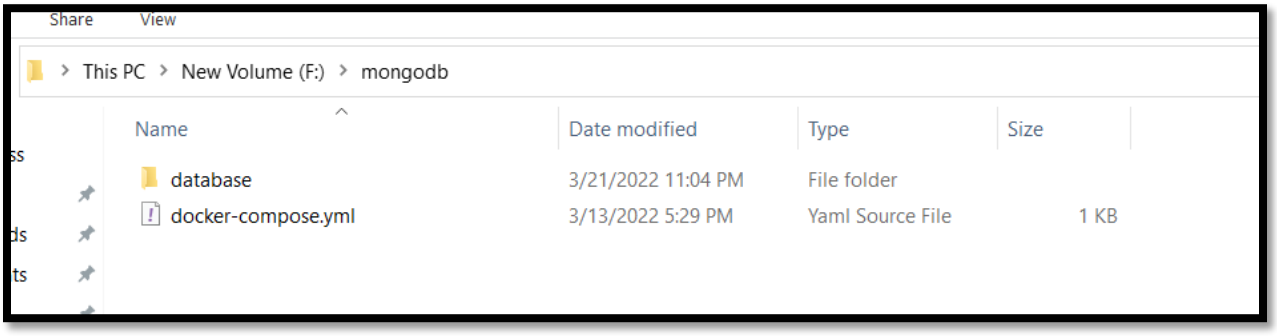
Dockerized MongoDB – Assignment

Submitted by: Muhammad Amin Ghias

ERP: 25366

- Created a new directory F:/mongodb/database
- The JSON file is stored/saved in database folder
- Docker-compose.yml in mongodb folder



- The docker-compose.yml file was modified

Following changes were made

```
volumes:
    - 'F:/mongodb/database:/data/db'
```

- As F drive is used

```
deploy:
    resources:
      limits:
        memory: 8000M
      reservations:
        memory: 8000M
```

- The resources are mentioned with high limits of 8000M as the JSON file size is large

Changes are show in picture

docker-compose.yml

F: > mongodb > docker-compose.yml > ...

docker-compose.yml (compose-spec.json)

```yaml
version: "3.8"

services:
  mongodb:
    image: mongo
    container_name: mongodb2

    deploy:
      resources:
        limits:
          memory: 8000M
        reservations:
          memory: 8000M
    environment:
      - PUID=1000
      - PGID=1000
    volumes:
      - 'F:/mongodb/database:/data/db'
    ports:
      - 27017:27017
    restart: "unless-stopped"

```

- Creating the container package using docker compose byrunning command in directory **F :\mongodb** directory

```
F:\mongodb>docker-compose up -d
[+] Running 1/0
 - Container mongodb2  Running                                                    0.0s

F:\mongodb>
```

- The new docker container is created and running

```
F:\mongodb>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED      STATUS       PORTS                      NAMES
3c0e975d3a4c   mongo     "docker-entrypoint.s…"   8 days ago   Up 2 hours   0.0.0.0:27017->27017/tcp   mongodb2
```

```
F:\mongodb>docker logs mongodb2
```

- Executing new created container to open its bash

```
F:\mongodb>docker exec -it mongodb2 bash
root@3c0e975d3a4c:/#
```

Importing the data set

The data set used is taken from Kaggle and its **JSON size is 2.38 GB**

https://www.kaggle.com/datasets/bitrook/us-county-historical demographics?select=us_county_demographics.json

About Dataset

US County & Zipcode Historical Demographics

Easily lookup US historical demographics by county FIPS or zipcode in seconds with this file containing over 5,901 different columns including:

- *Lat/Long
- *Boundaries
- *State FIPS
- *Population from 2010-2019
- *Death Rate from 2010-2019
- *Unemployment from 2001-2020
- *Education from 1970-2019
- *Gender and Age Population

```
root@15f569cdba34:/data/db# mongoimport --db MY_DB --collection users --drop --jsonArray --batchSize 1 --file ./us_county_demographics.json
2022-03-13T13:30:41.954+0000    connected to: mongodb://localhost/
2022-03-13T13:30:41.956+0000    dropping: MY_DB.users
2022-03-13T13:30:44.955+0000    [.......................] MY_DB.users  15.0MB/2.28GB (0.6%)
2022-03-13T13:30:47.954+0000    [.......................] MY_DB.users  24.5MB/2.28GB (1.1%)
2022-03-13T13:30:50.954+0000    [.......................] MY_DB.users  31.7MB/2.28GB (1.4%)
2022-03-13T13:30:53.954+0000    [.......................] MY_DB.users  35.2MB/2.28GB (1.5%)
2022-03-13T13:30:56.954+0000    [.......................] MY_DB.users  41.9MB/2.28GB (1.8%)
2022-03-13T13:30:59.954+0000    [.......................] MY_DB.users  46.4MB/2.28GB (2.0%)
2022-03-13T13:31:02.955+0000    [.......................] MY_DB.users  52.1MB/2.28GB (2.2%)
2022-03-13T13:31:05.955+0000    [.......................] MY_DB.users  58.3MB/2.28GB (2.5%)
```

- Starting the mongo server

```
root@3c0e975d3a4c:/# mongo
MongoDB shell version v5.0.6
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=
Implicit session: session { "id" : UUID("42eb84dd-43fa-4d45-9d86-e3eab22e0ef9") }
MongoDB server version: 5.0.6
================
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo" shell has been de
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
================
---
The server generated these startup warnings when booting:
        2022-03-21T16:16:31.747+00:00: Access control is not enabled for the data
        2022-03-21T16:16:31.747+00:00: /sys/kernel/mm/transparent_hugepage/enable
---
---
        Enable MongoDB's free cloud-based monitoring service, which will then rec
        metrics about your deployment (disk utilization, CPU, operation statistic

        The monitoring data will be available on a MongoDB website with a unique
        and anyone you share the URL with. MongoDB may use this information to ma
        improvements and to suggest MongoDB products and deployment options to yo

        To enable free monitoring, run the following command: db.enableFreeMonito
        To permanently disable this reminder, run the following command: db.disab
---
>
```

1. Viewing the database available and selecting db and viewing collections

```
> show dbs
MY_DB   0.294GB
admin   0.000GB
config  0.000GB
local   0.000GB
> use MY_DB
switched to db MY_DB
> show collections
users
> db.users.findOne()
{
        "_id" : ObjectId("622df20257e9d4cbb9f63b46"),
        "zipcode_type" : "Standard",
        "major_city" : "Verbena",
        "post_office_city" : "Verbena, AL",
        "county" : "Chilton County",
        "state" : "AL",
        "lat" : 32.8,
        "lng" : -86.5,
        "timezone" : "Central",
        "radius_in_miles" : 10,
        "land_area_in_sqmi" : 81.8,
        "water_area_in_sqmi" : 1.64,
        "zipcode" : "36091",
        "county_fips" : 1001,
        "state_fips" : 1,
        "state_name" : "Alabama",
        "county_sumlev" : 50,
        "county_region" : 3,
        "county_division" : 6,
        "county_region_name" : "South",
        "county_division_name" : "East South Central",
        "population" : {
                "rural_urban" : {
                        "continumm_code_2003" : 2,
```

2. Finding the documents in our collection

```
> db.users.count()
52002
```

3. Finding the name of county where major_city is ‘Verbana’

```
> db.users.find({"major_city":"Verbena"},{county:1,major_city:1})
{ "_id" : ObjectId("622df20257e9d4cbb9f63b46"), "major_city" : "Verbena", "county" : "Chilton County" }
{ "_id" : ObjectId("622df20757e9d4cbb9f63bd2"), "major_city" : "Verbena", "county" : "Chilton County" }
>
```

4. Finding the zipcode where major_city is ‘Verbana’

```
> db.users.find({major_city:'Verbena'},{zipcode:1,major_city:1})
{ "_id" : ObjectId("622df20257e9d4cbb9f63b46"), "major_city" : "Verbena", "zipcode" : "36091" }
{ "_id" : ObjectId("622df20757e9d4cbb9f63bd2"), "major_city" : "Verbena", "zipcode" : "36091" }
>
```

5. Showing the stats of the database

```
> db.stats()
{
        "db" : "MY_DB",
        "collections" : 1,
        "views" : 0,
        "objects" : 52002,
        "avgObjSize" : 21824.974943271412,
        "dataSize" : 1134942347,
        "storageSize" : 315097088,
        "indexes" : 1,
        "indexSize" : 1024000,
        "totalSize" : 316121088,
        "scaleFactor" : 1,
        "fsUsedSize" : 40118435840,
        "fsTotalSize" : 471858147328,
        "ok" : 1
}
```

6. Finding stats of our collection

```
> db.users.stats()
{
        "ns" : "MY_DB.users",
        "size" : 1134942347,
        "count" : 52002,
        "avgObjSize" : 21824,
        "storageSize" : 315097088,
        "freeStorageSize" : 253952,
        "capped" : false,
        "wiredTiger" : {
                "metadata" : {
                        "formatVersion" : 1
                },
                "creationString" : "access_pattern_hint=none,al
mestamp=none,write_timestamp=off),block_allocation=best,block_c
d=,name=),exclusive=false,extractor=,format=btree,huffman_key=,
false),internal_item_max=0,internal_key_max=0,internal_key_trun
f_value_max=64MB,log=(enabled=true),lsm=(auto_throttle=true,blo
5GB,chunk_size=10MB,merge_custom=(prefix=,start_generation=0,su
x=0,prefix_compression=false,prefix_compression_min=4,readonly=
ge=(auth_token=,bucket=,bucket_prefix=,cache_directory=,local_r
```

