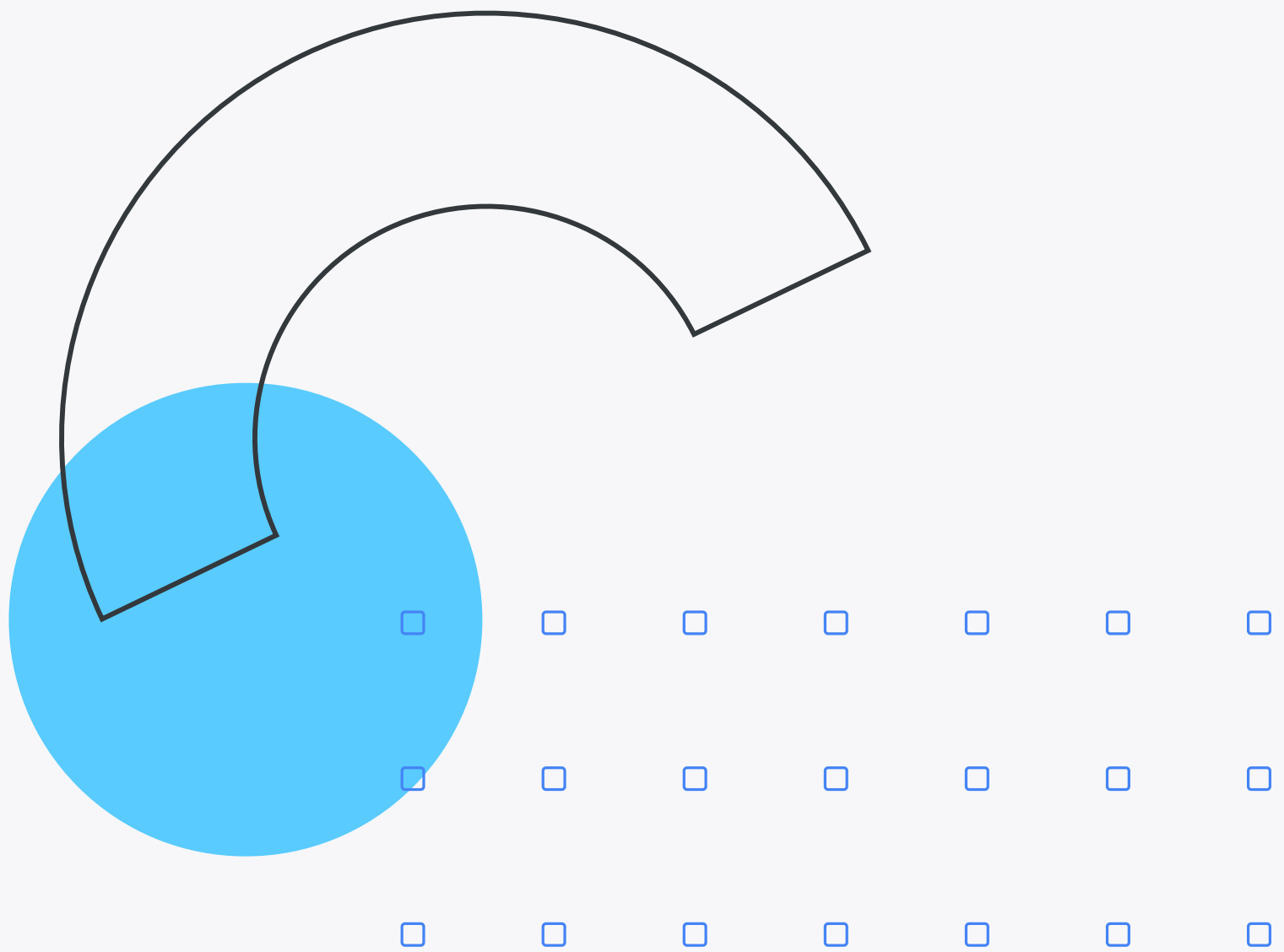
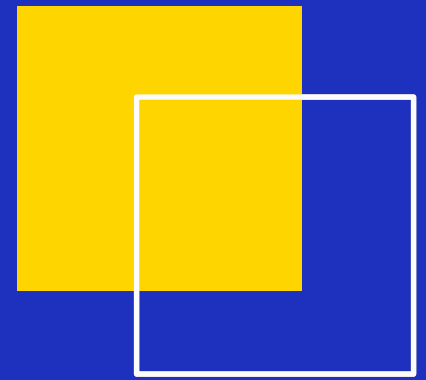


