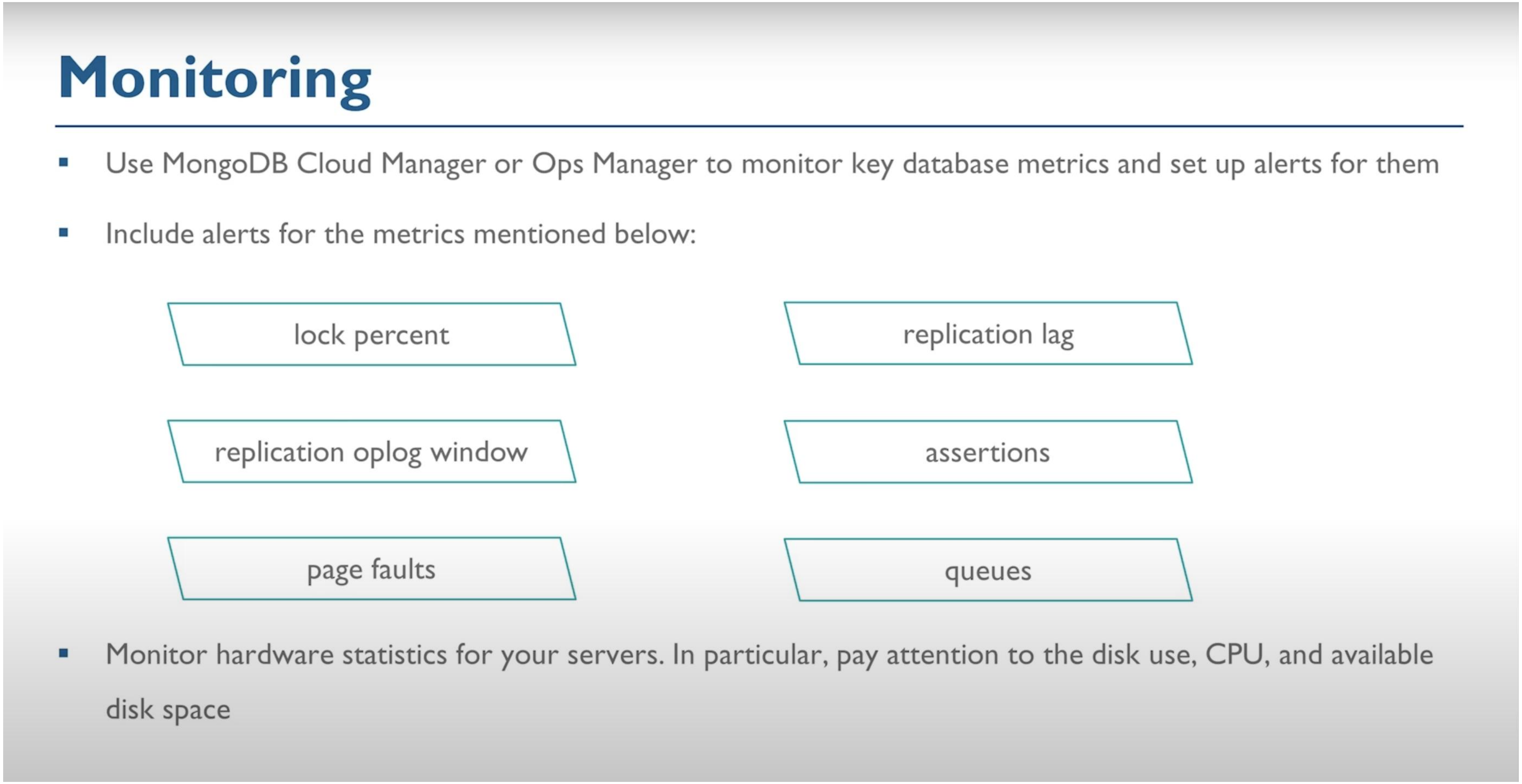


Monitoring

- Use MongoDB Cloud Manager or Ops Manager to monitor key database metrics and set up alerts for them
- Include alerts for the metrics mentioned below:

