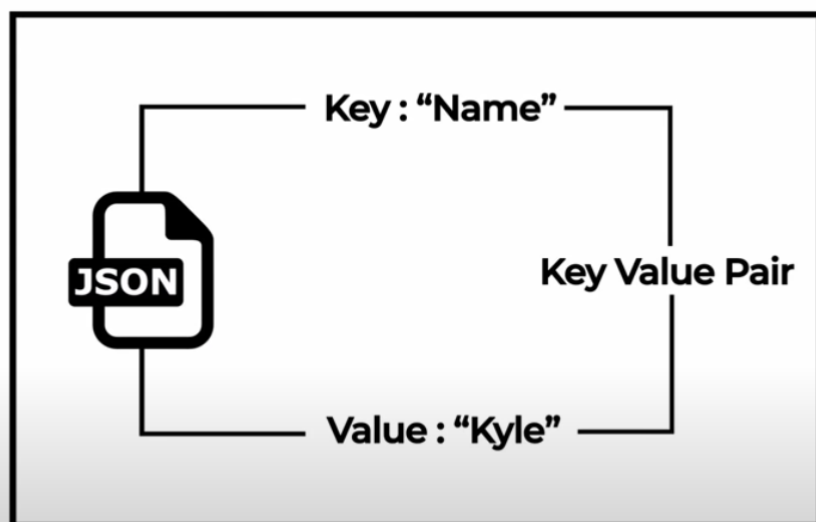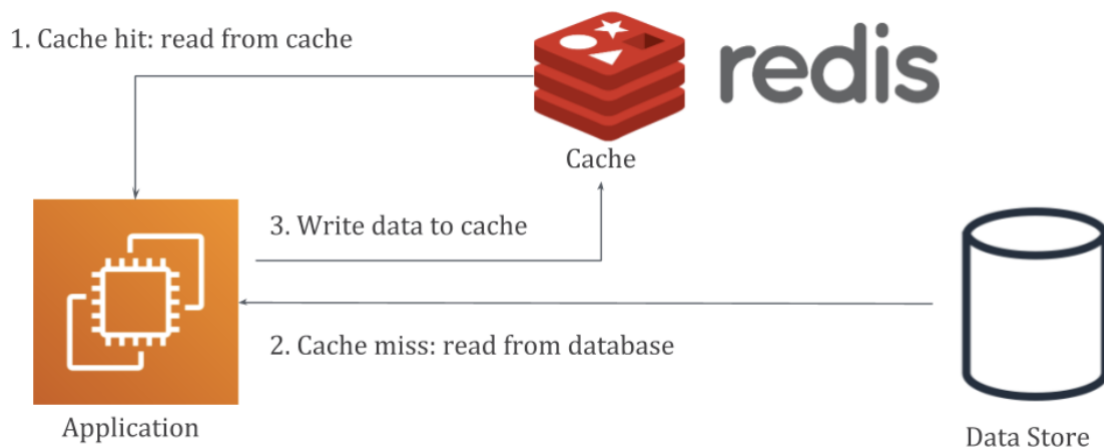# Redis

Singh Rounak.

1. It is kind of a Database (A giant KeyValue pair object in JSON format)
2. Very different from MySQL, NoSQL or MongoDB type of databases.

3. In-memory key-value Store (Runs on your computers RAM instead of some server) this makes it incredibly fast but more volatile.
4. Open Source -written in C
5. It Sits on top of a Database (Front-Redis ——>Back -MongoDB)

Web applications run on servers and when requested with some information from clients, the fetch data from databases in the backend. There might be a delay in rendering output since the databases have to be accessed first hand by the application using API calls.

With Redis, the Web Apps don't have to to access databases every time. Redis stores cache of the data and when thousands of servers are running thousands of applications at the same time with same data that doesn't change that much, Redis stores that data and whenever a call is made to fetch particular piece of data the chache render the data, thus saving a trip to the database.

Persistance Mechanism -

1.RDB Redis Snapshotting

2.AOF Append-Only File

Redis is not cloud based.

No XML Support, No SQL Support, No secondary index support

C , C++, Python, PHP are supported langs

Companies using Redis - Twitter, Github, Pinterest, Snapchat, Craiglist, Digg, StackOverflow and Flickr

NO SQL- Term used to determine High performance Non -Relational DB.

Data - Doc, Graphs, KV , etc

Benefits - Ease of development, scalable performance, High availability and resilience.

DBs -Documents, KV Pairs, GRAPHs

1. Documents - Pair each key with a complex data structure called a document(nested as well).
2. Graph Stores - Used to store info about networks of data like Social connections, Neo4J
3. Key-Value stores - Keys with Values, simplest form of data. Most Flexible
4. Columns - Cassandra
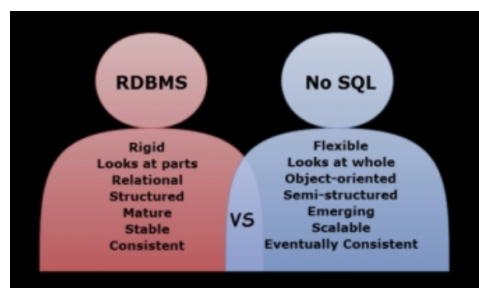
Application Utilization - More NoSQL DBs are used as it performs better than RDBMS.

KV Stores -

Scalable, Reliable, Portable, Low Operational Costs

Move application from one system to another without rewriting the code or constructing new architecture.

Massive Scalability

High Availability

DataTypes -

String	Key value
List	List of strings, sorted by insertion order
Set	Unordered collection of strings
Sorted Set	Similar to sets but each entry has an associated "score"
Hash	Map between string fields and string values

String

List

Set

Sorted Set

Hash

Redis Datatype

	GET / SET	INCREMENT / DECREMENT	PUSH/POP	Other
String	✓	✓		
List	✓(By index)	✓	✓	
Set	✓(By item)	✓	POP	Intersect / Union / Diff
Sorted Set	✓(By item, score or rank)	✓		Intersect / Union
Hash	✓(By field)	✓		

Other Datatypes
- **Bitmap**
- **HyperLogLogs**