Availability and Resilience in YugabyteDB

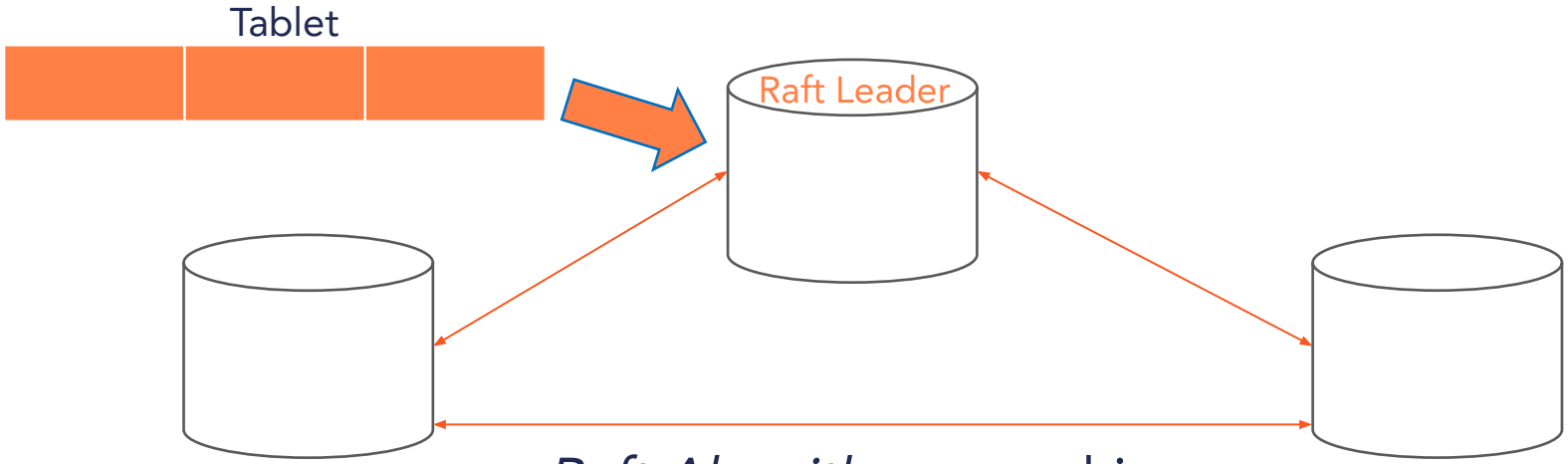
A high performance, distributed SQL DB built for the cloud

YFTT
YUGABYTEDB
FRIDAY
TECH TALKS

yugabyteDB

Achieving HA and Resilience with Consistency

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

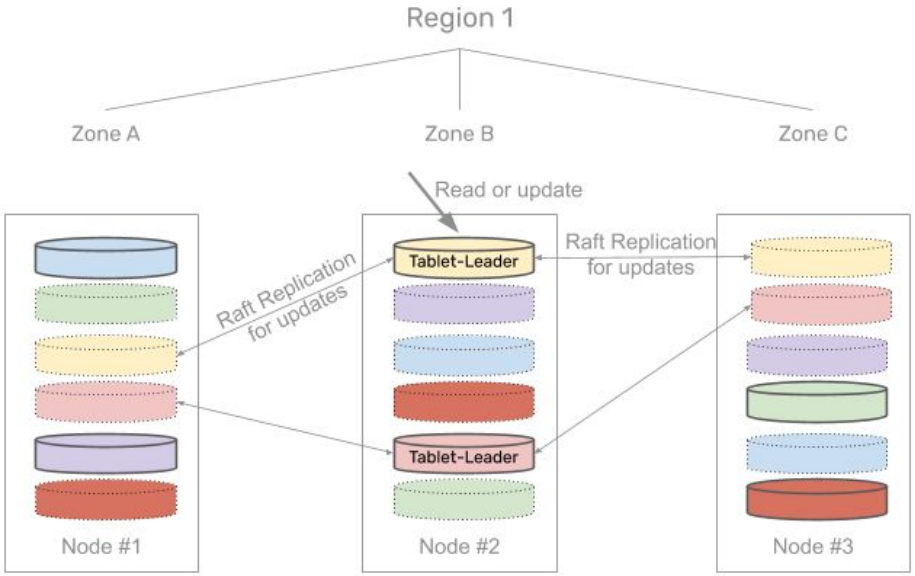
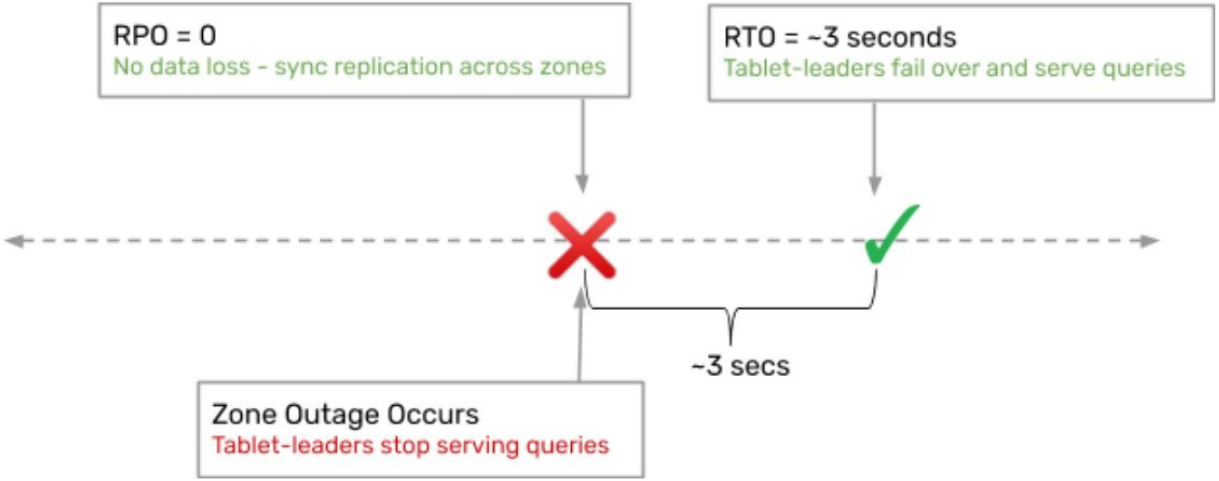


