

恒生电子股份有限公司

LightDB Enterprise Manager

用户手册

恒生研究院

2022 年 10 月

文档修改记录

版本	修订人	修订说明	批准人	发布日期
1.0.0.0		初稿		20220801
1.0.0.1		v22.3 发版更新		20221018

说 明

本文档中所包含的信息属于商业机密信息，如无恒生电子股份有限公司的书面许可，任何人都无权复制或利用。

模板版本信息

编辑部门：EPG

批准日期：2018/9/26

目录

目 录	3
1 引言	5
1.1 编写目的	5
1.2 预期读者	5
1.3 参考文档	5
2 LightDB 及 EM 安装	5
3 LightDB EM 功能介绍	6
3.1 登录页面	6
3.1 首页	6
3.2 功能模块	7
3.3 Web 命令行工具	8
3.4 执行计划解析器	8
3.5 集群管控	10
3.5.1 集群列表	10
3.5.2 主机管理	14
3.5.3 实例列表	19
3.5.4 全局非空集群列表	22
3.6 主机监控与管理	23
3.6.1 主机概览	23
3.7 数据库监控与管理	31
3.7.1 PWR 管理	31
3.7.2 数据库概览	33
3.7.3 实时 SQL	48
3.7.4 TOP SQL	50
3.7.1 TOP 对象	51
3.8 集中备份	51
3.8.1 备份服务器	51
3.8.2 备份定时任务	52
3.8.3 备份列表	55
3.8.4 备份删除	57
3.8.5 恢复列表	57
3.9 日志检索	59
3.9.1 数据库日志	59
3.9.2 系统日志	65
3.9.3 Itcluster 日志	66
3.9.4 HA 日志	66
3.9.5 监控代理日志	67
3.10 通知中心	67
3.10.1 告警通知	67
3.11 系统管理	69

3.11.1	EM 定时任务	69
3.11.2	图表参数设置	70
3.11.3	采集监控定时任务	71
3.11.4	数据库定时任务.....	73
3.12	告警待办	74
3.13	用户中心	75
3.13.1	修改信息.....	75
3.13.2	修改密码.....	76
3.13.3	锁屏	77
3.13.4	主题皮肤.....	77
3.13.5	系统信息.....	78
3.13.6	设置密保问题	78
3.13.7	在线帮助.....	79
3.13.8	安全退出.....	79
4	LightDB EM 注意事项	79
4.1	Lightdb EM 集群相关.....	79
4.2	22.3 版本升级内容	79

1 引言

1.1 编写目的

本文档为恒生电子股份有限公司 LightDB Enterprise Manager 用户手册说明书，本文档主要阐述 LightDB Enterprise Manager 的详细功能介绍，完整的数据库功能请参考《LightDB 用户手册》。LightDB Enterprise Manager（即 LightDB 数据库监控管理平台，下文均简称为 LightDB EM）是一个综合性的数据库监控和管理系统，旨在满足数据库用户的需求，提供强大的图形界面，简化了对 LightDB 数据库的维护和使用。

1.2 预期读者

本文档主要适用于 LightDB 数据库的：

- 数据库管理员
- 开发工程师
- 测试工程师
- 技术支持工程师

1.3 参考文档

《LightDB 数据库安装手册》

《LightDB EM Installer 手册》

2 LightDB 及 EM 安装

安装完整的 LightDB 数据库请参考《LightDB 数据库安装手册》；

安装完整的 LightDB EM 请参考《LightDB EM Installer 手册》。

3 LightDB EM 功能介绍

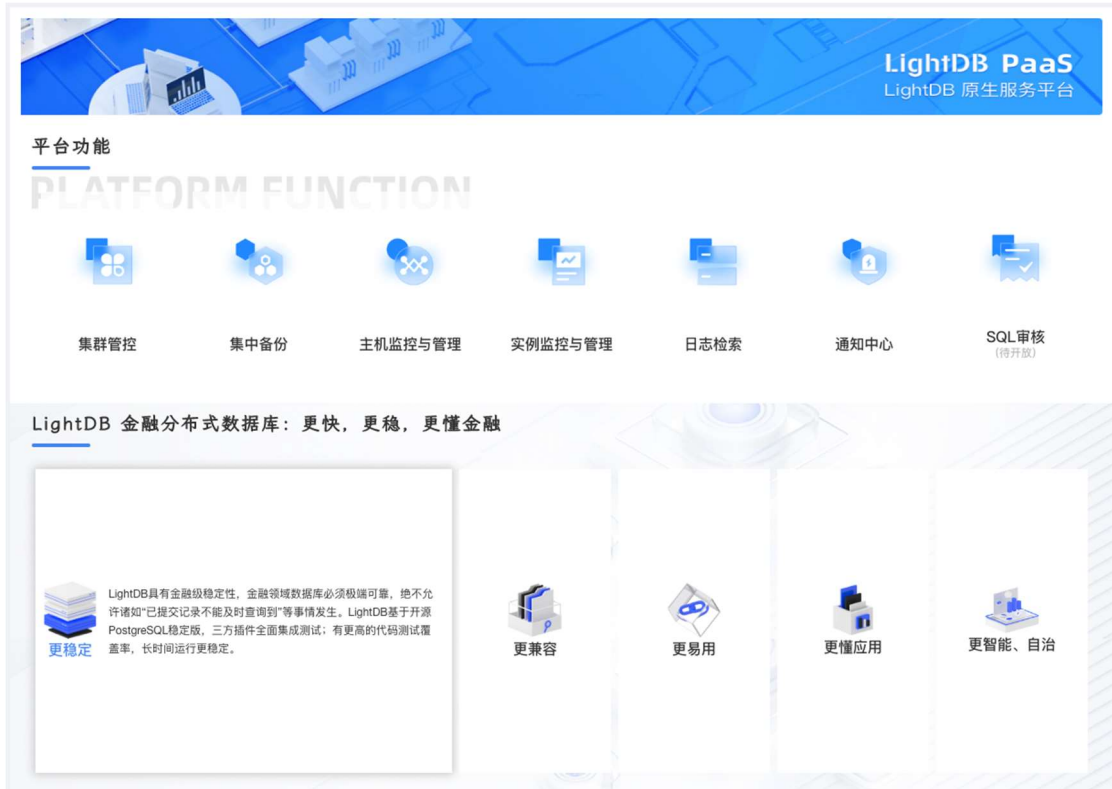
3.1 登录页面

LightDB EM 默认登录地址 http://your_em_server_ip:17333/em/login.html。首次成功安装后，登录用户名是 `system`，密码是 `hs123456`。首次登录会提示要修改密码，密码修改后再登录就直接进入首页。



3.1 首页

登录成功后进入首页。首页为平台介绍页，包含各种平台功能及其描述信息。



3.2 功能模块

功能模块可以展示系统所有功能清单，同时展示用户最新操作的功能，可以根据功能名称关键字进行查询，通过点击对应的功能可以进入功能的详细页面。

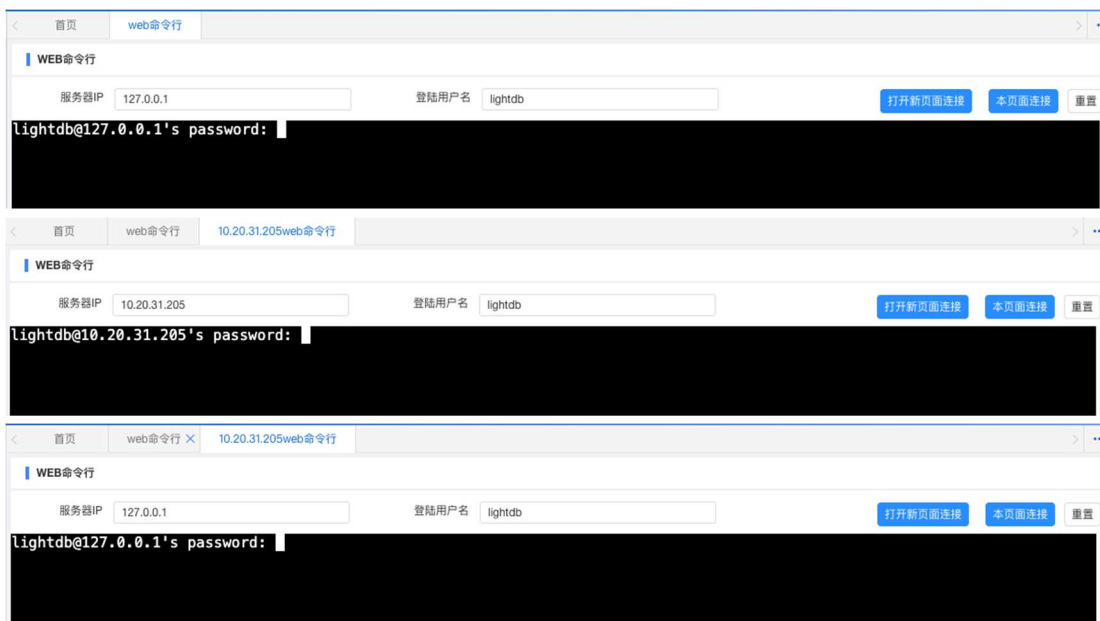


3.3 Web 命令行工具

Web 命令行提供一个 web 版本的命令行页面,可在此页面通过 ssh 连接到其他服务器进行操作。

初次打开会登陆到 em 所在服务器的命令行页面,需要输入对应账户的密码。

可以指定目标服务器 ip 和用户名,可以通过点击【打开新页面连接】打开新的连接,或者在本页面进行连接,点击重置按钮,本页面将连接重置到 127.0.0.1。



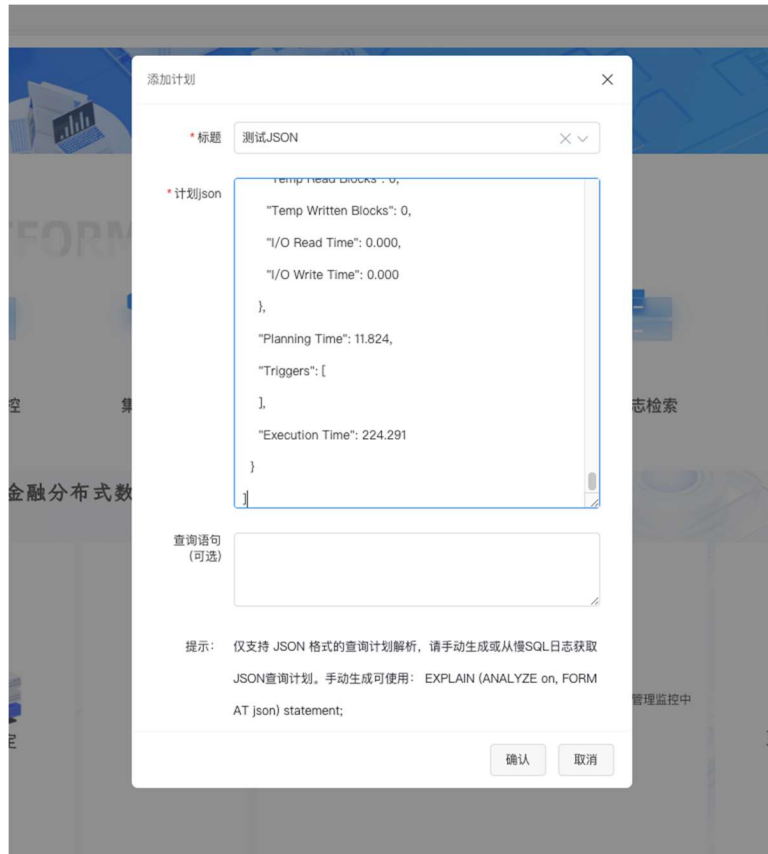
3.4 执行计划解析器

执行计划解析器提供在页面顶部第一位,点击后可打开添加计划窗口。



输入所需的标题和计划 json, 查询语句(可选填)后, 点击确定即可打开对应的执行计

划解析器。



执行计划解析器页面存在三个 tab 页, 分别是 html 表格解析结果, source 解析原格式, query 查询语句, html 页面中包含中间的表格数据和下方的 planing time、execution time 数据。

执行计划解析器

HTML SOURCE

序列	exclusive	inclusive	rows x	rows	loops	node
1	0.008	222.266	↑173000.0	1	1	WindowAgg (const=84855.41..2082520.98 rows=173000 width=144) (actual time=222.228..222.266 rows=1 loops=1) Output:row_number() OVER (?:t.fleet_id,a.service_provider_id,f.account_id,p.name,a.name,f.name,t.topic,t.trigger,(f.enabled AND a.enabled AND p.enabled)),to_json(array_agg(json_build_object('id',t.id,'user_id',t.user_id,'grant_id',g.grant_id,'user',u.name,'email_address',lower(trim(COALESCE(NULLIF((u.auth0_data->>'email':text),''):text),(u.email_address):text))), 'authorised',(g.grant_id IS NOT NULL),'enabled',(u.factory OR (u.enabled AND COALESCE(p_1.enabled,true) AND COALESCE((p_3.enabled AND a_2.enabled),true) AND COALESCE((p_2.enabled AND a_1.enabled AND f_1.enabled),true))), 'email_verified',COALESCE(CASE WHEN COALESCE(NOT COALESCE((u.auth0_id ~ '^auth0[0-9a-f]{24}\$::citext) AND ((u.auth0_id):text = (u.auth0_data->>'user_id':text))),false),true) THEN NULL:boolean ELSE COALESCE((u.auth0_data->>'email_verified':text)::boolean,false) END,false),'scope',g.scope,'tenant',g.tenant,'media',t.media,'create_time_utc',t.create_time_utc,'modified_time_utc',t.modified_time_utc,'uid',t.uid))Shared Hit Blocks:7783,Shared Read Blocks:0,Shared Dirty Blocks:0,Shared Written Blocks:0,Local Hit Blocks:0,Local Read Blocks:0,Local Dirty Blocks:0,Local Written Blocks:0,Temp Read Blocks:207,Temp Written Blocks:207,I/O Read Time:0,I/O Write Time:0
2	0.045	222.258	↑173000.0	1	1	Aggregate (const=84855.41..2079925.98 rows=173000 width=136) (actual time=222.221..222.258 rows=1 loops=1) Strategy:Sorted,Partial Mode:Simple,Output:t.fleet_id,a.service_provider_id,f.account_id,p.name,a.name,f.name,t.topic,t.trigger,(f.enabled AND a.enabled AND p.enabled),array_agg(json_build_object('id',t.id,'user_id',t.user_id,'grant_id',g.grant_id,'user',u.name,'email_address',lower(trim(COALESCE(NULLIF((u.auth0_data->>'email':text),''):text),(u.email_address):text))), 'authorised',(g.grant_id IS NOT NULL),'enabled',(u.factory OR (u.enabled AND COALESCE(p_1.enabled,true) AND COALESCE((p_3.enabled AND a_2.enabled),true) AND COALESCE((p_2.enabled AND a_1.enabled AND f_1.enabled),true))), 'email_verified',COALESCE(CASE WHEN COALESCE(NOT COALESCE((u.auth0_id ~ '^auth0[0-9a-f]{24}\$::citext) AND ((u.auth0_id):text = (u.auth0_data->>'user_id':text))),false),true) THEN NULL:boolean ELSE COALESCE((u.auth0_data->>'email_verified':text)::boolean,false) END,false),'scope',g.scope,'tenant',g.tenant,'media',t.media,'create_time_utc',t.create_time_utc,'modified_time_utc',t.modified_time_utc,'uid',t.uid))Group Key:t.fleet_id,a.service_provider_id,f.account_id,p.name,a.name,f.name,t.topic,t.trigger,(f.enabled AND a.enabled AND p.enabled))Shared Hit Blocks:7783,Shared Read Blocks:0,Shared Dirty Blocks:0,Shared Written Blocks:0,Local Hit Blocks:0,Local Read Blocks:0,Local Dirty Blocks:0,Local Written Blocks:0,Temp Read Blocks:207,I/O Read Time:0,I/O Write Time:0
3	0.023	222.213	↑173000.0	1	1	Incremental Sort (const=84855.41..2066880.98 rows=173000 width=1004) (actual time=222.177..222.213 rows=1 loops=1) Output:t.fleet_id,a.service_provider_id,f.account_id,p.name,a.name,f.name,t.topic,t.trigger,(f.enabled AND a.enabled AND p.enabled),t.id,t.user_id,g.grant_id,u.name,u.auth0_data,u.email_address,u.factory,u.enabled,p_1.enabled,(p_3.enabled AND a_2.enabled),(p_2.enabled AND a_1.enabled AND f_1.enabled),u.auth0_id,g.scope,g.tenant,t.media,t.create_time_utc,t.modified_time_utc,t.uid,Sort Key:a.service_provider_id,f.account_id,p.name,a.name,f.name,t.topic,t.trigger,(f.enabled AND a.enabled AND p.enabled)),Presorted Key:a.service_provider_id,Full-sort Groups:[object Object],Shared Hit Blocks:7783,Shared Read Blocks:0,Shared Dirty Blocks:0,Shared Written Blocks:0,Local Hit Blocks:0,Local Read Blocks:0,Local Dirty Blocks:0,Local Written Blocks:0,Temp Read Blocks:207,Temp Written Blocks:207,I/O Read Time:0,I/O Write Time:0
4	0.006	222.19	↑173000.0	1	1	Nested Loop (const=16529.68..1934882.3 rows=173000 width=1004) (actual time=208.24..222.19 rows=1 loops=1) Join Type:Left,Output:t.fleet_id,a.service_provider_id,f.account_id,p.name,a.name,f.name,t.topic,t.trigger,(f.enabled AND a.enabled AND p.enabled),t.id,t.user_id,g.grant_id,u.name,u.auth0_data,u.email_address,u.factory,u.enabled,p_1.enabled,(p_3.enabled AND a_2.enabled),(p_2.enabled AND f_1.enabled),u.auth0_id,g.scope,g.tenant,t.media,t.create_time_utc,t.modified_time_utc,t.uid,Inner Unique:false,Shared Hit Blocks:7783,Shared Read Blocks:0,Shared Dirty Blocks:0,Shared Written Blocks:0,Local Hit Blocks:0,Local Read Blocks:0,Local Dirty Blocks:0,Local Written Blocks:0,Temp Read Blocks:207,Temp Written Blocks:207,I/O Read Time:0,I/O Write Time:0

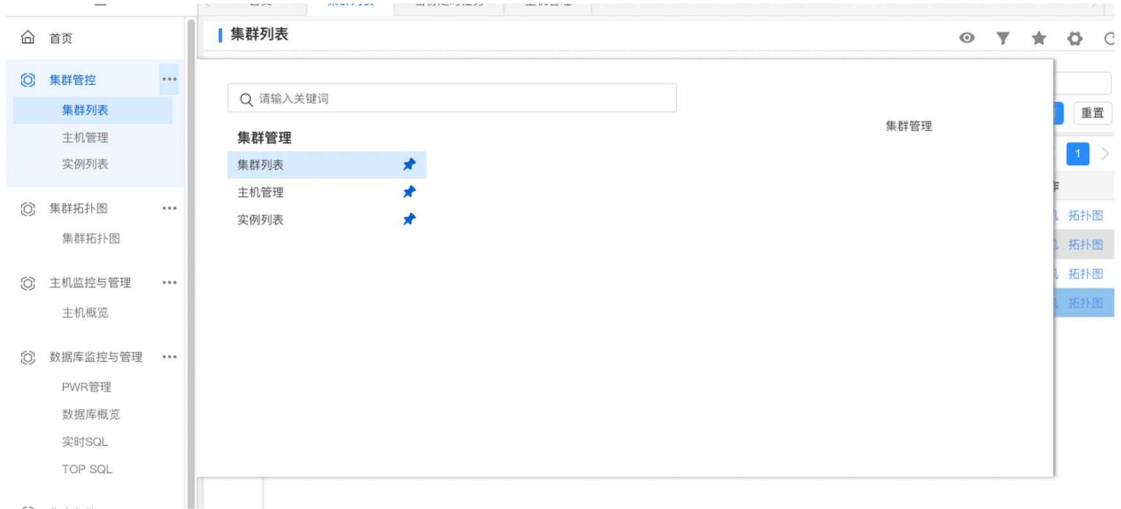
Planning time : 11.824 ms
Execution time :224.291 ms

执行计划解析器 html 中存在 7 列数据，分别是序列号，exclusive 独占时间、inclusive 所有时间，rowsex 行数 x，rows 行数，loops 循环数，node 节点信息。前四个节点有四种背景颜色，依次为红色，棕色，黄色，白色，分别对应严重到不严重的等级。

其中 exclusive, inclusive 的比例取决当前数值占总时间的比例，依次为 90%/50%/10%，rowsex 依据当前值的大小依次为 1000/100/10，rows 取决的移除数的比例，计算规则为 rows/(rows+remove)，比例依次为 90%/50%/10%。

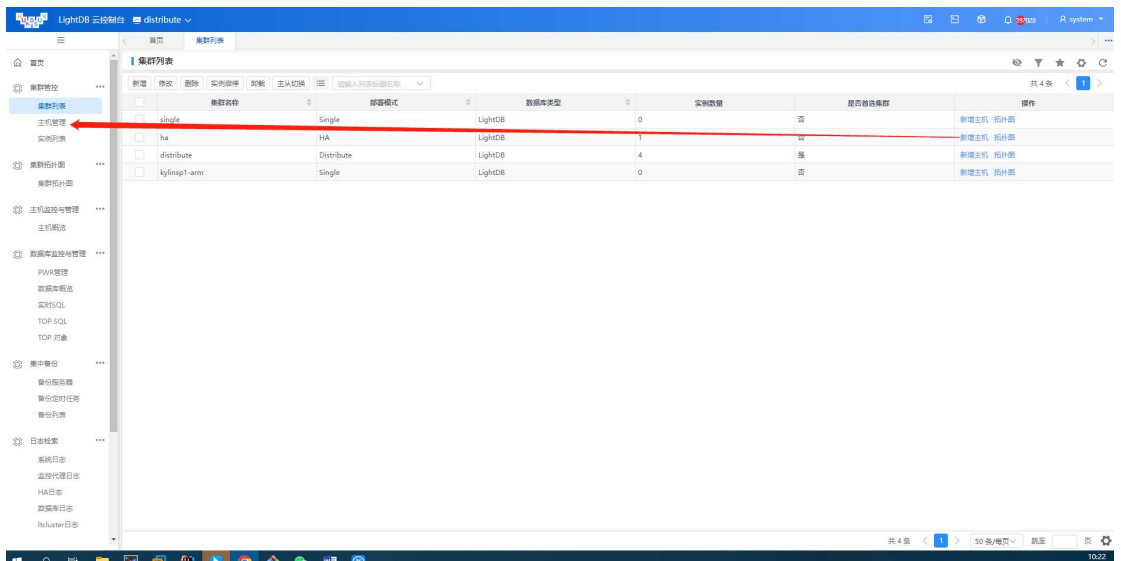
3.5 集群管控

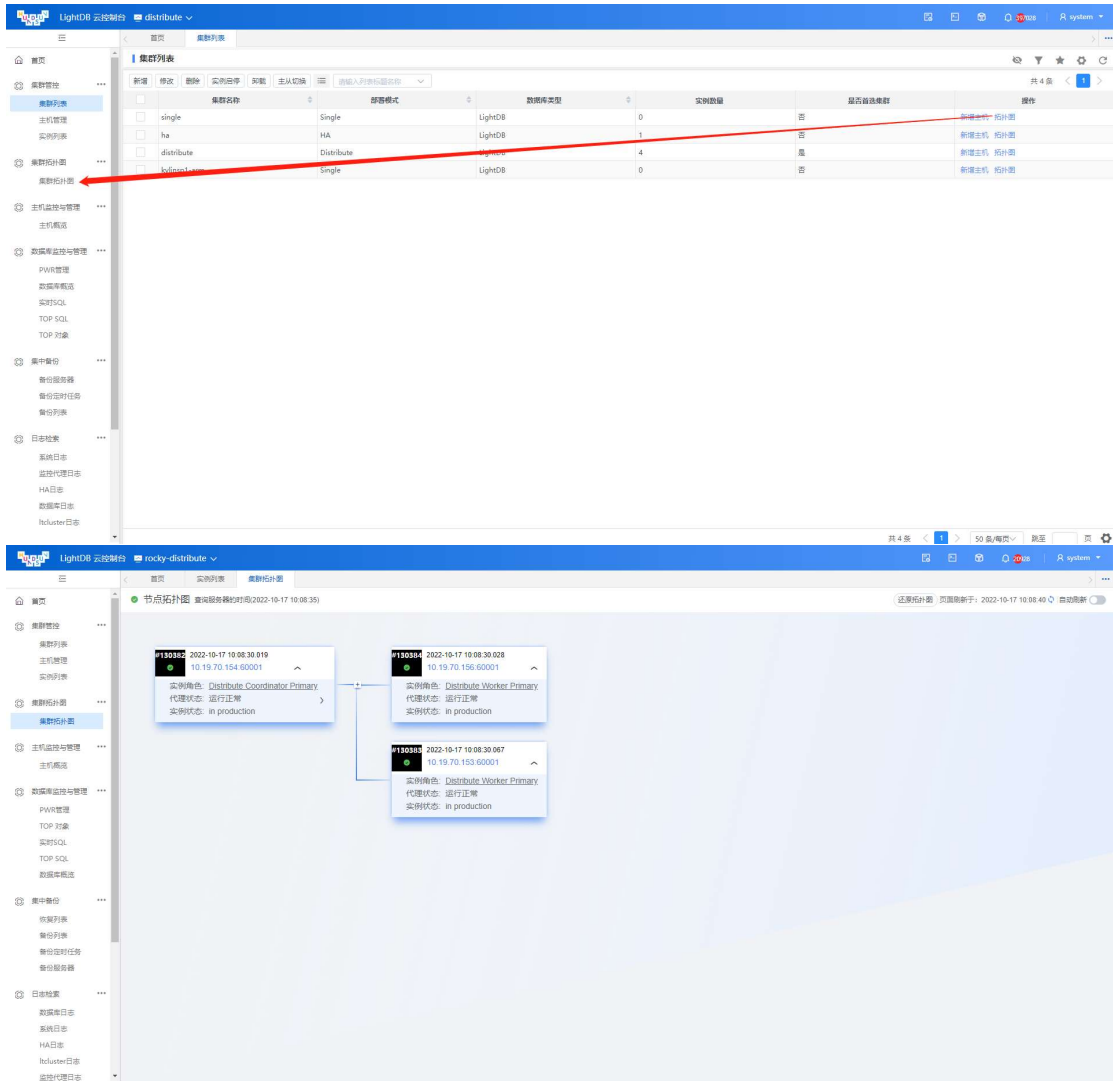
集群管控用于统一管理主机、代理、实例



3.5.1 集群列表

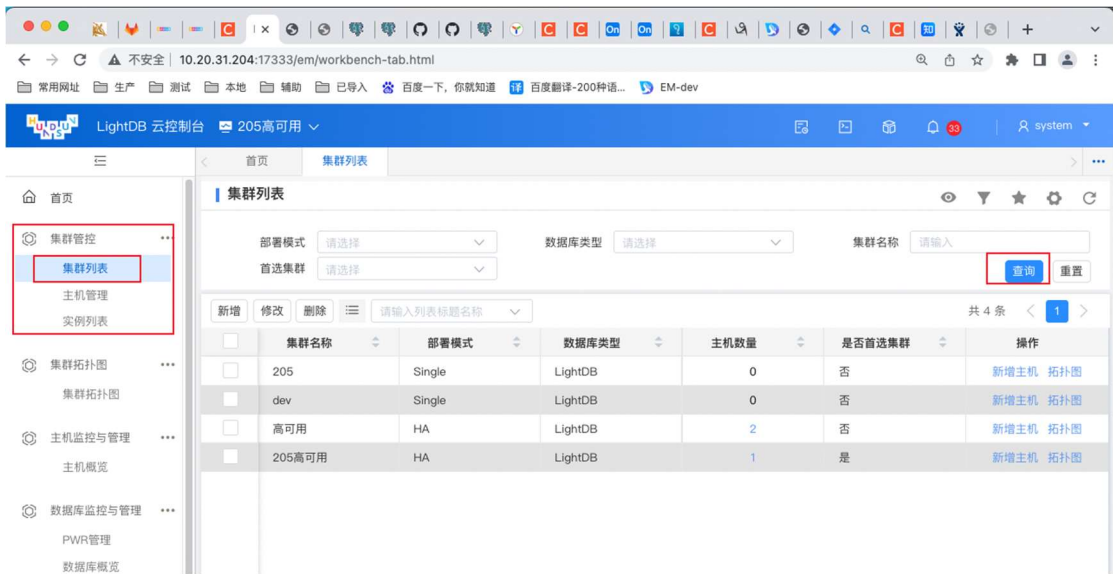
通过添加集群来管控集群下的主机和数据库实例。点击新增主机可跳转至主机管理。点击拓扑图可跳转至集群拓扑图查看集群下主机角色关联关系。





