Scaling resilient systems: a journey into Slack's database service

Rafael Chacón
Guido Iaquinti

slack

SPEAKERS

Rafael Chaon

he/him/his

twitter.com/rafaelchacon

Staff Software Engineer - Slack

Guido Iaquinti

he/him/his

twitter.com/guidoiaquinti

guido.iaquinti.com

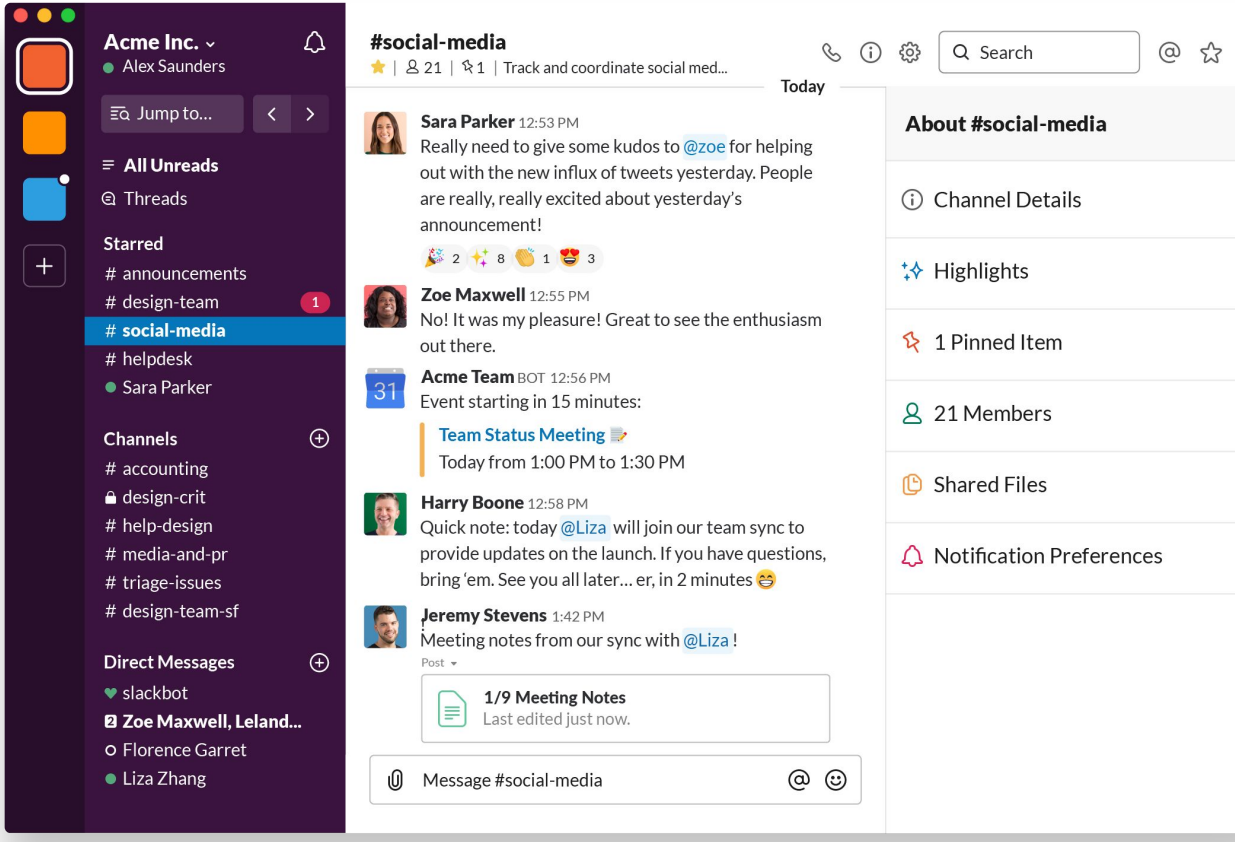
Site Reliability Engineer - Freelance

Agenda

1. Databases at Slack
2. Running databases in the cloud
3. Fault tolerance & Isolation
4. Key Lessons
5. Q&A

MISSION STATEMENT

Slack's mission is to make people's working lives **simpler**, more **pleasant**, and more **productive**.



Databases at Slack

Current status

In progress **migration** of our entire dataset to Vitess.

Two main types of clusters:

- Legacy shards
- Vitess shards

Why are we migrating?

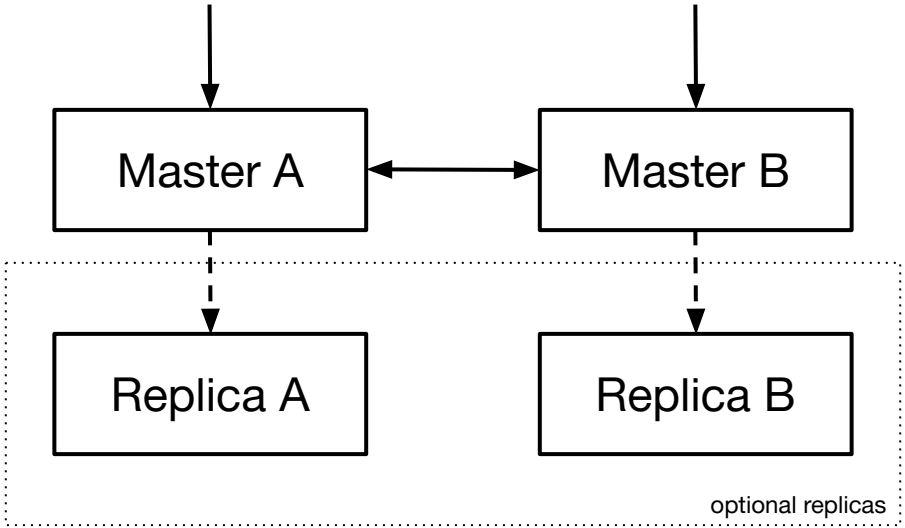
For more details please see the **presentations** on the side.

tl;dr; shard size limits, inefficient resource distribution, operational overhead, single sharding model

- *“Migrating to Vitess at (Slack) Scale” - Mike Demmer*
- *“Designing and launching the next-generation database system at Slack: from whiteboard to production” - Guido Iaquinti*
- *“Smooth scaling: Slack’s journey toward a new database” - Ameet Kotian*

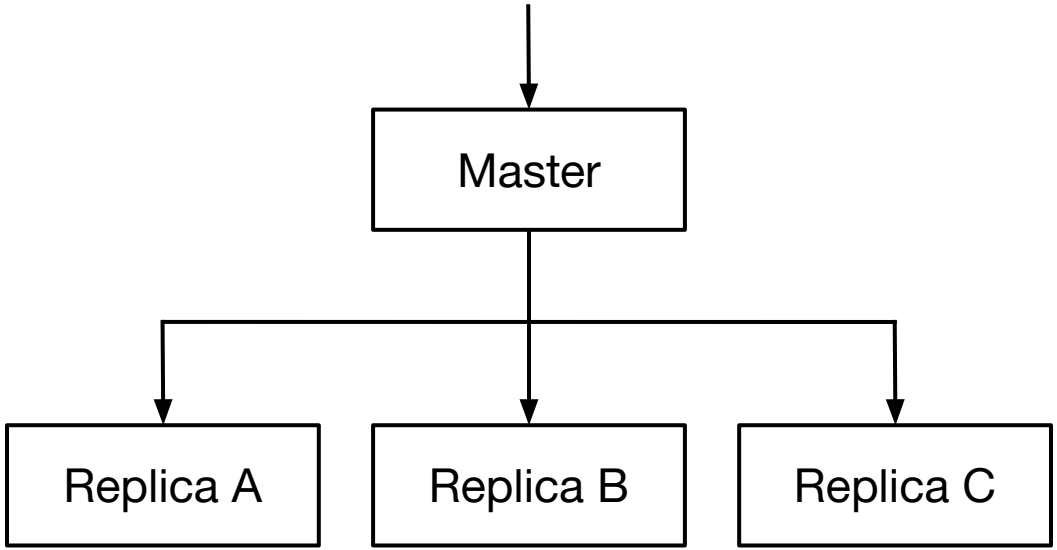
Legacy shards

Application level team-sharded **active master-master** MySQL setup.



Vitess shards

Master-replica
MySQL setup fully managed by Vitess.