Databend

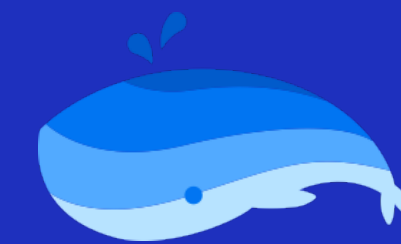
实战 Databend 助力 MySQL 归档分析

主讲人：吴炳锡





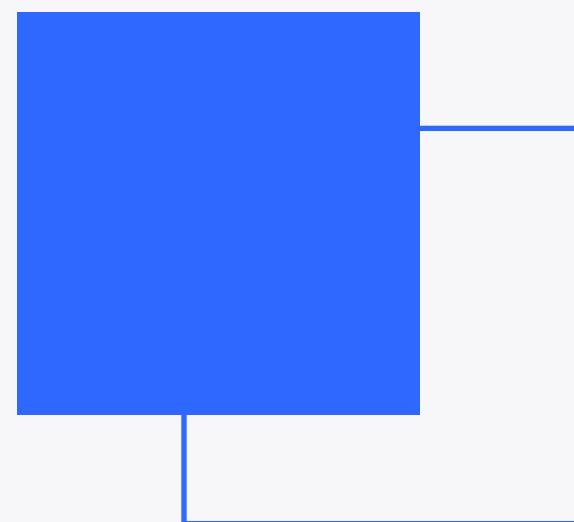
吴炳锡



Databend

- Datafuse Labs 联合创始人
- 腾讯 TVP 成员
- TGO 成员，原知数堂联合创始人
- 专注 Databend 大数据分析及应用
- 熟悉 MySQL 生态应用

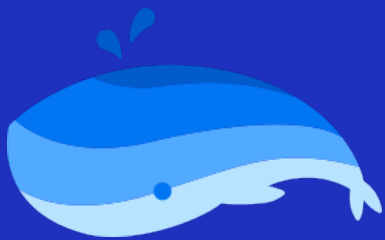
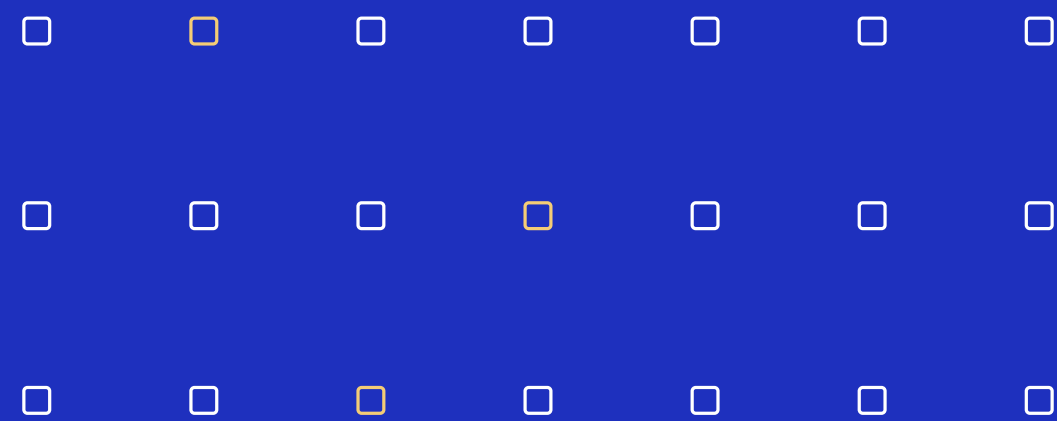




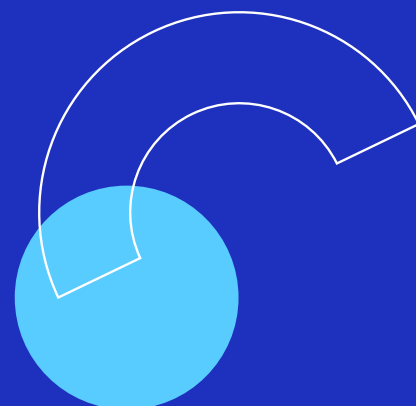
目录

CONTENTS

- MySQL 归档的方案及问题
- pt-archive & Databend
- TiDB dumping & Databend
- Metabase & Databend
- 使用 Databend 注意事项



Databend

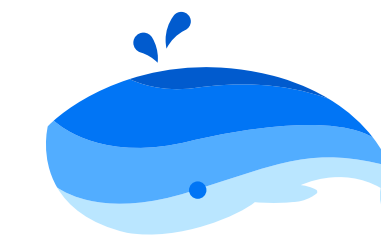


MySQL 归档的方案及问题





MySQL 归档的方案及问题



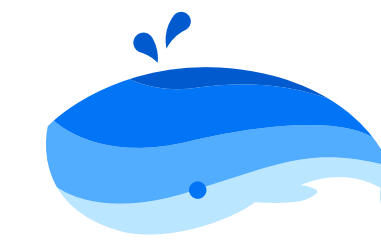
Databend

- MySQL 为什么需要归档分析服务
 1. MySQL 通常定位在 OLTP，使用服务器资源贵
 2. MySQL 本身不适合超大表
 3. 去上巨贵的 RDS，归档到线下提供查询
 4. 在 MySQL 中分库分表的业务，需要合并到一块进行数据分析
 - TiDB
 - Clickhouse