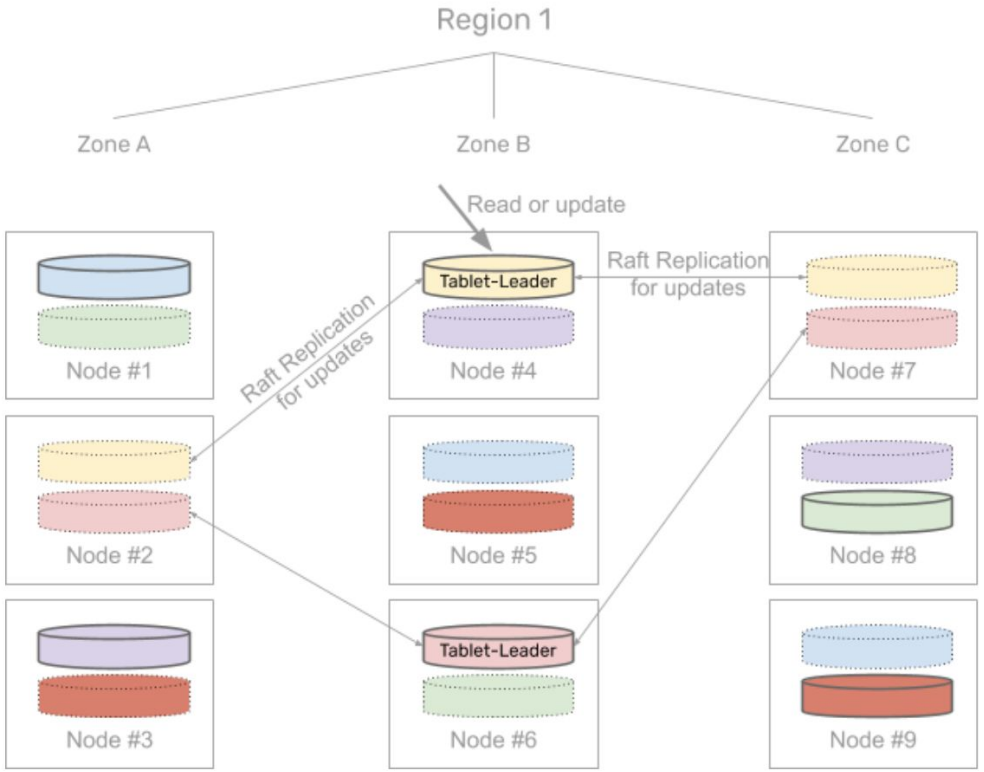
Diagram with replication factor = 3

How long to failover in multi-zone setup?



