



Deepthi Sigireddi, Matt Lord, Rohit Nayak



Deepthi Sigireddi



Technical Lead

@ATechGirl

Matt Lord



Maintainer

@mattalord

Rohit Nayak



Maintainer

@rohitnayak

Vitess Overview

What is Vitess?

Cloud
Native
Database

Massively
Scalable

Highly
Available

MySQL
5.7/8.0
Compatible

Works with

Database
Frameworks

ORMs

Legacy
Code

Third Party
Applications

Logical database (Keyspace)

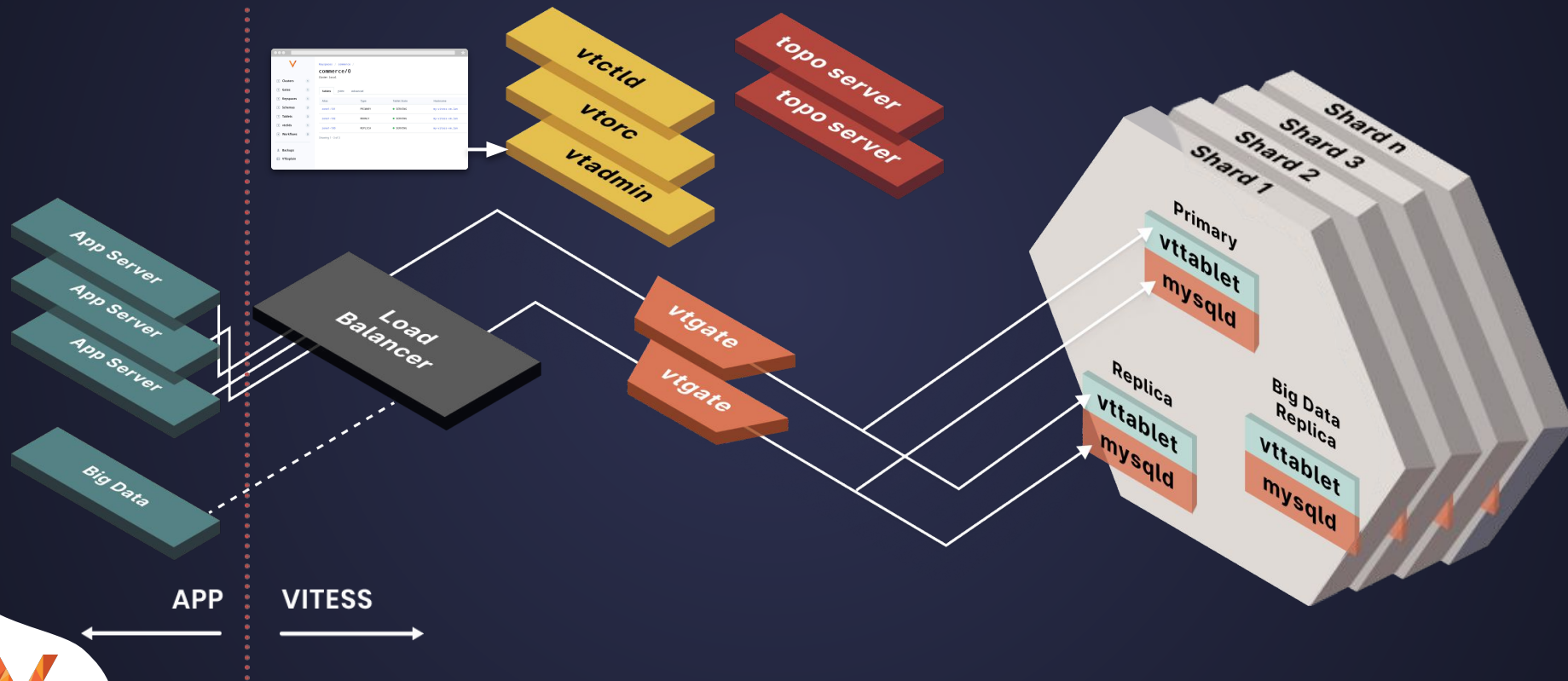
Many
physical
databases

Query
routing

gRPC /
MySQL
clients

Single
connection

Architecture



Vitess in production



Key Adopters

Slack
100% on
Vitess

GitHub

JD.com
10000+
databases

PlanetScale
Database
Service



Jeremy Tinley

@techwolf359

...