MySQL 归档的方案及问题

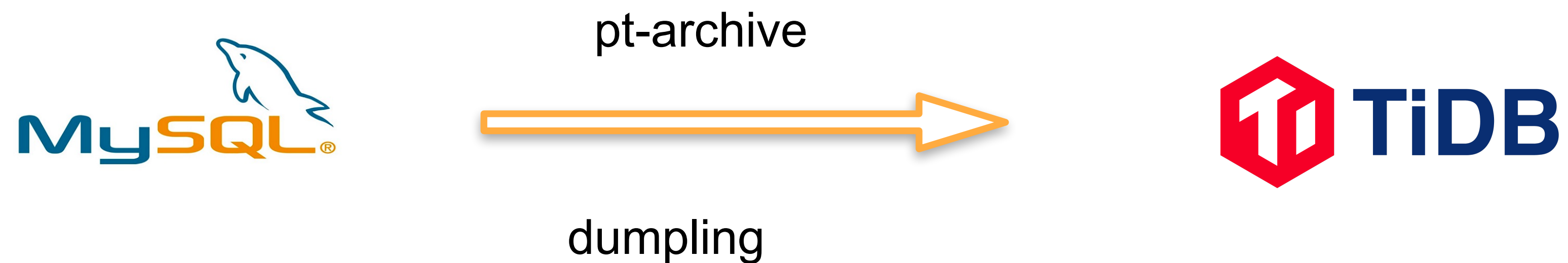
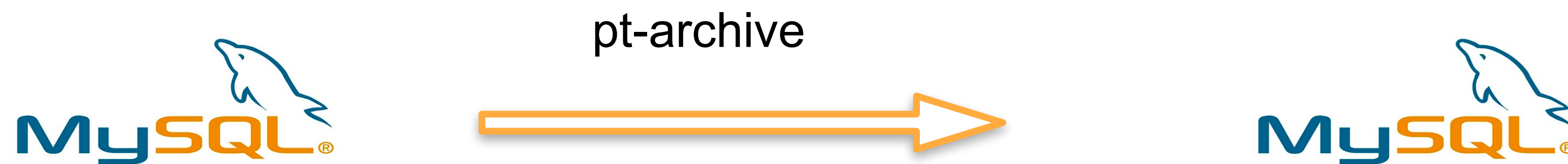


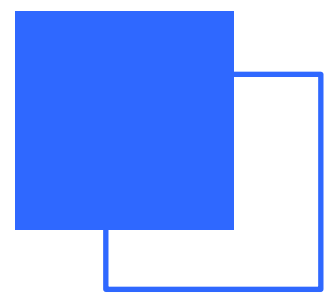
Databend

- MySQL 归档分析中引起的问题
 1. 原始 MySQL 删数据造成主从延迟严重
 2. 存储节点单点故障，恢复周期长
 3. 计算节点算力瓶颈，无法支撑复杂运算
 4. 非存储分离架构，大量资源闲置

