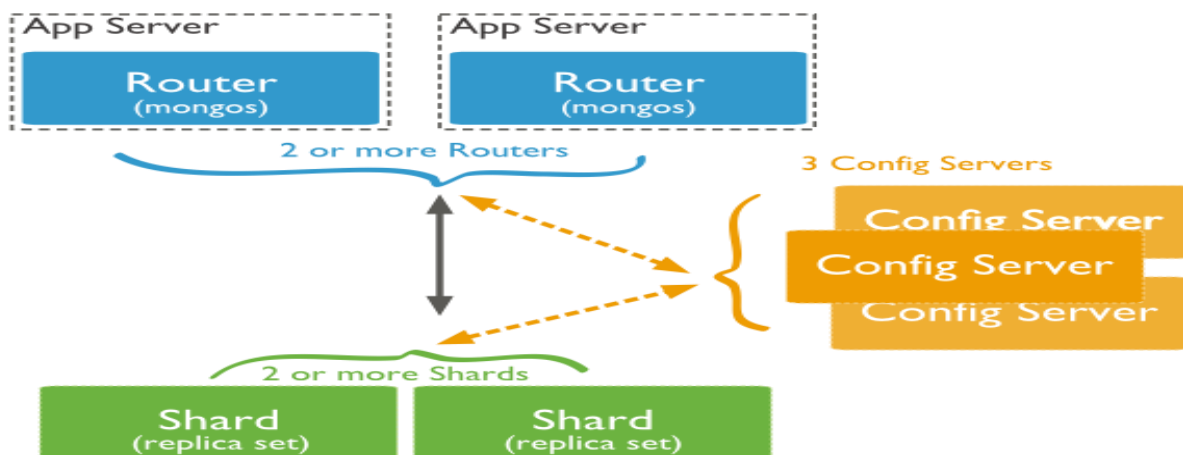


❖ What is sharding ?

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.

❖ Sharded Cluster

- A MongoDB sharded cluster consists of the following components.
 1. Shard: each shard contains a subset of the sharded data each sharded can be deployed as a replica set.
 2. Mongos: The mongos acts as a query router, providing an interface between client applications and the sharded cluster.
 3. Config Servers: Config servers store metadata and configuration settings for the cluster.



❖ In the following diagram, there are three main components –

- **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 mongo instances, interface with client applications and direct operations to the appropriate shard. The query router processes and targets the 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.

