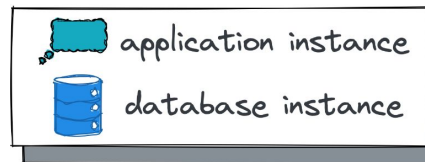
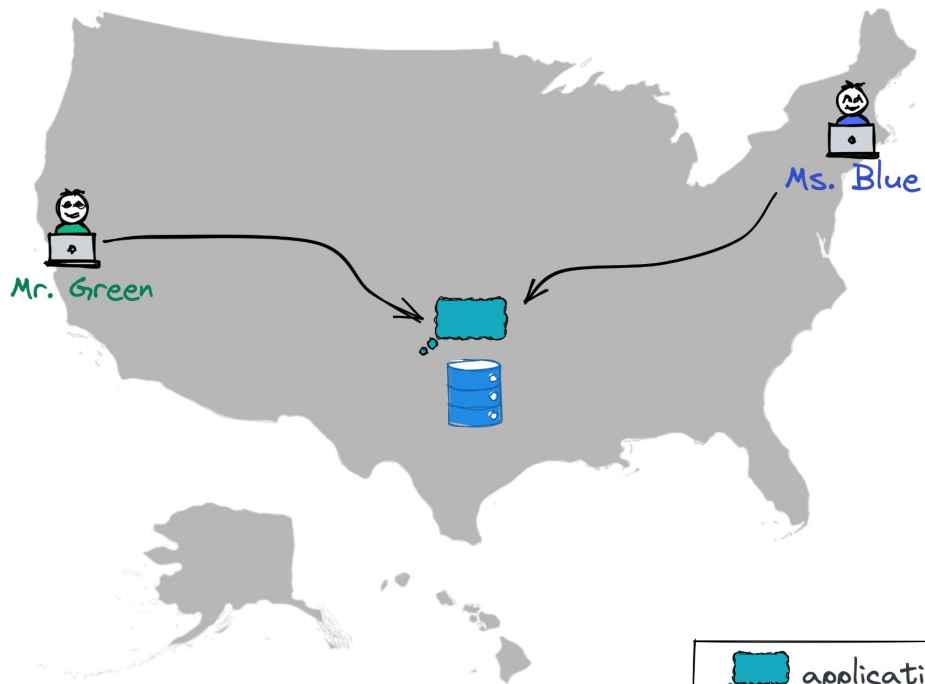


# Weathering The Cloud Storm: PostgreSQL High Availability Options

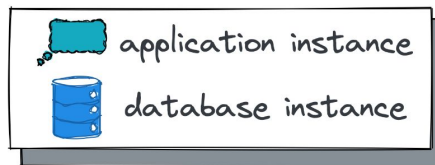
# The first release of an application

---



# And the first outage in the cloud

---



# Starting 2011, AWS alone had a major outage once a year

## AWS Post-Event Summaries

AWS Trusted Advisor Priority | Get visibility into critical optimization recommendations curated by your AWS account team »

### AWS Post-Event Summaries

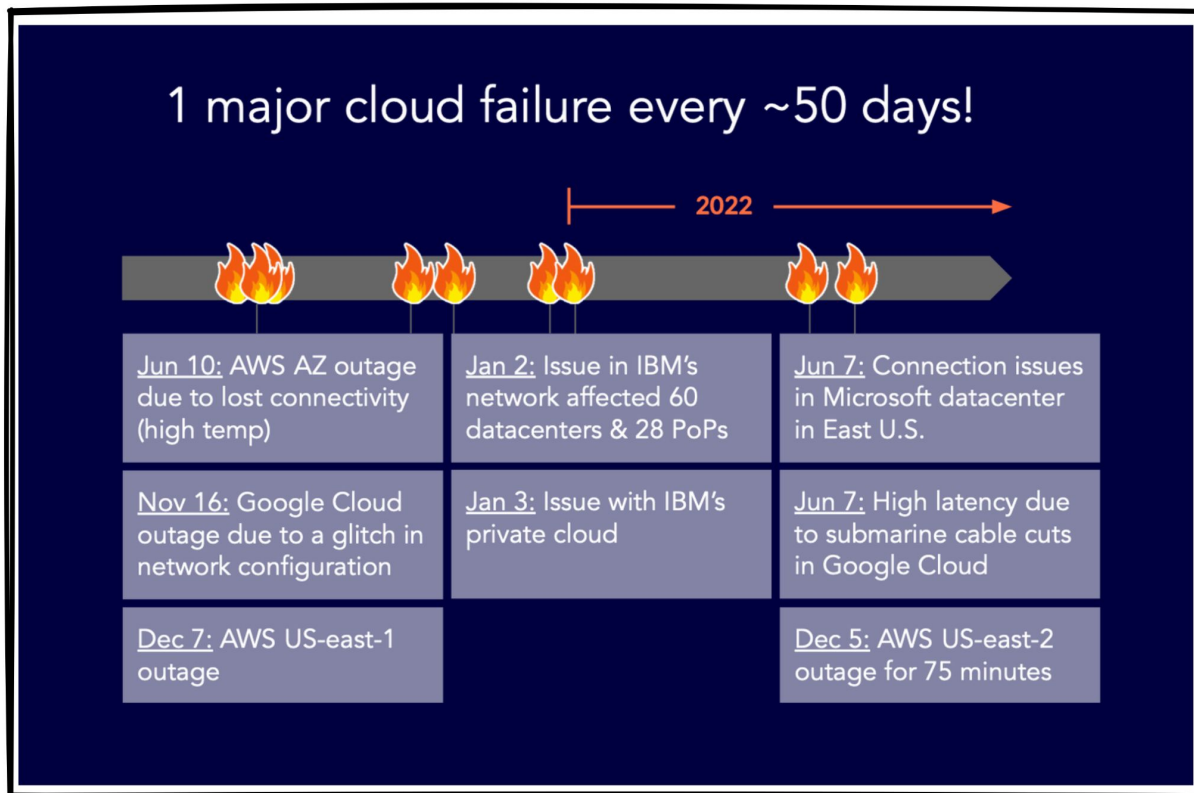
The following is a list of post-event summaries from major service events that impacted AWS service availability. AWS is committed to publishing post-event summaries for events that result in high error rates for an extended period of time across a broad set of AWS Services and customers in a Region:

- [Summary of the AWS Service Event in the Northern Virginia \(US-EAST-1\) Region, December, 10th 2021](#)
- [Summary of AWS Direct Connect Event in the Tokyo \(AP-NORTHEAST-1\) Region, September, 2nd 2021](#)
- [Summary of the Amazon Kinesis Event in the Northern Virginia \(US-EAST-1\) Region, November, 25th 2020](#)
- [Summary of the Amazon EC2 and Amazon EBS Service Event in the Tokyo \(AP-NORTHEAST-1\) Region, August 23, 2019](#)
- [Summary of the Amazon EC2 DNS Resolution Issues in the Asia Pacific \(Seoul\) Region \(AP-NORTHEAST-2\), November 24, 2018.](#)
- [Summary of the Amazon S3 Service Disruption in the Northern Virginia \(US-EAST-1\) Region, February 28, 2017.](#)
- [Summary of the AWS Service Event in the Sydney Region, June 8, 2016.](#)
- [Summary of the Amazon DynamoDB Service Disruption and Related Impacts in the US-East Region, September 20, 2015.](#)
- [Summary of the Amazon EC2, Amazon EBS, and Amazon RDS Service Event in the EU West Region, August 7, 2014.](#)
- [Summary of the Amazon SimpleDB Service Disruption, June 13, 2014.](#)
- [Summary of the December 17th event in the South America Region \(SA-EAST-1\), December 20, 2013.](#)
- [Summary of the December 24, 2012 Amazon ELB Service Event in the US-East Region, December 24, 2012.](#)
- [Summary of the October 22, 2012 AWS Service Event in the US-East Region, October 22, 2012.](#)
- [Summary of the AWS Service Event in the US East Region, July 2, 2012.](#)
- [Summary of the Amazon EC2 and Amazon RDS Service Disruption in the US East Region, April 29, 2011.](#)

<https://aws.amazon.com/premiumsupport/technology/pes/>



# That's not just about AWS...



# Club Rules

---

## Rule 1

Accept the possibility that  
your deployment can be impacted



# Club Rules

---

## Rule 1

Accept the possibility that your deployment can be impacted



## Rule 2

Develop & deploy with high availability in mind



For OtterTune's customers,  
70% of production Aurora  
clusters have read replicas.



Andy Pavlo

<https://ottertune.com/blog/why-the-faas-database-problem-wont-happen-in-aws/>

But only 32% of production non-Aurora RDS instances are replicating to at least one standby instance.



Andy Pavlo

<https://ottertune.com/blog/why-the-faas-database-problem-wont-happen-in-aws/>

# Club Rules

---

## Rule 1

Accept the possibility that your deployment can be impacted



## Rule 2

Develop & deploy with high availability in mind



# There are many ways to approach HA in PostgreSQL

---

1. Single database instance
2. Single primary instance with read replicas
3. Multi-master (sharding with a coordinator)
4. Multi-master (sharding without a coordinator)





## High Availability



the ability of a system to operate continuously by eliminating a single point of failure



## Disaster Recovery



is the process of getting a system back to an operational state



# Availability SLA

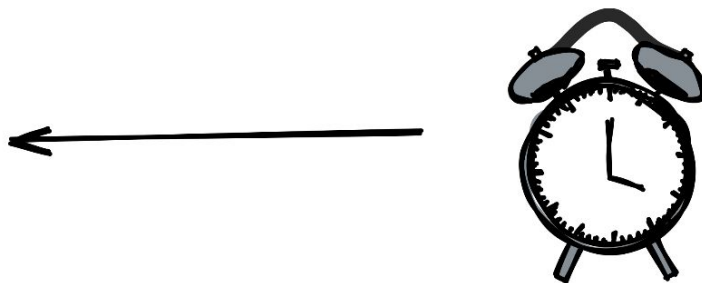
(aka. availability objective or the nines of availability)

Uptime	Downtime per year
99.9%	8.76 hours
99.99%	56.2 mins
99.999%	5.25 mins
99.9999%	31.56 secs

# Availability SLA

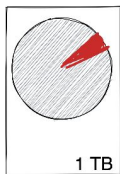
Uptime	Downtime per year
99.9%	8.76 hours
99.99%	56.2 mins
99.999%	5.25 mins
99.9999%	31.56 secs

## Recovery Time Objective (RTO)



the duration of time within which the system must be restored

## Recovery Point Objective (RPO)



the maximum acceptable amount of data loss after an incident

## Availability SLA

Uptime	Downtime per year
99.9%	8.76 hours
99.99%	56.2 mins
99.999%	5.25 mins
99.9999%	31.56 secs

