02 Back Up and Restore with MongoDB Tools

The mongodump and mongorestore utilities work with BSON data dumps, and are useful for creating backups of small deployments.

When backing up your data with MongoDB's tools, consider the following guidelines:

- Label files so that you can identify the contents of the backup as well as the point in time that the backup reflects.
- Use an alternative backup strategy such as Filesystem Snapshots or MongoDB Cloud Manager if the performance impact of mongodump and mongorestore is unacceptable for your use case.
- Use --oplog to capture incoming write operations during the mongodump operation to ensure that the backups reflect a consistent data state.
- Ensure that your backups are usable by restoring them to a test MongoDB deployment.

Basic mongodump Operations

The mongodump utility backs up data by connecting to a running mongod. The utility can create a backup for an entire server, database or collection, or can use a query to backup just part of a collection.

When you run mongodump without any arguments, the command connects to the MongoDB instance on the local system (e.g. localhost) on port 27017 and creates a database backup named dump/ in the current directory.

Sintaxis

mongodump --host=<dominio-o-ip> --port=27017

To specify a different output directory, you can use the --out or -o option:

mongodump ... --out=/data/backup/

Práctica

Creamos dos directorios

md data/server

md data/backup

y levantamos un servidor:

mongod --port 27100 --dbpath data/server

Escribimos:

```
> use getafe1
switched to db getafe1
> for (i=0; i < 1000; i++) {db.foo1.insert({a: i})}
WriteResult({ "nInserted" : 1 })
> use getafe2
switched to db getafe2
> for (i=0; i < 1000; i++) {db.foo2.insert({a: i})}
WriteResult({ "nInserted" : 1 })
```

Desde otra terminal lanzamos mongodump:

```
MacBook-Pro-de-Pedro:~ pedro$ mongodump --port 27100
--out=data/backup/mongodump-11-02-2020
2020-02-11T17:23:49.095+0100    writing admin.system.version to
2020-02-11T17:23:49.096+0100    done dumping admin.system.version (1
document)
2020-02-11T17:23:49.097+0100    writing getafe1.foo1 to
2020-02-11T17:23:49.098+0100    writing getafe2.foo2 to
2020-02-11T17:23:49.099+0100    done dumping getafe1.foo1 (1000 documents)
2020-02-11T17:23:49.100+0100    done dumping getafe2.foo2 (1000 documents)
```

To limit the amount of data included in the database dump, you can specify --db and --collection as options to mongodump. For example:

mongodump ... --collection=<colección> --db=<basededatos>

mongodump overwrites output files if they exist in the backup data folder. Before running the mongodump command multiple times, either ensure that you no longer need the files in the output folder (the default is the dump/ folder) or rename the folders or files.

Point in Time Operation Using Oplogs

Use the --oplog option with mongodump to collect the oplog entries to build a point-in-time snapshot of a database within a replica set. With --oplog, mongodump copies all the data from the source database as well as all of the oplog entries from the beginning to the end of the backup procedure. This

operation, in conjunction with mongorestore --oplogReplay, allows you to restore a backup that reflects the specific moment in time that corresponds to when mongodump completed creating the dump file.

Restore a Database with mongorestore

The mongorestore utility restores a binary backup created by mongodump. By default, mongorestore looks for a database backup in the dump/ directory.

The mongorestore utility restores data by connecting to a running mongod directly.

mongorestore can restore either an entire database backup or a subset of the backup.

To use mongorestore to connect to an active mongod, use a command with the following prototype form:

Sintaxis

mongorestore --port=<numero-puerto> <ruta>

Práctica

Borramos una de las colecciones:

```
> db.foo2.drop()
true
```

Y restauramos:

```
MacBook-Pro-de-Pedro:~ pedro$ mongorestore --port 27100
data/backup/mongodump-11-02-2020

...
2020-02-11T18:31:52.058+0100    no indexes to restore
2020-02-11T18:31:52.058+0100    finished restoring getafe1.foo1 (0 documents, 1000
failures)
2020-02-11T18:31:52.058+0100    1001 document(s) restored successfully. 1000
document(s) failed to restore.
```

Ahora comprobamos como está restituida la colección borrada.

```
> db.foo2.find()
{ "_id" : ObjectId("5e42d3864fac851c55bc1337"), "a" : 0 }
{ "_id" : ObjectId("5e42d3864fac851c55bc1338"), "a" : 1 }
```

```
{ "_id" : ObjectId("5e42d3864fac851c55bc1339"), "a" : 2 }
{ "_id" : ObjectId("5e42d3864fac851c55bc133a"), "a" : 3 }
{ "_id" : ObjectId("5e42d3864fac851c55bc133b"), "a" : 4 }
...
```