Databend

Databend

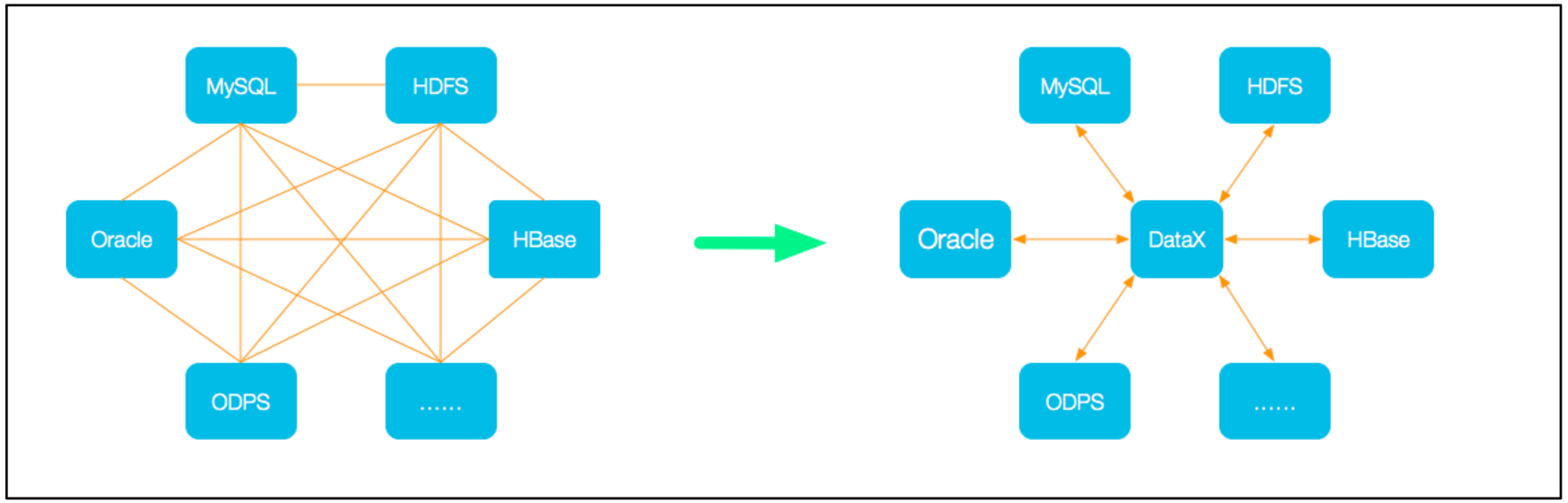
Data Integration

韩山杰

Databend

Datax

Datax 最适合的场景是 T+1 的离线数据同步，比如做全增量的数据归档

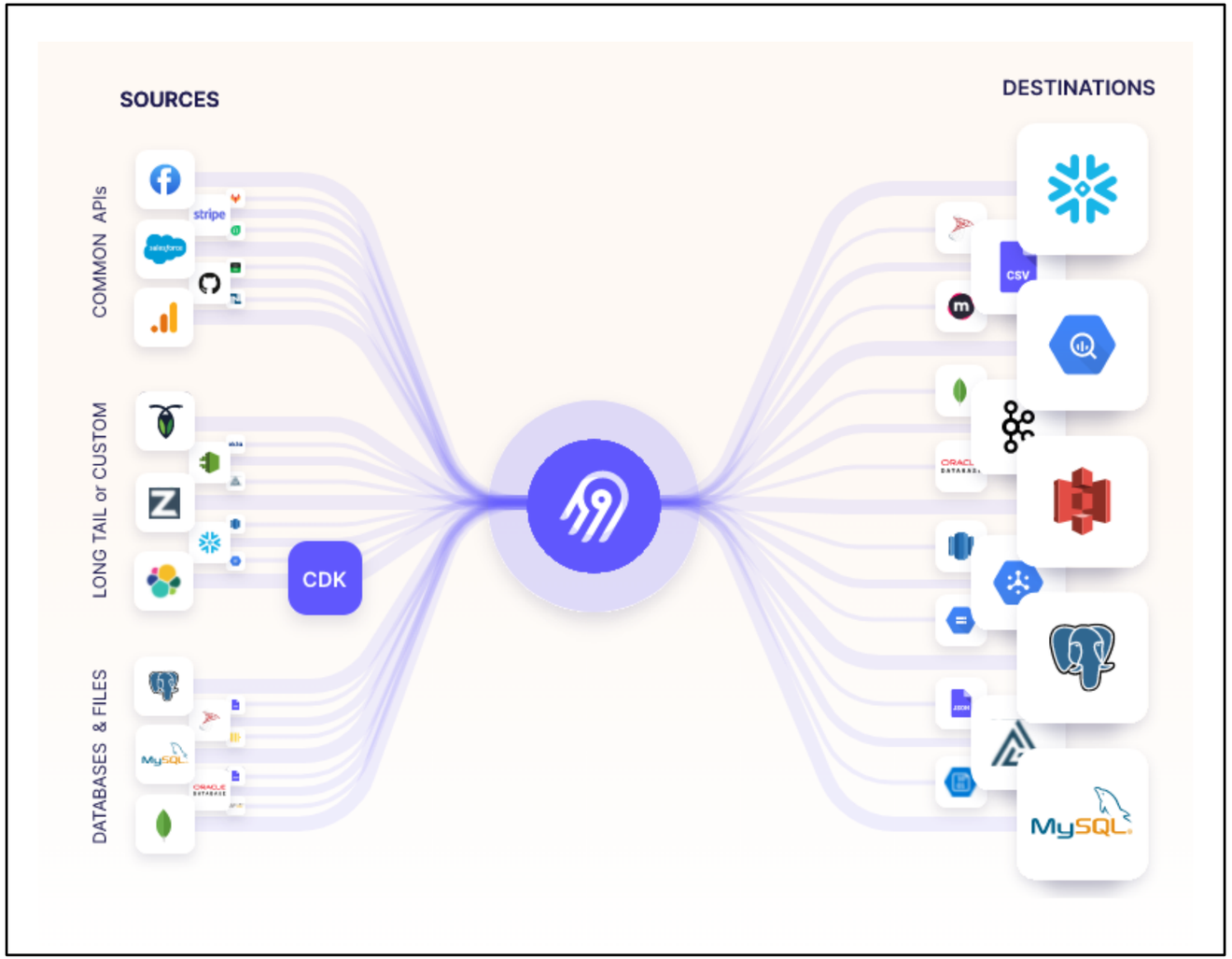


Databend 提供了 Databend Writer 的 Datax Plugin ，

可以支持从任意具有 Datax Reader 插件的数据库同步数据到 Databend。

支持 Inset 和 Upsert 两种 mode：https://github.com/alibaba/DataX/tree/master/databendwriter