lock percent

replication lag

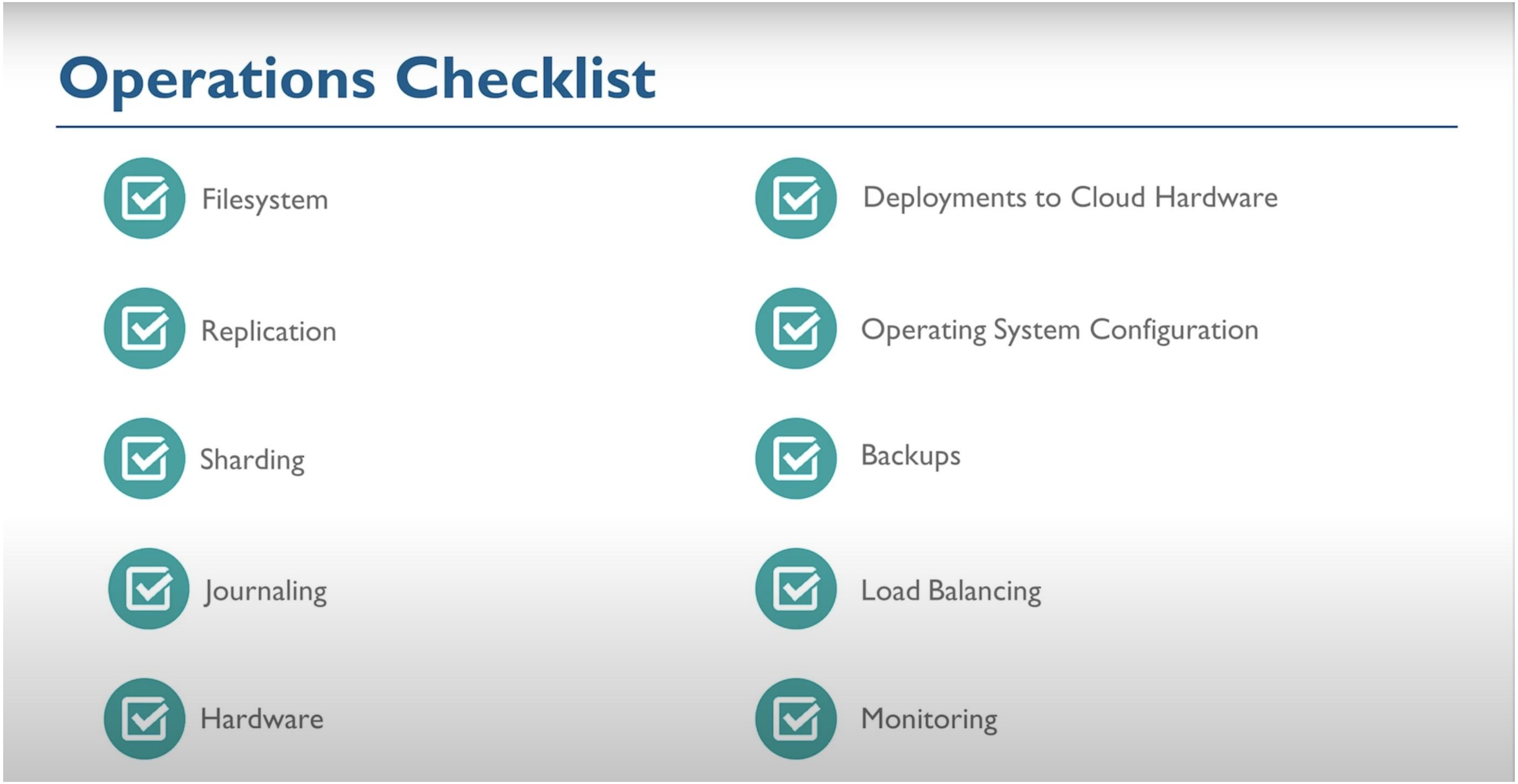
replication oplog window

assertions

page faults

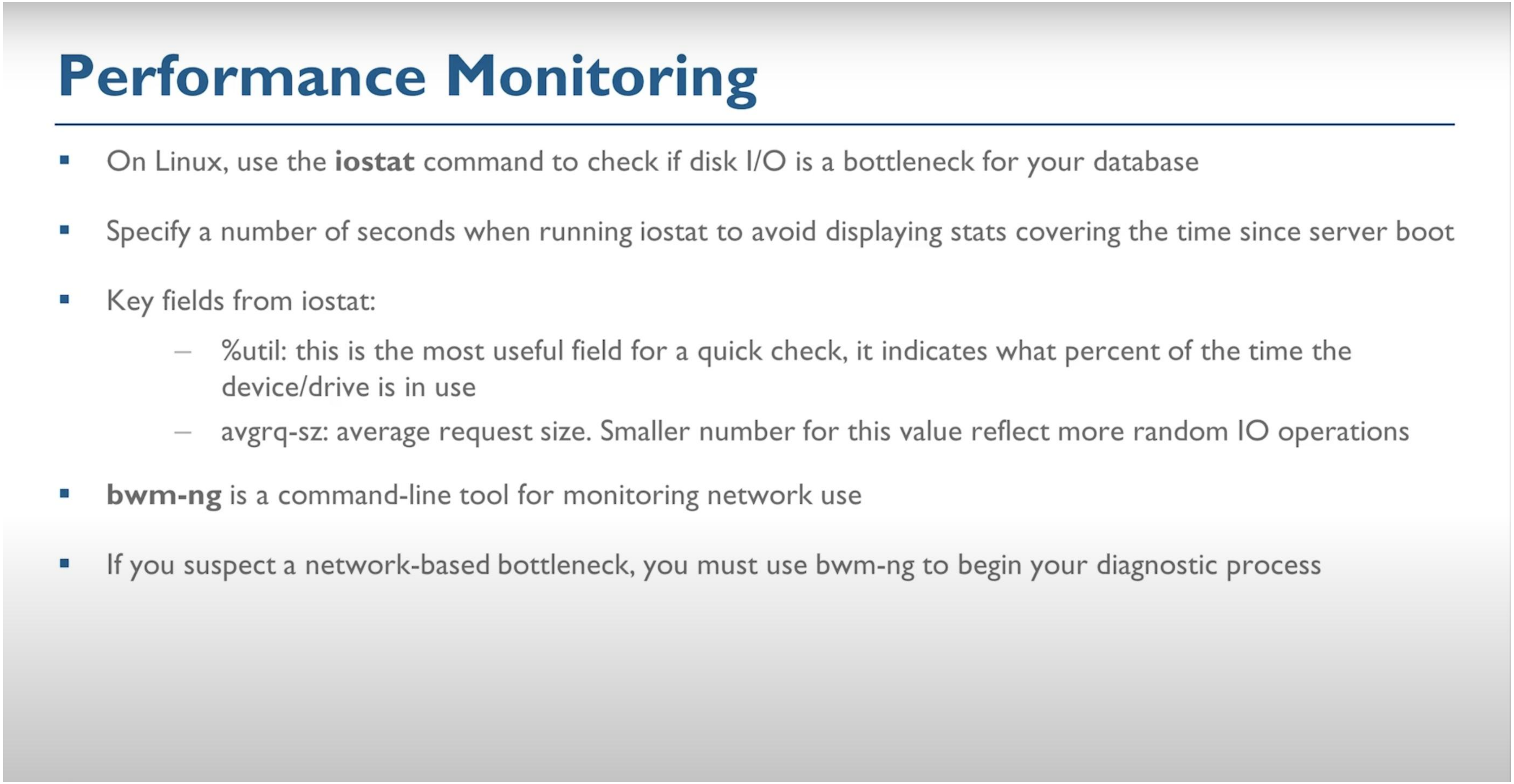
queues

- Monitor hardware statistics for your servers. In particular, pay attention to the disk use, CPU, and available disk space