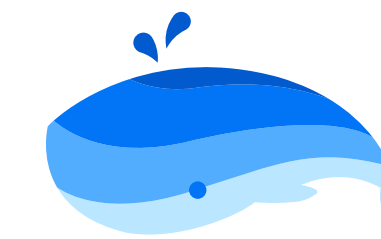


MySQL 归档的方案及问题





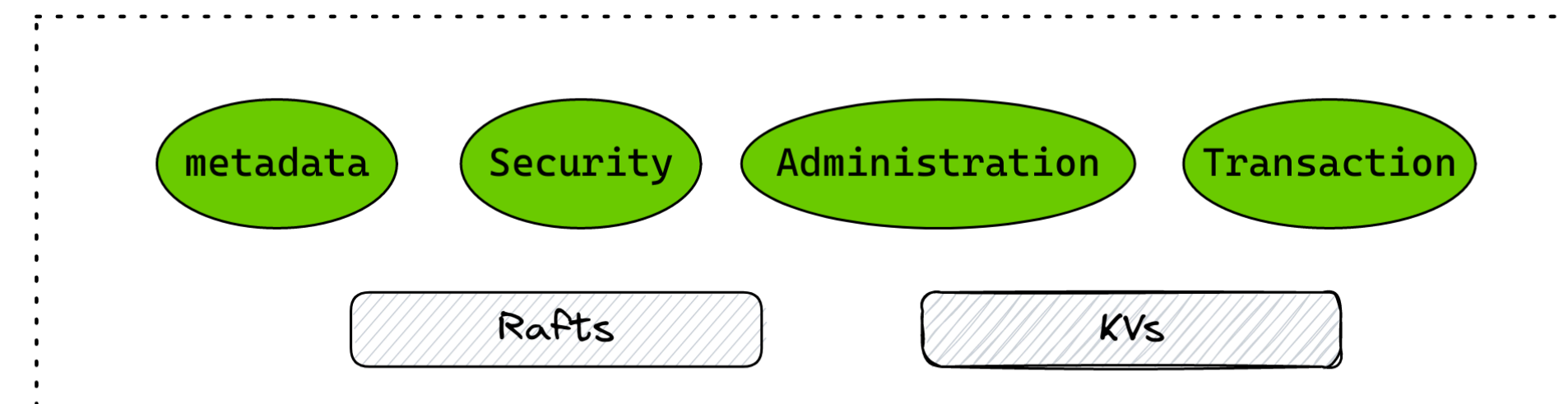
MySQL 归档的方案及问题



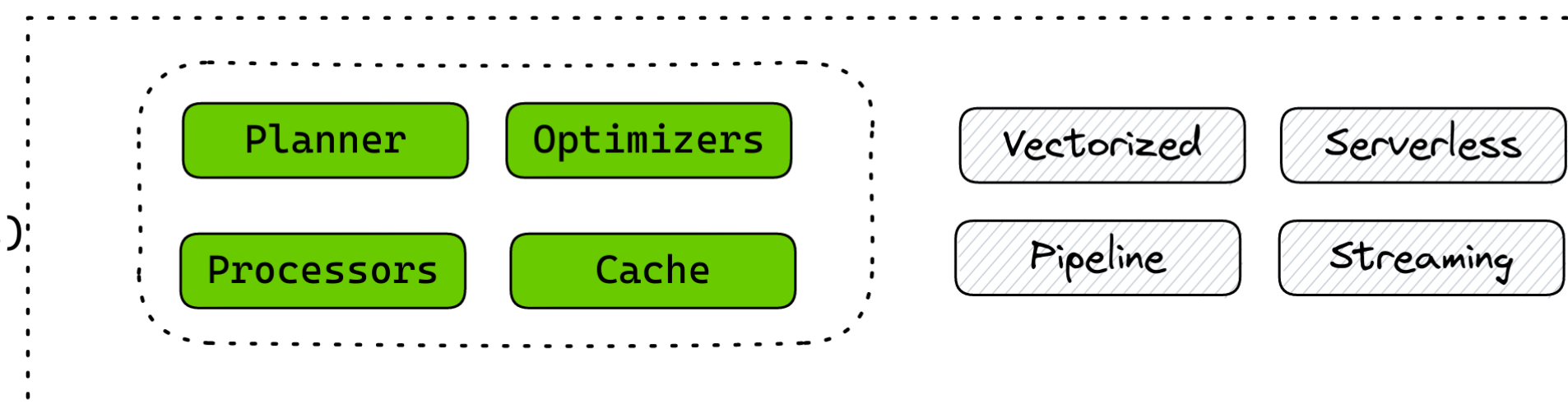
Databend

- Databend 介绍
 - 适用于云上的 DataLake 方案
 - 存算分离
 - 高性能，低成本，高扩展性
 - 易用：SQL 接入
 - 大数据存储分析方案

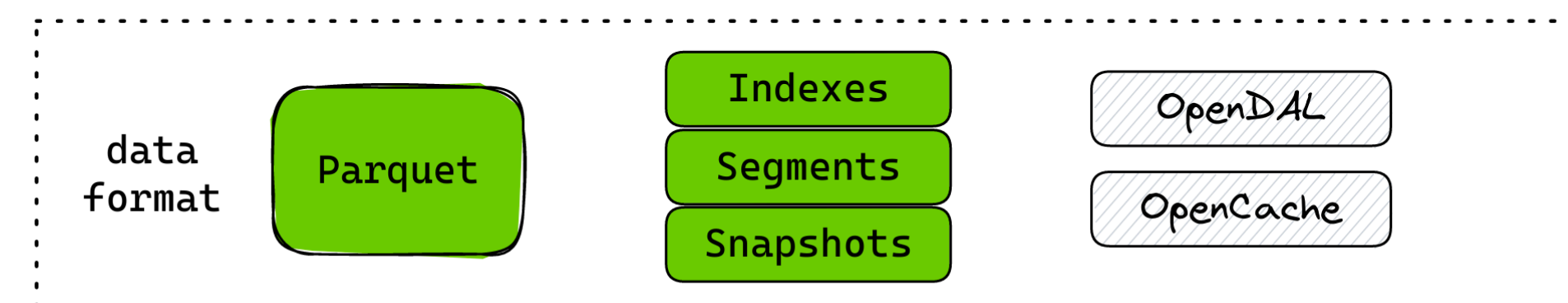
MetaService Layer
(elasticity)



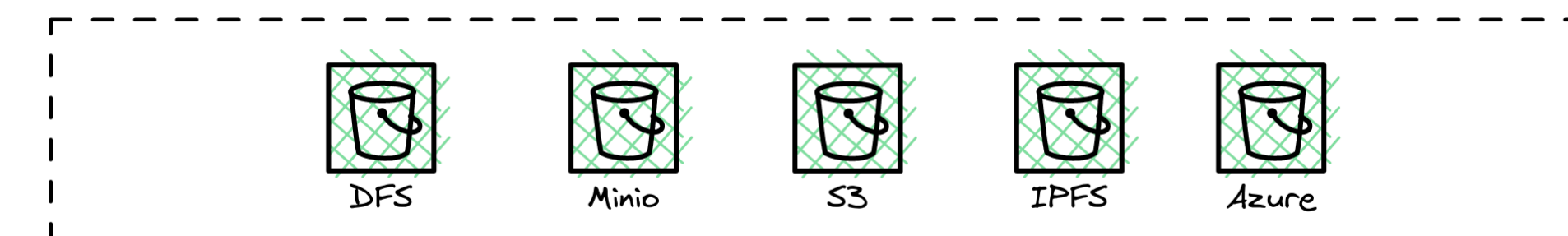
Query Layer
(elasticity & serverless)



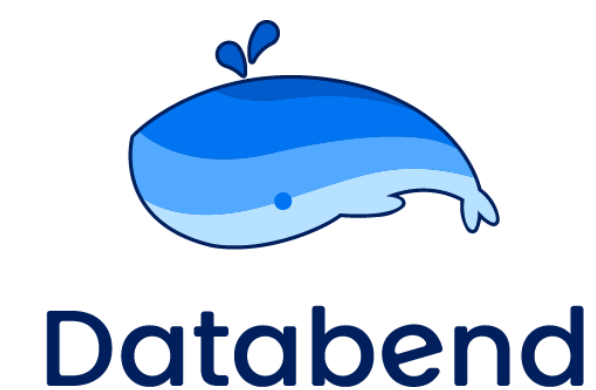
Storage Layer
(elasticity)



