



MongoDB Enterprise 4.0.10

FUNDAMENTOS DE MONGODB

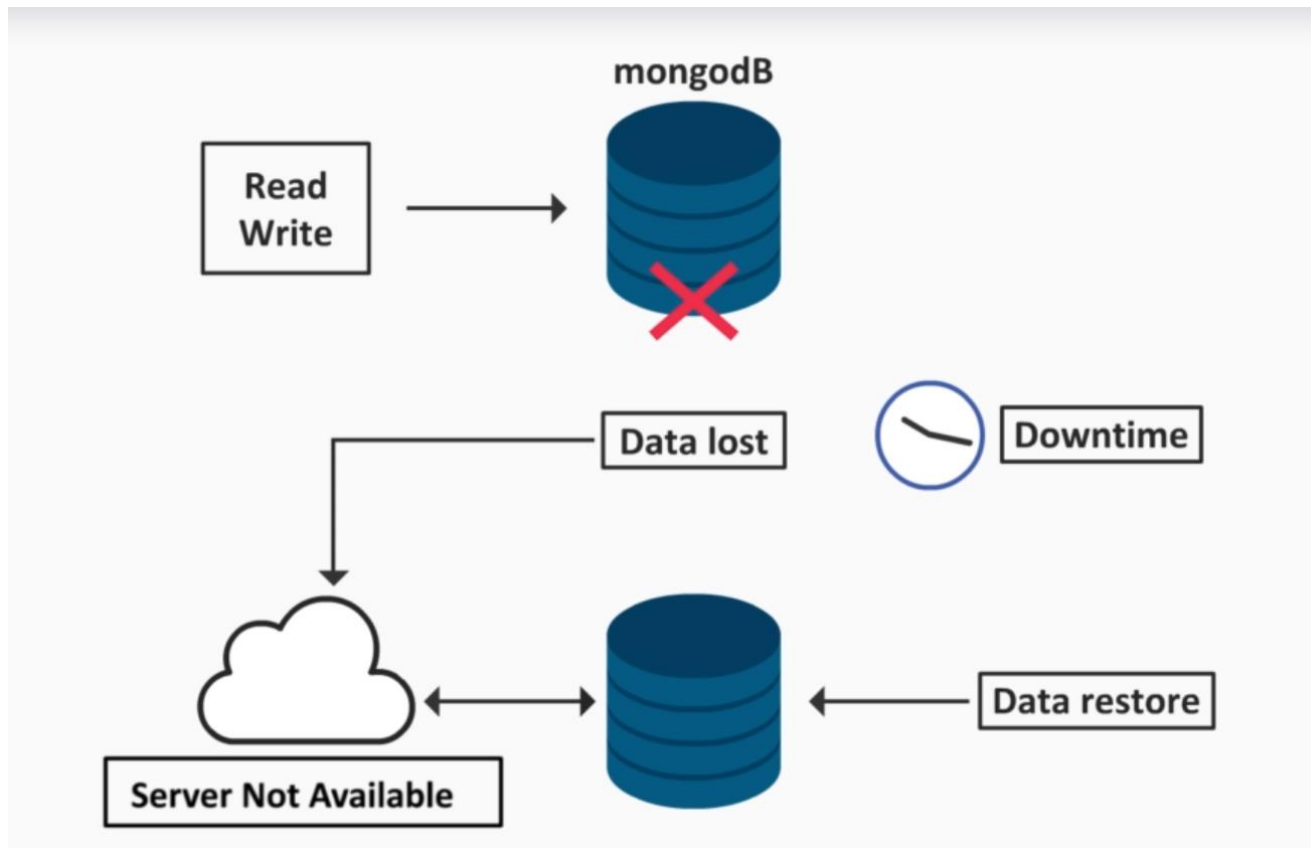
Instructor: Carlos Carreño
Email: ccarrenovi@gmail.com

Escalabilidad Horizontal

- Replicaset
- Sharding
- Balanceo
- El cluster
- Dimensionamiento de Tamaño de Colecciones de Mongodb (Storage)

