



Integrating Best of Breed Open Source Tools to Vitess: Orchestrator

Alkin Tezuysal
PlanetScale

Perconalive, May 2021

About me

Sr. Technical Manager at **PlanetScale**

Maintainer for **Vitess**

Open source database evangelist previously at **Percona**,
Pythian and others with Enterprise Background

Born to Sail, Forced to Work!



 [@ask_dba](https://twitter.com/ask_dba)





Founded Feb. 2018 by co-creators of Vite

~50 employees

HQ Mountain View, 100% remote team

Vitess

A database clustering system for horizontal scaling of MySQL / MariaDB



- CNCF graduated project
- Open source, Apache 2.0 licence
- Contributors from around the community
- Written in Golang



Agenda

Vitess architecture overview

Vitess use cases and sharding

Vitess Open Source Tools

- XtraBackup - Percona
- pt-online-schema change - Percona
- gh-ost - Github
- Orchestrator - Shlomi Noach
- And more ...





Vitess architecture basics

How the Vitess architecture enables transparent database infrastructure operations



@ask_dba

Glossary



- Keyspace: Logical database (sharded)
 - Keyspace ID
 - Primary Vindex
 - Vindex
- VTGate: Proxy server
- VTablet : Backend server
- Topology : Configuration server (etcd, zookeeper)



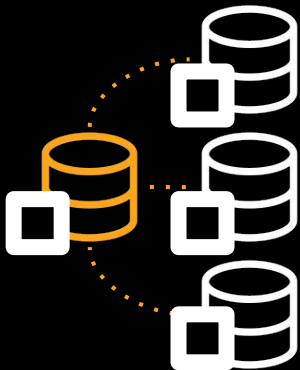


ViteSS architecture basics

Consider a common replication cluster



@ask_dba



ViteSS architecture basics

Each MySQL server is assigned a **vttablet**

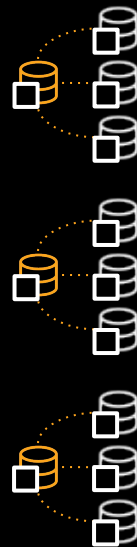
- A daemon/sidecar
- Controls the **mysqld** process
- Interacts with the **mysqld** server
- Typically on same host as **mysqld**



@ask_dba

Vitess architecture basics

In production you have multiple clusters

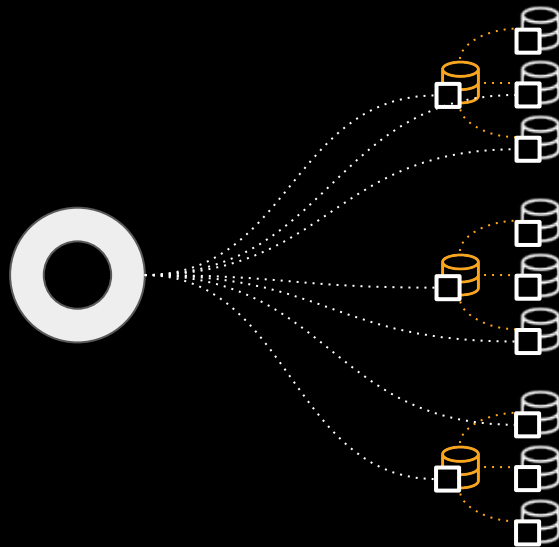


@ask_dba

Vitess architecture basics

User and application traffic is routed via **vtgate**

- A smart, stateless proxy
- Speaks the MySQL protocol
- Impersonates as a monolith MySQL server
- Relays queries to **vttablets**



