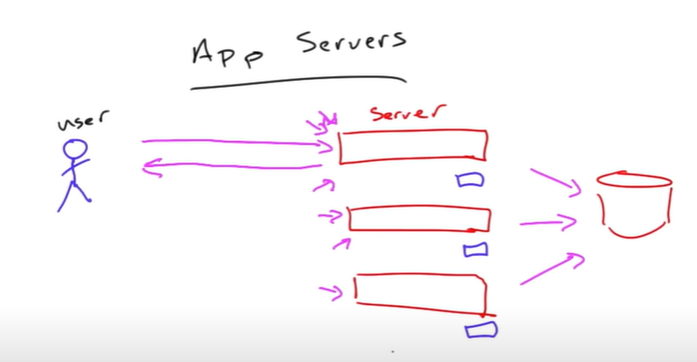
**Part 2 – Solution**

Problem : A plan for deploying and scaling the above application on a public cloud (AWS/GCP/Azure etc). This can include diagrams and documents and/or configuration scripts.

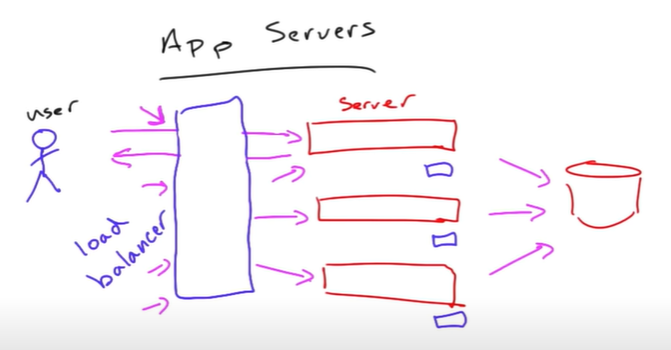
If you see any issues scaling this application, please provide suggestions as a document that you might have for resolving them.

**Solution**

1. Scaling of the application : When the load comes from the application side, we can place a load balancer in front of the EC2 instances. This can equally distribute the load between the instances. Or, we can horizontally scale it with new VMs.

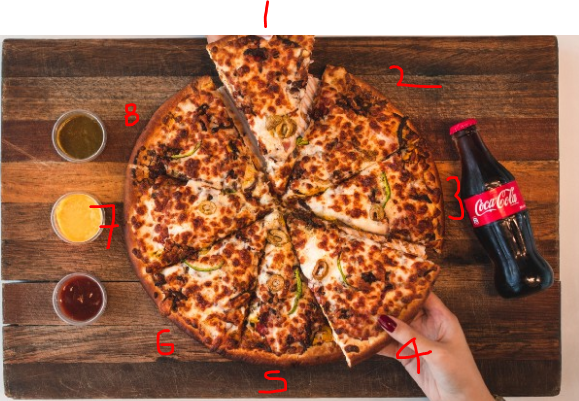


With Load Balancer

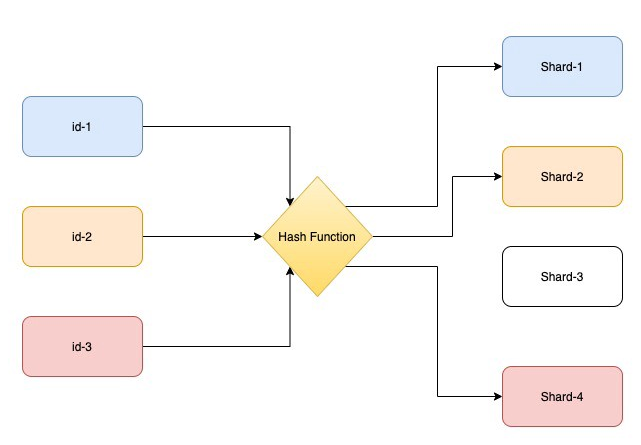


1. Scaling of the DB : Scaling of DB can be done vertically, but, max scaling can go to bare metal servers. Thus, the best practice is Sharding of the databases that should be done in the initial phase of the databases. Other ways are optimizing the queries, indexing the DB, replication etc

In sharding we divide the DB on the basis of the shard key/partition key using hash function.



Each number in the pizza here is the partition, and requests will go based on your keys. The hash function will convert the shard keys to a limited length of the key.



Since MySQL is an RDBMS it needs application level sharding.