I don't think I would ever consider running MySQL without [@vitessio](#) at this point. You'll have a higher curve to get started, but the tradeoff should be significantly less toil.

5:23 PM · Aug 1, 2022 · Twitter Web App



Derek Perkins

@Derek_Perkins

...

Replying to [@techwolf359](#) and [@vitessio](#)

Even if you're only running a single shard and don't foresee needing to shard in the near future, the control plane for backups, schema migrations, and replica management are more than worth the extra learning curve

6:37 PM · Aug 1, 2022 · Twitter Web App



@vitessio

Community

21
maintainers

429
contributors,
214 PR authors

63
companies

PR authors
from 30
companies

VReplication Workflows

VReplication

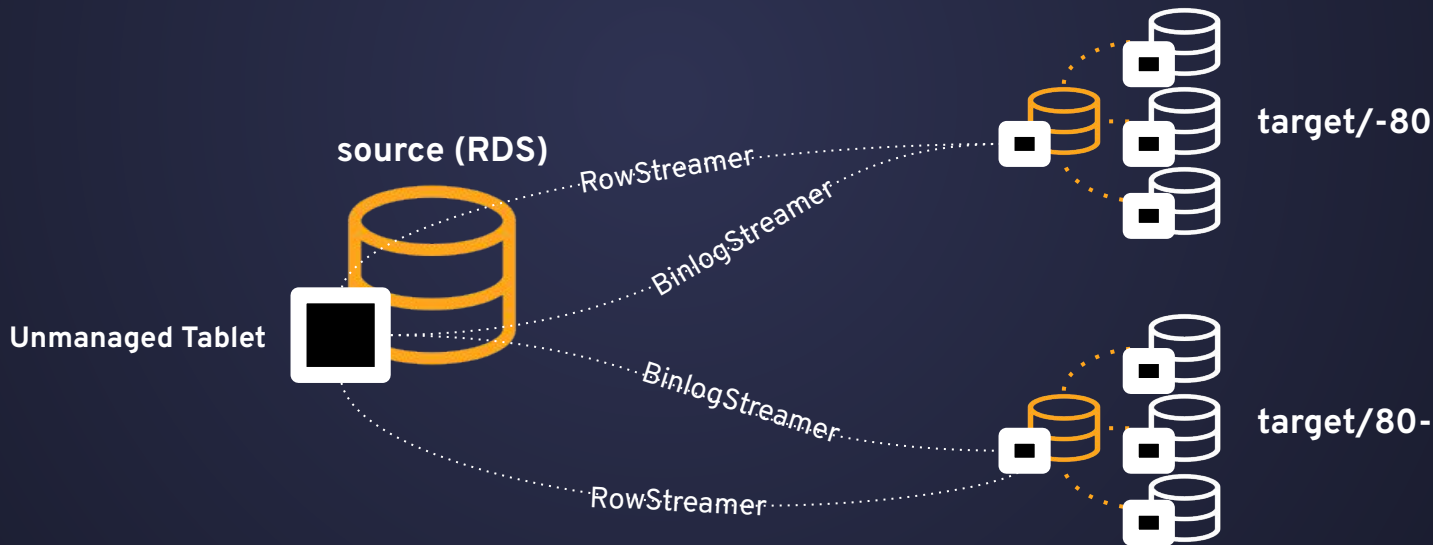
A Framework For Creating And Managing Data Streams And Workflows

When data matches a defined criteria then execute a defined workflow:

- Migrating into Vitess
- Sharding
- Online Schema Changes
- Realtime Materialization
- Event Streams / CDC
- ...