7. Showing the name of current database we are present in

```
> db.getName()
MY_DB
>
```

8. Finding the number of people whose education is less than high school diploma and major city is ‘Verbana’

```
> db.users.find({major_city:'Verbena'},{"education.less_than_a_high_school_diploma":1,major_city:1})
{ "_id" : ObjectId("622df20257e9d4cbb9f63b46"), "major_city" : "Verbena", "education" : { "less_than_a_
 "2000" : 5872, "2015-19" : 4291 } } }
{ "_id" : ObjectId("622df20757e9d4cbb9f63bd2"), "major_city" : "Verbena", "education" : { "less_than_a_
, "2000" : 8757, "2015-19" : 5474 } } }
```

9. Finding the number of people whose education is less than high school diploma in 1970 and major city is ‘Verbana’

```
> db.users.find({major_city:'Verbena'},{"education.less_than_a_high_school_diploma.1970":1,major_city:1})
{ "_id" : ObjectId("622df20257e9d4cbb9f63b46"), "major_city" : "Verbena", "education" : { "less_than_a_high_school_diploma" : { "1970" : 6611 } } }
{ "_id" : ObjectId("622df20757e9d4cbb9f63bd2"), "major_city" : "Verbena", "education" : { "less_than_a_high_school_diploma" : { "1970" : 10285 } } }
```

10. Finding the major city where population of male in 2019 is greater than equal to 35000

```
> db.users.find({"population_by_gender.summary.male.2019":{$gte:35000}},{major_city:1})
{ "_id" : ObjectId("622df20357e9d4cbb9f63b53"), "major_city" : "Atmore" }
{ "_id" : ObjectId("622df20357e9d4cbb9f63b54"), "major_city" : "Bay Minette" }
{ "_id" : ObjectId("622df20357e9d4cbb9f63b55"), "major_city" : "Spanish Fort" }
{ "_id" : ObjectId("622df20357e9d4cbb9f63b56"), "major_city" : "Fairhope" }
{ "_id" : ObjectId("622df20357e9d4cbb9f63b57"), "major_city" : "Orange Beach" }
{ "_id" : ObjectId("622df20357e9d4cbb9f63b58"), "major_city" : "Little River" }
{ "_id" : ObjectId("622df20357e9d4cbb9f63b59"), "major_city" : "Point Clear" }
{ "_id" : ObjectId("622df20357e9d4cbb9f63b5a"), "major_city" : "Stockton" }
{ "_id" : ObjectId("622df20357e9d4cbb9f63b5b"), "major_city" : "Daphne" }
{ "_id" : ObjectId("622df20357e9d4cbb9f63b5c"), "major_city" : "Loxley" }
{ "_id" : ObjectId("622df20357e9d4cbb9f63b5d"), "major_city" : "Magnolia Springs" }
{ "_id" : ObjectId("622df20357e9d4cbb9f63b5e"), "major_city" : "Gulf Shores" }
{ "_id" : ObjectId("622df20357e9d4cbb9f63b5f"), "major_city" : "Robertsdale" }
{ "_id" : ObjectId("622df20357e9d4cbb9f63b60"), "major_city" : "Fairhope" }
{ "_id" : ObjectId("622df20357e9d4cbb9f63b61"), "major_city" : "Silverhill" }
{ "_id" : ObjectId("622df20357e9d4cbb9f63b62"), "major_city" : "Summerdale" }
{ "_id" : ObjectId("622df20357e9d4cbb9f63b63"), "major_city" : "Gulf Shores" }
{ "_id" : ObjectId("622df20357e9d4cbb9f63b64"), "major_city" : "Montrose" }
{ "_id" : ObjectId("622df20357e9d4cbb9f63b65"), "major_city" : "Perdido" }
{ "_id" : ObjectId("622df20357e9d4cbb9f63b66"), "major_city" : "Lillian" }
Type "it" for more
```

11. Finding major city where area of land is less than or equal to 60 sqmile

```
> db.users.find({"land_area_in_sqmi":{$lte:60}},{"land_area_in_sqmi":1,major_city:1,_id:0})
{ "major_city" : "Prattville", "land_area_in_sqmi" : 19.87 }
{ "major_city" : "Marbury", "land_area_in_sqmi" : 50.78 }
{ "major_city" : "Spanish Fort", "land_area_in_sqmi" : 41.13 }
{ "major_city" : "Orange Beach", "land_area_in_sqmi" : 16.2 }
{ "major_city" : "Little River", "land_area_in_sqmi" : 53.93 }
{ "major_city" : "Point Clear", "land_area_in_sqmi" : 0.3 }
{ "major_city" : "Daphne", "land_area_in_sqmi" : 37.4 }
{ "major_city" : "Magnolia Springs", "land_area_in_sqmi" : 0.84 }
{ "major_city" : "Gulf Shores", "land_area_in_sqmi" : 52.49 }
{ "major_city" : "Silverhill", "land_area_in_sqmi" : 19.45 }
{ "major_city" : "Summerdale", "land_area_in_sqmi" : 53.76 }
{ "major_city" : "Montrose", "land_area_in_sqmi" : 0.36 }
{ "major_city" : "Perdido", "land_area_in_sqmi" : 44.59 }
{ "major_city" : "Lillian", "land_area_in_sqmi" : 27.91 }
{ "major_city" : "Bon Secour", "land_area_in_sqmi" : 4.67 }
{ "major_city" : "Seminole", "land_area_in_sqmi" : 14.01 }
{ "major_city" : "Stapleton", "land_area_in_sqmi" : 30.61 }
{ "major_city" : "Clopton", "land_area_in_sqmi" : 40.61 }
{ "major_city" : "Skipperville", "land_area_in_sqmi" : 54.21 }
{ "major_city" : "Green Pond", "land_area_in_sqmi" : 0.06 }
```

12. Finding major city where area of land is greater than or equal to 60 sqmile

```
> db.users.find({"land_area_in_sqmi":{$gte:60}},{"land_area_in_sqmi":1,major_city:1,_id:0})
{ "major_city" : "Verbena", "land_area_in_sqmi" : 81.8 }
{ "major_city" : "Plantersville", "land_area_in_sqmi" : 87.41 }
{ "major_city" : "Billingsley", "land_area_in_sqmi" : 71.47 }
{ "major_city" : "Prattville", "land_area_in_sqmi" : 239.23 }
{ "major_city" : "Selma", "land_area_in_sqmi" : 259.77 }
{ "major_city" : "Autaugaville", "land_area_in_sqmi" : 93.38 }
{ "major_city" : "Deatsville", "land_area_in_sqmi" : 90.78 }
{ "major_city" : "Selma", "land_area_in_sqmi" : 120.77 }
{ "major_city" : "Jones", "land_area_in_sqmi" : 61.91 }
{ "major_city" : "Atmore", "land_area_in_sqmi" : 328.21 }
{ "major_city" : "Bay Minette", "land_area_in_sqmi" : 334.81 }
{ "major_city" : "Fairhope", "land_area_in_sqmi" : 73.74 }
{ "major_city" : "Stockton", "land_area_in_sqmi" : 131.32 }
{ "major_city" : "Loxley", "land_area_in_sqmi" : 65 }
{ "major_city" : "Robertsdale", "land_area_in_sqmi" : 228.46 }
{ "major_city" : "Elberta", "land_area_in_sqmi" : 96.62 }
{ "major_city" : "Uriah", "land_area_in_sqmi" : 162.08 }
{ "major_city" : "Foley", "land_area_in_sqmi" : 91.48 }
{ "major_city" : "Clio", "land_area_in_sqmi" : 62.26 }
{ "major_city" : "Clayton", "land_area_in_sqmi" : 241.27 }
```

13. Finding major city where area of land is greater than or equal to 50 **and** less than equal to 50 sqmile

```
>  db.users.find({$and:[{"land_area_in_sqmi":{$gte:50}},{"land_area_in_sqmi":{$lte:50}}]},
... ... {"land_area_in_sqmi":1,major_city:1,_id:0})
{ "major_city" : "Clearfield", "land_area_in_sqmi" : 50 }
{ "major_city" : "Clearfield", "land_area_in_sqmi" : 50 }
{ "major_city" : "Wagram", "land_area_in_sqmi" : 50 }
{ "major_city" : "Madison", "land_area_in_sqmi" : 50 }
{ "major_city" : "Madison", "land_area_in_sqmi" : 50 }
{ "major_city" : "Madison", "land_area_in_sqmi" : 50 }
{ "major_city" : "Landisburg", "land_area_in_sqmi" : 50 }
{ "major_city" : "Huron", "land_area_in_sqmi" : 50 }
{ "major_city" : "Iron Belt", "land_area_in_sqmi" : 50 }
{ "major_city" : "Davenport", "land_area_in_sqmi" : 50 }
{ "major_city" : "Wewahitchka", "land_area_in_sqmi" : 50 }
```

