Vitess Demystified

Navigating the world of Distributed Databases

Manan Gupta

PlanetScale

Vitess

Manan Gupta @GuptaManan100

Maintainer, Vitess @vitessio

Software Engineer, PlanetScale @planetscaledata

GitHub @GuptaManan100

Vitess

What Modern Applications Need

Performance

Scalability

Uptime / High Availability

No data loss

NewSQL

- Bridge gap between SQL and NoSQL
- Provide structure like SQL
- Provide Scalability like NoSQL
- Vitess, CockroachDB, TiDB and many others

What is Vitess?

Cloud Native Database

Massively Scalable

Highly Available

MySQL 5.7/8.0 Compatible

Vitess is also

CNCF graduated

Apache v2.0 license

Community supported

Written in golang

Vitess

Vitess serves millions of QPS in production

Key Adopters

Slack
100% on
Vitess

Square Cash
100% on Vitess

JD.com
10000+
databases

PlanetScale
Database
Service

Needs to work with

- Database Frameworks
- ORMs
- Legacy code
- Third Party Applications

Vitess Architecture basics

How does the Vitess architecture enable transparent database operations?

Concepts

Keyspace: Logical database

Shard: Segment of a keyspace

Sharding Scheme: Which row goes where

Commerce

Customer_id	*Email*
101	alice@domain.com
102	bob@domain.com
103	charlie@domain.com

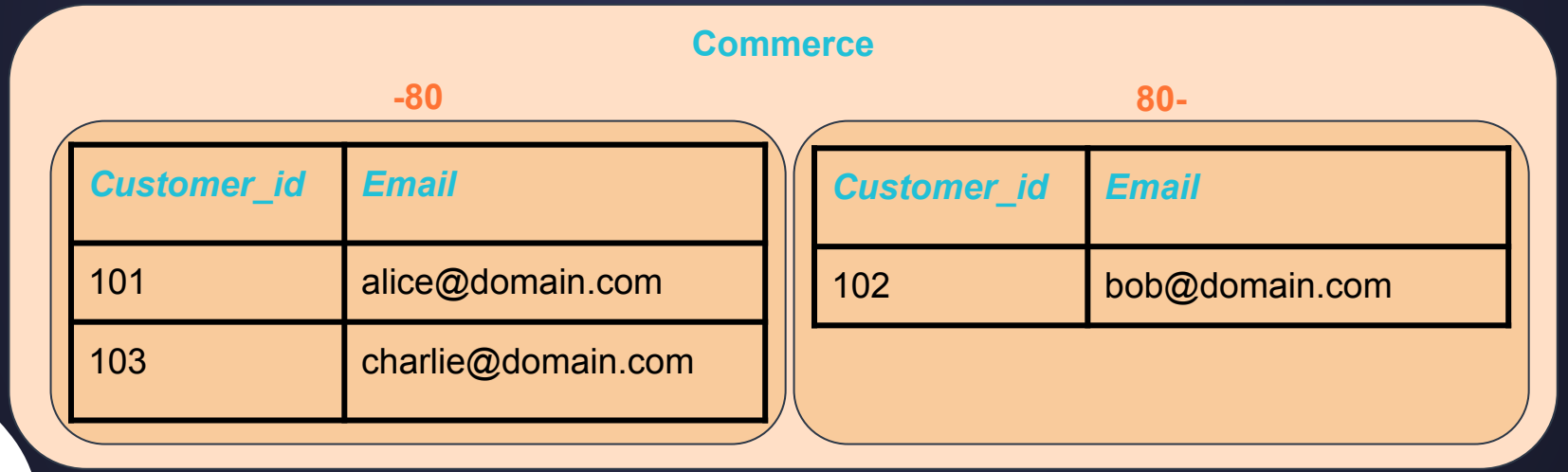
Vitess

Concepts

Keyspace: Logical database

Shard: Segment of a keyspace

Sharding Scheme: Which row goes where



Vitess

Vitess Architecture basics

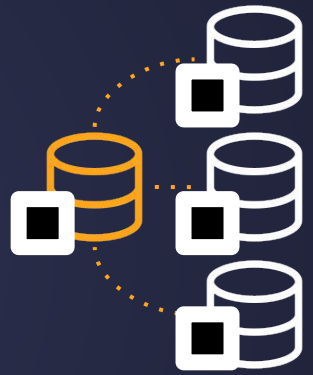
A common replicated database cluster with primary and replicas