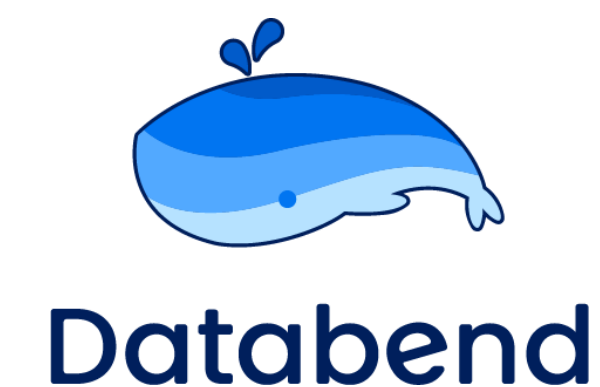
Storage Provision



MySQL 归档的方案及问题

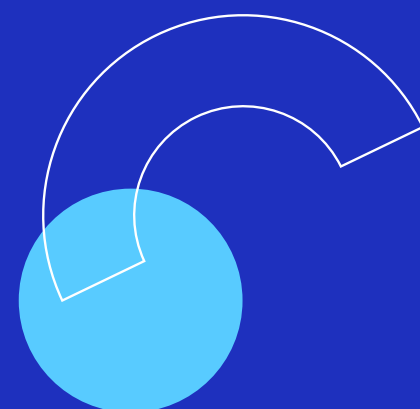
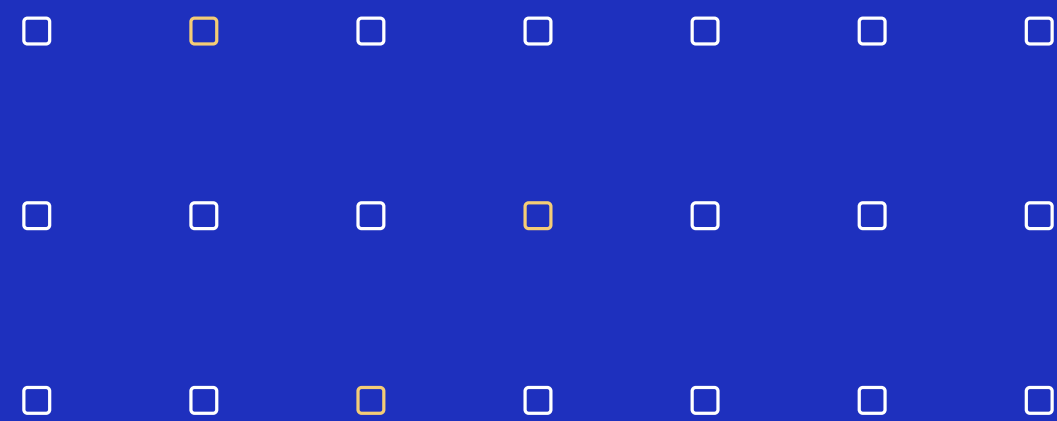


dumpling



海量存储
计算按需扩容

- Databend 介绍
 - SQL: MySQL, Clickhouse
 - Streaming_load: CSV, TSV, NDJSON
 - 基于对存存储的: COPY INTO @location
 - 基于网络: COPY INTO @http



pt-archive & Databend

- pt-archive 介绍
 - pt-tools 工具集成员
 - 从 MySQL 读取指定的数据到 csv 或是 MySQL
 - select -> insert (replace into) -> 慢
 - select -> load data -> JSON 支持不友好
 - select -> CSV -> JSON 支持不友好
 - 可以把指定的数据删除
 - 对 MySQL 主从环境友好

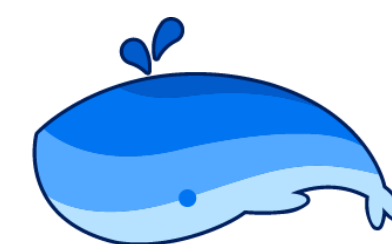
pt-archive & Databend



pt-archive



借助于 CSV | TSV



Databend

```
pt-archiver --source h=192.168.1.100,D=wubx,u=wubx,p=wubxwubx,t=sbtest1 \  
--where '1=1' \  
--file='all.csv' \  
--limit=10000 \  
--progress 10000 \  
--statistics \  
--no-check-charset \  
--no-delete
```

#Databend

```
curl -H "insert_sql:insert into wubx.sbtest1_bak format CSV" \  
-H "skip_header:0" \  
-H "record_delimiter:\n" \  
-F "upload=@./all.csv" \  
-u root: \  
-XPUT http://localhost:8000/v1/streaming_load
```


- 需要的额外技能
 - split 把大文件切成小文件
 - wc -l
 - split -l N filename all_
 - xargs |curl
 - ls all_* |xargs -l {} -P N curl ... {}

```
ls wubx.sbtest1.*.gz |xargs -l {} -P 4 curl -H "insert_sql:insert into wubx.sbtest1_bak format CSV" -H  
"skip_header:0" -H "compression:auto" -F "upload=@{}" -u root: -XPUT http://localhost:8000/v1/  
streaming_load
```