14. Finding major city where area of land is greater than or equal to 60 **or** less than equal to 30 sqmile

```
> db.users.find({$or:[{"land_area_in_sqmi":{$lte:30}},{"land_area_in_sqmi":{$gte:60}}]},
... {"land_area_in_sqmi":1,major_city:1,_id:0})
{ "major_city" : "Verbena", "land_area_in_sqmi" : 81.8 }
{ "major_city" : "Plantersville", "land_area_in_sqmi" : 87.41 }
{ "major_city" : "Billingsley", "land_area_in_sqmi" : 71.47 }
{ "major_city" : "Prattville", "land_area_in_sqmi" : 239.23 }
{ "major_city" : "Selma", "land_area_in_sqmi" : 259.77 }
{ "major_city" : "Autaugaville", "land_area_in_sqmi" : 93.38 }
{ "major_city" : "Deatsville", "land_area_in_sqmi" : 90.78 }
{ "major_city" : "Selma", "land_area_in_sqmi" : 120.77 }
{ "major_city" : "Jones", "land_area_in_sqmi" : 61.91 }
{ "major_city" : "Prattville", "land_area_in_sqmi" : 19.87 }
{ "major_city" : "Atmore", "land_area_in_sqmi" : 328.21 }
{ "major_city" : "Bay Minette", "land_area_in_sqmi" : 334.81 }
{ "major_city" : "Fairhope", "land_area_in_sqmi" : 73.74 }
{ "major_city" : "Orange Beach", "land_area_in_sqmi" : 16.2 }
{ "major_city" : "Point Clear", "land_area_in_sqmi" : 0.3 }
{ "major_city" : "Stockton", "land_area_in_sqmi" : 131.32 }
{ "major_city" : "Loxley", "land_area_in_sqmi" : 65 }
{ "major_city" : "Magnolia Springs", "land_area_in_sqmi" : 0.84 }
{ "major_city" : "Robertsdale", "land_area_in_sqmi" : 228.46 }
{ "major_city" : "Silverhill", "land_area_in_sqmi" : 19.45 }
Type "it" for more
```

15. Finding major city where area of land is greater than or equal to 55 **and** less than equal to 60 sqmile

```
> db.users.find("land_area_in_sqmi" > 55 && "land_area_in_sqmi" < 60,{"land_area_in_sqmi":1,major_city:1,_id:0})
{ "major_city" : "Verbena", "land_area_in_sqmi" : 81.8 }
{ "major_city" : "Plantersville", "land_area_in_sqmi" : 87.41 }
{ "major_city" : "Billingsley", "land_area_in_sqmi" : 71.47 }
{ "major_city" : "Prattville", "land_area_in_sqmi" : 239.23 }
{ "major_city" : "Selma", "land_area_in_sqmi" : 259.77 }
{ "major_city" : "Autaugaville", "land_area_in_sqmi" : 93.38 }
{ "major_city" : "Booth", "land_area_in_sqmi" : null }
{ "major_city" : "Prattville", "land_area_in_sqmi" : null }
{ "major_city" : "Deatsville", "land_area_in_sqmi" : 90.78 }
{ "major_city" : "Selma", "land_area_in_sqmi" : 120.77 }
{ "major_city" : "Jones", "land_area_in_sqmi" : 61.91 }
{ "major_city" : "Prattville", "land_area_in_sqmi" : 19.87 }
{ "major_city" : "Marbury", "land_area_in_sqmi" : 50.78 }
{ "major_city" : "Atmore", "land_area_in_sqmi" : 328.21 }
{ "major_city" : "Bay Minette", "land_area_in_sqmi" : 334.81 }
{ "major_city" : "Spanish Fort", "land_area_in_sqmi" : 41.13 }
{ "major_city" : "Fairhope", "land_area_in_sqmi" : 73.74 }
{ "major_city" : "Orange Beach", "land_area_in_sqmi" : 16.2 }
{ "major_city" : "Little River", "land_area_in_sqmi" : 53.93 }
{ "major_city" : "Point Clear", "land_area_in_sqmi" : 0.3 }
Type "it" for more
```

16. Finding the number of major city where area of land is greater than or equal to 55 **and** less than equal to 60 sqmile

```
> db.users.count({$or:[{"land_area_in_sqmi":{$lte:30}},{"land_area_in_sqmi":{$gte:60}}]})
36286
```

17. Finding major city where area of land is greater than or equal to 50 **and** less than equal to 52 sqmile and sorting the area in ascending order

```
> db.users.find({$and:[{"land_area_in_sqmi":{$gte:50}},{"land_area_in_sqmi":{$lte:52}}]},
... {"land_area_in_sqmi":1,major_city:1,_id:0}).sort({"land_area_in_sqmi":1})
{ "major_city" : "Clearfield", "land_area_in_sqmi" : 50 }
{ "major_city" : "Clearfield", "land_area_in_sqmi" : 50 }
{ "major_city" : "Wagram", "land_area_in_sqmi" : 50 }
{ "major_city" : "Madison", "land_area_in_sqmi" : 50 }
{ "major_city" : "Madison", "land_area_in_sqmi" : 50 }
{ "major_city" : "Madison", "land_area_in_sqmi" : 50 }
{ "major_city" : "Landisburg", "land_area_in_sqmi" : 50 }
{ "major_city" : "Huron", "land_area_in_sqmi" : 50 }
{ "major_city" : "Iron Belt", "land_area_in_sqmi" : 50 }
{ "major_city" : "Davenport", "land_area_in_sqmi" : 50 }
{ "major_city" : "Wewahitchka", "land_area_in_sqmi" : 50 }
{ "major_city" : "Gaston", "land_area_in_sqmi" : 50.01 }
{ "major_city" : "Gaston", "land_area_in_sqmi" : 50.01 }
{ "major_city" : "Frisco", "land_area_in_sqmi" : 50.01 }
{ "major_city" : "Frisco", "land_area_in_sqmi" : 50.01 }
{ "major_city" : "Crothersville", "land_area_in_sqmi" : 50.02 }
{ "major_city" : "Crothersville", "land_area_in_sqmi" : 50.02 }
{ "major_city" : "Casa Grande", "land_area_in_sqmi" : 50.02 }
{ "major_city" : "Nicolaus", "land_area_in_sqmi" : 50.02 }
{ "major_city" : "Beaver Falls", "land_area_in_sqmi" : 50.03 }
Type "it" for more
```

18. Finding major city where area of land is greater than or equal to 50 **and** less than equal to 52 sqmile and sorting the area in decending order

```
> db.users.find({$and:[{"land_area_in_sqmi":{$gte:50}},{"land_area_in_sqmi":{$lte:52}}]},
... {"land_area_in_sqmi":1,major_city:1,_id:0}).sort({"land_area_in_sqmi":-1})
{ "major_city" : "Bryce", "land_area_in_sqmi" : 51.99 }
{ "major_city" : "Holliday", "land_area_in_sqmi" : 51.98 }
{ "major_city" : "Fortson", "land_area_in_sqmi" : 51.97 }
{ "major_city" : "Fortson", "land_area_in_sqmi" : 51.97 }
{ "major_city" : "Coker", "land_area_in_sqmi" : 51.96 }
{ "major_city" : "Nixa", "land_area_in_sqmi" : 51.95 }
{ "major_city" : "Nixa", "land_area_in_sqmi" : 51.95 }
{ "major_city" : "Bradley", "land_area_in_sqmi" : 51.95 }
{ "major_city" : "Bradley", "land_area_in_sqmi" : 51.95 }
{ "major_city" : "Parrottsville", "land_area_in_sqmi" : 51.94 }
{ "major_city" : "Von Ormy", "land_area_in_sqmi" : 51.94 }
{ "major_city" : "Von Ormy", "land_area_in_sqmi" : 51.94 }
{ "major_city" : "Evans", "land_area_in_sqmi" : 51.94 }
{ "major_city" : "Henderson", "land_area_in_sqmi" : 51.92 }
{ "major_city" : "Princeton", "land_area_in_sqmi" : 51.92 }
{ "major_city" : "Princeton", "land_area_in_sqmi" : 51.92 }
{ "major_city" : "Princeton", "land_area_in_sqmi" : 51.92 }
{ "major_city" : "Hueysville", "land_area_in_sqmi" : 51.91 }
{ "major_city" : "Hueysville", "land_area_in_sqmi" : 51.91 }
{ "major_city" : "Cub Run", "land_area_in_sqmi" : 51.9 }
Type "it" for more
```

19. Creating index on state

```
> db.users.createIndex({"state":1})
{
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "createdCollectionAutomatically" : false,
        "ok" : 1
}
>
```

20. Creating index on state and county

```
> db.users.createIndex({"state":1,"county":1})
{
        "numIndexesBefore" : 2,
        "numIndexesAfter" : 3,
        "createdCollectionAutomatically" : false,
        "ok" : 1
}
```

21. Grouping state_name and summing all values, Finding all the sate names and number of more cities in the state

```
> db.users.aggregate([
... {"$group" : {_id:"$state_name", count:{$sum:1}}}
... ])
{ "_id" : "Alaska", "count" : 248 }
{ "_id" : "Florida", "count" : 1602 }
{ "_id" : "Arizona", "count" : 532 }
{ "_id" : "Indiana", "count" : 1520 }
{ "_id" : "Louisiana", "count" : 882 }
{ "_id" : "Alabama", "count" : 1122 }
{ "_id" : "New Jersey", "count" : 817 }
{ "_id" : "Mississippi", "count" : 127 }
{ "_id" : "California", "count" : 2645 }
{ "_id" : "Iowa", "count" : 1732 }
{ "_id" : "North Carolina", "count" : 1522 }
{ "_id" : "Ohio", "count" : 2050 }
{ "_id" : "Georgia", "count" : 1582 }
{ "_id" : "Tennessee", "count" : 1236 }
{ "_id" : "West Virginia", "count" : 1067 }
{ "_id" : "Kentucky", "count" : 1440 }
{ "_id" : "Minnesota", "count" : 1465 }
{ "_id" : "Wisconsin", "count" : 921 }
{ "_id" : "Illinois", "count" : 2240 }
{ "_id" : "Massachusetts", "count" : 728 }
Type "it" for more
```