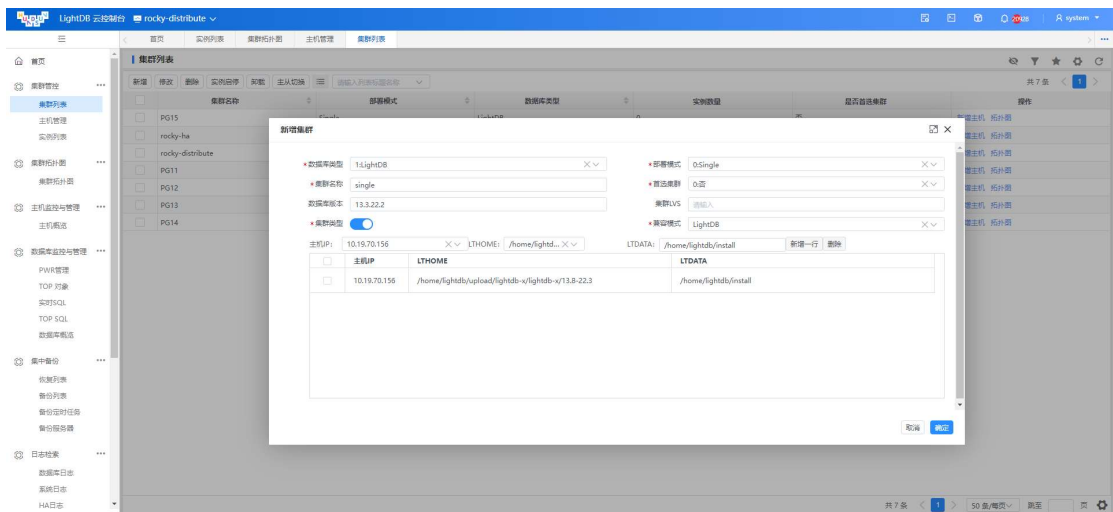
3.5.1.1 查询集群列表

集群列表展示用户添加的集群信息，可以根据部署模式、数据库类型、集群名称、首选集群进行查询，同时提供增删改操作功能。



3.5.1.2 添加集群

用户可以添加要监控的集群，添加成功后可通过添加主机、实例进行关联。



数据库类型：要监控的集群数据库类型包含 LightDB、PostgreSQL 等。

部署模式：Single 单机模式、HA 高可用模式、Distribute 分布式模式。

集群名称：定义部署的集群名称，不可重复。

首选集群：平台默认展示的监控集群。

数据库版本：可不填，由 agent 后台监控并填充。

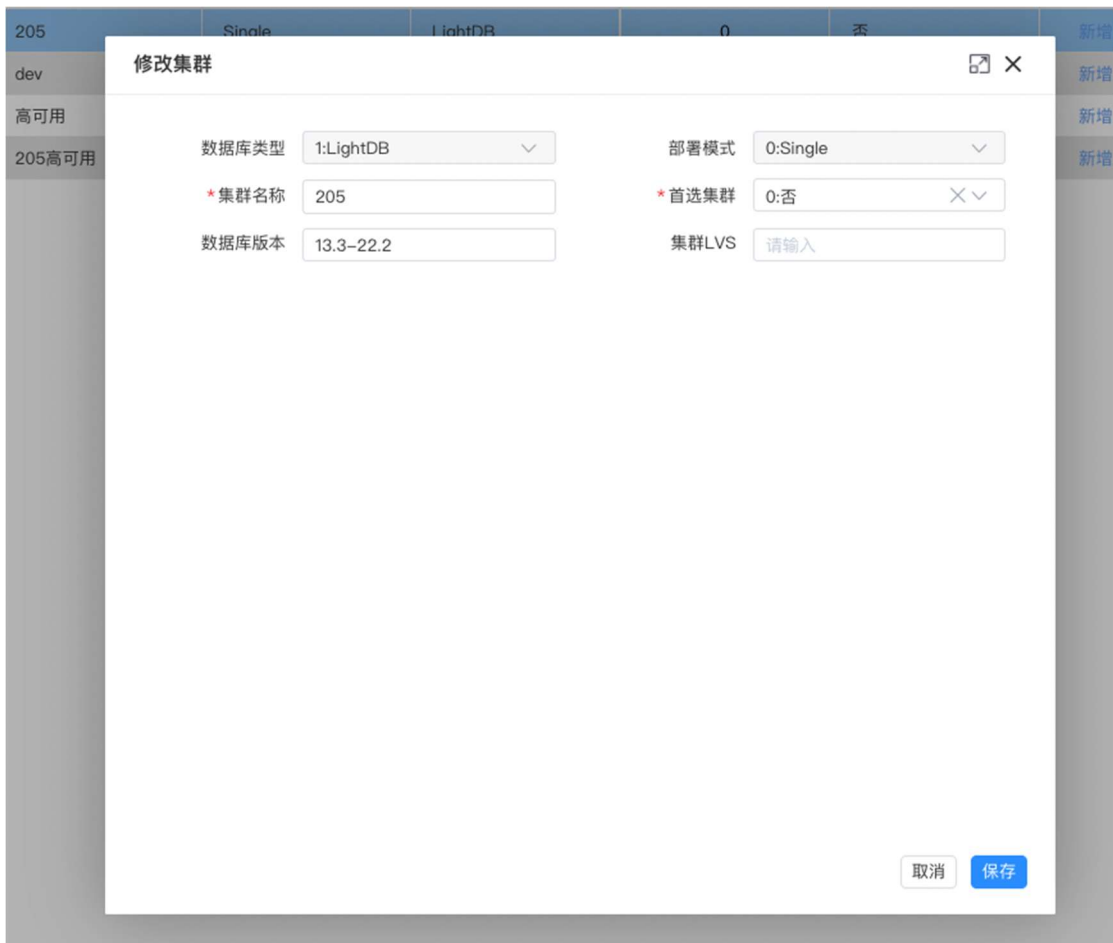
集群 LVS：分布式预留字段，可不填写。

集群类型：打开可在线安装数据库实例。

兼容模式：可选 LightDB、Oracle、MySQL。

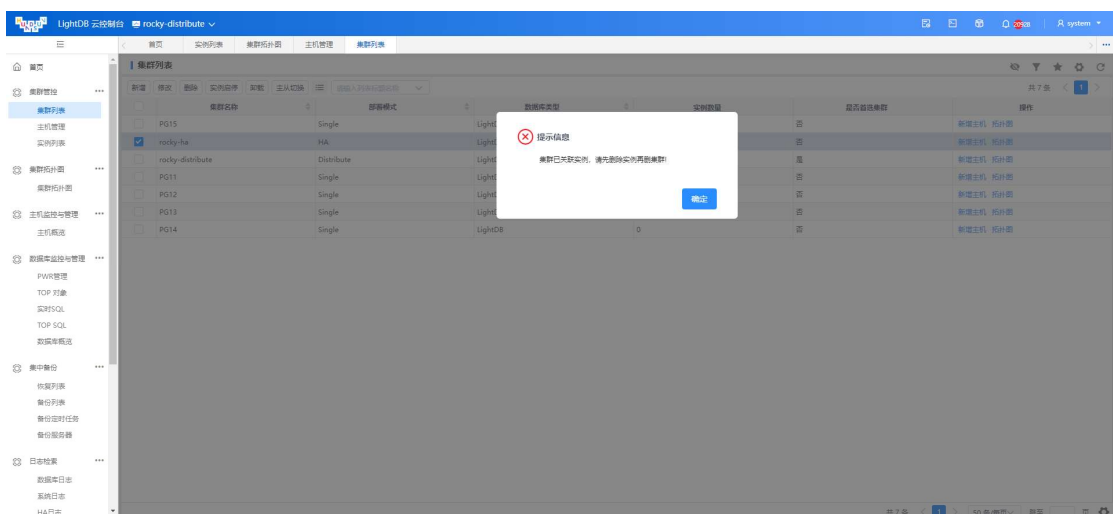
3.5.1.3 修改集群

修改集群不可修改数据库类型、部署模式。其余均可修改。



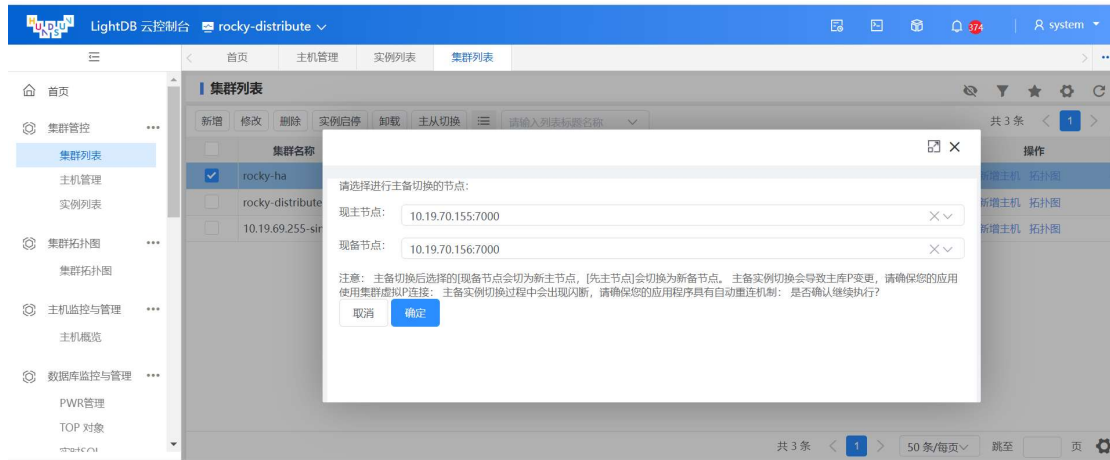
3.5.1.4 删除集群

不想要监控集群可进行删除。删除前必须先删除所监控的数据库实例。



3.5.1.5 主从切换

高可用模式下，主节点能够切换成备节点，备节点切换成主节点。

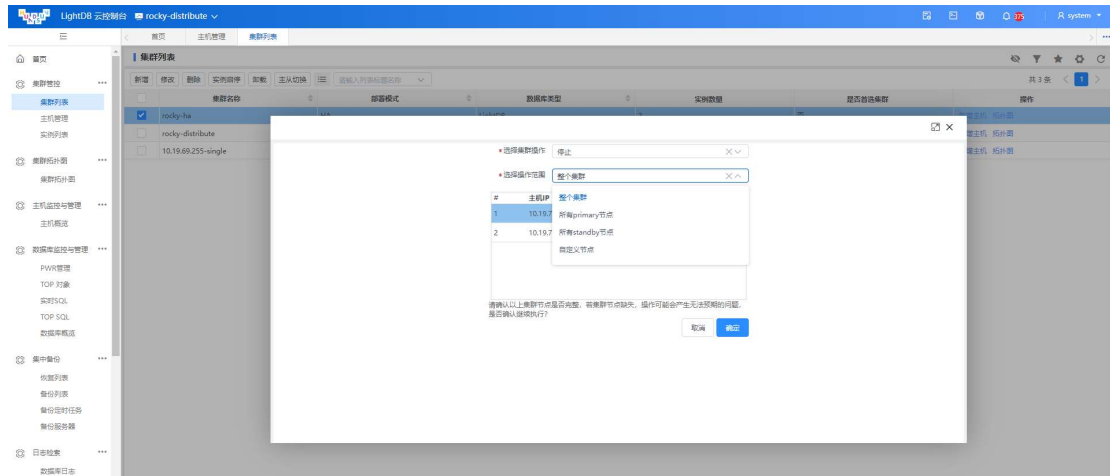


现主节点：现在高可用模式下的主节点 ip。

现备节点：现在高可用模式下的备节点 ip。

3.5.1.6 实例启停

实例启停操作可对整个集群中的节点进行启动，停止，重启操作。

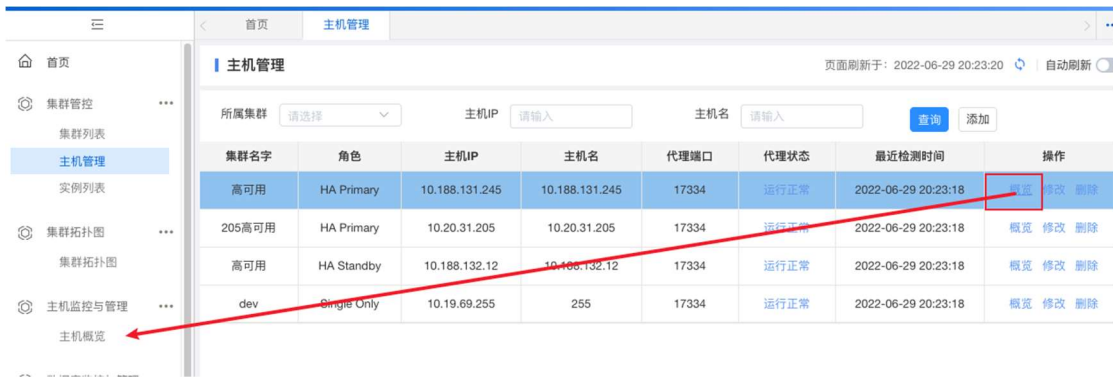


选择集群操作：可选择启动，停止，重启操作。

选择操作范围：可选择整个集群，所有 primary 节点，所有 standby 节点，自定义节点。

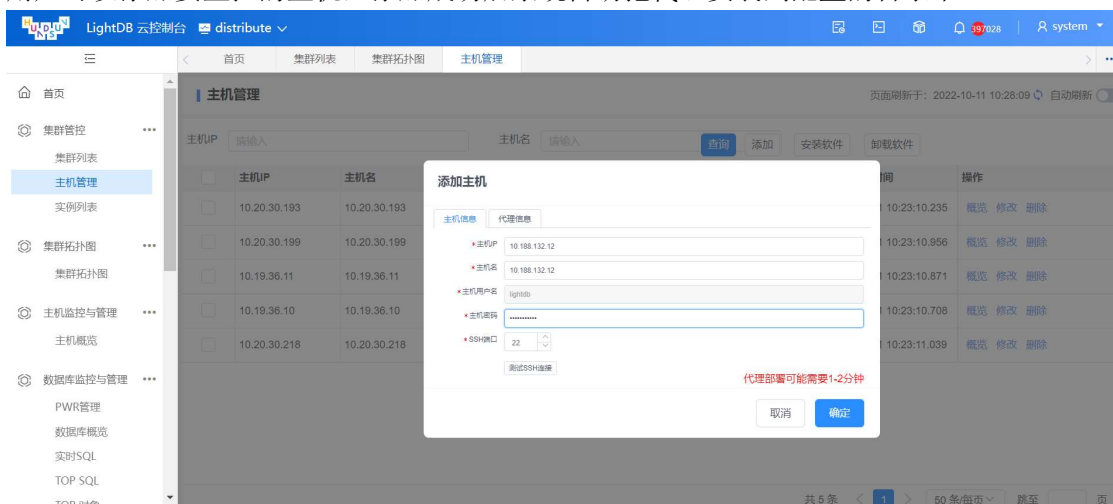
3.5.2 主机管理

通过添加主机部署监控实例代理。点击代理状态可跳转至监控代理日志。点击概览可以跳转到主机概览。



3.5.2.1 添加主机

用户可以添加要监控的主机，添加成功后系统自动把代理安装到配置的目录中



主机 IP: 要监控的主机 IP 地址

主机名: 要监控的主机名信息，方便识别主机

SSH 端口: 添加主机同时，系统通过 SCP 把代理安装到对应主机，因此需要对应的 SSH 端口

主机用户名: 登录主机的用户名

主机密码: 登录主机用户名对应的密码，用于安装代理

添加主机

主机信息
代理信息

*代理端口

*代理路径

*agent包上传

点击或将文件拖拽到这里上传

lightdb-agent-13.8-22.3-cecbc36-el7.x86_64.zip

✕

代理部署可能需要1-2分钟

取消
确定

代理端口：要监控主机中未使用的端口，默认 17334。建议 1733X 未占用相关端口；
代理安装路径：代理安装的路径，需要主机用户名有 RWX 权限的空目录。
agent 包上传：上传打包好的 agent 的 zip 包，根据服务器的版本选择对应的 zip 包。

主机管理 页面更新于：2022-06-29 19:25:05 | 自动刷新

所属集群
主机IP
主机名
查询
添加

集群名字	角色	主机IP	主机名	代理端口	代理状态	最近检测时间	操作
高可用	HA Primary	10.188.131.245	10.188.131.245	17334	运行正常	2022-06-29 19:43:10	概览 修改 删除
205高可用	HA Primary	10.20.31.205	10.20.31.205	17334	运行正常	2022-06-29 19:43:10	概览 修改 删除
高可用	HA Standby	10.188.132.12	10.188.132.12	17334	未安装	2022-06-29 19:43:13	概览 修改 删除

添加主机后代理部署可能需要 1~2 分钟，代理状态为运行正常时部署成功。

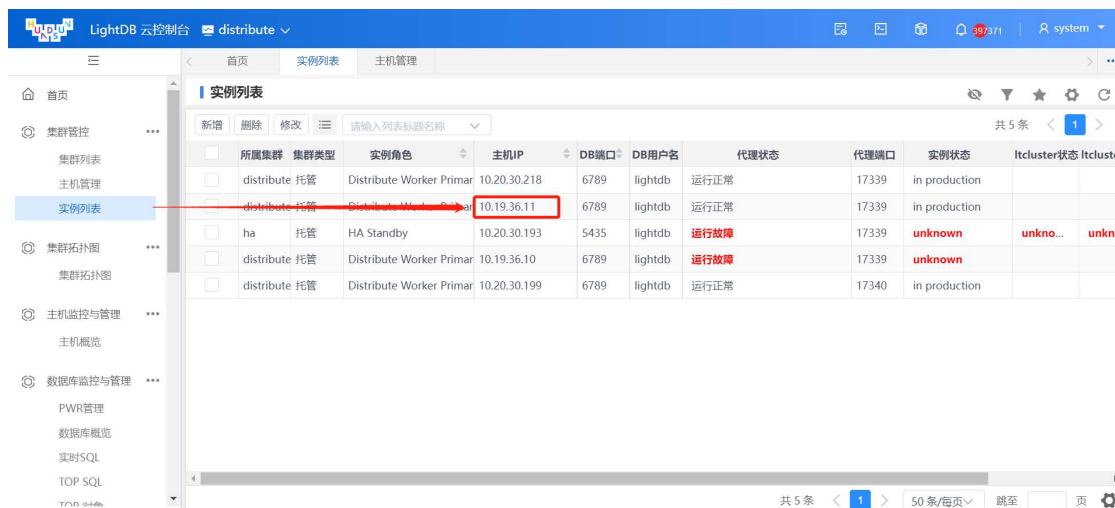
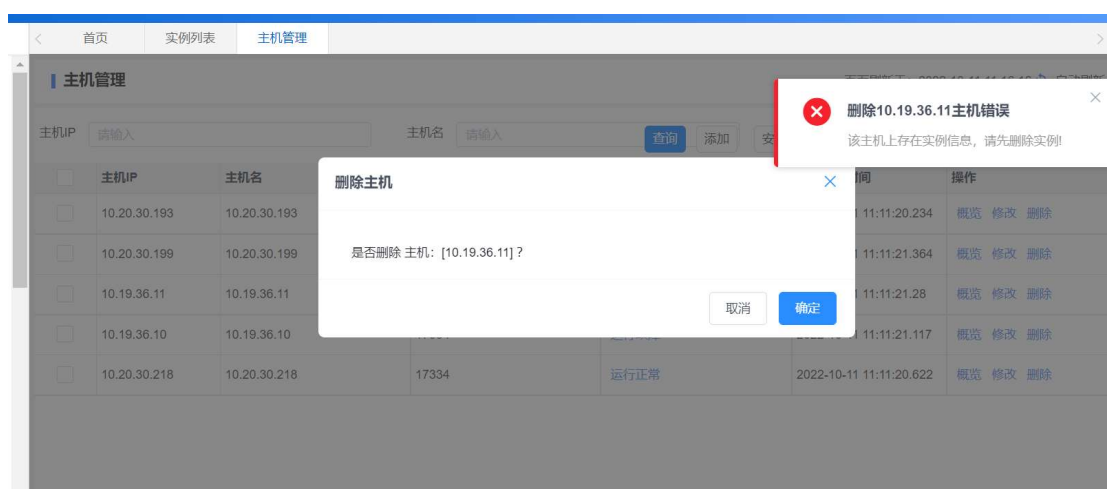
3.5.2.2 修改主机

修改主机只能修改主机密码、端口以及数据库实例密码。



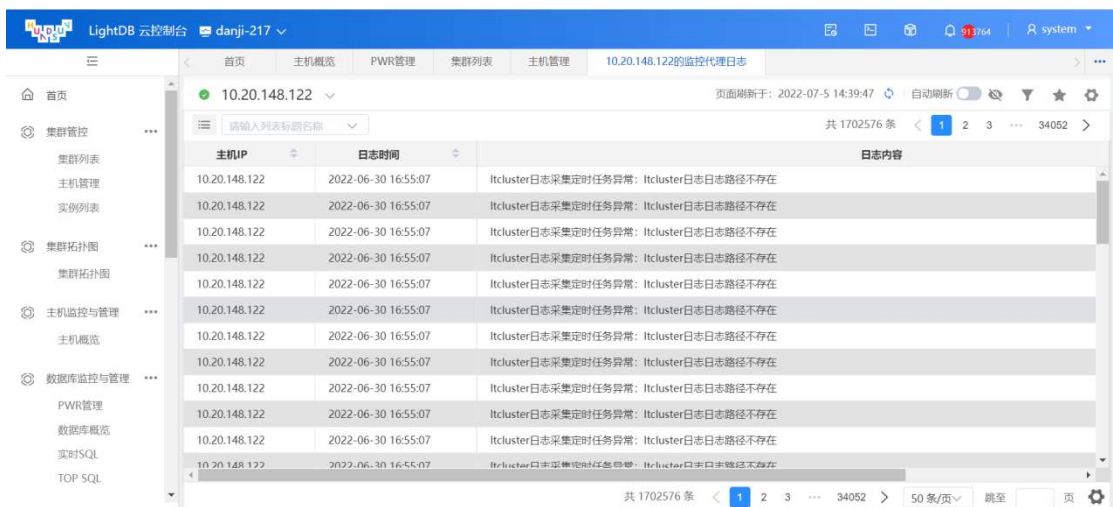
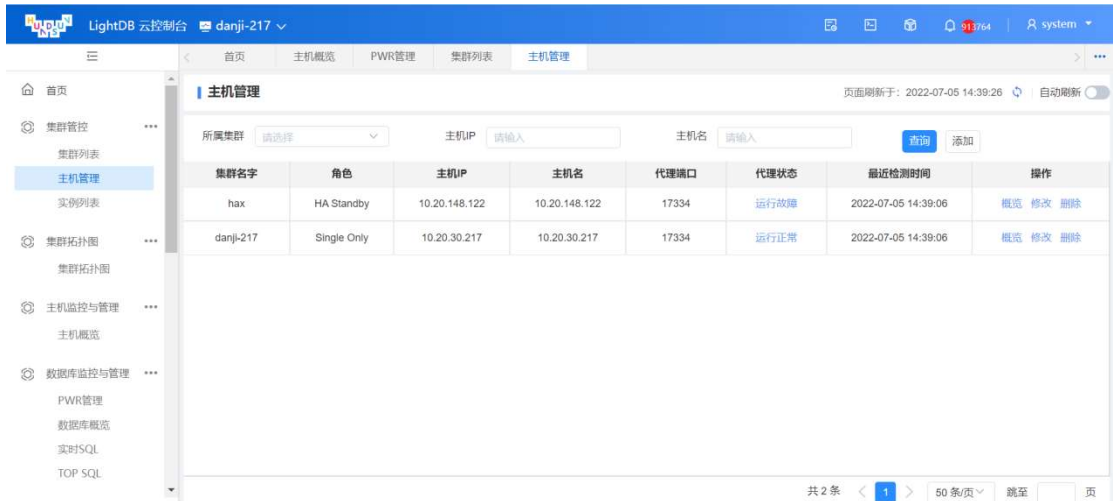
3.5.2.3 删除主机

如果不需要监控可以删除主机。删除前必须先删除部署在主机上的数据库实例监控。



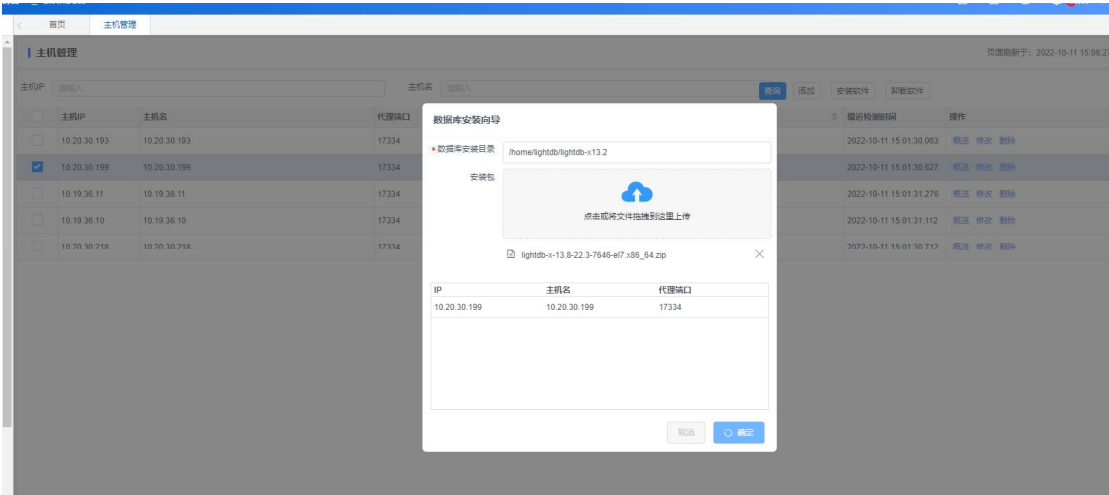
3.5.2.4 查看主机代理日志

在主机管理列表中，点击代理状态列的值可进入主机代理日志页面。

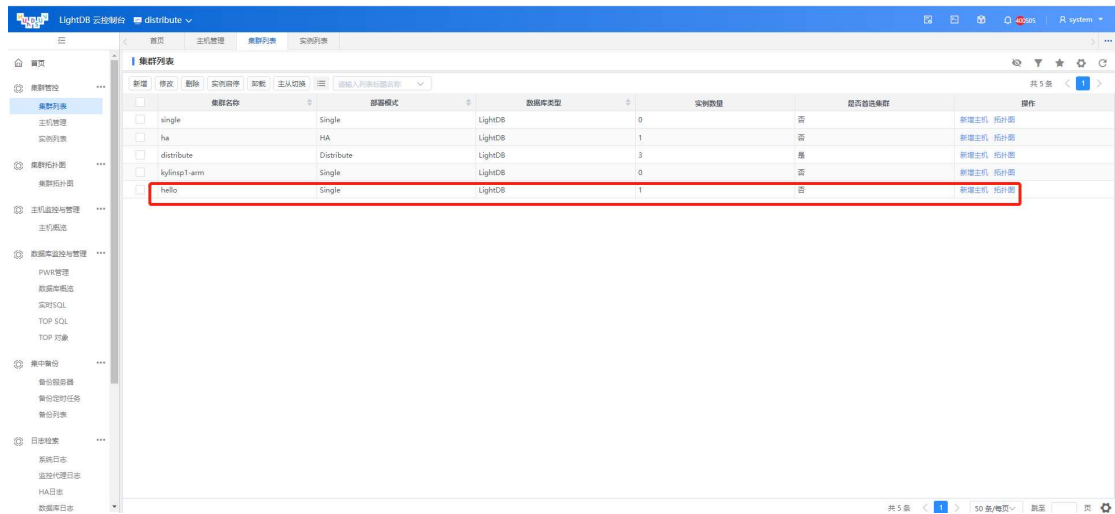
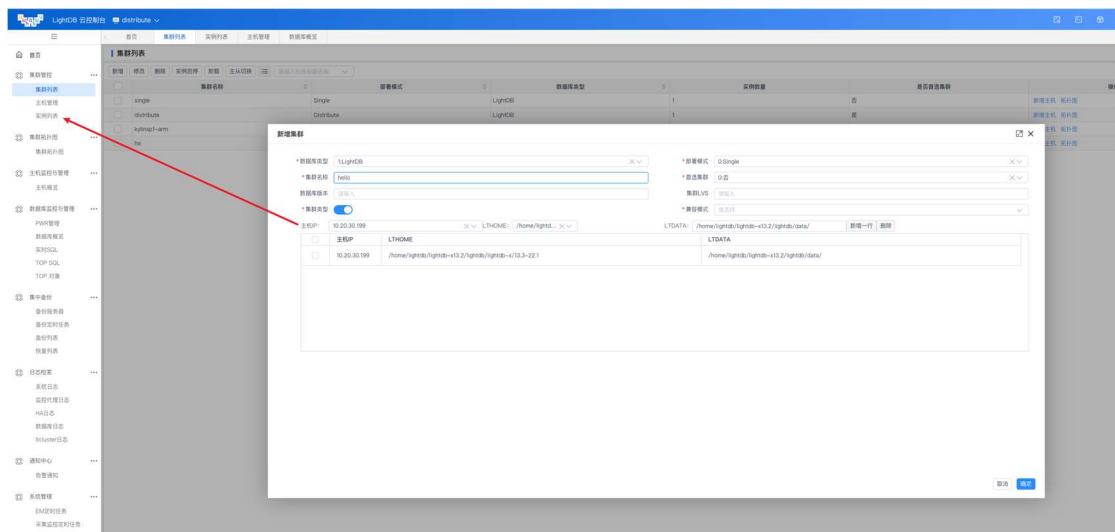