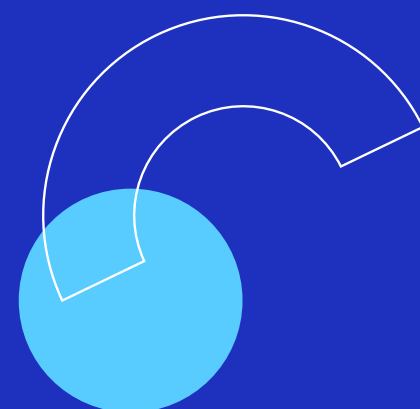
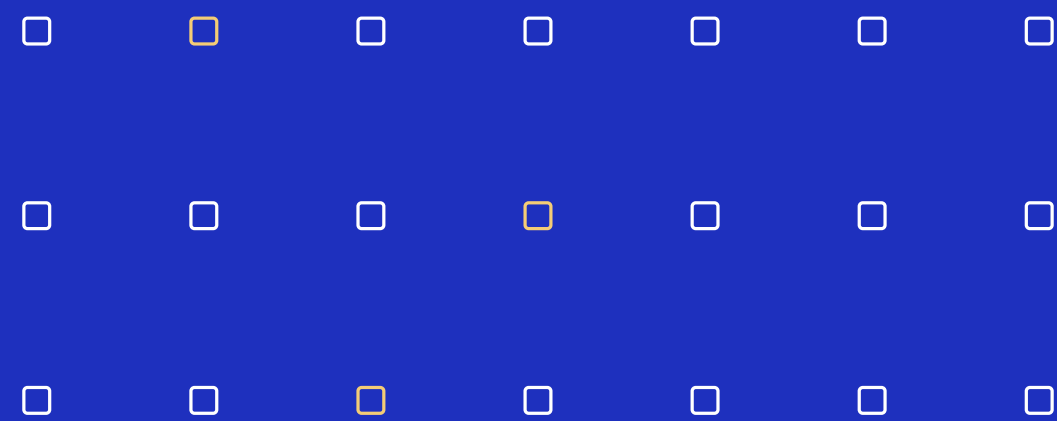
- streaming_load 接口: <https://databend.rs/doc/load-data/local>
- 参数 -H
 - insert_sql: 指定写入的表
 - skip_header: 是否跳过第一行
 - field_delimiter: 指定间隔符
 - record_delimiter: 指定行分隔符
 - compression: 压缩格式

```
curl -H "insert_sql:insert into wubx.sbtest1_bak format CSV" -H "skip_header:0" -H  
"compression:gzip" -F "upload=@a1.csv" -u root: -XPUT http://localhost:8000/v1/streaming_load
```

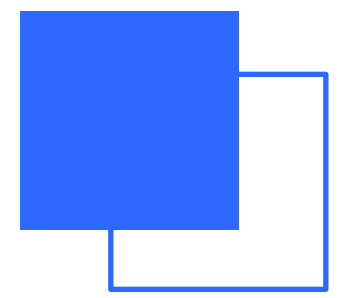
pt-archive & Databend



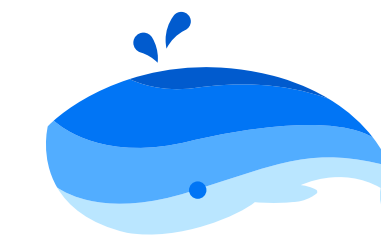
- streaming_load 接口
 - 支持的格式: CSV, TSV, NDJSON, Parquet
 - 支持并行
 - 更多参考: <https://databend.rs/doc/load-data/local>



TiDB dumping & Databend



TiDB dumping & Databend



Databend

- TiDB dumping
 - 灵感来源于 mydumper
 - 支持数据库备份成： SQL, CSV
 - 可以把数据备份在 S3 上
 - 更加丰富的 table-filter 支持
 - 支持并行

```
wget http://download.pingcap.org/tidb-toolkit-nightly-linux-amd64.tar.gz -O tools.tar.gz
```



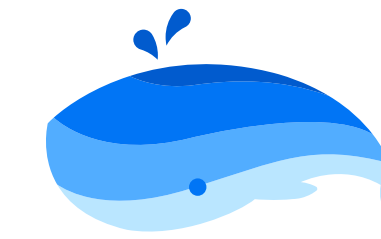
- dumping 使用

```
./dumping -h 127.0.0.1 -P 3306 -u wubx \  
-p wubxwubx \  
-T wubx.sbtest1 \  
--filetype csv \  
--no-header \  
-o ./ \  
-r 200000 \  
-c gzip \  
-m \  
-t 8
```

```
export AWS_ACCESS_KEY_ID= xxx  
export AWS_SECRET_ACCESS_KEY=xxx  
./dumping -h 127.0.0.1 -P 3306 -u wubx \  
-p wubxwubx -T wubx.sbtest1 \  
--filetype csv \  
--no-header \  
-o s3://databend-1255499614/ \  
--s3.endpoint=https://cos.ap-beijing.myqcloud.com \  
--s3.provider 'alibaba' \  
-r 200000 \  
-F 200M \  
-m \  
-t 8
```