Stats

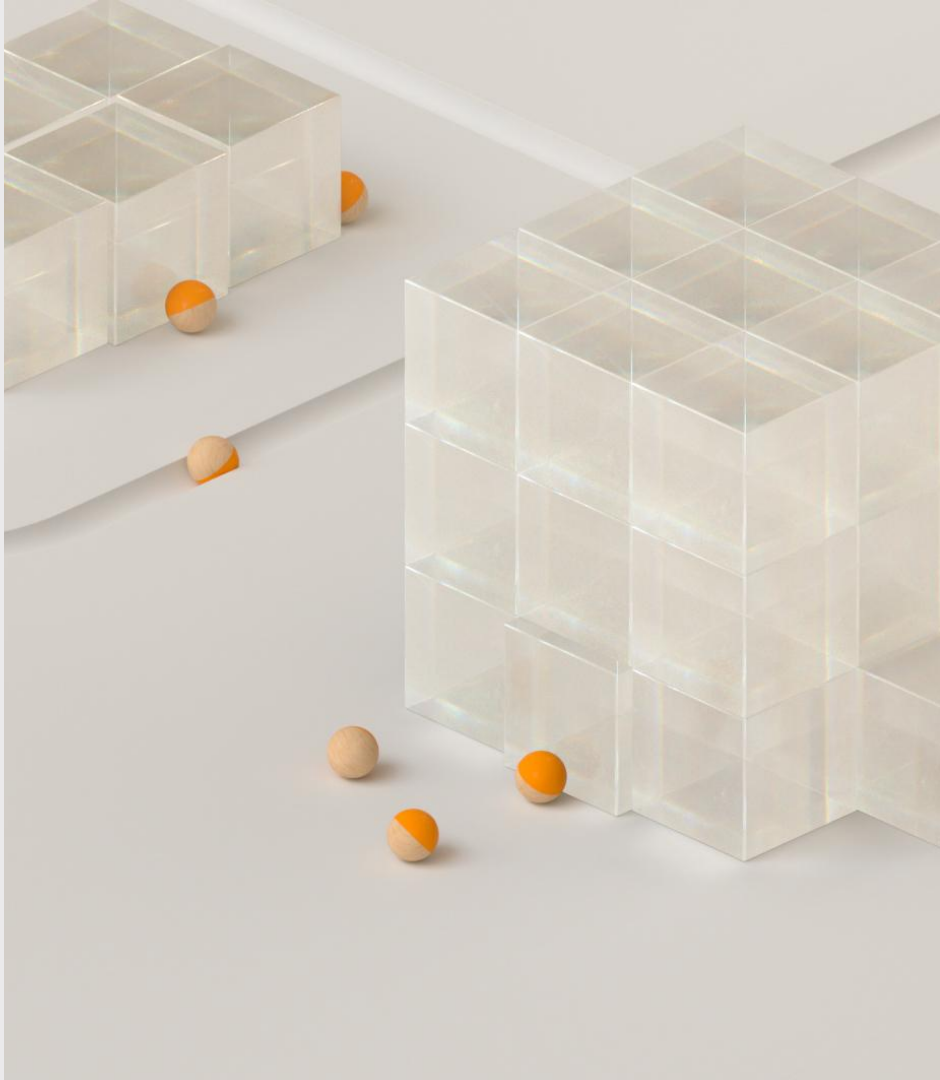
Queries per day: **53+ billion**

Storage provisioned: **7.5+ PB**

Served by legacy infrastructure: ~**60%**

Served by Vitess: ~**40%**

Target: **70%** served by Vitess by EOY



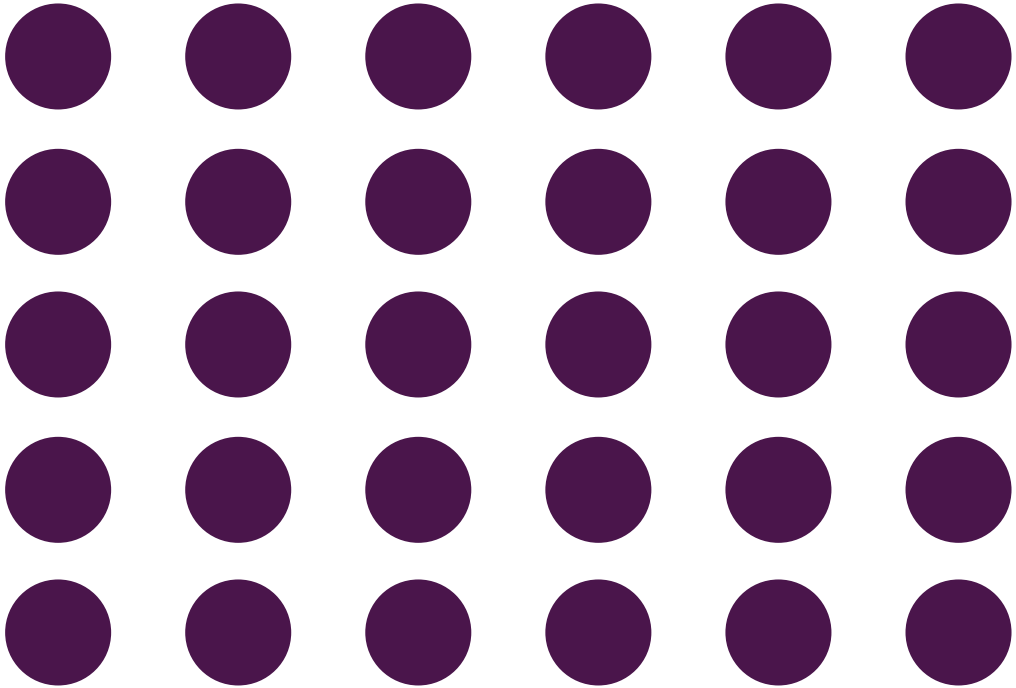
Running databases in the cloud

slack

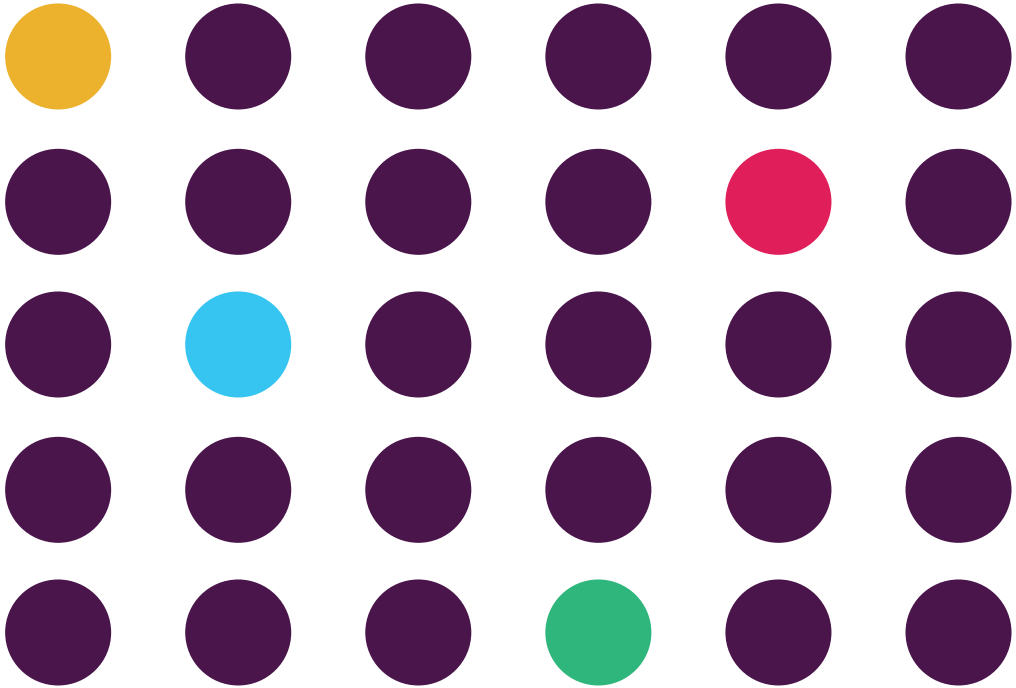
Variable infrastructure

Variable infrastructure

Variable infrastructure



Variable infrastructure



Immutable infrastructure

- Instances are untouched after provisioning
- Configuration changes happen only through reprovisioning
- No in-place patching allowed

Instance failure

The airplane analogy

Instance failure

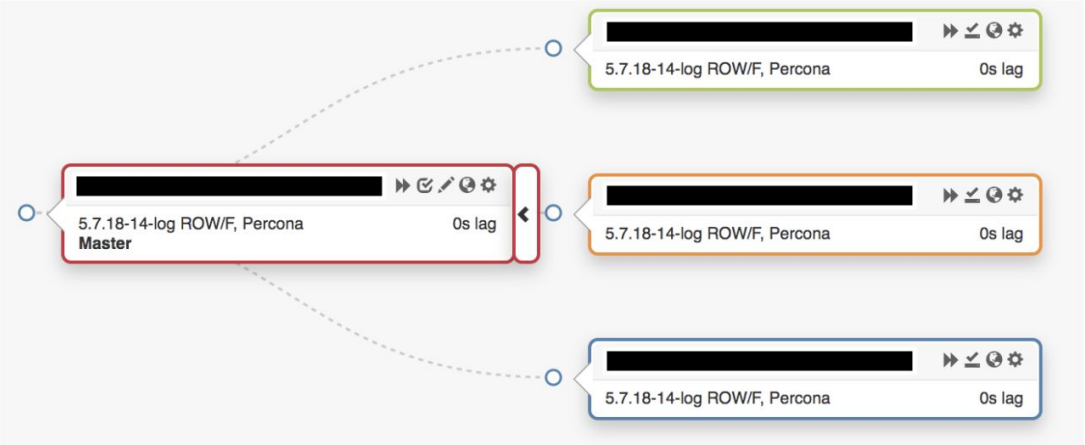
Always reprovision: challenges

- Network storage VS ephemeral instance storage
  - Network storage: stop & start instance
  - Instance storage: download the latest backup (NIC is the bottleneck)