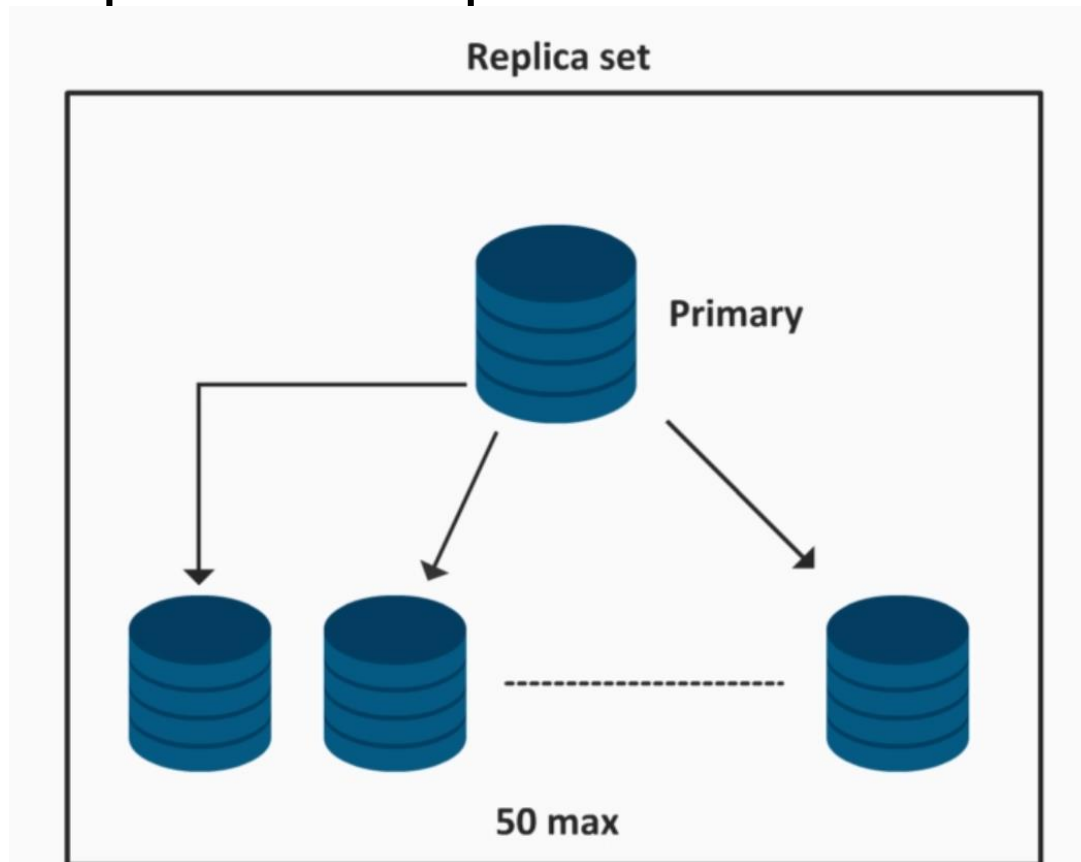
Introduction to MongoDB Replication

- Restore sin replicación



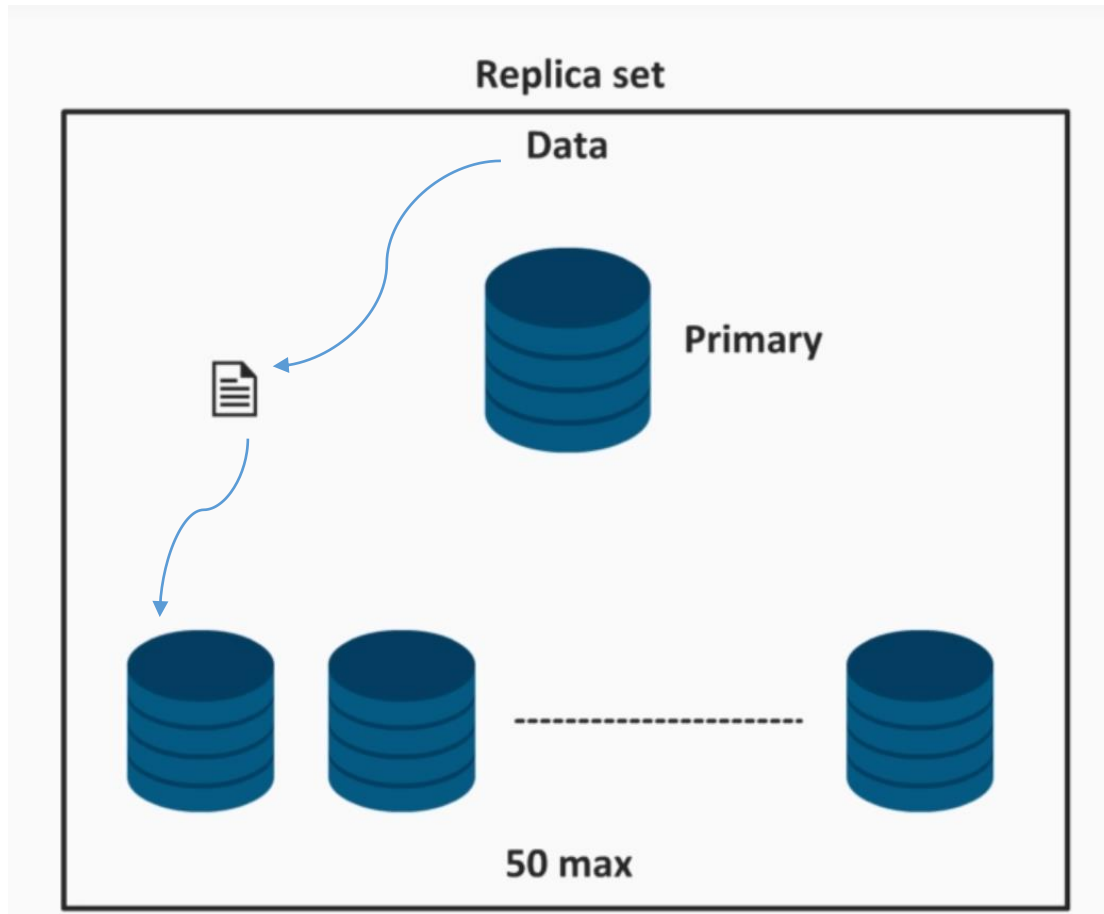
Replica Set

- Esquema de replica set



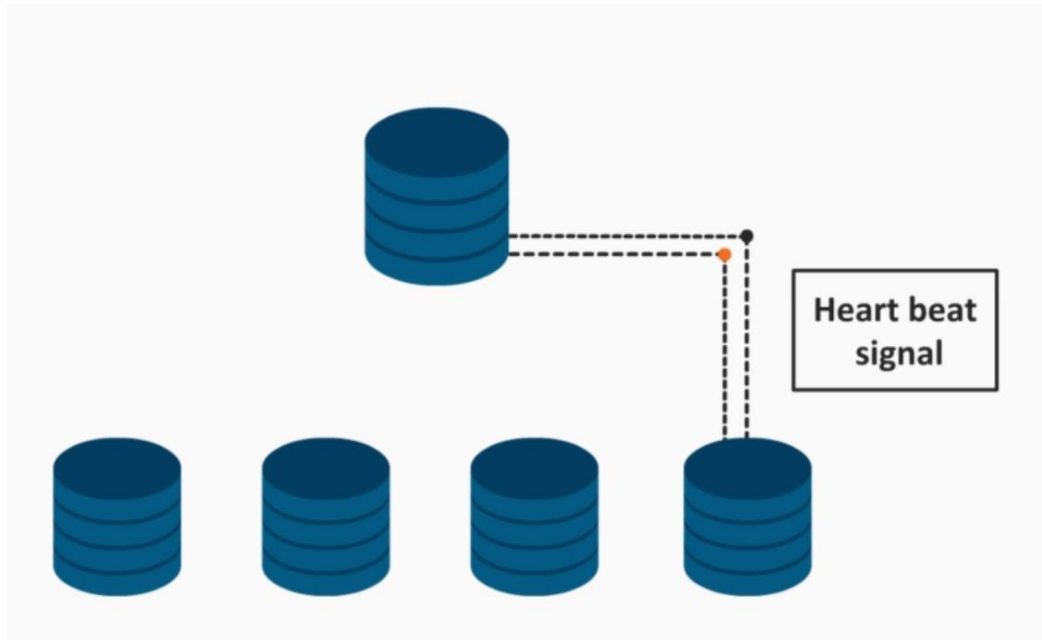
Data replication

- Data Replication from primary instance



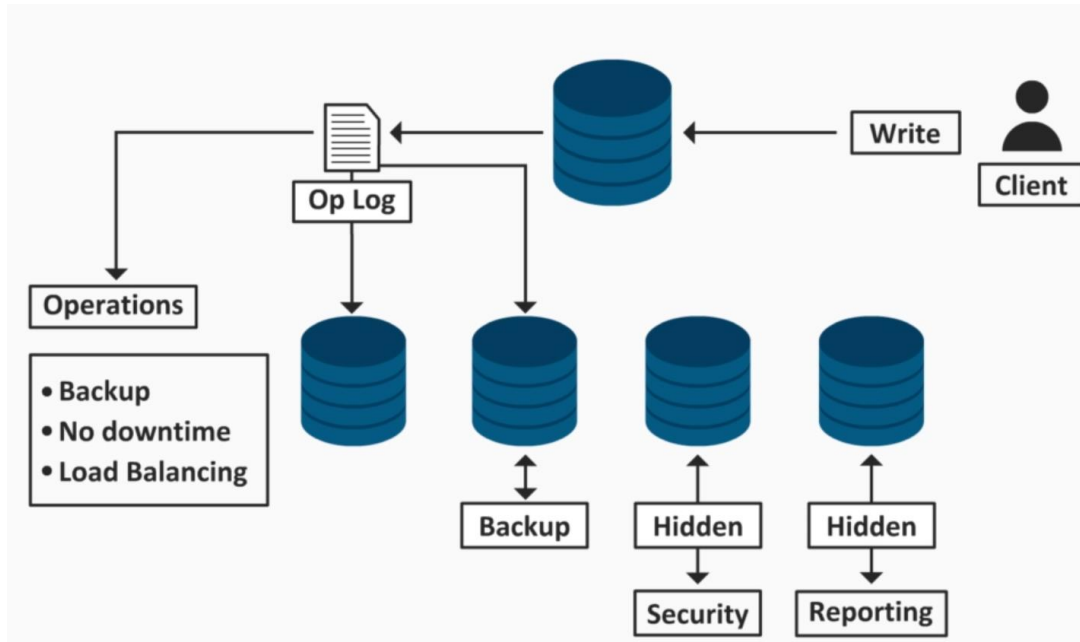
Continuous replication

- **Heart beat signal** : Si el servidor no responde por un largo tiempo se considera que está caído (downtime)