Vitess

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

Vitess Architecture basics

In production you have multiple clusters

Vitess

Vitess Architecture basics

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

- A smart, stateless proxy
- Speaks the MySQL protocol
- Impersonates a monolithic MySQL server
- Relays queries to **vttablet**s

Vitess

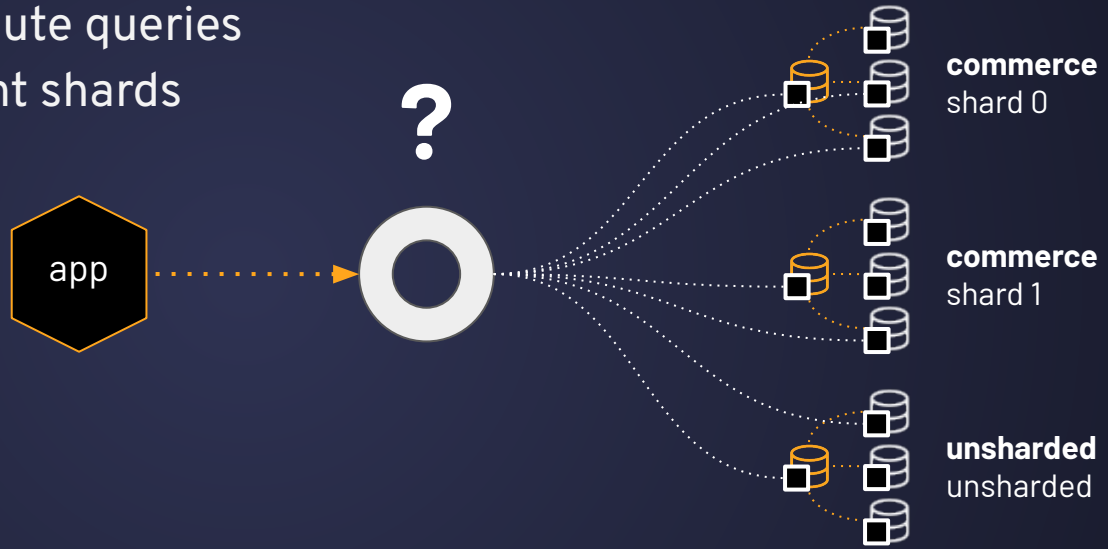
Vitess Architecture basics

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

Vitess

Vitess Architecture basics

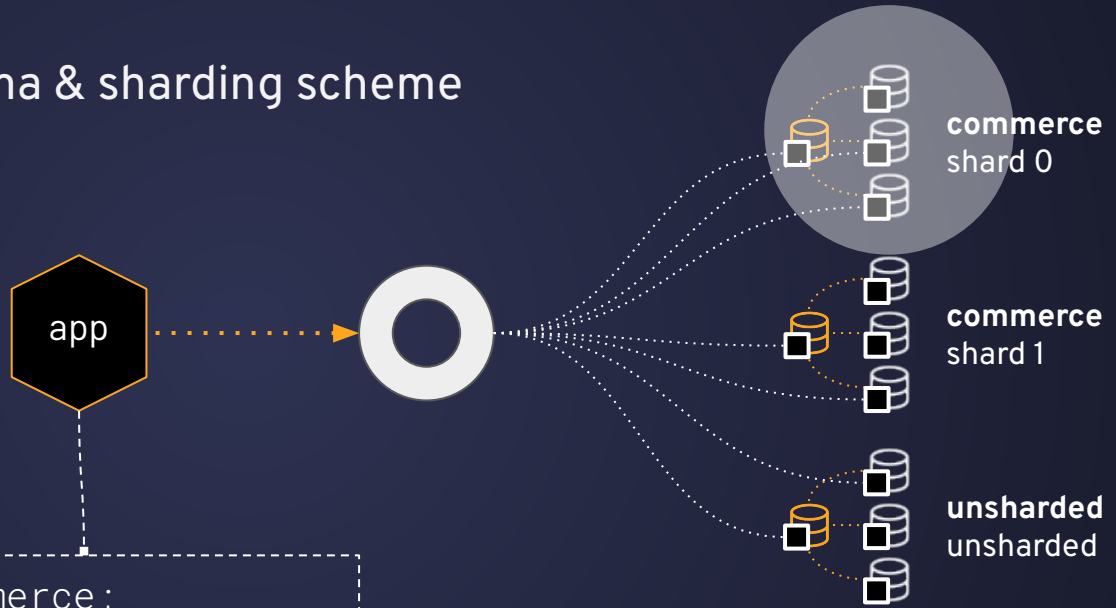
vtgate must transparently route queries
to correct clusters, to relevant shards



