

Sharding

Sharding (샤딩) 이란 여러 대의 머신으로 data access 를 분배하는 방법을 말한다.

기능적인 관점으로 보았을 때 여러 호스트에 데이터를 배포하는 것이고 , 여러 MongoDB 인스턴스에 걸쳐 큰 데이터 세트를 작은 데이터 세트로 분할하는 것이 샤딩처리이다

처리량이 많은 App 이나 매우 큰 데이터를 처리할 때 하드웨어의 성능이 매우 중요해집니다. 또한 높은 쿼리 속도는 머신에 부담을 주어 전체적인 성능을 저하시킨다.

이 문제를 처리하기 위한 방법은 크게 두가지가 있다.

1 수직 확장

: 단일 서버의 하드웨어 기능을 향상시키는 방법이다. 가장 간단하면서 직관적인 방법이다.

하드웨어의 성능을 업그레이드 한다는 것은 한계점에 도달하면 기술적 한계와 비용 제약이 있을 수 있다.

2 수평 확장

: 부하를 분산하는 것이다. 데이터 셋을 여러 서버 (노드) 에 나누고 로드를 분산한다.

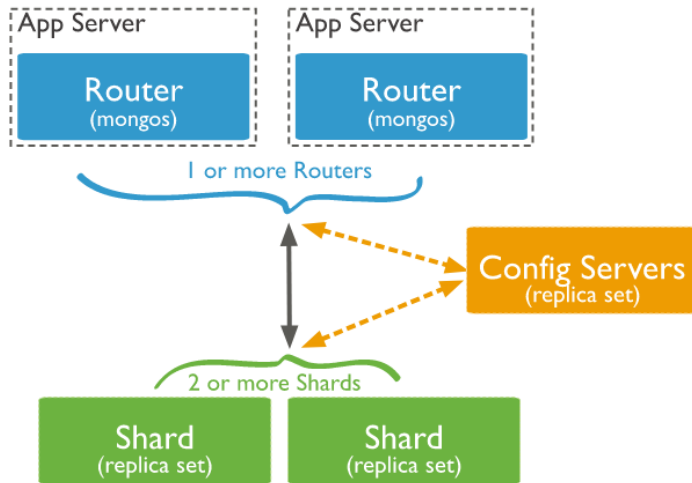
무작정 수평 확장을 하다보면 아키텍처의 복잡성을 굉장히 증가시킬 수 있다.

MongoDB 샤딩 구성 요소

: 몽고 DB 는 샤딩 구성을 하는데 대표적으로 3 가지 요소가 필요하다.

1. Shard - 가장 기본이 되는 인스턴스로 , 데이터의 하위 집합을 보유하고 있는 단일 mongoDB 인스턴스 라고 볼 수 있다.
2. Mongos - 다수 구성된 샤드의 인터페이스 역할이다. 클라이언트의 요청을 올바른 샤드로 라우팅하는 것이 주요 기능이다.

3. Config servers – 전체 클러스터의 메타 데이터, 구성 설정 등을 저장하는 서버이다.



Shard Keys

MongoDB 컬렉션을 샤딩 할 때 초기 단계 중 하나로 샤드 키가 생성된다 "샤드 키"는 MongoDB 컬렉션의 도큐먼트를 전체 샤드에 배포하는 데 사용되고 키는 모든 도큐먼트에서 단일 필드 또는 여러 필드로 구성된다.

샤딩 된 키는 샤딩 후에는 변경이 불가. 분할된 컬렉션에는 단일 분할 키만 포함됩니다. 채워진 컬렉션을 분할할 때 컬렉션에는 분할 키로 시작하는 인덱스가 있어야 한다. 적절한 인덱스가 없는 빈 컬렉션일 때 MongoDB 는 지정된 샤드 키에 대한 인덱스를 생성.

샤드 키는 클러스터의 성능에 직접 영향을 미칠 수 있기 때문에 클러스터와 관련된 응용 프로그램에서 병목 현상이 발생할 수 있다. 이를 완화하려면 컬렉션을 샤딩하기 전에 다음을 기반으로 샤드 키를 생성해야 한다.

- 데이터 세트의 스키마
- 데이터 세트를 쿼리 하는 방법

Chunks

청크는 공유 데이터의 하위 단위. MongoDB 는 샤딩 된 데이터를 공유 클러스터의 샤드에 분산된 청크로 분리한다. 각 클러스터에 특정한 밸런서가 청크 분배를 처리한다. 밸런서는 백그라운드 작업으로 실행되며 모든 샤드에서 청크의 균형을 맞추기 위해 필요에 따라 청크를 배포.