- Si un **primary** no está disponible (10 segundos), un **secondary** elegible realizará una elección para elegir la nueva primaria.



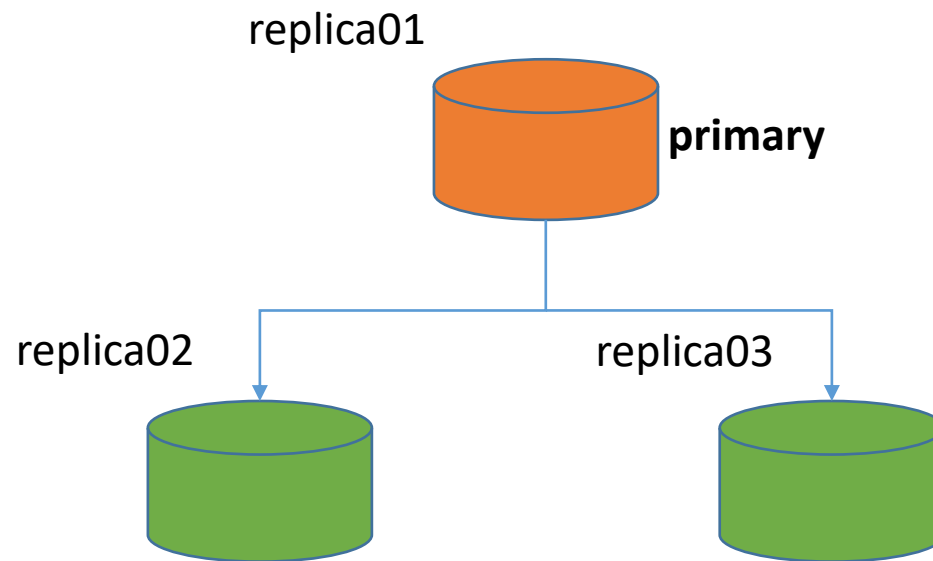
Operations in replica set

- Los servidores secundarios pueden estar especializados, si se cae el primary uno de ellos tomara el rol.
- El primario registra todos los cambios en sus conjuntos de datos en su registro de operaciones llamado **oplog**.