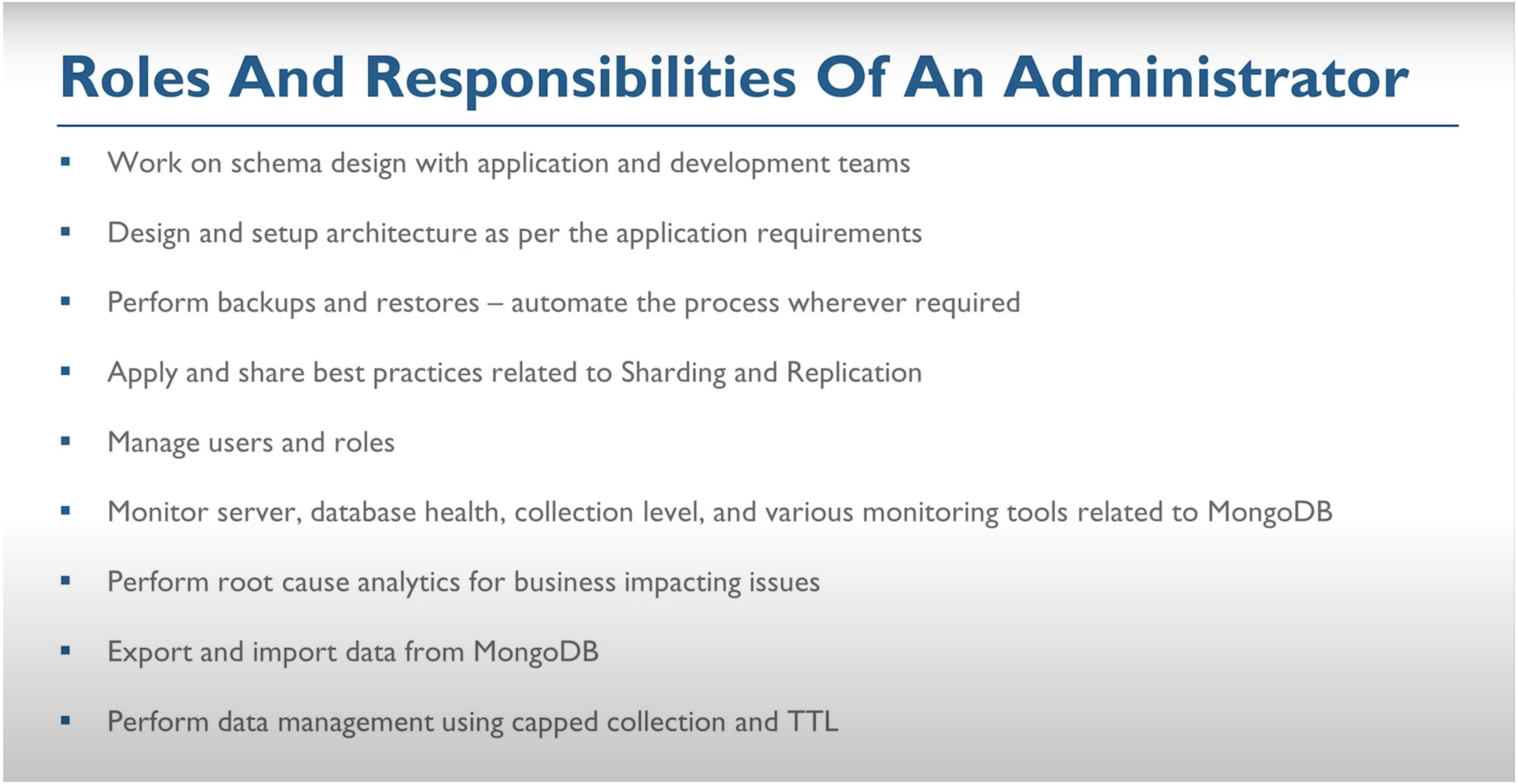
Operations Checklist

- ☑ Filesystem
- ☑ Replication
- ☑ Sharding
- ☑ Journaling
- ☑ Hardware
- ☑ Deployments to Cloud Hardware
- ☑ Operating System Configuration
- ☑ Backups
- ☑ Load Balancing
- ☑ Monitoring



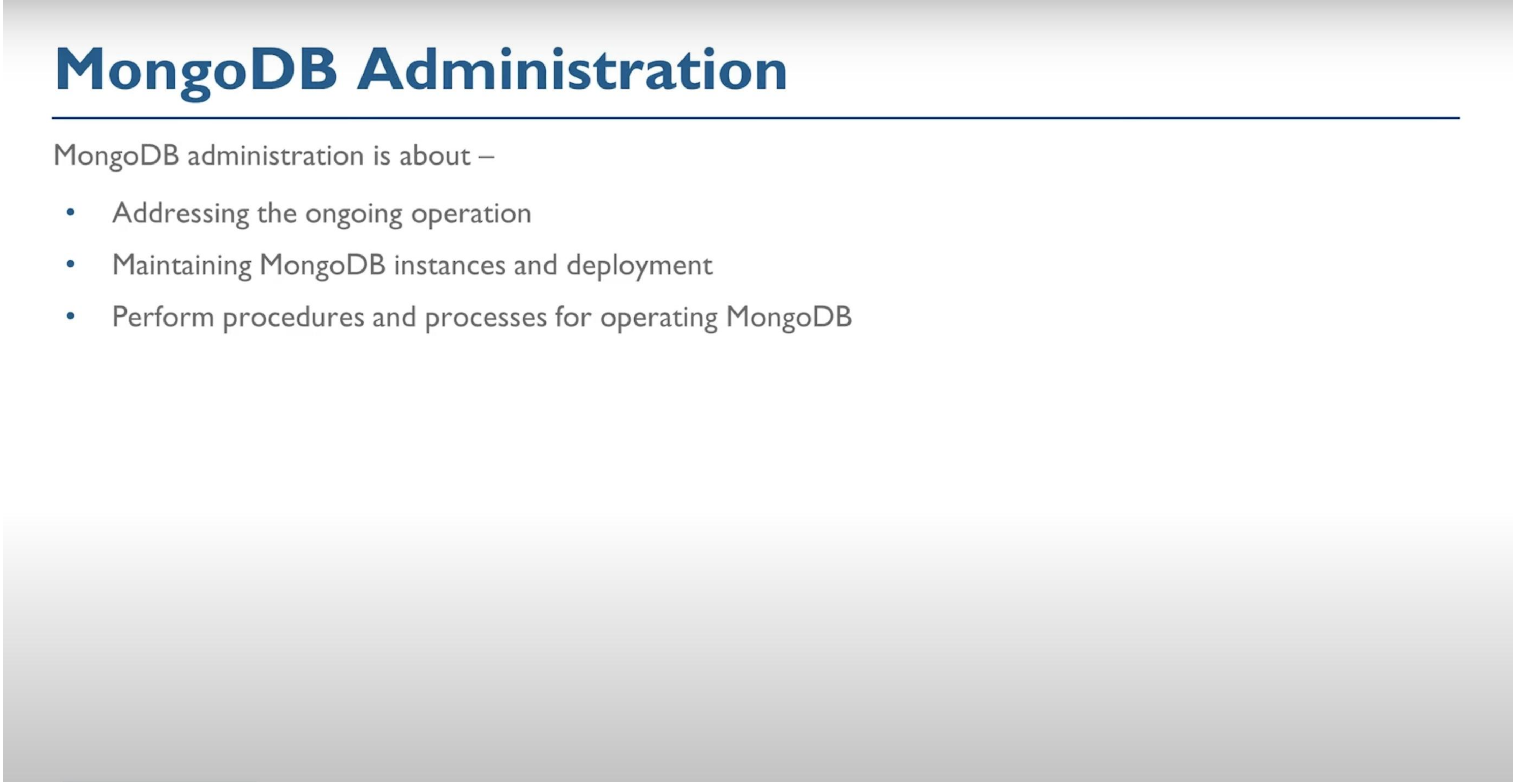
Performance Monitoring

- On Linux, use the **iostat** command to check if disk I/O is a bottleneck for your database
- Specify a number of seconds when running iostat to avoid displaying stats covering the time since server boot
- Key fields from iostat:
  - %util: this is the most useful field for a quick check, it indicates what percent of the time the device/drive is in use
  - avgrq-sz: average request size. Smaller number for this value reflect more random IO operations
- **bwm-ng** is a command-line tool for monitoring network use
- If you suspect a network-based bottleneck, you must use bwm-ng to begin your diagnostic process