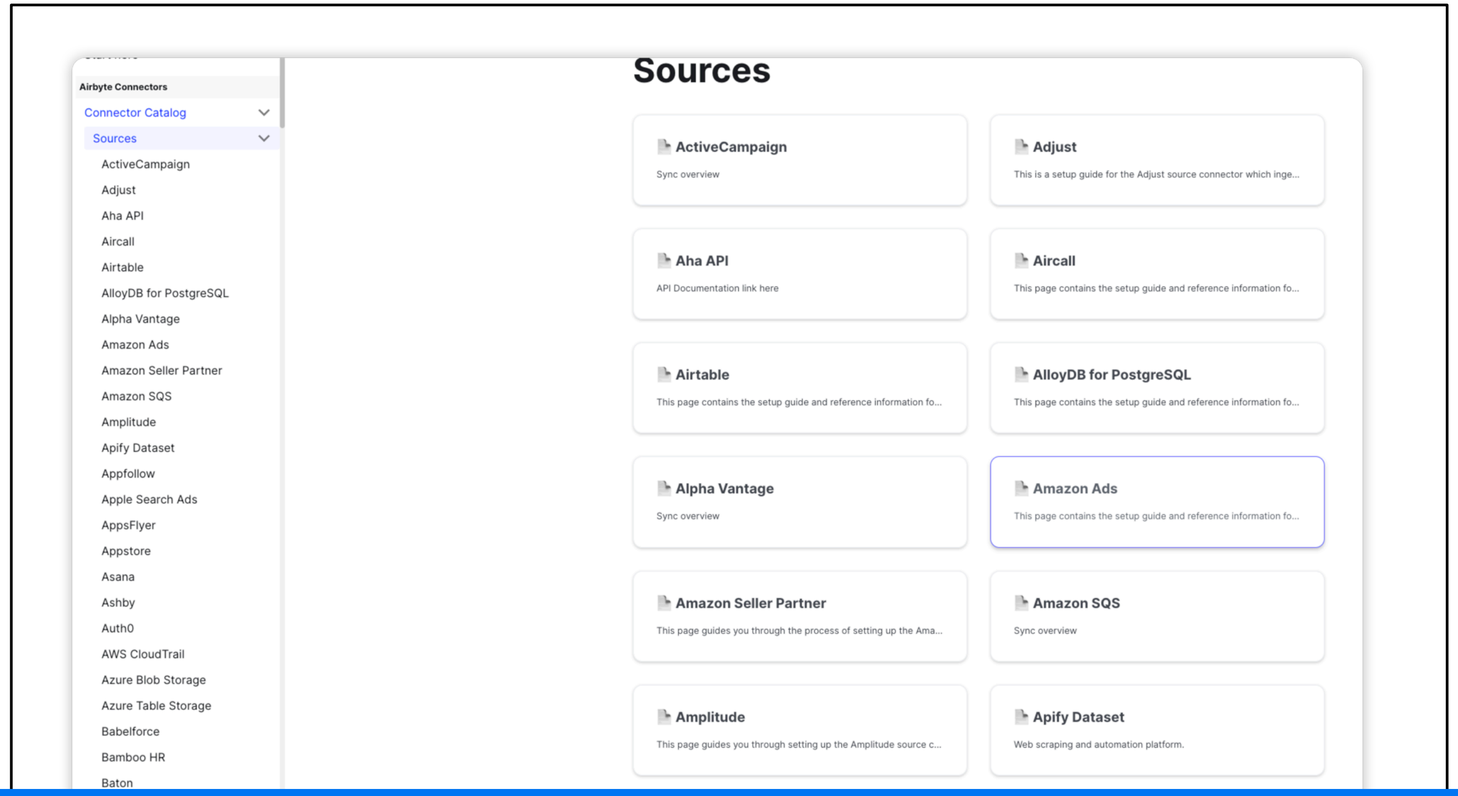
Airbyte



- https://docs.airbyte.com/integrations/destinations/databend/

Airbyte 支持上百种 data source

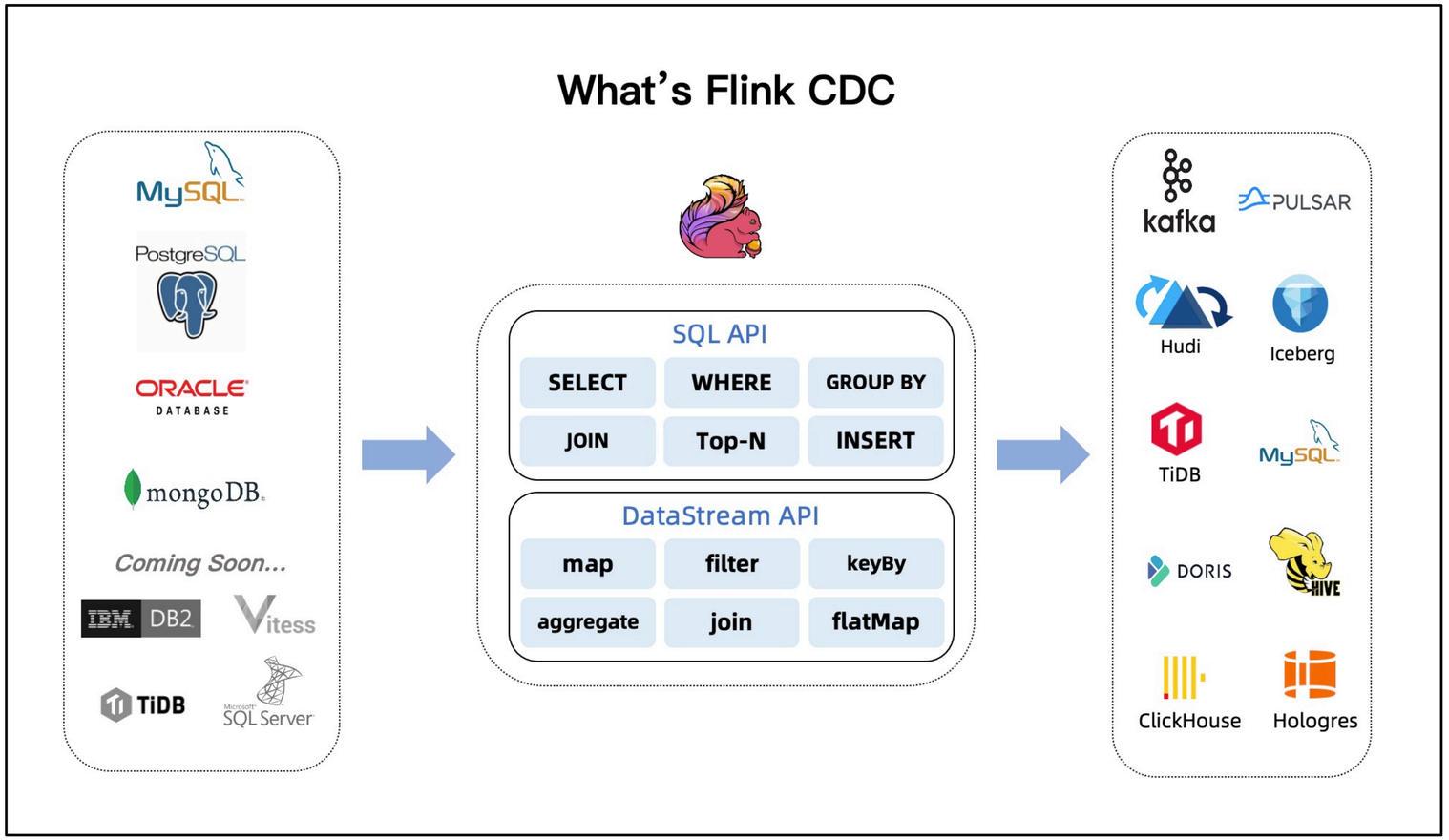
- 适合数据源多，同步任务多且需要统一管理的场景



Flink CDC