- Small shards VS big shards
  - Recovery time (if you don’t use network storage)
  - Blast radius
  - Distributed workload VS centralized workload
  - Less contention

Durability through replication

via semi-sync



Durability through replication

via semi-sync

```
rpl_semi_sync_master_timeout = 9999999999999

rpl_semi_sync_master_wait_no_slave = 1

sync_binlog = OFF

innodb_flush_log_at_trx_commit = 2
```

How we run Vitess

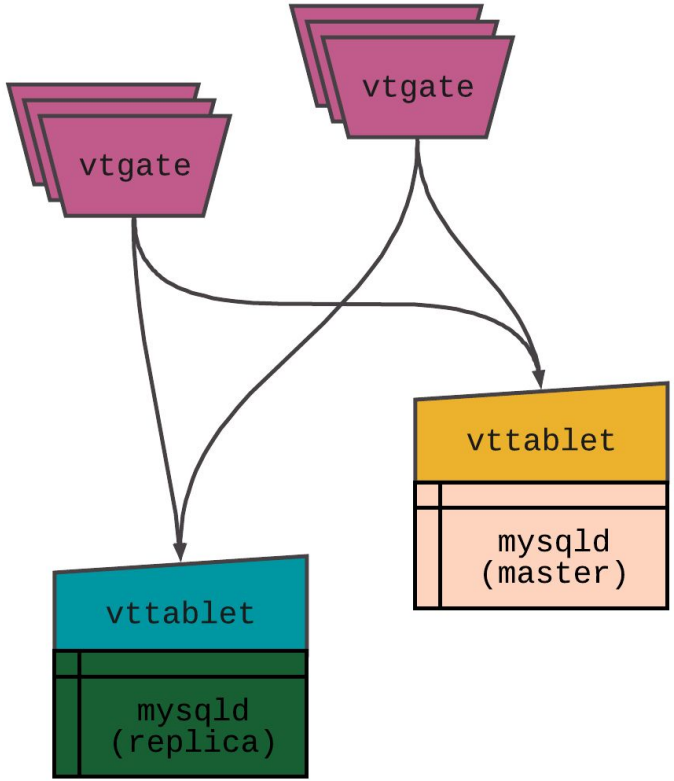
- AWS

How we run Vitess

- AWS
- EC2 not k8s

How we run Vitess

- AWS
- EC2 not k8s
- ASG for stateless components

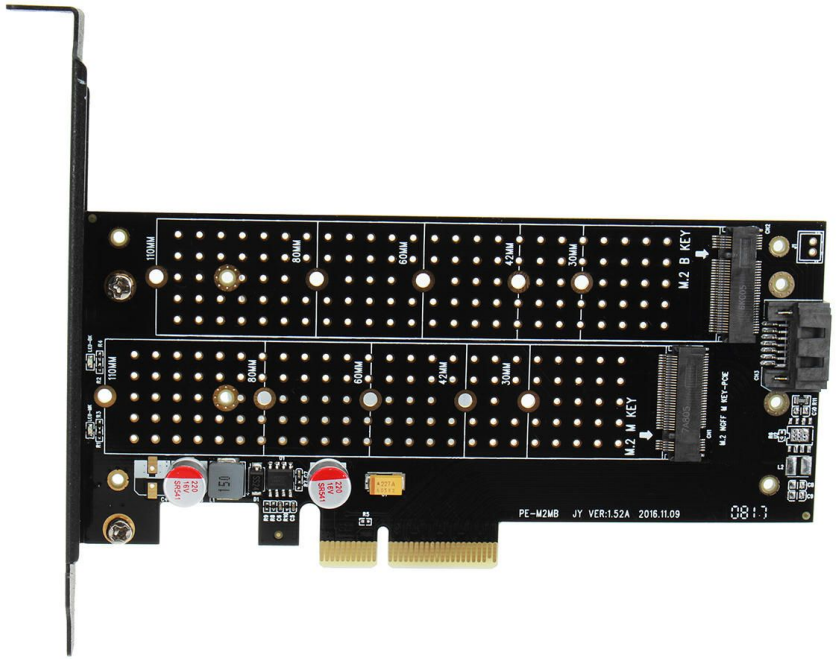


How we run Vitess

- AWS
- EC2 not k8s
- ASG for stateless components
- MySQL 5.7 (Percona)

How we run Vitess

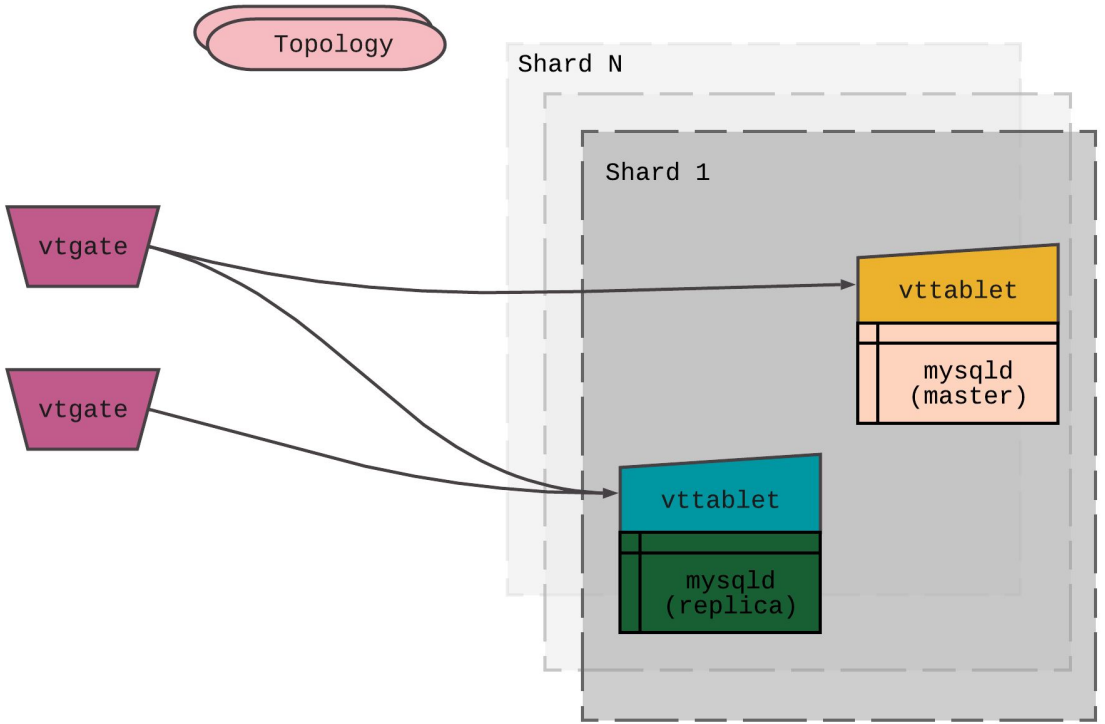
- AWS
- EC2 not k8s
- ASG for stateless components
- MySQL 5.7 (Percona)
- Ephemeral NVMe (no EBS)



