Sharding

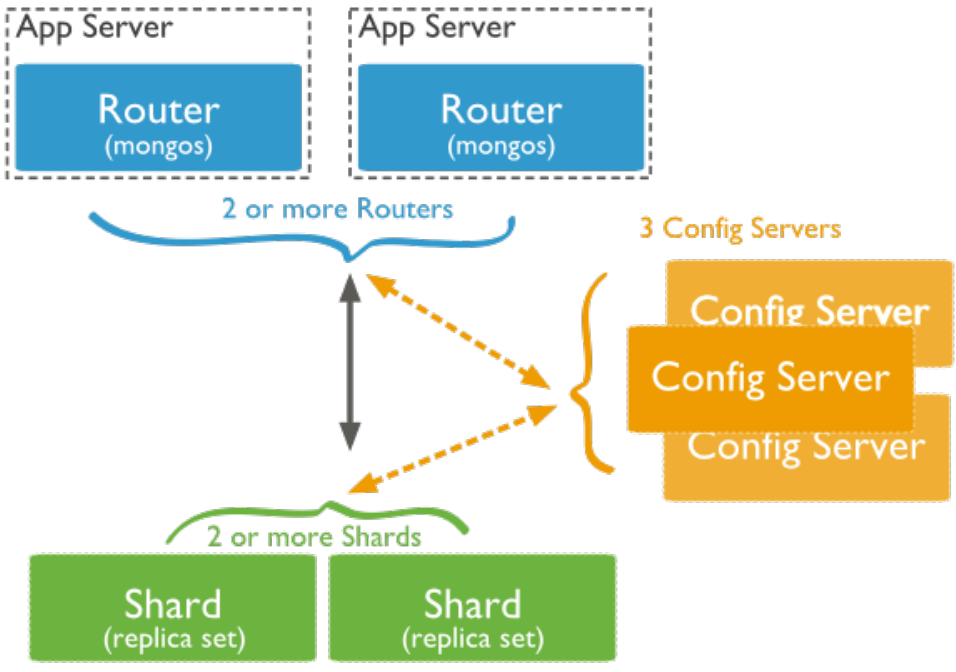
Sharding is the process of storing data records across multiple machines and it is MongoDB's approach to meeting the demands of data growth. As the size of the data increases, a single machine may not be sufficient to store the data nor provide an acceptable read and write throughput. Sharding solves the problem with horizontal scaling. With sharding, you add more machines to support data growth and the demands of read and write operations.

Why Sharding?

- In replication all writes go to master node
- Latency sensitive queries still go to master
- Single replica set has limitation of 12 nodes
- Memory can't be large enough when active dataset is big
- Local Disk is not big enough
- Vertical scaling is too expensive

Sharding in MongoDB

Below given diagram shows the sharding in MongoDB using sharded cluster.



In the above given diagram there are three main components which are described below:

- **Shards:** Shards are used to store data. They provide high availability and data consistency. In production environment each shard is a separate replica set.
- **Config Servers:** Config servers store the cluster's metadata. This data contains a mapping of the cluster's data set to the shards. The query router uses this metadata to target operations to specific shards. In production environment sharded clusters have exactly 3 config servers.
- **Query Routers:** Query Routers are basically mongos instances, interface with client applications and direct operations to the appropriate shard. The query router processes and

targets operations to shards and then returns results to the clients. A sharded cluster can contain more than one query router to divide the client request load. A client sends requests to one query router. Generally a sharded cluster have many query routers.