

YugabyteDB: Distributed SQL Open-Source PostgreSQL-compatible

Franck Pachot, Developer Advocate



Scaling-out the database: Why?

High Availability:

- the DB runs on more than one server
- because a server can fail at any time
- because servers must be upgraded



Elasticity:

- add more servers rather than stop + start a larger VM
- stop servers when the load is low to reduce cost



Cloud:

- commodity hardware can fail a any time
- infrastructure on multiple zones, regions, multi/hybrid cloud, serverless, multitenant



Scaling-out the database: How?



Traditional RDBMS:

the DB writes to shared memory -> cannot scale horizontally



NoSQL:

gives up on consistency, joins, integrity, secondary indexes...



Sharding:

Partitions to multiple traditional DB -> all features but no global indexes, no FKs...



Read replicas:

Primary and (sync) standby -> all features but scales only reads



Distributed SQL:

data is distributed and replicated, with global tx, fk, gsi and HA

Distributed SQL actors (spanner-like architecture)



Google Spanner:

SQL features on geo-distributed storage with atomic clocks



CockroachDB:

PostgreSQL drivers and syntax, runs on commodity hardware

also some MySQL compatible: TiDB, Xpand




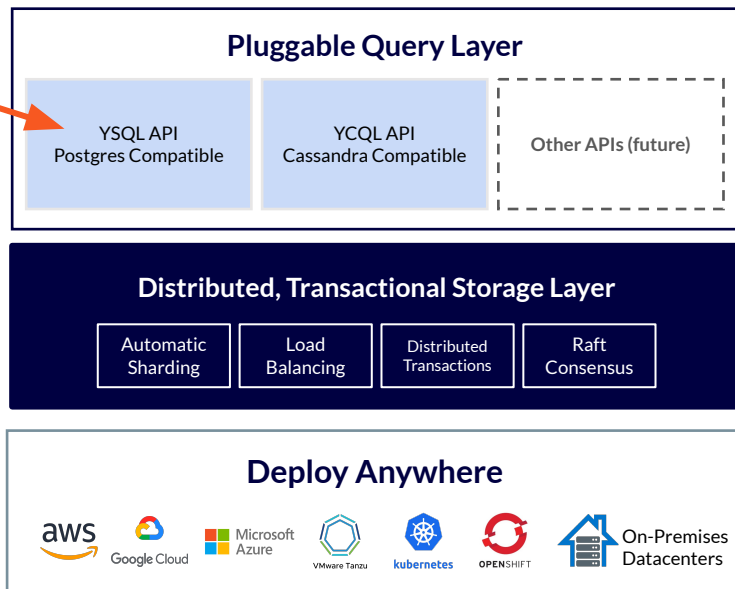
YugabyteDB

Open Source, PostgreSQL query layer, same runtime behavior without storage weaknesses (bloat, xid wraparound, vacuum,...)

distributes (by hash or range) on primary and index keys, with replication

YugabyteDB architecture

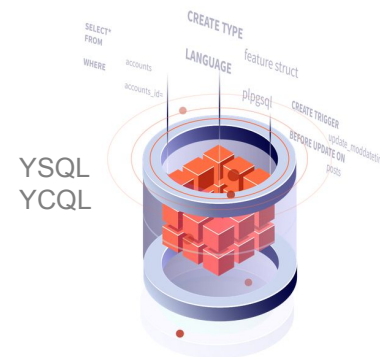
PostgreSQL Application 



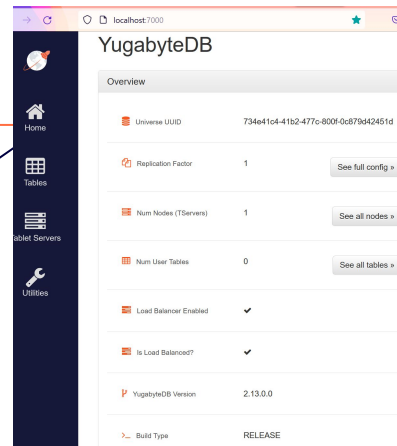
Open Source
Distributed
Database

SQL

★ <https://github.com/yugabyte/yugabyte-db>



Quick Start (dev environment)



Start YugabyteDB (3 nodes)

```
docker network create -d bridge yb
```

```
docker run -d --name yb0 --hostname yb0 --net=yb -p 5433:5433 -p7000:7000 \  
yugabytedb/yugabyte:latest yugabyted start --daemon=false --listen 0.0.0.0
```

```
docker run -d --name yb1 --hostname yb1 --net=yb -p 5434:5433 \  
yugabytedb/yugabyte:latest yugabyted start --daemon=false --listen 0.0.0.0 --join yb0
```

```
docker run -d --name yb2 --hostname yb2 --net=yb -p 5435:5433 \  
yugabytedb/yugabyte:latest yugabyted start --daemon=false --listen 0.0.0.0 --join yb0
```

Connect to any nodes

```
psql postgres://localhost:5433  
  select * from yb_servers();
```

```
-- Web console on http://localhost:7000
```

When it can be an **alternative to PostgreSQL**:

- OLTP (with some analytics push-downs)
- cloud native HA (Compute instances, kubernetes pods)
- multitenant, geo-distribution (data residency)

Community slack
for questions (or just say hello):

<https://communityinviter.com/apps/yugabyte-db/register>



E-mail:

fpachot@yugabyte.com

Blogs:

dev.to/FranckPachot

blog.yugabyte.com/author/fpachot

Twitter:

@FranckPachot

Youtube:

youtube.pachot.net

Twitch:

www.twitch.tv/franckpachot

LinkedIn:

www.linkedin.com/in/franckpachot