
PlanetScale

Data on Kubernetes / stateless storage

Matthias Crauwels
Enterprise Customer Engineer

@mcrauwel

SCaLE 23x - Cloud Native room
Pasadena, CA - Sat Mar 7th, 2026

Who am I?

- Living in Ghent, Belgium 🇧🇪
- Bachelor Computer Science
- ~30 years Linux user / admin
- ~20 years PHP developer
- ~15 years MySQL DBA
- 4th year at PlanetScale
- Currently Enterprise Customer Engineer
- Father of Leander

Kubernetes

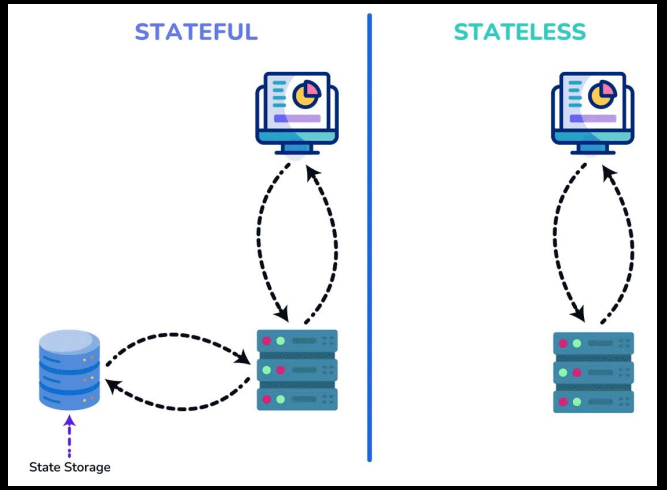
- Open source framework
- Management, deployment and scaling
- Containerized applications

Containerized applications

- Isolated
- Portable
- Lightweight
- Self-contained

Stateless vs Stateful

- Stateless
  - Each request is a new independent, isolated transaction. The server does not know anything about other requests
  - Think HTTP(S), DNS, …
- Stateful
  - History is stored to be reused
  - Think about a database



PlanetScale

- Founded in 2018
- Database-as-a-service product since 2021
- Originally only Vitess (MySQL based)
- Since 2025 also PostgreSQL
- Runs everything on Kubernetes

About Vitess

PlanetScale is powered by Vitess, the open-source database technology that was invented at YouTube in 2010 to solve the scaling issues they faced with their massive MySQL database.