Typical Multi-AZ Cluster

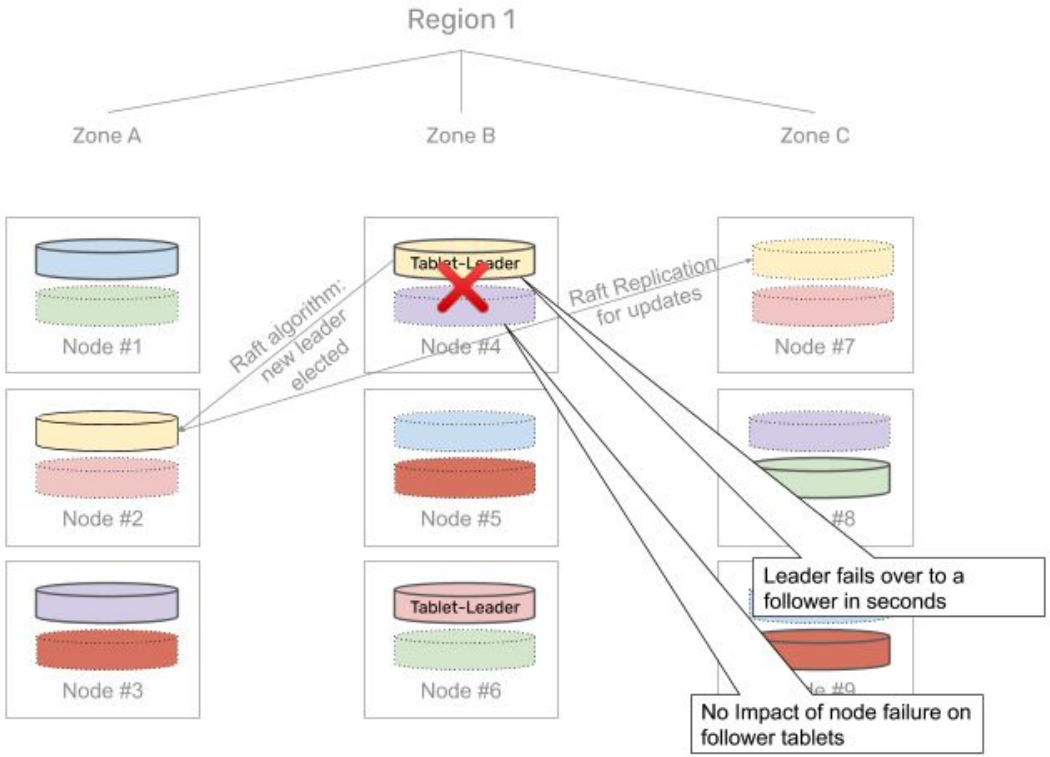
Scaling to 9 nodes



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

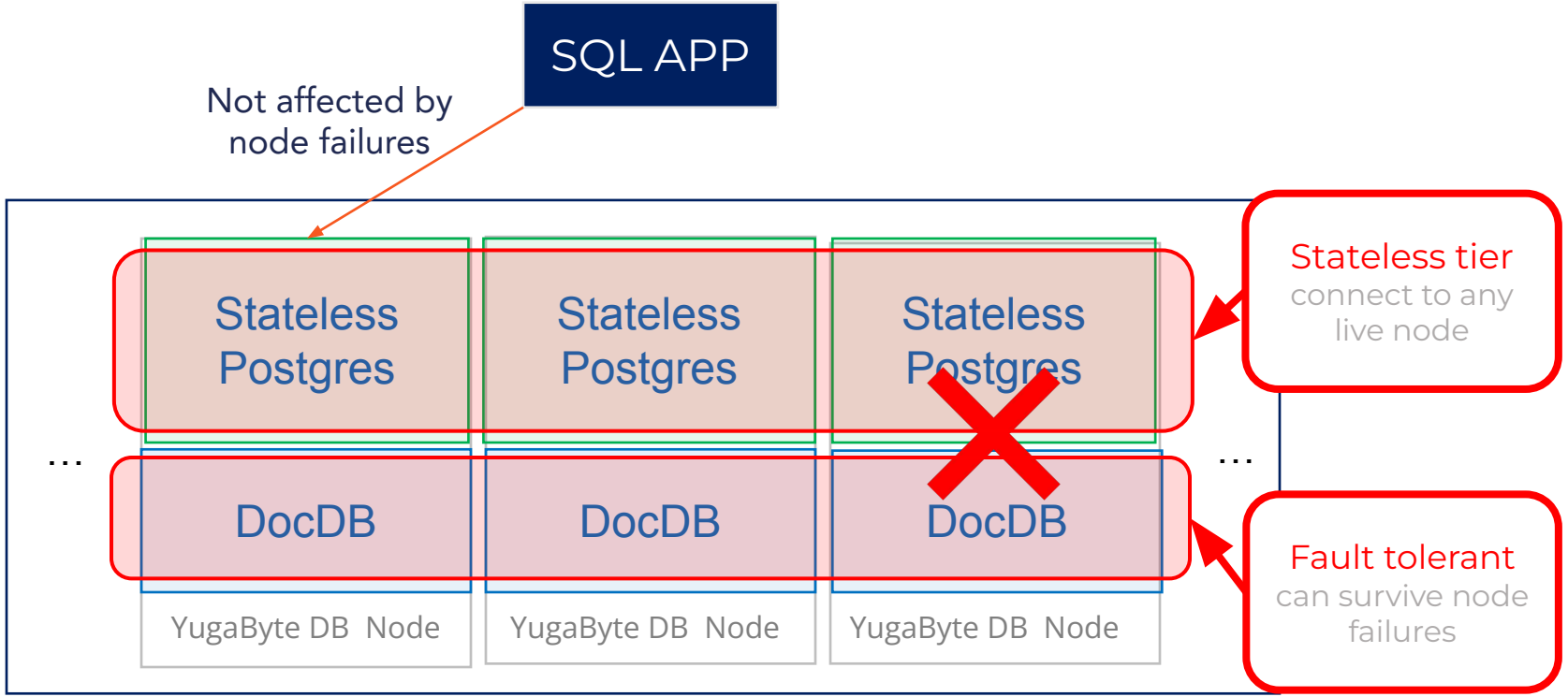
HA in Practice