Fault tolerance & isolation

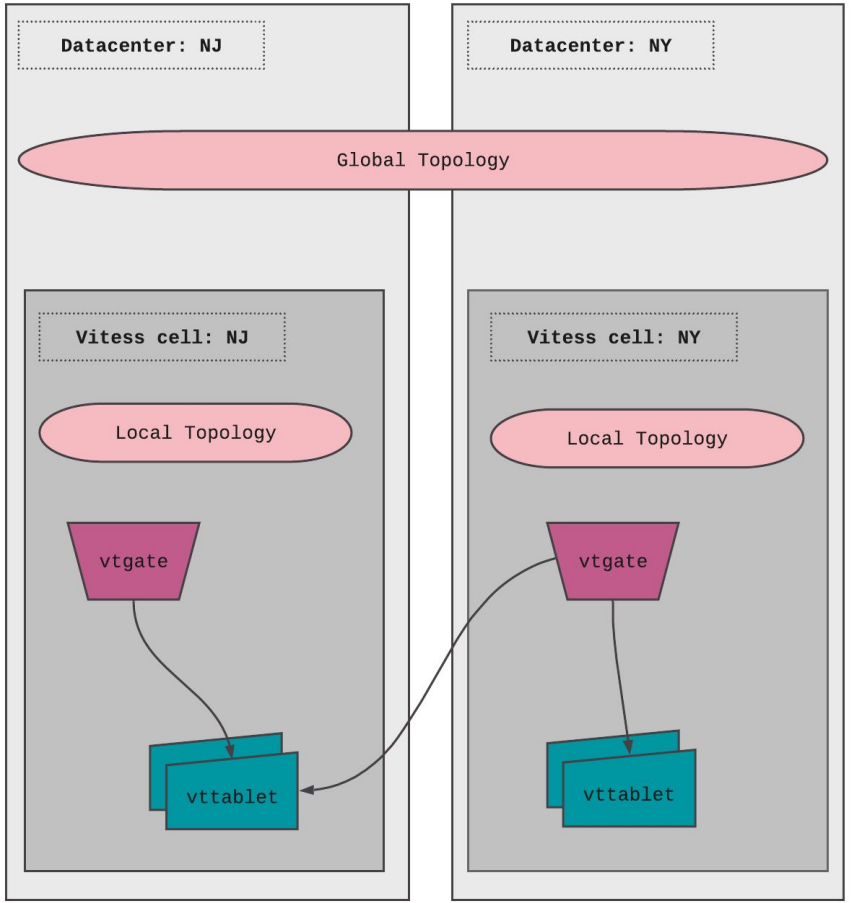
FAULT TOLERANCE & ISOLATION

Vitess architecture



FAULT TOLERANCE & ISOLATION

Vitess standard deployment



FAULT TOLERANCE & ISOLATION

Slack cloud infrastructure

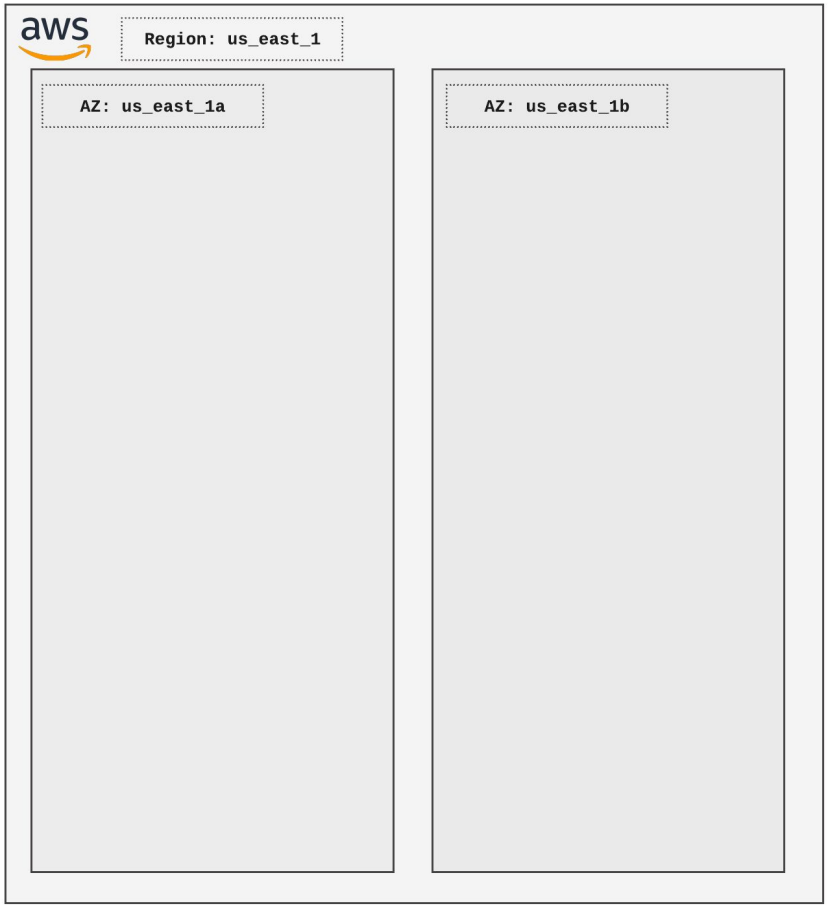
- Amazon EC2 is hosted in multiple locations world-wide.
- These locations are composed of Regions and Availability Zones (AZ's).
- Each Region is a separate geographic area.
- AZ's in a Region are connected through low-latency links.

source https://docs.aws.amazon.com/

FAULT TOLERANCE & ISOLATION

Vitess initial deployment

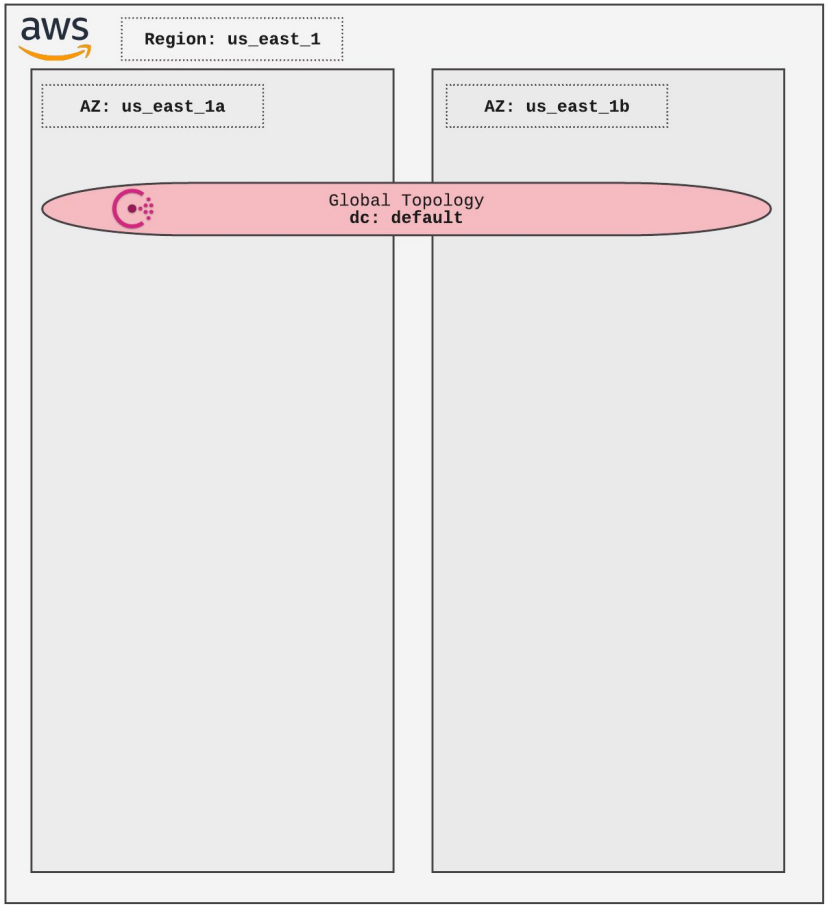
- We now have multiple clusters in different regions.



