



In-memory databases
Use cases and solutions for
modern apps

Patrocinadores

Patrocinadores Principales



Patrocinadores Platino



Patrocinadores Oro

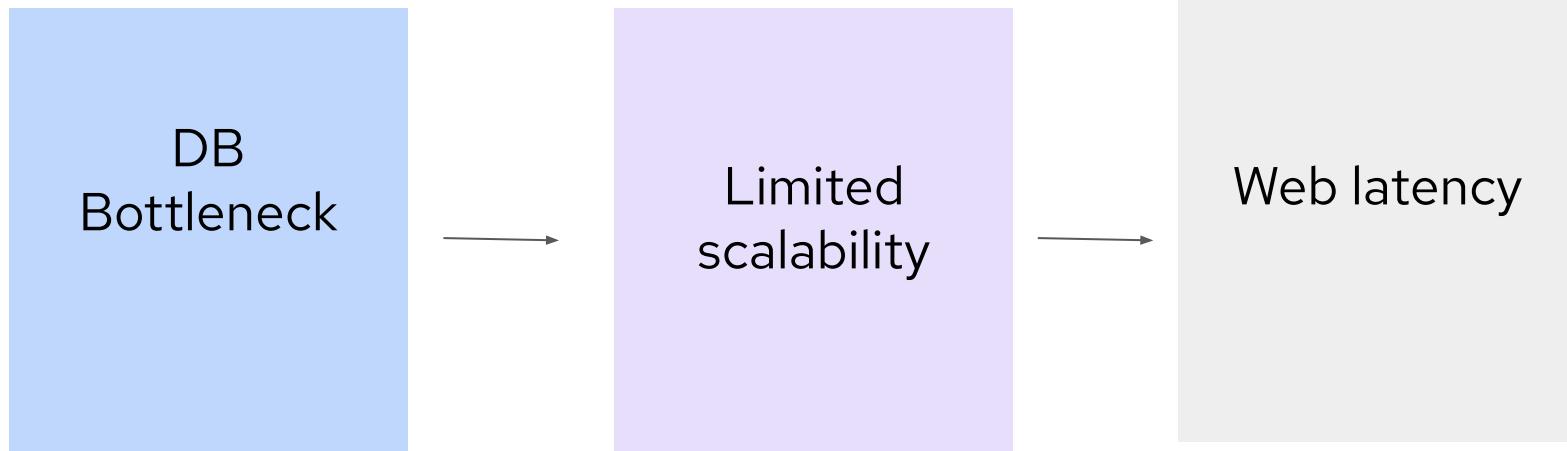


Patrocinadores Plata



Patrocinadores Bronce





What can we do ?





Katia Aresti

- Principal Software Engineer
Red Hat (Infinispan, Quarkus)
- Linkedin Learning
- Devoxx France, Java Champion
- NoSoloTech (10 ep.)
- Les Cast Codeurs Podcast

DISCLAIMER



Community

Infinispan is created and maintained by a group of open-source contributors who welcome your participation.

a Red Hat sponsored project



An Open Project

Infinispan is a 100% open source project, under the [Apache Public License 2.0](#).

Not only is the code open source, free to use and free to modify, the development process is also open.

Code and Issue Tracker

The source code is hosted on [GitHub](#). If you need to report a bug or request a new feature, look for a similar one on our [GitHub issues tracker](#). If you don't

Github discussions

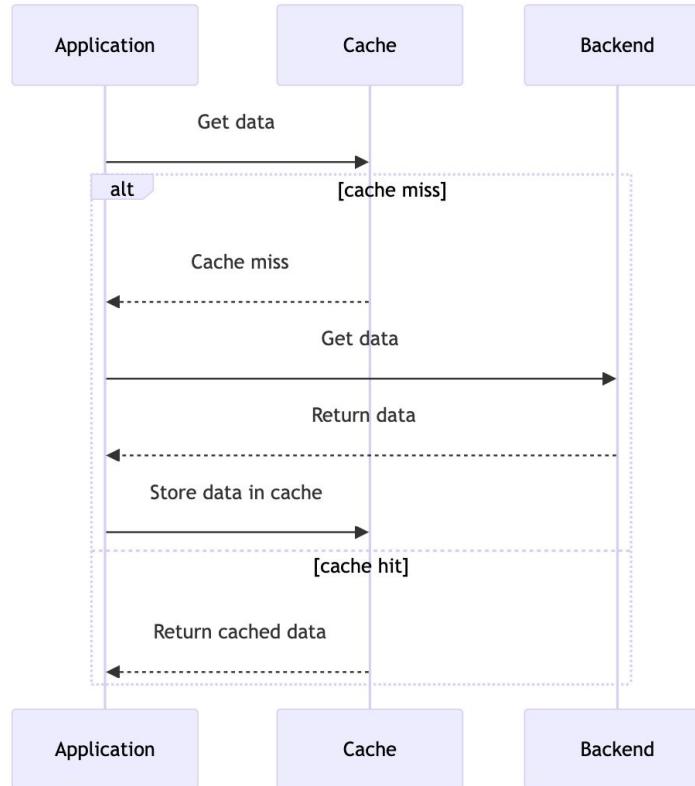
If you have questions, are experiencing a bug or want advice on using Infinispan, you can use [GitHub discussions](#). We will do our best to answer you as soon as we can.

Chat with us

The Infinispan community uses [ZulipChat](#) for real-time communications. Join us using either a web-browser or a [dedicated application](#) on the [Infinispan chat](#).

Caching library

Caso de uso: Caché simple



- High speed
- Load reduction
- Economic efficiency
- Adds availability
- Flexible data management
(expiration)

**Programming
Language**

Library/Framework

Java Caffeine (Java Cache Library)

Python `cachetools, diskcache`

JavaScript `node-cache, lru-cache`

C# `LazyCache, CacheManager`

Go `groupcache, golang-lru`

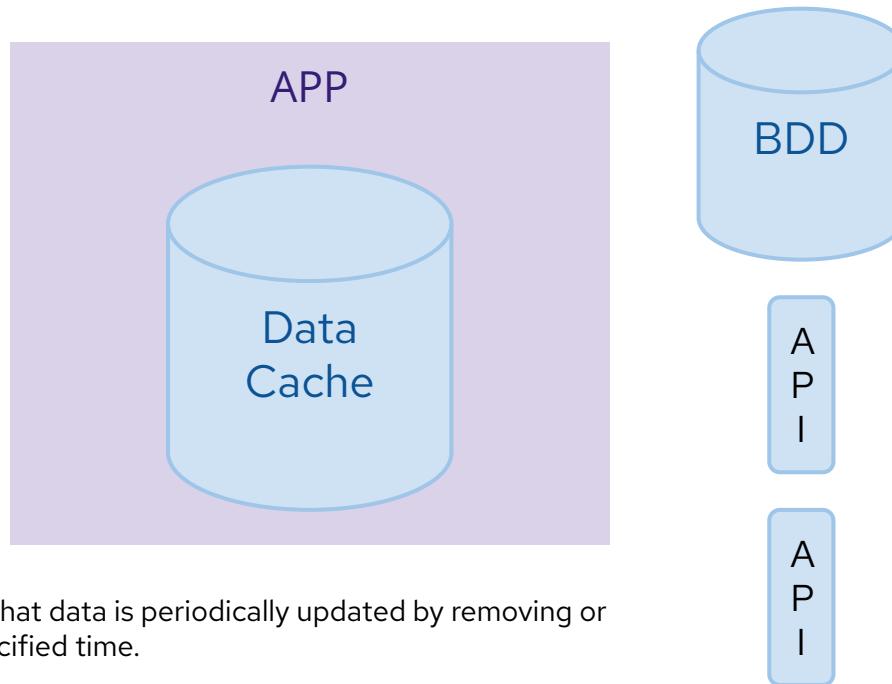
Ruby `Dalli, Ruby-LRU`

PHP `PHP-Cache, Laravel
Cache`

Rust `cached, lru-cache`

Kotlin Caffeine (same as Java)

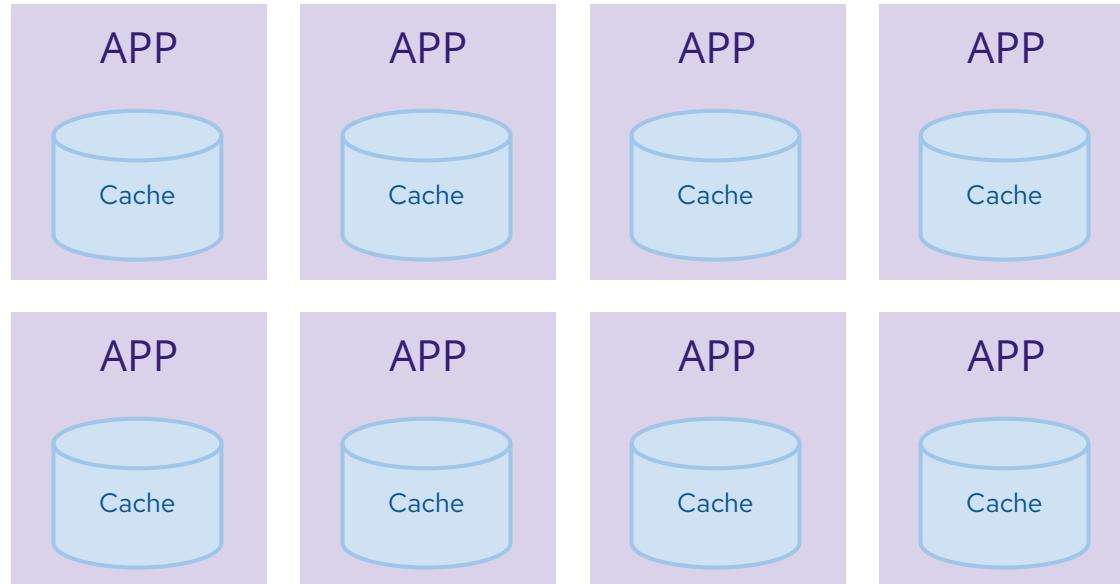
Caching library



TTL (Time-to-Live) ensures that data is periodically updated by removing or refreshing elements after a specified time.

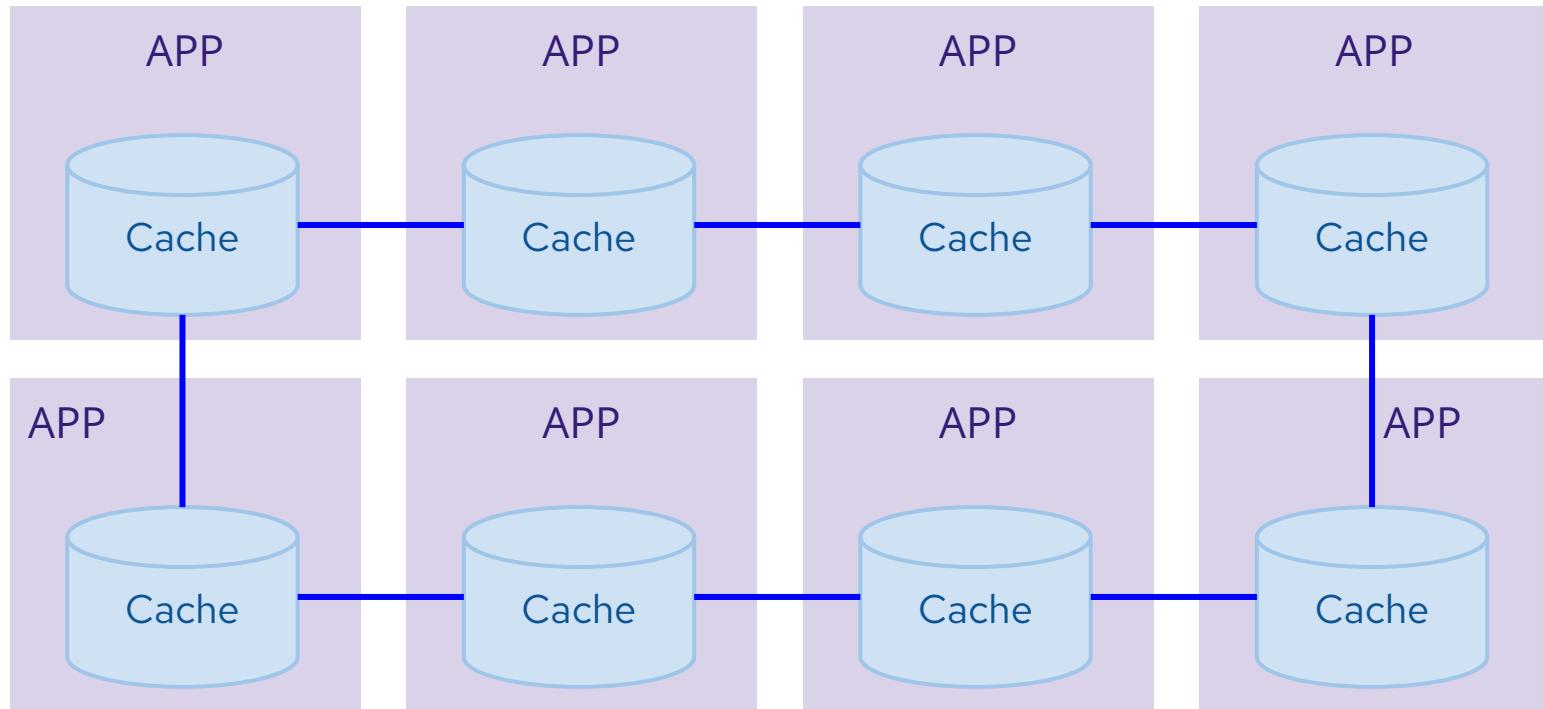
MAX IDLE is the maximum time an item can remain inactive in the cache before being removed, even if it hasn't expired due to TTL.

