



Redis Enterprise Technology Deep Dive

Shekhar Suman, Solution Architect, Redis
5th June, 2024

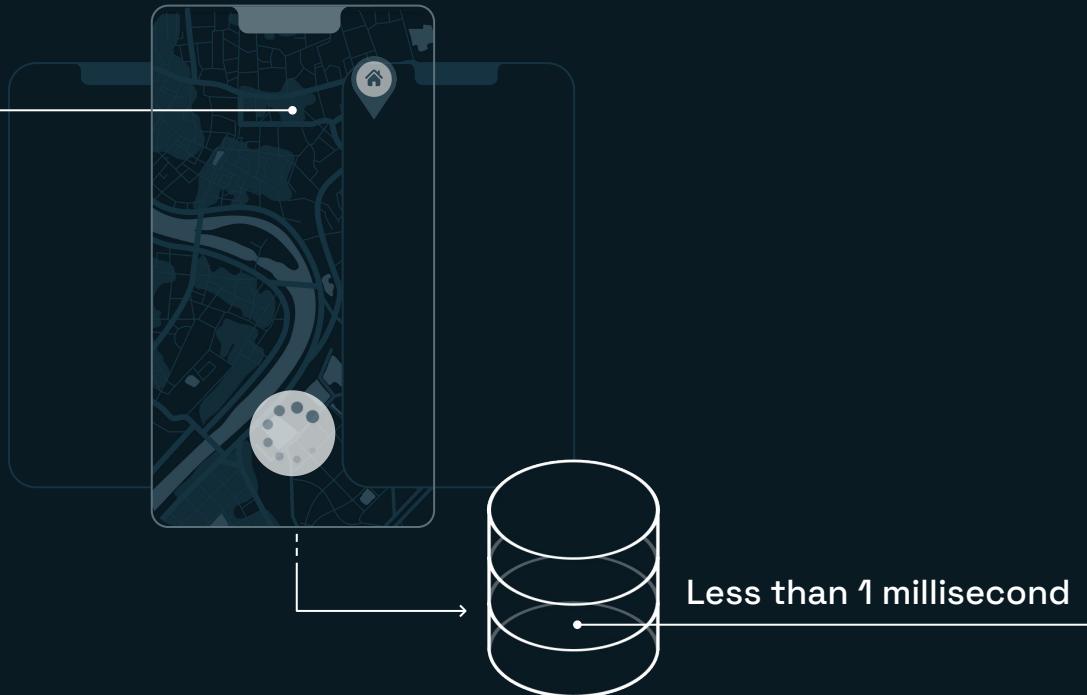
Speed is a competitive advantage.



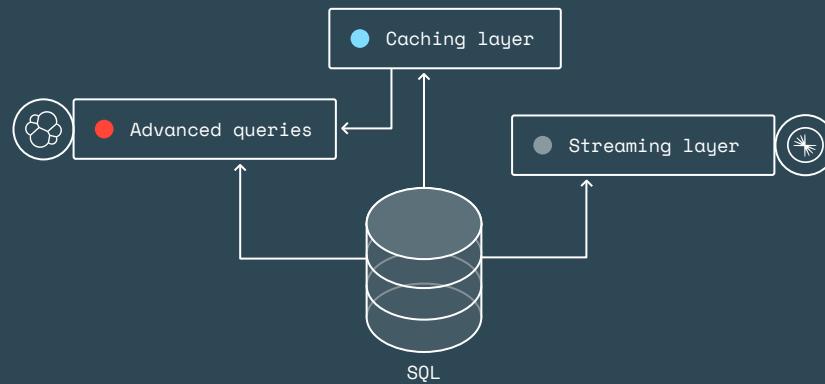
Your customers won't stand for spinning
wheels or long load times.

Milliseconds matter to your users.

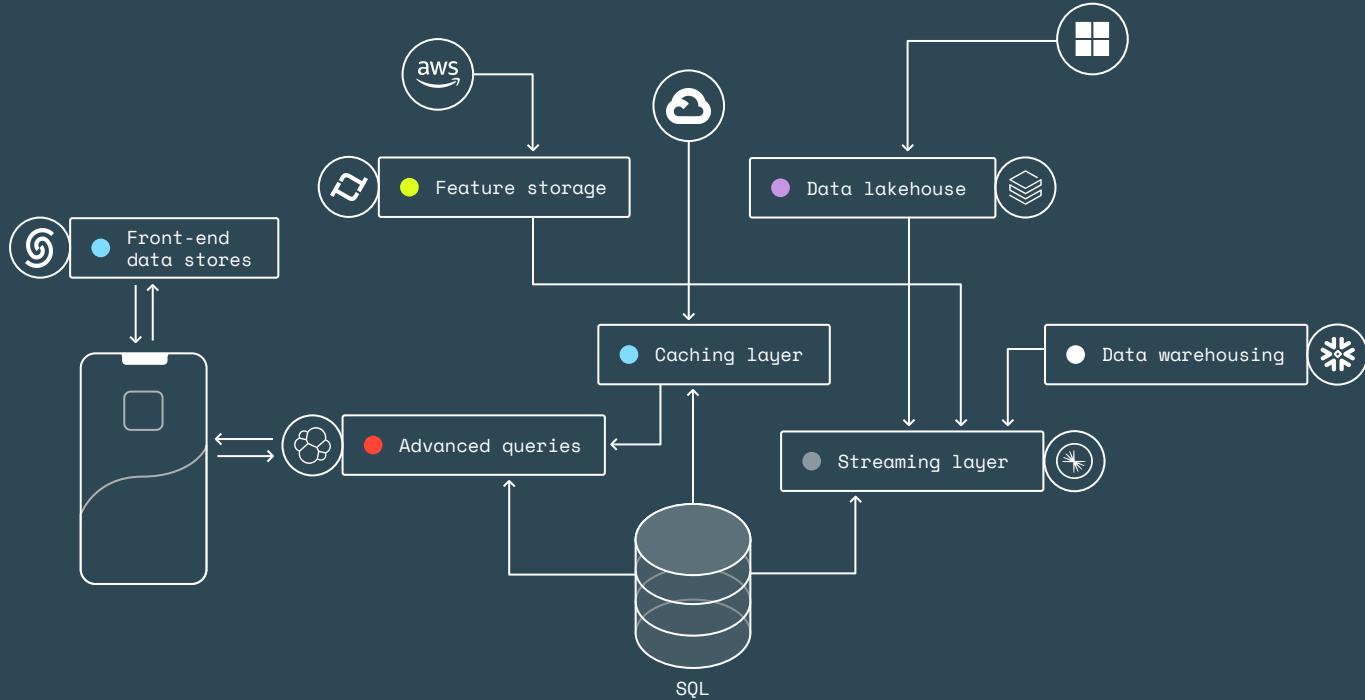
Less than 100 milliseconds
end-to-end response time



