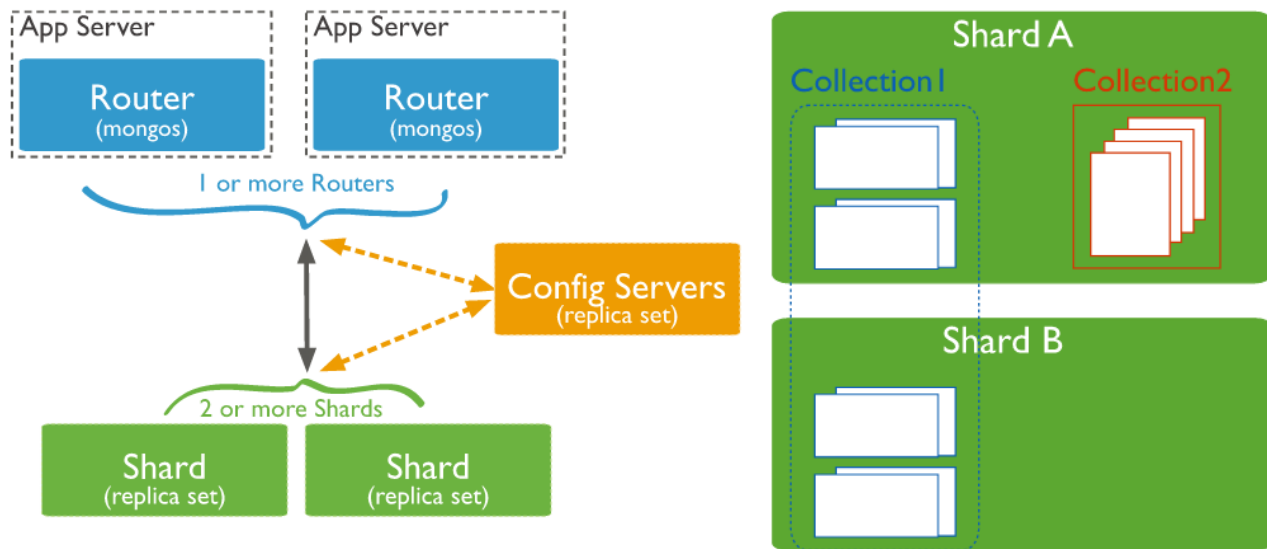


Sharding

- Es un método para distribuir datos en múltiples máquinas.
- Existen dos maneras de enfrentar crecimiento de un sistema:
 - Verticalmente: implica aumentar las capacidades de un solo servidor.
 - Horizontalmente: implica dividir la información en porciones más pequeñas y distribuir la carga en múltiples equipos.

MongoDB

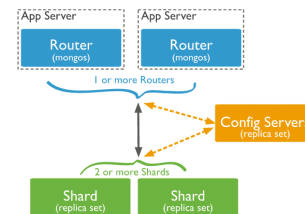
Sharding



MongoDB

Sharding

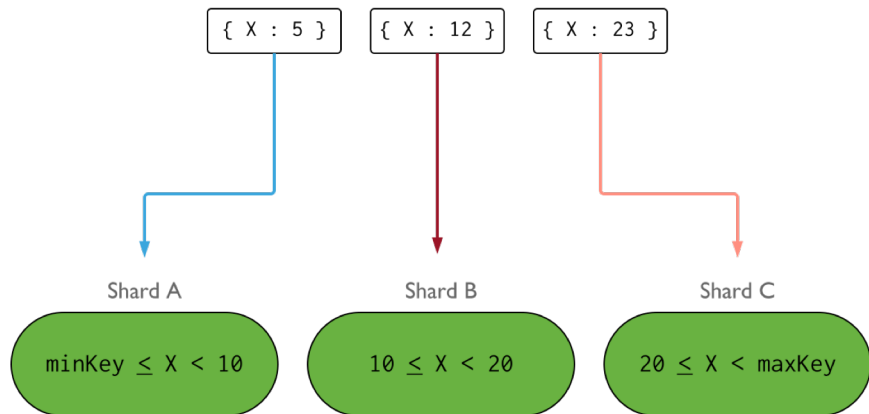
- Shard
 - Cada “shard” contiene un subconjunto de la información partida.
 - A cada parte se la debería distribuir como “replica set”.
- Mongos: actúan como los puntos de entrada de los clientes para rutear las consultas.
- Config servers: almacenan la configuración y metadata relacionada con el cluster.



MongoDB

Shards key

- Mongo realiza la partición a nivel de colecciones utilizando una “shard key”.
- Está compuesta de uno o más campos inmutables.
- Solamente puede existir una shard key por colección y no se puede cambiar.



