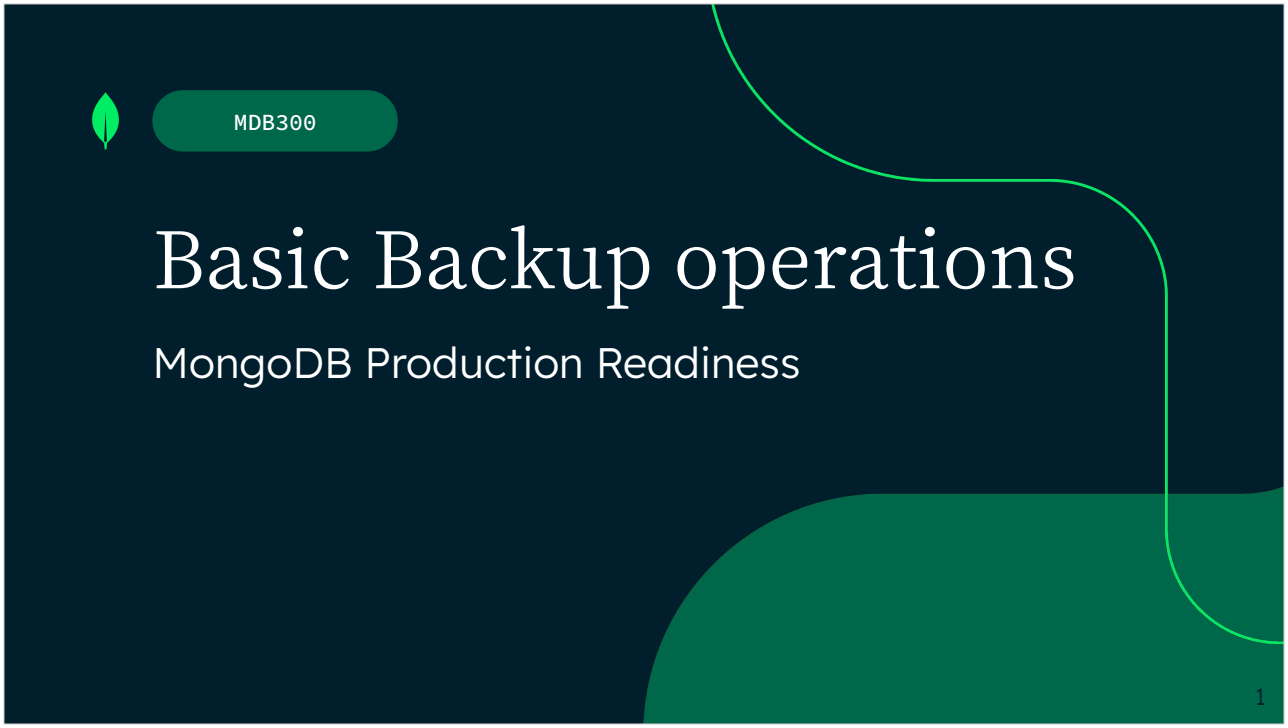


Release: 20250117

Topics we cover

Why are backups necessary?

Backup Plan

Backup Methods

mongodump and mongorestore

File System Backup

Why Backups?

Data is one of the company's most valuable assets

A severe data loss can cost a fortune or go out of business - like a bank losing all its data or an e-commerce company offline for one week

Data is a very valuable asset.
It is impossible to replace but easy to keep a copy of, compared to physical stock.
Therefore having appropriate backups - and being able to restore them is critical.

Causes of data loss

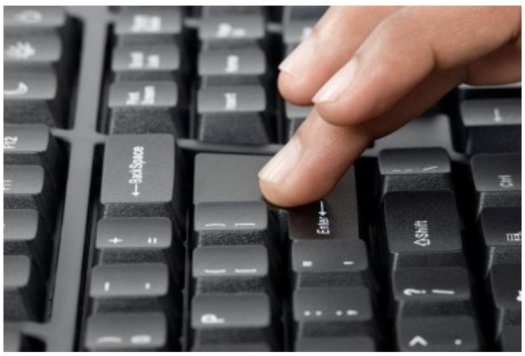
Data loss can occur due to many reasons - There are four different main situations:

Human Error

Database Failure / Corrupt

System Failure / Collapse

Security Breach



Question

What is the hardest thing about taking backups?

