



FZS 概览

FZS 是九桥同步基于多年对于数据同步、数据容灾与数据集成领域积累的技术与经验，自主研发的一款通过对数据库重做日志 redo 的捕获与分析，从而实现 Oracle、MySQL、Microsoft SQL Server 及 PostgreSQL 等数据库的全增量一体的同构 + 异构数据实时流转的数据同步平台，主要应用在：数据同构灾备、数据异构集成，信创数据库迁移等业务场景中。

FZS 数据同步平台在支持传统关系型数据库的基础上，对国产数据库、云原生数据库、大数据平台以及自定义 API 也提供广泛的支持，并在不断扩展产品能力边界。[完整支持列表](#)

硬件平台方面，FZS 支持主流硬件架构，如龙芯 / 飞腾 / 鲲鹏 / 海光 / 兆芯 / 申威 / 宏芯等兼容 x86、ARM、或自研架构的处理器。

操作系统方面，FZS 支持麒麟 / UOS / 凝思 / RHEL / Ubuntu / Windows / AIX / Solaris 等操作系统。

衍生背景

国家“十四五”数字经济发展规划中，明确指出了，要“强化高质量数据要素供给”，支持市场主体依法合规开展数据采集，聚焦数据的标注、清洗、脱敏、脱密、聚合、分析等环节，提升数据资源处理能力，培育壮大数据服务产业。推动数据资源标准体系建设，提升数据管理水平和数据质量，探索面向业务应用的共享、交换、协作和开放。加快推动各领域通信协议兼容统一，打破技术和协议壁垒，努力实现互通互操作，形成完整贯通的数据链。推动数据分类分级管理，强化数据安全风险评估、监测预警和应急处置。深化政务数据跨层级、跨地域、跨部门有序共享。建立健全国家公共数据资源体系，统筹公共数据资源开发利用，推动基础公共数据安全有序开放，构建统一的国家公共数据开放平台和开发利用端口，提升公共数据开放水平，释放数据红利。

面对上述市场机遇，九桥同步综合自身在数据行业深耕 10 余年的经验，推出了 FZS 数据同步平台：凭借自身超高的数据同步效率，多源异构的数据库支持能力，完善的数据脱敏等技术帮助各大型企业和政府实现了对数据的灾备、清洗与脱敏等操作，帮助客户促进数据的融合与共享。

产品介绍

设计理念

FZS 数据同步平台的设计理念可以总结为八个字：契合场景、高效稳定。

契合场景

在 toB 领域，数据同步场景多种多样，不同的场景对数据同步的要求也不尽相同。例如：

- 数据灾备场景中，对于数据同步的要求往往落在 RTO 与 RPO 上。
- 数据分析场景中，对于数据同步的要求往往落在对数据源支持的完整度上。
- 日常运维场景中，对于数据同步的要求往往落在数据同步的稳定性、操作的易用性与告警的及时性上。

FZS 数据同步平台在扩大自身对于不同数据同步场景的支持的同时，也在深耕现有业务场景，不断提升契合度，力求为每一种场景提供最优的数据同步体验。

高效稳定

如果把数据同步链路比作水管，那么用户对于同步系统的需求就变得显而易见：

- 高效：水管越粗，水流越大，同步速度越快。
- 稳定：水管越稳固，水流越不容易中断，同步越不容易失败。

一款成功的同步工具，应该让用户感知不到同步工具的存在。就像水管一样，默默支撑着无数个水龙头的正常运转，让用水成为生活的基础设施的一部分，基础到用户只有在水管坏了的时候才会想起它的存在。

FZS 就是这样一款「数据水管」，高效地、稳定地将数据从源端运输到备端，让数据同步成为客户业务基础设施中不可或缺的一部分。

系统架构

下图以数据流转的角度，展示了 FZS 的系统架构与核心组件：

如上图所示，示例用户在 FZS Web 上创建同步链路后，FZS Web 会通过 FZS Web Server 的 API 交互，将同步链路的配置信息下发到源备端各自的 FZS Agent，FZS Agent 源端会根据配置信息，从数据源端抽取数据，经过数据高速传输至 FZS Agent 备端，最后将数据装载到数据备端。

① 备注

- 实线箭头 ——> 表示数据流转方向
- 虚线箭头 ----> 表示 API 交互方向
- 虚线圆头 ----○ 表示触发事件

FZS Agent

FZS 的核心组件，负责数据源端抽取、数据高速传输与数据备端装载。

FZS Web Server

FZS Agent 与 FZS Web 的服务中间件，对外输出 Restful API 服务。

FZS Web

FZS 的管理平台，通过对接 FZS Web Server 的 Restful API，实现了数据同步链路的创建、启动、停止、监控等运维功能。

FZS Daemon

FZS Web 的预警组件，监听各个同步链路的报错信息，并通过邮件、短信、Webhook 等方

式通知管理员用户。

功能亮点

FZS 数据同步平台的功能亮点同样可以被总结为八个字：[灵活多样](#)、[直观易用](#)。

灵活多样

在 toB 领域，企业级产品与服务如何满足与应对复杂、多变的业务场景，是一个亘古不变的话题。

FZS 数据同步平台在设计之初所考虑的系统灵活性与多样性，是九桥产品技术团队给到客户的答案。主要体现在以下几个方面：

源端支持多样

满足客户业务系统中常见关系型数据库，与大部分国产信创数据库，包括但不限于 Oracle、MySQL、PostgreSQL、SQL Server、DB2、OceanBase、TiDB、GaussDB、openGauss、武汉达梦 DM、人大金仓 Kingbase、LightDB、SUNDB、云和恩墨 MogDB 等。

[查看完整支持列表](#)

备端支持完整

满足任何兼容 `OCI` 或 `ODBC` 接口的关系型数据库，包括但不限于 Oracle、MySQL、PostgreSQL、SQL Server、DB2、OceanBase、TiDB、GaussDB、openGauss、武汉达梦 DM、人大金仓 Kingbase、LightDB、SUNDB、云和恩墨 MogDB 等。

[查看完整支持列表](#)

部署方式多选

FZS Agent 支持四种私有部署方式：

- 点对点部署：源端 Agent 部署在源主机 + 备端 Agent 部署在备主机
- 中间机部署：
 - 源端 Agent + 备端 Agent 皆部署在中间机
 - 源端 Agent 部署在源主机 + 备端 Agent 部署在中间机
 - 源端 Agent 部署在中间机 + 备端 Agent 部署在备主机

直观易用

当一款同步产品的介入到企业 IT 运维体系中时，通常对于忙于满足企业自身业务需求的 IT 运维人员来说，产品的学习成本也会不可避免低随之而至。因此，如何在满足复杂多样的业务需求的同时，尽可能地降低产品的使用门槛与运维成本，是一个重要的设计平衡点。

因此，九桥产品技术团队将 FZS 数据同步平台的以下几个方面打磨成了亮点：

一键部署

通过 [FZS Web](#)，用户可以通过一键部署功能，快速部署 [FZS Web Server](#)，实现源备端的数据同步。



配置链路

通过 [FZS Web](#)，用户可以通过配置链路功能，快速配置源备端的数据同步链路。



链路监控

通过 [FZS Web](#) 的仪表盘大屏，用户可以实时监控多条数据同步链路的同步计数、增量延时、链路状态等数据。



告警平台

通过 [FZS Web](#) 的告警平台，用户可以实时监控多条数据同步链路的告警信息，并通过邮件、短信、Webhook 等方式通知管理员用户。

系统设置

用户管理

告警策略

软件管理

导入导出

Webhook 告警



开启 Webhook 告警后，链路报错会自动发送至指定渠道

POST ▾

https://qyapi.weixin.qq.com/cgi-bin/webhook/send?key=2e336418-da05-4b1f

HEADERS

Content-Type: application/json

BODY

```
1  {
2    "msgtype": "text",
3    "text": {
4      "content": "${message}"
5    }
6  }
```

注意：请使用 `${message}` 绑定告警内容

测试

保存

部署方案

在选择核心组件 FZS Agent 的部署方案时，我们提供两种选项：

- [点对点](#)（性能至上）
- [中间机](#)（安全至上）

每种拓扑结构都各自满足不同的业务运行和数据安全的需求，允许用户根据其特定要求选择最合适的选项。

点对点

对于追求高性能、低时延的用户，我们建议使用点对点部署方式。点对点部署方式下，源端和备端直接通信，不存在任何数据中途流转的环节，数据传输效率高，适用于对数据传输性能要求较高的场景。如下图所示：

为了达到最佳的性能，[FZS Web Server](#) 源端和备端需要分别部署在源端和备端主机上，以便对各自的 FZS Agent 进行管理与资源调度，数据库源端和备端之间需要直接通信。

中间机

对于追求数据安全、业务侵入性低的用户，我们建议使用中间机部署方式。中间机部署方式下，源端和备端通过中间机进行数据传输，中间机负责数据的中转，源端和备端之间不直接通信，数据传输过程中不会暴露源端和备端的真实地址，适用于对数据传输安全性要求较高的场景。如下图所示：

为了达到最佳的安全性与最低的业务侵入性，只需一个 [FZS Web Server](#) 部署在中间机上，它便可以对源端和备端的 FZS Agent 进行管理与资源调度，实现数据的远程抽取与装载，数据库源端和备端之间不直接通信。

i 备注

使用中间机部署方式时，FZS 暂不支持配置以 `Oracle` 或者 `SUNDB` 作为源端的同步链路。

案例汇总

附录

FZS 支持的数据库

i 备注

-  = 适配已完成，可以直接使用。
-  = 代表适配正在进行中，敬请期待。

以下是 FZS 的数据支持矩阵：

类型	名称	支持源端抽取	支持备端装载
<code>oracle</code>	Oracle		
<code>mysql</code>	MySQL		
<code>postgres</code>	PostgreSQL		
<code>sqlserver</code>	Microsoft SQL Server		
<code>db2</code>	IBM Db2		
<code>ob-oracle</code>	OceanBase Oracle 模式		
<code>ob-mysql</code>	OceanBase MySQL 模式		
<code>polardb</code>	PolarDB		
<code>tidb</code>	TiDB		
<code>gaussdb</code>	GaussDB		

类型	名称	支持源端抽取	支持备端装载
opengauss	openGauss	✓	✓
tdsql-mysql	TDSQL MySQL	✓	✓
tdsql-pg	TDSQL PostgreSQL		✓
dm	武汉达梦 DM	✓	✓
kingbase	人大金仓 Kingbase		✓
gbase	南大通用 GBASE		✓
goldendb	中兴通讯 GoldenDB		✓
lightdb	恒生电子 LightDB	✓	✓
sundb	科蓝软件 SUNDB	✓	✓
mogdb	云和恩墨 MogDB	✓	✓
kafka	Apache Kafka		✓
hive	Apache Hive		⚠
doris	Apache Doris		⚠
starrocks	StarRocks		⚠
gaussdb-dws	GaussDB (DWS)		✓

恒生电子 LightDB

LightDB 是恒生电子股份有限公司研发并将长期支持的一款同时支持在线事务处理与在线分析处理的融合型分布式数据库产品，具备 SQL 兼容性高、容量弹性伸缩、金融级高可用、现代硬件融合、纯内存计算等核心特性，主要适用于对可用性、一致性要求较高的系统。

Oracle 至 LightDB 链路适配

同步功能支持

Oracle 对象	FZS 是否支持
表	支持
分区表（1、2 级别分区表）	支持（但 1、2 级分区操作：合并、拆分、交换、移动分区、重建分区索引、哈希分区表添加分区、表空间的指定暂不支持）
临时表（事务、会话级）	支持
视图	暂不支持
物化视图	暂不支持
索引	支持
约束	支持

Oracle 对象	FZS 是否支持
序列	支持
同义词	暂不支持
触发器	暂不支持
函数	暂不支持
存储过程	暂不支持
包	暂不支持
Dblink	暂不支持
type	支持（详见数据类型映射关系）

数据类型映射

Oracle 数据类型	LightDB 数据类型（LightDB 内部数据类型转换）
<code>TIMESTAMP(1) WITH TIME ZONE</code>	<code>TIMESTAMP(1) WITH TIME ZONE (TIMESTAMP TZ(1))</code>
<code>INTERVAL YEAR(4) TO MONTH</code>	<code>INTERVAL YEAR TO MONTH</code>
<code>BFILE</code>	<code>BYTEA</code>

Oracle 数据类型	LightDB 数据类型 (LightDB 内部数据类型转换)
NCLOB	TEXT
INTERVAL DAY(n) TO SECOND(n)	INTERVAL DAY TO SECOND(6)
LONG RAW	BYTEA
BINARY_FLOAT	REAL (FLOAT4)
NCHAR	CHAR (BPCHAR)
UROWID	OID
TIMESTAMP(n)	TIMESTAMP(n)
BINARY_DOUBLE	DOUBLE PRECISION (FLOAT8)
FLOAT	FLOAT8
TIMESTAMP(n) WITH TIME ZONE	TIMESTAMP(n) WITH TIME ZONE (TIMESTAAMPTZ)
NVARCHAR2	NVARCHAR2
ROWID	TID
XMLTYPE	BYTEA

Oracle 数据类型	LightDB 数据类型 (LightDB 内部数据类型转换)
BLOB	BYTEA
LONG	TEXT
CLOB	TEXT
RAW	BYTEA
DATE	DATE
CHAR	CHAR (BPCHAR)
VARCHAR2	VARCHAR2
NUMBER	NUMERIC
INT	BIGINT (INT8)

LightDB 到 LightDB 链路适配

FZS Agent 全部参数

以下是 FZS Agent 支持的全部参数：

```
{
  "export.param_full_sync_user": "全同步时是否自动创建用户",
  "export.param_full_sync_tbspace": "全同步时是否自动创建表空间",
  "export.param_tgt_not_drop": "全同步时是否同步表结构",
  "export.param_ddl_only_truncate": "ddl 只同步 truncate 操作",
  "export.param_archive_log": "启动时切换一次redo日志",
  "export.param_proc_max_mem": "export 进程最大使用内存，单位为兆",
  "export.param_cpu_max": "export 最大使用cpu",
  "export.param_disk_quota": "允许 fzs 使用的磁盘空间，占用空间超过该值时停止抽取，空间释放后会继续",
  "export.param_is_ddl": "是否同步 ddl 操作",
  "export.param_tbl_drt_rnm": "是否将 drop 操作转换为 rename",
  "export.param_sync_table": "是否全同步表",
  "export.param_sync_priv": "是否全同步权限",
  "export.param_sync_temp_table": "是否全同步临时表",
  "export.param_sync_proc": "是否全同步存储过程",
  "export.param_sync_syn": "是否全同步同义词",
  "export.param_sync_profile": "是否全同步profile",
  "export.param_sync_seq": "是否全同步序列",
  "export.param_sync_view": "是否全同步视图",
  "export.param_sync_java": "是否全同步 java",
  "export.param_sync_role": "是否全同步角色",
  "export.param_trace_seq": "是否同步序列的增量",
  "export.param_sync_index": "全同步时创建索引的并发数，配置为 0 时表示不同步索引",
  "export.param_sync_tbspace": "增量是否同步创建表空间操作",
  "export.param_new_user": "增量是否同步创建用户操作",
  "export.param_exadata": "redo 大端格式存储，asm 的 db 模式下有效，一般不做修改",
  "export.param_off": "redo 读取偏移量，asm 的 raw 模式下有效，一般不做修改",
  "export.param_left_space": "磁盘剩余空间小于该值时停止抽取，单位为兆，空间释放后会继续",
}
```


快速开始

 快速部署

2 个项目

 快速上手

登录 FZS Web

快速部署

容器化部署

FZS 提供基于 Docker Compose 的容器化部署方式，可以做到真正意义上的一键部署。

软件包部署

对于 oracle 的同步链路，由于需要直接读取 redo 文件，因此除了需要启动容器化部署中的组件外，FZS Web Server 需要部署在装有数据库的本地环境（支持 Linux、Windows（还...