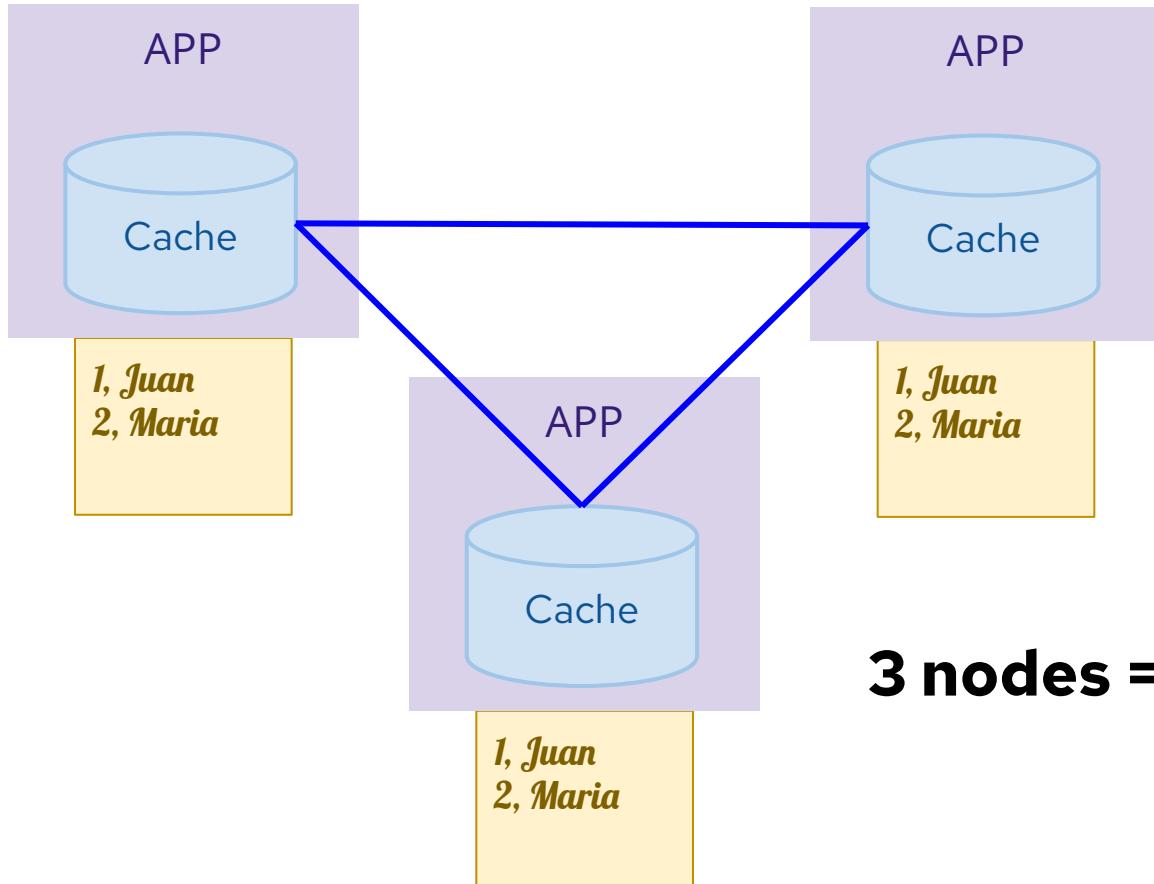
Multiple instances



Distributed cache

Apps and inside caches form a cluster

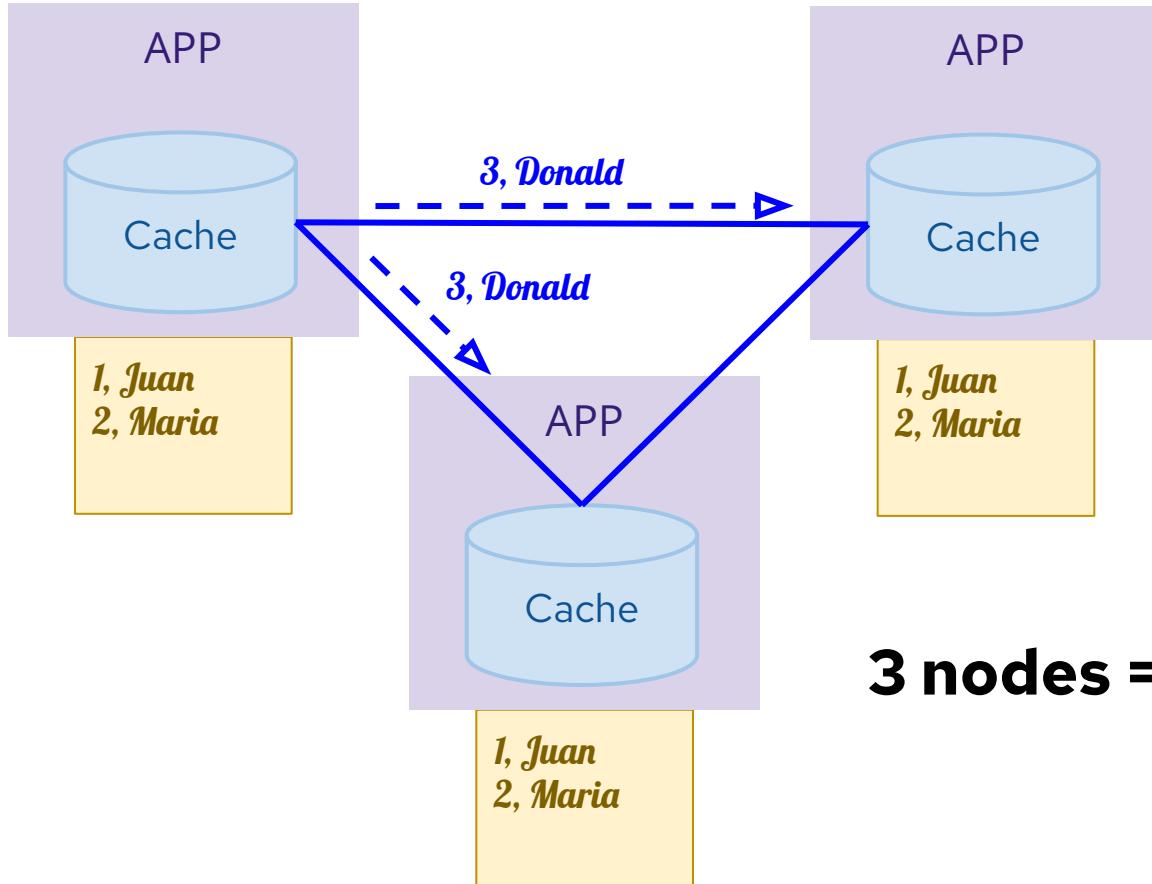




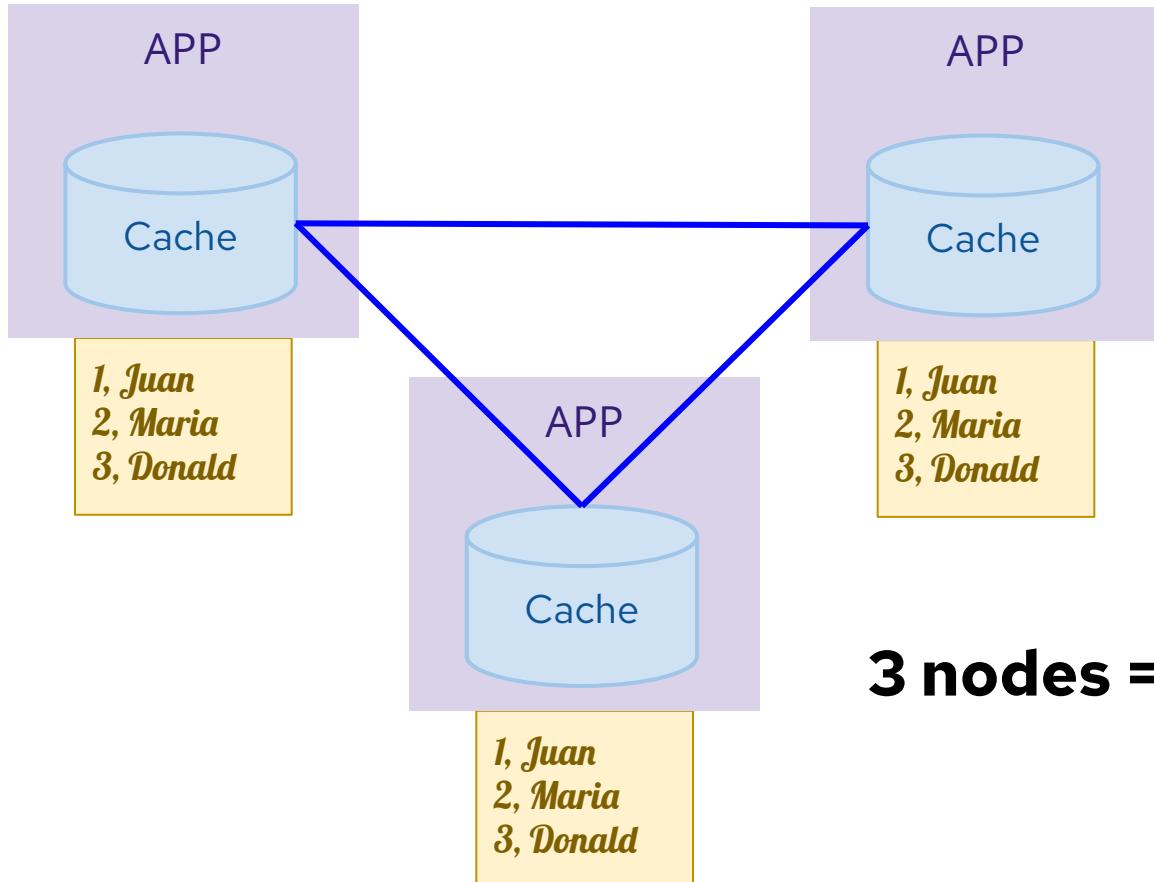
3 nodes = 3 copies

PUT

3, Donald

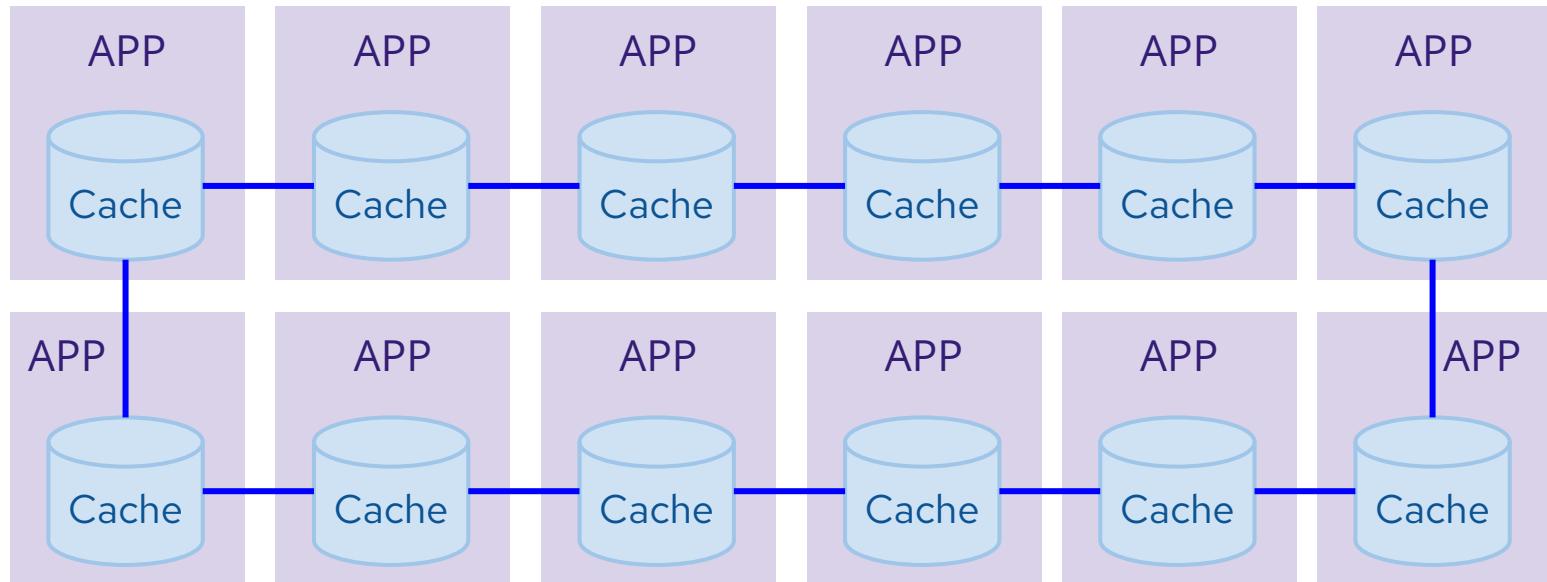


3 nodes = 3 copies

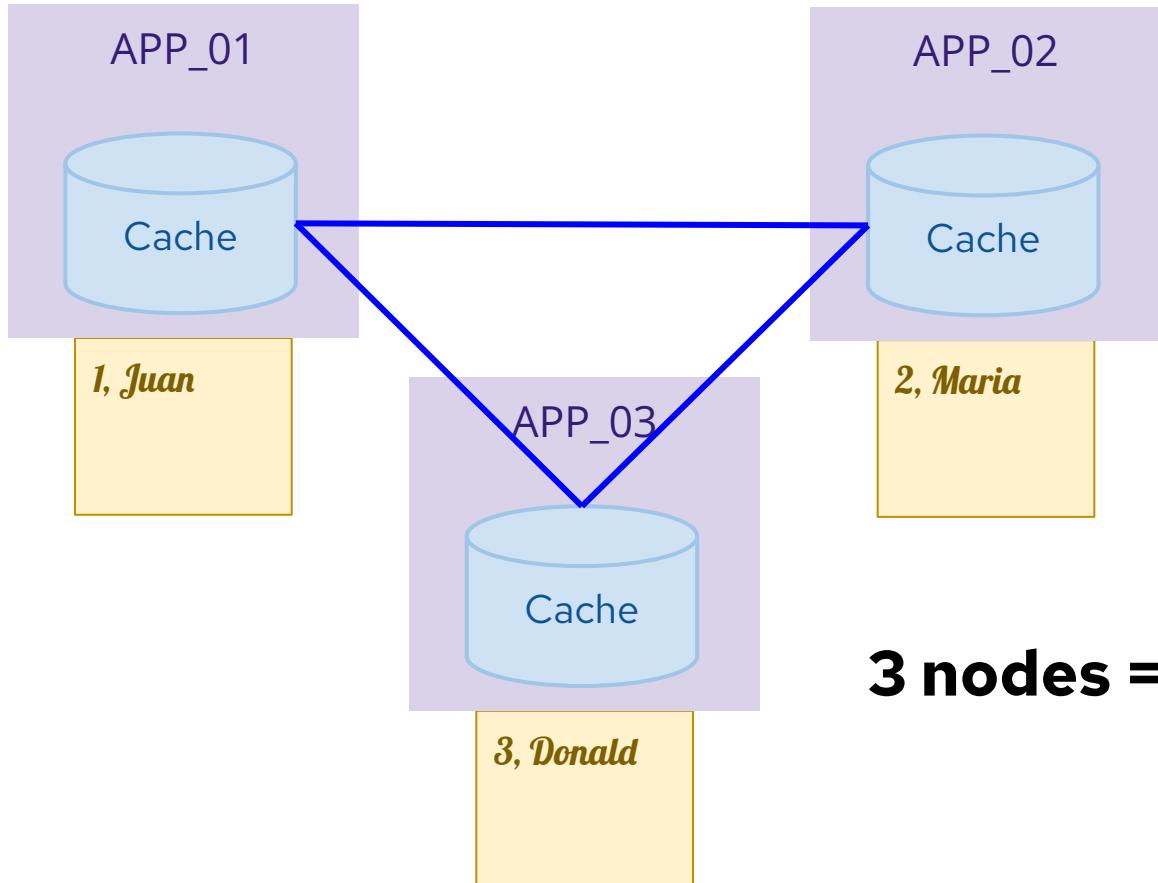