22. Finding the average of the are square mile of the cities of each state. That is grouping on state then fining average of the area sqmile of all the cities present in the state

```
> db.users.aggregate([
... {"$group" : {_id:"$state_name", avg_sqmi:{$avg:"$land_area_in_sqmi"}}}
... ])
{ "_id" : "Alaska", "avg_sqmi" : 1043.4522222222222 }
{ "_id" : "Florida", "avg_sqmi" : 60.05229202037352 }
{ "_id" : "Arizona", "avg_sqmi" : 245.62872685185187 }
{ "_id" : "Indiana", "avg_sqmi" : 56.249853479853485 }
{ "_id" : "Louisiana", "avg_sqmi" : 94.26875342465753 }
{ "_id" : "Alabama", "avg_sqmi" : 92.91329317269076 }
{ "_id" : "Mississippi", "avg_sqmi" : 134.84567796610168 }
{ "_id" : "California", "avg_sqmi" : 66.8800563236047 }
{ "_id" : "New Jersey", "avg_sqmi" : 13.988314763231198 }
{ "_id" : "Iowa", "avg_sqmi" : 66.25187120291616 }
{ "_id" : "North Carolina", "avg_sqmi" : 69.27273781902552 }
{ "_id" : "Ohio", "avg_sqmi" : 43.2251724137931 }
{ "_id" : "Georgia", "avg_sqmi" : 88.38996426018585 }
{ "_id" : "Tennessee", "avg_sqmi" : 77.30929347826087 }
{ "_id" : "West Virginia", "avg_sqmi" : 41.764702127659575 }
{ "_id" : "Kentucky", "avg_sqmi" : 73.71540769230769 }
{ "_id" : "Minnesota", "avg_sqmi" : 101.1127119831815 }
{ "_id" : "Wisconsin", "avg_sqmi" : 83.89591385331781 }
{ "_id" : "Illinois", "avg_sqmi" : 46.07235180722892 }
{ "_id" : "Massachusetts", "avg_sqmi" : 15.31331158238173 }
Type "it" for more
```

23. Finding all the distinct land areas

```
> db.users.distinct("land_area_in_sqmi",{"state_name":"Alaska"})
[
        null,
        0,
        0.12,
        0.19,
        0.27,
        0.55,
        0.68,
        0.88,
        0.91,
        0.92,
        0.96,
        1.01,
        1.04,
        1.11,
        1.53,
```

24. Finding the average of the are square mile of the cities of each state using project. That is grouping on state then fining average of the area sqmile of all the cities present in the state

```
> db.users.aggregate([
... {"$group" : {_id:"$state_name", avg_sqmi:{$avg:"$land_area_in_sqmi"}}},
... {$project : { avg_sqmi:1, _id: 0}}
... ])
{ "avg_sqmi" : 1043.4522222222222 }
{ "avg_sqmi" : 60.05229202037352 }
{ "avg_sqmi" : 245.62872685185187 }
{ "avg_sqmi" : 56.249853479853485 }
{ "avg_sqmi" : 94.26875342465753 }
{ "avg_sqmi" : 92.91329317269076 }
{ "avg_sqmi" : 13.988314763231198 }
{ "avg_sqmi" : 134.84567796610168 }
{ "avg_sqmi" : 66.8800563236047 }
{ "avg_sqmi" : 66.25187120291616 }
{ "avg_sqmi" : 69.27273781902552 }
{ "avg_sqmi" : 43.2251724137931 }
{ "avg_sqmi" : 88.38996426018585 }
{ "avg_sqmi" : 77.30929347826087 }
{ "avg_sqmi" : 41.764702127659575 }
{ "avg_sqmi" : 73.71540769230769 }
{ "avg_sqmi" : 101.1127119831815 }
{ "avg_sqmi" : 83.89591385331781 }
{ "avg_sqmi" : 46.07235180722892 }
{ "avg_sqmi" : 15.31331158238173 }
Type "it" for more
```

25. Finding all the major cities whose name begins with “Ver”

```
> db.users.find({major_city: /^Ver/ },{major_city:1,_id:0})
{ "major_city" : "Verbena" }
{ "major_city" : "Verbena" }
{ "major_city" : "Vernon" }
{ "major_city" : "Versailles" }
{ "major_city" : "Vermilion" }
{ "major_city" : "Vernon" }
{ "major_city" : "Vermont" }
{ "major_city" : "Verona" }
{ "major_city" : "Vergennes" }
{ "major_city" : "Vernon Hills" }
{ "major_city" : "Vermont" }
{ "major_city" : "Vernon" }
{ "major_city" : "Vernon" }
{ "major_city" : "Versailles" }
{ "major_city" : "Vermillion" }
{ "major_city" : "Vermillion" }
{ "major_city" : "Verona" }
{ "major_city" : "Verona" }
{ "major_city" : "Verona" }
{ "major_city" : "Versailles" }
Type "it" for more
```

26. Finding all the major cities whose name ends with “on”

```
> db.users.find({major_city: /on$/ },{major_city:1,_id:0})
{ "major_city" : "Stockton" }
{ "major_city" : "Stapleton" }
{ "major_city" : "Clayton" }
{ "major_city" : "Clopton" }
{ "major_city" : "Ariton" }
{ "major_city" : "West Blocton" }
{ "major_city" : "Marion" }
{ "major_city" : "Horton" }
{ "major_city" : "Pinson" }
{ "major_city" : "Anniston" }
{ "major_city" : "Anniston" }
{ "major_city" : "Anniston" }
{ "major_city" : "Anniston" }
{ "major_city" : "Anniston" }
{ "major_city" : "Anniston" }
{ "major_city" : "Wellington" }
{ "major_city" : "Stanton" }
{ "major_city" : "Clanton" }
{ "major_city" : "Clanton" }
{ "major_city" : "Jemison" }
Type "it" for more
```

27. Using unwind to see all the documents in unemployment

```
> db.users.aggregate({$unwind : "$unemployment"}).pretty()
```

```
                        "2016" : 10,
                        "2017" : 10,
                        "2018" : 8,
                        "2019" : 12
                }
        },
        "native_hawaiian_pacific_islander_combination" : {
                "male" : {
                        "2010_census" : 0,
                        "2010" : 0,
                        "2011" : 0,
                        "2012" : 0,
                        "2013" : 0,
                        "2014" : 0,
                        "2015" : 1,
                        "2016" : 1,
                        "2017" : 0,
                        "2018" : 1,
                        "2019" : 3
                },
                "female" : {
                        "2010_census" : 1,
                        "2010" : 1,
                        "2011" : 1,
                        "2012" : 0,
                        "2013" : 2,
                        "2014" : 0,
                        "2015" : 0,
                        "2016" : 1,
                        "2017" : 1,
                        "2018" : 1,
                        "2019" : 0
```

28. Finding all the states who has more than 1000 major cities. (Grouping on state, using sum to count major cities, then using match condition

```
> db.users.aggregate([
... {"$group" : {_id:"$state_name", count:{$sum:1}}},
... {$match : {count : {$gt : 1000}}},
... {$project : { count:1, _id: 0}}
... ])
{ "count" : 1595 }
{ "count" : 1002 }
{ "count" : 1489 }
{ "count" : 3298 }
{ "count" : 1602 }
{ "count" : 2505 }
{ "count" : 1652 }
{ "count" : 1189 }
{ "count" : 1028 }
{ "count" : 1048 }
{ "count" : 2558 }
{ "count" : 2240 }
{ "count" : 1465 }
{ "count" : 1440 }
{ "count" : 1582 }
{ "count" : 1236 }
{ "count" : 1732 }
{ "count" : 1522 }
{ "count" : 2050 }
{ "count" : 1122 }
Type "it" for more
```

29. Finding all the states who has more than 3000 major cities. (Grouping on state, using sum to count major cities, then using match condition

```
> db.users.aggregate([
... {"$group" : {_id:"$state_name", count:{$sum:1}}},
... {$match : {count : {$gt : 3000}}},
... {$project : { count:1, _id: 0}}
... ])
{ "count" : 3298 }
```

30. Finding only one document using limit

```
> db.users.find().pretty().limit(1)
{
        "_id" : ObjectId("622df20257e9d4cbb9f63b46"),
        "zipcode_type" : "Standard",
        "major_city" : "Verbena",
        "post_office_city" : "Verbena, AL",
        "county" : "Chilton County",
        "state" : "AL",
        "lat" : 32.8,
        "lng" : -86.5,
        "timezone" : "Central",
        "radius_in_miles" : 10,
        "land_area_in_sqmi" : 81.8,
        "water_area_in_sqmi" : 1.64,
        "zipcode" : "36091",
        "county_fips" : 1001,
        "state_fips" : 1,
        "state_name" : "Alabama",
        "county_sumlev" : 50,
        "county_region" : 3,
        "county_division" : 6,
        "county_region_name" : "South",
        "county_division_name" : "East South Central",
        "population" : {
                "rural_urban" : {
                        "continumm_code_2003" : 2,
                        "continumm_code_2013" : 2
                }
```

31. Finding the last document using limit and sort

```
> db.users.find().pretty().limit(1).sort({$natural:-1})
{
        "_id" : ObjectId("622dfde657e9d4cbb9f70667"),
        "zipcode_type" : "Standard",
        "major_city" : "Woodstock",
        "post_office_city" : "Woodstock, GA",
        "county" : "Cherokee County",
        "state" : "GA",
        "lat" : 34.13,
        "lng" : -84.57,
        "timezone" : "Eastern",
        "radius_in_miles" : 4,
        "land_area_in_sqmi" : 19.45,
        "water_area_in_sqmi" : 2.15,
        "zipcode" : "30189",
        "county_fips" : 13057,
        "state_fips" : 13,
        "state_name" : "Georgia",
        "county_sumlev" : 50,
        "county_region" : 3,
        "county_division" : 5,
        "county_region_name" : "South",
        "county_division_name" : "South Atlantic",
        "population" : {
                "rural_urban" : {
```

