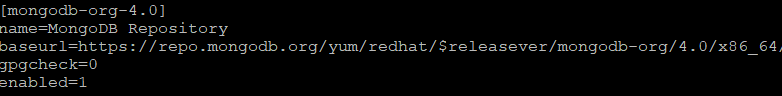
**MONGO DB RUN-Book**

1. **Mongo-db Installation:**

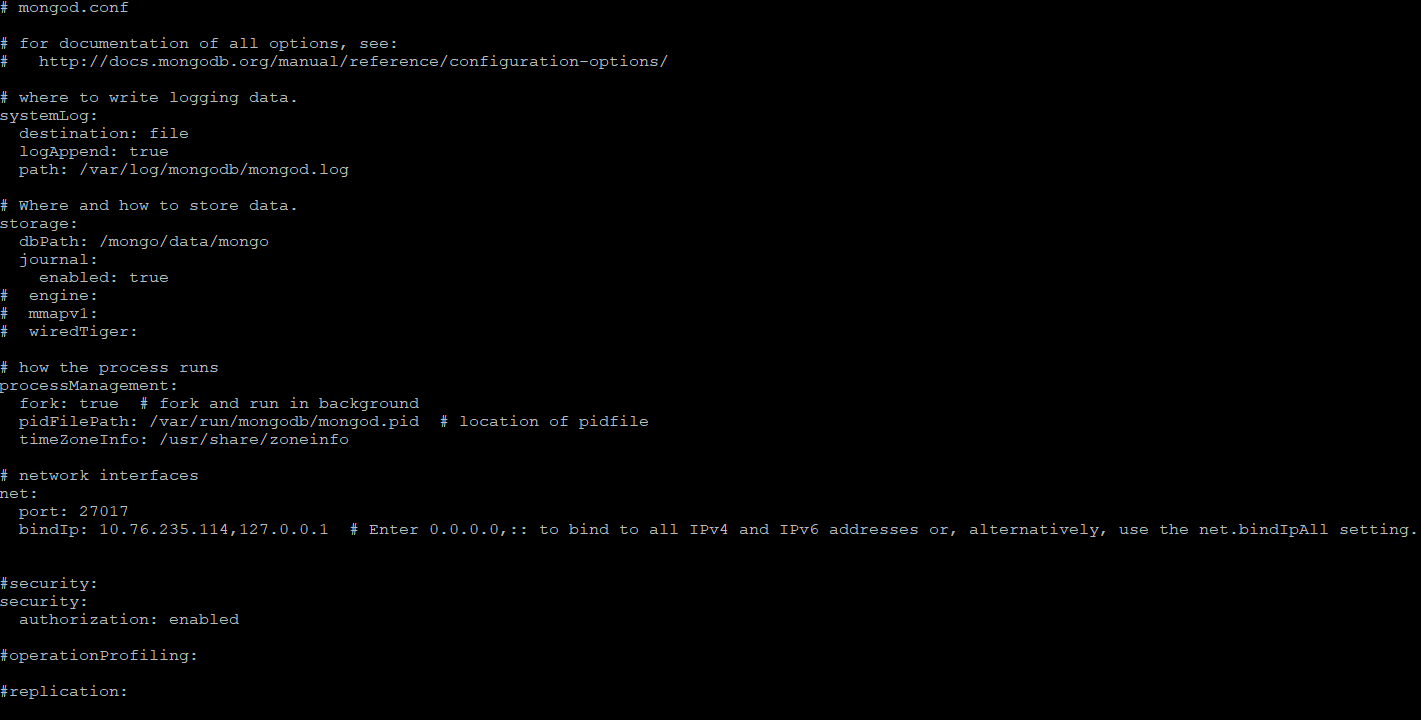
The Installation of mongo DB is pretty straight forward. You have to create a file with the name “mongodb-org-4.0.repo” in “/etc/yum.repos.d” directory. Please use the below configuration.



Then please execute yum install “yum install mongodb-org”

1. Mongo-db Configuration:

Once the installation with Yum is successful, it will by default create a configuration file known as “mongod.conf” under /etc folder.



Please edit the dbPath, bindIP and port (if you needed) as per the screen shot above. You can enable security at later point of time.