3 nodes = 3 copies

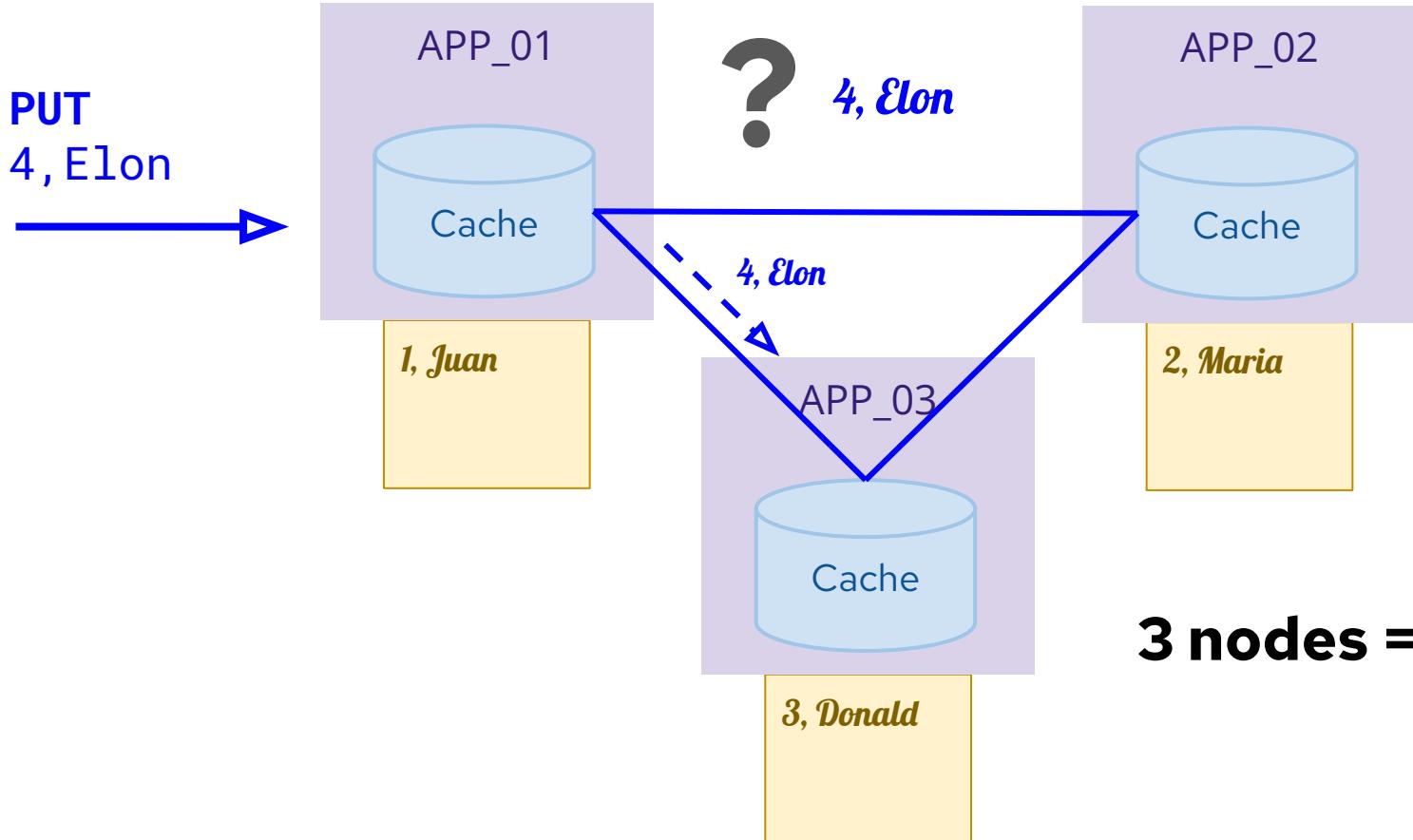
Does no scale

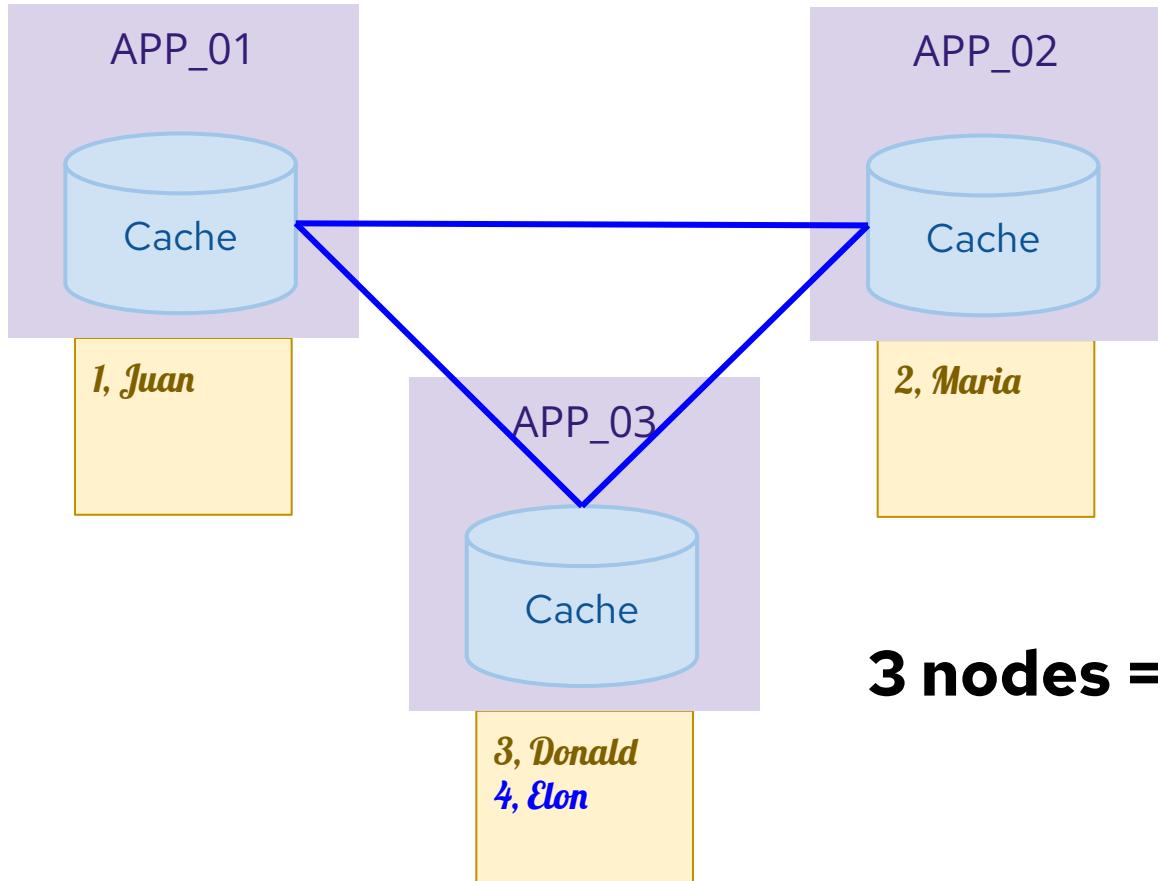


12 nodes = 12 copies



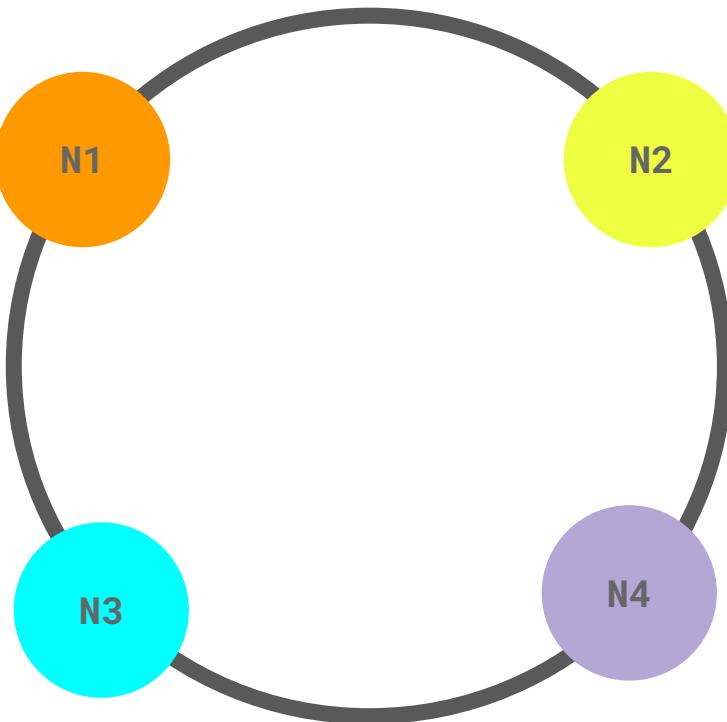
3 nodes = 1 copy

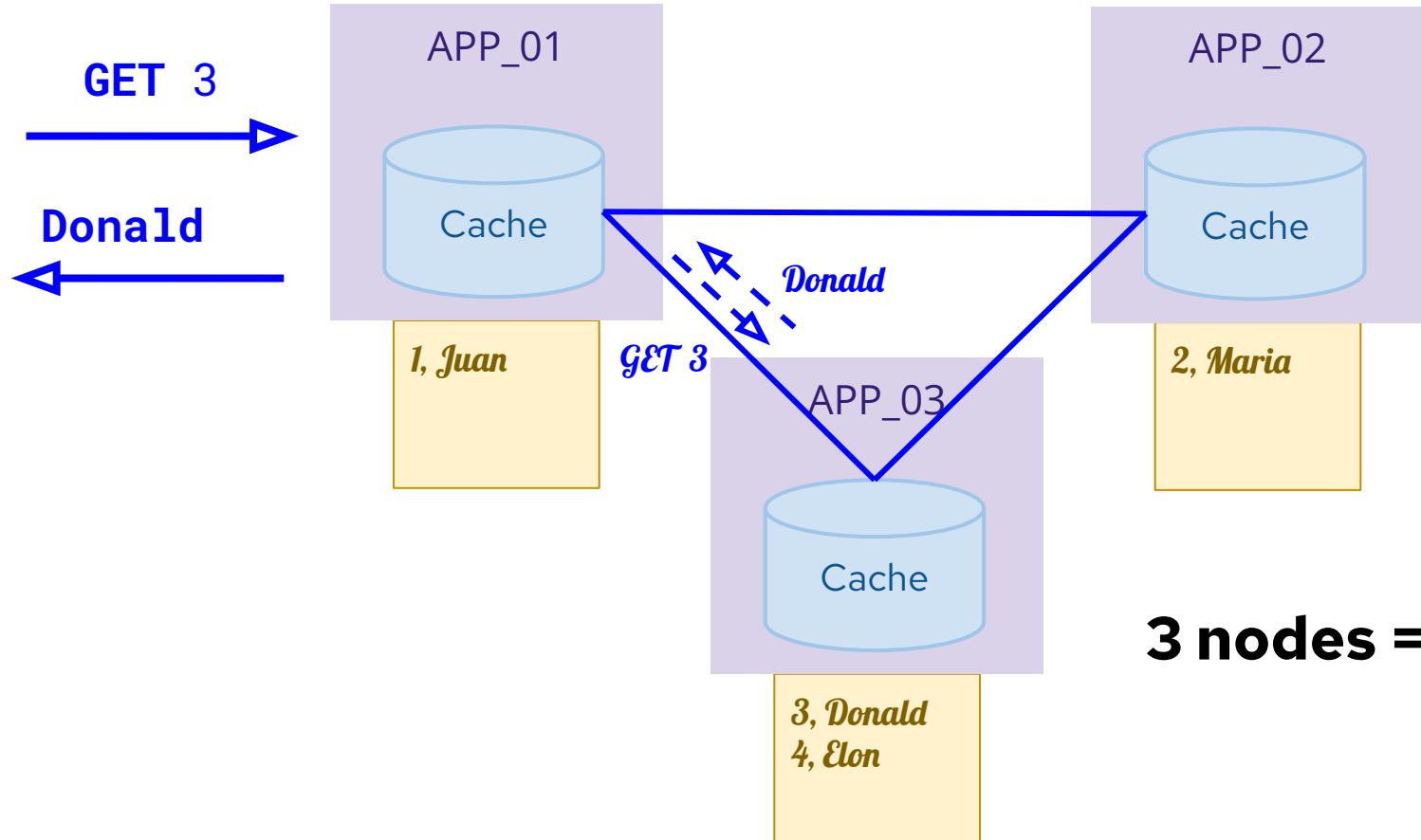


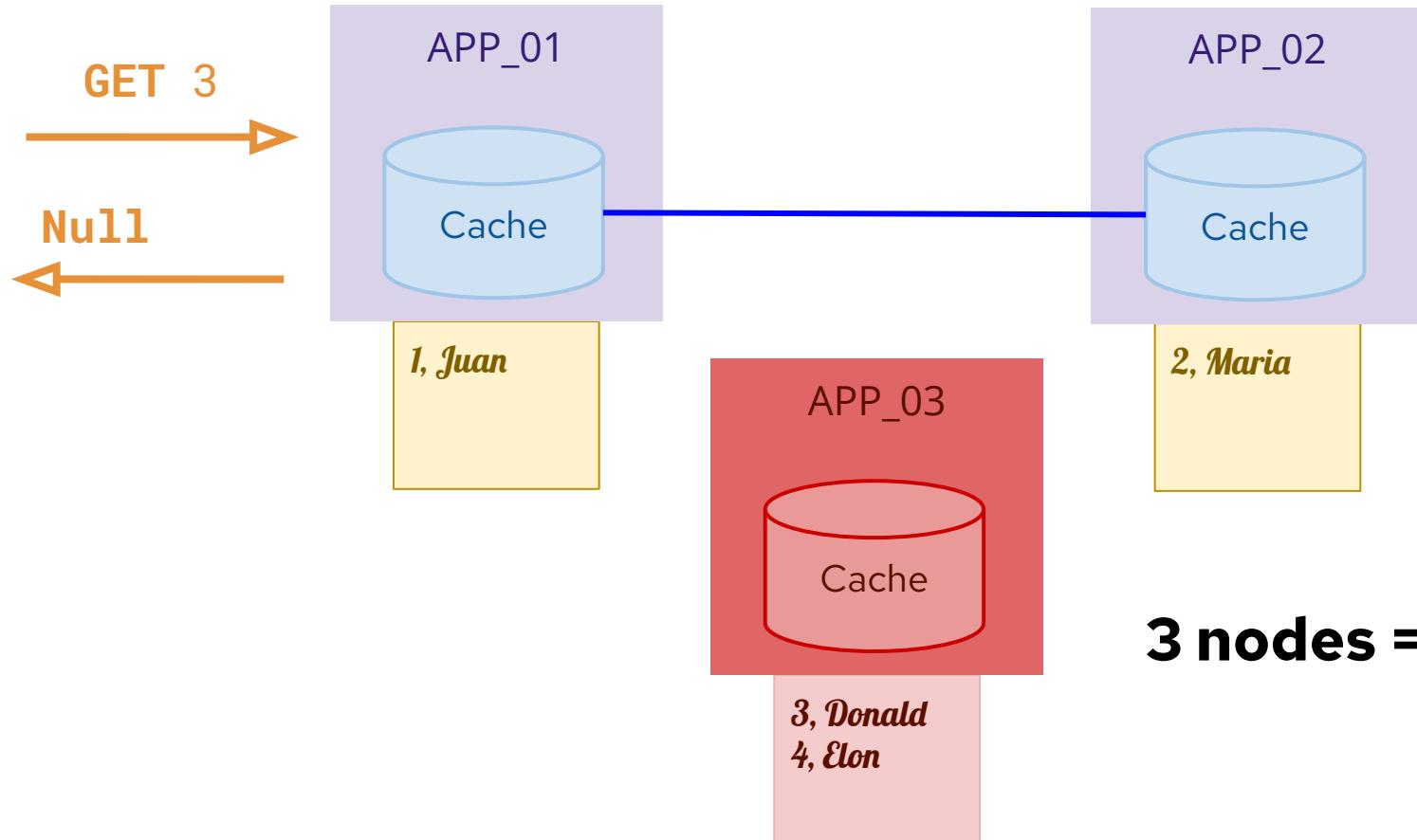


3 nodes = 1 copy

Consistent Hashing (or other system)