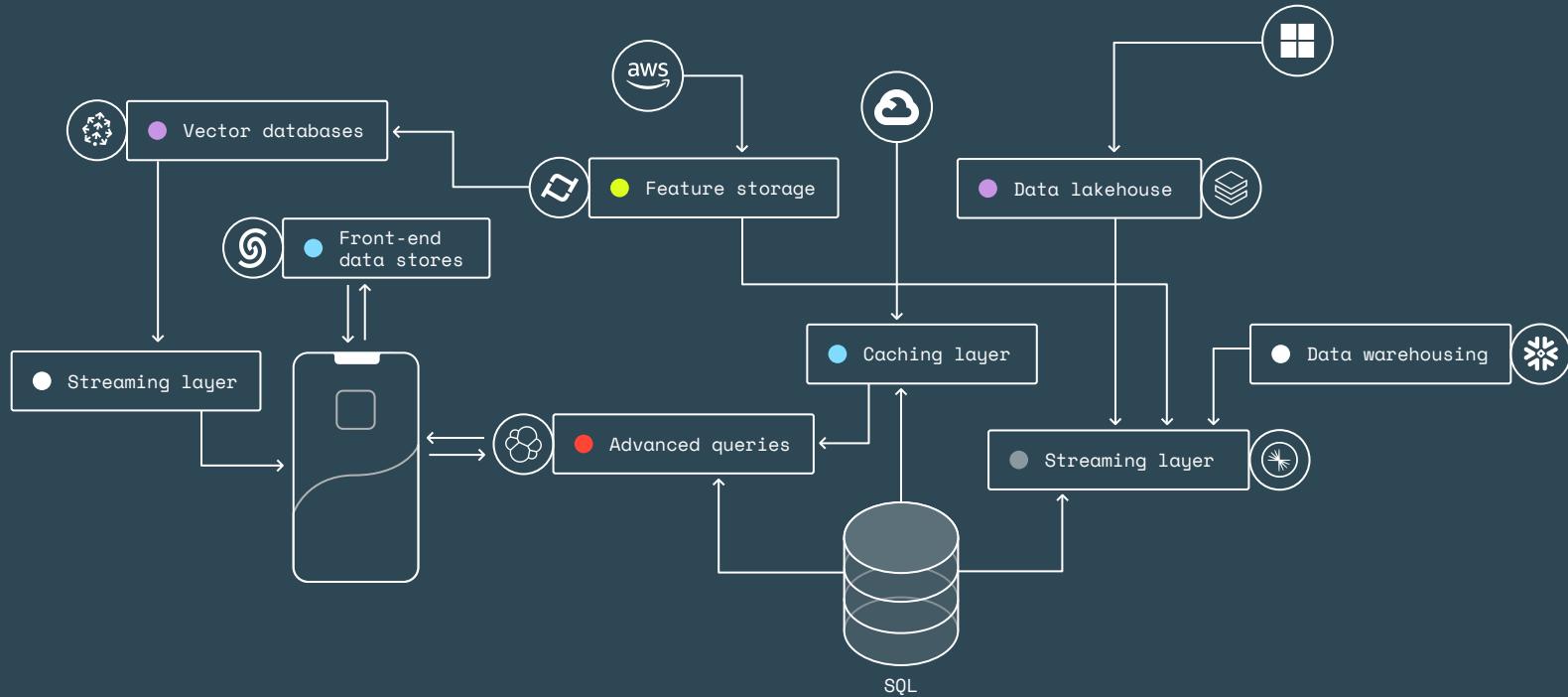
Traditional databases can't keep up.



They create fragmented, slow tech stacks.



GenAI apps only add to the chaos.



We set the standard for fast—so you can too.

- FAST

In-memory storage provides unparalleled data access speed



- SIMPLE

Built to handle a wide range of use cases to stop database sprawl



- AI READY

New vector and semantic search capabilities



Devs love us. Apps need us.

● MOST ADMIRE

NoSQL Database,
2023



● MOST POPULAR

Container Images -
2020, 2021, 2023



● MOST LAUNCHED

Database
technology



The ones who built open source Redis.

Redis OSS

The innovation of the vibrant open-source community.



Fast

In-memory, sub-millisecond latency



Simple

Easy-to-use CLI, intuitive API, clients
in all major programming languages



Versatile

Any data type, full text and semantic
search; JSON, Vector, and time series

Are the ones who make it possible to scale it anywhere.

Redis OSS

The innovation of the vibrant open-source community.



Fast

In-memory, sub-millisecond latency



Simple

Easy-to-use CLI, intuitive API, clients in all major programming languages



Versatile

Any data type, full text and semantic search; JSON, Vector, and time series

Redis Enterprise

An enterprise-proven real-time platform.



Global

Global distribution with local latency; supports hybrid, multi-cloud, on-prem



Enterprise-grade

Scalability, 99.999% uptime, highly secure



Cost-effective

Multi-tenancy, auto-tiering

Being production-ready at scale can be simple.

Redis OSS



Basic cache only

Self-managed

Redis Enterprise



Enterprise cache

Built, supported, and managed by Redis

Significant additional value, with full enterprise-grade functionality



Real-time data platform

Advance use cases like real-time search and vector database



Third-party Redis Services

Basic cache only

Third parties managing our open source offerings
Limited value over open source

Redis Enterprise core Data structures



Strings

"I'm a plain text string!"



Bitmaps

00110101010111001100010110



Bit fields

{23334}{112345569}{766538}



Hashes

{A: "foo", B: "bar", C: "baz"}



Lists

[A → B → C → D → E]



