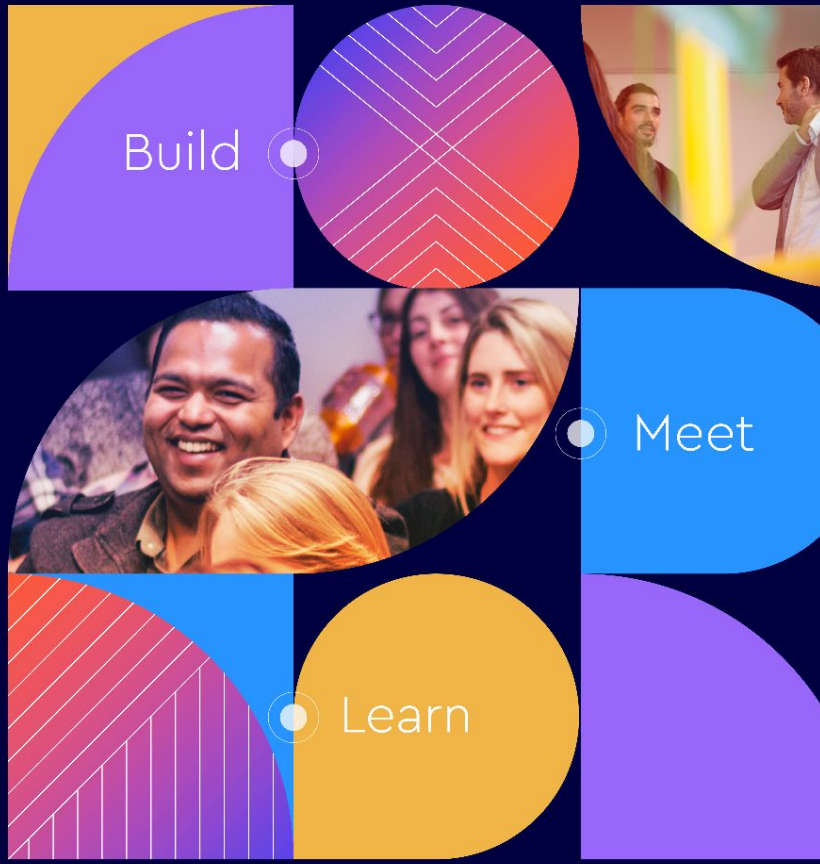
yugabyteDB

Enterprise Scale: Uncover how Distributed Architecture Delivers Scale and Resilience



Rahul Batta

Principal Pre-Sales Engineer

Amit Chauhan

Principal Pre-Sales Engineer

Why are Scale and Resilience so Important?

Agenda

- Why Yugabyte
- How Yugabyte does it..
  - DB Internals
  - Topologies
- Show and Tell Demos
  - YB Managed *peek thru*
  - Region Failure
  - Scaling - Horizontal & Vertical
  - Upgrade DB
- Q&A

Why Yugabyte's Distributed Architecture

- Cross-Category Competition
- Flexibility with Microservices
- Anywhere Data Access
- Edge Creation & Consumption
- Innovations Released Daily
- Application-led Buying

Application Modernization

Containers | App Frameworks | Kubernetes

Time for Data Layer Modernization?

Infrastructure Modernization

Cloud | Virtualization | Commodity HW

Today's solutions fail to advance the data layer without tradeoffs

Legacy Relational

- ✓ Consistency / ACID
- ✓ Indexes
- ✓ Relational data model

But tradeoffs include...

- ✗ Manual sharding
- ✗ Bolt-on replication
- ✗ No geo-distribution

Non-Relational

- ✓ Scalability
- ✓ Geo-distribution
- ✓ Native replication

But tradeoffs include...

- ✗ No consistency / ACID
- ✗ No indexes
- ✗ No relational model

Cloud DB

- ✓ Ease of use
- ✓ Resilience
- ✓ Consistency

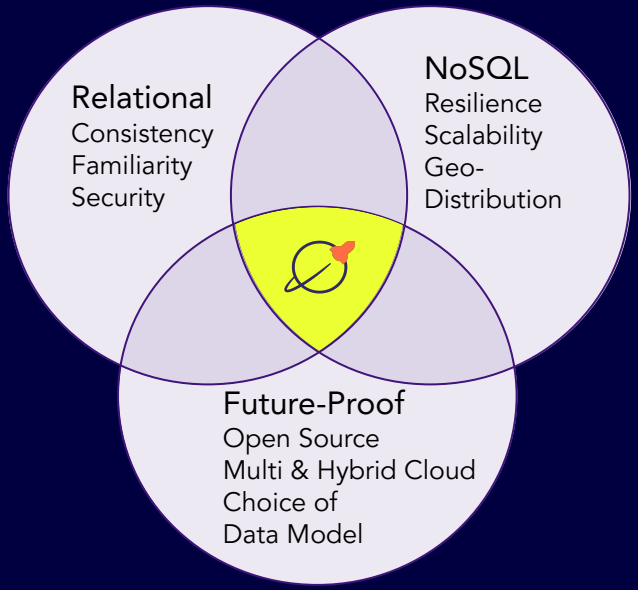
But tradeoffs include…

- ✗ Cloud lock-in
- ✗ No open source
- ✗ Limited scale and geo-distribution

Trade-offs Cost Your Business

Lower Productivity | Higher Cost | Higher Risks

Our Mission: Eliminate your data layer tradeoffs



Postgres Familiarity & Consistency

- Reuses the query layer of PostgreSQL
- Includes advanced features (triggers, stored procedures, security)

Top 5 financial services company modernized from DB2 to YugabyteDB

Rapid Horizontal Scalability

- Billions of op/day, hundreds of TBs
- Rapidly scale queries, storage and connections by adding nodes

Multiple top retailers easily scale for peak traffic (Black Friday/Cyber Monday)

Built-in Resiliency & Availability

- Survive multiple failures with automatic healing and re-replication
- Zero-downtime upgrades and security patching

