Backup and Restore Database in Mongo DB

What’s the most important thing about making backups?

RESTORING THEM!

Backup Options

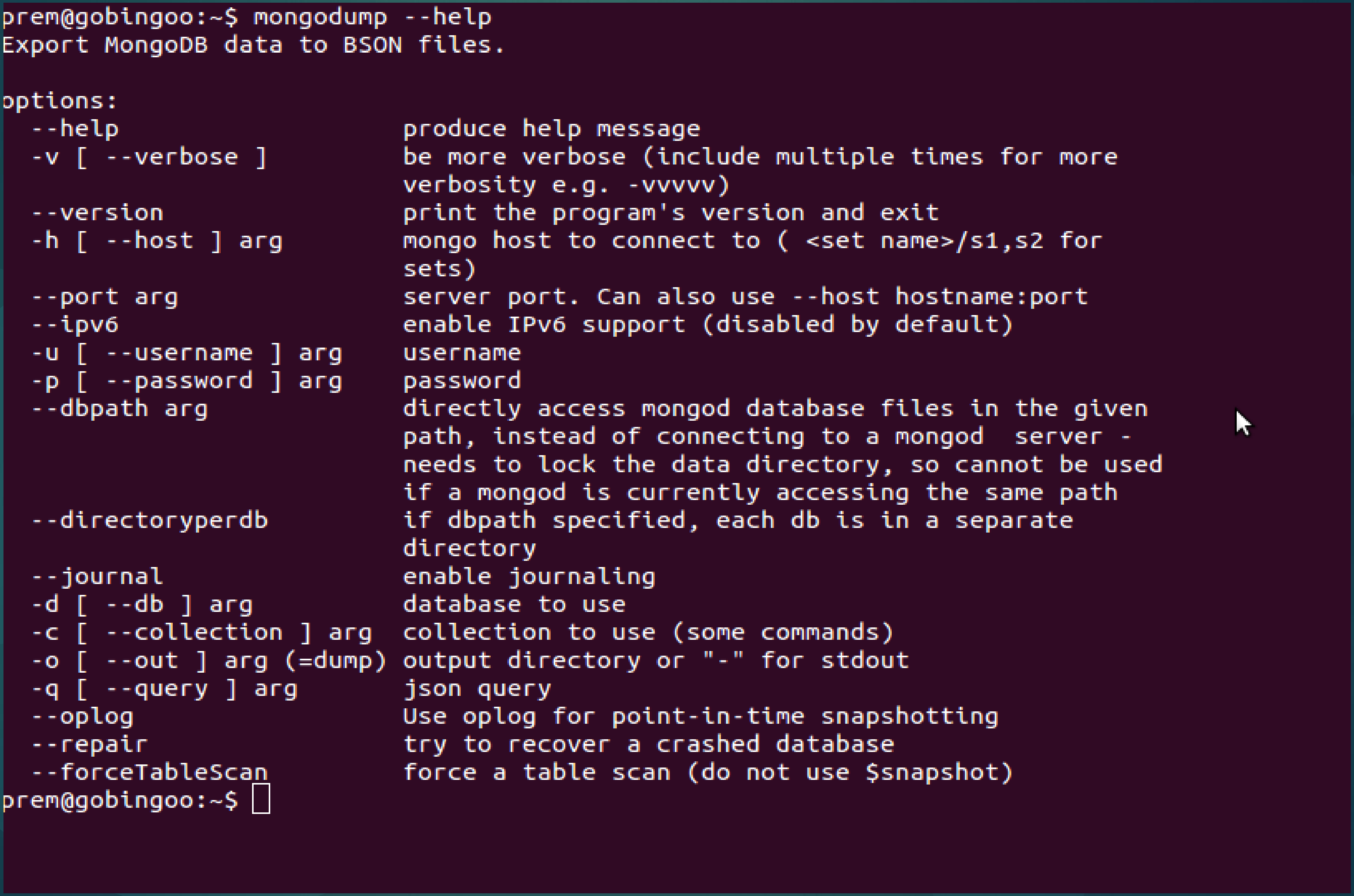
Mongodump utility

Copy the files

mongodump

- Dumps collections to *.bson files
- Mirrors your structure
- Can be run in live or offline mode-