Vitess

Vitess Architecture basics

Queries route based on schema & sharding scheme



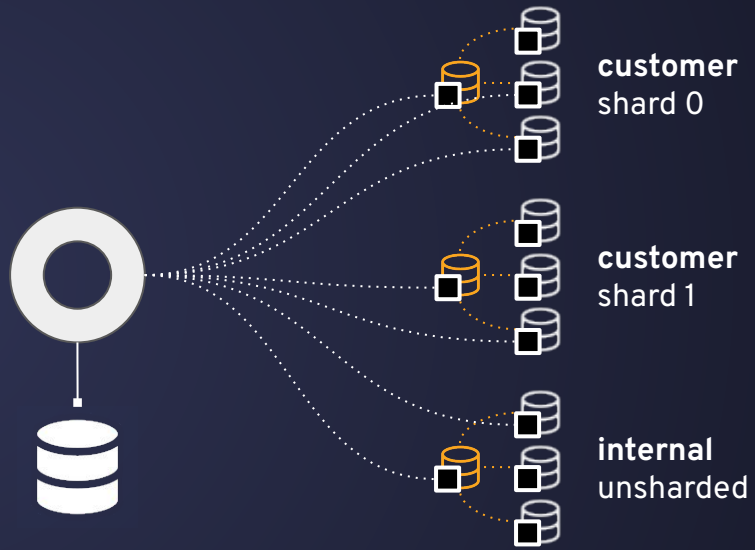

```
USE commerce;
SELECT customer_email,
customer_id
  FROM customer
  WHERE customer_id=4;
```