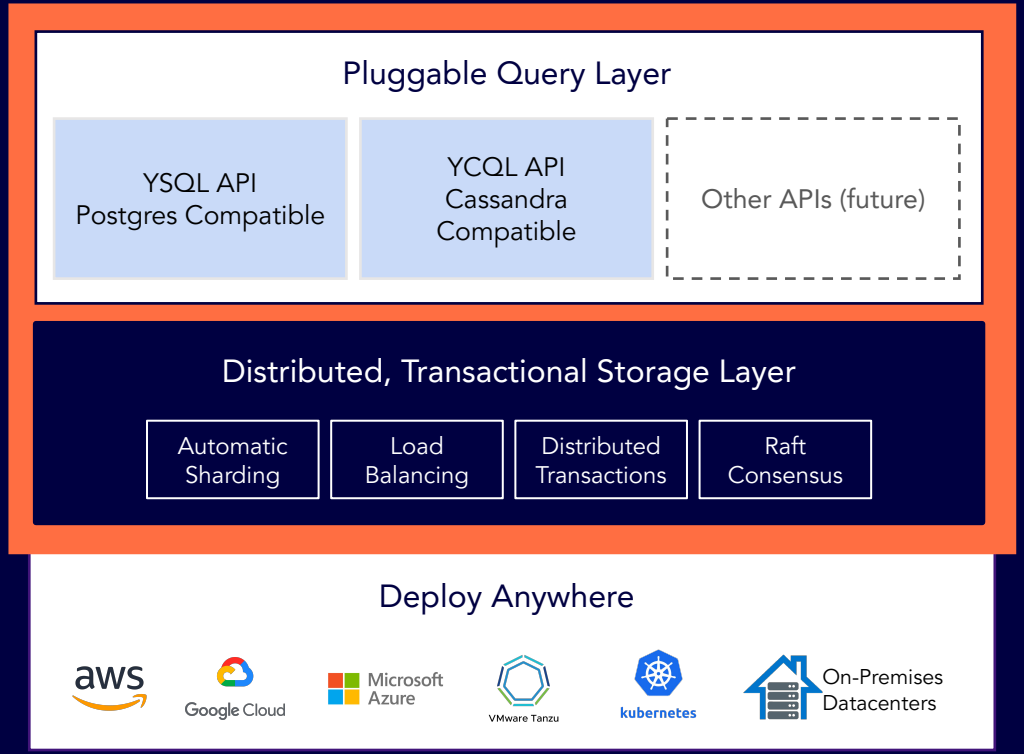
Top 5 retailer does upgrades & survives DC outage while serving >250K qps

Comprehensive Geo-Distribution

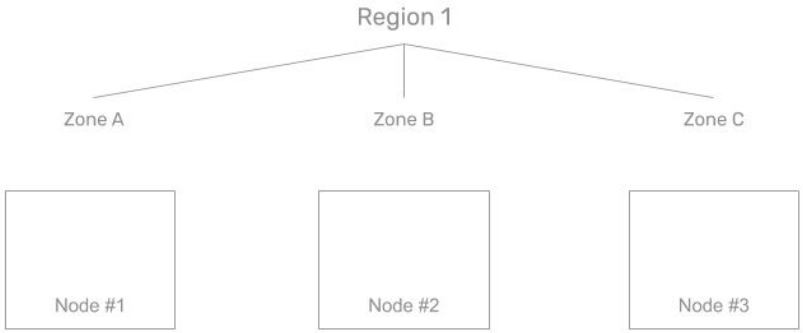
- Distribute data across zones, regions or clouds with ACID consistency
- Most complete global replication features: sync, async, geo-partitioning

Admiral distributes data across 5 regions to deliver <3ms global read latency

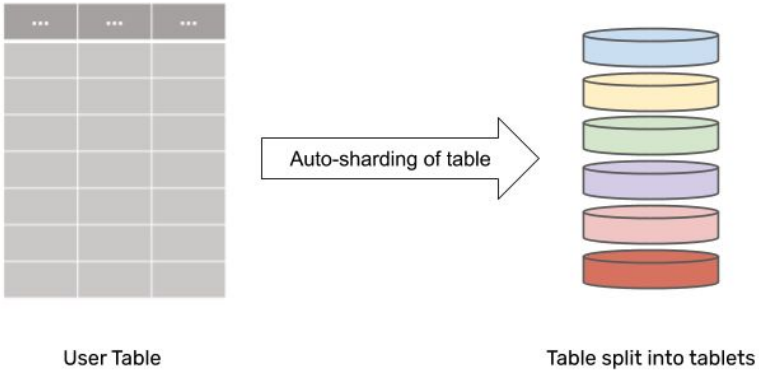
Innovative Architecture combines best of databases



Distributing Data For Horizontal Scalability



- Assume 3-nodes across zones
- How to distribute data across nodes?

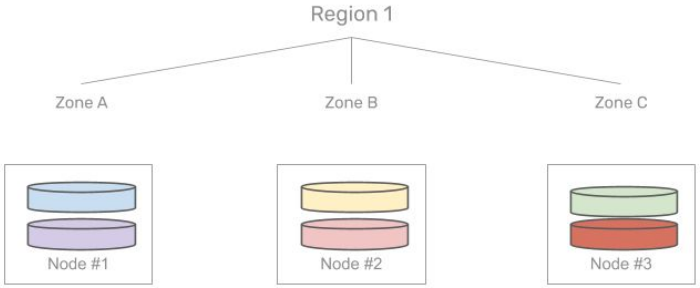


- User tables sharded into tablets
- Tablet = group of rows
- Sharding is transparent to user
  - **HASH, RANGE supported**

