**How to Create and Use MongoDB Backups on Ubuntu 14.04 and management CURD**

**Step 1 — Creating an Example Database**

**mongo**

**use myDatabase**

Output

switched to db myDatabase

db.myCollection.insert([

{'name': 'Alice', 'age': 30},

{'name': 'Bill', 'age': 25},

{'name': 'Bob', 'age': 35}

]);

Output of a successful insert() operation

BulkWriteResult({

"writeErrors" : [ ],

"writeConcernErrors" : [ ],

"nInserted" : 3,

"nUpserted" : 0,

"nMatched" : 0,

"nModified" : 0,

"nRemoved" : 0,

"upserted" : [ ]

})

**Step 2 — Checking the Size of the Database**

db.stats().dataSize;

For the current database, the value of dataSize will be a small number:

Output of db.stats().datasize

592

Note that the value of dataSize is only a rough estimate of the size of the backup.

**Step 3 — Creating a Backup**

mongodump -d myDatabase -o ~/backups/first\_backup

If the backup creation is successful, you will see the following log messages:

Directory structure of a MongoDB backup

first\_backup

└── myDatabase

├── myCollection.bson

├── myCollection.metadata.json

└── system.indexes.bson

**Step 4 — Deleting the Database**

Open the mongo shell and connect to myDatabase.

mongo myDatabase

Delete the database using the dropDatabase method.

db.dropDatabase();

If the deletion is successful, you'll see the following message:

Output of dropDatabase()

{ "dropped" : "myDatabase", "ok" : 1 }

You can now use the find method of your collection to see that all the data you inserted earlier is gone.

db.myCollection.find();

There will be no output from this command because there's no data to display in the database.

**Step 5 — Restoring the Database**

mongorestore ~/backups/first\_backup/

You'll see the following log messages if the restore operation is successful:

만일 mongorestoe 명령실행시 다음과 같은 오유 가 발생한다면 다음 option 을 추가한다.

오유: too many positional arguments (import 경로 에 공백문자 포함, 경로를 바꾼다.)

: don't know what to do with file

이경우는 다음 명령어를 실행한다.

mongorestore -d dbname dbpath

dbname은 자료기지이름, dbpath는 현재 저장되여 있는 자료기 지 backup 경로.

mongo myDatabase

Then, call the find method on your collection.

db.myCollection.find();

If everything went well, you should now be able to see all the data you inserted earlier.

Output of find()

{ "\_id" : ObjectId("5654e76f21299039c2ba8720"), "name" : "Alice", "age" : 30 }

{ "\_id" : ObjectId("5654e76f21299039c2ba8721"), "name" : "Bill", "age" : 25 }

{ "\_id" : ObjectId("5654e76f21299039c2ba8722"), "name" : "Bob", "age" : 35 }

MongoDB 자료기지 관리

-자료기지 목록 보기(show databases)

>show dbs

-자료기지 table (show tables)

>show collections

-새자료기 지 리용 (아래에서 sample 자료기지 이름이다. )

>use sample

-자료기지에서 usres 라는 collection 하나 생성

> db.users.insert({username: '실험'})

-실지 명령어로 수행된 내용은 다음과 같다.(mini는 자료기지 이름)

>use mini

>db.users.insert({username: '실험'})

>db.users.find()

db.user.find()로 sql의 select처럼 검색질의문을 보낼수 있다.

-Create & Read

>db.users.save({username:'실험'})

>db.users.count()

>db.users.find()

- insert() 와 save()차이

insert()는 무조건 값을 추가하지만 save()는 중복 id 값이 없는 경우에 추가하고 있다면 수정하는 명령어 이다.

mongodb 의 내용은 \_id 라는 마당이 반드시 필요하며 이값이 없으면 값을 자동으로 생성해서 추가하게 됩니다.

-update

>db.users.update({usrname: '실험'}, {$set : {country: '실험1'}})

>db.users.find()

$set 연산자는 해당하는 배열전체에 대해 다시쓰기를 한다고 한다.

새롭게 추가한 항목을 제거하기 위해서 $unset 연산자를 사용한다.

>db.users.find()

>db.users.update({username: '실험'}, {$unset:{country : 1}})

>db.users.find()

-delete

사제는 remove()를 호출한다.

>db.users.remove({username : '실험'})

>db.users.find()

그러나 remove 는 collection을 지우지는 않는다.

collection을 제거하려면 drop()을 사용하면 된다.

>db..users.remove()

>db.users.find()

>show collections

>db.users.drop()

>show collections

만약 db도 삭제하기위해서는 dropdatabase() 사용

>show dbs

>db.dropdatabase()

>show dbs