Key



Sets

{A,B,C,D,E}



Sorted sets

{A:0.1, B:0.3, C:100, D:1337}



Geospatial indexes

{A:(51.5, 0.12), B:(32.1, 34.7)}



Hyperlog-logs

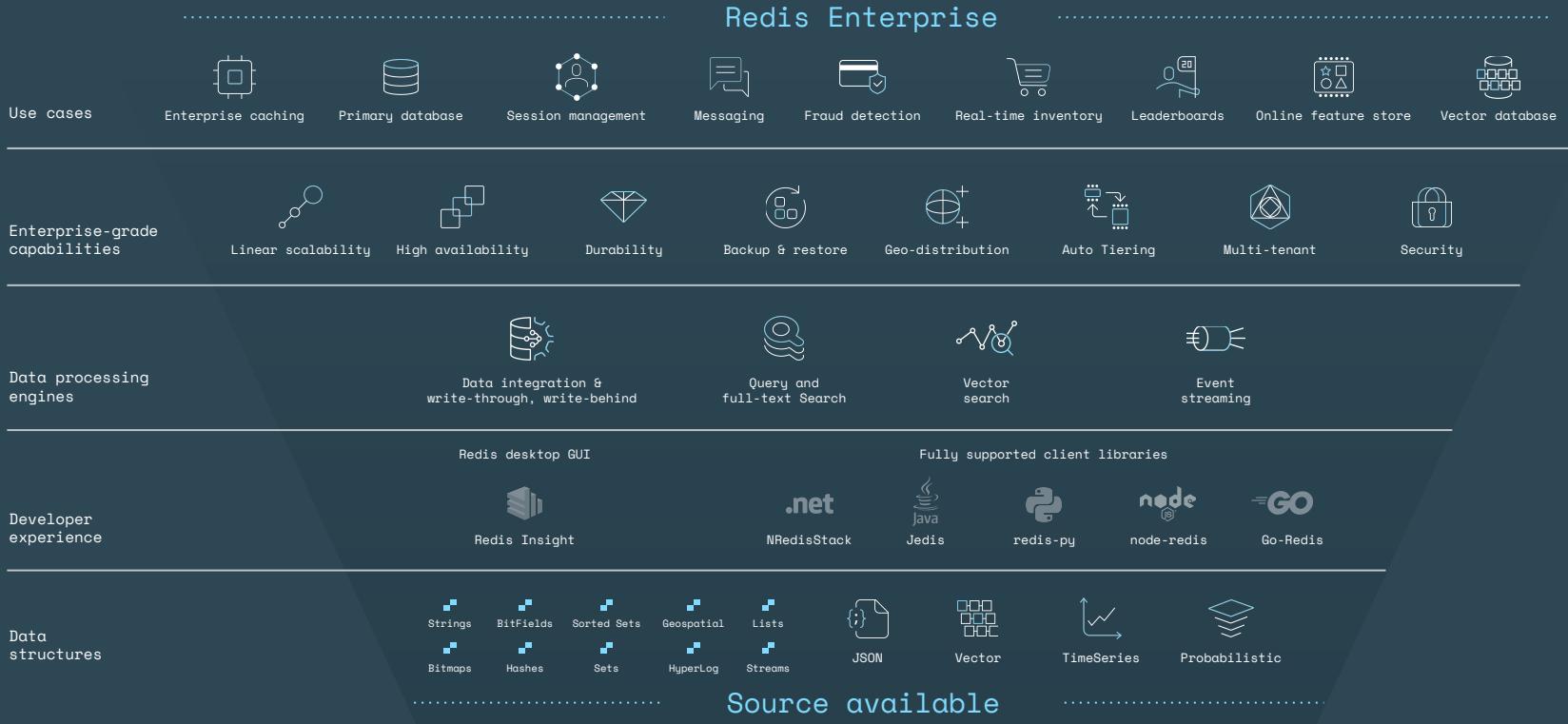
00110101 11001110 10101010



Streams

{id1=time1.seq(A:"xyz", B:"abc"),
id2=time2.seq(D:"123")}

We made the most-used database even better.



Go beyond caching.



Enterprise
caching



Primary
database



Session
management



Messaging



Fraud
detection



Real-time
inventory



Leaderboards

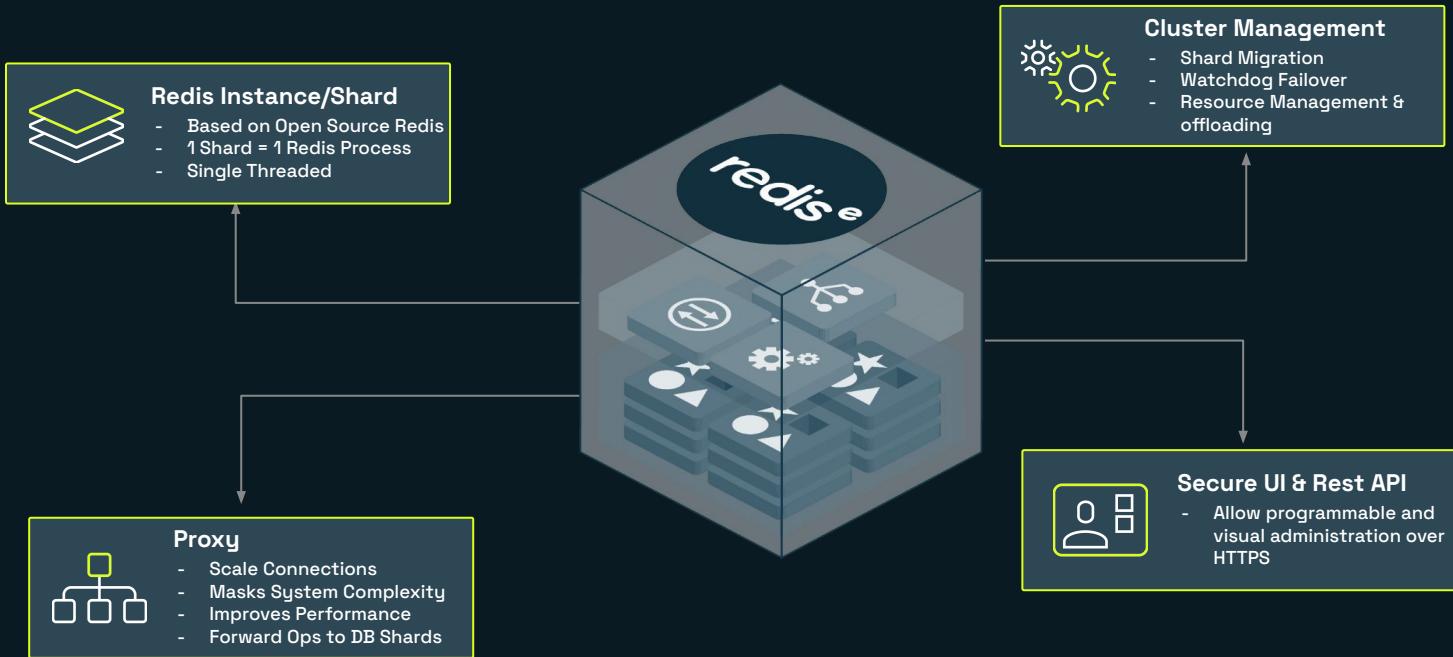


Vector
database



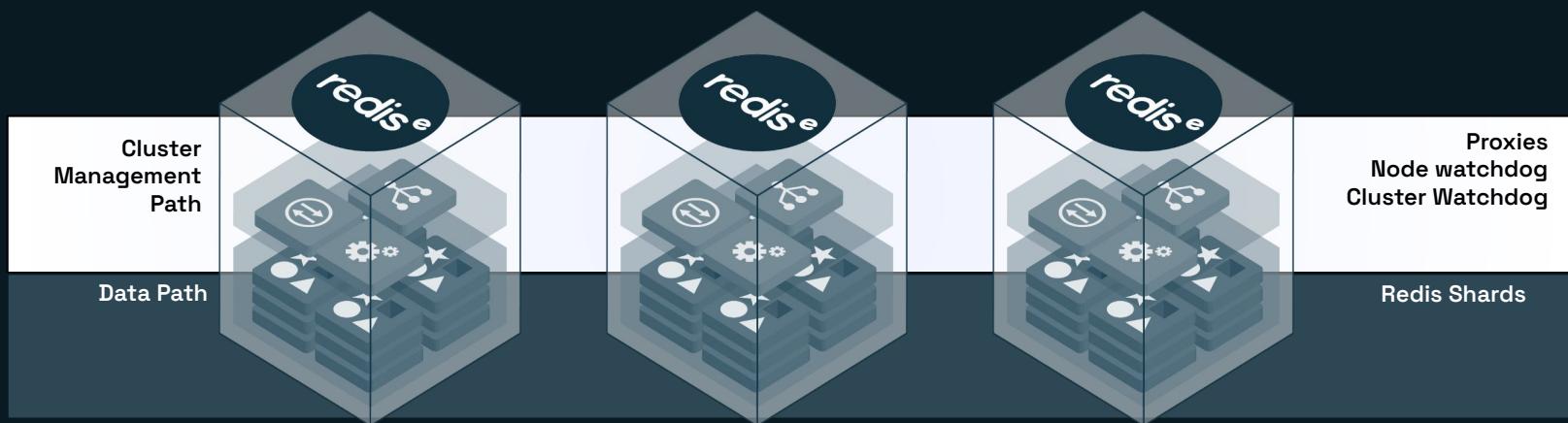
Online
feature store

Redis Enterprise node



Shared nothing symmetric architecture

Unique multi-tenant container like architecture enables running hundreds of databases over a single, average cloud instance without performance degradation and with maximum security provisions.