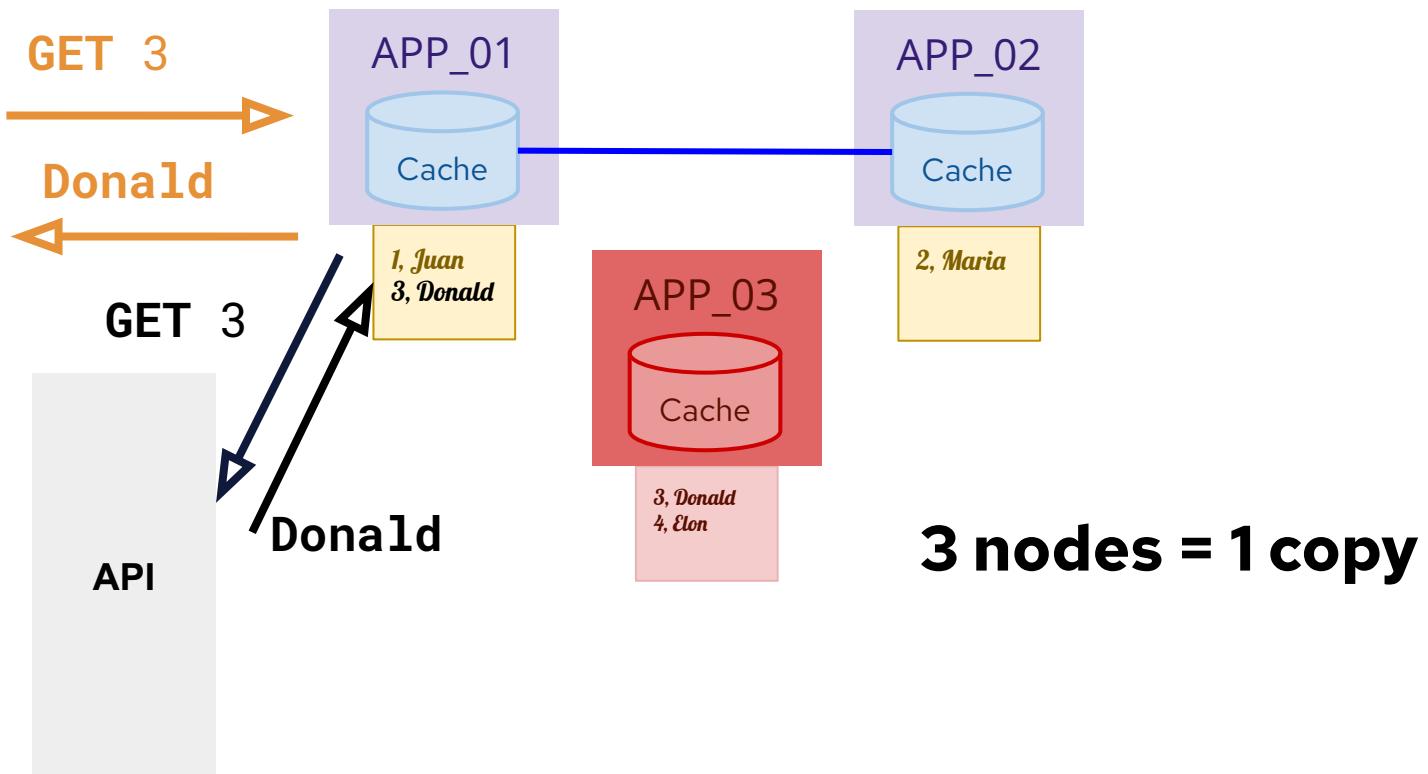




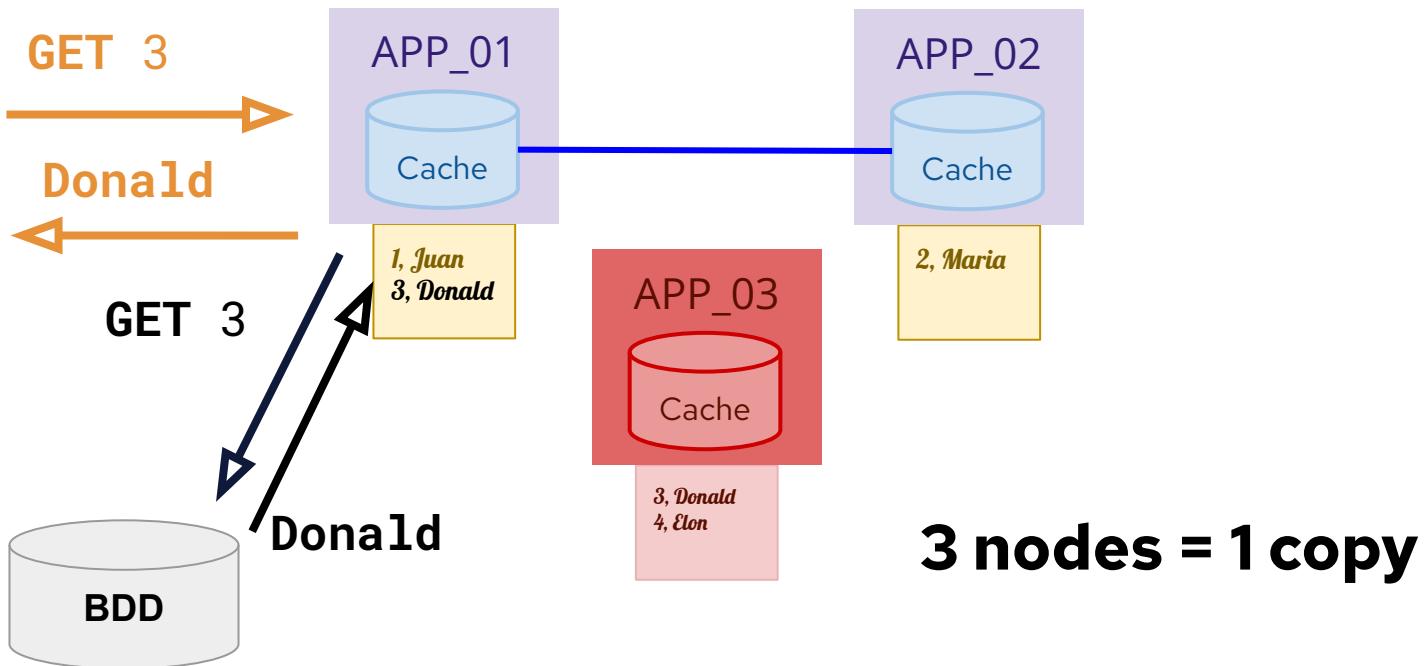


3 nodes = 1 copy

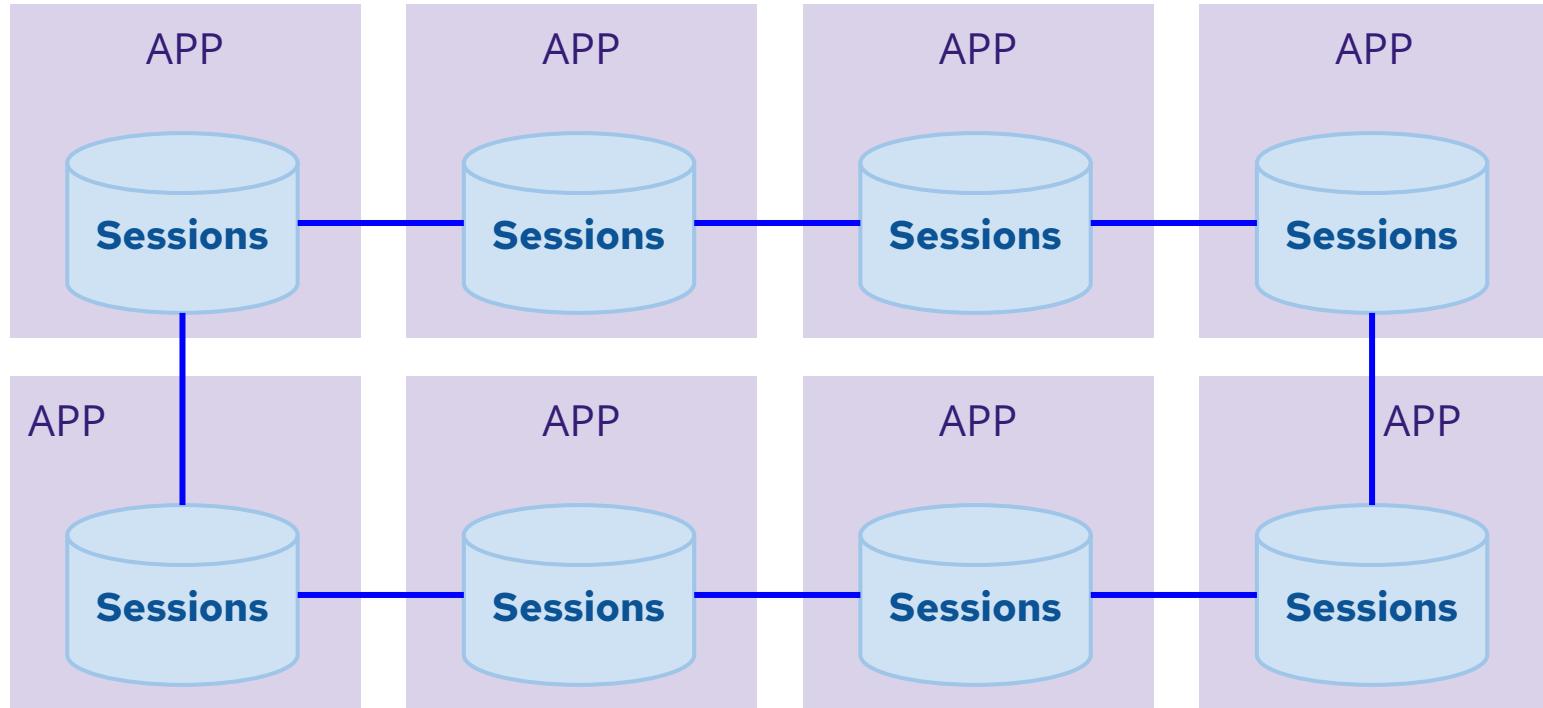
Caso de uso: Almacenamiento datos temporales

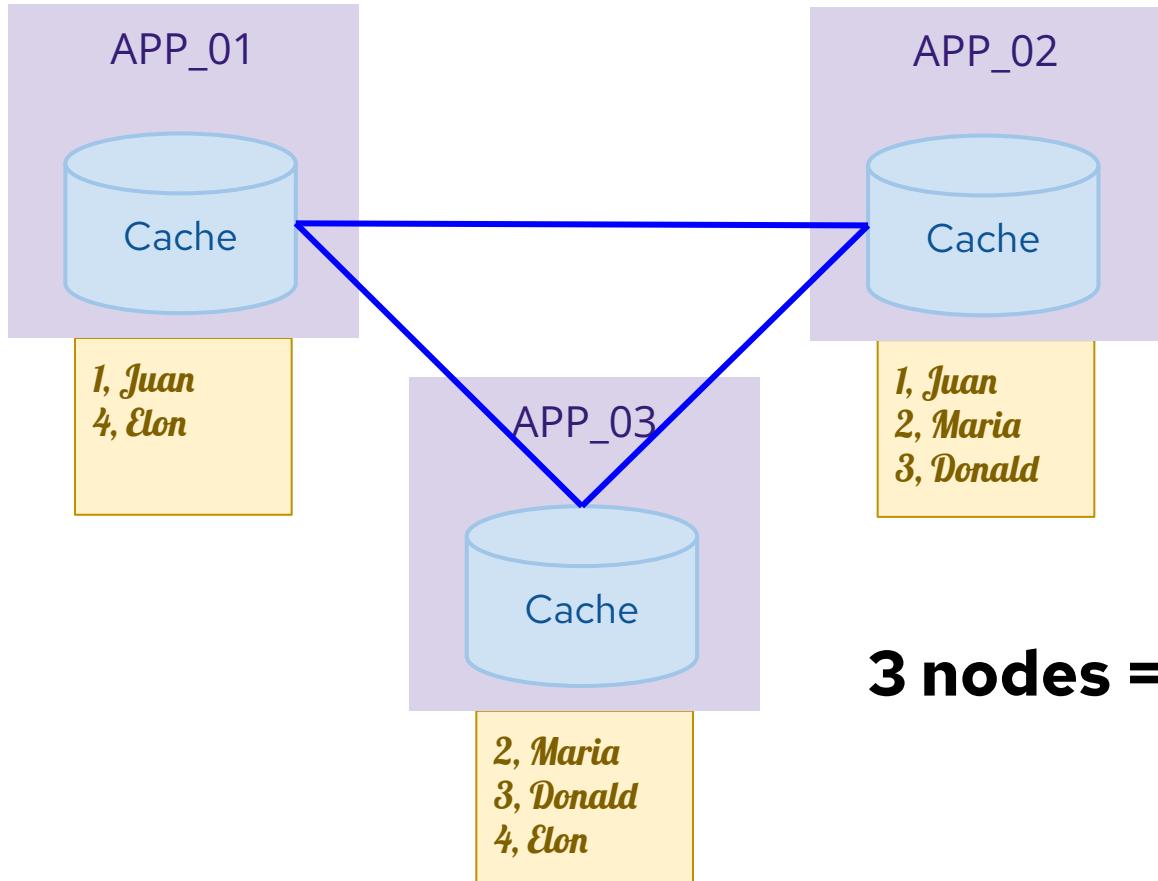


Caso de uso: Almacenamiento datos temporales

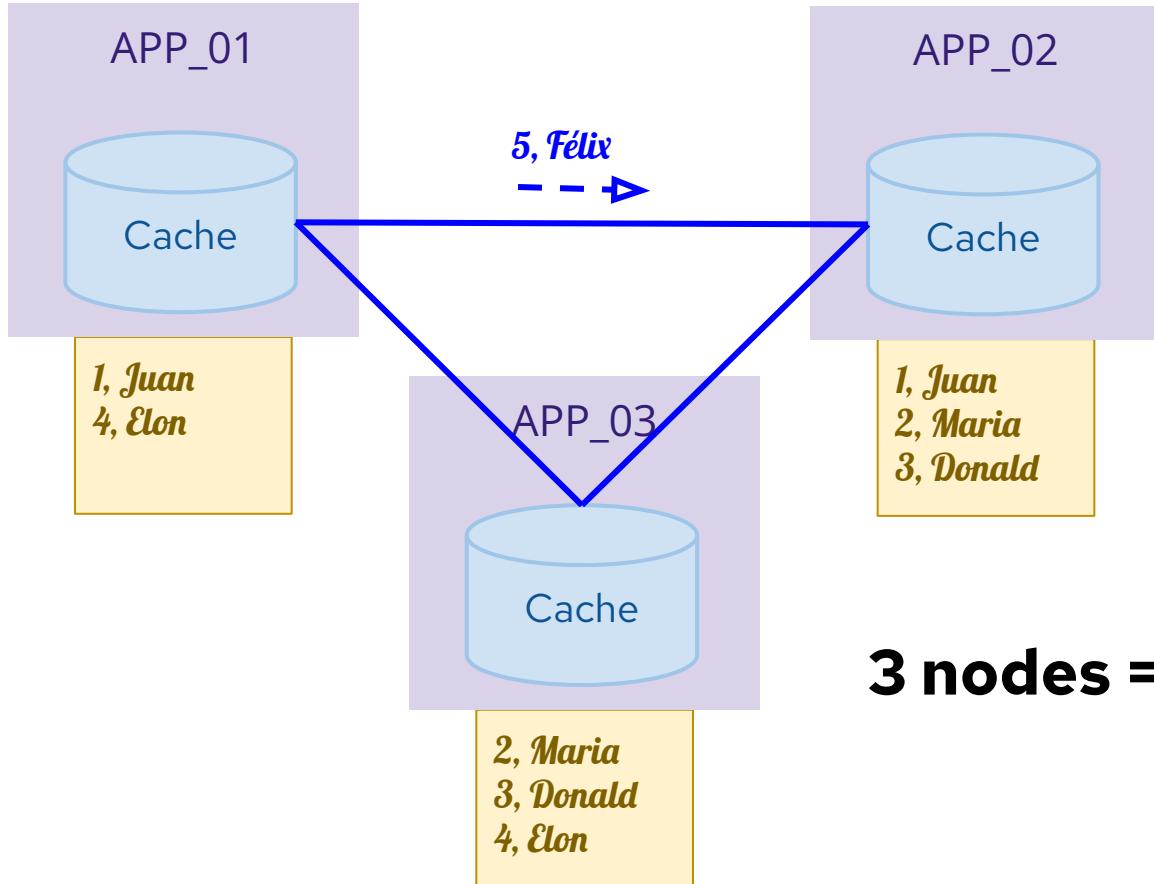


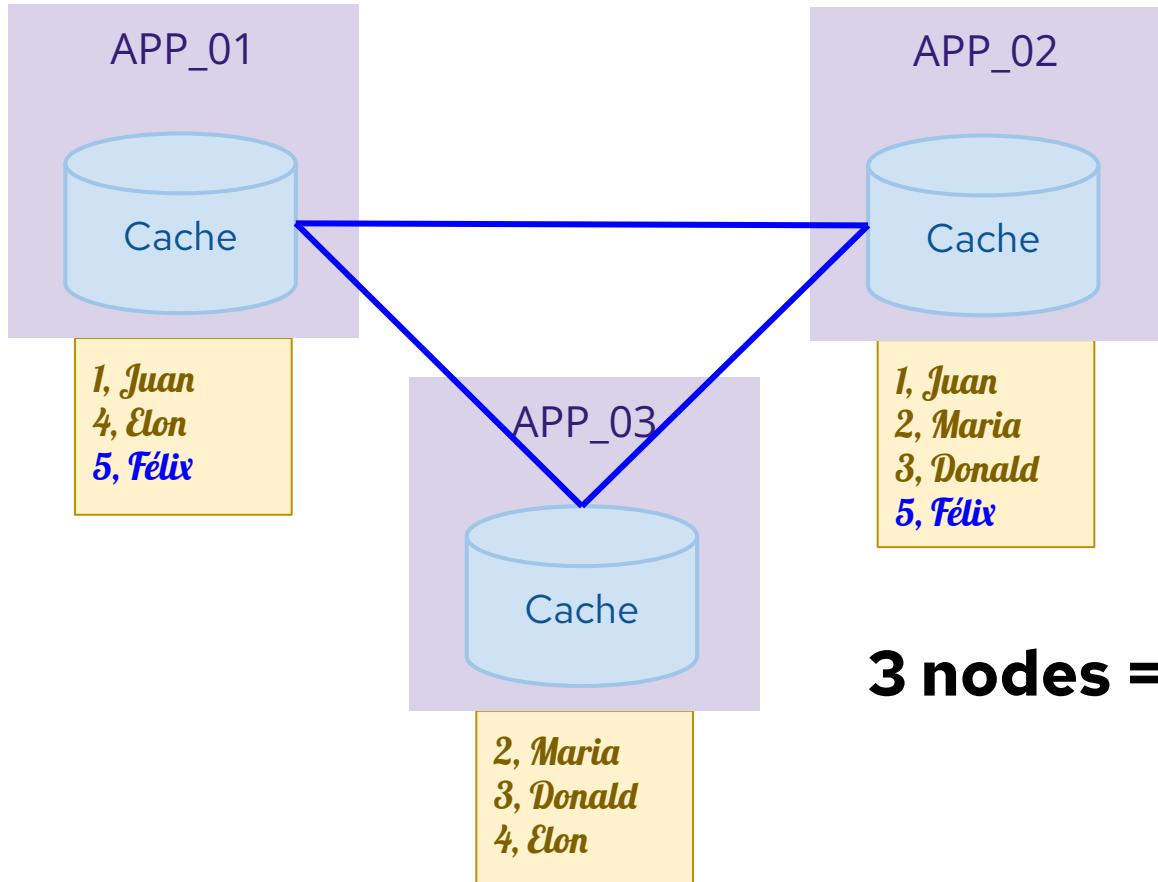
Caso de uso: User sessions and data ?