❖ Step for set up sharding

1. First create two shard servers
2. Start first shard server by specifying the following command

- F:\mongodb-win32-x86_64-2012plus-4.2.3\mongodb-win32-x86_64-2012plus-4.2.3\bin>mongod --shardsvr --dbpath "F:\mongodb-win32-x86_64-2012plus-4.2.3\mongodb-win32-x86_64-2012plus-4.2.3\data" --port 27020 --bind_ip 0.0.0.0

```
F:\mongodb-win32-x86_64-2012plus-4.2.3\mongodb-win32-x86_64-2012plus-4.2.3\bin>mongod --shardsvr --dbpath "F:\mongodb-win32-x86_64-2012plus-4.2.3\mongodb-win32-x86_64-2012plus-4.2.3\data" --port 27020 --bind_ip 0.0.0.0
2020-03-26T04:39:41.417+0530 I CONTROL [main] Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'
2020-03-26T04:39:41.433+0530 I CONTROL [initandlisten] MongoDB starting : pid=9884 port=27020 dbpath=F:\mongodb-win32-x86_64-2012plus-4.2.3\mongodb-win32-x86_64-2012plus-4.2.3\data 64-bit host=ThisPc
2020-03-26T04:39:41.433+0530 I CONTROL [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2020-03-26T04:39:41.433+0530 I CONTROL [initandlisten] db version v4.2.3
2020-03-26T04:39:41.433+0530 I CONTROL [initandlisten] git version: 6874650b362138df74be53d366bbefc321ea32d4
2020-03-26T04:39:41.433+0530 I CONTROL [initandlisten] allocator: tcmalloc
2020-03-26T04:39:41.433+0530 I CONTROL [initandlisten] modules: none
2020-03-26T04:39:41.434+0530 I CONTROL [initandlisten] build environment:
2020-03-26T04:39:41.434+0530 I CONTROL [initandlisten] distmod: 2012plus
2020-03-26T04:39:41.434+0530 I CONTROL [initandlisten] distarch: x86_64
2020-03-26T04:39:41.434+0530 I CONTROL [initandlisten] target_arch: x86_64
2020-03-26T04:39:41.434+0530 I CONTROL [initandlisten] options: { net: { bindIp: "0.0.0.0", port: 27020 }, sharding: { clusterRole: "shardsvr" }, storage: { dbPath: "F:\mongodb-win32-x86_64-2012plus-4.2.3\mongodb-win32-x86_64-2012plus-4.2.3\data" } }
2020-03-26T04:39:41.466+0530 I STORAGE [initandlisten] wiredtiger open config: create,cache size=1488M,cache overflow=(file_max=0M),session_max=33000,eviction=(thread_min=4,threads_max=4),config base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=100000,close_scan_interval=10,close_handle_minimum=250),statistics_log=(wait=0),verbose=[recovery_progress,checkpoint_progress],
2020-03-26T04:39:41.918+0530 I STORAGE [initandlisten] WiredTiger message [1585177781:917848][9884:140733739654736], txn-recover: Set global recovery timestamp: (0, 0)
2020-03-26T04:39:42.078+0530 I RECOVERY [initandlisten] WiredTiger recoveryTimestamp. Ts: Timestamp(0, 0)
2020-03-26T04:39:42.296+0530 I STORAGE [initandlisten] Timestamp monitor starting
2020-03-26T04:39:42.467+0530 I CONTROL [initandlisten]
2020-03-26T04:39:42.467+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-03-26T04:39:42.469+0530 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
2020-03-26T04:39:42.470+0530 I CONTROL [initandlisten]
2020-03-26T04:39:42.473+0530 I STORAGE [initandlisten] createCollection: admin.system.version with provided UUID: e61889ac-b989-47c6-9fc7-ffc7ea46e2e and options: {
  uuid: UUID("e61889ac-b989-47c6-9fc7-ffc7ea46e2e")
}
```

3. Start the client by specifying the following command.

- F:\mongodb-win32-x86_64-2012plus-4.2.3\mongodb-win32-x86_64-2012plus-4.2.3\bin>mongo --port 27020

```
C:\Windows\System32\cmd.exe - mongo --port 27020
Microsoft Windows [Version 10.0.18362.657]
(c) 2019 Microsoft Corporation. All rights reserved.

F:\mongodb-win32-x86_64-2012plus-4.2.3\mongodb-win32-x86_64-2012plus-4.2.3\bin>mongo --port 27020
MongoDB shell version v4.2.3
connecting to: mongodb://127.0.0.1:27020/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("22cd31f3-1789-4d8c-85c1-859c38ffcd1d") }
MongoDB server version: 4.2.3
Server has startup warnings:
2020-03-26T04:39:42.467+0530 I CONTROL [initandlisten]
2020-03-26T04:39:42.467+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-03-26T04:39:42.469+0530 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
2020-03-26T04:39:42.470+0530 I CONTROL [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
```

