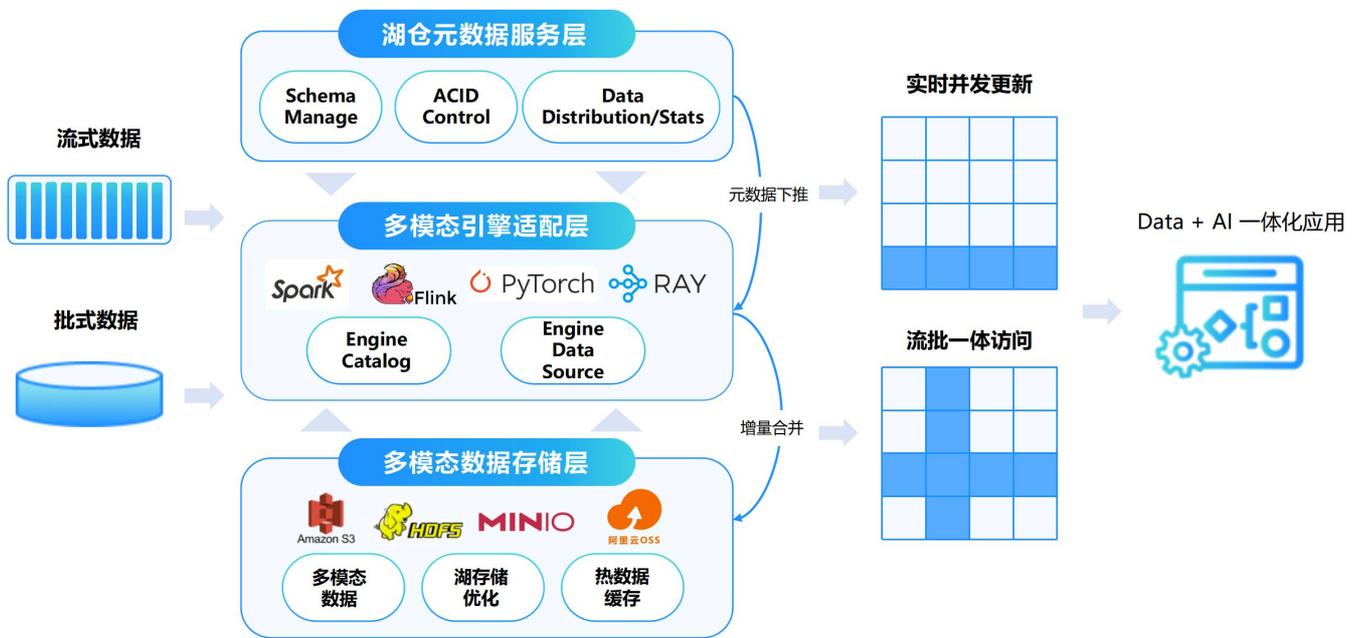


数元灵湖仓数据智能 LakeInsight 产品白皮书

1. 产品介绍

LakeInsight 是北京数元灵科技有限公司研发推出的一站式湖仓数据智能产品，涵盖实时数据集成、实时数据建模、实时数据服务、指标报表和 AI 开发等功能，一站式服务企业数据智能业务，构建数据智能中台。

LakeInsight 基于数元灵自研开源湖仓一体框架 LakeSoul。LakeSoul 是数元灵自主研发的全新一代湖仓一体框架，具备云原生、计算存储弹性、流批一体、Data+AI 一体化的设计。LakeSoul 的总体架构如下：



2. 产品核心功能

2.1 实时数据同步

(1) 支持单库单表、单数据库多表同步，根据数据库 CDC 事件增量同步数据包含的信息，自行解析出数据库，表名以及表的 schema 信息，然后实现建库建表的操作；

(2) 支持的数据类型：boolean, bit, binary, varbinary, blob, tinyblob, mediumblob, longblob, bigint, int, integer, mediumint, tinyint, smallint, float, double, date, datetime, timestamp, decimal, char, varchar, string, longtext, mediumtext, text, tinytext, json;

(3) 支持自动 Schema 变更同步。根据数据中的 Schema 信息，自动识别 DDL 变更，并同步到实时存储平台表中。支持的 Schema 变更类型包含增、减列、列数据类型改变操作(int -> long, float -> double)，并支持逻辑列删除(即数据本身删除某列后，平台可保留删除列的 schema，这样后续查询可查到源表删除列字段信息)；

(4) 保障数据准确性：实现全链路 Exactly-once，保证数据传输的不丢不重；实现数据延迟机制判断，避免因上游数据延迟导致相关异常；

(5) 支持多种断点重传机制：在上游数据出现问题等情况下，支持多种重传方式，包括指定时间戳消费，从最新数据消费等，方便快速恢复同步任务；

(6) 支持数据源数据安全机制。

2.2 实时计算存储

采用湖仓一体的实时数仓架构，兼容开源生态体系，支持多种开源计算引擎、存储引擎，能够进行增量计算、全量计算以及多种分层建模方式的 PB 级实时数仓框架，做到”流批一体、湖仓一体、AI 与 BI 一体“三位一体，满足数据可追踪、可管理、可查看以及集群可弹性伸缩的能力。具体功能包括：

(1) 元数据管理功能，支持高可用、分布式部署方案，能够满足单节点千万级别数据管理要求，支持 Domain(域)、Namespace、表、分区、数据文件多层级管理；支持高并发写入，支持 ACID，保证数据读写一致性。支持 TimeTravel，支持回滚、快照、增量读等；支持 Listen-Trigger-Notify 机制触发自动 Compaction 和数据清理支持高可用部署。

(2) 支持数据实时更新（append, upsert 方式），数据实时读取方式(MOR，增量，快照)：数据实时更新情况下，可根据主键情况，自适应采用不同数据更新方式。无主键采用 Append 方式，有主键可采用 Upsert 模式，数据在读取时会按照主键合并更新，保证同一个主键使用最新的数据；需提供 TimeTravel 能力，增量读和快照读。

(3) 支持计算引擎包括不限于批式、流式计算引擎：批式计算引擎包括 Spark 等；流式计算引擎支持 Flink、Spark Streaming 等；支持 AI、MPP 等相关计算引擎：AI 计算引擎支持 PyTorch、Pandas、Spark MLlib 等；MPP 计算引擎支持 Presto、Doris 等；提供 API 接口以支持各类开源引擎的接入和整合。

(4) 支持流批数仓建模：对接上游数据采集平台后，可使用实时、批量数据等方式写到存储计算平台；支持实时增量建模，支持以流式的方式增量读取上游数据，读取时支持 Changelog 语义，增量数据，支持 Flink 增量计算，包括双流 Join、LookupJoin、Aggregate 等，支持 CDC 输出，支持实时落盘到计算存储平台，支持实时推送到下游数据服务平台；支持批量计算建模，支持以周期调度的方式，批量执行建模计算任务。结果可以支持覆盖写（Overwrite）以及 Upsert 两种方式进行更新。批量计算作业支持 Spark SQL、Spark DataFrame API 进行相关开发。

2.3 实时数据建模分析

功能包括：基于 WEB-UI 的开发环境，在该平台可进行相关数据建模，任务提交，数据探查工作。

(1) 支持的开发语言：支持通过 SQL 方式进行相关数据研发工作；支持 Python、Java、Scala 等代码任务；支持数据任务的开发、测试、上线的一站式建模开发服务；

(2) 安全机制：支持企业内部单点登录对接；支持开发环境和生产环境隔离；支持数据域的划分，数据可读可写可执行等权限的隔离；支持各类租户的工作空间权限隔离与划分；支持并提供基于角色的权限管理功能，保证当前工作空间的各项业务及数据安全的要求；

(3) 数据建模开发：支持开发人员使用基于 WEB 的在线编辑器，进行数据处理作业设计开发工作；支持开发人员使用交互协作的方式使用 Flink、Spark 等多种语言进行实时数据建模工作，包括数据清洗、分析、可视化显示，支持报告生成等；支持管理员角色用户，将开发人员在平台中完成的任务以可配置化的方式，发布在生产环境，支持任务实时状态监控、任务的启停以及日志查询等；

(4) 平台管理：支持工作空间管理，不同用户在不同的工作空间下，不同模块有不同的操作权限；基于角色的数据开发和任务发布权限控制；基于各模块的自定义角色管理功能；支持配置开发集群和生产集群，支持开发环境和生产环境隔离；支持平台任务管理和监控，支持资源、计算任务监控报警。

2.4 实时指标报表

功能包括：可视化低代码的指标定义，方便业务人员自助定义指标。支持原子、衍生、复合指标。支持可视化大屏面板自定义图表，支持在线编辑可视化图文数据报告。

(1) 指标定义：支持原子指标定义，即指定数据库表和维度列、聚合列、聚合函数，生成指标；支持衍生指标定义，即任意指标添加过滤条件或聚合；支持复合指标定义，即任意多个指标四则运算；

(2) 图表定义：可以为每个指标，配置图表。支持折线图、柱状图、饼图、表格、数字等多种展示方式。支持将多个指标形成分组，每个分组形成图表面板，组成大屏展示；

(3) 图文报告：支持前端在线编辑图文报告，支持设置字体、段落等格式，支持在报告中选择指标和图表形式插入，打开报告时自动刷新最新的指标值。

2.5 产品部署和集成

LakeInsight 产品支持部署在 Hadoop 环境、Kubernetes 容器化环境等。支持麒麟、统信、欧拉等国产操作系统，支持海光、兆芯等国产 CPU 芯片。