Tolerating node outage

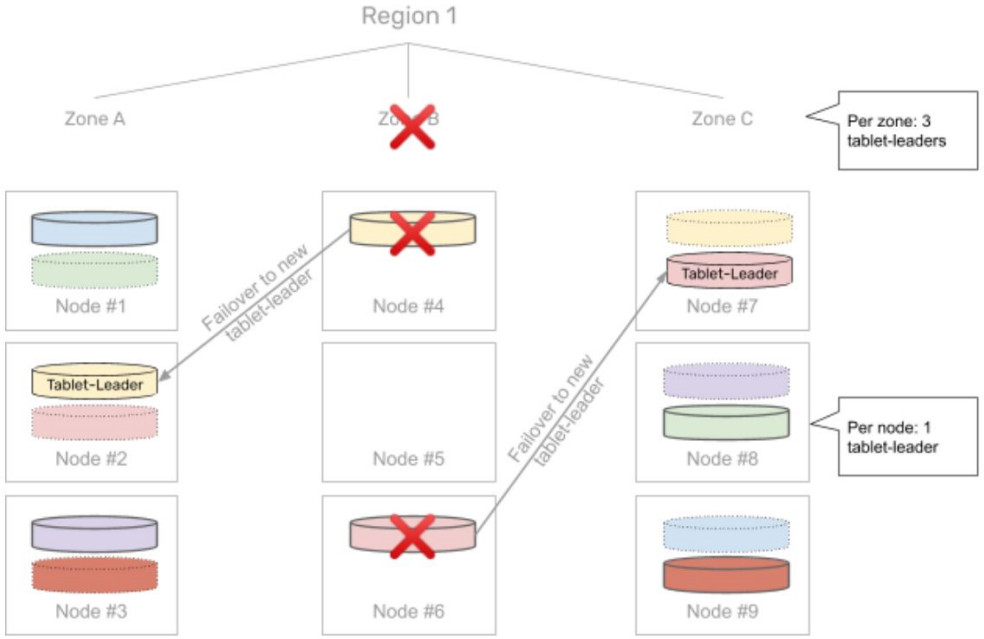


- New tablet leaders re-elected (~3 sec)
- No impact on tablet follower outage
- DB unavailable during re-election window

