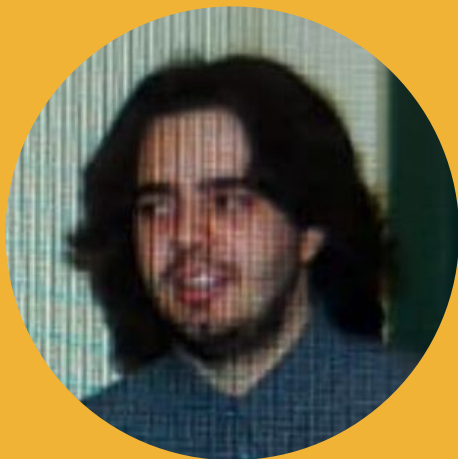




Backups and restore with Percona Backup for MongoDB (PBM)

Corrado Pandiani
Senior Architect

Milan, May 26th 2025



Corrado Pandiani
Senior Architect

This is me

- Open Source enthusiast
- MySQL and MongoDB expert
- Worked in the past as DBA, Web Developer, Project Manager, CRM and BI developer and instructor
- Spent 22 years in the football industry for a worldwide popular team
- Perconian since early 2018

Agenda

1. Terminology warm-up
2. Elements of MongoDB Backups
3. MongoDB Backup Options
4. Percona Backup for MongoDB (PBM)
5. Conclusions



Backups? What?

- Annoying task
- Require additional resources
- The most important duty for DBAs
- Backups are a matter of money, always
- You cannot permit you can lose data
 - Your business does not exist without data
- Even The Beatles were aware of backups
 - "Oh, I believe in yesterday"



Terminology warm-up

Logical vs Physical Backup and Restore

Logical

Collections dumped as BSON documents

Transfer by: DB Driver

Overhead: high

Backup: slow

Restore: very slow

Ideal for: selective backup/restore and portability

Impact on availability: none

Physical

DB's Data Files

Transfer by: File Copy

Overhead: minimal

Backup: fast

Restore: fast

Ideal for: disaster recovery

Impact on availability: none or may require downtime

Consistent Backup

All documents are restored at a version as of a single point in time

Logical	End time of the backup
Physical	Start time of the backup
Point-in-time restore	Whenere you like

Sharded clusters require some further synchronization

Hot Backup

- Taking the backup while the database is running
- No downtime
- Logical: always HOT
- Physical: depends on the methodology
 - shutdown required if copying the files
 - locking of the database required if you take a file system snapshot
 - HOT: if you use WiredTiger snapshot or PBM



Elements of MongoDB Backups

MongoDB Oplog for Replication

- MongoDB uses logical replication
- User writes are transformed to idempotent operations, written to the *"local"* db's *oplog.rs* collection
- Secondary nodes tail the oplog and apply those updates

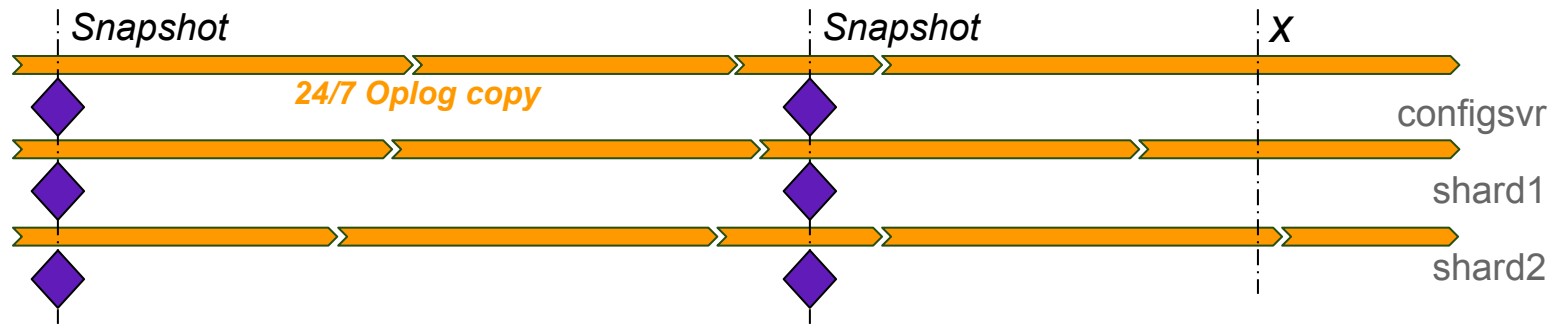
```
{ "ts" : Timestamp(1395663575, 1), "h" : ... , "v" : 2,  
  "op" : "i", "ns" : "db.coll_XYZ", "o" : { "_id" : ...,  
  "fieldA": ... } },  
{ "ts" : Timestamp(1395663575, 2), "h" : ... , "v" : 2,  
  "op" : "d", "ns" : "db.coll_XYZ", "o" : { "_id" : ... } },
```