Deep Dive

DB Internals | Topologies | Smart Driver

Distributing Data Across Nodes, Zones, Regions

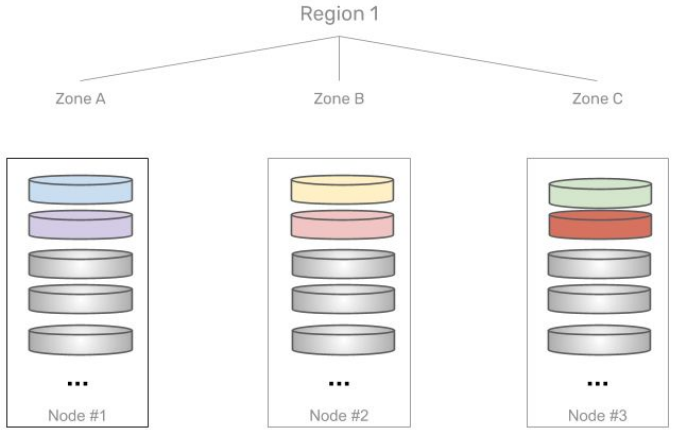


Tablets (per-table, across tables)

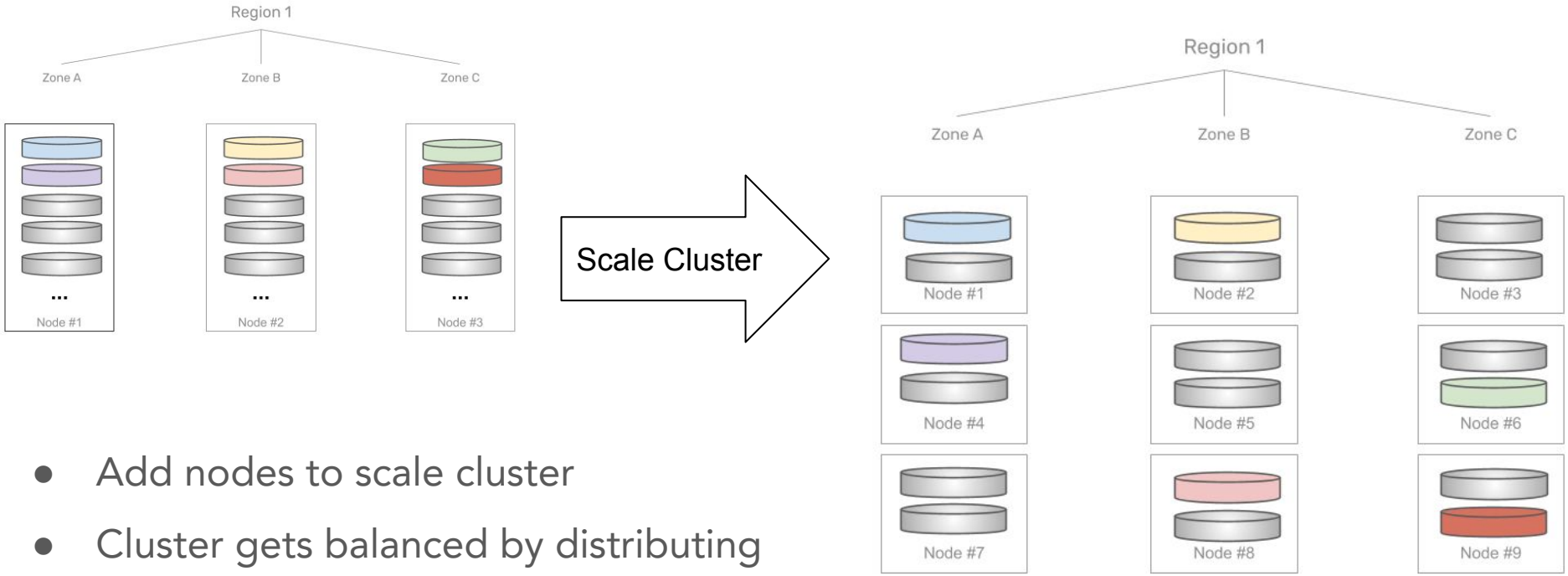
evenly distributed across nodes

In real deployments,

many tablets per node

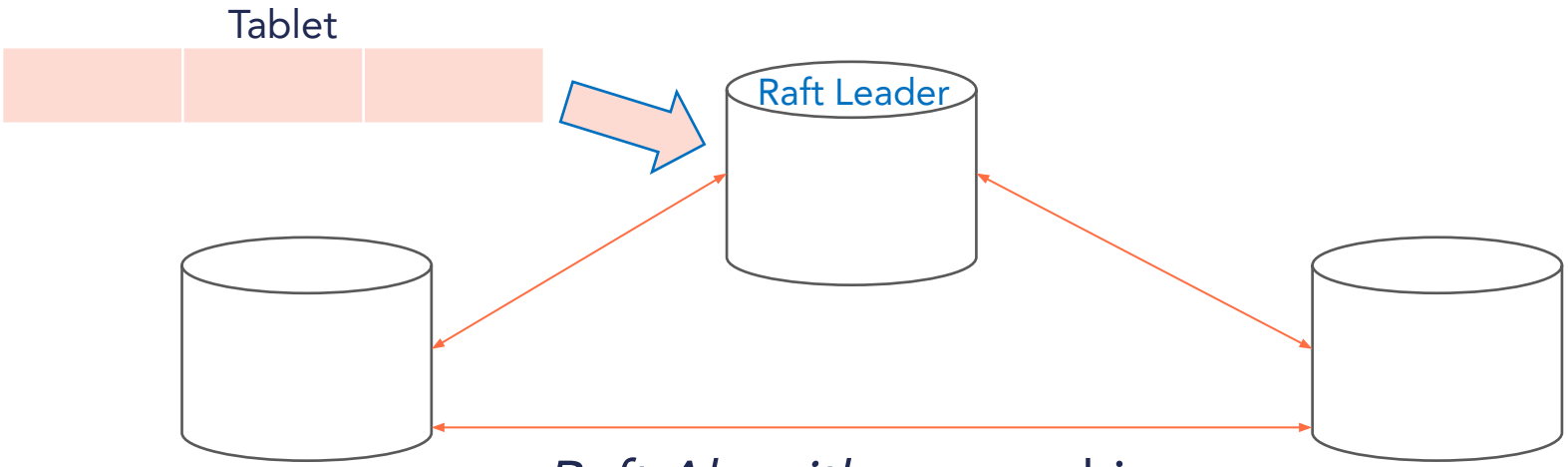


How Horizontal Scalability Works



- Add nodes to scale cluster
- Cluster gets balanced by distributing existing tablets to new nodes

