜像打标签。

**docker tag** [步骤6](#)中的[自定义组织名称]/[自定义镜像名称]:[自定义镜像版本] [镜像仓库地址]/[组织名称]/[自定义镜像名称:自定义版本名称]

示例：

```
docker tag mydli/spark:2.4 swr.xxx/testdli0617/spark:2.4.5.tensorflow
```

**步骤8** 上传自定义镜像。

**docker push** [镜像仓库地址]/[组织名称]/[自定义镜像名称:自定义版本名称]

上述命令中的“[镜像仓库地址]/[组织名称]/[自定义镜像名称:自定义版本名称]”保持和[步骤7](#)一致。

示例:

```
docker push swr.xxx/testdli0617/spark:2.4.5.tensorflow
```

**步骤9** 在DLI服务中提交Spark或者Flink jar作业时选择自定义镜像。

- 打开管理控制台的Spark作业或者Flink作业编辑页面，在自定义镜像列表中选择已上传并共享的镜像，运行作业，即可使用自定义镜像运行作业。

如果选择的镜像不是共享镜像，自定义镜像处会提示该镜像未授权，则需要授权后才可以使用。具体可以参考图15-3，提示处单击“立即授权”即可，填写其他作业执行参数后，再执行作业。

图 15-2 在 DLI Spark 作业编辑页面，选择自定义镜像

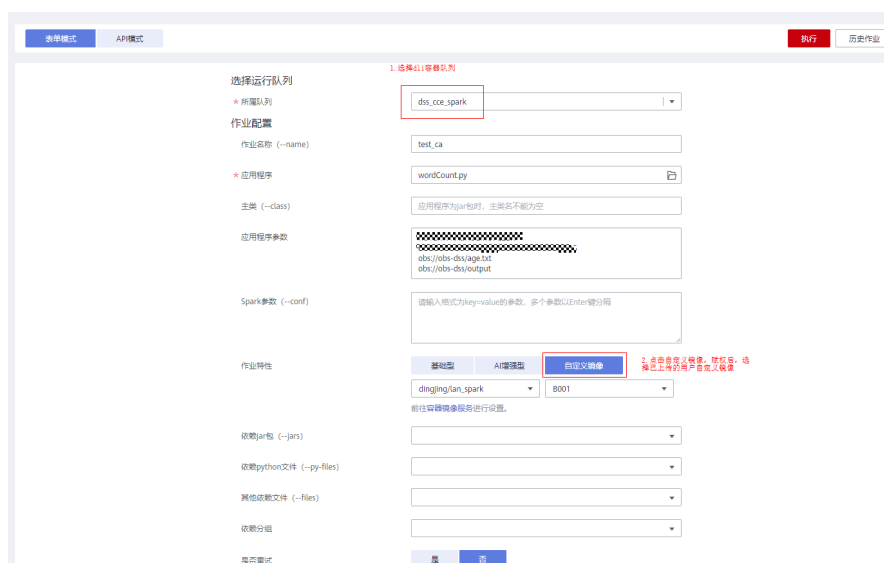


图 15-3 Spark 作业镜像授权操作

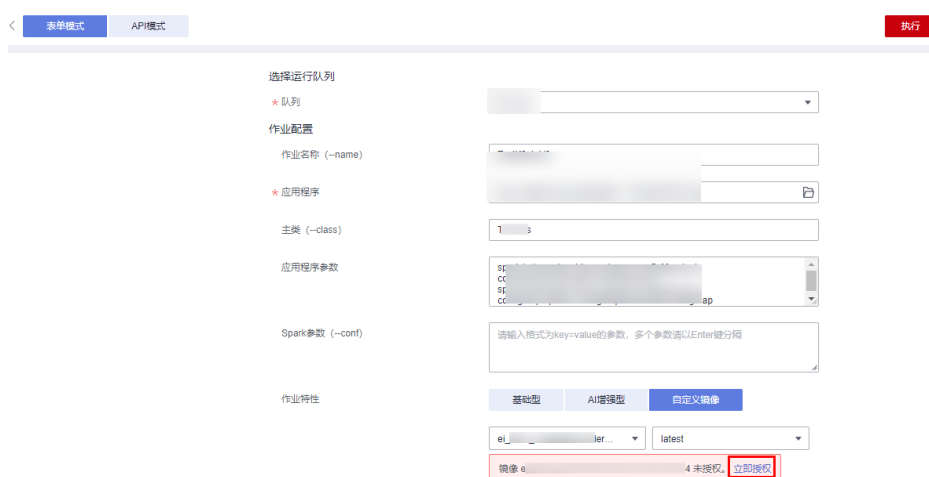


图 15-4 在 DLI Flink jar 作业编辑页面，选择自定义镜像

test\_5  
ID: 83508 | 作业类型: Flink Jar

---

\* 所属队列

\* 应用程序  [查看内置依赖包](#)

主类    
默认类名根据应用程序的Manifest文件指定。

参数

依赖jar包  [查看内置依赖包](#)

其他依赖文件

作业特性

[前往容器镜像服务进行设置。自定义镜像使用指南](#)

Flink版本

优化参数

- 在使用API时，在作业参数中指定image参数，即可使用自定义镜像运行作业。Spark作业请参考《[创建批处理作业](#)》，Flink jar作业请参考《[创建Flink Jar作业](#)》。

----结束

## 15.2 管理 DLI 全局变量

### 什么是全局变量

DLI支持在管理控制台设置全局变量，将作业开发过程中频繁使用的变量设置为全局变量，可以避免在编辑作业过程中重复定义，减少开发与维护成本。通过使用全局变量可以替换长难复杂变量，简化复杂参数，提升SQL语句可读性。

本节操作为您介绍如何创建全局变量。

### 创建全局变量

1. 在DLI控制台左侧导航栏中单击“全局配置 > 全局变量”。

- 在“全局变量”页面，单击右上角“创建变量”，可创建新的全局变量。

表 15-2 全局变量参数说明

| 参数名称 | 描述          |
|------|-------------|
| 变量名称 | 所创建的全局变量名称。 |
| 变量值  | 全局变量的值。     |

- 创建全局变量之后，在SQL语法中使用“{{xxxx}}”代替设置为全局变量的参数值即可，其中“xxxx”为变量名称。例如，在建表语句中，设置表名为全局变量abc，即可用{{abc}}代替实际的表名。

```
create table {{table_name}} (String1 String, int4 int, varchar1 varchar(10))
partitioned by (int1 int,int2 int,int3 int)
```

#### 说明

不推荐在建表语句的OPTIONS关键字中使用全局变量。

#### 相关操作：

##### - 修改全局变量

在“全局变量”页面，单击变量“操作”列中的“修改”，可修改对应的变量值。

#### 说明

如果同账号同项目下存在多个相同名称的全局变量时，需要将多余相同名称的全局变量删除，保证同账号同项目下唯一，此时具备该全局变量修改权限的用户均可以修改对应的变量值。

##### - 删除全局变量

在“全局变量”页面，单击变量“操作”列中的“删除”，可删除对应的变量。

#### 说明

- 如果同账号同项目下存在多个相同名称的全局变量，优先删除用户自建的。如果仅存在唯一名称的全局变量，则具备删除权限即的用户均可删除该全局变量。
- 变量删除后，SQL中将无法使用该变量。

## 全局变量权限管理

针对不同用户，可以通过权限设置分配不同的全局变量，不同用户之间互不影响。管理员用户和全局变量的所有者拥有所有权限，不需要进行权限设置且其他用户无法修改其全局变量权限。

给新用户设置全局变量权限时，该用户所在用户组的所属区域需具有Tenant Guest权限。关于Tenant Guest权限的介绍和开通方法，详细参见《[权限策略](#)》和《[统一身份认证服务用户指南](#)》中的[创建用户组](#)。

#### ● 全局变量用户授权

- 单击“全局配置 > 全局变量”页面，单击全局变量“操作”列中的“权限设置”，进入“用户权限信息”页面，可以对用户进行全局变量的授权、设置权限和回收权限。
- 单击页面右上角“授权”可对用户进行全局变量授权。

图 15-5 全局变量授权



表 15-3 全局变量参数说明

| 参数名称 | 描述                                                                                                                                                                                                                    |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 用户名  | 被授权的IAM用户的名称。<br><b>说明</b><br>该用户名称是已存在的IAM用户名称。                                                                                                                                                                      |
| 权限设置 | <ul style="list-style-type: none"> <li>更新：更新该全局变量。</li> <li>删除：删除该全局变量。</li> <li>赋权：当前用户可将全局变量的权限赋予其他用户。</li> <li>回收：当前用户可回收其他用户具备的该全局变量的权限，但不能回收该全局变量所有者的权限。</li> <li>查看其他用户具备的权限：当前用户可查看其他用户具备的该全局变量的权限。</li> </ul> |

- 设置全局变量权限**  
 单击对应子用户“操作”列中的“权限设置”可修改该用户的权限。详细权限描述如表15-3所示。  
 当“权限设置”中的选项为灰色时，表示您不具备修改此全局变量的权限。可以向管理员用户、组所有者等具有赋权权限的用户申请“全局变量”权限。
- 回收全局变量权限**  
 单击对应子用户“操作”列中的“回收”将删除该用户的权限。该子用户将不具备该全局变量的任意权限。

## 15.3 管理 Jar 作业程序包

### 15.3.1 程序包管理概述

在执行DLI作业前需要将UDF Jar包或Jar作业程序包上传到云平台进行统一的管理和维护。

有以下两种方式管理程序包：

- （推荐使用）上传至OBS管理程序包：提前将对应的jar包上传至OBS桶中，在作业配置时选择对应的OBS路径。

- （DLI程序包功能即将停用）上传至DLI管理程序包：提前将对应的jar包上传至OBS桶中，并在DLI管理控制台的“数据管理>程序包管理”中创建程序包，在作业配置时选择对应的DLI程序包。

本节操作介绍在DLI管理控制台上传并管理程序包的方式。

#### 📖 说明

- DLI程序包功能即将停用，使用Spark3.3.1及以上版本、和Flink1.15及以上版本执行作业时，推荐直接选择OBS中的程序包。
- 打包Spark或Flink jar作业jar包时，请不要上传平台已有的依赖包，以免与平台内置依赖包冲突。内置依赖包信息请参考[DLI内置依赖包](#)。

## 约束与限制

表 15-4 程序包使用约束限制

| 限制项 | 说明                                                                                                                                                                                                                                                                              |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 程序包 | <ul style="list-style-type: none"> <li>• 程序包支持删除，但不支持删除程序包组。</li> <li>• 支持上传的程序包类型：                             <ul style="list-style-type: none"> <li>- JAR：用户jar文件。</li> <li>- PyFile：用户Python文件。</li> <li>- File：用户文件。</li> <li>- ModelFile：用户AI模型文件。</li> </ul> </li> </ul> |

## 程序包管理页面

表 15-5 程序包管理参数

| 参数   | 参数说明                                                                                                                                                                 |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 分组名称 | 程序包所属分组的名称。如果不分组，则显示“--”。                                                                                                                                            |
| 名称   | 程序包名称。                                                                                                                                                               |
| 所有者  | 上传程序包的用户名称。                                                                                                                                                          |
| 包类型  | 程序包的类型。支持的包类型如下： <ul style="list-style-type: none"> <li>• JAR：用户jar文件。</li> <li>• PyFile：用户Python文件。</li> <li>• File：用户文件。</li> <li>• ModelFile：用户AI模型文件。</li> </ul> |
| 状态   | 创建程序包的状态。 <ul style="list-style-type: none"> <li>• 上传中（Uploading）：表示程序包正在上传。</li> <li>• 已成功（Finished）：表示程序包已经上传。</li> <li>• 已失败（Failed）：表示程序包上传失败。</li> </ul>        |

| 参数   | 参数说明                                                                                                                             |
|------|----------------------------------------------------------------------------------------------------------------------------------|
| 创建时间 | 创建程序包的时间。                                                                                                                        |
| 更新时间 | 更新程序包的时间。                                                                                                                        |
| 操作   | 权限管理：对程序包用户进行权限管理。<br>删除：删除程序包。<br>更多： <ul style="list-style-type: none"><li>● 修改所有者：修改程序包用户。</li><li>● 标签：添加或编辑程序包标签。</li></ul> |

