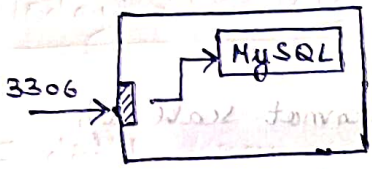
Database Sharding and Partitioning

Sharding :- Method of distributing data across multiple machines

Partitioning :- Splitting a subset of data within the same instance

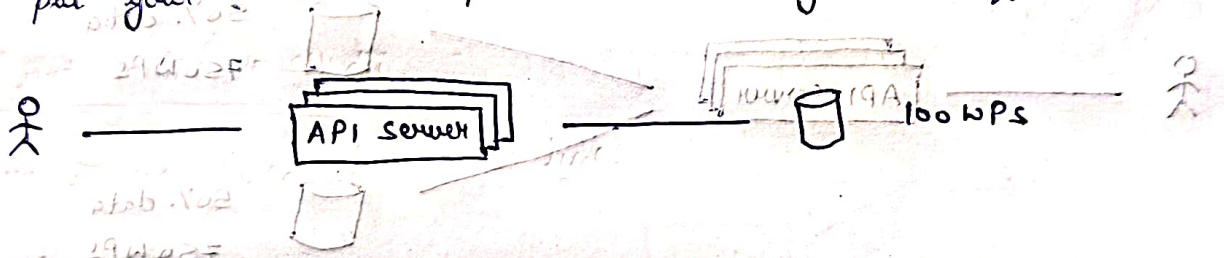
How a database is scaled ?



A database server is just a database process (mysqld, mongod) running on EC2 Machine

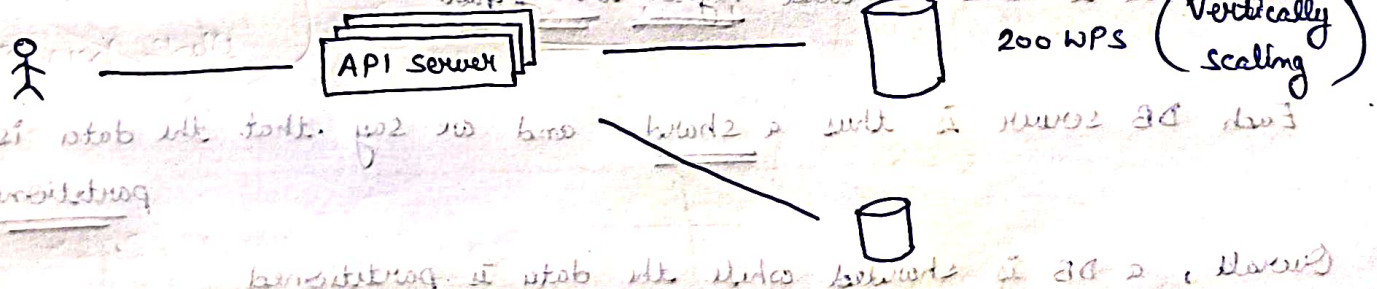
And we represent this as 

You put your database in production, serving real traffic



You are getting more users, that your DB is unable to manage

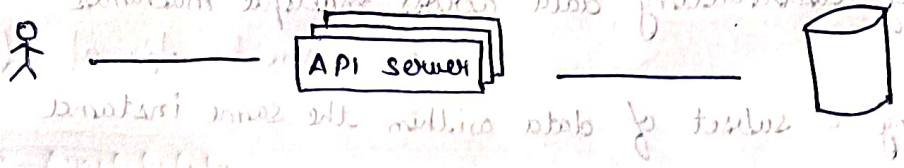
you scale up your DB : give it more CPU, RAM and DISK



↳ **Bulkier Server + Read Replica**

for handling writes & complex reads ; for handling only read requests

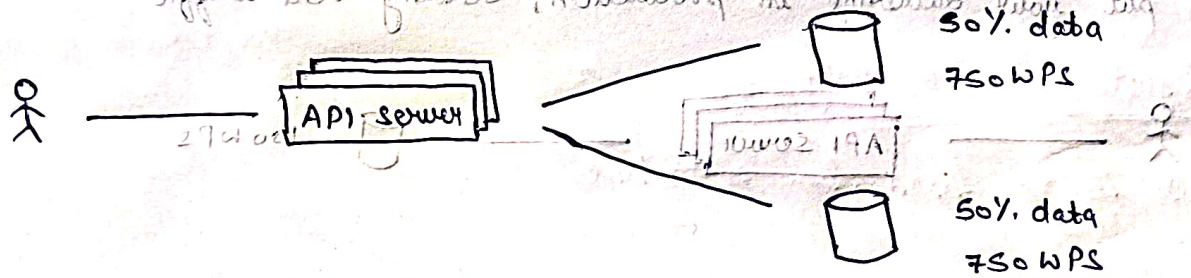
Your product went viral and your bulky DB is unable to handle the load, so you scale up again



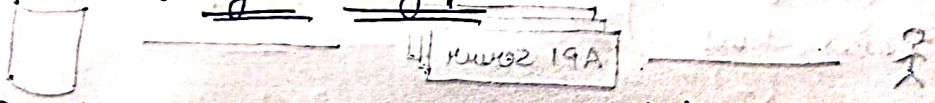
But after a certain stage you know you would not be able to scale up your DB because Vertical scaling has limit.