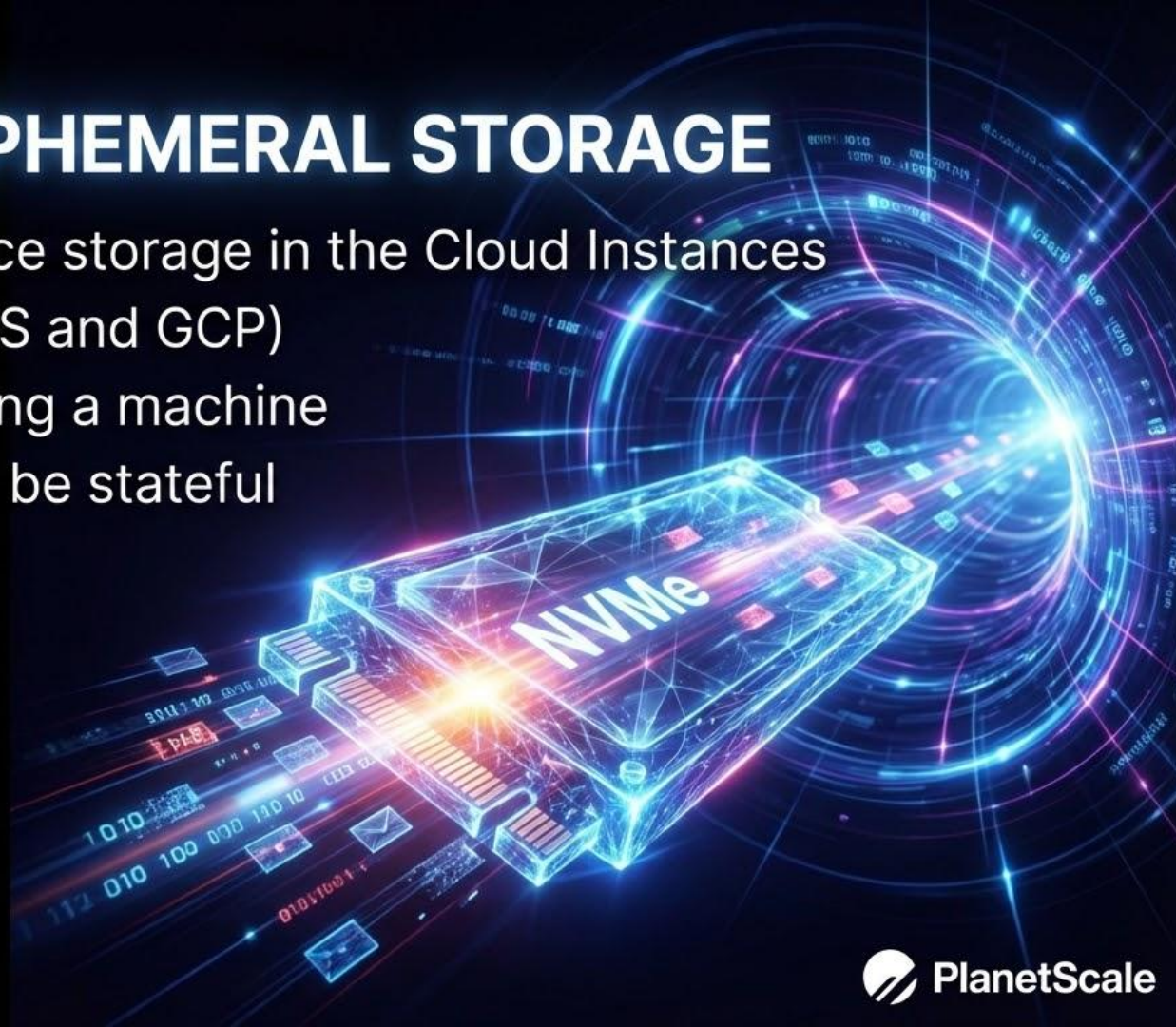
Vitess went on to become open source as a CNCF project and continues to scale massive companies like Slack, GitHub, and more.

Saving state

- Most people like their database to store its state
  (not everything runs on a BLACKHOLE storage engine)
- Databases can become quite large
  - Slow!
- Running in Cloud Block Storage
  - EBS in AWS / PD in GCP
  - Network-attached storage
  - SLOW!

RUNNING ON EPHEMERAL STORAGE

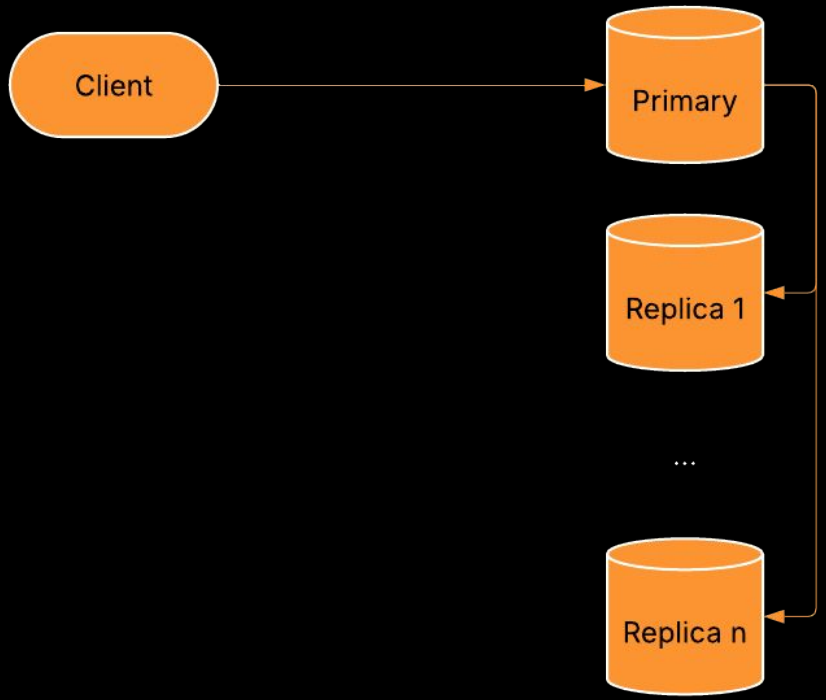
- Local NVMe instance storage in the Cloud Instances
  - (Available in AWS and GCP)
- Lost when destroying a machine
- Not really meant to be stateful
- Really f*cking fast!