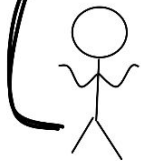
## Recovery Time Objective (RTO)



the duration of time within which the system must be restored

# Questions to Answer

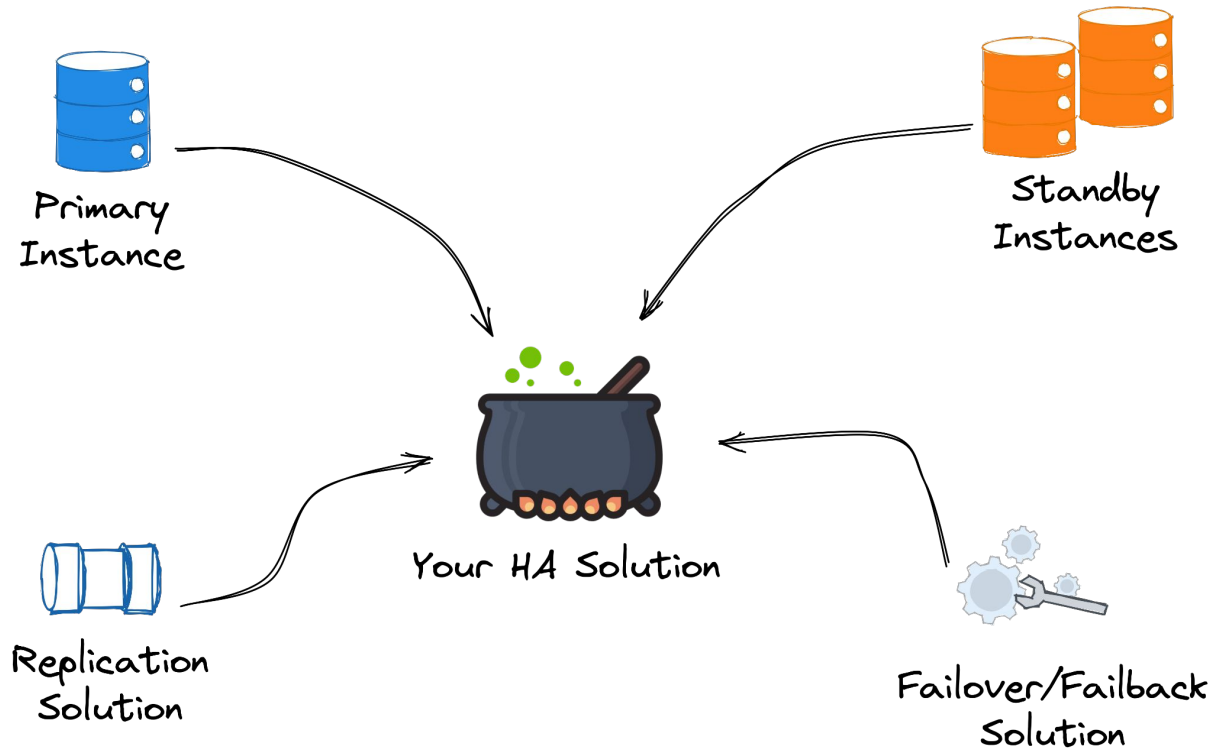
1. How is data important to you?
2. Do you have any predefined SLAs?
3. ???



## Your HA Objectives

- \* The Nines
- \* RPO
- \* RTO

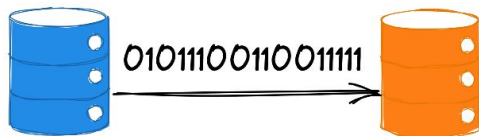
# High Availability With Single Database Instance





## A few popular built-in replication solutions

Streaming replication  
(log shipping)

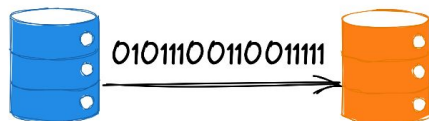


- \* replicates the entire database
- \* primary & standby configurations need to be similar



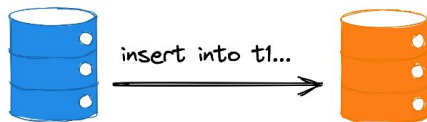
## A few popular built-in replication solutions

Streaming replication  
(log shipping)



- \* replicates the entire database
- \* primary & standby configurations need to be similar

Logical replication  
(log decoding)



- \* can replicate specific tables
- \* primary & standby can have different configs



# Asynchronous or Synchronous Replication

## synchronous\_commit modes

mode	primary durable commit	standby durable commit after PG crash	standby durable commit after OS crash	standby query consistency
remote_apply	✓	✓	✓	✓
on	✓	✓	✓	
remote_write	✓	✓		
local	✓			
off				





## Asynchronous or Synchronous Replication

synchronous\_commit modes

durability



performance

mode	primary durable commit	standby durable commit after PG crash	standby durable commit after OS crash	standby query consistency
remote_apply	✓	✓	✓	✓
on	✓	✓	✓	
remote_write	✓	✓		
local	✓			
off				



## A few popular failover & failback solutions



Pgpool

(failover, connection pooling,  
load balancing)

- \* custom scripts for failover and failback events
- \* manual intervention can be necessary



## A few popular failover & failback solutions



Pgpool

(failover, connection pooling,  
load balancing)

- \* custom scripts for failover and failback events
- \* manual intervention can be necessary



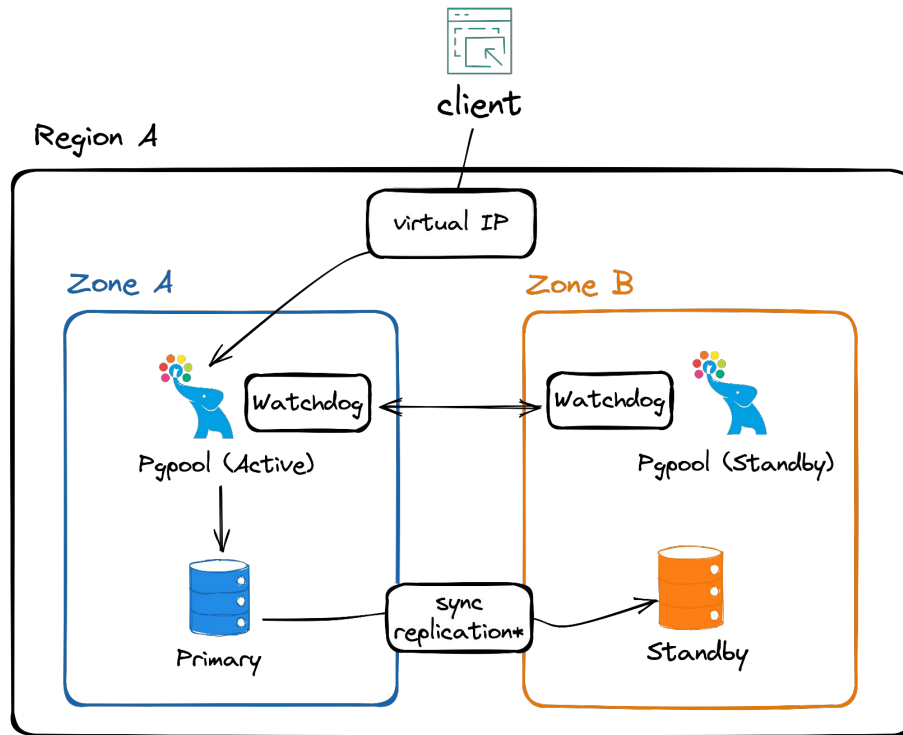
Patroni

(failover and failback)

- \* makes things more automated and flexible
- \* uses etcd (or alternatives) for quorum and split-brain resolution



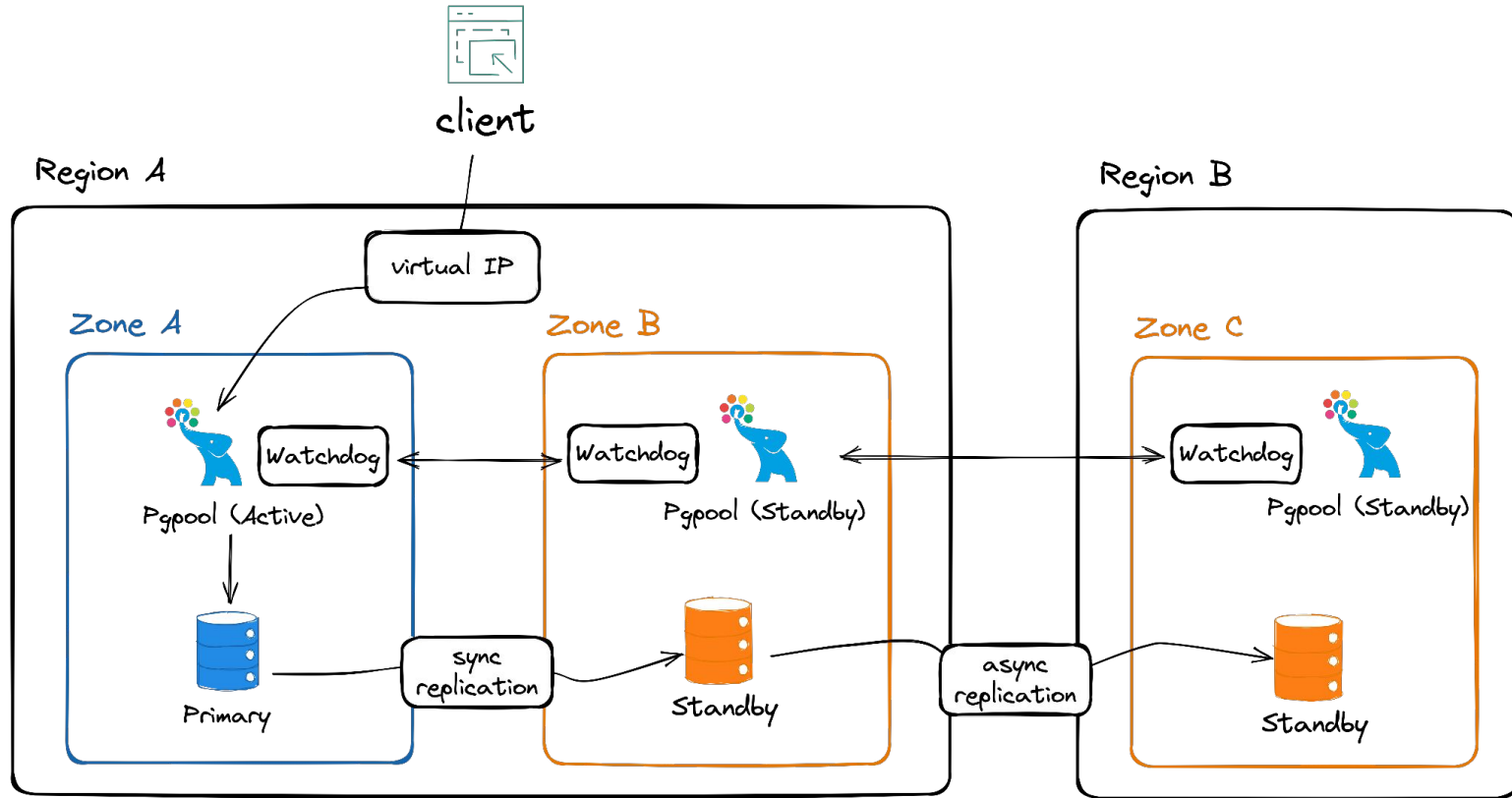
## Sample Configuration With Single Standby