So, you will have to resort to Horizontal scaling

Say, one DB server was handling 1000 WPS and we cannot scale up beyond that but we are getting 1500 WPS, we scale horizontally & split the data

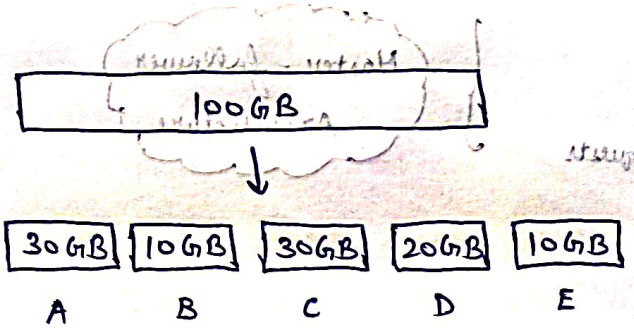


By adding one more DB server, we reduced the load to 750 WPS on each node & thus handled higher throughput



Each DB server is thus a shard and we say that the data is partitioned

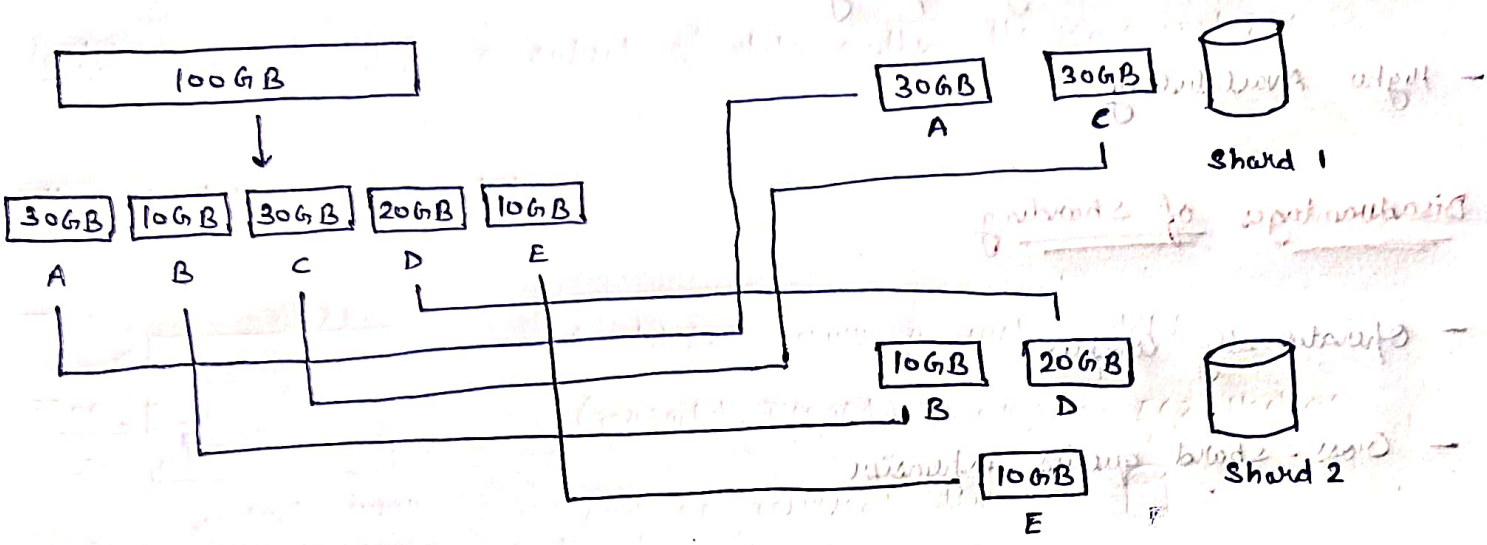
Overall, a DB is sharded while the data is partitioned



You partitioned the 100 GB of total data into 5 mutually exclusive partitions.

Each of these partitions can either live on one DB server or a couple of them can share one server.

And this depends on the no of shards you have.



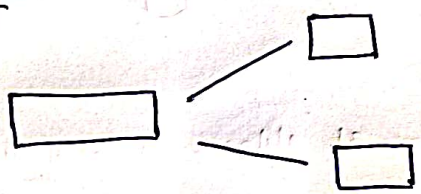
5 partitions of our 100 GB dataset is distributed across 2 shards

How to partition the data?

There are two categories of partitioning

① Horizontal Partitioning

② Vertical Partitioning



When we split the 100 GB data, we could have used either of the ways but deciding which one to pick depends on load, usecase & access pattern

Sharding \ Partitioning	No	Yes
No	 something you do locally on your machine	 Data is logically partitioned (like creating 2 DB in one MySQL server)
Yes	 Read Replica	 To handle large writes

Advantages of Sharding :-

- Handle large Reads and Writes
- Increase overall storage capacity
- Higher Availability

Disadvantages of Sharding

- Operationally Complex
- Cross-shard queries expensive