4. Start second shard server by specifying the following command

- F:\mongodb-win32-x86_64-2012plus-4.2.3\mongodb-win32-x86_64-2012plus-4.2.3\bin>mongod --shardsvr --dbpath "F:\mongodb-win32-x86_64-2012plus-4.2.3\mongodb-win32-x86_64-2012plus-4.2.3\data1" --port 27021 --bind_ip 0.0.0.0

```

C:\Windows\System32\cmd.exe - mongod --shardsvr --dbpath "F:\mongodb-win32-x86_64-2012plus-4.2.3\mongodb-win32-x86_64-2012plus-4.2.3\data1" --port 27021 --bind_ip 0.0.0.0
Microsoft Windows [Version 10.0.18362.657]
(c) 2019 Microsoft Corporation. All rights reserved.

F:\mongodb-win32-x86_64-2012plus-4.2.3>mongodb-win32-x86_64-2012plus-4.2.3\bin>mongod --shardsvr --dbpath "F:\mongodb-win32-x86_64-2012plus-4.2.3\data1" --port 27021 --bind_ip 0.0.0.0
2020-03-26T04:42:13.649+0530 I CONTROL [main] Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'
2020-03-26T04:42:14.306+0530 I CONTROL [initandlisten] MongoDB starting : pid=9324 port=27021 dbpath=F:\mongodb-win32-x86_64-2012plus-4.2.3\data1 64-bit host=ThisPc
2020-03-26T04:42:14.306+0530 I CONTROL [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2020-03-26T04:42:14.306+0530 I CONTROL [initandlisten] db version v4.2.3
2020-03-26T04:42:14.307+0530 I CONTROL [initandlisten] git version: 6874650b362138df74be53d366bbefc321ea32d4
2020-03-26T04:42:14.307+0530 I CONTROL [initandlisten] allocator: tcmalloc
2020-03-26T04:42:14.307+0530 I CONTROL [initandlisten] modules: none
2020-03-26T04:42:14.307+0530 I CONTROL [initandlisten] build environment:
2020-03-26T04:42:14.307+0530 I CONTROL [initandlisten] distmod: 2012plus
2020-03-26T04:42:14.308+0530 I CONTROL [initandlisten] distarch: x86_64
2020-03-26T04:42:14.308+0530 I CONTROL [initandlisten] target_arch: x86_64
2020-03-26T04:42:14.308+0530 I CONTROL [initandlisten] options: { net: { bindip: "0.0.0.0", port: 27021 }, sharding: { clusterRole: "shardsvr" }, storage: { dbPath: "F:\mongodb-win32-x86_64-2012plus-4.2.3\mongodb-win32-x86_64-2012plus-4.2.3\data1" } }
2020-03-26T04:42:14.312+0530 I STORAGE [initandlisten] wiredtiger open config: create,cache_size=1488M,cache_overflow=(file_max=0M),session_max=33000,eviction=(thread_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=100000,close_scan_interval=10,close_handle_minimum=250),statistics_log=(wait=0),verbose=[recovery_progress,checkpoint_progress],
2020-03-26T04:42:14.554+0530 I STORAGE [initandlisten] WiredTiger message [1585177934:553559][9324:140733739654736], txn-recover: Set global recovery timestamp: (0, 0)
2020-03-26T04:42:14.707+0530 I RECOVERY [initandlisten] WiredTiger recoveryTimestamp: Ts: Timestamp(0, 0)
2020-03-26T04:42:14.881+0530 I STORAGE [initandlisten] Timestamp monitor starting
2020-03-26T04:42:14.941+0530 I CONTROL [initandlisten]
2020-03-26T04:42:14.941+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-03-26T04:42:14.942+0530 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
2020-03-26T04:42:14.945+0530 I CONTROL [initandlisten]
2020-03-26T04:42:14.947+0530 I STORAGE [initandlisten] createCollection: admin.system.version with provided UUID: a45520b2-0456-4383-a02f-4e9c47d92720 and options: { uuid: UUID("a45520b2-0456-4383-a02f-4e9c47d92720") }
2020-03-26T04:42:15.083+0530 I INDEX [initandlisten] index build: done building index_id on ns admin.system.version
2020-03-26T04:42:15.085+0530 I SHARDING [initandlisten] Marking collection admin.system.version as collection version: <unsharded>
2020-03-26T04:42:15.085+0530 I COMMAND [initandlisten] setting featureCompatibilityVersion to 4.0
2020-03-26T04:42:15.090+0530 I SHARDING [initandlisten] Marking collection local.system.indexes as collection version: <unsharded>

```

5. Start the client by specifying the following command.

- F:\mongodb-win32-x86_64-2012plus-4.2.3\mongodb-win32-x86_64-2012plus-4.2.3\bin> mongo --port 27021

```

F:\mongodb-win32-x86_64-2012plus-4.2.3\mongodb-win32-x86_64-2012plus-4.2.3\bin>mongo --port 27021
MongoDB shell version v4.2.3
connecting to: mongodb://127.0.0.1:27021/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("a115e371-c0d3-4406-b23c-39c65ff24a6a") }
MongoDB server version: 4.2.3
Server has startup warnings:
2020-03-26T04:42:14.941+0530 I CONTROL [initandlisten]
2020-03-26T04:42:14.941+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-03-26T04:42:14.942+0530 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
2020-03-26T04:42:14.945+0530 I CONTROL [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
>

```

6. Create the config server replica set

- First make one data folder in that make another configdb folder.
- F:\mongodb-win32-x86_64-2012plus-4.2.3\mongodb-win32-x86_64-2012plus-4.2.3\bin> mongod --configsvr --dbpath "F:\mongodb-win32-x86_64-2012plus-4.2.3\mongodb-win32-x86_64-2012plus-4.2.3\data3\config" --port 27022 --bind_ip 0.0.0.0 --replSet yash