32. Inserting a document

```
> db.users.insert({"zipcode_type":"Non-Standardz"})
WriteResult({ "nInserted" : 1 })
> db.users.find().pretty().limit(1).sort({$natural:-1})
{
        "_id" : ObjectId("623ac0ac661c27431f97676f"),
        "zipcode_type" : "Non-Standardz"
}
```

33. Viewing the last inserted document

```
> db.users.find().pretty().limit(1).sort({$natural:-1})
{
        "_id" : ObjectId("623ac126661c27431f976770"),
        "zipcode_type" : "Non-Standardz",
        "major_city" : "Karz"
}
```

34. Inserting multiple (2) documents

```
> db.users.insert([{"zipcode_type":"Non-Standardz", "major_city":"Karz"},
...   {"zipcode_type":"Nz-Standardz", "major_city":"Larz"}])
BulkWriteResult({
        "writeErrors" : [ ],
        "writeConcernErrors" : [ ],
        "nInserted" : 2,
        "nUpserted" : 0,
        "nMatched" : 0,
        "nModified" : 0,
        "nRemoved" : 0,
        "upserted" : [ ]
})
> db.users.find().pretty().limit(1).sort({$natural:-1})
{
        "_id" : ObjectId("623ac254661c27431f976772"),
        "zipcode_type" : "Nz-Standardz",
        "major_city" : "Larz"
}
```

35. Viewing the newly inserted documents

```
> db.users.find().pretty().limit(4).sort({$natural:-1})
{
        "_id" : ObjectId("623ac254661c27431f976772"),
        "zipcode_type" : "Nz-Standardz",
        "major_city" : "Larz"
}
{
        "_id" : ObjectId("623ac254661c27431f976771"),
        "zipcode_type" : "Non-Standardz",
        "major_city" : "Karz"
}
{
        "_id" : ObjectId("623ac126661c27431f976770"),
        "zipcode_type" : "Non-Standardz",
        "major_city" : "Karz"
}
{
        "_id" : ObjectId("623ac0ac661c27431f97676f"),
        "zipcode_type" : "Non-Standardz"
```

36. Updating document with the id and changing major_city to Tarz

```
> db.users.update({"_id" : ObjectId("623ac254661c27431f976771")},
... {$set: {"major_city" : "Tarz"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.users.find().pretty().limit(4).sort({$natural:-1})
{
        "_id" : ObjectId("623ac254661c27431f976772"),
        "zipcode_type" : "Nz-Standardz",
        "major_city" : "Larz"
}
{
        "_id" : ObjectId("623ac254661c27431f976771"),
        "zipcode_type" : "Non-Standardz",
        "major_city" : "Tarz"
}
{
        "_id" : ObjectId("623ac126661c27431f976770"),
        "zipcode_type" : "Non-Standardz",
        "major_city" : "Karz"
}
{
        "_id" : ObjectId("623ac0ac661c27431f97676f"),
        "zipcode_type" : "Non-Standardz"
}
>
```

37. Finding the major city with name Karz

```
> db.users.find({major_city: "Karz"})
{ "_id" : ObjectId("623ac126661c27431f976770"), "zipcode_type" : "Non-Standardz", "major_city" : "Karz" }
```

38. Removing all the documents with zipcode type “Non-Standards”

```
> db.users.remove({"zipcode_type" : "Non-Standardz"})
WriteResult({ "nRemoved" : 3 })
```

39. Viewing after removing documents

```
> db.users.remove({"zipcode_type" : "Non-Standardz"})
WriteResult({ "nRemoved" : 3 })
> db.users.find().pretty().limit(1).sort({$natural:-1})
{
        "_id" : ObjectId("623ac254661c27431f976772"),
        "zipcode_type" : "Nz-Standardz",
        "major_city" : "Larz"
}
```

40. Finding all the major cities whose county_sumlev is greater than land_are_in_sqmi

```
> db.users.aggregate([
... { $project: {compareResult : {$gt : ["$county_sumlev", "$land_area_in_sqmi"]},
... "major_city" : 1, _id: 0 }}])
{ "major_city" : "Verbena", "compareResult" : false }
{ "major_city" : "Plantersville", "compareResult" : false }
{ "major_city" : "Billingsley", "compareResult" : false }
{ "major_city" : "Prattville", "compareResult" : false }
{ "major_city" : "Selma", "compareResult" : false }
{ "major_city" : "Autaugaville", "compareResult" : false }
{ "major_city" : "Booth", "compareResult" : true }
{ "major_city" : "Prattville", "compareResult" : true }
{ "major_city" : "Deatsville", "compareResult" : false }
{ "major_city" : "Selma", "compareResult" : false }
{ "major_city" : "Jones", "compareResult" : false }
{ "major_city" : "Prattville", "compareResult" : true }
{ "major_city" : "Marbury", "compareResult" : false }
{ "major_city" : "Atmore", "compareResult" : false }
{ "major_city" : "Bay Minette", "compareResult" : false }
{ "major_city" : "Spanish Fort", "compareResult" : true }
{ "major_city" : "Fairhope", "compareResult" : false }
{ "major_city" : "Orange Beach", "compareResult" : true }
{ "major_city" : "Little River", "compareResult" : false }
{ "major_city" : "Point Clear", "compareResult" : true }
Type "it" for more
```

41. Finding all the major cities whose county_sumlev is less than land_are_in_sqmi

```
> db.users.aggregate([
... { $project: {compareResult : {$lt : ["$county_sumlev", "$land_area_in_sqmi"]},
... "major_city" : 1, _id: 0 }}])
{ "major_city" : "Verbena", "compareResult" : true }
{ "major_city" : "Plantersville", "compareResult" : true }
{ "major_city" : "Billingsley", "compareResult" : true }
{ "major_city" : "Prattville", "compareResult" : true }
{ "major_city" : "Selma", "compareResult" : true }
{ "major_city" : "Autaugaville", "compareResult" : true }
{ "major_city" : "Booth", "compareResult" : false }
{ "major_city" : "Prattville", "compareResult" : false }
{ "major_city" : "Deatsville", "compareResult" : true }
{ "major_city" : "Selma", "compareResult" : true }
{ "major_city" : "Jones", "compareResult" : true }
{ "major_city" : "Prattville", "compareResult" : false }
{ "major_city" : "Marbury", "compareResult" : true }
{ "major_city" : "Atmore", "compareResult" : true }
{ "major_city" : "Bay Minette", "compareResult" : true }
{ "major_city" : "Spanish Fort", "compareResult" : false }
{ "major_city" : "Fairhope", "compareResult" : true }
{ "major_city" : "Orange Beach", "compareResult" : false }
{ "major_city" : "Little River", "compareResult" : true }
{ "major_city" : "Point Clear", "compareResult" : false }
Type "it" for more
```

42. Finding the sum of county_sumlev is less than land_are_in_sqmi

```
> db.users.aggregate([
... { $project: {combined : {$add : ["$county_sumlev", "$land_area_in_sqmi"]},
... "major_city" : 1, _id: 0 }}])
{ "major_city" : "Verbena", "combined" : 131.8 }
{ "major_city" : "Plantersville", "combined" : 137.41 }
{ "major_city" : "Billingsley", "combined" : 121.47 }
{ "major_city" : "Prattville", "combined" : 289.23 }
{ "major_city" : "Selma", "combined" : 309.77 }
{ "major_city" : "Autaugaville", "combined" : 143.38 }
{ "major_city" : "Booth", "combined" : null }
{ "major_city" : "Prattville", "combined" : null }
{ "major_city" : "Deatsville", "combined" : 140.78 }
{ "major_city" : "Selma", "combined" : 170.76999999999998 }
{ "major_city" : "Jones", "combined" : 111.91 }
{ "major_city" : "Prattville", "combined" : 69.87 }
{ "major_city" : "Marbury", "combined" : 100.78 }
{ "major_city" : "Atmore", "combined" : 378.21 }
{ "major_city" : "Bay Minette", "combined" : 384.81 }
{ "major_city" : "Spanish Fort", "combined" : 91.13 }
{ "major_city" : "Fairhope", "combined" : 123.74 }
{ "major_city" : "Orange Beach", "combined" : 66.2 }
{ "major_city" : "Little River", "combined" : 103.93 }
{ "major_city" : "Point Clear", "combined" : 50.3 }
Type "it" for more
```