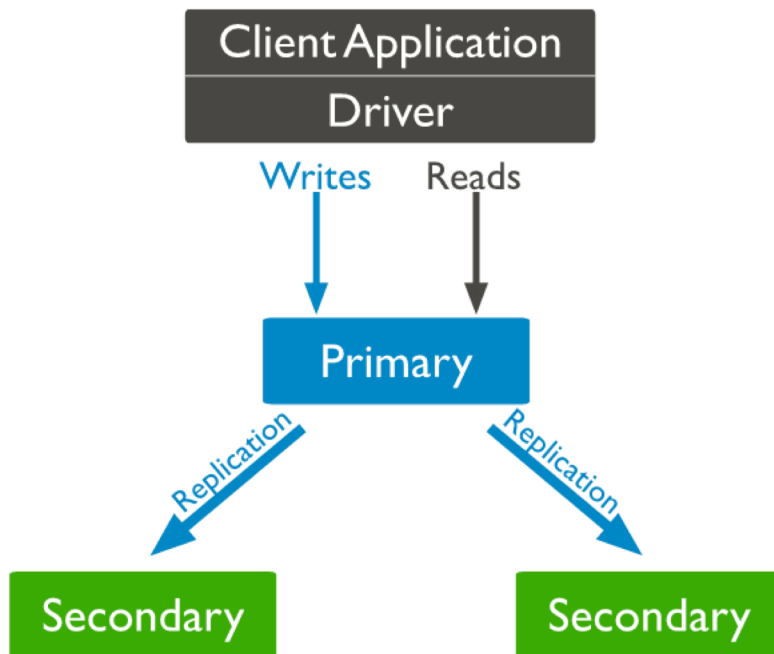
MongoDB

Ventajas y beneficios

- Lecturas / escrituras
 - La carga se distribuye horizontalmente en todo el cluster.
- Capacidad de almacenamiento
 - Si el espacio requerido aumenta, simplemente se agregan nuevos nodos al cluster.
- Alta disponibilidad
 - Aún cuando algún nodo no funcione, se puede seguir resolviendo parcialmente el pedido.

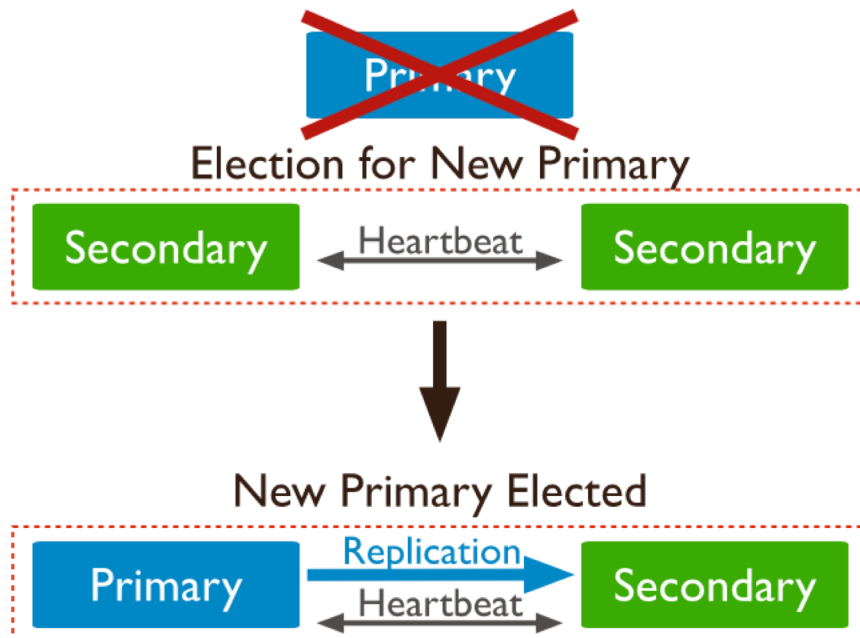
MongoDB

Replication



MongoDB

Automatic failover



MongoDB

Trabajo con MongoDB desde Java

Dependencias de Maven

```
1 <dependency>
2   <groupId>org.mongodb</groupId>
3   <artifactId>mongo-java-driver</artifactId>
4   <version>3.4.1</version>
5 </dependency>
```

Conectarse a un servidor

```
1 MongoClient mongoClient = new MongoClient("localhost", 27017);
```

Acceder a una base de datos

```
1 DB database = mongoClient.getDB("myMongoDb");
```

Obtener todas las bases de datos

```
1 mongoClient.getDatabaseNames().forEach(System.out::println);
```

Crear una nueva colección

```
1 database.createCollection("customers", null);
```


MongoDB

Trabajo con MongoDB desde Java

Guardar

```
1 DBCollection collection = database.getCollection("customers");
2 BasicDBObject document = new BasicDBObject();
3 document.put("name", "Shubham");
4 document.put("company", "Bieldung");
5 collection.insert(document);
```

Realizar una consulta

```
1 BasicDBObject searchQuery = new BasicDBObject();
2 searchQuery.put("name", "John");
3 DBCursor cursor = collection.find(searchQuery);
4
5 while (cursor.hasNext()) {
6     System.out.println(cursor.next());
7 }
```