Self-Healing against failures



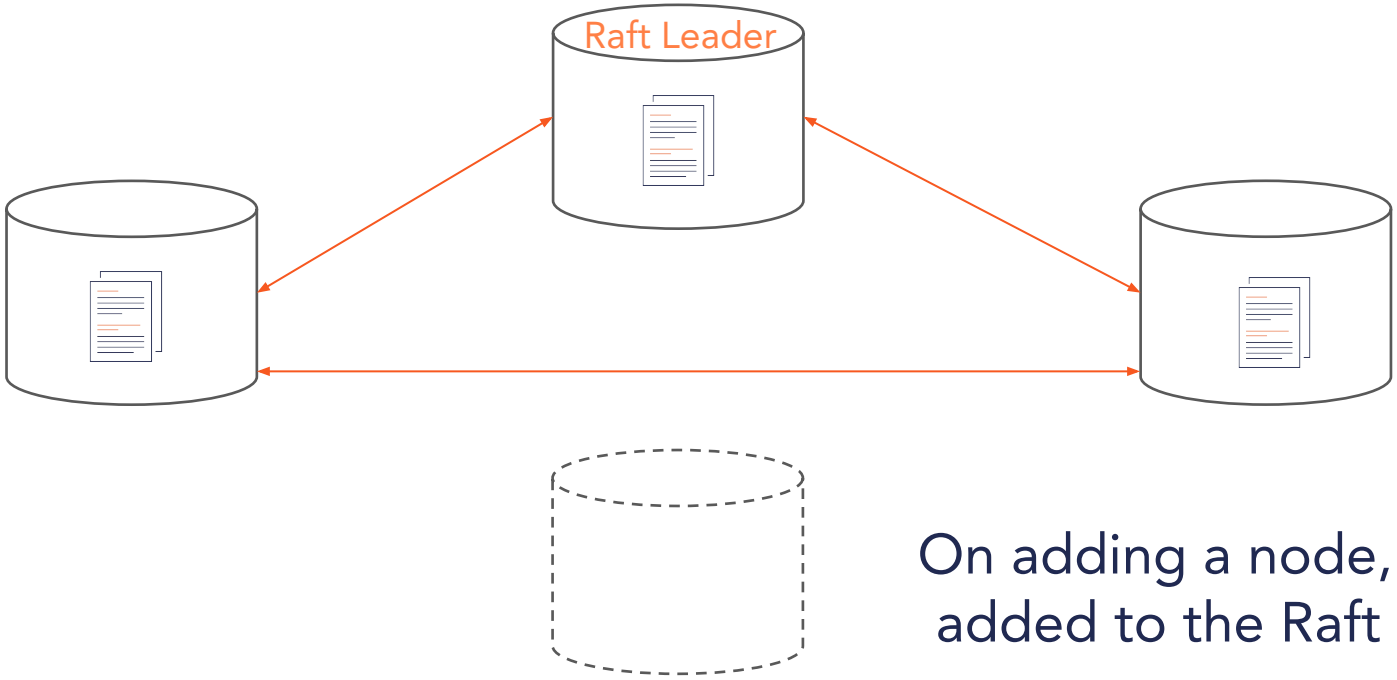
Automatic rebalancing



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

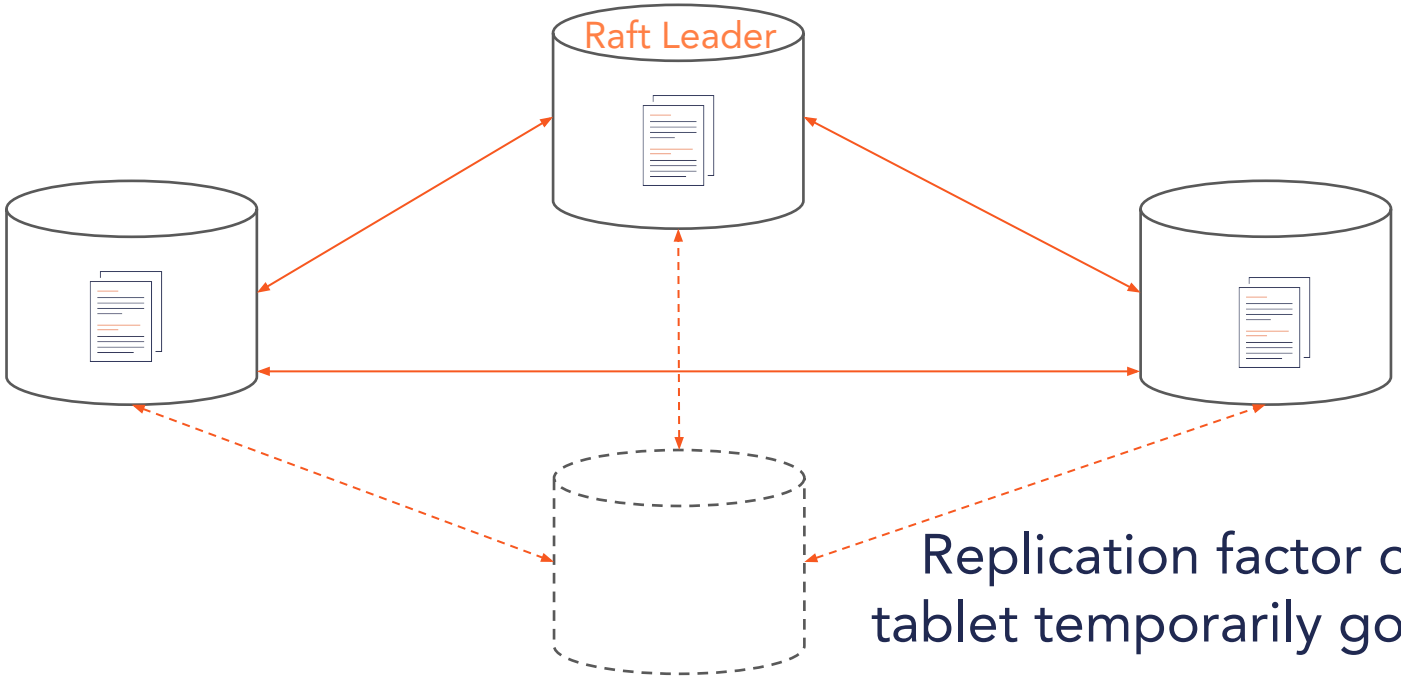
Planned Upgrades and Node Maintenance