High Availability - Replication uses Raft Consensus algorithm



Raft Algorithm can achieve per-row consistency across nodes

On failure, HA achieved because new leader elected quickly

Replication in a 3 node cluster

- Assume rf = 3
- Survives 1 node or zone failure
- Tablets replicated across 3 nodes
- Follower (replica) tablets balanced across nodes in cluster

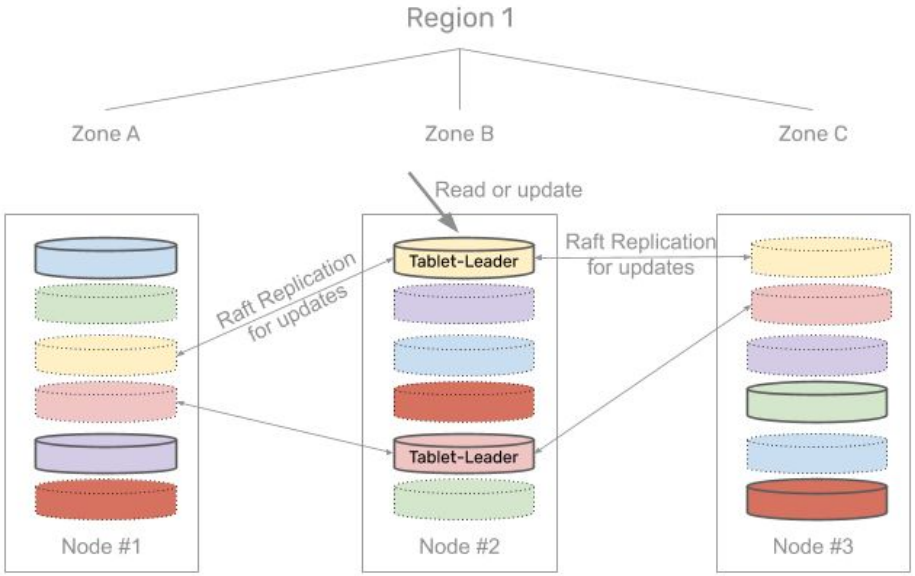
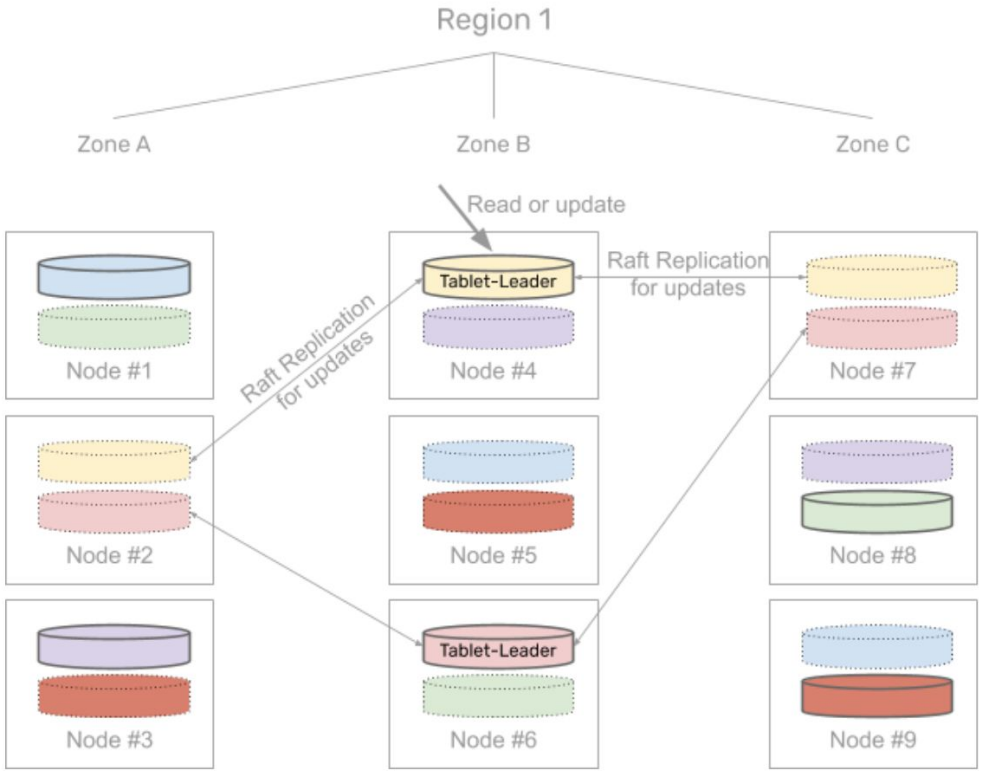