FAULT TOLERANCE & ISOLATION

Vitess initial deployment

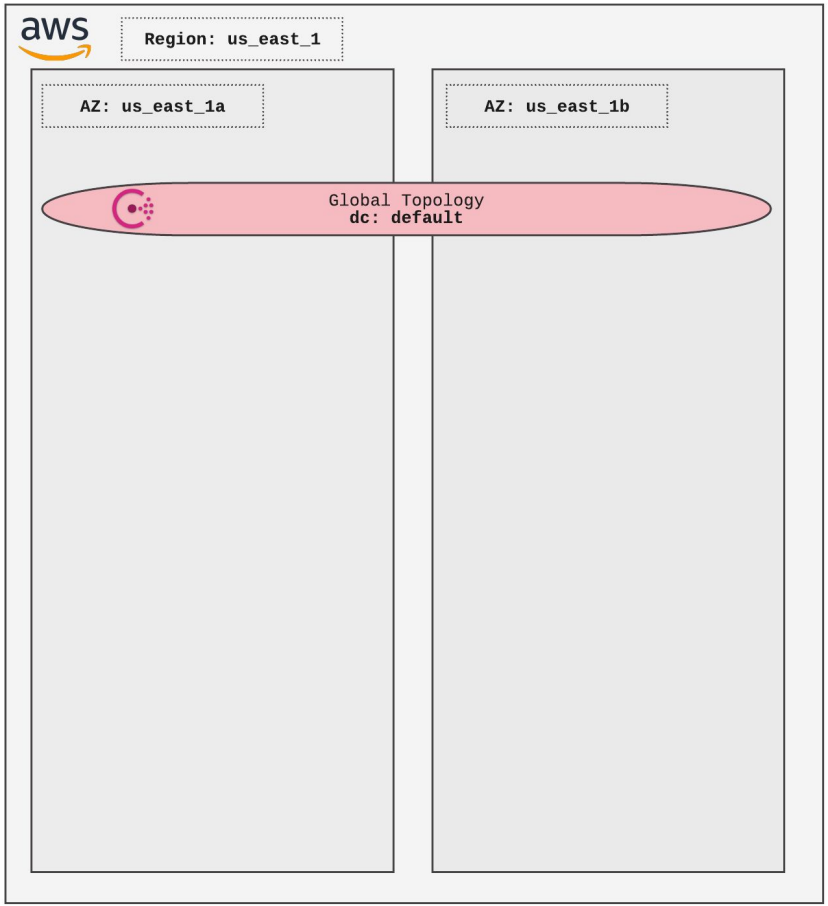
- We use Consul



FAULT TOLERANCE & ISOLATION

Vitess initial deployment

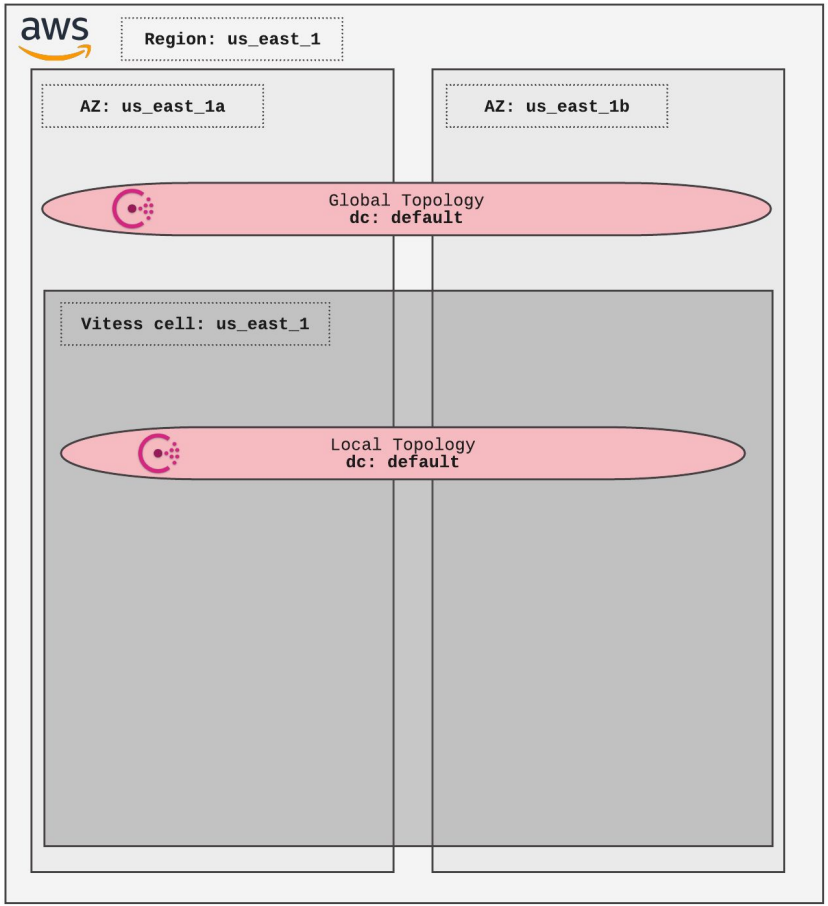
- We use Consul
- Notice **default** datacenter (dc) in Consul.



FAULT TOLERANCE & ISOLATION

Vitess initial deployment

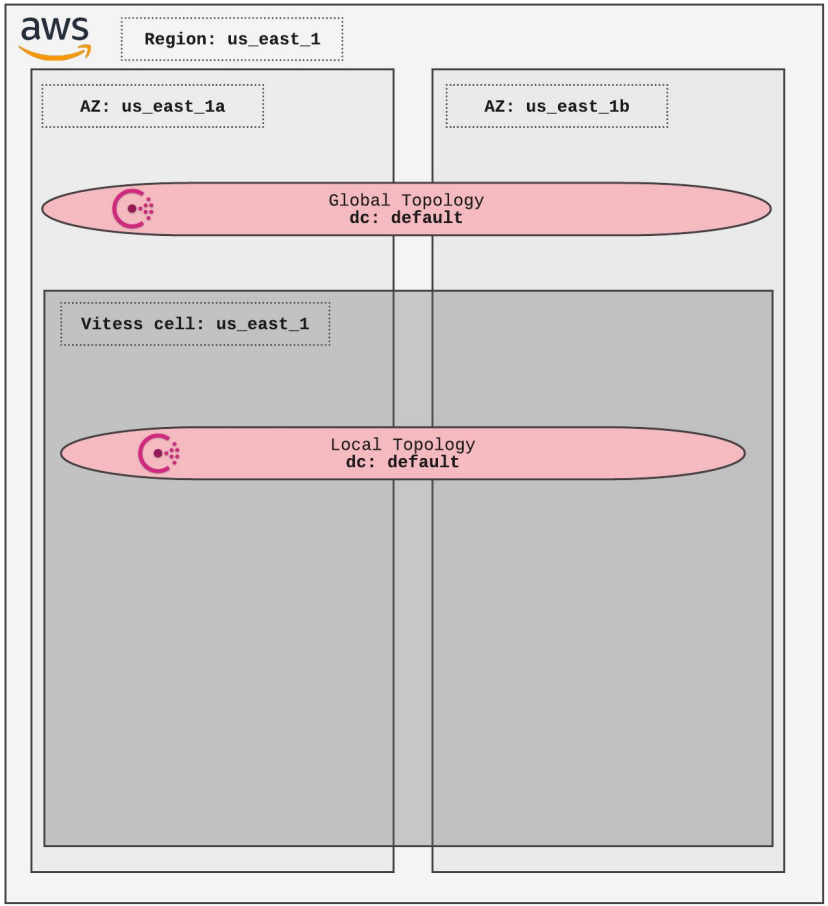
- Single **cell**.



FAULT TOLERANCE & ISOLATION

Vitess initial deployment

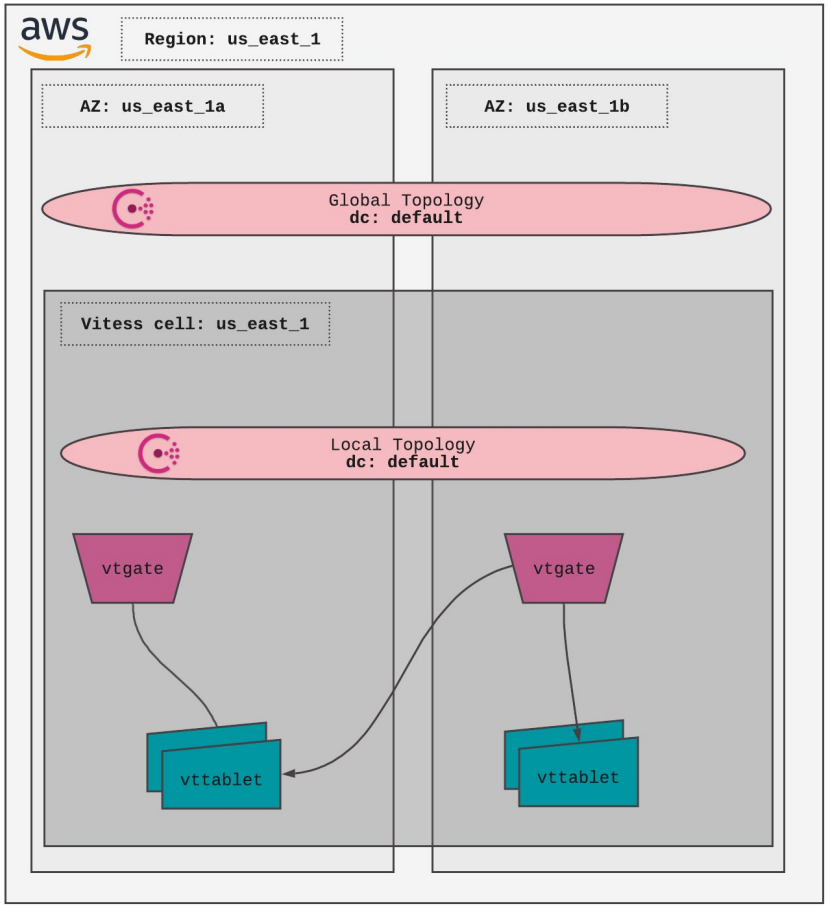
- Single **cell**.
- Same Consul dc: **default.**



FAULT TOLERANCE & ISOLATION

Vitess initial deployment

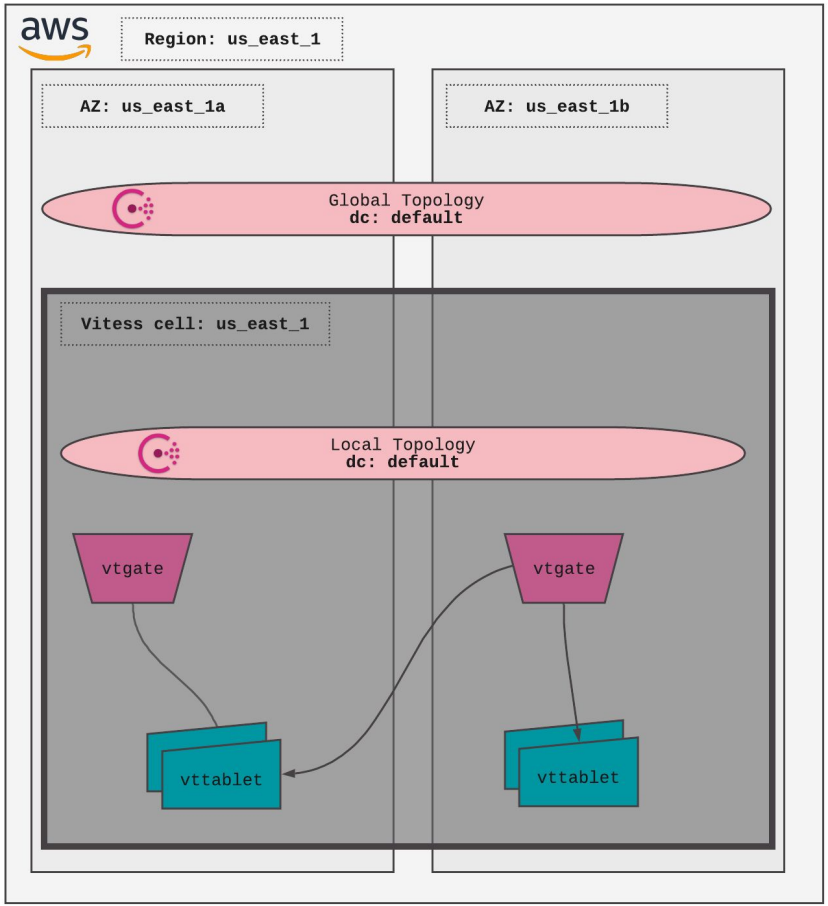
- Single **cell**.
- Same Consul dc: **default.**
- vtgates/tablets in different AZ's.



