



GOTC 2023

全球开源技术峰会

THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE

OPEN SOURCE, INTO THE FUTURE

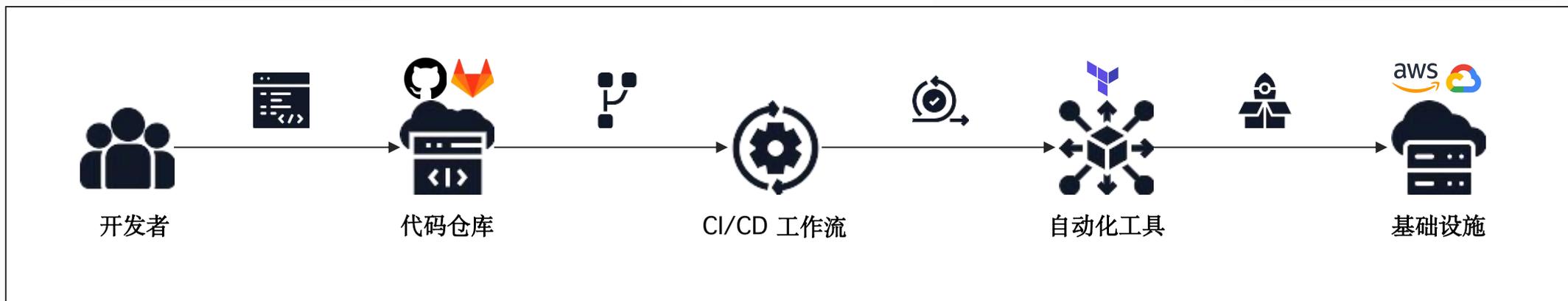
「数据与数据库技术」专场

本期议题：构建基于 GitOps 的数据库变更 workflow

王长煜 2023年05月28日

什么是 GitOps

GitOps 是一种基于 Git 仓库的持续交付工作流，将基础设施和应用程序的配置或状态信息存储在 Git 仓库中，并使用自动化工具对其进行变更管理。



使用 Git 来存储定义和配置信息，轻松实现版本化管理，确保目标基础设施的配置一致性。



GitOps 的 CI/CD 流程可以自动化应用程序的部署和回滚，减少人为错误，提升部署效率。



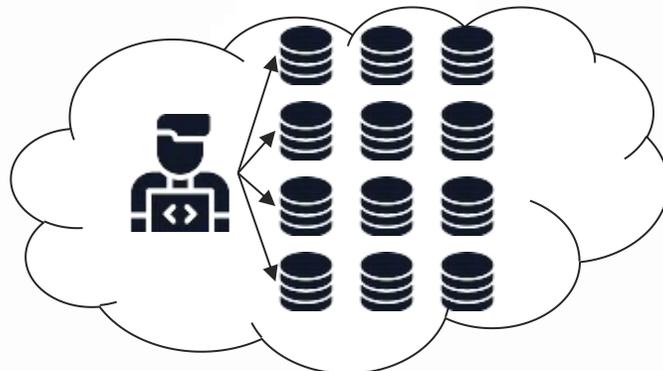
GitOps 可以轻松地在多个环境和平台上部署。



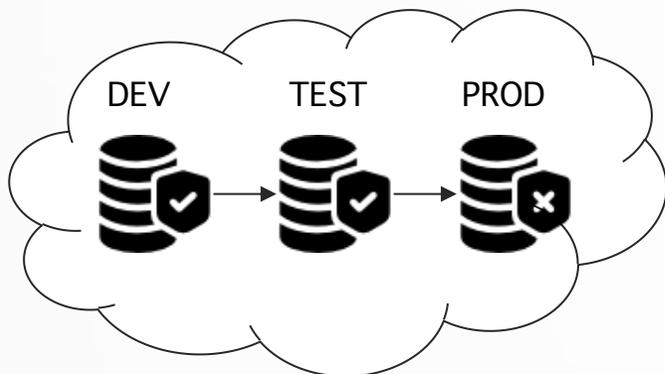
GitOps 可以提供实时状态和配置信息，以及所有变更的记录。

数据库的变更是否需要 GitOps?