LakeInsight 提供数据写入和读取接口。提供 JDBC 标准数据访问连接，可以与各类 BI 软件兼容集成。

3. 使用文档

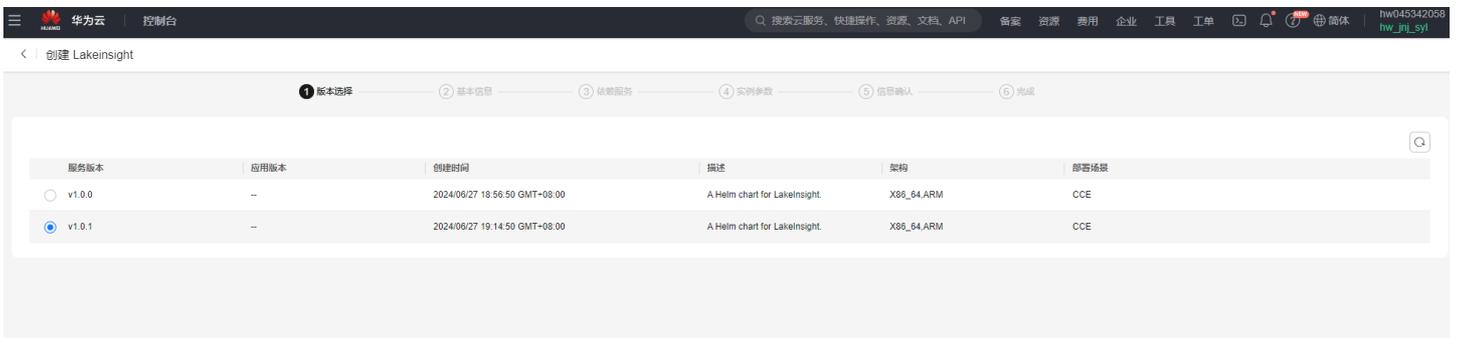
3.1 安装

3.1.1 创建实例

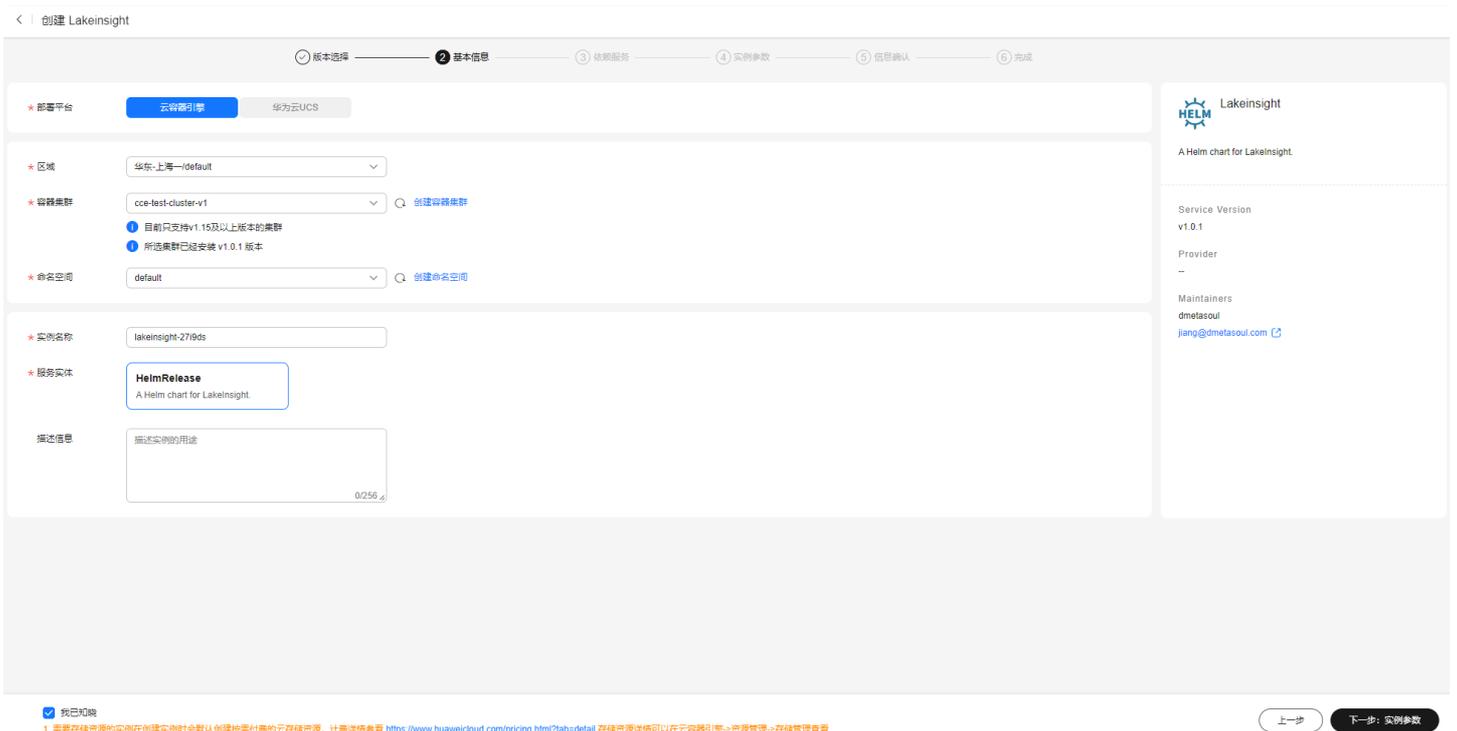
- 用户可以从【云原生服务中心】进入【我的服务】，点击【创建实例】



- 勾选选择相应的版本，并点击右下角的【下一步】。

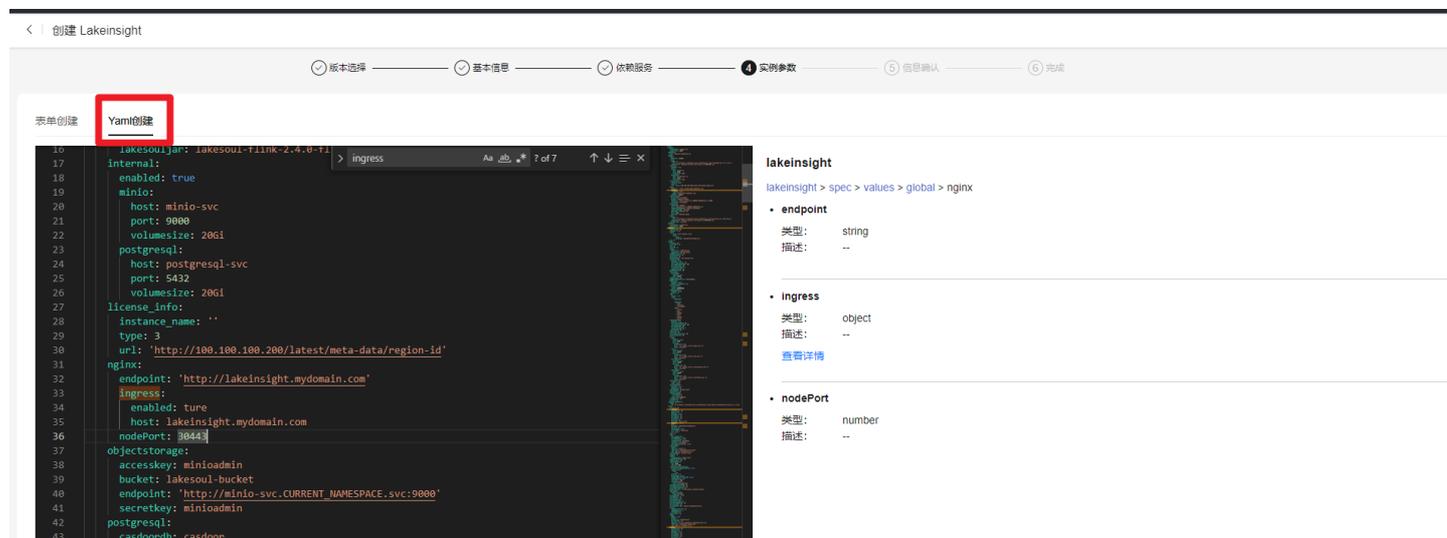
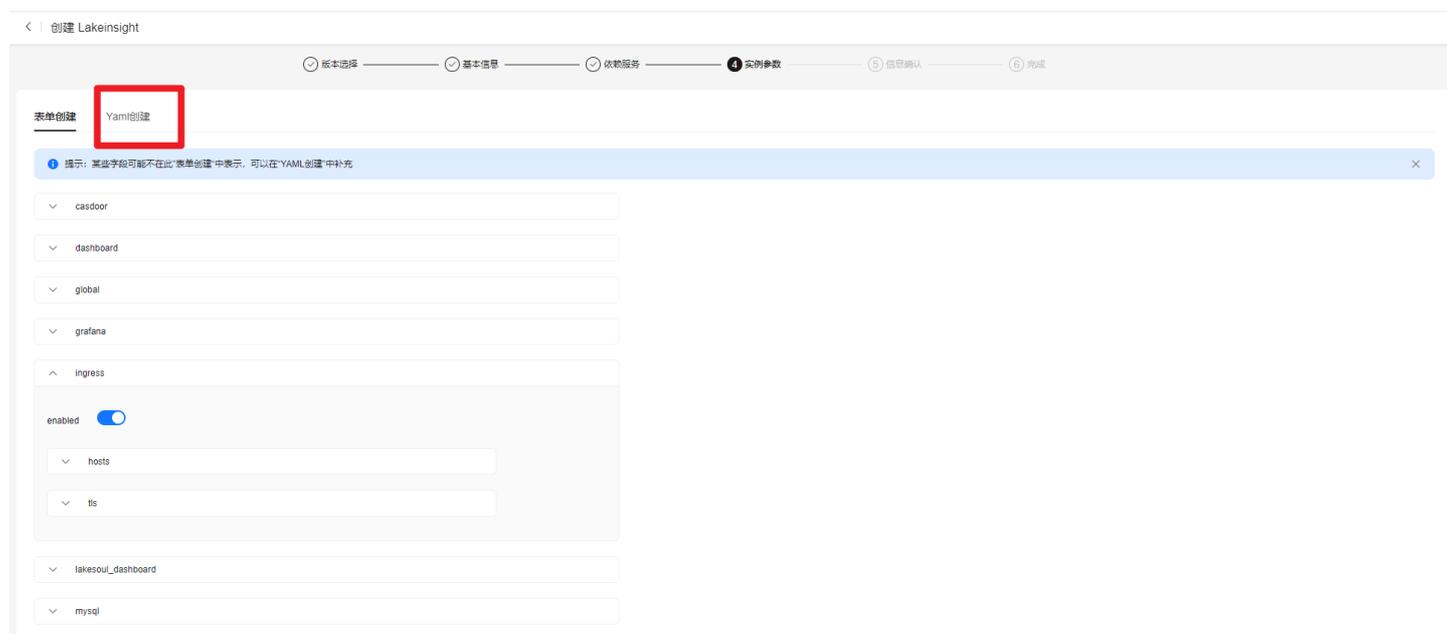


- 勾选相应的【基本信息】，并点击右下角的【下一步】。



- 配置访问方式 (Ingress 和 NodePort 两种方式)

选择 Yaml 创建：



• 方式一：Ingress配置方式（默认）

如果用户希望通过Ingress独立域名的去访问，需要将ingress开启，为endpoint配置 http://ingress_host，可参考如下配置方式：

```
1 #####Important Values#####
2 global:
3   admin:
4     password: 123456
5   nginx:
6     endpoint: http://lakeinsight.mydomain.com
7     #if use nodePort, set endpoint like: http://192.168.18.167:30443
8     nodePort: 30447
9     #if use ingress, set endpoint like: http://lakeinsight.mydomain.com
10  ingress:
11    enabled: true
12    host: lakeinsight.mydomain.com
13
14  internal:
15    enabled: true
16  minio:
17    host: minio-svc
18    port: 9000
19    volumeSize: 8Gi
20  postgresql:
21    host: postgresql-svc
22    port: 5432
23    volumeSize: 8Gi
```

- 方式二：NodePort配置方式

如果用户希望通过NodePort方式访问，需要预定义集群中未被占用的nodePort，为endpoint配置 <http://NodeIP:NodePort>,可参考如下配置方式：

```
1 #####Important Values#####
2 global:
3   admin:
4     password: 123456
5   nginx:
6     endpoint: http://192.168.18.167:30447
7     #if use nodePort, set endpoint like: http://192.168.18.167:30443
8     nodePort: 30447
9     #if use ingress, set endpoint like: http://lakeinsight.mydomain.com
10  ingress:
11    enabled: false
12    host: lakeinsight.mydomain.com
13
14  internal:
15    enabled: true
16  minio:
17    host: minio-svc
18    port: 9000
19    volumeSize: 8Gi
20  postgresql:
21    host: postgresql-svc
22    port: 5432
23    volumeSize: 8Gi
```

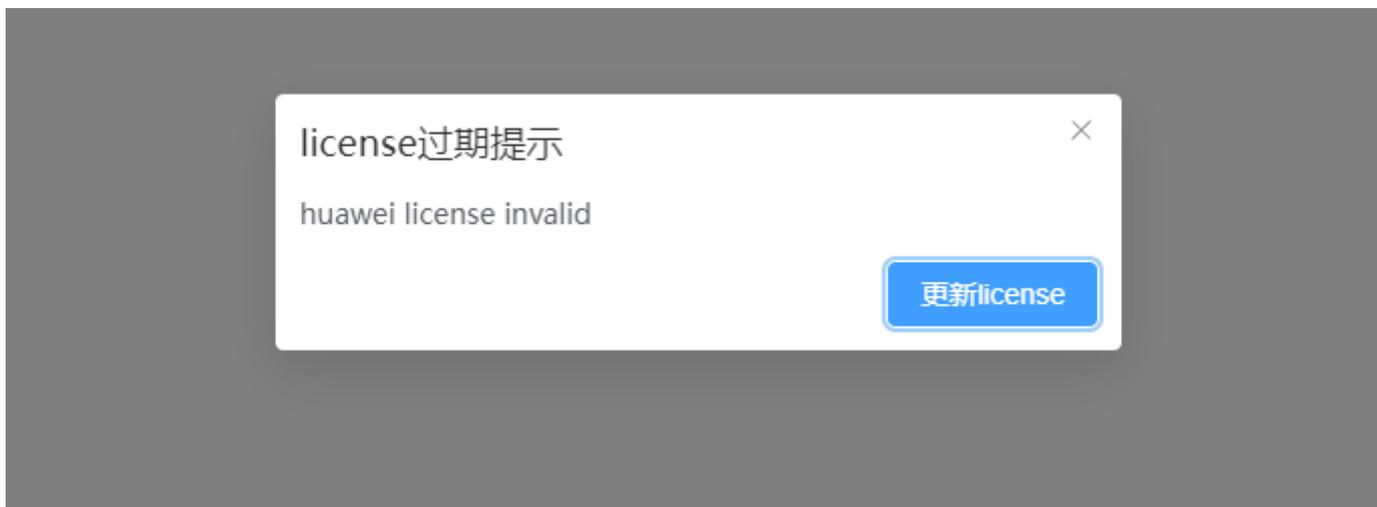
- 信息确认并完成创建

基本信息

部署平台	云容器引擎	区域	华东-上海一
容器集群	cce-test-cluster-v1	命名空间	lakeinsight
实例名称	lakeinsight-27f9ds	描述信息	--