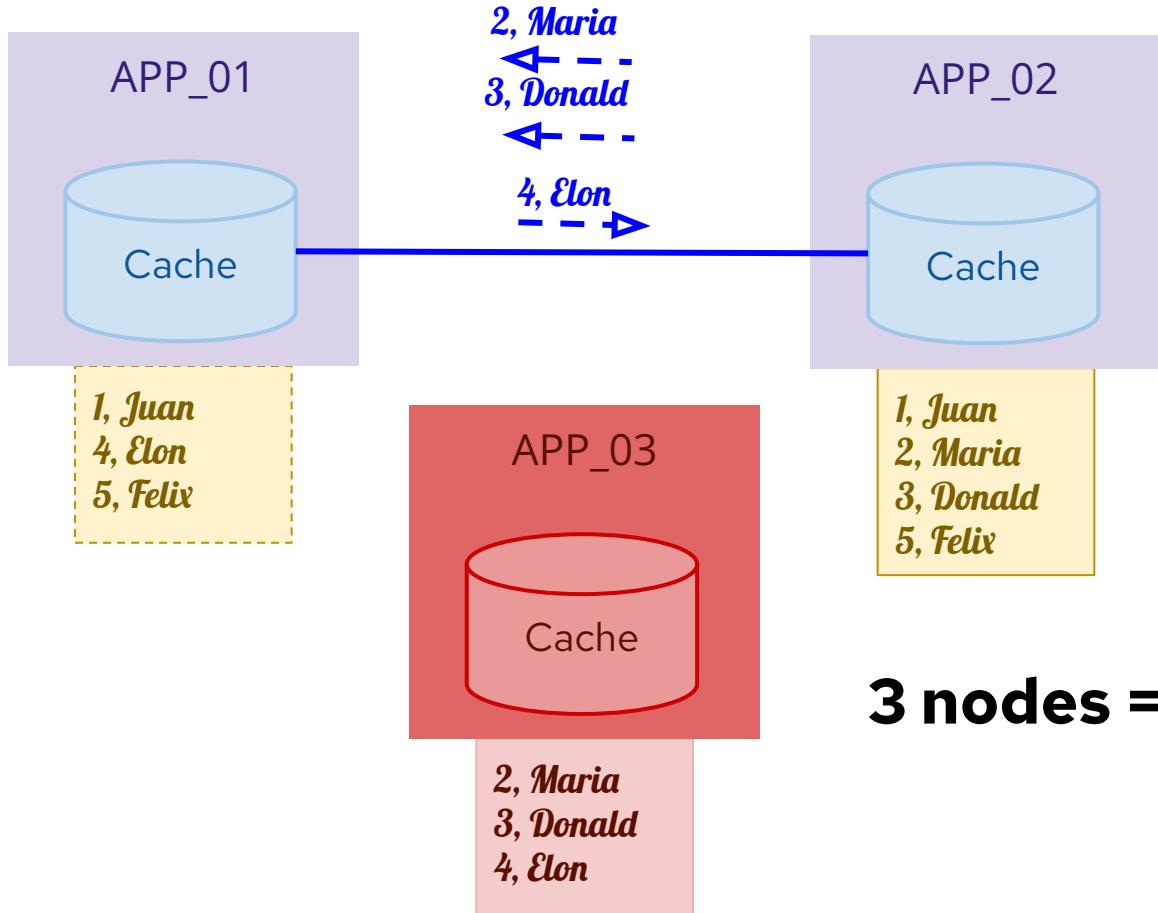


PUT
5, Félix

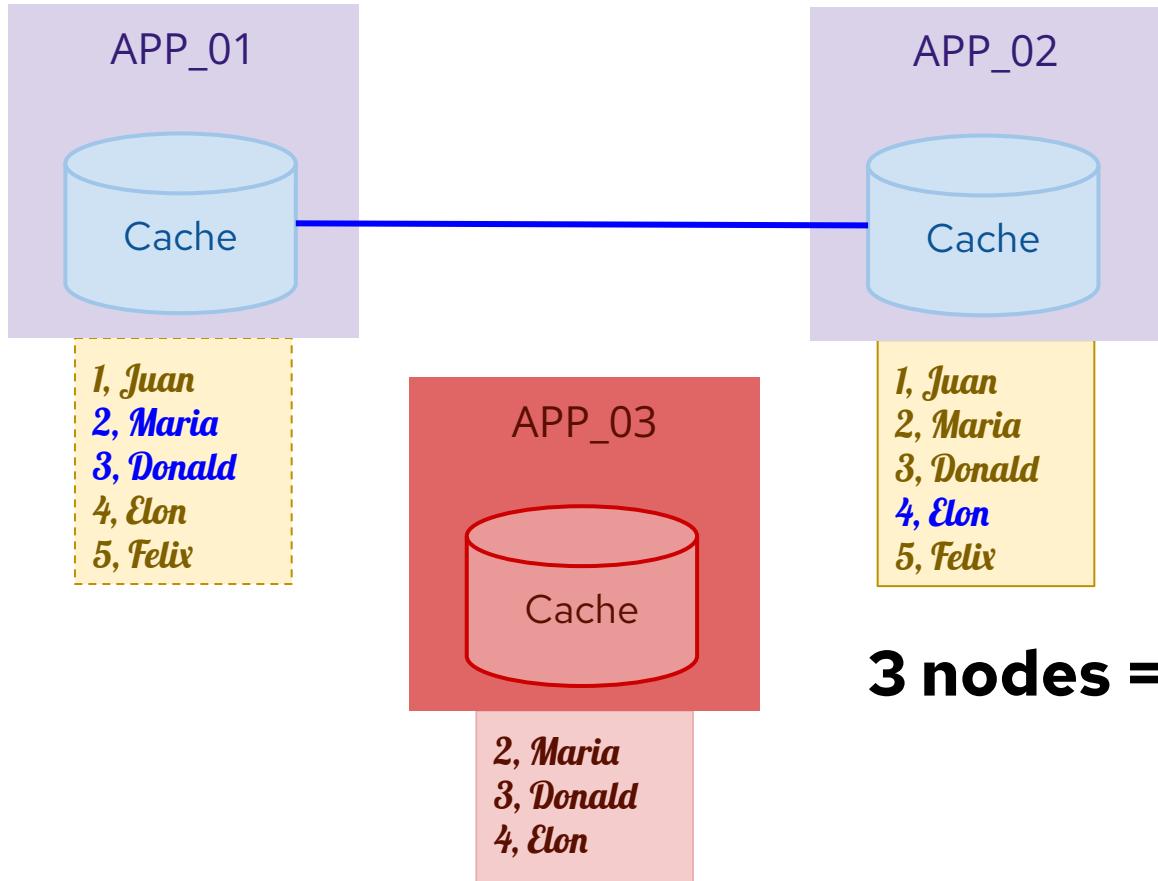




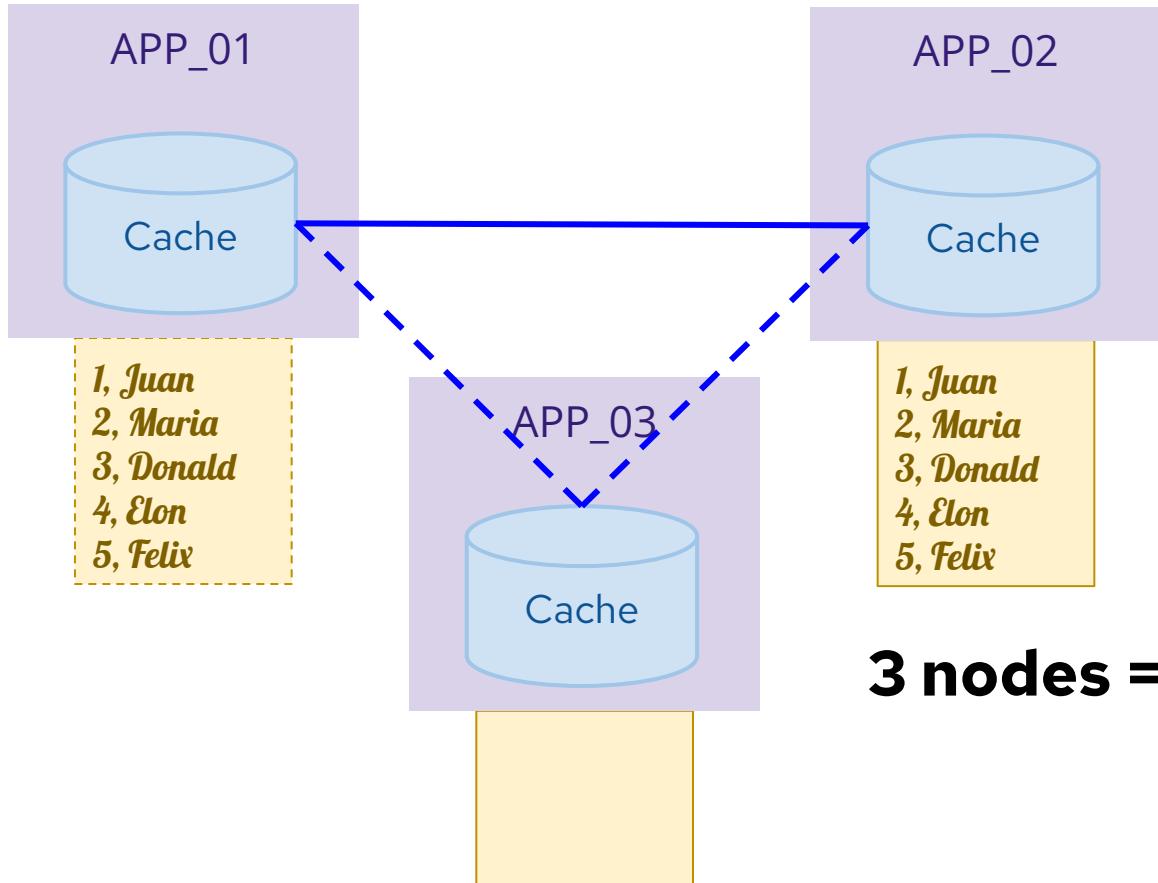
3 nodes = 2 copies

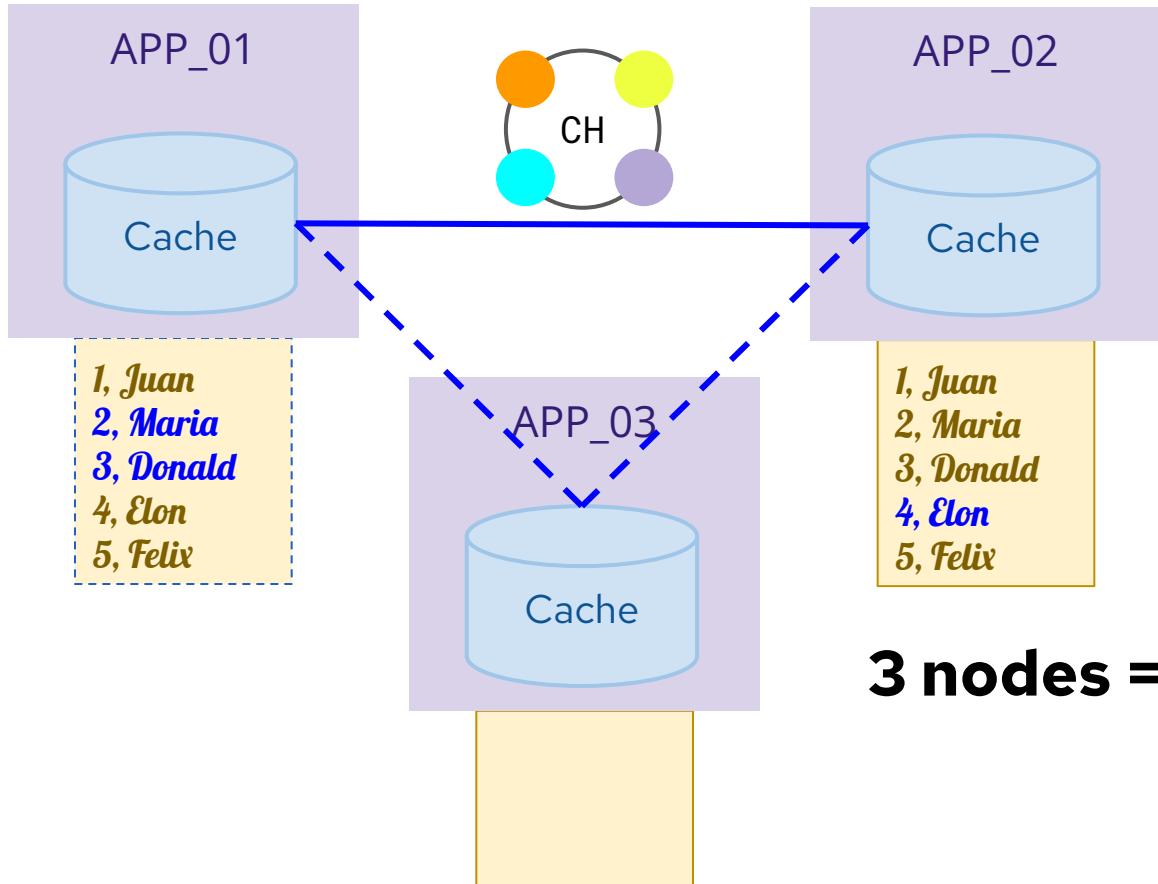


3 nodes = 2 copies

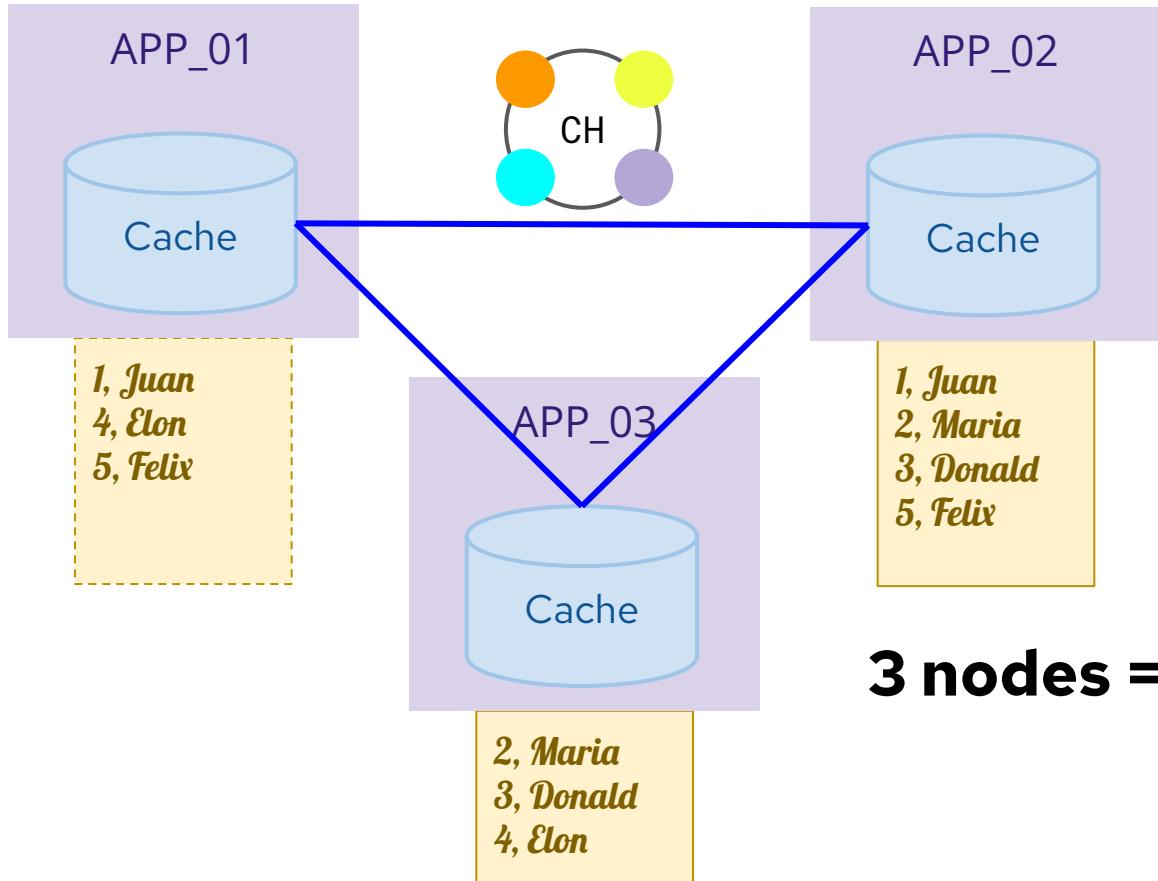


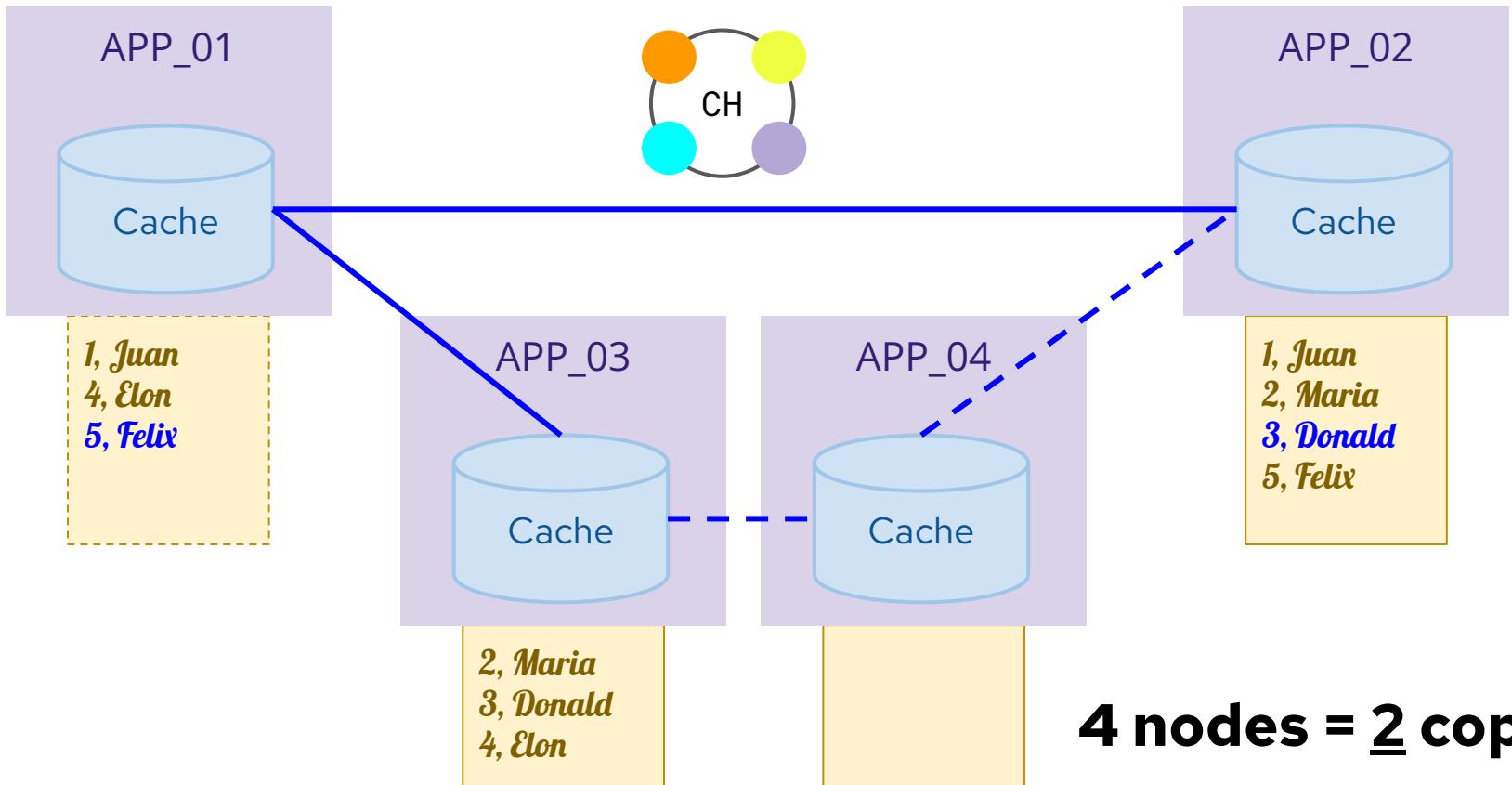
3 nodes = 2 copies



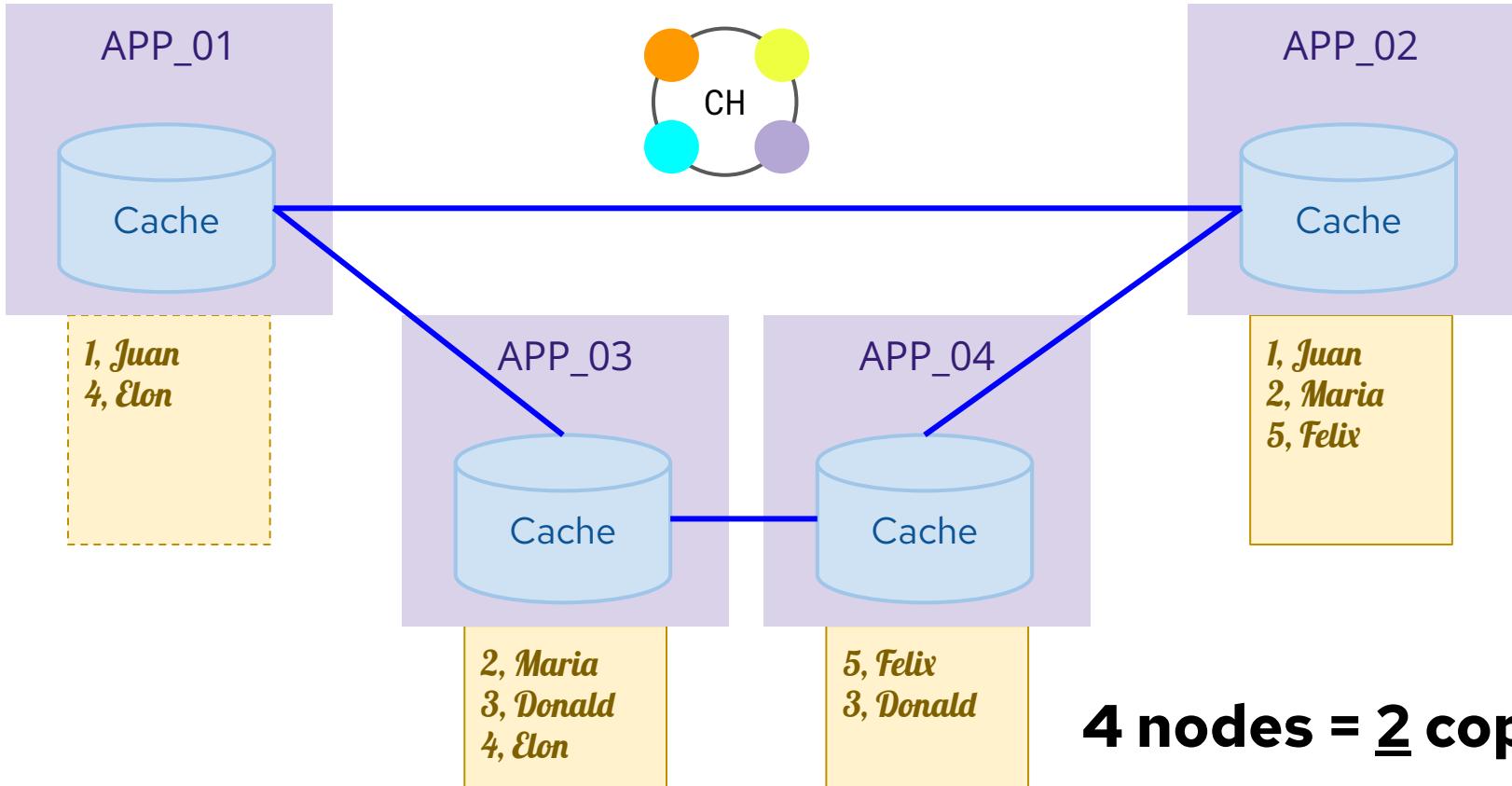


3 nodes = 2 copies

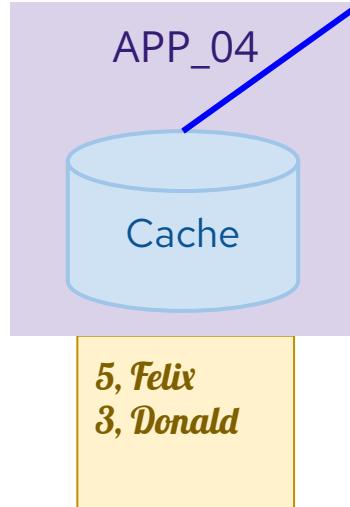
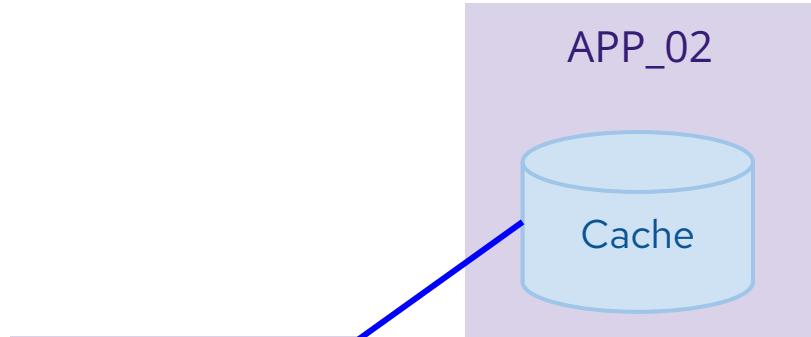
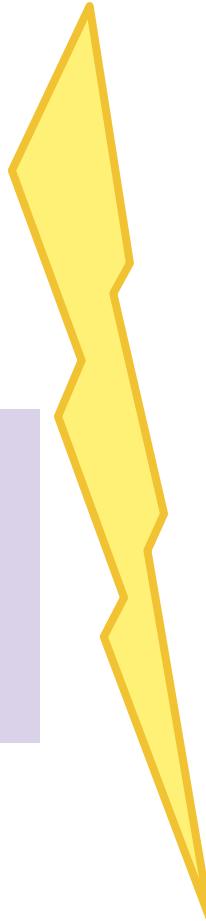
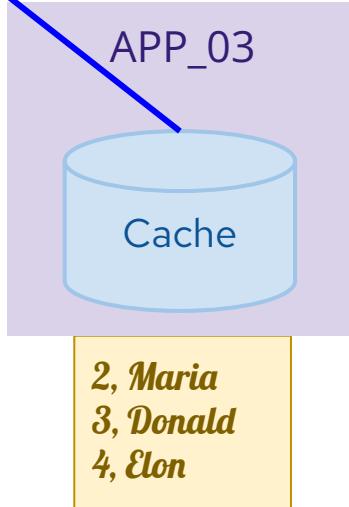
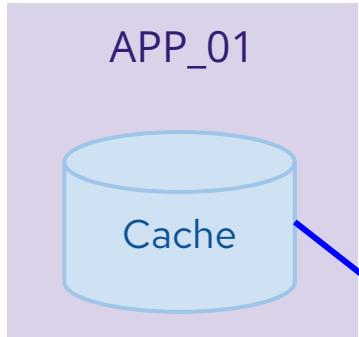




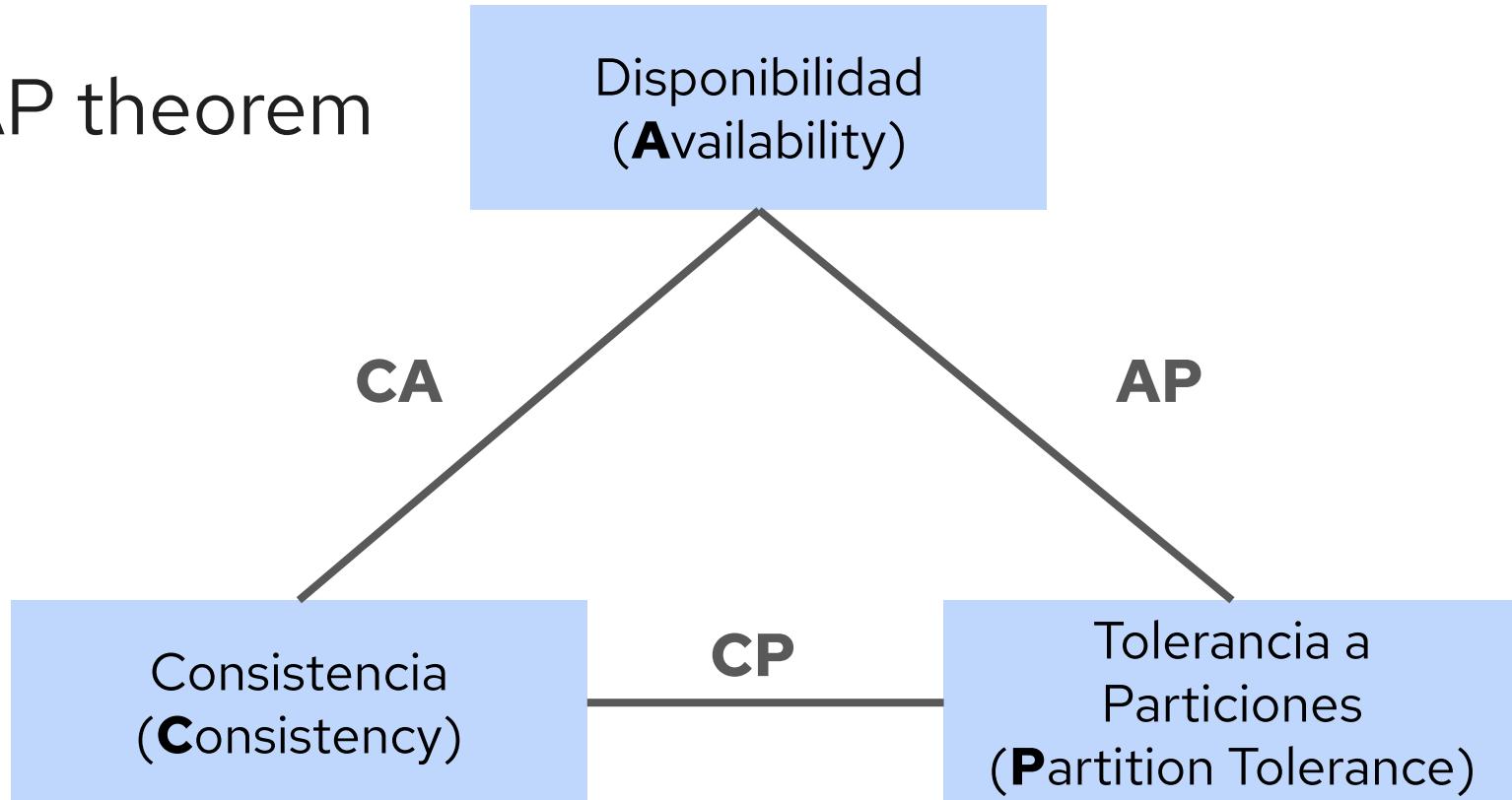
4 nodes = 2 copies



4 nodes = 2 copies



CAP theorem



Language	Libraria / Framework
Java	Hazelcast, Infinispan, Ignite
C#	NCache
Kotlin	Hazelcast, Infinispan, Ignite
Otros	-





Working on the cloud
Deploying with Kubernetes

WHAT DO WE WANT?