## 15.3.2 创建 DLI 程序包

DLI支持用户通过批处理方式将程序包提交至通用队列中运行。

### 说明

如果用户需要更新程序包，可以使用相同的程序包或文件上传至DLI的同一个位置（同一个分组），直接覆盖原有的程序包或文件。

### 前提条件

所使用的程序包需提前上传至OBS服务中保存。

### 创建程序包步骤

1. 在管理控制台左侧，单击“数据管理” > “程序包管理”。
2. 在“程序包管理”页面，单击右上角“创建”可创建程序包。
3. 在“创建程序包”对话框，参见[表15-6](#)设置相关参数。



图 15-6 创建程序包

创建程序包
×

包类型  JAR  PyFile  File  ModelFile

\* OBS路径 多个参数请以Enter键分隔

分组设置  已有分组  创建新分组  不分组

\* 分组名称

标签 如果您需要使用同一标签识别多种云资源，即所有服务均可在标签输入框下拉选择同一标签，建议在TMS中创建预定义标签。 [查看预定义标签](#)

在下方键/值输入框输入内容后单击添加，即可将标签加入此处

您还可以添加10个标签。

表 15-6 参数说明

| 参数名称  | 描述                                                                                                                                                        |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| 包类型   | 支持的包类型如下： <ul style="list-style-type: none"> <li>• JAR：用户jar文件</li> <li>• PyFile：用户Python文件</li> <li>• File：用户文件</li> <li>• ModelFile：用户AI模型文件</li> </ul> |
| OBS路径 | 选择对应程序包的OBS路径。<br><b>说明</b> <ul style="list-style-type: none"> <li>• 程序包需提前上传至OBS服务中保存。</li> <li>• 只支持选择文件。</li> </ul>                                    |
| 分组设置  | 可选择“已有分组”，“创建新分组”或“不分组”。                                                                                                                                  |

| 参数名称 | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 分组名称 | <ul style="list-style-type: none"> <li>选择“已有分组”：可选择已有的分组。</li> <li>选择“创建新分组”：可输入自定义的组名称。</li> <li>选择“不分组”：不需要选择或输入组名称。</li> </ul> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>如果选择分组，则对应的权限管理为对应程序包组的权限管理。</li> <li>如果选择不分组，则对应的权限管理为对应程序包的权限管理。</li> </ul> <p>程序包组和程序包权限管理请参考<a href="#">程序包权限管理</a>。</p>                                                                                                                                                                                                                                                                                                                 |
| 标签   | <p>使用标签标识云资源。包括标签键和标签值。如果您需要使用同一标签标识多种云资源，即所有服务均可在标签输入框下拉选择同一标签，建议在标签管理服务（TMS）中创建预定义标签。</p> <p>如您的组织已经设定DLI的相关标签策略，则需按照标签策略规则为资源添加标签。标签不符合标签策略的规则，则可能会导致资源创建失败，请联系组织管理员了解标签策略详情。</p> <p>具体请参考《<a href="#">标签管理服务用户指南</a>》。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>最多支持20个标签。</li> <li>一个“键”只能添加一个“值”。</li> <li>每个资源中的键名不能重复。</li> <li>标签键：在输入框中输入标签键名称。</li> </ul> <p><b>说明</b></p> <p>标签的键的最大长度为128个字符，标签的键可以包含任意语种字母、数字、空格和_ . : + - @ , 但首尾不能含有空格，不能以_sys_开头。</p> <ul style="list-style-type: none"> <li>标签值：在输入框中输入标签值。</li> </ul> <p><b>说明</b></p> <p>标签值的最大长度为255个字符，标签的值可以包含任意语种字母、数字、空格和_ . : + - @ 。</p> |

4. 单击“确定”，完成创建程序包。

程序包创建成功后，您可以在“程序包管理”页面查看和选择使用对应的包。作业执行完成后，如果不再使用程序包，您可以程序包管理页面及时删除程序包，释放DLI存储空间。

### 15.3.3 配置 DLI 程序包权限

针对不同用户，可以通过权限设置分配不同的程序包组或程序包，不同用户之间的作业效率互不影响，保障作业性能。

- 管理员用户、程序包组拥有程序包组的所有权限。不需要进行权限设置，且其他用户无法修改其程序包组权限。
- 管理员用户、程序包的所有者拥有程序包的所有权限。不需要进行权限设置，且其他用户无法修改其程序包权限。
- 程序包组作为一个单元，用于管理行为一致的程序包，所以支持赋权给程序包组相关权限，但不支持对程序包组中的程序包单独赋权。
- 管理员用户给新用户设置程序包组或程序包权限时，管理员用户所在用户组的所属区域需具有Tenant Guest权限。

关于Tenant Guest权限的介绍和开通方法，详细参见《[权限策略](#)》和《[统一身份认证服务用户指南](#)》中的[创建用户组](#)。

## 配置程序包组或程序包权限

1. 在“程序包管理”页面，选择要授权的程序包组或程序包，单击“操作”列中的“权限管理”。
2. 进入“用户权限信息”页面，单击页面右上角“授权”新增授权用户，并选择对应的权限。

### 说明

- 如果创建程序包时选择了分组，则权限管理为对应程序包组的权限管理。
- 如果创建程序包时选择了不分组，则权限管理为对应程序包的权限管理。

### 程序包组授权

图 15-7 程序包组授权

授权

\* 用户名

请勾选需要授予的权限

全选

使用组                       更新组                       查询组

删除组                       赋权                       回收

查看其他用户具备的权限

表 15-7 程序包组授权参数说明

| 参数名称 | 描述                                                                                                                                                                                                                                                                                   |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 用户名  | 被授权的IAM用户的名称。<br><b>说明</b><br>该用户名称是已存在的IAM用户名称。                                                                                                                                                                                                                                     |
| 权限设置 | <ul style="list-style-type: none"> <li>• 使用组：使用该组的程序包。</li> <li>• 更新组：更新该组内程序包，包含创建组内程序包。</li> <li>• 查询组：查询组内程序包详情。</li> <li>• 删除组：删除该组的程序包。</li> <li>• 赋权：当前用户可将组的权限赋予其他用户。</li> <li>• 回收：当前用户可回收其他用户具备的该组的权限，但不能回收该组所有者的权限。</li> <li>• 查看其他用户具备的权限：当前用户可查看其他用户具备的该组的权限。</li> </ul> |

### 程序包授权

图 15-8 程序包授权

授权

\* 用户名

请勾选需要赋予的权限

全选

使用程序包       更新程序包       查询程序包

删除程序包       赋权       回收

查看其他用户具备的权限

表 15-8 程序包授权参数说明

| 参数名称 | 描述                                                                                                                                                                                                                                                                                |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 用户名  | 被授权的IAM用户的名称。<br><b>说明</b><br>该用户名称是已存在的IAM用户名称。                                                                                                                                                                                                                                  |
| 权限设置 | <ul style="list-style-type: none"> <li>● 使用程序包：使用该程序包。</li> <li>● 更新程序包：更新该程序包。</li> <li>● 查询程序包：查询该程序包。</li> <li>● 删除程序包：删除该程序包。</li> <li>● 赋权：当前用户可将程序包的权限赋予其他用户。</li> <li>● 回收：当前用户可回收其他用户具备的该程序包的权限，但不能回收该程序包所有者的权限。</li> <li>● 查看其他用户具备的权限：当前用户可查看其他用户具备的该程序包的权限。</li> </ul> |

## 修改程序包组或程序包权限

1. 在“程序包管理”页面，选择要程序包组或程序包，单击“操作”列中的“权限管理”。
2. 进入“用户权限信息”页面，单击对应子用户“操作”列中的“权限设置”可修改该用户的权限。

详细权限描述如表15-7和表15-8所示。

### 📖 说明

- 如果创建程序包时选择了分组，则修改的是对应程序包组的权限。
- 如果创建程序包时选择了不分组，则修改的是对应程序包的权限。

如果用户权限信息页面的“权限设置”选项为灰色时，表示您不具备修改此程序包组或程序包权限的权限。

您可以向管理员用户、组所有者等具有赋权权限的用户申请“程序包组或程序包的赋权”和“程序包组或程序包权限的回收”权限。

## 回收程序包组或程序包权限

DLI提供了一键回收程序包组或程序包权限的功能。

1. 在“程序包管理”页面，选择要程序包组或程序包，单击“操作”列中的“权限管理”。
2. 进入“用户权限信息”页面，单击对应子用户“操作”列中的“回收”将删除该用户的权限。

回收后该子用户将不具备该程序包组或程序包的任意权限。

### 说明

- 如果创建程序包时选择了分组，则回收的是对应程序包组的权限。
- 如果创建程序包时选择了不分组，则回收的是对应程序包的权限。

## 15.3.4 修改 DLI 程序包所有者

DLI提供了修改程序包组或程序包的所有者的功能。

1. 登录DLI管理控制台，选择“数据管理 > 程序包管理”。
2. 在“程序包管理”页面，单击程序包“操作”列中的“更多 > 修改所有者”。
  - 如果该程序包进行过分组设置，选择“组”或者“程序包”进行修改。

图 15-9 修改程序包所有者

### 修改所有者

分组名称

名称

选择类型  组  程序包

用户名

- 如果该程序包没有进行过分组设置，则可以参考下图，直接修改该程序包的所有者。

图 15-10 程序包管理-修改程序包所有者



表 15-9 参数说明

| 参数名称 | 描述                                                                                                                        |
|------|---------------------------------------------------------------------------------------------------------------------------|
| 分组名称 | <ul style="list-style-type: none"> <li>如果创建程序包时选择了分组，显示所在的分组名称。</li> <li>如果创建程序包时没有选择分组，则不显示该参数。</li> </ul>               |
| 名称   | 程序包名称。                                                                                                                    |
| 选择类型 | <ul style="list-style-type: none"> <li>如果创建程序包时选择了分组，可选择修改“组”的所有者或者“程序包”的所有者。</li> <li>如果创建程序包时没有选择分组，则不显示该参数。</li> </ul> |
| 用户名  | 程序包所有者的名称。<br><b>说明</b><br>该用户名称是已存在的IAM用户名称。                                                                             |

- 单击“确定”修改完成。

### 15.3.5 DLI 程序包标签管理

标签是用户自定义的、用于标识云资源的键值对，它可以帮助用户对云资源进行分类和搜索。标签由标签“键”和标签“值”组成。

DLI支持对程序包组或程序包添加标签。

- 在DLI管理控制台单击“数据管理 > 程序包管理”。
- 选择程序包，单击操作列的“更多 > 标签”，显示当前程序包组或程序包的标签信息。
- 单击“添加/编辑标签”，弹出“添加/编辑标签”对话框。
- 在“添加/编辑标签”对话框中配置标签参数。

表 15-10 标签配置参数

| 参数  | 参数说明                                                                                                                                                                                                                                                                                                                                                                          |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 标签键 | <p>您可以选择：</p> <ul style="list-style-type: none"> <li>在输入框的下拉列表中选择预定义标签键。<br/>如果添加预定义标签，用户需要预先在标签管理服务中创建好预定义标签，然后在“标签键”的下拉框中进行选择。用户可以通过单击“查看预定义标签”进入标签管理服务的“预定义标签”页面，然后单击“创建标签”来创建新的预定义标签。<br/>具体请参见《标签管理服务用户指南》中的“<a href="#">创建预定义标签</a>”章节。</li> <li>在输入框中输入标签键名称。</li> </ul> <p><b>说明</b><br/>标签的键的最大长度为128个字符，标签的键可以包含任意语种字母、数字、空格和_ . : + - @ ,但首尾不能含有空格，不能以_sys_开头。</p> |
| 标签值 | <p>您可以选择：</p> <ul style="list-style-type: none"> <li>在输入框的下拉列表中选择预定义标签值。</li> <li>在输入框中输入标签值。</li> </ul> <p><b>说明</b><br/>标签值的最大长度为255个字符，标签的值可以包含任意语种字母、数字、空格和_ . : + - @ 。</p>                                                                                                                                                                                              |