Migrating into Vitess

- Set up the Vitess cluster
- Set up a tablet for the external MySQL instance
- Add this *unmanaged* tablet to Vitess
- Use **MoveTables** to move [and shard] the data into Vitess



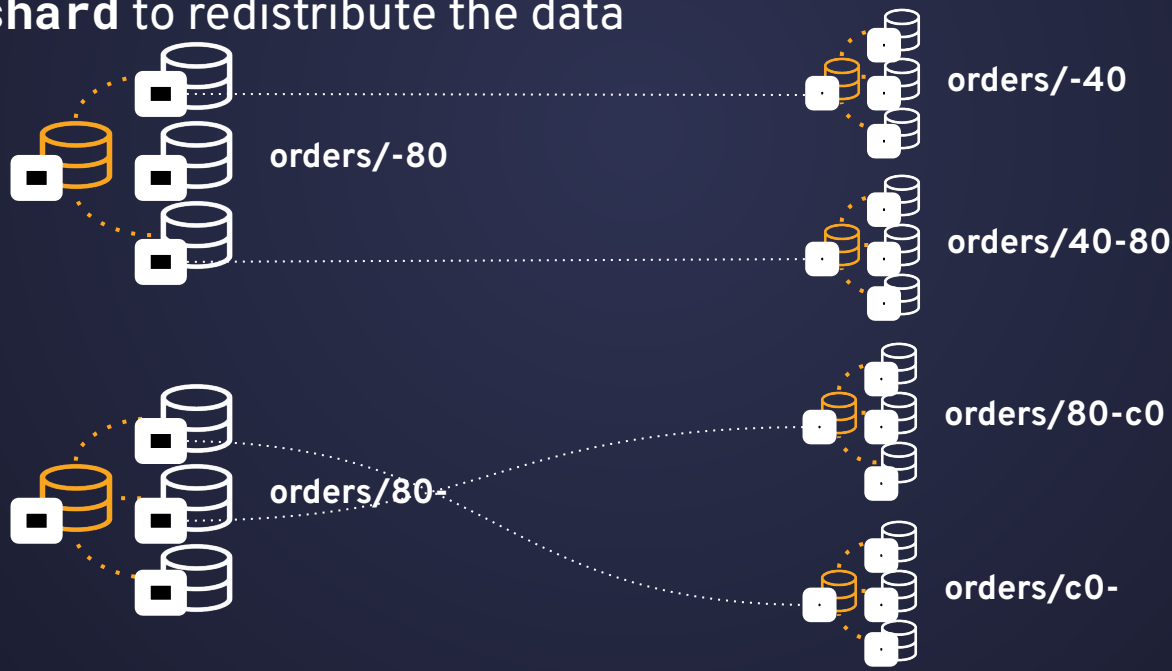
Vertical / Functional Sharding

- You have many/all tables in a single keyspace (logical database)
- You want to move some of the tables to a new keyspace
- You want to achieve this without making significant changes to your application or incurring downtime
- Use **MoveTables** to split the tables into N keyspaces



Horizontal Sharding

- Going from 1 to 2 shards, 2 to 4, 4 to 8, to ... as your dataset and usage grows
 - Can also shrink by merging shards and/or keyspaces when needed
- Add new shards to manage new keyspace range splits
- Use **Reshard** to redistribute the data



VDiff (v2)

- Runs a diff between the source and target shards
- Asynchronous
- Progress reporting and ETA
- Auto resume on error
- Resilient to failures and topology changes