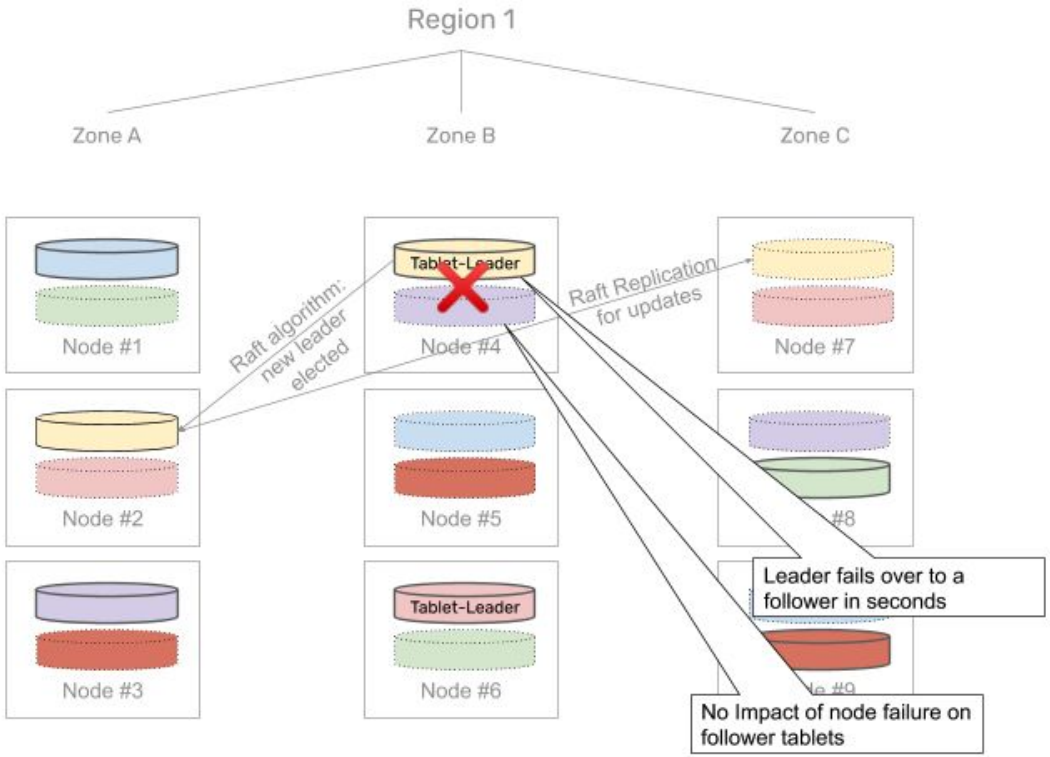
Diagram with replication factor = 3

Scaling to 9 nodes



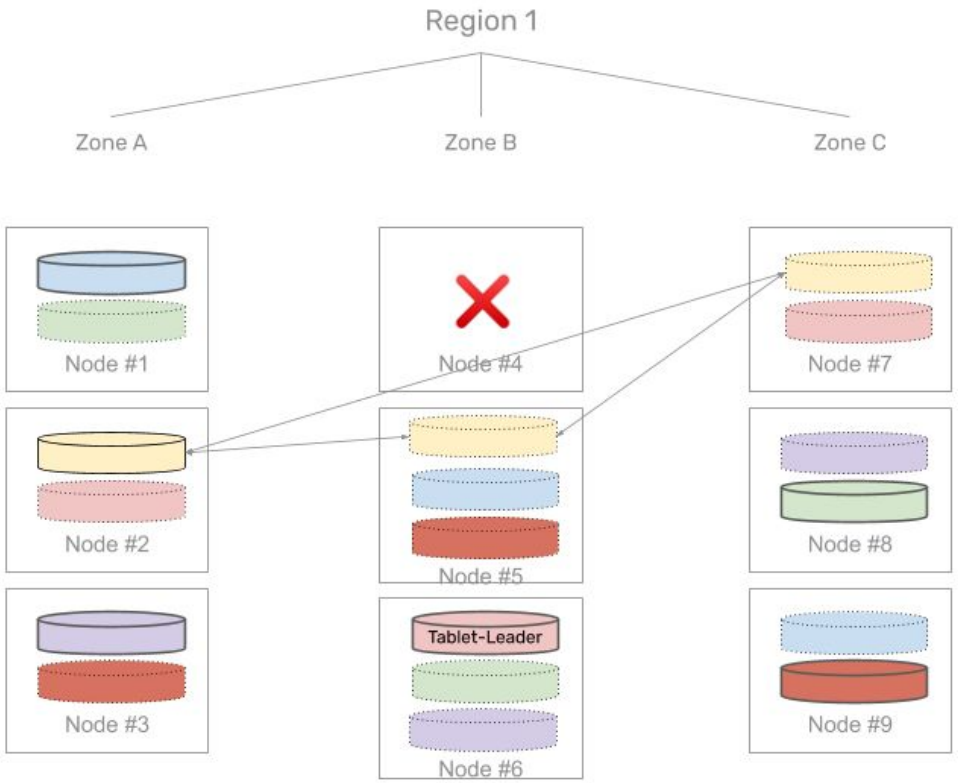
- Multi-region is similar
- 6 tablets in table
- Replication = 3
- 1 replica per zone

Tolerating Node Outage



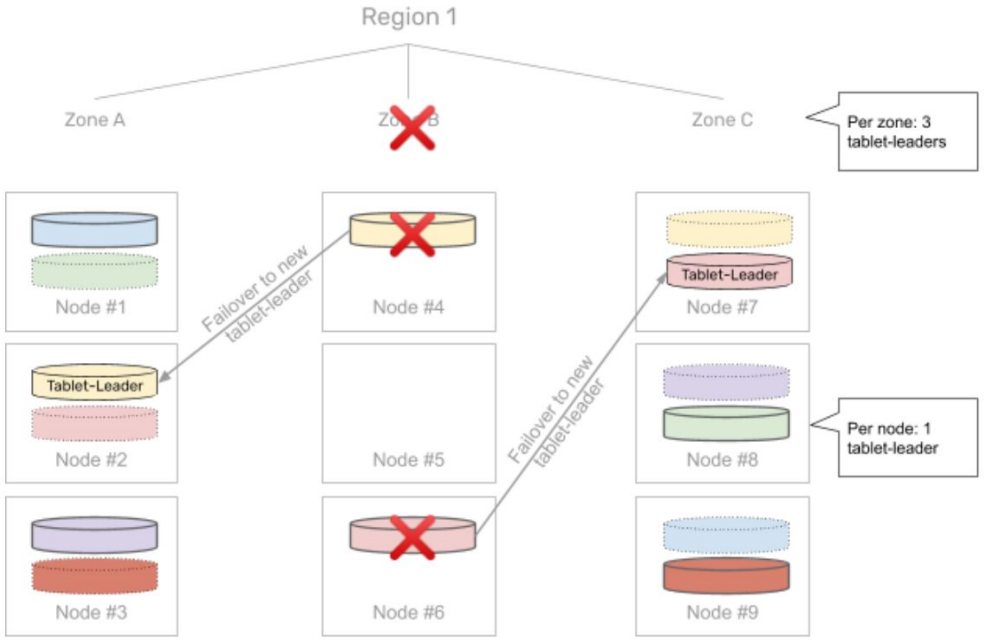
- New tablet leaders re-elected (~3 sec)
- No impact on tablet follower outage
- Follower reads ok