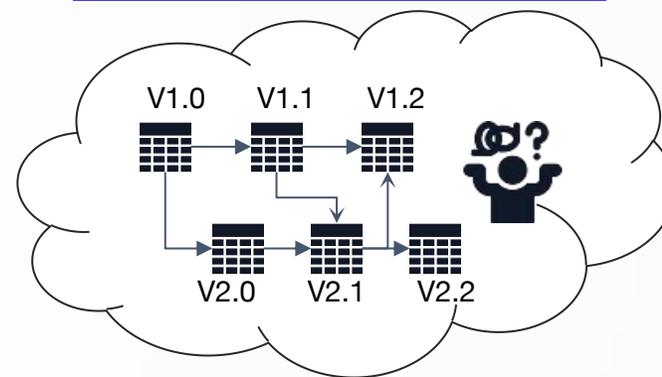
手工批量变更上百个库



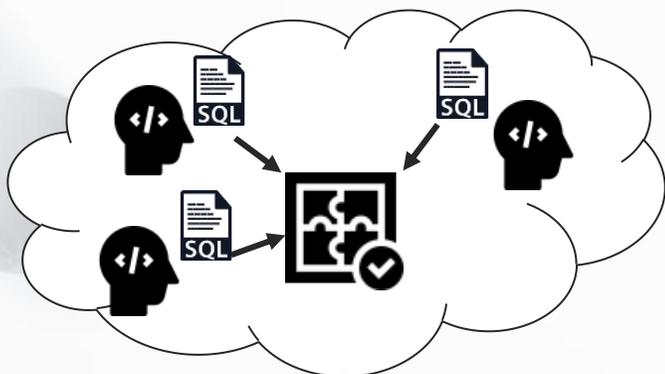
测试到生产脚本不一致



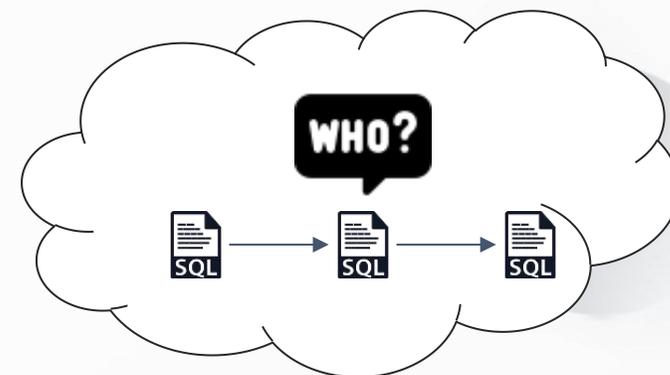
多版本 Schema 管理混乱



脚本散落在多人电脑中

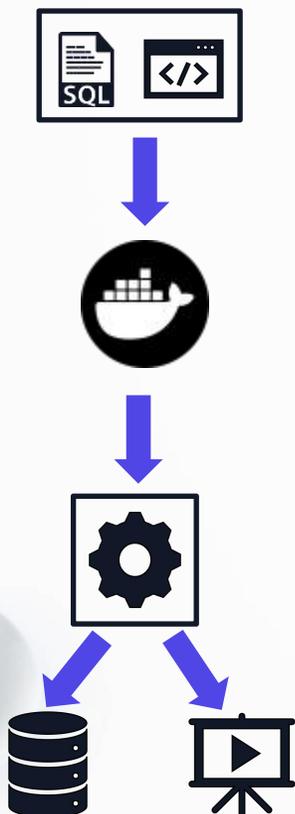


变更历史无法追溯



现有的数据库 GiOps 方案

数据库变更脚本与
应用代码打包发布
紧耦合



- ❖ **语法正确性**: 验证脚本语句是否符合模板引擎的语法
- ❖ **业务正确性**: 是否满足应用发布需求
- ❖ **架构正确性**: 是否符合性能与扩展性等要求



- ❖ **大型变更**: 对于数据量庞大的表发起变更
- ❖ **批量发布**: 一次对多个同构库发起变更
- ❖ **流水发布**: 将一个变更从测试环境逐阶段发布到生产环境
- ❖ **灰度发布**: 挑选部分库先行发布, 再逐步扩展到所有库
- ❖ **临时性变更**: 不涉及应用变更的数据库变更



- ❖ **脚本修正**: 基于错误原因快速调整脚本语句
- ❖ **快速撤销**: 执行反向语句撤销本次变更
- ❖ **全库回滚**: 将数据库回退到变更前的某个时间点



具备强大的数据库变更管理能力，与代码仓库集成，提供独立的发布流，同时能与应用发布联动



原生 Git 集成

代码仓库内审核

独立发布工作流

联动应用发布



自动化审核

语法分析

业务规范

架构规范

自定义规范集成



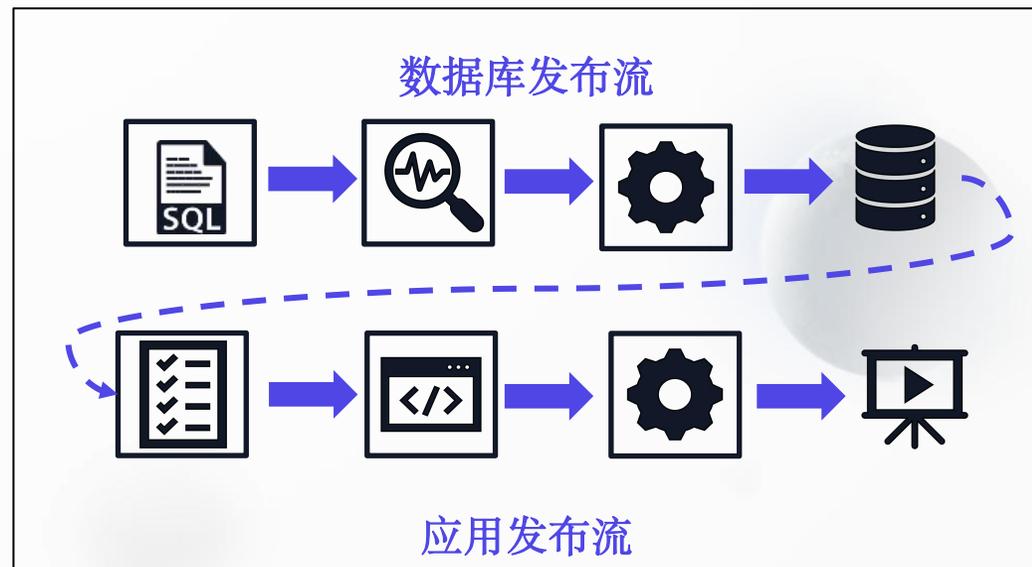
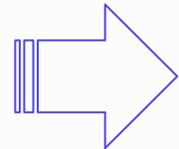
发布管理

权限控制

多种发布功能

版本管理

变更回滚



以项目（Project）为边界的自助管理

- ❖ 自定义审批，基于角色与风险的审批流程
- ❖ 自动审核，支持近百条常见审核规范
- ❖ 自动发布，审核通过的变更可以一键自动发布到目标库

The screenshot displays the Bytebase interface for a 'Create product table' task. The task is currently in a 'Waiting Approval' state. The workflow progress bar shows: Dev (Add initial schema) -> Integration (Add initial schema) -> Staging (Add initial schema) -> Prod (Add initial schema). The 'Integration' stage is the current active stage. Below the progress bar, there is a table for 'Add initial schema' with one entry: 'Done' by 'Bytebase' on 'Jun 28'. The SQL query is: `CREATE TABLE product (id INTEGER PRIMARY KEY AUTO_INCREMENT, name TEXT NOT NULL, description TEXT NOT NULL);`. The task description is 'Create product table'. The right sidebar shows task details: Status: Open, Assignee: Jerry DBA, Stage: Integration, Database: shop, Instance: On-premises Dev MySQL, Environment: Integration, Project: Shop (Git), Updated: July 4, 2021 8:53 PM, Created: June 28, 2021 5:48 PM.