@ask_dba

Vitess architecture basics

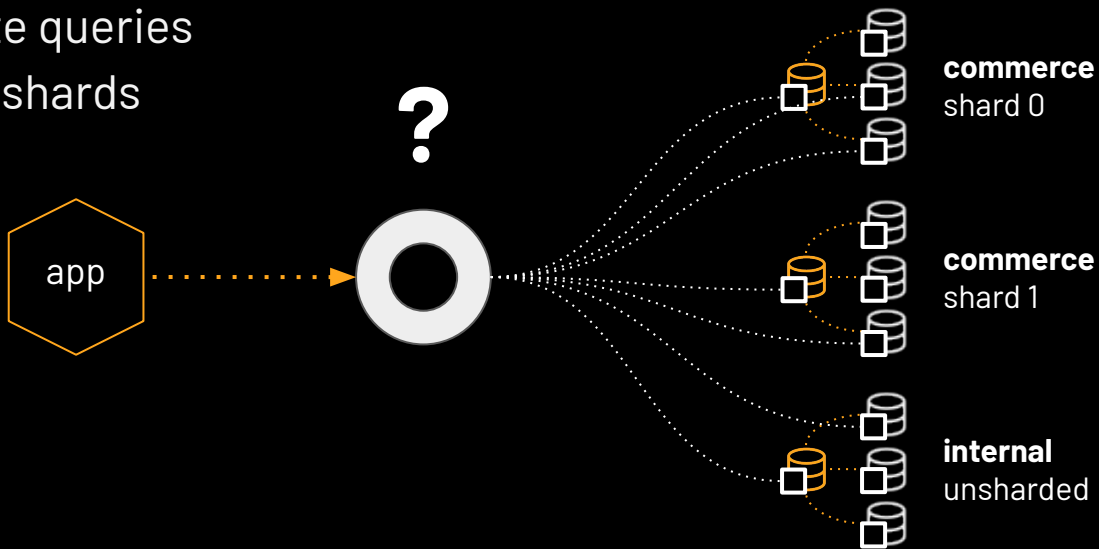
A vitess deployment will run multiple **vtgate** servers for scale out



@ask_dba

Vitess architecture basics

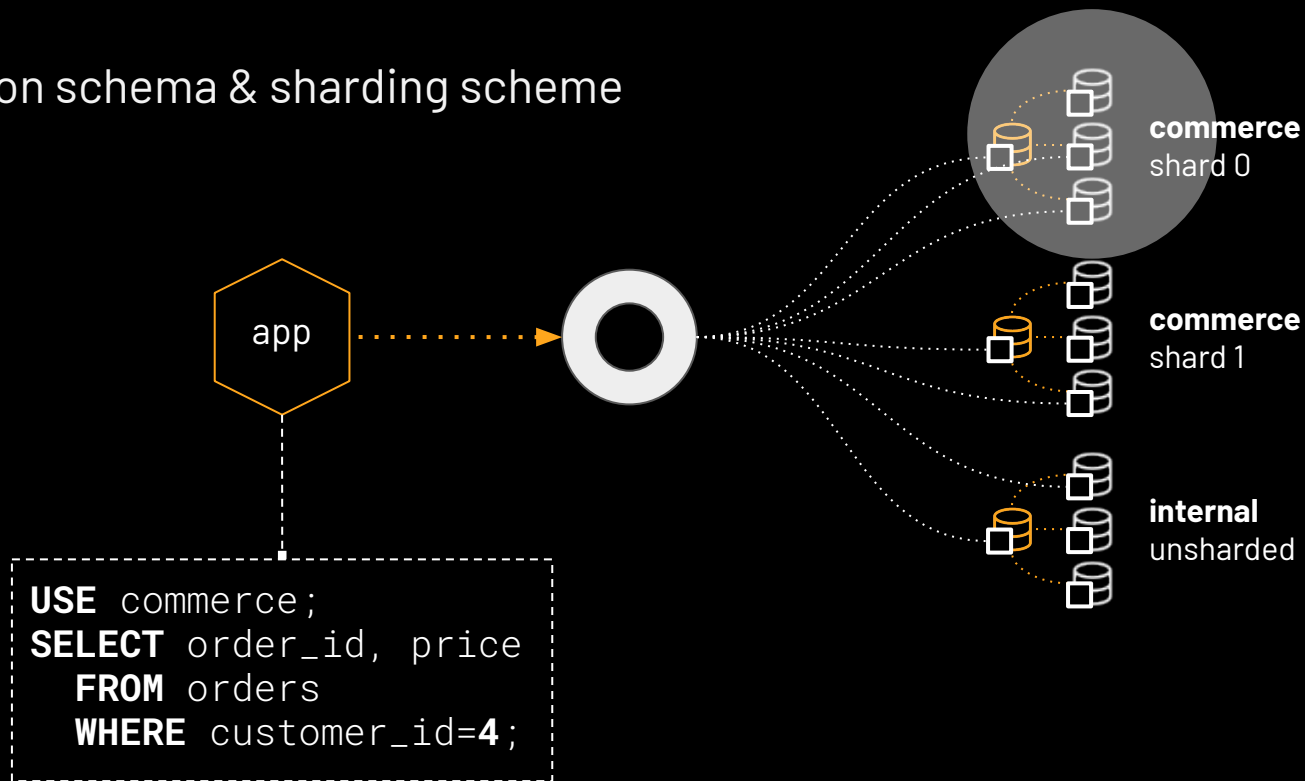
vtgate must transparently route queries to correct clusters, to relevant shards