Automatic Resilience



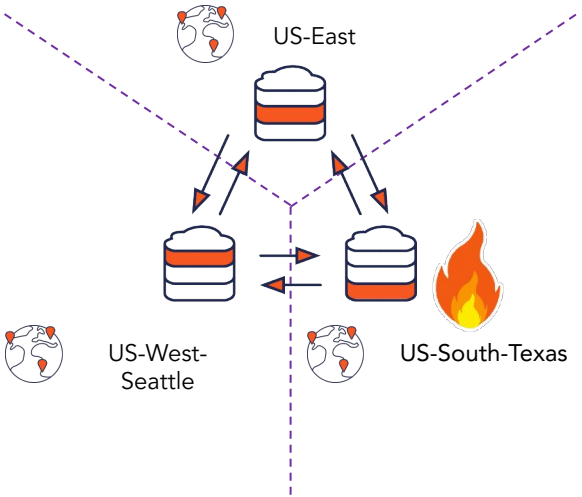
- After 15 mins, data is re-replicated (if possible)
- On failed node recovery, automatically catch up
- Tablet leaders auto-rebalanced

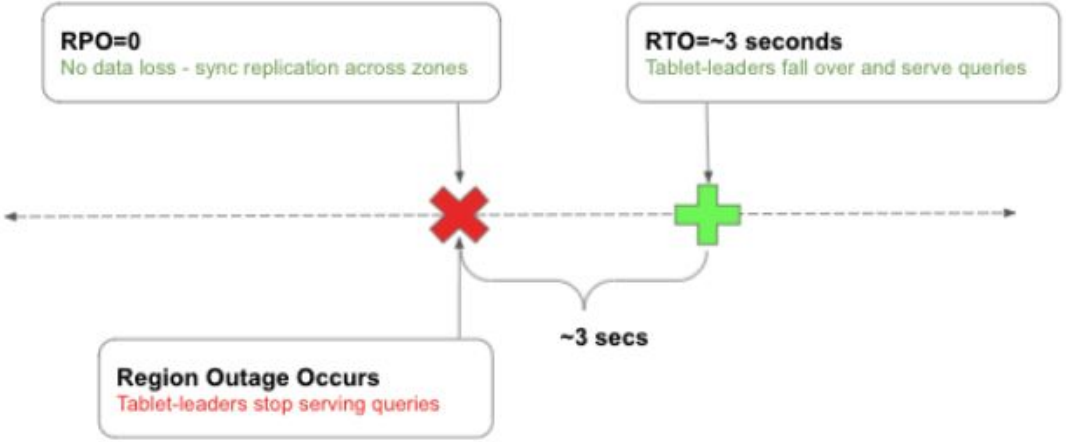
Automatic rebalancing



- New leaders evenly rebalanced
- On failed node recovery, automatically catch up

How a Fortune 500 Retailer Weathered a Regional Cloud Outage with YugabyteDB?

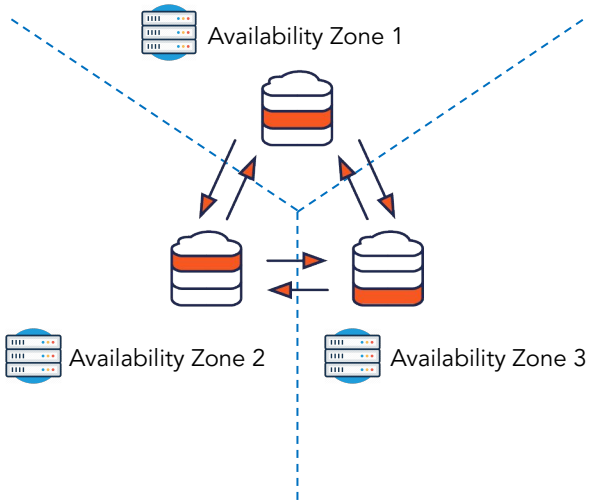




https://blog.yugabyte.com/weathering-a-regional-cloud-outage-with-yugabytedb/

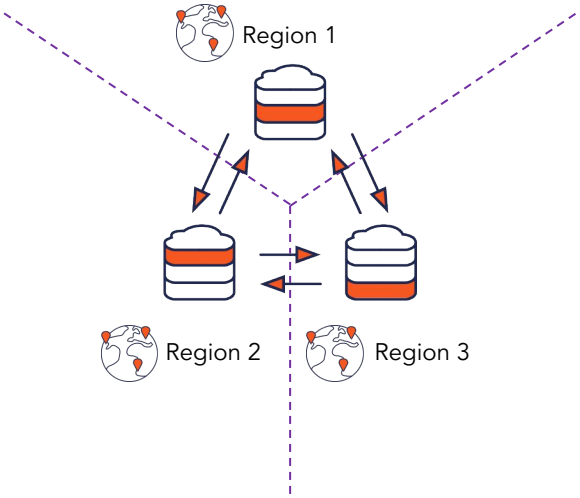
Resilient, Scalable and Strongly Consistent across failure domains

1. Single Region, Multi-Zone



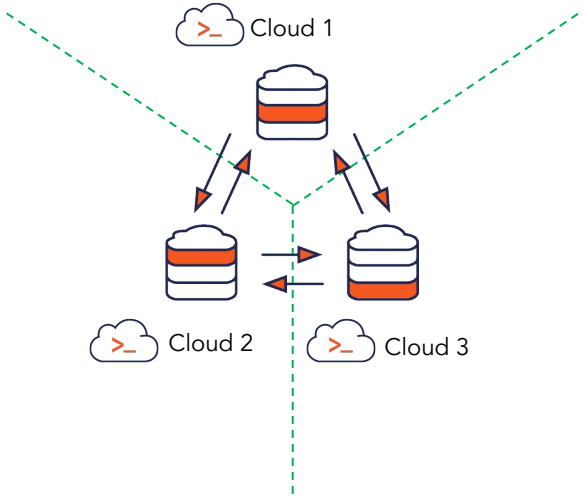
Consistent Across Zones
No WAN Latency But No
Region-Level Failover/Repair