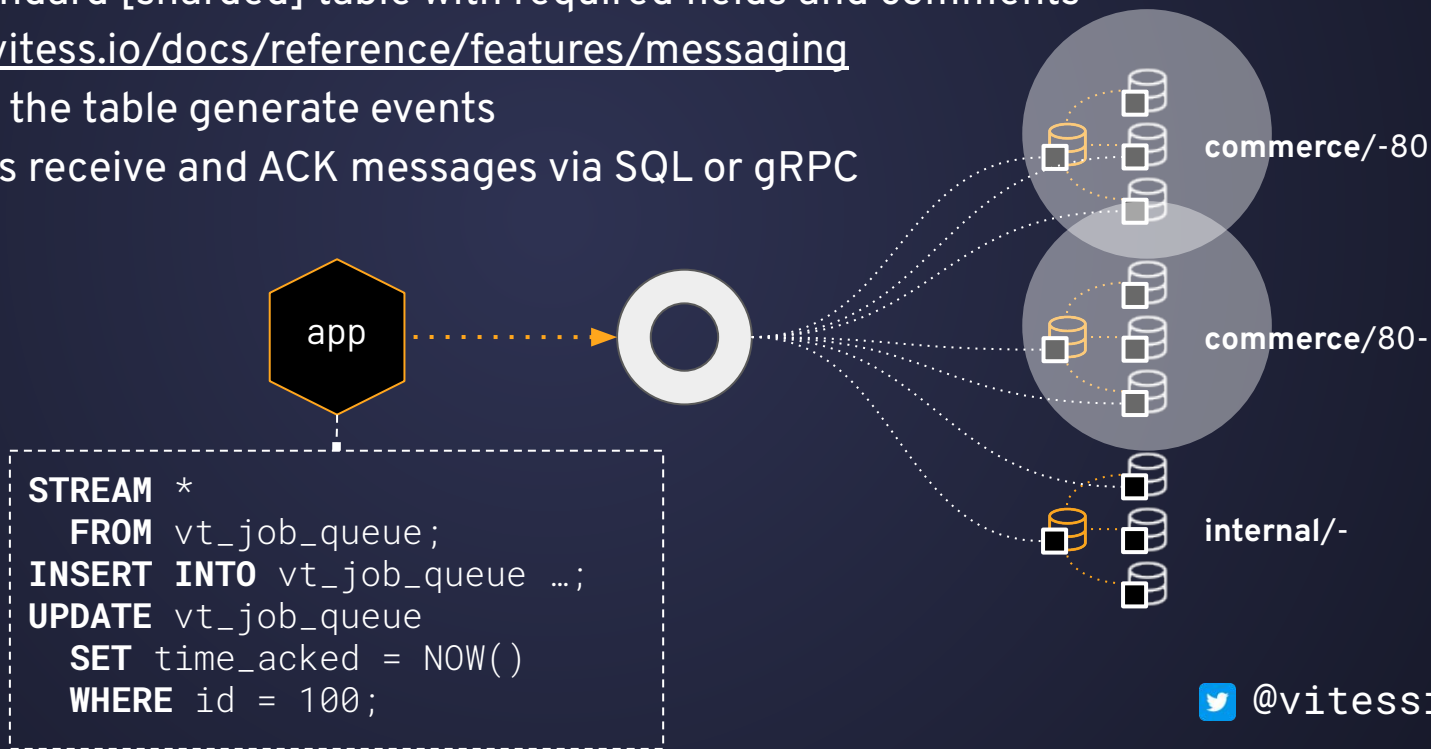
```
$ vtctlclient VDiff -- --v2 customer.commerce2customer --format=json show last
{
  "Workflow": "commerce2customer",
  "Keyspace": "customer",
  "State": "started",
  "UUID": "1922fd46-48fd-11ed-96c4-920702940ee0",
  "RowsCompared": 3370000,
  "HasMismatch": false,
  "Shards": "-80,80-",
  "StartedAt": "2022-10-11 00:39:00",
  "Progress": {
    "Percentage": 16.99,
    "ETA": "2022-10-11 00:39:49"
  }
}
```