@ask_dba

Vitess architecture basics

Queries route based on schema & sharding scheme

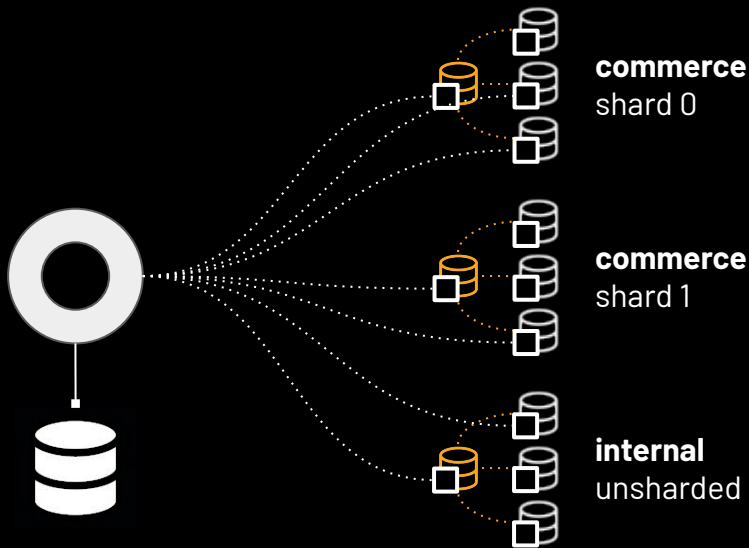


Vitess architecture basics

topo: distributed key/value store

- Stores the state of vitess: schemas, shards, sharding scheme, tablets, roles, etc.
- etcd/consul/zookeeper
- Small dataset, mostly cached by **vtgate**

vtgate

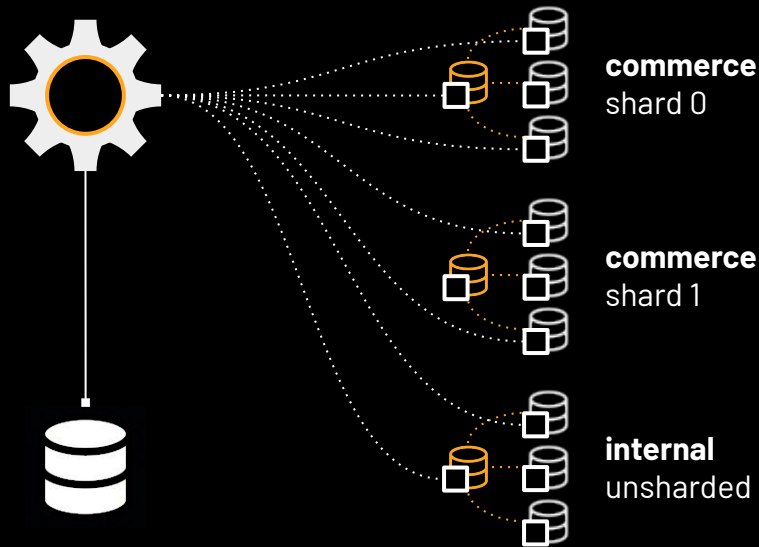


@ask_dba

Vitess architecture basics

vtctld: control daemon

- Runs ad hoc operations
- API server
- Reads/writes **topo**
- Uses locks
- Operates on tablets



