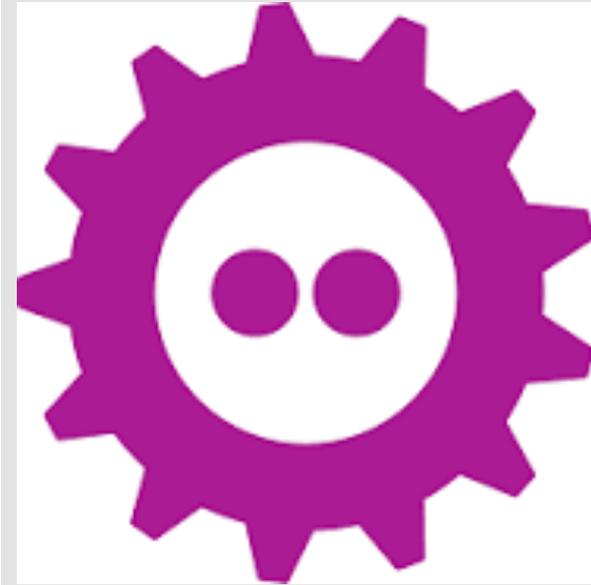


# Distributed Storage in the Cloud



Peter Zaitsev,  
Founder, Percona  
February 4th, 2023

## About Presentation

High Level Overview, Do not  
Expect Deep Details

I'm not Expert in all  
technologies covered, speak up  
if I'm wrong and correct me

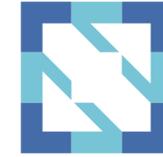
How do you  
take your  
Cloud ?



# Freedom or Serfdom ?



- Lock-in with Cloud Vendor
- Use Proprietary Solutions
- Highly Differentiated Cloud
- Hostage
- No Vendor Choice



**CLOUD NATIVE**  
COMPUTING FOUNDATION



- Freedom to Run Anywhere
- Use Open Source
- Cloud Is Commodity
- Customer
- Choice of Vendors



# Giving Cloud Its Originally Intended Role of Commodity Infrastructure

## What is Cloud Computing?

An analogy: think of electricity services...

You simply plug into a vast electrical grid managed by experts to get a low cost, reliable power supply – available to you with much greater efficiency than you could generate on your own.



Power is a utility service - available to you on-demand and you pay only for what you use.





**Can I deploy solution in my Environment without incurring additional costs ?**



**Do I have a broad choice of vendors if I need help ?**



**Can I improve software so it solves my needs better ?**

# Practical Open Source



# kubernetes

Kubernetes – Leading Open Source Orchestration for Public or Private Cloud

Applications



Uses Orchestration

Orchestration



Uses Provider

Provider



# Storage Types to Consider

- **Node Local Storage**
- **Network Attached Block Storage**
- **Network File System**
- **HTTP(S) Accessible Object Store**
- **Queues/Streams/Pipelines**
- **Databases**



**DATA MODEL**



**QUERY  
LANGUAGE**



**PURPOSE**



**INTERNAL DESIGN  
CONSIDERATIONS**

**Databases are Complicated!**

**Relational**

**Key Value**

**Document**

**Time  
Series**

**Graph**

**Other**

## Common Data Models



# Multi-Model

Some Databases Support Multiple Data Models, Some even Talk Different Languages/Protocols

## Shapechangers

---

Clickhouse can speak PostgreSQL and MySQL Protocols

---

VictoriaMetrics has InfluxDB and Graphite API

---

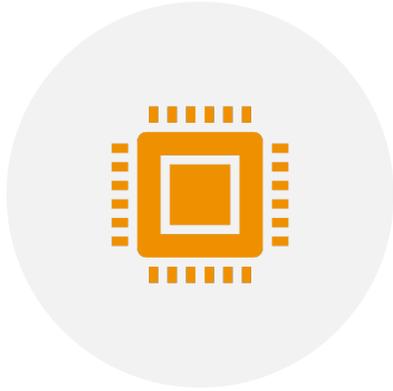
FerretDB allows to use PostgreSQL as if it were MongoDB

---

Babelfish turns PostgreSQL in MS SQL Compatible Database

# Purpose and Design

- **Operational/Transactional vs Analytical**
- **Cache vs Persistence**
- **Fully in Memory vs Storage Based**
- **Natively Distributed vs Fully Replicated**
- **Column Store vs Row Store**
- **Blockchain Based**



