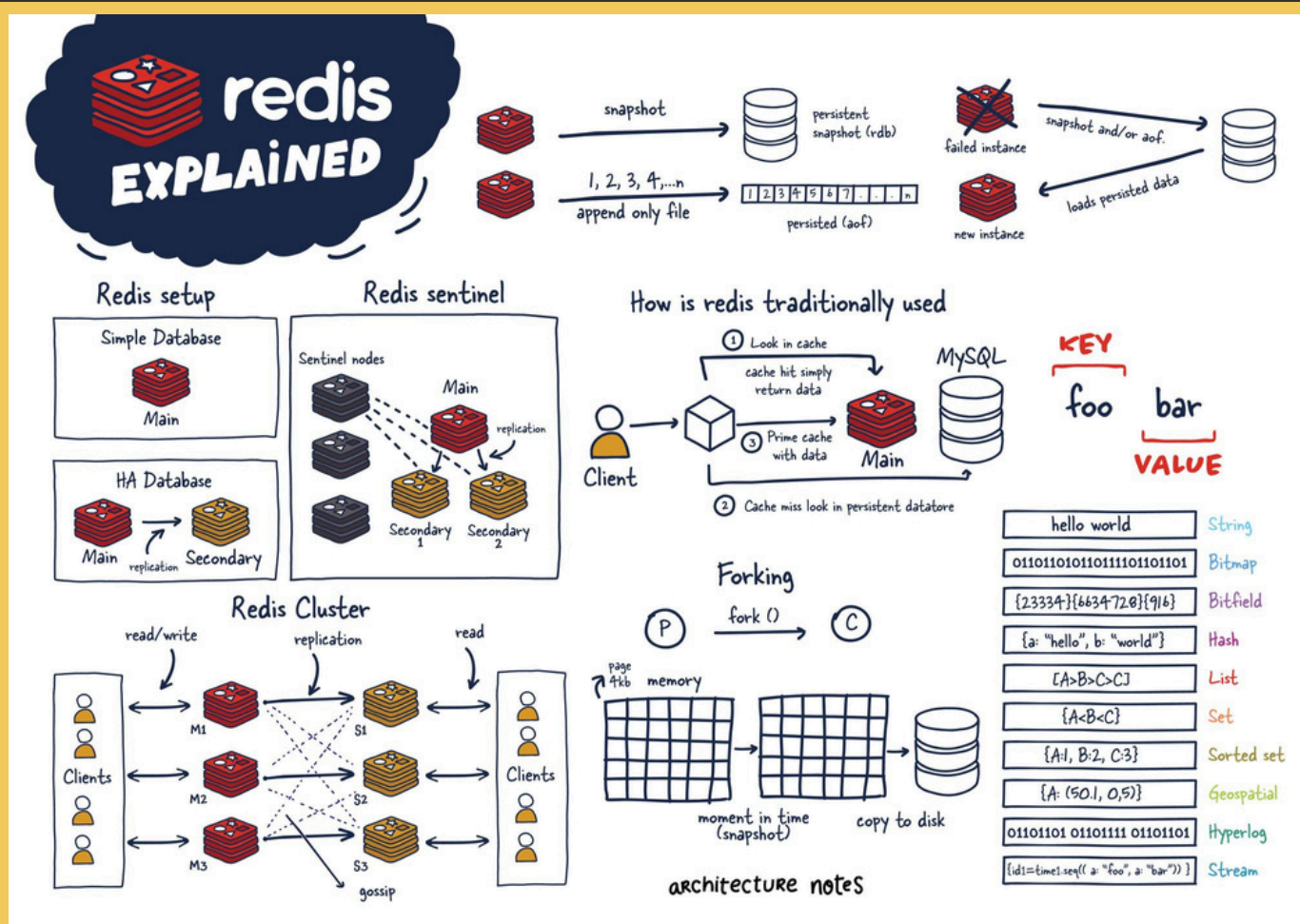




- Type: Key-Value Store (NoSQL)
- Main Features:**
 - In-memory data storage
 - Supports data persistence to disk
 - Advanced data structures (Strings, Lists, Sets, etc.)
 - Pub/Sub messaging
 - Lua scripting

Redis (Remote Dictionary Server) is a blazing fast in-memory data structure store that functions as a database, cache, and message broker. It supports a variety of data structure, such as strings, hashes, lists, sets, sorted sets, bitmaps, and more. Its primary strength is its speed, as it stores all data in memory, making it suitable for low-latency, high-performance applications.



Redis follows a single-threaded event-driven architecture optimized for performance and simplicity. Here's how Redis operates:

- **In-memory Store:** Data is stored in RAM for ultra-fast read/write speeds.
- **Persistence Mechanisms:**
 - **RDB** (Redis Database File): Snapshot of the dataset is saved on disk at intervals.
 - **AOF** (Append Only File): Logs every write operation to disk, ensuring durability.
- **Replication:** Redis supports master-slave replication, allowing read scaling and failover.
- **Sharding:** Horizontal partitioning of data is done to distribute large datasets across multiple nodes.

Why Redis Over Others?

Redis offers distinct advantages over other NoSQL databases:

- **Ultra-fast Performance:**
 - Redis is in-memory, so data access and manipulation are almost instantaneous.
 - Low latency (less than a millisecond).
- **Flexible Data Structures:**
 - Beyond key-value pairs, Redis supports lists, sets, hashes, sorted sets, bitmaps, and more. These advanced structures make Redis versatile for a wide range of use cases.

Persistence Options:

- Redis combines the speed of an in-memory store with the durability of disk persistence through RDB and AOF, providing flexibility based on requirements.

Easy Scaling:

- Redis supports horizontal scaling via replication and partitioning (sharding), making it suitable for large-scale applications.

Use as Cache and Message Broker:

- Redis doubles as a cache due to its fast memory access and can also be used for Pub/Sub messaging systems for real-time updates.

Real-life Applications of Redis

Caching:

- Reduces server load and speeds up responses for platforms like Twitter, GitHub, and StackOverflow by storing frequently accessed data in memory.

Session Management:

- Stores user session data for quick access and low latency, used by e-commerce sites and social media platforms.

Real-time Analytics:

- Processes and analyzes data streams in real-time, commonly used in IoT systems and ad-tech for live insights.

Message Queue (Pub/Sub):

- Enables real-time messaging and notifications in chat applications, data streaming, and live event updates.

Leaderboard Systems:

- Provides fast updates for maintaining leaderboards in real-time, used in gaming platforms to track player rankings.

Name: Deep Salunkhe
Roll.No:21102A0014