Redis Enterprise Terminology



Redis Instance (Shard)

- Single-threaded Redis OSS database process
- Commonly referred to as a shard which is either a primary or a replica



Redis Enterprise Database

- Logical entity that manages your entire dataset across multiple Redis Instances
- Not to be confused with the OSS database inside every Redis instance
- Multi-tenant architecture maximizing infrastructure utilization (Reduces overall TCO)

Redis Enterprise Nodes

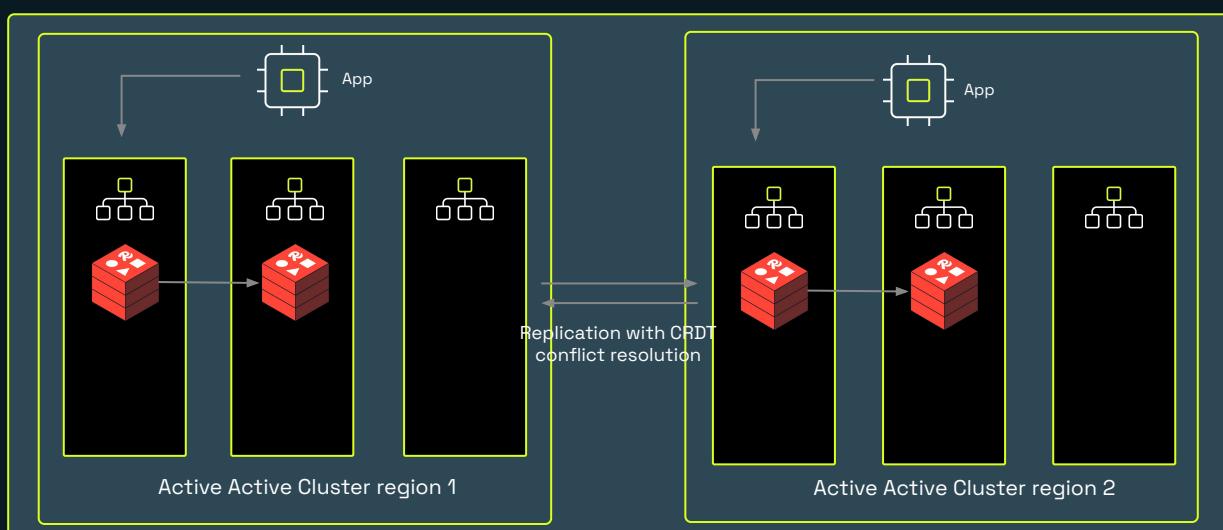
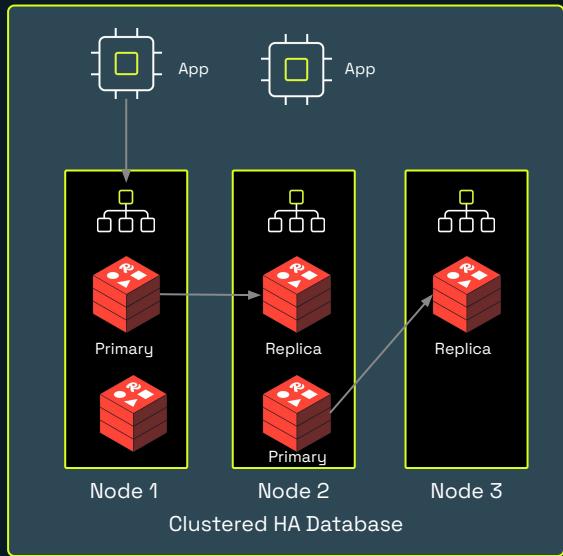
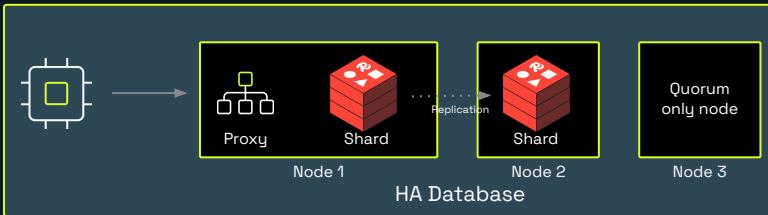
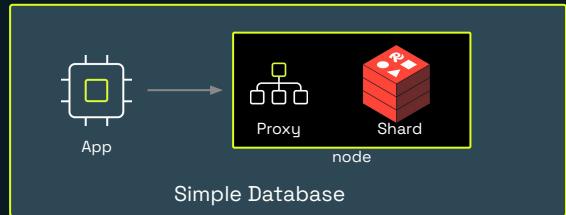
- Physical servers, virtual machines, containers, and/or cloud instances on which the Redis Enterprise installation package is installed and configured
- Each node is a container capable of running multiple Redis Instances ("shards")



Redis Enterprise Cluster

- Set of Redis Enterprise nodes pooling resources
- Allows you to create any number of Redis Enterprise nodes to scale up/out

Topologies



Redis Enterprise Proxy

The Redis Enterprise Proxy is a low-latency intermediary that simplifies application interaction with the database by providing a single endpoint for client connections and masking the complexity of the database's topology. It enables efficient traffic management and scaling without requiring developers to handle these aspects

Utilizes techniques like multiplexing and pipelining for higher throughput

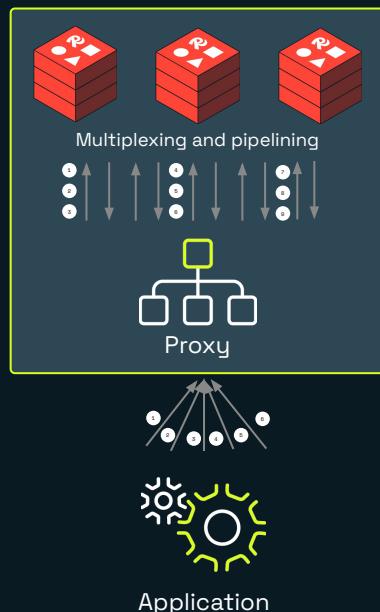
Helps in maintaining persistence in connections with the clients

Provides a single endpoint for connections, simplifying application interaction with the database

Manages traffic efficiently, even during node loss, by spreading connections across nodes

Executes multikey commands across multiple shards

Maintains uninterrupted client access to the database endpoint during topology changes