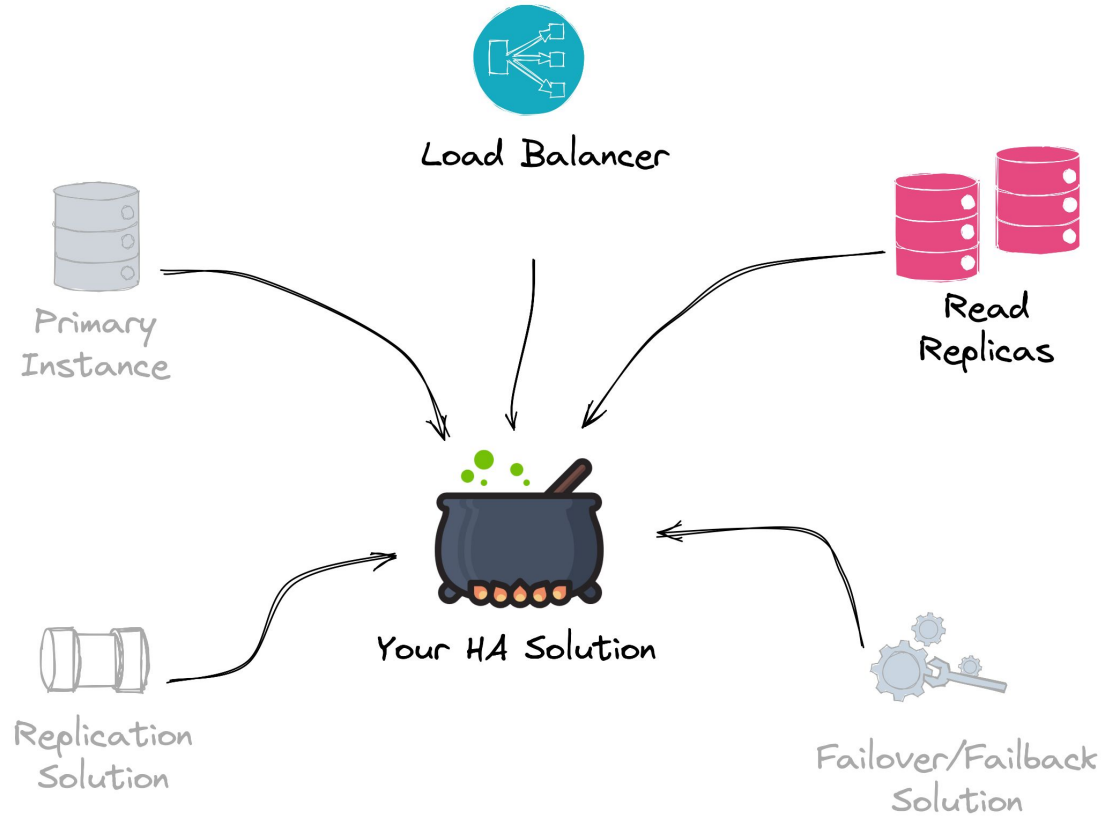
**sync replication\*** - the primary can't commit if the standby is down. Use 2+ standbys.



# Sample Multi-Region Configuration With Two Standbys



# When you need to scale reads





## A few popular load balancing solutions



Pgpool

(failover, connection pooling,  
load balancing)



HAPROXY

(load balancing for  
TCP and HTTP traffic)

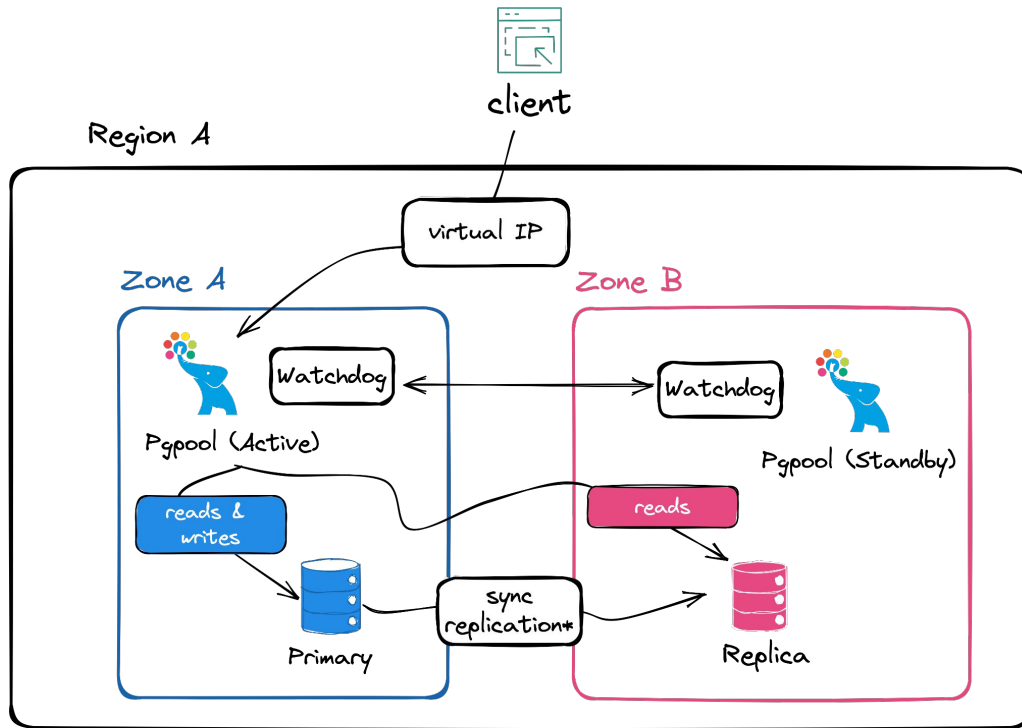
+

PG\_Bouncer

(connection pooling)



# Sample Configuration With Single Replica



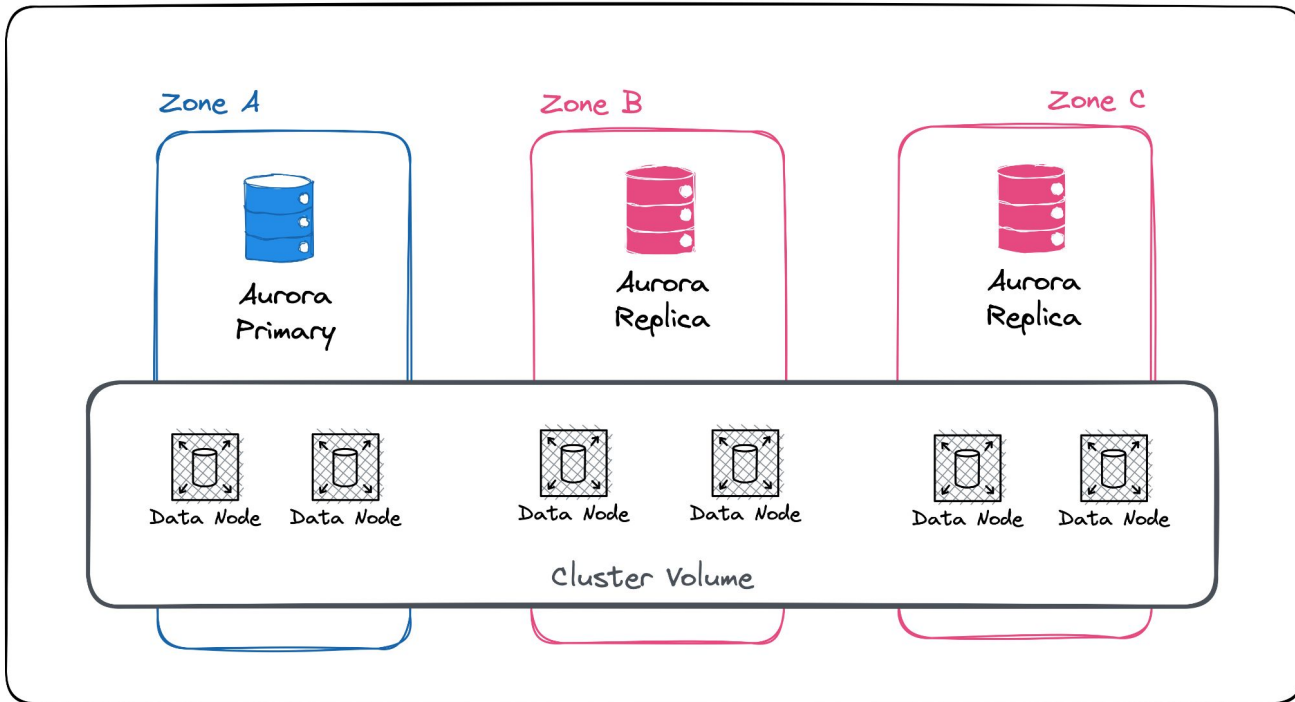
**sync replication\*** - the primary can't commit if the replica is down. Have 2+ replicas.