FAULT TOLERANCE & ISOLATION

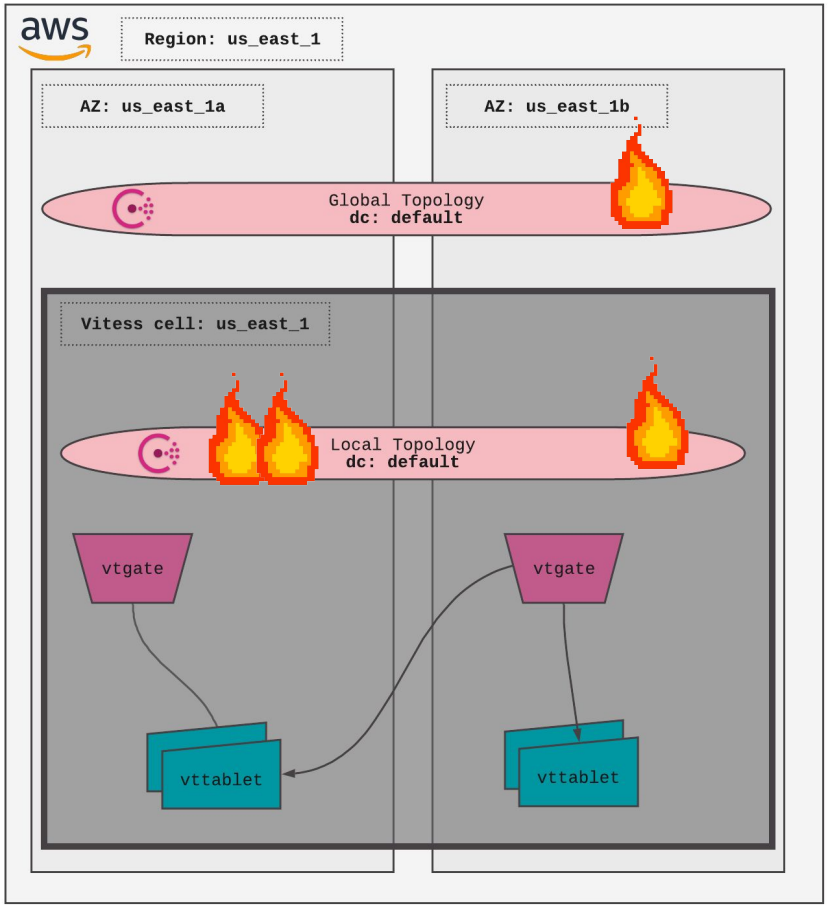
Vitess initial deployment

- A **single** cell across multiple AZ's (fundamental).
- Global and local topology using the same Consul cluster (circumstantial).



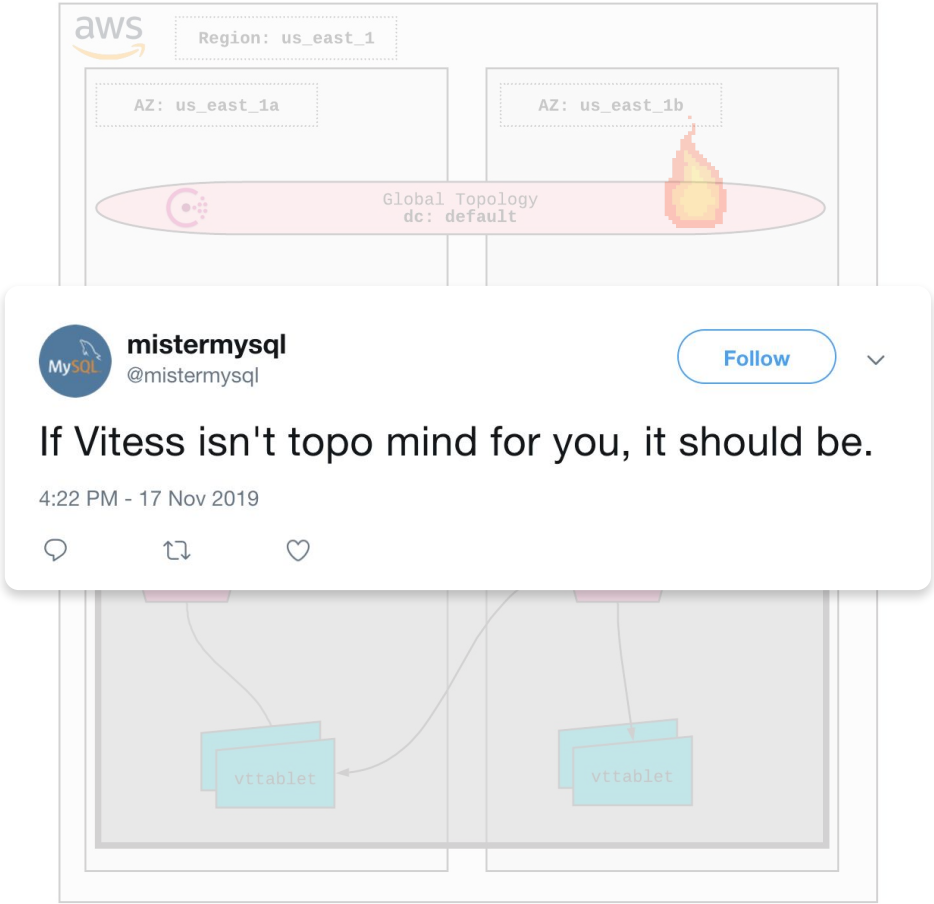
FAULT TOLERANCE & ISOLATION

Vitess initial deployment



FAULT TOLERANCE & ISOLATION

Vitess initial deployment



mistermysql
@mistermysql
Follow

If Vitess isn't topo mind for you, it should be.

4:22 PM - 17 Nov 2019

FAULT TOLERANCE & ISOLATION

We fixed things!

- Defensive programming.
- Fixing bugs.

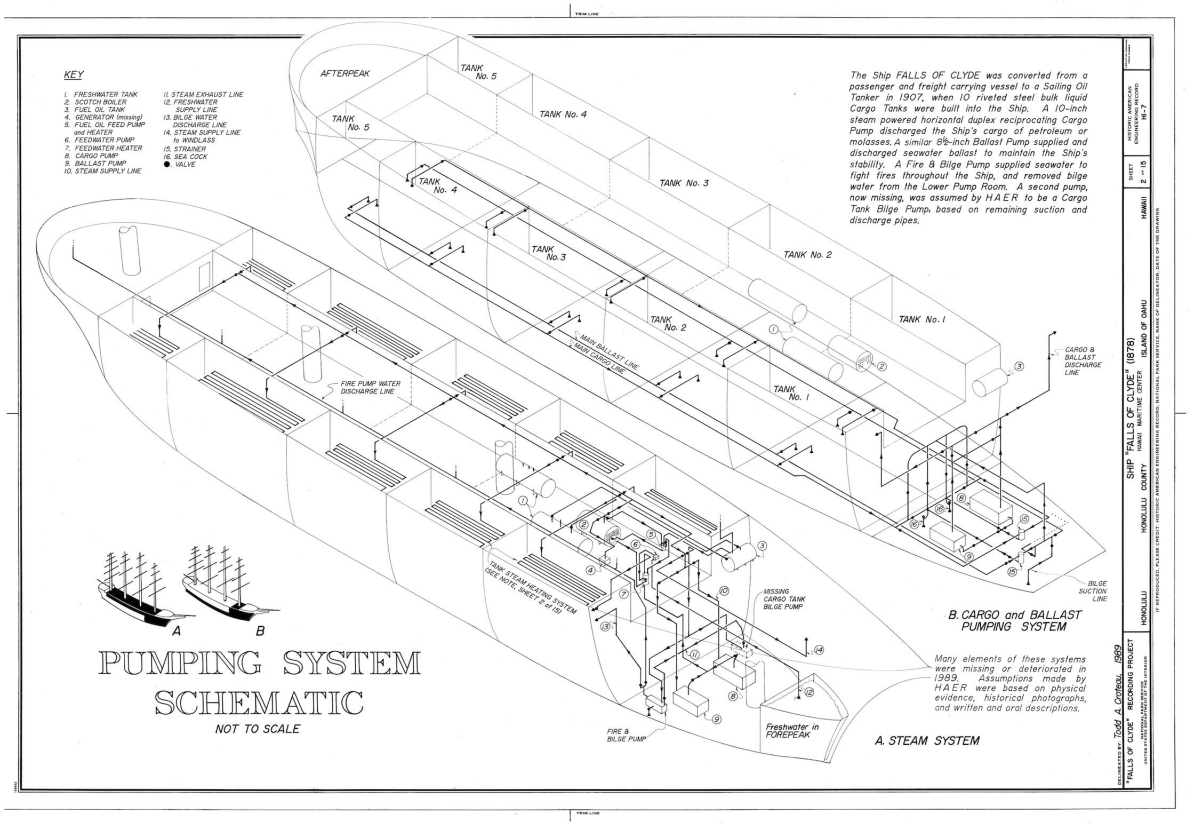
FAULT TOLERANCE & ISOLATION

Problem

Surfaced a fundamental issue in our deployment.