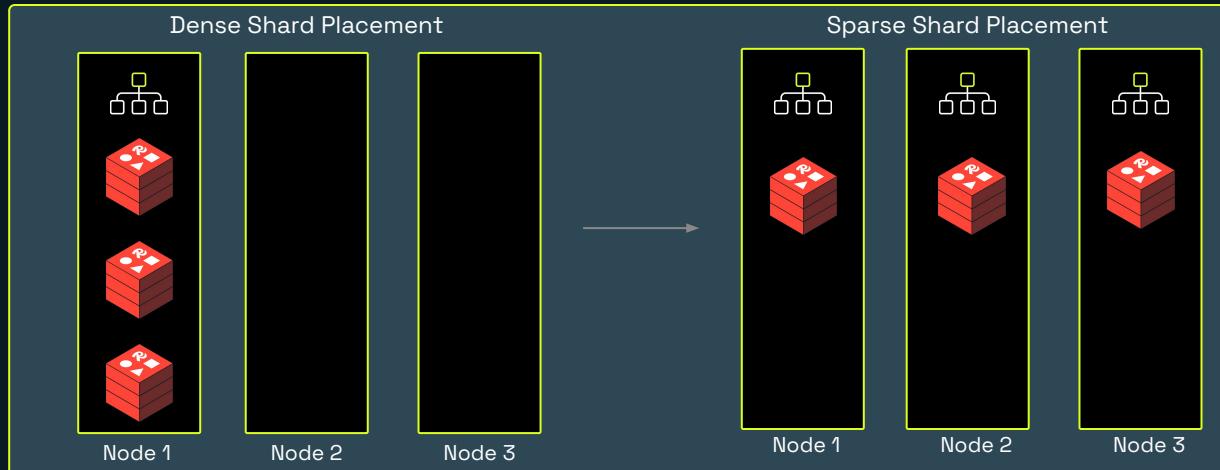
Proxy Policies

Single Proxy - Only one node will have the proxy enabled. All the clients connect this node.

Each node would need to have several unutilized CPU cores to be able to take the traffic in case of a failure. The single proxy policy is designed to handle a large number of operations efficiently without consuming additional resources

Multiple proxies - Each node will have proxy enabled.

The connections are evenly spread across the nodes. Losing a node, means that the surviving nodes need to be able to handle the additional traffic (i.e. of 1 node or max. 1 AZ) - (this option allows > CPU usage and can allow for more connections and greater throughput, should be combined with All-primary shards policy)



High Availability

Topologies for High availability

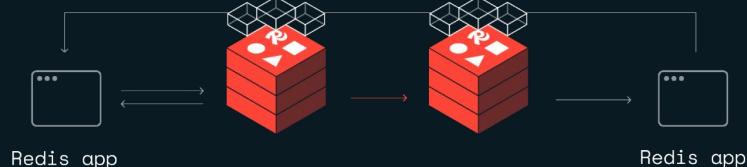
ACTIVE-PASSIVE

- Passive server is a cold standby
- Uses: high availability, disaster recovery, data durability



ACTIVE-READ/REPLICA

- Read-replica is available in read-only mode
- Uses: distributed caching



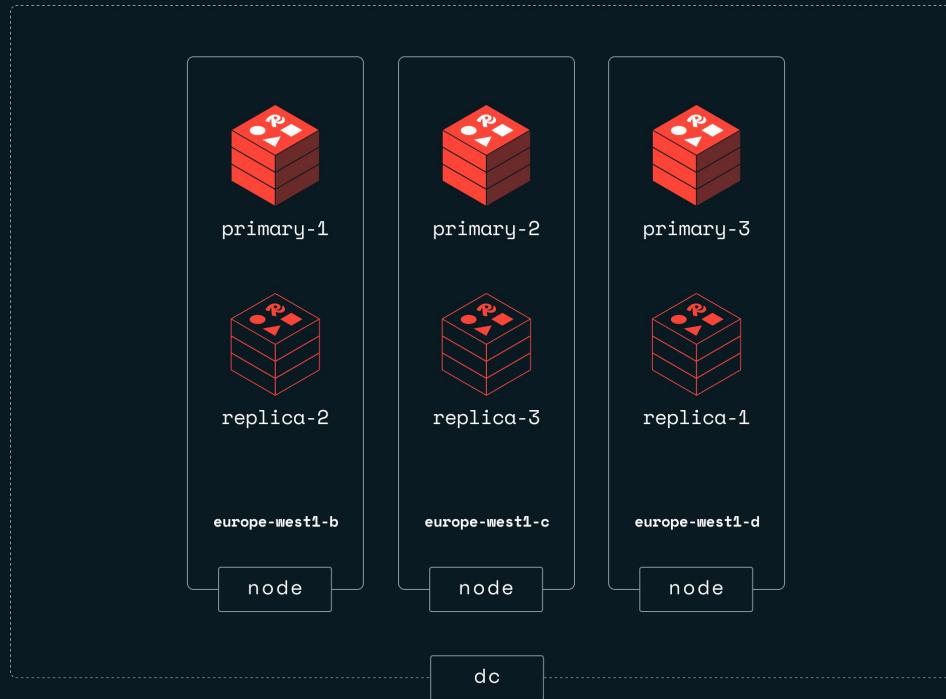
ACTIVE-ACTIVE

- All database instances are available for read and write operations
- Uses: local latencies for geo-distributed apps, load distribution, data consolidation



Rack Zone Awareness

Redis can ensure replica shard placement exists in alternate zones



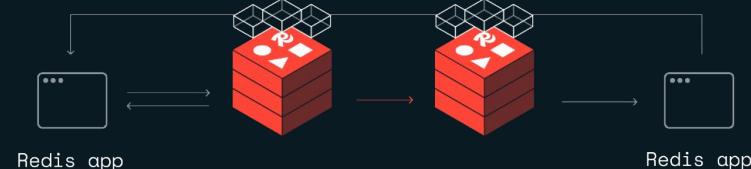
ACTIVE-PASSIVE

- Passive server is a cold standby
- Uses: high availability, disaster recovery, data durability