43. Finding the sum of county_sumlev is less than land_are_in_sqmi and displaying with project

```
> db.users.aggregate([
... { $project: {combined : {$add : ["$county_sumlev", "$land_area_in_sqmi"]},
... "major_city" : 1, "county_sumlev" :1, "land_area_in_sqmi": 1, _id: 0 }}])
{ "major_city" : "Verbena", "land_area_in_sqmi" : 81.8, "county_sumlev" : 50, "combined" : 131.8 }
{ "major_city" : "Plantersville", "land_area_in_sqmi" : 87.41, "county_sumlev" : 50, "combined" : 137.41 }
{ "major_city" : "Billingsley", "land_area_in_sqmi" : 71.47, "county_sumlev" : 50, "combined" : 121.47 }
{ "major_city" : "Prattville", "land_area_in_sqmi" : 239.23, "county_sumlev" : 50, "combined" : 289.23 }
{ "major_city" : "Selma", "land_area_in_sqmi" : 259.77, "county_sumlev" : 50, "combined" : 309.77 }
{ "major_city" : "Autaugaville", "land_area_in_sqmi" : 93.38, "county_sumlev" : 50, "combined" : 143.38 }
{ "major_city" : "Booth", "land_area_in_sqmi" : null, "county_sumlev" : 50, "combined" : null }
{ "major_city" : "Prattville", "land_area_in_sqmi" : null, "county_sumlev" : 50, "combined" : null }
{ "major_city" : "Deatsville", "land_area_in_sqmi" : 90.78, "county_sumlev" : 50, "combined" : 140.78 }
{ "major_city" : "Selma", "land_area_in_sqmi" : 120.77, "county_sumlev" : 50, "combined" : 170.76999999999998 }
{ "major_city" : "Jones", "land_area_in_sqmi" : 61.91, "county_sumlev" : 50, "combined" : 111.91 }
{ "major_city" : "Prattville", "land_area_in_sqmi" : 19.87, "county_sumlev" : 50, "combined" : 69.87 }
{ "major_city" : "Marbury", "land_area_in_sqmi" : 50.78, "county_sumlev" : 50, "combined" : 100.78 }
{ "major_city" : "Atmore", "land_area_in_sqmi" : 328.21, "county_sumlev" : 50, "combined" : 378.21 }
{ "major_city" : "Bay Minette", "land_area_in_sqmi" : 334.81, "county_sumlev" : 50, "combined" : 384.81 }
{ "major_city" : "Spanish Fort", "land_area_in_sqmi" : 41.13, "county_sumlev" : 50, "combined" : 91.13 }
{ "major_city" : "Fairhope", "land_area_in_sqmi" : 73.74, "county_sumlev" : 50, "combined" : 123.74 }
{ "major_city" : "Orange Beach", "land_area_in_sqmi" : 16.2, "county_sumlev" : 50, "combined" : 66.2 }
{ "major_city" : "Little River", "land_area_in_sqmi" : 53.93, "county_sumlev" : 50, "combined" : 103.93 }
{ "major_city" : "Point Clear", "land_area_in_sqmi" : 0.3, "county_sumlev" : 50, "combined" : 50.3 }
Type "it" for more
```

REPLICATION

44. Updating the mongoshell

```
root@3c0e975d3a4c:/# apt-get update
Get:1 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Ign:3 http://repo.mongodb.org/apt/ubuntu focal/mongodb-org/5.0 InRelease
Get:4 http://repo.mongodb.org/apt/ubuntu focal/mongodb-org/5.0 Release [4406 B]
Get:5 http://repo.mongodb.org/apt/ubuntu focal/mongodb-org/5.0 Release.gpg [801 B]
Get:6 http://repo.mongodb.org/apt/ubuntu focal/mongodb-org/5.0/multiverse amd64 Packages [13.2 kB]
Get:7 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:8 http://archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [1068 kB]
Get:10 http://archive.ubuntu.com/ubuntu focal/universe amd64 Packages [11.3 MB]
Get:11 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [1683 kB]
Get:12 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [25.8 kB]
Get:13 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [860 kB]
```

45. Installing nano in mongoshell

```
root@3c0e975d3a4c:/# apt-get install nano
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  hunspell
The following NEW packages will be installed:
  nano
0 upgraded, 1 newly installed, 0 to remove and 18 not upgraded.
Need to get 269 kB of archives.
After this operation, 868 kB of additional disk space will be used.
```

46. Creating 3 configuration files of node1, node2, node 3

```
root@3c0e975d3a4c:/# nano node1.conf
root@3c0e975d3a4c:/# nano node1.conf
root@3c0e975d3a4c:/# nano node2.conf
root@3c0e975d3a4c:/# nano node3.conf
```

47. Node1.conf file creation

root@3c0e975d3a4c: /

```
  GNU nano 4.8
storage:
  dbPath: /var/mongodb/db/1
net:
  bindIp: localhost
  port: 27001
security:
  authorization: enabled
  keyFile: /var/mongodb/pki/m103-keyfile
systemLog:
  destination: file
  path: /var/mongodb/logs/mongod1.log
  logAppend: true
processManagement:
  fork: true
replication:
  replSetName: "repl"
```

48. Node2.conf file creation

root@3c0e975d3a4c: /

```
  GNU nano 4.8
storage:
  dbPath: /var/mongodb/db/2
net:
  bindIp: localhost
  port: 27002
security:
  authorization: enabled
  keyFile: /var/mongodb/pki/m103-keyfile
systemLog:
  destination: file
  path: /var/mongodb/logs/mongod2.log
  logAppend: true
processManagement:
  fork: true
replication:
  replSetName: "repl"
```

49. Node3.conf file creation

root@3c0e975d3a4c: /

```
  GNU nano 4.8
storage:
  dbPath: /var/mongodb/db/3
net:
  bindIp: localhost
  port: 27003
security:
  authorization: enabled
  keyFile: /var/mongodb/pki/m103-keyfile
systemLog:
  destination: file
  path: /var/mongodb/logs/mongod3.log
  logAppend: true
processManagement:
  fork: true
replication:
  replSetName: "repl"
```

50. Creating directories for node1

```
root@3c0e975d3a4c:/# mkdir -p /var/mongodb/db/1
root@3c0e975d3a4c:/# mkdir -p /var/mongodb/pki
root@3c0e975d3a4c:/# mkdir -p /var/mongodb/logs
```

51. Changing ownership of folder, generating keyfile and changing permission of the keyfile, then starting node1

```
root@3c0e975d3a4c:/# chown root:root /var/mongodb/pki/
root@3c0e975d3a4c:/# openssl rand -base64 741 > /var/mongodb/pki/m103-keyfile
root@3c0e975d3a4c:/# chmod 600 /var/mongodb/pki/m103-keyfile
root@3c0e975d3a4c:/# mongod -f node1.conf
about to fork child process, waiting until server is ready for connections.
forked process: 549
child process started successfully, parent exiting
root@3c0e975d3a4c:/#
```

52. Creating directories for node2 and node3 respectively

```
root@3c0e975d3a4c:/# mkdir -p /var/mongodb/db/2
root@3c0e975d3a4c:/# mkdir -p /var/mongodb/db/3
```

53. Starting node2

```
root@3c0e975d3a4c:/# mongod -f node2.conf
about to fork child process, waiting until server is ready for connections.
forked process: 995
child process started successfully, parent exiting
root@3c0e975d3a4c:/#
```

54. Staring node3

```
root@3c0e975d3a4c:/# mongod -f node3.conf
about to fork child process, waiting until server is ready for connections.
forked process: 1081
child process started successfully, parent exiting
root@3c0e975d3a4c:/#
```

55. Connecting to node1 port

```
root@3c0e975d3a4c:/# mongo --port 27001
MongoDB shell version v5.0.6
connecting to: mongodb://127.0.0.1:27001/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("d6c95b1f-fbe4-4ceb-99f2-d1569e977812") }
MongoDB server version: 5.0.6
===============
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
===============
>
```

56. Initiating a replica set

```
> rs.initiate()
{
        "info2" : "no configuration specified. Using a default configuration for the set",
        "me" : "localhost:27001",
        "ok" : 1
}
repl:SECONDARY>
```

57. Creating a user

```
repl:SECONDARY> use admin
switched to db admin
repl:PRIMARY>
```

```
repl:PRIMARY> db.createUser({
...     user: "m103-admin",
...     pwd: "m103-pass",
...     roles: [
...       {role: "root", db: "admin"}
...     ]
... })
Successfully added user: {
        "user" : "m103-admin",
        "roles" : [
                {
                        "role" : "root",
                        "db" : "admin"
                }
        ]
}
repl:PRIMARY>
```

58. Exiting out of the Mongo shell and connecting to the entire replica set

```
repl:PRIMARY> exit
bye
```

```
root@3c0e975d3a4c:/# mongo --host 'repl/127.0.0.1:27001' -u 'm103-admin' -p 'm103-pass' --authenticationDatabase 'admin'
MongoDB shell version v5.0.6
connecting to: mongodb://127.0.0.1:27001/?authSource=admin&compressors=disabled&gssapiServiceName=mongodb&replicaSet=repl
Implicit session: session { "id" : UUID("8f479ce4-5ee2-40bb-85b4-f36472c0be43") }
MongoDB server version: 5.0.6
================
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
================
---
The server generated these startup warnings when booting:
        2022-03-23T12:16:26.652+00:00: You are running this process as the root user, which is not recommended
        2022-03-23T12:16:26.656+00:00: /sys/kernel/mm/transparent_hugepage/enabled is 'always'. We suggest setting it to 'never'
---
---
        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).

        The monitoring data will be available on a MongoDB website with a unique URL accessible to you
        and anyone you share the URL with. MongoDB may use this information to make product
        improvements and to suggest MongoDB products and deployment options to you.

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
repl:PRIMARY>
```

