

AWS S3 로 MongoDB 백업 자동화

참고 : <https://velog.io/@seunghoking/Aws-S3%EB%A1%9C-Mongo-DB-%EB%B0%B1%EC%97%85-%EB%B0%8F-%EC%9E%90%EB%8F%99%ED%99%94-%EC%B4%9D%EC%A0%95%EB%A6%AC>

구성 :

AWS EC2 Seoul region

3 shard replica set

mongodump 를 사용하면 mongo 서버를 종료하지 않고 백업 할 수 있다.

정해진 시간에 백업 될 수 있도록 자동화해보려고한다

정해진 시간에 주기적으로 dump 를 하기 위해서는 예약 스케줄러 Cron 으로 셸 스크립트를 돌리면 된다.

* Cron : 작업 예약 스케줄러

* 셸 스크립트 (.sh) : shell 의 명령어를 파일로 저장해서 파일을 실행하면 명령어 실행

crontab → cron 작업을 설정하는 파일 , 특정 시간을 정하고 그 시간에 맞춰 특정 sh 파일을 실행시키는 역할을 한다.

```
mongos> for (i=0;i<1000000;i++) db.item.insertOne({"item":i})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("635b81cdf448a1f96d00046c")
}
```

데이터 100 만개 추가

```
Shard shard1 at shard1/172.32.130.55:27017,172.32.176.183:27017,172.32.82.245:27017
data : 10.4MiB docs : 303056 chunks : 2
estimated data per chunk : 5.2MiB
estimated docs per chunk : 151528

Shard shard2 at shard2/172.32.21.124:27017,172.32.212.60:27017,172.32.30.9:27017
data : 15.59MiB docs : 454364 chunks : 3
estimated data per chunk : 5.19MiB
estimated docs per chunk : 151454

Shard shard3 at shard3/172.32.162.46:27017,172.32.42.154:27017,172.32.51.237:27017
data : 15.53MiB docs : 452585 chunks : 3
estimated data per chunk : 5.17MiB
estimated docs per chunk : 150861

Totals
data : 41.54MiB docs : 1210005 chunks : 8
Shard shard1 contains 25.04% data, 25.04% docs in cluster, avg obj size on shard : 36B
Shard shard2 contains 37.55% data, 37.55% docs in cluster, avg obj size on shard : 36B
Shard shard3 contains 37.4% data, 37.4% docs in cluster, avg obj size on shard : 36B
```

기존 데이터까지 총 1210005 개 데이터를 샷딩했다.

Test DB 를 mongos 를 통해 mongodump 로 BSON 파일로 추출해보자

```
# mongodump -h 172.32.114.98:27018 -d test -u cluster_admin_user -p
124578 -o db_backups/ --authenticationDatabase=admin
```

에러 발견

```
[root@ip-172-32-114-98 centos]# mongodump --host 172.32.114.98:27018 --db test -u cluster_admin_user -p 124578 --out db_backups/
2022-10-28T08:40:25.399+0800 Failed: can't create session; could not connect to server: connection() error occurred during connection handshake: auth error: sasl conversation error: s
nable to authenticate using mechanism "SCRAM-SHA-1": (AuthenticationFailed) Authentication failed.
```

- 이 에러는 권한 에러이다. auth 를 진행하는 디비를 지정하지 않아서 나오는 문제인 듯 싶다.

```
[root@ip-172-32-114-98 centos]# mongodump -h 172.32.114.98:27018 -d test -u cluster_admin_user -p 124578 -o db_backups/ --authenticationDatabase=admin
2022-10-28T08:45:03.370+0800 writing test.item to db_backups/test/item.bson
2022-10-28T08:45:05.638+0800 done dumping test.item (1210005 documents)
[root@ip-172-32-114-98 centos]#
```

성공

이제 이 파일을 s3 에 저장하면 된다.

먼저 AWS user 를 만들어주어야한다. S3 에 대한 권한을 가져야 하니 IAM 콘솔에서 새 user 를 만들고 만드는 부분에서 권한 설정시 [AmazonS3FullAccess](#) 를 추가해 주어야한다.

aws configure 를 입력하여 aws 접속 권한을 부여해 주어야한다.

```
[root@ip-172-32-114-98 centos]# aws configure
AWS Access Key ID [None]: 
AWS Secret Access Key [None]: 
Default region name [None]: ap-northeast-2
Default output format [None]:
```

aws s3 sync test "s3://shkimmongobackup"

[DB] [내 S3 버킷 이름]