3.5.2.5 安装软件

安装软件是指安装数据库，主机管理填写数据库安装目录，需要主机用户名有 RWX 权限的空目录，上传操作系统版本相匹配的 lightdb-x 安装包。

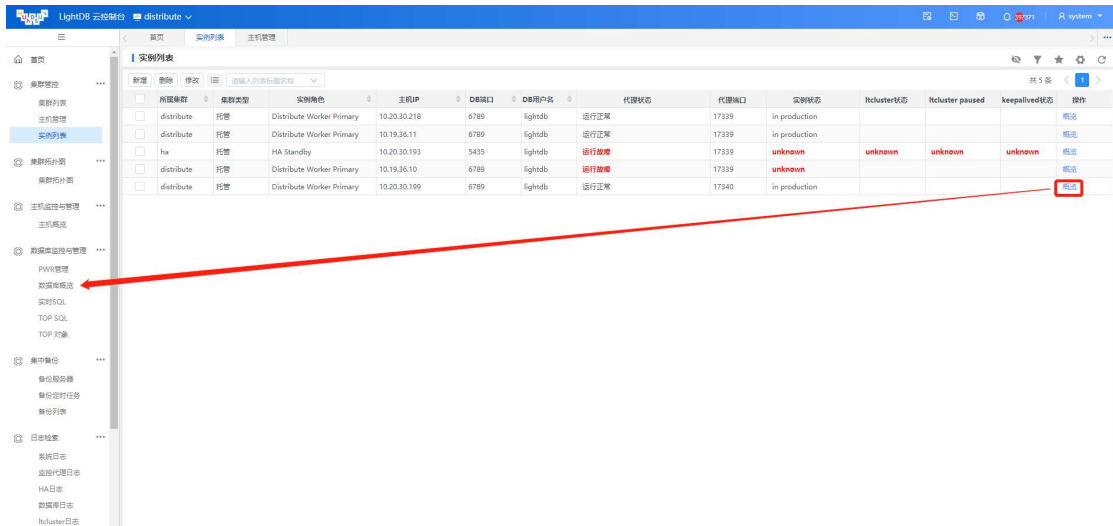


集群列表中点击新增集群可以基于已安装数据库软件的主机，在线安装数据库实例。



3.5.3 实例列表

实例列表展示用户添加集群下主机对应部署的数据库实例信息。包含所属集群、实例角色、主机 IP、数据库端口、数据库用户名、数据库实例状态。可以根据所属集群、实例角色、主机 IP 和实例状态进行查询，同时提供点击概览跳转至数据库概览界面功能。



3.5.3.1 添加实例

用户可以添加要监控的数据库实例，添加成功后系统自动把代理安装到对应主机下代理路径的目录



代理端口：要监控主机中未使用的端口，建议 1733X 未占用相关端口；
 代理安装路径：代理安装的路径，需要主机用户名有 RWX 权限的空目录
 agent 包上传：上传打包好的 agent 的 zip 包，根据服务器的版本选择对应的 zip 包。

代理信息
实例信息

* 所属集群 X v

* 所属主机 X v

* 实例角色 X v

* 数据库家目录

* 数据库

* 数据库端口 ^ v

* 数据库用户

* 数据库密码

代理部署可能需要1-2分钟

所属集群：所要监控的数据库实例对应的集群。

所属主机：所要监控的数据库实例安装在哪个主机上。

实例角色：所要监控的数据库实例对应的角色。跟集群中部署模式层级联动，Single 单机模式对应 Single Only;HA 高可用模式 对应 HA Primary、HA Standby、HA Witness; Distribute 分布式模式对应 Distribute Coordinator Primary、Distribute Coordinator Standby、Distribute Coordinator Witness、Distribute Worker Primary、Distribute Worker Standby、Distribute Worker Witness;

数据库家目录：安装数据实例的家目录。

数据库：默认 postgres

数据库端口：数据库实例对应端口。

数据库用户：登录数据库的用户名

数据库密码：登录数据库的密码

3.5.3.2 修改实例

修改主机只能修改主机密码、端口以及数据库实例密码。

代理信息 | **实例信息**

* 代理端口: 17339

* 代理路径: /home/lightdb/auto-agent/em217/instance

代理部署可能需要1-2分钟

取消 确定

代理信息 | **实例信息**

* 所属集群: distribute

* 所属主机: 10.20.30.218

* 实例角色: Distribute Worker Primary

* 数据库目录: /home/lightdb/zhj/stage/lightdb-x

* 数据库: postgres

* 数据库端口: 6789

* 数据库用户: lightdb

* 数据库密码:

代理部署可能需要1-2分钟

取消 确定

3.5.3.3 删除实例

如果不需要监控用户可以删除数据库。删除前提必须先删除部署在主机上的备份定时任务。

实例名称	所属集群	实例角色	主机IP	DB端口	DB用户名	代理状态	代理端口	实例状态	hcluster状态	hcluster paused	keepalived状态	操作
distribute	托管	Distribute Worker Primary	10.19.36.11	6789	lightdb	运行正常	17339	in production			未知	删除
ha	托管	HA Standby	10.20.30.193	5435	lightdb	运行故障	17339	unknown	unknown	unknown	unknown	删除
distribute	托管	Distribute Worker Primary	10.20.30.218	6789	lightdb	运行正常	17339	in production			未知	删除
distribute	托管	Distribute Worker Primary	10.19.36.10	6789	lightdb	运行故障	17339	unknown			未知	删除

3.5.4 全局非空集群列表

集群列表展示集群监控的实例信息。

集群名称	部署模式	数据库类型	实例数量	是否首选集群	操作
PG15	Single	LightDB	0	否	新建主机 添加副本
rocky-ha	HA	LightDB	2	否	新建主机 添加副本
rocky-distribute	Distribute	LightDB	3	是	新建主机 添加副本
PG11	Single	LightDB	0	否	新建主机 添加副本
PG12	Single	LightDB	0	否	新建主机 添加副本
PG13	Single	LightDB	0	否	新建主机 添加副本
PG14	Single	LightDB	0	否	新建主机 添加副本

3.6 主机监控与管理

主机监控与管理展示了数据库服务器的主机概览信息。

3.6.1 主机概览

主机概览包括数据时间范围统一选择、自动刷新、主机信息、主机配置、CPU 使用率、网络流量分析、内存使用率、交换空间使用量、文件系统、磁盘活动和性能测试。

3.6.1.1 时间范围统一选择

时间范围统一选择提供的选择范围有 0.5 小时、1 小时、6 小时、12 小时、24 小时和自定义范围。通过选择不同的时间范围，页面下面的图表就可以展示不同时间范围的数据，除了自定义时间范围外，其他的时间范围都是以服务器当前时间作为结束时间向前推导开始时间来查询数据。



自定义范围是用户根据实际需要可以选择图表数据范围的起止时间。确定后，页面上展示选择的时间范围，同时图表数据按照选定的时间范围进行加载。首先选择日期范围

(2022-01-29 16:29:03)

时间范围 0.5小时 ^

主机配置

启动 | 持续 3月20天6小时

CPU: HUAWEI,Kunp

内存: 765G

Swap: 16G

内核参数 >

选择时间

清空 确定

2022年 1月 2022年 2月

日	一	二	三	四	五	六	日	一	二	三	四	五	六
26	27	28	29	30	31	1	30	31	1	2	3	4	5
2	3	4	5	6	7	8	6	7	8	9	10	11	12
9	10	11	12	13	14	15	13	14	15	16	17	18	19
16	17	18	19	20	21	22	20	21	22	23	24	25	26
23	24	25	26	27	28	29	27	28	1	2	3	4	5
30	31	1	2	3	4	5	6	7	8	9	10	11	12

日期范围选择后，可以选择时间范围

(2022-01-29 16:29:03)

时间范围 0.5小时 ^

主机配置

启动 | 持续 3月20天6小时

CPU: HUAWEI,Kunp

内存: 765G

Swap: 16G

内核参数 >

选择日期

清空 确定

开始时间			结束时间		
00	00	00	00	00	00
01	01	01	01	01	01
02	02	02	02	02	02
03	03	03	03	03	03
04	04	04	04	04	04
05	05	05	05	05	05
06	06	06	06	06	06
07	07	07	07	07	07
08	08	08	08	08	08

3.6.1.2 自动刷新

自动刷新默认是开启的且刷新频率为 5 秒，用户可以根据需要开启。自动刷新频率选择的范围有 1 秒、3 秒、5 秒、15 秒、30 秒、1 分钟和 5 分钟。开启自动刷新，选择频率后，页面会根据选择的时间频率加载图表的数据。



3.6.1.3 主机信息和主机配置

主机概览展示监控的数据库**主机信息**，包括主机名、Linux 版本、内核版本、启动时间和内核参数入口。

主机信息

主机名: 10.19.69.193

启动 | 持续 2月28天19小时

Linux版本: CentOS Linux 7.6.1810

[内核参数 >](#)

内核版本: CentOS Linux 4.14.0-115.el7a.0.1.aarch64

点击内核参数入口进入内核参数列表

内核参数

参数名	参数值	建议值
aio_max_nr	65536	1048576
dirty_background_ratio	10	5
dirty_expire_centiseecs	3000	500
dirty_ratio	20	40
dirty_writeback_centiseecs	500	250
file_max	1607571	131072
overcommit_memory	0	2
overcommit_ratio	50	50
sem	32000 1024000000 500 32000	500 2048000 200 4096
shmall	18446744073692774399	4230005915648
shmmax	18446744073692774399	17326104230494207
shmmni	4096	4096

内核参数项介绍

参数	说明
shmmni	整个系统的共享内存段的最大数目（个）
shmmax	内核所允许的最大共享内存段的大小（bytes）
shmall	在任何给定时刻，系统上可以使用的共享内存的总量（bytes）
sem	用于控制内核信号量，信号量是 System V IPC 用于进程间通讯的方法
swappiness	控制换出运行时内存的相对权重。swappiness 参数值可设置范围在 0 到 100 之间。低参数值会让内核尽量少用交换，更高参数值会使内核更多的去使用交换空间。默认值为 60
overcommit_memory	内核针对内存分配的策略，其值可以是 0、1、2。0 表示内核将检查是否有足够的可用内存供应用进程使用；如果有足够的可用内存，内存申请允许；否则，内存申请失败，并把错误返回给应用进程。1 表示内核允许分配所有的物理内存，而不管当前的内存状态如何。2 表示内核允许分配超过所有物理内存和交换空间总和的内存（参照 overcommit_ratio）
overcommit_ratio	如果 overcommit_memory=2，可以过载内存的百分比，通过以下公式来计算系统整体可用内存。 系统可分配内存=交换空间+物理内存*overcommit_ratio/100
dirty_background_ratio	表示脏数据到达系统整体内存的百分比，此时触发 pdflush 进程把脏数据写回磁盘
dirty_ratio	表示如果进程产生的脏数据到达系统整体内存的百分比，此时进程自行把脏数据写回磁盘
dirty_expire_centiseecs	表示如果脏数据在内存中驻留时间超过该值，pdflush 进程在下次将把这些数据写回磁盘
dirty_writeback_centiseecs	表示 pdflush 进程周期性间隔多久把脏数据写回磁盘
aio-max-nr	该参数决定了系统中所允许的文件句柄最大数目
file-max	指定了可以分配的文件句柄的最大数目。如果用户得到的错误消息声明由于打开文件数已经达到了最大值，从而他们不能打开更多文件，则可能需要增加该值

somaxconn	定义了系统中每一个端口最大的监听队列的长度,这是个全局的参数,默认值为 1024
tcp_max_syn_backlog	对于那些依然还未获得客户端确认的连接请求, 需要保存在队列中最大数目
tcp_tw_reuse	是否允许重新应用处于 TIME-WAIT 状态的 socket 用于新的 TCP 连接。缺省设置为 0
tcp_syn_retries	示本机向外发起 TCP SYN 连接超时重传的次数, 不应该高于 255; 该值仅仅针对外出的连接, 对于进来的连接由 tcp_retries1 控制
tcp_retries2	表示放弃在已经建立通讯状态下的一个 TCP 数据包前进行重传的次数。缺省设置为 15
tcp_slow_start_after_idle	关闭 tcp 的连接传输的慢启动, 即先休止一段时间, 再初始化拥塞窗口

主机概览展示监控的数据库主机配置, 包括 CPU、存储、内存和 Swap 交换区。

主机配置

CPU: HUAWEI,Kunpeng 920 96核 存储: 47896G

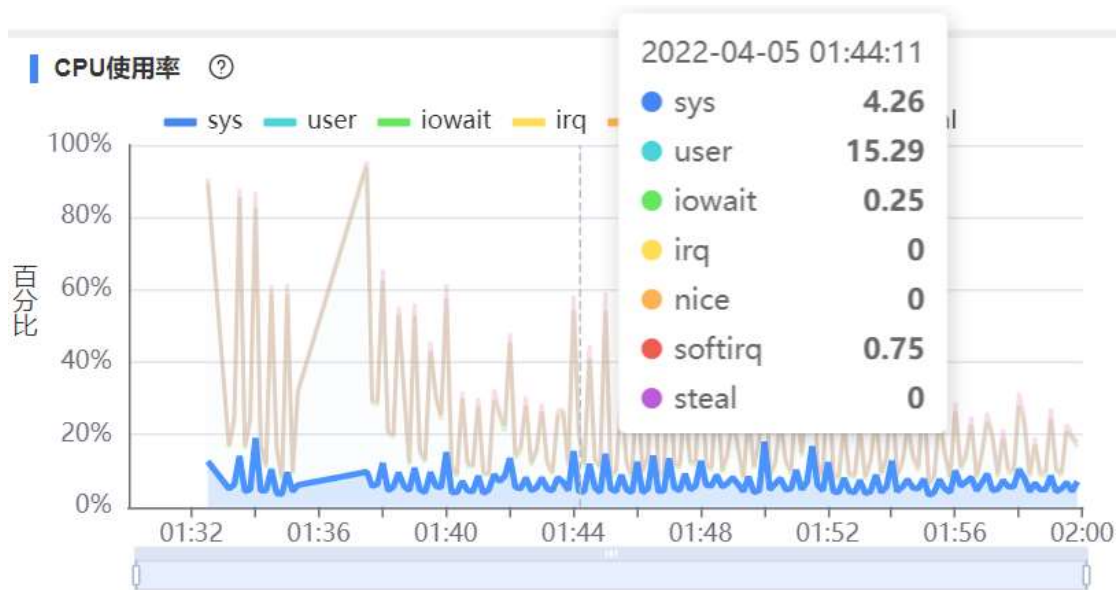
内存: 765G

Swap: 16G

3.6.1.4 CPU 使用率

目前系统每 10 秒采集一次 CPU 使用时间信息, 主要采集包括 user、nice、system、idle、iowait、irq、softirq 和 steal。采集数据项介绍:

数据项	说明
user	代表用户态 CPU 时间比率。通常缩写为 us
nice	代表低优先级用户态 CPU 时间比率, 也就是进程的 nice 值被调整为 1-19 之间是的 CPU 时间。通常缩写为 ni
system	代表内核态 CPU 时间比率。通常缩写为 sys
idle	代表空闲时间, 它不包括 I/O 等待时间 (iowait)。通常缩写为 id
iowait	代表等待 I/O 的 CPU 时间比率。通常缩写为 wa
irq	代表处理硬中断的 CPU 时间比率。通常缩写为 hi
softirq	代表处理软中断的 CPU 时间比率。通常缩写为 si
steal	代表当系统运行在虚拟机中的时候, 被其他虚拟机占用的 CPU 时间比率。通常缩写为 st



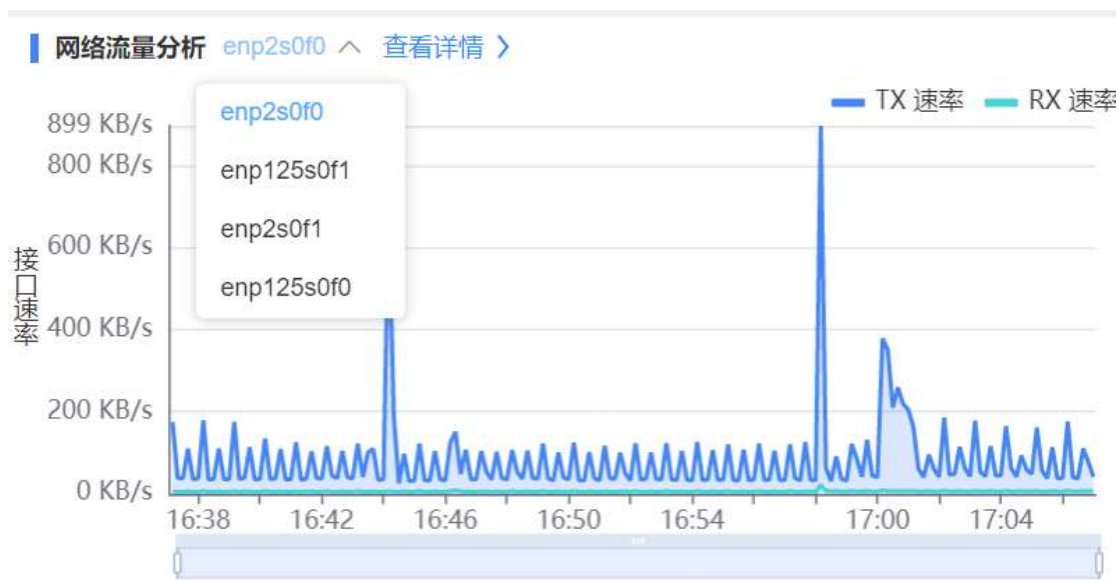
CPU 使用率计算公式，以 user 项为例：

user 使用率 = $\frac{user}{system+user+iowait+irq+nice+softirq+steal}$

3.6.1.5 网络流量分析

网络流量每 10 秒采集一次，采集每个网卡的发送和接收流量速率。采集数据项介绍：

数据项	说明
TX 速率	从网卡发出的流量速率
RX 速率	网卡接收流量速率



同时提供针对网卡进程的实时流量监控，页面端提供自动刷新功能：

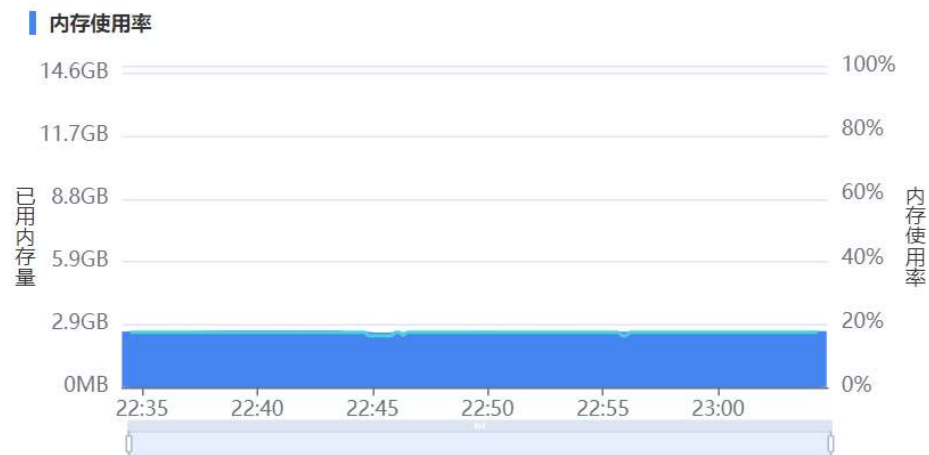
10.19.36.10 页面刷新于: 2022-01-20 18:12:34 自动刷新 15秒

enp2s0f0 实时网络流量分析

进程PID	进程名	发送字节数	接收字节数	总字节数
54790	lightdb	37094	2016	39110
219012	nginx	570	698	1268

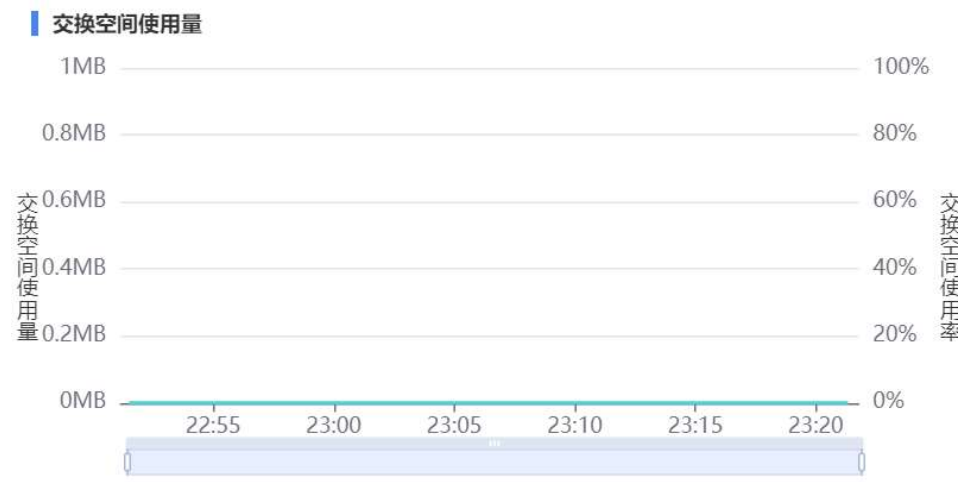
3.6.1.6 内存使用率

内存使用每 10 秒采集一次，每次展示的是当前采集时间点的系统内存使用情况。



3.6.1.7 交换空间使用量

交换空间使用每 10 秒采集一次，每次展示的是当前采集时间点的系统交换空间使用情况。



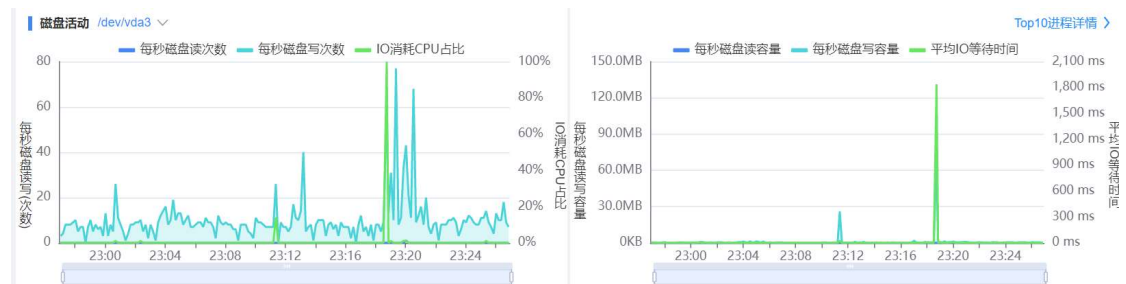
3.6.1.8 文件系统

文件系统展示最新的文件系统名称、类型、挂载点和最新的使用情况。

设备名称	文件系统	类型	挂载点	总空间	可用空间	使用率(%)
/dev/vda	/dev/vda1	xfs	/	300G	272.3G	9.22

3.6.1.9 磁盘活动

磁盘活动展示磁盘每秒读写次数、每秒读写容量、IO 消耗 CPU 占比和平均 IO 等待时间



同时展示 Top10 进程详情

Top10进程 (按%uti排序)

页面刷新于: 2022-01-23 23:26:56 | 自动刷新

PID	进程名称	读取速度(KB/s)	写入速度(KB/s)	IO占用(%)
2	kthreadd	0.00	0.00	0.00
4	kworker/0:0H	0.00	0.00	0.00
6	mm_percpu_wq	0.00	0.00	0.00
7	ksoftirqd/0	0.00	0.00	0.00
8	rcu_sched	0.00	0.00	0.00
9	rcu_bh	0.00	0.00	0.00
10	migration/0	0.00	0.00	0.00
11	watchdog/0	0.00	0.00	0.00
12	cpuhp/0	0.00	0.00	0.00
13	cpuhp/1	0.00	0.00	0.00

3.6.1.10 性能测试

此处生成磁盘性能测试报告：

性能测试 [下载性能报告](#) 重新测试

1 second per test
O_DIRECT supported on this platform for open_datasync and open_sync.

Compare file sync methods using one 8kB write:
(in wal_sync_method preference order, except fdatsync is Linux's default)
open_datasync 1237.102 ops/sec 808 usecs/op
fdatsync 1186.694 ops/sec 843 usecs/op
fsync 1035.138 ops/sec 966 usecs/op
fsync_writethrough n/a
open_sync 1027.339 ops/sec 973 usecs/op

3.7 数据库监控与管理

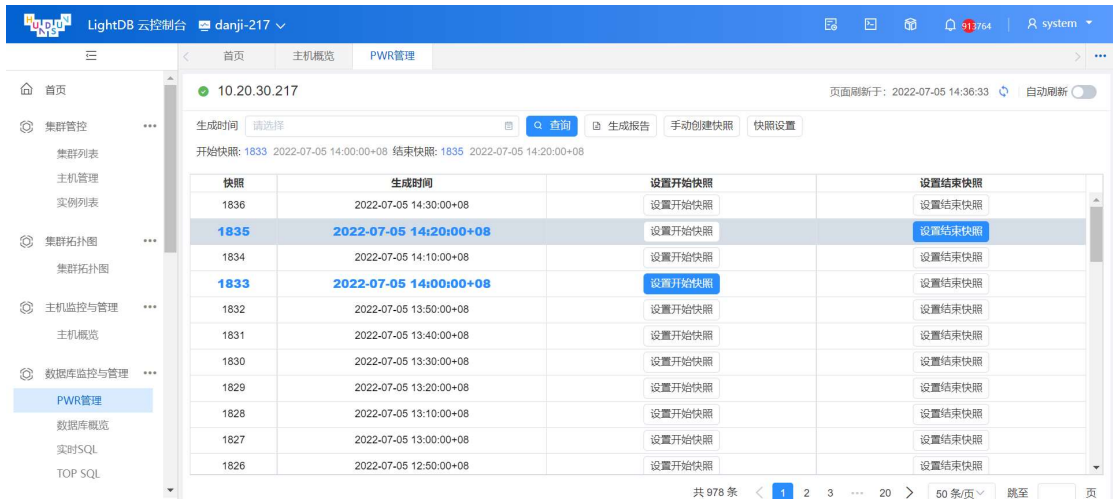
数据库监控与管理监控数据库的一些核心功能，展示数据库的运行状态，方便用户查看数据库运行是否监控，如果出现问题，用户也可以及时的根据这些功能分析定位数据库存在的问题。数据库监控与管理包括 PWR 管理、数据库概览、代理监控状态和 TOP SQL。

3.7.1 PWR 管理

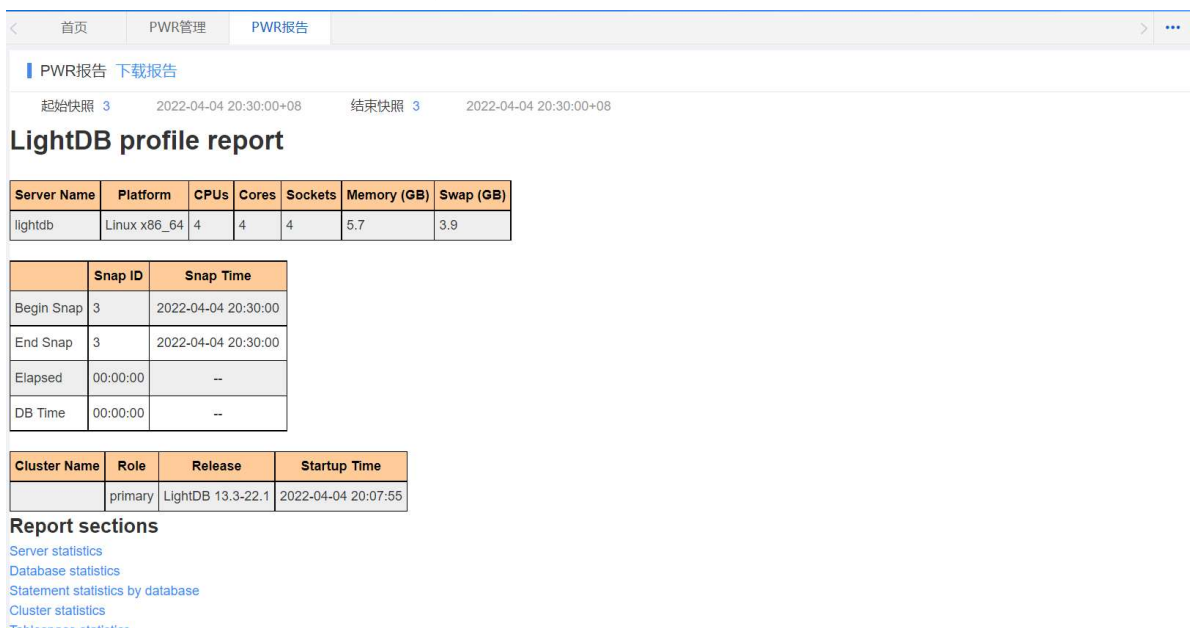
数据库每 10 秒定时生成数据库快照。PWR 管理功能真是所有快照，根据时间范围可以查询快照列表。

The screenshot shows a web interface for PWR management. At the top, there are navigation tabs: '首页', 'PWR管理', and 'PWR报告'. Below the tabs, the IP address '192.168.86.10' is displayed. The main content area features a table with columns: '快照', '生成时间', '设置开始快照', and '设置结束快照'. The table contains 11 rows of data, each representing a snapshot taken at 10-second intervals from 20:10:00+08 to 21:50:00+08. Below the table, there is a pagination bar showing '共 36 条' and '1' of 50 items per page.