Resilient systems

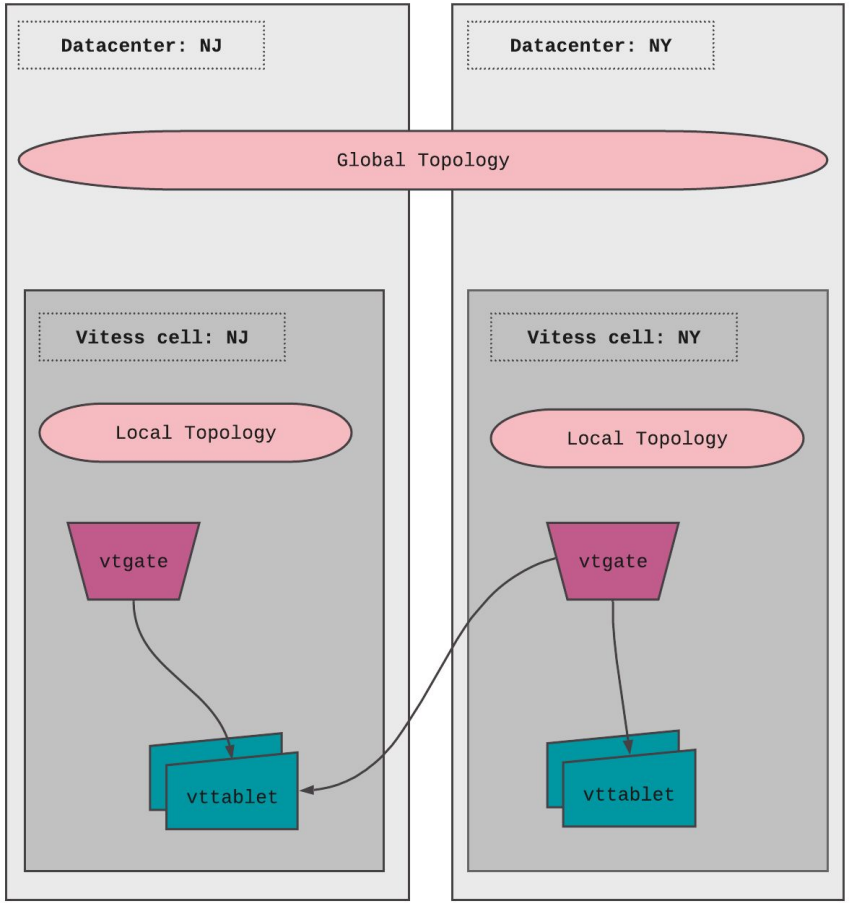
- Minimize the blast radius.
- Isolation is key.
- Understand your dependencies.



FAULT TOLERANCE & ISOLATION

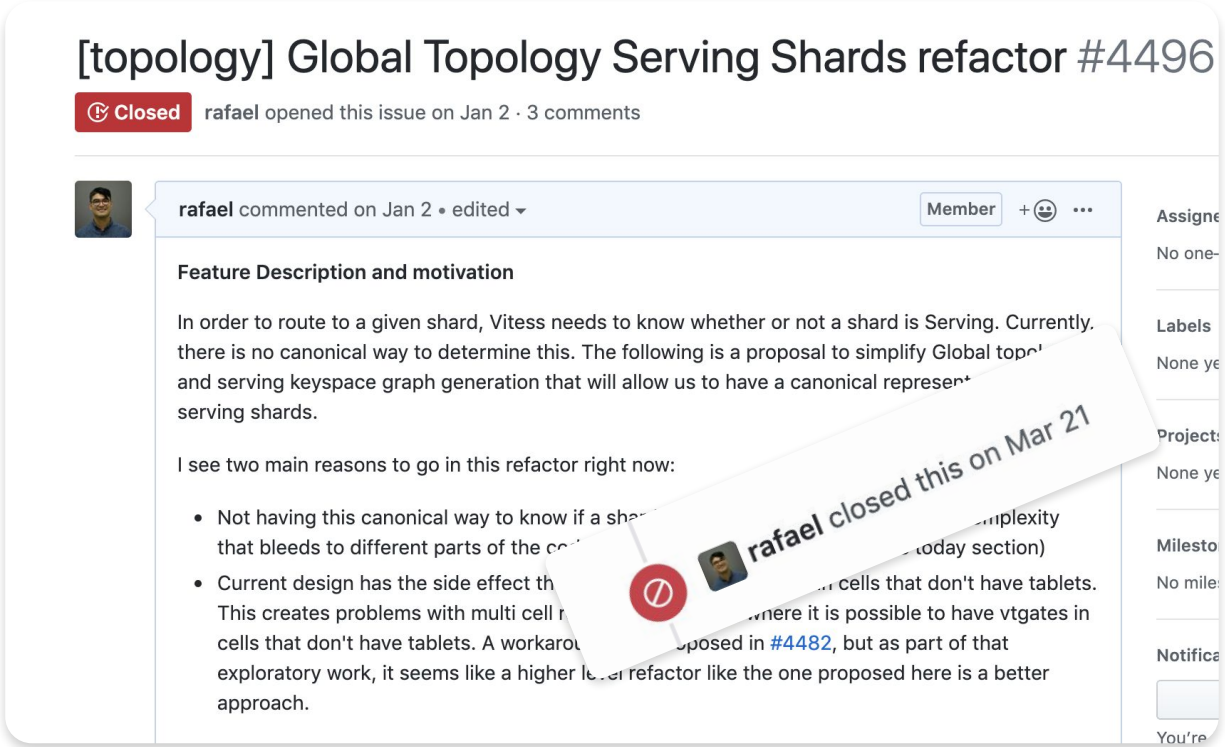
Current deployment

Easy right?



Current deployment

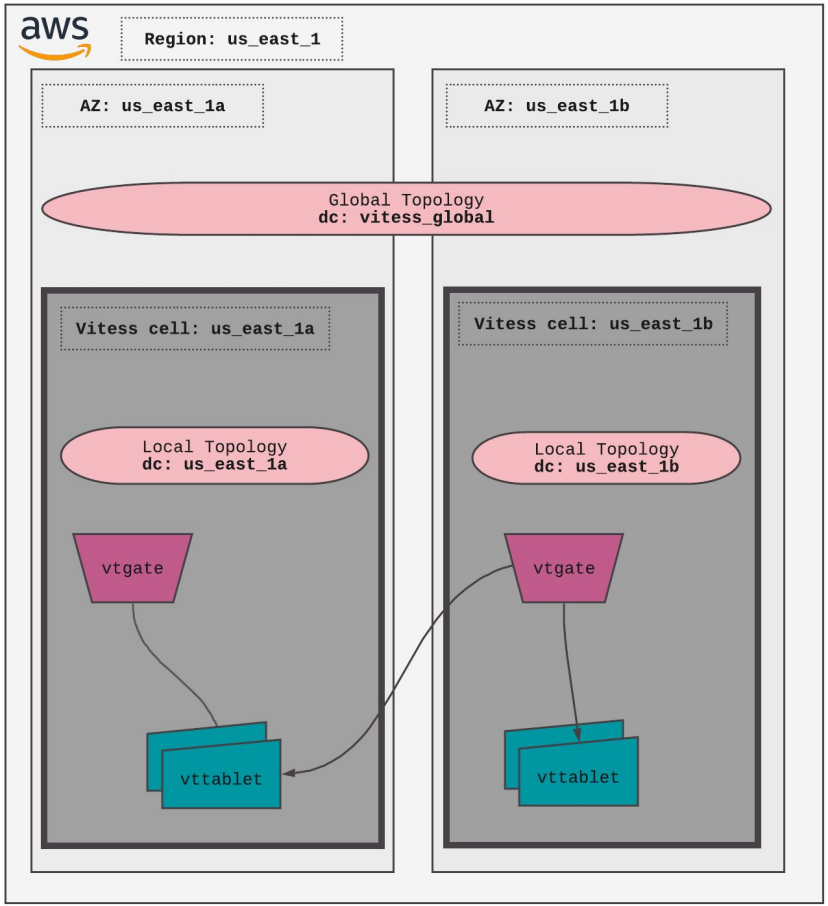
Easy right?



