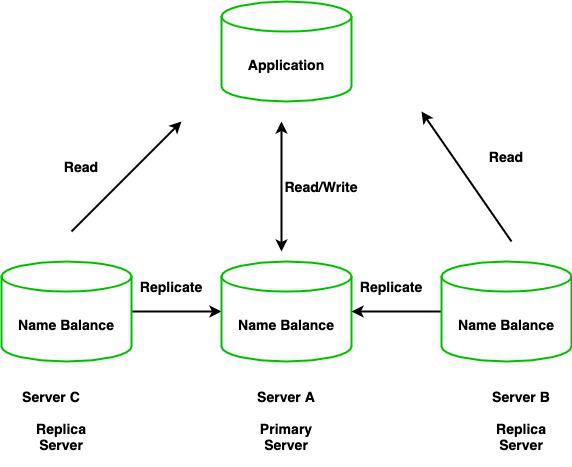
**MongoDB – Replication and Sharding**

In context to the scaling of the MongoDB database, it has some features know as Replication and Sharding. Replication can be simply understood as the duplication of the data-set whereas sharding is partitioning the data-set into discrete parts. By sharding, you divided your collection into different parts. Replicating your database means you make imagers of your data-set. In terms of functionality delivered.

**Replication**

Replication is the method of duplication of data across multiple servers. For example, we have an application and it reads and writes data to a database and says this server A has a name and balance which will be copied/replicate to two other servers in two different locations.



By doing this, will get redundancy and increases data availability with multiple copies of data on different database servers. So, it will increase the performance of reading scaling. The set of servers that maintain the same copy of data is known as replica servers or MongoDB instances.

**Replication Key Features :**

* Replica sets are the clusters of N different nodes that maintain the same copy of the data set.
* The primary server receives all write operations and record all the changes to the data i.e, oplog.
* The secondary members then copy and apply these changes in an asynchronous process.
* All the secondary nodes are connected with the primary nodes. there is one heartbeat signal from the primary nodes. If the primary server goes down an eligible secondary will hold the new primary.

**Why Replication?**

* High Availability of data disasters recovery
* No downtime for maintenance ( like backups index rebuilds and compaction)
* Read Scaling (Extra copies to read from)

**Sharding**

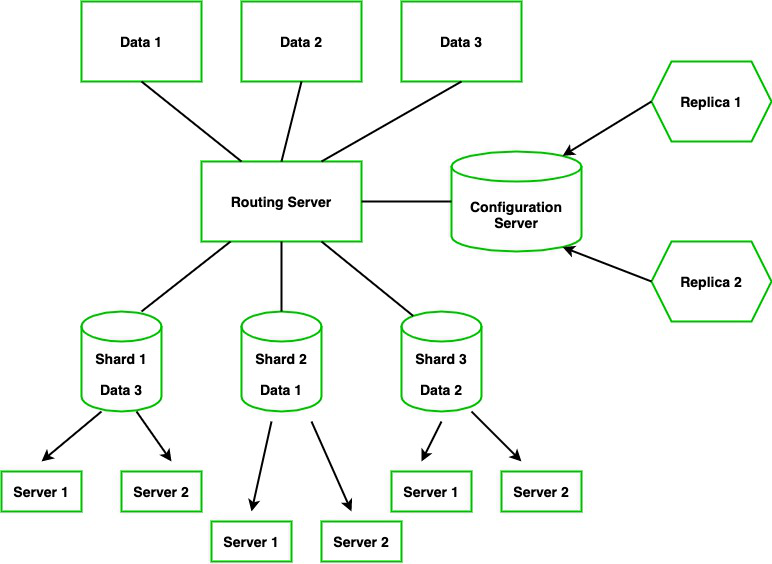
Sharding is a method for allocating data across multiple machines. MongoDB used sharding to help deployment with very big data sets and large throughput the operation. By sharding, you combine more devices to carry data extension and the needs of read and write operations.

**Why Sharding?**

* Database systems having big data sets or high throughput requests can doubt the ability of a single server.
* For example, High query flows can drain the CPU limit of the server.
* The working set sizes are larger than the system’s RAM to stress the I/O capacity of the disk drive.

**How does Sharding work?**

Sharding determines the problem with horizontal scaling breaking the system dataset and store over multiple servers, adding new servers to increase the volume as needed.



Now, instead of one signal as primary, we have multiple servers called Shard. We have different routing servers that will route data to the shard servers. For example: Let say we have Data 1, Data 2, and Data 3 this will be going to the routing server which will route the data (i.e, Different Data will go to a particular Shard ) Each Shard holds some pieces of data. Here the configuration server will hold the metadata and it will configure the routing server to integrate the particular data to a shard however configure server is the MongoDB instance if it goes down then the entire server will go down, So it again has Replica Configure database.

**Advantages of Sharding :**

* Sharding adds more server to a data field automatically adjust data loads across various servers.
* The number of operations each shard manage got reduced.
* It also increases the write capacity by splitting the write load over multiple instances.
* It gives high availability due to the deployment of replica servers for shard and config.
* Total capacity will get increased by adding multiple shards.

In order to create sharded clusters in MongoDB, We need to configure the shard, a config server, and a query router.