3.2 注册 Licence

登录首页，如果没有申请过 licence 或 licence 已过期，则会弹出提示窗口，点击 **【更新licence】**



鼠标移动到右下角，会提示向邮箱 [【hw-support@dmetasoul.com】](mailto:hw-support@dmetasoul.com) 发送邮件申请 license

LICENSE

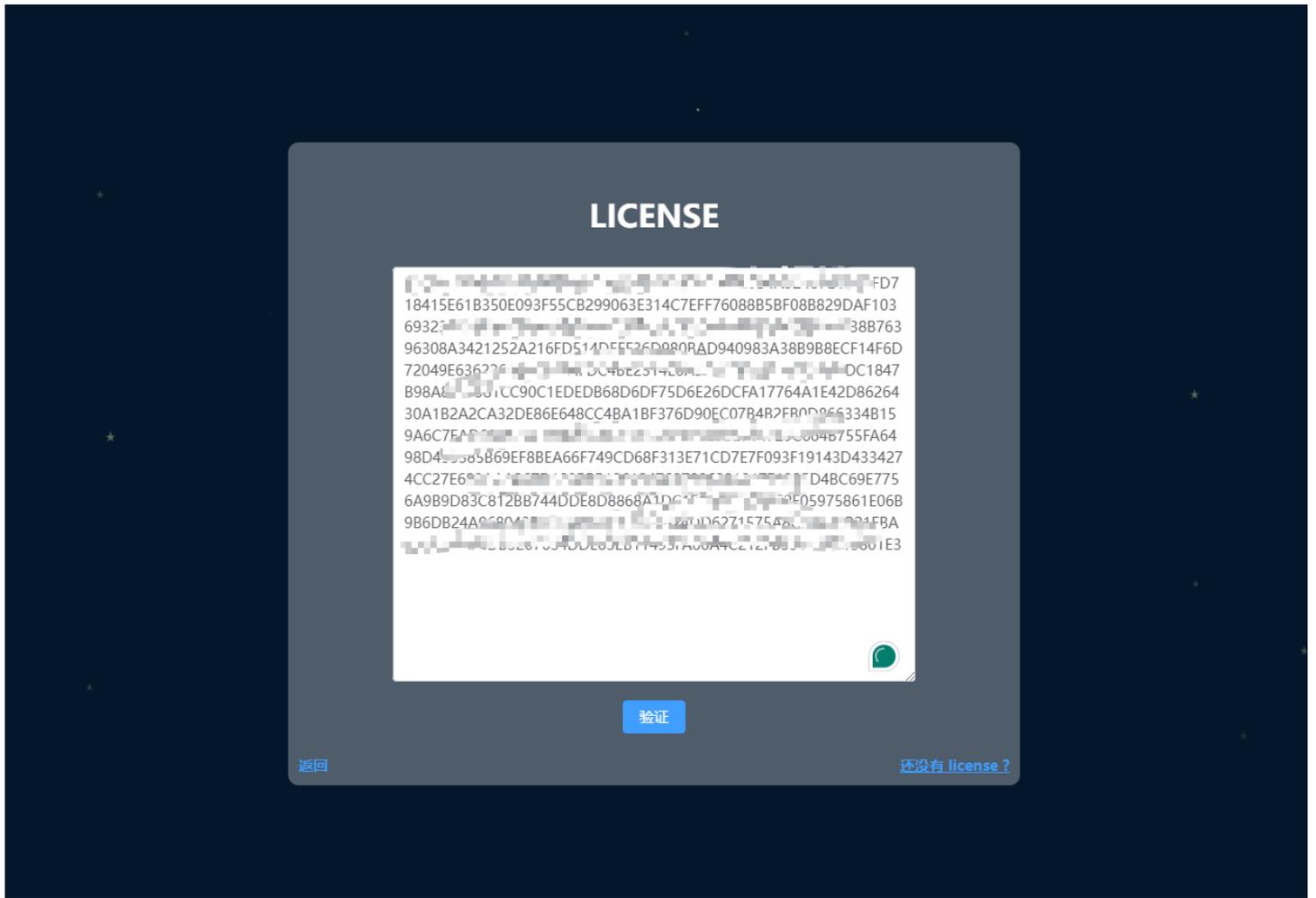
返回

验证

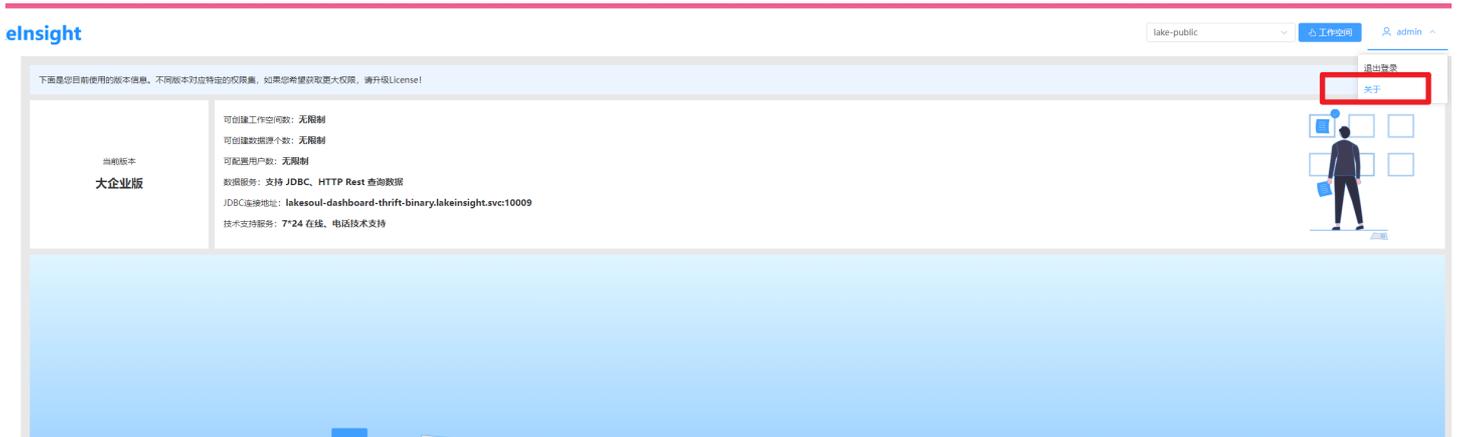
还没有 license ?

请联系邮箱 hw-support@dmetasoul.com 获取license

将获取到的 licence 粘贴至输入框里，并点击【验证】



右上角【关于】可以查看当前 licence 版本。

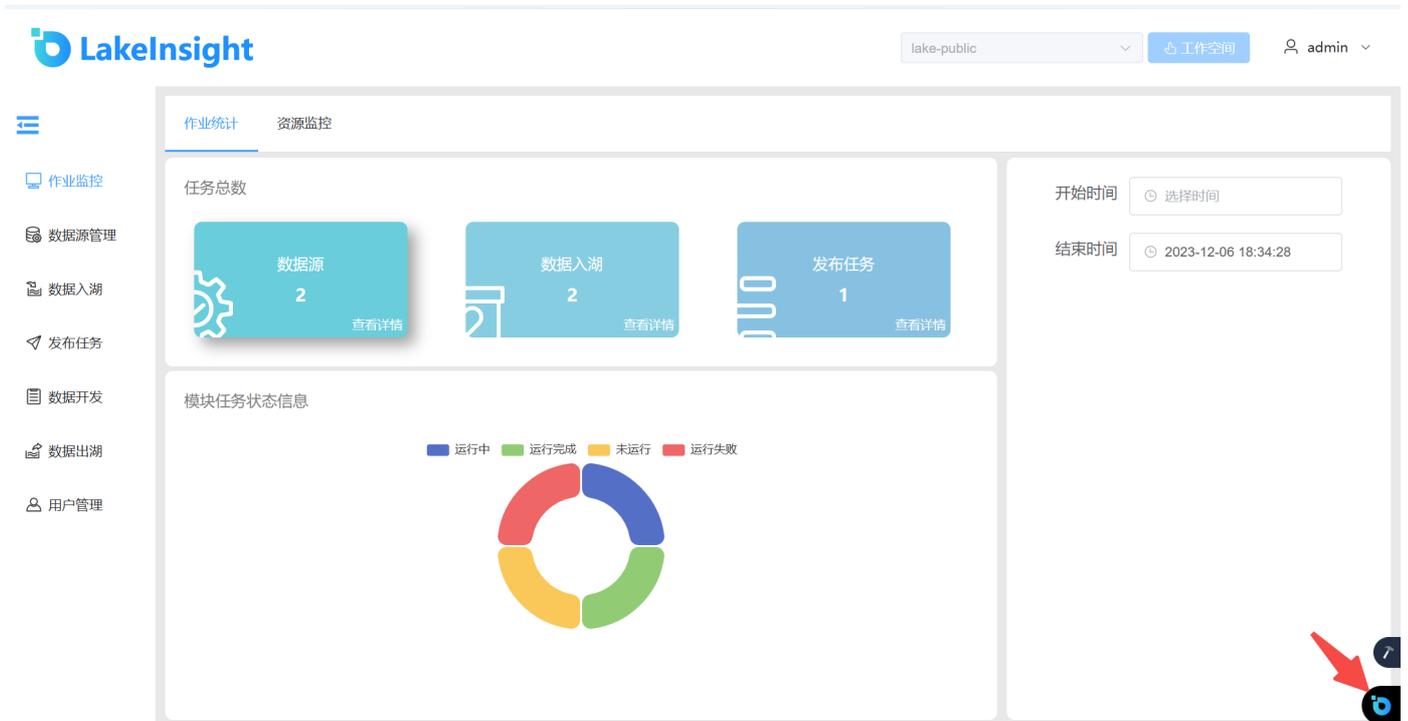


3.3 登录

1. 进入 LakeInsight 前，未登录的情况下，先跳转到登录页进行登录认证。使用上述安装时配置的账号和密码进行登录（我们当前的用户管理，引入了第三方的用户管理系统casdoor）。