创新的 GitOps 实践：融入代码仓库的 SQL 审核能力



- ❖ 符合开发团队工作习惯与流程，无须在多个工具间切换
- ❖ CI 阶段预审核大幅降低上线风险，增加容错时间



Level	Line	Name	Code	Doc
ERROR	Line 1	table.require-pk	601	docs
ERROR	Line 1	table.comment	605	docs
WARN	Line 1	column.required	401	docs
WARN	Line 1	column.no-null	402	docs
ERROR	Line 1	column.comment	408	docs
WARN	Line 1	column.require-default	400	docs

```
@@ -0,0 +1 @@
1 create table demo_app(id int);
```

Bytebase / SQL Review
[ERROR] table.require-pk (601)
Table `demo_app` requires PRIMARY KEY
Docs: <https://www.bytebase.com/docs/reference/error-code/advisor#601>

Bytebase / SQL Review
[ERROR] table.comment (605)
Table `demo_app` requires comments
Docs: <https://www.bytebase.com/docs/reference/error-code/advisor#605>

Bytebase / SQL Review
[WARN] column.required (401)
Table `demo_app` requires columns: created_ts, creator_id, updated_ts, updater_id
Docs: <https://www.bytebase.com/docs/reference/error-code/advisor#401>



Update bytebase/Prod/hublocal##20221014##ddl##create_table.sql

Request to merge ed-main-patch-35429 into main

Detached merge request pipeline #181 failed for cf8c16f0 1 week ago

Test summary contained 3 failed out of 3 total tests

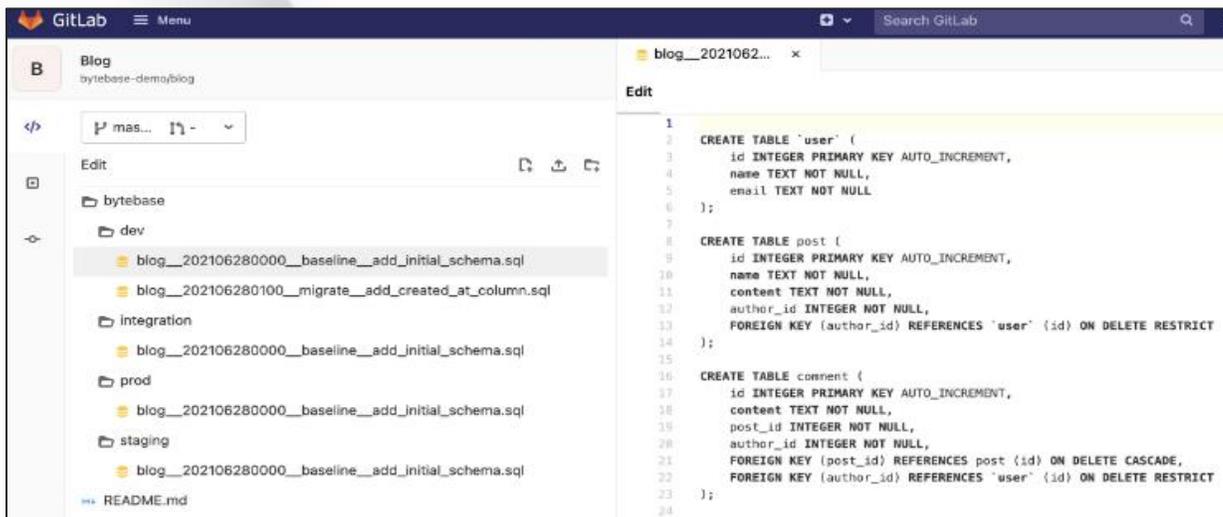
bytebase-vcs-sql-review found 3 failed out of 3 total tests

New

- column.required
- column.no-null
- table.require-pk

Merge Delete source branch

Adds 1 commit and 1 merge commit to main. [Modify merge commit](#)



```
1 databases:
2   - name: supermarket
3   - name: supermarket_east
4   - name: supermarket_west
5   - name: supermarket_south
6   - name: supermarket_north
7 statement: |
8   INSERT INTO company (id, name, address) VALUES (1,
9     'Bytebase', '1 DevOps street');
```

由代码仓库触发的数据库变更请求

❖ 自动触发数据库审核与发布流程

- 监听指定仓库、指定分支、指定目录
- 基于 Webhook

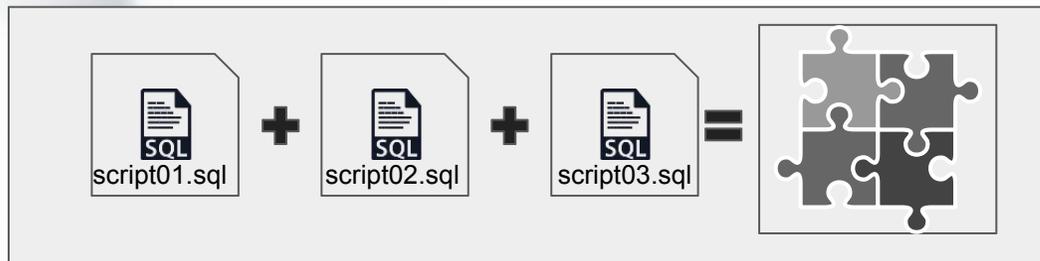
❖ 支持主流代码仓库

- GitLab
- GitHub
- BitBuket

❖ 多种文件识别模式

- 模式一：基于文件名模板
- 模式二：基于 YAML 格式

SQL 发布：基于脚本的命令式变更发布



版本控制系统（VCS）中保存每一次变更的脚本，依序执行即可得到任意时间点的数据库结构。

Projects Databases Instances Environments Settings Plan Free Team Role Owner DBA Developer DO