快照	生成时间	设置开始快照	设置结束快照
1	2022-04-04 20:10:00+08	设置开始快照	设置结束快照
2	2022-04-04 20:20:00+08	设置开始快照	设置结束快照
3	2022-04-04 20:30:00+08	设置开始快照	设置结束快照
4	2022-04-04 20:40:00+08	设置开始快照	设置结束快照
5	2022-04-04 20:50:00+08	设置开始快照	设置结束快照
6	2022-04-04 21:00:00+08	设置开始快照	设置结束快照
7	2022-04-04 21:10:00+08	设置开始快照	设置结束快照
8	2022-04-04 21:20:00+08	设置开始快照	设置结束快照
9	2022-04-04 21:30:00+08	设置开始快照	设置结束快照
10	2022-04-04 21:40:00+08	设置开始快照	设置结束快照
11	2022-04-04 21:50:00+08	设置开始快照	设置结束快照



用户选择 2 个快照可以生成 PWR 报告



用户可以设置快照保存时间期限



用户可以自己创建快照



3.7.2 数据库概览

数据库概览展示了数据库核心的配置和运行信息。包括 GUC 参数、冻结事务比例、节点信息、上次备份、活跃会话数、事务数、缓存命中率、DML 影响行数、等待事件、表膨胀监控、临时文件、WAL 数据量信息和数据库物理写信息



3.7.2.1 数据库实例切换

数据库实例切换功能计划针对所有数据库实例进行监控。目前只实现了针对主节点的监控。

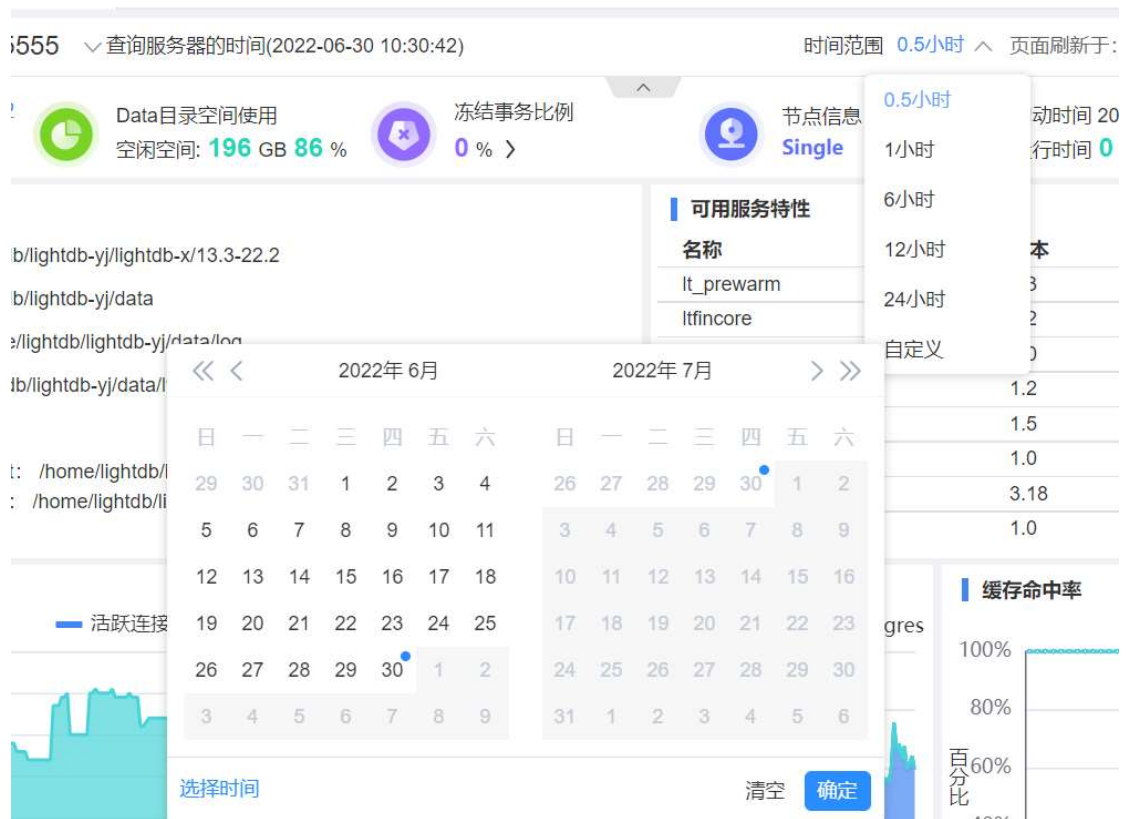


3.7.2.2 时间范围统一选择

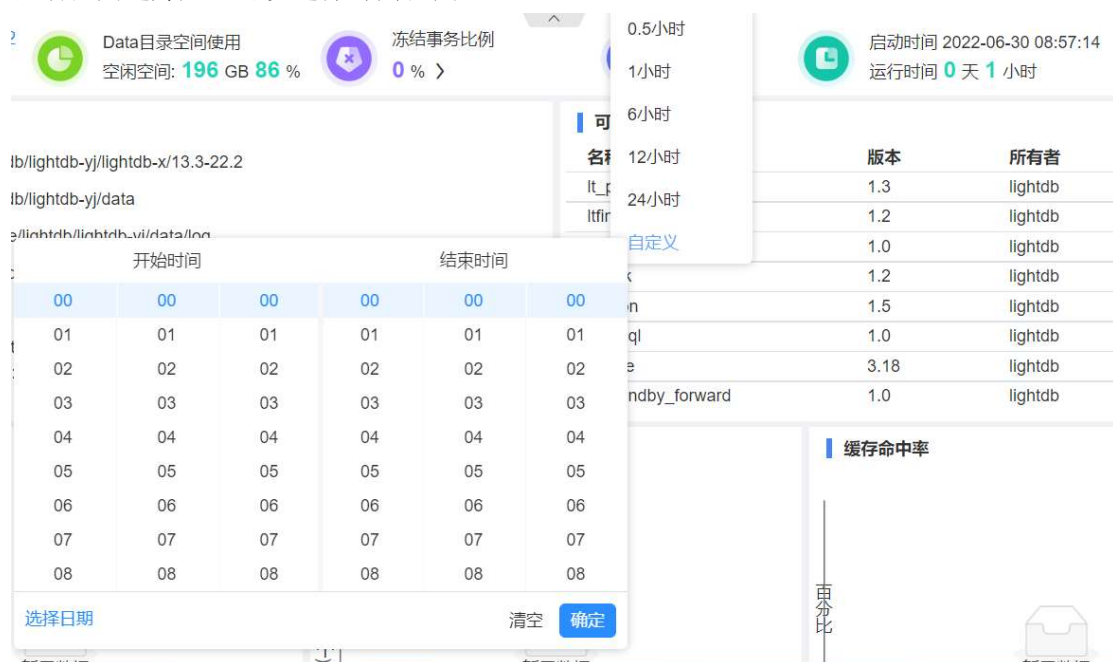
时间范围统一选择提供的选择范围有 0.5 小时、1 小时、6 小时、12 小时、24 小时和自定义范围。通过选择不同的时间范围，页面下面的图表就可以展示不同时间范围的数据，除了自定义时间范围外，其他的时间范围都是以服务器当前时间作为结束时间向前推导开始时间来查询数据。



自定义范围是用户根据实际需要可以选择图表数据范围的起止时间。确定后，页面上展示选择的时间范围，同时图表数据按照选定的时间范围进行加载。首先选择日期



日期范围选好后，可以选择时间范围



3.7.2.3 自动刷新

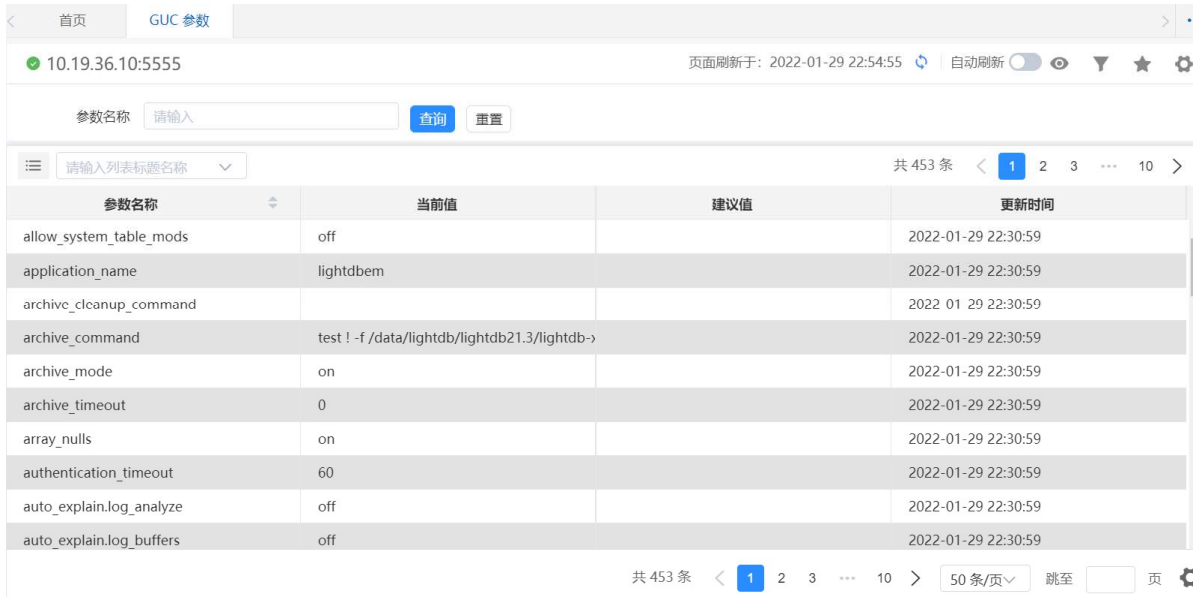
自动刷新默认是开启的且刷新频率为 5 秒，用户可以根据需要开启。自动刷新频率选择的范围有 1 秒、3 秒、5 秒、10 秒、15 秒、30 秒、1 分钟和 5 分钟。开启自动刷新，选

择频率后，页面会根据选择的时间频率加载图表的数据。

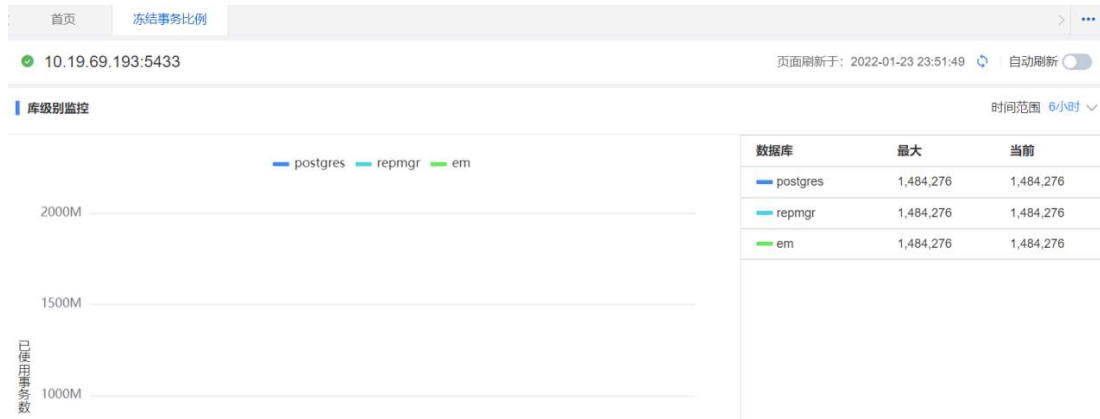


3.7.2.4 GUC 参数

GUC: Grand Unified Configuration 指的是 Lightdb 数据库的一种对数据库变量进行设置对数据库进行控制的机制。通常理解是对 lightdb.conf 文件中变量进行修改，或通过 set 命令对参数进行设置。具体以数据库配置为准。



3.7.2.5 冻结事务比例

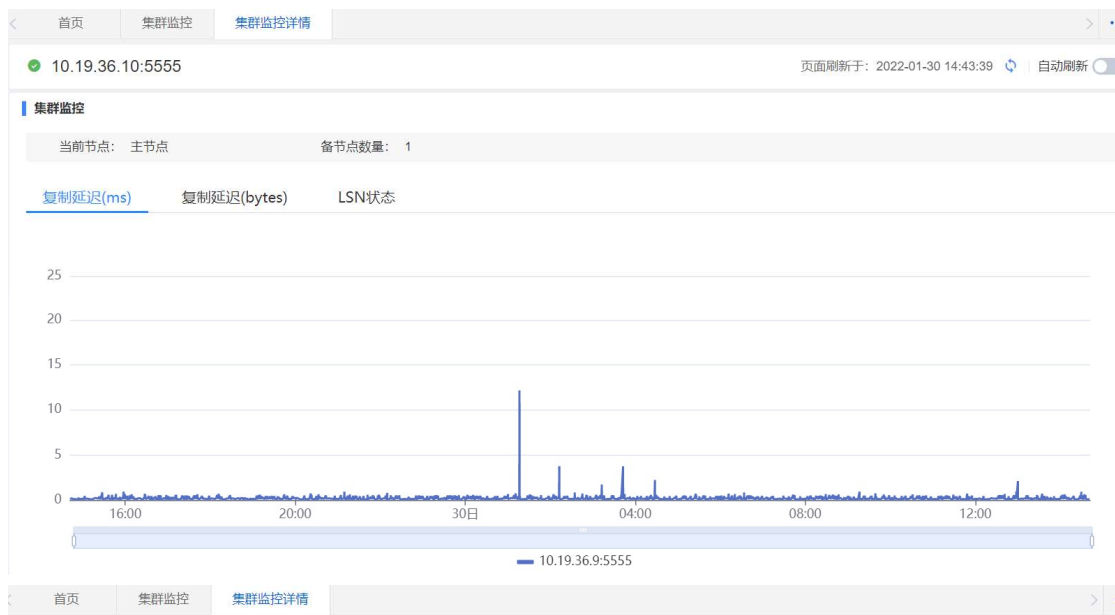


3.7.2.6 主节点信息

主节点信息展示数据库集群信息。数据库集群通过 Itcluster 来实现，集群状态通过执行 Itcluster 命令来查询 `Itcluster -f $Itcluster_config_path cluster show`

IP	端口	角色	实例状态	守护进程状态	复制模式	查看详情
10.19.69.193	5433	primary	* running	running		查看详情
10.19.69.192	5433	standby	running	running	sync	

通过查看详情，可以查看集群主节点复制延迟和 LSN 状态



10.19.36.10:5555 页面刷新于: 2022-01-30 14:43:39 自动刷新

集群监控

当前节点: 主节点 备节点数量: 1

复制延迟(ms) 复制延迟(bytes) LSN状态

instanceId	sampleTime	writesn	flushsn	replaysn
10.19.36.9	2022-01-30 14:42:00	A/2A4F5758(709,842,776)	A/2A4F5758(709,842,776)	A/2A4F5758(709,842,776)
10.19.36.9	2022-01-30 14:41:50	A/2A4BAB68(709,602,152)	A/2A4BAB68(709,602,152)	A/2A4BAB68(709,602,152)
10.19.36.9	2022-01-30 14:41:40	A/2A4452C0(709,120,704)	A/2A4452C0(709,120,704)	A/2A4452C0(709,120,704)
10.19.36.9	2022-01-30 14:41:30	A/2A377B70(708,279,152)	A/2A377B70(708,279,152)	A/2A377B70(708,279,152)
10.19.36.9	2022-01-30 14:41:20	A/2A331D88(707,992,968)	A/2A331D88(707,992,968)	A/2A331D88(707,992,968)
10.19.36.9	2022-01-30 14:41:10	A/2A2DE1A0(707,649,952)	A/2A2DE1A0(707,649,952)	A/2A2DE1A0(707,649,952)
10.19.36.9	2022-01-30 14:41:00	A/2A1A0FA8(706,351,016)	A/2A1A0FA8(706,351,016)	A/2A1A0FA8(706,351,016)
10.19.36.9	2022-01-30 14:40:50	A/2A144478(705,971,320)	A/2A144478(705,971,320)	A/2A144478(705,971,320)
10.19.36.9	2022-01-30 14:40:40	A/2A0F6298(705,651,352)	A/2A0F6298(705,651,352)	A/2A0F6298(705,651,352)

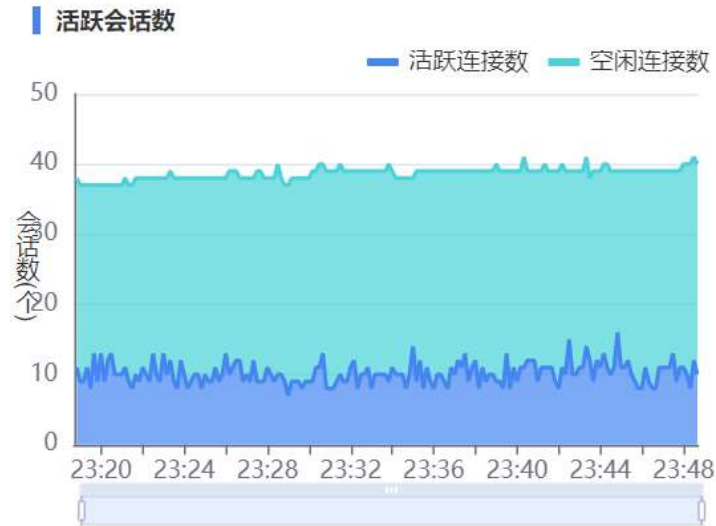
共 39034 条 < 1 2 3 ... 781 > 50 条/页 跳至 页

3.7.2.7 上次备份

本实例在集中备份模块最后一次执行备份的时间。

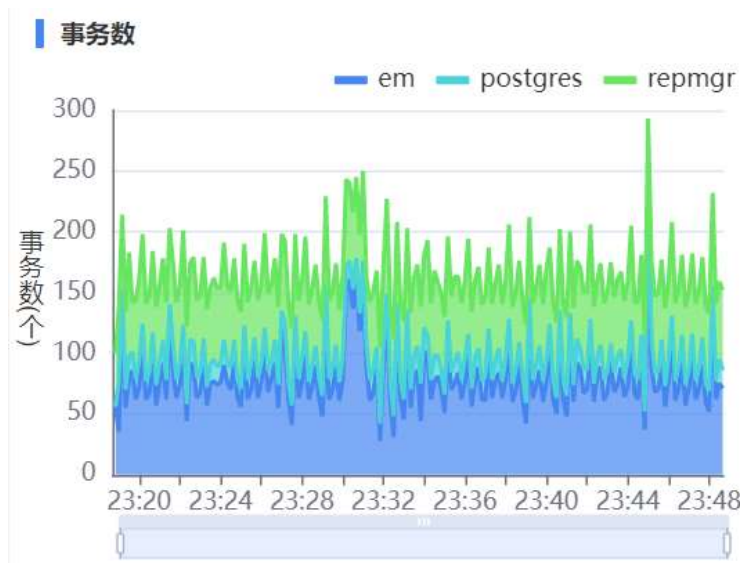
3.7.2.8 活跃会话数

活跃会话数统计视图 `lt_stat_activity` 总连接数, 然后根据条件 `state like 'idle%'` 查询空闲连接数, 根据总连接数-空闲连接数=活跃连接数



3.7.2.9 事务数

数据库事务数从视图 `pg_stat_database` 中获取 `xact_commit` 的值，`xact_commit` 代表此数据库中已提交的事务数。



3.7.2.10 缓存命中率

缓存命中率从视图 `pg_statio_all_tables` 中数据。`pg_statio_all_table` 视图中当前数据库每个表一行(包括 TOAST 表)，显示：表 OID、模式名、表名、从该表中读取的磁盘块总数、缓冲区命中次数、该表上所有索引的磁盘块读取总数、该表上所有索引的缓冲区命中总数、在该表的辅助 TOAST 表(如果存在)上的磁盘块读取总数、在该表的辅助 TOAST 表(如果存在)上的缓冲区命中总数、TOAST 表的索引的磁盘块读取总数、TOAST 表的索引的缓冲区命中总数。当前数据库中的每个表一行，显示有关在指定表上 I/O 的统计信息。

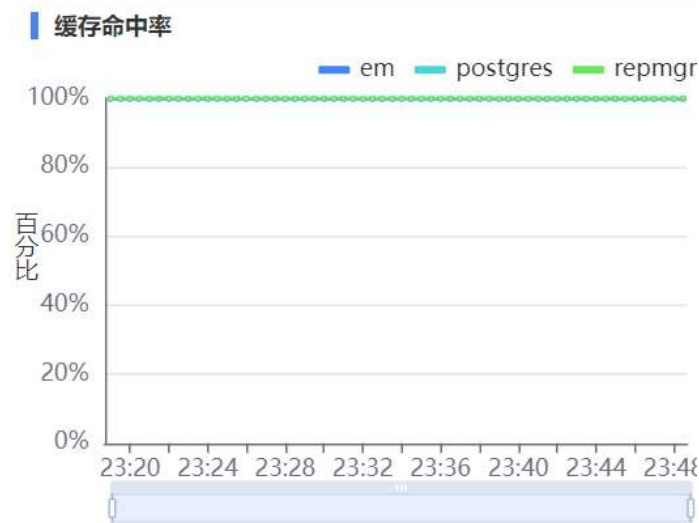
缓存命中率图表展示总的命中率数据，计算公式：

`sum(COALESCE(t.heap_blks_hit,0)`

```

+COALESCE(t.idx_blks_hit,0)
+COALESCE(t.toast_blks_hit,0)
+COALESCE(t.tidx_blks_hit,0)) over() *100
/
sum(COALESCE(t.heap_blks_read,0)
+COALESCE(t.heap_blks_hit,0)
+COALESCE(t.idx_blks_read,0)
+COALESCE(t.idx_blks_hit,0)
+COALESCE(t.toast_blks_read,0)
+COALESCE(t.toast_blks_hit,0)
+COALESCE(t.tidx_blks_read,0)
+COALESCE(t.tidx_blks_hit,0)) over() as decimal(10,2)

```



点击命中率图表上的每个点，可以查看缓存命中率明细：

repmgr

模式名称 表名称

请输入列表标题名称 共 44 条 < 1 >

模式名称	表名称	缓存命中率	当前表读取磁	表缓冲区命中	表索引读取磁	表索引缓冲区	表TOAST表	表TOAST表	表TOAST表	表TOAST表	采集时间点
repmgr	events	72.73	3	8	0	0	0	0	0	0	2022-01-30
pg_catalog	lt_class_ext	100	6	175161	6	350314	0	0	0	0	2022-01-30
repmgr	monitoring_	99.46	2365	259954	886	338290	0	0	0	0	2022-01-30
repmgr	nodes	100	3	338948	6	262442	0	0	0	0	2022-01-30
pg_catalog	pg_aggregat	99.79	4	1569	6	3139	0	0	0	0	2022-01-30
pg_catalog	pg_am	100	3	900239	0	0	0	0	0	0	2022-01-30
pg_catalog	pg_amop	100	21	3276985	25	3545464	0	0	0	0	2022-01-30
pg_catalog	pg_amproc	100	11	1703579	9	4130691	0	0	0	0	2022-01-30
pg_catalog	pg_attrdef	100	3	189198	8	365918	0	0	0	0	2022-01-30
pg_catalog	pg_attribute	100	155	10593913	40	21533782	0	0	0	0	2022-01-30

共 44 条 < 1 > 50 条/页 跳至 页

pg_statio_all_tables 视图列说明：

列名	类型	描述
relid	oid	表的 OID
schemaname	name	该表所在的模式名

relname	name	这个表的名称
heap_blks_read	bigint	从该表中读取的磁盘块的数量
heap_blks_hit	bigint	该表中的缓冲区命中数
idx_blks_read	bigint	从这个表上所有索引读取的磁盘块数
idx_blks_hit	bigint	这个表上所有索引中的缓冲区命中数
toast_blks_read	bigint	从这个表的 TOAST 表中读取的磁盘块的数量(如果有的话)
toast_blks_hit	bigint	这个表的 TOAST 表中的缓冲区命中数(如果有的话)
tidx_blks_read	bigint	从这个表的 TOAST 表索引中读取的磁盘块的数量(如果有的话)
tidx_blks_hit	bigint	这个表的 TOAST 表索引中的缓冲区命中数(如果有的话)

3.7.2.11 DML 影响行数

DML 影响行数从视图 `pg_stat_all_tables` 中数据。`pg_stat_all_tables` 视图将为当前数据库中的每一个表（包括 TOAST 表）包含一行，该行显示与对该表的访问相关的统计信息。



点击 DML 影响行数图表上的每个点，可以查看 DML 影响行数明细：

postgres

模式名称 表名称

共 211 条 < 1 2 3 ... 5 >

模式名称	表名称	Insert 行数	Update 行数	Delete 行数	Fetch 行数	采集时间点
public	baselines	0	0	0	0	2022-01-30 23:09:00
public	bl_samples	0	0	0	0	2022-01-30 23:09:00
columnar	chunk	0	0	0	0	2022-01-30 23:09:00
columnar	chunk_group	0	0	0	0	2022-01-30 23:09:00
public	funcs_list	0	0	0	0	2022-01-30 23:09:00
hint_plan	hints	0	0	0	0	2022-01-30 23:09:00
public	import_queries	0	0	0	0	2022-01-30 23:09:00
public	import_queries_version	0	0	0	0	2022-01-30 23:09:00
public	indexes_list	0	0	0	464	2022-01-30 23:09:00
cron	job	0	0	0	8	2022-01-30 23:09:00

共 211 条 < 1 2 3 ... 5 > 50 条/页 跳至 页

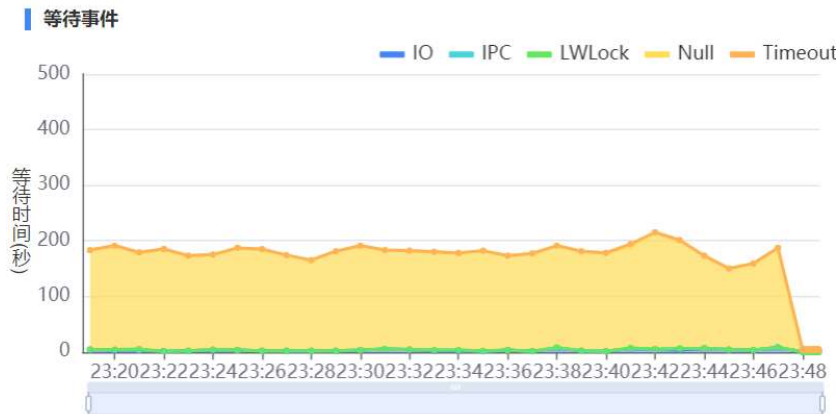
pg_stat_all_tables 视图列说明:

列名	类型	描述
relid	oid	表的 OID
schemaname	name	该表所在的模式的名称
relname	name	这个表的名称
seq_scan	bigint	在此表上启动的顺序扫描数
seq_tup_read	bigint	连续扫描获取的实时行数
idx_scan	bigint	对这个表发起的索引扫描数
idx_tup_fetch	bigint	索引扫描获取的实时行数
n_tup_ins	bigint	插入的行数
n_tup_upd	bigint	更新的行数(包括 HOT 更新的行)
n_tup_del	bigint	删除的行数
n_tup_hot_upd	bigint	HOT 更新的行数(即, 不需要单独的索引更新)
n_live_tup	bigint	活的行的估计数量
n_dead_tup	bigint	僵死行的估计数量
n_mod_since_analyze	bigint	自上次分析此表以来修改的行的估计数量
n_ins_since_vacuum	bigint	自上次清空此表以来插入的行的估计数量
last_vacuum	timestamp with time zone	最后一次手动清理这个表(不包括 VACUUM FULL)
last_autovacuum	timestamp with time zone	这个表最后一次被自动清理守护进程清理的时间
last_analyze	timestamp with time zone	上一次手动分析这个表
last_autoanalyze	timestamp with time zone	自动清理守护进程最后一次分析这个表
vacuum_count	bigint	这个表被手动清理的次数(VACUUM FULL 不计数)
autovacuum_count	bigint	这个表被 autovacuum 守护进程清理的次数
analyze_count	bigint	手动分析这个表的次数

autoanalyze_count	bigint	这个表被 autovacuum 守护进程分析的次数
-------------------	--------	---------------------------