Dynamic membership changes



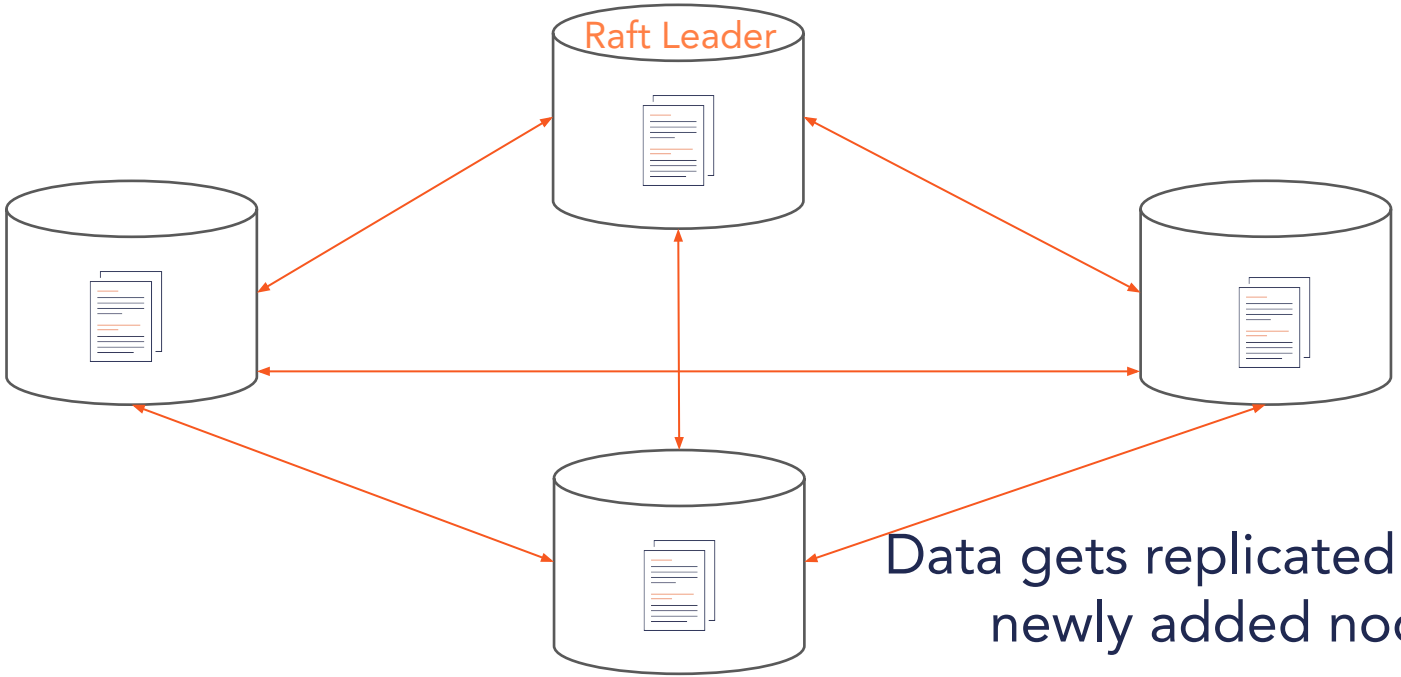
On adding a node, it gets added to the Raft group

Adding a Replica



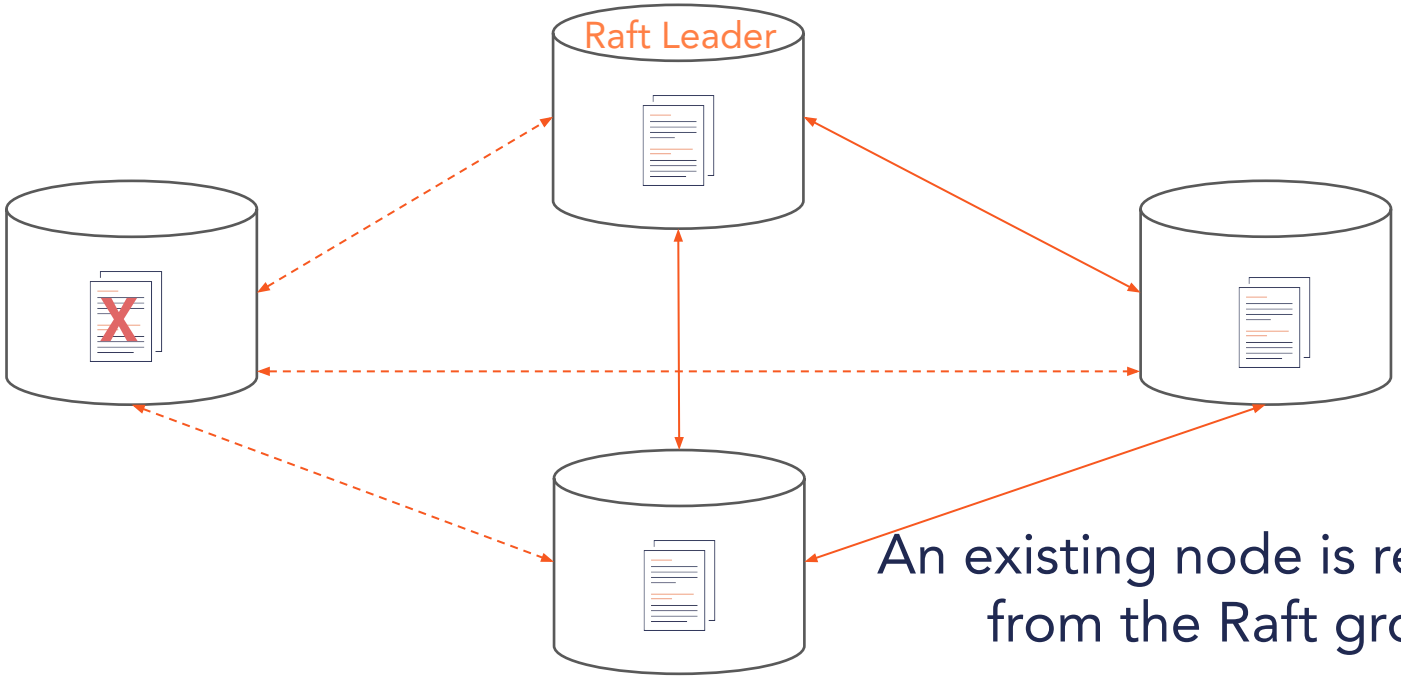
Replication factor of the tablet temporarily goes to 4