2. 登录认证通过后，会进入到LakeInsight的首页，根据需求通过切换左侧导航选择内容。（可以通过右下角logo处 单独打开LakeInsight）

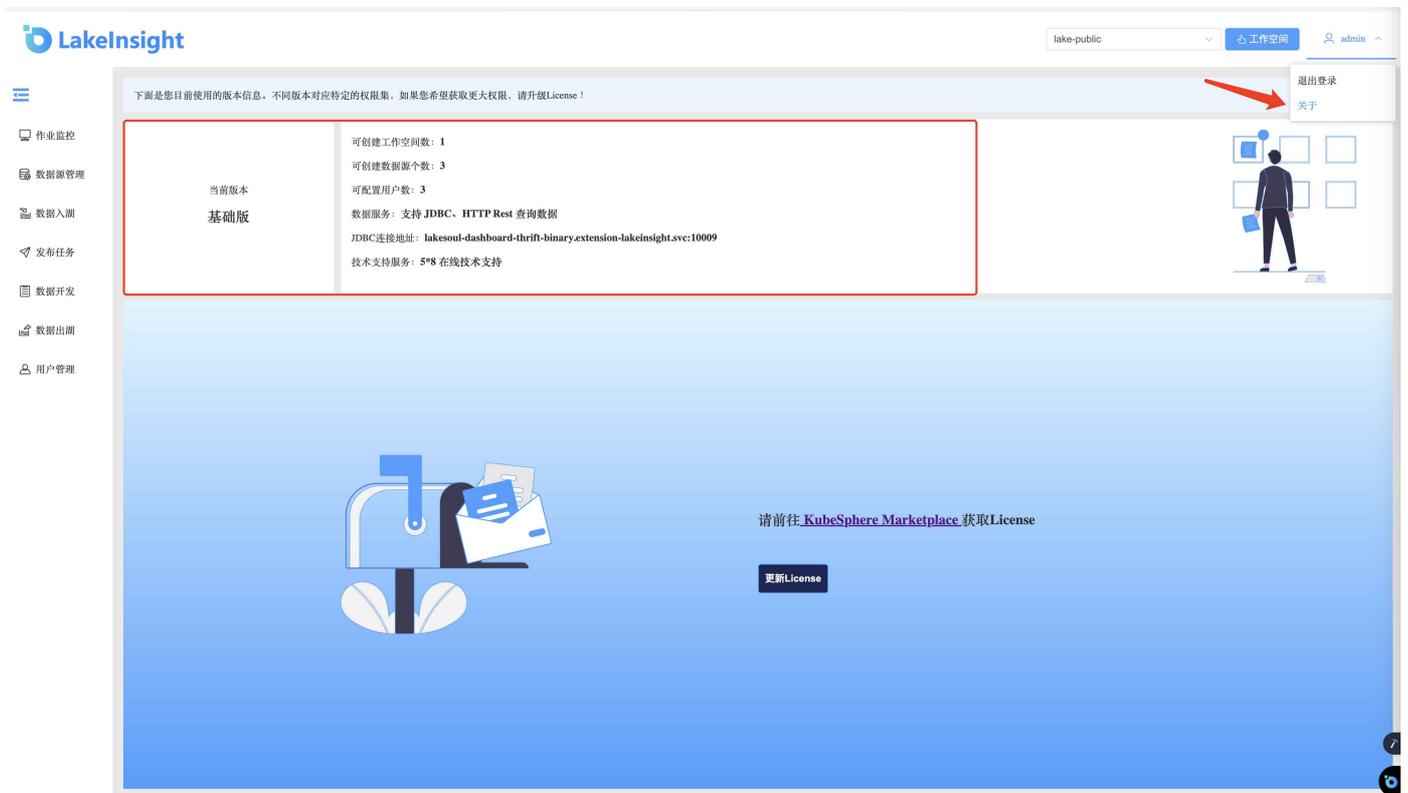


3. 登出

点击**退出登陆**按钮，可以退出当前系统。

4. 关于

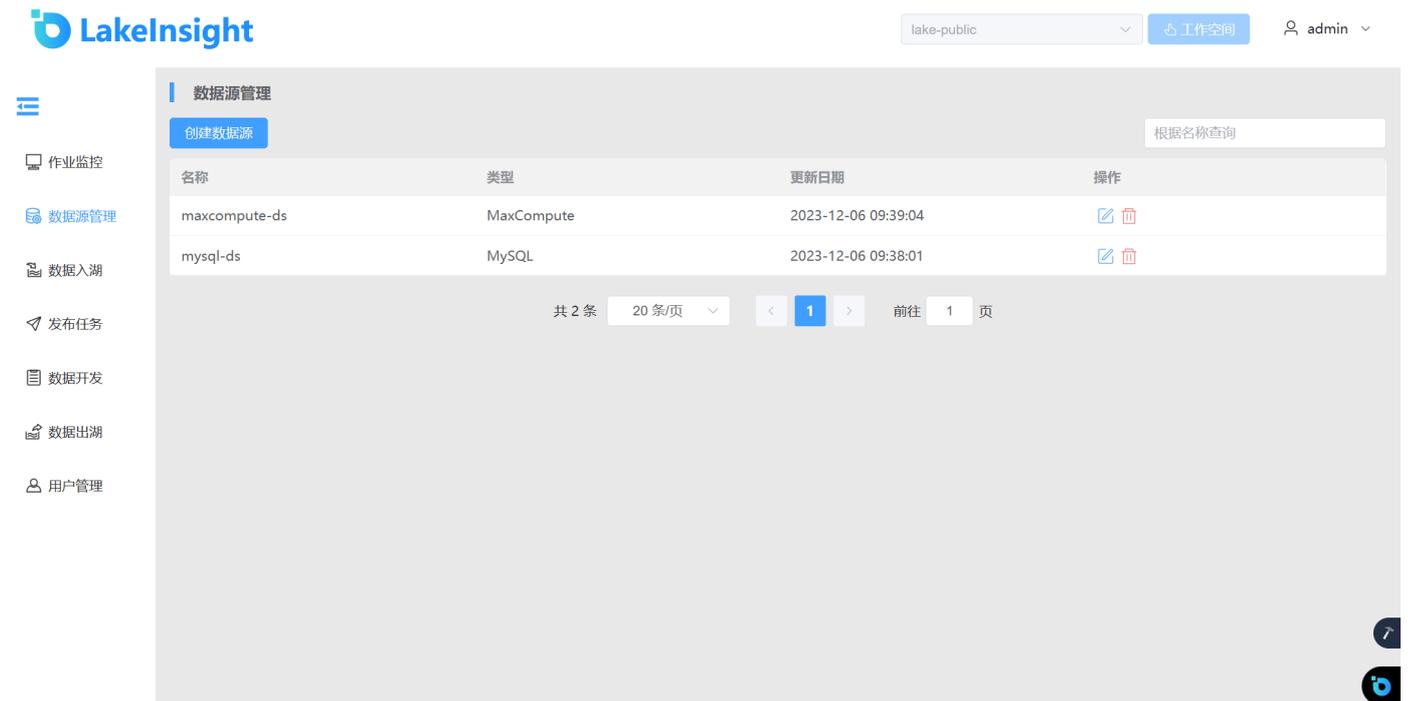
点击**关于**按钮，可以查看当前平台的版本信息。



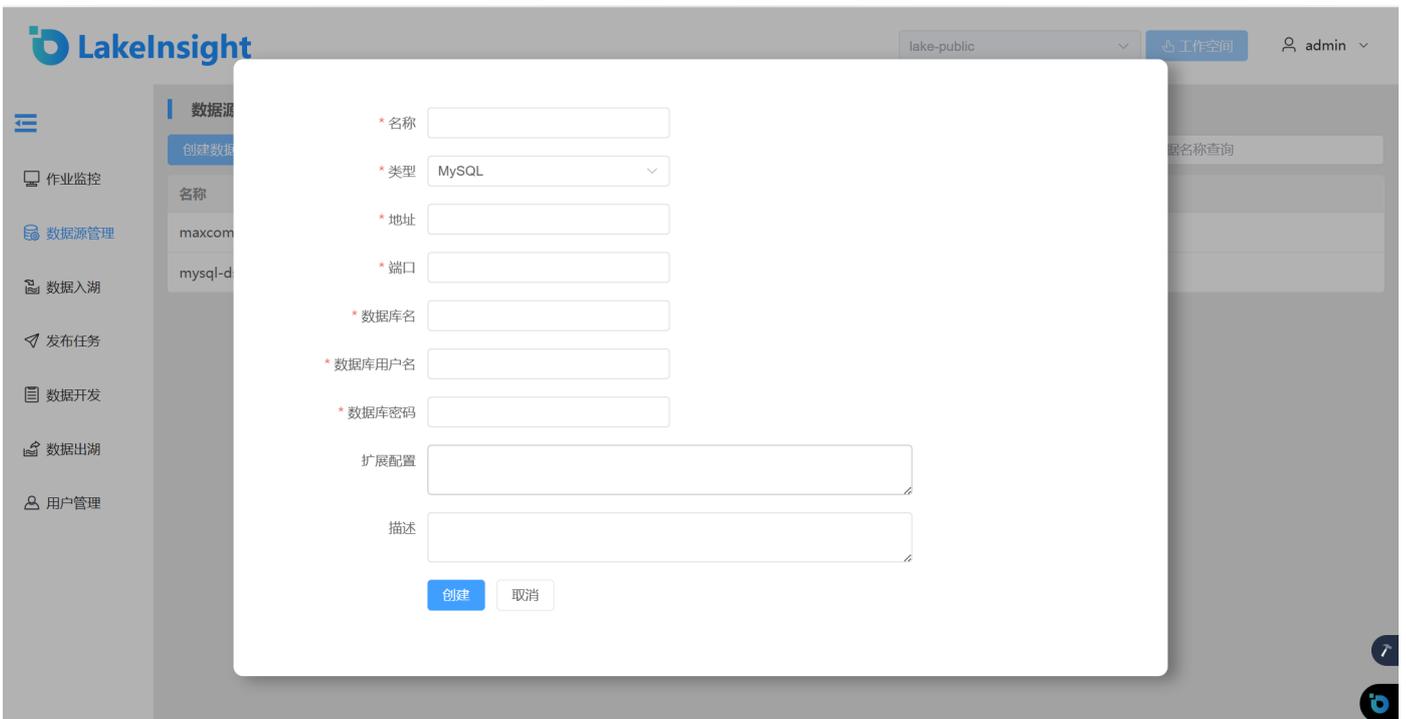
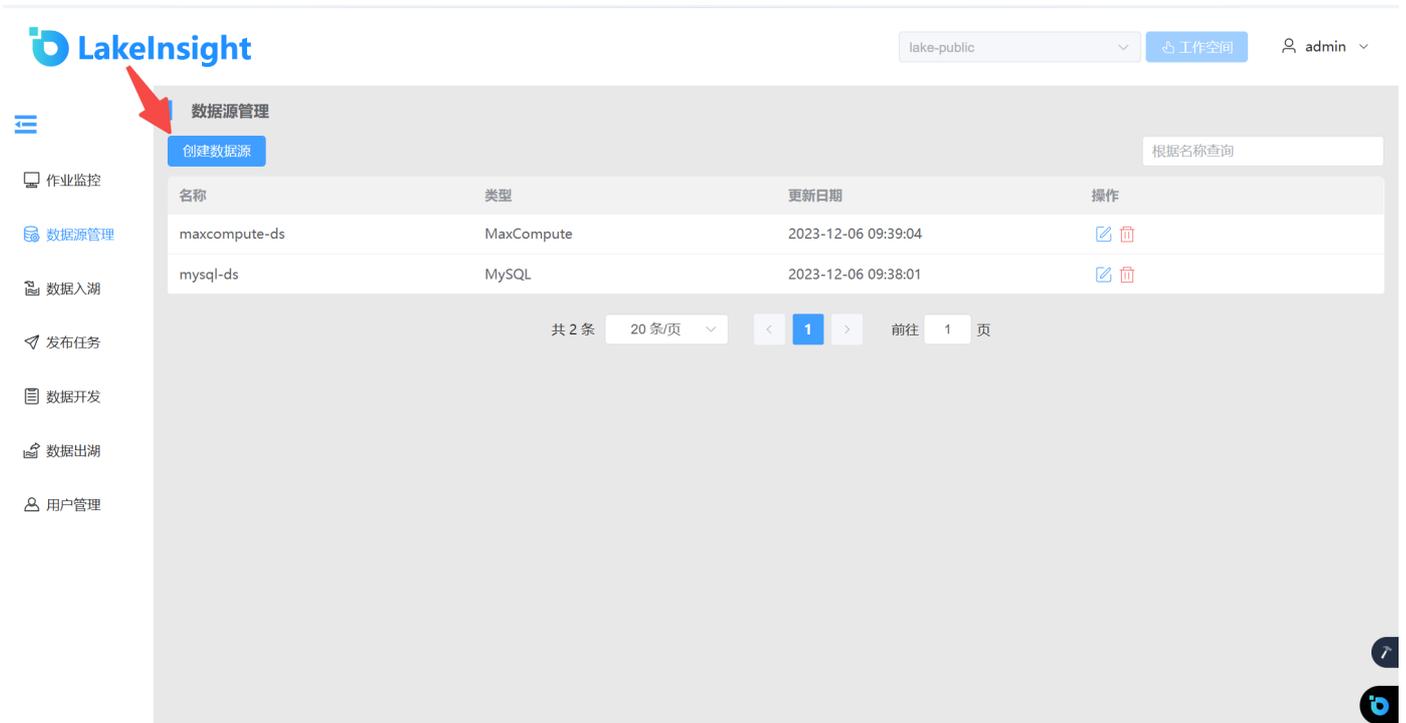
3.4 数据源管理