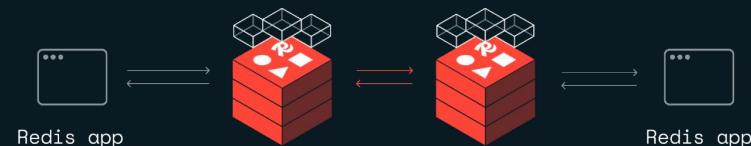
ACTIVE-READ/REPLICA

- Read-replica is available in read-only mode
- Uses: distributed caching

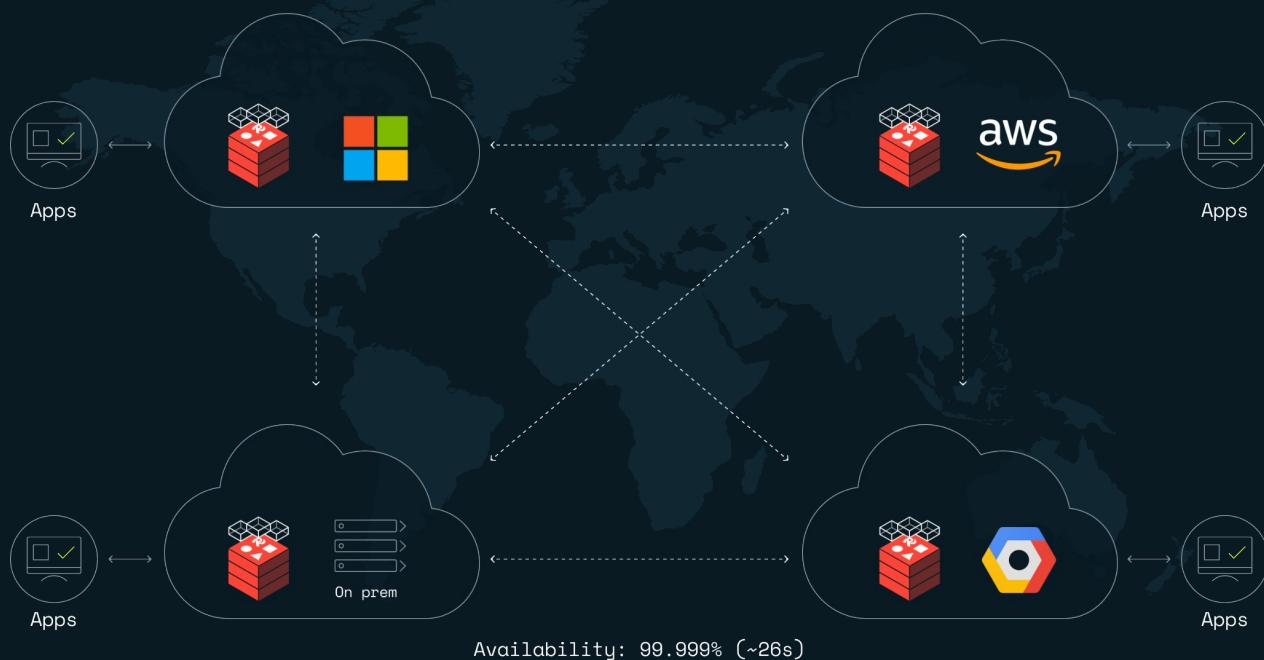


ACTIVE-ACTIVE

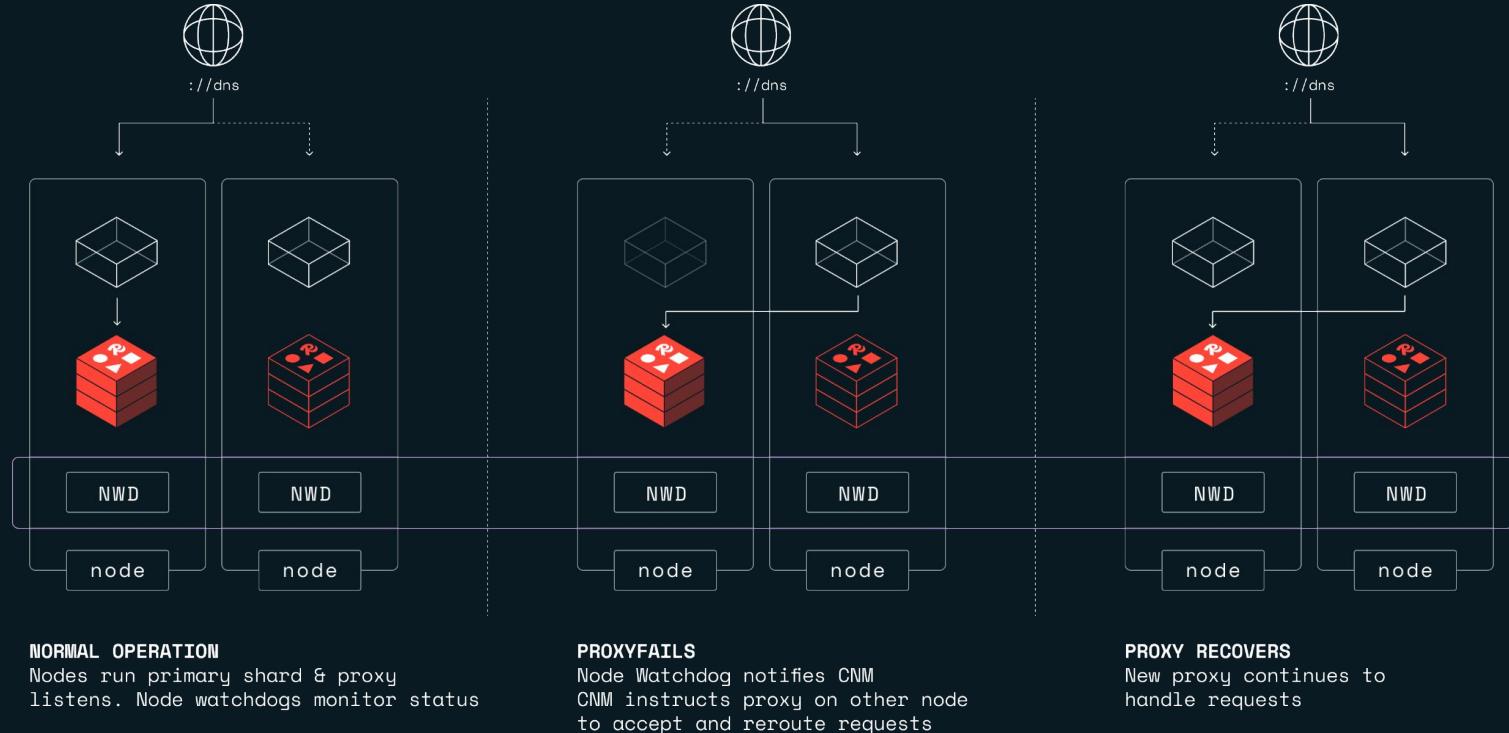
- All database instances are available for read and write operations
- Uses: local latencies for geo-distributed apps, load distribution, data consolidation