- F:\mongodb-win32-x86_64-2012plus-4.2.3\mongodb-win32-x86_64-2012plus-4.2.3\bin>mongos --configdb yash/127.0.0.1:27022 --port 27024
- “yash” is the name of config server replica set.
- “27022” is the port number of the config server.

```

C:\Windows\System32\cmd.exe - mongos --configdb yash/127.0.0.1:27022 --port 27024
Microsoft Windows [Version 10.0.18362.657]
(c) 2019 Microsoft Corporation. All rights reserved.

F:\mongodb-win32-x86_64-2012plus-4.2.3\mongodb-win32-x86_64-2012plus-4.2.3\bin>mongos --configdb yash/127.0.0.1:27022 --port 27024
2020-03-26T04:51:16.756+0530 W SHARDING [main] Running a sharded cluster with fewer than 3 config servers should only be done for testing purposes and is not recommended for production.
2020-03-26T04:51:16.769+0530 I CONTROL [main] Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'
2020-03-26T04:51:17.173+0530 I CONTROL [main]
2020-03-26T04:51:17.173+0530 I CONTROL [main] ** WARNING: Access control is not enabled for the database.
2020-03-26T04:51:17.173+0530 I CONTROL [main] ** Read and write access to data and configuration is unrestricted.
2020-03-26T04:51:17.174+0530 I CONTROL [main]
2020-03-26T04:51:17.174+0530 I CONTROL [main] ** WARNING: This server is bound to localhost.
2020-03-26T04:51:17.174+0530 I CONTROL [main] ** Remote systems will be unable to connect to this server.
2020-03-26T04:51:17.174+0530 I CONTROL [main] ** Start the server with --bind_ip <address> to specify which IP
2020-03-26T04:51:17.174+0530 I CONTROL [main] ** addresses it should serve responses from, or with --bind_ip_all to
2020-03-26T04:51:17.174+0530 I CONTROL [main] ** bind to all interfaces. If this behavior is desired, start the
2020-03-26T04:51:17.174+0530 I CONTROL [main] ** server with --bind_ip 127.0.0.1 to disable this warning.
2020-03-26T04:51:17.174+0530 I CONTROL [main]
2020-03-26T04:51:17.176+0530 I SHARDING [mongosMain] mongos version v4.2.3
2020-03-26T04:51:17.176+0530 I CONTROL [mongosMain] db version v4.2.3
2020-03-26T04:51:17.176+0530 I CONTROL [mongosMain] git version: 6874650b362138df74be53d366bbefc321ea32d4
2020-03-26T04:51:17.176+0530 I CONTROL [mongosMain] allocator: tcmalloc
2020-03-26T04:51:17.176+0530 I CONTROL [mongosMain] modules: none
2020-03-26T04:51:17.176+0530 I CONTROL [mongosMain] build environment:
2020-03-26T04:51:17.176+0530 I CONTROL [mongosMain] distmod: 2012plus
2020-03-26T04:51:17.177+0530 I CONTROL [mongosMain] distarch: x86_64
2020-03-26T04:51:17.178+0530 I CONTROL [mongosMain] target_arch: x86_64
2020-03-26T04:51:17.179+0530 I CONTROL [mongosMain] options: { net: { port: 27024 }, sharding: { configDB: "yash/127.0.0.1:27022" } }
2020-03-26T04:51:17.185+0530 I NETWORK [mongosMain] Starting new replica set monitor for yash/127.0.0.1:27022
2020-03-26T04:51:17.188+0530 I SHARDING [ReplicaSetMonitor-TaskExecutor] Connecting to 127.0.0.1:27022
2020-03-26T04:51:17.188+0530 I SHARDING [thread1] creating distributed lock ping thread for process ThisPc:27024:1585178477-6398319450128296495 (sleeping for 30000ms)
2020-03-26T04:51:17.234+0530 I NETWORK [ReplicaSetMonitor-TaskExecutor] Confirmed replica set for yash is yash/127.0.0.1:27022
2020-03-26T04:51:17.234+0530 I SHARDING [Sharding-Fixed-0] Updating sharding state with confirmed set yash/127.0.0.1:27022
2020-03-26T04:51:17.252+0530 I SHARDING [ShardRegistry] Received reply from config server node (unknown) indicating config server optime term has increased, previous optime { ts: Timestamp(0, 0), t: -1 }, now { ts: Timestamp(1585178474, 1), t: 1 }
2020-03-26T04:51:17.294+0530 W SHARDING [replSetDistLockPinger] pinging failed for distributed lock pinger :: caused by :: LockStateChangeFailed: findAndModify query predicate didn't match any lock document
2020-03-26T04:51:17.295+0530 W FTDC [mongosMain] FTDC is disabled because neither '--logpath' nor set parameter 'diagnosticDataCollectionDirectoryPath' are specified.
2020-03-26T04:51:17.826+0530 I FTDC [mongosMain] Initializing full-time diagnostic data capture with directory ''
2020-03-26T04:51:17.844+0530 I NETWORK [listener] Listening on 127.0.0.1
2020-03-26T04:51:17.844+0530 I NETWORK [listener] waiting for connections on port 27024
2020-03-26T04:51:17.844+0530 I SH REPR [configServerCatalogCacheLoader-0] Refresh for database config from version 1 to version 1 uuid: UUID("774323f4h-8f08-49c0-8150-")