# Amazon Aurora Standard Configuration

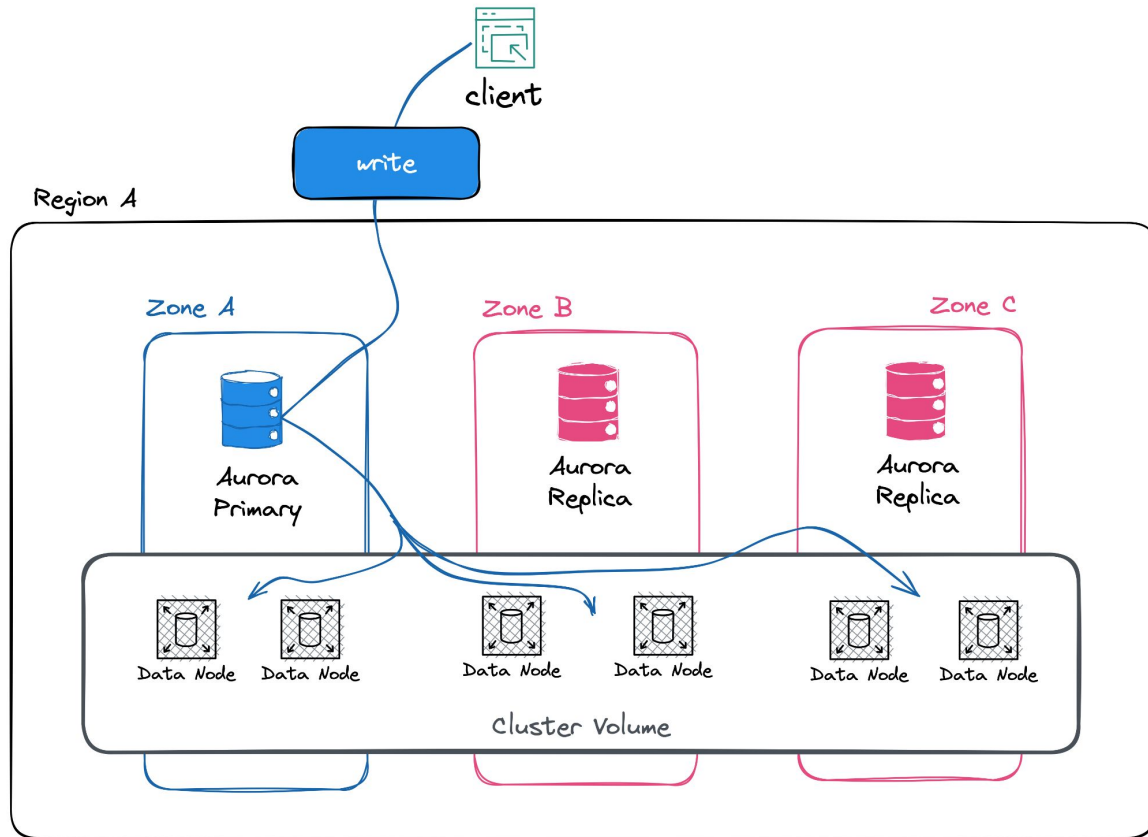
Region A





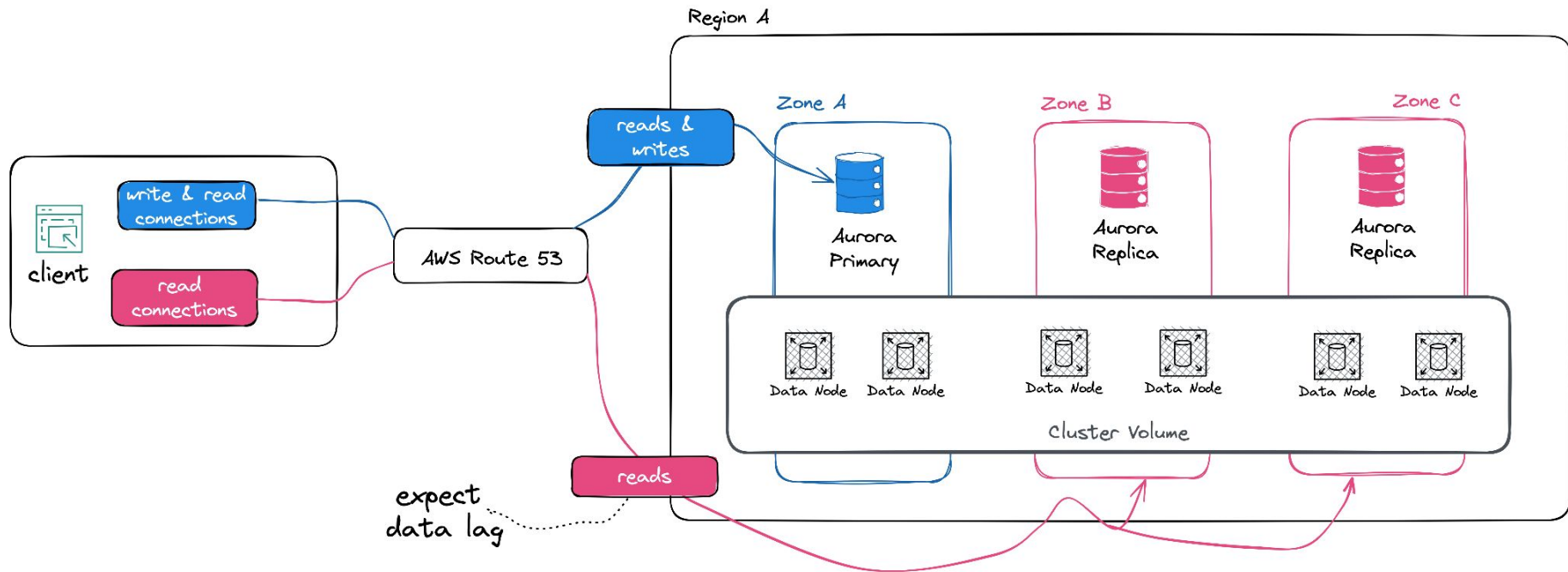
Aurora

# Writes Require a Quorum of 4 Data Nodes

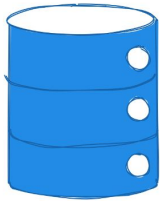




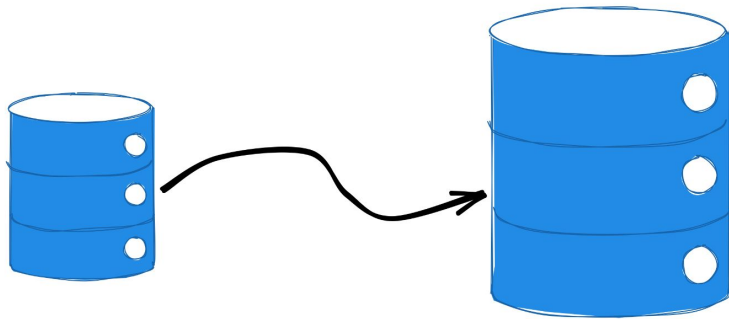
# An approach to scale reads in Aurora



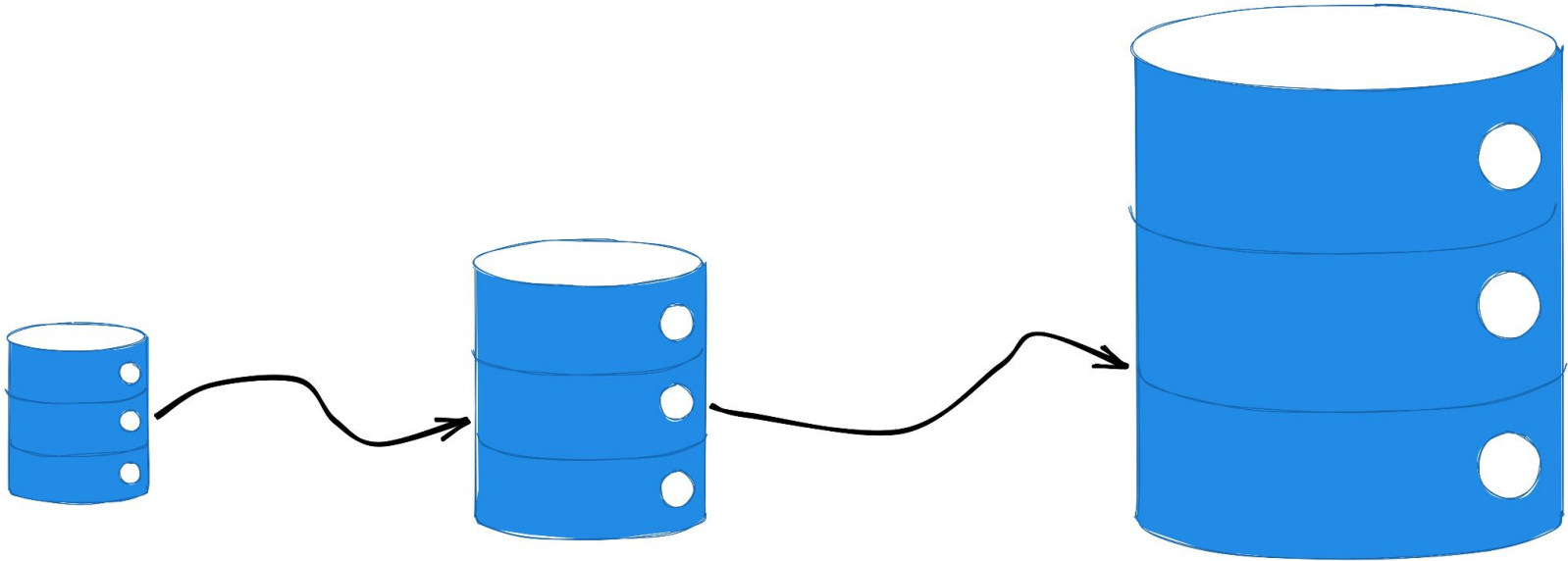
