

# Vitess

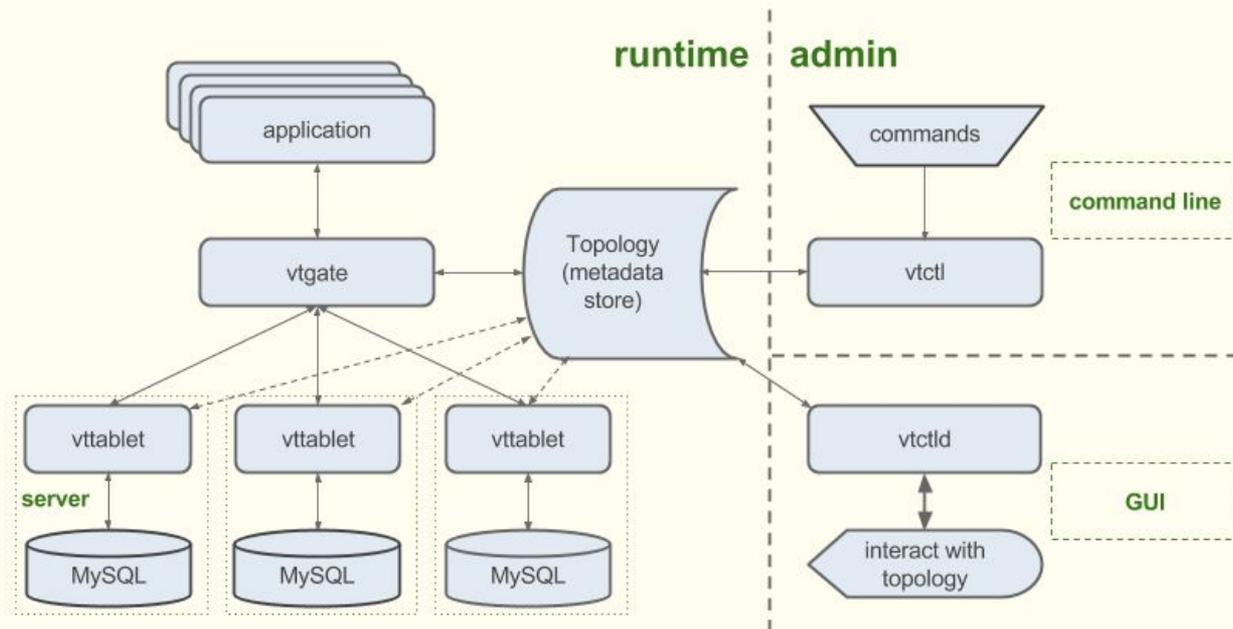
<https://vitess.io>

# Что это такое?

Система для горизонтального масштабирования MySQL

Обеспечивает read-tolerance и переключение мастер-баз в случае выхода мастера из строя

# Архитектура



# Что умеет(из коробки)

- горизонтальное масштабирование(шардирование)
- пуллинг соединений
- кеширование запросов(удаление дублей)
- 2хфазный коммит(с недавнего времени)
- топология(при помощи consul, etcd, ZooKeeper)
- ACL на запросы!

# Что не умеет

- last\_inserted\_id на шардированном кейспейсе: **SELECT last\_inserted\_id() FROM user;**
- distinct & функции агрегирования: **SELECT DISTINCT a, COUNT(\*) FROM user**
- если используется группировка по полю, то поле должно обязательно быть написано в SELECT: **SELECT a FROM user GROUP BY b;**
- удаление по шардам с лимитом: **DELETE FROM user LIMIT 10;**
- Полный список:  
[https://github.com/vitessio/vitess/blob/master/go/vt/vtgate/planbuilder/testdata/unsupported\\_cases.txt](https://github.com/vitessio/vitess/blob/master/go/vt/vtgate/planbuilder/testdata/unsupported_cases.txt)

# Cell

Cell — группа серверов и сетевой инфраструктуры, изолированной от аварий в других cell(датацентр)

Обычно, это или ДЦ или часть ДЦ, иногда называется “зона” или “зона доступности”.

# Keyspace

Keyspace — логическая база данных

Если используется шардирование, то кейспейс указывает на несколько MySQL баз данных, иначе на одну базу

Чтение данных из кейспейса представляет собой чтение из MySQL

# Keyspace graph

Keyspace graph позволяет Vitess решать, какой шард использовать для конкретного запроса

# Shard

Shard — логическая часть внутри кейспейса. Обычно содержит один мастер и много MySQL слейвов

В конфигурации имеет следующий вид

SHARD=-80, где -80 — это ключи [0x0, 0x80)

Или же SHARD=80-, где 80- — это ключи [0x80, 0xff]

# Tablet

Таблетка — это комбинация процесса `mysql` и `vtablet`, запущенных на одной и той же машине.

# Состояния Tablet

**master** — мастер-база

**replica** — слейв, который может быть продвинут до мастера

**rdonly** — слейв, который не может быть продвинут до мастера. Обычно, используется в качестве обработки background задач, дампа данных, тяжелых запросов

**backup** — база с остановленной репликацией

**restore** — база, в которую наливаются данные из бекапа.

**drained** — разрезервированная база для background задач Vitess'a

# Topology Service

Сервера, которые хранят данные о топологии и предоставляют распределенный сервис блокировок

Например, etcd, ZooKeeper или Consul

- Роутит запросы
- Координирует таблетки между собой
- Хранит конфигурацию серверов

# VSchema

Описание того, как данные лежат внутри кейспейсов и шардов.

Чаще всего описывается JSONом, например

```
{ "sharded": true,  
  "tables": {  
    "dinosaurs": {  
      "column_vindexes": [  
        {"column": "id", "name": "hash"}]}}
```

# Виды индексов в VSchema

- hash — 3DES null-key hash
- consistent\_lookup\_unique — хеш по уникальным значениям в таблице(primary key)
- unicode\_loose\_md5 — md5 по ключу в таблице

<https://vitess.io/docs/reference/vschema/> — полный список

# vtgate

Прокси-сервер между пользователем и vttablet.

Общается с клиентом по MySQL протоколу

Общается с vttablet по gRPC

# vtctl, vtctld

**vtctl** — CLI для администрирования vitess-кластера

# vtworker

Занимается долгими задачами, например, рещардингом