STORMATICS

Clustering in PostgreSQL

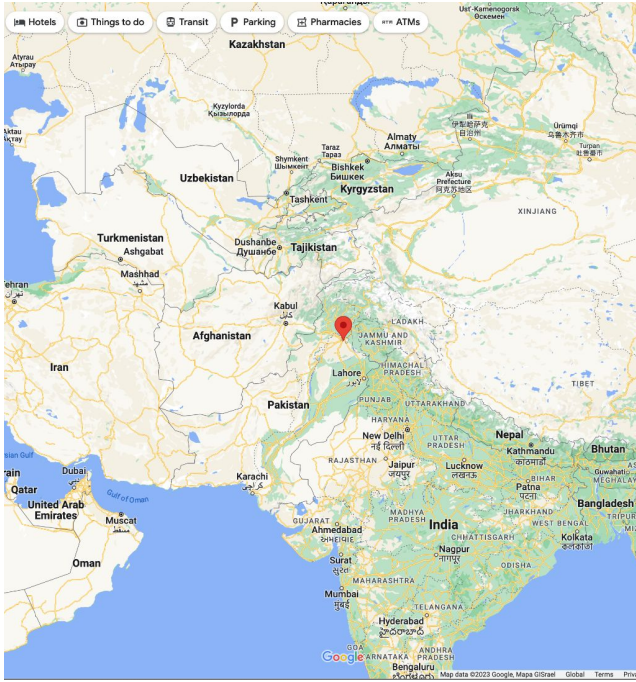
Because one database server is never enough (and neither is two)

PGConf Nepal
May 11-12, 2023

STORMATICS

```
postgres=# select * from umair;
```

```
-[ RECORD 1 ]-----------------------------------------------
name        | Umair Shahid
description | 20 year veteran of the PostgreSQL community
company     | Stormatics
designation | Founder
location    | Islamabad, Pakistan
family      | Mom, Wife & 2 kids
kid1        | Son, 16 year old
kid2        | Daughter, 13 year old
```



EDB POWER TO POSTGRES

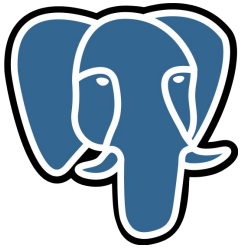
OpenSCG PostgreSQL, Java & Linux Experts