```

9. Connect to the router server.

- F:\mongodb-win32-x86_64-2012plus-4.2.3\mongodb-win32-x86_64-2012plus-4.2.3\bin>mongo --port 27024

```

F:\mongodb-win32-x86_64-2012plus-4.2.3\mongodb-win32-x86_64-2012plus-4.2.3\bin>mongo --port 27024
MongoDB shell version v4.2.3
connecting to: mongodb://127.0.0.1:27024/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("60f14528-108a-428a-8fac-6571bc909800") }
MongoDB server version: 4.2.3
Server has startup warnings:
2020-03-26T04:51:17.173+0530 I CONTROL [main]
2020-03-26T04:51:17.173+0530 I CONTROL [main] ** WARNING: Access control is not enabled for the database.
2020-03-26T04:51:17.173+0530 I CONTROL [main] ** Read and write access to data and configuration is unrestricted.
2020-03-26T04:51:17.174+0530 I CONTROL [main]
2020-03-26T04:51:17.174+0530 I CONTROL [main] ** WARNING: This server is bound to localhost.
2020-03-26T04:51:17.174+0530 I CONTROL [main] ** Remote systems will be unable to connect to this server.
2020-03-26T04:51:17.174+0530 I CONTROL [main] ** Start the server with --bind_ip <address> to specify which IP
2020-03-26T04:51:17.174+0530 I CONTROL [main] ** addresses it should serve responses from, or with --bind_ip_all to
2020-03-26T04:51:17.174+0530 I CONTROL [main] ** bind to all interfaces. If this behavior is desired, start the
2020-03-26T04:51:17.174+0530 I CONTROL [main] ** server with --bind_ip 127.0.0.1 to disable this warning.
2020-03-26T04:51:17.174+0530 I CONTROL [main]

```

10. Now add both the shard servers into the cluster by specifying the following command.

- sh.addShard("127.0.0.1:27020")

```

C:\Windows\System32\cmd.exe - mongo --port 27024
Microsoft Windows [Version 10.0.18362.657]
(c) 2019 Microsoft Corporation. All rights reserved.

F:\mongodb-win32-x86_64-2012plus-4.2.3\mongodb-win32-x86_64-2012plus-4.2.3\bin>mongo --port 27024
MongoDB shell version v4.2.3
connecting to: mongodb://127.0.0.1:27024/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("60f14528-108a-428a-8fac-6571bc909800") }
MongoDB server version: 4.2.3
Server has startup warnings:
2020-03-26T04:51:17.173+0530 I CONTROL [main]
2020-03-26T04:51:17.173+0530 I CONTROL [main] ** WARNING: Access control is not enabled for the database.
2020-03-26T04:51:17.173+0530 I CONTROL [main] ** Read and write access to data and configuration is unrestricted.
2020-03-26T04:51:17.174+0530 I CONTROL [main]
2020-03-26T04:51:17.174+0530 I CONTROL [main] ** WARNING: This server is bound to localhost.
2020-03-26T04:51:17.174+0530 I CONTROL [main] ** Remote systems will be unable to connect to this server.
2020-03-26T04:51:17.174+0530 I CONTROL [main] ** Start the server with --bind_ip <address> to specify which IP
2020-03-26T04:51:17.174+0530 I CONTROL [main] ** addresses it should serve responses from, or with --bind_ip_all to
2020-03-26T04:51:17.174+0530 I CONTROL [main] ** bind to all interfaces. If this behavior is desired, start the
2020-03-26T04:51:17.174+0530 I CONTROL [main] ** server with --bind_ip 127.0.0.1 to disable this warning.
2020-03-26T04:51:17.174+0530 I CONTROL [main]
mongos> sh.addShard("127.0.0.1:27020")
{
  "shardAdded" : "shard0000",
  "ok" : 1,
  "operationTime" : Timestamp(1585178821, 4),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1585178821, 4),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"),
      "keyId" : NumberLong(0)
    }
  }
}