容器化部署

FZS 提供基于 Docker Compose 的容器化部署方式，可以做到真正意义上的一键部署。

前置条件

部署前，请确保已经安装了 [Docker Engine](#) 以及 [Docker Compose](#)，并且已经启动了 Docker 服务。

支持架构

- x86_64
- arm64

硬件要求

- 最低配置：4 CPU，16 GB 内存，50 GB 硬盘
- 推荐配置：8 CPU，32 GB 内存，50 GB 硬盘
- 网络端口：
 - 对外开放 80 端口，用于 HTTP 访问
 - 对内开放所有端口，用于内部通信

开始部署

获取 `gen.sh` 自动化脚本

前往 [FZS Release](#) 页面，获取最新版本的 `gen.sh` 文件。

生成 `docker-compose.yml` 文件

执行 `./gen.sh` 根据提示设置参数，生成 `docker-compose.yml` 文件。

一键启动

在包含 `docker-compose.yml` 文件的目录下，执行以下命令：

```
docker compose up -d
```

检查容器是否启动成功:

```
docker ps
```

如果 `web` 与 `daemon` 的容器状态类似以下输出, 说明容器启动成功:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
3a0da025e1de	192.168.0.198:880/9bridges/fzs-web-server:0.4.0	"/usr/local/s2i/run"	27 seconds ago	Up 26 seconds	
fzs-web-server-1					
73a1648c3409	9bridges/fzs-daemon:0.2.2-beta.2-node	"node index.js"	2 hours ago	Up About an hour	
fzs-daemon-1					
8ca3c3460a0e	192.168.0.198:880/9bridges/synjq-query:0.3.4	"/usr/local/s2i/run"	2 hours ago	Up 2 hours	8080/tcp, 8443/tcp, fzs-query-1
0e7a6d8d0000	9bridges/fzs-web:0.5.1-rc.2	"/start.sh"	2 hours ago	Up About an hour	0.0.0.0:5555->5555/tcp, :::5555->5555/tcp, 0.0.0.0:80->8080/tcp, :::80->8080/tcp fzs-web-1
56c5b25fc7b2	provectuslabs/kafka-ui:latest	"/bin/sh -c 'java --..."	3 hours ago	Up 3 hours	0.0.0.0:8080->8080/tcp, :::8080->8080/tcp kafka-ui
391e8bd349c8	192.168.0.198:880/9bridges/fzs-source-connector:2.6.1.Final.86f50f.2.4	"/docker-entrypoint..."	3 hours ago	Up 3 hours	8778/tcp, 9092/tcp, 0.0.0.0:18083->8083/tcp, :::18083->8083/tcp root-connect-1
88e7fa4f48b9	bitnami/kafka:3.6	"/opt/bitnami/script..."	3 hours ago	Up 3 hours	
0.0.0.0:9092-9094->9092-9094/tcp, :::9092-9094->9092-9094/tcp					root-kafka-1

软件包部署

对于 oracle 的同步链路，由于需要直接读取 redo 文件，因此除了需要启动容器化部署中的组件外，FZS Web Server 需要部署在装有数据库的本地环境（支持 Linux、Windows（还在研发中）以及 aix 的多种操作系统部署）

部署条件

- 硬件配置：单条链路磁盘预留空间至少为数据库中同步数据量的 1/3，内存与 cpu 无特定要求（机器性能与软件运行效率成正比）
- 网络端口：
 - 默认对外开放 8081 端口，用于 HTTP 访问
 - 对内开放所有端口，用于内部通信

❗ 信息

请确认 FZS Web Server 与 FZS Web 的各自宿主机系统时间一致，否则可能会导致同步链路监控数据异常。

部署流程

创建系统用户 (Linux 系统)

- 在 oracle 所在机器上创建和 oracle 或者 grid 用户(ASM 存储环境上一般有 grid 用户)同组的用户,执行下面脚本创建 fzs (可任意修改用户名) 用户，web-server 需要部署在这个新建的系统用户下

```

#!/bin/sh
echo "please input system username to create(default:fzs):"
read sysuser
if [ ! -n "$sysuser" ]; then
    sysuser=fzs
fi
SYSTEM=$(uname -s)
case $SYSTEM in
Linux | AIX)
    j=3
    ;;
SunOS | HP-UX)
    j=1
    ;;
*)
    echo unknow system
    exit 1
    ;;
esac

if id -u grid >/dev/null 2>&1; then
    grp=$(groups grid | awk '{for(i=3;i<=NF;i++) if(i!=NF) printf $i",";
else printf $i;}')
else
    grp=$(groups oracle | awk '{for(i=3;i<=NF;i++) if(i!=NF) printf
$i","; else printf $i;}')
fi
if [ $SYSTEM = "AIX" ]; then
    UserExists=17
else
    UserExists=9
fi
if [ $SYSTEM = "SunOS" ]; then
    ONLY_SUN="-s /bin/bash"
fi

useradd $ONLY_SUN -G $grp -m $sysuser
if [ $? = $UserExists ]; then

```

- 创建 oracle 数据库用户并授权

源端

```
/* 由于权限问题，普通用户无法直接查询下面几个表，因此需要创建视图来代替 */
/* 数据库增量相关，fzs 通过下面两个视图判断当前的增量位点信息 */
create or replace view XKCCLE as select * from sys.x$kccl;
create or replace view XKCCCP as select * from sys.x$kcclcp;
/* redo 格式相关，不同的 redo 格式，需要不同的解析方法 */
create or replace view FZS_XKSPPI as select * from X$KSPPI;
create or replace view FZS_XKSPPSV as select * from X$KSPPSV;

/* 补全日志信息 */
alter database force logging;
alter database add supplemental log data;
alter database add supplemental log data (primary key, unique index)
columns;
create user fzs1 identified by fzs1;
create role fzs_role identified by fzs_role;
grant connect,lock any table,select any table,select any
dictionary,alter system,execute any type to fzs_role,fzs1;
grant fzs_role to fzs1;
grant execute on dbms_flashback to fzs_role,fzs1;
/* fzs 查询数据需要使用一些表空间，这里给 fzs1 用户分配 2M 的表空间 */
alter user fzs1 quota 2m on users;
/* rac 多节点环境下，fzs 会定期向 fzs1 表写入数据，非 rac 环境可以不创建 */
create table fzs1.fzs1 (f1 int);

## 12c 以上版本源端还需要执行以下授权
/* 创建 fzs_pdb 表用于查询 PDB 信息 */
create table sys.fzs_pdb as select * from sys.v$pdb;
/* 由于 12c 以上的数据库权限管理方式改变，select any table 权限不足以查询一些系
统表，因此需要单独赋权 */
grant create session to fzs1;
grant select on sys.ts$ to fzs1;
grant select on sys.tabsubpart$ to fzs1;
grant select on sys.obj$ to fzs1;
grant select on sys.seg$ to fzs1;
grant select on sys.coltype$ to fzs1;
```

解压 web-server 对应安装包

- 以 fzsweb.a39731c-0.1.2-Linux.tar.gz 软件包为例，在上一步创建的系统用户下解压安装包

```
$ tar -xvf fzsweb.a39731c-0.1.2-Linux.tar.gz
```

- 解压目录预览：

```
[lcy@centos-2 fzs-soft]$ ls -R
.:
fzsweb  fzsweb.a39731c-0.1.2-Linux.tar.gz

./fzsweb:
bin  lib64  run

./fzsweb/bin:
fzsweb-exe  fzsweb-exe-ob

./fzsweb/lib64:
oci

./fzsweb/lib64/oci:
adrci      genezi      libclntsh.so.10.1  libclntsh.so.18.1  libmql1.so  libocci.so.10.1  libocci.so.18.1  libocijdbc19.so  ojdbc8.jar  xstreams.jar
BASIC_LICENSE  libclntshcore.so.19.1  libclntsh.so.11.1  libclntsh.so.19.1  libnnz19.so  libocci.so.11.1  libocci.so.19.1  liboramysql19.so  ucp.jar
BASIC_README  libclntsh.so      libclntsh.so.12.1  libipcl.so      libocci.so  libocci.so.12.1  libociei.so      network      uidrvci

./fzsweb/lib64/oci/network:
admin

./fzsweb/lib64/oci/network/admin:
README

./fzsweb/run:
fzsAgent.sh  sql  start.bat  start.sh  stop.bat  stop.sh

./fzsweb/run/sql:
init.sql
```

启动 web-server

进入 fzsweb/run

修改端口

vim 打开 start.sh（Windows 环境需要修改 start.bat）脚本将 FZS_WEB_PORT=8081 替换成 FZS_WEB_PORT=1963（假设自定义对外端口为 1963）

```
export LD_LIBRARY_PATH=`pwd`/../../lib64/oci:$LD_LIBRARY_PATH
#export FZS_WEB_PORT=8081
export FZS_WEB_PORT=1963
```

ESC 退出编辑模式输入:wq 保存修改

执行启动命令

```
#Oracle数据库环境
$./start.sh
#ob数据库环境
$./start.sh -o
```

检查进程是否启动成功:

```
$ps x
```

如果 web-server 启动成功，会有以下进程状态打印:

PID	TTY	STAT	TIME	COMMAND
72372	pts/0	S	0:00	../bin/fzsweb-exe
72373	pts/0	Sl	0:00	../bin/fzsweb-exe

停止 web-server

```
$./stop.sh
```

检查容器是否停止成功:

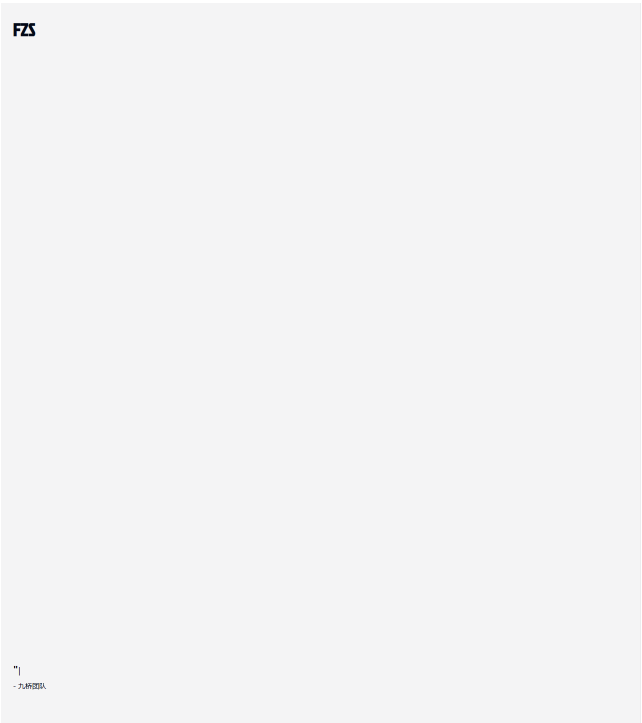
```
$ps x
```

如果 web-server 停止成功，之前启动进程状态将会被清除:

PID	TTY	STAT	TIME	COMMAND
-----	-----	------	------	---------

快速上手

登录 FZS Web



早上好, 欢迎回来

输入用户名和密码登录

用户名

密码

登录

添加 Web Server

oracle-yuan

oracle-yuan-cx

oracle-bei-cx

dws

gaussdb

arm-opengauss

arm-mysql

mogdb-ley

opengauss-ley

opengauss

kafka

81mogdb

lx yuan

lxbei

lx1

lx2

43kafka

数据校验

root 超级管理员

系统设置

用户管理

告警策略

软件管理

导入导出

Web Server

FZS Web 绑定的 Web Server

点击添加 Web Server

172.21.16.3:9081836 天

172.21.16.4:8081861 天

172.21.16.4:48881836 天

172.21.16.4:19111861 天

172.21.16.3:19081永久授权

172.21.16.4:19123837 天

120.46.69.135:8081836 天

172.21.16.3:8081永久授权

172.21.16.4:19110837 天

172.21.16.4:29111837 天

172.21.16.4:29222837 天

172.21.16.4:29333837 天

172.21.16.4:88111未授权

FZS License

已上传的 FZS License 授权

sn87.txt

sn_nolimit.txt

sn_nolimit_host.txt

Choose Files No file chosen

FZS Agent

已上传的 FZS Agent 软件包

0531.md5.a3afb7.feature-cyrmteleg.3029ab59.fzs.linux.11g.web.tar.gz

0612.md5.e825bc.feature-hdb-dbz.56341a5c.fzs.linux.11g.web.tar.gz

0613.main.7a18b5d3.fzs.linux.11g.web.tar.gz

0731.md5.1e72f7.feature-ora-mogdb.753cd8b0.fzs.linux.12c.web.tar.gz

0801.adg.md5.990976.feature-charset.a2b089ad.fzs.linux.11g.web.tar.gz

0807.adg.md5.5033de.feature-charset-kafka.88d3552a.fzs.linux.11g.web.tar.gz

0808.adg.md5.49c35f.feature-charset-kafka.607388d2.fzs.linux.11g.web.tar.gz

0808.adg.md5.f4b9b4.feature-charset-kafka.607388d2.fzs.linux.12c.web.tar.gz

0813.adg.md5.3315cc.feature-charset-kafka.dcf147b3.fzs.linux.12c.web.tar.gz

0814.adg.md5.1ca4a2.feature-charset-kafka.2703a76c.fzs.linux.12c.web.tar.gz

0814.md5.e7f166.feature-mogodb.5b1705fa.fzs.linux.11g.web.tar.gz

0815.md5.b6bcb4.feature-mogodb.c66c49bf.fzs.linux.11g.web.tar.gz

1009.feature-zsufo-ohbtarget.b096042.fzs.linux.11g.web.tar.gz

11g-0126.feature-arm.41dc7ab4.gz

11g-0131.feature-sundb-proc.afb6e5a2.gz

11g-0201.feature-sundb-proc.61759b28.gz

11g-0227.odbc.21b98dda.gz

11g-0320.feature-dbz-nopk.53516e52.gz

11g-0402.feature-odbc-igtidisable.ed6bda57.gz

11g-0403.feature-odbc-igtidisable.0905a7cd.gz

11g-0428.odbc.caec1258.gz

11g-0627.feature-lightdb.12bf6489.gz

11g-0702.feature-foreign-key-test.eF0dd376.gz

oracle-yuan

oracle-yuan-cx

oracle-bei-cx

dws

gaussdb

arm-opengauss

arm-mysql

mogdb-ley

opengauss-ley

opengauss

kafka

81mogdb

lx yuan

lxbei

lx1

lx2

43kafka

数据校验

root 超级管理员

系统设置

用户管理

告警策略

软件管理

导入导出

Web Server

FZS Web 绑定的 Web Server

点击添加 Web Server

172.21.16.3:9081836 天

172.21.16.4:8081861 天

172.21.16.4:48881836 天

172.21.16.4:19111861 天

172.21.16.3:19081永久授权

172.21.16.4:19123837 天

120.46.69.135:8081836 天

172.21.16.3:8081永久授权

172.21.16.4:19110837 天

172.21.16.4:29111837 天

172.21.16.4:29222837 天

172.21.16.4:29333837 天

172.21.16.4:88111未授权

FZS License

已上传的 FZS License 授权

sn87.txt

sn_nolimit.txt

sn_nolimit_host.txt

Choose Files No file chosen

FZS Agent

已上传的 FZS Agent 软件包

0531.md5.a3afb7.feature-cyrmteleg.3029ab59.fzs.linux.11g.web.tar.gz

0612.md5.e825bc.feature-hdb-dbz.56341a5c.fzs.linux.11g.web.tar.gz

0613.main.7a18b5d3.fzs.linux.11g.web.tar.gz

0731.md5.1e72f7.feature-ora-mogdb.753cd8b0.fzs.linux.12c.web.tar.gz

0801.adg.md5.990976.feature-charset.a2b089ad.fzs.linux.11g.web.tar.gz

0807.adg.md5.5033de.feature-charset-kafka.88d3552a.fzs.linux.11g.web.tar.gz

0808.adg.md5.49c35f.feature-charset-kafka.607388d2.fzs.linux.11g.web.tar.gz

0808.adg.md5.f4b9b4.feature-charset-kafka.607388d2.fzs.linux.12c.web.tar.gz

0813.adg.md5.3315cc.feature-charset-kafka.dcf147b3.fzs.linux.12c.web.tar.gz

0814.adg.md5.1ca4a2.feature-charset-kafka.2703a76c.fzs.linux.12c.web.tar.gz

0814.md5.e7f166.feature-mogodb.5b1705fa.fzs.linux.11g.web.tar.gz

0815.md5.b6bcb4.feature-mogodb.c66c49bf.fzs.linux.11g.web.tar.gz

1009.feature-zsufo-ohbtarget.b096042.fzs.linux.11g.web.tar.gz

11g-0126.feature-arm.41dc7ab4.gz

11g-0131.feature-sundb-proc.afb6e5a2.gz

11g-0201.feature-sundb-proc.61759b28.gz

11g-0227.odbc.21b98dda.gz

11g-0320.feature-dbz-nopk.53516e52.gz

11g-0402.feature-odbc-igtidisable.ed6bda57.gz

11g-0403.feature-odbc-igtidisable.0905a7cd.gz

11g-0428.odbc.caec1258.gz

11g-0627.feature-lightdb.12bf6489.gz

11g-0702.feature-foreign-key-test.eF0dd376.gz

您是否已部署 Web Server ?