Roles And Responsibilities Of An Administrator

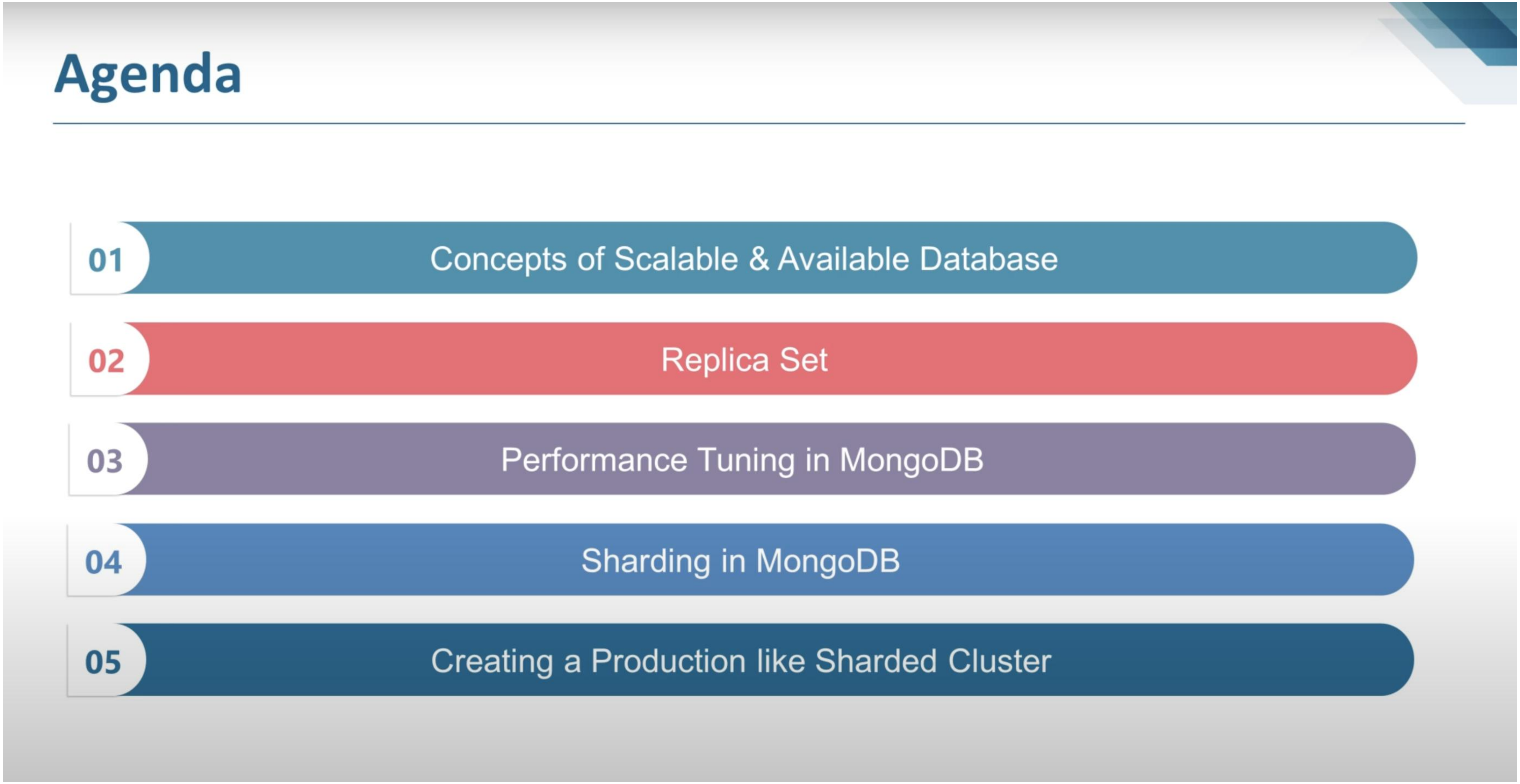
- Work on schema design with application and development teams
- Design and setup architecture as per the application requirements
- Perform backups and restores – automate the process wherever required
- Apply and share best practices related to Sharding and Replication
- Manage users and roles
- Monitor server, database health, collection level, and various monitoring tools related to MongoDB
- Perform root cause analytics for business impacting issues
- Export and import data from MongoDB
- Perform data management using capped collection and TTL



MongoDB Administration

MongoDB administration is about –

- Addressing the ongoing operation
- Maintaining MongoDB instances and deployment
- Perform procedures and processes for operating MongoDB



