01 Intro

Conceptos

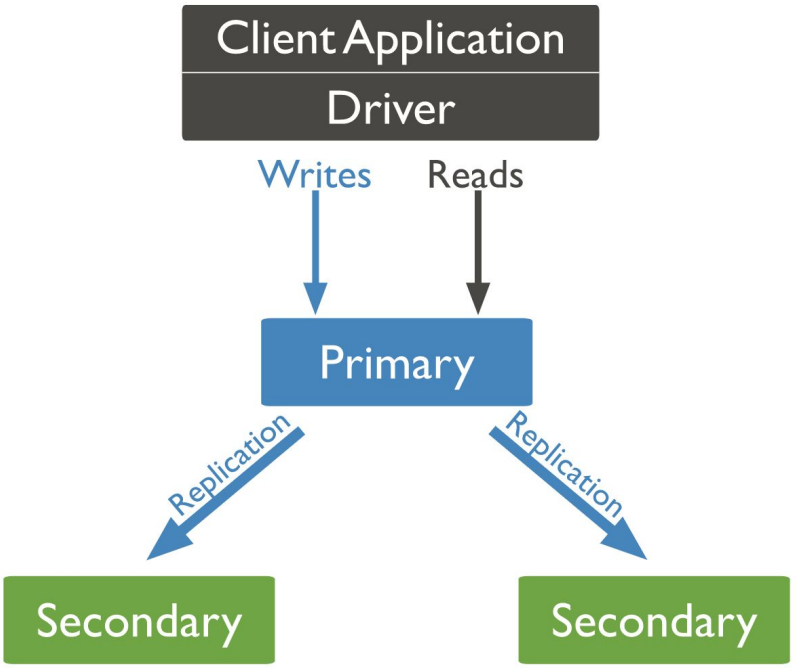
Un replica set o cluster en MongoDB es un grupo de procesos o servidores mongod que mantienen el mismo set de datos proporcionando redundancia y alta disponibilidad, y son la base de todos los despliegues en producción.

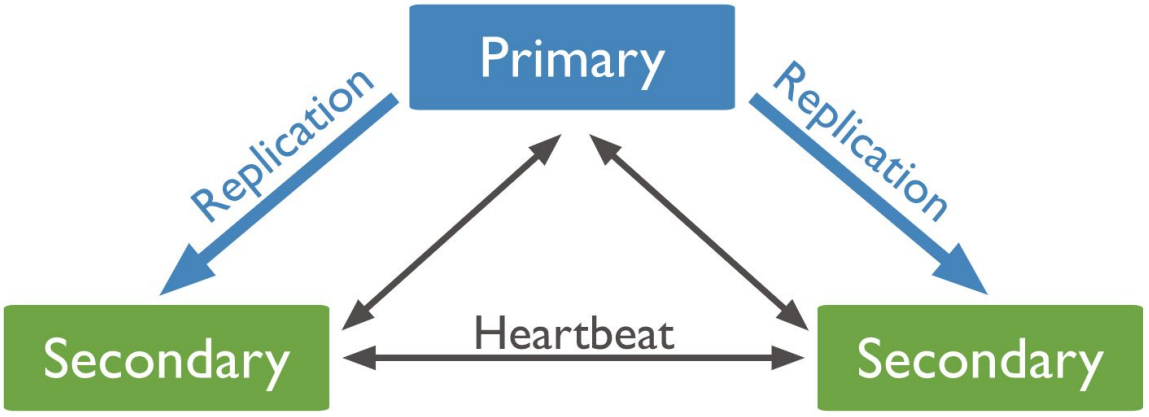
La replicación proporciona redundancia y incrementa la disponibilidad de datos mediante múltiples copias de los mismos en diferentes servidores de bases de datos, proveyendo un nivel de tolerancia a fallos por la pérdida de un servidor.

En algunos casos la replicación puede proporcionar un incremento de la capacidad de lectura al enviar las operaciones a diferentes servidores. Al mantener copias de datos en diferentes data centers se puede ampliar la disponibilidad local para aplicaciones distribuidos.

También se pueden crear copias adicionales con propósitos dedicados como recuperación de desastres, reporting o backup.

Un replica set contiene varios nodos de producción y opcionalmente un nodo árbitro. De los nodos de producción, solo uno será denominado primario, mientras que los demás serán secundarios.



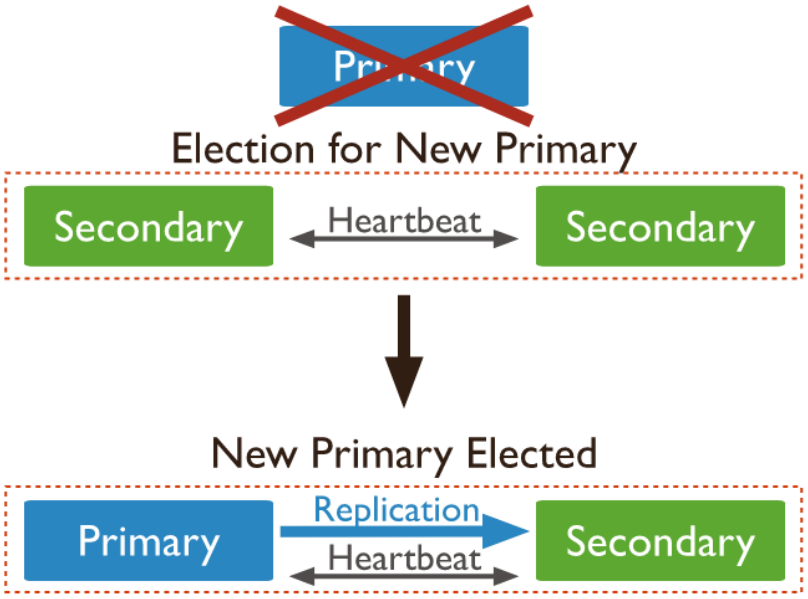


Para comprobar la disponibilidad de cada miembro del cluster se comunican entre ellos con un ping cada 2s (por defecto).

Replicación asíncrona

Los nodos secundarios replican el oplog (colección que almacena las operaciones de escritura) y aplican esas operaciones sobre su set de datos de manera asíncrona. Esta replicación de datos permite continuar funcionando al cluster a pesar de la falla de uno o más miembros.

Conmutación (failover) automática



1.- Cuando el primario no se comunica con los otros miembros por más de 10 segundos (valor por defecto de electionTimeoutMillis) uno de los secundarios elegible llama a las elecciones nominándose a sí mismo como el nuevo primario.

2.- Mientras se producen las elecciones se interrumpen todas las operaciones de escritura.

3.- Las operaciones de lectura se podrían mantener durante las elecciones, pero para ello uno de los secundarios debe estar configurado para aceptar operaciones de lectura.

4.- El tiempo medio de las elecciones no suele exceder de 12 segundos, pero se podría rebajar configurando el valor de interrupción de comunicaciones por debajo de 10 segundos. En este caso el problema es que pequeños fallos de comunicación desencadenen elecciones innecesarias.

5.- Una vez resueltas las elecciones, el nuevo primario establece las operaciones de escritura y lectura y la replicación a los miembros vivos.

6.- Una vez que se recupere el servidor caído se puede incorporar al cluster como secundario para recuperar la información generada desde su falla y posteriormente volver a ser primario o no en función de la configuración del cluster.

7.- La lógica de las aplicaciones deberá incorporar tolerancia a este automatic failover así como las elecciones. Algunos driver proporcionan esta tolerancia de manera nativa y configurable.