Materialized Views

- Real-time views into a [transformed] [subset] of data
- Persisted as a table, could be sharded
- This data is automatically kept correct and up-to-date
- Use **Materialize** to setup the materialized view
- Some applications
 - Aggregations for analytics
 - Denormalized tables
 - Anonymization of PII data
 - Local copies of a reference tables

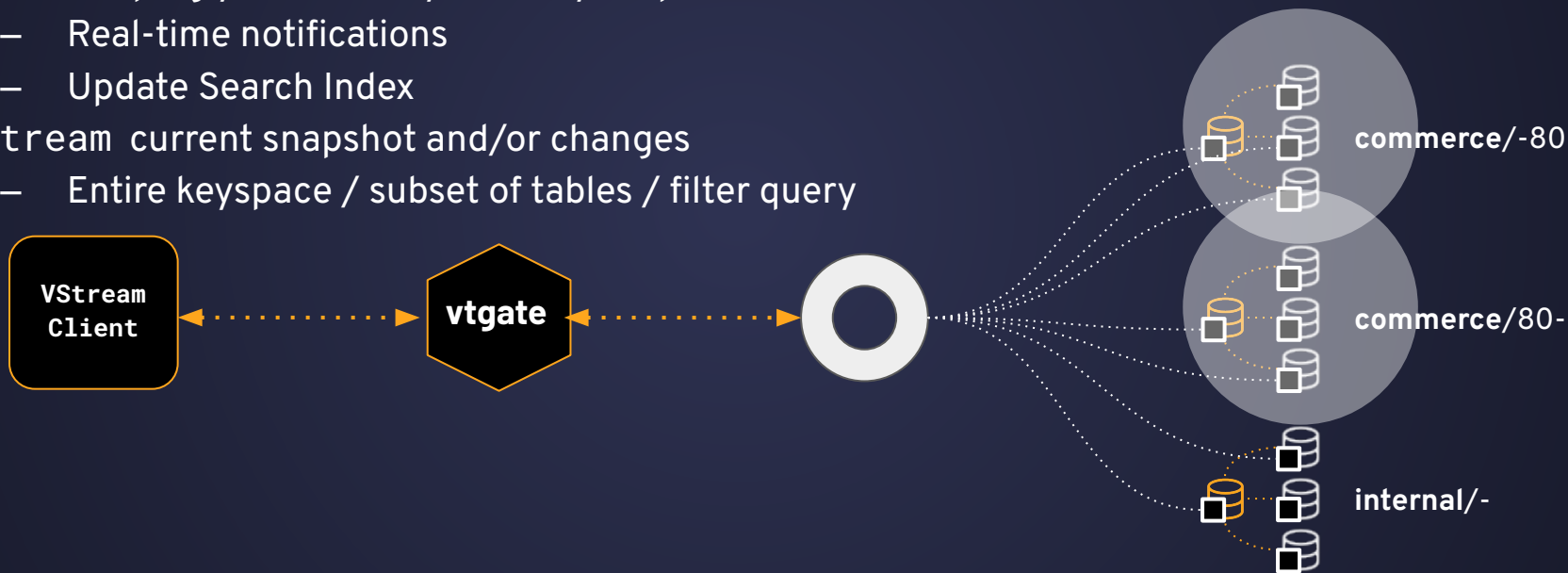
Message Bus / Job Queue

- Use a Vitess managed message bus or job queue
 - For example, managing “offline” data processing pipelines
- CREATE a standard [sharded] table with required fields and comments
 - <https://vitess.io/docs/reference/features/messaging>
- DMLs against the table generate events
- SUBSCRIBERS receive and ACK messages via SQL or gRPC



Change Data Capture

- VTGate VStream GRPC API
- Vitess managed change event stream
 - Keeping your DW in sync with your production DB
 - Real-time notifications
 - Update Search Index
- Stream current snapshot and/or changes
 - Entire keyspace / subset of tables / filter query



Any New Built-in Workflows...