数据源管理界面，可以配置各种类型的数据源，目前包括Mysql、Kafka以及MaxCompute三类数据源。针对不同的License，创建的数据源实例个数不同。

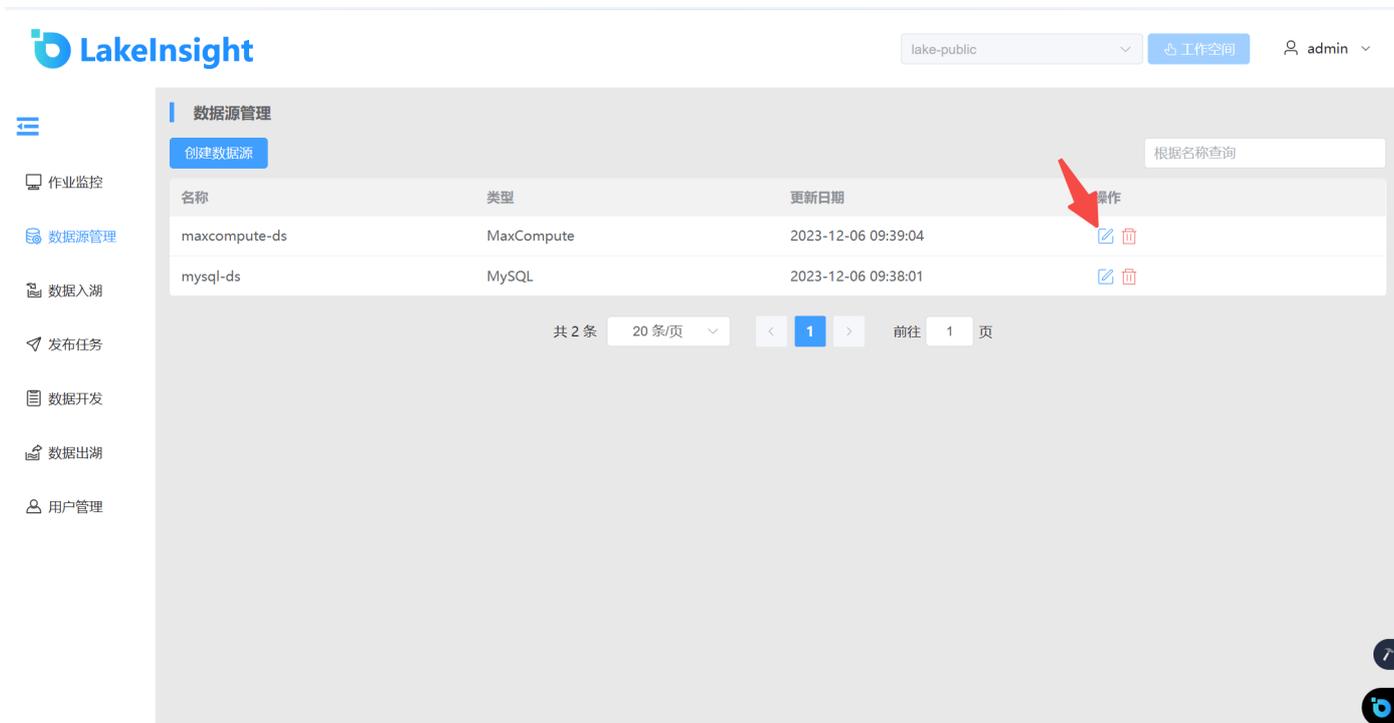
1. 点击左侧**数据源管理**，查看当前已有的数据源列表。



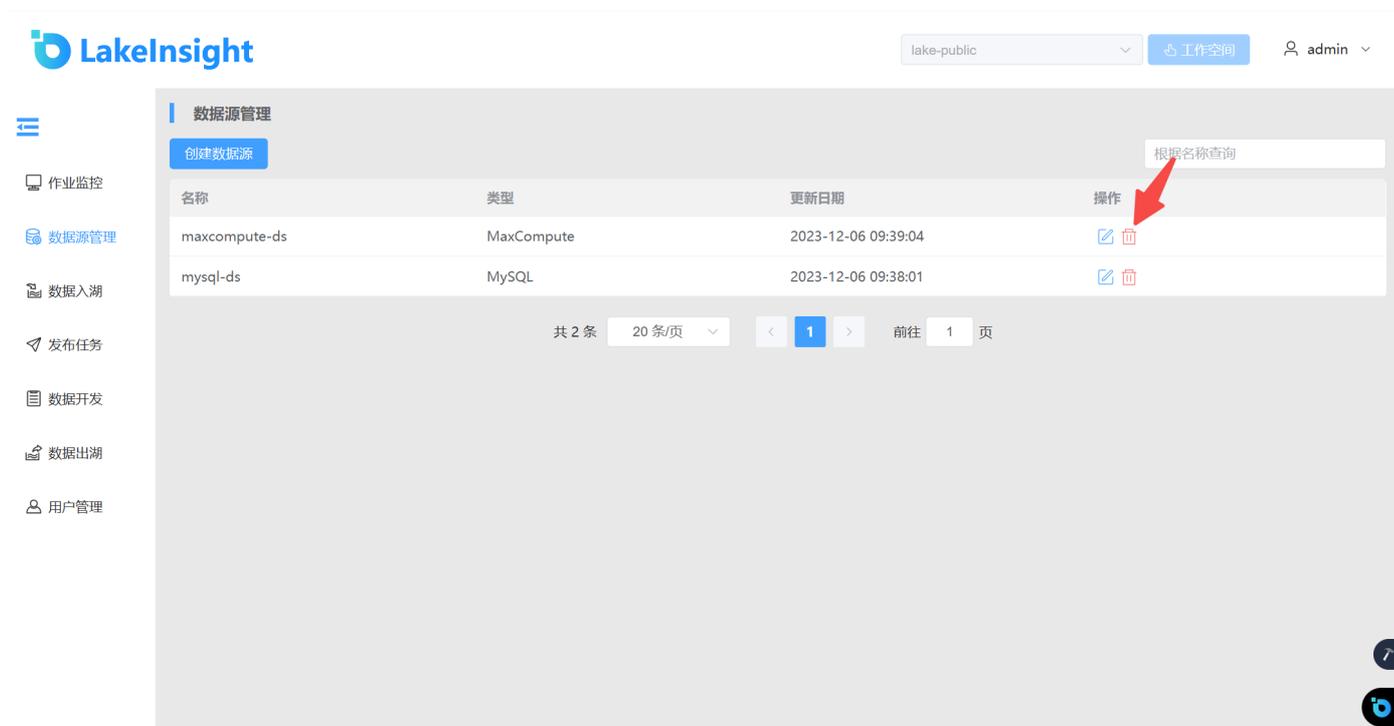
2. 点击**创建数据源**按钮，新增数据源。红色星号标识着必填项。需要输入数据源名称、选择数据源类型（目前支持 MySQL、Kafka和MaxCompute 三种数据类型），输入不同数据源类型所需要的服务地址以及配置相关安全信息。最后点击**创建按钮**即可新增一条数据源。



