Mongo DBA

Contents

[Chapter 1: Authentication, authorization; creating users and granting permissions 1](#_Toc456113784)

[*1.* *How to enable authentication for mongod server?* 1](#_Toc456113785)

[*2.* *What if you created no user but start server with --auth mode?* 1](#_Toc456113786)

[*3.* *How to create users and grant permissions to them?* 1](#_Toc456113787)

[*4.* *Update user of a database?* 3](#_Toc456113788)

[*5.* *Get all users of a database* 3](#_Toc456113789)

[*6.* *Delete a user* 3](#_Toc456113790)

[*7.* *Authentication in Replica mode?????? (Need research more)* 4](#_Toc456113791)

[Chapter 2: Managing Servers 5](#_Toc456113792)

[Chapter 3: Upgrade MongoDB server 5](#_Toc456113793)

[Chapter 4: Aggregate framework 5](#_Toc456113794)

[Chapter 5: Indexing and managing indexes 5](#_Toc456113795)

[Chapter 6: GridFS 5](#_Toc456113796)

[Chapter 7: Optimization 5](#_Toc456113797)

[Chapter 8: Backing Up and Restoring 5](#_Toc456113798)

[Chapter 9: Replication 5](#_Toc456113799)

[Chapter 10: Sharding 5](#_Toc456113800)

[Chapter 11: Misc (balancers, config servers, …) 5](#_Toc456113801)

# Chapter 1: Authentication, authorization; creating users and granting permissions

1. *How to enable authentication for mongod server?*

Restart server with option **--auth** (or **auth=true** in config file)

**mongod --auth**

1. *What if you created no user but start server with --auth mode?*

You cannot run any command

1. *How to create users and grant permissions to them?*

Firstly, stop your mongod server, and then start the server without --auth parameter.

**mongod --port <Your Port>**

Then, use admin database.

**use admin**

Then create user:

**db.createUser(**

**{**

**user : "admin",**

**pwd : "pass",**

**roles: [**

**{**

**role : "readWrite",**

**db : "admin"**

**},**

**{**

**role: "userAdminAnyDatabase",**

**db : "admin"**

**}**

**]**

**});**

A success message will be returned if everything is okay.

In MongoDB use role-base-authentication. New users are granted permissions by ROLEs.

Table of roles available in MongoDB:

|  |  |  |
| --- | --- | --- |
| **Role** | **Description** | **VD** |
| read | Allows user to read from a given database | **{role: “read”, db: “neosend”}** |
| readWrite | Allows user to read/write access to given database | **{role: “readWrite”, db: “neosend”}** |
| dbAdmin | Perform administrative functions within the given dbs:   * Creating/deleting indexes * Viewing statistics * Accessing **system.profile** collections | **{role: “dbAdmin”, db: “neosend”}** |
| userAdmin | Allows user to write to the **system.users** collection:   * Create/delete users for this db * Administer the users for this db | **{role: “userAdmin”, db: “neosend”}** |
| dbOwner | Combines permissions of **readWrite**, **dbAdmin** and **userAdmin** roles | **{role: “dbOwner”, db: “neosend”}** |
| clusterManager | Only for **admin** database. Manage the cluster, not data. | **aa** |
| clusterMonitor | For **admin** database only. Access to stats and commands that collect stats (for the **admin** database) |  |
| hostManager | For **admin** database only. Manage and monitor host level services |  |
| clusterAdmin | For **admin** database only. Full admin access to all functions relate to sharding and replica set.  Combination of **clusterManager, clusterMonitor** and **hostManager** roles. | **Read more at** [**http://docs.mongodb.org/manual/reference/user-privileges/#clusterAdmin%23clusterAdmin**](http://docs.mongodb.org/manual/reference/user-privileges/#clusterAdmin%23clusterAdmin) |
| backup | For **admin** database only.  Have permissions to back up the entire system. |  |
| restore | For **admin** database only. Have rights to restore entire system from backups |  |
| readAnyDatabase | Available for **admin** database only.  Have permissions to read all databases | **Read more at** [**http://docs.mongodb.org/manual/reference/user-privileges/#readAnyDatabase%23readAnyDatabase**](http://docs.mongodb.org/manual/reference/user-privileges/#readAnyDatabase%23readAnyDatabase) |
| readWriteAnyDatabase | For **admin** database only. Have permissions to read/write to all databases | **Read more at** [**http://docs.mongodb.org/manual/reference/user-privileges/#readWriteAnyDatabase%23readWriteAnyDatabase**](http://docs.mongodb.org/manual/reference/user-privileges/#readWriteAnyDatabase%23readWriteAnyDatabase) |
| userAdminAnyDatabase | For **admin** database only. Have userAdmin permission on all databases | **Read more at** [**http://docs.mongodb.org/manual/reference/user-privileges/#userAdminAnyDatabase%23userAdminAnyDatabase**](http://docs.mongodb.org/manual/reference/user-privileges/#userAdminAnyDatabase%23userAdminAnyDatabase) |
| dbAdminAnyDatabase | For **admin** database only. Have dbAdmin permission on all databases | **Read more at** [**http://docs.mongodb.org/manual/reference/user-privileges/#dbAdminAnyDatabase%23dbAdminAnyDatabase**](http://docs.mongodb.org/manual/reference/user-privileges/#dbAdminAnyDatabase%23dbAdminAnyDatabase) |

When you are using a certain database, you have to **auth(u, p)** and have permission to create user in order to create new user for the database.

1. *Update user of a database?*

You need to have at least dbAdmin role to use this method:

**db.updateUser( "foo", { roles: [ { role : "dbAdmin", db : "blog" } ] })**

1. *Get all users of a database*

**db.getUsers()**

1. *Delete a user*

**db.removeUser(“username”)**

1. *Authentication in Replica mode?????? (Need research more)*

# Chapter 2: Managing Servers

(prepare linux environment for practicing)

1. Starting and stopping mongodb

* MongoDB server is started with the **mongod** executable
* Some param for **mongod** command:
  + --dbpath: path to the directory containing data files
  + --port : as the name implies. Default port is 27017
  + --fork: Fork the server process, run MongoDB as a daemon (not for MS Windows)
  + --logpath: log file to let MongoDB output logs (consider append mode by --logappend)
  + --config: specify a file for configuration instead of using command line params
* File based configs: put all that params in a .config file, and when start **mongod** pass the path of the file to --**config** param. With params like --auth or --fork, in the config file, put something like this: auth=true or fork=true
* Safely stop mongodb: run the **shutdown** command as below

**> use admin**

**> db.shutdownServer()**

When use this command in a primary node of a replica set, it will step down the primary and wait for a secondary being elected before shutting down the server completely. I no secondary is available, the primary won’t shut down.

In that case, we can force the primary to shut down by using this command:

**> db.adminCommand({“shutdown”: 1, “force”: true})**

**mongod** then will wait for any running operations or file preallocations to finish, close all connections and flush all data to disk before shutting down.

1. Monitoring MongoDB
2. Deploying MongoDB
3. Backing up MongoDB
4. Upgrade MongoDB

# Chapter 3: Aggregate framework

# Chapter 4: Indexing and managing indexes

# Chapter 5: GridFS

# Chapter 6: Optimization

# Chapter 7: Replication

# Chapter 8: Sharding

# Chapter 9: Misc (balancers, config servers, …)

a