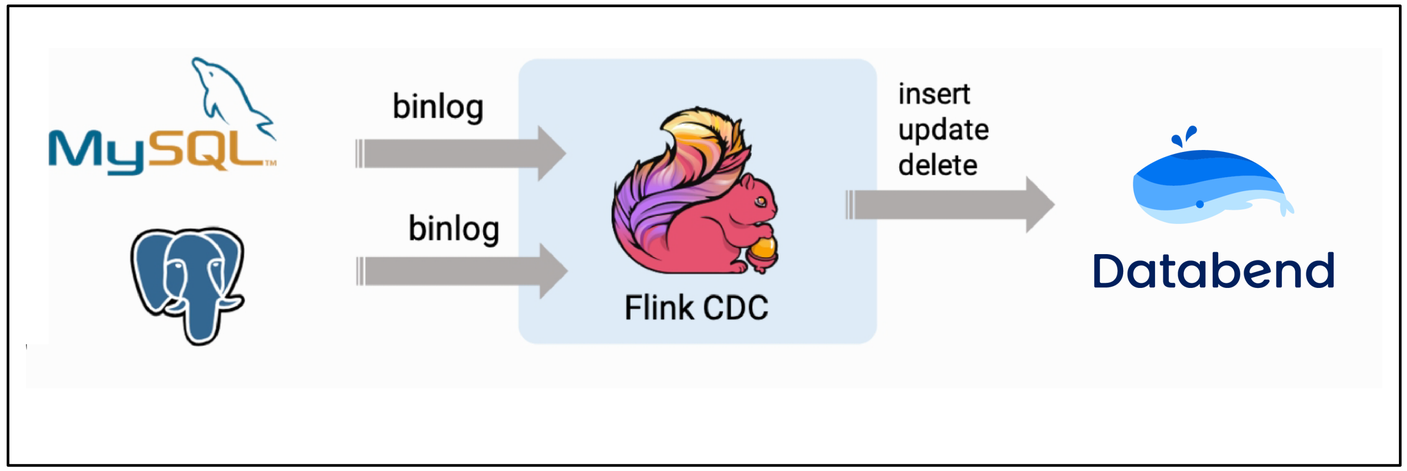
Flink 作为实时数据集成框架，具有无锁读取、并行读取、表模式自动同步、分布式架构等优势，能够实现 exactly once 语义。

CDC (Change Data Capture) 是一种用于捕捉数据库变更数据的技术，Flink 从 1.11 版本开始原生支持 CDC 数据（changelog）的处理，目前已经是非常成熟的变更数据处理方案。