请根据您的实际情况选择下面的选项:

未部署 Web Server

已部署 Web Server

我已经部署了一个 Web Server, 我需要通过其绑定到 FZS Web 中。

Web Server IP 地址 *

请输入 Web Server IP 地址

Web Server 端口 *

请输入 Web Server 端口

提交

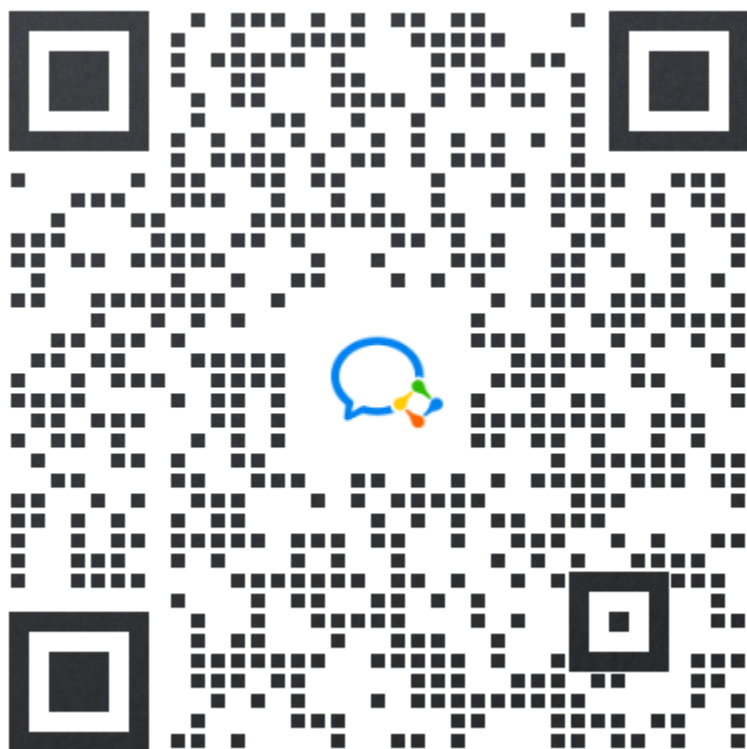
添加授权

❗ 信息

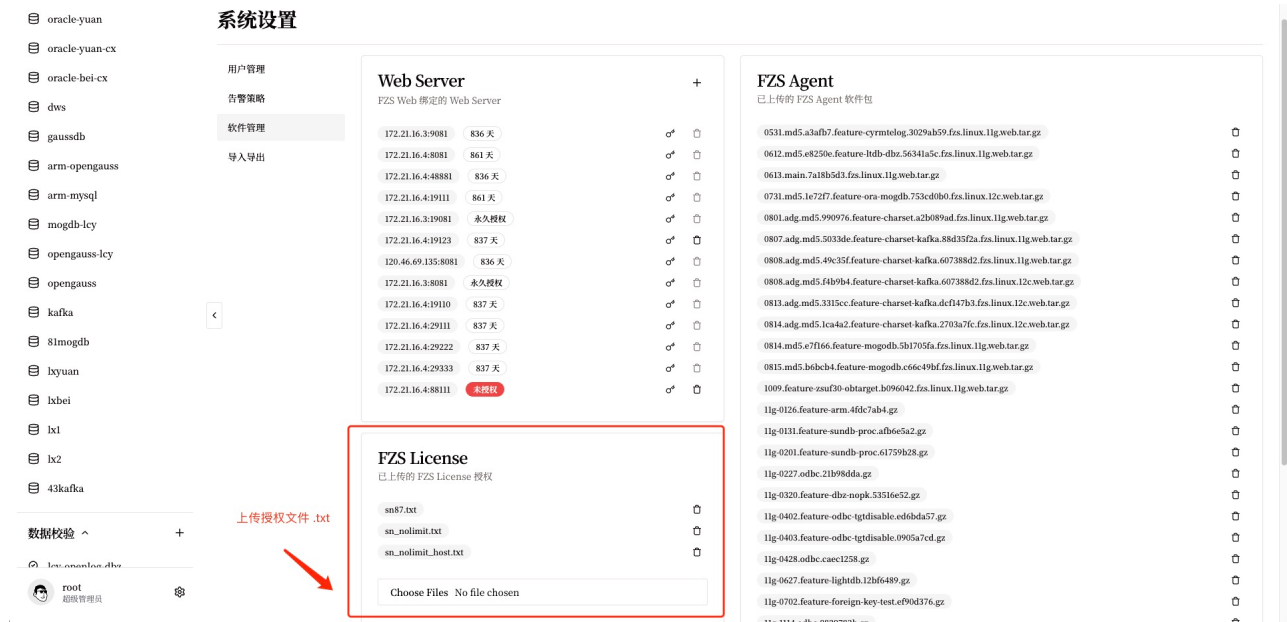
FZS 数据同步平台的正常使用，需要配合有效的授权文件。

获取授权

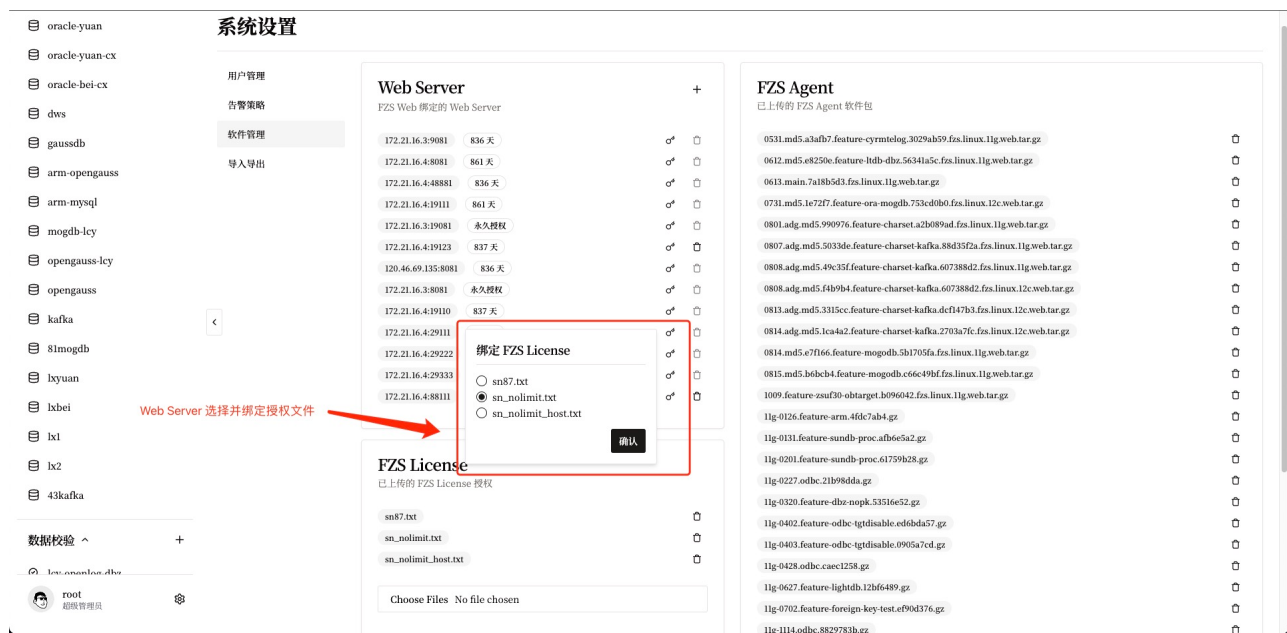
FZS 的授权需要联系 FZS 的产品团队获取，获取到的授权文件为 `license.txt`。