Database > db1

db1 Transfer Project Alter Schema

Environment - Dev Instance - Localhost Dev MySQL Project - Default

Overview **Migration History** Backups

Migration History Establish new baseline

Workflow	Version	Type	Issue	Description	Duration	Created	Creator
✓ UI	20211008114339.11017	MIGRATE	13010	Update db1 schema	a few seconds	11:43	Demo Owner
✓ UI	20211008014909.11019	MIGRATE	13012	Update db1 schema	a few seconds	Oct 8 01:49	Demo Owner
✓ UI	20211008014847.11018	MIGRATE	13011	Update db1 schema	a few seconds	Oct 8 01:48	Demo Owner
✓ UI	20211008014815.11017	MIGRATE	13010	Update db1 schema	a few seconds	Oct 8 01:48	Demo Owner
✓ UI	20211008012549.11017	MIGRATE	13010	Update db1 schema	a few seconds	Oct 8 01:25	Demo Owner
✓ UI	20211006233228.11020	BASELINE	13013	Establish db1 baseline	a few seconds	Oct 6 23:32	Demo Owner

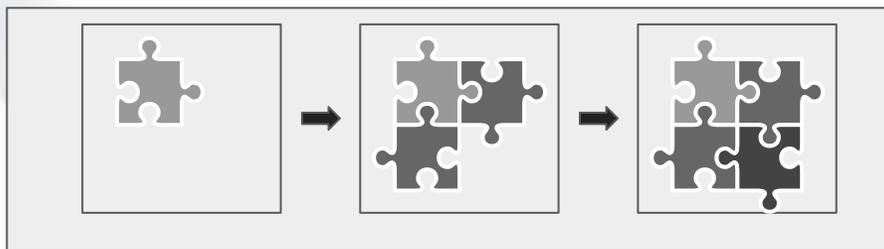
优点：

- ❖ 符合当前数据库变更工作习惯
- ❖ 变更过程透明可控
- ❖ 兼容所有主流数据库
- ❖ 灵活挑选特定变更发布
- ❖ 支持 DML

缺点：

- ❖ 需要维护大量脚本执行顺序
- ❖ 需要解决冲突问题

SQL 发布：基于终态的声明式变更发布



版本控制系统（VCS）中保存一份唯一标准的数据库结构文件，所有的 Schema 变更都通过修改该文件执行，Bytebase 自动将变更同步到对应数据库。

```
Issues Projects Databases Instances Environments Settings
> Databases > employee > Change > 20230406034143
Statement
ALTER TABLE
`employee`
ADD COLUMN
(`Nickname` VARCHAR(16) NOT NULL);

Schema Snapshot
Show diff Prev (20230404072400) vs This version

@@ -316,20 +316,21 @@
316 `gender` enum('M','F') NOT NULL,
317 `hire_date` date NOT NULL,
318 `EduBackgrounds` varchar(50) NOT NULL,
319 `Email` varchar(40) NOT NULL,
320 `PhoneNo` varchar(15) NOT NULL,
321 `Credentials_type` char(4) NOT NULL,
322 `Credentials_No` varchar(10) NOT NULL,
323 `Manager` varchar(14) DEFAULT NULL,
324 `Emergency_name` varchar(16) NOT NULL,
325 `Emergency_phoneNo` varchar(15) NOT NULL,
326 + `Nickname` varchar(16) NOT NULL,
327 PRIMARY KEY (`emp_no`)
328 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
329
330 -- Table structure for `ffadfa`
331 --
332 CREATE TABLE `ffadfa` (
333 `id` int NOT NULL,
334 PRIMARY KEY (`id`)
335 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

优点：

- ❖ 简化脚本编写，无需编写复杂的 DDL 语句
- ❖ 简化执行管理，无需管理语句执行顺序
- ❖ 自动变更合并，将多个开发提交的变更汇总执行，大幅降低变更执行时间
- ❖ 维护一份唯一的数据库结构，不再有结构冲突等原因导致发版失败

缺点：

- ❖ 无法应对 DML 语句与对象重命名
- ❖ 需要数据库引擎支持

常见的数据库变更发布场景

1

单库发布

对单库进行标准的SQL 脚本发布

2

批量发布

对多个数据库发起相同的变更，多见于分库分表、SaaS 多租户等场景

3

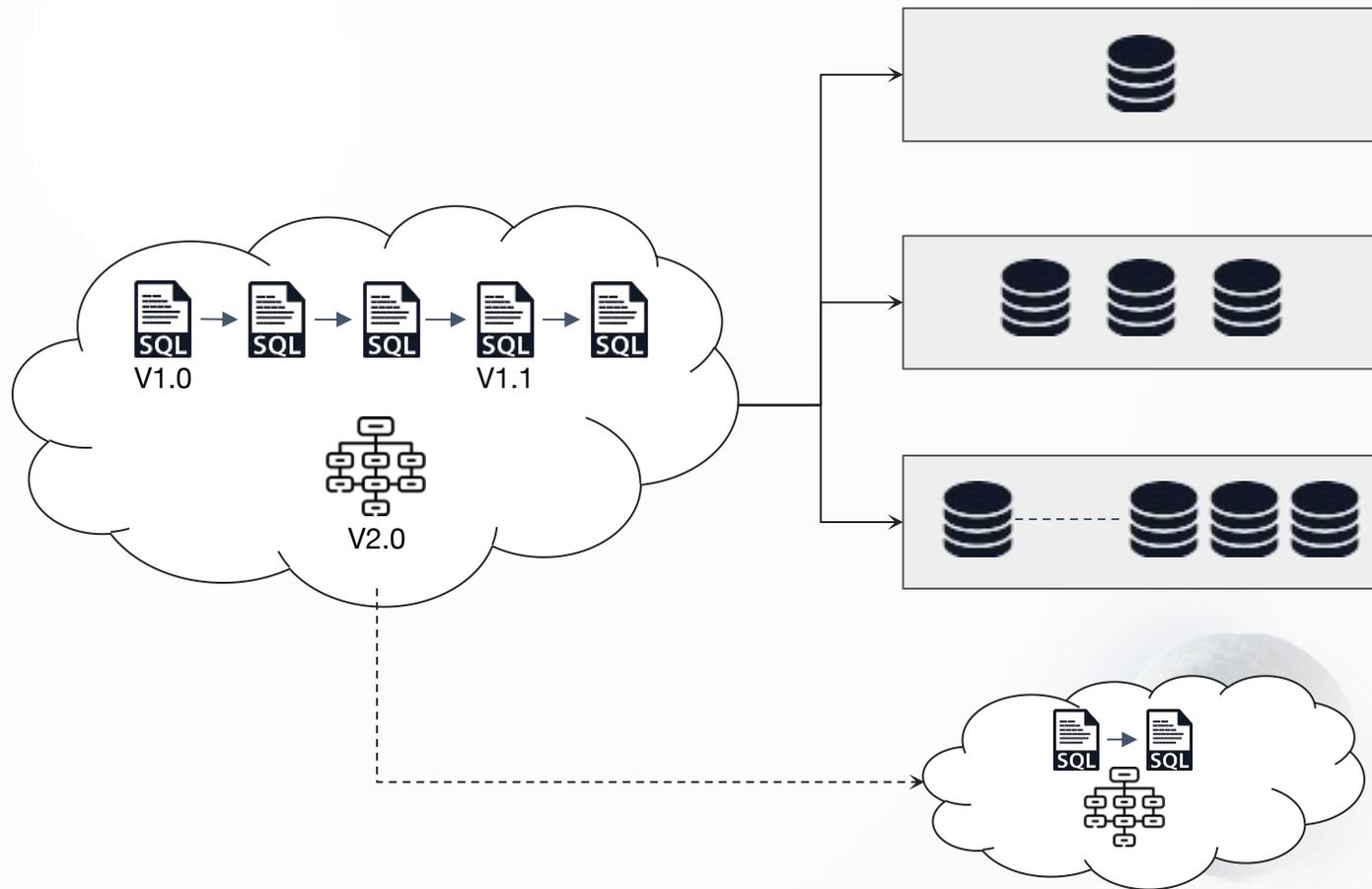
分阶段发布

一次变更贯穿测试、预上线、生产等多个环境，或对生产环境进行分批灰度发布

4

跨环境发布

物理隔离的环境下的变更发布，多见于用户的私有化部署场景



常见发布场景的 GitOps 实现

1

单库发布

对单库进行标准的SQL 脚本发布