Adding a Replica



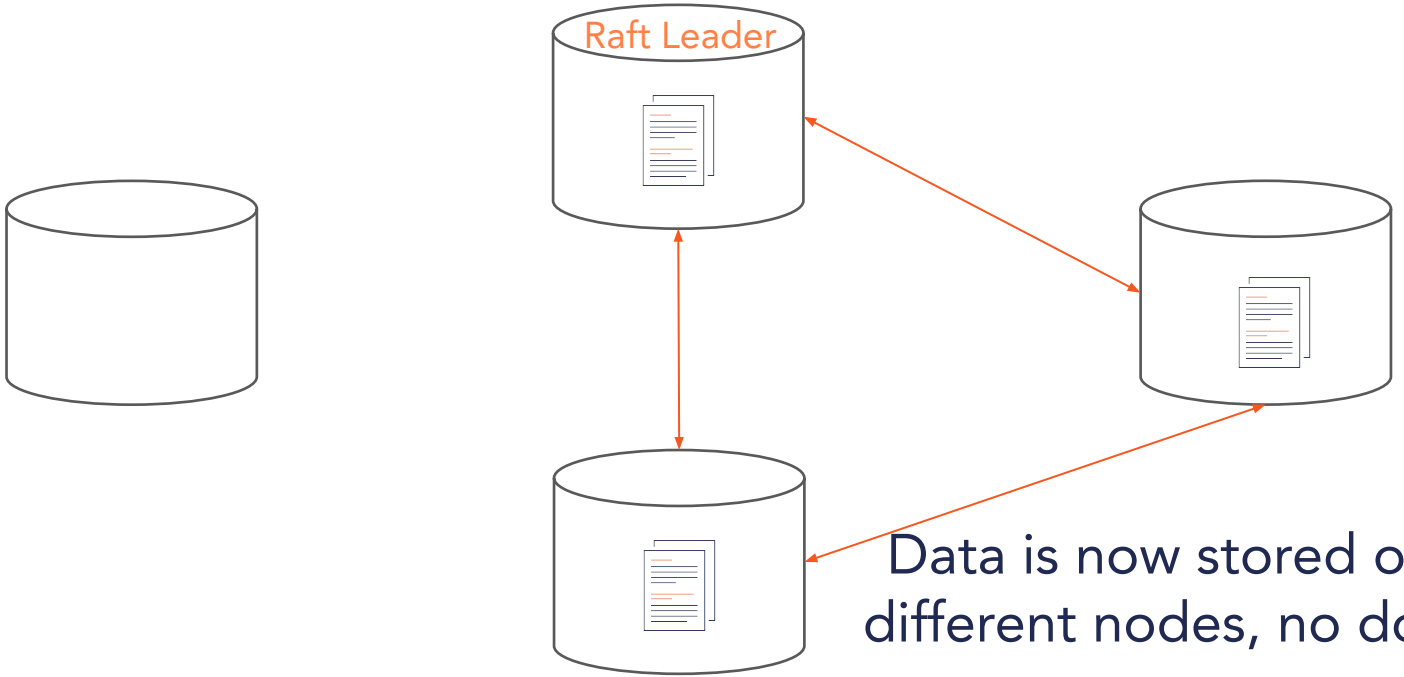
Data gets replicated to the newly added node