When you need to scale out



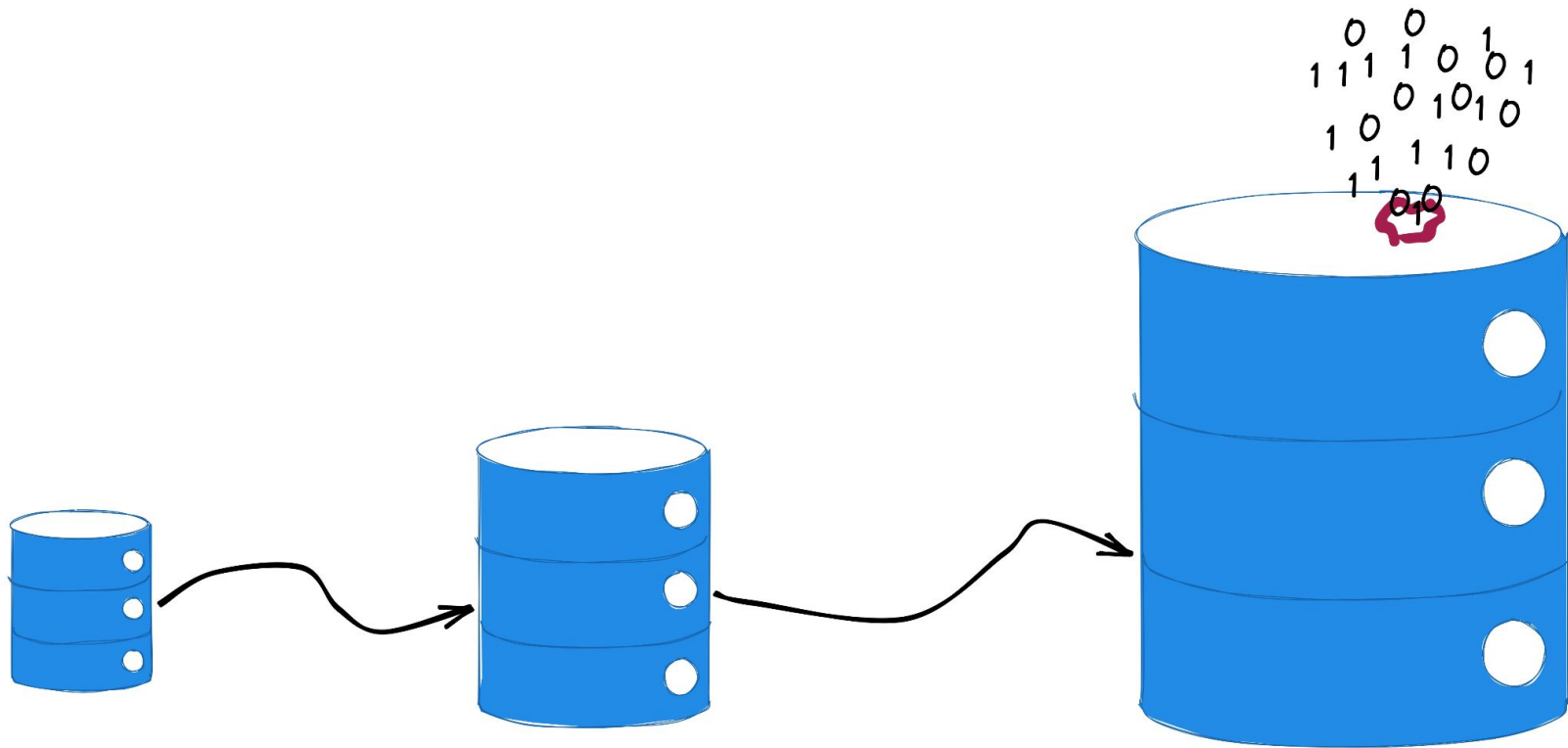
When you need to scale out



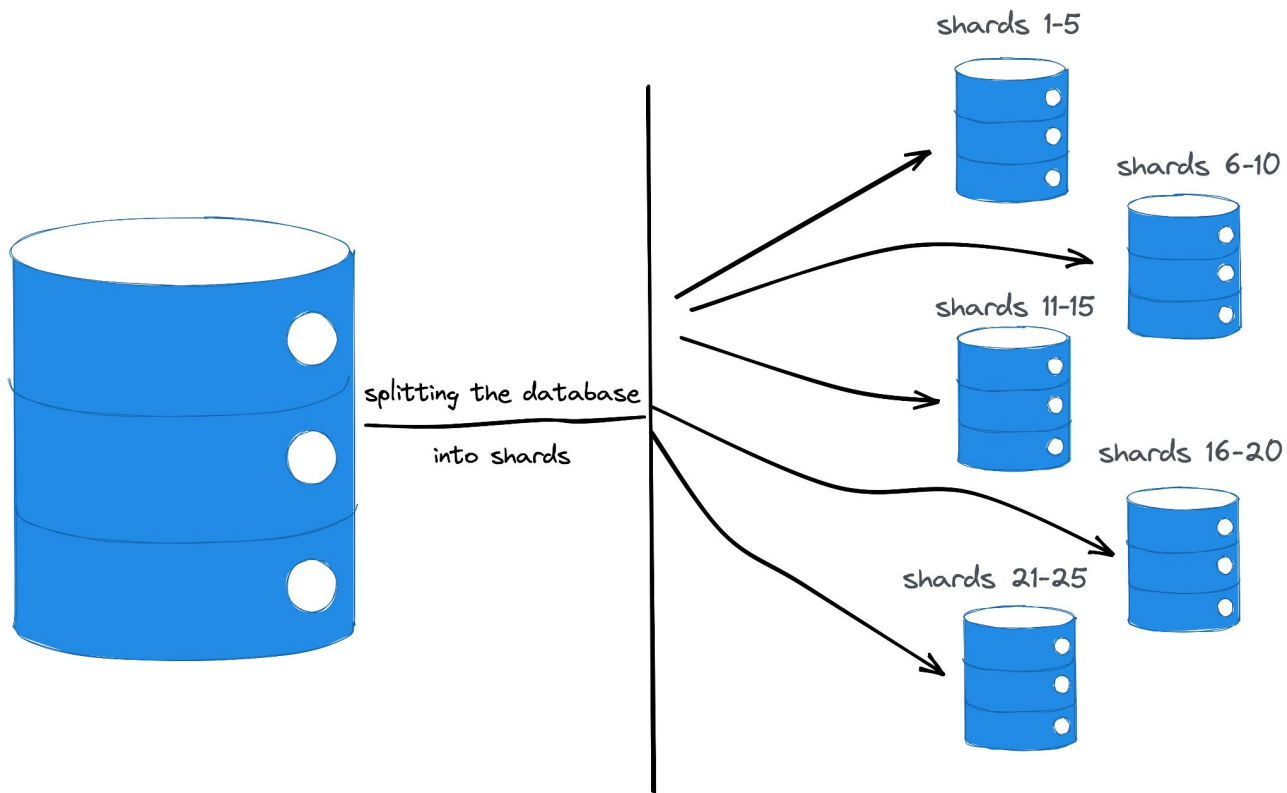
When you need to scale out



When you need to scale out



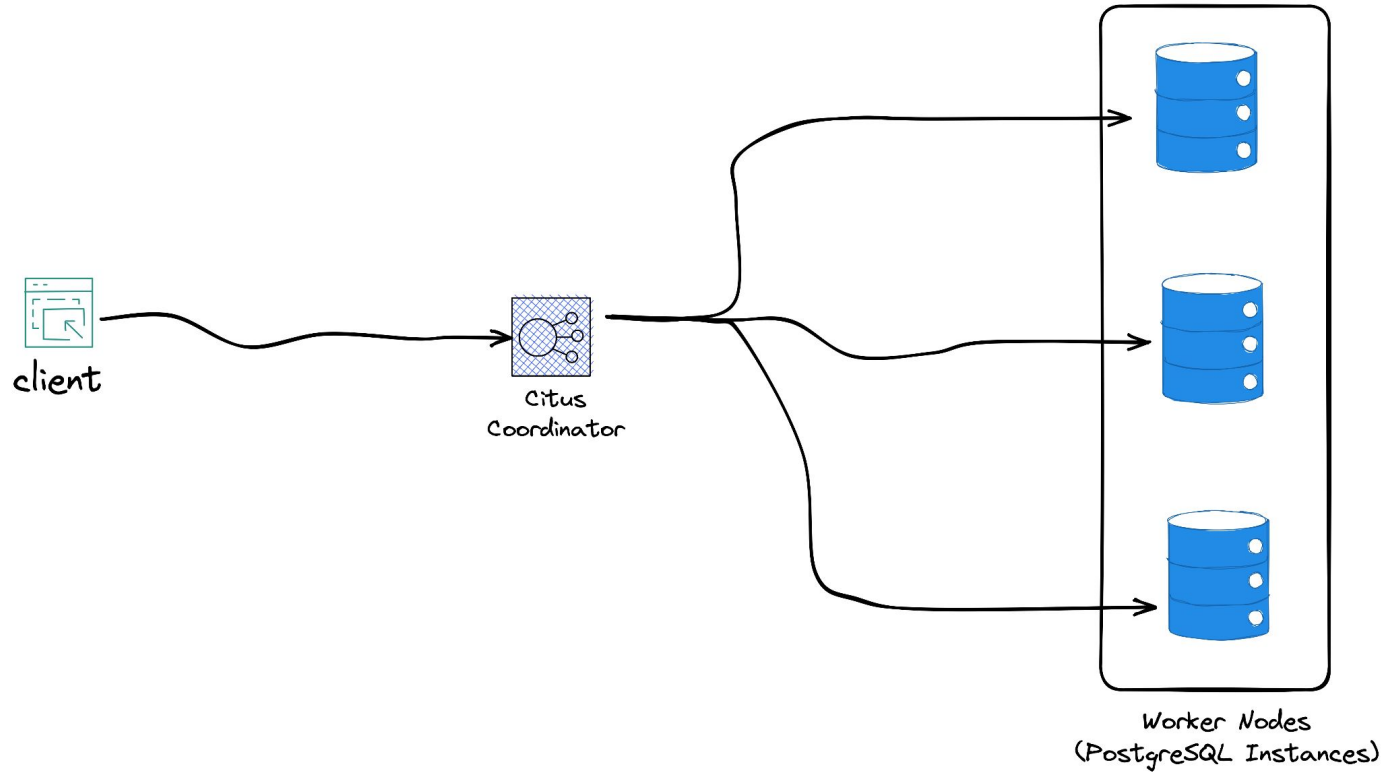
# Sharding





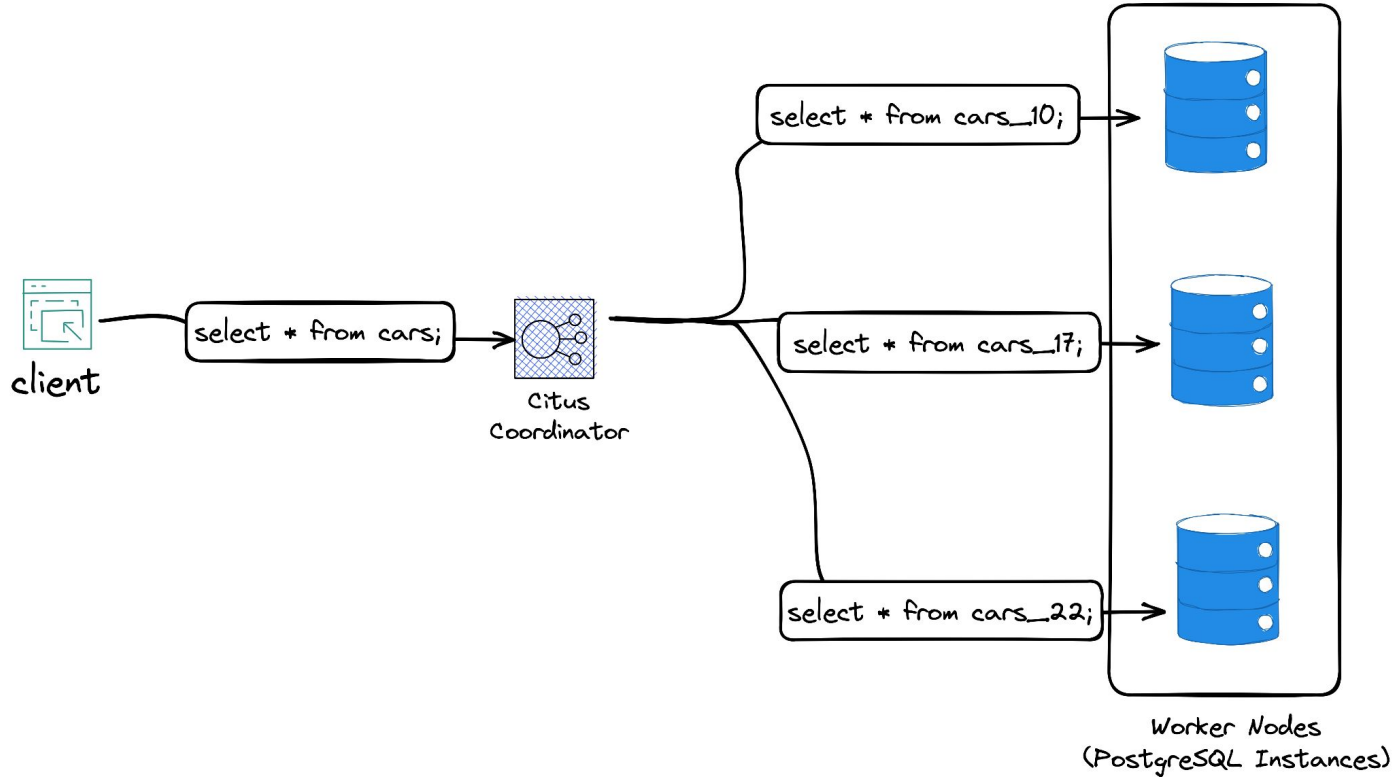