3.7.2.12 等待事件

等待事件从表 sample_activity_profile 关联 sample_activity_history 表获取数据



通过等待事件图表上的点可以访问等待事件明细：

等待事件类别	等待事件	等待时长/秒	SQL详情	采集时间
Null	Null	33	select count(*) from cron.job_run_details	2022-01-30 23:17:00
Null	Null	33		2022-01-30 23:17:00
Null	Null	33	SELECT collect_activity_history()	2022-01-30 23:17:00
Null	Null	33	SELECT take_sample()	2022-01-30 23:17:00
Null	Null	1	SELECT collect_activity_profile()	2022-01-30 23:17:00
Null	Null	1	select jrd.jobid,jrd.runid,jrd.job_pid as jobPid,jrd.database,jrd.u	2022-01-30 23:17:00
Null		134	汇总	2022-01-30 23:17:00

等待事件类型：

Activity	服务器进程空闲。此事件类型表示在其主处理循环中等待活动的进程。wait_event 将识别特定的等待点。
BufferPin	服务器进程正在等待对数据缓冲的独占访问。如果另一个进程持有一个打开的游标，该游标最后一次从相关缓冲区读取数据，则缓冲区销等待可能是漫长的。
Client	服务器进程正在等待连接到用户应用程序的套接字上的活动。因此，服务器预计发生一些独立于其内部进程的事情。wait_event 将识别特定的等待点。
Extension	服务器进程正在等待扩展模块定义的某个条件。
IO	服务器进程正在等待一个 I/O 操作完成。wait_event 将识别特定的等待点。
IPC	服务器进程正在等待与另一个服务器进程进行交互。wait_event 将识别特定的等待点。

Lock	服务器进程正在等待一个重量级锁。重量级锁，也称为锁管理器锁或简单锁，主要保护表等 SQL 可见对象。然而，它们也用于确保某些内部操作的互斥，例如关系扩展。wait_event 将识别等待的锁的类型。
LWLock	服务器进程正在等待一个轻量级锁。大多数这样的锁保护共享内存中的特定数据结构。wait_event 将包含标识轻量级锁用途的名称。(有些锁有特定的名称；其他锁是一组锁的一部分，每个锁具有类似的目的。)
Timeout	服务器进程正在等待超时过期。wait_event 将识别特定的等待点。

3.7.2.13 表膨胀监控

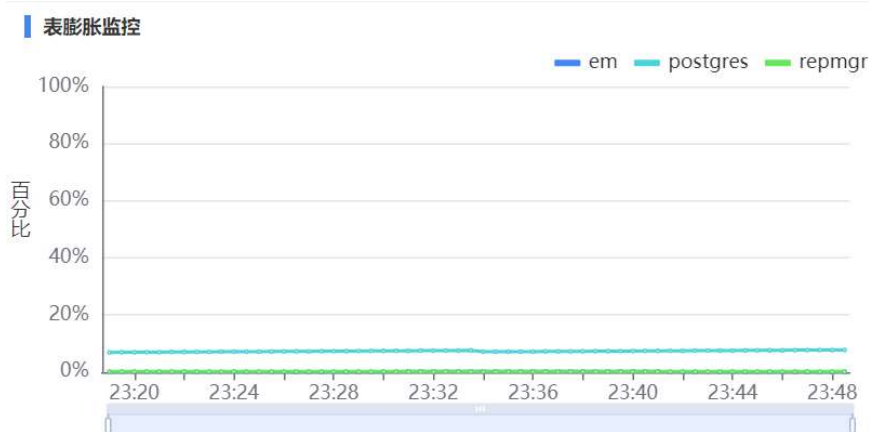
表膨胀也是从视图 pg_stat_all_tables 中获取数据计算出来结果。

表膨胀采集逻辑较为苛刻，页面没数据是正常的情况。获取表膨胀逻辑：

```

with tmp as(select schemaname,relname as
tablename,pg_relation_size(''||schemaname||'.'||relname||'') tab_size,
n_dead_tup,
n_live_tup,
coalesce(round(n_dead_tup * 100 / (case when n_live_tup + n_dead_tup = 0
then null else n_live_tup + n_dead_tup end ),2),0.00) as dead_tup_ratio,
round( case when (sum(n_live_tup + n_dead_tup) over())=0 then 0
else (sum( n_dead_tup) over())*100/(sum(n_live_tup +
n_dead_tup) over()) end ,2) dead_tup_ratio_total
from pg_stat_all_tables)
SELECT * FROM tmp WHERE schemaname not in ('pg_catalog','lt_catalog')
and tab_size > 100*1024*1024
and n_dead_tup > 1000
and n_live_tup > 1000
and dead_tup_ratio > 20
ORDER BY dead_tup_ratio DESC LIMIT 1000;

```



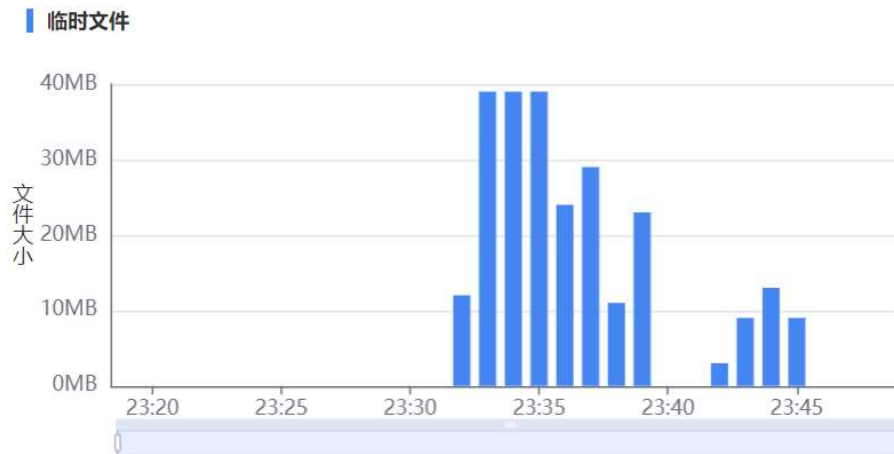
通过表膨胀监控图表上的点可以访问表膨胀明细：

模式名称	表名称	表大小/字节	存活元组数量	死元组数量	膨胀率	采集时间点
public	baselines	0	0	0	0	2022-01-30 23:25:30
public	bl_samples	0	0	0	0	2022-01-30 23:25:30
columnar	chunk	0	0	0	0	2022-01-30 23:25:30
columnar	chunk_group	0	0	0	0	2022-01-30 23:25:30
public	funcs_list	0	0	0	0	2022-01-30 23:25:30
hint_plan	hints	0	0	0	0	2022-01-30 23:25:30
public	import_queries	49152	41	0	0	2022-01-30 23:25:30
public	import_queries_version	8192	4	0	0	2022-01-30 23:25:30
public	indexes_list	32768	140	27	16	2022-01-30 23:25:30
cron	job	8192	4	4	50	2022-01-30 23:25:30

3.7.2.14 临时文件

数据库临时文件信息是从数据库日志中解析出来的数据。系统会把数据库日志定时采集到 `lem_db_temp_file_log` 表中，通过解析 `lem_db_temp_file_log` 中的数据即得到临时文件信息。临时文件数据获取逻辑：

```
select db_log_time_minute as minuteTime , round(sum(temp_file_size)/(1024*1024),2) as size
from lem_db_temp_file_log group by db_log_time_minute order by 1 asc
```



通过点击临时文件图表中的柱子，可以查看临时文件日志明细：

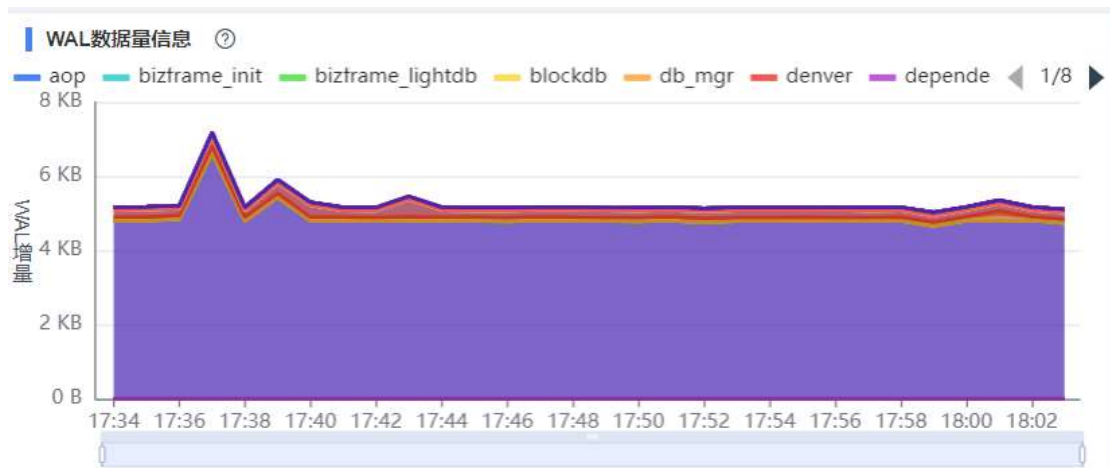
临时文件	文件大小(MB)	时间	sql
"base/pgsql_tmp/pgsql_tmp464341.474175"	0.86	2022-01-25 14:38:26 CST	SQL statement "delete from cron.job_run_details where runid in (select runid from cron.job_run_details order by runid desc limit (select count(runid) from cron.job_run_details) offset 100000)"
"base/pgsql_tmp/pgsql_tmp464341.474180"	0.86	2022-01-25 14:38:26 CST	SQL statement "delete from cron.job_run_details where runid in (select runid from cron.job_run_details order by runid desc limit (select count(runid) from cron.job_run_details) offset 100000)"
"base/pgsql_tmp/pgsql_tmp464341.474172"	0.86	2022-01-25 14:38:26 CST	SQL statement "delete from cron.job_run_details where runid in (select runid from cron.job_run_details order by runid desc limit (select count(runid) from cron.job_run_details) offset 100000)"
"base/pgsql_tmp/pgsql_tmp464341.47417"	0.85	2022-01-25 14:38:26 CST	SQL statement "delete from cron.job_run_details where runid in (select runid from cron.job_run_details order by runid desc limit (select count(runid) from cron.job_run_details) offset 100000)"

共 150 条 < 1 2 3 ... 15 > 10 条/页 跳至

3.7.2.15 WAL 数据量信息

WAL 数据量信息从视图 `pg_stat_statements` 中获取数据。WAL 汇总数据获取逻辑：

```
select pd.datname as dbName, sum(pss.wal_bytes) as walAmount
from Pg_stat_statements pss
join pg_database pd on pss.dbid = pd.oid
where pss.wal_bytes > 0 and datname != 'template1'
group by pd.datname
order by dbName;
```

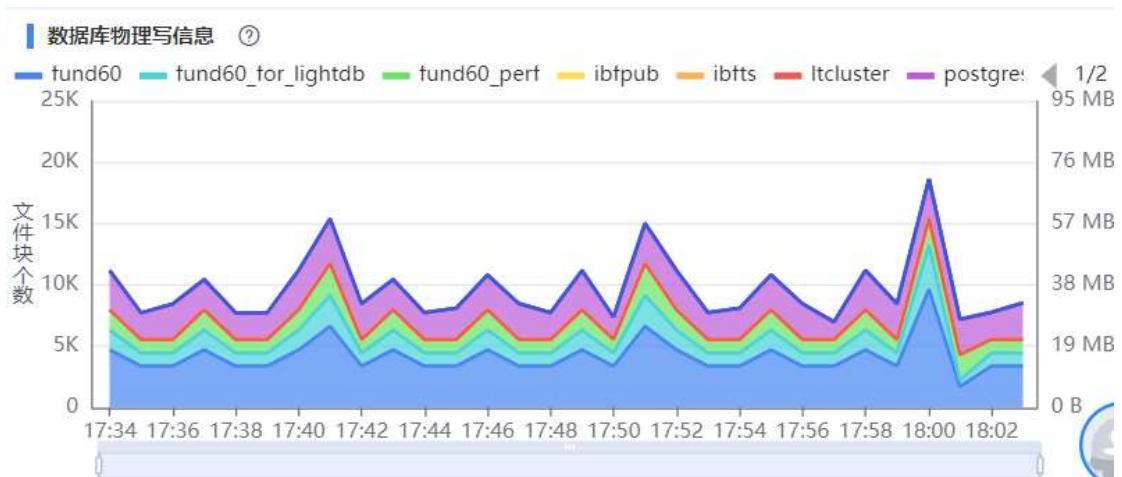


通过 WAL 数据量信息图表上的点可以访问 WAL 增量明细：

数据库名称	WAL日志数据量(字节)	SQL详情	采集时间
repmgr	38339	INSERT INTO repmgr.monitoring_history (primary_node_id, standby_node_id, last_mon	2022-01-23 23:36:00

3.7.2.16 数据库物理写信息

数据库物理写从视图 `pg_stat_statements` 中获取数据。
从节点无物理写信息。



通过数据库物理写信息图表上的点可以访问物理写明细：

数据库物理写信息

repmgr

请输入列表标题名称

共 1 条

数据库名称	写入共享块总数	写入本地块总数	写入临时块总数	写入总块数	SQL详情	采集时间
repmgr	3	0	0	3	INSERT INTO repmgr.monitoring_history (primary_node_id, standby_node_id, ...)	2022-01-23 23:...

共 1 条

50 条/页

跳至 页

3.7.3 实时 SQL

实时 SQL 展示实时活动的 sql。通过表 lt_stat_activity 得到实时 SQL，得到的数据中文化展示。

10.19.69.255:5555

页面更新于: 2022-06-30 13:32:42

共 7 条

Pid	LeaderPid	用户名称	数据库名称	应用名称	客户端地址	客户端主机名	客户端端口	状态	状态变更时间	等待事件类型	等待事件	查询开始时间	QueryId	查询语句
29106		em	lightdb	lightdbem	10.20.31.205		58520	idle in transact	2022-06-30 13	Client	ClientRead	2022-06-30 13		select host_id from lem_host where host...
26023		em	lightdb	lightdbem	10.188.131.245		58214	idle in transact	2022-06-30 13	Client	ClientRead	2022-06-30 13		select host_id from lem_host where host...
22677		em	lightdb	lightdbem	10.188.132.12		58374	idle in transact	2022-06-30 13	Client	ClientRead	2022-06-30 13		select cron_id,host_id,host_ip,cron_key,c...
27518		em	lightdb	lightdbem	10.188.131.245		53246	idle in transact	2022-06-30 13	Client	ClientRead	2022-06-30 13		select host_id from lem_host where host...
14480		em	lightdb	lightdbem	10.188.131.245		58366	idle in transact	2022-06-30 13	Client	ClientRead	2022-06-30 13		select host_id from lem_host where host...
27687		postgres	lightdb	lightdbem	10.19.69.255		45598	active	2022-06-30 13			2022-06-30 13		SELECT count(0) FROM lt_stat_activity AS

实时 SQL 搜索条件数据也是中文化展示。

10.19.69.255:5555

页面更新于: 2022-06-30 13:32:42

自动刷新

用户名称: 请输入

应用名称: 请输入

QueryId: 请输入

状态: 请选择

Pid: 请输入

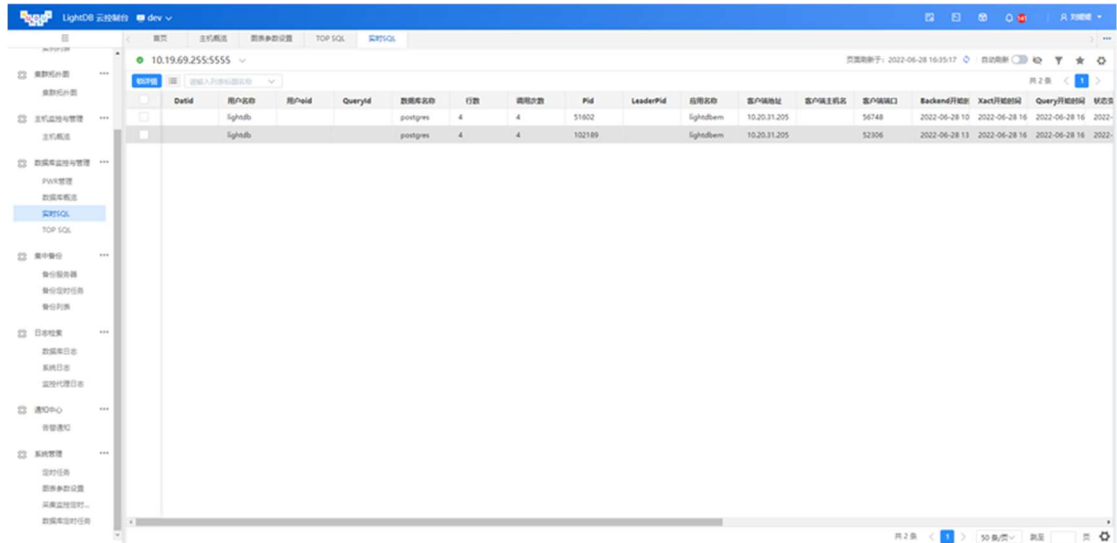
Xact启动时间范围: 选择日期

查询启动时间范围: 选择日期

客户端地址: 请输入

查询 重置

锁详情页面



实时 SQL 表和数据项说明:

对象	类型	说明
It_stat_activity	视图	每个服务器进程将有一行，显示与该进程当前活动相关的信息
username	数据项	登录到这个后端的用户的 OID
queryid		内部哈希码，从语句的解析树计算得来
dbname		数据库名称
rows		语句检索或影响的总行数
calls		语句被执行的次数
datid		这个后端连接到的数据库的 OID
pid		这个后端的进程 ID
leader_pid		并行组组长的进程 ID，如果该进程是并行查询工作者。如果该进程是一个并行组的组长或不参与并行查询，则为 NULL
usesysid		登录到这个后端的用户的 OID
application_name		连接到这个后端的应用的名称
client_addr		连接到这个后端的客户端的 IP 地址。如果这个字段为空，它表示客户端通过服务器机器上的一个 Unix 套接字连接或者这是一个内部进程，如自动清理
client_hostname		已连接的客户端的主机名，由 client_addr 的反向 DNS 查找报告。这个字段将只对 IP 连接非空，并且只有 log_hostname 被启用时才会非空
client_port		客户端用于与此后端通信的 TCP 端口号，如果使用 Unix 套接字，则为-1。如果该字段为空，它表示这是一个内部服务器进程
backend_start		这个进程被启动的时间。对客户端后端来说，这就是客户端连接到服务器的时间
xact_start		这个进程的当前事务被启动的时间，如果没有活动事务则为空。如果当前查询是它的第一个事务，这一列等于 query_start 列
query_start		当前活动查询被开始的时间，如果 state 不是 active，则为上一个查询开始的时间
state_change		state 上一次被改变的时间

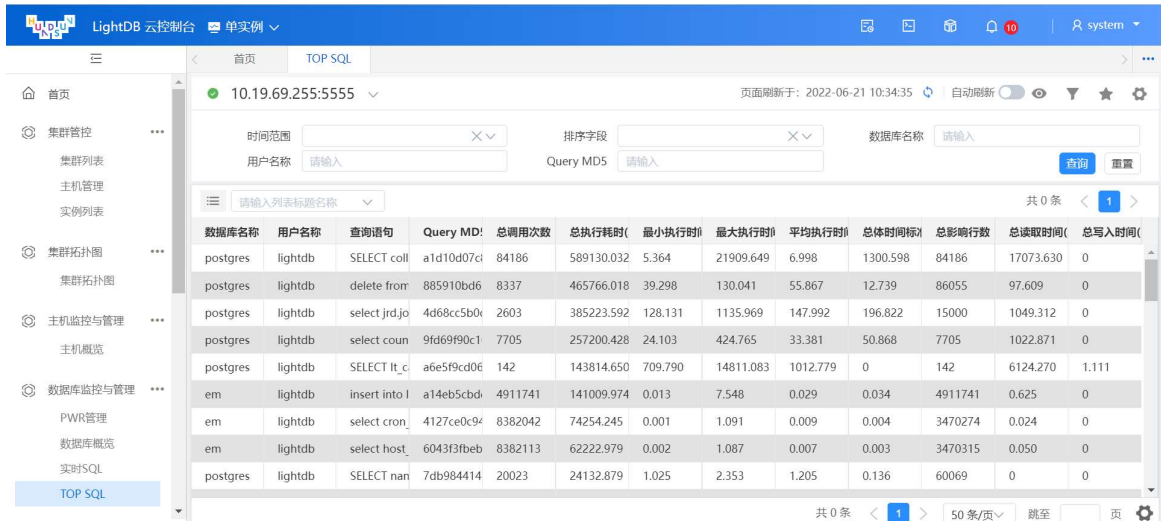
wait_event_type	后端等待的事件类型，如果有的话;否则 NULL
wait_event	如果后端当前正在等待，则等待事件名称，否则为 NULL
state	这个后端的当前总体状态。可能的值为： active : 后端正在执行一个查询。 idle : 后端正在等待一个新的客户端命令。 idle in transaction : 后端在一个事务中,但是当前没有正在执行一个查询。 idle in transaction (aborted) : 这个状态与 idle in transaction 相似,除了在该事务中的一个语句导致了一个错误。 fastpath function call : 后端正在执行一个 fast-path 函数。 disabled : 如果在这个后端中 track_activities 被禁用,则报告这个状态。
backend_xid	这个后端的顶层事务标识符，如果存在
backend_xmin	当前后端的 xmin 范围
backend_type	当前后端的类型。可能的类型为 autovacuum launcher , autovacuum worker , logical replication launcher , logical replication worker , parallel worker , background writer , client backend , checkpointer , startup , walreceiver , walsender and walwriter . 除此以外,由扩展注册的后台 Worker 可能有额外的类型
total_exec_time	执行语句所花费的总时间,以毫秒为单位
query	这个后端最近查询的文本。如果 state 为 active , 这个字段显示当前正在执行的查询。在所有其他状态下,它显示上一个被执行的查询。默认情况下,查询文本会被截断至 1024 个字节,这个值可以通过参数 track_activity_query_size 更改

3.7.4 TOP SQL

TOP SQL 是对最近 24h 查询语句的多维统计排行榜;

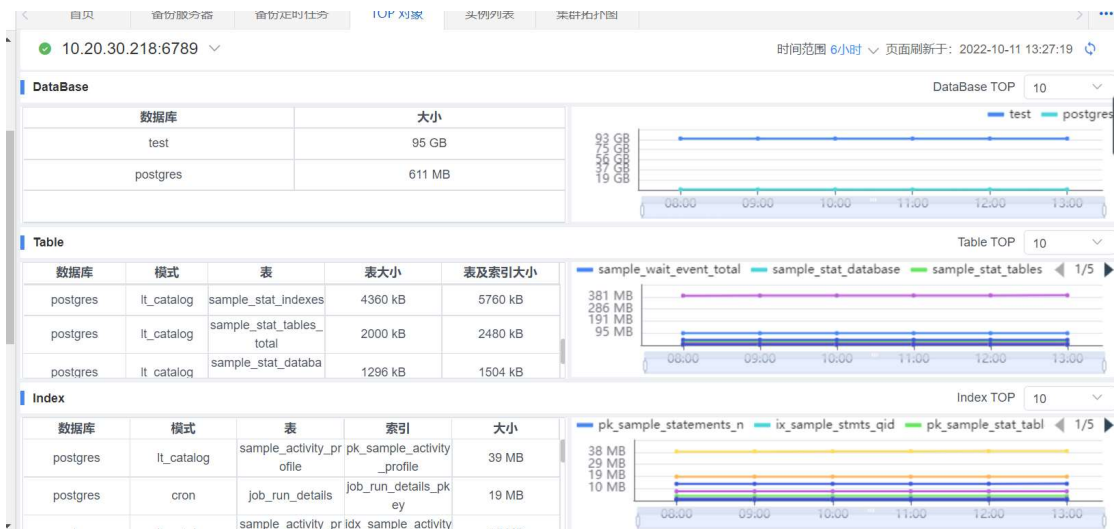
基于 **query md5** 来对语句分组,

主要排行维度有: 总调用次数、总执行耗时、最小执行时间、最大执行时间、平均执行时间、总体时间标准差、总影响行数、总读取时间、总写入时间。



3.7.1 TOP 对象

TOP 对象是对数据库大小，数据库表大小，索引大小进行排行；



3.8 集中备份

集中备份基于 It_probackup 实现数据库实例的数据备份和 WAL 归档；

一个备份服务器配备一个备份实例，目前仅支持备份服务器的备份实例与被备份实例同版本的情况。一个备份服务器可以初始化多个备份目录，一个备份目录下可以添加多个实例（一个实例只允许被添加一次），一个实例可以执行多次备份，增量备份之前必须有成功的全量备份。

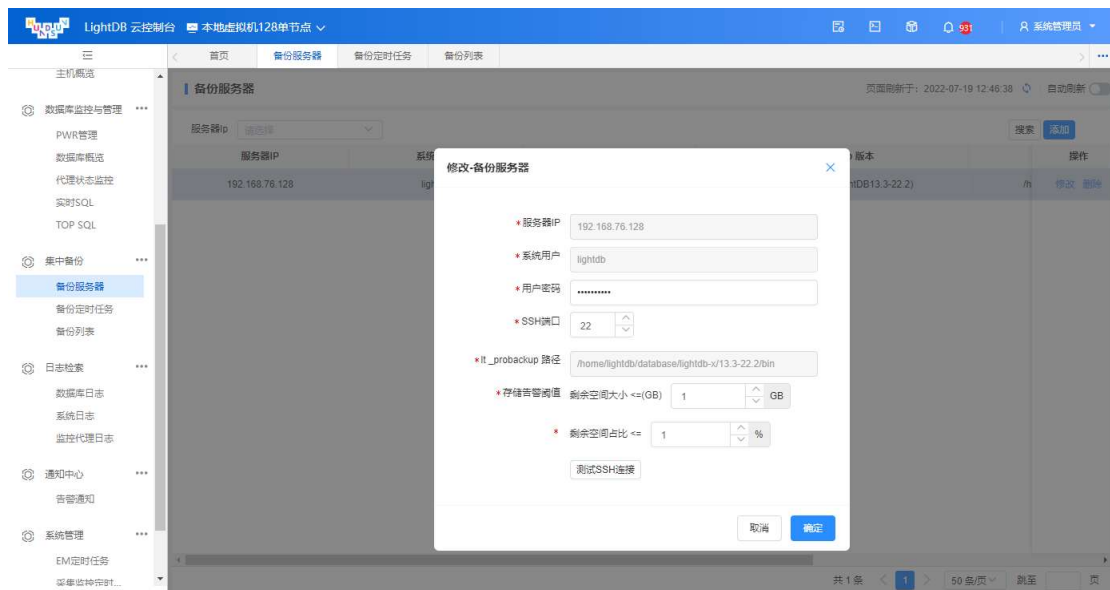
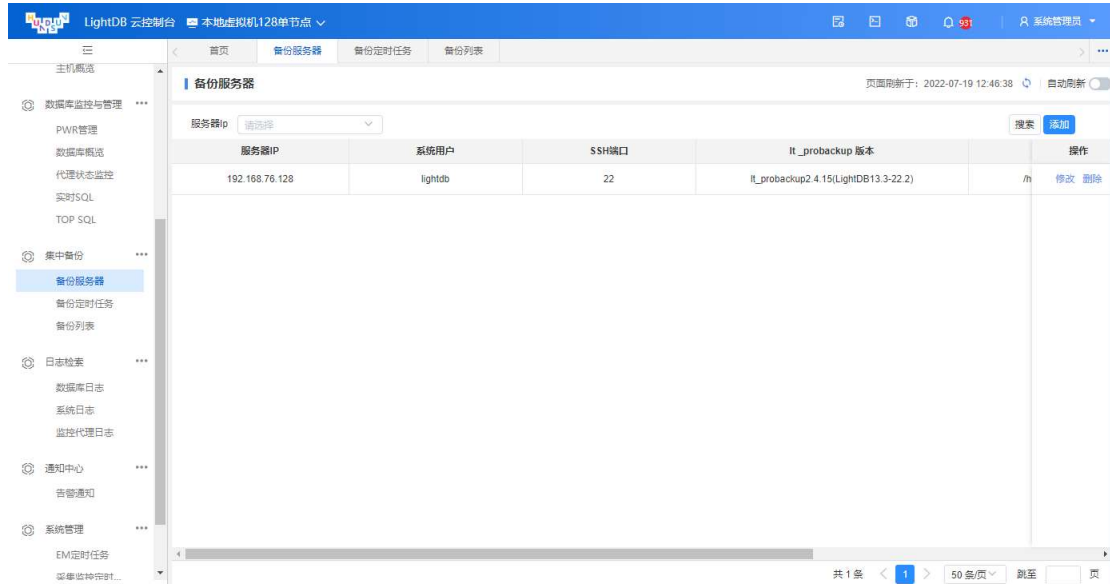
3.8.1 备份服务器