```
bytebase/prod/employee##202303040001##ddl##create_table_salary.sql
```

2

批量发布

对多个数据库发起相同的变更，多见于分库分表、SaaS 多租户等场景

```
databases:  
  - name: supermarket_east  
  - name: supermarket_west  
statement: |  
  INSERT INTO company (id, name, address) VALUES (1, 'Bytebase', '1  
  DevOps street');
```

3

分阶段发布

一次变更贯穿测试、预上线、生产等多个环境，或对生产环境进行分批灰度发布

```
bytebase/employee##202303040001##ddl##create_table_salary.sql
```

4

跨环境发布

物理隔离的环境下的变更发布，多见于用户的私有化部署场景

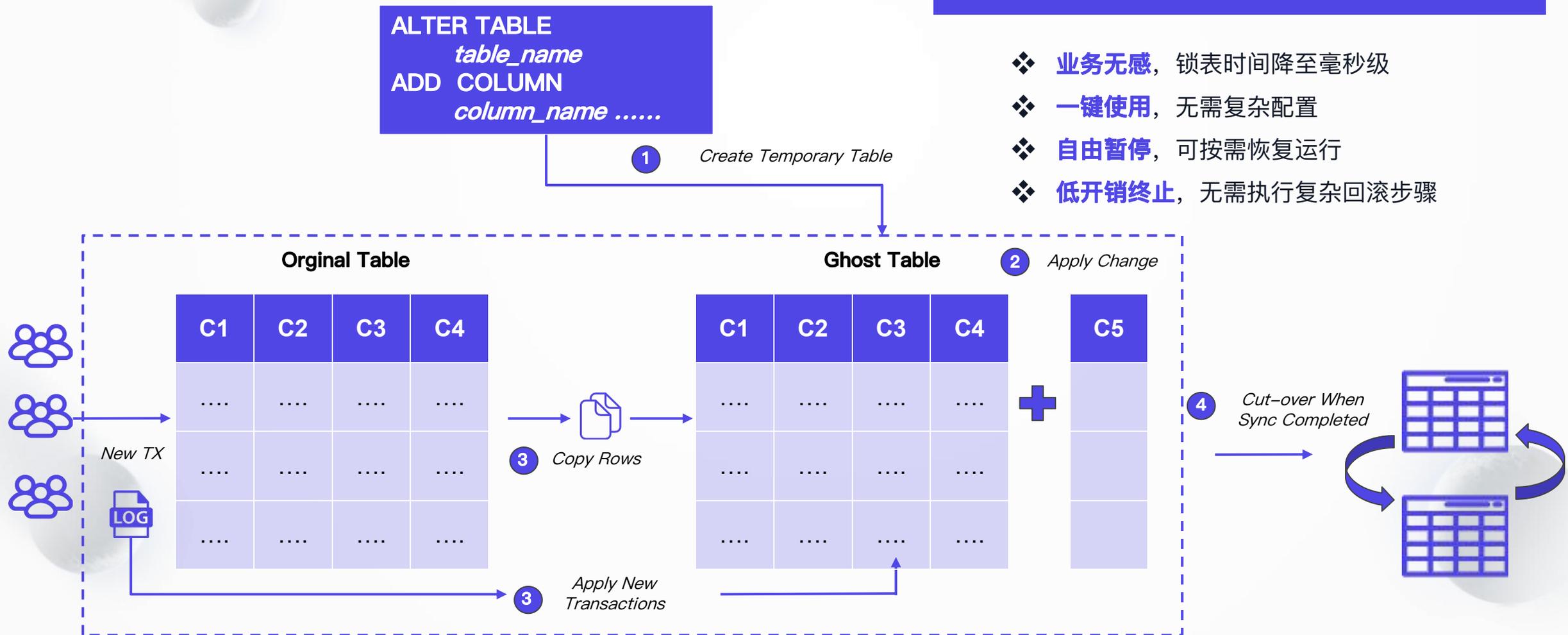
```
cp  
bytebase/test/  
  employee##202303020001##ddl##create_table.sql  
  employee##202303020002##ddl##create_index.sql  
  employee##202303030001##ddl##add_column.sql  
to  
bytebase/prod/
```

提供高级变更方案应对复杂生产环境

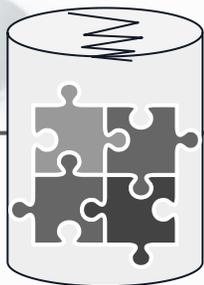
基于 Gh-ost 的 MySQL 大表无锁变更