 说明

- 最多支持20个标签。
  - 一个“键”只能添加一个“值”。
  - 每个资源中的键名不能重复。
5. 单击“确定”。
  6. （可选）在标签列表中，单击“操作”列中“删除”可对选中的标签进行删除。

### 15.3.6 DLI 内置依赖包

DLI内置依赖包是平台默认提供的依赖包，用户打包Spark或Flink jar作业jar包时，不需要额外上传这些依赖包，以免与平台内置依赖包冲突。

#### Spark 3.1.1 依赖包

表 15-11 Spark 3.1.1 依赖包

| 依赖包名称                   |                                                   |                            |
|-------------------------|---------------------------------------------------|----------------------------|
| accessors-smart-1.2.jar | hive-shims-scheduler-3.1.0-h0.cbu.mrs.321.r10.jar | metrics-graphite-4.1.1.jar |
| activation-1.1.1.jar    | hive-spark-client-3.1.0-h0.cbu.mrs.321.r10.jar    | metrics-jmx-4.1.1.jar      |

| 依赖包名称                            |                                                        |                                       |
|----------------------------------|--------------------------------------------------------|---------------------------------------|
| aggdesigner-algorithm-6.0.jar    | hive-standalone-metastore-3.1.0-h0.cbu.mrs.321.r10.jar | metrics-json-4.1.1.jar                |
| aircompressor-0.16.jar           | hive-storage-api-2.7.2.jar                             | metrics-jvm-4.1.1.jar                 |
| algebra_2.12-2.0.0-M2.jar        | hive-vector-codegen-3.1.0-h0.cbu.mrs.321.r10.jar       | minlog-1.3.0.jar                      |
| annotations-17.0.0.jar           | hk2-api-2.6.1.jar                                      | netty-3.10.6.Final.jar                |
| ant-1.10.9.jar                   | hk2-locator-2.6.1.jar                                  | netty-all-4.1.86.Final.jar            |
| ant-launcher-1.10.9.jar          | hk2-utils-2.6.1.jar                                    | netty-buffer-4.1.86.Final.jar         |
| antlr4-runtime-4.8-1.jar         | hppc-0.7.2.jar                                         | netty-codec-4.1.86.Final.jar          |
| antlr-runtime-3.5.2.jar          | httpclient-4.5.6.jar                                   | netty-codec-dns-4.1.86.Final.jar      |
| aopalliance-1.0.jar              | httpcore-4.4.10.jar                                    | netty-codec-haproxy-4.1.86.Final.jar  |
| aopalliance-repackaged-2.6.1.jar | istack-commons-runtime-3.0.8.jar                       | netty-codec-http2-4.1.86.Final.jar    |
| apiguardian-api-1.1.0.jar        | ivy-2.5.0.jar                                          | netty-codec-http-4.1.86.Final.jar     |
| arpack_combined_all-0.1.jar      | jackson-annotations-2.13.2.jar                         | netty-codec-memcache-4.1.86.Final.jar |
| arrow-format-2.0.0.jar           | jackson-core-2.13.2.jar                                | netty-codec-mqtt-4.1.86.Final.jar     |
| arrow-memory-core-2.0.0.jar      | jackson-core-asl-1.9.13-atlassian-4.jar                | netty-codec-redis-4.1.86.Final.jar    |
| arrow-memory-netty-2.0.0.jar     | jackson-databind-2.13.2.jar                            | netty-codec-smtp-4.1.86.Final.jar     |
| arrow-vector-2.0.0.jar           | jackson-dataformat-yaml-2.13.2.jar                     | netty-codec-socks-4.1.86.Final.jar    |
| asm-5.0.4.jar                    | jackson-datatype-jsr310-2.11.2.jar                     | netty-codec-stomp-4.1.86.Final.jar    |
| audience-annotations-0.5.0.jar   | jackson-mapper-asl-1.9.13-atlassian-4.jar              | netty-codec-xml-4.1.86.Final.jar      |
| automaton-1.11-8.jar             | jackson-module-jaxb-annotations-2.13.2.jar             | netty-common-4.1.86.Final.jar         |



| 依赖包名称                        |                                      |                                                               |
|------------------------------|--------------------------------------|---------------------------------------------------------------|
| avatica-1.22.0.jar           | jackson-module-scala_2.12-2.13.2.jar | netty-handler-4.1.86.Final.jar                                |
| avatica-core-1.16.0.jar      | jaeger-client-1.6.0.jar              | netty-handler-proxy-4.1.86.Final.jar                          |
| avatica-metrics-1.16.0.jar   | jaeger-core-1.6.0.jar                | netty-handler-ssl-ocsp-4.1.86.Final.jar                       |
| avatica-server-1.16.0.jar    | jaeger-thrift-1.6.0.jar              | netty-resolver-4.1.86.Final.jar                               |
| avro-1.8.2.jar               | jaeger-tracerresolver-1.6.0.jar      | netty-resolver-dns-4.1.86.Final.jar                           |
| avro-ipc-1.8.2.jar           | jakarta.activation-api-1.2.1.jar     | netty-resolver-dns-classes-macos-4.1.86.Final.jar             |
| avro-mapred-1.8.2.jar        | jakarta.annotation-api-1.3.5.jar     | netty-resolver-dns-native-macos-4.1.86.Final-osx-aarch_64.jar |
| java-sdk-bundle-1.11.856.jar | jakarta.el-3.0.3.jar                 | netty-resolver-dns-native-macos-4.1.86.Final-osx-x86_64.jar   |
| base64-2.3.8.jar             | jakarta.el-api-3.0.3.jar             | netty-transport-4.1.86.Final.jar                              |
| bcpkix-jdk15on-1.69.jar      | jakarta.inject-2.6.1.jar             | netty-transport-classes-epoll-4.1.86.Final.jar                |
| bcprov-jdk15on-1.69.jar      | jakarta.servlet-api-4.0.3.jar        | netty-transport-classes-kqueue-4.1.86.Final.jar               |
| bcutil-jdk15on-1.69.jar      | jakarta.validation-api-2.0.2.jar     | netty-transport-native-epoll-4.1.86.Final-linux-aarch_64.jar  |
| bonecp-0.8.0.RELEASE.jar     | jakarta.ws.rs-api-2.1.6.jar          | netty-transport-native-epoll-4.1.86.Final-linux-x86_64.jar    |
| breeze_2.12-1.0.jar          | jakarta.xml.bind-api-2.3.2.jar       | netty-transport-native-kqueue-4.1.86.Final-osx-aarch_64.jar   |
| breeze-macros_2.12-1.0.jar   | jamon-runtime-2.4.1.jar              | netty-transport-native-kqueue-4.1.86.Final-osx-x86_64.jar     |

| 依赖包名称                             |                                |                                                     |
|-----------------------------------|--------------------------------|-----------------------------------------------------|
| caffeine-2.8.1.jar                | janino-3.0.16.jar              | netty-transport-native-unix-common-4.1.86.Final.jar |
| calcite-core-1.22.0.jar           | JavaEWAH-0.3.2.jar             | netty-transport-rxtx-4.1.86.Final.jar               |
| calcite-druid-1.19.0.jar          | java-sdk-core-3.0.12.jar       | netty-transport-sctp-4.1.86.Final.jar               |
| calcite-linq4j-1.22.0.jar         | javassist-3.25.0-GA.jar        | netty-transport-udt-4.1.86.Final.jar                |
| cats-kernel_2.12-2.0.0-M4.jar     | javax.activation-api-1.2.0.jar | nimbus-jose-jwt-8.19.jar                            |
| checker-qual-3.5.0.jar            | javax.annotation-api-1.3.2.jar | objenesis-2.5.1.jar                                 |
| chill_2.12-0.9.5.jar              | javax.inject-1.jar             | okhttp-3.14.9.jar                                   |
| chill-java-0.9.5.jar              | javax.jdo-3.2.0-m3.jar         | okio-1.17.2.jar                                     |
| classmate-1.5.1.jar               | java-xmlbuilder-1.1.jar        | opencsv-2.3.jar                                     |
| commons-beanutils-1.9.4.jar       | javax.servlet-api-3.1.0.jar    | opentelemetry-api-1.16.0.jar                        |
| commons-cli-1.2.jar               | javax.transaction-api-1.3.jar  | opentelemetry-context-1.16.0.jar                    |
| commons-codec-1.15.jar            | javax.ws.rs-api-2.1.1.jar      | opentelemetry-semconv-1.16.0-alpha.jar              |
| commons-collections-3.2.2.jar     | javalution-5.5.1.jar           | opentracing-api-0.33.0.jar                          |
| commons-compiler-3.0.16.jar       | jaxb-api-2.2.11.jar            | opentracing-noop-0.33.0.jar                         |
| commons-compress-1.21.jar         | jaxb-runtime-2.3.2.jar         | opentracing-tracerresolver-0.1.8.jar                |
| commons-configuration2-2.1.1.jar  | jboss-logging-3.4.1.Final.jar  | opentracing-util-0.33.0.jar                         |
| commons-crypto-1.0.0-20191105.jar | jboss-threads-2.3.3.Final.jar  | orc-core-1.6.8.jar                                  |
| commons-daemon-1.0.13.jar         | jcip-annotations-1.0-1.jar     | orc-mapreduce-1.6.8.jar                             |
| commons-dbcp-1.4.jar              | jcl-over-slf4j-1.7.36.jar      | orc-shims-1.6.8.jar                                 |

| 依赖包名称                        |                                        |                                         |
|------------------------------|----------------------------------------|-----------------------------------------|
| commons-dbc2-2.6.0.jar       | jcodings-1.0.57.jar                    | orc-tools-1.6.7-h0.cbu.mrs.321.r10.jar  |
| commons-digester-2.1.jar     | jdo-api-3.2.jar                        | oro-2.0.8.jar                           |
| commons-httpclient-3.1.jar   | jersey-client-2.34.jar                 | osgi-resource-locator-1.0.3.jar         |
| commons-io-2.8.0.jar         | jersey-common-2.34.jar                 | paranamer-2.8.jar                       |
| commons-lang-2.4.jar         | jersey-container-servlet-2.34.jar      | parquet-column-1.12.2.jar               |
| commons-lang-2.6.jar         | jersey-container-servlet-core-2.34.jar | parquet-common-1.12.2.jar               |
| commons-lang3-3.10.jar       | jersey-hk2-2.34.jar                    | parquet-encoding-1.12.2.jar             |
| commons-logging-1.2.jar      | jersey-server-2.34.jar                 | parquet-format-structures-1.12.2.jar    |
| commons-math3-3.4.1.jar      | jets3t-0.9.4-1.0.0.jar                 | parquet-hadoop-1.12.2.jar               |
| commons-net-3.1.jar          | jettison-1.1.jar                       | parquet-hadoop-bundle-1.12.0-ei-2.0.jar |
| commons-pool2-2.6.1.jar      | jetty-http-9.4.41.v20210516.jar        | parquet-jackson-1.12.2.jar              |
| commons-text-1.10.0.jar      | jetty-io-9.4.41.v20210516.jar          | postgresql-42.3.5.jar                   |
| commons-validator-1.7.jar    | jetty-rewrite-9.4.43.v20210629.jar     | protobuf-java-2.5.0.jar                 |
| compress-lzf-1.0.3.jar       | jetty-security-9.4.41.v20210516.jar    | py4j-0.10.9.jar                         |
| core-1.1.2.jar               | jetty-server-9.4.41.v20210516.jar      | pyrolite-4.30.jar                       |
| curator-client-2.13.0.jar    | jetty-servlet-9.4.41.v20210516.jar     | re2j-1.1.jar                            |
| curator-framework-2.13.0.jar | jetty-util-9.4.41.v20210516.jar        | RoaringBitmap-0.9.0.jar                 |
| curator-recipes-2.13.0.jar   | jetty-util-ajax-9.4.41.v20210516.jar   | scala-collection-compat_2.12-2.1.1.jar  |