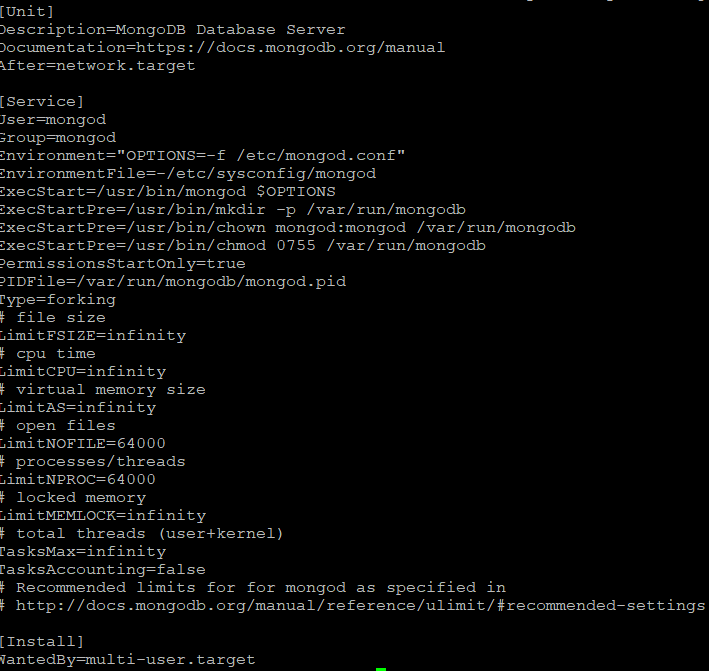
After this change, check is there any symbolic link created under “/etc/system/system” directory, which is UNIT file for systemctl process start.

If not, it will be in the below location “/usr/lib/systemd/system” with the name “mongod.service”.

Cross check the mongod.service file with the below one, and change it accordingly(if needed)

Please install openJDK1.8 before you start the mongo service. Use the below command for the same:

“yum install java-1.8.0-openjdk-devel”



1. Mongo-db Process and Data directories

Go to /mongo and create a “data” directory and give the permissions as mongod:mongod.

You have to make sure that you should have mongod user and group should be present before you do any of this.

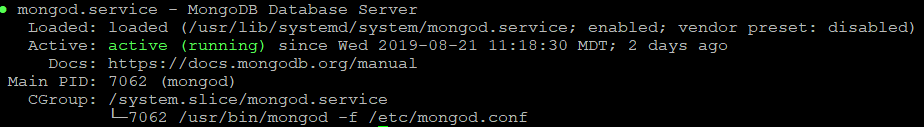
Use “sudo systemctl daemon-reload” command to load the Unit file.

Use “sudo systemctl enable mongod.service”

To start the process use “sudo systemctl start mongod”

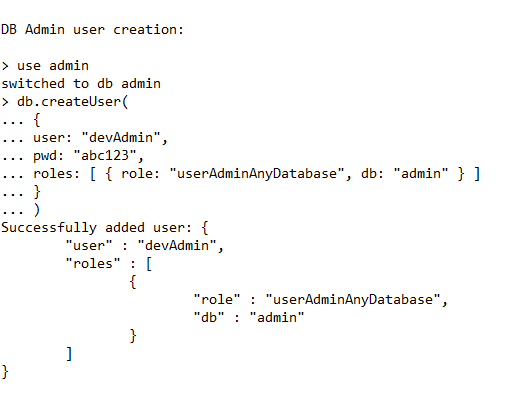
To check the status use “sudo systemctl status mongod”

To stop the process use “sudo systemctl stop mongod”



1. Admin User creation & Enabling security

We have created a user with the name “devAdmin” in the “admin” db.



In the mongodb.conf file under /etc, there is a security enabling tab that you need to enable. And you have to restart the mongo service.

After that you can use the below command to authenticate against mongo secured shell.

**mongo --port 27017 -u "devAdmin" -p "abc123" --authenticationDatabase "admin**"

1. Creating Data Bases other than Admin db

As per the application team requests we have created the below data bases.

**use scheduled\_jobs**

**use sgs\_apscheduler**

**use sgs\_jam\_history**

**use sgs\_whitelist**

**use sgs\_messaging**

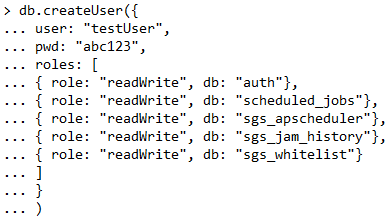
**use test**

1. Creating application User at AUTH DB

You have to create an application user apart from the Admin user, so that all the databases can use that id to read/write/update the data bases as required.

We have created an application user with the name “testUser” and provided “readWrite” access role.

So we have created a data base with the name “AUTH”, and created a user under it. And we have provided that access in such a way it can have readWrite access on the remaining data bases. With this, we are controlling that authentication against empty data base which is “AUTH”. So that there won’t be any interaction with “Admin” data base.