2. Single Cloud, Multi-Region



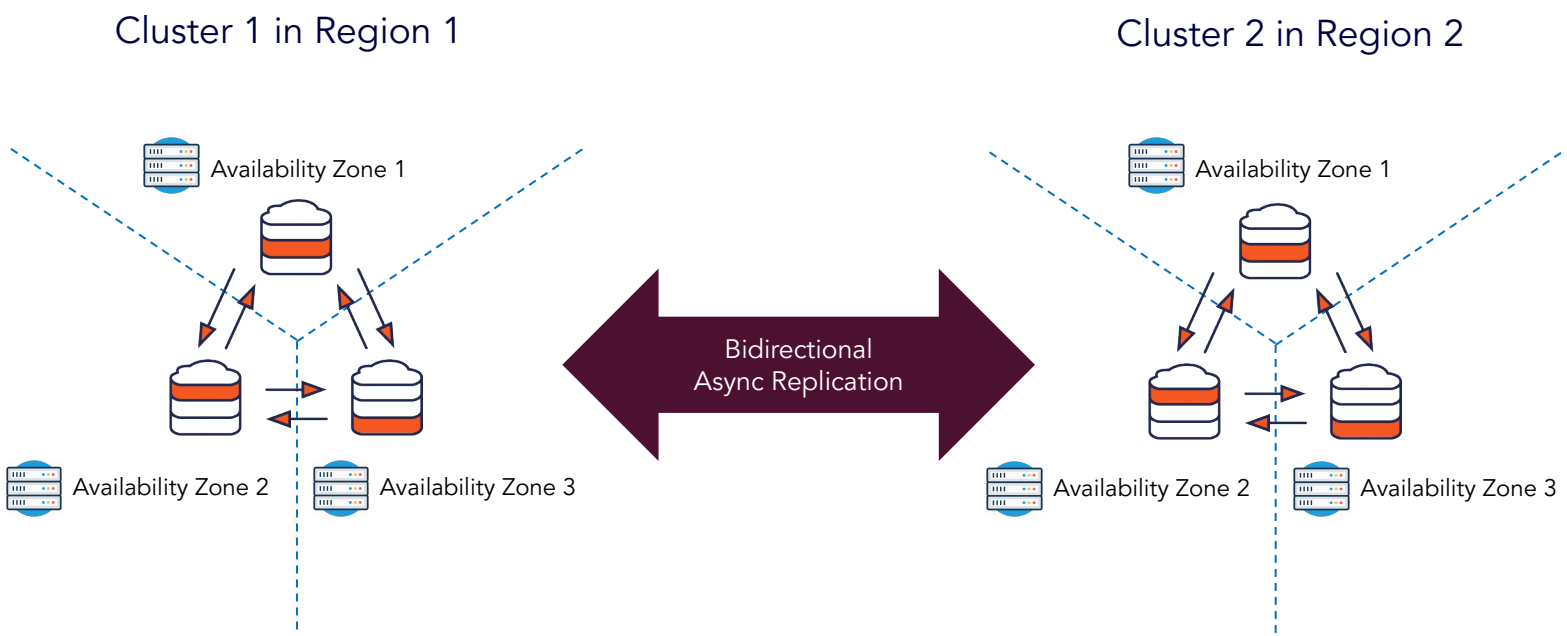
Consistent Across Regions
with Auto Region-Level
Failover/Repair

3. Multi-Cloud, Multi-Region



Consistent Across Clouds
with Auto Cloud-Level
Failover/Repair

Multi-Cluster Deployments w/ xCluster Replication



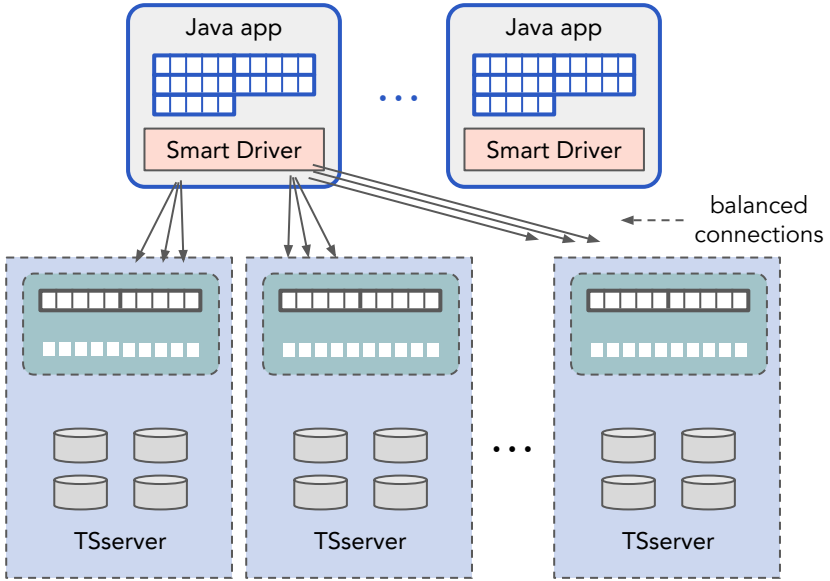
Consistent Across Zones
No Cross-Region Latency for Both Writes & Reads
App Connects to Cluster in Region 2 on Failure

Consistent Across Zones
No Cross-Region Latency for Both Writes & Reads
App Connects to Cluster in Region 1 on Failure

YB Smart Driver Overview

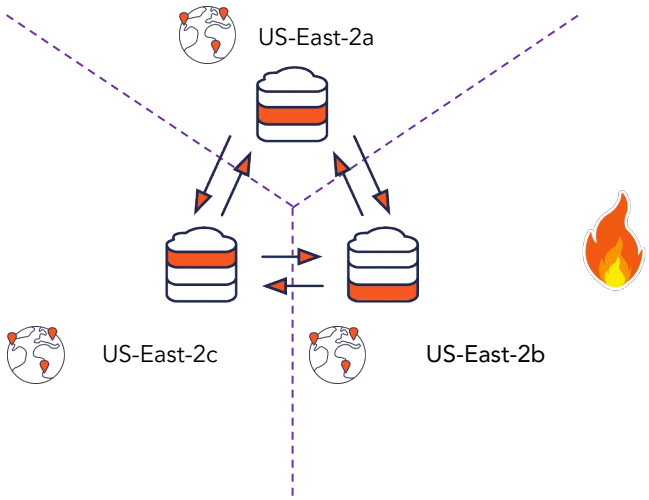
The YB Smart Driver is a distributed driver for YSQL based on the Postgres