| 依赖包名称                                                    |                                   |                                                      |
|----------------------------------------------------------|-----------------------------------|------------------------------------------------------|
| datanucleus-api-jdo-4.2.4.jar                            | jetty-webapp-9.4.41.v20210516.jar | scala-compiler-2.12.16.jar                           |
| datanucleus-core-4.1.17.jar                              | jetty-xml-9.4.41.v20210516.jar    | scala-library-2.12.16.jar                            |
| datanucleus-rdbms-fi-4.1.19-302022.jar                   | JLargeArrays-1.5.jar              | scala-parser-combinators_2.12-1.1.2.jar              |
| derby-10.14.2.0.jar                                      | jline-3.21.0.jar                  | scala-reflect-2.12.16.jar                            |
| disruptor-3.4.2.jar                                      | joda-time-2.10.5.jar              | scala-xml_2.12-1.2.0.jar                             |
| dli-catalog-client-2.3.7-20240108.090504-101.jar         | jodd-core-3.5.2.jar               | secComponentApi-1.1.8.jar                            |
| dli-catalog-hive3-client-2.3.7-20240108.090513-100.jar   | jodd-util-6.0.0.jar               | serializer-2.7.2.jar                                 |
| dli-catalog-hive-extension-2.3.7-20240108.090517-100.jar | joni-2.1.43.jar                   | shapeless_2.12-2.3.3.jar                             |
| dnsjava-2.1.7.jar                                        | jpam-1.1.jar                      | shims-0.9.0.jar                                      |
| dropwizard-metrics-hadoop-metrics2-reporter-0.1.2.jar    | jsch-0.1.72.jar                   | sketches-core-0.9.0.jar                              |
| error_prone_annotations-2.18.0.jar                       | json-20210307.jar                 | slf4j-api-1.7.30.jar                                 |
| esdk-obs-java-optimised-3.22.10.2.jar                    | json4s-ast_2.12-3.7.0-M5.jar      | slf4j-log4j12-1.7.25.jar                             |
| esri-geometry-api-2.2.0.jar                              | json4s-core_2.12-3.7.0-M5.jar     | snakeyaml-1.30.jar                                   |
| fastutil-6.5.6.jar                                       | json4s-jackson_2.12-3.7.0-M5.jar  | snappy-java-1.1.8.2.jar                              |
| flatbuffers-java-1.9.0.jar                               | json4s-scalap_2.12-3.7.0-M5.jar   | spark-avro_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar     |
| generex-1.0.2.jar                                        | json-path-2.4.0.jar               | spark-catalyst_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar |
| glassfish-corba-omgapi-4.2.2.jar                         | json-smart-2.3.jar                | spark-core_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar     |
| gson-2.8.9.jar                                           | jsr305-3.0.0.jar                  | spark-graphx_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar   |

| 依赖包名称                                           |                          |                                                             |
|-------------------------------------------------|--------------------------|-------------------------------------------------------------|
| gson-fire-1.8.5.jar                             | JTransforms-3.1.jar      | spark-hive_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar            |
| guava-14.0.1.jar                                | jul-to-slf4j-1.7.36.jar  | spark-kubernetes_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar      |
| guice-3.0.jar                                   | kafka-clients-2.8.0.jar  | spark-kvstore_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar         |
| guice-assistedinject-3.0.jar                    | kerb-admin-2.0.2.jar     | spark-launcher_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar        |
| guice-servlet-4.0.jar                           | kerb-client-2.0.2.jar    | spark-mllib_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar           |
| hadoop-annotations-3.1.1-h0.cbu.mrs.313.r9.jar  | kerb-common-2.0.2.jar    | spark-mllib-local_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar     |
| hadoop-archives-3.3.1-h0.cbu.mrs.321.r10.jar    | kerb-core-2.0.2.jar      | spark-network-common_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar  |
| hadoop-auth-3.3.1-h0.cbu.mrs.321.r16.jar        | kerb-crypto-2.0.2.jar    | spark-network-shuffle_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar |
| hadoop-3.3.1-h0.cbu.mrs.321.r16.jar             | kerb-identity-2.0.2.jar  | spark-quota-manager_2.12-3.1.1-2.3.7.dli-SNAPSHOT.jar       |
| hadoop-client-3.1.1-h0.cbu.mrs.313.r9.jar       | kerb-server-2.0.2.jar    | spark-repl_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar            |
| hadoop-common-3.3.1-h0.cbu.mrs.321.r10.jar      | kerb-simplekdc-2.0.2.jar | spark-sketch_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar          |
| hadoop-distcp-3.3.1-h0.cbu.mrs.321.r10.jar      | kerb-util-2.0.2.jar      | spark-sql_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar             |
| hadoop-hdfs-3.3.1-h0.cbu.mrs.321.r16.jar        | kerby-asn1-2.0.2.jar     | spark-streaming_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar       |
| hadoop-hdfs-client-3.3.1-h0.cbu.mrs.321.r10.jar | kerby-config-2.0.2.jar   | spark-tags_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar            |
| hadoop-3.1.1-52.1.jar                           | kerby-pkix-2.0.2.jar     | spark-unsafe_2.12-3.1.1-h1.cbu.dli.20230607.r1.jar          |

| 依赖包名称                                                           |                                                           |                                                |
|-----------------------------------------------------------------|-----------------------------------------------------------|------------------------------------------------|
| hadoop-mapreduce-client-common-3.1.1-h0.cbu.mrs.313.r9.jar      | kerby-util-2.0.2.jar                                      | spark-uquery_2.12-3.1.1-2.3.7.dli-SNAPSHOT.jar |
| hadoop-mapreduce-client-core-3.1.1-h0.cbu.mrs.313.r9.jar        | kerby-xdr-2.0.2.jar                                       | spire_2.12-0.17.0-M1.jar                       |
| hadoop-mapreduce-client-jobclient-3.1.1-h0.cbu.mrs.313.r9.jar   | kotlin-stdlib-1.4.21.jar                                  | spire-macros_2.12-0.17.0-M1.jar                |
| hadoop-mapreduce-client-nativetask-3.3.1-h0.cbu.mrs.321.r10.jar | kotlin-stdlib-common-1.4.21.jar                           | spire-platform_2.12-0.17.0-M1.jar              |
| hadoop-registry-3.3.1-h0.cbu.mrs.321.r10.jar                    | kryo-shaded-4.0.2.jar                                     | spire-util_2.12-0.17.0-M1.jar                  |
| hadoop-shaded-guava-1.1.1.jar                                   | kubernetes-client-5.4.1-20211025.jar                      | sqlline-1.3.0.jar                              |
| hadoop-shaded-protobuf_3_7-1.1.1.jar                            | kubernetes-model-admissionregistration-5.4.1-20211025.jar | ST4-4.0.4.jar                                  |
| hadoop-yarn-api-3.1.1-h0.cbu.mrs.313.r9.jar                     | kubernetes-model-apiextensions-5.4.1-20211025.jar         | stax2-api-4.2.1.jar                            |
| hadoop-yarn-client-3.1.1-h0.cbu.mrs.313.r9.jar                  | kubernetes-model-apps-5.4.1-20211025.jar                  | stax-api-1.0.1.jar                             |
| hadoop-yarn-registry-3.3.1-h0.cbu.mrs.321.r10.jar               | kubernetes-model-autoscaling-5.4.1-20211025.jar           | stream-2.9.6.jar                               |
| hbase-asyncfs-2.4.14-h0.cbu.mrs.321.r10.jar                     | kubernetes-model-batch-5.4.1-20211025.jar                 | streamingClient                                |
| hbase-client-2.4.14-h0.cbu.mrs.321.r10.jar                      | kubernetes-model-certificates-5.4.1-20211025.jar          | streamingClient010                             |
| hbase-common-2.4.14-h0.cbu.mrs.321.r10.jar                      | kubernetes-model-common-5.4.1-20211025.jar                | swagger-annotations-2.2.8.jar                  |
| hbase-hadoop2-compat-2.4.14-h0.cbu.mrs.321.r10.jar              | kubernetes-model-coordination-5.4.1-20211025.jar          | tephra-api-0.6.0.jar                           |

| 依赖包名称                                               |                                                  |                                           |
|-----------------------------------------------------|--------------------------------------------------|-------------------------------------------|
| hbase-hadoop-compat-2.4.14-h0.cbu.mrs.321.r10.jar   | kubernetes-model-core-5.4.1-20211025.jar         | tephra-core-0.6.0.jar                     |
| hbase-http-2.4.14-h0.cbu.mrs.321.r10.jar            | kubernetes-model-discovery-5.4.1-20211025.jar    | tephra-hbase-compat-1.0-0.6.0.jar         |
| hbase-logging-2.4.14-h0.cbu.mrs.321.r10.jar         | kubernetes-model-events-5.4.1-20211025.jar       | threetenbp-1.3.5.jar                      |
| hbase-metrics-2.4.14-h0.cbu.mrs.321.r10.jar         | kubernetes-model-extensions-5.4.1-20211025.jar   | threeten-extra-1.5.0.jar                  |
| hbase-metrics-api-2.4.14-h0.cbu.mrs.321.r10.jar     | kubernetes-model-flowcontrol-5.4.1-20211025.jar  | tink-1.6.0.jar                            |
| hbase-procedure-2.4.14-h0.cbu.mrs.321.r10.jar       | kubernetes-model-metrics-5.4.1-20211025.jar      | token-provider-2.0.2.jar                  |
| hbase-protocol-2.4.14-h0.cbu.mrs.321.r10.jar        | kubernetes-model-networking-5.4.1-20211025.jar   | tomcat-servlet-api-8.5.61.jar             |
| hbase-protocol-shaded-2.4.14-h0.cbu.mrs.321.r10.jar | kubernetes-model-node-5.4.1-20211025.jar         | transaction-api-1.1.jar                   |
| hbase-replication-2.4.14-h0.cbu.mrs.321.r10.jar     | kubernetes-model-policy-5.4.1-20211025.jar       | twill-api-0.6.0-incubating.jar            |
| hbase-server-2.4.14-h0.cbu.mrs.321.r10.jar          | kubernetes-model-rbac-5.4.1-20211025.jar         | twill-common-0.6.0-incubating.jar         |
| hbase-shaded-gson-4.1.4.jar                         | kubernetes-model-scheduling-5.4.1-20211025.jar   | twill-core-0.6.0-incubating.jar           |
| hbase-shaded-jersey-4.1.4.jar                       | kubernetes-model-storageclass-5.4.1-20211025.jar | twill-discovery-api-0.6.0-incubating.jar  |
| hbase-shaded-jetty-4.1.4.jar                        | leveldbjni-all-1.8-20191105.jar                  | twill-discovery-core-0.6.0-incubating.jar |
| hbase-shaded-miscellaneous-4.1.4.jar                | libfb303-0.9.3.jar                               | twill-zookeeper-0.6.0-incubating.jar      |
| hbase-shaded-netty-4.1.4.jar                        | libthrift-0.14.1-ei-311001.jar                   | univocity-parsers-2.9.1.jar               |