Setting Up Replica Set Servers

- Planificar el conjunto de instancias de mongodb que formaran parte del Replica Set



Replica Set Server

- Crea un directorio por cada servidor y ejecutar el comando mongod con las parámetros adecuados

```
$mkdir replica01
```

```
$chmod -R 777 replica01
```

```
$mongod -dbpath replica01 -port 27020 -logpath replica01/log -logappend  
-fork -bind_ip_all -replSet myreplicaset01
```

```
$ps ef | grep mongo
```

Configure Replica Set

- [admin@odiseo ~]\$ **mongo** --**port** 27020
- MongoDB Enterprise > **rs.initiate();**

```
MongoDB Enterprise > rs.initiate();
{
  "info2" : "no configuration specified. Using a default configuration for the set",
  "me" : "odiseo.example.com:27020",
  "ok" : 1,
  "$clusterTime" : {
    "clusterTime" : Timestamp(1572210317, 1),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  },
  "operationTime" : Timestamp(1572210317, 1)
}
MongoDB Enterprise myreplicaset01:OTHER>
MongoDB Enterprise myreplicaset01:PRIMARY> █
```

Nota: Presiona Enter después de ejecutar el comando en la consola CLI, deberás ver el cambio a **PRIMARY**

Check Configuration and status

- Chequeando la configuración

MongoDB Enterprise myreplicaset01:PRIMARY> **rs.conf()**;

MongoDB Enterprise myreplicaset01:PRIMARY> **rs.status()**;

- Agregando servidores secundarios al set

MongoDB Enterprise myreplicaset01:PRIMARY> **rs.add("odiseo.example.com:27021")**

```
MongoDB Enterprise myreplicaset01:PRIMARY> rs.add("odiseo.example.com:27021")
{
  "ok" : 1,
  "$clusterTime" : {
    "clusterTime" : Timestamp(1572210913, 1),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  },
  "operationTime" : Timestamp(1572210913, 1)
}
MongoDB Enterprise myreplicaset01:PRIMARY> rs.add("odiseo.example.com:27022")
{
  "ok" : 1,
  "$clusterTime" : {
    "clusterTime" : Timestamp(1572210921, 1),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  },
  "operationTime" : Timestamp(1572210921, 1)
}
MongoDB Enterprise myreplicaset01:PRIMARY> █
```

Setting Secondary Server

- [admin@odiseo ~]\$ **mongo** --port 27022
- MongoDB Enterprise myreplicaset01:SECONDARY> **rs.slaveOk(true);**

```
MongoDB Enterprise myreplicaset01:SECONDARY> show dbs
2019-10-28T00:02:44.675-0400 E QUERY [js] uncaught exception: Error: listDatabases failed:{"operationTime" : Timestamp(1572235164, 1),
"ok" : 0,
"errmsg" : "not master and slaveOk=false",
"code" : 13435,
"codeName" : "NotMasterNoSlaveOk",
"$clusterTime" : {
  "clusterTime" : Timestamp(1572235164, 1),
  "signature" : {
    "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
    "keyId" : NumberLong(0)
  }
}
} :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
Mongo.prototype.getDBs/<@src/mongo/shell/mongo.js:135:19
Mongo.prototype.getDBs@src/mongo/shell/mongo.js:87:12
shellHelper.show@src/mongo/shell/utils.js:906:13
shellHelper@src/mongo/shell/utils.js:790:15
@(shellhelp2):1:1
MongoDB Enterprise myreplicaset01:SECONDARY> rs.slaveOk(true);
MongoDB Enterprise myreplicaset01:SECONDARY> show dbs
admin      0.000GB
config     0.000GB
local      0.000GB
prodmongodb 0.000GB
```

Connection using Mongo to secondary server

- [admin@odiseo ~]\$ **mongo** --port 27022

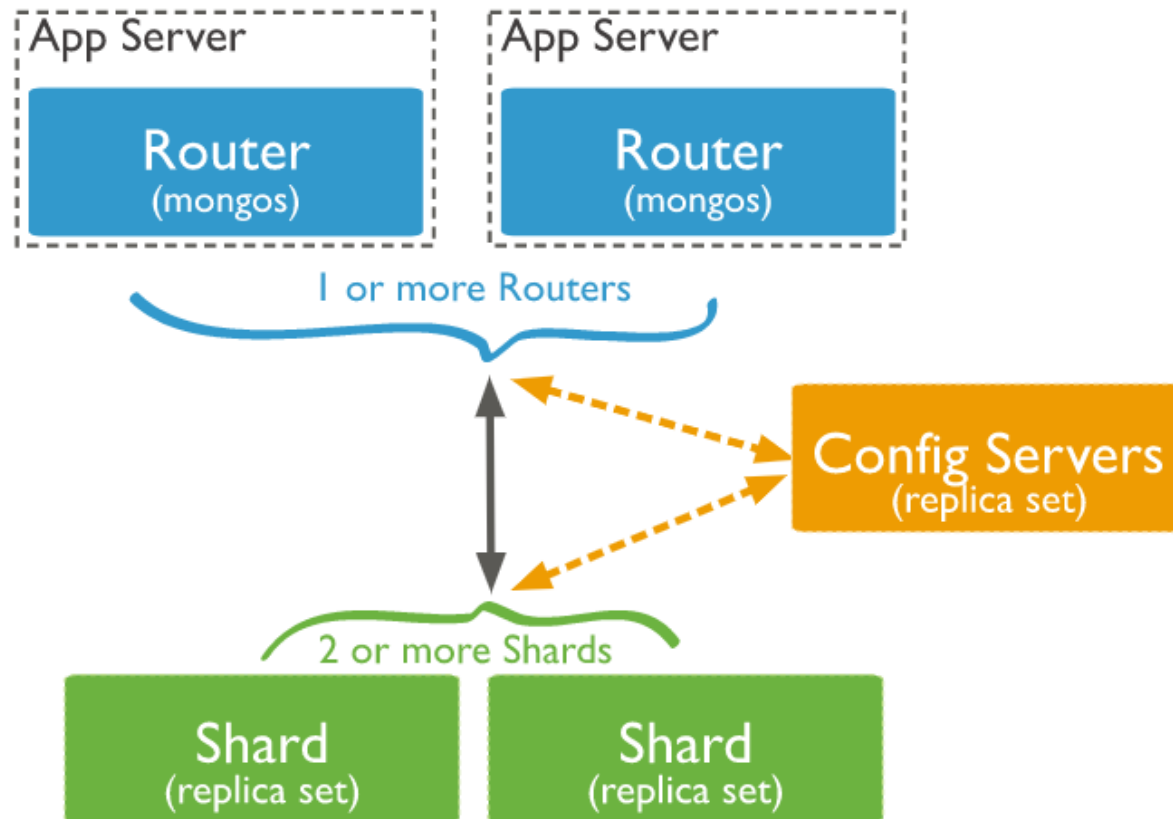
```
MongoDB Enterprise myreplicaset01:SECONDARY> use prodmongodb
switched to db prodmongodb
MongoDB Enterprise myreplicaset01:SECONDARY> db.employees.find();
{ "_id" : ObjectId("5db664ef3c49e8e78b25e093"), "name" : "Charles", "age" : 20, "salary" : 1000 }
MongoDB Enterprise myreplicaset01:SECONDARY> db.employees.insert({name:"Yenny",salary:1500});
WriteResult({
  "nInserted" : 0,
  "writeError" : {
    "code" : 189,
    "errmsg" : "Not primary while writing to prodmongodb.employees"
  }
})
MongoDB Enterprise myreplicaset01:SECONDARY> █
```