2ndQuadrant PostgreSQL

PERCONA

PostgreSQL Solutions for the Enterprise

Clustering & Analytics, backed by 24/7 Support and Professional Services

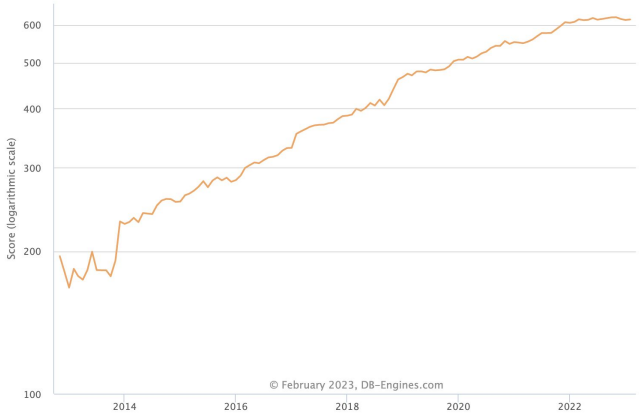


10 years of popularity

As measured by

DB-ENGINES

https://db-engines.com/en/ranking_trend

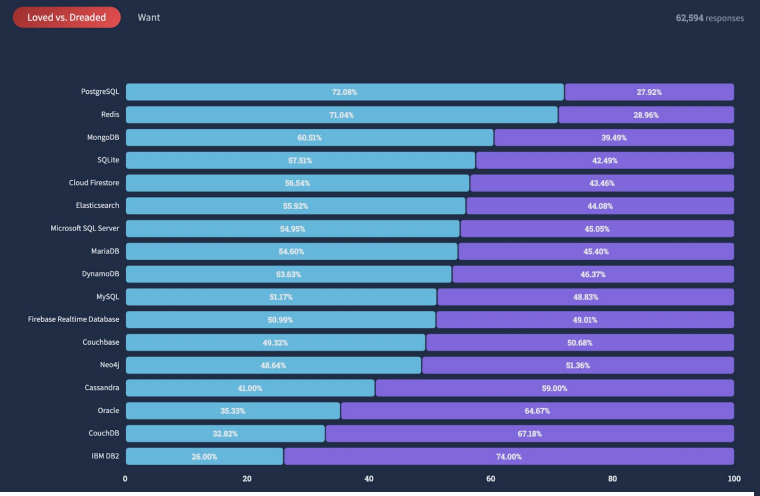


PostgreSQL

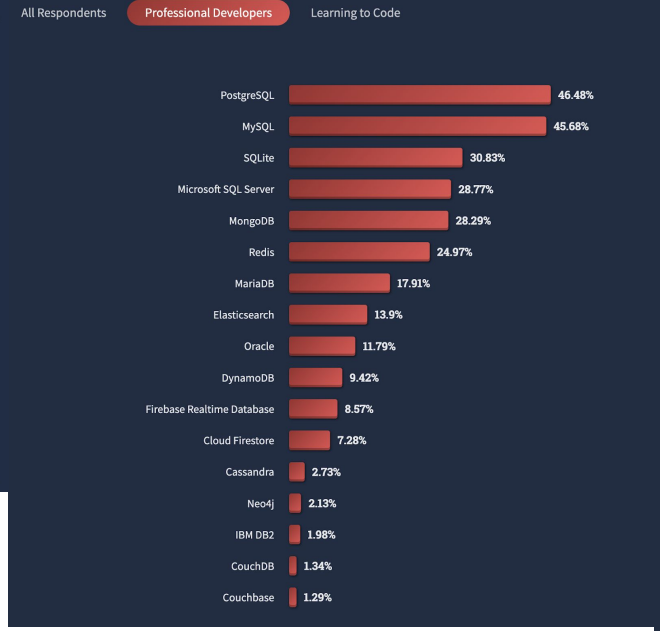
Oracle

MySQL

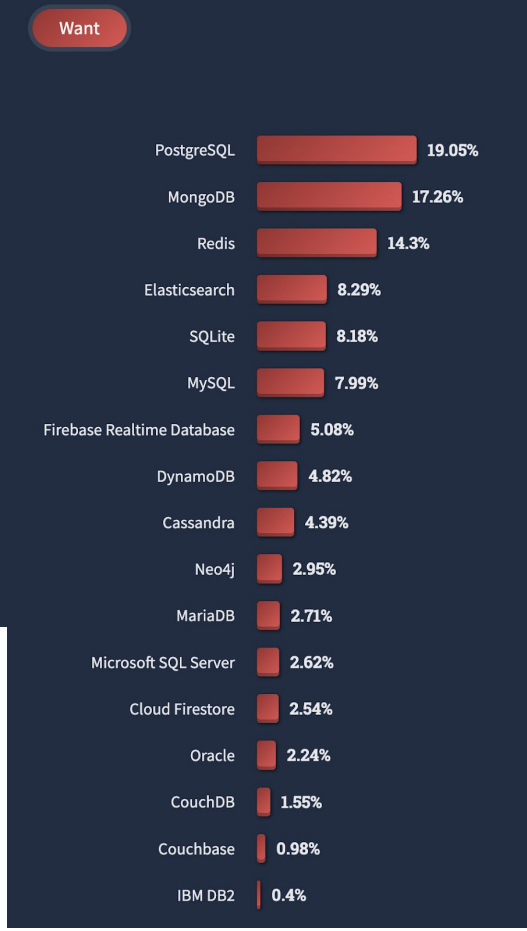
SQL Server

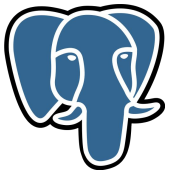


Most Loved



Most Used (Professional Developers)





Loved by Developers

stack overflow
2022 Developer Survey

Most Wanted

On to the topic now!

STORMATICS

What is High Availability?

- Remain operational even in the face of hardware or software failure
- Minimize downtime
- Essential for mission-critical applications that require 24/7 availabili
- Measured in ‘Nines of Availability’

Nines of Availability

Availability	Downtime per year
90% (one nine)	36.53 days
99% (two nines)	3.65 days
99.9% (three nines)	8.77 hours
99.99% (four nines)	52.60 minutes
99.999% (five nines)	5.26 minutes

But my database resides in the cloud, and the cloud is always available

Right?