| 依赖包名称                                              |                                                                          |                                       |
|----------------------------------------------------|--------------------------------------------------------------------------|---------------------------------------|
| hbase-shaded-protobuf-4.1.4.jar                    | log4j-1.2.17-cloudera1.jar                                               | us-common-1.0.66.jar                  |
| hbase-unsafe-4.1.4.jar                             | log4j-api-2.17.1.jar                                                     | velocity-1.7.jar                      |
| hbase-zookeeper-2.4.14-h0.cbu.mrs.321.r10.jar      | log4j-rolling-appender-20131024-2017.jar                                 | velocity-engine-core-2.3.jar          |
| hibernate-validator-6.2.5.Final.jar                | logging-interceptor-3.14.9.jar                                           | wildfly-client-config-1.0.1.Final.jar |
| HikariCP-2.6.1.jar                                 | luxor-cluster-quota-manager-transport_2.12-2.3.7-20231226.034700-559.jar | wildfly-common-1.5.2.Final.jar        |
| hive-classification-3.1.0-h0.cbu.mrs.321.r10.jar   | luxor-encrypt-2.3.7-20231226.034423-1046.jar                             | woodstox-core-5.4.0.jar               |
| hive-common-3.1.0-h0.cbu.mrs.321.r10.jar           | luxor-fs3-2.3.7-20231226.034438-1039.jar                                 | xalan-2.7.2.jar                       |
| hive-exec-3.1.0-h0.cbu.mrs.321.r10-core.jar        | luxor-obs-fs3-2.3.7-20231226.034443-1038.jar                             | xbean-asm7-shaded-4.15.jar            |
| hive-llap-client-2.3.3-ei-12-20210120.005053-2.jar | luxor-rpc_2.12-2.3.7-20231226.034653-560.jar                             | xercesImpl-2.12.2.jar                 |
| hive-llap-common-3.1.0-h0.cbu.mrs.321.r10.jar      | luxor-scc-adapter-2.3.7-20231226.034418-1045.jar                         | xml-apis-1.4.01.jar                   |
| hive-llap-tez-3.1.0-h0.cbu.mrs.321.r10.jar         | luxor-transport-2.3.7-20231226.034433-1038.jar                           | xnio-api-3.8.4.Final.jar              |
| hive-metastore-3.1.0-h0.cbu.mrs.321.r10.jar        | lz4-java-1.7.1.jar                                                       | xz-1.5.jar                            |
| hive-serde-3.1.0-h0.cbu.mrs.321.r10.jar            | machinist_2.12-0.6.8.jar                                                 | zjsonpatch-0.3.0.jar                  |
| hive-service-rpc-3.1.0-h0.cbu.mrs.321.r10.jar      | macro-compatible_2.12-1.1.1.jar                                          | zookeeper-3.5.6-ei-302002.jar         |
| hive-shims-0.23-3.1.0-h0.cbu.mrs.321.r10.jar       | memarts-ccsdk-1.0.jar                                                    | zookeeper-jute-3.5.6-ei-302002.jar    |
| hive-shims-3.1.0-h0.cbu.mrs.321.r10.jar            | memory-0.9.0.jar                                                         | zstd-jni-1.4.9-1.jar                  |



| 依赖包名称                                          |                        |   |
|------------------------------------------------|------------------------|---|
| hive-shims-common-3.1.0-h0.cbu.mrs.321.r10.jar | metrics-core-4.1.1.jar | - |

## Spark 2.4.5 依赖包

表 15-12 Spark 2.4.5 依赖包

| 依赖包名称                                   |                                           |                                              |
|-----------------------------------------|-------------------------------------------|----------------------------------------------|
| JavaEWAH-1.1.7.jar                      | httpclient-4.5.6.jar                      | lucene-queryparser-7.7.2.jar                 |
| RoaringBitmap-0.7.45.jar                | httpcore-4.4.10.jar                       | lucene-sandbox-7.7.2.jar                     |
| ST4-4.3.1.jar                           | ivy-2.4.0.jar                             | luxor-encrypt-2.0.0-20220623.010726-213.jar  |
| accessors-smart-1.2.jar                 | jackson-annotations-2.11.4.jar            | luxor-fs3-2.0.0-20220623.010750-209.jar      |
| activation-1.1.1.jar                    | jackson-core-2.11.4.jar                   | luxor-obs-fs3-2.0.0-20220623.010756-209.jar  |
| aircompressor-0.16.jar                  | jackson-core-asl-1.9.13-atlassian-4.jar   | luxor-rpc_2.11-2.0.0-20220623.010737-182.jar |
| alluxio-2.3.1-luxor-SNAPSHOT-client.jar | jackson-databind-2.11.4.jar               | luxor-transport-2.0.0-20220623.010744-71.jar |
| annotations-17.0.0.jar                  | jackson-dataformat-yaml-2.11.4.jar        | lz4-java-1.7.1.jar                           |
| antlr-2.7.7.jar                         | jackson-datatype-jsr310-2.11.2.jar        | machinist_2.11-0.6.1.jar                     |
| antlr-runtime-3.4.jar                   | jackson-jaxrs-base-2.10.3.jar             | macro-compat_2.11-1.1.1.jar                  |
| antlr4-runtime-4.8-1.jar                | jackson-jaxrs-json-provider-2.10.3.jar    | metrics-core-3.1.5.jar                       |
| aopalliance-1.0.jar                     | jackson-mapper-asl-1.9.13-atlassian-4.jar | metrics-graphite-3.1.5.jar                   |

| 依赖包名称                                |                                            |                                 |
|--------------------------------------|--------------------------------------------|---------------------------------|
| aopalliance-repackaged-2.4.0-b34.jar | jackson-module-jaxb-annotations-2.10.3.jar | metrics-jmx-4.1.12.1.jar        |
| apache-log4j-extras-1.2.17.jar       | jackson-module-paranamer-2.11.4.jar        | metrics-json-3.1.5.jar          |
| arpack_combined_all-0.1.jar          | jackson-module-scala_2.11-2.11.4.jar       | metrics-jvm-3.1.5.jar           |
| arrow-format-0.12.0.jar              | jakarta.activation-api-1.2.1.jar           | minlog-1.3.0.jar                |
| arrow-memory-0.12.0.jar              | jakarta.xml.bind-api-2.3.2.jar             | mssql-jdbc-6.2.1.jre7.jar       |
| arrow-vector-0.12.0.jar              | janino-3.0.9.jar                           | netty-all-4.1.51.Final.jar      |
| asm-5.0.4.jar                        | java-util-1.9.0.jar                        | nimbus-jose-jwt-8.19.jar        |
| audience-annotations-0.5.0.jar       | java-xmlbuilder-1.1.jar                    | objenesis-2.5.1.jar             |
| automaton-1.11-8.jar                 | javassist-3.18.1-GA.jar                    | okhttp-3.14.9.jar               |
| avro-1.8.2.jar                       | javax.annotation-api-1.2.jar               | okio-1.17.2.jar                 |
| avro-ipc-1.8.2.jar                   | javax.inject-1.jar                         | opencsv-2.3.jar                 |
| avro-mapred-1.8.2.jar                | javax.inject-2.4.0-b34.jar                 | opencsv-4.6.jar                 |
| java-sdk-bundle-1.11.856.jar         | javax.servlet-api-3.1.0.jar                | opencv-4.3.0-2.jar              |
| base64-2.3.8.jar                     | javax.ws.rs-api-2.0.1.jar                  | orc-core-1.6.8-nohive.jar       |
| bcpkix-jdk15on-1.66.jar              | javalution-5.3.1.jar                       | orc-mapreduce-1.6.8-nohive.jar  |
| bcprov-jdk15on-1.67.jar              | jaxb-api-2.2.11.jar                        | orc-shims-1.6.8.jar             |
| bonecp-0.8.0.RELEASE.jar             | jcip-annotations-1.0-1.jar                 | oro-2.0.8.jar                   |
| breeze-macros_2.11-0.13.2.jar        | jcl-over-slf4j-1.7.30.jar                  | osgi-resource-locator-1.0.1.jar |
| breeze_2.11-0.13.2.jar               | jdo-api-3.0.1.jar                          | paranamer-2.8.jar               |
| calcite-avatica-1.2.0-incubating.jar | jersey-client-2.23.1.jar                   | parquet-column-1.12.2.jar       |
| chill-java-0.9.3.jar                 | jersey-common-2.23.1.jar                   | parquet-common-1.12.2.jar       |

| 依赖包名称                             |                                          |                                         |
|-----------------------------------|------------------------------------------|-----------------------------------------|
| chill_2.11-0.9.3.jar              | jersey-container-servlet-2.23.1.jar      | parquet-encoding-1.12.2.jar             |
| commons-beanutils-1.9.4.jar       | jersey-container-servlet-core-2.23.1.jar | parquet-format-structures-1.12.2.jar    |
| commons-cli-1.2.jar               | jersey-guava-2.23.1.jar                  | parquet-hadoop-1.12.2.jar               |
| commons-codec-1.15.jar            | jersey-media-jaxb-2.23.1.jar             | parquet-hadoop-bundle-1.6.0.jar         |
| commons-collections-3.2.2.jar     | jersey-server-2.23.1.jar                 | parquet-jackson-1.12.2.jar              |
| commons-collections4-4.2.jar      | jets3t-0.9.4.jar                         | postgresql-42.2.14.jar                  |
| commons-compiler-3.0.9.jar        | jettison-1.1.jar                         | protobuf-java-2.5.0.jar                 |
| commons-compress-1.4.1.jar        | jetty-http-9.4.34.v20201102.jar          | py4j-0.10.7.jar                         |
| commons-configuration2-2.1.1.jar  | jetty-io-9.4.34.v20201102.jar            | pyrolite-4.13.jar                       |
| commons-crypto-1.0.0-20191105.jar | jetty-security-9.4.34.v20201102.jar      | re2j-1.1.jar                            |
| commons-daemon-1.0.13.jar         | jetty-server-9.4.34.v20201102.jar        | scala-compiler-2.11.12.jar              |
| commons-dbcp2-2.7.0.jar           | jetty-servlet-9.4.34.v20201102.jar       | scala-library-2.11.12.jar               |
| commons-httpclient-3.1.jar        | jetty-util-9.4.34.v20201102.jar          | scala-parser-combinators_2.11-1.1.2.jar |
| commons-io-2.5.jar                | jetty-util-ajax-9.4.34.v20201102.jar     | scala-reflect-2.11.12.jar               |
| commons-lang-2.6.jar              | jetty-webapp-9.4.34.v20201102.jar        | scala-xml_2.11-1.0.5.jar                |
| commons-lang3-3.5.jar             | jetty-xml-9.4.34.v20201102.jar           | secComponentApi-1.0.6.jar               |
| commons-logging-1.2.jar           | joda-time-2.9.3.jar                      | shapeless_2.11-2.3.2.jar                |
| commons-math3-3.4.1.jar           | jodd-core-3.5.2.jar                      | shims-0.7.45.jar                        |

| 依赖包名称                         |                               |                                                                   |
|-------------------------------|-------------------------------|-------------------------------------------------------------------|
| commons-net-3.1.jar           | json-20200518.jar             | slf4j-api-1.7.30.jar                                              |
| commons-pool2-2.8.0.jar       | json-io-2.5.1.jar             | slf4j-log4j12-1.7.30.jar                                          |
| commons-text-1.3.jar          | json-sanitizer-1.2.1.jar      | snakeyaml-1.26.jar                                                |
| compress-lzf-1.0.3.jar        | json-smart-2.3.jar            | snappy-java-1.1.8.2.jar                                           |
| core-1.1.2.jar                | json4s-ast_2.11-3.5.3.jar     | solr-core-7.7.2.jar                                               |
| crypter-0.0.6.jar             | json4s-core_2.11-3.5.3.jar    | solr-solrj-7.7.2.jar                                              |
| curator-client-4.2.0.jar      | json4s-jackson_2.11-3.5.3.jar | spark-avro_2.11-2.4.5.0100-2.0.0.dli-20220617.085536-9.jar        |
| curator-framework-4.2.0.jar   | json4s-scalap_2.11-3.5.3.jar  | spark-avro_2.11-4.0.0.jar                                         |
| curator-recipes-2.7.1.jar     | jsp-api-2.1.jar               | spark-catalyst_2.11-2.4.5.0100-2.0.0.dli-20220617.085405-16.jar   |
| datanucleus-api-jdo-3.2.6.jar | jsr305-1.3.9.jar              | spark-core_2.11-2.4.5.0100-2.0.0.dli-20220617.085327-16.jar       |
| datanucleus-core-3.2.10.jar   | jta-1.1.jar                   | spark-graphx_2.11-2.4.5.0100-.0.dli-20220617.085336-16.jar        |
| datanucleus-rdbms-3.2.9.jar   | jtransforms-2.4.0.jar         | spark-hive_2.11-2.4.5.0100-2.0.0.dli-20220617.085423-16.jar       |
| derby-10.14.2.0.jar           | jts-core-1.16.1.jar           | spark-kubernetes_2.11-2.4.5.0100-2.0.0.dli-20220617.085519-16.jar |
| dnsjava-2.1.7.jar             | jul-to-slf4j-1.7.30.jar       | spark-kvstore_2.11-2.4.5.0100-2.0.0.dli-20220617.085249-16.jar    |
| ecj-3.21.0.jar                | junit-4.11.jar                | spark-launcher_2.11-2.4.5.0100-2.0.0.dli-20220617.085435-16.jar   |