Sharding

- Sharding es un método para distribuir datos a través de múltiples máquinas.
- El sharding se utiliza con grandes cantidades de datos y alto throughput

Cluster

- Son tres los componentes principales:



Components of Sharded Cluster

- **Shard**: cada shard contiene un subconjunto de los datos fragmentados. Cada shard se puede implementar como un conjunto de réplicas.
- **mongos**: el mongos actúa como un enrutador de consultas, proporcionando una interfaz entre las aplicaciones del cliente y el clúster fragmentado.
- **servidores de configuración**: los servidores de configuración almacenan metadatos y ajustes de configuración para el clúster

Balanceo con Shard keys

- MongoDB usa la **shard key** para distribuir los documentos de la colección entre shards. La shard key **consiste en un campo o campos** que existen en cada documento en la colección de destino.
- Usted elige la shard key cuando fragmentas una colección. **La elección de la shard key no se puede cambiar después de fragmentar** la colección. Una colección fragmentada solo puede tener una shard key.
- Para fragmentar una colección no vacía, **la colección debe tener un índice** que comience con la shard key.

Chunks

- Las particiones MongoDB fragmentan los datos en chunks.
- Cada chunk tiene un rango superior e inferior inclusivo basado en la shard key.
- Cada chunk por defecto es de 64 MB

Balancer and Even Chunk Distribution

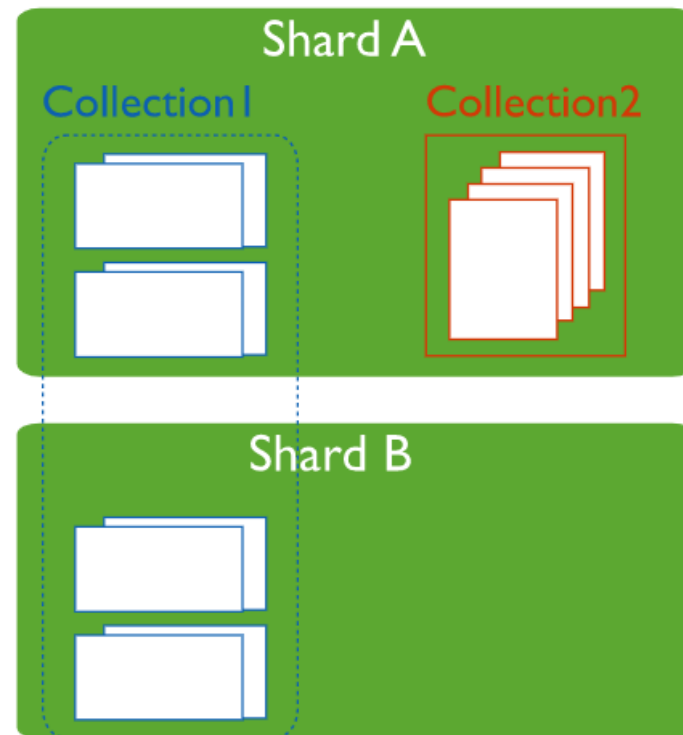
- En un intento por lograr una distribución uniforme de los chunks en todos los shard del clúster, se ejecuta un balanceador en segundo plano para migrar chunks en los shards del cluster.

Ventajas del Sharding

- Mejora el desempeño en operaciones de lectura y escritura
- Implementa una mejor gestión de la capacidad de almacenamiento
- Proporciona una solución de alta disponibilidad

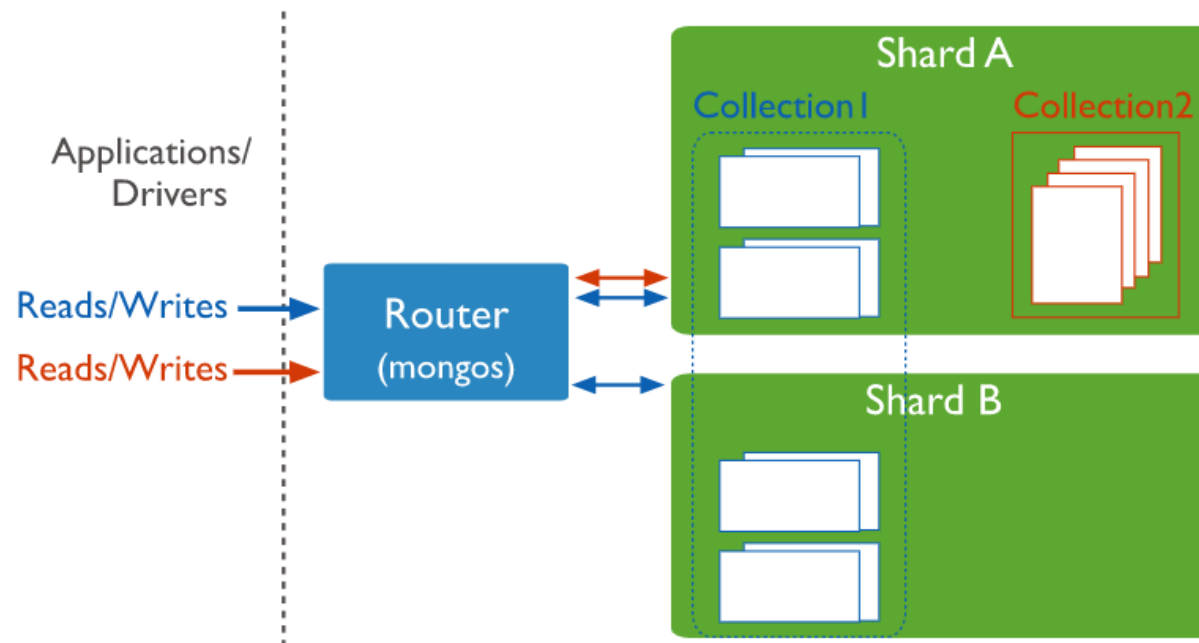
Sharded and Non- Sharded collection

- Una base de datos en mongo puede trabajar en modo mixto puede tener algunas colecciones en sharded y otras en modo no sharded