上传授权

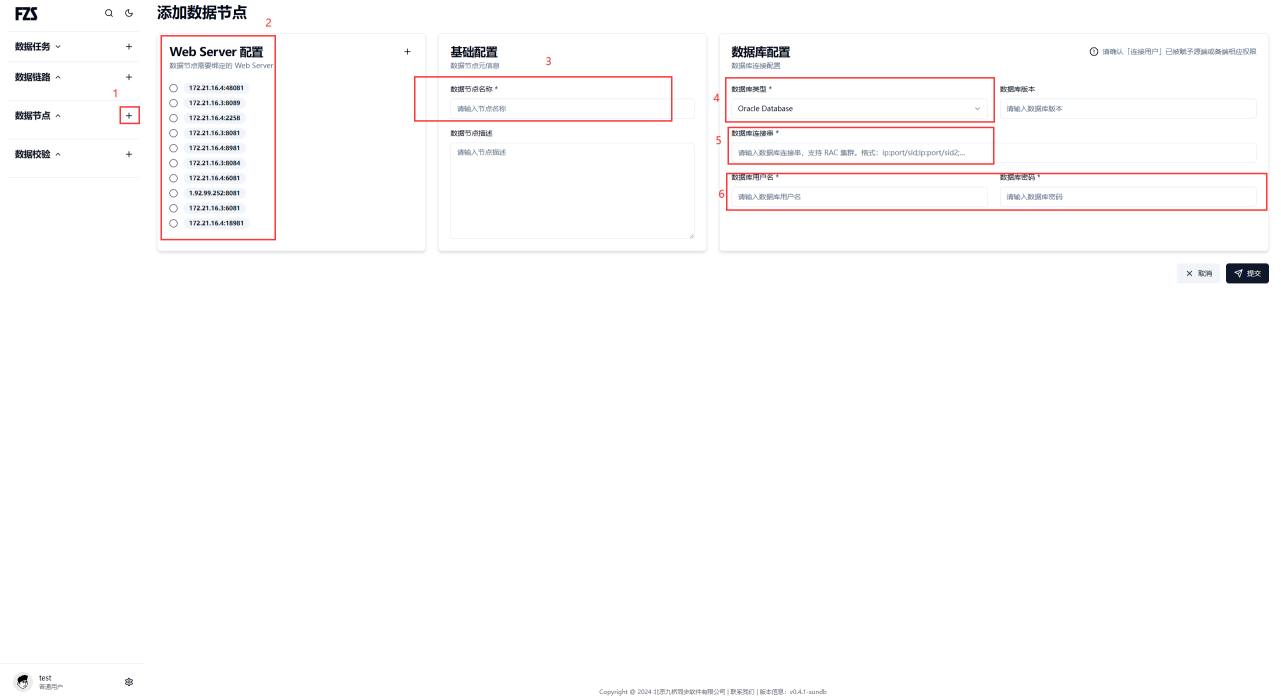


绑定授权



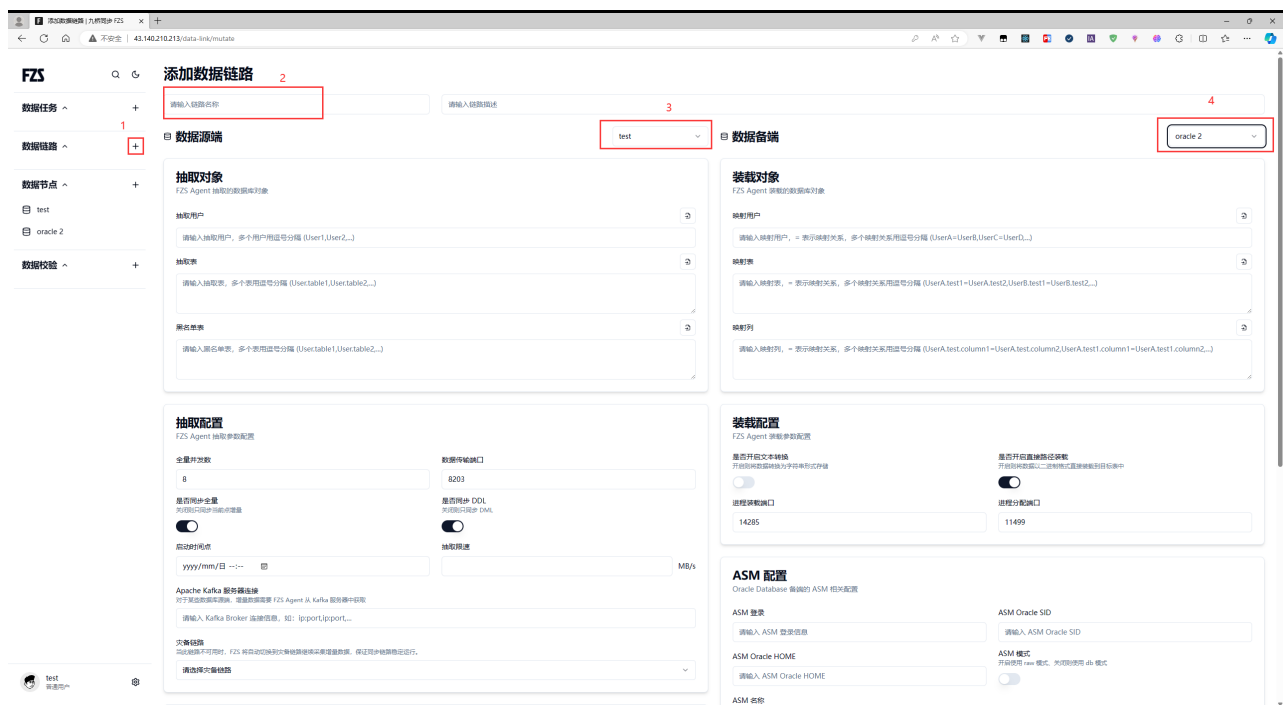
添加数据节点

选中源备端对应的webserver配置（注意ip和端口），添加数据节点名称，选择数据库的类型，添加数据库的连接串及fzs的数据库连接用户，点击提交



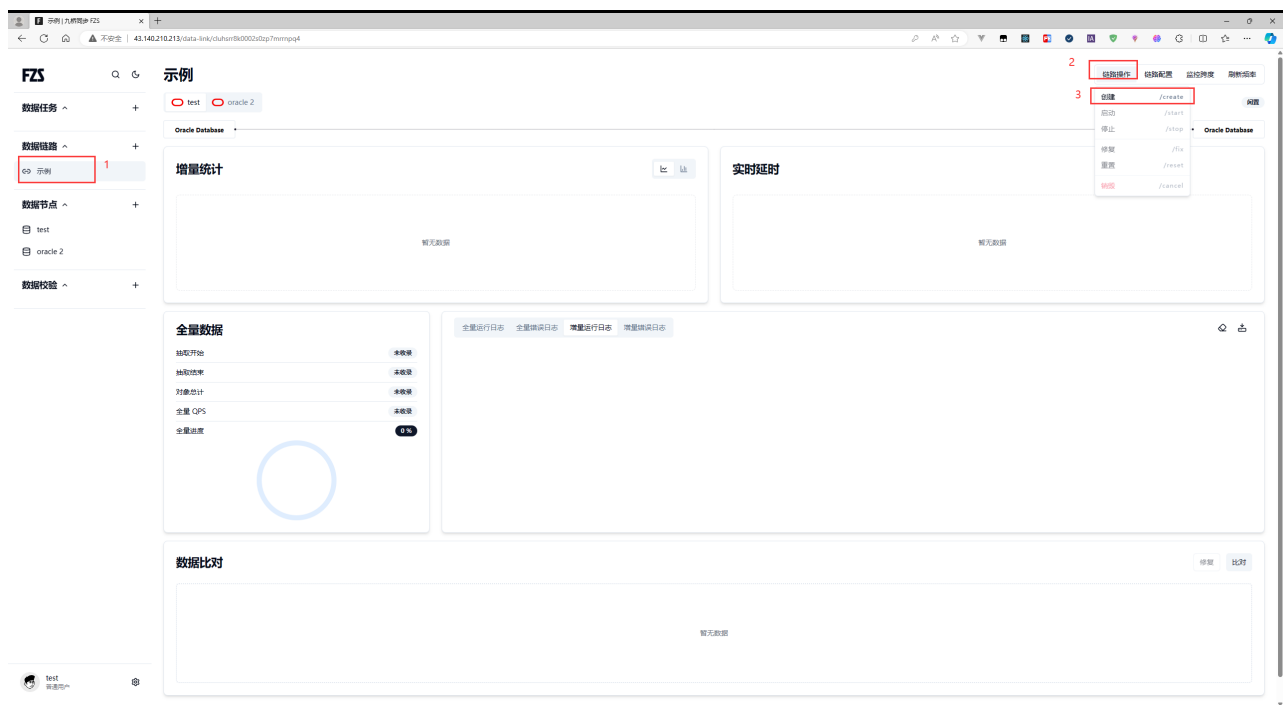
添加数据链路

点击添加链路，填写链路名称，选择源备端的数据节点，选择目标端的数据节点，点击提交

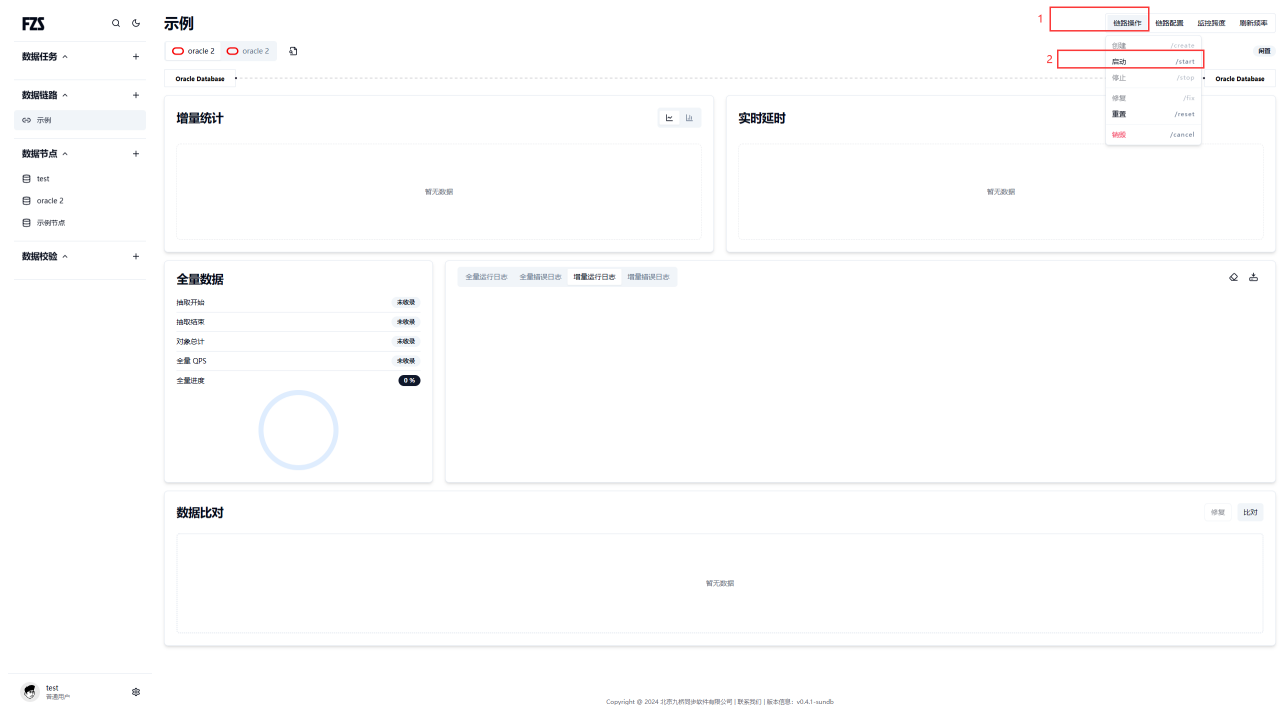


创建及启动数据链路

首先选中要创建的链路，点击创建,链路创建后，创建按钮置灰

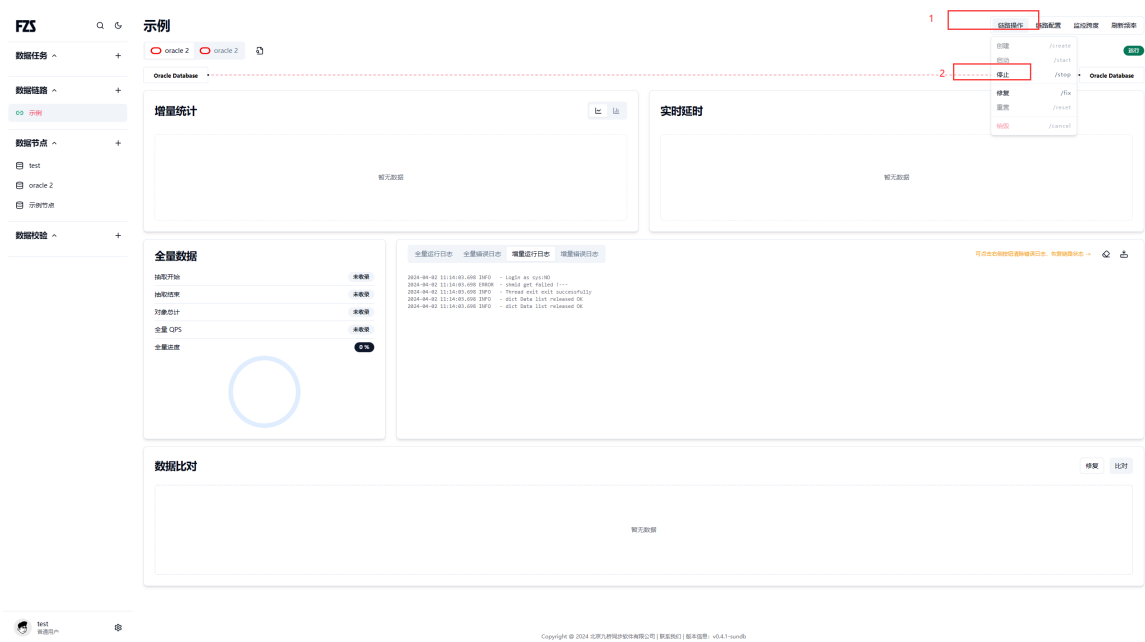


之后点击启动按钮，启动链路。链路启动后，链路状态从闲置变为运行

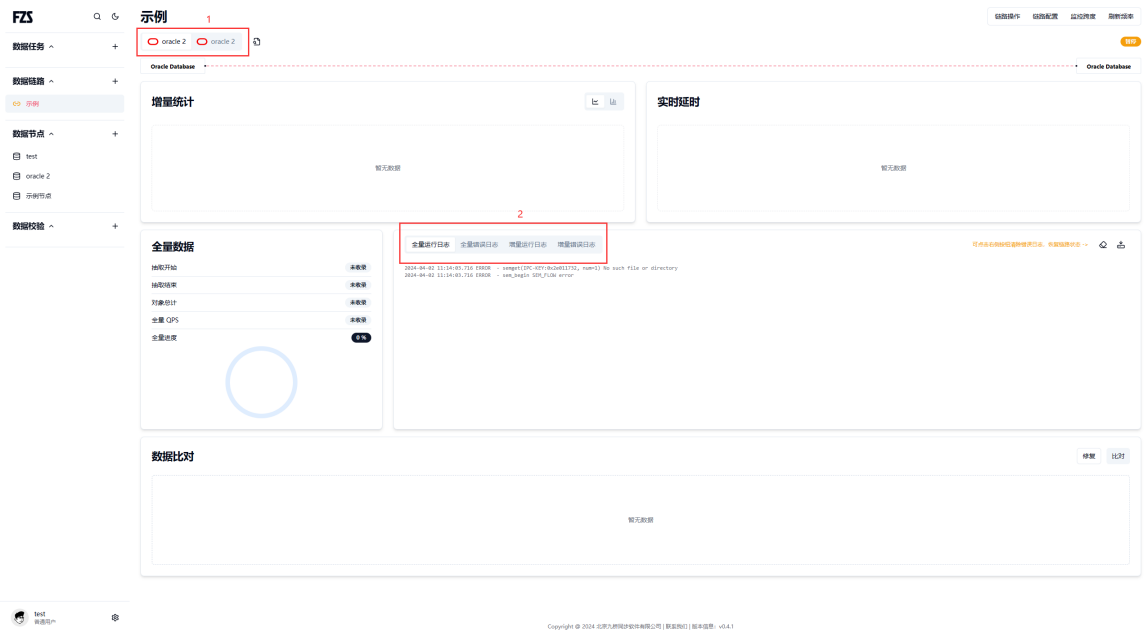


日常运维

- 重新全同步
 - 点击停止按钮，停止链路，链路状态变为暂停

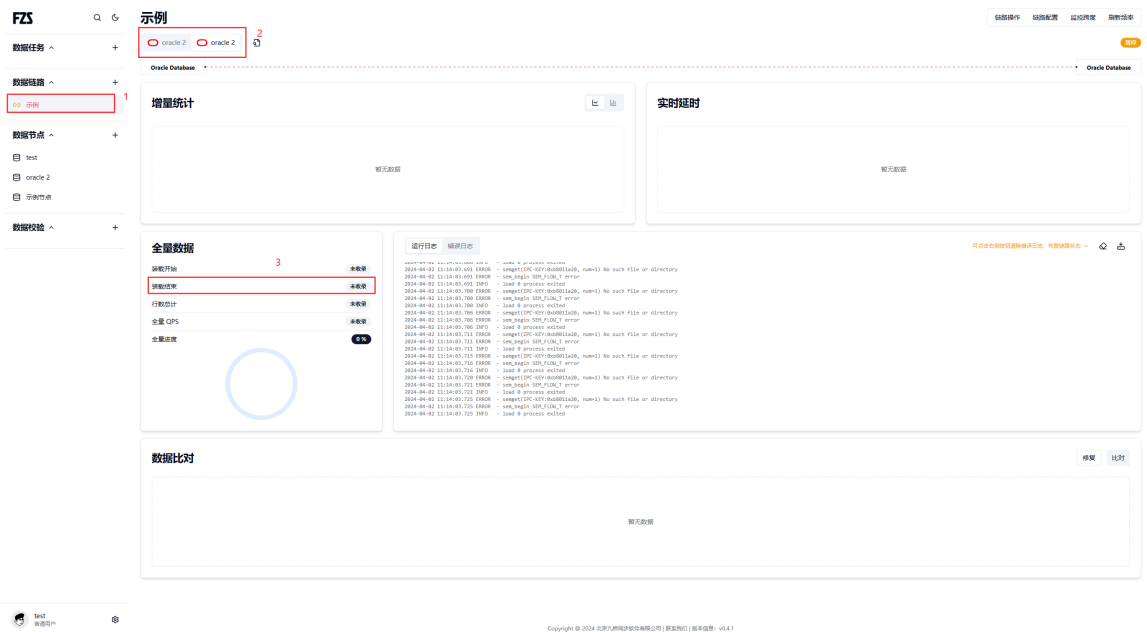


- 点击 链路操作-重置，重置链路，清除上次同步产生文件和日志。



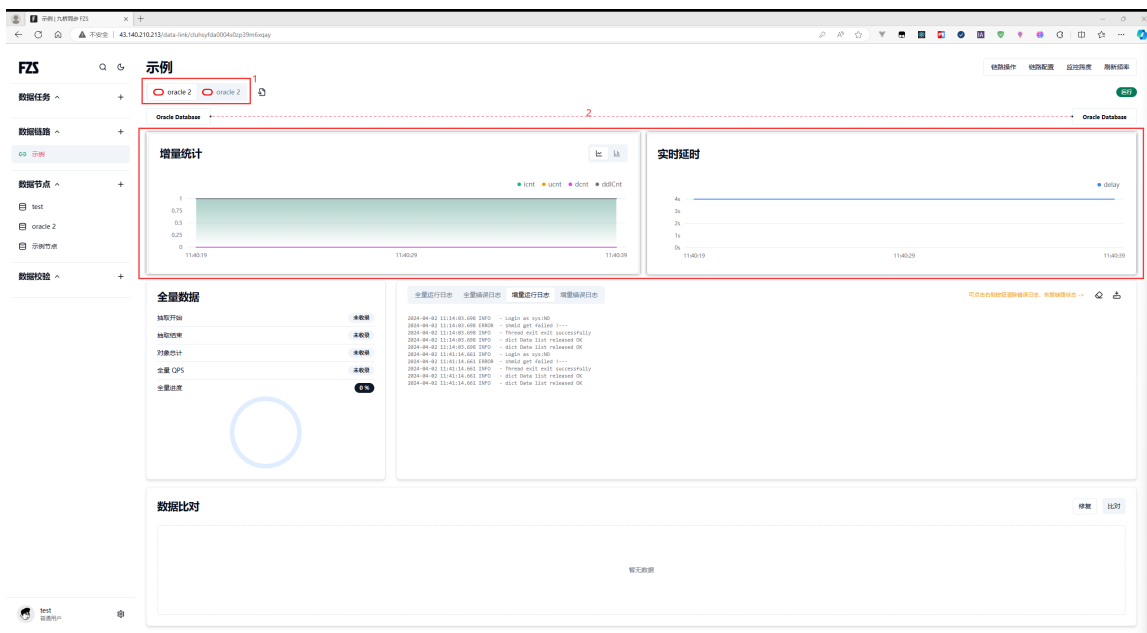
- 全同步完成标志

- 备端出现装载结束时间，说明全同步完成



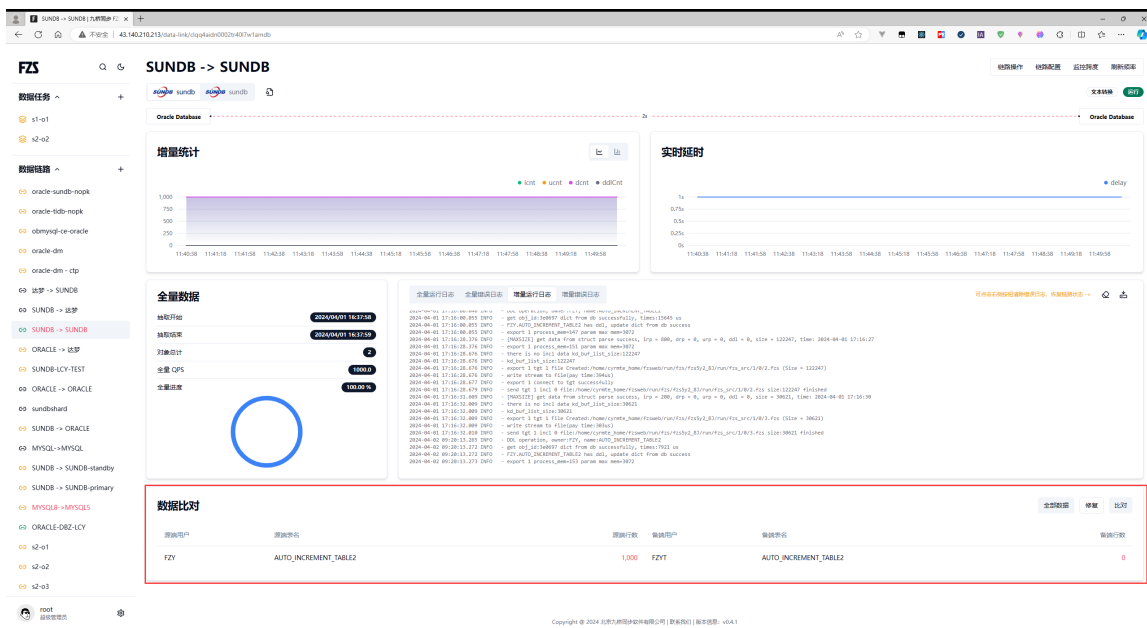
- 查看增量同步的状态

- 选中对应链路，点击源备端查看增量同步状态



- 数据比对

- 在链路详情页最下方数据比对卡片，点击比对，如果显示数据一致，则代表同步正常

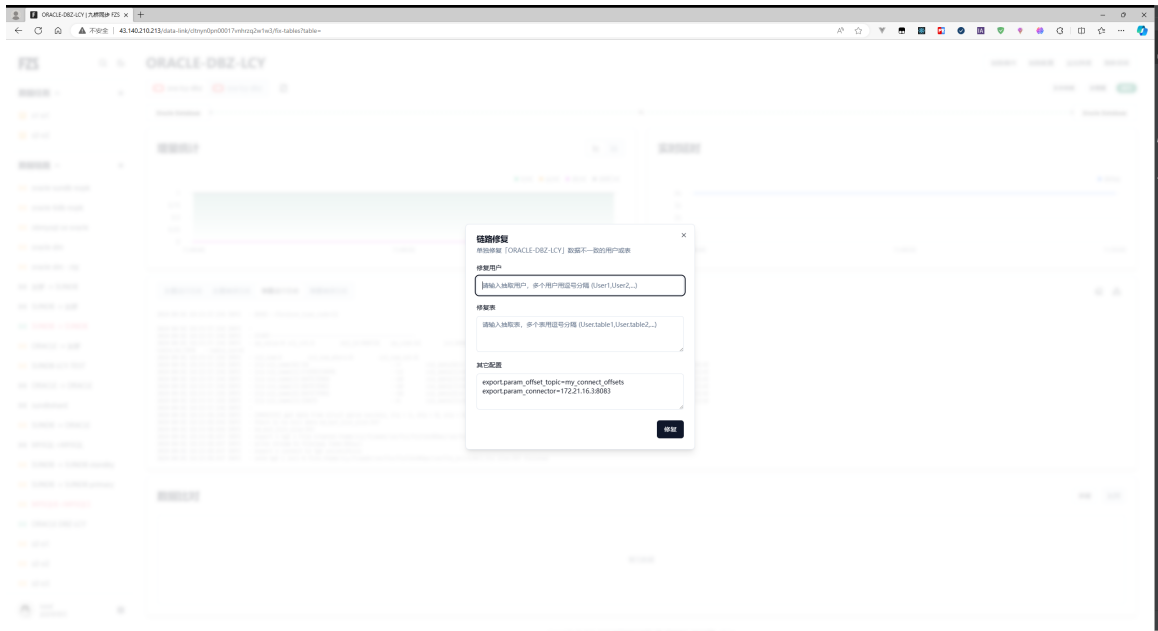


- 如果显示数据不一致，可点击卡片右上角修复，对数据不一致的表进行修复

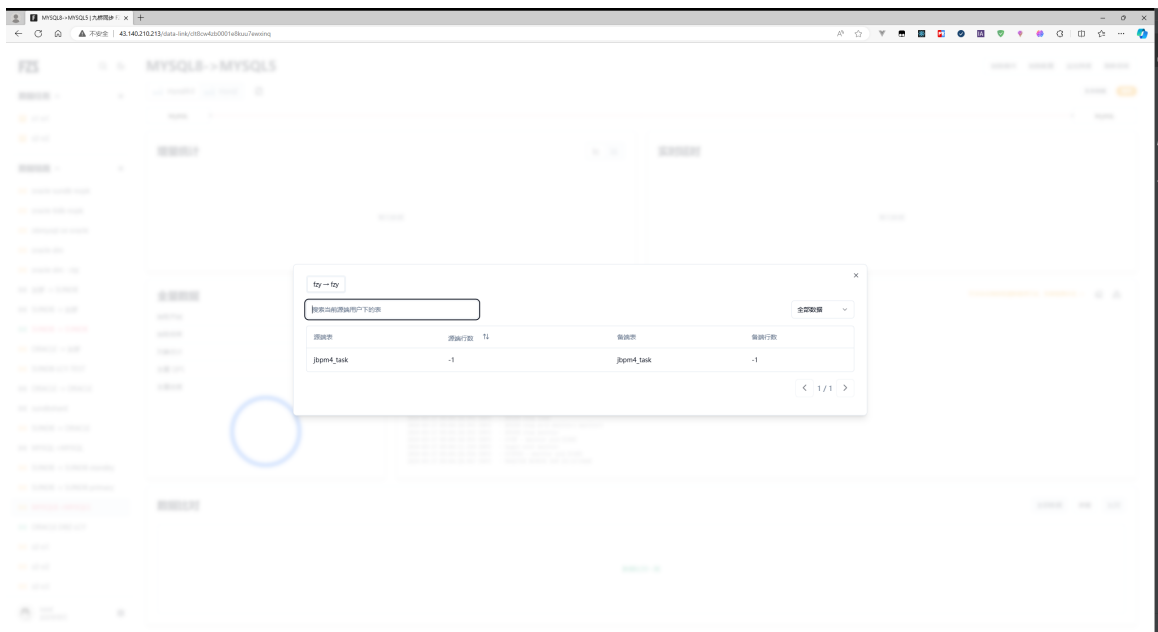


危险

修复操作需要在备端没有延迟和非业务期间进行



- 点击卡片右上角全部数据，查看所有表的数据比对情况



FZS Web

本章节将介绍 FZS Web 的部署以及使用方式。



使用的 FZS Web 版本和 FZS Web Server 版本必须一致，否则可能会出现不可预知的错误。

浏览器最低兼容版本

- Chrome 92+ (推荐)
- Edge 92+ (推荐)
- Firefox 90+
- Safari 15.4+
- Opera 78+

使用手册

本章节将介绍 FZS Web 各个功能模块的使用方式。

我们追求的是简单、易用、高效的使用体验，希望能够让您在使用 FZS Web 的过程中，能够感受到我们的用心。

数据节点

5 个项目

数据链路

12 个项目

数据任务

10 个项目

数据节点

在 FZS Web 中，数据节点是一个数据源的抽象，您可以将一个数据库配置为一个数据节点，然后在 FZS Web 中对该数据节点进行管理和操作。

数据节点是 FZS Web 的基本组成单元，我们支持多种类型的数据库，您可以通过数据节点来定义不同类型数据库的连接信息，且可以定义数据链路的源端节点和备端节点。

节点管理

- [添加节点](#)
- [编辑节点](#)
- [删除节点](#)