TiDB dumping & Databend



Databend

- 加到数据到 Databend

```
mysql>create stage mydata url='s3://databend-1255499614' \  
connection=(endpoint_url='https://cos.ap-beijing.myqcloud.com' aws_key_id='xxx' secret_access_key='xxx');  
Query OK, 0 rows affected (0.01 sec)
```

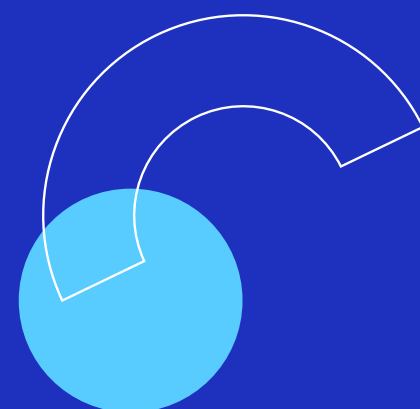
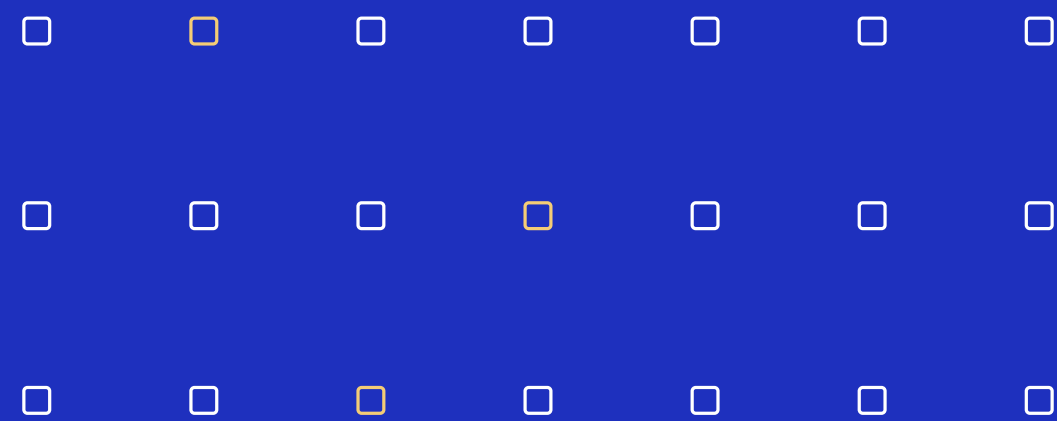
```
mysql>COPY INTO sbtest11 FROM @mydata pattern = 'wubx.sbtest1.*' FILE_FORMAT = ( TYPE=CSV  
compression=auto );  
Query OK, 0 rows affected (3 min 37.73 sec)
```

更多帮助: <https://databend.rs/doc/load-data/stage>



- 对比

	streaming_load	Copy into (s3)
sbtest 数据gz 文件加载 数据大小：17.86G (gz) 文件数量：196文件	并行： 16 用时： 3m47.823s block 数量: 388	并行： 16 用时： 3 min 37.73 sec Block 数量： 387
sbtest 解压后的 csv 文件加载 数据大小：39G 文件数量： 196个	并行： 16 用时： 3m48.221s	并行： 16 用时： 3 min 37.22 sec



Metabase & Databend

Metabase & Databend



- Metabase 介绍
 - 开源的 BI 分析工具
 - 支持丰富的数据源
 - mysql
 - ...
- 推荐场景
 - BI
 - 内部数据查询平台