Making sure you can restore them - an untested backup is worthless.

Backup plan

A company should have an appropriate backup plan as per their use case

The same plan may not be feasible for everyone - Choose the best for your company

- RPO - Recovery Point Objective
- RTO - Recovery Time Objective

Replication of data is not a substitute for a backup - why?

RPO vs RTO

Recovery Point Objective (RPO)

How much data can you afford to lose?

At what point in time must the backups be when there is a data loss incident

Important to know how often a backup needs to be made

Recovery Time Objective (RTO)

How long can you afford to be offline?

How long should it take to have my application back online?

You need to plan your backup for:

- Recovery Time (RTO) - how long will it take to get back to normal.
- Recovery Point (RPO) - if you have to go to backups, how much did you lose
- Retention Time - If you need to go back to older data - how far can you go back.

Replicated data keeps you up in case of hardware failure, but bad code, bad actors, or human error all replicate, and so the replicated data may be destroyed too. Also, you don't have an 'Old' or archive version in case you need to look back.

Backup methods

There are a variety of environments and tooling related to each environment that can backup the database's data

- MongoDB Atlas Continuous Cloud Backups, or Cloud Backups
- MongoDB Cloud Manager or Ops Manager, backup snapshot
- Non-managed / self-service approaches

Comparing Backup methods

Considerations	mongodump	File System	Cloud Manager	Ops Manager
Initial Complexity	Medium	High	Low	High
Replica Set PIT	Yes	No	Yes	Yes
Sharded Snapshot	Yes	No	Yes	Yes
Restore Time	Slow	Fast	Medium	Medium
Incremental	No	No	Yes	Yes

Self-Service Backup Options

Document Level

- Logical
- mongodump / mongorestore

File System Level

- Physical
- Copy Files
- Volume / Disk snapshots

mongodump
mongorestore

mongodump

mongodump is the simplest option to backup online MongoDB databases

Offline you could just copy the files

There are two steps to perform disaster recovery when using mongodump

- Create the dump file using mongodump
- Restore the dump file with mongorestore in a disaster.

- mongodump can backup a database by copying it to BSON files along with metadata for indexes etc.
- mongodump doesn't make incremental backups so backups can take a long time and lot of space if data is large.
- It can also backup the oplog allowing you to restore to a consistent state.
  - If you do not use the oplog, then all data is backed up at a different point in time.
  - With the oplog - after restoring collections, it rolls the oplog forward to a consistent point in time.
  - This feature is not available when working with sharded clusters

mongodump

mongodump can be used to create backups of subparts of an instance, specific to:

Databases

Collections

Documents - based on a query

mongodump allows you to specify what parts of the database you want to back up.

mongodump - Pros and Cons

Pros

- Simple to use
- Can backup a subset of the data
- BSON files can be read without restoring

Cons

- Cannot backup only changes since last time so large backups are slow
- By default does not provide a point in time snapshot
- Restore time can take longer as indexes need to be rebuilt
- Cannot be part of the backup strategy with a sharded cluster

For a sharded cluster, instead of using mongodump manually we rely on MongoDB Atlas, Cloud Manager or Ops Manager to get it done automatically.

mongodump

Dumps a collection to BSON files - Mirrors your structure

Does not include indexes (rebuilt during restore), only captures the documents

`--db`	Dump a specific database
`--collection`	Dump a specific collection
`--oplog`	Record oplog while backing up
`--query/filter`	Selective dump
`--readPreference`	Read preference for source of the dump
`--gzip`	Compresses the backup
`--archive`	Outputs the backup to a file specified with this option

mongorestore

`--oplogReplay`	Replay oplog to point-in-time
`--oplogLimit`	Prevents replay oplog with timestamps newer than given time
`--writeConcern`	Specifies the write concern
`--numberInsertionWorkersPerCollection`	Defaults to 1 but for large imports. Can be increased to specify how many workers insert per collection concurrently
`--gzip`	Restores compressed files or data stream
`--archive`	Restores from a specified archive file

A backup without an oplog is not a snapshot of a point in time - each collection is copied and may be changing during the backups.
This is resolved by replaying the oplog, if you play the oplog entries between the start of taking your backup and the end then you end up
with a consistent snapshot of what the database looked like at the end of your backup process.
This does not result in consistency in a sharded cluster though unless you are careful to synchronise the oplogs, a truly consistent sharded cluster backup can only be done online by the enterprise backup tools.