# PostgreSQL Extension for Sharding

(aka. Azure CosmosDB for PostgreSQL)



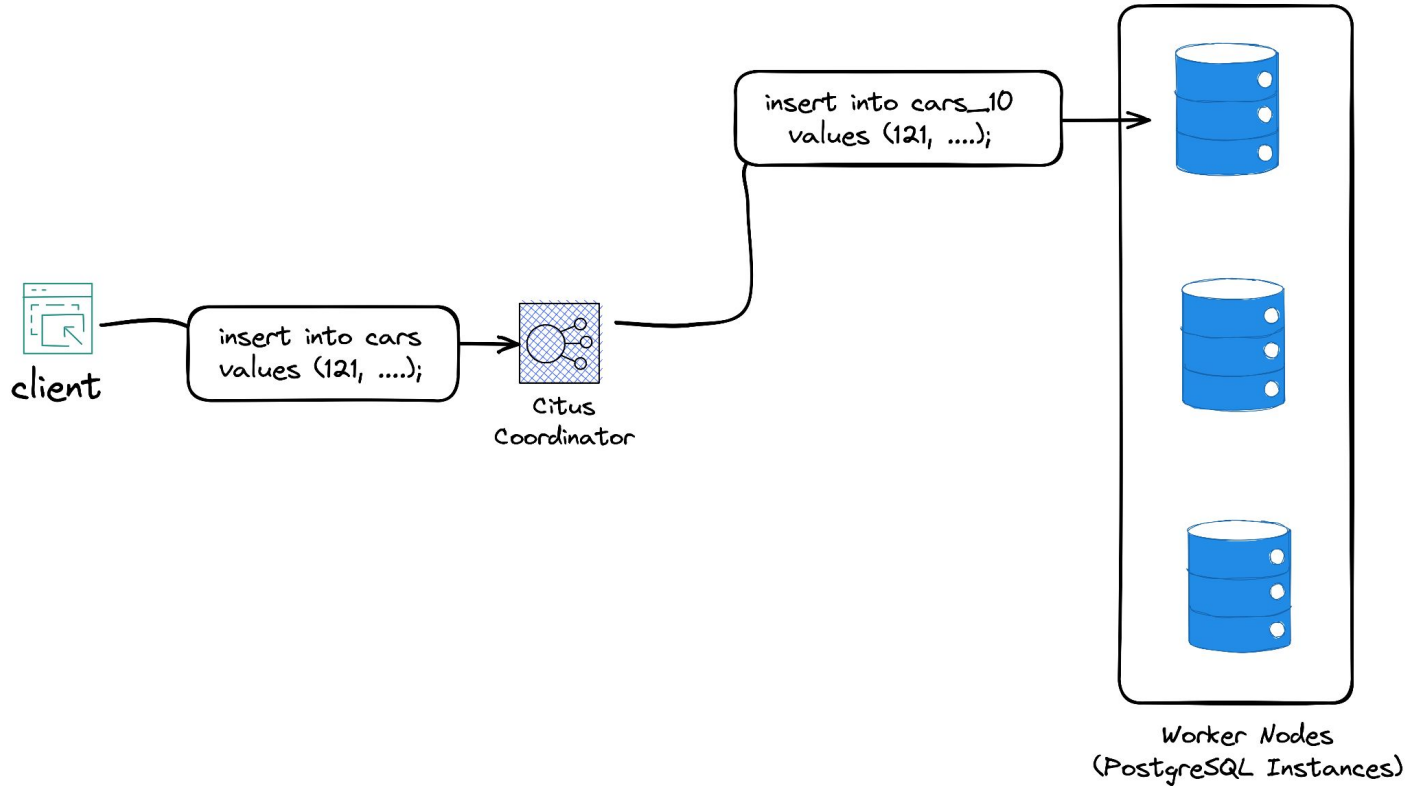
# PostgreSQL Extension for Sharding

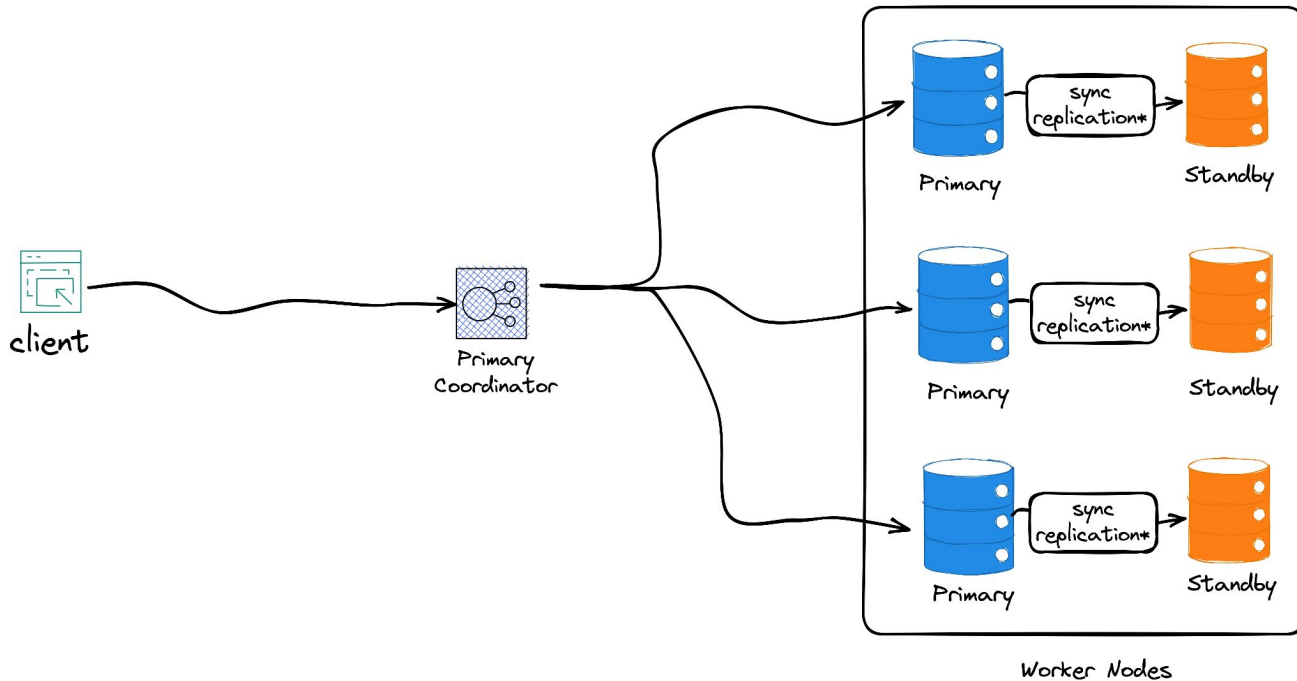
(aka. Azure CosmosDB for PostgreSQL)



# PostgreSQL Extension for Sharding

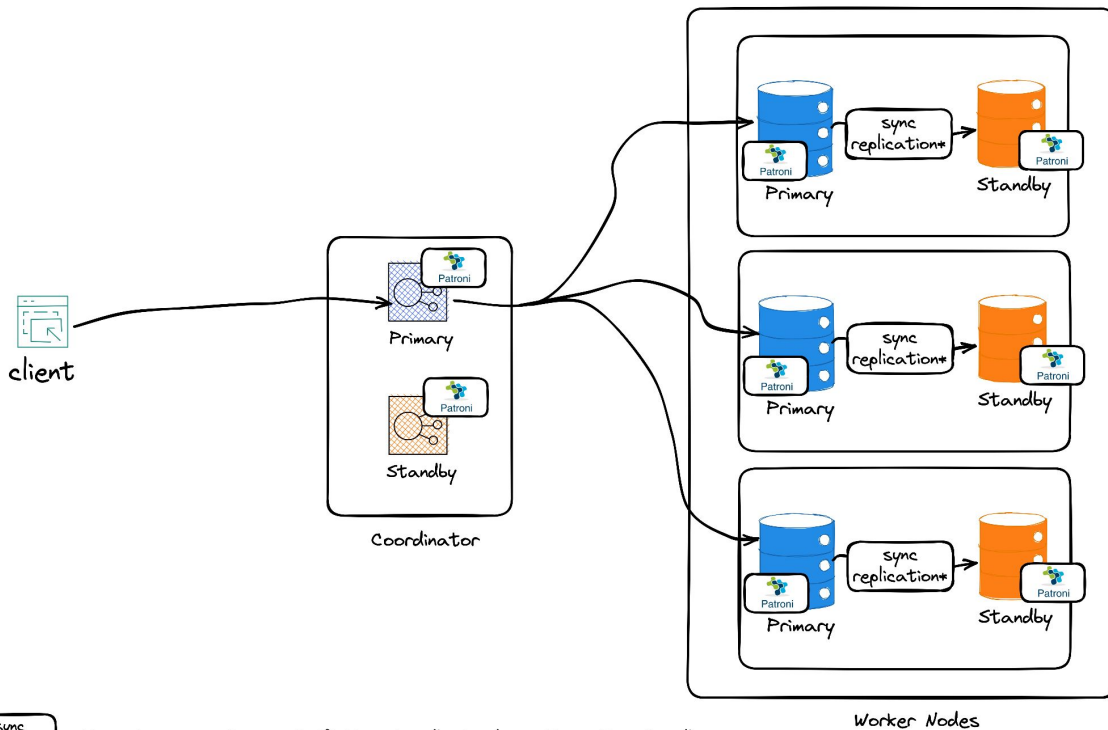
(aka. Azure CosmosDB for PostgreSQL)