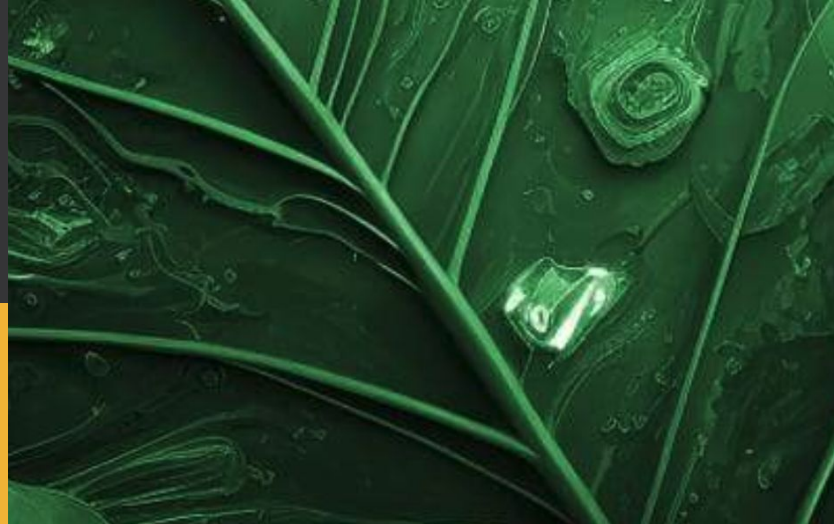
MongoDB Oplog extra use in backups

- A database snapshot might not be consistent
 - Logical backups read documents while updates are being made
 - Sharded cluster backups are made of multiple backups – one per replica set – which probably have (slightly) different capture times
- Solution
 - use the idempotent updates in the Oplog to ‘fast-forward’ to an identical time

Point-in-time Restore

- Add oplog capture 24/7 to enable PITR
- After the restore of backup snapshot, replay oplog from the snapshot backup time to X





MongoDB Backup Options

Backup & Restore solutions list

Logical

mongodump + mongorestore

Percona Backup for MongoDB (PBM)

Physical

Shutdown secondary and copy files

(Eventually) Lock database and take file system snapshot

WiredTiger Snapshot ('hot backup' in PBM)

Percona Backup for MongoDB (PBM)

MongoDB OpsManager Backup

Not free

mongodump + mongorestore

```
# full backup
$ mongodump --username=root --authenticationDatabase=admin --password=mypwd

# a collection backup
$ mongodump -d test -c mycoll --username=root --authenticationDatabase=admin --password=mypwd

# oplog backup
$ mongodump -d local -c oplog.rs -o oplogDumpDir/ --username=root --authenticationDatabase=admin --password=percona

# restore full backup
$ mongorestore --username=root --authenticationDatabase=admin --password=mypwd dump/

# restore a collection into a different namespace
$ mongorestore --username=root --authenticationDatabase=admin --password=mypwd --nsFrom=test.mycoll
--nsTo=test.mycoll2

# replay the Oplog
$ mongorestore --username=root --authenticationDatabase=admin --password=mypwd --oplogReplay oplogDumpDir/

# replay with point-in-time
$ mongorestore --username=root --authenticationDatabase=admin --password=mypwd --oplogReplay --oplogLimit 166791793
oplogDumpDir/
```

Stop Secondary and copy files

```
$ sudo systemctl stop mongod
```

```
...
```

```
$ aws s3 cp --recursive /data/db s3://mongodb_backups/20250313_1500
```

```
Completed 4 of 74 part(s) with 70 file(s) remaining
```

```
...
```

```
$ sudo systemctl start mongod
```

```
...
```

WiredTiger Snapshot (Hot Backup)

- Available for free in PSMDB
 - save to local path or stream to s3 bucket
- Not for free on MongoDB Enterprise

```
> use admin
switched to db admin
> db.runCommand({createBackup: 1, backupDir: "/tmp/backup"})
{ "ok" : 1 }

> db.runCommand({createBackup: 1, s3: {bucket: "backup", path:
"year2019/day42"}})
```