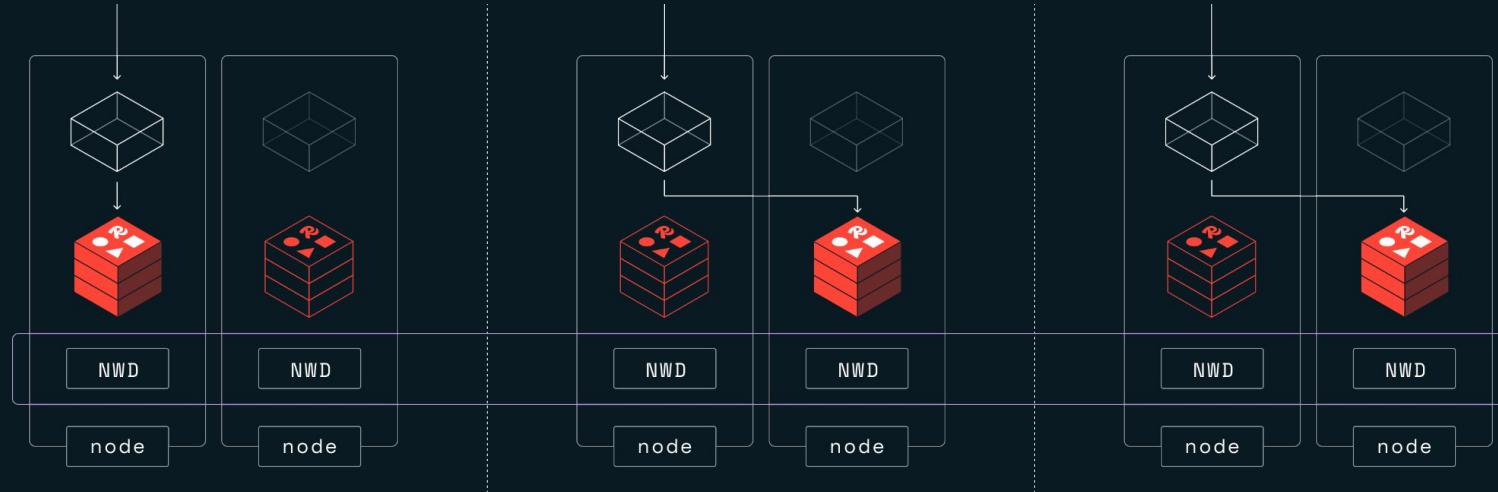
Multi Cloud Support



Proxy Failover



Shard Failover



NORMAL OPERATION

Node runs primary shard & proxy
listens
Node watchdogs monitor status

MASTER SHARD FAILS

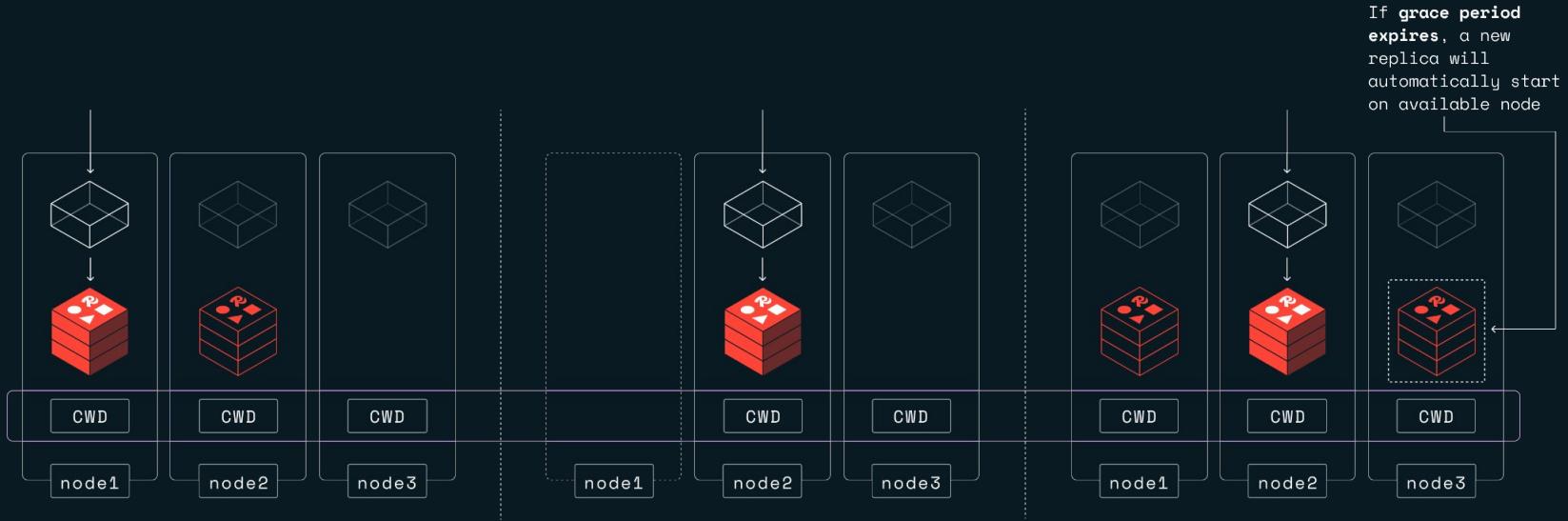
Node watchdog notifies CNM
CNM promotes replica shard
Proxy re-routes request

LOST SHARD RECOVERS

Shard synchronizes
Proxy continues re-routing requests

NWD = Node watchdog

Node Failover



NORMAL OPERATION

Node runs primary shard & proxy listens
Node watchdogs monitor status

NODE WITH MASTER FAILS

Cluster watchdogs notifies CNM
CNM instructs proxy to auto-failover
Proxy failovers via DNS

NODE RECOVERS

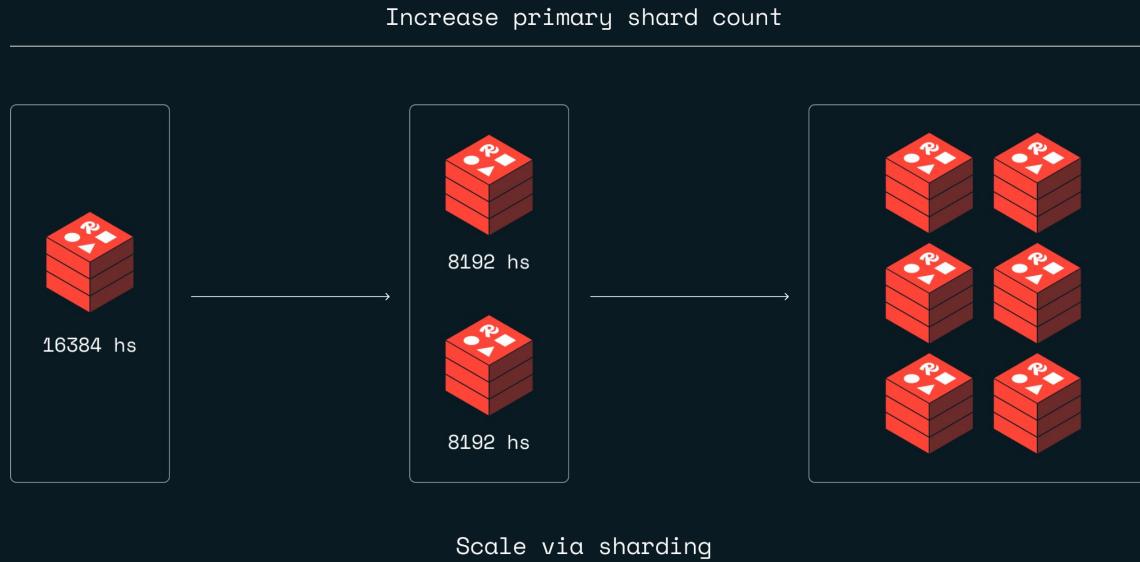
Nodes recovers and replica shard re-syncs