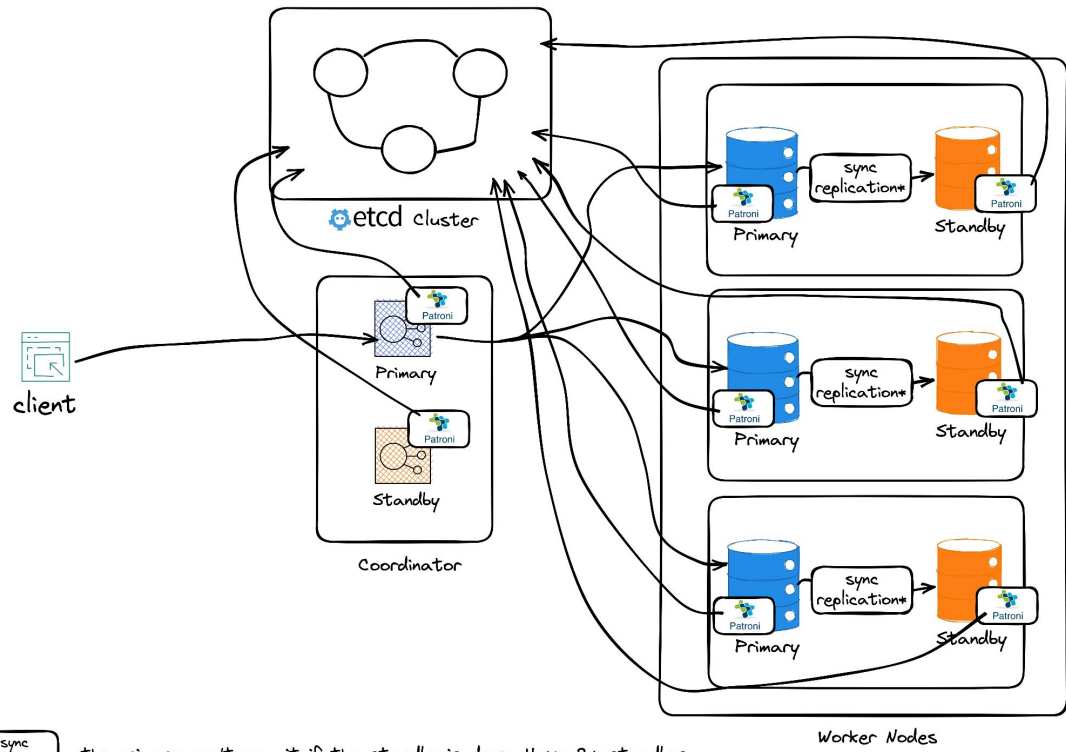
sync replication\*


- the primary can't commit if the standby is down. Have 2+ standbys.



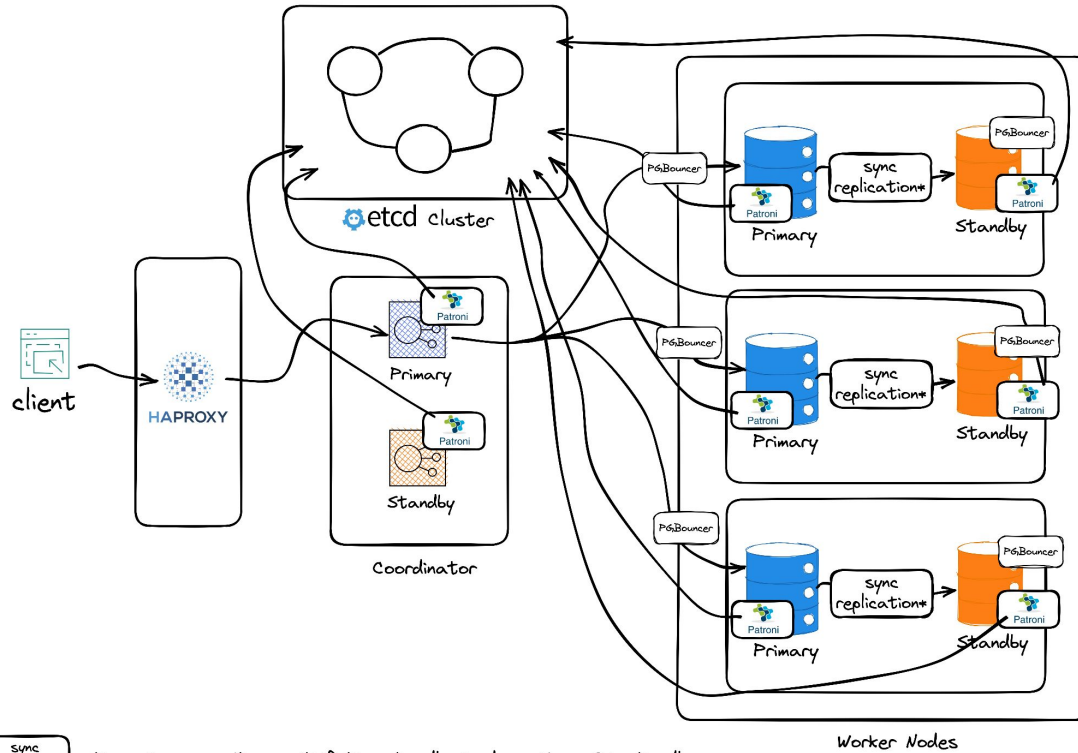
 - the primary can't commit if the standby is down. Have 2+ standbys.


 - Patroni agents




 - the primary can't commit if the standby is down. Have 2+ standbys.

 - Patroni agents



 - the primary can't commit if the standby is down. Have 2+ standbys.