- This will grow over time based on new use cases that present themselves
- Most recently, Vitess Native OnlineDDL
- Please let us know if you have feature suggestions/requests!



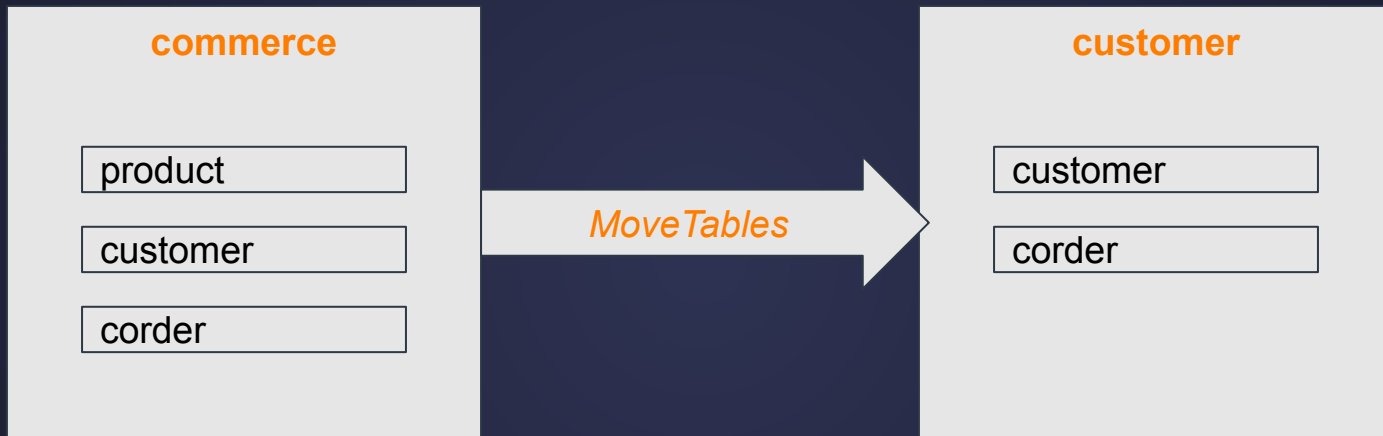
Safety Mechanisms

- VDiff
 - Compare the full set of logical rows on both sides using a consistent snapshot
- Limit impact on production traffic
 - configurable tablet throttling
 - `--tablet_types`, `--max_replication_lag`, `--filtered_replication_wait_time`, `--vreplication_copy_phase_max_innodb_history_list_length`, `--vreplication_copy_phase_max_mysql_replication_lag`, ...
- `--cells` to avoid cross datacenter traffic
- Custom `RoutingRules` to make cutovers seamless for apps/users
- Rollbacks via `$ vtctl ReverseTraffic`