Wrong!

Amazon RDS Service Level Agreement

Multi-AZ configurations for MySQL, MariaDB, Oracle, and PostgreSQL are covered by the Amazon RDS Service Level Agreement ("SLA"). The RDS SLA affirms that AWS will use commercially reasonable efforts to make Multi-AZ instances of Amazon RDS available with a Monthly Uptime Percentage of at least 99.95% during any monthly billing cycle. In the event Amazon RDS does not meet the Monthly Uptime Percentage commitment, affected customers will be eligible to receive a service credit.*

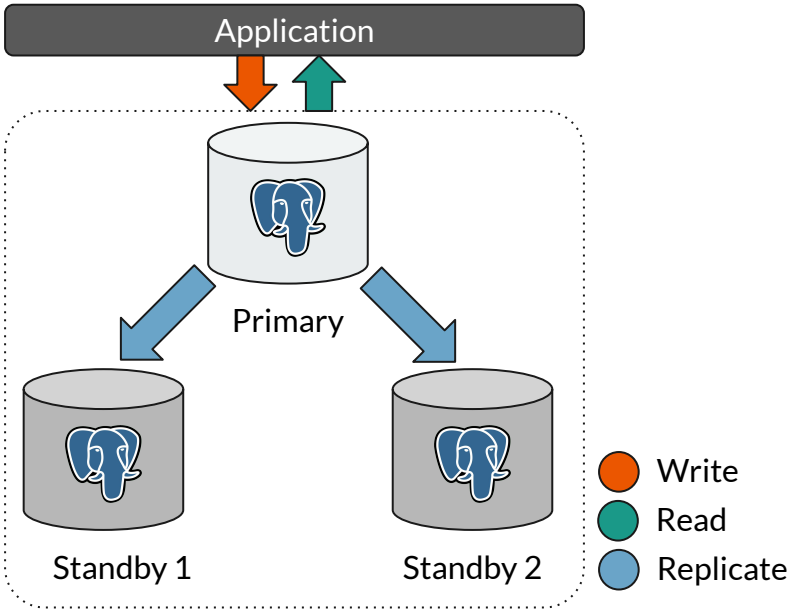
99.95% = three and a half nines = 4.38 hours of downtime per year!!!

* https://aws.amazon.com/rds/ha/

So - what do I do if I want better reliability for my mission-critical data?

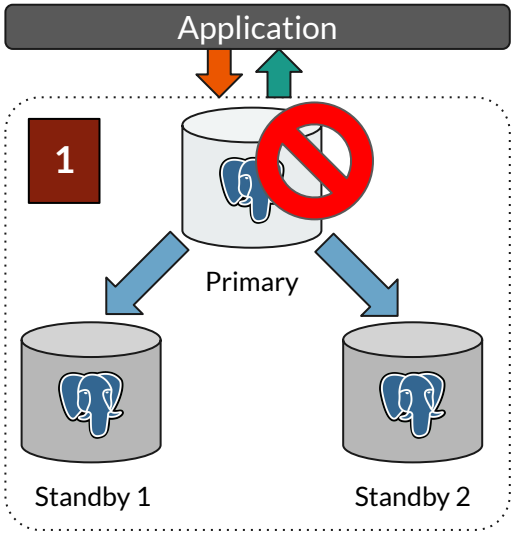
Clustering!

What is clustering?

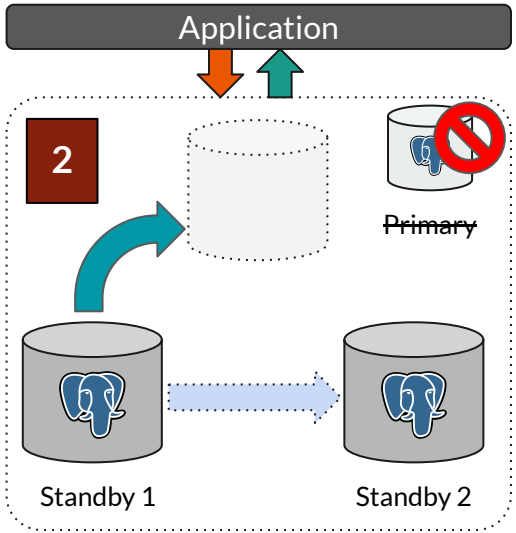


- Multiple database servers work together to provide redundancy
- Gives the appearance of a single database server
- Application communicates with the primary PostgreSQL instance
- Data is replicated to standby instances
- Auto failover in case the primary node goes down