@ask_dba



Vitess knows

Vitess keeps known schemas, shards, clusters, server roles, all in **topo**

It keeps a *state*



@ask_dba

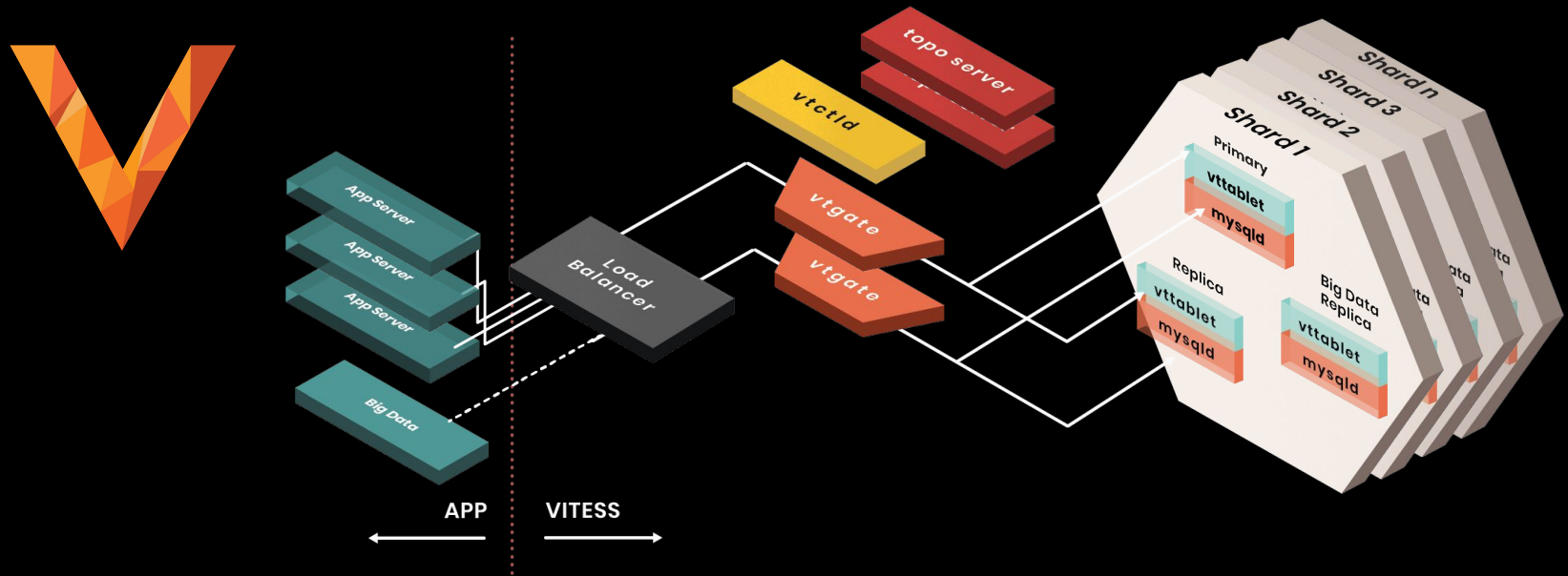


Vitess Controlplane Includes

- Proxy server (vtgate)
- Managed Backup and Recovery (XtraBackup)
- Integrated failover (a.k.a Orchestrator/vtorc)
- Sharding Schemes (Horizontal)
- Advanced Replication (Vreplication, Vstream)
- Online DDL (gh-ost, pt-osc)
- And more



Vitess architecture summary





Supported Backend Databases

- MySQL 5.7 / 8.0
- MariaDB 10.3
- ~~Postgresql~~



Vitess Use Cases and Sharding



- Part or entire application scaling
- Management of existing MySQL topology
- Sharding and resharding
- Minimizing backup/recovery scenarios

Vitess Open Source Tools



- XtraBackup - Percona ✓
- pt-online-schema change - Percona ✓
- gh-ost - Github ✓
- Orchestrator - Shlomi Noach → vtorc - in progress
- And more ...

XtraBackup

- For MySQL 5.7 and MySQL 8.0
- MariaDB 10.3 is not compatible with XtraBackup.



```
$ vtctl Backup <tablet-alias>
```

```
$ vtctl BackupShard [-allow_master=false]  
<keyspace/shard>
```

pt-online-schema



- Supported since version 9.0.0 GA
- Automatic handling of flags
- Uses @@ddl_strategy session variable

```
mysql> set @@ddl_strategy='pt-osc --max-load  
Threads_running=200';
```

```
mysql> ALTER TABLE user MODIFY user_id BIGINT
```

```
$ vtctlclient ApplySchema -ddl_strategy "pt-osc  
--max-load Threads_running=200" -sql "ALTER TABLE  
user MODIFY user_id INT " commerce
```

gh-ost



- Supported since version 9.0.0 GA
- Automatic handling of flags
- Uses @@ddl_strategy session variable

```
mysql> set @@ddl_strategy='gh-ost --max-load  
Threads_running=200';
```

```
mysql> ALTER TABLE user MODIFY user_id BIGINT
```

```
$ vtctlclient ApplySchema -ddl_strategy "gh-ost  
--max-load Threads_running=200" -sql "ALTER TABLE  
user MODIFY user_id INT " commerce
```

Choosing a DDL strategy

- Online / VReplication
- gh-ost or pt-osc



Read more on Vitess [Documentation](#).

Orchestrator - Vtorc

- Experimental
- Development in progress, help needed.



Orchestrator Highlights

- Developed by Shlomi Noach , several contributors
- MySQL Replication Topology Management and High Availability solution
- Provides HTTP API & Web Interface
- Supports Discovery, Refactoring, Recovery with pre/post hooks



DEMO and Q/A



@ask_dba

Resources

Docs: vitess.io/docs/

Code: github.com/vitessio/vitess

Slack: vitess.slack.com



@ask_dba



Thank you!

Questions?

github.com/askdba

[@ask_dba](https://twitter.com/ask_dba)



@ask_dba