 - Patroni agents

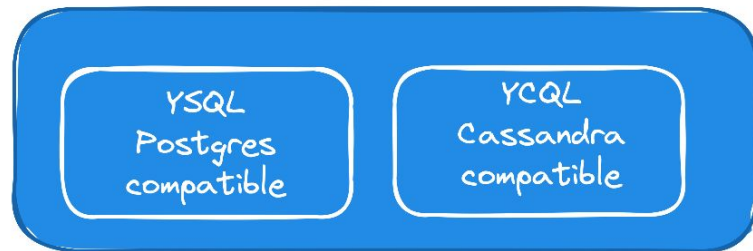
 - Connection pooler

open source  
distributed SQL database

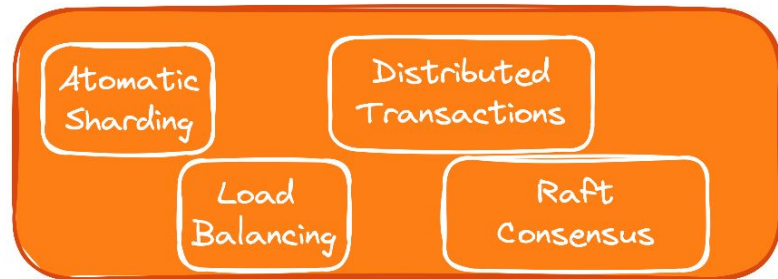
Built on PostgreSQL

With Google Spanner-like storage

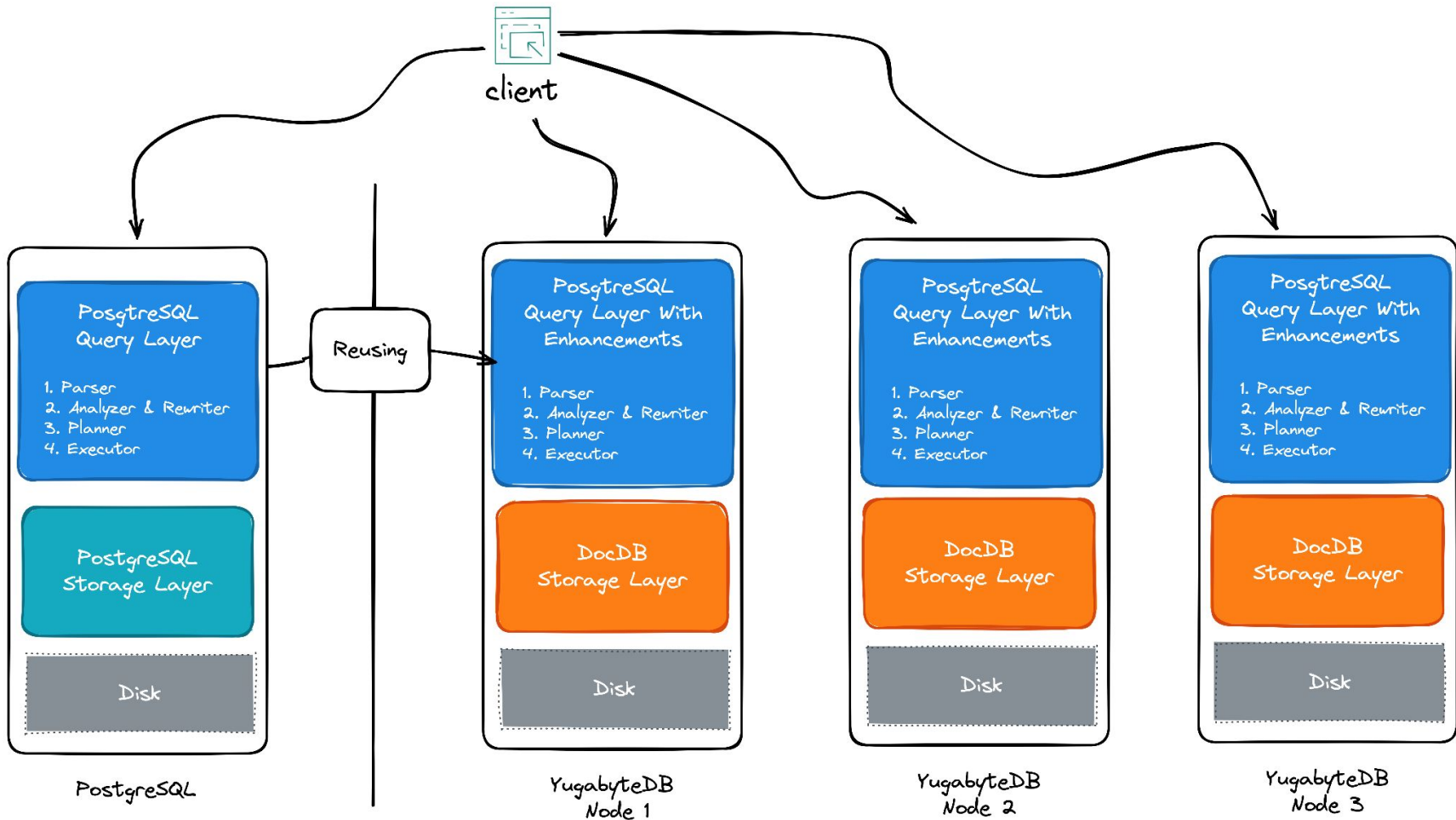
Distributed Query Layer



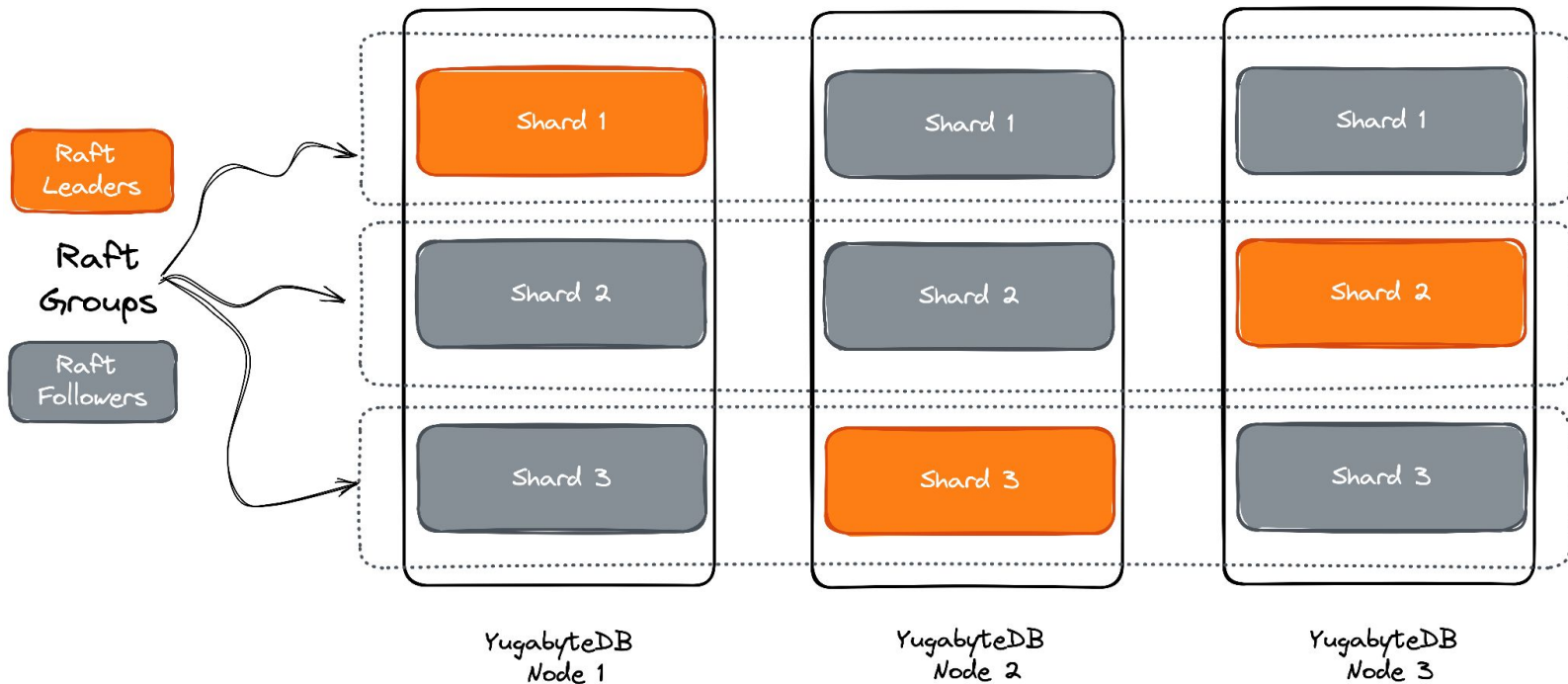
Distributed Storage Layer



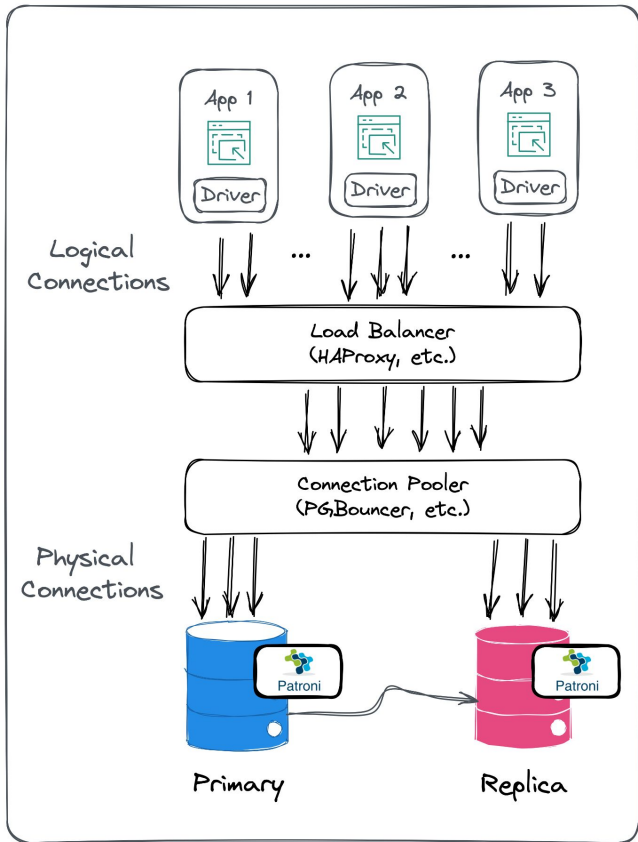




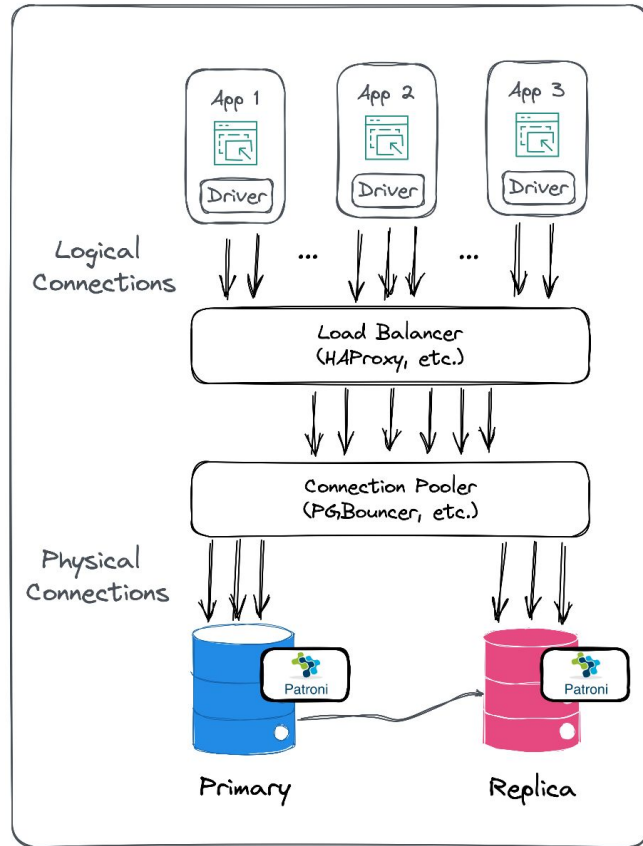
# YugabyteDB Distributes Data With Sharding and Synchronizes Via Raft



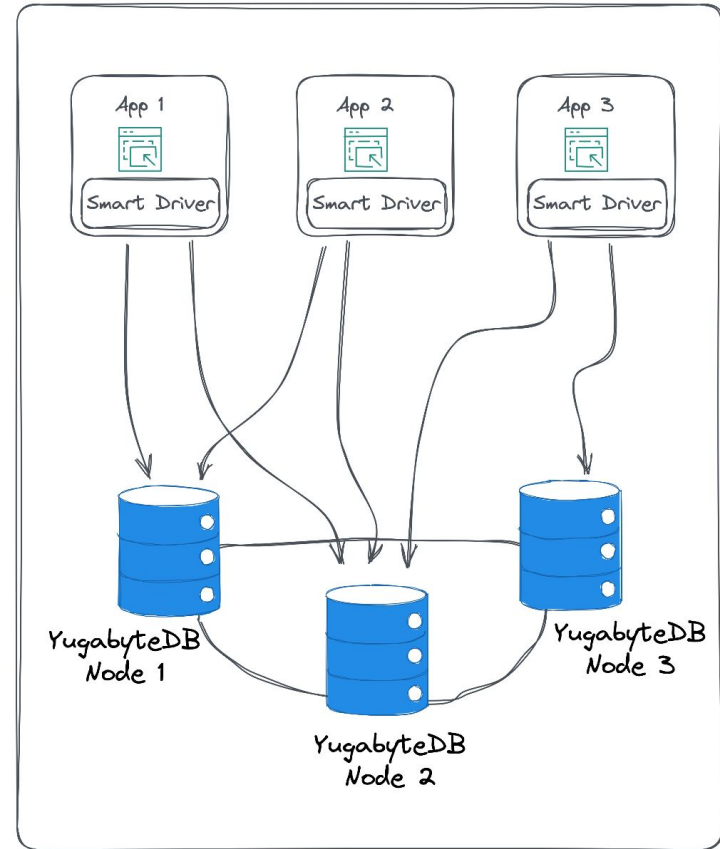
# Application Architecture With A Centralized SQL Database



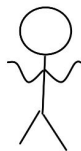
## Application Architecture With A Centralized SQL Database



## Application Architecture With A Distributed SQL Database



What if I want  
to learn more?



YugabyteDB University



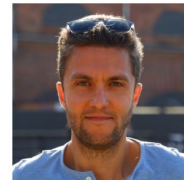
yugabyteDB

yugabyte.com

## Books



Me



denismagda

# Thank You

Join us on Slack:

[www.yugabyte.com/slack](http://www.yugabyte.com/slack)

Star us on GitHub:

[github.com/yugabyte/yugabyte-db](https://github.com/yugabyte/yugabyte-db)