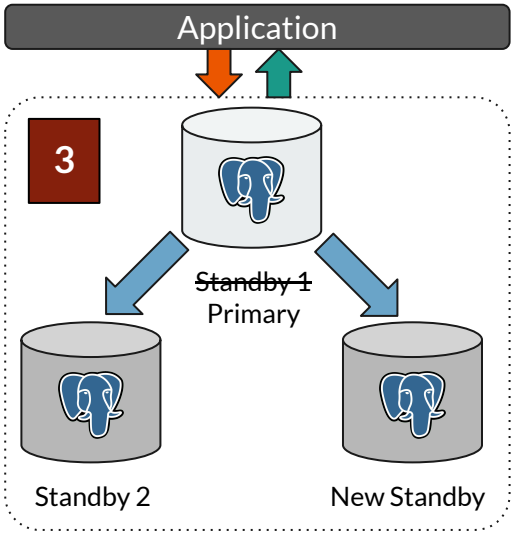
What is auto failover?



- Primary node goes down

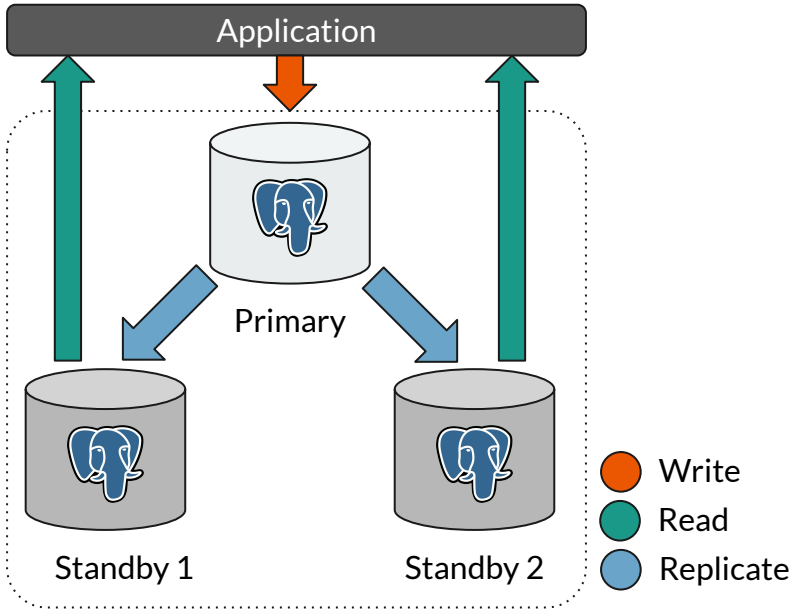


- Standby 1 gets promoted to Primary
- Standby 2 becomes subscriber to Standby 1



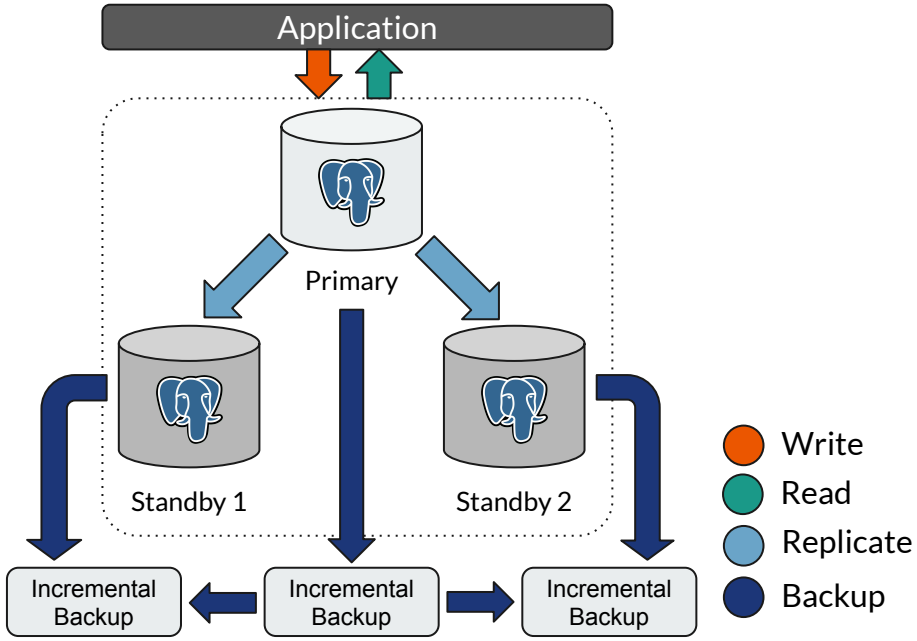
- New Standby is added to the cluster
- Application talks to the new Primary

Clusters with load balancing



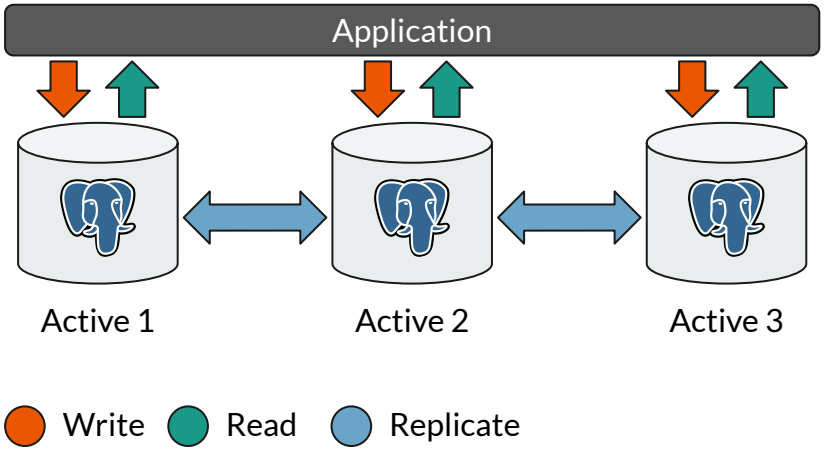
- Write to the primary PostgreSQL instance and read from standbys