Removing a Replica



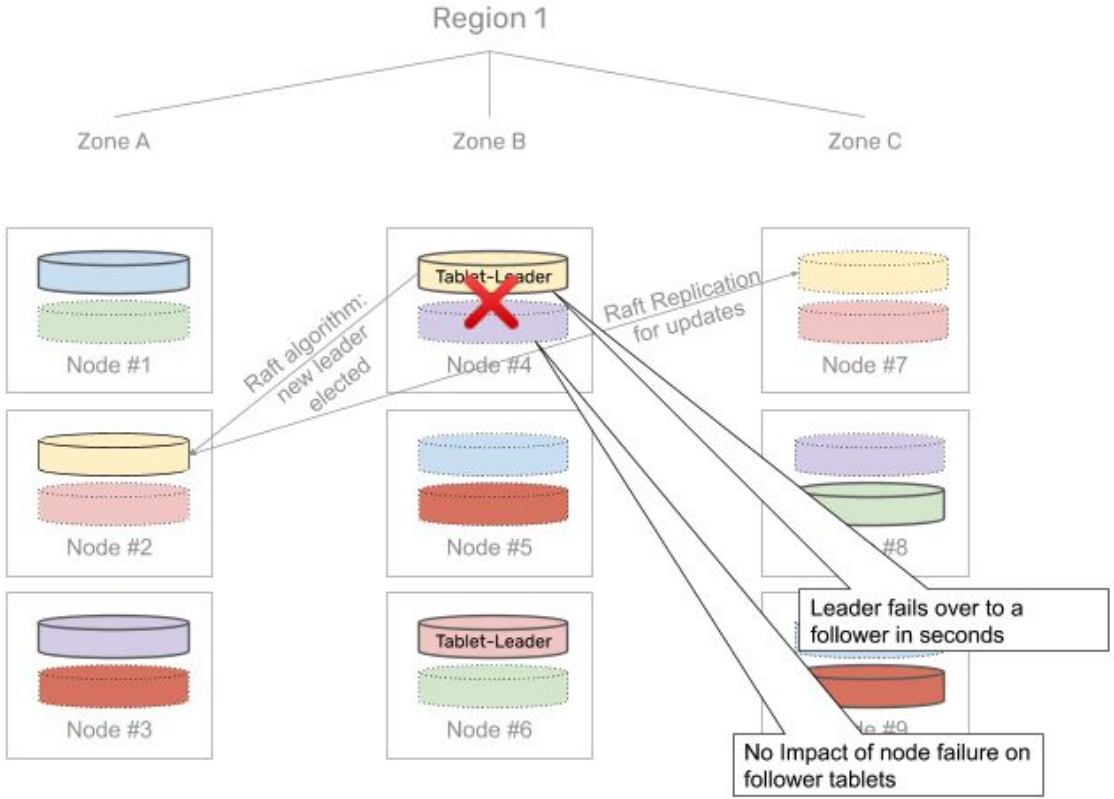
An existing node is removed from the Raft group

Decommissioning a node

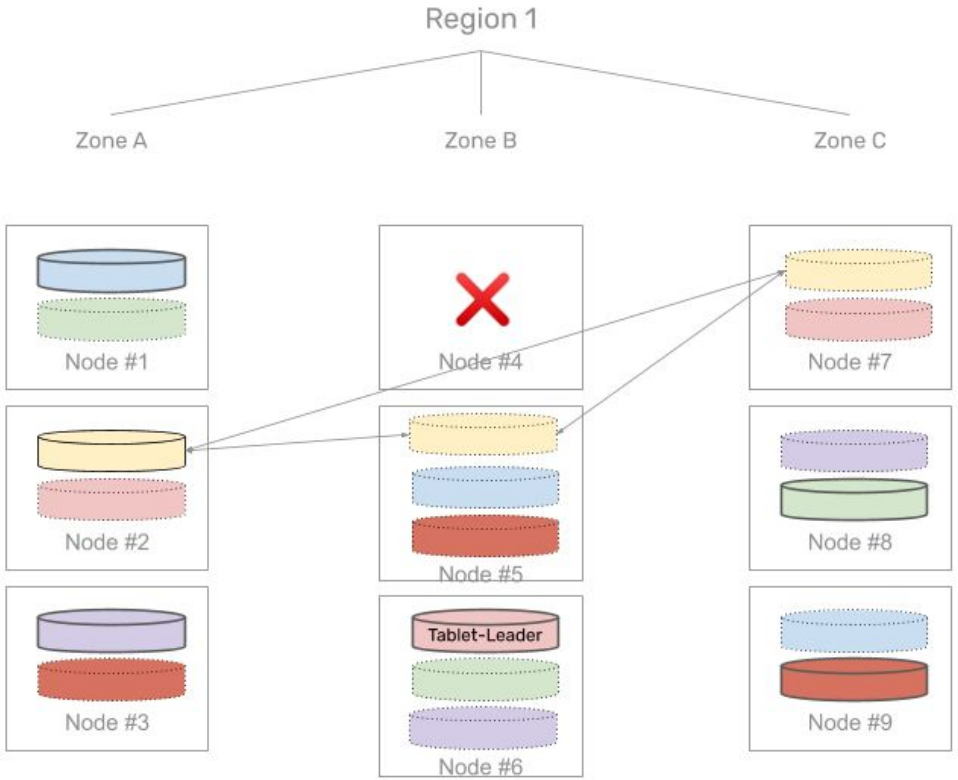


Resilience in Practice

Tolerating node outage



Automatic Resilience



- After 15 mins, data is re-replicated (if possible)

Thank You

Join us on Slack: yugabyte.com/slack (#yftt channel)

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

YFTT YUGABYTEDB FRIDAY TECH TALKS

yugabyteDB