Flink CDC

- Databend 也提供了 flink-databend-connector，可以与 MySQL，PG 等 RDBMS 构建实时数据同步。
- 启动 Flink Client 后，可使用以上 Flink SQL 创建 CDC 的 Flink Job

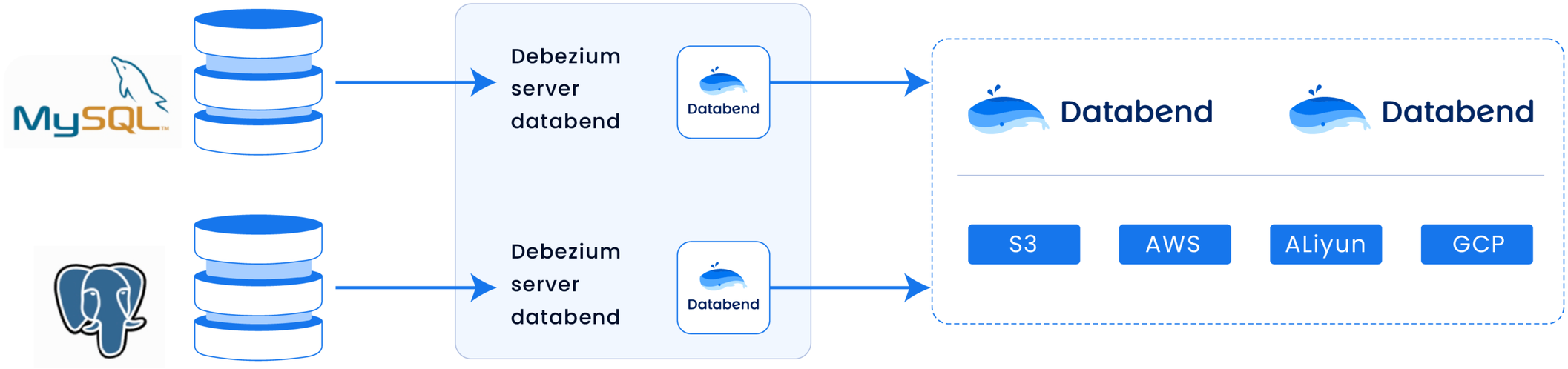



```sql
-- Flink SQL
create table d_products (id INT,name String,description String, PRIMARY KEY (`id`) NOT EI
with ('connector' = 'databend',
'url'='databend://localhost:8000',
'username'='databend',
'password'='databend',
'database-name'='default',
'table-name'='bend_products',
'sink.batch-size' = '5',
'sink.flush-interval' = '1000',
'sink.max-retries' = '3');
```