- Data redundancy through replication to two standbys
- Auto failover in case the primary node goes down

Clusters with backups and disaster recovery



- Incremental backups
- Redundancy introduced from primary as well standbys
- RTO and RPO balance achieved per organizational requirements
- Point-in-time recovery

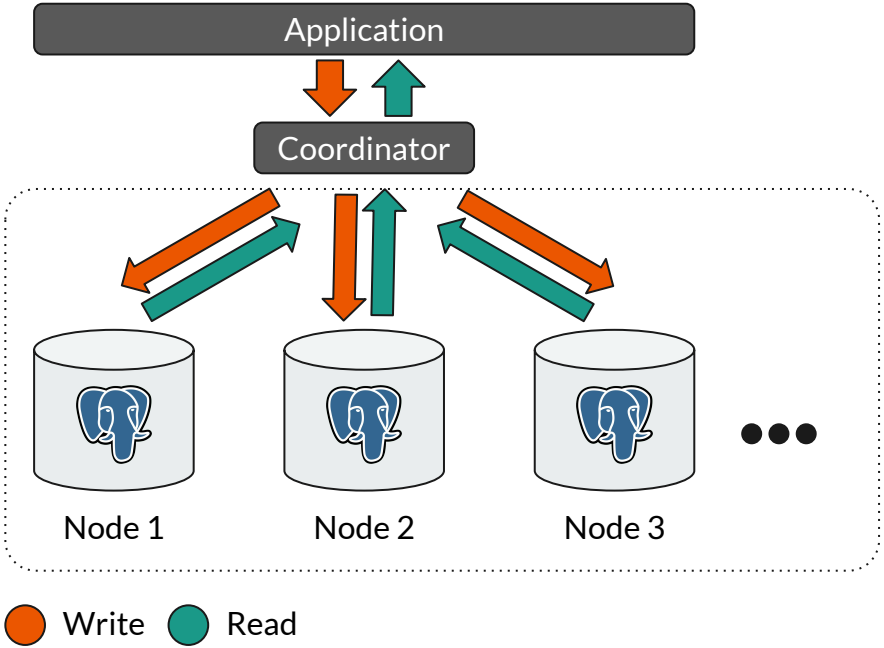
Multi-node clusters with Active-Active configuration*