| 依赖包名称                                    |                          |                                                                        |
|------------------------------------------|--------------------------|------------------------------------------------------------------------|
| ehcache-3.3.1.jar                        | kerb-admin-1.0.1.jar     | spark-mllib-local_2.11-2.4.5.0100-2.0.0.dli-20220617.085349-16.jar     |
| expiringmap-0.5.9.jar                    | kerb-client-1.0.1.jar    | spark-mllib_2.11-2.4.5.0100-2.0.0.dli-20220617.085342-16.jar           |
| fastutil-8.2.3.jar                       | kerb-common-1.0.1.jar    | spark-network-common_2.11-2.4.5.0100-2.0.0.dli-20220617.085254-16.jar  |
| flatbuffers-java-1.9.0.jar               | kerb-core-1.0.1.jar      | spark-network-shuffle_2.11-2.4.5.0100-2.0.0.dli-20220617.085300-16.jar |
| fst-2.50.jar                             | kerb-crypto-1.0.1.jar    | spark-om_2.11-2.4.5.0100-2.0.0.dli-20220617.085316-16.jar              |
| generex-1.0.2.jar                        | kerb-identity-1.0.1.jar  | spark-repl_2.11-2.4.5.0100-2.0.0.dli-20220617.085430-16.jar            |
| geronimo-jcache_1.0_spec-1.0-alpha-1.jar | kerb-server-1.0.1.jar    | spark-sketch_2.11-2.4.5.0100-2.0.0.dli-20220617.085243-16.jar          |
| gson-2.2.4.jar                           | kerb-simplekdc-1.0.1.jar | spark-sql_2.11-2.4.5.0100-2.0.0.dli-20220617.085414-16.jar             |
| guava-14.0.1.jar                         | kerb-util-1.0.1.jar      | spark-streaming_2.11-2.4.5.0100-2.0.0.dli-20220617.085359-16.jar       |
| guice-4.0.jar                            | kerby-asn1-1.0.1.jar     | spark-tags_2.11-2.4.5.0100-2.0.0.dli-20220617.085322-16.jar            |

| 依赖包名称                                                 |                                                           |                                                               |
|-------------------------------------------------------|-----------------------------------------------------------|---------------------------------------------------------------|
| guice-servlet-4.0.jar                                 | kerby-config-1.0.1.jar                                    | spark-unsafe_2.11-2.4.5.0100-2.0.0.dli-20220617.085311-16.jar |
| hadoop-annotations-3.1.1-ei-302002.jar                | kerby-pkix-1.0.1.jar                                      | spark-uquery_2.11-2.4.5.0100-2.0.0.dli-SNAPSHOT.jar           |
| hadoop-auth-3.1.1-ei-302002.jar                       | kerby-util-1.0.1.jar                                      | spark-yarn_2.11-2.4.5.0100-2.0.0.dli-20220617.085531-16.jar   |
| hadoop-3.1.1-ei-302002.jar                            | kerby-xdr-1.0.1.jar                                       | spire-macros_2.11-0.13.0.jar                                  |
| hadoop-client-3.1.1-ei-302002.jar                     | kryo-shaded-4.0.2.jar                                     | spire_2.11-0.13.0.jar                                         |
| hadoop-common-3.1.1-ei-302002.jar                     | kubernetes-client-5.4.1-20211025.jar                      | stax-api-1.0-2.jar                                            |
| hadoop-hdfs-3.1.1-ei-302002.jar                       | kubernetes-model-admissionregistration-5.4.1-20211025.jar | stax2-api-3.1.4.jar                                           |
| hadoop-hdfs-client-3.1.1-ei-302002.jar                | kubernetes-model-apiextensions-5.4.1-20211025.jar         | stream-2.7.0.jar                                              |
| hadoop-3.1.1-46.jar                                   | kubernetes-model-apps-5.4.1-20211025.jar                  | stringtemplate-3.2.1.jar                                      |
| hadoop-mapreduce-client-common-3.1.1-ei-302002.jar    | kubernetes-model-autoscaling-5.4.1-20211025.jar           | threeten-extra-1.5.0.jar                                      |
| hadoop-mapreduce-client-core-3.1.1-ei-302002.jar      | kubernetes-model-batch-5.4.1-20211025.jar                 | tink-1.6.0.jar                                                |
| hadoop-mapreduce-client-jobclient-3.1.1-ei-302002.jar | kubernetes-model-certificates-5.4.1-20211025.jar          | token-provider-1.0.1.jar                                      |
| hadoop-minikdc-3.1.1-ei-302002.jar                    | kubernetes-model-common-5.4.1-20211025.jar                | tomcat-api-9.0.39.jar                                         |
| hadoop-yarn-api-3.1.1-ei-302002.jar                   | kubernetes-model-coordination-5.4.1-20211025.jar          | zookeeper-jute-3.5.6-ei-302002.jar                            |

| 依赖包名称                                                            |                                                  |                                |
|------------------------------------------------------------------|--------------------------------------------------|--------------------------------|
| hadoop-yarn-client-3.1.1-ei-302002.jar                           | kubernetes-model-core-5.4.1-20211025.jar         | tomcat-el-api-9.0.39.jar       |
| hadoop-yarn-common-3.1.1-ei-302002.jar                           | kubernetes-model-discovery-5.4.1-20211025.jar    | tomcat-jasper-9.0.39.jar       |
| hadoop-yarn-registry-3.1.1-ei-302002.jar                         | kubernetes-model-events-5.4.1-20211025.jar       | tomcat-jasper-el-9.0.39.jar    |
| hadoop-yarn-server-applicationhistoryservice-3.1.1-ei-302002.jar | kubernetes-model-extensions-5.4.1-20211025.jar   | tomcat-jsp-api-9.0.39.jar      |
| hadoop-yarn-server-common-3.1.1-ei-302002.jar                    | kubernetes-model-flowcontrol-5.4.1-20211025.jar  | tomcat-juli-9.0.39.jar         |
| hadoop-yarn-server-resourcemanager-3.1.1-ei-302002.jar           | kubernetes-model-metrics-5.4.1-20211025.jar      | tomcat-servlet-api-9.0.39.jar  |
| hadoop-yarn-server-web-proxy-3.1.1-ei-302002.jar                 | kubernetes-model-networking-5.4.1-20211025.jar   | tomcat-util-9.0.39.jar         |
| hamcrest-core-1.3.jar                                            | kubernetes-model-node-5.4.1-20211025.jar         | tomcat-util-scan-9.0.39.jar    |
| hive-common-1.2.1-2.0.0.dli-20220528.090500-402.jar              | kubernetes-model-policy-5.4.1-20211025.jar       | univocity-parsers-2.7.3.jar    |
| hive-exec-1.2.1-2.0.0.dli-20220528.090521-401.jar                | kubernetes-model-rbac-5.4.1-20211025.jar         | zstd-jni-1.4.9-1.jar           |
| hive-metastore-1.2.1-2.0.0.dli-20220528.090509-402.jar           | kubernetes-model-scheduling-5.4.1-20211025.jar   | validation-api-1.1.0.Final.jar |
| hive-shims-0.23-1.2.1-2.0.0.dli-20220528.090445-403.jar          | kubernetes-model-storageclass-5.4.1-20211025.jar | velocity-1.7.jar               |
| hive-shims-1.2.1-2.0.0.dli-20220528.090455-403.jar               | leveldbjni-all-1.8-20191105.jar                  | woodstox-core-5.0.3.jar        |

| 依赖包名称                                                        |                                          |                               |
|--------------------------------------------------------------|------------------------------------------|-------------------------------|
| hive-shims-common-1.2.1-2.0.0.dli-20220528.090441-404.jar    | libfb303-0.9.3.jar                       | xbean-asm6-shaded-4.8.jar     |
| hive-shims-scheduler-1.2.1-2.0.0.dli-20220528.090450-403.jar | libthrift-0.12.0.jar                     | xercesImpl-2.12.0.jar         |
| hk2-api-2.4.0-b34.jar                                        | log4j-1.2.17-cloudera1.jar               | xml-apis-1.4.01.jar           |
| hk2-locator-2.4.0-b34.jar                                    | log4j-rolling-appender-20131024-2017.jar | xz-1.0.jar                    |
| hk2-utils-2.4.0-b34.jar                                      | logging-interceptor-3.14.9.jar           | zjsonpatch-0.3.0.jar          |
| hppc-0.7.2.jar                                               | lucene-analyzers-common-7.7.2.jar        | zookeeper-3.5.6-ei-302002.jar |
| htrace-core4-4.2.0-incubating-1.0.0.jar                      | lucene-core-7.7.2.jar                    | -                             |

## Spark 2.3.2 依赖包

表 15-13 Spark 2.3.2 依赖包

| 依赖包名称                                   |                                                       |                                             |
|-----------------------------------------|-------------------------------------------------------|---------------------------------------------|
| accessors-smart-1.2.jar                 | HikariCP-java7-2.4.12.jar                             | logging-interceptor-3.14.4.jar              |
| activation-1.1.1.jar                    | hive-common-1.2.1-2.1.0.dli-20201111.064115-91.jar    | luxor-encrypt-2.1.0-20201106.065437-53.jar  |
| aircompressor-0.8.jar                   | hive-exec-1.2.1-2.1.0.dli-20201111.064444-91.jar      | luxor-fs3-2.1.0-20201106.065612-53.jar      |
| alluxio-2.3.1-luxor-SNAPSHOT-client.jar | hive-metastore-1.2.1-2.1.0.dli-20201111.064230-91.jar | luxor-obs-fs3-2.1.0-20201106.065616-53.jar  |
| antlr-2.7.7.jar                         | hk2-api-2.4.0-b34.jar                                 | luxor-rpc_2.11-2.1.0-20201106.065541-53.jar |



| 依赖包名称                                |                                            |                                                  |
|--------------------------------------|--------------------------------------------|--------------------------------------------------|
| antlr4-runtime-4.8-1.jar             | hk2-locator-2.4.0-b34.jar                  | luxor-rpc-protobuf2-2.1.0-20201106.065551-53.jar |
| antlr-runtime-3.4.jar                | hk2-utils-2.4.0-b34.jar                    | lz4-java-1.7.1.jar                               |
| aopalliance-1.0.jar                  | hppc-0.7.2.jar                             | machinist_2.11-0.6.1.jar                         |
| aopalliance-repackaged-2.4.0-b34.jar | htrace-core4-4.2.0-incubating-1.0.0.jar    | macro-compatible_2.11-1.1.1.jar                  |
| apache-log4j-extras-1.2.17.jar       | httpClient-4.5.4.jar                       | metrics-core-3.1.5.jar                           |
| arpack_combined_all-0.1.jar          | httpcore-4.4.7.jar                         | metrics-graphite-3.1.5.jar                       |
| arrow-format-0.8.0.jar               | ivy-2.4.0.jar                              | metrics-jmx-4.1.12.1.jar                         |
| arrow-memory-0.8.0.jar               | j2objc-annotations-1.3.jar                 | metrics-json-3.1.5.jar                           |
| arrow-vector-0.8.0.jar               | jackson-annotations-2.10.0.jar             | metrics-jvm-3.1.5.jar                            |
| asm-5.0.4.jar                        | jackson-core-2.10.0.jar                    | minlog-1.3.0.jar                                 |
| audience-annotations-0.5.0.jar       | jackson-core-asl-1.9.13-atlassian-4.jar    | mssql-jdbc-6.2.1.jre7.jar                        |
| automaton-1.11-8.jar                 | jackson-databind-2.10.0.jar                | netty-3.10.6.Final.jar                           |
| avro-1.7.7.jar                       | jackson-dataformat-yaml-2.10.0.jar         | netty-all-4.1.51.Final.jar                       |
| avro-ipc-1.7.7.jar                   | jackson-datatype-jsr310-2.10.3.jar         | nimbus-jose-jwt-8.19.jar                         |
| avro-ipc-1.7.7-tests.jar             | jackson-jaxrs-base-2.10.3.jar              | objenesis-2.1.jar                                |
| avro-mapred-1.7.7-hadoop2.jar        | jackson-jaxrs-json-provider-2.10.3.jar     | okhttp-3.14.4.jar                                |
| java-sdk-bundle-1.11.271.jar         | jackson-mapper-asl-1.9.13-atlassian-4.jar  | okio-1.17.2.jar                                  |
| base64-2.3.8.jar                     | jackson-module-jaxb-annotations-2.10.3.jar | opencsv-2.3.jar                                  |
| bcpkix-jdk15on-1.66.jar              | jackson-module-paranamer-2.10.0.jar        | opencsv-4.6.jar                                  |
| bcprov-jdk15on-1.66.jar              | jackson-module-scala_2.11-2.10.0.jar       | opencv-4.3.0-2.jar                               |

