**Introduction to sharding**

**Sharding :**

Sharding is a process of distributing the data across multiple machines.The distribution is done based on the specific key called the shard key.This allows MongoDB to handle large datasets and higher read/write operations.

MongoDB supports horizontal scaling through Sharding.

**Benefits of using sharding**

**Scalability:** allows you to distribute the data across multiple servers, making it easier to scale horizontally as your data and can add as many shards as needed.

**Availability:**sharding improves fault tolerance by replicating data across multiple servers.if a shard fails, the other shards can continue to operate ensuring high data availability.

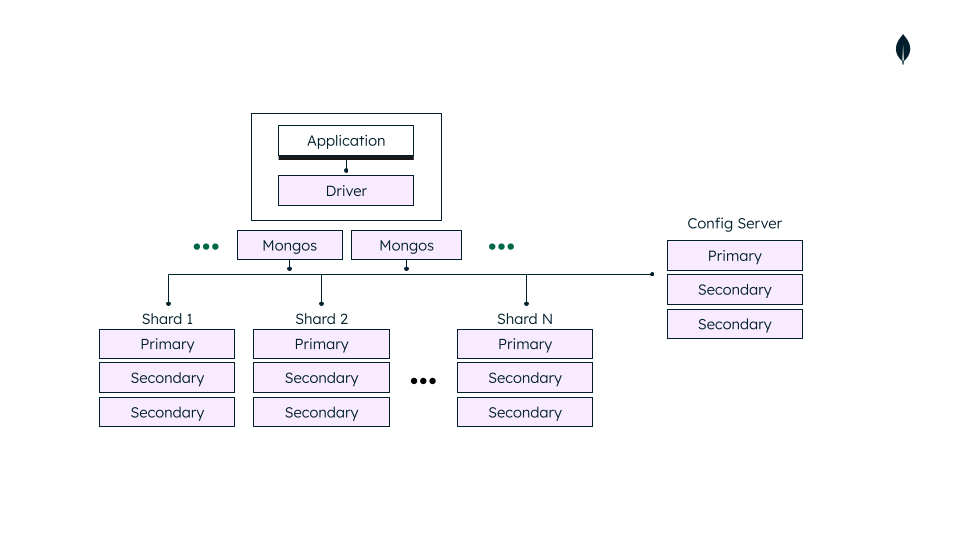
**Improved performance:** by using sharding queries can be distributed across the multiple shards leading to faster query responses times.mainly used for the application that have high throughput workloads.

**Isolation of workloads:** sharding enables you to isolate different types of workloads or applications on separate shards.

**Components in shared cluster**

1. Shard server
2. Config server
3. Query router

The image below gives you a brief idea about the sharding components.



**Shared server :**

* It is a single MongoDB server or any replica set that stores a subset of the data in a sharded cluster.
* Shards are responsible for storing and managing the portion of the total data.
* Each shard server is a replica set that consists of multiple copies of the data for the redundancy and high availability.
* MongoDB allows you to add more shards as your data grows.

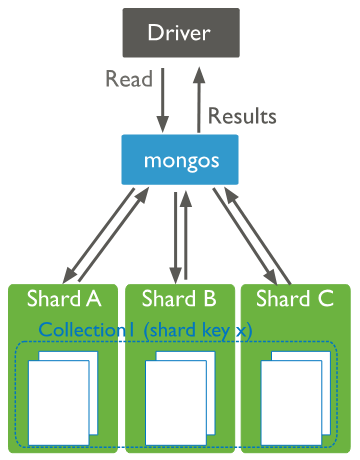
**Config servers:**

* config servers store metadata and configuration settings about the sharded cluster which includes the mapping between the chunks and shard keys,data stored on each shard, current state of each shard, shard key ranges and some settings that are related to the cluster.
* Provide the configuration that is required for query routing.
* Responsible for managing cluster configuration changes like adding and removing the shards on the requirements.
* Config servers also deployed in a replica set for reliability and redundancy.

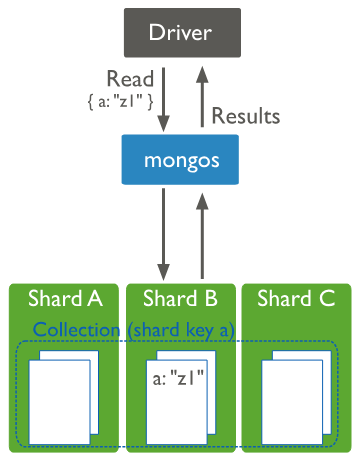
**Query router(mongos) :**

* Query routers are also known as mongos instances which act as intermediate between the client and sharded cluster, ensuring efficient and transparent data retrieval.
* These are responsible for directing the client request to the appropriate shard based on the sharding key within the sharded cluster.
* Query routers also handle load balancing to distribute the query traffic equally among the shards.
* Query routes use the metadata which is stored in the config servers to route the queries which are from the client.

The image below shows mongos instances sending the queries to all the shards for the collection if they did not figure out to which shard stores the data that is requested from the client.



If you have any shard key then mongos will send queries to that specific shard or set of shards which is shown below.



**Applications of sharding**

We use sharding when there is a need to manage the large and growing datasets. Below are some of the applications.

1. E-commerce and retail
2. Social media
3. Gaming
4. Internet of Things
5. Big data analytics