```
ALTER TABLE
  table_name
ADD COLUMN
  column_name .....
```

- ❖ 业务无感，锁表时间降至毫秒级
- ❖ 一键使用，无需复杂配置
- ❖ 自由暂停，可按需恢复运行
- ❖ 低开销终止，无需执行复杂回滚步骤



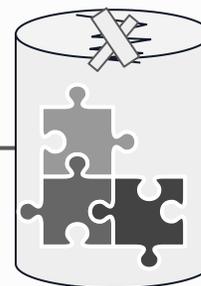
变更失败的应对



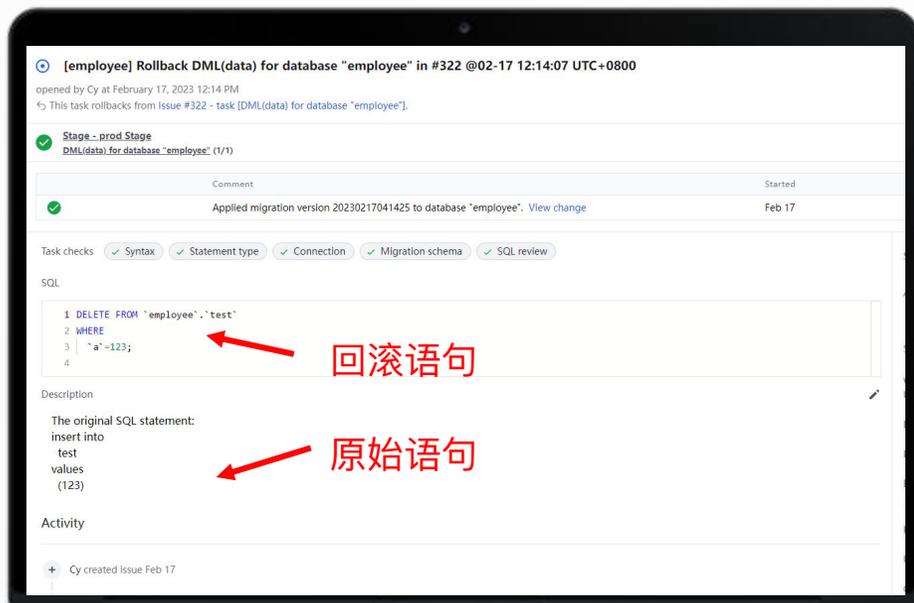
变更失败



日志挖掘



变更撤销



提供产品化的变更失败处理能力

- ❖ 一键挖掘日志生成撤销语句
- ❖ 预览工单，按需选择所需回滚语句
- ❖ 可在变更前后随时激活

变更完成自动回写 Schema 并联动应用发布

记录所有变更历史

向代码仓库回写 Schema

The screenshot shows the Bytebase interface. On the left is a sidebar with navigation options like Home, SQL Editor, Anomaly Center, and Projects. The main area displays a list of database changes with columns for ID, type, description, duration, and date. A SQL editor window is open, showing MySQL database structure for 'db1' and 't10'.

ID	Type	Description	Duration	Date	Environment
20230216094838	UI	insert into test values (10);	a few seconds	Feb 16 2022	Cy
20230214072448	UI	insert into test values (100);	a few seconds	Feb 14 2022	Cy
20221202081843	UI	create table test78 (id int PRIMARY KEY);	a few seconds	Dec 12 2022	Cy