Agenda

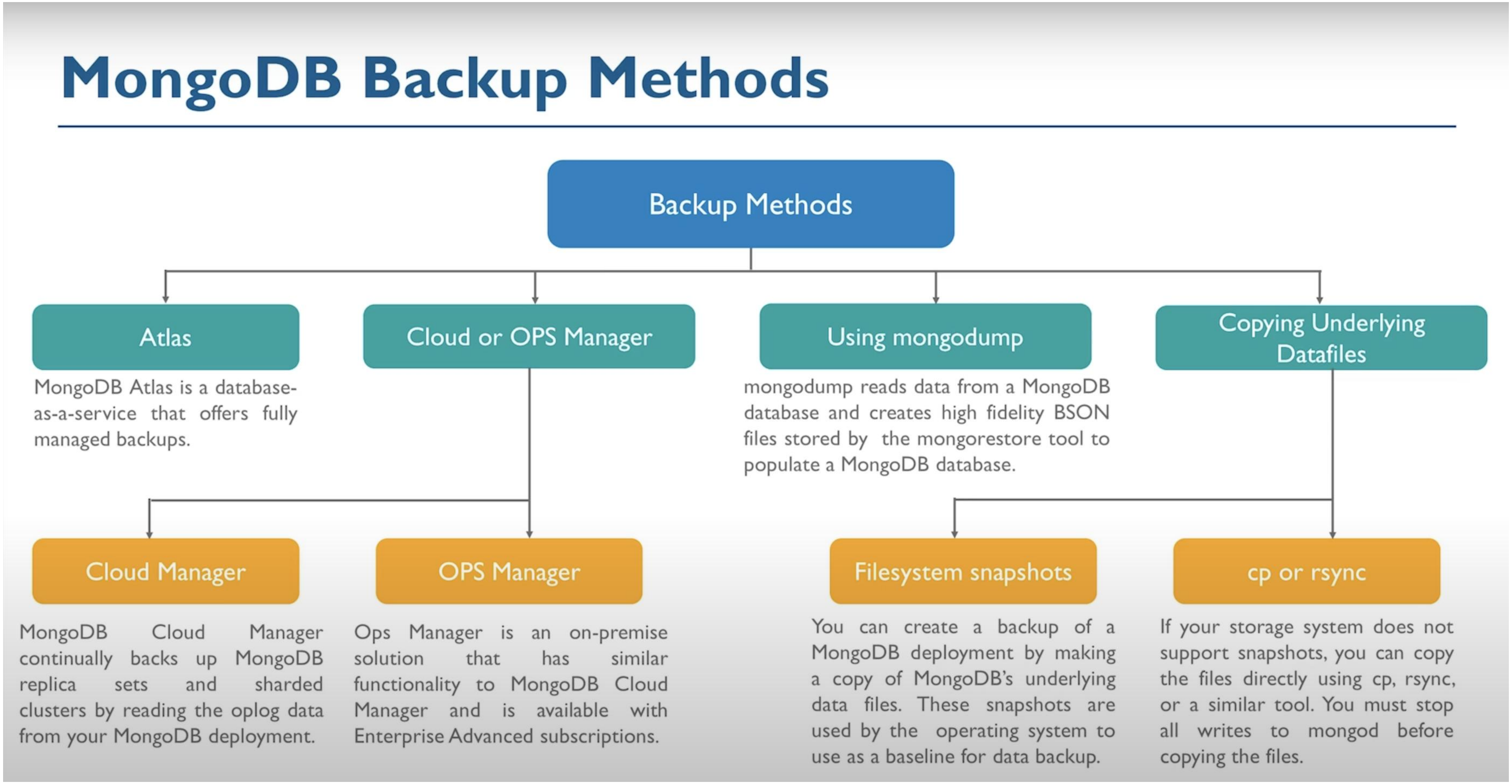
01 Concepts of Scalable & Available Database

02 Replica Set

03 Performance Tuning in MongoDB

04 Sharding in MongoDB

05 Creating a Production like Sharded Cluster



MongoDB Backup Methods

Backup Methods

Atlas

MongoDB Atlas is a database-as-a-service that offers fully managed backups.

Cloud or OPS Manager

Cloud Manager

MongoDB Cloud Manager continually backs up MongoDB replica sets and sharded clusters by reading the oplog data from your MongoDB deployment.

OPS Manager

Ops Manager is an on-premise solution that has similar functionality to MongoDB Cloud Manager and is available with Enterprise Advanced subscriptions.

Using mongodump

mongodump reads data from a MongoDB database and creates high fidelity BSON files stored by the mongorestore tool to populate a MongoDB database.

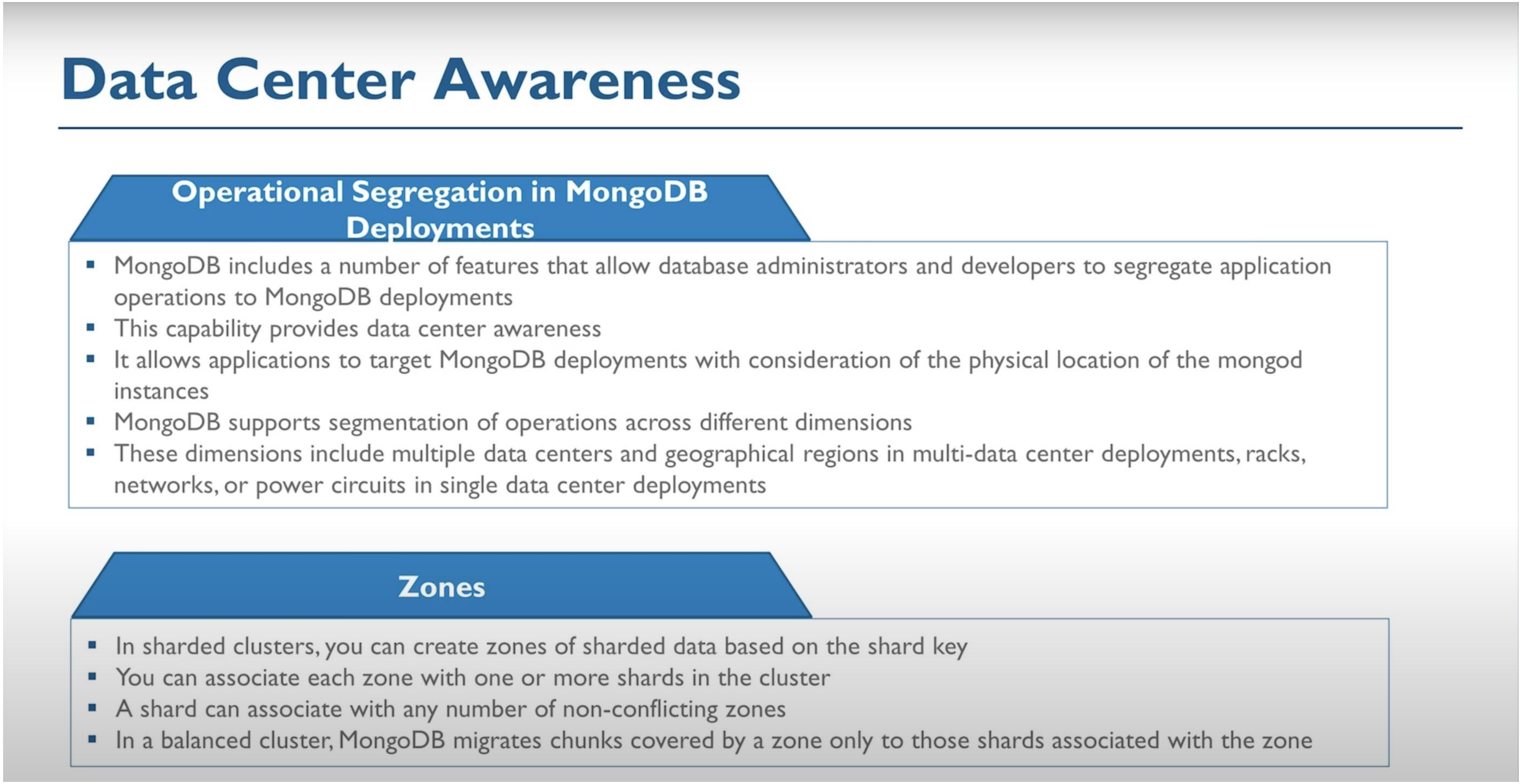
Copying Underlying Datafiles

Filesystem snapshots

You can create a backup of a MongoDB deployment by making a copy of MongoDB's underlying data files. These snapshots are used by the operating system to use as a baseline for data backup.