3. 对创建完成的数据源可以点击**编辑**按钮来进行修改，更新数据源相关信息。

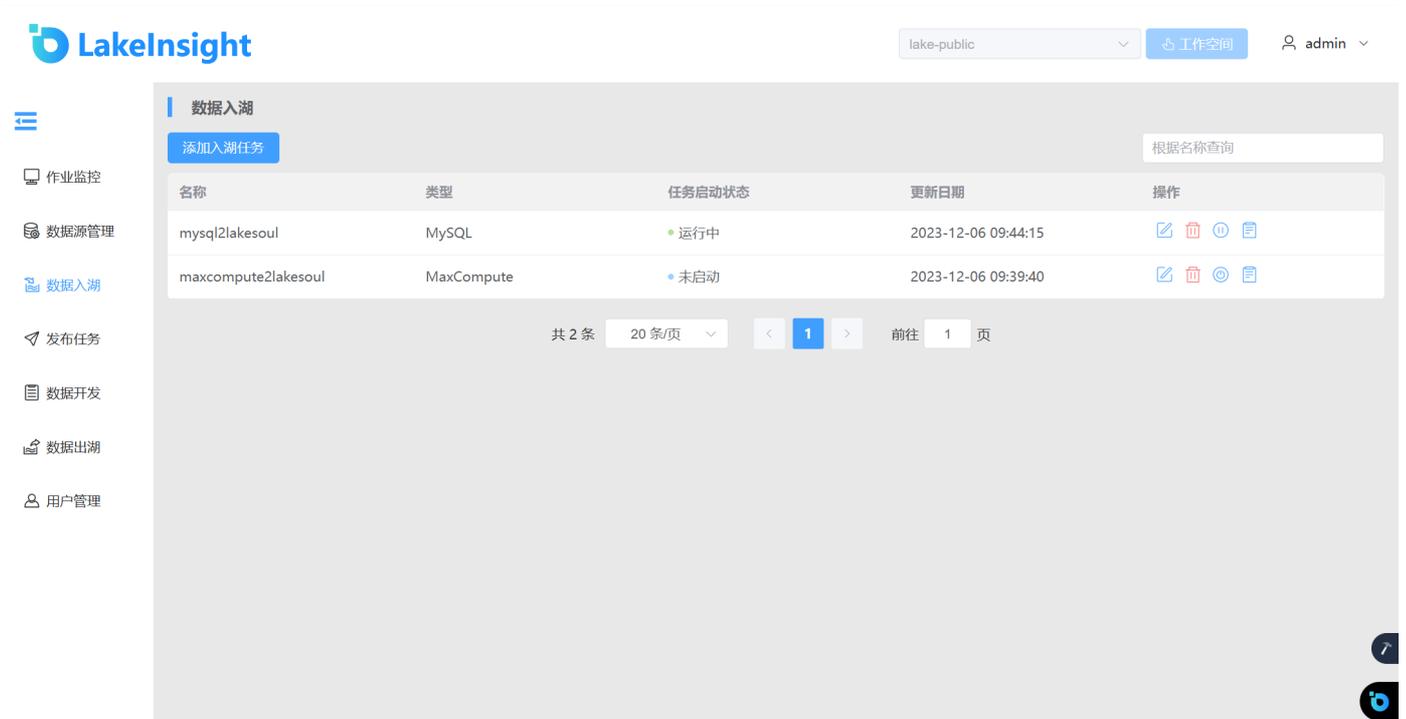


4. 点击删除按钮，可以删除一条数据源。

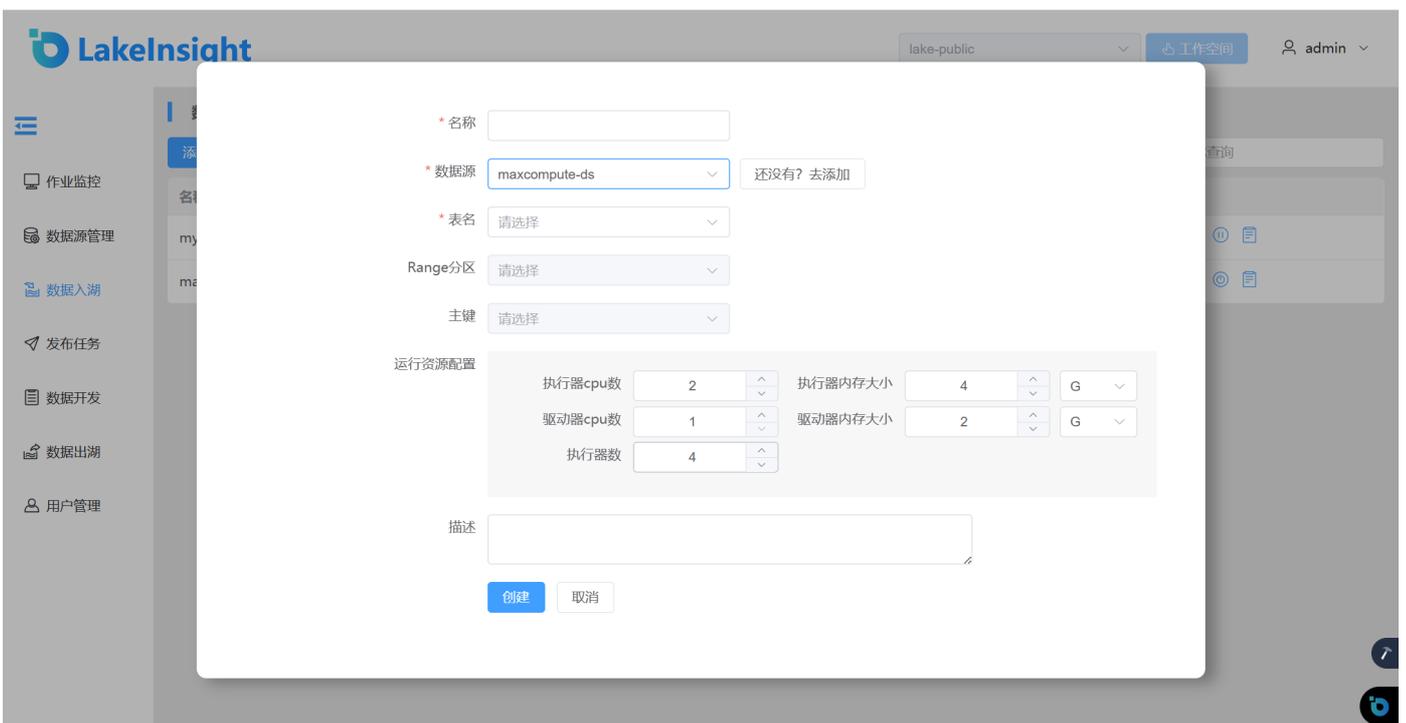


3.5 数据入湖

1. 点击左侧数据入湖，查看当前已有的数据入湖任务列表。



2. 点击**添加入湖任务**按钮，可以创建一个入湖任务。请从上到下依次填写资源信息，分别输入入湖任务名称、选择数据源（不同数据源会显示不同的配置项），以及其他所需的资源信息。点击**创建按钮**即可新增一个入湖任务。



3. 点击**编辑**按钮，可以修改并更新一条数据入湖任务的相关信息（操作同上述数据源管理的编辑操作）。
4. 点击**删除**按钮，可以删除数据入湖任务（操作同上述数据源管理的删除操作）。
5. 点击**启动任务**按钮，即可以启动一个数据入湖任务，可以实时查看任务的运行状态，启动后可以随时停止运行中的任务。

LakeInsight lake-public 工作空间 admin

数据入湖

添加入湖任务

名称 类型 任务启动状态 更新日期 操作

mysql2lakesoul	MySQL	停止	2023-12-06 09:44:15	启动任务
maxcompute2lakesoul	MaxCompute	未启动	2023-12-06 09:39:40	

共 2 条 20 条/页 1 前往 1 页

LakeInsight lake-public 工作空间 admin

数据入湖

添加入湖任务

名称 类型 任务启动状态 更新日期 操作

mysql2lakesoul	MySQL	启动中	2023-12-06 09:44:15	停止任务
maxcompute2lakesoul	MaxCompute	未启动	2023-12-06 09:39:40	

共 2 条 20 条/页 1 前往 1 页

6. 点击**日志**按钮在任务启动成功后，可以查看日志信息。



作业监控

数据源管理

数据入湖

发布任务

数据开发

数据出湖

用户管理

数据入湖

添加入湖任务

根据名称搜索

名称	类型	任务启动状态	更新日期	操作
mysql2lakesoul	MySQL	启动中	2023-12-06 09:44:15	
maxcompute2lakesoul	MaxCompute	未启动	2023-12-06 09:39:40	

共 2 条 20 条/页 < 1 > 前往 1 页

数据入湖

添加入湖任务

名称	类型
maxcompute2lakesoul	MaxCompute
mysql2lakesoul	MySQL

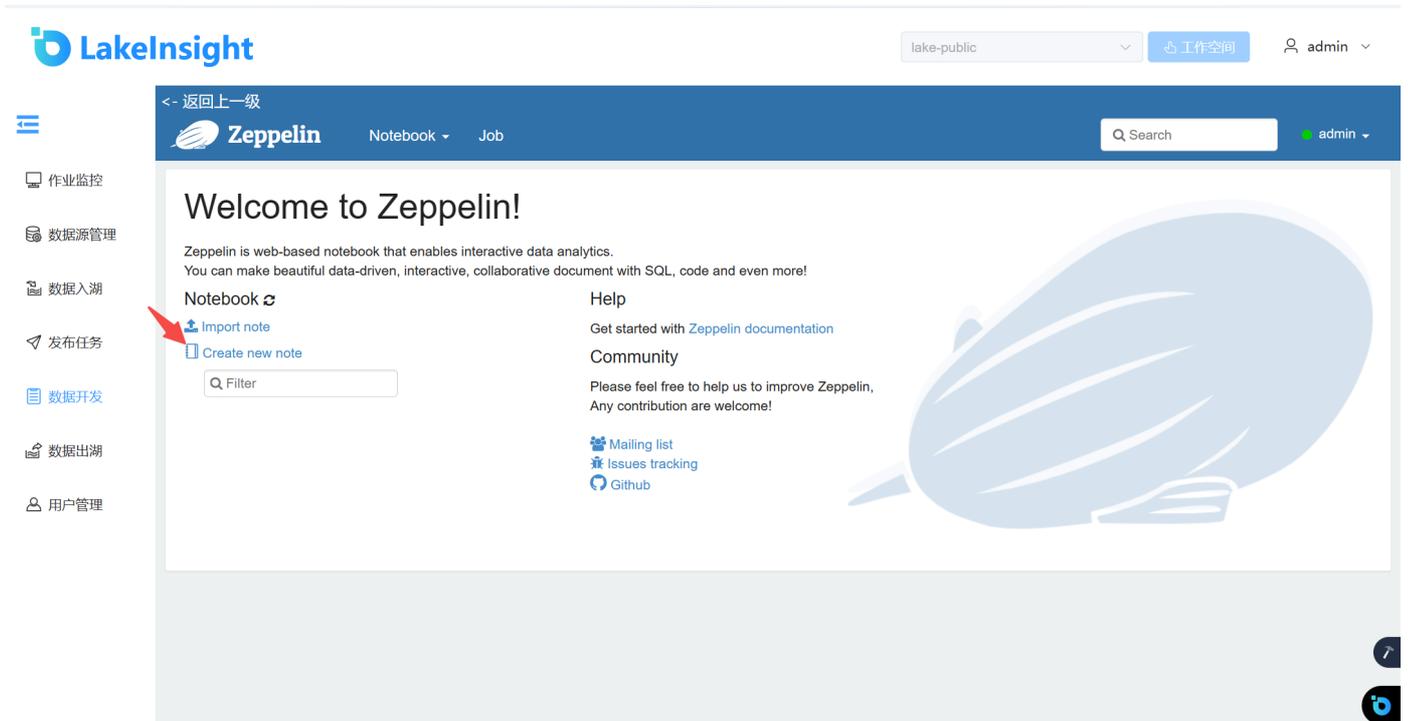
日志

```

++ id -u
+ myuid=185
++ id -g
+ mygid=0
+ set -e
++ getent passwd 185
+ uidentry=
+ set -e
+ '[' -z '' ']'
+ '[' -w /etc/passwd ']'
+ echo '185:x:185:0:anonymous uid:/opt/spark:/bin/false'
+ '[' -z /usr/local/openjdk-11 ']'
+ SPARK_CLASSPATH="/opt/spark/jars/*"
+ env
+ grep SPARK_JAVA_OPTS
+ sed 's/[=]*=\\(.*)/\\1/g'
+ sort -t_ -k4 -n
+ readarray -t SPARK_EXECUTOR_JAVA_OPTS
+ '[' -n '' ']'
+ '[' -z '' ']'
+ '[' -z '' ']'
+ '[' -n '' ']'
+ '[' -z '' ']'
+ '[' -z x ']'
+ SPARK_CLASSPATH="/opt/spark/conf:/opt/spark/jars/*"
+ case "$1" in
+ shift 1
+ CMD=$( { JAVA_HOME } /bin/java "${SPARK_EXECUTOR_JAVA_OPTS[@]}" -Xms$SPARK_EXECUTOR_MEMORY -Xmx$SPARK_EXECUTOR_MEMORY
+ exec /usr/bin/tini -s -- /usr/local/openjdk-11/bin/java -XX:IgnoreUnrecognizedVMOptions --add-opens=java.base/java.
Using Spark's default log4j profile: org/apache/spark/log4j2-defaults.properties
23/12/05 09:22:10 INFO KubernetesExecutorBackend: Started daemon with process name: 200maxcompute2lakesoul-11-73df9f8c
    
```

3.6 数据开发

1. 点击左侧数据开发按钮，进入Zeppelin界面，进行相关Flink SQL任务开发。



详细的 Flink SQL 语法请参考 LakeSoul Flink Connector, Flink SQL 以及 Flink on Zeppelin 文档:

 **数元灵**
D Meta Soul

<https://www.dmetasoul.com/docs/lakesoul/Usage%20Docs/flink-lakeso...>

元灵数智 - 一站式湖仓智能新基建 - 北京数元灵科技

数元灵是一家专注于一站式湖仓智能平台新基建的SaaS公司, 我们的在研项目包括云原生湖仓一体框架, 分布式训练引擎和算法框架, 高性能数据、模型在线服务...

<https://nightlies.apache.org/flink/flink-docs-release-1.17/docs/dev/table/sql/overview/>

SQL

SQL # This page describes the SQL language supported in Flink, including Data Definition Language (DDL), Data Manipulation Language (DML) and Query Language. Flink's SQL support is based on Apache Cal

<https://zeppelin.apache.org/docs/latest/interpreter/flink.html>

zeppelin.apache.org

2. 在 LakeInsight 上使用 Flink SQL 进行实时任务开发的流程:

进入Zeppelin界面, 点击**Create New Note**按钮, 创建一个新的Zeppelin Note进行开发, 亦可导入已有的Zeppelin的文档。下面是一个简单例子, 展示创建Lakesoul Catalog、数据写入以及数据查询的过程。

o 创建catalog

```
1 create catalog lakesoul with('type'='lakesoul');
2 use catalog lakesoul;
3 use `default`;
```

- 作业监控
- 数据源管理
- 数据入湖
- 发布任务
- 数据开发
- 数据出湖
- 用户管理

返回上一级

Zeppelin Notebook Job

flink-sql-dev-02

```

%flink.sql
create catalog lakesoul with('type'='lakesoul');
use catalog lakesoul;
use `default`;
Catalog has been created.
    
```

```

%flink.sql
use `default`;
create table if not exists test_table (`id` INT, name STRING, score INT, `date` STRING,region STRING, PRIMARY KEY (`id`,`name`) NOT ENFORCED ) PARTITIONED BY (`region`,`date`) WITH ( 'connector'='lakeSoul', 'use_cdc'='true','format'='lakesoul', 'path'='s3://lakesoul-bucket/sink/test', 'hashBucketNum'='4');
Table has been created.
    
```

```

%flink.sql
use `default`;
SET 'table.local-time-zone' = 'Asia/Shanghai';
insert into `lakesoul`.`default`.test_table values (1,'AAA', 100, '2023-05-11', 'China');
insert into `lakesoul`.`default`.test_table values (2,'BBB', 100, '2023-05-11', 'China');
insert into `lakesoul`.`default`.test_table values (3,'AAA', 98, '2023-05-10', 'China');
Insertion successfully.
Insertion successfully.
Insertion successfully.
    
```

```

%flink.sql
use `default`;
select * from `lakesoul`.`default`.test_table;
    
```

id	name	score	date	region
1	AAA	100	2023-05-11	China
2	BBB	100	2023-05-11	China
3	AAA	98	2023-05-10	China