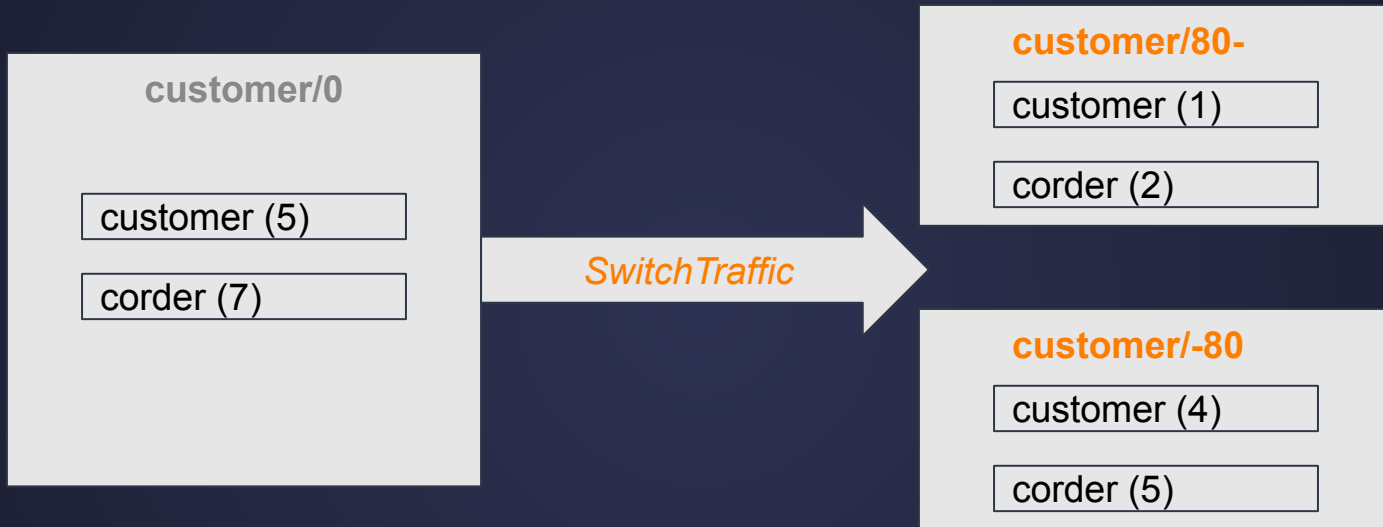
...

Demo

MoveTables Demo (Vertical Sharding)



Reshard Demo (Horizontal Sharding)



Materialize Demo

corder

sku	credit_card	created_at
SKU-1002	4026 1111 1111 1115	2022-10-24 11:14:24
SKU-1003	6011 3099 0000 1248	2021-03-11 15:35:08
SKU-1001	4111 1111 1111 1111	2020-02-20 08:30:01

Materialize

corder_facts

sku	credit_card	month	year
SKU-1002	xxxx xxxx xxxx 1115	10	2022
SKU-1001	xxxx xxxx xxxx 1111	2	2020

Materialize

sales

sku	num_orders	total_sales	total_qty
SKU-1001	1	100	1
SKU-1002	5	572	20

Demo Flow

- Create initial unsharded keyspace *commerce*
- **MoveTables** from *commerce* to unsharded *customer* (*Vertical Sharding*)
- **Reshard** from 1 to 2 shards (0=> -80/80-) (*Horizontal Sharding*)
- **Materialize**

- Denormalized view *corder_facts* of *corder* table

```
select order_id, customer_id, email, sku, CONCAT("xxxx-xxxx-xxxx-",  
RIGHT(credit_card,4)) as credit_card, price, qty, total(price, qty)  
total_price, month(created) as created_month, year(created) as  
created_year from corder where sku <> "SKU-1003" and price >= 2
```

- Aggregated *sales* view

```
select sku, count(*) as num_orders, sum(qty) as total_qty,  
sum(total_price) as total_sales from corder_facts group by sku
```

New Features

New Features

- **15.0 release GA this week**
- VTOrc
 - Automatic failure/problem detection and handling
 - Durability policies
- VTAdmin
 - Easy and intuitive cluster management web UI
- Backup/Restore compression algorithms
- New (Gen4) Query Planner / Optimizer GA
 - Increased MySQL SQL feature support and compatibility
- Schema Tracking GA
- Native View support (coming soon)

New Features

- VReplication
 - Shard level migrations
 - Migrate a keyspace to another Vitess cluster one shard at a time
 - Intelligent workflow throttling
 - Ensure SLA/SLO objectives continue to be met
 - Native OnlineDDL GA
 - No downtime, no impact on prod traffic, revertable
 - VDiff v2
 - Resumable (auto-resume on error)
 - Progress and ETA
- Incremental Backups and Point-in-Time Recovery (coming soon)

Q & A

Resources

- Website: vitess.io
 - [Docs](#)
 - [Tutorials](#)
 - [Source code](#)
- Slack: vitess.slack.com



<https://sched.co/182PH>

Thank you!