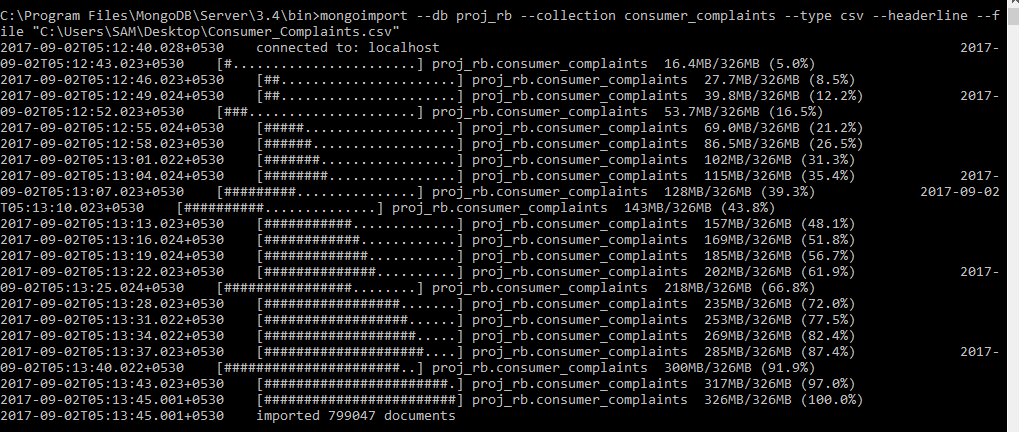
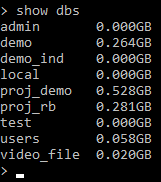
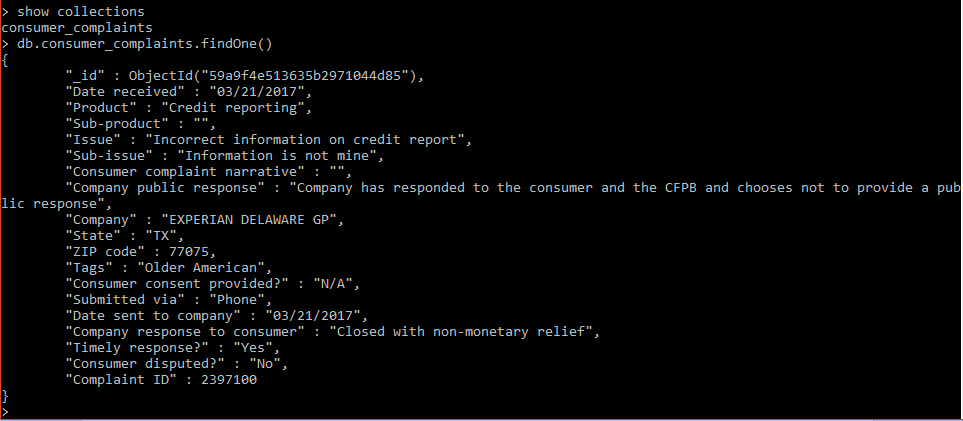
mongoimport --db proj\_rb --collection consumer\_complaints --type csv --headerline --file "C:\Users\SAM\Desktop\Consumer\_Complaints.csv"







Cold Backup.

Here after loading the data .

Now close all the instances of mongodb.

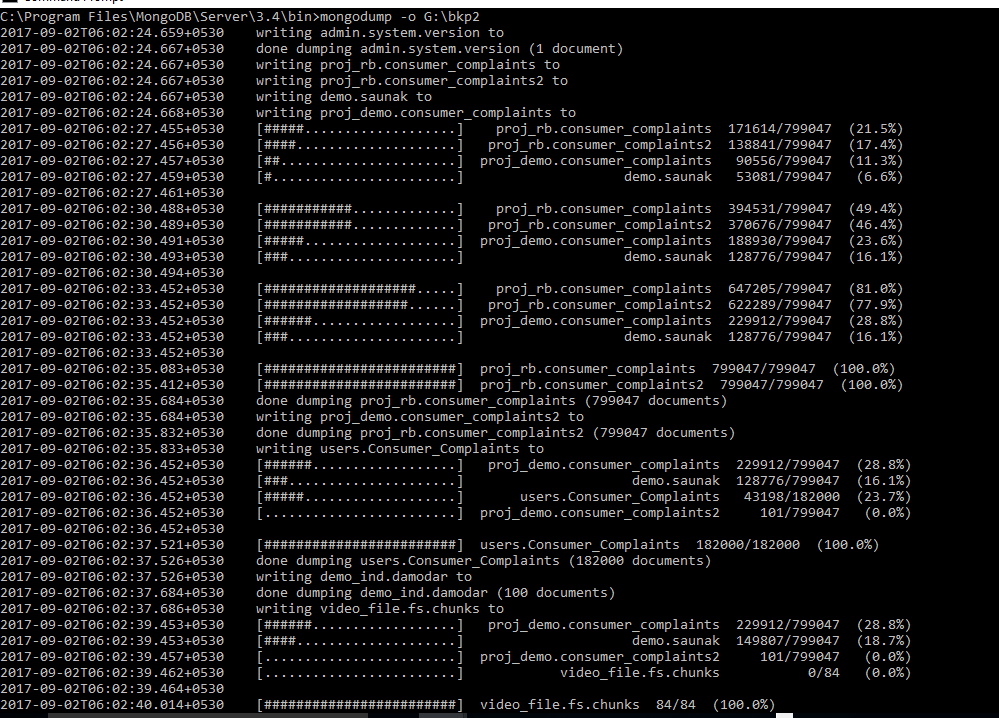
We need to go to the directory data\db\ where the data is stored.