创建table

- 1 use `default`;
- 2 create table if not exists test_table (`id` INT, name STRING, score INT,`date` STRING,region STRING, PRIMARY KEY (`id`,`name`) NOT ENFORCED) PARTITIONED BY (`region`,`date`) WITH ('connector'='lakeSoul', 'use_cdc'='true', 'format'='lakesoul', 'path'='s3://lakesoul-bucket/sink/test', 'hashBucketNum'='4');

- 作业监控
- 数据源管理
- 数据入湖
- 发布任务
- 数据开发
- 数据出湖
- 用户管理

返回上一级

Zeppelin Notebook Job

flink-sql-dev-02

```

%flink.sql
use `default`;
create table if not exists test_table (`id` INT, name STRING, score INT, `date` STRING,region STRING, PRIMARY KEY (`id`,`name`) NOT ENFORCED ) PARTITIONED BY (`region`,`date`) WITH ( 'connector'='lakeSoul', 'use_cdc'='true','format'='lakesoul', 'path'='s3://lakesoul-bucket/sink/test', 'hashBucketNum'='4');
Table has been created.
    
```

```

%flink.sql
use `default`;
SET 'table.local-time-zone' = 'Asia/Shanghai';
insert into `lakesoul`.`default`.test_table values (1,'AAA', 100, '2023-05-11', 'China');
insert into `lakesoul`.`default`.test_table values (2,'BBB', 100, '2023-05-11', 'China');
insert into `lakesoul`.`default`.test_table values (3,'AAA', 98, '2023-05-10', 'China');
Insertion successfully.
Insertion successfully.
Insertion successfully.
    
```

```

%flink.sql
use `default`;
select * from `lakesoul`.`default`.test_table;
    
```

id	name	score	date	region
1	AAA	100	2023-05-11	China
2	BBB	100	2023-05-11	China
3	AAA	98	2023-05-10	China

插入数据

```
1 use `default`;
2 SET 'table.local-time-zone' = 'Asia/Shanghai';
3 insert into `lakesoul`.`default`.test_table values (1,'AAA', 100, '2023-05-11', 'China');
4 insert into `lakesoul`.`default`.test_table values (2,'BBB', 100, '2023-05-11', 'China');
5 insert into `lakesoul`.`default`.test_table values (3,'AAA', 98, '2023-05-10', 'China');
```

The screenshot shows the Lakelnsight web interface. The top navigation bar includes the Lakelnsight logo, a dropdown menu for 'lake-public', a '去工作空间' button, and a user profile 'lakeinsight-user-01'. The left sidebar contains navigation icons for '作业监控', '数据源管理', '数据入湖', '发布任务', '数据开发', '数据出湖', and '用户管理'. The main content area displays a Flink job execution log. The log shows the following SQL commands:

```
%flink.bsq
use `default`;
create table if not exists test_table ('id' INT, name STRING, score INT, 'date' STRING, region STRING, PRIMARY KEY ('id','name') NOT ENFORCED ) PARTITIONED BY ('region','date') WITH ( 'connector'='lakeSoul', 'use_cdc'='true', 'format'='lakesoul', 'path'='s3://lakesoul-bucket/sink/test', 'hashBucketNum'=4);
Table has been created.
Took 9 sec. Last updated by lakeinsight-user-01 at December 06 2023, 6:52:43 PM.
```

The log then shows the execution of the insert statements, which completed successfully:

```
%flink.bsq
use `default`;
SET 'table.local-time-zone' = 'Asia/Shanghai';
insert into `lakesoul`.`default`.test_table values (1,'AAA', 100, '2023-05-11', 'China');
insert into `lakesoul`.`default`.test_table values (2,'BBB', 100, '2023-05-11', 'China');
insert into `lakesoul`.`default`.test_table values (3,'AAA', 98, '2023-05-10', 'China');
Insertion successfully.
Insertion successfully.
Insertion successfully.
Took 1 min 29 sec. Last updated by lakeinsight-user-01 at December 06 2023, 7:05:24 PM.
```

Below the log, a table displays the data inserted into the 'test_table' table:

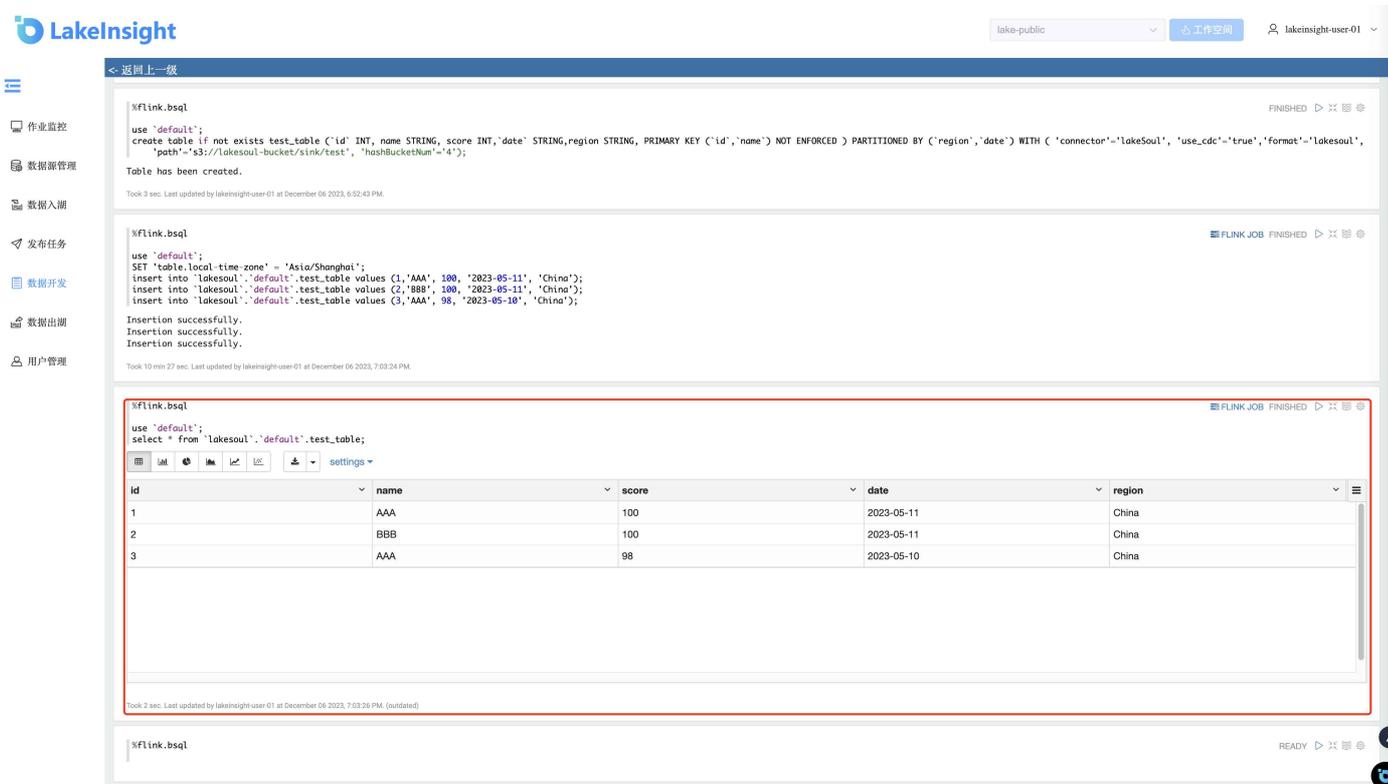
id	name	score	date	region
1	AAA	100	2023-05-11	China
2	BBB	100	2023-05-11	China
3	AAA	98	2023-05-10	China

The bottom of the screenshot shows another Flink job execution log with the following SQL commands:

```
%flink.bsq
use `default`;
select * from `lakesoul`.`default`.test_table;
```

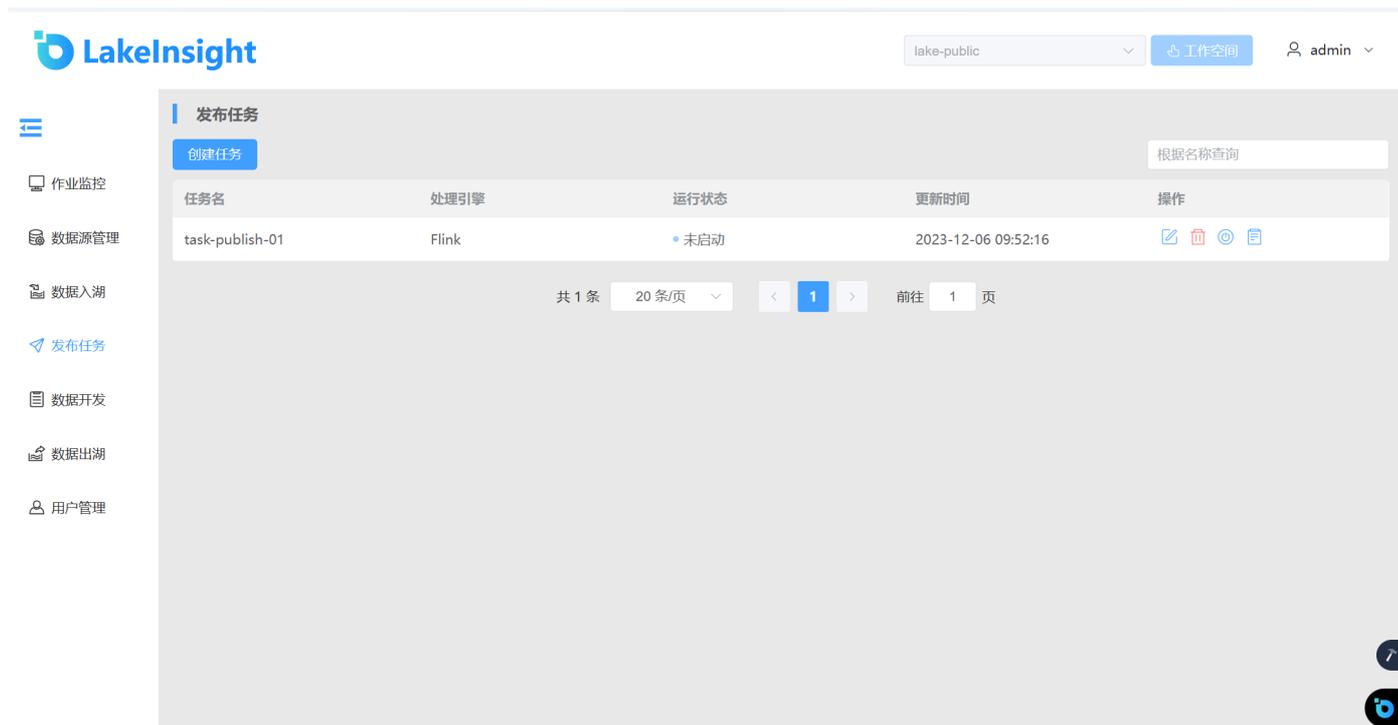
数据查询

```
1 use `default`;
2 select * from `lakesoul`.`default`.test_table;
```



