



#ASLI ENGINEERING

How to Implement Vertical Sharding



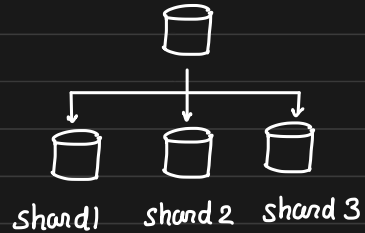
BY

ARPIT BHAYANI

How to implement Vertical Sharding

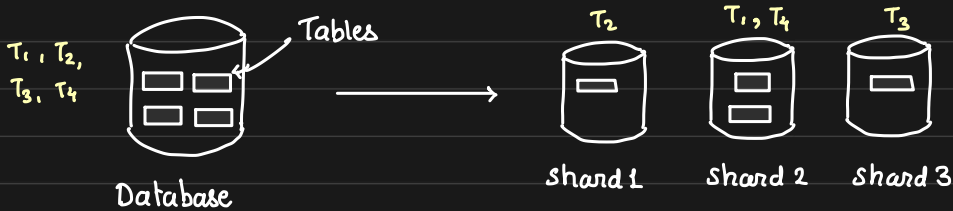
What is sharding?

Splitting a database across multiple machines is called Sharding



Vertical Sharding

Distributing tables across multiple shards



Idea: separate databases for set of tables

- Better load handling

eg: payments related tables in one DB server

- Handle larger scale

auth related tables in another DB server

* You will have to vertically shard the DB when moving to microservices

So, how do we implement Vertical Sharding?

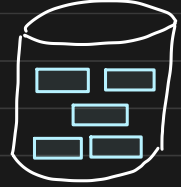


Move tables from one database server to another

Storing meta information and reactive update

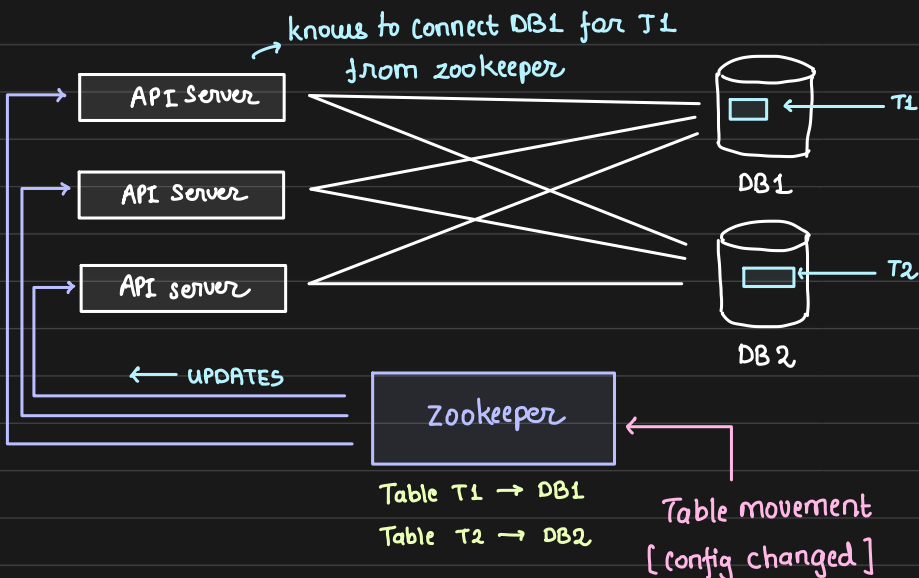
Given that we will have multiple databases
among which the tables will be distributed

We would need a way to store this info
enabling everyone to have a consistent view



Also, upon any changes in configuration (Table ownership) the API servers need to be notified.

Hence, we use **Zookeeper** to address both of these concerns



How to move tables from one MySQL to another?

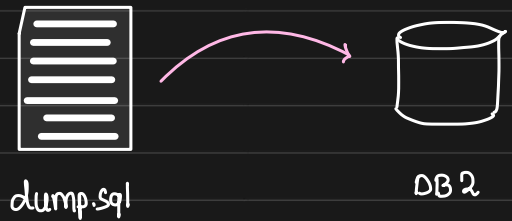


Move Table T_2 from DB1 to DB2
with minimal downtime

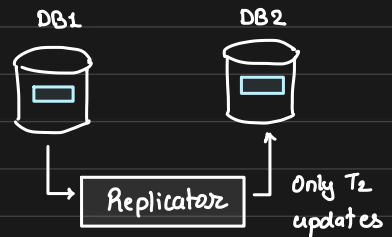
1. Dump the table T_2 along with Binlog position using 'mysqldump'. The dump will have the entire data of the table along with the binlog position.



2. Restore the dump to another database
load dump.sql to DB2



3. Start the replication between the two databases ensuring we only apply changes specific to the intended table T_2



Newer changes will thus flow to DB2

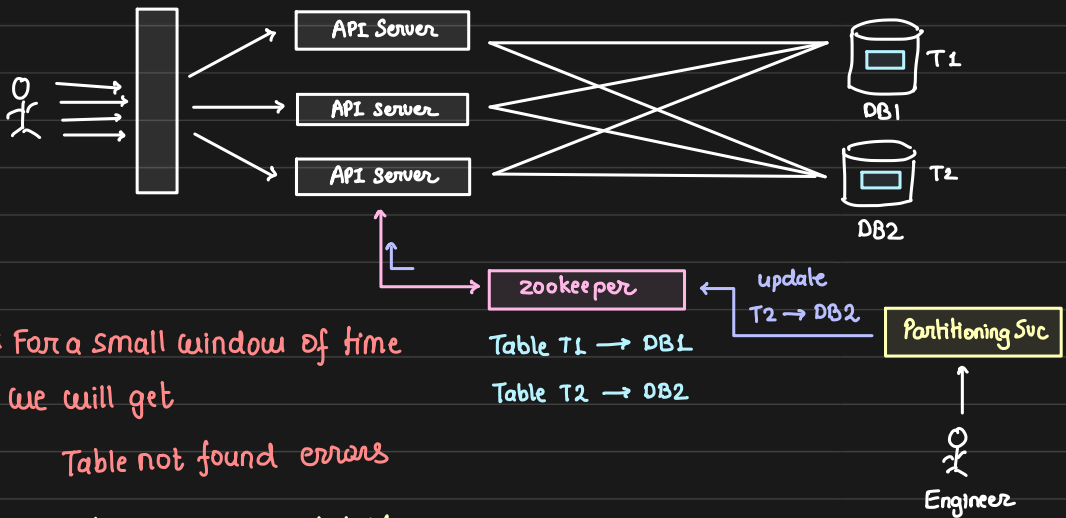
4. Once other database DB2 is almost caught-up,

a. Rename the table T_2 to $T_2\text{-bak}$

↳ sync b/w the 2 DB should stop as it encounters this statement

b. Update the entry in **Zookeeper** to say that table T_2 is now present on DB2

Once the changes are made to Zookeeper, they are reachively sent to API servers which then starts connecting to DB2 for table T_2



* The approach works well for small/medium tables
for huge tables replication lag will be an issue.