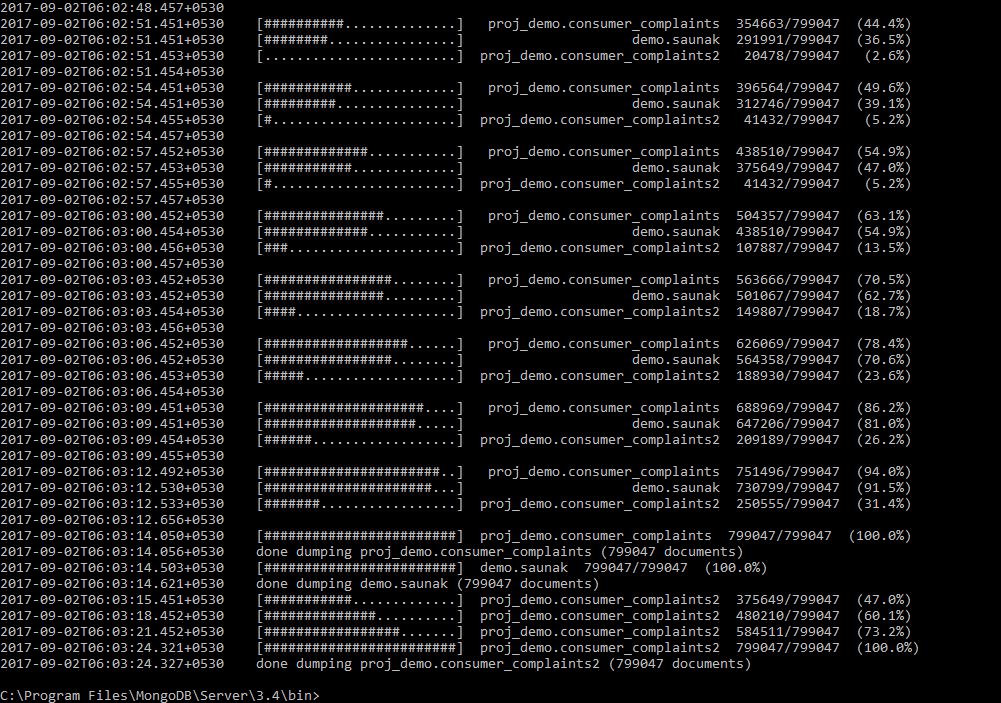
Copy all the files , and store them in another folder (Backup) . Now in case of emergency data gets corrupted we can

Just copy all the files from the backup folder and replace all the files in the mongodb data directory where the data gets stored.

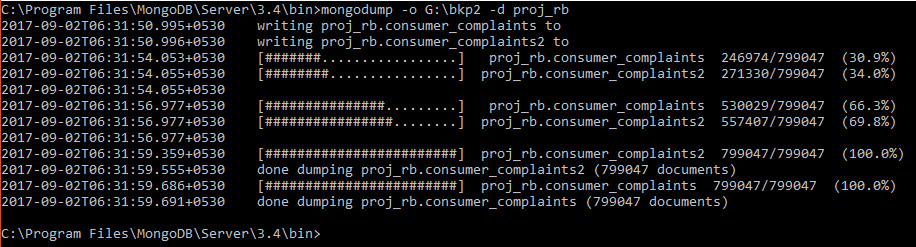
Full backup by mongodump

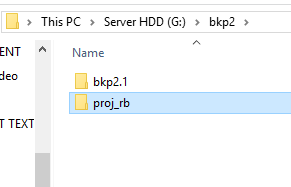
At instant level (everting thing is taken in the dump)

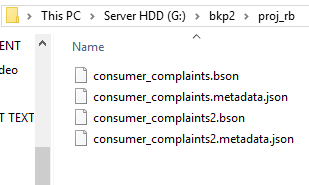




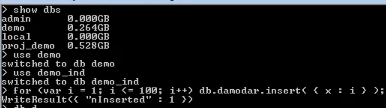
(at database level)

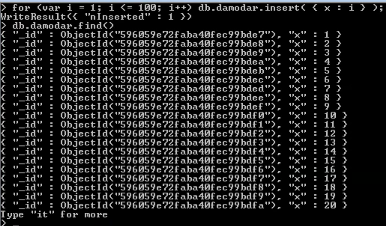


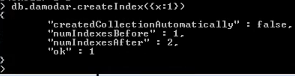




Backup of indexes









Open this metadata





**Backups and Recovery**

**Premise**

"I thought I was on my development machine and I’ve accidentally dropped a collection in production."

The backup system captured its most recent backup last night, which is in the /home/ec2-user/scripts/dump directory, but the system subsequently continued to accumulate new data through the night. You need to restore your data to the point in time just before the you dropped the collection.

The collection is test.users and the total number of documents prior to the drop was 20,000.

Ans:- here we have to drop the current data and restore from the last night’s dump, so that it could be brought back to the state before the collection was dropped.

I use the following command to backup a production/live database and restore it to my local development machine - the command assumes the following:

* The path to the directory containing the backup you made using mongodump
* An instance of MongoDB running on the machine the command is run on.
* Drop the collection before using the below command.

mongorestore --nsInclude test.users /home/ec2-user/scripts/dump directory/dump