cp or rsync

If your storage system does not support snapshots, you can copy the files directly using cp, rsync, or a similar tool. You must stop all writes to mongod before copying the files.



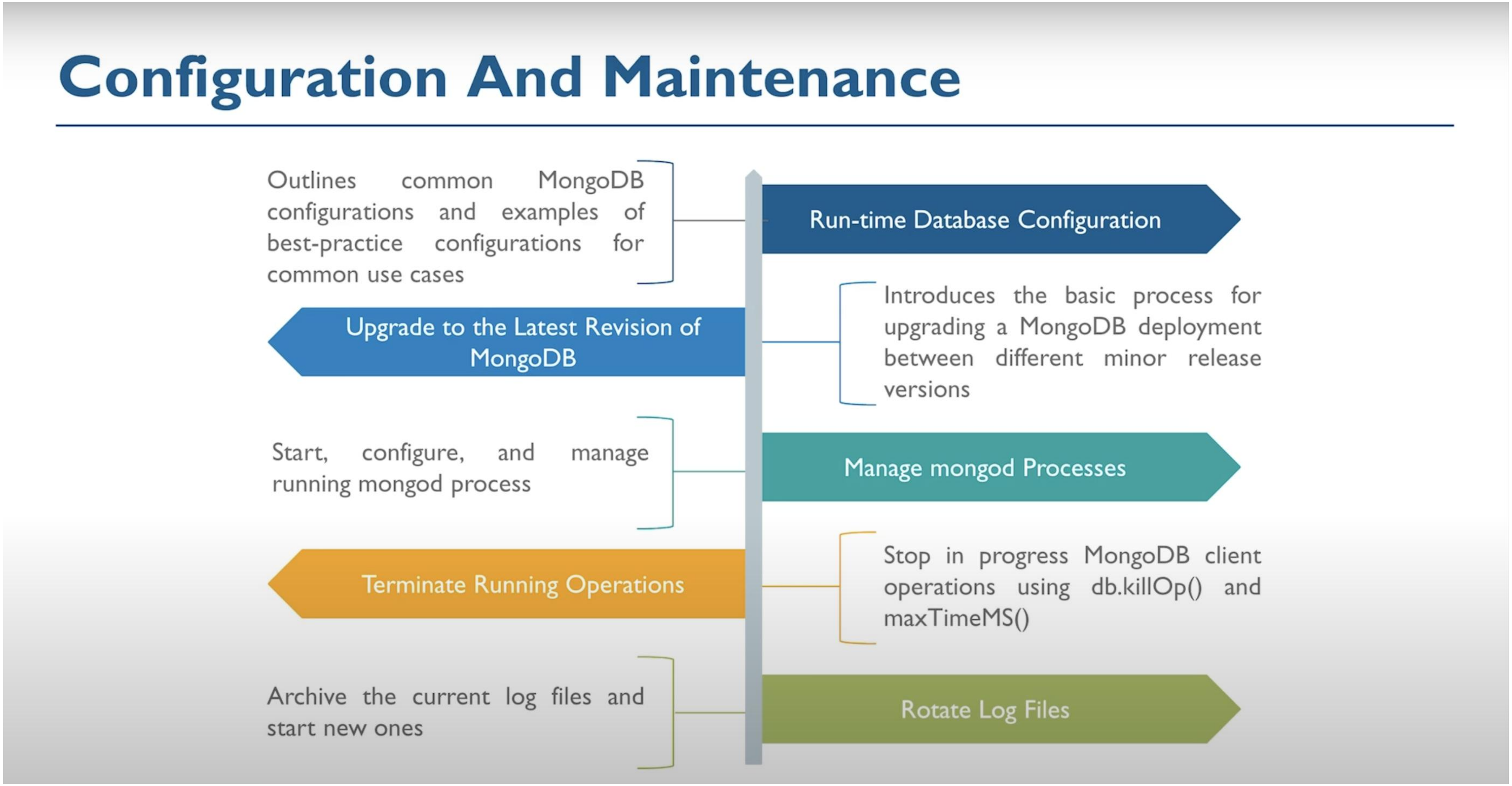
Data Center Awareness

Operational Segregation in MongoDB Deployments

- MongoDB includes a number of features that allow database administrators and developers to segregate application operations to MongoDB deployments
- This capability provides data center awareness
- It allows applications to target MongoDB deployments with consideration of the physical location of the mongod instances
- MongoDB supports segmentation of operations across different dimensions
- These dimensions include multiple data centers and geographical regions in multi-data center deployments, racks, networks, or power circuits in single data center deployments

Zones

- In sharded clusters, you can create zones of sharded data based on the shard key
- You can associate each zone with one or more shards in the cluster
- A shard can associate with any number of non-conflicting zones
- In a balanced cluster, MongoDB migrates chunks covered by a zone only to those shards associated with the zone



Configuration And Maintenance

Run-time Database Configuration

Outlines common MongoDB configurations and examples of best-practice configurations for common use cases

Upgrade to the Latest Revision of MongoDB

Introduces the basic process for upgrading a MongoDB deployment between different minor release versions