备份服务器是用于存储备份文件的服务器，系统提供自定义存储告警阈值，触发阈值将

会发送告警通知。

添加备份服务器前准备:

1. 参与备份的服务器平台要一致, 如 e17.x87 或者 KylinV10spl.arm。
2. 预先在备份服务器安装 lightdb 实例, 版本与被备份实例需一致。



注意: It_probackup 路径表示 It_probackup 这个命令在备份服务器上的绝对路径(该路径为目录不包含 It_probackup)

3.8.2 备份定时任务

备份定时任务开放定时上线, 仅支持手动执行备份(增量备份前需要已成功全量备份), 支持手动删除备份(本期仅允许从最新备份往前删)。

添加备份任务前准备:

1. 已部署被备份实例。
2. 在被备份实例配置数据库用户 It_probackup 的访问权限和免密, 编辑

\$LTDATA/lt_hba.conf 添加如下内容:

```
host all lt_probackup 10.20.128.128/32 trust
```

字段内容说明:

host: 此记录匹配使用 TCP/IP 进行的连接尝试。

all: 匹配所有的数据库

lt_probackup: 特定的数据库用户名(该用户用于备份)

10.20.128.128/32: 客户端计算机地址。该地址下的客户端允许访问数据库

trust: 无条件允许连接。即允许客户端地址为 10.20.128.128/32 的数据库用户

lt_probackbap 访问(不需要密码)。

```
# "local" is for Unix domain socket connections only
local all all trust
# IPv4 local connections:
host all all 127.0.0.1/32 trust
host all lt_probackup 192.168.76.128/32 trust
host all all 0.0.0.0/0 md5
# IPv6 local connections:
host all all ::1/128 trust
# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication all trust
host replication all 127.0.0.1/32 trust
host replication all ::1/128 trust
```

提示:

未在数据库实例的\$LTDATA/lt_hba.conf 文件中配置数据库用户 lt_probackup 访问免密, 则为该数据库实例添加备份定时任务失败, 失败提示信息为“需要在\$LTDATA/lt_hba.conf 配置备份用户 lt_probackup 的访问权限和免密”。

22.1 版本的数据库实例, 为该实例添加备份定时任务失败, 错误提示信息中\$LTDATA 对应 22.1 版本的数据库实例为\$PGDATA。lt_hba.conf 对应 22.1 版本的数据库实例为 pg_hba.conf。

3. 在被备份主节点配置连续 WAL 归档(用于备份归档), 编辑 \$LTDATA/lightdb.conf 配置文件添加以下内容(三个配置项):
 - a) 配置项 wal_level 为 replica 或者 logical。(wal_level 取值范围 minimal, replica(默认), or logical)
 - b) 在主节点配置连续 WAL 归档, 配置项 archive_mode 必须设置为 on 或者 always(如果想在备份节点也设置归档需要设置为 always)
 - c) 配置项 archive_command
 - i. 配置项 archive_command 和 archive_mode 在数据库单机版需要配置, 数据库高可用版不需要配置。该功能暂不支持数据库分布式版本。
 - ii. 目前在添加备份定时任务的过程中在\$LTHOME/bin 目录下生成一个 shell 脚本 lt_probackup_for_em.sh (不再需要单独在配置项 archive_command 中显示设置 lt_probackup archive-push 子命令)。
 - iii. archive_command='... && \$LTHOME/bin/lt_probackup_for_em.sh %p %f'
4. 继续编辑 \$LTDATA/lightdb.conf 文件, 调整配置项 archive_timeout =300 #自动归档的间隔时间, 单位为秒, 默认 5 分钟, 建议 1 分钟。
5. 编辑 \$LTDATA/lightdb.conf 文件后, 需要**重启数据库实例服务** (因为配置项 archive_command 重载(reload)无法生效, 必须重启数据库实例) 保证配置生效。
6. 使用命令行 **itsql 连接到数据库实例, 查看配置项(archive_mode、wal_level、archive_command、archive_timeout)内容:**

```
show archive_mode;
show wal_level;
show archive_timeout;
show archive_command;
```

7. 配置备份服务器主机 与 被备份实例主机之间 SSH 双向免密互通；

```
ssh-keygen -t rsa -P "" -f ~/.ssh/id_rsa
ssh-copy-id lightdb@backup_server_ip
```

```
ssh-keygen -t rsa -P "" -f ~/.ssh/id_rsa
ssh-copy-id lightdb@instance_server_ip
```

8. 远程备份的并行数受 SSH 连接限制，因为 SSH 连接数默认上限为 10，当备份并行数超过 10 时，需要修改被备份实例所在主机的 SSH 连接数限制，参考如下（超级用户执行）：

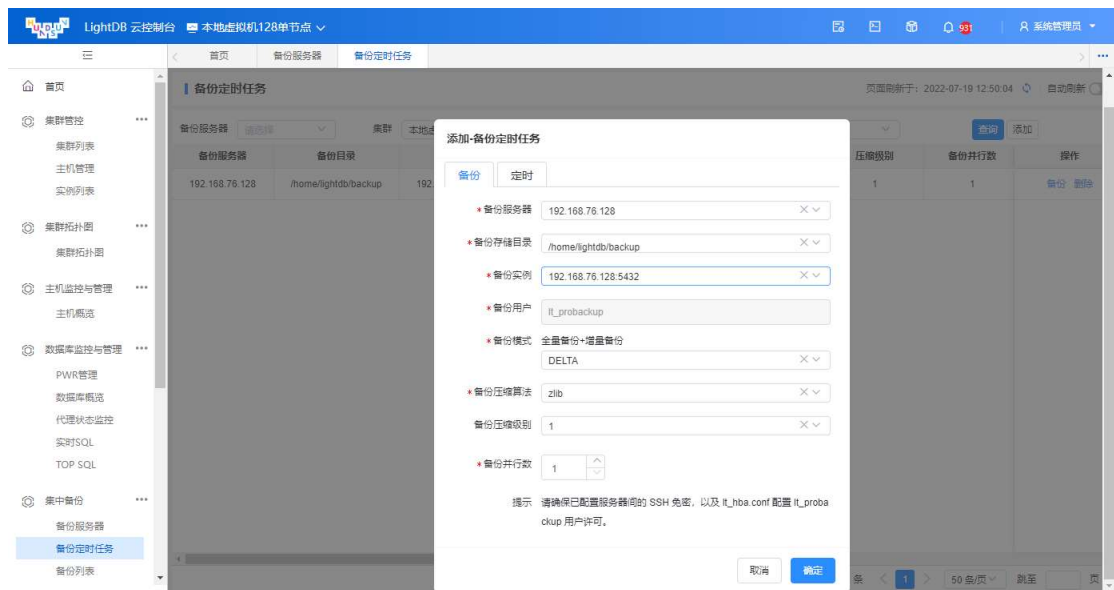
```
vim /etc/ssh/sshd_config
```

修改 MaxStartup、MaxSessions 参数，将最小连接数设置为 80（建议值为最大并行数+10）。保存后退出。

重启 sshd 服务：systemctl restart sshd 或者 service sshd reload。

```
MaxSessions 80
```

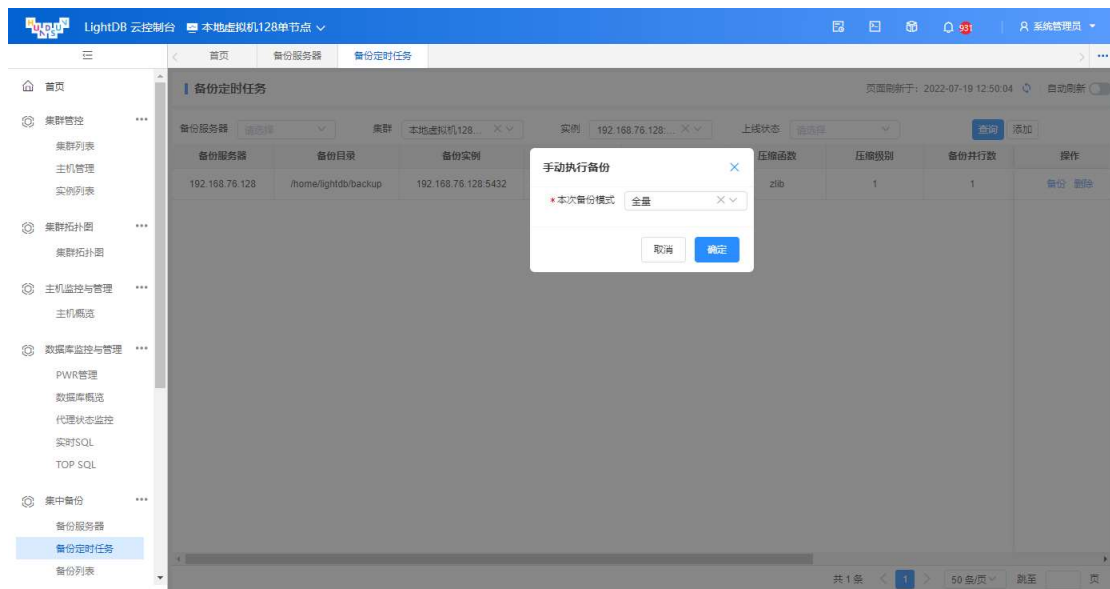
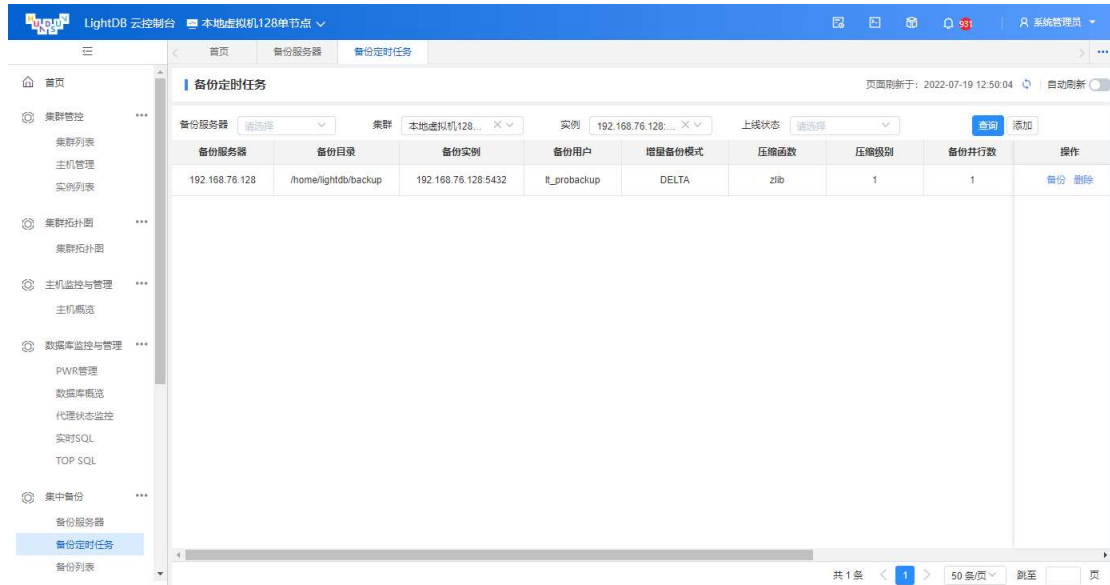
```
MaxStartup 80
```



备份模式(增量)说明：

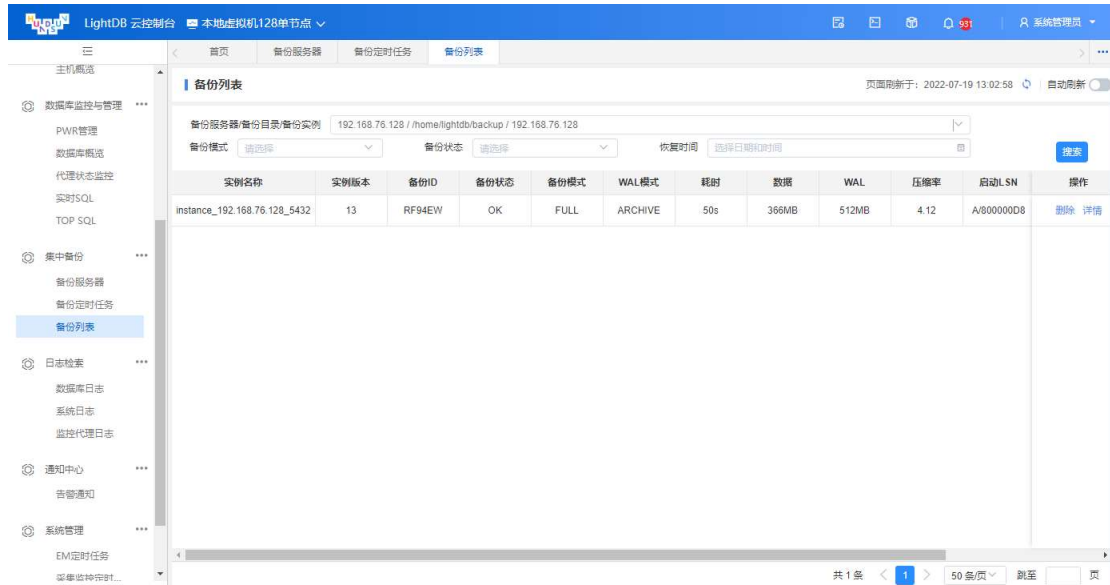
DELTA: 在这种模式下，lt_probackup 命令 读取 数据库实例的 DATA 目录中的所有数据文件，并仅复制自上次备份以来发生更改的那些页面。

PAGE: lt_probackup 命令会从前一次完整备份或增量备份开始扫描归档文件中的所有 WAL 文件。新创建的备份只包含 WAL 记录中提到的页面。这要求自上一次备份以来的所有 WAL 文件都存在于 WAL 存档中。要求配置持续 WAL 归档。



3.8.3 备份列表

备份列表页面展示备份服务器-备份目录下的备份列表，点击“详情”可以查看备份详情。



备份状态说明:

OK: 备份完成并且有效

DONE: 备份已完成，但尚未验证。

RUNNING: 备份正在进行中。

MERGING: 备份正在合并。

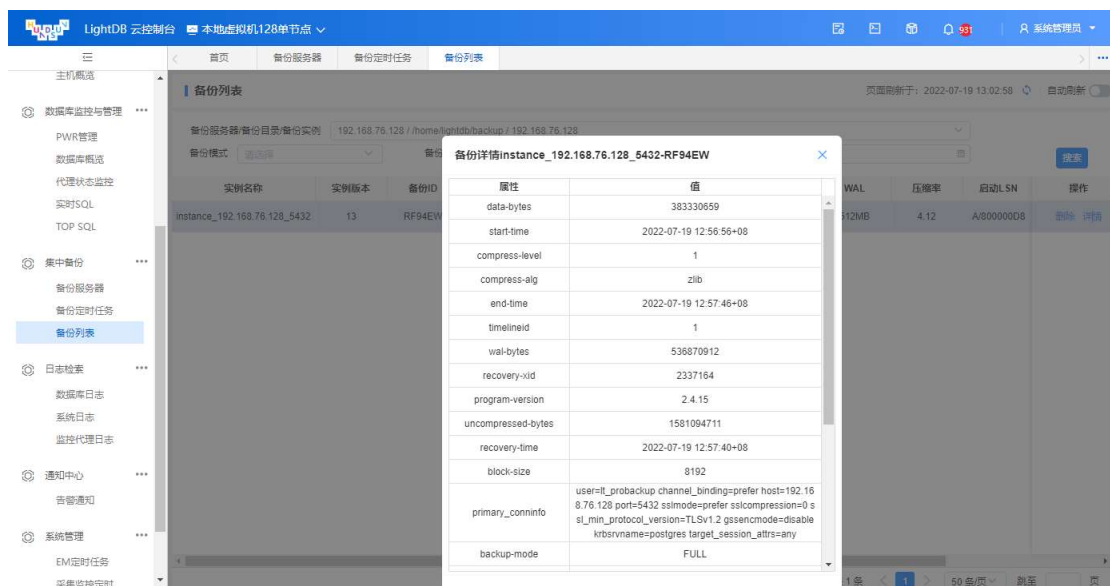
MERGED: 备份数据文件已成功合并，但其元数据正在更新过程中。只有全量备份才能具有此状态。

CORRUPT: 一些备份文件已经破坏。

DELETING: 正在删除备份文件。

ERROR: 备份由于意外错误而中止。

ORPHAN: 备份无效，因为其父备份之一已损坏或丢失。



3.8.4 备份删除

备份删除策略：指定备份 id 记录删除。

实例名称	实例版本	备份ID	备份状态	备份模式	WAL模式	耗时	数据	WAL	压缩率	启动LSN	停止LSN	恢复时间	操作
instance_10.15	13	RJY7BV	OK	DELTA	ARCHIVE	1m:27s	3518kB	1024MB	1.00	4/1C28	4/4023C7E8	2022-10-18 20	删除 详情
instance_10.15	13	RJY7BR	OK	DELTA	ARCHIVE	1m:40s	3518kB	1536MB	1.00	4/50	4/600001B0	2022-10-18 20	详情
instance_10.15	13	RJY6RO	OK	FULL	ARCHIVE	3m:44s	326MB	512MB	1.00	3/80000120	3/A00BACA0	2022-10-18 20	详情
instance_10.15	13	RJY6QL	ERROR	FULL	ARCHIVE	5m:13s	0	0	1.00	3/200000F8	0/0	---	详情

3.8.5 恢复列表

恢复前准备：

恢复实例所在服务器、备份实例所在服务器、备份实例的备份服务器，三个服务器主机需要满足以下：

1. 主机之间配置 SSH 双向免密互通；

```
ssh-keygen -t rsa -P "" -f ~/.ssh/id_rsa  
ssh-copy-id lightdb@backup_server_ip
```

```
ssh-keygen -t rsa -P "" -f ~/.ssh/id_rsa  
ssh-copy-id lightdb@restore_instance_server_ip
```

2. 远程还原的并行线程数受 SSH 连接限制，因为 SSH 连接数默认上限为 10，当并行数超过 10 时，需要修改被恢复实例所在主机的 SSH 连接数限制，参考如下（超级用户执行）：

```
vim /etc/ssh/sshd_config
```

#修改 MaxStartup、MaxSessions 参数，将最小连接数设置为 80（建议值为最大并行数+10）。保存后退出。

```
MaxSessions 80
```

```
MaxStartup 80
```

#重启 sshd 服务：

```
systemctl restart sshd 或者 service sshd reload。
```

3. 主机之间的 ntp 服务开启并配置为相同的时间服务器；（详情参考《LightDB 数据库安装手册》）

#以下 root 用户操作：

```
# 1. 生成配置文件 $LTHOME/etc/ntp/ntp.conf 并初始化配置，后续也可用于修改  
sh $LTHOME/etc/ntp/generate_ntp_conf.sh $PRIMARY_IP $LOCAL_IP
```

#2.启动（已经启动的需要停止后再启动）

\$LTHOME/tools/bin/ntpd -g -c \$LTHOME/etc/ntp/ntp.conf

3.查看 ntp 同步状态（不需要 root）

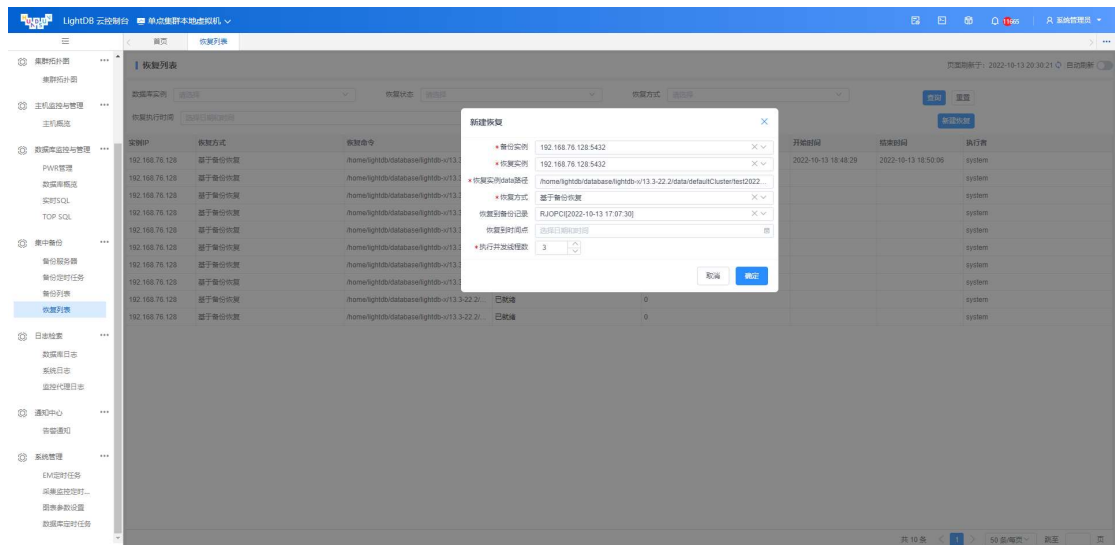
[lightdb@lightdb-2 script]\$ \$LTHOME/tools/bin/ntpq -p

remote	refid	st	t	when	poll	reach	delay	offset	jitter
10.20.30.218	10.19.30.3	6	u	45	64	3	0.321	-1.046	0.500

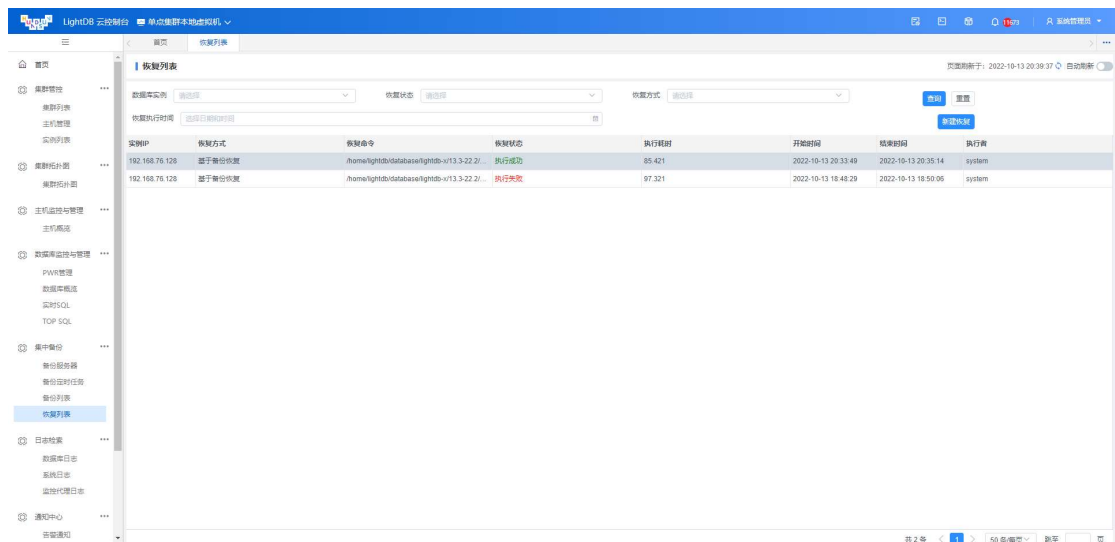
执行恢复：

恢复方式有两种：基于时间点恢复、基于备份恢复，选择基于时间点，则需要指定恢复的时间点，选择基于备份，则需要指定恢复的备份 ID；

基于备份的恢复方式如下：



在当前页面可以查询本次执行恢复的状态：



3.9 日志检索

3.9.1 数据库日志

数据库日志定时从 LightDB 数据库日志文件中采集。日志文件地址\$LTDDATA/log/。

```
[lightdb@localhost data]$  
[lightdb@localhost data]$ cd $PGDATA/log  
[lightdb@localhost log]$ ls -l  
总用量 254644  
-rw----- 1 lightdb lightdb 208329 1月 25 14:36 lightdb-2022-01-25_143320.log  
-rw----- 1 lightdb lightdb 7770 1月 25 14:37 lightdb-2022-01-25_143652.log  
-rw----- 1 lightdb lightdb 6069161 1月 26 00:00 lightdb-2022-01-25_143826.log  
-rw----- 1 lightdb lightdb 13121824 1月 27 00:00 lightdb-2022-01-26_000000.log  
-rw----- 1 lightdb lightdb 71435518 1月 28 00:00 lightdb-2022-01-27_000000.log  
-rw----- 1 lightdb lightdb 127025056 1月 29 00:00 lightdb-2022-01-28_000000.log  
-rw----- 1 lightdb lightdb 41330823 1月 29 08:07 lightdb-2022-01-29_000000.log  
[lightdb@localhost log]$  
[lightdb@localhost log]$
```

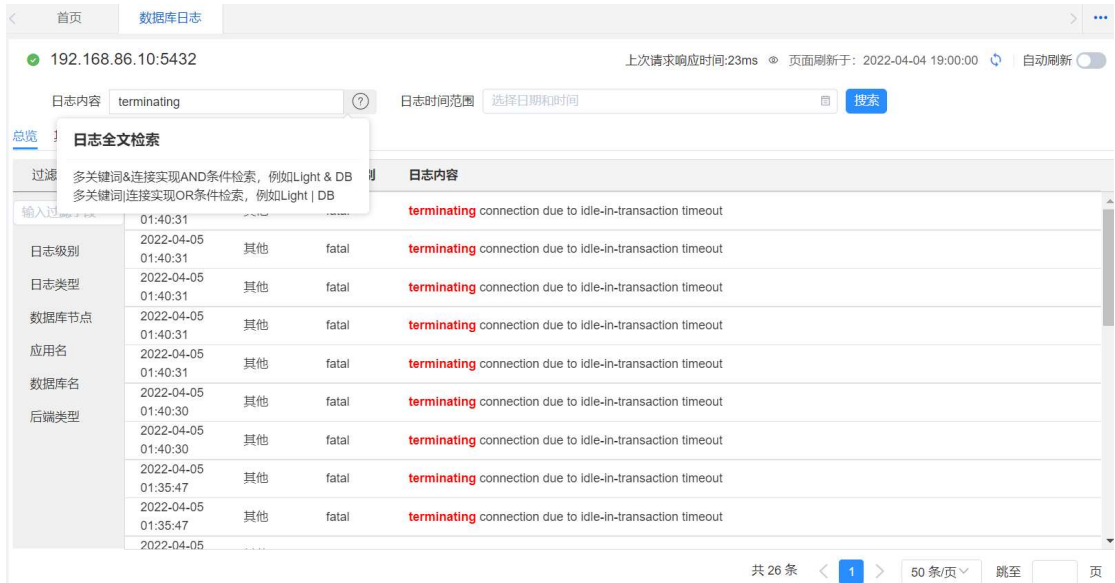
数据库日志采集以行为单位进行读取，以段为单位进行采集。每段的规定是以日期开头到下一个日期开头之前的所有行的内容为一行。

The screenshot shows a web-based log search interface. At the top, there are search filters for IP address (10.188.131.245:5432), search content, and time range. Below the filters is a table with columns: ID, 时间戳 (Timestamp), 日志类型 (Log Type), 日志级别 (Log Level), 日志消息 (Log Message), and 解析计划 (Parsing Plan). The table contains 391 entries, with the first few rows showing log messages about failed archive commands and process exits. On the left side, there are filter tabs for '过滤字段' (Filter Fields) and '日志级别' (Log Level). At the bottom, there are pagination controls showing '共 391 条' (Total 391 items) and '50 条/页' (50 items per page).