STATEFUL APPS!!

HOW DO WE WANT THEM?

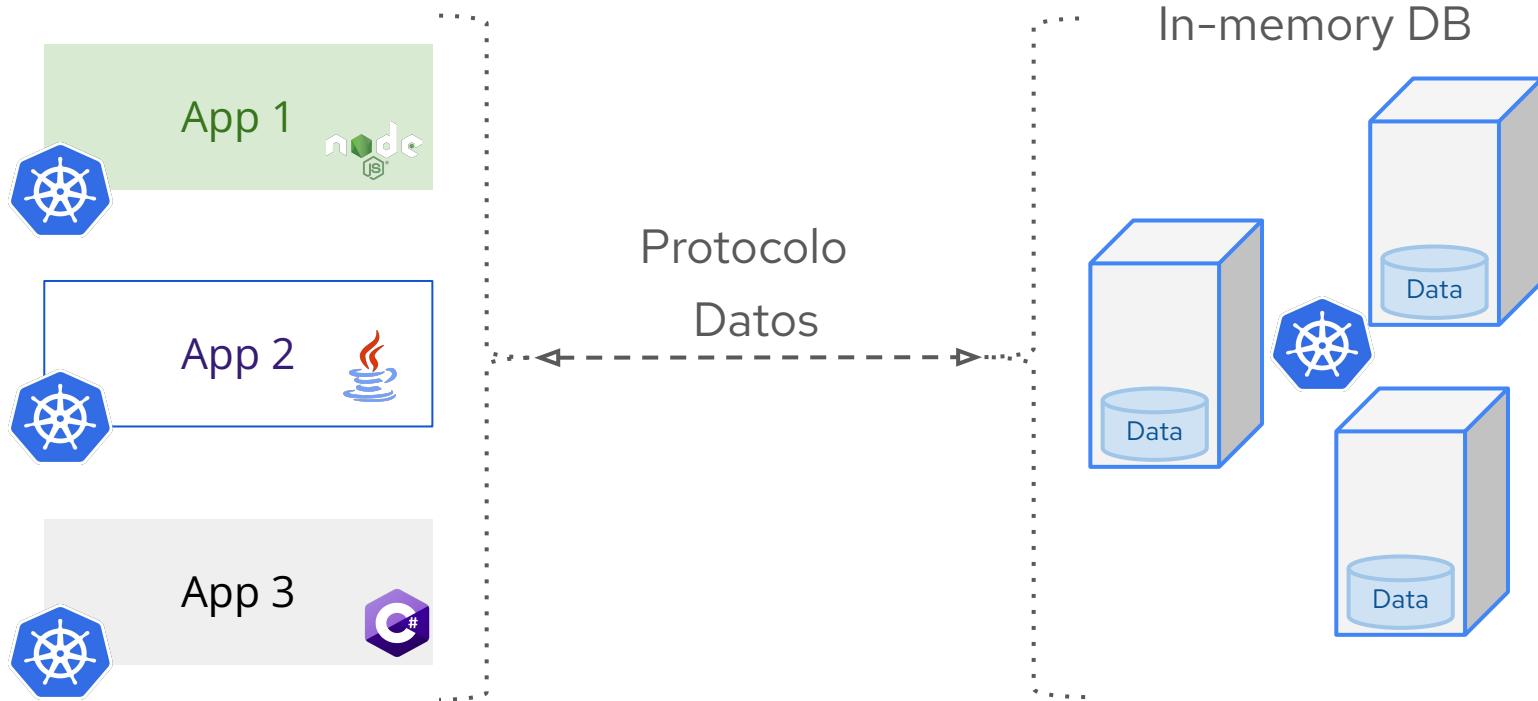


STATELESS!!

In-memory databases

Work as distributed caches too

In-memory databases



Use case: main data store

- High performance and low latency
- Fast transaction processing
- Support for complex operations
- Data durability
- Horizontal scalability
- Advanced security

Cluster execution

Security

Full-text search

In-memory transactions

Generic use cases

- Distributed caches for processing speed
- Session storage
- Real-time recommendations
- Collaborative tools
- Sectors: banking (fraud detection...), insurance, product buying/selling, transportation, telecommunications, etc.



redis



Tarantool

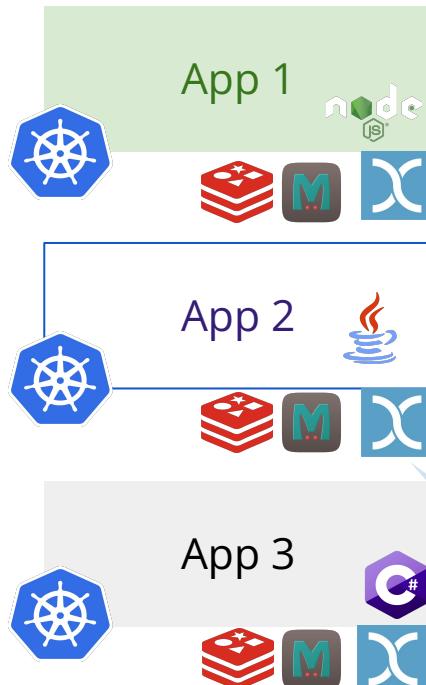


Infinispan



Couchbase

In-memory database, Infinispan example



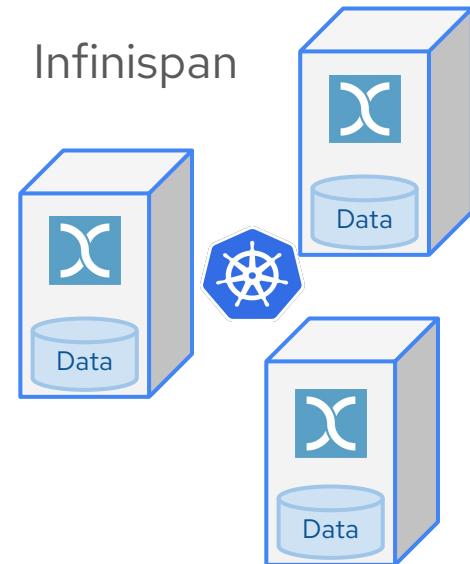
Protocolos:

- HotRod
- RESP (2,3)
- Memcached
- REST

Datos:

- Protobuf
- Texto, JSON, XML
- Java

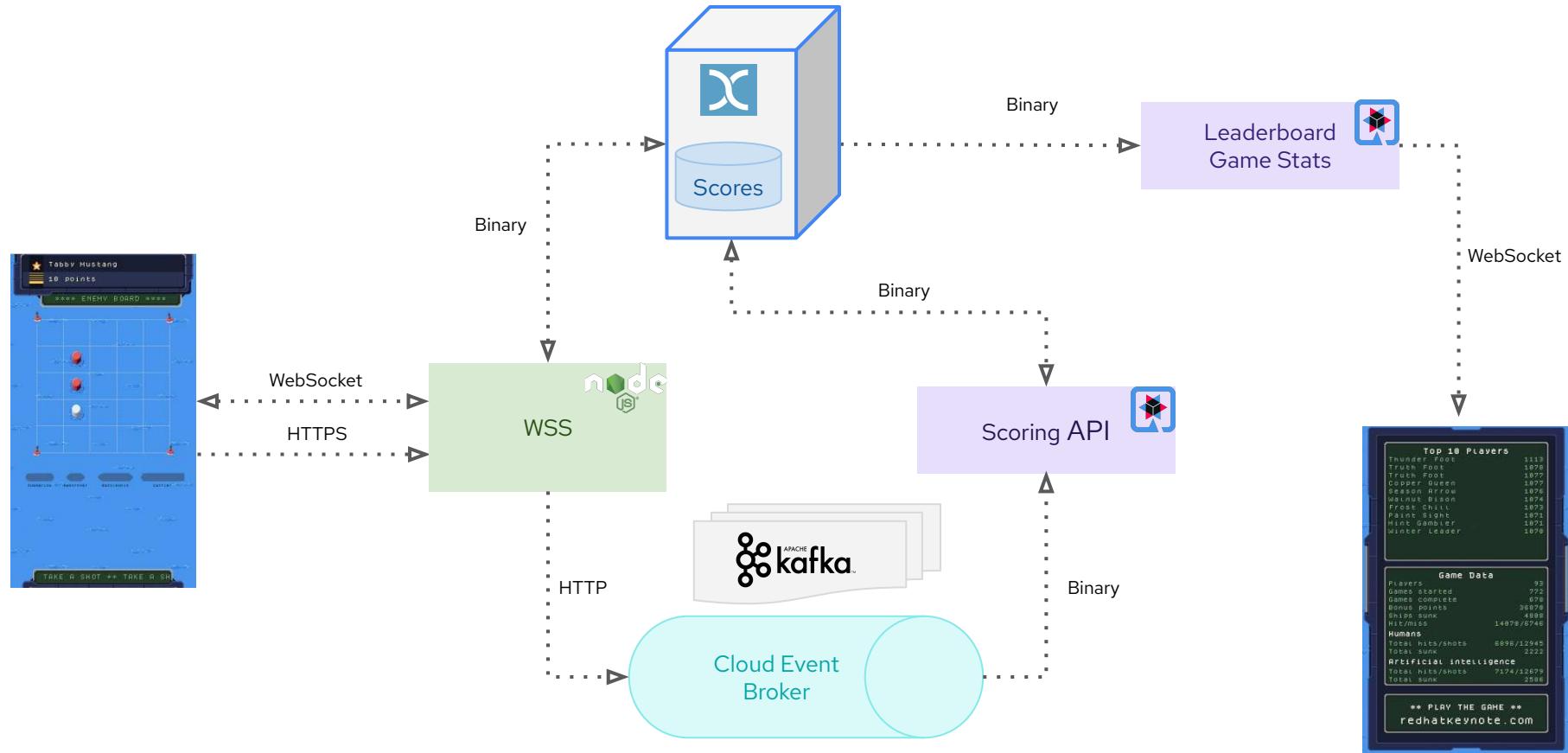
3 nodos = 2 copias



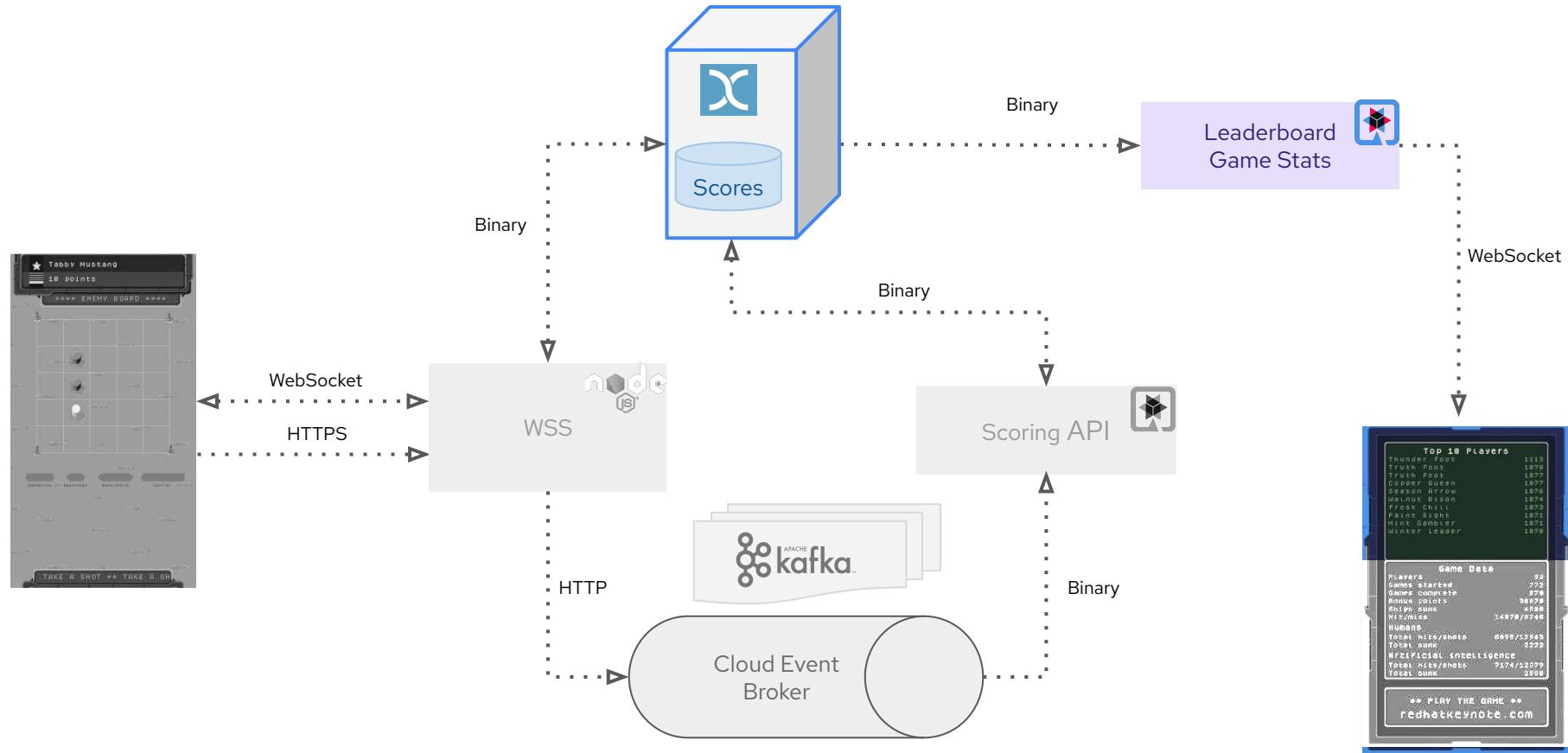
Use case: real-time analytics



Use case: real-time analytics

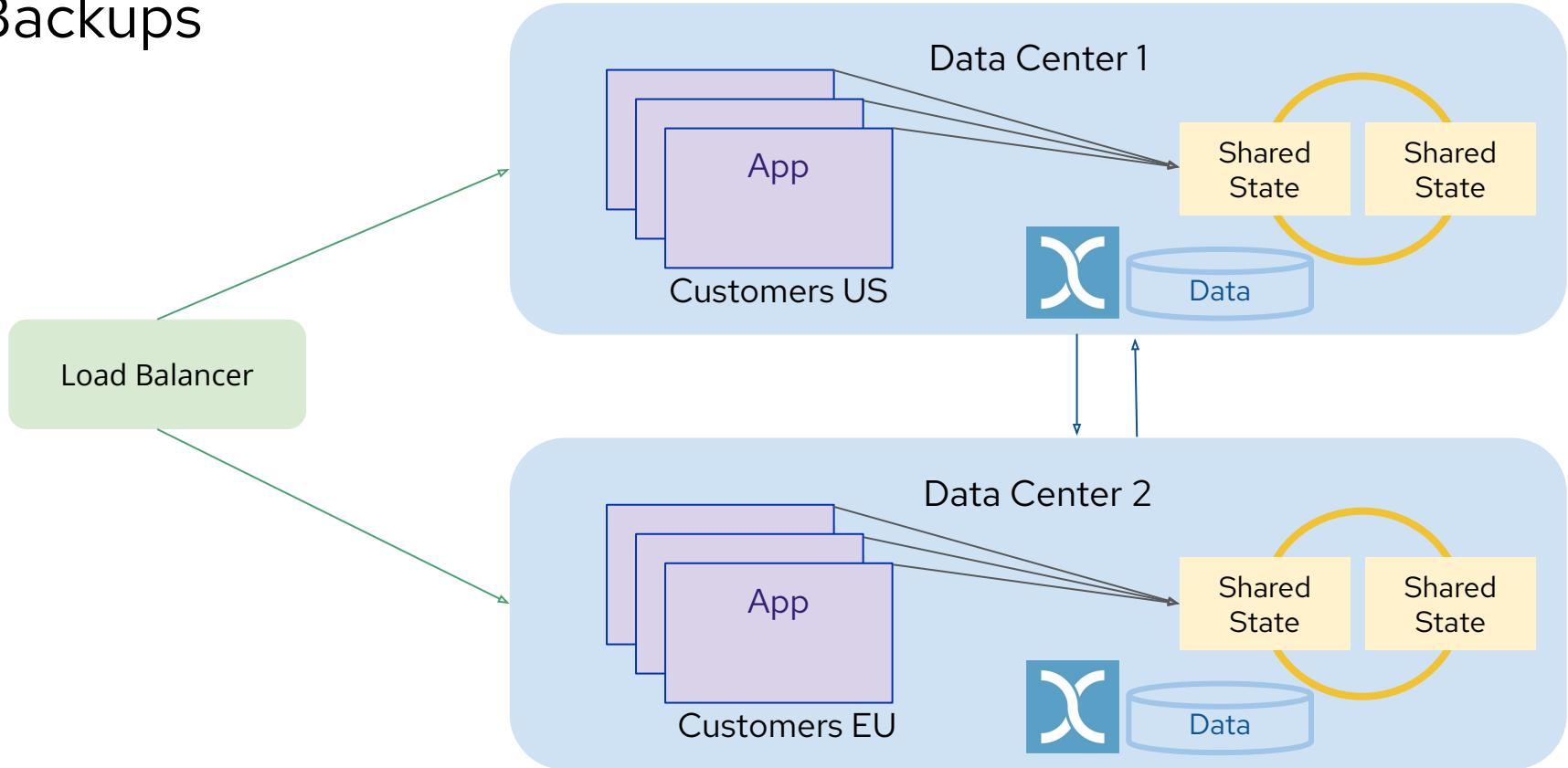


Use case: real-time analytics



Security Backups

Backups



Use case: Keycloak HA (High Availability)

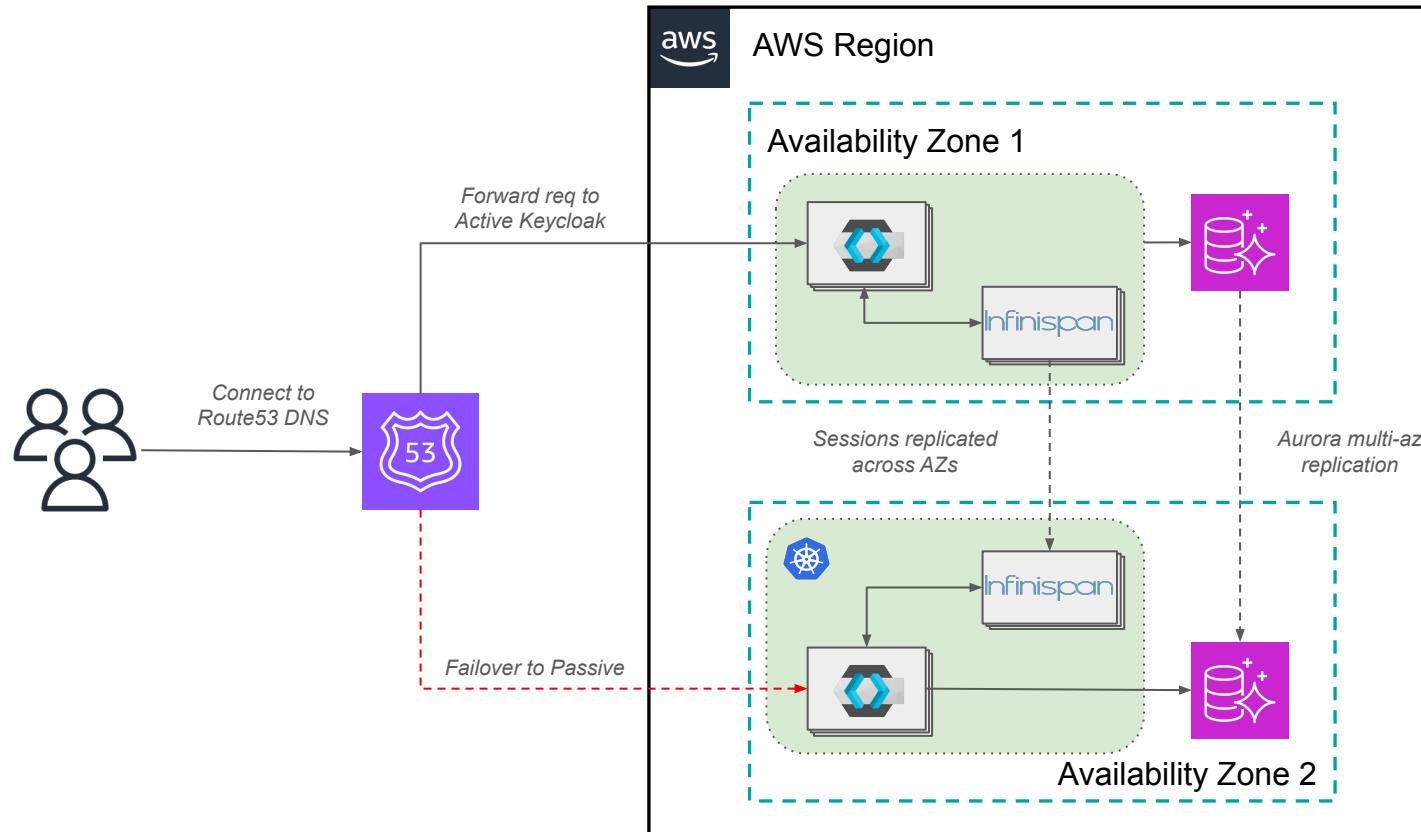
<https://www.youtube.com/watch?v=y1yE7Rd3lGE&t=5s>

The video player displays a presentation slide titled "HA Architecture". On the left, a speaker is visible on stage at the DEVOXX France 2024 event. The slide itself contains a diagram illustrating the Keycloak HA architecture within an AWS Region.

The diagram shows two Availability Zones (AZ1 and AZ2) separated by a dashed line. In each zone, there is a "Keycloak" icon. A "Forward req to Active Keycloak" arrow points from the Keycloak icon in AZ1 to a blue circle representing an active session. A "Sync replication" arrow points from the active session back to the Keycloak icon. A "Session replicated across AZs" arrow connects the active session in AZ1 to the Keycloak icon in AZ2. Another "Sync replication" arrow points from the Keycloak icon in AZ2 back to its own active session (blue circle). A "Failover to Passive" arrow points from the Keycloak icon in AZ2 to a pink circle representing a passive session. A "Sync replication" arrow points from the passive session in AZ2 back to the Keycloak icon in AZ2.

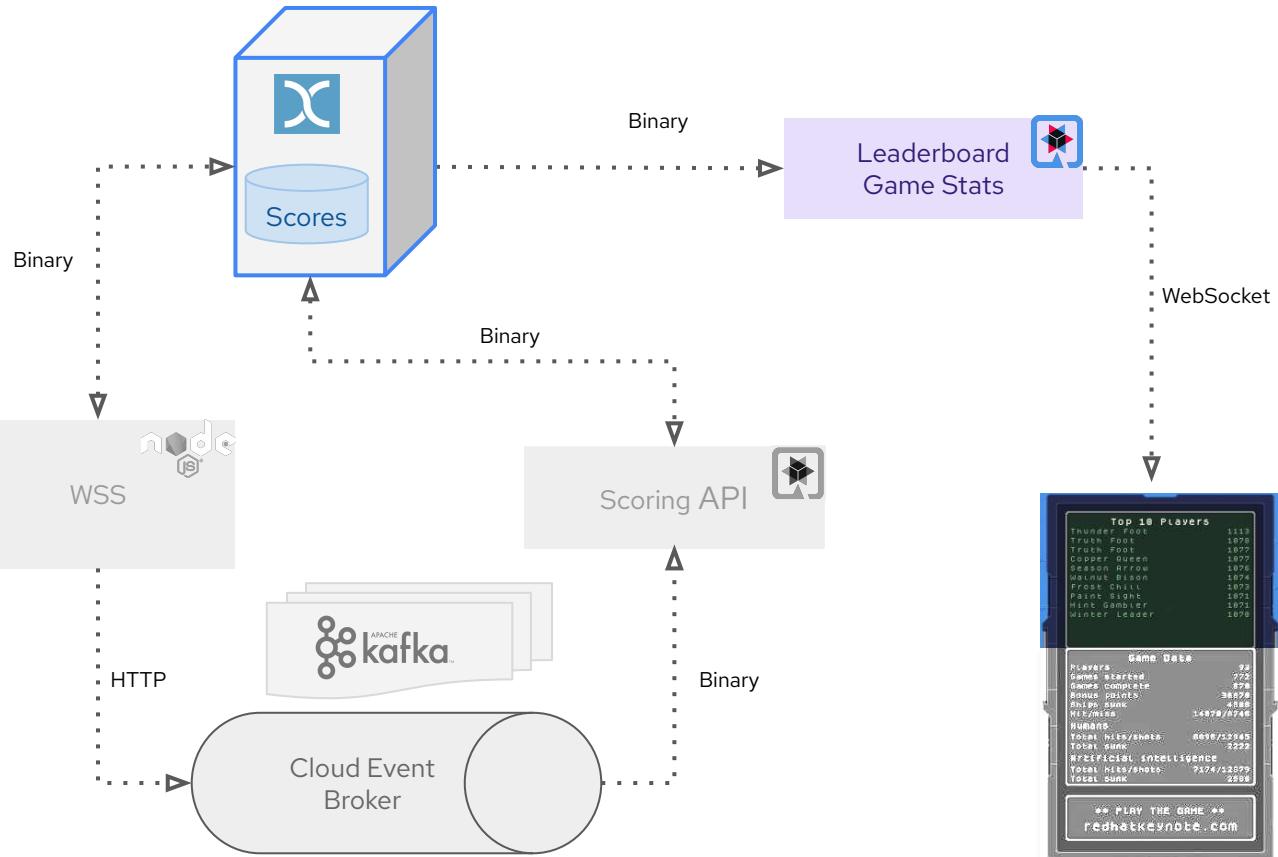
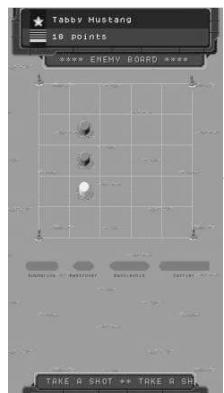
Below the diagram, the "KEYCLOAK" logo is present. At the bottom of the video player, there is a progress bar indicating the video is at 23:42 / 46:10, and standard video control icons.