Manage mongod Processes

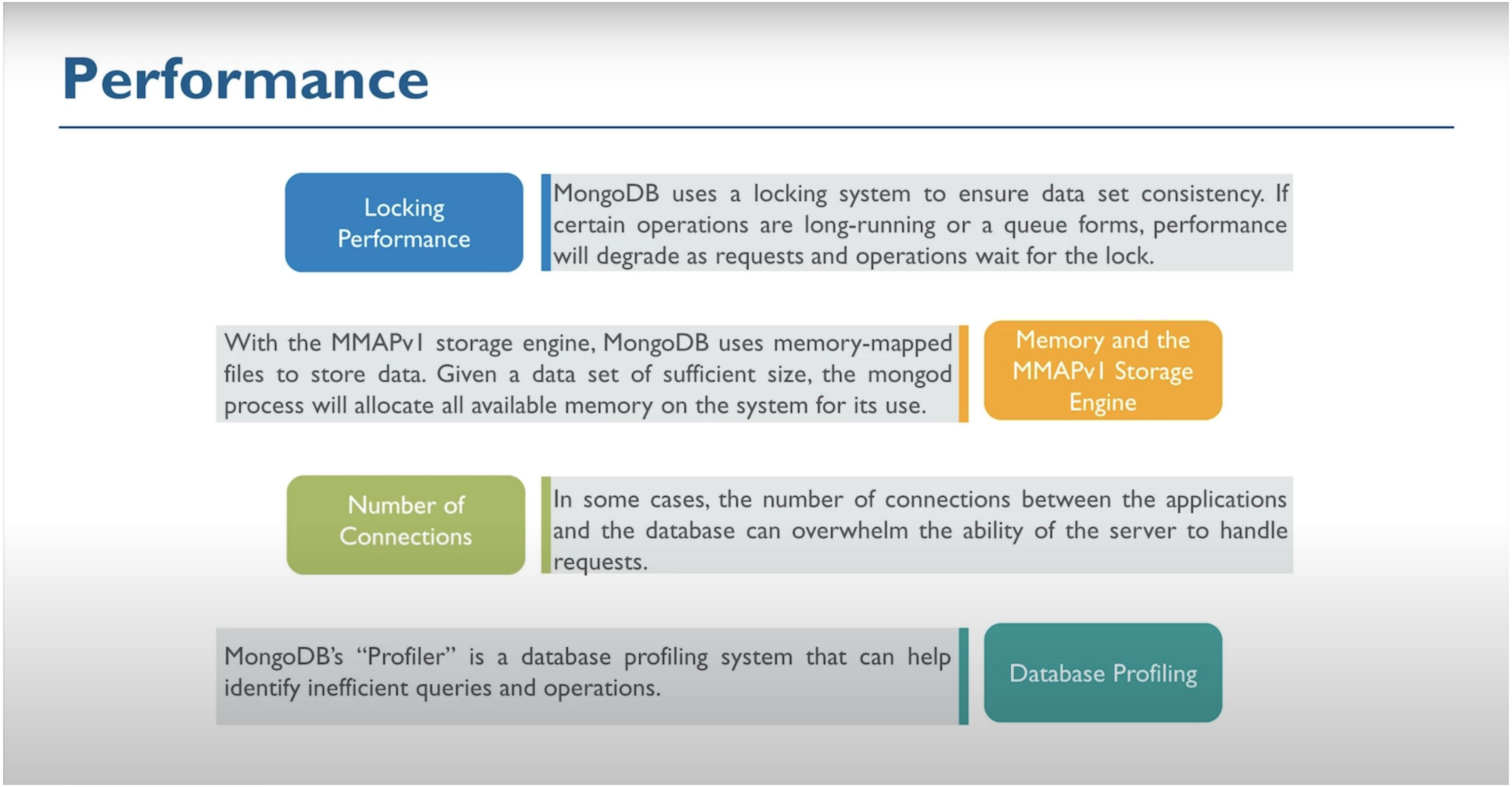
Start, configure, and manage running mongod process

Terminate Running Operations

Stop in progress MongoDB client operations using db.killOp() and maxTimeMS()

Rotate Log Files

Archive the current log files and start new ones



Performance

Locking Performance

MongoDB uses a locking system to ensure data set consistency. If certain operations are long-running or a queue forms, performance will degrade as requests and operations wait for the lock.

Memory and the MMAPv1 Storage Engine

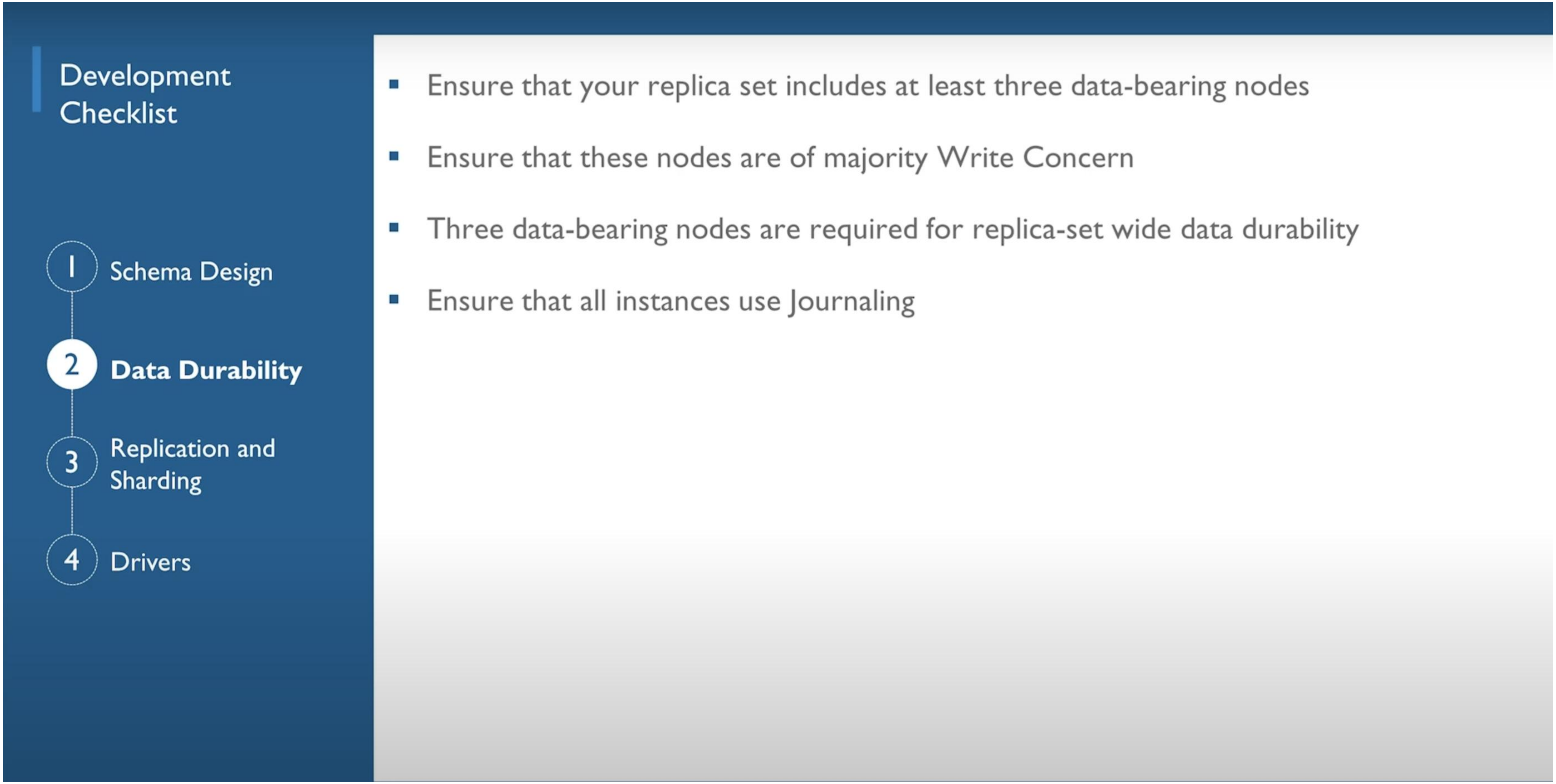
With the MMAPv1 storage engine, MongoDB uses memory-mapped files to store data. Given a data set of sufficient size, the mongod process will allocate all available memory on the system for its use.