File System Level

Physical Backups

Device Level Snapshot (like LVM)

- Needs no additional steps to ensure consistency
- A disk snapshot is always healthy or able to self restore
- Simply copying files is not enough

File Copy

- Must have the database stopped from changing
- Stop accepting writes

db.fsyncLock() or shutdown instance (could be a Secondary)
Copy files
db.fsyncUnlock()

To use the File System backup on a sharded cluster you need to stop the balancer alongside writes to the entire cluster to ensure a precise and consistent backup.

File System Restores

Restore all database files to disk - Snapshot or copy

Start a MongoDB instance pointing at those files

Add secondaries and allow them to sync

File System Backup - Pros and Cons

Pros

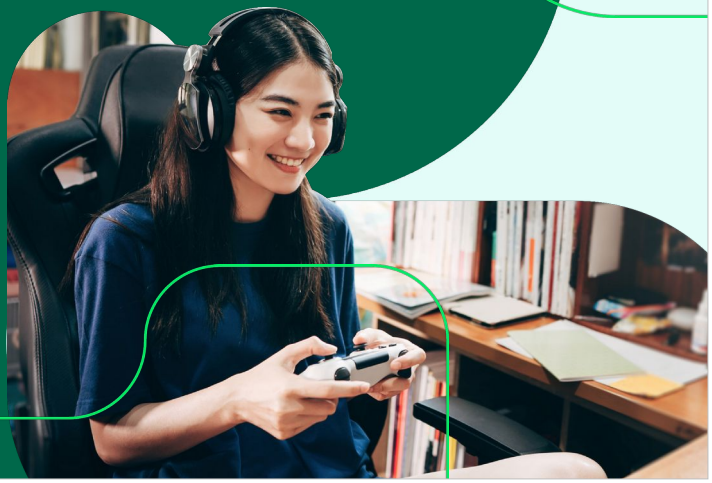
- With an LVM, is simple to do
- Easy to understand
- Quick to restore

Cons

- Without LVM, needs downtime
- Does not scale to a large system where copy time is long.
- Always backs up an entire database; hence slow.

When creating a backup in a sharded cluster downtime is always required using a File System Backup to ensure consistency.

Quiz Time!



#1. What are reliable methods for backing up MDB while it's running?

A	tar to write to tape archive	B	mongodump to tape archive	C	rsync
D	Cloud manager backup	E	mongodump to a new network drive		

Answer in the next slide.

#1. What are reliable methods for backing up MDB while it's running?

A. tar to write to tape archive

B. mongodump to tape archive

C. rsync

D. Cloud manager backup

E. mongodump to a new network drive

Enterprise solutions like CM backup can easily do this. If there is not enterprise option, then you'd have to rely on using mongodump and restoring to a tape archive or a new network drive. There isn't a tar option.

#2. What needs to be done to ensure a point in time backup with mongodump?

A	Backup the database schemas
B	Backup the data in all collections
C	Backup the admin database
D	Backup the Oplog
E	Backup the server logs

Answer in the next slide.

#2. What needs to be done to ensure a point in time backup with mongodump?

A	Backup the database schemas
B	Backup the data in all collections
C	Backup the admin database
D	Backup the Oplog
E	Backup the server logs

A mongodump has to have the information related to the oplog in addition to the data of the collections in order to do a point in time backup/restore.

#3. Select what the backups protect from but replication does not:

A	Hardware failure	B	Network failure	C	Storage failure
D	Human error	E	Code errors		

Answer in the next slide.

#3. Select what the backups protect from but replication does not:

A	Hardware failure	B	Network failure	C	Storage failure
D	Human error	E	Code errors		

Human error/code error can't be prevented with replication as it will replicate the bad code to the other nodes in the RS. However, using a restore from a backup taken prior to this can protect the consistency of your data.

Recap

Backups are needed even with Replication

Backups should be tested

MongoDB has Enterprise tooling for differential backups with low RPO and RTO

mongodump and file backups are a simple alternative - both have serious tradeoffs