20221202081649	UI	create table test789 (id int PRIMARY KEY);	a few seconds	Dec 2 2022	Cy
20221202074022	UI	create table test123 (id int PRIMARY KEY);	a few seconds	Dec 2 2022	Cy
202206270013	VCS	insert into test values (123);	a few seconds	Sep 9 2022	cy
202206270011	VCS	insert into test values(1);	a few seconds	Sep 9 2022	cy
202206270010	VCS	insert into test values(1);	a few seconds	Sep 9 2022	cy

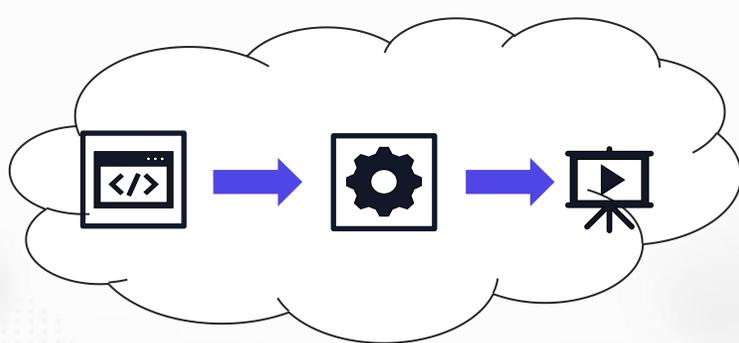
The screenshot shows a commit in a code repository. The commit message is "[Bytebase] Update latest schema for 'employee' after migration 202206270010". The code content shows SQL statements for creating tables like 'T_Employee', 'demo001', 'demo005', and 'department'.

webhook (可选)

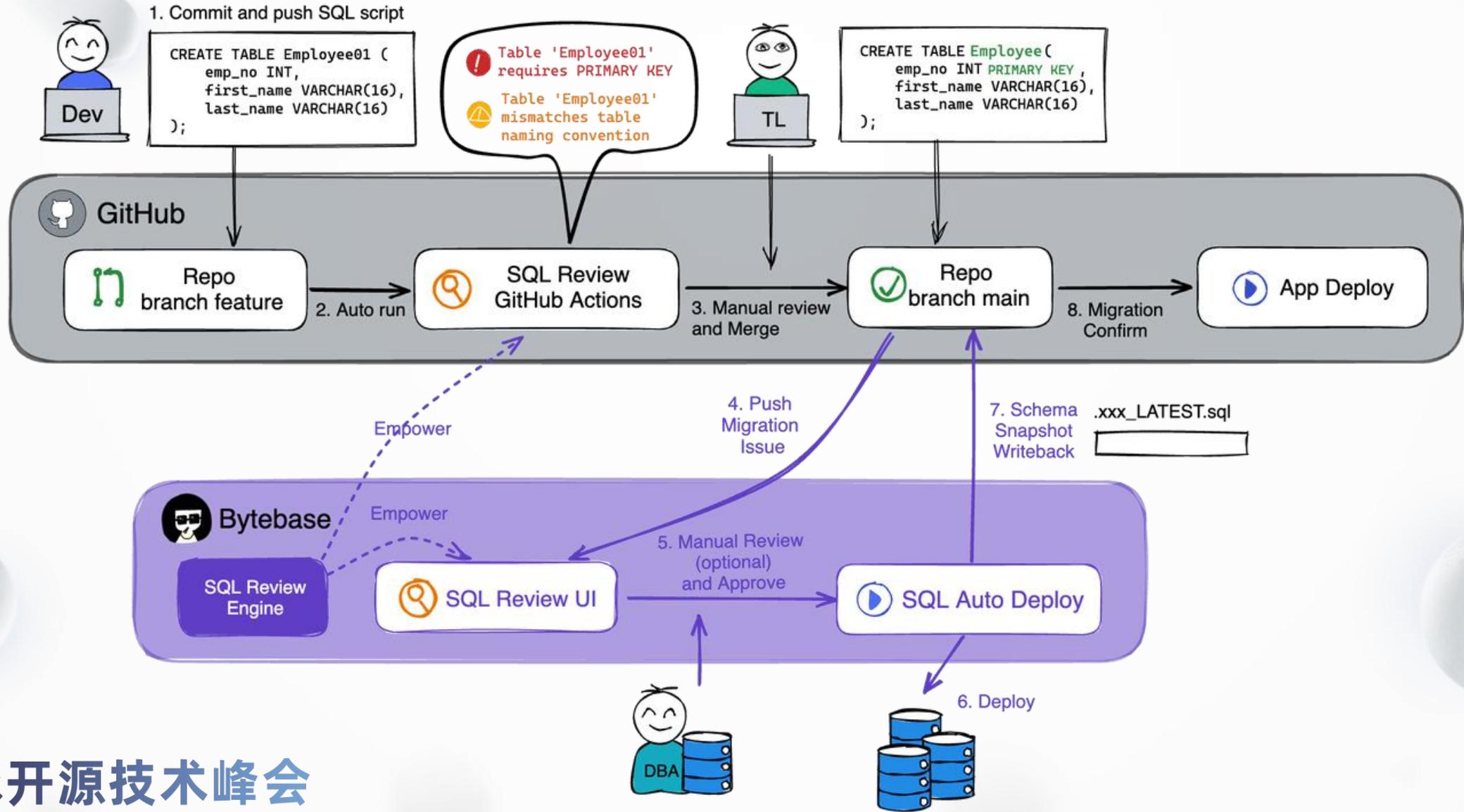
Git Pipeline (可选)

全球开源技术峰会

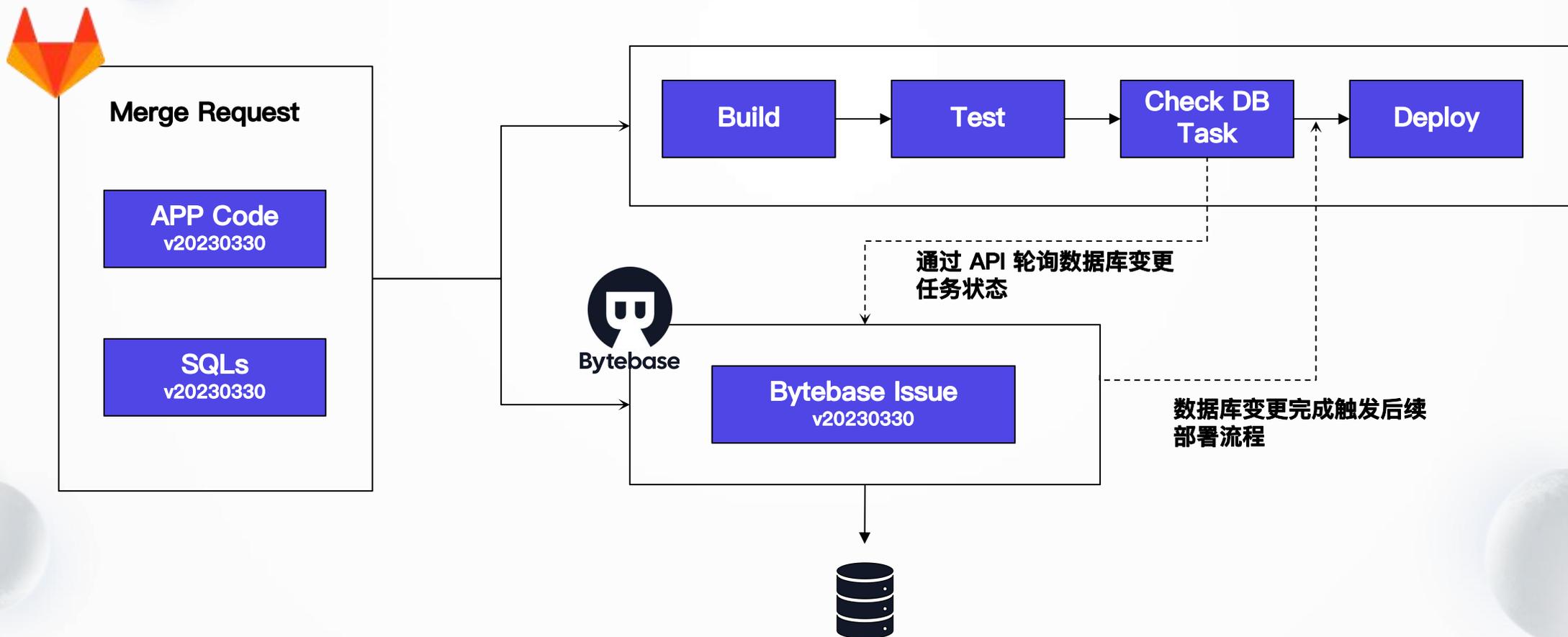
THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE



场景示例：一个完整的 Database CI/CD 流程



场景示例：将数据库变更纳入应用发布流程实现版本化管理