PBM

PBM

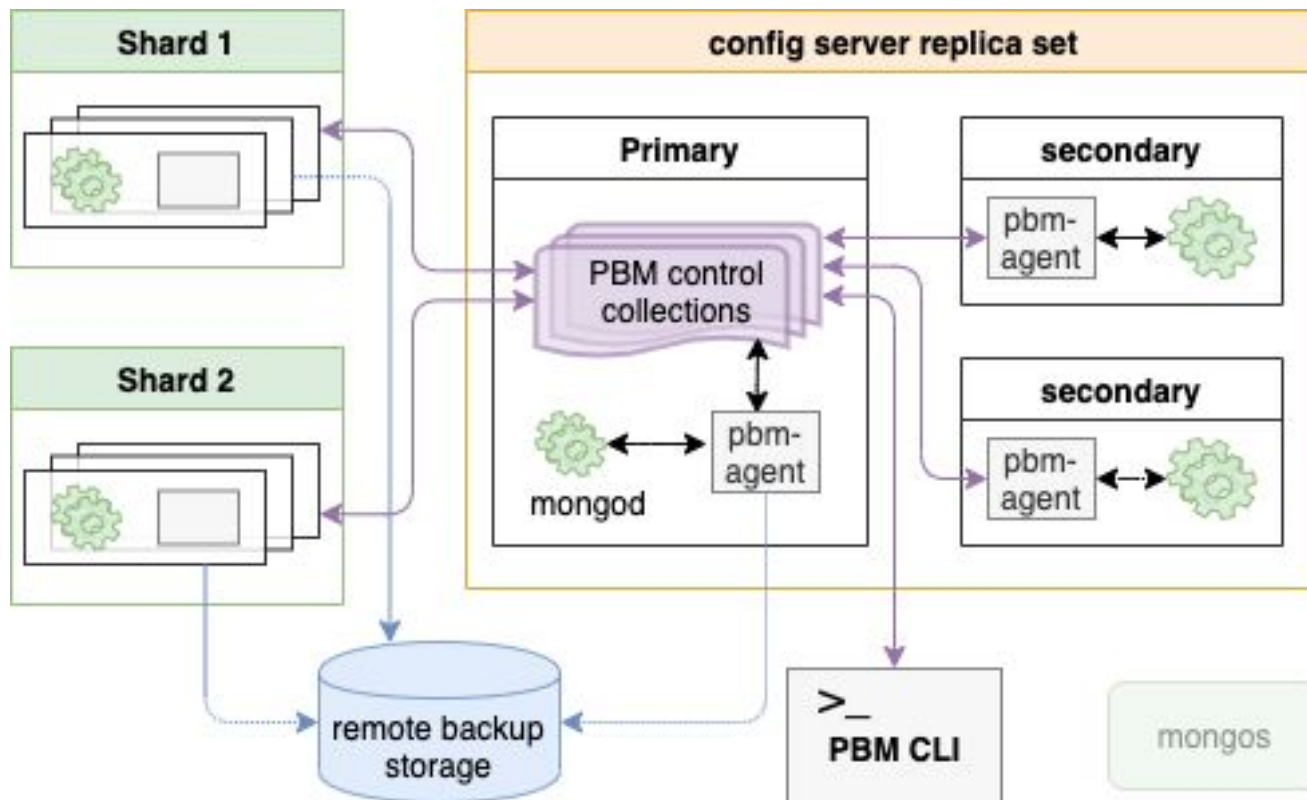
- Low-impact solution for taking consistent backups of MongoDB Sharded Cluster and Replica Sets
- Open Source
- Supports PSMDB and MongoDB Community v3.6 or higher
- Doesn't work in standalone mode. Require replication enabled
- Requires an agent running on each node
- Included in our Kubernetes Operator for MongoDB

<https://docs.percona.com/percona-backup-mongodb/index.html>

PBM Features

- Logical Backup: the tool is basically a wrapper for `mongodump` and `mongorestore`
- Simple command-line management utility
- Integrated with MongoDB authentication
- Physical Backup
- Point-in-time recovery
- Selective Backup
- Incremental Backup
- Compression
- Use S3 compatible storage or volume mounted on all nodes (e.g. NFS)

Architecture



Basic operations

- Perform a backup

```
$ pbm backup
```

- List available backups on the storage

```
$ pbm list
```

```
Backup snapshots:
```

```
2025-03-10T10:44:52Z <logical> [restore_to_time: 2025-03-10T10:44:56Z]
```

```
2025-03-10T10:49:20Z <physical> [restore_to_time: 2025-03-10T10:49:23Z]
```

```
2025-03-10T10:50:22Z <incremental> [restore_to_time: 2025-03-10T10:50:25Z]
```