59. Getting replica set status

```
repl:PRIMARY> rs.status()
{
        "set" : "repl",
        "date" : ISODate("2022-03-23T12:45:24.887Z"),
        "myState" : 1,
        "term" : NumberLong(1),
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "heartbeatIntervalMillis" : NumberLong(2000),
        "majorityVoteCount" : 1,
        "writeMajorityCount" : 1,
        "votingMembersCount" : 1,
        "writableVotingMembersCount" : 1,
        "optimes" : {
                "lastCommittedOpTime" : {
                        "ts" : Timestamp(1648039518, 1),
                        "t" : NumberLong(1)
                },
                "lastCommittedWallTime" : ISODate("2022-03-23T12:45:18.676Z"),
                "readConcernMajorityOpTime" : {
                        "ts" : Timestamp(1648039518, 1),
                        "t" : NumberLong(1)
                },
                "appliedOpTime" : {
                        "ts" : Timestamp(1648039518, 1),
                        "t" : NumberLong(1)
                },
                "durableOpTime" : {
                        "ts" : Timestamp(1648039518, 1),
                        "t" : NumberLong(1)
                },
                "lastAppliedWallTime" : ISODate("2022-03-23T12:45:18.676Z"),
                "lastDurableWallTime" : ISODate("2022-03-23T12:45:18.676Z")
```

```
                "lastAppliedWallTime" : ISODate("2022-03-23T12:45:18.676Z"),
                "lastDurableWallTime" : ISODate("2022-03-23T12:45:18.676Z")
        },
        "lastStableRecoveryTimestamp" : Timestamp(1648039498, 1),
        "electionCandidateMetrics" : {
                "lastElectionReason" : "electionTimeout",
                "lastElectionDate" : ISODate("2022-03-23T12:35:58.577Z"),
                "electionTerm" : NumberLong(1),
                "lastCommittedOpTimeAtElection" : {
                        "ts" : Timestamp(1648038958, 1),
                        "t" : NumberLong(-1)
                },
                "lastSeenOpTimeAtElection" : {
                        "ts" : Timestamp(1648038958, 1),
                        "t" : NumberLong(-1)
                },
                "numVotesNeeded" : 1,
                "priorityAtElection" : 1,
                "electionTimeoutMillis" : NumberLong(10000),
                "newTermStartDate" : ISODate("2022-03-23T12:35:58.608Z"),
                "wMajorityWriteAvailabilityDate" : ISODate("2022-03-23T12:35:58.641Z")
        },
        "members" : [
                {
                        "_id" : 0,
                        "name" : "localhost:27001",
                        "health" : 1,
                        "state" : 1,
                        "stateStr" : "PRIMARY",
                        "uptime" : 1739,
                        "optime" : {
                                "ts" : Timestamp(1648039518, 1),
                                "t" : NumberLong(1)
                        },
                        "optimeDate" : ISODate("2022-03-23T12:45:18Z"),
                        "lastAppliedWallTime" : ISODate("2022-03-23T12:45:18.676Z"),
                        "lastDurableWallTime" : ISODate("2022-03-23T12:45:18.676Z"),
                        "syncSourceHost" : "",
```

```
                                "t" : NumberLong(1)
                        },
                        "optimeDate" : ISODate("2022-03-23T12:45:18Z"),
                        "lastAppliedWallTime" : ISODate("2022-03-23T12:45:18.676Z"),
                        "lastDurableWallTime" : ISODate("2022-03-23T12:45:18.676Z"),
                        "syncSourceHost" : "",
                        "syncSourceId" : -1,
                        "infoMessage" : "",
                        "electionTime" : Timestamp(1648038958, 2),
                        "electionDate" : ISODate("2022-03-23T12:35:58Z"),
                        "configVersion" : 1,
                        "configTerm" : 1,
                        "self" : true,
                        "lastHeartbeatMessage" : ""
                }
        ],
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1648039518, 1),
                "signature" : {
                        "hash" : BinData(0,"TzTGtBeNsaXXHUUjlyWt+5zDQ0k="),
                        "keyId" : NumberLong("7078273427143917573")
                }
        },
        "operationTime" : Timestamp(1648039518, 1)
}
repl:PRIMARY>
```

60. Adding other members node2 to replica set

```
repl:PRIMARY> rs.add("localhost:27002")
{
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1648039745, 1),
                "signature" : {
                        "hash" : BinData(0,"VX0TS37SlhHo8XvPyiXDBuaLN+4="),
                        "keyId" : NumberLong("7078273427143917573")
                }
        },
        "operationTime" : Timestamp(1648039745, 1)
}
repl:PRIMARY>
```

61. Adding other members node3 to replica set

```
repl:PRIMARY> rs.add("localhost:27003")
{
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1648039785, 1),
                "signature" : {
                        "hash" : BinData(0,"bk9z5E5lJdOXLj16Np/O2Pn6ehU="),
                        "keyId" : NumberLong("7078273427143917573")
                }
        },
        "operationTime" : Timestamp(1648039785, 1)
}
repl:PRIMARY>
```

62. Getting replica set status

```
repl:PRIMARY> rs.status()
{
        "set" : "repl",
        "date" : ISODate("2022-03-23T13:45:59.123Z"),
        "myState" : 1,
        "term" : NumberLong(3),
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "heartbeatIntervalMillis" : NumberLong(2000),
        "majorityVoteCount" : 2,
        "writeMajorityCount" : 2,
        "votingMembersCount" : 3,
        "writableVotingMembersCount" : 3,
        "optimes" : {
                "lastCommittedOpTime" : {
                        "ts" : Timestamp(1648043152, 1),
                        "t" : NumberLong(3)
                },
                "lastCommittedWallTime" : ISODate("2022-03-23T13:45:52.894Z"),
                "readConcernMajorityOpTime" : {
                        "ts" : Timestamp(1648043152, 1),
                        "t" : NumberLong(3)
                },
                "appliedOpTime" : {
                        "ts" : Timestamp(1648043152, 1),
                        "t" : NumberLong(3)
                },
                "durableOpTime" : {
                        "ts" : Timestamp(1648043152, 1),
                        "t" : NumberLong(3)
                },
                "lastAppliedWallTime" : ISODate("2022-03-23T13:45:52.894Z"),
                "lastDurableWallTime" : ISODate("2022-03-23T13:45:52.894Z")
        },
        "lastStableRecoveryTimestamp" : Timestamp(1648043113, 1),
```

```
        },
        "lastStableRecoveryTimestamp" : Timestamp(1648043113, 1),
        "electionCandidateMetrics" : {
                "lastElectionReason" : "electionTimeout",
                "lastElectionDate" : ISODate("2022-03-23T13:40:32.856Z"),
                "electionTerm" : NumberLong(3),
                "lastCommittedOpTimeAtElection" : {
                        "ts" : Timestamp(0, 0),
                        "t" : NumberLong(-1)
                },
                "lastSeenOpTimeAtElection" : {
                        "ts" : Timestamp(1648042812, 1),
                        "t" : NumberLong(2)
                },
                "numVotesNeeded" : 1,
                "priorityAtElection" : 1,
                "electionTimeoutMillis" : NumberLong(10000),
                "numCatchUpOps" : NumberLong(0),
                "newTermStartDate" : ISODate("2022-03-23T13:40:32.861Z"),
                "wMajorityWriteAvailabilityDate" : ISODate("2022-03-23T13:40:32.865Z")
        },
        "members" : [
                {
                        "_id" : 0,
                        "name" : "localhost:27001",
                        "health" : 1,
                        "state" : 1,
                        "stateStr" : "PRIMARY",
                        "uptime" : 339,
                        "optime" : {
                                "ts" : Timestamp(1648043152, 1),
                                "t" : NumberLong(3)
                        },
                        "optimeDate" : ISODate("2022-03-23T13:45:52Z"),
                        "lastAppliedWallTime" : ISODate("2022-03-23T13:45:52.894Z"),
                        "lastDurableWallTime" : ISODate("2022-03-23T13:45:52.894Z"),
                        "syncSourceHost" : "",
                        "syncSourceId" : -1,
```

```
                "lastDurableWallTime" : ISODate("2022-03-23T13:45:52.894Z"),
                "syncSourceHost" : "",
                "syncSourceId" : -1,
                "infoMessage" : "",
                "electionTime" : Timestamp(1648042832, 1),
                "electionDate" : ISODate("2022-03-23T13:40:32Z"),
                "configVersion" : 5,
                "configTerm" : 3,
                "self" : true,
                "lastHeartbeatMessage" : ""
        },
        {
                "_id" : 1,
                "name" : "localhost:27002",
                "health" : 1,
                "state" : 2,
                "stateStr" : "SECONDARY",
                "uptime" : 119,
                "optime" : {
                        "ts" : Timestamp(1648043152, 1),
                        "t" : NumberLong(3)
                },
                "optimeDurable" : {
                        "ts" : Timestamp(1648043152, 1),
                        "t" : NumberLong(3)
                },
                "optimeDate" : ISODate("2022-03-23T13:45:52Z"),
                "optimeDurableDate" : ISODate("2022-03-23T13:45:52Z"),
                "lastAppliedWallTime" : ISODate("2022-03-23T13:45:52.894Z"),
                "lastDurableWallTime" : ISODate("2022-03-23T13:45:52.894Z"),
                "lastHeartbeat" : ISODate("2022-03-23T13:45:57.348Z"),
                "lastHeartbeatRecv" : ISODate("2022-03-23T13:45:57.351Z"),
                "pingMs" : NumberLong(0),
                "lastHeartbeatMessage" : "",
                "syncSourceHost" : "localhost:27001",
                "syncSourceId" : 0,
                "infoMessage" : "",
                "configVersion" : 5,
                "configTerm" : 3
        },
        {
                "_id" : 2,
```

```
                {
                        "_id" : 2,
                        "name" : "localhost:27003",
                        "health" : 1,
                        "state" : 2,
                        "stateStr" : "SECONDARY",
                        "uptime" : 47,
                        "optime" : {
                                "ts" : Timestamp(1648043152, 1),
                                "t" : NumberLong(3)
                        },
                        "optimeDurable" : {
                                "ts" : Timestamp(1648043152, 1),
                                "t" : NumberLong(3)
                        },
                        "optimeDate" : ISODate("2022-03-23T13:45:52Z"),
                        "optimeDurableDate" : ISODate("2022-03-23T13:45:52Z"),
                        "lastAppliedWallTime" : ISODate("2022-03-23T13:45:52.894Z"),
                        "lastDurableWallTime" : ISODate("2022-03-23T13:45:52.894Z"),
                        "lastHeartbeat" : ISODate("2022-03-23T13:45:57.347Z"),
                        "lastHeartbeatRecv" : ISODate("2022-03-23T13:45:58.353Z"),
                        "pingMs" : NumberLong(0),
                        "lastHeartbeatMessage" : "",
                        "syncSourceHost" : "localhost:27002",
                        "syncSourceId" : 1,
                        "infoMessage" : "",
                        "configVersion" : 5,
                        "configTerm" : 3
                }
        ],
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1648043152, 1),
                "signature" : {
                        "hash" : BinData(0,"woKHpSMIjl2hHygLg2rjbrqlRdI="),
                        "keyId" : NumberLong("7078273427143917573")
                }
        },
        "operationTime" : Timestamp(1648043152, 1)
}
repl:PRIMARY>
```