**REDUNDANCY, PERFORMANCE,  
SCALE**



**CLOUD DOES NOT WORK WELL  
WITH "PET" MODEL**



**COUNT ON ANY NODE CAN  
DISAPPEAR WITHOUT A TRACE**

**Why Distributed?**



# Let's Look into Storage Types and Options!

# Commodity Storage



**Has Relatively Simple Interface**



**Small/Medium Effort for Migration**



**Does not Create Strong Vendor Lock-In**



**Often Makes Sense to use Cloud Vendors  
solution as a “Building Block”**

## Node Local Storage



Basically, Local Disk

All Major Clouds Have Options

Performance can Differ by a Lot!

NVMe Flash storage tends to be the fastest

Great a Building Block for Distributed Storage or Local Processing needs



**AWS: Elastic Block Store (EBS)**



**Azure: Managed Disks**



**GCP: Persistent Disk (Zonal and Regional)**

# Network Block Storage - Clouds

**NetApp Cloud  
Volumes  
ONTAP**

**Portworx  
Persistent  
Volumes**

Network Block Storage: Vendors



**Ceph**



**Rook**



**Longhorn**



**OpenEBS**



**OpenStack Block Storage (cinder)**

Network Block Storage: Open Source



**AWS: Elastic File System  
(EFS)**



**Azure: Azure Files**



**GCP: Filestore**

## File Storage: Clouds

**NetApp**

**PortWorx**

File Storage: Vendors

**Ceph**

**Longhorn**

**Rook**

**OpenEBS**

**OpenStack  
Shared File  
System (manila)**

# File Storage: Open Source



**AWS: S3**



**Azure: Blob Storage**



**GCP: Google Storage**

# Object Storage: Clouds

Object Storage:  
Vendors

---

**NetApp**

---

**PortWorks**

---

**Wasabi**

---

**Blackblaze B2**

---

**Digital Ocean**

---

**Open Telekom Cloud**



Minio

Ceph

Rook

Object Store: Open Source

## Datastores and Databases

---

Highly Differentiated

---

“Similar” Offerings are NOT Easily  
Replicable

---

Important to use Open Source Solutions  
if you want to avoid Vendor LockIn

Queues,  
Streams, Data  
Pipelines

**Moving Data around with  
Persistence**

**Is NOT conventional Database**

**Key part of many modern data  
architectures**

## AWS

- Kinesis
- Data Pipeline
- Simple Queueing Service (SQS)
- Simple Notification Service (SNS)
- Managed Kafka

## Azure

- Data Factory
- Event Hubs
- Event Hubs for Apache Kafka

## GCP

- Dataflow
- Pub/Sub

# Queues: Clouds

**Confluent  
Kafka**

**Aiven  
Managed  
Kafka**

**Red Panda**

Queues: Proprietary

Apache  
Kafka

Apache  
Pulsar

RabbitMQ

ActiveMQ

Many  
Others....

Queues: Open Source



## AWS

Aurora (MySQL, PostgreSQL)  
RDS (MySQL, PostgreSQL,  
MariaDB, Oracle, SQL Server)



## Azure

SQL Database  
Azure Database (MySQL,  
PostgreSQL, MariaDB)  
Hyperscale



## GCP

Spanner  
CloudSQL (MySQL, PostgreSQL,  
SQL Server)

# Relational Transactional - Cloud

**Oracle**

**Microsoft  
(SQL Server)**

**Yugabyte  
Cloud**

**CochroachDB  
Cloud**

**Instaclustr  
Managed  
PostgreSQL**

**Aiven  
Managed  
Databases**

**SkySQL  
(MariaDB in  
the cloud)**

**TiDB Cloud**

**Relational Transactional Proprietary**

---

**PostgreSQL**

---

**MySQL**

---

**MariaDB**

---

**Yugabyte**

---

**TiDB**

---

**Percona Distributions for MySQL and PostgreSQL**

---

**Relational Transactional – Open Source**

# Relational Analytical - Cloud

## AWS

- RedShift
- Athena

## Azure

- Synapse Analytics
- Data Lake Analytic

## GCP

- BigQuery

Relational  
Analytical -  
Proprietary

---

**Oracle**

---

**Snowflake**

---

**Vertica**

---

**Managed DataBricks**

---

**SingleStore**

---

**Oracle HeatWave**

Spark

Hadoop

Presto

Trino

ClickHouse

MariaDB  
ColumnStore

TiDB (HTAP)