Conexión al sharding cluster

- El cliente debería conectarse a un mongos router para poder interactuar con el sharding cluster, tanto con colecciones sharded como con no sharded.

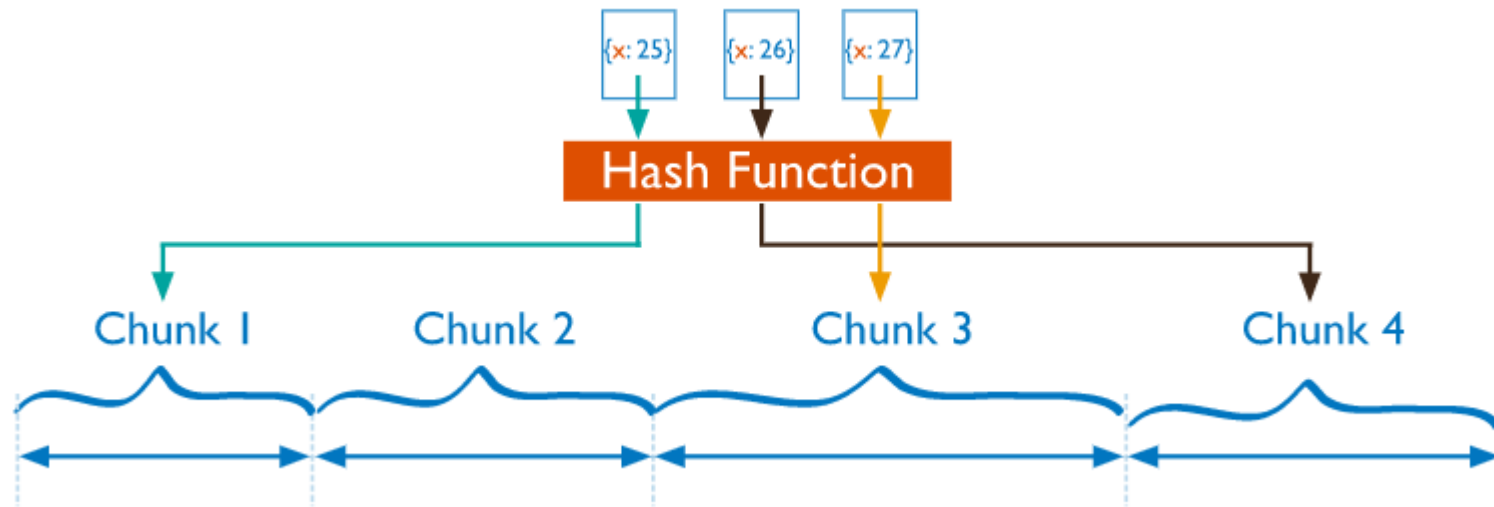


Sharding Strategy

- Mongo soporta dos estrategias para distribuir los datos en el cluster:
 - Hashed Sharding
 - Ranged sharding

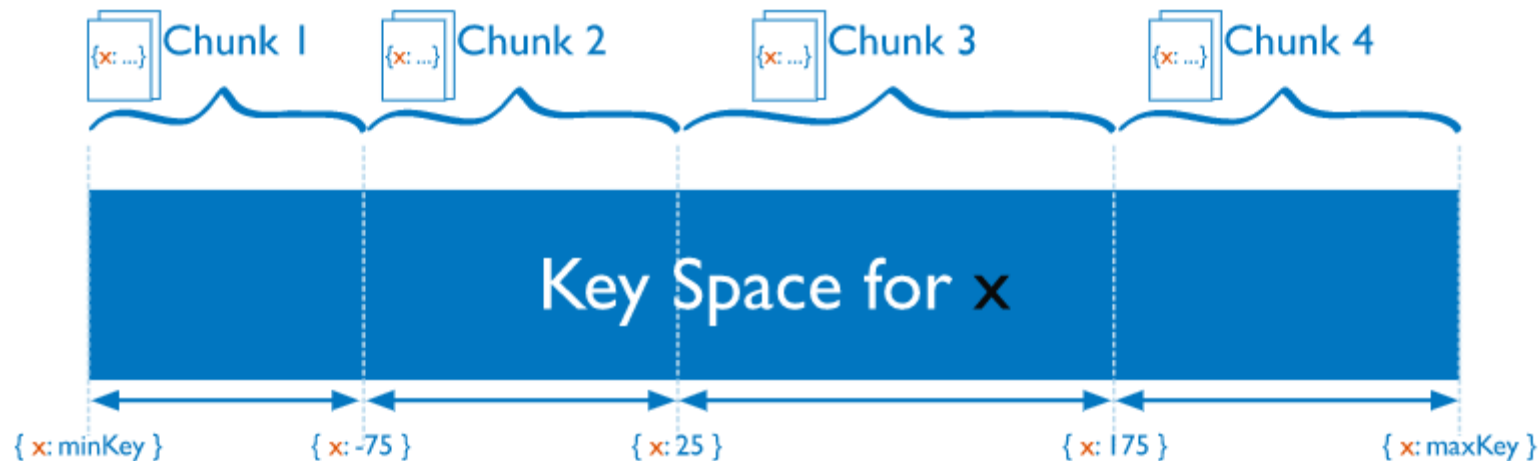
Hashed Sharding

- Hashed Sharding implica calcular un hash del valor del campo shard key. A cada chunk se le asigna un rango basado en los valores del shard key hash.