Vitess

Vitess Architecture basics

topo: distributed key/value store

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

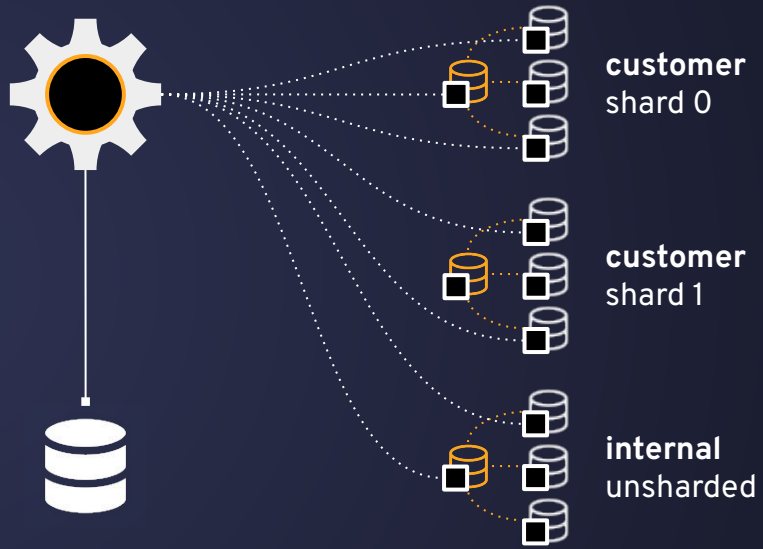


Vitess

Vitess Architecture basics

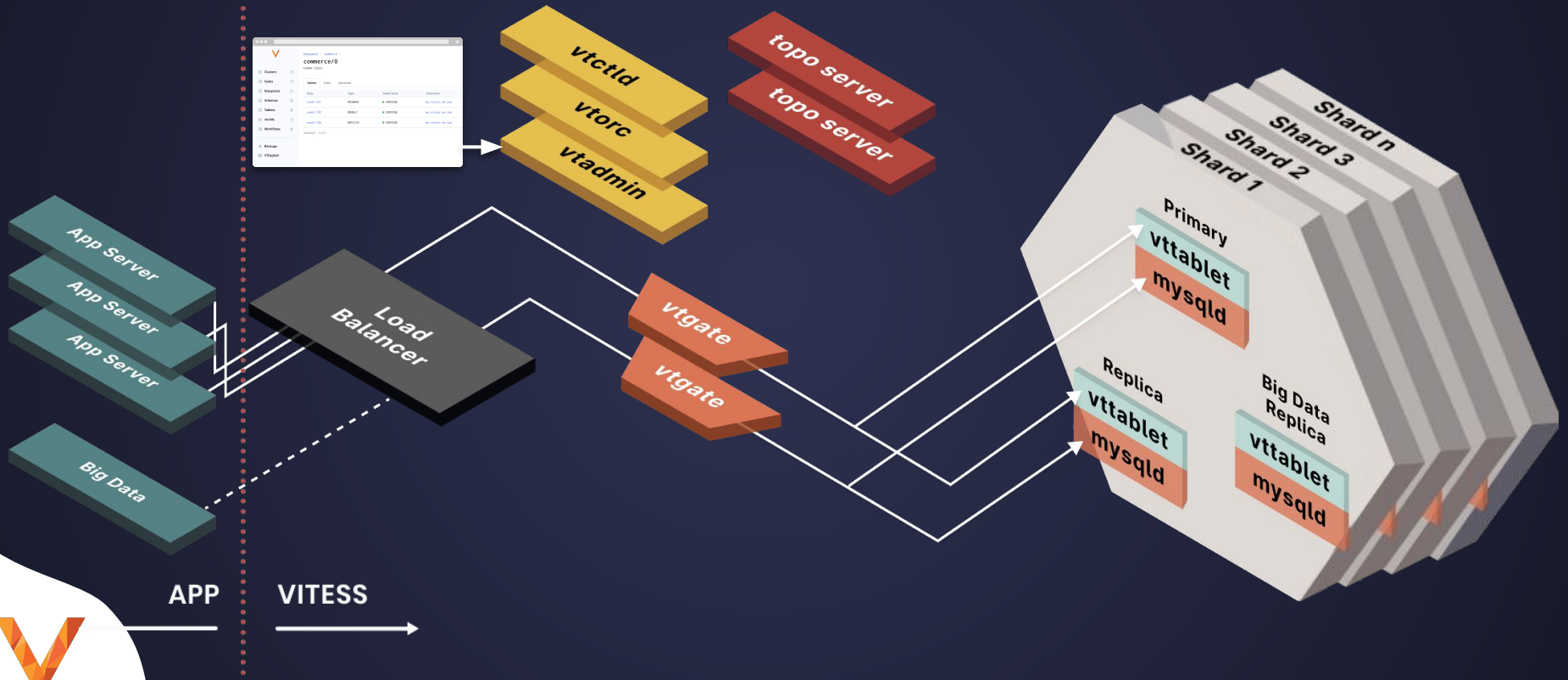
vtctld: control daemon

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



Vitess

Vitess Architecture summary



Vitess

Demo

Query Serving In Vitess

Features

MySQL compatibility

- Compatible with popular frameworks

Management features

- Connection Pooling
- Online schema changes
- Query consolidation

Features

Scalability through Sharding

High Availability

- Failure detection and failover

Materialized Views

Change data capture / notification

Data migrations

Sequence Tables

Resources

Join vitess slack

https://vitess.slack.com

Try one of our tutorials

https://vitess.io/docs/18.0/get-started/

Source code

https://github.com/vitessio/vitess

Resources

Blog on Query Planning

https://planetscale.com/blog/what-is-a-query-planner

Blog on Cluster Management

https://vitess.io/blog/2022-09-21-vtorc-vitess-native-orchestrator/

Questions?

@GuptaManan100