CWD = Cluster watchdog

Scaling

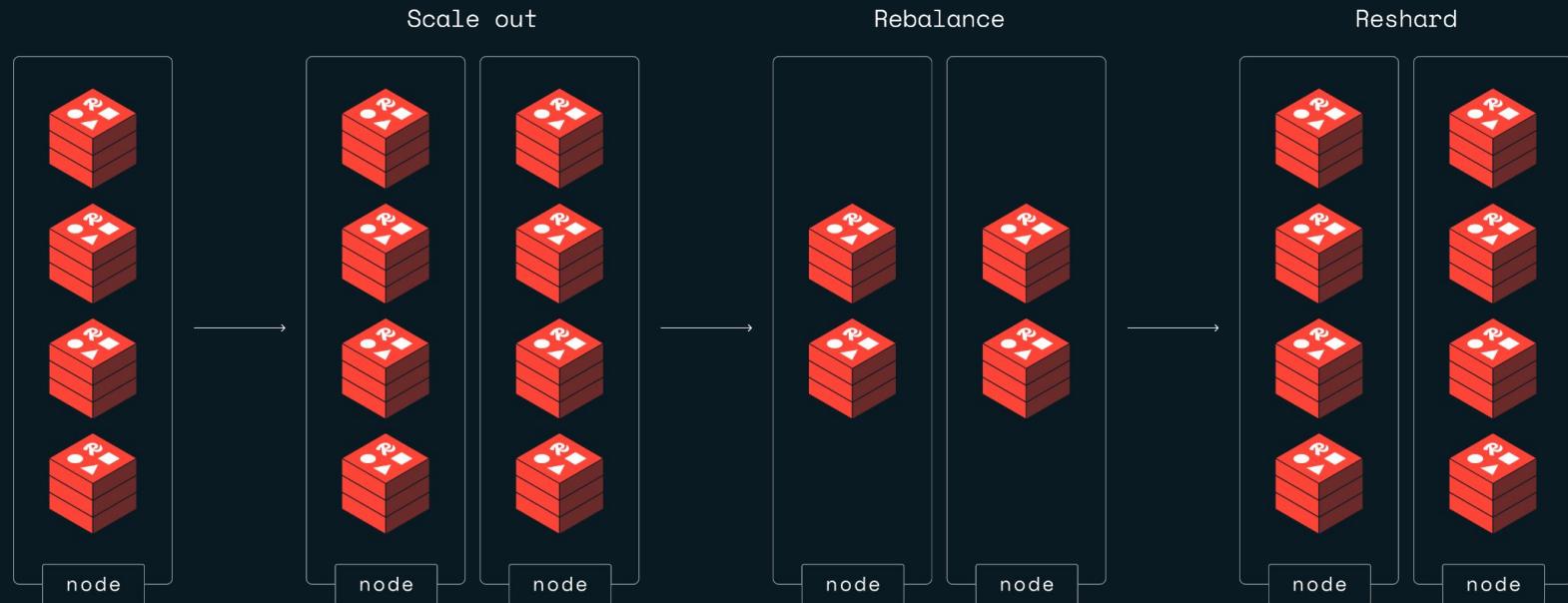
Vertical Scaling

This diagram shows how to increase the number of primary shards of an existing database to scale up (vertically) on a single node.

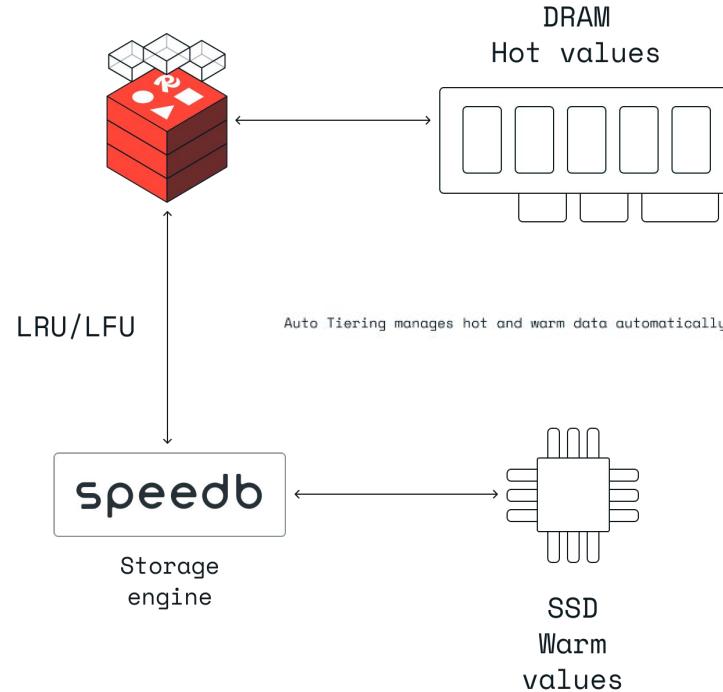


Horizontal Scaling

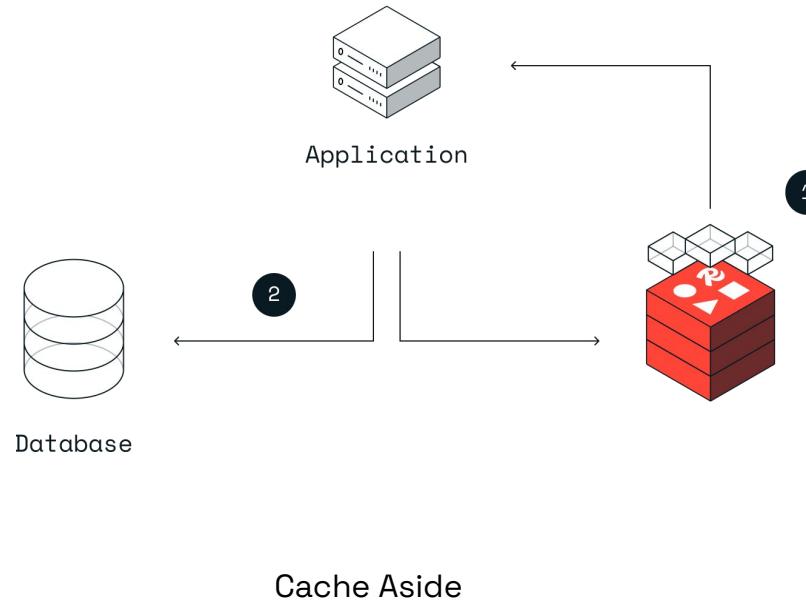
Add additional Redis nodes and scale out



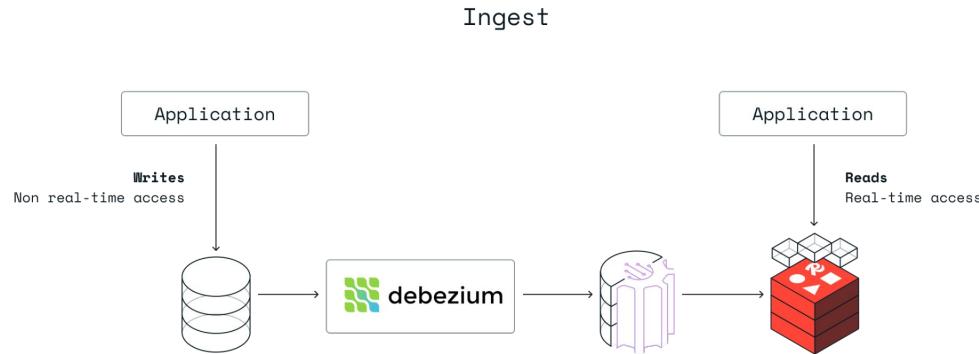
Auto-Tiering



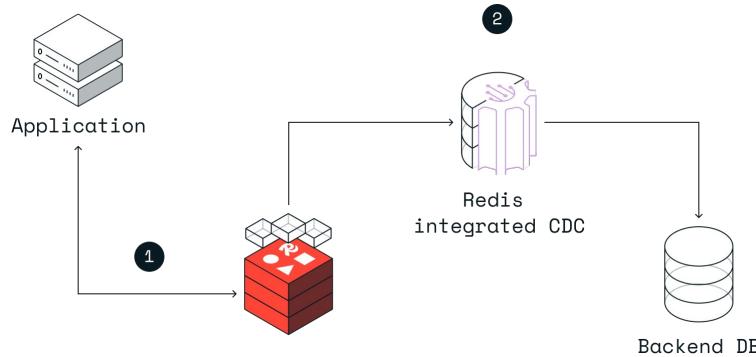
Caching Patterns



Caching Patterns



Caching Patterns



Write Behind