```


- sh.addShard("127.0.0.1:27021")

```

C:\Windows\System32\cmd.exe - mongo --port 27024
connecting to: mongod://127.0.0.1:27024/?compressors=disabled&gssapiServiceName=mongod
Implicit session: session { "id" : UUID("60f14528-108a-428a-8fac-6571bc909800") }
MongoDB server version: 4.2.3
Server has startup warnings:
2020-03-26T04:51:17.173+0530 I CONTROL [main] ** WARNING: Access control is not enabled for the database.
2020-03-26T04:51:17.173+0530 I CONTROL [main] **          Read and write access to data and configuration is unrestricted.
2020-03-26T04:51:17.174+0530 I CONTROL [main] ** WARNING: This server is bound to localhost.
2020-03-26T04:51:17.174+0530 I CONTROL [main] **          Remote systems will be unable to connect to this server.
2020-03-26T04:51:17.174+0530 I CONTROL [main] **          Start the server with --bind_ip <address> to specify which IP
2020-03-26T04:51:17.174+0530 I CONTROL [main] **          addresses it should serve responses from, or with --bind_ip_all to
2020-03-26T04:51:17.174+0530 I CONTROL [main] **          bind to all interfaces. If this behavior is desired, start the
2020-03-26T04:51:17.174+0530 I CONTROL [main] **          server with --bind_ip 127.0.0.1 to disable this warning.
mongos> sh.addShard("127.0.0.1:27020")
{
  "shardAdded" : "shard0000",
  "ok" : 1,
  "operationTime" : Timestamp(1585178821, 4),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1585178821, 4),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"),
      "keyId" : NumberLong(0)
    }
  }
}
mongos> sh.addShard("127.0.0.1:27021")
{
  "shardAdded" : "shard0001",
  "ok" : 1,
  "operationTime" : Timestamp(1585178848, 3),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1585178848, 3),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"),
      "keyId" : NumberLong(0)
    }
  }
}

```

11. Now check whether shard servers added or not .

- mongos> db.adminCommand({listShards:1})

```

C:\Windows\System32\cmd.exe - mongo --port 27024
"keyId" : NumberLong(0)
}
}
mongos> sh.addShard("127.0.0.1:27021")
{
  "shardAdded" : "shard0001",
  "ok" : 1,
  "operationTime" : Timestamp(1585178848, 3),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1585178848, 3),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"),
      "keyId" : NumberLong(0)
    }
  }
}
mongos> db.adminCommand({listShards:1})
{
  "shards" : [
    {
      "_id" : "shard0000",
      "host" : "127.0.0.1:27020",
      "state" : 1
    },
    {
      "_id" : "shard0001",
      "host" : "127.0.0.1:27021",
      "state" : 1
    }
  ],
  "ok" : 1,
  "operationTime" : Timestamp(1585178908, 1),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1585178908, 1),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"),
      "keyId" : NumberLong(0)
    }
  }
}

```

12 . Now enable the sharding and use that database and make a collection by specifying the following command.

- mongos> sh.enableSharding("Employee")

```

mongos> sh.enableSharding("Employee")
{
  "ok" : 1,
  "operationTime" : Timestamp(1585178963, 5),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1585178963, 5),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  }
}

```

- mongos> use Employee

```

mongos> use Employee
switched to db Employee
mongos> db.createCollection("Employee_info")
{
  "ok" : 1,
  "operationTime" : Timestamp(1585179021, 4),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1585179021, 4),

```

- mongos> db.createCollection("Employee_info")

```

C:\Windows\System32\cmd.exe - mongo --port 27024
{
  "ok" : 1,
  "operationTime" : Timestamp(1585178963, 5),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1585178963, 5),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  }
}
mongos> use Employee
switched to db Employee
mongos> db.createCollection("Employee_info")
{
  "ok" : 1,
  "operationTime" : Timestamp(1585179021, 4),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1585179021, 4),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  }
}