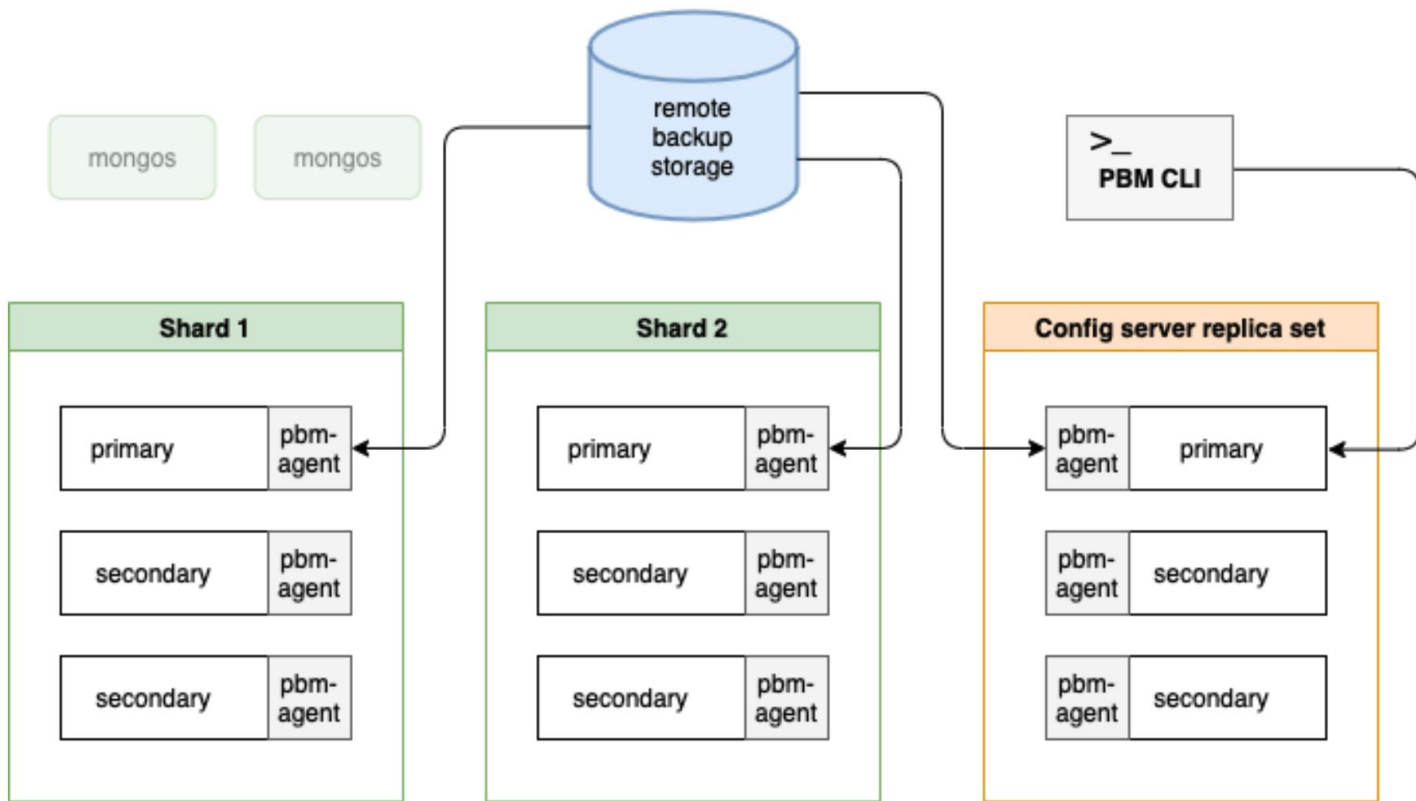
- Restore a backup

```
$ pbm restore 2025-03-10T10:49:20Z
```

Logical Backup and Restore

- Logical backup is the done by using `mongodump`
- The *pbm-agent* connects to the database
 - If possible, a Secondary node is selected for running `mongodump`
 - The files are eventually compressed on the fly and sent to the remote storage
- Logical restore is done by running `mongorestore`
 - Dump files are retrieved from the remote storage
 - Mongorestore is executed on each Primary

