

redis

REmote DIctionary Server

Historia de Redis

Origen en 2009

- Creado por **Salvatore Sanfilippo** (también conocido como *antirez*), un desarrollador italiano.
- Nació como solución para mejorar la escalabilidad de su propia startup tecnológica.



Evolución y desarrollo

- 2010:** Introducción de características clave como persistencia en disco y soporte para múltiples estructuras de datos.
- 2013:** Lanzamiento de Redis 2.6, incluyendo scripting con Lua.
- 2015:** Fundación de Redis Labs (ahora Redis Ltd.), impulsando el desarrollo y soporte empresarial..

Código Abierto y Adopción

- En marzo de 2009, Redis se lanzó como proyecto de código abierto bajo la licencia BSD.
- Rápida adopción por parte de la comunidad debido a su alto rendimiento y flexibilidad

Estado actual

- **Comunidad Activa:** Amplio ecosistema de contribuyentes y usuarios.
- **Innovación Continua:** Incorporación de módulos como RedisJSON, RedisGraph y mejoras en rendimiento y seguridad.

Casos de uso Redis:

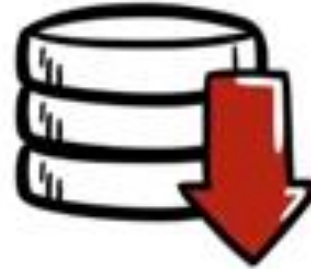


Real time
analytics



High speed
data ingest

Session storage



Application job
management



High speed
transactions



Message queues

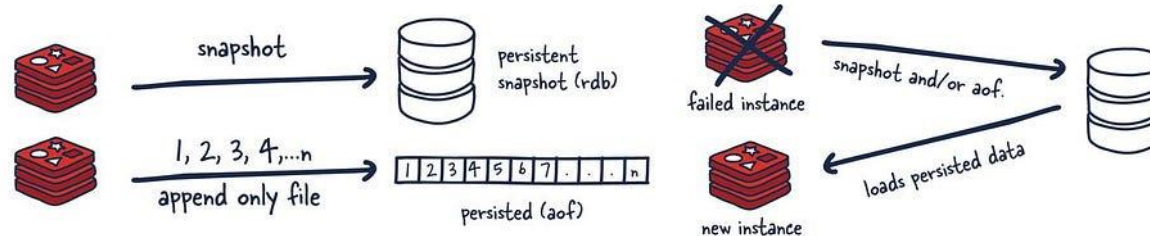


In-app social
functionality

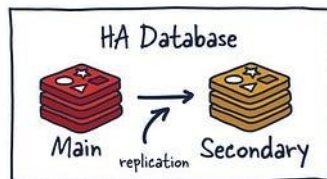
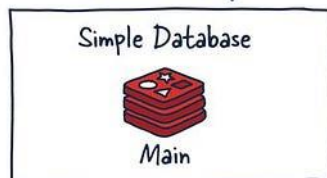
Geo Search
and Caching



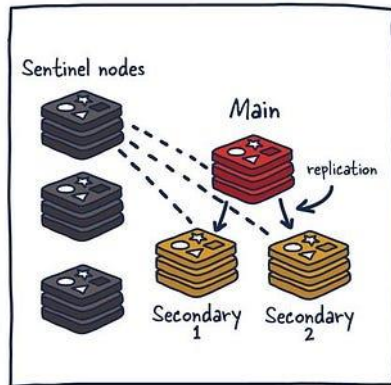
¿Cómo funciona?



