How to shard MariaDB like Pro?

Alkin Tezuysal
PlanetScale

FOSDEM Feb 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 Vitess

~45 employees

HQ Mountain View, 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
Agenda

Vitess architecture overview

Vitess use cases and sharding

Vitess meets MariaDB 10.3

- Local Docker Install
- Kubernetes Operator
- Build
Vitess architecture basics

How the Vitess architecture enables transparent database infrastructure operations
Glossary

- **Keyspace**: Logical database (sharded)
  - Keyspace ID
  - Primary Vindex
  - Vindex
- **VTGate**: Proxy server
- **VTTablet**: Backend server
- **Topology**: Configuration server (etcd, zookeeper)
Vitess architecture basics

Consider a common replication cluster
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`
Vitess architecture basics

In production you have multiple clusters
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
Vitess architecture basics

A vitess deployment will run multiple vtgate servers for scale out
Vitess architecture basics

vtgate must transparently route queries to correct clusters, to relevant shards
Vitess architecture basics

Queries route based on schema & sharding scheme

```
USE commerce;
SELECT order_id, price
FROM orders
WHERE customer_id=4;
```
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
Vitess architecture basics

vtctl: control daemon
- Runs ad hoc operations
- API server
- Reads/writes topo
- Uses locks
- Operates on tablets
Vitess knows

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

It keeps a state
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
MariaDB Compatibility 10.3

- No extensive work has been done
- Looking for contributors and users
- Feedback is valuable to enhance project
- 10.4 compatibility is pending #issue
Resources

Docs: vitess.io/docs/

Code: github.com/vitessio/vitess

Slack: vitess.slack.com
DEMO and Q/A
Thank you!

Questions?

github.com/askdba
@ask_dba
@ask_dba