节点操作

- [Web Server 连通性校验](#)
- [数据库可用性校验](#)



提示

节点操作的功能可能在详情页其他位置有重复，这里只列出了节点操作入口，所实现的效果是相同的。

添加数据节点

功能概述

添加数据节点，用于保存数据源的连接信息。

操作步骤

1. 在页面的数据节点列表中，点击  按钮，进入添加数据节点界面。
2. 在 Web Server 配置中，选择一个可用的 Web Server。如果您还没有添加 Web Server，您可以点击卡片右上角  按钮，跳转到 Web Server 管理页面。
3. 在基础配置中，填写数据节点的名称、描述信息。
4. 在数据库配置中，填写数据库的连接信息，不同类型的数据库连接信息可能不同，请根据页面的提示信息填写。
5. 点击  按钮，会对数据节点的信息进行 [可用性校验](#)，如果校验通过，将会保存数据节点的信息，保存成功后将会跳转到数据节点详情页面。

提示

- 添加节点前请确保您有一个可用的 Web Server。
- 数据节点的名称必须是唯一的，不能与已有的数据节点重复。
- 请确保您的数据库可用，否则数据节点将无法正常保存。
- 带有 * 的字段为必填项。

编辑数据节点

功能概述

对于已添加的数据节点进行编辑，用于修改数据源的连接信息。

操作步骤

1. 在数据节点详情界面，点击节点配置，点击菜单中的 **编辑** 按钮。
2. 跳转到编辑数据节点界面，您可以修改数据节点的名称、描述信息、数据库连接信息。
3. 其余部分参见 [添加数据节点](#)

删除数据节点

功能概述

添加数据节点，用于保存数据源的连接信息。

操作步骤

1. 在数据节点详情界面，点击节点配置，点击菜单中的 **删除** 按钮，将会弹出确认框。
2. 点击确认将会删除该数据节点。
3. 删除完成后，将会跳转到首页。

提示

如果节点已被某个链路或校验使用，将无法进行删除。如需删除，请先解除链路或者校验和该节点的关联关系，或者删除相关链路和校验。

危险

已被删除的节点后将无法恢复。

Web Server 连通性校验

功能概述

快速验证数据节点的 Web Server 是否可用。

操作步骤

1. 在数据节点详情界面，点击节点操作按钮，点击菜单中的 **连通性** 按钮。
2. 如果出现 **连接成功** 的提示，说明 Web Server 连接正常，否则将会显示连接失败的原因

数据库可用性校验

功能概述

快速的验证数据库的连接状态及权限验证。

操作步骤

1. 在数据节点详情界面，点击节点操作按钮，点击菜单中的 **可用性**，选择源端或者备端。
2. 点击后将会验证当前数据库的权限是否满足当前节点作为源端或者备端的要求。
3. 如果出现 **连接成功** 的提示，说明数据库权限满足当前节点作为源端或者备端的要求，否则将会显示连接失败的原因。

提示

数据库可用性校验分为两种：**源端** 和 **备端**，因为在数据链路中，该节点作为源端和备端时所需的数据库权限可能不同，所以需要分别进行验证。

数据链路

在 FZS Web 中，数据链路是一个非常重要的概念，它是用来描述数据源之间的关系的，它可以帮助您快速的了解数据源之间的关系及其同步状态。

链路的界面及交互也是非常的复杂及重要的，下面将会对链路的界面及交互进行详细的介绍。

链路管理

- [添加链路](#)
- [编辑链路](#)
- [复制链路](#)
- [切换链路](#)
- [删除链路](#)

链路操作

- [创建链路](#)
- [启动链路](#)
- [停止链路](#)
- [修复链路](#)
- [重置链路](#)
- [销毁链路](#)

添加数据链路

功能概述

添加一条新的数据链路至系统中，您需要完成链路的基本配置，包括链路名称、数据源端和数据备端的数据节点，以及其他配置信息。

操作步骤

1. 在页面的数据链路列表中，点击数据链路标题旁的  按钮，进入添加数据链路界面。
2. 填写数据链路的名称，并选择源端和备端的数据节点。
3. 在选择数据节点后，会自动加载与之对应的其他配置信息，根据需要进行填写。
4. 点击  按钮，完成数据链路的添加。

提示

- 在数据链路中，数据源端和数据备端的节点必须是已经添加的数据节点。
- 在数据链路中，只有名称、数据源端和数据备端是必填项，其他参数可以根据需要进行填写。

编辑数据链路

功能概述

编辑数据链路指修改已存在的数据链路的配置内容。

操作步骤

1. 在数据链路详情界面，点击节点配置，点击菜单中的 **编辑** 按钮。
2. 跳转到编辑数据链路界面，您可以修改数据链路的名称、描述信息、源备端节点，及其他配置信息等。
3. 其余部分参见 [添加数据节点](#)

提示

- 如果链路在启动状态将无法编辑，需要先停止链路。详见 [停止数据链路](#)
- 在编辑链路时，源备端的配置项将会比添加链路多出一个 **Agent** 软件包的配置。在这个配置中，您可以将该链路中默认自带的 **Agent** 软件包，修改为系统中自己上传的 **Agent** 软件包。

复制数据链路

功能概述

复制已配置好的数据链路，复制后将会生成一条新的数据链路，您可以在新的数据链路中修改配置信息。

操作步骤

1. 在数据链路详情界面，点击链路配置，点击菜单中的 **复制** 按钮。
2. 跳转到新数据链路的详情界面，之后您可以进行其他的操作了。

❗ 信息

复制生成的链路名称是被复制的数据链路名称后面加上 **-副本**。

切换数据链路

功能概述

将数据链路中的数据源端和数据备端进行切换。切换后，数据源端将变为数据备端，数据备端将变为数据源端。

操作步骤

1. 在数据链路详情界面，点击链路配置，点击菜单中的 **切换** 按钮，将会弹出确认框。
2. 点击确认后数据源端和数据备端将会切换。
3. 切换完成后，将会重新刷新当前页面。

提示

如果链路在创建状态将无法切换，需要先停止销毁链路

危险

切换后，数据源端将变为数据备端，数据备端将变为数据源端。

删除数据链路

功能概述

删除已配置好的数据链路，删除后将无法恢复。

操作步骤

1. 在数据链路详情界面，点击链路配置，点击菜单中的 **删除** 按钮，将会弹出确认框。
2. 点击确认将会删除该数据链路。
3. 删除完成后，将会跳转到首页。



提示

如果链路已经被创建，将无法删除，需要先销毁链路。详见 [销毁数据链路](#)



危险

已被删除的节点后将无法恢复。

创建数据链路

功能概述

创建一条数据链路，创建后，数据链路处于未启动状态，创建链路的目的是为了将数据源和数据目标进行关联，从而实现数据的同步。

操作步骤

在数据链路详情界面，点击链路操作，点击菜单中的 **创建** 按钮。

启动数据链路

功能概述

开启一条数据链路的任务，启动后，数据链路将会会产生同步数据。

操作步骤

在数据链路详情界面，点击链路操作，点击菜单中的 **启动** 按钮。



提示

启动链路前需要先创建的数据链路。

停止数据链路

功能概述

停止后，数据链路不会再产生新的同步数据。

操作步骤

在数据链路详情界面，点击链路操作，点击菜单中的 停止 按钮即可。

 提示

只有已经启动的链路才能被停止。

修复数据链路

功能概述

修复在同步过程中出现的坏表，修复后，数据链路将会会产生同步数据。

操作步骤

1. 在数据链路详情界面，点击链路操作，点击菜单中的 **修复** 按钮。
2. 弹出链路修复的弹窗，按照需求填入需要修复的数据，点击 **修复** 按钮。
3. 弹窗关闭后，链路将会进行修复，修复过程中，将会产生同步数据。



修复只有在链路状态是运行状态和错误状态时才能进行。

灾备链路

功能概述

在新建或编辑链路时，可以选择灾备链路，灾备链路是一种特殊的配置项，在链路启动后，如果主链路出现异常，灾备链路将会自动启动，用于保障数据的同步。

操作步骤

1. 在数据链路配置界面 **抽取配置** 选项卡中，配置灾备链路
2. 点击 **提交** 按钮，完成数据链路的添加或编辑。
3. 之后，在数据链路详情界面，点击链路操作，点击菜单中的 **创建** 按钮，创建数据链路。
4. 创建完毕后再点击菜单中的 **启动 Cymte** 按钮，启动Cymte。



提示

主链路和灾备链路都需要进行 3,4 操作。

5. 之后正常启动主链路，当主链路出现异常时，灾备链路将会自动启动。

重置数据链路

功能概述

重置数据链路，重置后，数据链路的同步数据将会被清空，状态不会变更。

操作步骤

在数据链路详情界面，点击链路操作，点击菜单中的 重置 按钮。



启动或者暂停的链路才能被重置。

销毁数据链路

功能概述

销毁数据链路，销毁后，数据链路的所有数据将会被清空。

操作步骤

在数据链路详情界面，点击链路操作，点击菜单中的 **销毁** 按钮。



提示

链路被创建后才能够销毁



危险

此操作不可恢复，销毁后，数据链路的所有数据将会被清空！

数据任务

在 FZS Web 中，数据任务是为了实现链路的一对多和n条的一对一，让每个独立的数据链路能有一定的关联，达到集中管理的目的。

任务管理

- [添加任务](#)
- [编辑任务](#)
- [删除任务](#)

任务操作

- [创建任务](#)
- [启动任务](#)
- [停止任务](#)
- [修复任务](#)
- [重置任务](#)
- [销毁链路](#)
- [灾备任务](#)

添加数据任务

功能概述

添加一条新的数据任务至系统中，您需要完成数据任务的基本配置，包括任务名称、数据链路选择、任务策略以及其他配置信息。

操作步骤

1. 在页面的数据任务列表中，点击数据任务标题旁的  按钮，进入添加数据任务界面。
2. 填写数据任务的名称，并选择需要添加到当前创建任务的数据链路。
3. 之后，您可以根据需要进行配置，包括任务策略、其他配置信息等。
4. 点击  按钮，完成数据任务的添加。

编辑数据任务



数据任务在运行状态下无法编辑，需要先停止数据任务。详见 [停止数据任务](#)

功能概述

编辑数据任务指修改已存在的数据任务的配置内容。

操作步骤

1. 在数据任务详情界面，点击任务配置，点击菜单中的 **编辑** 按钮。
2. 跳转到编辑数据任务界面，您可以修改数据任务的名称、描述信息、数据链路、任务策略、其他配置信息等。
3. 其余部分参见 [添加数据任务](#)

注意事项

- 。

删除数据任务

提示

如果任务已经被创建，将无法删除，需要先销毁任务。详见 [销毁数据任务](#)

危险

已被删除的任务后将无法恢复。

功能概述

删除已配置好的数据任务，删除后将无法恢复。

操作步骤

1. 在数据任务详情界面，点击任务配置，点击菜单中的  按钮，将会弹出确认框。
2. 点击确认将会删除该数据任务。
3. 删除完成后，将会跳转到首页。

创建数据任务

功能概述

创建数据任务，创建后，数据任务将会创建一条新的进程用来同时控制内部的所有链路。

操作步骤

在数据任务详情界面，点击任务操作，点击菜单中的 **创建** 按钮。

启动数据任务



提示

启动任务前需要先创建的数据任务。

功能概述

启动数据任务，启动后，数据任务内的所有链路将会产生新的同步数据。

操作步骤

在数据任务详情界面，点击任务操作，点击菜单中的 **启动** 按钮。

停止数据任务

功能概述

停止后，数据任务内的所有链路将不会再产生新的同步数据。

操作步骤

在数据任务详情界面，点击任务操作，点击菜单中的 停止 按钮即可。

修复数据任务



修复只有在任务状态是运行状态和错误状态时才能进行。

功能概述

修复数据任务中的数据链路在同步过程中出现的坏表，修复后，任务中的数据链路将会产生同步数据。

操作步骤

1. 在数据任务详情界面，点击任务操作，点击菜单中的 **修复** 按钮。
2. 弹出任务修复的弹窗，按照需求填入需要修复的数据，点击 **修复** 按钮。
3. 弹窗关闭后，任务将会进行修复，修复过程中，将会产生同步数据。

重置数据任务



创建后的任务且不在运行状态下任务才能被重置。

功能概述

重置数据任务，重置后，数据任务内的数据链路的上一次同步过程中产生的同步信息将会被清空，状态不会变更。

操作步骤

在数据任务详情界面，点击任务操作，点击菜单中的 **重置** 按钮。

销毁数据任务



销毁后，数据任务中的所有数据将会被清空！所有数据将无法恢复，请谨慎操作。

功能概述

销毁数据链路，销毁后，数据链路的所有数据将会被清空。

操作步骤

在数据链路详情界面，点击链路操作，点击菜单中的 **销毁** 按钮。

灾备任务

功能概述

在新建或编辑任务时，可以选择灾备任务，灾备任务是一种特殊的配置项，在任务启动后，如果主任务出现异常，灾备任务将会自动启动，用于保障数据的同步。

操作步骤

1. 在数据任务配置界面 **其他配置** 选项卡中，配置灾备任务
2. 点击 **提交** 按钮，完成数据任务的添加或编辑。
3. 之后，在数据任务详情界面，点击任务操作，点击菜单中的 **创建** 按钮，创建数据任务。
4. 创建完毕后再点击菜单中的 **启动 Cyrmte** 按钮，启动Cyrmte。



主任务和灾备任务都需要进行 3,4 操作。

FZS Web Server

本章节将介绍 FZS Web Server 的部署以及 API 调用方式。



使用的 FZS Web 版本和 FZS Web Server 版本必须一致，否则可能会出现不可预知的错误。

API 文档

FZS Web Server 提供 API 功能，你可以通过 API 对 FZS 同步工具进行运维操作。

你可以通过 API 完成 FZS 同步工具的如下运维操作：

数据节点管理

- [测试数据节点连接](#)
- [获取数据节点授权](#)
- [测试数据库可用性](#)
- [上传数据节点授权](#)

数据链路管理

备注

数据链路相关 API 需要在源端节点与备端节点上分别调用。

- [创建数据链路](#)
- [启动数据链路](#)
- [停止数据链路](#)
- [重置数据链路](#)
- [修复数据链路](#)
- [销毁数据链路](#)
- [获取 FZS 软件路径](#)
- [获取链路全量同步信息](#)
- [获取数据链路增量延时](#)
- [获取数据链路增量计数](#)

