

Global DR

* DR 이란?

: disaster recovery , 말 그대로 재해 복구 이다. 물론 데이터베이스 분야의 DR 이란 실시간 복제 (replication) 등을 통해서 복구를 진행하고 있다.

MongoDB 에서는 기본적으로 replica set 을 제공하여 타 DBMS 보다 손쉬운 DR 가능성을 보여준다.

다음은 MongoDB sharded cluster 의 trouble shooting 과정을 나타내고 이를 DR 로 표현할 수 있을 것 같다.

1. Shard 의 Replica set 중 하나가 작동하지 않는다면 ?

- 특정 상황에서 Replica set 을 구성하는 한 멤버가 작동하지 않는 문제가 생길 수 있다. 이 때를 위해 Replication 이 존재한다고도 할 수 있다.

만약 (1) Primary node 가 작동하지 않는다면, Replica set 에서는 곧 (heartbeat 의 주기가 2 초기 때문에 그 안팎으로 매우 짧은 시간이라고 할 수 있다.) 새로운 Primary node 를 새로 선출 한다.

(2) Secondary node 가 작동하지 않는다면 나머지 2 개 (혹은 그이상) 의 node 와 연결을 끊는다. 3 개의 멤버를 가진 replica set 에서는 한 개의 node 가 연결되지 않는다고 해도 타 노드로의 복제로 가용성이 보장된다.

2. 그럼 만약 한 Shard 의 모든 노드가 작동하지 않는다면 ?

- 매우 암울한 상황인 것은 맞을 것이다. Sharded Cluster 는 mongod 와 mongos 노드 (인스턴스와 같은 말이다) 가 각 replica set (config replica set / shard replica set) 을 모니터링하는데

Shard 내의 모든 멤버가 작동하지 않을 수 있는데 다른 모든 샤드에 대한 데이터는 계속 사용할 수 있으며 다른 샤드에 데이터를 읽고 쓸 수 있다. 그러나 application 은 부분적인 결과를 처리할 수 있어야 하며, 중단 원인을 조사하고 가능한 한 빨리 샤드를 복구해야 한다

이때 유용한것이 백업 , 동기화를 통한 DR 이다.

3. Config Server 의 Replica set 의 멤버가 작동하지 않는다면 ?

- 사실 1 번과 과정은 동일하게 진행된다. 하지만 만약 primary 와의 연결이 끊기고 새로운 primary 선출에 실패하면 config server 는 read-only 가 되게 된다.

정확히는 read 와 write 가 모두 가능하지만 chunk migration 이나 chunk split 등 balancing 은 primary 가 작동하기 전까지 사용할 수 없다.

공식 문서에서는 2 개 이상의 데이터 센터를 사용할 것을 권장한다.

DR 과정에서 생기는 데이터의 유실 등을 가장 효과적으로 방지할 수 있는 방법은 바로 백업 (Back Up)이다.

모든 백업 시스템은 과거 시점의 시스템을 스냅샷 (Snap-shot)을 캡처 합니다.

백업 전략에서는 RTO(Recovery Time Object) 와 RPO(Recovery Point Object) 가 있는데

RTO : 복구 시간 목표 (복구에 얼마나 빨리 걸리는지를 나타낸다.)

RPO : 복구 시점 목표 (사고 발생 시 손실될 가능성이 있는 데이터의 양을 나타낸다.)

RTO 와 RPO 는 각 애플리케이션 별로 평가되어야한다.

RTO 및 RPO 에 대한 요구 사항은 백업 유지보수의 경제적 및 성능 비용을 증가시킨다.

예를 들어, 화재 발생 시 최대 1 년 분량의 데이터를 손실하는 데 만족하면 1 년에 한 번 스냅샷을 생성하여 원격 위치의 연결이 끊긴 물리적 테이프에 백업을 저장할 수 있다.