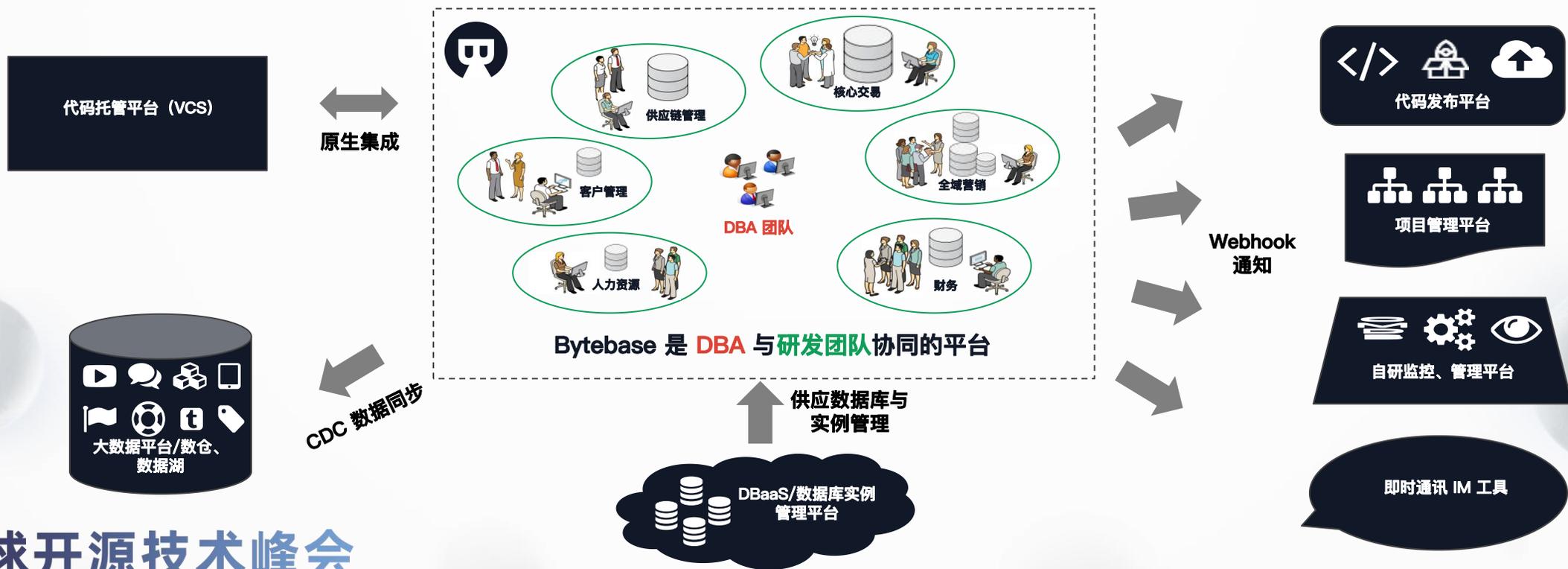
Bytebase 愿景是打造一款开发者友好的 Database CI/CD 工具

代码变更 -> GitLab 

基础设施变更 -> Terraform 

数据库变更 -> ???

数据库变更 (Database Change) 需要 **Bytebase** —— 面向研发组织设计, **聚焦业务数据库开发管理**, 让数据库变更与版本管理 **更安全** 的 DevOps 工具



全球开源技术峰会

THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE

企业可依托 Bytebase 实现安全高效的数据库协同开发 workflow



Bytebase 提供了一套完整的企业级解决方案，提供了全流程



1 查询访问控制

构建安全易用的 SQL 协同开发平台

2 变更审核

提供专业、高效、可解释的开发规范审核能力

3 变更发布

支持手动、自动等多种变更 workflow

4 失败回滚

提供基于时间点的恢复能力，一键还原错误变更

5 变更版本管理

自动创建并管理数据库 Schema 版本资料库

已支持数据库:



PostgreSQL



redis



MongoDB



snowflake



ClickHouse



全球开源技术峰会

THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE

唯一入选 CNCF 目录，2022 全球开源初创企业增速 TOP 50



官网 – www.bytebase.com

GitHub – <https://github.com/bytebase/bytebase>

Twitter – [@Bytebase](https://twitter.com/Bytebase)

LinkedIn – <https://www.linkedin.com/in/bytebase/>

微信公众号



用户群



全球开源技术峰会

THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE

THANKS