https://github.com/databendcloud/flink-connector-databend

Databend

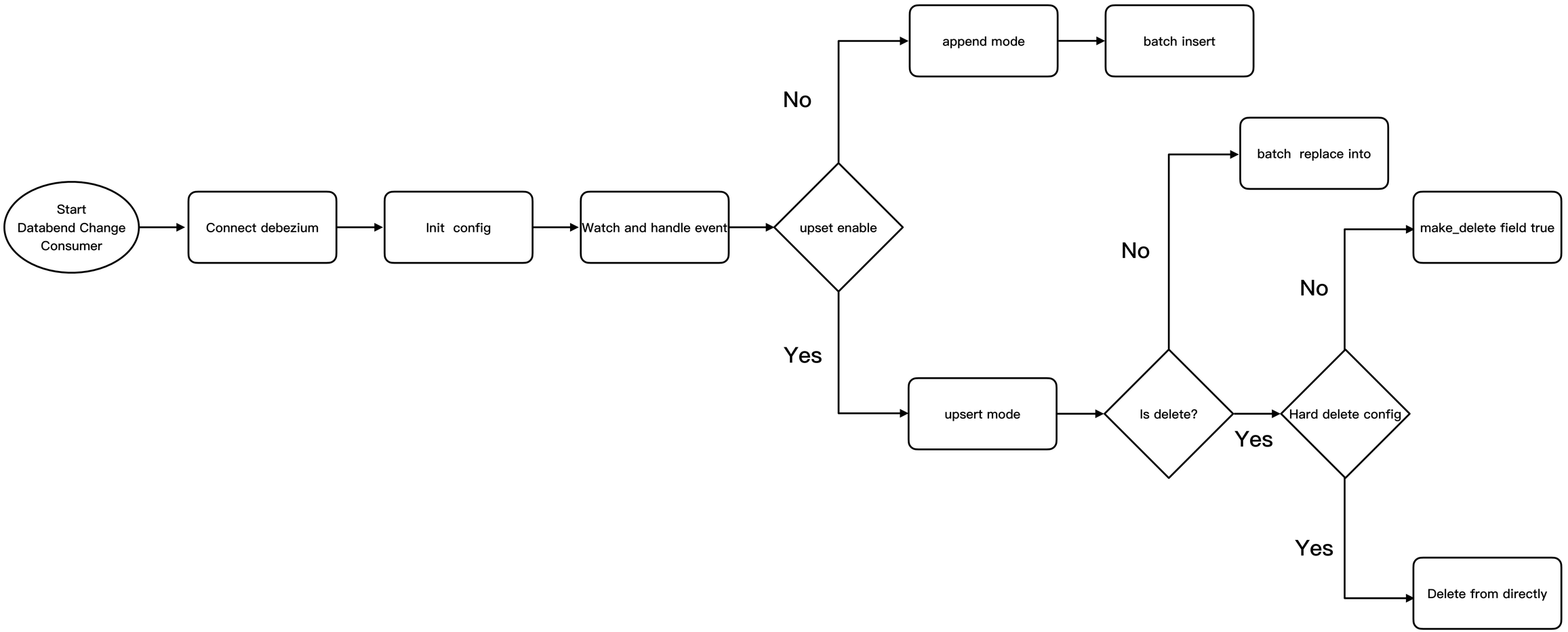
Debezium Server Databend



Debezium Server Databend 是 Databend 基于 Debezium Engine 自研的轻量级 CDC 项目，提供了一种简单的方式来监控和捕获关系型数据库的变化，并将这些变化转换为可消费的事件。无须依赖大型的 Data Infra 如 Flink, Kafka等，只需一个启动脚本即可开启实时数据同步。

https://github.com/databendcloud/debezium-server-databend

Debezium Databend 实现原理



步骤：

1. 下载 debezium databend https://github.com/databendcloud/debezium-server-databend/releases
2. 配置文件 mkdiraconf & touch conf/application.properties
3. 执行启动脚本bash run.sh