| 依赖包名称                                    |                                  |                                 |
|------------------------------------------|----------------------------------|---------------------------------|
| bonecp-0.8.0.RELEASE.jar                 | jakarta.activation-api-1.2.1.jar | orc-core-1.4.4-nohive.jar       |
| breeze_2.11-0.13.2.jar                   | jakarta.xml.bind-api-2.3.2.jar   | orc-mapreduce-1.4.4-nohive.jar  |
| breeze-macros_2.11-0.13.2.jar            | janino-3.0.8.jar                 | oro-2.0.8.jar                   |
| calcite-avatica-1.2.0-incubating.jar     | javacpp-1.5.4.jar                | osgi-resource-locator-1.0.1.jar |
| calcite-core-1.2.0-incubating.jar        | javacpp-1.5.4-linux-x86_64.jar   | paranamer-2.8.jar               |
| calcite-linq4j-1.2.0-incubating.jar      | javacv-1.5.4.jar                 | parquet-column-1.8.3.jar        |
| checker-qual-2.11.1.jar                  | JavaEWAH-1.1.7.jar               | parquet-common-1.8.3.jar        |
| chill_2.11-0.8.4.jar                     | javassist-3.18.1-GA.jar          | parquet-encoding-1.8.3.jar      |
| chill-java-0.8.4.jar                     | javax.annotation-api-1.2.jar     | parquet-format-2.3.1.jar        |
| commons-beanutils-1.9.4.jar              | javax.inject-1.jar               | parquet-hadoop-1.8.3.jar        |
| commons-cli-1.2.jar                      | javax.inject-2.4.0-b34.jar       | parquet-hadoop-bundle-1.6.0.jar |
| commons-codec-2.0-20130428.202122-59.jar | javax.servlet-api-3.1.0.jar      | parquet-jackson-1.8.3.jar       |
| commons-collections-3.2.2.jar            | javax.ws.rs-api-2.0.1.jar        | parquet-format-2.3.1.jar        |
| commons-collections4-4.2.jar             | java-xmlbuilder-1.1.jar          | parquet-hadoop-1.8.3.jar        |
| commons-compiler-3.0.8.jar               | javolution-5.3.1.jar             | parquet-hadoop-bundle-1.6.0.jar |
| commons-compress-1.4.1.jar               | jaxb-api-2.2.11.jar              | parquet-jackson-1.8.3.jar       |
| commons-configuration2-2.1.1.jar         | jcip-annotations-1.0-1.jar       | postgresql-42.2.14.jar          |
| commons-crypto-1.0.0-20191105.jar        | jcl-over-slf4j-1.7.26.jar        | protobuf-java-2.5.0.jar         |

| 依赖包名称                       |                                          |                                                                  |
|-----------------------------|------------------------------------------|------------------------------------------------------------------|
| commons-daemon-1.0.13.jar   | jdo-api-3.0.1.jar                        | py4j-0.10.7.jar                                                  |
| commons-dbcp-1.4.jar        | jersey-client-2.23.1.jar                 | pyrolite-4.13.jar                                                |
| commons-dbcp2-2.7.0.jar     | jersey-common-2.23.1.jar                 | re2j-1.1.jar                                                     |
| commons-httpclient-3.1.jar  | jersey-container-servlet-2.23.1.jar      | RoaringBitmap-0.5.11.jar                                         |
| commons-io-2.5.jar          | jersey-container-servlet-core-2.23.1.jar | scala-compiler-2.11.12.jar                                       |
| commons-lang-2.6.jar        | jersey-guava-2.23.1.jar                  | scala-library-2.11.12.jar                                        |
| commons-lang3-3.5.jar       | jersey-media-jaxb-2.23.1.jar             | scalap-2.11.0.jar                                                |
| commons-logging-1.2.jar     | jersey-server-2.23.1.jar                 | scala-parser-combinators_2.11-1.1.0.jar                          |
| commons-math3-3.4.1.jar     | jets3t-0.9.4.jar                         | scala-reflect-2.11.12.jar                                        |
| commons-net-2.2.jar         | jetty-http-9.4.31.v20200723.jar          | scala-xml_2.11-1.0.5.jar                                         |
| commons-pool-1.5.4.jar      | jetty-io-9.4.31.v20200723.jar            | secComponentApi-1.0.5c.jar                                       |
| commons-pool2-2.8.0.jar     | jetty-security-9.4.31.v20200723.jar      | shapeless_2.11-2.3.2.jar                                         |
| commons-text-1.3.jar        | jetty-server-9.4.31.v20200723.jar        | slf4j-api-1.7.30.jar                                             |
| compress-lzf-1.0.3.jar      | jetty-servlet-9.4.31.v20200723.jar       | slf4j-log4j12-1.7.30.jar                                         |
| core-1.1.2.jar              | jetty-util-9.4.31.v20200723.jar          | snakeyaml-1.24.jar                                               |
| curator-client-4.2.0.jar    | jetty-util-ajax-9.4.31.v20200723.jar     | snappy-java-1.1.7.5.jar                                          |
| curator-framework-4.2.0.jar | jetty-webapp-9.4.31.v20200723.jar        | spark-catalyst_2.11-2.3.2.0101-2.1.0.dli-20201111.073826-143.jar |

| 依赖包名称                             |                                |                                                                         |
|-----------------------------------|--------------------------------|-------------------------------------------------------------------------|
| curator-recipes-2.7.1.jar         | jetty-xml-9.4.31.v20200723.jar | spark-core_2.11-2.3.2.0101-0.dli-20201111.073836-134.jar                |
| datanucleus-api-jdo-3.2.6.jar     | joda-time-2.9.3.jar            | spark-graphx_2.11-2.3.2.0101-2.1.0.dli-20201111.073847-129.jar          |
| datanucleus-core-3.2.10.jar       | jodd-core-4.2.0.jar            | spark-hive_2.11-2.3.2.0101-0.dli-20201111.073854-132.jar                |
| datanucleus-rdbms-3.2.9.jar       | json-20200518.jar              | spark-kubernetes_2.11-2.3.2.0101-2.1.0.dli-20201111.073916-85.jar       |
| derby-10.12.1.1.jar               | json4s-ast_2.11-3.2.11.jar     | spark-kvstore_2.11-2.3.2.0101-2.1.0.dli-20201111.073933-127.jar         |
| dnsjava-2.1.7.jar                 | json4s-core_2.11-3.2.11.jar    | spark-launcher_2.11-2.3.2.0101-2.1.0.dli-20201111.073940-127.jar        |
| ehcache-3.3.1.jar                 | json4s-jackson_2.11-3.2.11.jar | spark-mllib_2.11-2.3.2.0101-2.1.0.dli-20201111.073946-127.jar           |
| eigenbase-properties-1.1.5.jar    | json-sanitizer-1.2.1.jar       | spark-mllib-local_2.11-2.3.2.0101-2.1.0.dli-20201111.073953-127.jar     |
| error_prone_annotations-2.3.4.jar | json-smart-2.3.jar             | spark-network-common_2.11-2.3.2.0101-2.1.0.dli-20201111.073959-127.jar  |
| failureaccess-1.0.1.jar           | jsp-api-2.1.jar                | spark-network-shuffle_2.11-2.3.2.0101-2.1.0.dli-20201111.074007-127.jar |

| 依赖包名称                                    |                         |                                                                    |
|------------------------------------------|-------------------------|--------------------------------------------------------------------|
| fastutil-8.2.3.jar                       | jsr305-3.0.2.jar        | spark-om_2.11-2.3.2.0101-.0.dli-20201111.074019-125.jar            |
| ffmpeg-4.3.1-1.5.4.jar                   | jta-1.1.jar             | spark-repl_2.11-2.3.2.0101-2.1.0.dli-20201111.074028-125.jar       |
| ffmpeg-4.3.1-1.5.4-linux-x86_64.jar      | jtransforms-2.4.0.jar   | spark-sketch_2.11-2.3.2.0101-2.1.0.dli-20201111.074035-125.jar     |
| flatbuffers-1.2.0-3f79e055.jar           | jul-to-slf4j-1.7.26.jar | spark-sql_2.11-2.3.2.0101-2.1.0.dli-20201111.074041-126.jar        |
| generex-1.0.2.jar                        | junit-4.11.jar          | spark-streaming_2.11-2.3.2.0101-2.1.0.dli-20201111.074100-123.jar  |
| geronimo-jcache_1.0_spec-1.0-alpha-1.jar | kerb-admin-1.0.1.jar    | spark-tags_2.11-2.3.2.0101-2.1.0.dli-20201111.074136-123.jar       |
| gson-2.2.4.jar                           | kerb-client-1.0.1.jar   | spark-tags_2.11-2.3.2.0101-2.1.0.dli-20201111.074141-124-tests.jar |
| guava-29.0-jre.jar                       | kerb-common-1.0.1.jar   | spark-unsafe_2.11-2.3.2.0101-2.1.0.dli-20201111.074144-123.jar     |
| guice-4.0.jar                            | kerb-core-1.0.1.jar     | spark-uquery_2.11-2.3.2.0101-2.1.0.dli-20201111.074906-210.jar     |
| guice-servlet-4.0.jar                    | kerb-crypto-1.0.1.jar   | spark-yarn_2.11-2.3.2.0101-2.1.0.dli-20201111.074151-123.jar       |
| hadoop-annotations-3.1.1-ei-302002.jar   | kerb-identity-1.0.1.jar | spire_2.11-0.13.0.jar                                              |

| 依赖包名称                                                 |                                                                |                                |
|-------------------------------------------------------|----------------------------------------------------------------|--------------------------------|
| hadoop-auth-3.1.1-ei-302002.jar                       | kerb-server-1.0.1.jar                                          | spire-macros_2.11-0.13.0.jar   |
| hadoop-3.1.1-ei-302002.jar                            | kerb-simplekdc-1.0.1.jar                                       | ST4-4.3.1.jar                  |
| hadoop-client-3.1.1-ei-302002.jar                     | kerb-util-1.0.1.jar                                            | stax2-api-3.1.4.jar            |
| hadoop-common-3.1.1-ei-302002.jar                     | kerby-asn1-1.0.1.jar                                           | stax-api-1.0-2.jar             |
| hadoop-hdfs-3.1.1-ei-302002.jar                       | kerby-config-1.0.1.jar                                         | stream-2.7.0.jar               |
| hadoop-hdfs-client-3.1.1-ei-302002.jar                | kerby-pkix-1.0.1.jar                                           | stringtemplate-3.2.1.jar       |
| hadoop-3.1.1-41.jar                                   | kerby-util-1.0.1.jar                                           | token-provider-1.0.1.jar       |
| hadoop-mapreduce-client-common-3.1.1-ei-302002.jar    | kerby-xdr-1.0.1.jar                                            | univocity-parsers-2.5.9.jar    |
| hadoop-mapreduce-client-core-3.1.1-ei-302002.jar      | kryo-shaded-3.0.3.jar                                          | validation-api-1.1.0.Final.jar |
| hadoop-mapreduce-client-jobclient-3.1.1-ei-302002.jar | kubernetes-client-4.9.2-20200804.jar                           | woodstox-core-5.0.3.jar        |
| hadoop-minikdc-3.1.1-ei-302002.jar                    | kubernetes-model-4.9.2-20200804.jar                            | xbean-asm5-shaded-4.4.jar      |
| hadoop-yarn-api-3.1.1-ei-302002.jar                   | kubernetes-model-common-4.9.2-20200804.jar                     | xercesImpl-2.12.0.jar          |
| hadoop-yarn-client-3.1.1-ei-302002.jar                | leveldbjni-all-1.8-20191105.jar                                | xml-apis-1.4.01.jar            |
| hadoop-yarn-common-3.1.1-ei-302002.jar                | libfb303-0.9.3.jar                                             | xz-1.0.jar                     |
| hadoop-yarn-registry-3.1.1-ei-302002.jar              | libthrift-0.12.0.jar                                           | zjsonpatch-0.3.0.jar           |
| hadoop-yarn-server-common-3.1.1-ei-302002.jar         | listenablefuture-9999.0-empty-to-avoid-conflict-with-guava.jar | zookeeper-3.5.6-ei-302002.jar  |

| 依赖包名称                                            |                                          |                                    |
|--------------------------------------------------|------------------------------------------|------------------------------------|
| hadoop-yarn-server-web-proxy-3.1.1-ei-302002.jar | log4j-1.2.17-cloudera1.jar               | zookeeper-jute-3.5.6-ei-302002.jar |
| hamcrest-core-1.3.jar                            | log4j-rolling-appender-20131024-2017.jar | zstd-jni-1.4.4-11.jar              |