이것은 비교적 저렴한 해결책이 된다.. 그러나 재해가 발생하면 테이프를 데이터 센터에 물리적으로 보내야 하고 백업은 1 년이 지났을 수 있으며, 백업 도착까지 며칠이 걸릴 수 있으며, 운영 상태로 복원하는 데 더 오래 걸릴 수 있는 것이다.

한편, 1 분 또는 2 분 이상의 데이터 손실이나 몇 분 이상의 다운타임을 초래하지 않으려면 시점 복구 기능을 갖춘 지속적인 백업 솔루션이 필요하다.

MongoDB 에서 사용하는 백업 전략은 다음과 같다.

1. MongoDump : MongoDB Database 의 붙어있는 utility 이다
2. File System snapshot (예를 들면 Linux LVM 이나 AWS EBS)
3. MongoDB Management Service (MMS)

참고 :

<https://www.mongodb.com/backup-and-its-role-disaster-recovery>

MongoDB Backup Management Tips for Sharded Clusters

	mongodump	Filesystem Snapshots	MMS
RPO	Snapshot 시점에 정해지는 한계가 있음	Snapshot 시점에 정해지는 한계가 있음 하지만 Snapshot 이 적은 overhead 를 가졌기 때문에 더 짧은 주기로 캡처가 가능함	Replicat set 에서의 Point-in-time 가능 Snapshot 시점을 전체 sharded cluster 로 가능
RTO	mongostore 을 실행해야하는데 mongostore 의 레이턴시는 dump 의 위치 (dumps location) 와 복구중인 항목의 세부 내용에 달렸다.	스냅샷을 Production 서버에 가져와서 현재 실행중인 파일 시스템으로 확장하는 지연시간에 달렸습니다.	네트워크를 통해 백업 스냅샷을 전송하는 데 걸리는 시간에 따라 달라집니다. 저장된 스냅샷보다 사용자 지정 시점 스냅샷을 구성하고 복원하는 경우 복원 시간이 길어집니다.

Isolation : 운영 중단이 백업 중단과 관련되지 않도록 백업은 운영 시스템과 분리되어야 합니다	백업 스냅샷이 프로덕션에서 유지되는 정도에 따라 달라집니다.	백업 스냅샷이 프로덕션에서 유지되는 정도에 따라 달라집니다.	백업은 AWS 외부에 있는 MongoDB 의 중복되고 안전한 데이터 센터에 저장됩니다.
Perfomance Impact	Online-mode (실시간) 에서 실행할 시에는 매우 위험합니다.	구현에 따라 상당히 영향이 없을 수 있습니다.	초기 동기화 시 상당한 로드가 추가될 수 있습니다. 그 후, 영향은 매우 낮습니다.
Restore Process	mongorestore 를 사용하여 BSON 파일의 압축을 풉니다.	일반적으로 스냅샷을 새 파일 시스템으로 확장합니다.	MMS 에서 스냅샷을 전송한 다음 확장합니다. MMS 는 BSON 덤프가 아닌 실제 데이터베이스 파일 을 복원합니다.
sharding	전체 클러스터에 걸쳐 스크립팅하고 스냅샷을 동기화해야 합니다.	전체 클러스터에 걸쳐 스크립팅하고 스냅샷을 동기화해야 합니다.	샤드 클러스터의 일관된 스냅샷을 지원합니다.
Deployment Complexity	소규모 시스템의 경우에는 낮은 편이지만 대규모 cluster 에 경우에는 매우 높을 수 있다.	소규모 시스템의 경우에는 낮은 편이지만 대규모 cluster 에 경우에는 중간 정도의 complexity 를 가집니다.	낮음.

Flexibility	Script 를 사용자가 직접 custom 해야합니다.	특정 데이터베이스를 대상으로 지정할 수 없습니다.	중요하지 않은 수집을 배제하고, 사용자 지정 스냅샷 예약 및 보존 정책을 설정할 수 있는 유연성을 제공합니다.
Cost	Free	사용하는 tool 에 따라 유/무료	유료