application.properties

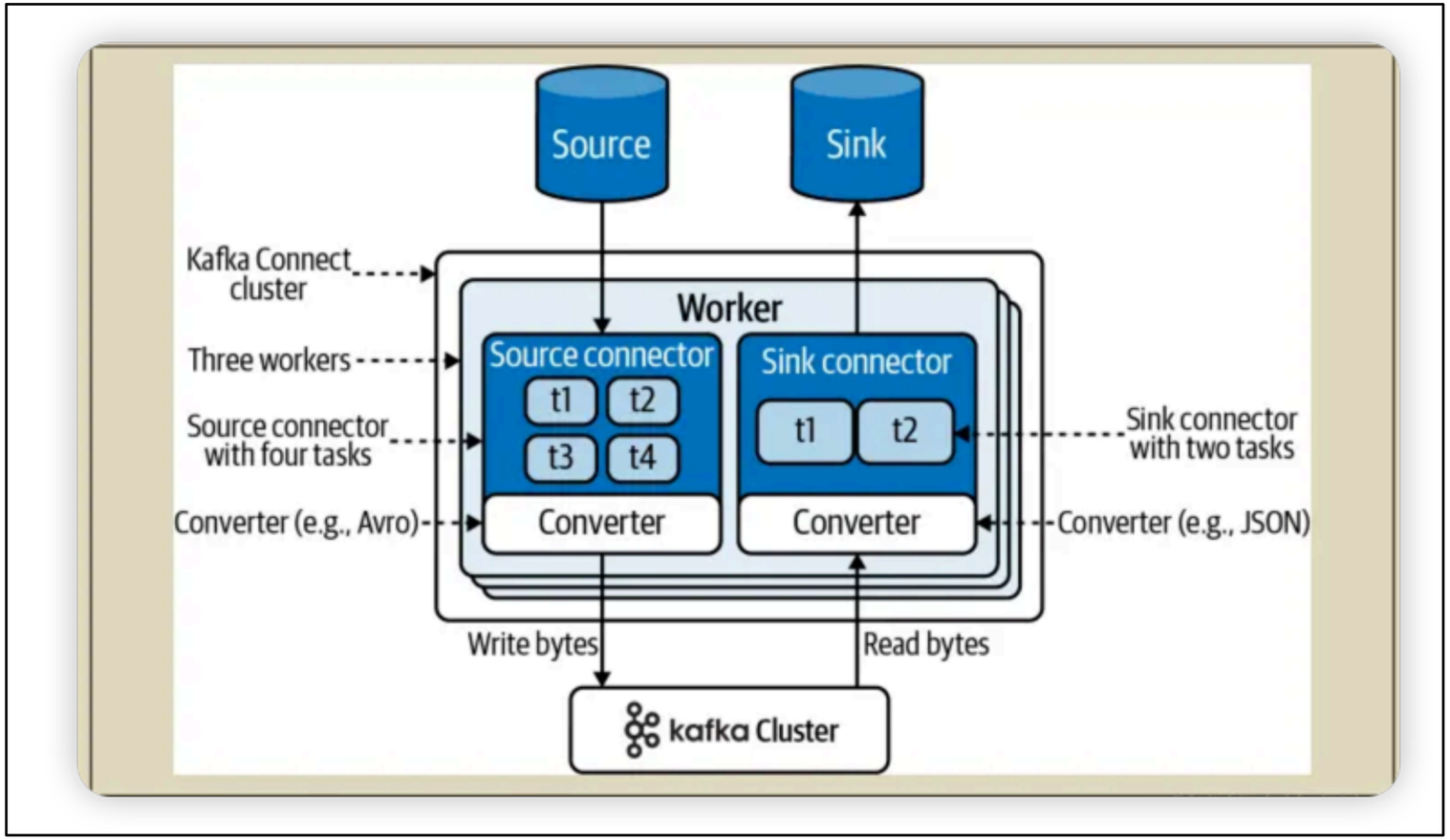
```
debezium.sink.type=databend
debezium.sink.databend.upsert=true
debezium.sink.databend.upsert-keep-deletes=false
debezium.sink.databend.database.databaseName=debezium
debezium.sink.databend.database.url=jdbc:databend://tnf34b0rm--medium-mms5.gw.aliyun-cn-beijing.default.databend.cn:443
debezium.sink.databend.database.username=cloudapp
debezium.sink.databend.database.password=password
debezium.sink.databend.database.primaryKey=id
debezium.sink.databend.database.tableName=products
debezium.sink.databend.database.param.ssl=true
# additional databend parameters

# enable event schemas
debezium.format.value.schemas.enable=true
debezium.format.key.schemas.enable=true
debezium.format.value=json
debezium.format.key=json

# mysql source
debezium.source.connector.class=io.debezium.connector.mysql.MySqlConnector
debezium.source.offset.storage.file.filename=data/offsets.dat
debezium.source.offset.flush.interval.ms=60000
```