mongodump



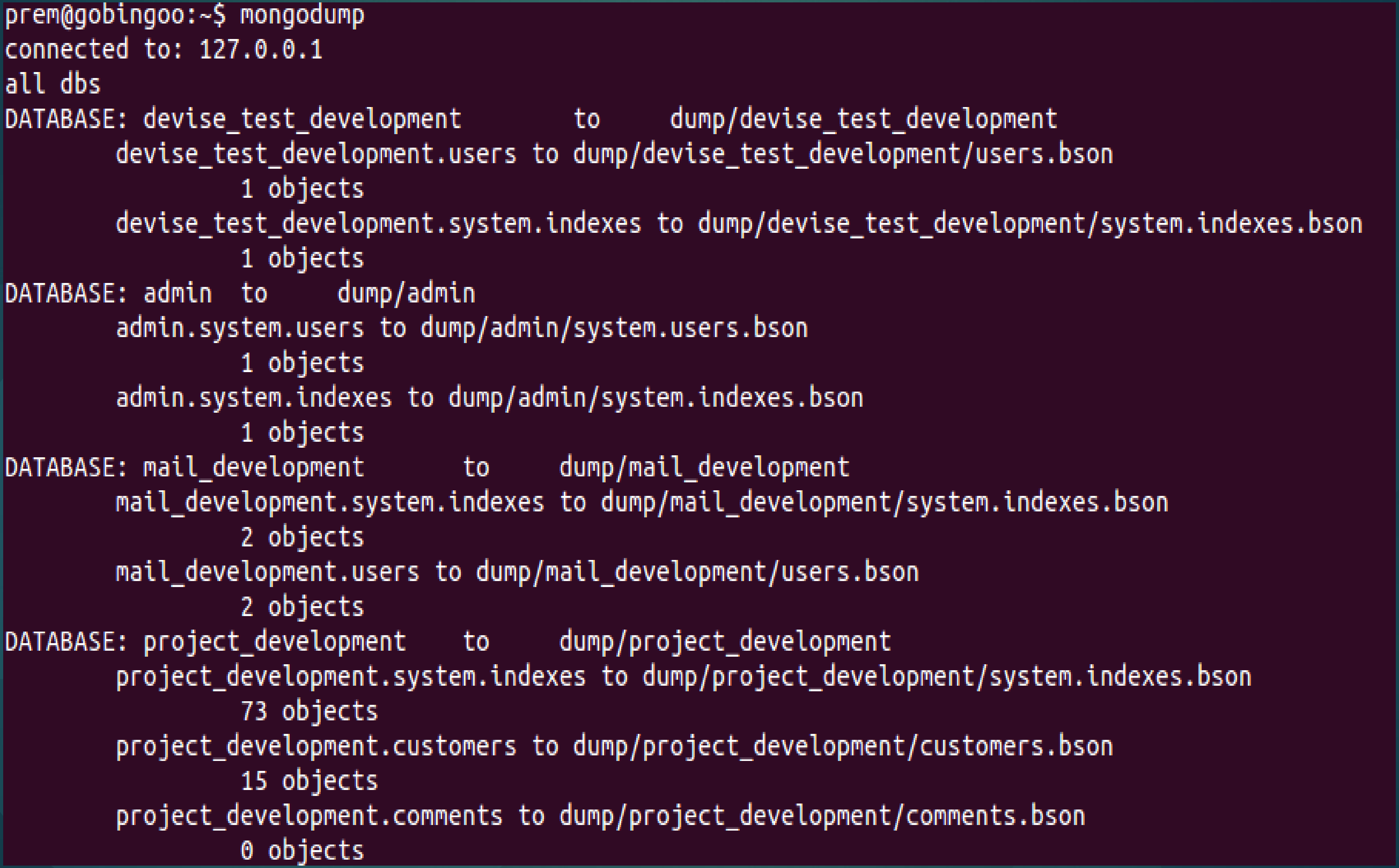
```
prem@gobingoo:~$ mongodump --help
Export MongoDB data to BSON files.

options:
  --help                     produce help message
  -v [ --verbose ]           be more verbose (include multiple times for more
                             verbosity e.g. -vvvvv)
  --version                  print the program's version and exit
  -h [ --host ] arg          mongo host to connect to ( <set name>/s1,s2 for
                             sets)
  --port arg                 server port. Can also use --host hostname:port
  --ipv6                     enable IPv6 support (disabled by default)
  -u [ --username ] arg      username
  -p [ --password ] arg      password
  --dbpath arg               directly access mongod database files in the given
                             path, instead of connecting to a mongod  server -
                             needs to lock the data directory, so cannot be used
                             if a mongod is currently accessing the same path
  --directoryperdb           if dbpath specified, each db is in a separate
                             directory
  --journal                  enable journaling
  -d [ --db ] arg            database to use
  -c [ --collection ] arg    collection to use (some commands)
  -o [ --out ] arg (=dump)   output directory or "-" for stdout
  -q [ --query ] arg         json query
  --oplog                    Use oplog for point-in-time snapshotting
  --repair                   try to recover a crashed database
  --forceTableScan           force a table scan (do not use $snapshot)
prem@gobingoo:~$ 
```