Use case: Keycloak HA



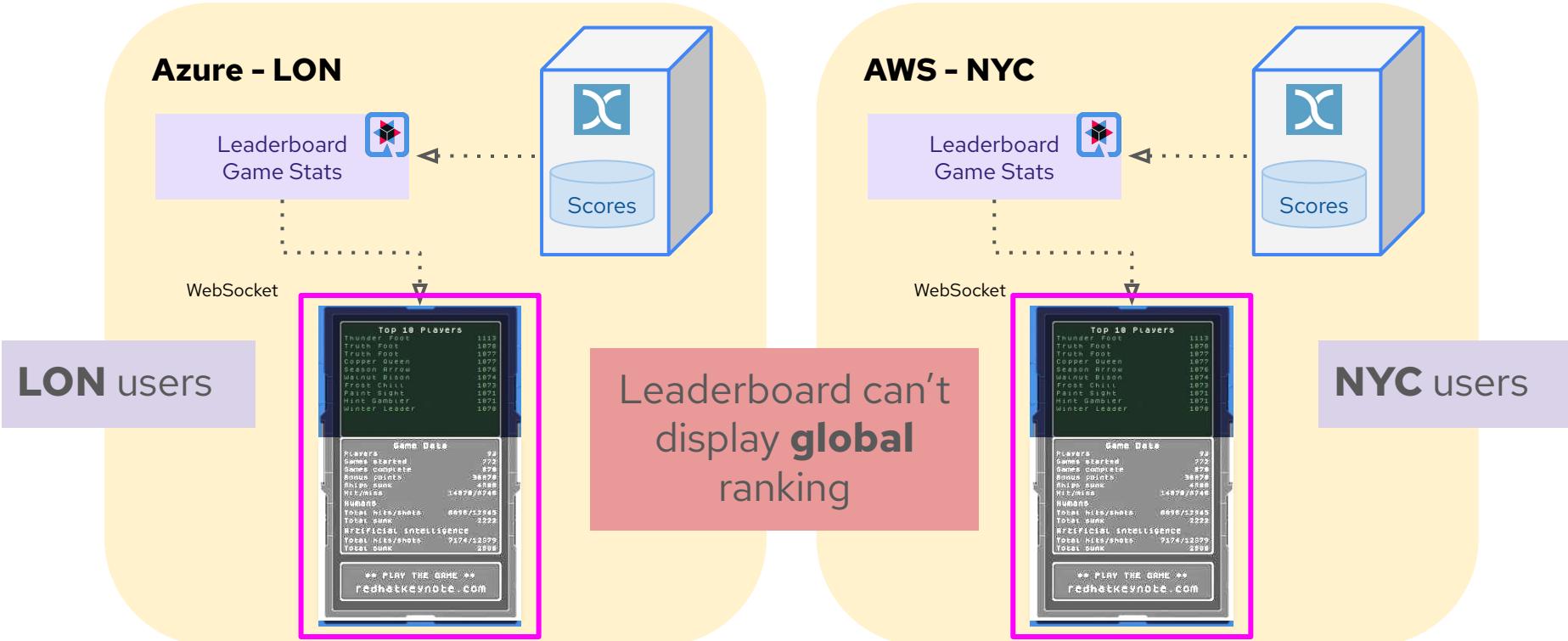
Use case: real-time analytics

Multi-Cloud

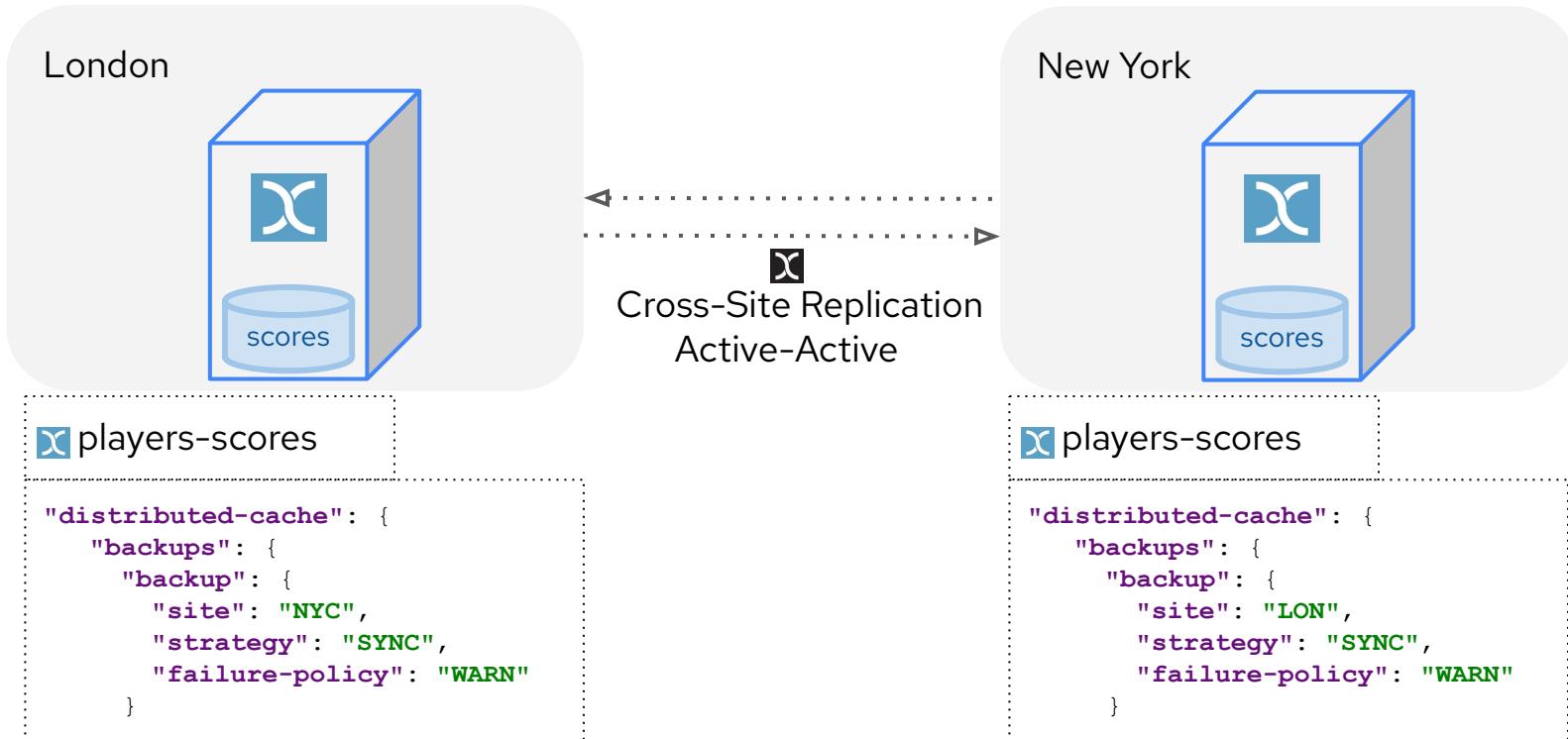


Use case: real-time analytics

Multi-Cloud = 2 clusters

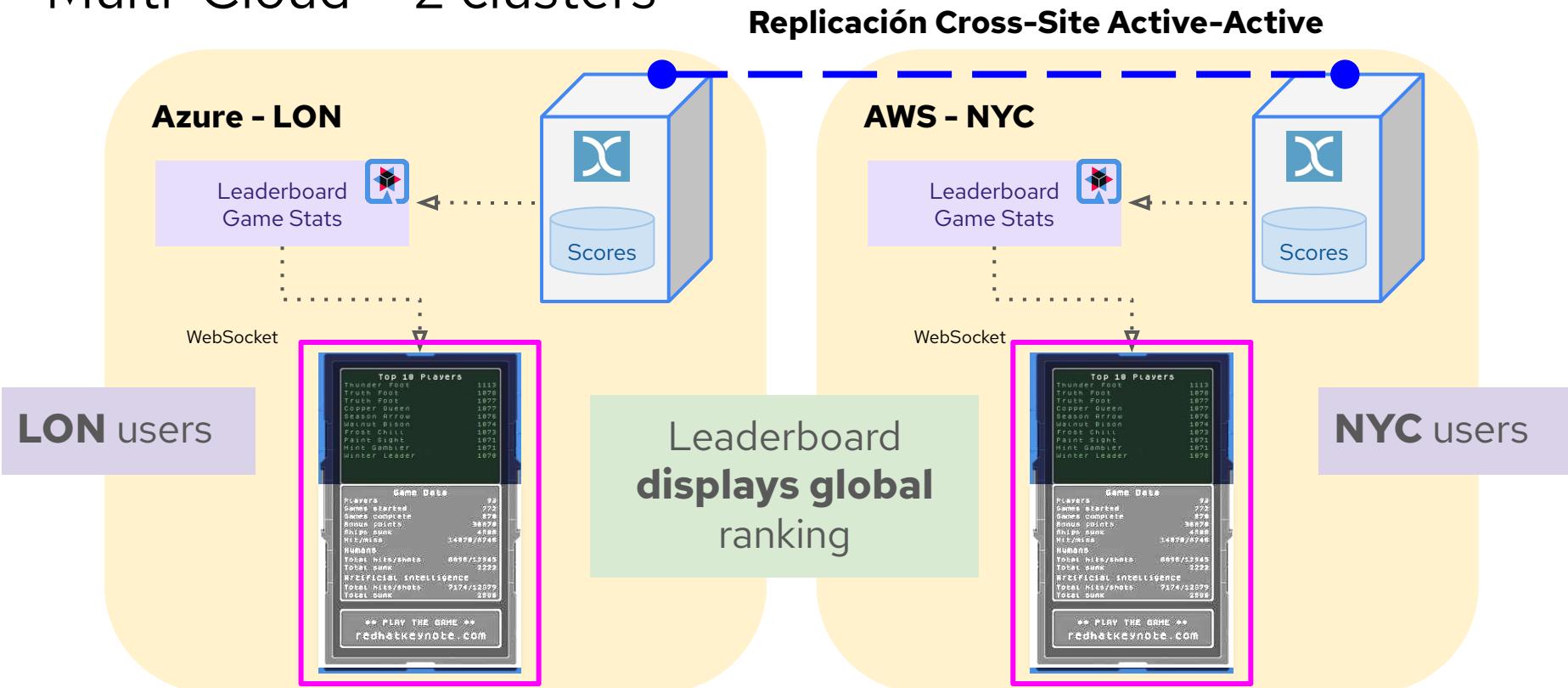


Cluster Backup Configuration

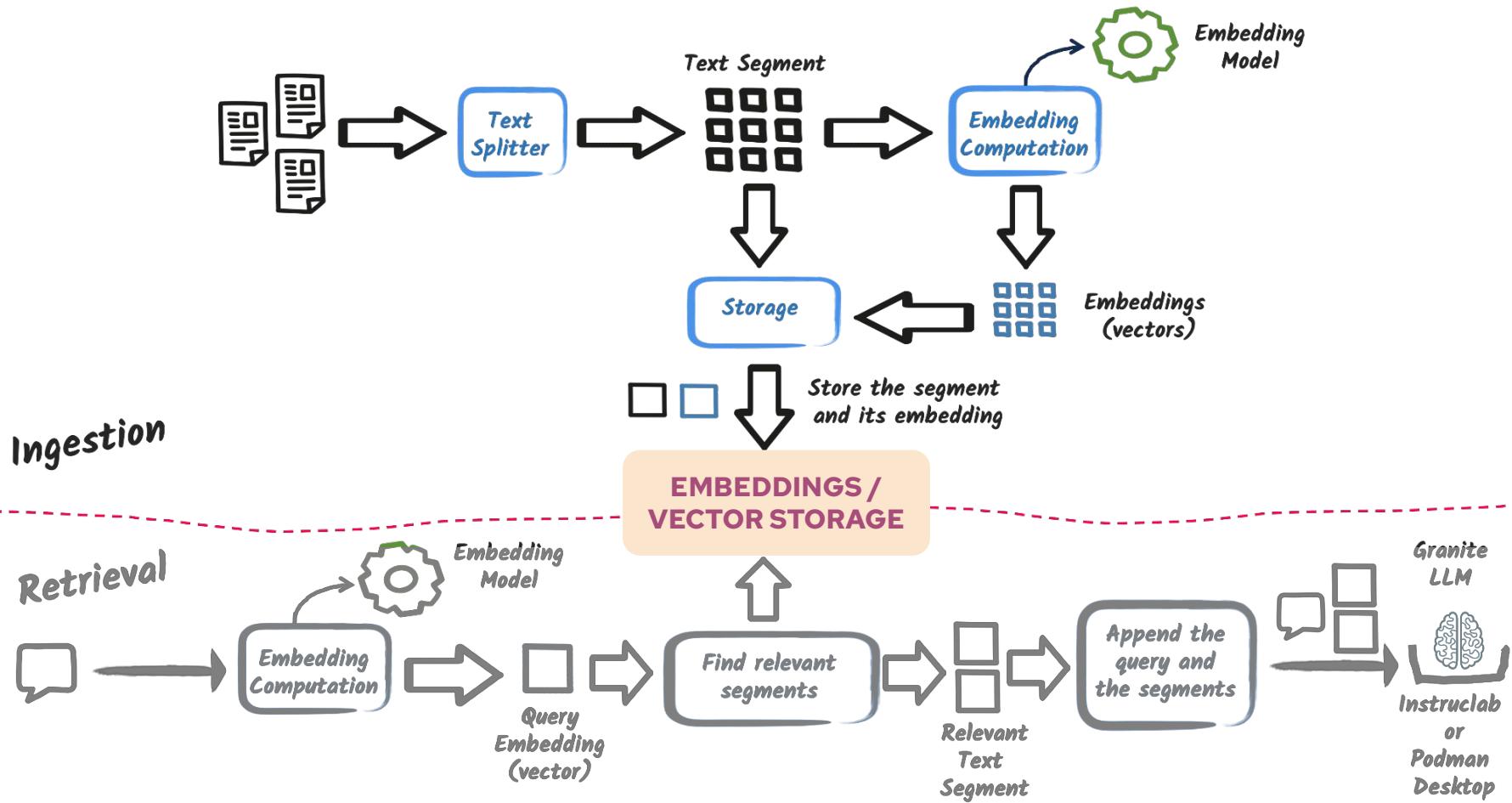


Use case: real-time analytics

Multi-Cloud = 2 clusters



Vector search



LangChain Integrations API Reference More ▾

Providers

- Anthropic
- AWS
- Google
- Hugging Face
- Microsoft
- OpenAI
- More

Components

- Chat models
- Retrievers
- Tools/Toolkits

[Home](#) > Components > Vector stores > Couchbase

Couchbase

Couchbase is an award-winning distributed NoSQL cloud database that provides scalability, and financial value for all of your cloud, mobile, AI, and edge needs. It includes built-in search and indexing, and provides a simple interface for developers and vector search for their applications.

Vector Search is a part of the Full Text Search Service (Search Service).

This tutorial explains how to use Vector Search in Couchbase. You can follow along with the notebook on the managed Couchbase Server.

Components

- Chat models
- Retrievers
- Tools/Toolkits

LangChain Integrations API Reference More ▾

Providers

- Anthropic
- AWS
- Google
- Hugging Face
- Microsoft
- OpenAI
- More

Components

- Chat models
- Retrievers
- Tools/Toolkits

[Home](#) > Components > Vector stores > Infinispan

Infinispan

Infinispan is an open-source key-value data grid, it can work as single node as well as distributed.

Vector search is supported since release 15.x For more: [Infinispan Home](#)

```
# Ensure that all we need is installed
# You may want to skip this
%pip install sentence-transformers
%pip install langchain
%pip install langchain_core
%pip install langchain_community
```

Components

- Chat models
- Retrievers
- Tools/Toolkits

[Home](#) > Components > Vector stores > Redis

Redis Vector Store

This notebook covers how to get started with the Redis vector store.

Redis is a popular open-source, in-memory data structure store that can be used as a database, cache, and queue. It now includes vector similarity search capabilities, making it suitable for applications that require fast, low-latency search and retrieval of large amounts of data.

<https://python.langchain.com/docs/integrations/vectorstores/>

<https://quarkus.io/blog/granite-rag/>

Mona Museum

M

Mona

Welcome, my name is Mona, how can I help you today?

M

Me

What exhibitions do you propose?

M

Mona

Based on the descriptions provided, here are some suggested exhibitions for your youth group:

1. ****Isabella Vanderwood Exhibition****: As an Impressionist inspired by African trips, Isabella's exhibit would be perfect for your group. Encourage them to observe her loose brushwork and vibrant colors, and discuss how they capture the fleeting impressions of light and atmosphere.
2. ****Elijah Montrose Exhibition****: Elijah's Surrealist portraits and whimsical style would spark curiosity and imagination. His artworks, inspired by dreams and the subconscious mind, can lead to fascinating discussions about the power of the unconscious and creativity.
3. ****Adelaide Delacroix Exhibition****: For a unique blend of futuristic symbolism and Renaissance art, Adelaide's exhibit would be an excellent choice. Her inspirations from her pet chinchilla named C-3PO can lead to conversations about the intersection of technology and art, as well as the influence of animals on human creativity.

For larger youth groups, we offer customized tours and overnight experiences. Our overnight tour includes a special scavenger hunt to encourage exploration, refreshments, and a more intimate learning environment. Make sure to have each individual bring their own sleep bag and pillow for a comfortable and engaging experience.

Q&A

Eskerrik asko



<https://infinispan.org/>



<https://quarkus.io/guides/infinispan-client>



<https://quarkus.io/guides/infinispan-client-reference>



<https://quarkus.io/blog/granite-rag/>



<https://github.com/infinispan-demos/riviera-dev-2024-demo>



<https://github.com/infinispan-demos/harry-potter-quarkus>



Real time data on the open hybrid cloud with Quarkus and Infinispan



[Highly available Identity and Access Management with multi-site Keycloak deployments](#)



[Create and manage Red Hat Data Grid services in the hybrid cloud](#)