## Flink 1.15 依赖包

请在Flink作业的日志中获取Flink 1.15相关依赖包信息：

1. 查看Flink日志。
  - a. 登录DLI管理控制台，选择“作业管理 > Flink作业”。
  - b. 单击作业名称，选择“运行日志”。
  - c. 控制台只展示最新的运行日志，更多日志信息请查看保存日志的OBS桶。
2. 在日志中搜索依赖包信息。  
在日志中搜索“Classpath:”即可查看相关依赖包信息。

## Flink 1.12 依赖包

表 15-14 Flink 1.12 依赖包

| 依赖包名称                                         |                                                                   |                                             |
|-----------------------------------------------|-------------------------------------------------------------------|---------------------------------------------|
| bcpkix-jdk15on-1.60.jar                       | flink-json-1.12.2-ei-313001-dli-2022011002.jar                    | libtensorflow-1.12.0.jar                    |
| bcprov-jdk15on-1.60.jar                       | flink-kubernetes_2.11-1.12.2-ei-313001-dli-2022011002.jar         | log4j-1.2-api-2.17.1.jar                    |
| clickhouse-jdbc-0.3.1-ei-313001-SNAPSHOT.jar  | flink-metrics-prometheus_2.11-1.12.2-ei-313001-dli-2022011002.jar | log4j-api-2.17.1.jar                        |
| commons-codec-1.9.jar                         | flink-obs-hadoop-fs-2.0.0-20220226.034421-73.jar                  | log4j-core-2.17.1.jar                       |
| commons-configuration-1.7.jar                 | flink-s3-fs-hadoop-1.12.2.jar                                     | log4j-slf4j-impl-2.17.1.jar                 |
| dataflow-fs-obs-2.0.0-20220226.034402-190.jar | flink-shaded-zookeeper-3.6.3-ei-313001-SNAPSHOT.jar               | luxor-encrypt-2.0.0-20220405.072004-199.jar |

| 依赖包名称                                                    |                                                                       |                                                           |
|----------------------------------------------------------|-----------------------------------------------------------------------|-----------------------------------------------------------|
| deeplearning4j-core-0.9.1.jar                            | flink-sql-avro-1.12.2-ei-313001-dli-2022011002.jar                    | luxor-fs3-2.0.0-20220405.072025-195.jar                   |
| deeplearning4j-nlp-0.9.1.jar                             | flink-sql-avro-confluent-registry-1.12.2-ei-313001-dli-2022011002.jar | luxor-obs-fs3-2.0.0-20220405.072030-195.jar               |
| deeplearning4j-nn-0.9.1.jar                              | flink-table_2.11-1.12.2-ei-313001-dli-2022011002.jar                  | manager-hadoop-security-crypter-8.1.3-313001-SNAPSHOT.jar |
| ejml-cdense-0.33.jar                                     | flink-table-blink_2.11-1.12.2-ei-313001-dli-2022011002.jar            | manager-wc2frm-8.1.3-313001-SNAPSHOT.jar                  |
| ejml-core-0.33.jar                                       | guava-18.0.jar                                                        | mrs-obs-provider-3.1.1.49.jar                             |
| ejml-ddense-0.33.jar                                     | guava-26.0-jre.jar                                                    | nd4j-api-0.9.1.jar                                        |
| ejml-dsparse-0.33.jar                                    | hadoop-hdfs-client-3.1.1-ei-302002.jar                                | nd4j-native-0.9.1.jar                                     |
| ejml-experimental-0.33.jar                               | hadoop-3.1.1-46.jar                                                   | nd4j-native-api-0.9.1.jar                                 |
| ejml-fdense-0.33.jar                                     | hadoop-plugins-8.1.3-313001-SNAPSHOT.jar                              | nd4j-native-platform-0.9.1.jar                            |
| ejml-simple-0.33.jar                                     | httpasyncclient-4.1.2.jar                                             | okhttp-3.14.8.jar                                         |
| ejml-zdense-0.33.jar                                     | httpClient-4.5.3.jar                                                  | okio-1.14.0.jar                                           |
| elsa-3.0.0-M7.jar                                        | httpcore-4.4.4.jar                                                    | ranger-obs-client-0.1.1.jar                               |
| flink-changelog-json-1.12.2-ei-313001-dli-2022011002.jar | httpcore-nio-4.4.4.jar                                                | secComponentApi-1.0.5.jar                                 |
| flink-csv-1.12.2-ei-313001-dli-2022011002.jar            | java-xmlbuilder-1.1.jar                                               | slf4j-api-1.7.26.jar                                      |
| flink-dist_2.11-1.12.2-ei-313001-dli-2022011002.jar      | jna-4.1.0.jar                                                         | tensorflow-1.12.0.jar                                     |



## Flink 1.10 依赖包

Flink 1.10作业程序开发的样例代码可以参考[使用Flink Jar写入数据到OBS开发指南](#)。  
2020年12月之后创建的新队列才能使用Flink 1.10依赖包。

表 15-15 Flink 1.10 依赖包

| 依赖包名称                         |                                               |                                |
|-------------------------------|-----------------------------------------------|--------------------------------|
| bcpkix-jdk15on-1.60.jar       | esdk-obs-java-3.20.6.1.jar                    | java-xmlbuilder-1.1.jar        |
| bcprov-jdk15on-1.60.jar       | flink-cep_2.11-1.10.0.jar                     | jna-4.1.0.jar                  |
| commons-codec-1.9.jar         | flink-cep-scala_2.11-1.10.0.jar               | libtensorflow-1.12.0.jar       |
| commons-configuration-1.7.jar | flink-dist_2.11-1.10.0.jar                    | log4j-over-slf4j-1.7.26.jar    |
| deeplearning4j-core-0.9.1.jar | flink-python_2.11-1.10.0.jar                  | logback-classic-1.2.3.jar      |
| deeplearning4j-nlp-0.9.1.jar  | flink-queryable-state-runtime_2.11-1.10.0.jar | logback-core-1.2.3.jar         |
| deeplearning4j-nn-0.9.1.jar   | flink-sql-client_2.11-1.10.0.jar              | nd4j-api-0.9.1.jar             |
| ejml-cdense-0.33.jar          | flink-state-processor-api_2.11-1.10.0.jar     | nd4j-native-0.9.1.jar          |
| ejml-core-0.33.jar            | flink-table_2.11-1.10.0.jar                   | nd4j-native-api-0.9.1.jar      |
| ejml-ddense-0.33.jar          | flink-table-blink_2.11-1.10.0.jar             | nd4j-native-platform-0.9.1.jar |
| ejml-dsparse-0.33.jar         | guava-26.0-jre.jar                            | okhttp-3.14.8.jar              |
| ejml-experimental-0.33.jar    | hadoop-3.1.1-41.jar                           | okio-1.14.0.jar                |
| ejml-fdense-0.33.jar          | httpasyncclient-4.1.2.jar                     | secComponentApi-1.0.5.jar      |
| ejml-simple-0.33.jar          | httpClient-4.5.3.jar                          | slf4j-api-1.7.26.jar           |
| ejml-zdense-0.33.jar          | httpcore-4.4.4.jar                            | tensorflow-1.12.0.jar          |
| elsa-3.0.0-M7.jar             | httpcore-nio-4.4.4.jar                        | -                              |

## Flink 1.7.2 依赖包

Flink 1.7.2作业程序开发的样例代码可以参考[DLI样例代码](#)中的“luxor-demo\dlf-flink-demo”。

表 15-16 Flink 1.7.2 依赖包

| 依赖包名称                         |                                              |                                |
|-------------------------------|----------------------------------------------|--------------------------------|
| bcpkix-jdk15on-1.60.jar       | esdk-obs-java-3.1.3.jar                      | httpcore-4.4.4.jar             |
| bcprov-jdk15on-1.60.jar       | flink-cep_2.11-1.7.0.jar                     | httpcore-nio-4.4.4.jar         |
| commons-codec-1.9.jar         | flink-cep-scala_2.11-1.7.0.jar               | java-xmlbuilder-1.1.jar        |
| commons-configuration-1.7.jar | flink-dist_2.11-1.7.0.jar                    | jna-4.1.0.jar                  |
| deeplearning4j-core-0.9.1.jar | flink-gelly_2.11-1.7.0.jar                   | libtensorflow-1.12.0.jar       |
| deeplearning4j-nlp-0.9.1.jar  | flink-gelly-scala_2.11-1.7.0.jar             | log4j-over-slf4j-1.7.21.jar    |
| deeplearning4j-nn-0.9.1.jar   | flink-ml_2.11-1.7.0.jar                      | logback-classic-1.2.3.jar      |
| ejml-cdense-0.33.jar          | flink-python_2.11-1.7.0.jar                  | logback-core-1.2.3.jar         |
| ejml-core-0.33.jar            | flink-queryable-state-runtime_2.11-1.7.0.jar | nd4j-api-0.9.1.jar             |
| ejml-ddense-0.33.jar          | flink-shaded-curator-1.7.0.jar               | nd4j-native-0.9.1.jar          |
| ejml-dsparse-0.33.jar         | flink-shaded-hadoop2-uber-1.7.0.jar          | nd4j-native-api-0.9.1.jar      |
| ejml-experimental-0.33.jar    | flink-table_2.11-1.7.0.jar                   | nd4j-native-platform-0.9.1.jar |
| ejml-fdense-0.33.jar          | guava-26.0-jre.jar                           | okhttp-3.14.8.jar              |
| ejml-simple-0.33.jar          | hadoop-3.1.1-41-20201014.085840-4.jar        | okio-1.14.0.jar                |
| ejml-zdense-0.33.jar          | httpasyncclient-4.1.2.jar                    | slf4j-api-1.7.21.jar           |
| elsa-3.0.0-M7.jar             | httpClient-4.5.12.jar                        | tensorflow-1.12.0.jar          |
| log4j-api-2.16.0.jar          | log4j-core-2.16.0.jar                        | log4j-api-2.8.2.jar            |
| log4j-core-2.8.2.jar          | -                                            | -                              |

## 15.4 管理 DLI 资源配额

### 什么是配额？

为防止资源滥用，平台限定了各服务资源的配额，对用户的资源数量和容量做了限制。

如果当前资源配额限制无法满足使用需要，您可以申请扩大配额。

### 怎样查看我的配额


1. 登录管理控制台。
2. 单击管理控制台左上角的 ，选择区域和项目。
3. 在页面右上角，选择“资源 > 我的配额”。  
系统进入“服务配额”页面。

图 15-11 我的配额



4. 您可以在“服务配额”页面，查看各项资源的总配额及使用情况。  
如果当前配额不能满足业务要求，请参考后续操作，申请扩大配额。

### 如何申请扩大配额？

1. 登录管理控制台。
2. 在页面右上角，选择“资源 > 我的配额”。  
系统进入“服务配额”页面。

图 15-12 我的配额



3. 单击“申请扩大配额”。
4. 在“新建工单”页面，根据您的需求，填写相关参数。  
其中，“问题描述”项请填写需要调整的内容和申请原因。
5. 填写完毕后，勾选协议并单击“提交”。