Terminal Local Local (2)


```
rocessName":"io.debezium.server.Main","processId":34268}
{"timestamp":"2023-08-29T22:38:54.848+08:00","sequence":256,"loggerClassName":"org.slf4j.impl.Slf4jLogger","loggerName":"io.debezium.connector.mysql.MySqlStreamingChangeEventSource","level":"INFO","message":"Connected to MySQL binlog at pc-bp1cx14ktrq5hn3sn-public.mysql.polardb.rds.aliyuncs.c
sourceInfoSchema=Schema{io.debezium.connector.mysql.Source:STRUCT}, sourceInfo=SourceInfo [currentGtid=null, currentBinlogFilename=mysql-bin.000004, currentBinlogPosition=37696, currentRowNumber=0, serverId=0, sourceTime=2023-08-29T14:38:54Z, threadId=-1, currentQuery=null, tableIds=[mydb.pro
eted=true, transactionContext=TransactionContext [currentTransactionId=null, perTableEventCount={}, totalEventCount=0], restartGtidSet=81fd75d9-1ee9-11ee-bb4b-0c42a16494dc:1-119, currentGtidSet=81fd75d9-1ee9-11ee-bb4b-0c42a16494dc:1-119, restartBinlogFilename=mysql-bin.000004, restartBinlogPo
tEventsToSkip=0, currentEventLengthInBytes=0, inTransaction=false, transactionId=null, incrementalSnapshotContext =IncrementalSnapshotContext [windowOpened=false, chunkEndPosition=null, dataCollectionsToSnapshot=[], lastEventKeySent=null, maximumKey=null]]","threadName":"blc-pc-bp1cx14ktrq5hn
:3306","threadId":32,"mdc":{"dbz.taskId":"0","dbz.connectorName":"from_mysql","dbz.connectorType":"MySQL","dbz.connectorContext":"binlog"},"ndc":"","hostName":"hanshanjiedembp","processName":"io.debezium.server.Main","processId":34268}
{"timestamp":"2023-08-29T22:38:54.849+08:00","sequence":257,"loggerClassName":"org.slf4j.impl.Slf4jLogger","loggerName":"io.debezium.connector.mysql.MySqlStreamingChangeEventSource","level":"INFO","message":"Waiting for keepalive thread to start","threadName":"debezium-mysqlconnector-from_mys
adId":31,"mdc":{"dbz.taskId":"0","dbz.connectorName":"from_mysql","dbz.connectorType":"MySQL","dbz.connectorContext":"streaming"},"ndc":"","hostName":"hanshanjiedembp","processName":"io.debezium.server.Main","processId":34268}
{"timestamp":"2023-08-29T22:38:54.85+08:00","sequence":258,"loggerClassName":"org.slf4j.impl.Slf4jLogger","loggerName":"io.debezium.util.Threads","level":"INFO","message":"Creating thread debezium-mysqlconnector-from_mysql-binlog-client","threadName":"blc-pc-bp1cx14ktrq5hn3sn-public.mysql.pol
,"mdc":{"dbz.taskId":"0","dbz.connectorName":"from_mysql","dbz.connectorType":"MySQL","dbz.connectorContext":"binlog"},"ndc":"","hostName":"hanshanjiedembp","processName":"io.debezium.server.Main","processId":34268}
{"timestamp":"2023-08-29T22:38:54.85+08:00","sequence":259,"loggerClassName":"org.slf4j.impl.Slf4jLogger","loggerName":"io.debezium.connector.mysql.MySqlStreamingChangeEventSource","level":"INFO","message":"Keepalive thread is running","threadName":"debezium-mysqlconnector-from_mysql-change-e
dc":{"dbz.taskId":"0","dbz.connectorName":"from_mysql","dbz.connectorType":"MySQL","dbz.connectorContext":"streaming"},"ndc":"","hostName":"hanshanjiedembp","processName":"io.debezium.server.Main","processId":34268}
{"timestamp":"2023-08-29T22:38:55.119+08:00","sequence":260,"loggerClassName":"org.slf4j.impl.Slf4jLogger","loggerName":"io.debezium.server.databend.tablewriter.RelationalTable","level":"WARN","message":"Loaded Databend table debezium.debezium.products \nColumns:{name=DatabendRawType{type=nam
dRawType{type=description, isNullable=false}, id=DatabendRawType{type=id, isNullable=false}} \nPK:{id=1}","threadName":"pool-7-thread-1","threadId":19,"mdc":{},"ndc":"","hostName":"hanshanjiedembp","processName":"io.debezium.server.Main","processId":34268}
{"timestamp":"2023-08-29T22:38:56.408+08:00","sequence":261,"loggerClassName":"org.slf4j.impl.Slf4jLogger","loggerName":"io.debezium.server.databend.DatabendChangeConsumer","level":"INFO","message":"current record size: 25 time write into databend: 1542.80475 ms","threadName":"pool-7-thread-1
me":"hanshanjiedembp","processName":"io.debezium.server.Main","processId":34268}
```

Kafka Connect (coming soon)

Kafka Connect 可以很容易地将数据从多个数据源流到 Kafka，并将数据从 Kafka 借助 kafka connector 流到多个数据目标。Kafka Connector 支持上百种不同的数据连接器。



Databend

Databend Driver

Rust Driver: https://github.com/datafuselabs/bendsql/tree/main/driver

Node.js Driver: https://github.com/datafuselabs/bendsql/tree/main/bindings/nodejs

Python Driver: https://github.com/databendcloud/databend-py

SQLAlchemy: https://github.com/databendcloud/databend-sqlalchemy

Golang Driver: https://github.com/databendcloud/databend-go

JDBC: https://github.com/databendcloud/databend-jdbc

Databend

Thank you!