Number of Connections

In some cases, the number of connections between the applications and the database can overwhelm the ability of the server to handle requests.

Database Profiling

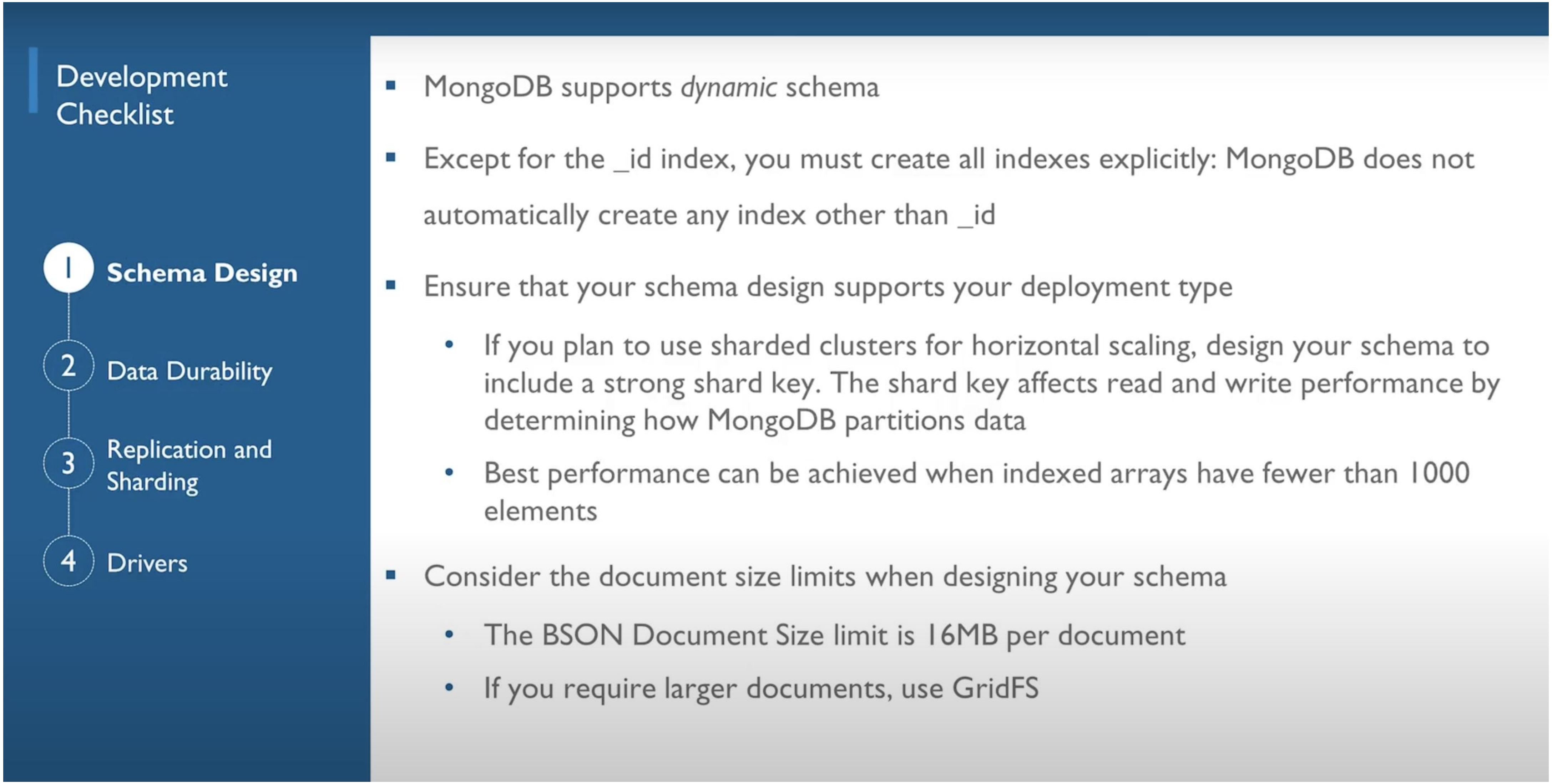
MongoDB's "Profiler" is a database profiling system that can help identify inefficient queries and operations.



Development Checklist

1. Schema Design
2. **Data Durability**
3. Replication and Sharding
4. Drivers

- Ensure that your replica set includes at least three data-bearing nodes
- Ensure that these nodes are of majority Write Concern
- Three data-bearing nodes are required for replica-set wide data durability
- Ensure that all instances use Journaling



Development Checklist

1. **Schema Design**
2. Data Durability
3. Replication and Sharding
4. Drivers

- MongoDB supports *dynamic* schema
- Except for the _id index, you must create all indexes explicitly: MongoDB does not automatically create any index other than _id
- Ensure that your schema design supports your deployment type
  - If you plan to use sharded clusters for horizontal scaling, design your schema to include a strong shard key. The shard key affects read and write performance by determining how MongoDB partitions data
  - Best performance can be achieved when indexed arrays have fewer than 1000 elements
- Consider the document size limits when designing your schema
  - The BSON Document Size limit is 16MB per document
  - If you require larger documents, use GridFS