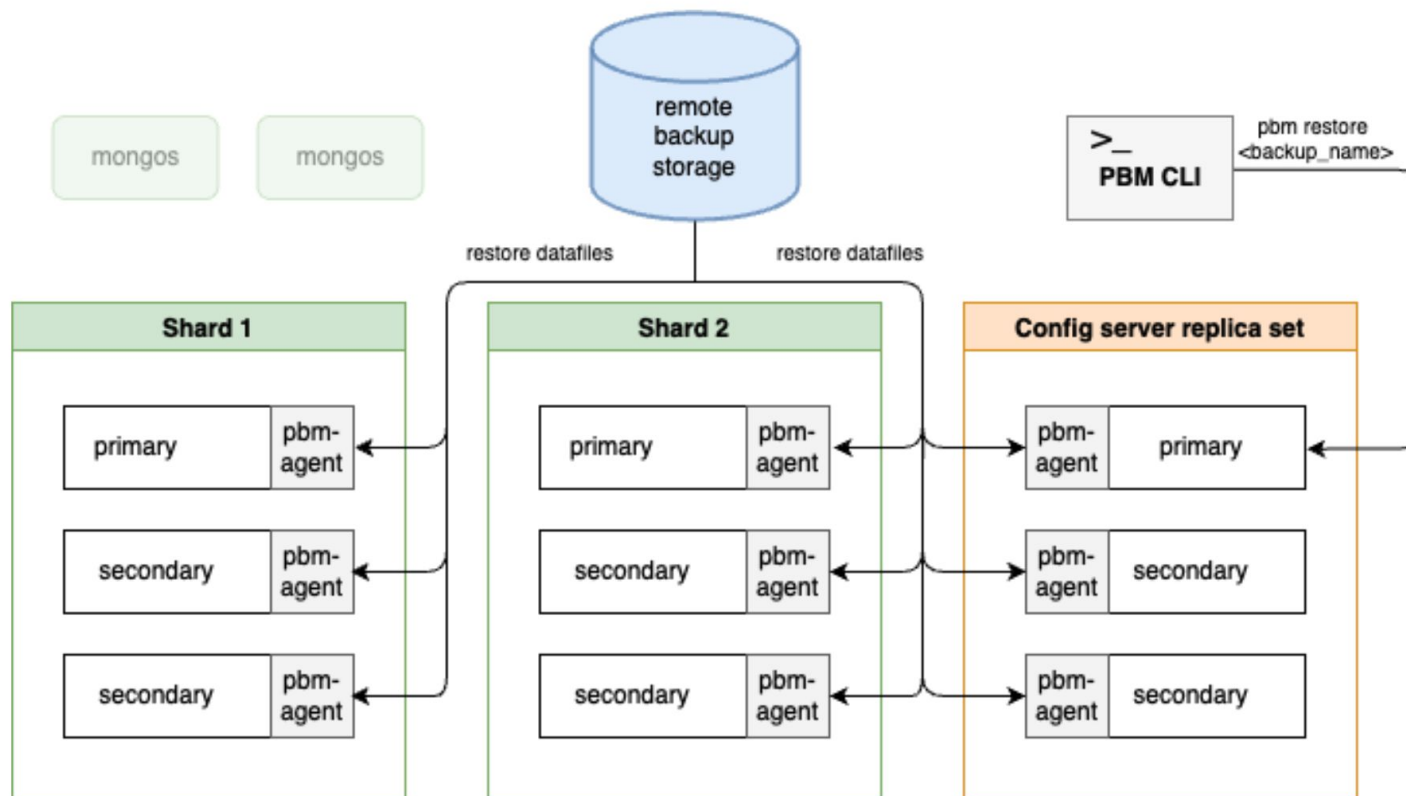
Logical Restore Schema



Physical Backup and Restore

- Available for PSMDB starting from versions 4.2.15–16, 4.4.6–8, 5.0 and higher
- Physical backups rely on the WiredTiger \$backupCursor functionality
- Physical files are copied from the PSMDB *dbPath* data directory into the remote storage
 - These include data files, journal, index files, etc.
- Physical Restore is the reverse process: *pbm-agents* shut down all `mongod` nodes, clean up the *dbPath* and copy the files from the remote storage
 - All nodes are involved in the restore in parallel

Physical Restore Schema



Point-in-time Recovery

- The process includes restoring the data from a backup snapshot and replaying all events that occurred to this data up to a specified time from oplog slices
- The feature is not enabled by default. You need to explicitly enabling it

```
$ pbm config --set pitr.enabled=true
```

- When enabled, the oplog is copied in slices to the remote storage
- Slices cover a 10-minute span of oplog events
- Before restoring, you need to disable it



Conclusions

Conclusions

- PBM is the most mature open source tool on the market for managing both Logical and Physical backups and restores with ease
- Plan your backup policy according to your needs
- The topology of the cluster matters
- Several TB of data should not scare that much. Rely on Physical approach or add shards if you can for Logical
- If you use PBM or not, remember Percona provides support



Questions?





Thank You

corrado.pandiani@percona.com