MongoDB Security, Monitoring & Backup (Mongodump)

- One of the key concepts in MongoDB is the management of databases. Important aspects such as security, backup, access to databases are all important concepts when it comes to database administration.

MongoDB Security Overview

- MongoDB has the ability to define security mechanisms to databases. By default one wouldn't want everyone to have an open access to every database in MongoDB, hence the requirement for having some sort of security mechanism in MongoDB is important.

Following are the best practices when implementing security in databases

1. Enable access control – Create users so that all applications and users are enforced to have some sort of authentication mechanism when accessing databases on MongoDB.
2. Configure role based access control – Sometimes there can be a logical grouping of permissions which may be required, which can be clubbed in roles. Users can then be assigned to these roles.
3. Try to configure MongoDB to use some sort of encryption protocol such as TLS or SSL. These protocols can be used to encrypt the traffic which flows between the client and the mongo DB environment.
4. Configure auditing – Administrators normally need to know who is doing what, which helps in analyzing problems later on. The best way is to enable auditing in MongoDB.
5. Run MongDB server instance with a separate user id which has access to the required resources on the server environment.

Mongodb Backup Procedures - mongodump

When working with MongDB it is important to always ensure a backup procedure is in place in case the data within MongoDB gets corrupted for any reason.

Below are the backup mechanisms available from within MongoDB

1. **Backup by Copying Underlying Data Files** – This is probably the easiest mechanism , all that needs to be done is to copy the data files on which MongoDB resides and copy it to another location which ideally should be another server.
2. **Backup a Database with mongodump** - The mongodump tool reads data from a MongoDB database and creates high fidelity BSON files. What needs to be taken into consideration is that if the data set is large in volume, then mongodump can be very resource intensive, so then to mitigate this problem, the utility should be run on a secondary server.
3. **MongoDB Cloud Manager Backup** - MongoDB Cloud Manager continually backs up MongoDB replica sets and sharded clusters by reading the oplog data from the MongoDB environment. MongoDB Cloud Manager can create a point in time recovery by storing oplog data so that it can create a restore at any moment in time for a particular replica set or sharded cluster.

Mongodb Monitoring

Monitoring is one of the most critical administrative activities in MongoDB. This is because you can be more proactive by monitoring the environment for possible issues which could crop up.

Below are some of the examples for implementing monitoring

1. mongostat will tell you how many time database operations such as insert, query, update, delete, etc. actually occur on the server. This will give a good idea on how much the load the server is handling and will indicate whether you need additional resources on the server or maybe additional servers to distribute the load.
2. mongotop tracks and reports the current read and write activity of a MongoDB instance, and reports these statistics on a per collection basis.

3. MongoDB provides a web interface that exposes diagnostic and monitoring information in a simple web page. One can browse to the below url on your local server to open the web administration utility http://localhost:28017
4. The serverStatus command, or db.serverStatus() from the shell, returns an overview of the status of the database, with details on the disk usage, memory use, connections established to the MongoDB environment, etc.

MongoDB Indexing and Performance Considerations

1. Indexes are very important in any database and can be used to improve the efficiency of search queries in MongoDB. If you are continually performing searches in your document, then it's better to add indexes on the fields of the document that are used in the search criteria.
2. Try to always limit the number of query results returned. Suppose you have 2 field names in a document, but you just want to see 2 fields from the document. Then make sure your query only targets to display the 2 fields you require and not all of the fields.
3. If you want to view certain field values, then only use those fields in the query. Do not query for all the fields in the collection if they are not required.

Summary:

- It is very important to implement security in databases to ensure that the data in the database is kept safe.
- Users can be created in the database with the createUser command. Specific roles can be assigned to the users to give them specific permissions on the database itself.
- Administrators can be added for all databases are for just specific databases. This is achieved by giving either the userAdmin or the userAdminAnyDatabase role.
- Always backup your MongoDB environment so that in that in the case of any disaster, the data would be easily recoverable.
- Always monitor your MongoDB environment for being more proactive and see issues before they occur.