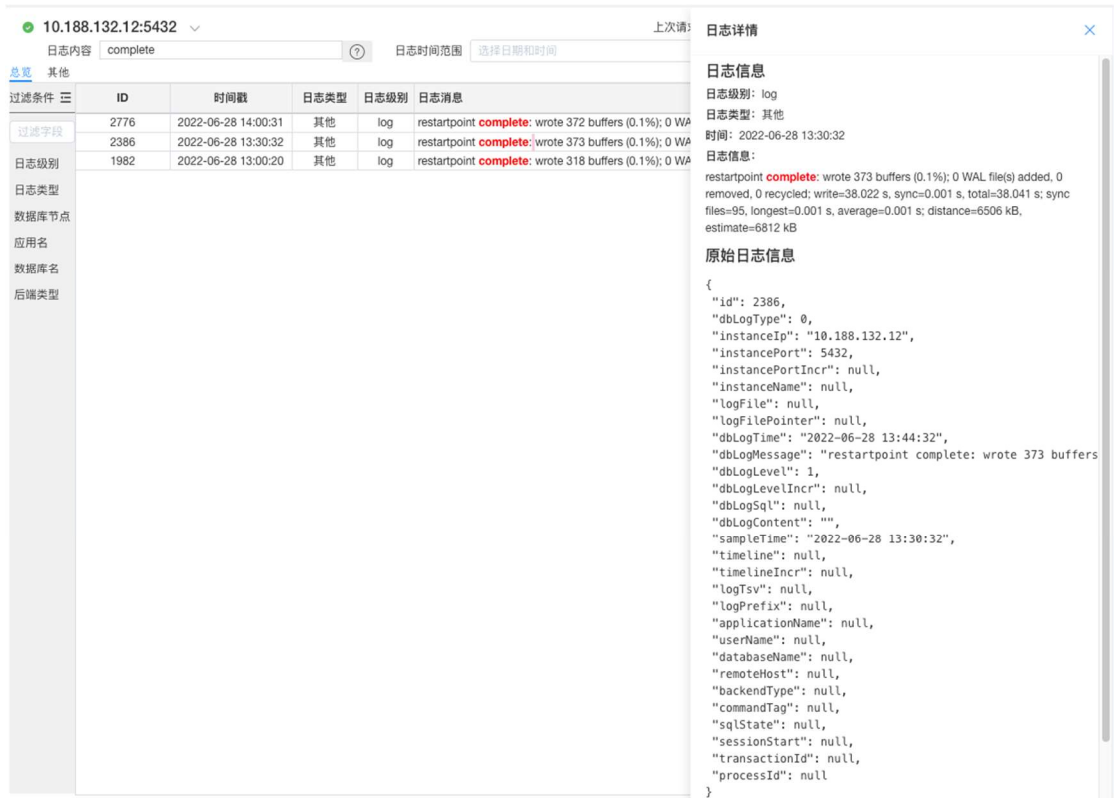
过滤条件	ID	时间戳	日志类型	日志级别	日志消息	解析计划
过滤字段	480	2022-06-28 11:15:44	归档	log	archiver process (PID 52273) exited with exit code 1	
日志级别	479	2022-06-28 11:15:44	归档	fatal	archive command failed with exit code 127	
日志类型	488	2022-06-28 11:16:14	归档	log	The failed archive command was: rm -f /usr/local/lightdb/lightdb-x/13.3-22.2/archive/000000010000000000000001 && ...	
数据库节点	487	2022-06-28 11:16:14	归档	fatal	archiver process (PID 52730) exited with exit code 1	
应用名	478	2022-06-28 11:15:44	归档	log	archive command failed with exit code 127	
数据库名	477	2022-06-28 11:15:44	归档	fatal	The failed archive command was: rm -f /usr/local/lightdb/lightdb-x/13.3-22.2/bin/lt_probackup_for_em.sh: No such file or directory	
后端类型	446	2022-06-28 11:13:54	归档	log	archiver process (PID 50536) exited with exit code 1	
	445	2022-06-28 11:13:54	归档	fatal	archive command failed with exit code 127	
	442	2022-06-28 11:13:40	归档	log	The failed archive command was: rm -f /usr/local/lightdb/lightdb-x/13.3-22.2/archive/000000010000000000000001 && ...	
	441	2022-06-28 11:13:40	归档	fatal	archiver process (PID 50275) exited with exit code 1	
	472	2022-06-28 11:15:12	归档	log	archive command failed with exit code 127	
	471	2022-06-28 11:15:12	归档	fatal	The failed archive command was: rm -f /usr/local/lightdb/lightdb-x/13.3-22.2/bin/lt_probackup_for_em.sh: No such file or directory	
	464	2022-06-28 11:14:42	归档	log	archiver process (PID 51509) exited with exit code 1	
	463	2022-06-28 11:14:42	归档	fatal	The failed archive command was: rm -f /usr/local/lightdb/lightdb-x/13.3-22.2/archive/000000010000000000000001 && ...	
	397	2022-06-28 11:10:02	归档	log	archiver process (PID 51217) exited with exit code 1	
	396	2022-06-28 11:10:02	归档	fatal	archive command failed with exit code 127	
	482	2022-06-28 11:15:50	归档	log	The failed archive command was: rm -f /usr/local/lightdb/lightdb-x/13.3-22.2/bin/lt_probackup_for_em.sh: No such file or directory	
	481	2022-06-28 11:15:50	归档	fatal	archiver process (PID 52385) exited with exit code 1	
	456	2022-06-28 11:14:10	归档	log	archive command failed with exit code 127	
	455	2022-06-28 11:14:10	归档	fatal	The failed archive command was: rm -f /usr/local/lightdb/lightdb-x/13.3-22.2/archive/000000010000000000000001 && ...	
	385	2022-06-28 11:09:00	归档	log	archiver process (PID 42820) exited with exit code 1	
	384	2022-06-28 11:09:00	归档	fatal	archive command failed with exit code 127	
	486	2022-06-28 11:16:14	归档	log	The failed archive command was: rm -f /usr/local/lightdb/lightdb-x/13.3-22.2/archive/000000010000000000000001 && ...	

日志搜索查询过滤条件如下：

日志搜索：根据关键词进行搜索，关键词实现了全文检索；



日志信息详情：双击单条日志可查看日志消息详情



慢 sql 查询计划：可通过点击慢日志在表格最右一栏的解析按钮查询该条慢日志的查询计划

10.20.31.206:5432 上次请求响应时间:12ms 页面刷新于: 2022-06-30 14:23:01 自动刷新

日志内容 搜索内容 日志时间范围 选择日期和时间 搜索

总览 其他 慢sql

过滤条件	ID	时间戳	日志类型	日志级别	日志消息	解析计划
过滤字段	5731	2022-06-30 13:47:35	慢sql	log	duration: 0.051 ms plan: { "Query Text": "update cron.job_run_details set status = \$1, return_message = \$2, end_time = ...	解析
日志级别	5759	2022-06-30 13:48:03	其他	context	SQL statement "update cron.job_run_details set status = \$1, return_message = \$2, end_time = \$3 where runid = \$4"	解析
日志类型	5730	2022-06-30 13:47:35	慢sql	log	duration: 0.028 ms plan: { "Query Text": "update cron.job_run_details set status = \$1, start_time = \$2 where runid = \$3"...	解析
数据库节点	5758	2022-06-30 13:48:03	其他	context	SQL statement "update cron.job_run_details set status = \$1, start_time = \$2 where runid = \$3"	解析
应用名	5729	2022-06-30 13:47:35	慢sql	log	duration: 0.052 ms plan: { "Query Text": "update cron.job_run_details set job_pid = \$1, status = \$2 where runid = \$3", "...	解析
数据库名	5757	2022-06-30 13:48:03	其他	context	SQL statement "update cron.job_run_details set job_pid = \$1, status = \$2 where runid = \$3"	解析
后端类型	5728	2022-06-30 13:47:35	慢sql	log	duration: 0.033 ms plan: { "Query Text": "update cron.job_run_details set status = \$1 where runid = \$2", "Plan": { "Nod..."	解析
	5756	2022-06-30 13:48:03	其他	context	SQL statement "update cron.job_run_details set status = \$1 where runid = \$2"	解析
	5727	2022-06-30 13:47:35	慢sql	log	duration: 0.038 ms plan: { "Query Text": "insert into cron.job_run_details (jobid, runid, database, username, command, ...	解析
	5755	2022-06-30 13:48:03	其他	context	SQL statement "insert into cron.job_run_details (jobid, runid, database, username, command, status) values (\$1,\$2,\$3,..."	解析
	5726	2022-06-30 13:47:35	慢sql	log	duration: 0.018 ms plan: { "Query Text": "select timezone from cron.it_job_ext where jobid = 2", "Plan": { "Node Type": ...	解析
	5754	2022-06-30 13:48:03	其他	context	SQL statement "select timezone from cron.it_job_ext where jobid = 2"	解析
	5725	2022-06-30 13:47:35	慢sql	log	duration: 0.038 ms plan: { "Query Text": "update cron.job_run_details set status = \$1, return_message = \$2, end_time = ...	解析
	5753	2022-06-30 13:48:03	其他	context	SQL statement "update cron.job_run_details set status = \$1, return_message = \$2, end_time = \$3 where runid = \$4"	解析
	5724	2022-06-30 13:47:35	慢sql	log	duration: 0.024 ms plan: { "Query Text": "update cron.job_run_details set status = \$1, start_time = \$2 where runid = \$3"...	解析
	5752	2022-06-30 13:48:03	其他	context	SQL statement "update cron.job_run_details set status = \$1, start_time = \$2 where runid = \$3"	解析
	5723	2022-06-30 13:47:35	慢sql	log	duration: 0.046 ms plan: { "Query Text": "update cron.job_run_details set job_pid = \$1, status = \$2 where runid = \$3", "...	解析
	5751	2022-06-30 13:48:03	其他	context	SQL statement "update cron.job_run_details set job_pid = \$1, status = \$2 where runid = \$3"	解析
	5722	2022-06-30 13:47:35	慢sql	log	duration: 0.037 ms plan: { "Query Text": "update cron.job_run_details set status = \$1 where runid = \$2", "Plan": { "Nod..."	解析
	5750	2022-06-30 13:48:03	其他	context	SQL statement "update cron.job_run_details set status = \$1 where runid = \$2"	解析
	5127	2022-06-30 13:47:04	慢sql	log	duration: 0.026 ms plan: { "Query Text": "update cron.job_run_details set status = \$1 where runid = \$2", "Plan": { "Nod..."	解析
	4934	2022-06-30 13:47:00	其他	context	SQL statement "update cron.job_run_details set status = \$1 where runid = \$2"	解析
	5721	2022-06-30 13:47:35	慢sql	log	duration: 0.034 ms plan: { "Query Text": "insert into cron.job_run_details (jobid, runid, database, username, command, ...	解析
	5749	2022-06-30 13:48:03	其他	context	SQL statement "insert into cron.job_run_details (jobid, runid, database, username, command, status) values (\$1,\$2,\$3,..."	解析
	5720	2022-06-30 13:47:35	慢sql	log	duration: 0.017 ms plan: { "Query Text": "select timezone from cron.it_job_ext where jobid = 2", "Plan": { "Node Type": ...	解析
	5126	2022-06-30 13:47:04	慢sql	log	duration: 0.034 ms plan: { "Query Text": "insert into cron.job_run_details (jobid, runid, database, username, command, ...	解析
	5748	2022-06-30 13:48:03	其他	context	SQL statement "select timezone from cron.it_job_ext where jobid = 2"	解析
	4933	2022-06-30 13:47:00	其他	context	SQL statement "insert into cron.job_run_details (jobid, runid, database, username, command, status) values (\$1,\$2,\$3,..."	解析
	5125	2022-06-30 13:47:04	慢sql	log	duration: 0.016 ms plan: { "Query Text": "select timezone from cron.it_job_ext where jobid = 2", "Plan": { "Node Type": ...	解析
	4932	2022-06-30 13:47:00	其他	context	SQL statement "select timezone from cron.it_job_ext where jobid = 2"	解析
	5719	2022-06-30 13:47:35	慢sql	log	duration: 0.035 ms plan: { "Query Text": "update cron.job_run_details set status = \$1, return_message = \$2, end_time = ...	解析
	5747	2022-06-30 13:48:03	其他	context	SQL statement "update cron.job_run_details set status = \$1, return_message = \$2, end_time = \$3 where runid = \$4"	解析
	5124	2022-06-30 13:47:04	慢sql	log	duration: 0.035 ms plan: { "Query Text": "update cron.job_run_details set status = \$1, return_message = \$2, end_time = ...	解析
	4931	2022-06-30 13:47:00	其他	context	SQL statement "update cron.job_run_details set status = \$1, return_message = \$2, end_time = \$3 where runid = \$4"	解析

首页 数据库日志 5727的执行计划解析器 添加

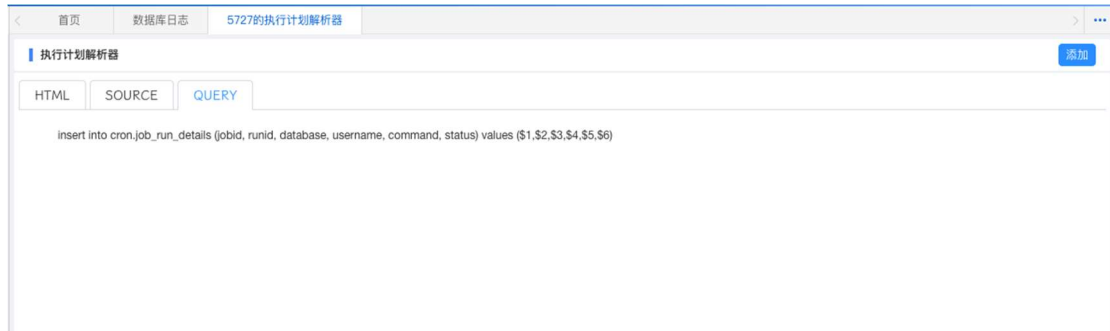
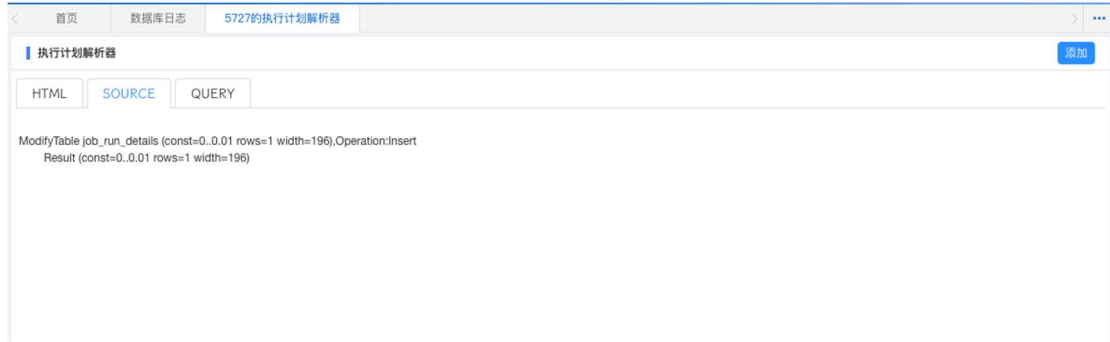
执行计划解析器

HTML SOURCE QUERY

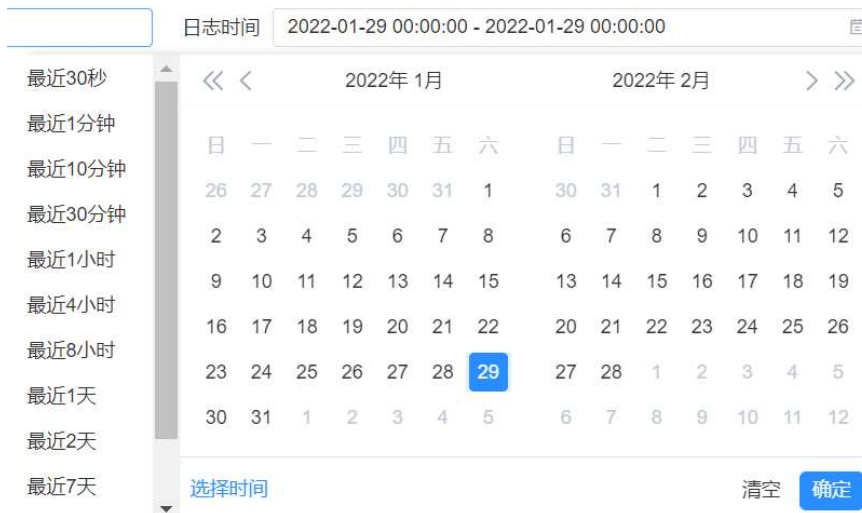
序列	exclusive	inclusive	rows x	rows	loops	node
1	0	0	↓0.0	0	0	▼ModifyTable job_run_details (const=0.0.01 rows=1 width=196),Operation:Insert
2	0	0	↓0.0	0	0	Result (const=0.0.01 rows=1 width=196)

Planning time : 0 ms
Execution time :0 ms

该查询计划包含 source 数据和该条日志包含的 query 语句



日志时间：可以选择日志时间范围进行搜索，先选择日期：



再选择时间

日志时间 2022-01-29 00:00:00 - 2022-01-29 00:00:00

	开始时间			结束时间		
最近30秒	00	00	00	00	00	00
最近1分钟	01	01	01	01	01	01
最近10分钟	02	02	02	02	02	02
最近30分钟	03	03	03	03	03	03
最近1小时	04	04	04	04	04	04
最近4小时	05	05	05	05	05	05
最近8小时	06	06	06	06	06	06
最近1天	07	07	07	07	07	07
最近2天	08	08	08	08	08	08
最近7天						

选择日期 清空 确定

日志级别：日志级别包括 fatal、notice、info、error 和 log

日志级别(总共2个值) 输入过滤值

值	事件数	占比	<input type="checkbox"/>
log	197	48.88%	<input type="checkbox"/>
fatal	206	51.12%	<input type="checkbox"/>

确认

日志类型：日志类型包括临时文件、慢 SQL、其他、auto vacuum 和自动检查点

日志类型(总共3个值) 输入过滤值

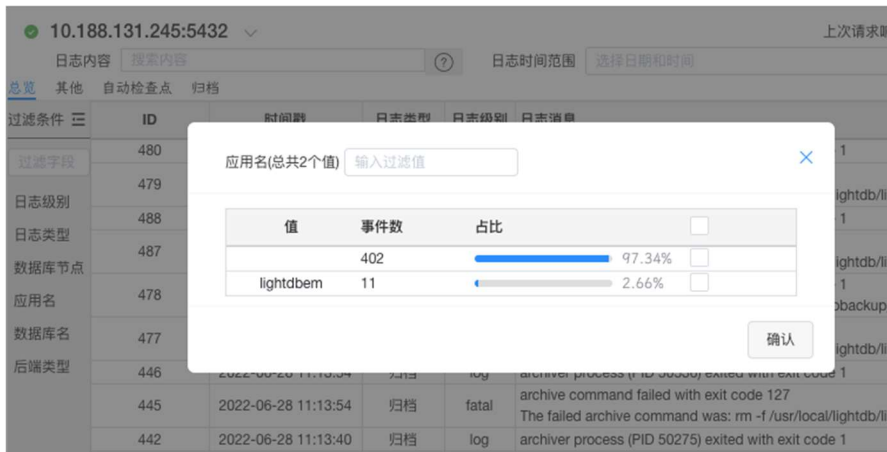
值	事件数	占比	<input type="checkbox"/>
其他	11	2.74%	<input type="checkbox"/>
自动检查点	2	0.50%	<input type="checkbox"/>
归档	388	96.76%	<input type="checkbox"/>

确认

数据库节点：根据实例 IP 和实例端口进行精确搜索



应用名：连接数据库的应用名称




数据库名：连接的数据库名称

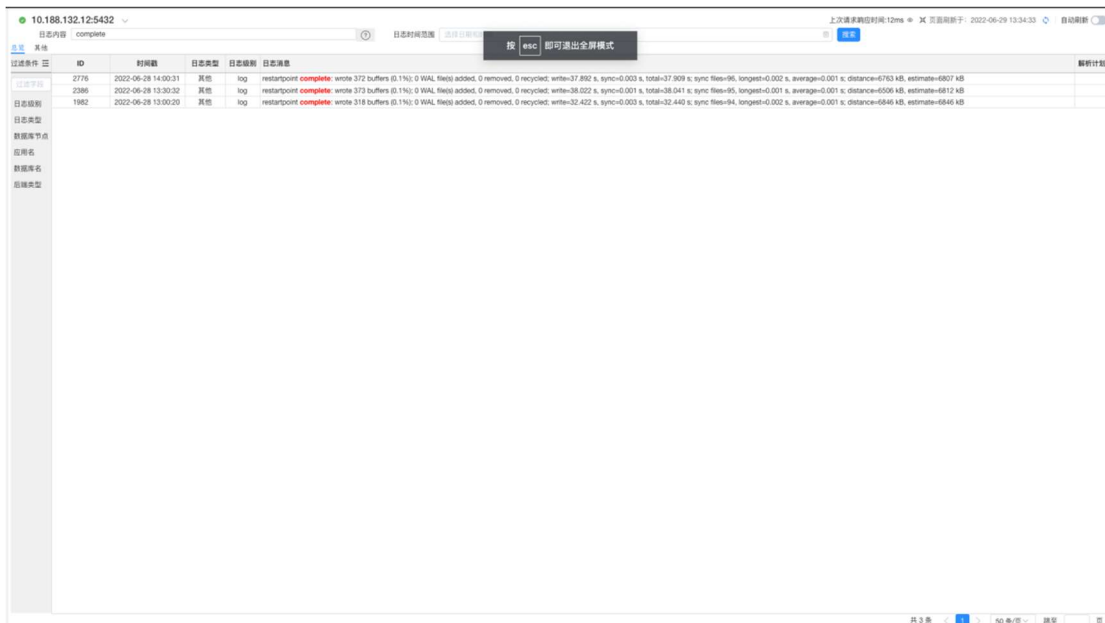


后端类型：后端类型对应视图 `pg_stat_activity` 中的 `backend_type` 列。但是其他类型可能会出现在日志中而不显示在该视图中。



日志全屏功能：

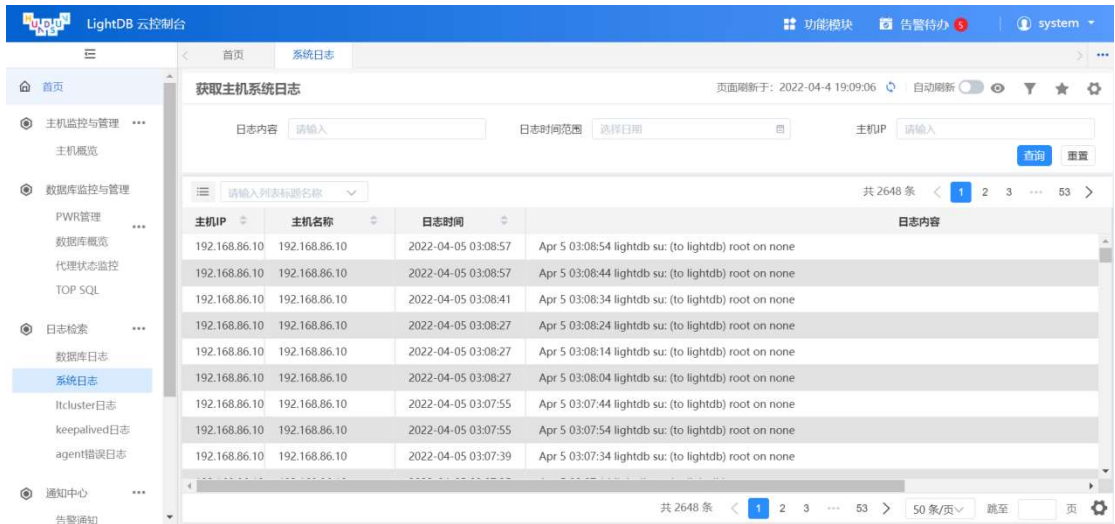
点击上方  按钮可以使当前窗口全屏展示



点击右上角的  或者 `esc` 都可以退出全屏状态。

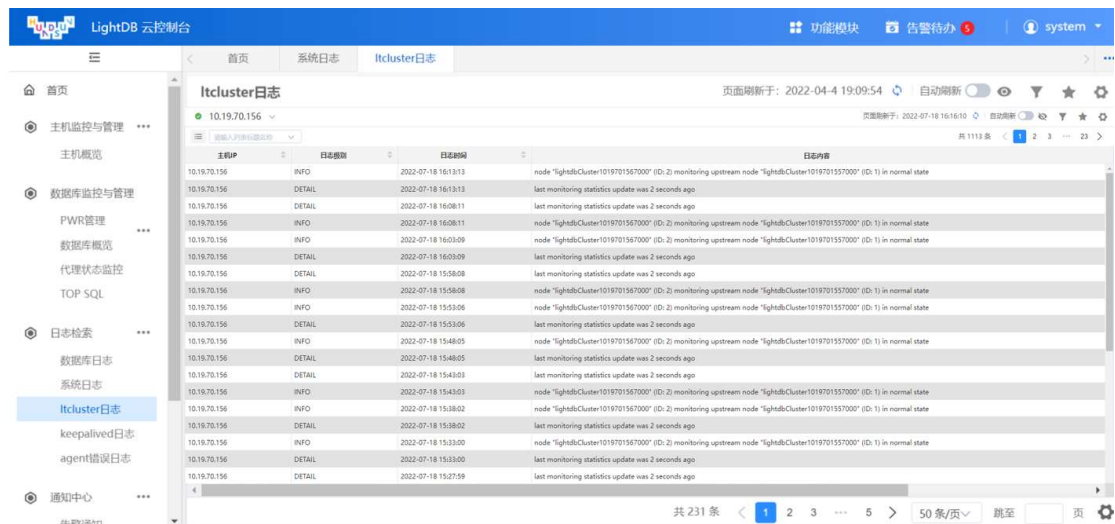
3.9.2 系统日志

系统日志从 `/var/log/messages` 中采集所有日志。



3.9.3 Itcluster 日志

Itcluster 日志记录 Itcluster 组件生成的日志，Itcluster 日志只有在集群环境才有。非集群环境看不到 Itcluster 日志菜单。



Itcluster 用于管理 LightDB 服务器集群中的复制和故障转移。它通过设置备用服务器、监控复制和执行管理任务（如故障转移或手动切换操作）的工具增强了 LightD 的内置热备用功能。Itcluster 为 LightDB 的内置复制机制提供了高级支持，例如通过复制协议的级联复制、时间线切换和基本备份。

3.9.4 HA 日志

HA 日志记录 keepalived 组件生成的日志，HA 日志只有在集群环境才有。非集群环境看不到 keepalived 日志菜单。

告警代码	告警名称	通知信息	通知方式	消息结果	发送时间
1	cpu总使用率告警	当前主机 10.19.69.193,cpu总使用率 80.21%	站内通知	已发送	2022-01-24 00:05:31
1	cpu总使用率告警	当前主机 10.19.69.193,cpu总使用率 80.36%	站内通知	已发送	2022-01-24 00:05:01
1	cpu总使用率告警	当前主机 10.19.69.193,cpu总使用率 81.66%	站内通知	已发送	2022-01-24 00:04:31
1	cpu总使用率告警	当前主机 10.19.69.193,cpu总使用率 82.79%	站内通知	已发送	2022-01-24 00:04:01
1	cpu总使用率告警	当前主机 10.19.69.193,cpu总使用率 81.22%	站内通知	已发送	2022-01-24 00:03:01
1	cpu总使用率告警	当前主机 10.19.69.193,cpu总使用率 83.29%	站内通知	已发送	2022-01-24 00:02:31
1	cpu总使用率告警	当前主机 10.19.69.193,cpu总使用率 84.35%	站内通知	已发送	2022-01-24 00:01:31
1	cpu总使用率告警	当前主机 10.19.69.193,cpu总使用率 95.99%	站内通知	已发送	2022-01-24 00:00:01
1	cpu总使用率告警	当前主机 10.19.69.193,cpu总使用率 80.88%	站内通知	已发送	2022-01-23 23:59:31
1	cpu总使用率告警	当前主机 10.19.69.193,cpu总使用率 82.49%	站内通知	已发送	2022-01-23 23:57:01

主机和数据库告警阈值配置表：

告警名称	用途	监控间隔	阈值	备注
cpu 总使用率告警	cpu 总使用率超过 80%	根据数据采集周期而定，实时	cpu_total_usage>80	lem_host_cpz 表 cpu_total_usage 字段
磁盘使用率告警	磁盘使用率告警超过 80%	同上	disk_usage>80	lem_host_diskfiles system 表 usage 字段
内存使用率告警	内存使用率告警超过 80%	同上	memory_usage_percent >80	lem_host_memory 表 memory_usage_percent 字段
日志告警		同上	db_log_level=error,fatal	lem_db_log 表 db_log_level 字段
数据库失效对象告警		同上		暂未实现
剩余冻结时间 (vacuum) 告警		同上	percent_towards_emergency_autovac>90	lem_db_age 表 percent_towards_emergency_autovac 字段
数据库内置定时任务告警	数据库定时任务告警	同上	status=failed	(select * from cron.job_run_details where status='failed')
data 目录可用空间	data 目录可用空间不足告警	同上	free_space<10,free_space_percent<10	

不足告警				
备份服务器备份目录可用空间不足告警	备份服务器备份目录可用空间不足告警	同上	free_space<10,free_space_percent<10	
ltcluster 状态异常告警	ltcluster 状态异常告警	同上		
pwr20分钟内没有数据产生告警	pwr20 分钟内没有数据产生告警	同上		
数据库节点状态异常告警	数据库节点状态异常告警	同上		
定时清理任务异常告警	定时清理任务异常告警	同上		
备份失败告警	备份失败告警	同上		
恢复失败告警	恢复失败告警	同上		
数据库日志解析异常告警	数据库日志解析异常告警	同上		