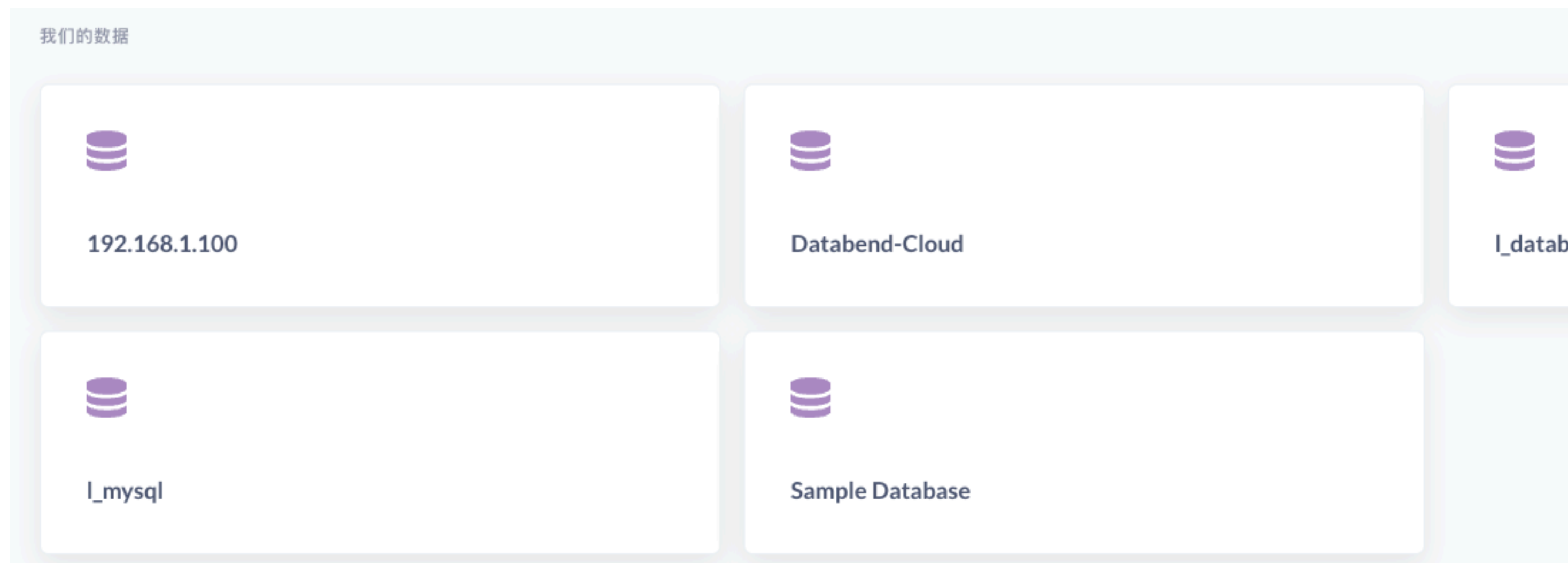
Metabase & Databend



- Metabase & Databend

```
docker pull metabase/metabase:latest
```

```
docker run -d -p 3000:3000 --name metabase metabase/metabase
```



数据库 > 添加数据库

数据库类型

MySQL

显示名称

192.168.1.100

Host

192.168.1.100

Port

3307

Database name

wubx

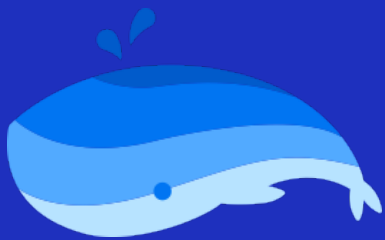
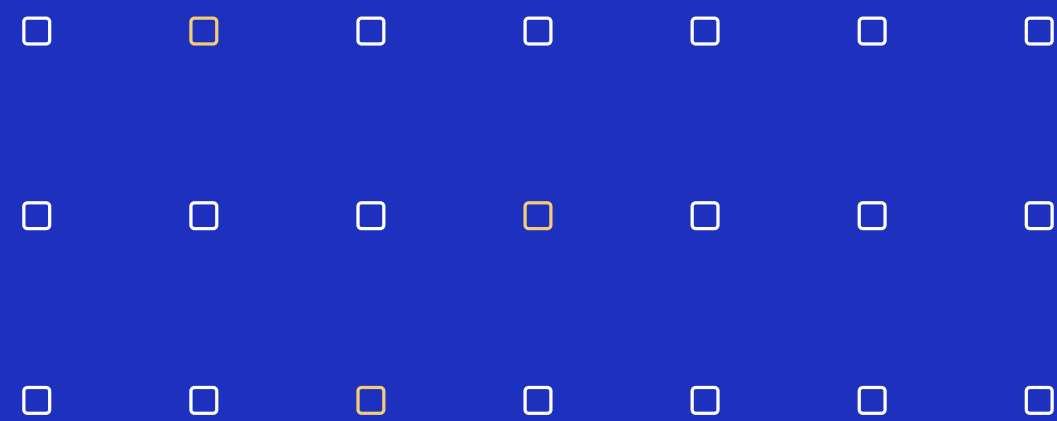
Username

wubx

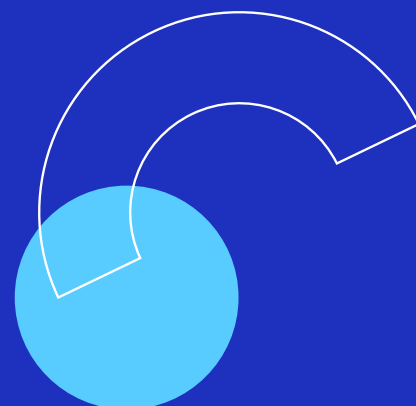
Password

••••••••

<https://www.metabase.com/docs/latest/installation-and-operation/running-metabase-on-docker>



Databend

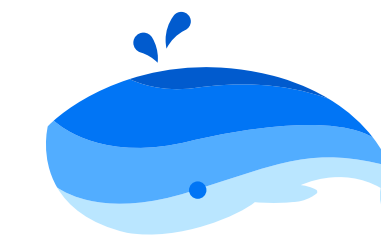


使用 Databend 注意事项





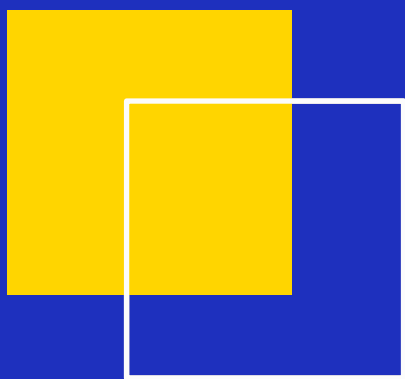
使用 Databend 注意事项



Databend

- 建议批量写入： `copy into` , `stream load`
- 比较适合分析类 SQL
- 复杂 SQL 建议使用 Cluster 架构
- 考虑集群专用： 数据导入集群 & 计算集群
 - 数据要不要进行 compact 操作？
- 选用合适的存储： 优先云上对象存储， 次优先： minio, ceph 类
- 利用 github issue 反馈问题





THANKS!