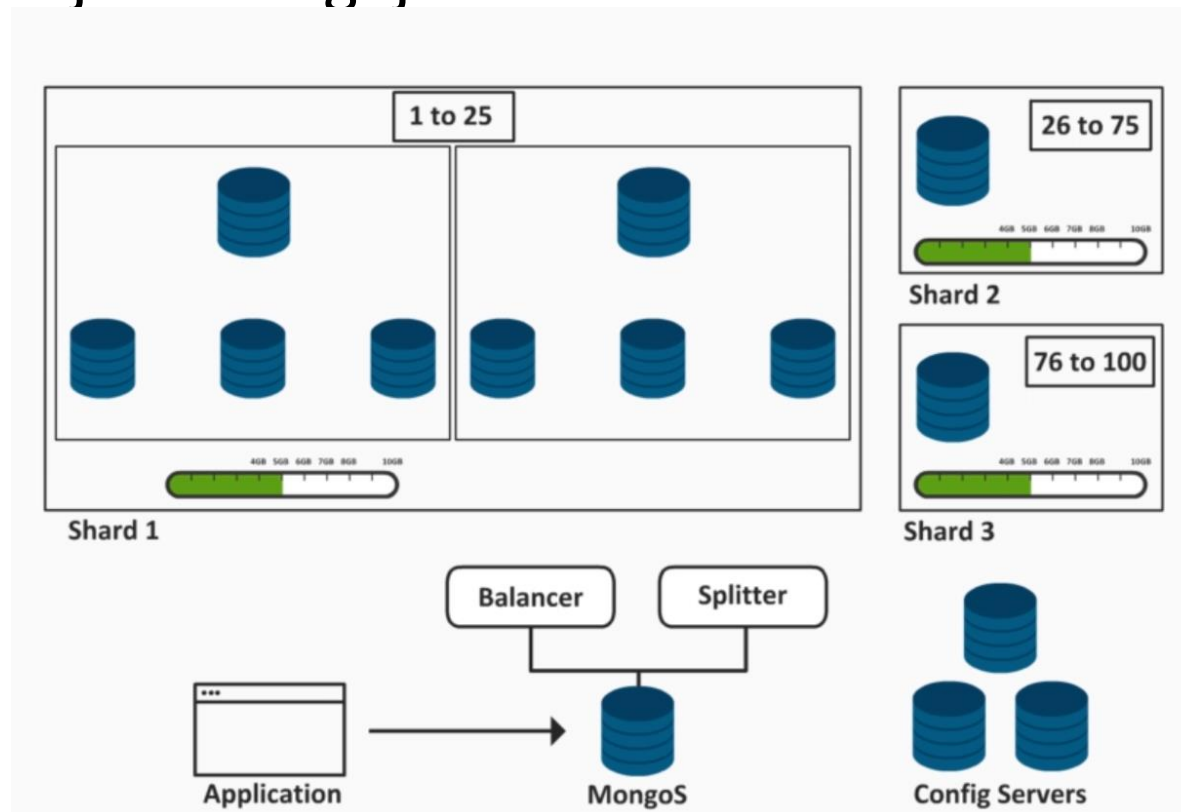
Ranged sharding

- Ranged sharding divide los datos en rangos basados en el valor del shard key. Cada chunk es asignado a un rango basado en el valor del shard key.