3.7 任务发布

1. 点击左侧**发布任务**，查看当前已有的发布任务。

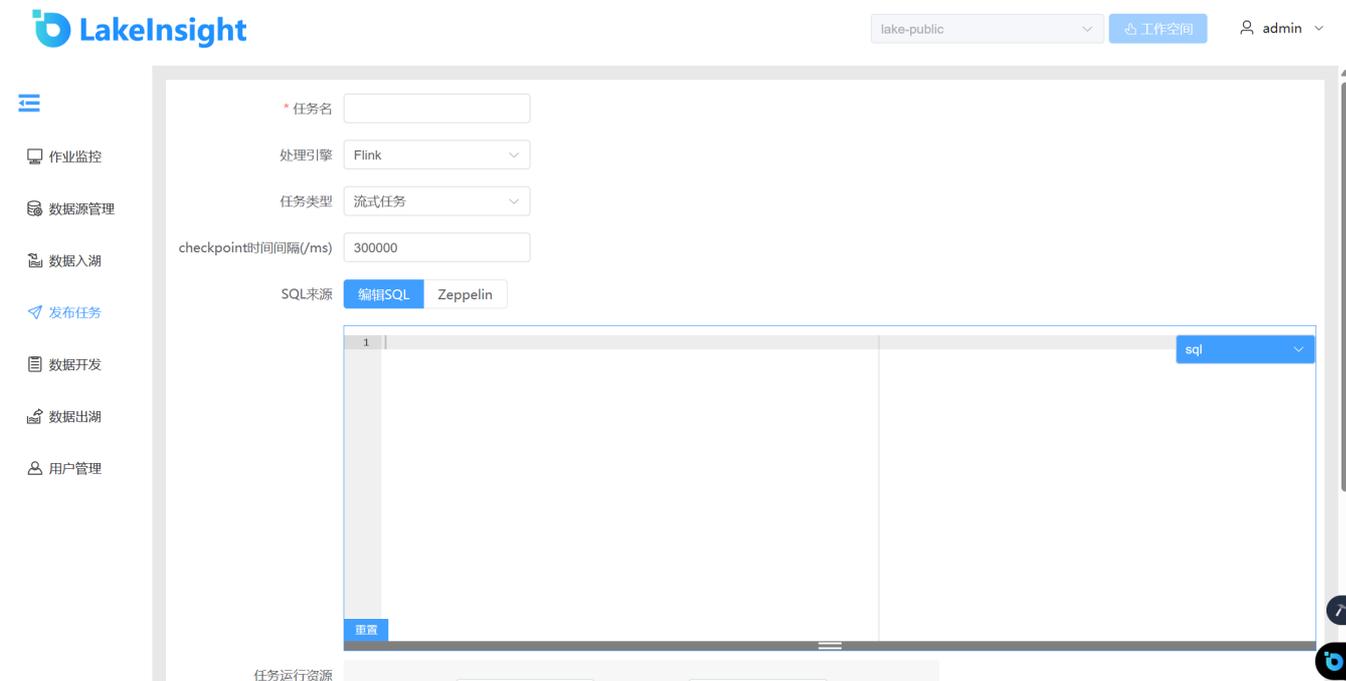


2. 点击**创建任务**按钮，弹出创建发布任务页面，依次输入任务名称、选择任务处理引擎、...选择sql语句来源(自助SQL或者来自zeppelin)、任务运行资源以及描述信息，输入完成后，点击**发布**按钮，即可完成一个发布任务。

a. SQL语句来源包括：自助SQL以及zeppelin

如果选择**编辑SQL**，可以在面板中直接写入Flink SQL语句

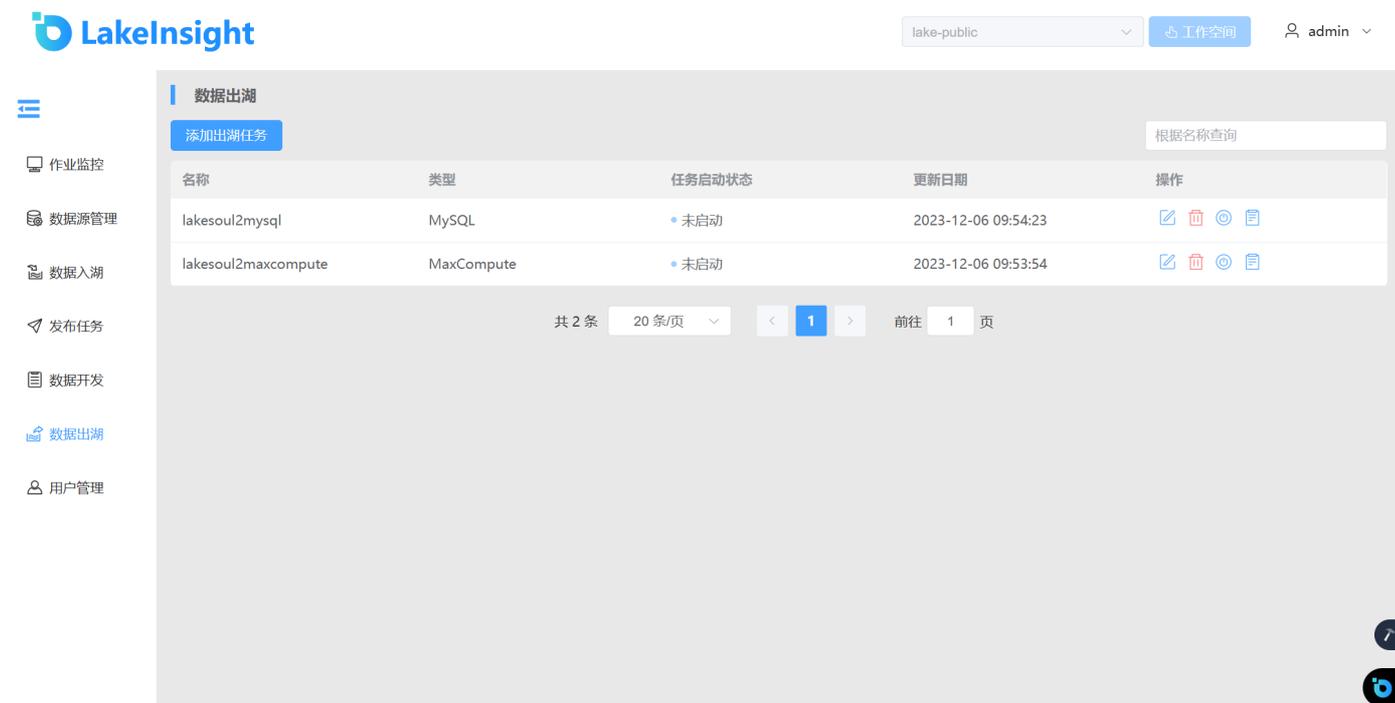
如果选择**Zeppelin**，可以从已有的zeppelin中选择已开发的Flink SQL任务（即上述数据开发所创建的数据）。选择后，数据信息会自动导入下面面板中



3. 点击**编辑**按钮，可以修改并更新任务发布相关信息（操作同上述数据源管理的编辑操作）。
4. 点击**删除**按钮，可以删除一条已发布的任务（操作同上述数据源管理的删除操作）。
5. 点击**启动任务**按钮，即可以启动一个发布任务，可以实时查看任务的运行状态（操作同上述数据入湖的启动/停止操作）。
6. 任务启动成功后，可点击**日志**按钮在，查看日志信息（操作同上述数据入湖的查看日志操作）。

3.8 数据出湖

1. 点击左侧**数据出湖**，查看当前已有的数据出湖任务列表。



2. 点击**添加数据出湖**按钮，弹出数据出湖任务页面。依次输入出湖任务名称、选择出湖数据包括数据库名和表名；选中导出到数据库信息包括选择数据源（不同类型的数据源对应的配置信息不同），设置表名库名等信息，设置任务运行资源以及描述信息。输入完成后，点击**创建**按钮，即可创建一个数据出湖任务。

The screenshot shows the 'Add Data Outflow Task' form in the Laka system. The form is divided into several sections:

- Outflow Task Name:** A text input field with a red asterisk indicating it is required.
- Outflow Data:** A section containing two dropdown menus: 'Database Name' and 'Table Name', both with red asterisks.
- Export to:** A section containing a 'Data Source' dropdown menu with a '还没有? 去添加' link, and a 'Table Name' dropdown menu with a red asterisk. There are also buttons for 'Create New Table' and 'Select Existing Table'.
- Run Resource Configuration:** A section containing two dropdown menus for 'source parallelism' and 'sink parallelism', both set to '4'. It also has a 'checkpoint storage interval (ms)' input field set to '300000' and a 'Database Time Zone' dropdown menu set to 'UTC'.
- Description:** A text input field at the bottom.

The interface includes a sidebar with navigation options: 作业监控, 数据源管理, 数据入湖, 发布任务, 数据开发, 数据出湖, and 用户管理. The user 'admin' is logged in.

3. 点击**编辑**按钮，可以修改并更新数据出湖相关信息（操作同上述数据源管理的编辑操作）。
4. 点击**删除**按钮，可以删除一条数据出湖任务（操作同上述数据源管理的删除操作）。
5. 点击**启动任务**按钮，即可以启动一个数据入湖任务，可以实时查看任务的运行状态（操作同上述数据入湖的启动/停止操作）。
6. 在任务启动成功后，点击**日志**按钮，可以查看日志信息（操作同上述数据入湖的查看日志操作）。

3.9 作业监控

1. 点击左侧**作业监控**按钮。
 - a. 选择**作业统计**，查看当前各模块任务数及运行状态信息，可通过右侧筛选出某段时间内的相关信息。

The screenshot shows the LakeInsight dashboard. At the top left is the LakeInsight logo. The top right shows the user 'lake-public' and 'admin'. The main navigation menu on the left includes: 作业统计, 作业监控, 数据源管理, 数据入湖, 发布任务, 数据开发, 数据出湖, and 用户管理. The '作业统计' (Job Statistics) tab is active, indicated by a red arrow. The dashboard content is divided into two main sections. The top section, titled '任务总数' (Total Tasks), contains three cards: '数据源' (Data Sources) with a count of 2, '数据入湖' (Data Ingestion) with a count of 2, and '发布任务' (Released Tasks) with a count of 1. Each card has a '查看详情' (View Details) link. The bottom section, titled '模块任务状态信息' (Module Task Status Information), features a donut chart showing the distribution of task states: 运行中 (Running) in blue, 运行完成 (Completed) in green, 未运行 (Not Running) in yellow, and 运行失败 (Failed) in red. A legend above the chart identifies these colors. On the right side of the dashboard, there are two input fields for '开始时间' (Start Time) and '结束时间' (End Time), with the end time set to '2023-12-06 18:51:27'.

b. 选择资源监控，查看当前任务运行的状态。

The screenshot shows the LakeInsight dashboard with the '资源监控' (Resource Monitoring) tab selected, indicated by a red arrow. The top navigation and user information are the same as in the previous screenshot. The main navigation menu on the left is also the same. The '资源监控' (Resource Monitoring) tab is active. The dashboard content is divided into two main sections. The top section, titled 'Dashboards', contains a search bar, a 'Filter by tag' dropdown, a 'Starred' checkbox, and a 'Sort' dropdown. Below this is a table of dashboards. The table has two columns: 'Name' and 'Tags'. The 'Name' column contains two entries: 'Flink Running App Monitor' and 'Spark Running App Monitor'. The 'Tags' column contains four tags for each entry: 'apache', 'flink', 'job manager', and 'task manager'. The bottom section, titled 'Playlists', is currently empty.



作业监控

数据源管理

数据入湖

发布任务

数据开发

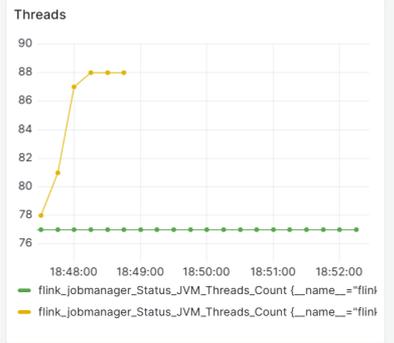
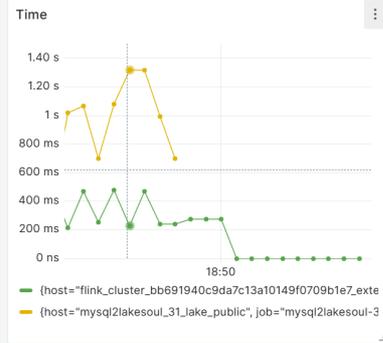
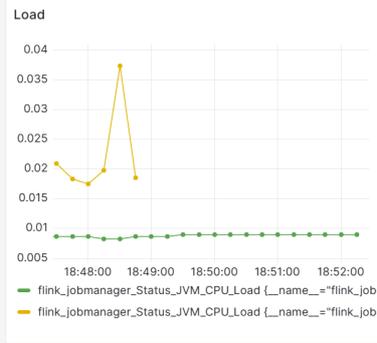
数据出湖

用户管理

作业统计 资源监控

Source Prometheus Job All Task Manager All Job Manager All Job Name All Task Name All Ad Hoc +

Job Manager (JVM - CPU)



Job Manager (JVM - Memory Usage)