mongodump

- Type mongodump in your terminal
- mongodump --db hero(hero is the database name)
- Database save inside dump folder

mongodump



```
prem@gobingoo:~$ mongodump
connected to: 127.0.0.1
all dbs
DATABASE: devise_test_development        to     dump/devise_test_development
        devise_test_development.users to dump/devise_test_development/users.bson
                 1 objects
        devise_test_development.system.indexes to dump/devise_test_development/system.indexes.bson
                 1 objects
DATABASE: admin  to     dump/admin
        admin.system.users to dump/admin/system.users.bson
                 1 objects
        admin.system.indexes to dump/admin/system.indexes.bson
                 1 objects
DATABASE: mail_development       to     dump/mail_development
        mail_development.system.indexes to dump/mail_development/system.indexes.bson
                 2 objects
        mail_development.users to dump/mail_development/users.bson
                 2 objects
DATABASE: project_development    to     dump/project_development
        project_development.system.indexes to dump/project_development/system.indexes.bson
                 73 objects
        project_development.customers to dump/project_development/customers.bson
                 15 objects
        project_development.comments to dump/project_development/comments.bson
                 0 objects
```

mongorestore

- *Mongorestore utility.*
- Restore all database inside dump

Repair

- Repair all databases on the server:
  - $ mongod –repair
- Repair Database command to repair a single
- database:

```
 > use project_development
> db.runCommand({repairDatabase: 1})
```

- which operate on a single collection :

 rebuild indexes

```
        >use tuotorial_development
        > db.employee_qualifications.reIndex()
```

 compact the collection

```
      > db.runCommand({ compact: "employee_qualifications" })
```

Recover Mongo DB

- Unexpected Shutdown
- **Error: couldn't connect to server 127.0.0.1 shell/mongo.js:84**

Solution

- Remove the mongod.lock file
- Run the *--repair* process

OTHER MONGODB COMMAND LINE UTILITIES

MongoStat

The mongostat utility provides a quick overview of the status of a currently running mongod or mongos instance. mongostat is functionally similar to the UNIX/Linux file system utility vmstat,
but provides data regarding mongod and mongos instances

Examples

```
mongostat --rowcount 20 1
mongostat --rowcount 20
mongostat -n 20 1
mongostat -n 20
```

MongoImport

The mongoimport tool provides a route to import content from a JSON, CSV, or TSV export
created by mongoexport, or potentially, another third-party export tool .
See the Import and Export MongoDB Data document for a more in depth usage overview ,
and the mongoexport document for more information regarding mongoexport ,
which provides the inverse "exporting "capability.

Example

```
mongoimport --db sales --collection contacts --stopOnError --dbpath
/srv/mongodb/
```

MongoExport

mongoexport is a utility that produces a JSON or CSV export of data stored in a MongoDB instance .
See the Import and Export MongoDB Data document for a more in depth usage overview ,
and the mongoimport document for more information regarding the mongoimport utility, which provides
the inverse “importing ”capability.

Example

```
mongoexport --db sales --collection contacts --out contacts.json --
journal
```

MongoTop

mongotop provides a method to track the amount of time a MongoDB instance spends reading and writing data .
mongotop provides statistics on a per-collection level .
By default, mongotop returns values every second.

Example

```
  mongotop 15
```

BsonDump

The bsondump converts BSON files into human-readable formats, including JSON.
For example, bsondump is useful for reading the output files generated by mongodump

Example

```
bsondump collection.bson > collection.json
```

BACKUP AND RESTORE LAB