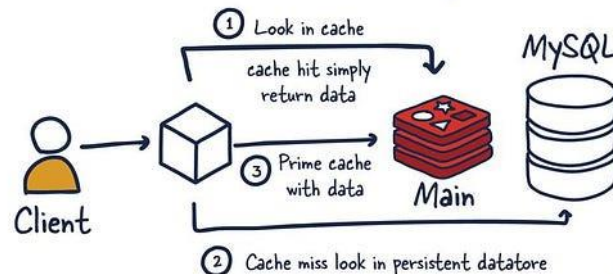
Redis setup



Redis sentinel

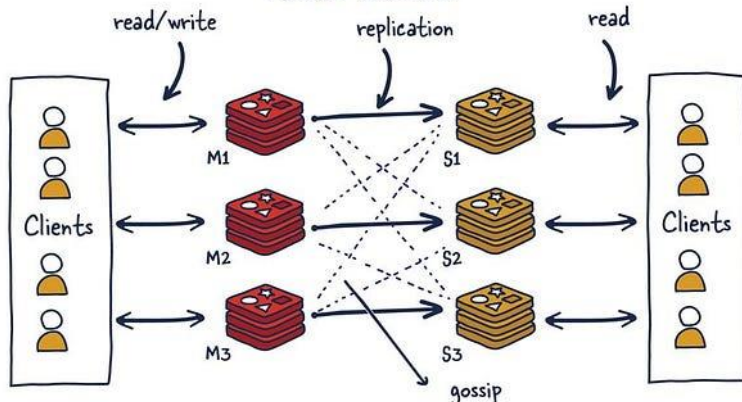


How is redis traditionally used

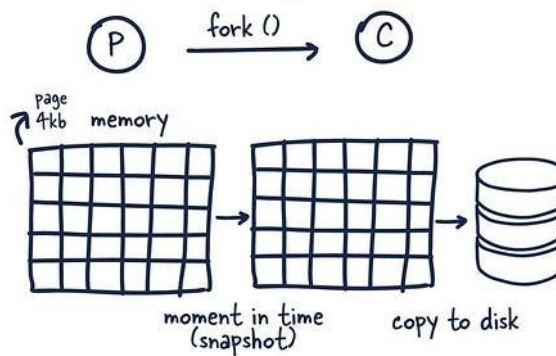


KEY
foo bar
VALUE

Redis Cluster



Forking



architecture notes

hello world	String
011011010110111101101101	Bitmap
{23334}{6634720}{916}	Bitfield
{a: "hello", b: "world"}	Hash
[A>B>C>C]	List
{A<B<C}	Set
{A:1, B:2, C:3}	Sorted set
{A: (50.1, 0.5)}	Geospatial
01101101 01101111 01101101	Hyperlog
{id1=time1.seq((a: "foo", a: "bar"))}	Stream

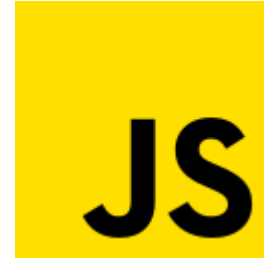
Top 10 comandos principales

- SET: Establece el valor de una clave.
- GET: Obtiene el valor de una clave.
- DEL: Elimina una o más claves.
- INCR: Incrementa el valor almacenado en una clave numérica por 1.
- LPUSH: Inserta uno o más valores al principio de una lista.
- LRANGE: Devuelve un rango específico de elementos en una lista.
- SADD: Añade uno o más miembros a un set.
- SMEMBERS: Devuelve todos los miembros de un set.
- HSET: Establece el valor de un campo en un hash.
- HGET: Obtiene el valor de un campo en un hash.

```
●▲★  
redis> SET user:name "Samuel"  
"OK"  
redis> GET user:name  
"Samuel"  
redis> DEL user:name  
(integer) 1  
redis> INCR page:views  
(integer) 1  
redis> LPUSH tasks "Task 1"  
(integer) 1  
redis> LRANGE tasks 0 -1  
1) "Task 1"
```

```
●▲★  
redis> SADD tags "redis"  
(integer) 1  
redis> SMEMBERS tags  
1) "redis"  
redis> HSET user:1000 name "Samuel"  
(integer) 1  
redis> HGET user:1000 name  
"Samuel"
```


Lenguajes que soporta



¿Quienes usan Redis?



- **X:** Para mantener el timeline de sus usuarios. El timeline es una lista de los tuits de las personas a las que se sigue, y X usa Redis para poder actualizar en el menor tiempo posible los timelines de todos sus usuarios.

- **hulu:** Para mantener la posición en la que un usuario se encuentra en un video, así como el historico de visualizaciones de los usuarios.

hulu



- **Pinterest:** Para mantener la información de los usuarios y los tablonos que sigue cada uno.

- **Caché:** Mejorar la velocidad al almacenar datos frecuentemente accedidos.
- **Colas de trabajo:** Manejar tareas en segundo plano eficientemente.
- **Gestión de sesiones:** Administrar sesiones de usuario de manera rápida y segura



Bibliografía

- Redis. (n.d.). Retrieved from <https://redis.io>
- Yusuf, M. (2022). Redis Explained. Architecture Notes. Retrieved from <https://architecturenotes.co/p/redis>
- Coder, D. (2020, November 17). Introducción a Redis. Medium. Retrieved from <https://medium.com/@diego.coder/introducción-a-redis-bdcdccbf596e>
- Backendless. (2023, August 10). Redis: What it is, what it does, and why you should care. Retrieved from <https://backendless.com/redis-what-it-is-what-it-does-and-why-you-should-care/>