## Relational Analytical – Open Source

**AWS:  
DocumentDB**

**Azure:  
CosmosDB**

**GCP:  
FireStore**

## Document Store - Cloud

---

MongoDB Atlas

---

MongoDB Enterprise

---

Couchbase Cloud

---

Couchbase Enterprise

Document Store - Proprietary



**MONGODB  
COMMUNITY**



**COUCHBASE  
COMMUNITY**



**RELATIONAL  
DATABASES**



**PERCONA SERVER  
FOR MONGODB**

Document Store – Open Source (and Source Available)

# Key-Value Stores for Caching: Cloud

## AWS

- ElastiCache

## Azure

- Cache
- Cache for Redis

## GCP

- Memstore

**Redis  
Enterprise**

**Redis  
Cloud**

Key-Value Stores for Caching – Proprietary

Redis

Memcached

KeyDB

Key-Value Stores for Caching: Open Source

**AWS:  
DynamoDB**

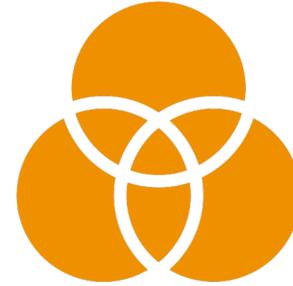
**Azure:  
CosmosDB**

**GCP: Big  
Table**

Persistent Key-Value Stores : Cloud



Redis Cloud



“Enterprise” Versions of Open  
Source Solutions

# Persistent Key Value Stores - Proprietary

Persistent Key-  
Value Stores:  
Open Source

Redis

Cassandra

ScyllaDB

Aerospike

Tarantool

Apache  
Ignite

Relational  
Databases



**AWS: Timestream**



**Azure: Azure Data Explorer**



**GCP: Cloud BigTable**

# Time Series Databases: Cloud

**Enterprise  
Versions of Open  