Authenticate the data base after enabling security and creating a user in AUTH db:

[ctummalachervu@indevl-stamdb01 mongodb-sgs-dev-dump-08222019-1030]$ mongo -u testUser -p abc123 --authenticationDatabase auth

MongoDB shell version v4.0.12

connecting to: mongodb://127.0.0.1:27017/?authSource=auth&gssapiServiceName=mongodb

Implicit session: session { "id" : UUID("604aa7c0-340b-475d-8e79-64fe4f204f9f") }

MongoDB server version: 4.0.12

1. Assign application user to other dbs

To check the USers info, you have to authenticate to Admin, then switch to "auth" db, then execute db.getUsers() command.

mongo -u devAdmin -p abc123 --authenticationDatabase admin

> use auth

switched to db auth

> db.getUsers()

[

{

"\_id" : "auth.testUser",

"userId" : UUID("140dbb54-753d-49b7-980a-dc74ca9e29fd"),

"user" : "testUser",

"db" : "auth",

"roles" : [

{

"role" : "readWrite",

"db" : "auth"

},

{

"role" : "readWrite",

"db" : "scheduled\_jobs"

},

{

"role" : "readWrite",

"db" : "sgs\_apscheduler"

},

{

"role" : "readWrite",

"db" : "sgs\_jam\_history"

},

{

"role" : "readWrite",

"db" : "sgs\_whitelist"

}

],

"mechanisms" : [

"SCRAM-SHA-1",

"SCRAM-SHA-256"

]

}

]

1. Migrating the Data base/Restoring from Dump

We have received the data dump from application team and we have used “mongorestore” command to migrate the data dump.

[ctummalachervu@indevl-stamdb01 mongodb-sgs-dev-dump-08222019-1030]$ mongorestore --host 10.76.235.114:27017 --username testUser --password abc123 --authenticationDatabase auth --db scheduled\_jobs /tmp/mongodb-sgs-dev-dump-08222019-1030/scheduled\_jobs

2019-08-22T11:03:32.311-0600 the --db and --collection args should only be used when restoring from a BSON file. Other uses are deprecated and will not exist in the future; use --nsInclude instead

2019-08-22T11:03:32.311-0600 building a list of collections to restore from /tmp/mongodb-sgs-dev-dump-08222019-1030/scheduled\_jobs dir

2019-08-22T11:03:32.333-0600 reading metadata for scheduled\_jobs.jobs from /tmp/mongodb-sgs-dev-dump-08222019-1030/scheduled\_jobs/jobs.metadata.json

2019-08-22T11:03:32.513-0600 restoring scheduled\_jobs.jobs from /tmp/mongodb-sgs-dev-dump-08222019-1030/scheduled\_jobs/jobs.bson

2019-08-22T11:03:32.545-0600 no indexes to restore

2019-08-22T11:03:32.545-0600 finished restoring scheduled\_jobs.jobs (114 documents)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

[ctummalachervu@indevl-stamdb01 mongodb-sgs-dev-dump-08222019-1030]$ mongorestore --host 10.76.235.114:27017 --username testUser --password abc123 --authenticationDatabase auth --db test /tmp/mongodb-sgs-dev-dump-08222019-1030/Test

2019-08-22T11:54:15.845-0600 the --db and --collection args should only be used when restoring from a BSON file. Other uses are deprecated and will not exist in the future; use --nsInclude instead

2019-08-22T11:54:15.845-0600 building a list of collections to restore from /tmp/mongodb-sgs-dev-dump-08222019-1030/Test dir

2019-08-22T11:54:15.868-0600 reading metadata for test.Test from /tmp/mongodb-sgs-dev-dump-08222019-1030/Test/Test.metadata.json

2019-08-22T11:54:15.989-0600 restoring test.Test from /tmp/mongodb-sgs-dev-dump-08222019-1030/Test/Test.bson

2019-08-22T11:54:16.015-0600 no indexes to restore

2019-08-22T11:54:16.015-0600 finished restoring test.Test (1 document)

2019-08-22T11:54:16.015-0600 done

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

[ctummalachervu@indevl-stamdb01 mongodb-sgs-dev-dump-08222019-1030]$ mongorestore --host 10.76.235.114:27017 --username testUser --password abc123 --authenticationDatabase auth --db graylog /tmp/mongodb-sgs-dev-dump-08222019-1030/graylog

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

[ctummalachervu@indevl-stamdb01 mongodb-sgs-dev-dump-08222019-1030]$ mongorestore --host 10.76.235.114:27017 --username testUser --password abc123 --authenticationDatabase auth --db sgs\_messaging /tmp/mongodb-sgs-dev-dump-08222019-1030/sgs\_messaging

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

If there is any need to Create a New DB and assing an existing a user to it :

First switch to "AUTH" where the User exists. Which means you have to create the user in AUTH.

Then grant the roles like below:

db.grantRolesToUser( "testUser", [ { role: "readWrite", db: "sgs\_messaging"}])

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

The Adminuser(devAdmin) has only permission to create and manage database users. If you try to read /write in the data base with Adminuser, mongodb will return error.

So we need to create additional users to read//write on the data base.