**MongoDB Sharding**

Sharding is a method for distributing data across multiple machines. MongoDB uses sharding to support deployments with very large data sets and high throughput operations.

There are two methods for addressing system growth:   
vertical and horizontal scaling.

**Vertical Scaling**

involves increasing the capacity of a single server, such as using a more powerful CPU, adding more RAM, or increasing the amount of storage space. Limitations in available technology may restrict a single machine from being sufficiently powerful for a given workload. Additionally, Cloud-based providers have hard ceilings based on available hardware configurations. As a result, there is a practical maximum for vertical scaling.

**Horizontal Scaling**

involves dividing the system dataset and load over multiple servers, adding additional servers to increase capacity as required. While the overall speed or capacity of a single machine may not be high, each machine handles a subset of the overall workload, potentially providing better efficiency than a single high-speed high-capacity server. Expanding the capacity of the deployment only requires adding additional servers as needed, which can be a lower overall cost than high-end hardware for a single machine. The trade off is increased complexity in infrastructure and maintenance for the deployment.

**Sharded Cluster**

A MongoDB sharded cluster consists of the following components:

shard: Each shard contains a subset of the sharded data. Each shard can be deployed as a replica set.

mongos: The mongos acts as a query router, providing an interface between client applications and the sharded cluster. Starting in MongoDB 4.4, mongos can support hedged reads to minimize latencies.

config servers: Config servers store metadata and configuration settings for the cluster.

**Advantages of Sharding**

**Read/Writes**

1. MongoDB distributes the read and write workload across the shards in the sharded cluster, allowing each shard to process a subset of cluster operations. Both read and write workloads can be scaled horizontally across the cluster by adding more shards.

2. For queries that include the shard key or the prefix of a compound shard key, mongos can target the query at a specific shard or set of shards. These targeted operations are generally more efficient than broadcasting to every shard in the cluster.

**Storage Capacity**

Sharding distributes data across the shards in the cluster, allowing each shard to contain a subset of the total cluster data. As the data set grows, additional shards increase the storage capacity of the cluster.

**High Availability**

The deployment of config servers and shards as replica sets provide increased availability.

Even if one or more shard replica sets become completely unavailable, the sharded cluster can continue to perform partial reads and writes. That is, while data on the unavailable shard(s) cannot be accessed, reads or writes directed at the available shards can still succeed.

**Troubles with Sharding**

**What If? Mongos Instance Becomes Absent**

If each application server has its own mongos instance, other application servers can continue to access the database. Furthermore, mongos instances do not maintain persistent state, and they can restart and become unavailable without losing any state or data. When a mongos instance starts, it retrieves a copy of the config database and can begin routing queries.

**What If? A Member Becomes Absent From the Shard Replica Set**

Start by checking the status of the shard by running the command sh.status(). If the returned result does not have the clusterId then the shard is indeed unavailable. Always investigate availability interruptions and failures  and if you are unable to recover it in the shortest time possible, create a new member to replace it  as soon as possible so as to avoid more data loss.

**What If? All Members of a Replica Set are Absent**

Data held in a shard  will be unavailable if all members of a replica set shard become absent. Since the other shards remain available, read and write operations are still possible except that the application will be served with partial data. You will need to investigate the cause of the interruptions and attempt reactivating the shard as soon as possible. Check which query profiler or the explain method what might have led to that problem.