Core Products**

**Cloud DBaaS of  
Open  
Source/Open  
Core Products**

**Time Series: Proprietary**

**TimescaleDB**

**InfluxDB**

**OpenTSB**

**Prometheus**

**VictoriaMetrics**

**Cortex**

**M3**

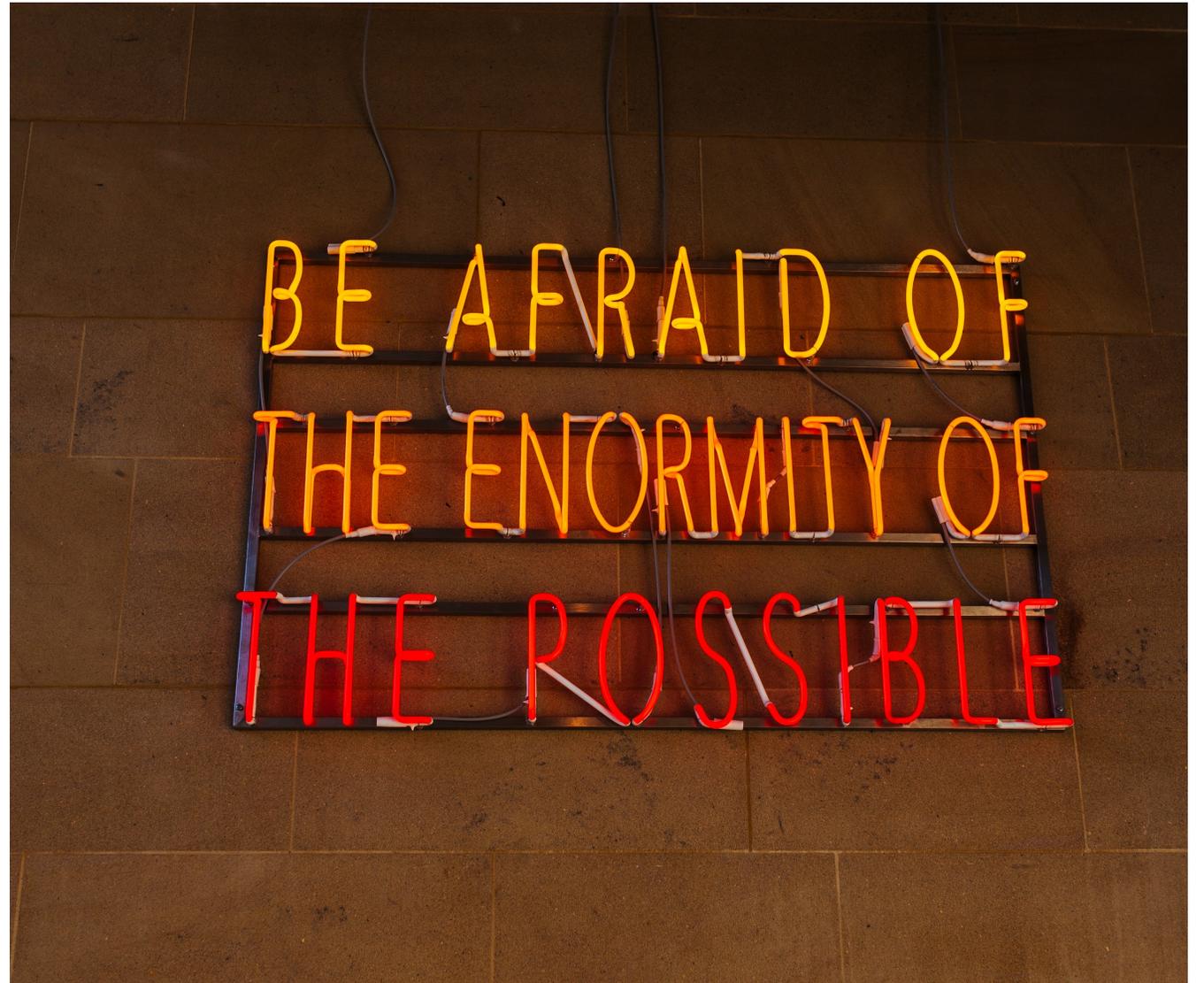
**Graphite**

# Time Series: Open Source

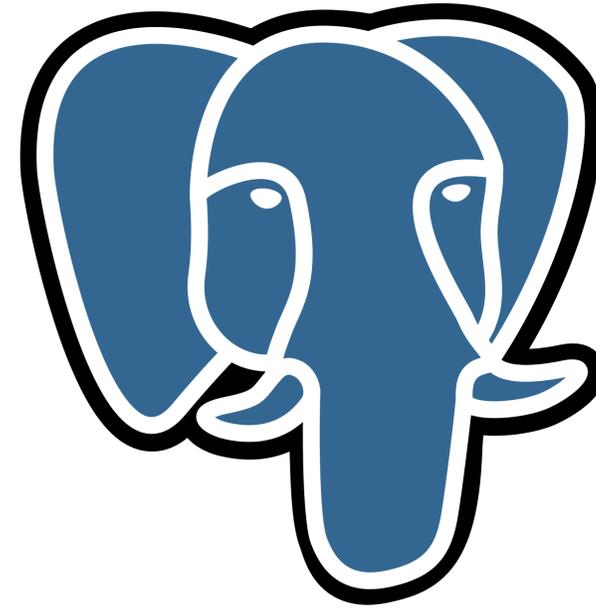
## Percona's Role



# Pushing Boundaries of Open Source Databases



Focus on MySQL,  
MongoDB,  
PostgreSQL



Percona  
Distributions for  
MySQL, MongoDB,  
PostgreSQL

Linux and  
Kubernetes  
(Operator)  
Deployment

100% Free and Open Source Database Software

# Percona Monitoring and Management (PMM)

Single Pane of Glass for MySQL, MongoDB, PostgreSQL

Observability

Management

DBaaS Experience



**PERCONA**  
Monitoring and Management

# Distributed Storage in the Cloud



**Is Quite Complicated**



**No One Size Fits all – look for the best tool for the job**



**If you so desire there are great Open Source Solutions available**

Thank you, Let's Connect!

<https://www.linkedin.com/in/peterzaitsev/>

<https://twitter.com/PeterZaitsev>

<http://www.peterzaitsev.com>