입력하여 백업을 진행한다.

```
[root@ip-172-32-114-98 db_backups]# aws s3 sync test "s3://shkimmongobackup"
upload: test/item.metadata.json to s3://shkimmongobackup/item.metadata.json
upload: test/item.bson to s3://shkimmongobackup/item.bson
```

```
[root@ip-172-32-114-98 db_backups]# aws s3 ls "s3://shkimmongobackup"
2022-10-28 08:51:06 43560202 item.bson
2022-10-28 08:51:06 239 item.metadata.json
[root@ip-172-32-114-98 db_backups]# aws s3 ls "s3://shkimmongobackup"
```

백업 성공. 자동화를 다음주에 구현해보자

10/31 추가

자동화 추가

crontab 이라는 스케줄러를 이용하여 지정된 시간에 Backup 이 진행되도록 할 것이다.

백업 설정은

매일 11 시 10 분 mongodump 실행 /

매일 11 시 15 분 aws s3 로의 sync 실행 이다.

db_backup.sh 파일을 생성하고 `chmod +x /home/centos/scripts/db_backup.sh` 로 실행권한을 부여해준다.

[illegible]

다음은 sync 를 위한 shell script 를 만들고 위와 같이 실행권한까지 부여해준다.

[illegible]

```
> crontab -e
: edit crontab

> crontab -l
: crontab 내용 조회

> crontab -r
: crontab 내용 모두 삭제
```

The diagram illustrates the components of a date in Korean. It starts with a vertical line on the left, branching out to the right to connect to various date parts. The parts are listed from top to bottom:

- 요일 (0 - 6)**: Day of the week (0:일요일, 1:월요일, 2:화요일, ..., 6:토요일)
- 월 (1 - 12)**: Month
- 일 (1 - 31)**: Day
- 시 (0 - 23)**: Hour
- 분 (0 - 59)**: Minute

[illegible]

cron 스케줄러가 백그라운드에서 실행 될 수 있도록 service crond start 해주는 것을 잊지말자. (항상 crontab -e 를 통해 수정을 거친 후에는 service crond restart 를 통해 재시작 해주어야한다.)

crontab 에 대한 로그는 cat /var/log/cron 명령어를 통해 확인 가능하다.

d

crontab 이 지정한 시간에 실행되지 않는 문제 발생 .

```
Oct 31 02:10:39 ip-172-32-114-98 crond[23067]: (CRON) INFO (Syslog will be used instead of sendmail.)
Oct 31 02:10:39 ip-172-32-114-98 crond[23067]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 63% if used.)
Oct 31 02:10:40 ip-172-32-114-98 crond[23067]: (CRON) INFO (running with inotify support)
Oct 31 02:10:40 ip-172-32-114-98 crond[23067]: (CRON) INFO (@reboot jobs will be run at computer's startup.)
Oct 31 02:12:08 ip-172-32-114-98 crontab[23155]: (root) BEGIN EDIT (root)
Oct 31 02:12:32 ip-172-32-114-98 crontab[23155]: (root) REPLACE (root)
Oct 31 02:12:32 ip-172-32-114-98 crontab[23155]: (root) END EDIT (root)
Oct 31 02:12:59 ip-172-32-114-98 crond[23067]: (CRON) INFO (Shutting down)
Oct 31 02:12:59 ip-172-32-114-98 crond[23215]: (CRON) INFO (Syslog will be used instead of sendmail.)
Oct 31 02:12:59 ip-172-32-114-98 crond[23215]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 24% if used.)
Oct 31 02:12:59 ip-172-32-114-98 crond[23215]: (CRON) INFO (running with inotify support)
Oct 31 02:12:59 ip-172-32-114-98 crond[23215]: (CRON) INFO (@reboot jobs will be run at computer's startup.)
Oct 31 02:14:08 ip-172-32-114-98 crontab[23287]: (root) BEGIN EDIT (root)
Oct 31 02:15:06 ip-172-32-114-98 crontab[23287]: (root) END EDIT (root)
```

로그를 확인해 보았을 때 실행에 대한 로그가 존재하지 않는 걸로 보아 아예 실행 자체가 되지 않는 듯 해보임

```
[root@ip-172-32-114-98 centos]# date
Mon Oct 31 02:19:48 UTC 2022
```