- Shared-Everything architecture
- Load balancing for read as well as write operations
- Database redundancy to achieve high availability
- Asynchronous replication between nodes for better efficiency

*with conflict resolution at the application layer

Multi-node clusters with data sharding and horizontal scaling



- Shared-Nothing architecture
- Automatic data sharding based on defined criteria
- Read and write operations are auto directed to the relevant node
- Each node can have its own standbys for high availability

Globally distributed clusters

- Spin up clusters on the public cloud, private cloud, on-prem, bare metal, VMs, or a hybrid of all the above
- Geo fencing for regulatory compliance
- High availability across data centers and geographies

Replication - synchronous vs asynchronous

Synchronous

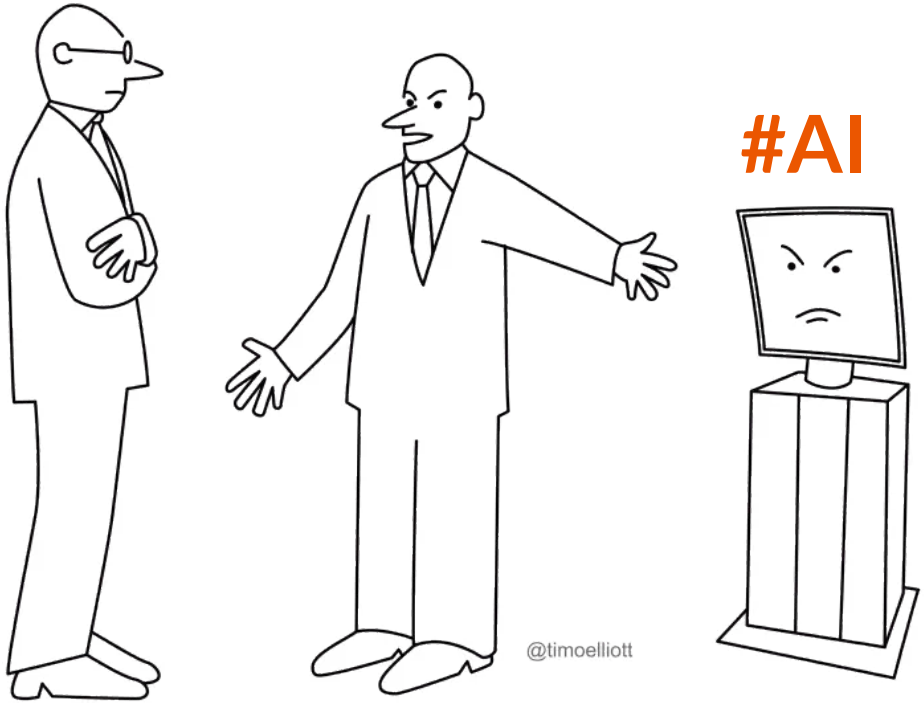
- Data is transferred immediately
- Transaction waits for confirmation from replica before it commits
- Ensures data consistency across all nodes
- Performance overhead caused by latency
- Used where data accuracy is critical, even at the expense of performance

Asynchronous

- Data may not be transferred immediately
- Transaction commits without waiting for confirmation from replica
- Data may be inconsistent across nodes
- Faster and more scalable
- Used where performance matters more than data accuracy

Challenges in Clustering

- Split brain
- Network Latency
- False alarms
- Data inconsistency



His decisions aren't any better than yours
— but they're WAY faster...