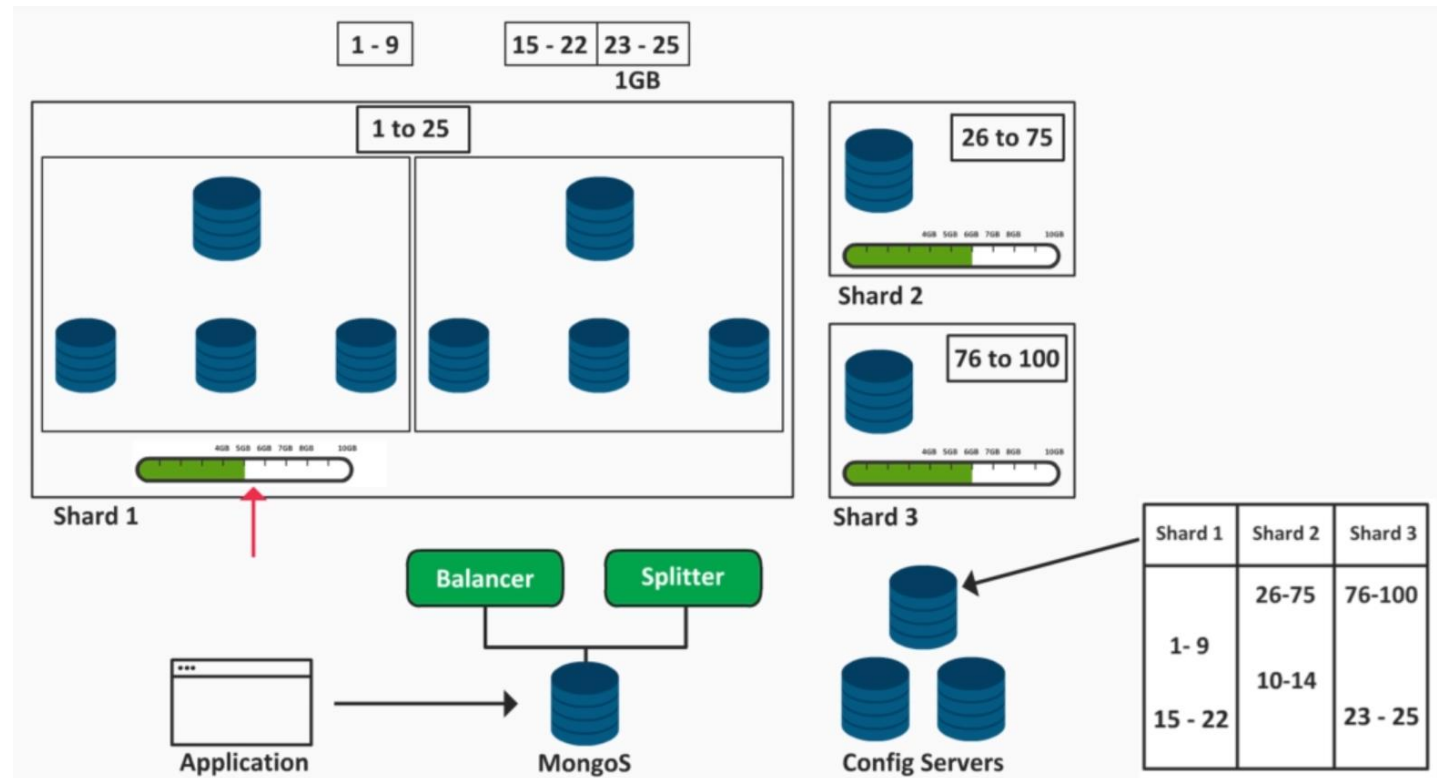
Sharding Workflow

- Flujo de trabajo de una solución de sharding



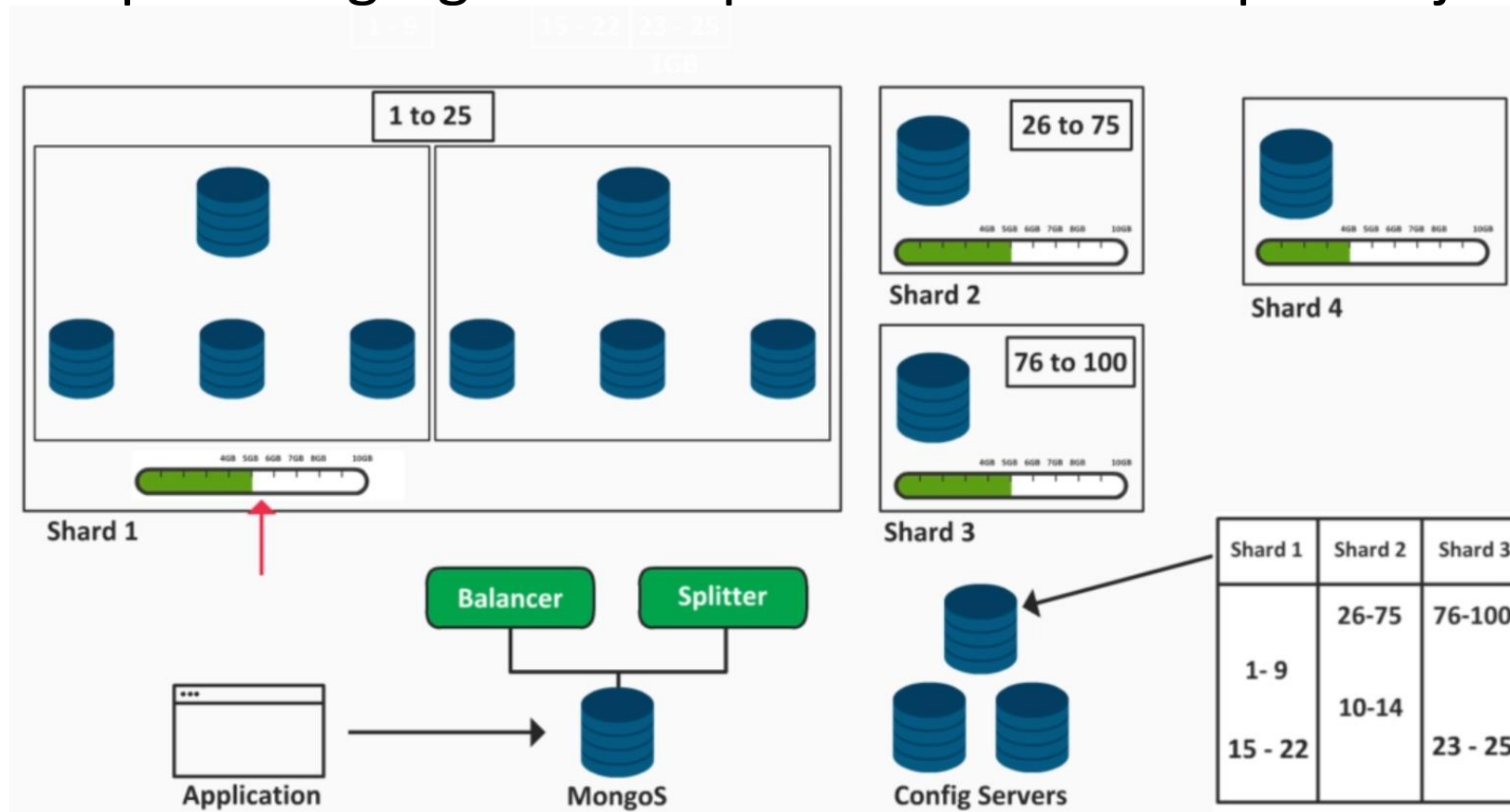
Sharding Workflow & Balancer

- Si algún shard está con sobre carga esta se puede balancear entre los Sharp disponibles



Sharding Workflow & Add replica set

- Se puede agregar una replica set adicional par mejorar el desempeño



Dimensionamiento de Tamaño de Colecciones de MongoDB (Storage)

- En un clúster de producción, asegúrese de que los datos sean redundantes y que sus sistemas estén altamente disponibles. Considere lo siguiente para una implementación de un shard clúster:
 - Implementar servidores de configuración como un conjunto de réplicas de 3 miembros
 - Implemente cada shard como un conjunto de réplicas de 3 miembros
 - Implemente uno o más enrutadores mongos

Referencias

- <https://devops.ionos.com/tutorials/configure-mongodb-replica-set/>
- <https://medium.com/codeops/how-to-setup-a-mongodb-replica-set-918f21da50ed>