- 获取数据链路日志
- 清除链路错误日志
- 下载数据链路日志
- 获取 FZS 软件包
- 更新 FZS 软件包

测试数据节点连接

该接口旨在测试数据节点连接，请求成功会返回对应节点的状态信息。

请求 URI

GET /

使用样例

以下请求会获取 IP 地址为 `172.21.16.3`，端口号为 `8089` 的 FZS Web Server 节点的状态信息。

```
curl -X GET http://172.21.16.3:8089
```

```
{
  "version": "v0.1.2",
  "compileTime": "23-10-08 16:27:24",
  "gitHash": "915a23f"
}
```

以上返回信息的字段解释如下：

- `version`：当前 FZS Web Server 版本号。
- `compileTime`：编译时间。
- `git_hash`：Git 哈希值。

获取数据节点授权

该接口旨在获取数据节点的授权，请求成功会返回对应节点的授权信息。

请求 URI

GET /license-info

使用样例

以下请求会获取 IP 地址为 172.21.16.3，端口号为 8089 的 FZS Web Server 节点的授权信息。

```
curl -X GET http://172.21.16.3:8089/license-info
```

```
{
  "host": "007f0100",
  "machine": "x86_64",
  "RemainingDay": -1
}
```

以上返回信息的字段解释如下：

- host：主机 ID
- machine：主机架构
- RemainingDay：授权剩余时间，-1 则表示永久授权

测试数据库可用性

该接口旨在测试数据库可用性，请求成功会返回对应节点的数据库可用性信息。

请求 URI

POST /test-db-validity

参数说明

```
{
  "agentType": 0,
  "databaseConfig": [
    {
      "host": "string",
      "userName": "string",
      "passWord": "string",
      "dbType": "string"
    }
  ]
}
```

参数说明如下：

参数名	说明
agentType	NUMBER 类型，软件包的类型：0 为源端，1 为备端。
databaseConfig	数据库连接参数。

databaseConfig 参数说明如下：

参数名	说明
<code>host</code>	<code>STRING</code> 类型，格式： <code>IP/端口</code> 。
<code>userName</code>	<code>STRING</code> 类型，数据库用户名。
<code>passWord</code>	<code>STRING</code> 类型，数据库登录密码。
<code>dbType</code>	<code>STRING</code> 类型，数据库类型。

使用样例

以下请求会测试一个 `oracle` 数据库的源端可用性。

```
curl -i -X POST \  
  -H "Content-Type:application/json" \  
  -d \  
'{  
  "agentType": 0,  
  "databaseConfig": [  
    {  
      "host": "172.21.16.4:1621/lhr11g",  
      "userName": "***",  
      "passWord": "***",  
      "dbType": "oracle"  
    }  
  ]  
' \  
'http://172.21.16.3:8089/test-db-validity'
```

响应体格式

```
{ "stateCode": 0, "stateMsg": "" }
```

参数说明如下：

参数名	说明
stateCode	NUMBER 类型，错误码。0 表示成功
stateMsg	STRING 类型，错误消息

上传数据节点授权

该接口可以上传数据节点上传文件，上传成功会返回相应的状态信息。

请求 URI

POST /upload-license

参数说明

```
{
  "name": "string",
  "filename": "string",
  "file": "string"
}
```

参数说明如下：

参数名	说明
<code>name</code>	<code>STRING</code> 类型，授权文件名称。
<code>filename</code>	<code>STRING</code> 类型，授权文件名称。
<code>file</code>	<code>STRING</code> 类型，base64 字符串，授权文件的内容。

使用样例

以下请求会上传一个 `sn_nolimit.txt` 授权文件，上传成功会返回 `OK`。

```
curl -i -X POST \  
  -H "Content-Type:multipart/form-data" \  
  -F "name=sn_nolimit" \  
  -F "filename=sn_nolimit" \  
  -F "file=@\"./sn_nolimit.txt\";type=text/  
plain;filename=\"sn_nolimit.txt\"" \  
  'http://172.21.16.4:8089/upload-license'
```

响应体格式

OK

创建数据链路

该接口可以在某个数据节点上创建一条链路，创建成功会返回相应的状态信息。

请求 URI

POST /create

参数说明

```
{
  "id": "string",
  "dataPort": "string",
  "ip": "string",
  "agentType": 0,
  "databaseConfig": [
    {
      "host": "string",
      "userName": "string",
      "passWord": "string",
      "dbType": "string"
    }
  ],
  "sourceConfig": {
    "mapUser": "string",
    "mapTable": "string",
    "mapNotTable": "string",
    "fullSyncCnt": "string",
    "mustFullSync": "string",
    "mustSyncDDL": "string",
    "rateLimit": "string",
    "beginTime": "string",
    "ifFzsText": "string",
    "obIncSyncUsername": "string",
```

参数说明如下：

参数名	说明
<code>id</code>	<code>STRING</code> 类型，FZS Agent ID。此 ID 用于生成 Web Server 上的 FZS Agent 路径。
<code>dataPort</code>	<code>STRING</code> 类型，同步链路的数据传输端口。
<code>ip</code>	<code>STRING</code> 类型，同步链对端节点的 IP 地址。例如：如果此接口向 Web Server 请求，则填写备端 Web Server 的 IP 地址，反之亦然。
<code>agentType</code>	<code>STRING</code> 类型，FZS Agent 的类型， <code>0</code> 则代表源端， <code>1</code> 则代表备端。
<code>databaseConfig</code>	Web Server 数据库连接参数。
<code>sourceConfig</code>	Web Server 源端同步参数。注意：仅当请求的是源端 Web Server 时添加。
<code>sinkConfig</code>	Web Server 备端同步参数。注意：仅当请求的是备端 Web Server 时添加。
<code>asmLogin</code>	<code>STRING</code> 类型，Oracle ASM 登录字符串，注意：仅当数据节点的数据库类型为 <code>Oracle</code> 时添加。
<code>asmOracleSID</code>	<code>STRING</code> 类型，Oracle ASM 登录 SID，注意：仅当数据节点的数据库类型为 <code>Oracle</code> 时添加。
<code>asmOracleHome</code>	<code>STRING</code> 类型，Oracle ASM Home 路径，注意：仅当数据节点的

参数名	说明
	数据库类型为 <code>Oracle</code> 时添加。
<code>asmMode</code>	<code>STRING</code> 类型，Oracle ASM 模式， <code>0</code> 代表 <code>db</code> 模式， <code>1</code> 代表 <code>raw</code> 模式。注意：仅当数据节点的数据库类型为 <code>Oracle</code> 时添加。
<code>asmConfig</code>	ASM 配置参数。
<code>execSwitch</code>	<code>STRING</code> 类型，是否执行源端备端的灾备切换， <code>1</code> 则代表执行， <code>0</code> 则代表不执行。
<code>extraConfig</code>	FZS Agent 的其他配置项。

`databaseConfig` 参数说明如下：

参数名	说明
<code>host</code>	<code>STRING</code> 类型，格式： <code>IP/端口</code> 。
<code>userName</code>	<code>STRING</code> 类型，数据库用户名。
<code>passWord</code>	<code>STRING</code> 类型，数据库登录密码。
<code>dbType</code>	<code>STRING</code> 类型，数据库类型。

`sourceConfig` 参数说明如下：

参数名	说明
<code>mapUser</code>	<code>STRING</code> 类型，需要同步的用户名。格式： <code>USER1,USER2,USER3</code> ，多个用户逗号分隔
<code>mapTable</code>	<code>STRING</code> 类型，需要同步的表名。格式： <code>USER.TABLE1,USER.TABLE2,USER.TABLE3</code> ，多张表逗号分隔
<code>mapNotTable</code>	<code>STRING</code> 类型，需要排除的表名。格式： <code>USER.TABLE1,USER.TABLE2,USER.TABLE3</code> ，多张表逗号分隔
<code>fullSyncCnt</code>	<code>STRING</code> 类型，全量同步的并发数。默认值为 <code>8</code>
<code>mustFullSync</code>	<code>STRING</code> 类型，是否进行全量同步。是则填写 <code>1</code> ，否则填写 <code>0</code>
<code>mustSyncDDL</code>	<code>STRING</code> 类型，是否同步 DDL 语句。是则填写 <code>1</code> ，否则填写 <code>0</code>
<code>rateLimit</code>	<code>STRING</code> 类型，同步限速。
<code>beginTime</code>	<code>STRING</code> 类型，同步指定历史时间。注意：此时间不能大于当前时间
<code>ifFzsText</code>	<code>STRING</code> 类型，是否开启文本转换。是则填写 <code>1</code> ，否则填写 <code>0</code>
<code>obIncSyncUsername</code>	<code>STRING</code> 类型，OceanBase 增量模块使用的用户名。注意：

参数名	说明
	仅当源端数据节点类型为 <code>OceanBase</code> 时添加。
<code>obIncSyncPassword</code>	<code>STRING</code> 类型，OceanBase 增量模块使用的密码。注意：仅当源端数据节点类型为 <code>OceanBase</code> 时添加。
<code>obIncSyncConfigURL</code>	<code>STRING</code> 类型，OceanBase 增量模块使用的集群配置 URL。注意：仅当源端数据节点类型为 <code>OceanBase</code> 时添加。

`sinkConfig` 参数说明如下：

参数名	说明
<code>mapUser</code>	<code>STRING</code> 类型，需要同步的用户名。格式： <code>USER1,USER2,USER3</code> ，多个用户逗号分隔
<code>mapTable</code>	<code>STRING</code> 类型，需要同步的表名。格式： <code>USER.TABLE1,USER.TABLE2,USER.TABLE3</code> ，多张表逗号分隔
<code>mapColumn</code>	<code>STRING</code> 类型，需要同步的列名。格式： <code>USER.TABLE.COL1,USER.TABLE.COL2,USER.TABLE.COL3</code> ，多个列逗号分隔
<code>ifFzsText</code>	<code>STRING</code> 类型，是否开启文本转换。是则填写 <code>1</code> ，否则填写 <code>0</code>
<code>ifDirectLoad</code>	<code>STRING</code> 类型，是否开启直接路径装载。是则填写 <code>1</code> ，否则填写 <code>0</code>

参数名	说明
<code>portToQueue</code>	<code>STRING</code> 类型，备端进程装载端口。
<code>portToAlloc</code>	<code>STRING</code> 类型，备端进程分配端口。
<code>fullSyncCnt</code>	<code>STRING</code> 类型，全量同步的并发数。注意：此参数的值须与 <code>sourceConfig</code> 中的 <code>fullSyncCnt</code> 相同

`asmConfig` 参数说明如下：

参数名	说明
<code>asmDev</code>	<code>STRING</code> 类型，ASM 路径。
<code>asmDisk</code>	<code>STRING</code> 类型，ASM 名称。

`extraConfig` 参数说明如下：

[点我查看](#)

使用样例

提示

创建一条完整的数据链路，需要在源端节点与备端节点上分别调用 `/create` 接口。

以下请求会在 `172.21.16.3:8089` 的 Web Server 上创建一个源端 FZS Agent，创建成功则会返回相应状态。

```
curl -i -X POST \
  -H "Content-Type:application/json" \
  -d \
  '{
    "id": "V76b2z",
    "dataPort": "8203",
    "ip": "192.168.0.24",
    "agentType": 0,
    "asmLogin": "",
    "asmOracleSID": "",
    "asmOracleHome": "",
    "asmMode": "db",
    "asmConfig": [
      {
        "asmDev": "",
        "asmDisk": ""
      }
    ],
    "databaseConfig": [
      {
        "host": "192.168.0.24:1521/orcl",
        "userName": "fzs1",
        "passWord": "fzs1",
        "dbType": "oracle"
      }
    ],
    "sourceConfig": {
      "dataPort": "8203",
      "mapUser": "",
      "mapTable": "",
      "mapNotTable": "",
      "fullSyncCnt": "8",
      "mustfullSync": "1",
      "mustSyncDDL": "1",
      "rateLimit": null,
      "asmLogin": "",
      "asmOracleSID": "",
      "asmOracleHome": "",
    }
  }'
```

以下请求会在 `172.21.16.4:8089` 的 Web Server 上创建一个备端 FZS Agent，创建成功则会返回相应状态。

```
curl -i -X POST \
  -H "Content-Type:application/json" \
  -d \
  '{
    "id": "-ChB4l",
    "dataPort": "8203",
    "ip": "192.168.0.24",
    "agentType": 1,
    "asmLogin": "",
    "asmOracleSID": "",
    "asmOracleHome": "",
    "asmMode": "db",
    "asmConfig": [
      {
        "asmDev": "",
        "asmDisk": ""
      }
    ],
    "databaseConfig": [
      {
        "host": "192.168.0.24:1521/orcl",
        "userName": "fzs1",
        "passWord": "fzs1",
        "dbType": "oracle"
      }
    ],
    "sinkConfig": {
      "id": "69ab59fd-72f7-4f71-a349-032c6f91e96b",
      "usedById": "clonqfx6i0001tzso2l208qob",
      "mapUser": "",
      "mapTable": "",
      "mapColumn": "",
      "ifFzsText": "0",
      "ifDirectLoad": "1",
      "portToQueue": "18409",
```

响应体格式

```
{ "stateCode": 0, "stateMsg": "" }
```

参数说明如下：

参数名	说明
stateCode	NUMBER 类型，错误码。0 表示成功
stateMsg	STRING 类型，错误消息

启动数据链路

该接口可以在某个数据节点上启动一个 FZS Agent，启动成功会返回相应的状态信息。

请求 URI

POST /start

参数说明

同 /create 接口参数。

使用样例

同 /create 接口使用样例，将接口路径中的 /create 替换为 /start 即可。

响应体格式

```
{ "stateCode": 0, "stateMsg": "" }
```

参数说明如下：

参数名	说明
stateCode	NUMBER 类型，错误码。0 表示成功
stateMsg	STRING 类型，错误消息

停止数据链路

该接口可以停止某个数据节点上的 FZS Agent，停止成功会返回相应的状态信息。

请求 URI

POST /stop

参数说明

```
{  
  "id": "string"  
}
```

参数说明如下：

参数名	说明
<code>id</code>	<code>STRING</code> 类型，FZS Agent ID。

使用样例

以下请求会停止 ID 为 `V76b2z` 的 FZS Agent。

```
curl -i -X POST \  
  -H "Content-Type:application/json" \  
  -d \  
'{
```

响应体格式

```
{ "stateCode": 0, "stateMsg": "" }
```

参数说明如下：

参数名	说明
stateCode	NUMBER 类型，错误码。0 表示成功
stateMsg	STRING 类型，错误消息

重置数据链路

该接口可以重置某个数据节点上的 FZS Agent，重置成功会返回相应的状态信息。

请求 URI

POST /reset

参数说明

```
{  
  "id": "string"  
}
```

参数说明如下：

参数名	说明
<code>id</code>	<code>STRING</code> 类型，FZS Agent ID。

使用样例

以下请求会重置 ID 为 `V76b2z` 的 FZS Agent。

```
curl -i -X POST \  
  -H "Content-Type:application/json" \  
  -d \  
'{
```

响应体格式

```
{ "stateCode": 0, "stateMsg": "" }
```

参数说明如下：

参数名	说明
stateCode	NUMBER 类型，错误码。0 表示成功
stateMsg	STRING 类型，错误消息

修复数据链路



提示

此接口仅需要向[源端](#) Web Server 请求即可。

该接口可以修复某数据链路的某些用户或者表，使其重新进行全同步，调用成功会返回相应的状态信息。

请求 URI

POST /fix

参数说明

同 `/create` [接口参数](#)。

使用样例

同 `/create` [接口使用样例](#)，将接口路径中的 `/create` 替换为 `/fix` 即可。

响应体格式

```
{ "stateCode": 0, "stateMsg": "" }
```

参数说明如下：

参数名	说明
stateCode	NUMBER 类型，错误码。0 表示成功
stateMsg	STRING 类型，错误消息

销毁数据链路

该接口可以销毁某个数据节点上的 FZS Agent，销毁成功会返回相应的状态信息。

请求 URI

POST /cancel

参数说明

```
{  
  "id": "string"  
}
```

参数说明如下：

参数名	说明
<code>id</code>	<code>STRING</code> 类型，FZS Agent ID。

使用样例

以下请求会销毁 ID 为 `V76b2z` 的 FZS Agent。

```
curl -i -X POST \  
  -H "Content-Type:application/json" \  
  -d \  
'{
```

响应体格式

```
{ "stateCode": 0, "stateMsg": "" }
```

参数说明如下：

参数名	说明
stateCode	NUMBER 类型，错误码。0 表示成功
stateMsg	STRING 类型，错误消息

获取 FZS 软件路径

该接口可以获取 FZS Agent 的运行路径。

请求 URI

POST /run-path

参数说明

```
{  
  "id": "string"  
}
```

参数说明如下：

参数名	说明
<code>id</code>	<code>STRING</code> 类型，FZS Agent ID。

使用样例

以下请求会获取 ID 为 `V76b2z` 的 FZS Agent 运行路径。

```
curl -i -X POST \  
  -H "Content-Type:application/json" \  
  -d \  
'{
```

响应体格式

```
/home/web/fzsweb/run/fzs/fzsV76b2z
```

获取链路全量同步信息

该接口可以获取 FZS Agent 的全量同步实时信息。

请求 URI

POST /full-sync

参数说明

```
{
  "id": "string"
}
```

参数说明如下：

参数名	说明
<code>id</code>	<code>STRING</code> 类型，FZS Agent ID。

使用样例

以下请求会获取 ID 为 `8AouMB` 的 FZS Agent 的全量信息。

```
curl -i -X POST \
  -H "Content-Type:application/json" \
  -d \
  '{
```