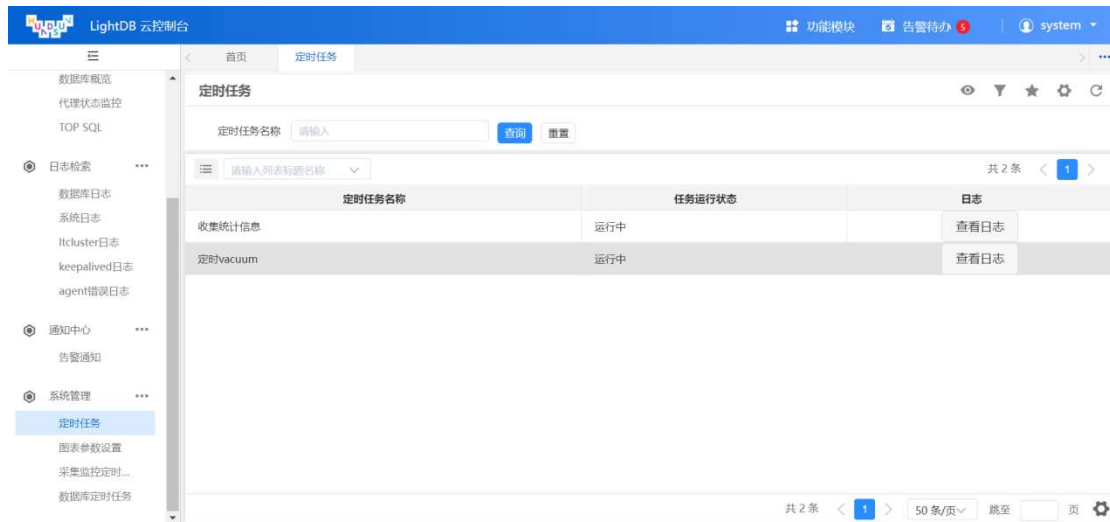
3.11 系统管理

系统管理模块主要包括 LightDB EM 全局设置的功能。

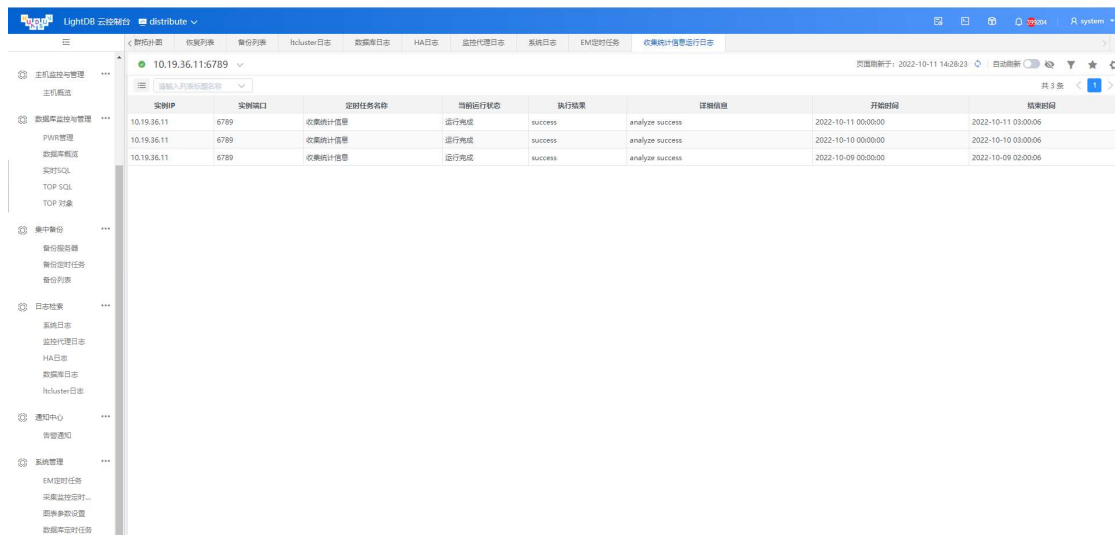
3.11.1 EM 定时任务

EM 定时任务配置的是采集和数据库定时任务外的其它定时任务。目前定时任务有获取最新的消息通知、收集统计信息、定时 vacuum、定时获取 agent 心跳和定时备份任务。

定时任务	执行周期 (cron 表达式)	是否启用 (1: 是; 0: 否)
收集统计信息	0 0 1 * * ?	1
定时 vacuum	0 0 1 * * ?	1



通过查看日志可以查看定时任务执行的日志明细：

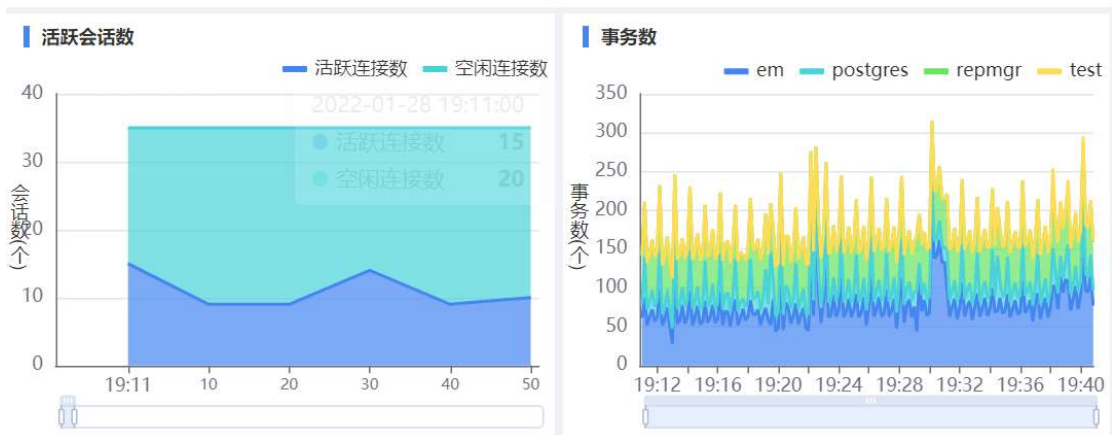


3.11.2 图表参数设置

图表参数配置的是 LightDB EM 每个图表的滑轨最小值和最大值，滑轨大小目前都是以毫秒为单位。

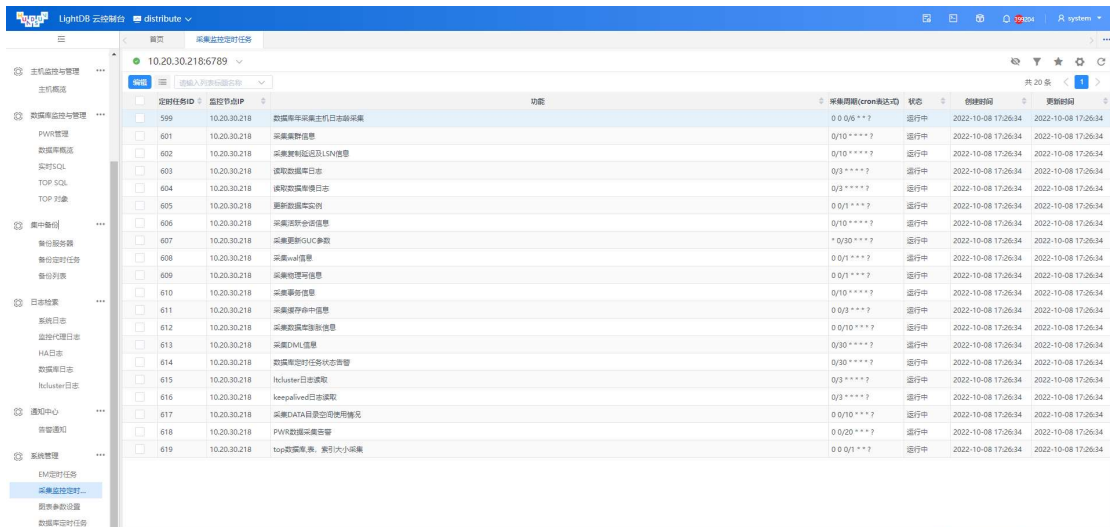
处理	图表内容	滑轨最小值	滑轨最大值	修改时间
<input type="checkbox"/>	缓存命中率图	60000	86400000	2021-11-22 15:19:19
<input type="checkbox"/>	活跃会话数图	20000	86400000	2021-11-22 15:28:18
<input type="checkbox"/>	表膨胀监控图	60000	86400000	2021-11-22 15:32:35
<input type="checkbox"/>	事务数图	20000	86400000	2021-11-22 15:32:38
<input type="checkbox"/>	wal数据量信息图	20000	86400000	2021-11-22 15:32:43
<input type="checkbox"/>	数据库物理写信息图	20000	86400000	2021-11-22 15:32:47
<input type="checkbox"/>	cpu使用率图	20000	86400000	2021-11-22 15:33:00
<input type="checkbox"/>	内存使用率图	20000	86400000	2021-11-22 15:33:04
<input type="checkbox"/>	磁盘使用量图	20000	86400000	2021-11-22 15:33:07
<input type="checkbox"/>	网络流量分析图	20000	86400000	2021-11-22 15:33:11
<input type="checkbox"/>	dml影响行数图	60000	86400000	2021-11-22 15:34:34

滑轨最小值代表图标的滑轨可以缩放的最小时间间隔，滑轨最大值代表图标的滑轨可以放大的最大时间间隔。

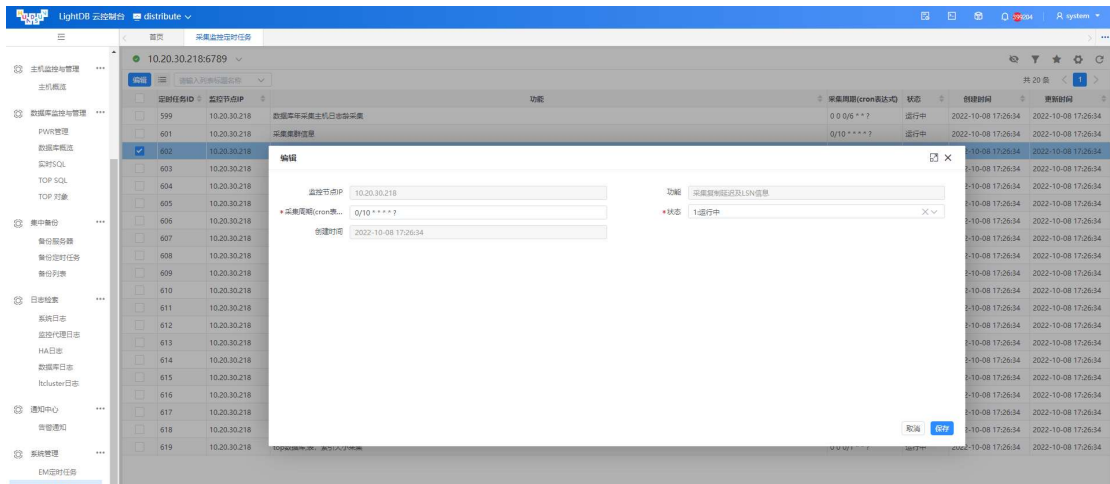


3.11.3 采集监控定时任务

采集监控定时任务配置了 LigthDB EM 中监控的主机和数据库采集业务点的定时任务信息。Agent 根据这些配置定时采集主机和数据库的信息



通过编辑操作，我们可以调整定时任务的采集周期和状态。采集定时任务目前使用 Spring Schedule 来实现，采集周期是标准的 cron 表达式。



采集周期 Cron 表达式是一个字符串，字符串以 5 或 6 个空格隔开，分为 6 或 7 个域，每一个域代表一个含义，Cron 有如下两种语法格式：

Seconds Minutes Hours DayofMonth Month DayofWeek Year 或
Seconds Minutes Hours DayofMonth Month DayofWeek

每一个域可出现的字符如下：

Seconds:可出现", - * /"四个字符，有效范围为 0-59 的整数

Minutes:可出现", - * /"四个字符，有效范围为 0-59 的整数

Hours:可出现", - * /"四个字符，有效范围为 0-23 的整数

DayofMonth:可出现", - * / ? L W C"八个字符，有效范围为 0-31 的整数

Month:可出现", - * /"四个字符，有效范围为 1-12 的整数或 JAN-DEC

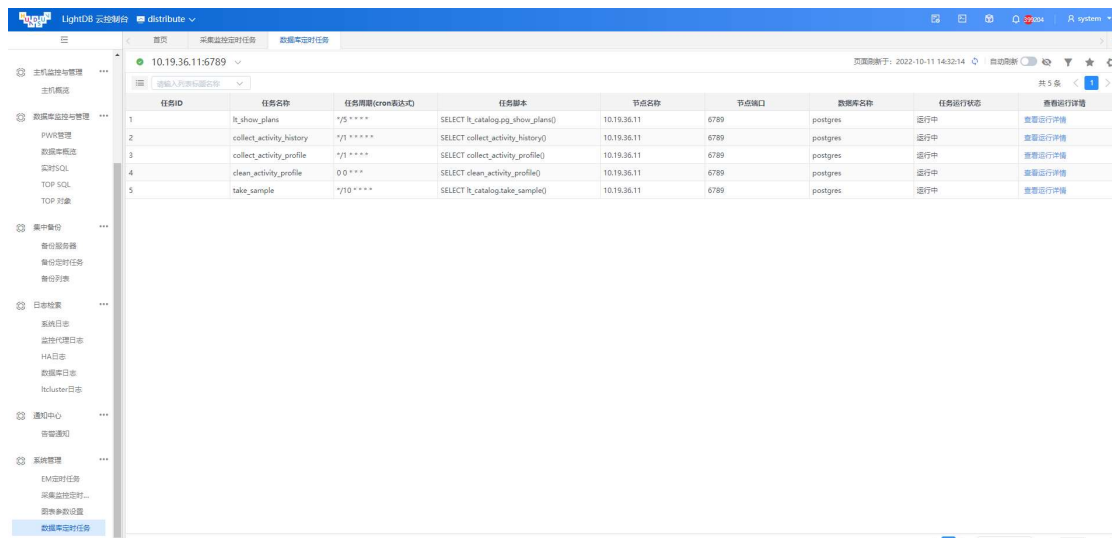
DayofWeek:可出现", - * / ? L C #"四个字符，有效范围为 1-7 的整数或 SUN-SAT 两个范围。1 表示星期天，2 表示星期一，依次类推

Year:可出现", - * /"四个字符，有效范围为 1970-2099 年

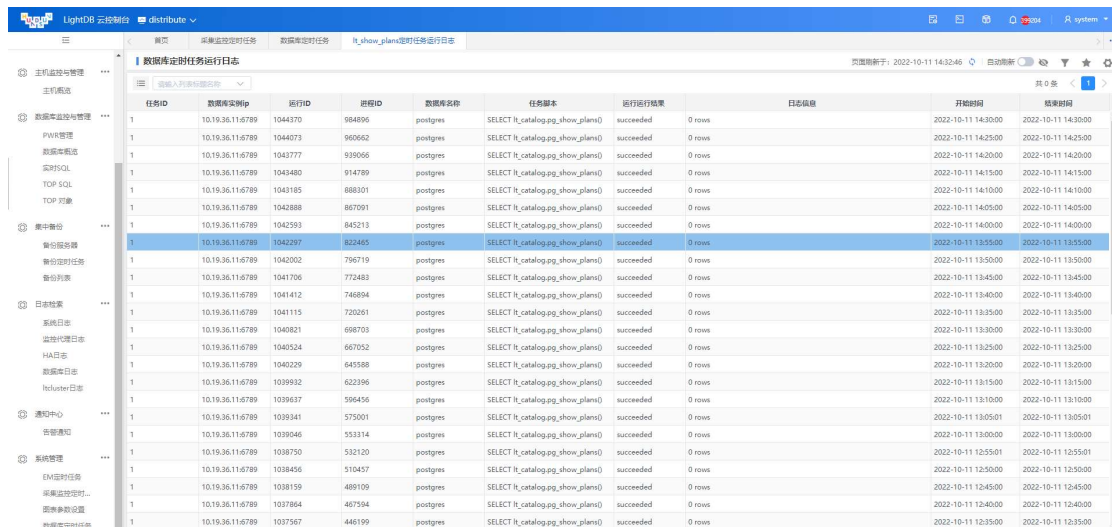
3.11.4 数据库定时任务

数据库定时任务配置在表 `cron.job`，需要在 `postgres` 库中查询。目前主要四个定时任务。

定时任务	周期	功能说明
<code>SELECT pg_show_plans()</code>	<code>* / 5 * * * *</code>	清理 <code>show plan</code> 数据的
<code>SELECT collect_activity_history()</code>	<code>* / 1 * * * * *</code> 每秒一次	对活跃会话历史 <code>active_session_history</code> 和进程活动信息历史 <code>lt_stat_activity_history</code> 进行采样
<code>SELECT collect_activity_profile()</code>	<code>* / 1 * * * *</code> 每分钟一次	对活跃会话汇总 <code>active_session_profile</code> 和进程活动汇总 <code>lt_stat_activity_profile</code> 进行采样
<code>SELECT clean_activity_profile()</code>	<code>0 0 * * *</code> 每 0 点 0 分一次	汇总表清理 <code>sample_active_session_profile</code> <code>sample_activity_profile</code>
<code>SELECT take_sample()</code>	<code>* / 1 * * * *</code> 每分钟一次	定时生成快照

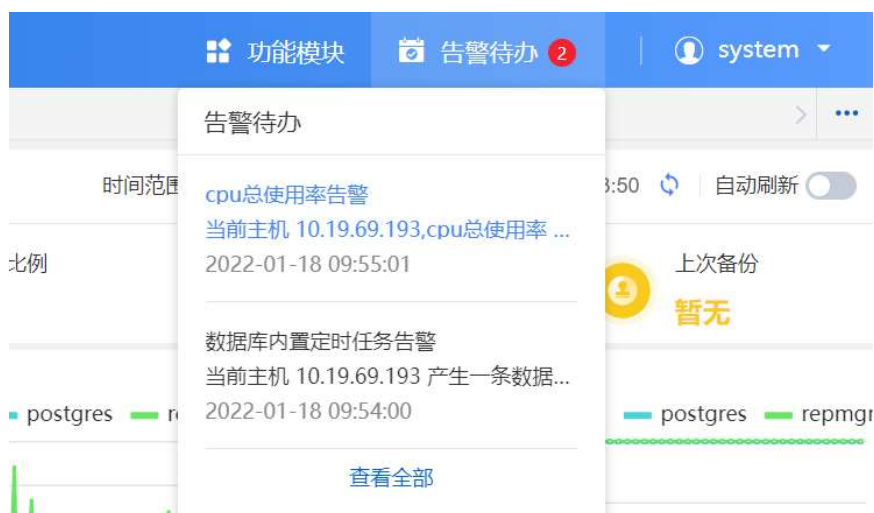


通过查看运行详情可以查看数据库定时任务执行的日志详情，方便定位问题：



3.12 告警待办

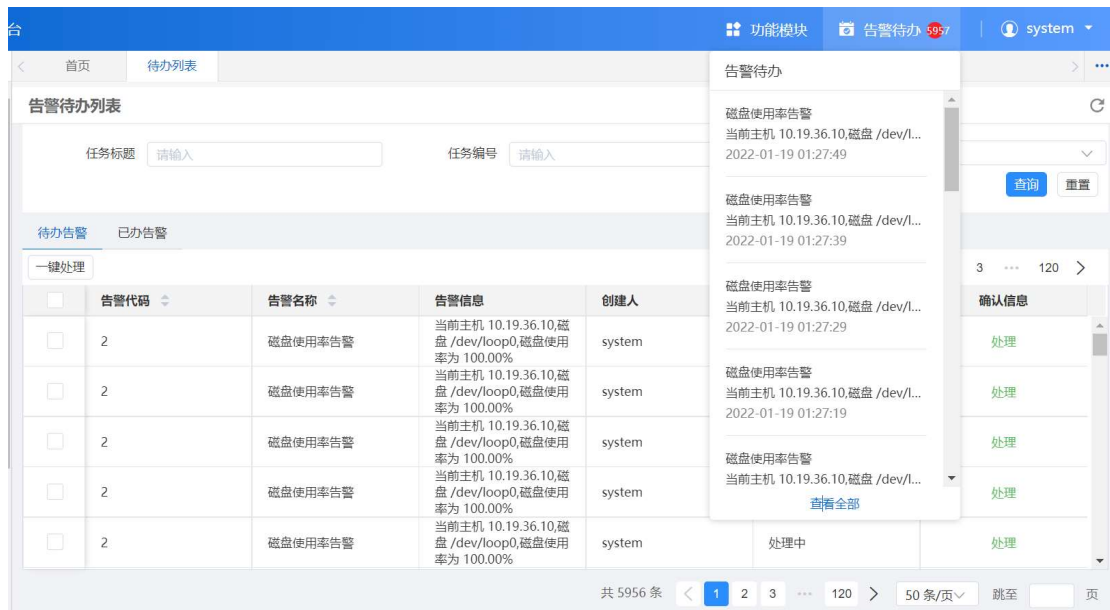
告警待办展示所有待处理的告警待办信息，数据库及主机信息如果超过阈值或者异常，就会自动生成告警待办。在实际运维过程中，我们需要特别关注告警待办，一般待办表示数据库及主机出现了问题，需要及时处理。



问题真正解决后，点击对应的告警待办输入确认信息保存，告警待办状态即为处理成功，并且会在顶部不再展示，只展示在已办告警列表中。



点击查看全部进入待办列表



3.13 用户中心

登录用户信息栏展示当前登录用户名，鼠标移动到用户信息上会展示用户对系统信息及修改操作。



3.13.1 修改信息

修改信息可以修改当前用户的信息

修改信息 ×

* 账号

* 姓名

外部用户名

邮箱

手机

电话

地址

3.13.2 修改密码

修改密码可以修改当前用户的密码，密码强度是根据包含类型（支持大写字母、小写字母、数字、其他字符）和长度，建议使用 8 位字符串、至少包含两种字符类型；

修改密码 ×

* 旧密码

* 新密码

* 确认密码

密码强度

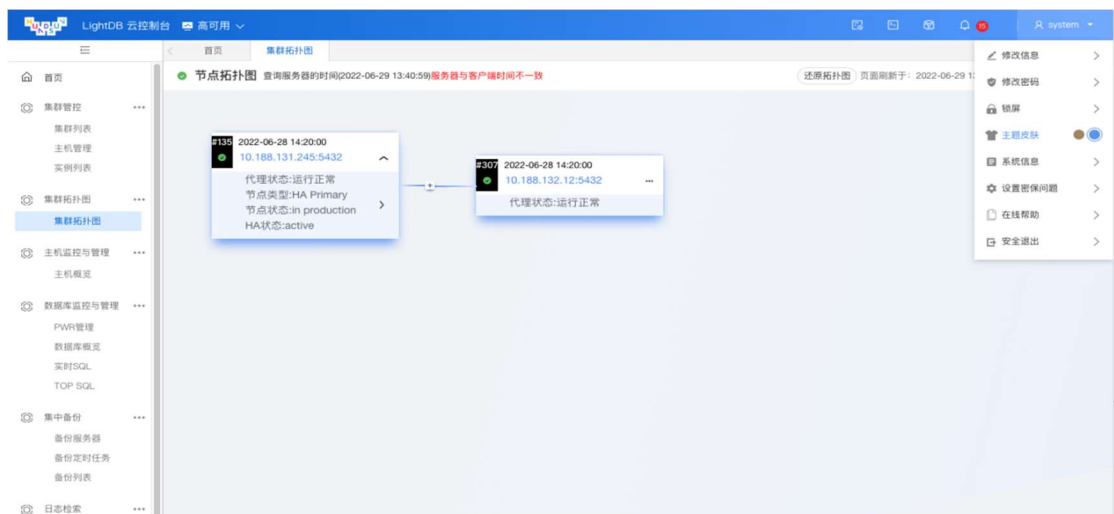
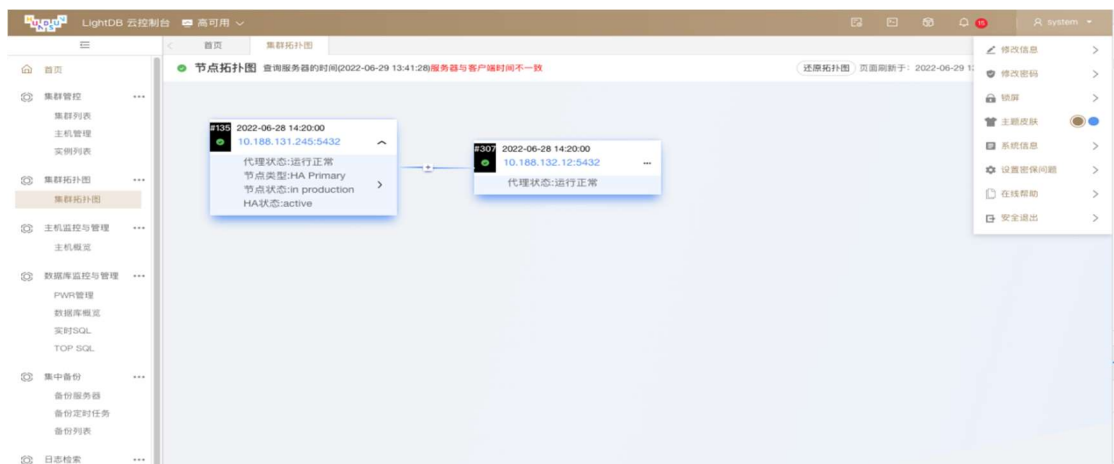
3.13.3 锁屏

锁屏可以遮盖当前系统防止其他用户操作界面，解除锁屏需要输入当前用户的密码。



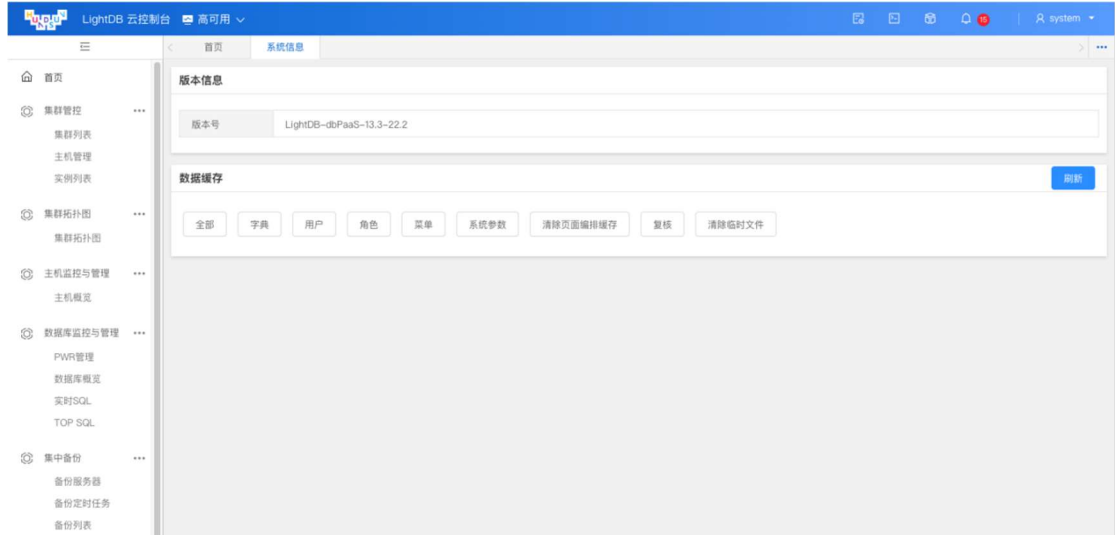
3.13.4 主题皮肤

主题皮肤可以选择 2 种皮肤中的一种，咖啡色和蓝色



3.13.5 系统信息

系统信息目前展示的是 DBPaaS 的版本系统和缓存



3.13.6 设置密保问题

设置好密保问题，用户可以根据密保问题找回密码

设置密保问题

问题一

答案

问题二

答案

问题三

答案

3.13.7 在线帮助

在线帮助展示的是 DBPaaS 的用户手册



3.13.8 安全退出

安全退出既退出当前用户登录，进入到系统登录页面。

4 LightDB EM 注意事项

4.1 Lightdb EM 集群相关

1. Single（单实例）部署模式下只能添加一套主机；
2. HA（高可用）部署模式下，HA Primary 只能有一个；
3. 添加主机时需要进行测试 SSH 连接校验，测试数据库连接校验；
4. 代理路径目录必须为空目录且需有可执行权限；
5. 实例角色需要选择实际部署的角色；
6. 一个主机只能部署一个代理监控。

4.2 22.3 版本升级内容

1. 主机监控和数据库实例监控分开部署；
2. 集群列表添加实例启停和主从切换，主机管理新增安装软件，配合集群管理可以初始化数据库实例；
3. 新增 top 对象，可查看数据库，数据库表，索引大小；
4. 采集监控添加下拉列表区分主机还是数据库实例的定时任务；
5. 新增备份恢复功能。