혹시나 하는 마음에 시스템 상의 시간을
체크해본 결과 다른 시간을 확인

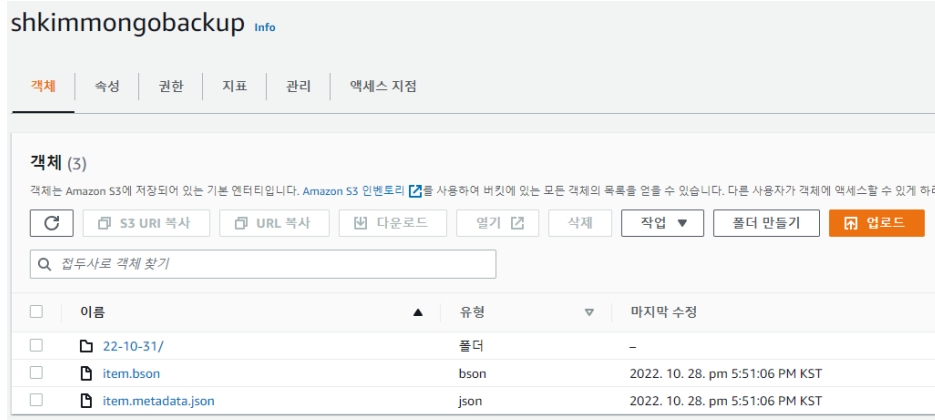
시간을 올바르게 고친 후 재 시도 해보았을 데 아래 오류 출력

```
Oct 31 05:05:02 ip-172-32-114-98 CROND[32434]: (root) CMD (/home/centos/scripts/db_backup.sh >> /home/centos/scripts/backup.txt)
Oct 31 05:05:02 ip-172-32-114-98 CROND[32431]: (root) CMDOUT (2022-10-31T05:05:02.689+0000)Writing test.item to /home/centos/db_backups/31-10-22/test/item.bson)
Oct 31 05:05:04 ip-172-32-114-98 CROND[32431]: (root) CMDOUT (2022-10-31T05:05:04.317+0000)@ollidone dumping test.item (1210005 documents))
Oct 31 05:07:01 ip-172-32-114-98 CROND[32560]: (root) CMD (/home/centos/scripts/db_sync.sh >> /home/centos/scripts/sync.log)
Oct 31 05:07:01 ip-172-32-114-98 CROND[32558]: (root) CMDOUT (/home/centos/scripts/db_sync.sh: line 2: aws: command not found)
```

aws 명령어를 찾을 수 없다고 하는데 db_sync.sh 파일에서 aws s3 ... 이 명령어가 존재하지 않기 때문에 /usr/local/bin/aws s3 ... 으로 바꾸고 재 실행하겠다.

```
backup.txt db_backup.sh db_sync.sh sync.log
[root@ip-172-32-114-98 centos]# cat scripts/backup.txt
[root@ip-172-32-114-98 centos]# cat scripts/sync.log
upload: ../home/centos/db_backups/31-10-22/test/item.metadata.json to s3://shkimmongobackup/31-10-22/test/item.metadata.json
upload: ../home/centos/db_backups/test/item.metadata.json to s3://shkimmongobackup/test/item.metadata.json
upload: ../home/centos/db_backups/22-10-31/test/item.metadata.json to s3://shkimmongobackup/22-10-31/test/item.metadata.json
upload: ../home/centos/db_backups/31-10-22/test/item.bson to s3://shkimmongobackup/31-10-22/test/item.bson
upload: ../home/centos/db_backups/test/item.bson to s3://shkimmongobackup/test/item.bson
[root@ip-172-32-114-98 centos]#
```

지정된 시간에 정상적으로 실행된 모습이다.



성공

- S3 storage class 는 쓰임새에 따라 크게 3 가지가 있다.

- Standard → 짧은 지연 시간 및 높은 처리량 성능 / 여러 가용 영역에 걸쳐 99.999999999%의 객체 내구성을 제공하도록 설계
- Standard-IA → 저렴한 GB 당 스토리지 요금과 GB 당 검색 요금으로 제공합니다. 저렴한 비용과 높은 성능의 조합을 제공하는 S3 Standard-IA 는 장기 스토리지, 백업 및 재해 복구 파일용 데이터 스토어에 이상적입니다.
- One Zone-IA → . S3 One Zone-IA 는 자주 액세스하지 않는 데이터에 대한 저렴한 옵션을 원하지만 S3 Standard 또는 S3 Standard-IA 스토리지와 같은 가용성 및 복원력이 필요 없는 고객에게 적합합니다

현재 사용하려고 하는 재해 복구 데이터 저장을 위해서는 Standard-IA 클래스가 적합하다. 저렴한 비용과 장기 보관 가능 한 부분이 강점인 것이다.

이를 위해 script 상에서 class 를 바꾸는 방법은 option 을 주는 것인데

sync shell script 에서

```
/user/local/bin/aws s3 ( — ) ( — ) --storage-class STANDARD_IA
```

이 옵션을 추가해주면 된다.