响应体格式

```
{
  "id": "8AouMB",
  "checkPoint": 1699335671,
  "startTime": null,
  "endTime": null,
  "fullStart": 1699330259,
  "fullEnd": 1699330280,
  "fullTotalCnt": 24,
  "fullFinishCnt": 24
}
```

参数说明如下：

参数名	说明
id	STRING 类型，FZS Agent ID。
checkPoint	NUMBER 类型，当前检查点 unix 时间戳
startTime	NUMBER 类型，查询起始时间 unix 时间戳
endTime	NUMBER 类型，查询结束时间 unix 时间戳
fullStart	NUMBER 类型，全量开始时间 unix 时间戳
fullEnd	NUMBER 类型，全量结束时间 unix 时间戳
fullTotalCnt	NUMBER 类型，全量总计行数。
fullFinishCnt	NUMBER 类型，全量完成行数。

获取数据链路增量延时

该接口可以获取 FZS Agent 的增量同步延时信息。

请求 URI

POST /delay

参数说明

```
{
  "id": "string",
  "startTime": 1699336083,
  "endTime": 1699336083
}
```

参数说明如下：

参数名	说明
id	STRING 类型，FZS Agent ID。
startTime	NUMBER 类型，查询起始时间 unix 时间戳。
endTime	NUMBER 类型，查询结束时间 unix 时间戳。

使用样例

以下请求会获取 ID 为 8AouMB 的 FZS Agent 的增量同步延时信息。

```
curl -i -X POST \
  -H "Content-Type:application/json" \
  -d \
  '{
    "id": "8AouMB"
  }' \
  'http://172.21.16.4:48081/delay'
```

响应体格式

```
[
  {
    "id": "8AouMB",
    "checkPoint": 1699336184,
    "startTime": null,
    "endTime": null,
    "delay": 1
  }
]
```

响应体返回一个数组，数组中每个元素为一个对象，对象中的参数说明如下：

参数名	说明
id	STRING 类型，FZS Agent ID。
checkPoint	NUMBER 类型，当前检查点 <code>unix</code> 时间戳
startTime	NUMBER 类型，查询起始时间 <code>unix</code> 时间戳
endTime	NUMBER 类型，查询结束时间 <code>unix</code> 时间戳
delay	NUMBER 类型，延时数据，单位为秒

获取数据链路增量计数

该接口可以获取 FZS Agent 的增量链路增量计数。

请求 URI

POST /count

参数说明

```
{
  "id": "string",
  "startTime": 1699336083,
  "endTime": 1699336083
}
```

参数说明如下：

参数名	说明
id	STRING 类型，FZS Agent ID。
startTime	NUMBER 类型，查询起始时间 unix 时间戳。
endTime	NUMBER 类型，查询结束时间 unix 时间戳。

使用样例

以下请求会获取 ID 为 8AouMB 的 FZS Agent 的增量链路增量计数。

```
curl -i -X POST \
  -H "Content-Type:application/json" \
  -d \
  '{
    "id": "8AouMB"
  }' \
  'http://172.21.16.4:48081/count'
```

响应体格式

```
[
  {
    "id": "8AouMB",
    "checkPoint": 1699336365,
    "startTime": null,
    "endTime": null,
    "icnt": 6634,
    "ucnt": 603604,
    "dcnt": 603604,
    "ddlCnt": 603
  }
]
```

响应体返回一个数组，数组中每个元素为一个对象，对象中的参数说明如下：

参数名	说明
<code>id</code>	<code>STRING</code> 类型，FZS Agent ID。
<code>checkPoint</code>	<code>NUMBER</code> 类型，当前检查点 <code>unix</code> 时间戳
<code>startTime</code>	<code>NUMBER</code> 类型，查询起始时间 <code>unix</code> 时间戳

参数名	说明
endTime	NUMBER 类型，查询结束时间 unix 时间戳
icnt	NUMBER 类型，增量插入计数
ucnt	NUMBER 类型，增量更新计数
dcnt	NUMBER 类型，增量删除计数
ddlCnt	NUMBER 类型，增量 DDL 计数

获取数据链路日志

该接口可以获取 FZS Agent 的增实时运行日志。

请求 URI

POST /logs

参数说明

```
{
  "id": "string"
}
```

参数说明如下：

参数名	说明
<code>id</code>	<code>STRING</code> 类型，FZS Agent ID。

使用样例

以下请求会获取 ID 为 `8AouMB` 的 FZS Agent 的实时运行日志。

```
curl -i -X POST \
  -H "Content-Type:application/json" \
  -d \
  '{
```

响应体格式

```
{
  "infoMsg": "20047 BLK:19 BLKOFF:0 MEM:0 CNT:22<br/>2023-11-07
13:55:27.353 INFO    - ANA REDO RETURN:LOG_NEED_AGAIN<br/>2023-11-07
13:55:27.353 INFO    - th:1 scn_max:b2a4b1e
scn_sub_max:1<br/>2023-11-07 13:55:27.353 INFO    - stream clean out
size:515 pay time : 2us<br/>2023-11-07 13:55:27.353 INFO    - merge
(pay time:38us)<br/>2023-11-07 13:55:27.353 INFO    - export 1 process
tgtid = 1, IRP = 1, DRP = 1, URP = 1, QMI = 0, DDL = 0, TRANS = 3,
SCN_TIME:2023-11-07 13:55:24)<br/>2023-11-07 13:55:27.353 INFO    -
SHOULD SAVE 0 NOT OUTPUT CMT TRANS TO DISK, NEED NOT
SAVE<br/>2023-11-07 13:55:27.353 INFO    - SAVED 0 CMT AND UNCMT TRANS
TO DISK BEFORE NEXT ANALYSIS<br/>2023-11-07 13:55:27.353 INFO    -
th_deal:0x1<br/>2023-11-07 13:55:27.354 INFO    - th:1 scn_max:b2a4b1e
scn_sub_max:1<br/>2023-11-07 13:55:27.354 INFO    -
th_deal:0x1<br/>2023-11-07 13:55:27.354 INFO    - HAS NO
CHANGE<br/>2023-11-07 13:55:27.853 INFO    - there is no incl data
kd_buf_list_size:551<br/>2023-11-07 13:55:27.853 INFO    -
kd_buf_list_size:551<br/>2023-11-07 13:55:27.854 INFO    - export 1 tgt
1 File Created:/home/fzs/web-server/fzsweb/run/fzs/fzs8AouMB/run/
fzs_src/1/1/1861.fzs (Size = 551)<br/>2023-11-07 13:55:27.854 INFO    -
write stream to file(pay time:165us)<br/>2023-11-07 13:55:27.854 INFO    -
send tgt 1 incl 0 file:/home/fzs/web-server/fzsweb/run/fzs/
fzs8AouMB/run/fzs_src/1/1/1861.fzs size:551 finished<br/>2023-11-07
13:55:28.355 INFO    - th:1 scn_max:b2a4b1e
scn_sub_max:1<br/>2023-11-07 13:55:28.355 INFO    -
th_deal:0x1<br/>2023-11-07 13:55:28.355 INFO    - HAS NO
CHANGE<br/>2023-11-07 13:55:29.356 INFO    - th:1 scn_max:b2a4b1e
scn_sub_max:1<br/>2023-11-07 13:55:29.356 INFO    -
th_deal:0x1<br/>2023-11-07 13:55:29.356 INFO    - HAS NO
CHANGE<br/>2023-11-07 13:55:30.357 INFO    - REDO CHANGE:[/u01/app/
oracle/oradata/LHR11G/redo01.log], RAC_ID:1, SEQ:120047,
MAX_SEQ:120047, DIFF_SEQ:0, CUR_BLK:19, NEXT_BLK:1253, BLK_OFF:0
LEFLAG:8<br/>2023-11-07 13:55:30.357 INFO    - TH:1 READ
SIZE:631808<br/>2023-11-07 13:55:30.358 INFO    - REC TH:1 SEQ:120047
BLK:1253 BLKOFF:0 MEM:0 CNT:2087<br/>2023-11-07 13:55:30.359 INFO    -
```

参数说明如下：

参数名	说明
infoMsg	STRING 类型，FZS Agent 的运行日志。
errMsg	STRING 类型，FZS Agent 的错误日志。

清除链路错误日志

该接口可以清除 FZS Agent 的错误日志。

请求 URI

POST /reset-error

参数说明

```
{  
  "id": "string"  
}
```

参数说明如下：

参数名	说明
<code>id</code>	<code>STRING</code> 类型，FZS Agent ID。

使用样例

以下请求会清除 ID 为 `8AouMB` 的 FZS Agent 的错误日志。

```
curl -i -X POST \  
  -H "Content-Type:application/json" \  
  -d \  
'{
```

响应体格式

```
{ "stateCode": 0, "stateMsg": "" }
```

参数说明如下：

参数名	说明
stateCode	NUMBER 类型，错误码。0 表示成功
stateMsg	STRING 类型，错误消息

下载数据链路日志

该接口可以打包下载 FZS Agent 的运行日志。

请求 URI

POST /log-tar

参数说明

```
{
  "id": "string"
}
```

参数说明如下：

参数名	说明
<code>id</code>	<code>STRING</code> 类型，FZS Agent ID。

使用样例

以下请求会获取 ID 为 `8AouMB` 的 FZS Agent 的实时运行日志。

```
curl -i -X POST \
  -H "Content-Type:application/json" \
  -d \
  '{
```

响应体格式

```
{
  "bytes": [123, 233, 221]
}
```

参数说明如下：

参数名	说明
bytes	<code>BUFFER</code> 类型，FZS Agent 的运行日志文件 <code>buffer</code> 。

获取 FZS 软件包

该接口获取 Web Server 上的 FZS Agent 软件包。

请求 URI

POST /agent-version

参数说明

```
{
  "id": "string"
}
```

参数说明如下：

参数名	说明
id	STRING 类型，FZS Agent ID。注意：传入空值则返回当前 Web Server 上的所有软件包。

使用样例

以下请求会获取 172.21.16.3:35081 Web Server 上的所有 FZS Agent 包。

```
curl -i -X POST \
  -H "Content-Type:application/json" \
```

响应体格式

```
{
  "agentVersions": [
    {
      "agentType": "oracle",
      "dbVersion": "10g",
      "softVersion": "main.e45bf1ad"
    },
    {
      "agentType": "oracle",
      "dbVersion": "11g",
      "softVersion": "main.odbc.pos.61e430b1"
    },
    {
      "agentType": "oracle",
      "dbVersion": "12c",
      "softVersion": "main.e45bf1ad"
    },
    {
      "agentType": "ora2ob",
      "dbVersion": "10g",
      "softVersion": "feature-zsuf30-obtarget.b0960426"
    },
    {
      "agentType": "ora2ob",
      "dbVersion": "11g",
      "softVersion": "feature-zsuf30-obtarget.b0960426"
    },
    {
      "agentType": "ora2ob",
      "dbVersion": "12c",
      "softVersion": "feature-zsuf30-obtarget.b0960426"
    },
    {
      "agentType": "odbc",
      "dbVersion": "10g",
```

参数说明如下：

参数名	说明
agentVersions	Web Server 上的单个或多个 Agent 包信息。

agentVersions 参数说明如下：

参数名	说明
agentType	STRING 类型，FZS Agent 包的类型。
dbVersion	STRING 类型，FZS Agent 包对应的数据库版本。
softVersion	STRING 类型，FZS Agent 包的版本。

更新 FZS 软件包

该接口用于更新 Web Server 上的 FZS Agent 软件包。

请求 URI

POST /upload-agent

参数说明

```
{
  "name": "string",
  "filename": "string",
  "file": "string"
}
```

参数说明如下：

参数名	说明
name	STRING 类型，FZS Agent ID。
filename	STRING 类型，软件包名称。
file	STRING 类型，base64 字符串，软件包的内容。

使用样例

以下请求将为 ID `rt0RekZe` 的 Agent 更新一个

`0131.oracle.afb6e5a2.fzs.linux.11g.web.tar.gz` 软件包，更新成功会返回相应信息。

```
curl -i -X POST \  
  -H "Content-Type:multipart/form-data" \  
  -F "rt0RekZe=@/tmp/0131.oracle.afb6e5a2.fzs.linux.11g.web.tar.gz" \  
  'http://172.21.16.4:8981/upload-agent'
```

响应体格式

```
{ "stateCode": 0, "stateMsg": "" }
```

参数说明如下：

参数名	说明
<code>stateCode</code>	<code>NUMBER</code> 类型，错误码。 <code>0</code> 表示成功
<code>stateMsg</code>	<code>STRING</code> 类型，错误消息

数据库配置

MySQL

以下是部署 MySQL 相关数据链路的一些注意事项。

MySQL 做源端

拓扑结构支持

1. 单机
2. 主从 / 双主
3. MHA 集群
4. 多主 NDB 集群
5. 托管（云服务商提供的 MySQL 服务）

详情参考：[MySQL 支持的拓扑结构](#)

配置 MySQL 数据库

1. 创建 fzs 用户

```
CREATE USER 'fzs'@'localhost' IDENTIFIED BY 'fzs';
```

2. 赋予 fzs 用户权限

```
GRANT SELECT, RELOAD, SHOW DATABASES, REPLICATION SLAVE, REPLICATION CLIENT ON *.*  
TO 'fzs' IDENTIFIED BY 'fzs';  
FLUSH PRIVILEGES;
```

3. 启用 binlog

- 查看 binlog 状态

```
mysql> show variables like '%log_bin%';
```

- 如果 binlog 是 OFF 状态，修改 my.cnf 文件

```
server-id                = 223344 # Querying variable is called server_id,
e.g. SELECT variable_value FROM information_schema.global_variables WHERE
variable_name='server_id';
log_bin                  = mysql-bin
binlog_format             = ROW
binlog_row_image          = FULL
binlog_expire_logs_seconds = 864000
```

- 重启 MySQL 服务



重启 MySQL 服务会导致数据库服务中断，需要谨慎操作。

```
systemctl restart mysql
```

4. 启用 GTIDs

- 查看 GTID 状态

```
mysql> show variables like '%gtid_mode%';
```

- 如果 GTID 是 OFF 状态，执行以下命令

```
SET GLOBAL ENFORCE_GTID_CONSISTENCY = 'WARN';
SET GLOBAL ENFORCE_GTID_CONSISTENCY = 'ON';
SET GLOBAL GTID_MODE = 'OFF_PERMISSIVE';
SET GLOBAL GTID_MODE = 'ON_PERMISSIVE';
SET GLOBAL GTID_MODE = 'ON';
```

5. 配置会话超时

- 查看会话超时时间

```
mysql> show variables like '%wait_timeout%';
```

- 如果会话超时时间不是默认值，执行以下命令：

```
mysql> set binlog_rows_query_log_events=ON;
```

6. 开启 **query log events**

- 查看 query log events 状态

```
mysql> show variables like '%binlog_rows_query_log_events%';
```

- 如果 query log events 是 **OFF** 状态，执行以下命令：

```
mysql> set binlog_rows_query_log_events=ON;
```

7. 校验 **binlog row value options**

- 查看 binlog row value options 状态

```
mysql> show global variables where variable_name = 'binlog_row_value_options';
```

- 如果 binlog row value options 是 **PARTIAL_JSON** 状态，执行以下命令：

```
mysql> set @@global.binlog_row_value_options='';
```

详情参考：[MySQL 配置](#)

部署 **FZS Source Connector**

1. 准备以下两个镜像：

- bitnami/kafka**
- 9bridges/fzs-source-connector**

2. 创建一个 **docker-compose-source.yml** 文件，内容如下：

```
version: '3'
name: fzs-source
services:
```

3. 启动 FZS Source Connector:

```
docker compose -f docker-compose-source.yml up -d
```

MySQL 做目标端

1. 创建 `fzs` 用户

```
CREATE USER 'fzs'@'localhost' IDENTIFIED BY 'fzs';
```

2. 赋予 `fzs` 用户权限

```
GRANT ALL PRIVILEGES ON *.* TO 'fzs'@'localhost' WITH GRANT OPTION;
```

OceanBase

创建 OceanBase 数据库用户

1. 在 sys 租户中，为 fzs 创建一个带密码的用户，用于增量抽取。

```
obclient> CREATE USER ${sys_username} IDENTIFIED BY  
'${sys_password}';  
obclient> GRANT ALL PRIVILEGES ON *.* TO ${sys_username} WITH  
GRANT OPTION;
```

2. OceanBase 增量抓取需要配置 `config-url`，可以使用以下命令获取：

```
obclient> show parameters like 'obconfig_url';
```

3. 登录业务租户，为你想要监控的租户创建一个用户，这个用户用来读取快照数据。

OceanBase 数据库 MySQL 租户

```
obclient> CREATE USER __oceanbase_inner_drc_user IDENTIFIED BY  
'<password>';  
obclient> GRANT ALL ON *.* TO __oceanbase_inner_drc_user;
```