FAULT TOLERANCE & ISOLATION

Current deployment

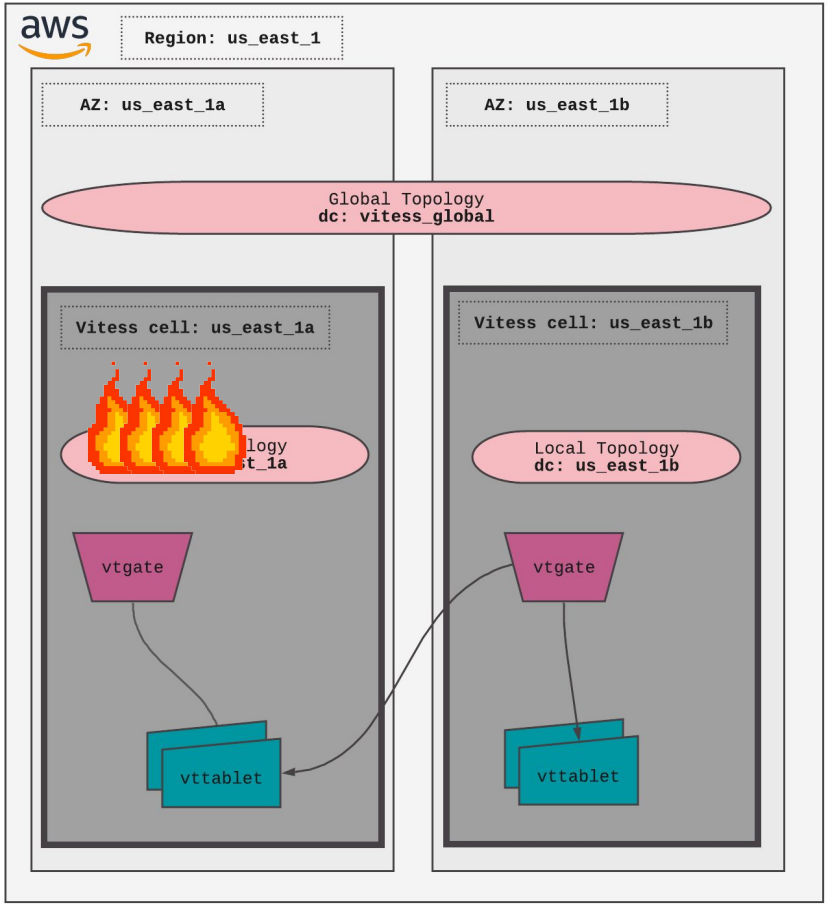
- Isolated topologies (one dc for each AZ and one for the global topo).
- Blast radius is mapped to physical infrastructure.



FAULT TOLERANCE & ISOLATION

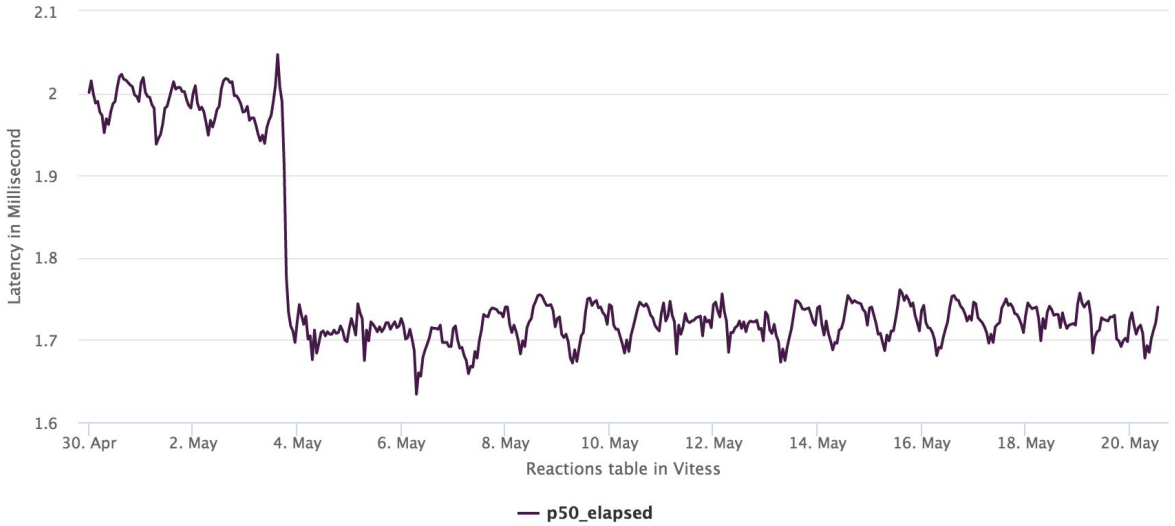
We have benefited already

- AZ failure during backup time.
- Single cell was affected!



FAULT TOLERANCE & ISOLATION

Performance wins



Key Lessons

KEY LESSONS

Complex system failures

- Complex systems are intrinsically dangerous systems.
- Complex systems are heavily and successfully defended against failure.
- Catastrophe is always just around the corner.
- Complex systems contain changing mixtures of failures latent within them.

A Short Treatise on the Nature of Failure; How Failure is Evaluated; How Failure is Attributed to Proximate Cause; and the Resulting New Understanding of Patient Safety - Richard I. Cook, MD (2000)

KEY LESSONS

Complex system failures

Humility towards complexity.

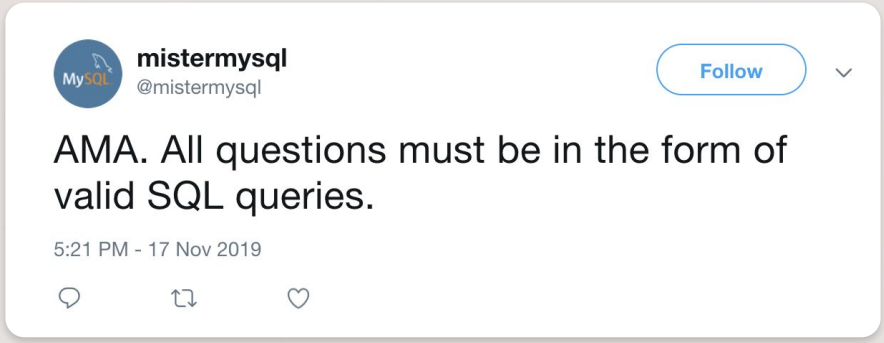
Reach out to other fields and learn from their experience.

Thank you!

P.S. We are hiring!

slack

Q&A


mistermysql
@mistermysql
Follow
AMA. All questions must be in the form of valid SQL queries.
5:21 PM - 17 Nov 2019

slack

Thank you!

P.S. We are hiring!

slack

Appendix

APPENDIX

Vitess configuration

Here are some of the configuration settings that we use in our setup.

They are the result of several years of tuning. We are sharing them so that the community can benefit as well.

```
./vtgate
    -buffer_size 10000
    -discovery_high_replication_lag_minimum_serving 5m
    -discovery_low_replication_lag 30s
    -enable_buffer
    -gateway_implementation discoverygateway
    -gateway_initial_tablet_timeout 120s
    -grpc_initial_conn_window_size 1073741824
    -grpc_initial_window_size 1073741824
    -grpc_keepalive_time 10s
    -grpc_keepalive_timeout 10s
    -grpc_server_initial_conn_window_size 1073741824
    -grpc_server_initial_window_size 1073741824
    -min_number_serving_vttablets 2
    -mysql_server_query_timeout 60s
    -mysql_server_read_timeout 60s
    -mysql_server_write_timeout 60s
    -normalize_queries
    -service_map grpc-vtgateservice
    -srv_topo_cache_refresh 5s
    -srv_topo_cache_ttl 8760h
    -tablet_refresh_interval 60s
    -tablet_refresh_known_tablets=false
    -tablet_types_to_wait MASTER,REPLICA
    -topo_read_concurrency 1
    -transaction_mode MULTI
```

APPENDIX

Vitess configuration

Here are some of the configuration settings that we use in our setup.

They are the result of several years of tuning. We are sharing them so that the community can benefit as well.

```
./vttablet
    -binlog_use_v3_resharding_mode
    -degraded_threshold 30s
    -enable-autocommit
    -enable_replication_reporter
    -enable_semi_sync
    -grpc_initial_conn_window_size 1073741824
    -grpc_initial_window_size 1073741824
    -grpc_server_initial_conn_window_size 1073741824
    -grpc_server_initial_window_size 1073741824
    -grpc_server_keepalive_enforcement_policy_min_time 2s
    -health_check_interval 1s
    -queryserver-config-idle-timeout 1200
    -queryserver-config-passthrough-dmls
    -queryserver-config-pool-size 150
    -queryserver-config-schema-reload-time 300
    -queryserver-config-transaction-cap 150
    -queryserver-config-txpool-timeout 3
    -unhealthy_threshold 1h
```