MongoDB Import/Export Backup

Ramesh S

mongoexport

$mongoexport

Export MongoDB data to CSV or JSON files.

Options:

/v [--verboselevel]	more detailed log output
/h [--host]	mongo host to connect to
/u [--username]	username
/p [--password]	password
/d [--db]	database to use
/c [--collection]	collection to use
/q [--query]	query filter, as a JSON string
/o [--out]	output file; if not specified, `stdout` is used
/f [--fields]	comma separated list of field names (required for exporting CSV)

Export all documents into a file

```
$ mongoexport -d accounts -c vendors -o vendors.json
```

- Exports the collection vendors from accounts database into a JSON file named `vendors.json`

Export to CSV

```
mongoexport -d accounts -c vendors --type=csv  -o vendors.csv -f _id,name,email
```

- This command will export the fields `_id`, `name`, `email` from *vendors* collection in *accounts* database into a csv file named `vendors.csv`

Export Based on a Query

```
$mongoexport -d accounts -c vendors -q “{profit:{$gt:100000}}” -o vendors.json
```

Export From a Remote Server

```
mongoexport -h xyz.abc.com -d employees -c qualifications
```

mongoimport

mongoimport

Import CSV, TSV or JSON data into MongoDB.

Options	
/h [--host]	mongo host to connect to (<set name>/s1,s2 for sets)
/u [--username]	username
/p [--password]	password
/d [--db]	database to use
/c [--collection]	collection to use (some commands)
/f [--fields]	comma separated list of field names e.g. -f name,age
[--file]	file to import from; if not specified stdin is used
[--drop]	drop collection before inserting documents
/mode: [insert|upsert|merge]	insert: insert only;
	upsert: insert or replace existing documents;
	merge: insert or modify existing documents

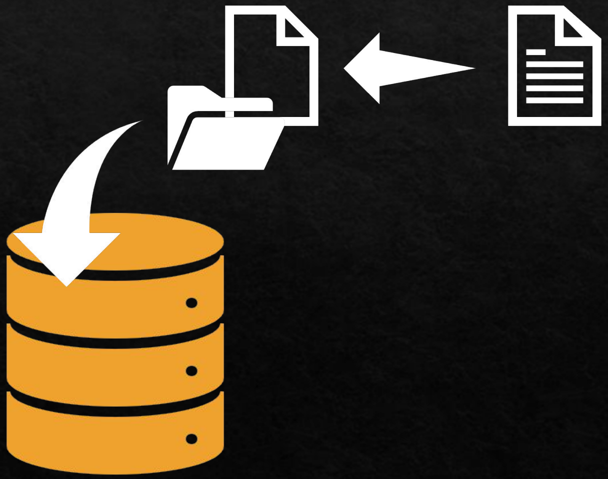
Import from File

```
$ mongoimport -d accounts -c audits --file audit.json
```

- If database ”*accounts*” and collection ”*audits*” do not exist, they are created automatically.

Insert or Update if Existing

```
$ mongoimport -d database -c collection --file my.json --upsert
```



Import with Username & Password

```
mongoimport -h xyz.abc.com -u trump -p america123 -d employees -c
qualifications --file data.json
```

Importing From CSV

```
mongoimport -d accounts -c vendors --type=csv  --file vendors.csv -f _id,name,email
```

mongodump

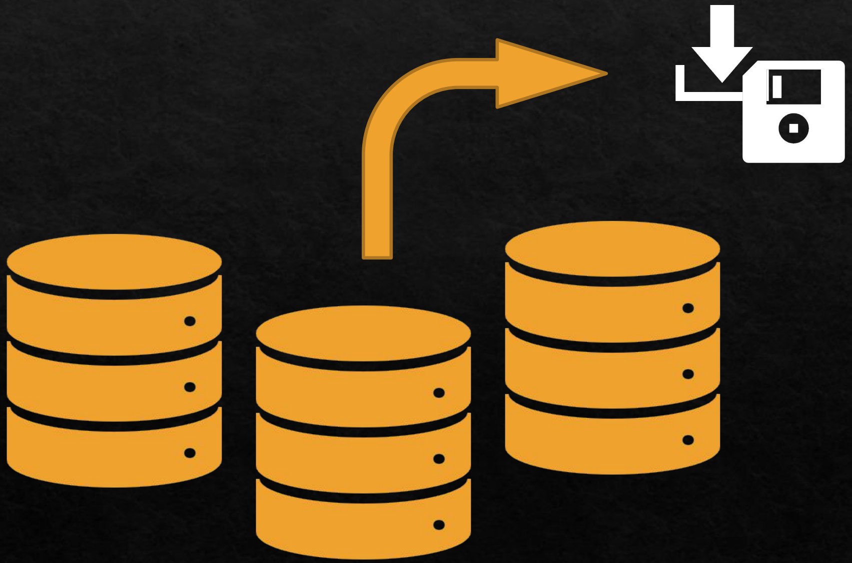
mongodump

- `mongodump` can export data from either `mongod` or `mongos` instances
- Options:

/h [hostname]	mongodb host to connect to
/ssl	connect to a mongod or mongos that has ssl enabled
/u [username]	username for authentication
/p [password]	password for authentication
/d [db:<database-name>]	database to use
/c [collection]	collection to use
/q [query]	query filter as a JSON string
/o [/out:<path>]	output directory (- for stdout)

Backup a Specific Database

```
mongodump --db customers
```



Backup a Specific Collection

```
mongodump --db customers --collection info
```

mongorestore

mongorestore

- Loads data from either a binary database dump created by `mongodump` or
- The standard input (starting in version 3.0.0) into a `mongod` or `mongos` instance

Options	
/h [hostname]	mongodb host to connect to
/ssl	connect to a mongod or mongos that has ssl enabled
/u [username]	username for authentication
/p [password]	password for authentication
/d [db:<database-name>]	database to use
/c [collection]	collection to use
/objcheck	query filter as a JSON string
/dir:<path>]	input directory (- for stdin)
/drop	drop each collection before import
/writeConcern:<option>	write concern options
/j [/numParallelCollections]	number of collections to restore in parallel (default: 4)

Restore a Specific Database

```
mongorestore --db customer dump/customer
```

- Restore the database customer from the directory `dump/customer`

Restore a Specific Collection

```
mongorestore --db company dump/company/emp.bson --collection emp
```

- Restore the collection emp into database company.
- The data is contained in the file `dump/company/emp.bson`