OceanBase 数据库 Oracle 租户

```
obclient> CREATE USER '__OCEANBASE_INNER_DRC_USER' IDENTIFIED BY  
<your_password>;  
obclient> GRANT DBA TO '__OCEANBASE_INNER_DRC_USER';
```

恒生电子 LightDB

修改数据库参数，使其支持逻辑复制

1. 修改 `lightdb.conf` 文件，设置以下参数：

```
wal_level = logical
archive_mode = on
max_wal_senders = 10
max_replication_slots = 10
```

2. 重启数据库

创建数据库登录用户

1. 创建数据库用户

```
CREATE USER fzs1 WITH PASSWORD 'fzs1';
```

2. 赋予超级用户权限，因为需要对所有表进行读取和创建 `publication` 操作

```
ALTER USER fzs1 WITH SUPERUSER;
```

数据对象操作

为所有同步表开启逻辑复制

1. 修改下面的 `schema` 为需要同步的 `schema`，执行查询结果中的 `sql` 语句

```
SELECT
  'alter table ' || schemaname || '.' || tablename || ' replica identity full;'
FROM
  pg_tables
WHERE
  schemaname IN ('fzs', 'test')
  AND tablename NOT IN (
    SELECT
      relname
    FROM
      pg_class c
      JOIN pg_inherits i ON
        c.oid = i.inhrelid
  )
```

创建用于同步 `ddl` 操作的相关表和触发器

```
CREATE SCHEMA IF NOT EXISTS fzs;
CREATE TABLE fzs.ddl_command
(
  ddl_text TEXT COLLATE pg_catalog."default",
  id BIGSERIAL PRIMARY KEY,
  EVENT TEXT COLLATE pg_catalog."default",
  tag TEXT COLLATE pg_catalog."default",
  username CHARACTER VARYING COLLATE pg_catalog."default",
```


版本发布

❗ 信息

FZS 部署脚本下载: gen.sh

⚠ 警告

当链路源端为 Oracle 时，部署 Web Server 需要使用其二进制版本，软件包下载：
[webserver-0.4.4.gz](#)

此页记录自 `v0.3.0` 以来 FZS 的版本发布信息，包括新增功能、功能优化、功能修复等。

0.5.5

日期: 2024-11-15

新增功能

- 新增 `Kerberos` 认证功能：当备端节点为 `Apache Kafka` 时，可选择 `Kerberos` 认证方式，保证数据链路的安全性。
- 数据节点、数据链路、数据任务表单页新增误退出提示：当数据节点、数据链路、数据任务表单页存在未保存的数据时，离开页面时会提示是否确认离开。

功能优化

- 用户登录 `Session` 支持 7 天有效期：用户登录 `Session` 支持 7 天有效期，避免了频繁登录的操作。
- 完善页面路由对于用户登录状态的判断：完善了页面路由对于用户登录状态的判断，避

免了未登录用户访问页面的情况。

- **OceanBase** 数据节点 **Icon** 展示优化：在数据链路中展示 **OceanBase** 数据库时，使用崭新的数据库图标。

0.5.4

日期：2024-11-01

新增功能

- 新增 **Oracle** 远程抓取功能：当链路源端为 **Oracle** 时，可通过远程抓取的方式获取增量数据，避免了在源端部署 Agent 的繁琐操作。

功能优化

- 数据链路详情页同步效果升级：优化了数据链路详情页的同步的动画效果，使得同步的过程更加直观。

缺陷修复

- **MEM** 占用取值修复：修复了仪表盘中 **MEM** 占用取值不准确的问题。

0.5.3

日期：2024-10-18

新增功能

- 新增邮件告警渠道：新增了邮件告警渠道，当链路出现异常时，系统会自动发送告警邮

件通知。

功能优化

- 日志下载详细命名：优化了日志下载的命名，更加直观地展示日志的内容，方便用户快速定位问题。
- 邮件样式优化：优化了邮件告警的样式，使得邮件告警更加美观。

0.5.2

日期：2024-09-30

新增功能

- 延时告警：新增了延时告警功能，当链路延时超过预设值时，系统会自动发送告警通知。
- 空闲告警：新增了空闲告警功能，当链路增量空闲超过预设值时，系统会自动发送告警通知。

功能优化

- **Webhook** 配置优化：移除 `monaco-editor`，改为 `textarea`，优化了 webhook 配置的体验
- **Webhook** 告警去重：优化了 webhook 告警的去重逻辑，避免了告警频繁发送
- 数据节点图片新增：新增了 `SQL Server`、`DB2` 与 `Informix` 数据库的 `svg` 图标
- 滚动效果全新升级：得益于 `Lenis`，FZS 的滚动效果得到了全新的升级，为用户提供更加流畅的滚动体验

缺陷修复

- **Webhook** 告警双引号转义：修复了 webhook 告警时双引号转义的问题
- **SSH** 部署 **Web Server** 环境变量缺失：修复了通过 SSH 一键部署 Web Server 时，无法获取当前用户环境变量的问题

0.5.1

日期：2024-08-28

功能优化

- 可以通过配置环境变量 `WEB_DATA_PORT_START` 与 `WEB_DATA_PORT_END`，在启动的时默认添加 web server
- 优化了首页展示 CPU 和内存使用率的展示，更加直观地查看系统的运行状态。
- 优化了系统日志请求过多的问题。
- 优化了部分UI的展示及交互。

功能修复

- 修复了添加其他配置项时，无法添加的问题。
- 修复了链路切换失败的问题。
- 修复 asm 配置 placeholder 的问题。

0.5.0

日期：2024-08-12

新增功能

- 新增视图和其他对象: 数据比对新增了视图和其他对象的比对数据, 更加全面且详细的展示数据比对的结果。
- 新增表空间使用率和用户数据量: 在节点详情页新增了表空间使用率和用户数据量的展示, 可直观地查看数据库相关信息。
- 新增一键导入坏表: 在链路修复页新增一键导入同步出错的坏表, 方便用户快速修复坏表。
- 数据任务新增 **Agent** 信息: 在数据任务详情页新增了 **Agent** 信息, 可以一键查看所有相关链路的 **Agent** 信息。
- 新增增量统计平均值显示: 在链路详情页的增量统计中新增了平均值的显示, 可直观地查看增量数据的平均值。
- 数据任务的数据比对: 在数据任务详情页新增了数据比对功能, 一键对所有链路进行数据比对, 更加方便的查看数据链路比对结果。
- 数据比对修复: 在数据比对完成后新增修复功能, 弹出框内会显示比对不一致的表, 可以更加简易的选择要修复的表。
- 其他配置项可选: 在数据链路修复页和数据任务的修复页增加了其他配置的选择, 更加方便的填写其他配置参数。
- 其他配置项入表: 将其他配置脱离文件控制, 方便运维人员进行配置。
- 新增链路同步模式: 新增了链路同步模式, 可选择实时模式、增量模式、迁移模式, 更加灵活的选择链路同步模式。

功能优化

- 数据链路在配置节点时展示对应 icon
- 其他配置在搜索时支持搜索描述和对应键名
- 优化了数据校验 UI, 移除点击名称跳转节点详情页
- 在查看链路配置时将不再显示 agent 列表

- 优化了各模块数据为空时的提示信息
- 优化了链路修复页和任务修复页的展示效果
- 当删除相关链路，节点，校验，任务时会删除对应的操作日志。
- 优化了部分 UI 的展示及交互

功能修复

- 修复了删除数据任务时无法删除的问题。
- 修复了在数据比对无数据时跳转不成功的问题。
- 修复了数据节点验证时多提示框，切换页面提示框不消失的问题。
- 修复了部分已知问题。

0.4.4

日期：2024-06-07

新增功能

- 新增内容校验：新增了内容校验功能，可根据配置的校验规则对数据进行校验，校验结果将会以文件的形式保存在系统中。
- 新增定时启动链路：新增了定时启动链路功能，可根据配置的时间定时启动链路。
- 新增校验快捷导航：在校验列表新增了校验快捷导航，可快速操作校验。
- 新增节点入口：在校验详情页新增了节点入口，可快速查看节点详情。

功能优化

- 优化了数据链路配置中配置项的默认值。
- 优化了数据链路日志下载按钮的展示和提示信息。日志清除及下载按钮增加提示信息
- 优化了数据校验中校验任务的展示及状态展示。

- 优化了部分UI的展示及交互。

功能修复

- 修复了停止任务时间过长 loading 效果不准确的 bug。
- 修复了配置链路上传文件自动解析配置的 bug。
- 修复系统日志显示校验名称的 bug。
- 修复校验模块下载文件名为中文无法下载的 bug。
- 修复了部分已知问题。

0.4.3

日期：2024-05-11

新增功能

- 侧边栏可折叠：新增了侧边栏可折叠的功能，可以更加专注
- 自定义 **sql** 功能升级：
 - 支持上传sql文件自动解析sql语句
 - 支持下载校验结果文件
 - 支持定时自动执行校验任务
- 链路详情页增加快速跳转相应节点：在链路详情页增加了快速跳转相应节点的功能，方便用户查看对应节点详情。

功能优化

- 优化了数据校验中校验任务的展示及状态展示。
- 禁用数据任务增加二次确认框。
- 首页显示更多主机相关信息。

- 删除校验任务增加loading效果

功能修复

- 修复了数据校验编辑框无法编辑的bug。
- 修复了数据校验任务无法删除的bug。
- 修复了部分已知问题。

0.4.2

日期：2024-04-19

新增功能

- 新增数据比对过滤：新增了数据比对过滤功能，可根据有无主键来进行数据比对。
- 新增数据任务可禁用某链路：新增了数据任务可禁用某链路的功能，可禁用不需要的链路。

功能优化

- 数据比对功能优化升级，解决数据量过大无法比对，并附带全新新UI，更加直观的展示数据比对的结果。
- 添加到任务的链路，将无法点击，并限制只能从任务进入链路详情。
- 优化了部分其他 UI 细节，使得界面更加美观。

功能修复

- 修复了数据比对屏幕过小无法滚动的问题。
- 修复了部分已知问题。

0.4.1

日期：2024-04-02

新增功能

- 新增校验任务的删除：新增了校验任务的删除功能，可删除不需要的校验任务。

功能优化

- 优化上传 Agent 包时会校验包的合法性。
- 链路切换不再限制数据库类型。
- 优化了部分其他 UI 细节，使得界面更加美观。

功能修复

- 重置按钮使用时机的问题。
- 修复数据任务中的数据链路可修复的问题。
- 链路中其他配置问题。
- 修复了部分已知问题。

0.4.0

日期：2024-03-15

新增功能

- 新增灾备切换：在配置链路时可指定灾备链路，当主链路出现故障时，将自动切换到配

置的灾备链路，保证数据的连续性。

- **新增数据任务**：新增了数据任务功能，在配置相应数据任务后，可实现链路的一对多同步或者多个链路的一对一，方便集中管理数据链路。
- **新增 SQL 自定义查询**：在数据校验位置新增了自定义查询功能，对于 SQL 数据源，可根据需求自定义 SQL 查询相应数据。
- **新增 SSO 登录**：新增了 SSO 登录功能，在用户管理页有提供相对应接口，可通过调用相应接口实现 SSO 登录 FZS。
- **新增全部系统日志**：新增了全部系统日志功能，在系统首页可查看系统的全部日志。

功能优化

- 优化了首页展示卡片间距。
- 优化了链路的重置按钮可使用时机。
- 优化了部分其他 UI 细节，使得界面更加美观

功能修复

- 修复数据比对在多次比对时，数据一直不变的问题。
- 修复普通用户可删除 ROOT 用户的问题。
- 修复了部分已知问题。

0.3.5

❗ 信息

此版本需要 FZS Daemon 0.1.1 及以上版本配合使用。

日期：2024-02-04

新增功能

- 链路和节点列表可折叠：在链路和节点列表新增了折叠功能，可根据需求展开或折叠列表，更加便捷地管理链路和节点。
- 查看链路配置：在链路链路在 running 状态可以查看当前链路配置，方便用户查看当前链路的配置信息。

功能优化

- 优化设置页面的布局，使得设置页面更加美观。
- 优化了部分其他 UI 细节，使得界面更加美观

功能修复

- 修复数据比对在报错情况下，数据遗留的问题。
- 修复了部分已知问题。

0.3.4

日期：2024-01-19

新增功能

- 新增数据比对筛选功能：在数据比对页面新增了筛选功能，可根据数据比对的结果进行筛选，更加直观地展示数据比对的结果。
- 新增销毁、重置的二次确认：在销毁、重置链路时，新增了二次确认，避免误操作。
- 新增数据库图标：在相关数据库展示的地方增加了数据库的图标，更加直观地展示数据库的类型。
- 新增增量累计柱状图：在链路详情页的增量累计图表中新增了柱状图显示，可以根据时

间段切分增量数据，更加直观地展示增量数据。

- 新增 **Web Server** 一键部署：在「系统设置」中新增了 Web Server 一键部署，可一键部署 Web Server，避免了手动部署 Web Server 的繁琐操作。

功能优化

- 优化数据比对数据展示：优化了数据比对的展示，更加直观地展示数据比对的结果。
- 优化数据链路更换节点：数据链路创建后禁止更换节点的操作，避免agent含义不明确及减少资源消耗。
- 优化了部分 UI 细节，使得界面更加美观

功能修复

- 修复数据比对数据不一致时，数据不标红的问题。
- 修复比对数据会自动消失的问题。
- 修复链路在创建后
- 修复了部分已知问题。

0.3.3

❗ 信息

此版本需要 FZS Web Server 0.3.3 及以上版本配合使用。

日期：2023-12-29

新增功能

- 首页新增 **CPU** 和内存使用率展示：在首页新增了 CPU 和内存使用率的展示，可直观地查看系统的运行状态。

- 链路详情页新增全量日志和增量日志：更加颗粒度的展示链路相关日志，更好的帮助用户定位问题。

功能优化

- 数据比对 **UI** 及 **UX** 的重构：更好的展示数据比对的结果，更加清晰的展示数据比对的结果。
- 链路详情页合并增量相关数据：优化了增量累计图表的显示，将增量相关的数据合并到同一个图表内,更加直观地展示增量数据。
- 首页增量和延时图表的重构：优化了首页对各链路增量和延时的展示，
- 链路详情页增量和延时图表的重构：优化了链路详情页对各链路增量和延时的展示，
- 优化了部分 **UI** 细节，使得界面更加美观

功能修复

- 修复了部分已知问题。

0.3.2

日期：2023-12-15

新增功能

- 数据链路新增源备端区分：在创建数据链路时，一些节点只能作为，一些节点只能作为备端，此功能使得 FZS 在创建数据链路时，更加清晰的区分源备端。

功能优化

- 配置抽取，装载对象时自动去重：优化了对于配置的抽取，装载对象时自动去重，避免了重复配置的出现。

- 链路操作错误提示：优化了链路操作时的错误提示，更加清晰的提示用户操作失败的原因。
- 优化了部分 UI 细节，使得界面更加美观。

功能修复

- 修复了链路报错时，无法在链路列表体现出来的问题。
- 修复了日志自动滚动影响页面的 bug。
- 修复了部分已知问题。

0.3.1

日期：2023-12-04

新增功能

- 列表新增悬浮菜单：在「数据链路」和「数据节点」列表项悬浮会出现菜单按钮，点击按钮弹出菜单，更加便捷的管理链路和节点。
- 链路详情页新增操作日志：在链路详情页增加对于此链路进行过的操作，更加清晰的管理链路。

功能优化

- **Web Server** 有效期自动校验：优化了对于 web server 有效期的校验方式，后台会在每天自动更新有效期，如果过期将会在首页的系统日志内提示。
- **Agent** 信息展示：优化了在链路详情页展示 Agent 信息的方式，更加清晰的展示 Agent 信息。
- 链路错误状态的分离：优化了链路错误状态的展示方式，将错误状态分离出来，更加清晰的展示链路状态。

- 节点配置优化：节点配置时，Web Server 配置列表只会显示可用的。

功能修复

- 修复了部分已知问题

0.3.0

日期：2023-11-24

新增功能

- 数据节点可共享 **Web Server** 配置：数据节点与 Web Server 不再是一对一的关系，一个 Web Server 可以绑定多个数据节点。此功能使得 FZS 在处理多条同步链路时，减少了 Web Server 的部署数量，不仅降低了系统资源的消耗，也减少了系统管理的复杂度。
- 数据链路添加更换 **FZS** 软件包的入口：当 FZS 软件包需要升级时，可事先「停止」同步链路，通过数据链路的配置界面上传并更新的软件包后，点击「启动」链路来恢复同步。

功能优化

- 数据链路详情页新增 **Flow Chart** 状态显示：在数据链路详情页新增了 Flow Chart 状态显示，可直观地查看数据链路的运行状态。
- 数据节点详情页 **UI** 布局优化：数据节点详情页的 UI 布局进行了优化，使得界面更加美观，组件之间的布局更加平衡。
- 「系统设置」入口放到左下角用户信息的右侧。
- 「主题切换」入口放到左上方的「FZS」图标的最右侧。
- UI 配色使用 `css variables`，可按客户需求灵活配置。

功能修复

- 修复了部分已知问题