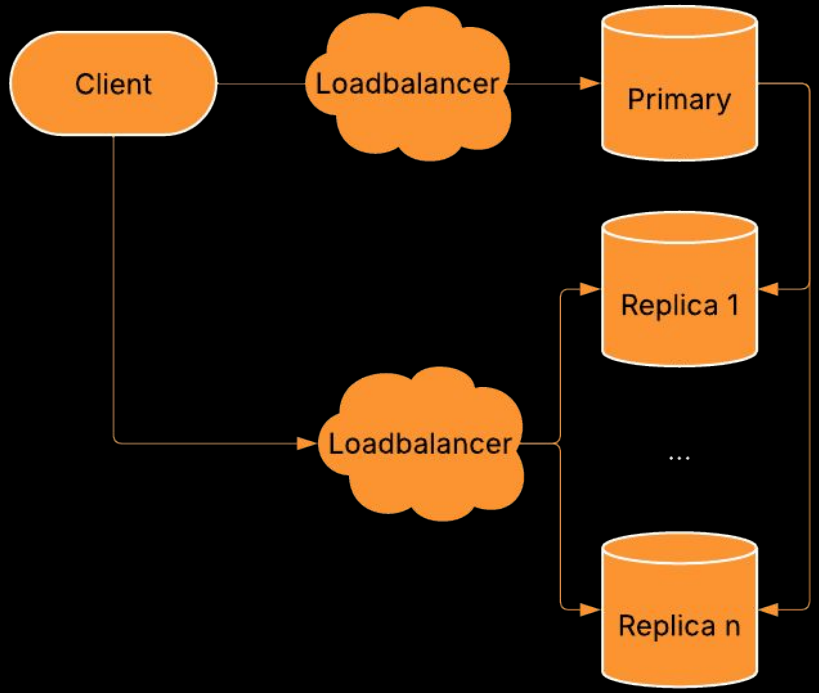
PlanetScale

Problems with this approach

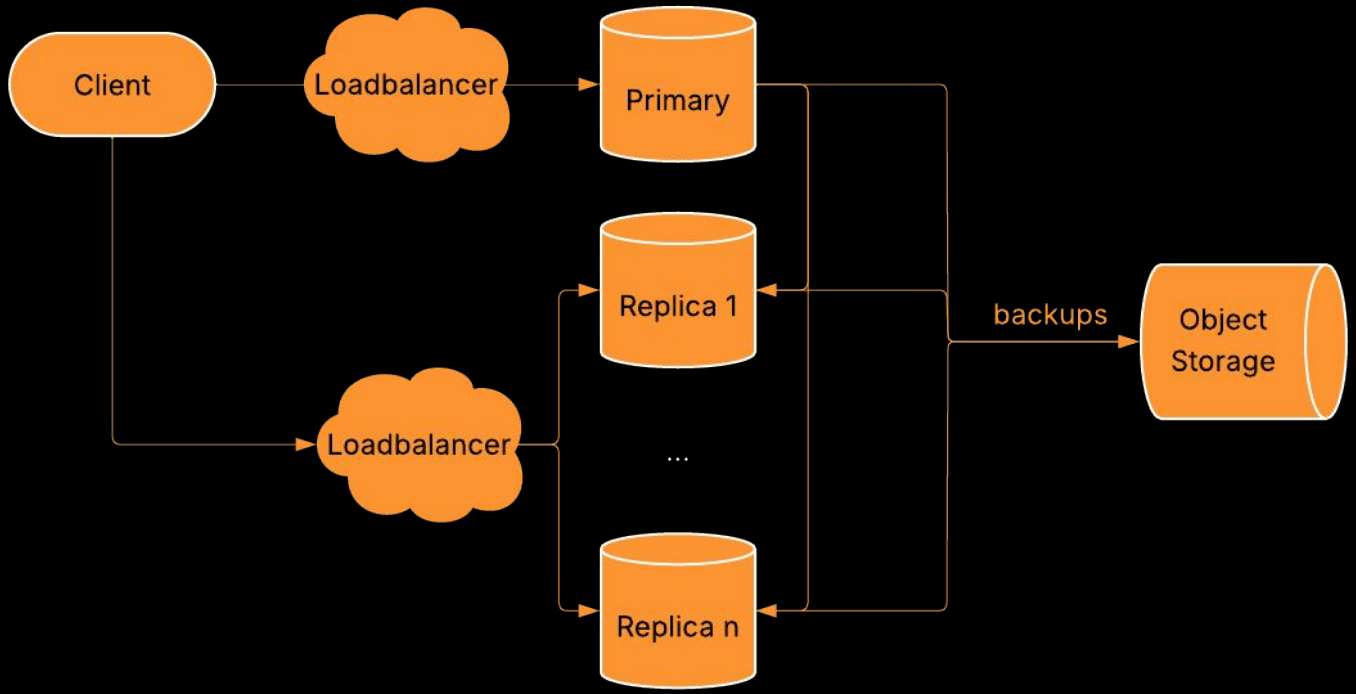
Problems with this approach



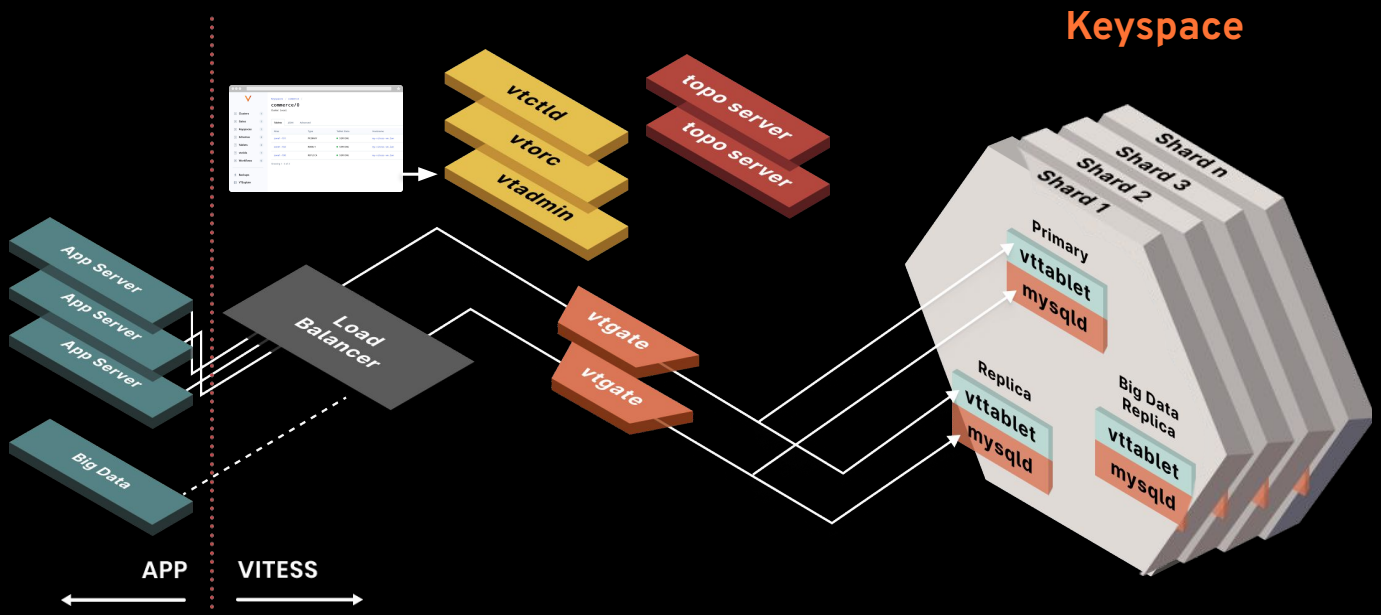
Problems with this approach



Problems with this approach



Vitess Architecture



Vitess serves millions of QPS in production

PlanetScale Metal

- Running your databases on ephemeral storage
- Makes the database run really fast with virtually unlimited IOPS
  - Multi-AZ deployment (minimal 3 AZs for production workloads)
  - Durability guarantee
    - MySQL: Semi-sync replication (requiring a cross-AZ acknowledgement)
    - PostgreSQL: at least one synchronous replica in another AZ
  - Daily full backup (verification) stored on object storage
    - GCS in GCP
    - S3 in AWS
  - Surge for upgrades

Questions?