```

13. Now shard collection by specifying the following command.

- mongos> sh.shardCollection("Employee.Employee_info",{_id:"hashed"});

```

C:\Windows\System32\cmd.exe - mongo --port 27024
{
  "$clusterTime" : {
    "clusterTime" : Timestamp(1585179021, 4),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  }
}
mongos> sh.shardCollection("Employee.Employee_info",{_id:"hashed"});
{
  "collectionsharded" : "Employee.Employee_info",
  "collectionUUID" : UUID("e1fd5085-721e-4e2d-a03a-2515a5aa5801"),
  "ok" : 1,
  "operationTime" : Timestamp(1585179194, 3),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1585179194, 3),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  }
}

```

14. Now insert the records in the collection

- mongos> db.Employee_info.insert({"id":1,"name":"yash"})

```

C:\Windows\System32\cmd.exe - mongo --port 27024
}
mongo> sh.shardCollection("Employee.Employee_info",{_id:"hashed"});
{
  "collectionsharded" : "Employee.Employee_info",
  "collectionUUID" : UUID("e1fd5085-721e-4e2d-a03a-2515a5aa5801"),
  "ok" : 1,
  "operationTime" : Timestamp(1585179194, 3),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1585179194, 3),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  }
}
mongo> db.Employee_info.insert({id:1,"name":yash})
2020-03-26T05:07:14.389+0530 E QUERY [js] uncaught exception: ReferenceError: yash is not defined :
@(<shell>):1:31
mongo> db.Employee_info.insert({id:1,name:yash})
2020-03-26T05:07:29.790+0530 E QUERY [js] uncaught exception: ReferenceError: yash is not defined :
@(<shell>):1:31
mongo> db.Employee_info.insert({"id":1,"name":"yash"})
WriteResult({ "nInserted" : 1 })
mongo> db.Employee_info.insert({"id":2,"name":"Tilak"})
WriteResult({ "nInserted" : 1 })
mongo> db.Employee_info.insert({"id":3,"name":"Mayank"})
WriteResult({ "nInserted" : 1 })
mongo>

```

15. Go to the shard servers and verify that the data are inserted or not

- Shard server 1

```

C:\Windows\System32\cmd.exe - mongo --port 27020
Microsoft Windows [Version 10.0.18362.657]
(c) 2019 Microsoft Corporation. All rights reserved.

F:\mongodb-win32-x86_64-2012plus-4.2.3\mongodb-win32-x86_64-2012plus-4.2.3\bin>mongo --port 27020
MongoDB shell version v4.2.3
connecting to: mongodb://127.0.0.1:27020/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("22cd31f3-1789-4d8c-85c1-859c38ffc1dd") }
MongoDB server version: 4.2.3
Server has startup warnings:
2020-03-26T04:39:42.467+0530 I CONTROL [initandlisten]
2020-03-26T04:39:42.467+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-03-26T04:39:42.469+0530 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
2020-03-26T04:39:42.470+0530 I CONTROL [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> show dbs
Employee  0.000GB
admin     0.000GB
config   0.000GB
local    0.000GB
> use Employee
switched to db Employee
> show collections
Employee_info
> db.Employee_info.find()
{ "_id" : ObjectId("5e7beb5a25a24b34be377faf"), "id" : 1, "name" : "yash" }
{ "_id" : ObjectId("5e7beb6d25a24b34be377fb0"), "id" : 2, "name" : "Tilak" }
{ "_id" : ObjectId("5e7bebd925a24b34be377fb1"), "id" : 3, "name" : "Mayank" }

```

- Shared server 2


```
> show dbs
Employee  0.000GB
admin     0.000GB
config    0.000GB
local     0.000GB
> use Employee
switched to db Employee
> show collections
Employee_info
> db.Employee_info.find()
{ "_id" : ObjectId("5e7beb5a25a24b34be377faf"), "id" : 1, "name" : "yash" }
{ "_id" : ObjectId("5e7beb6d25a24b34be377fb0"), "id" : 2, "name" : "Tilak" }
{ "_id" : ObjectId("5e7bebd925a24b34be377fb1"), "id" : 3, "name" : "Mayank" }
>
```