Challenges in Clustering

- Split brain
- Network Latency
- False alarms
- Data inconsistency

Split Brain

- **Defined:** Node in a highly available cluster lose connectivity with each other but continue to function independently

- **Challenge:** More than one node believes that it is the primary leading to inconsistencies and possible data loss

Split Brain - Prevention

- Quorum based voting
  - Majority nodes must agree on primary
  - Cluster stops if quorum is not achieved
- Redundancy and failover
  - Prevents a single point of failure
- Proper configuration
- Split brain resolver
  - Software based detection & resolution
  - Can shut down nodes in case of scenario
- Network segmentation
  - Physical network separation
  - Avoid congestion

Split Brain - Resolution

- Witness node
  - Third node that acts as a tie-breaker
  - The winner acts as primary, the loser is shut down
- Consensus algorithm
  - Paxos and Raft protocols are popular
- Manual resolution
  - DBA observes which nodes are competing to be the primary
  - Takes action based on best practices learnt from experience

Challenges in Clustering

- Split brain
- **Network Latency**
- False alarms
- Data inconsistency

Network Latency

- **Defined:** Time delay when data is transmitted from one point to another
- **Challenge:** Delayed replication can result in data loss. Delayed signals can trigger a false positive.

Network Latency - Preventing False Positives

- Deploy nodes in proximity
  - Reduces time delay and network hops
- Implement quorum-based system
  - If quorum isn't achieved failover won't occur
- High speed, low latency networking
  - High quality hardware and associated software
- Optimize database configuration
  - Parameter tuning based on workloads
  - max_connections, tcp_keealive_idle, ...
- Alerting system
  - Detect and alert admins of possible issues to preempt false positives

Challenges in Clustering

- Split brain
- Network Latency
- False alarms
- Data inconsistency

False Alarms

- **Defined:** A problem is reported, but in reality, there is no issue
- **Challenge:** Can trigger a failover when one isn't required, leading to unnecessary disruptions and impacting performance

False Alarms - Prevention

- Proper configuration
  - Best practices, past experience, and some hit & trial is required to ensure that the thresholds are configured appropriately
- Regular maintenance
  - Latest version of software and firmware to be used in order to avoid known bugs and exploits
- Regular testing
  - Testing of various use cases can help identify bottlenecks and possible misconfigurations
- Multiple monitoring tools
  - Multiple tools can help cross-reference alerts and confirm if failover is required

False Alarms - Resolution

In case a false alarm is triggered ...

- Check logs
  - Check the logs of cluster nodes, network devices, and other infrastructure components to identify any anomalies
- Notify stakeholders
  - Notify all stakeholders involved in the cluster's operations to prevent any unnecessary action
- Monitor cluster health
  - Monitor to cluster's health closely to ensure that it is functioning correctly and no further false alarms are triggered

Challenges in Clustering

- Split brain
- Network Latency
- False alarms
- Data inconsistency

Data Inconsistency

- **Defined:** Situations where data in different nodes of a cluster becomes out of sync, leading to inconsistent results and potential data corruption

- **Challenge:** Inaccurate query results that vary based on which node is queried. Such issues are very hard to debug.

Data Inconsistency - Prevention

- Synchronous replication
  - Ensures data is synchronized across all nodes before it is committed
  - Induces a performance overhead
- Load balancer
  - Ensure that all queries from a specific application are sent to the same node
  - Relies on eventual consistency of the cluster
- Monitoring tools
  - Help with early identification of possible issues so you can take preventive measures
- Maintenance windows
  - Minimize data disruption with planned downtime
- Regular testing
  - Test production use cases regularly to ensure the cluster is behaving as expected

Data Inconsistency - Resolution

- Resynchronization
  - Manual sync between nodes to correct data inconsistencies
- Rollback
  - Point-in-time recovery using tools like pg_rewind and pg_backrest
- Monitoring
  - Diligent monitoring to ensure that the issue doesn’t recur

This all sounds really hard

Open source clustering tools for PostgreSQL

- repmgr
  - https://repmgr.org/
  - GPL v3
  - Provides automatic failover
  - Manage and monitor replication
- pgpool-II
  - https://pgpool.net/
  - Similar to BSD & MIT
  - Middleware between PostgreSQL and client applications
  - Connection pooling, load balancing, caching, and automatic failover
- Patroni
  - https://patroni.readthedocs.io/en/latest/
  - MIT
  - Template for PostgreSQL high availability clusters
  - Automatic failover, configuration management, & cluster management

Questions?

pg_umair