- Cluster Awareness to eliminate need for a load balancer
- Topology Awareness to enable geo-distributed apps
- Shard Awareness for high performance (future)
- We have drivers available for most popular languages (Java, Python, Go, NodeJS, C, and many others)



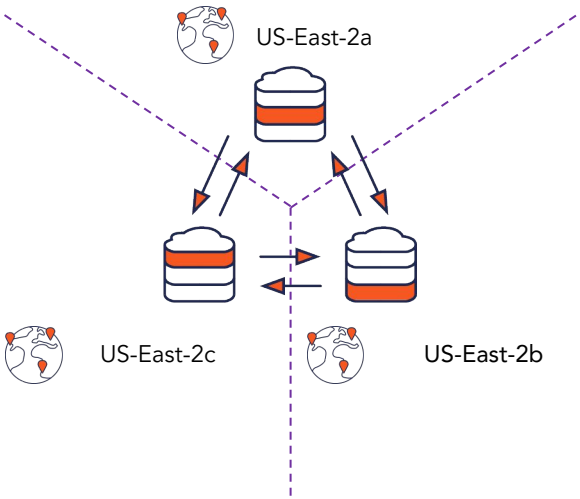
Show & Tell Demos

Live Demo: Application Zone Failure



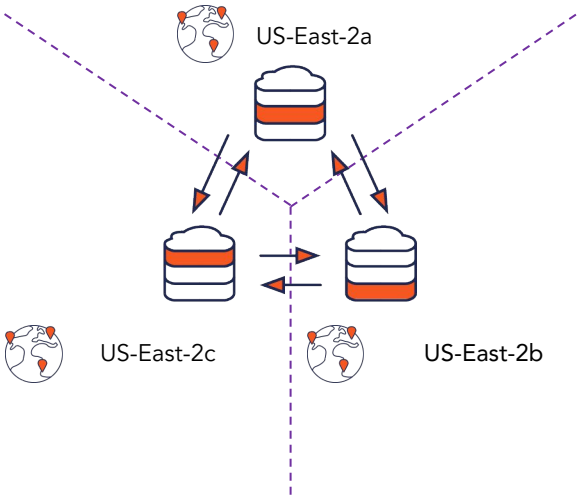
- Terminate all Nodes in a zone
- HA & Resiliency: No downtime for client apps

Live Demo: Cluster scaling



- Horizontal and Vertical scaling
- Rebalance and Reconfig - All Handsoff

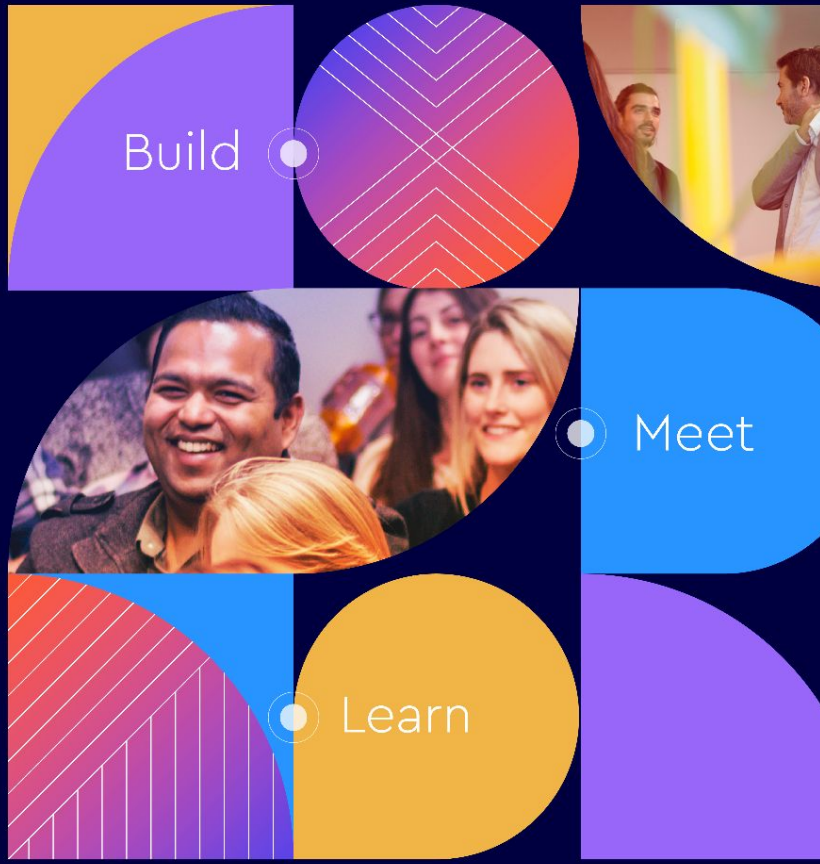
Live Demo: DB Upgrade



- Database Version Upgrade
- HA & Resiliency: No downtime for client apps

YugabyteDB Standout Features

Postgres Compatible API		Reuse PG Query Layer; Developers/ Admins are immediately productive!
100% open source; Impeccable Community Support		Releasing DB under permissive Apache v2.0 license
Multi-Cloud		Platform agnostic deployments - VMs, Kubernetes or Bare Metals, on-premise or across multiple cloud providers
Geo Distribution		Data may be spread across AZs, regions and even cloud providers while supporting GDPR/CCPA/etc
Resilience & HA		Survives multiple failures, even entire data centers
TCO		Lower cost per operation vs Oracle, DynamoDB, etc



Thank You

Join us on Slack: yugabyte.com/slack

Star us on Github: github.com/yugabyte/yugabyte-db