63. Getting an overview of the replica set topology

```
repl:PRIMARY> rs.isMaster()
{
        "topologyVersion" : {
                "processId" : ObjectId("623b234414a9689fc3e27c53"),
                "counter" : NumberLong(8)
        },
        "hosts" : [
                "localhost:27001",
                "localhost:27002",
                "localhost:27003"
        ],
        "setName" : "repl",
        "setVersion" : 5,
        "ismaster" : true,
        "secondary" : false,
        "primary" : "localhost:27001",
        "me" : "localhost:27001",
        "electionId" : ObjectId("7fffffff0000000000000003"),
        "lastWrite" : {
                "opTime" : {
                        "ts" : Timestamp(1648043342, 1),
                        "t" : NumberLong(3)
                },
                "lastWriteDate" : ISODate("2022-03-23T13:49:02Z"),
                "majorityOpTime" : {
                        "ts" : Timestamp(1648043342, 1),
                        "t" : NumberLong(3)
                },
                "majorityWriteDate" : ISODate("2022-03-23T13:49:02Z")
        },
        "maxBsonObjectSize" : 16777216,
        "maxMessageSizeBytes" : 48000000,
        "maxWriteBatchSize" : 100000,
        "localTime" : ISODate("2022-03-23T13:49:04.523Z"),
        "logicalSessionTimeoutMinutes" : 30,
        "connectionId" : 25,
        "minWireVersion" : 0,
        "maxWireVersion" : 13,
        "readOnly" : false,
        "ok" : 1,
        "$clusterTime" : {
```

```
        "$clusterTime" : {
                "clusterTime" : Timestamp(1648043342, 1),
                "signature" : {
                        "hash" : BinData(0,"BuTxLEi8TSvdr0aHY5sj0+mDirU="),
                        "keyId" : NumberLong("7078273427143917573")
                }
        },
        "operationTime" : Timestamp(1648043342, 1)
}
repl:PRIMARY>
```

64. Stepping down the current primary which will initiate voting

```
repl:PRIMARY> rs.stepDown()
{
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1648043432, 1),
                "signature" : {
                        "hash" : BinData(0,"LA7lTHUd1i8ugp3BKrhZzOdo+64="),
                        "keyId" : NumberLong("7078273427143917573")
                }
        },
        "operationTime" : Timestamp(1648043432, 1)
}
repl:PRIMARY>
```

65. Checking replica set overview after election

```
repl:PRIMARY> rs.isMaster()
{
        "topologyVersion" : {
                "processId" : ObjectId("623b241ebc44a516a6f9fb59"),
                "counter" : NumberLong(8)
        },
        "hosts" : [
                "localhost:27001",
                "localhost:27002",
                "localhost:27003"
        ],
        "setName" : "repl",
        "setVersion" : 5,
        "ismaster" : true,
        "secondary" : false,
        "primary" : "localhost:27002",
        "me" : "localhost:27002",
        "electionId" : ObjectId("7fffffff0000000000000004"),
        "lastWrite" : {
                "opTime" : {
                        "ts" : Timestamp(1648043478, 1),
                        "t" : NumberLong(4)
                },
                "lastWriteDate" : ISODate("2022-03-23T13:51:18Z"),
                "majorityOpTime" : {
                        "ts" : Timestamp(1648043478, 1),
                        "t" : NumberLong(4)
                },
                "majorityWriteDate" : ISODate("2022-03-23T13:51:18Z")
        },
        "maxBsonObjectSize" : 16777216,
        "maxMessageSizeBytes" : 48000000,
        "maxWriteBatchSize" : 100000,
        "localTime" : ISODate("2022-03-23T13:51:20.964Z"),
        "logicalSessionTimeoutMinutes" : 30,
        "connectionId" : 41,
        "minWireVersion" : 0,
        "maxWireVersion" : 13,
```

```
        "logicalSessionTimeoutMinutes" : 30,
        "connectionId" : 41,
        "minWireVersion" : 0,
        "maxWireVersion" : 13,
        "readOnly" : false,
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1648043478, 1),
                "signature" : {
                        "hash" : BinData(0,"BHkNmi2T/kQTnqvr82mzZDgAetA="),
                        "keyId" : NumberLong("7078273427143917573")
                }
        },
        "operationTime" : Timestamp(1648043478, 1)
}
repl:PRIMARY>
```

SHARDING

66. Creating csrs_1.conf

```
root@3c0e975d3a4c:/# nano csrs_1.conf
root@3c0e975d3a4c:/#
```

root@3c0e975d3a4c: /

```
  GNU nano 4.8
sharding:
  clusterRole: configsvr
replication:
  replSetName: m103-csrs
security:
  keyFile: /var/mongodb/pki/m103-keyfile
net:
  bindIp: localhost
  port: 27004
systemLog:
  destination: file
  path: /var/mongodb/db/csrs1.log
  logAppend: true
processManagement:
  fork: true
storage:
  dbPath: /var/mongodb/db/csrs1
```

67. Creating csrs_2.conf

```
root@3c0e975d3a4c:/# nano csrs_2.conf
```

```
root@3c0e975d3a4c: /
  GNU nano 4.8
sharding:
  clusterRole: configsvr
replication:
  replSetName: m103-csrs
security:
  keyFile: /var/mongodb/pki/m103-keyfile
net:
  bindIp: localhost
  port: 27005
systemLog:
  destination: file
  path: /var/mongodb/db/csrs2.log
  logAppend: true
processManagement:
  fork: true
storage:
  dbPath: /var/mongodb/db/csrs2
```

68. Creating csrs_3.conf

```
root@3c0e975d3a4c:/# nano csrs_3.conf
```

```
root@3c0e975d3a4c: /
  GNU nano 4.8
sharding:
  clusterRole: configsvr
replication:
  replSetName: m103-csrs
security:
  keyFile: /var/mongodb/pki/m103-keyfile
net:
  bindIp: localhost
  port: 27006
systemLog:
  destination: file
  path: /var/mongodb/db/csrs3.log
  logAppend: true
processManagement:
  fork: true
storage:
  dbPath: /var/mongodb/db/csrs3
```

69. Creating directories for 3 configservers

```
root@3c0e975d3a4c:/# mkdir -p /var/mongodb/db/csrs1
root@3c0e975d3a4c:/# mkdir -p /var/mongodb/db/csrs2
root@3c0e975d3a4c:/# mkdir -p /var/mongodb/db/csrs3
```

70. Staring config server 1

```
root@3c0e975d3a4c:/# mongod -f csrs_1.conf
about to fork child process, waiting until server is ready for connections.
forked process: 1445
child process started successfully, parent exiting
```

71. Staring config server 2

```
root@3c0e975d3a4c:/# mongod -f csrs_2.conf
about to fork child process, waiting until server is ready for connections.
forked process: 1514
child process started successfully, parent exiting
root@3c0e975d3a4c:/#
```

72. Staring configserver 3

```
root@3c0e975d3a4c:/# mongod -f csrs_3.conf
about to fork child process, waiting until server is ready for connections.
forked process: 1600
child process started successfully, parent exiting
root@3c0e975d3a4c:/#
```

73. Connect to one of the config servers

```
root@3c0e975d3a4c:/# mongo --port 27004
MongoDB shell version v5.0.6
connecting to: mongodb://127.0.0.1:27004/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("7b86fc37-ca49-461e-bb86-5cf4dd773ce3") }
MongoDB server version: 5.0.6
================
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
================
>
```

74. Initiating the CSRS

```
> rs.initiate()
{
        "info2" : "no configuration specified. Using a default configuration for the set",
        "me" : "localhost:27004",
        "ok" : 1,
        "$gleStats" : {
                "lastOpTime" : Timestamp(1648048641, 1),
                "electionId" : ObjectId("000000000000000000000000")
        },
        "lastCommittedOpTime" : Timestamp(1648048641, 1)
}
m103-csrs:SECONDARY>
```

75. Creating super user on CSRS

```
m103-csrs:SECONDARY> use admin
switched to db admin
m103-csrs:PRIMARY> db.createUser({
...    user: "m103-admin",
...    pwd: "m103-pass",
...    roles: [
...      {role: "root", db: "admin"}
...    ]
... })
Successfully added user: {
        "user" : "m103-admin",
        "roles" : [
                {
                        "role" : "root",
                        "db" : "admin"
                }
        ]
}
m103-csrs:PRIMARY>
```

76. Authenticating as the super user

```
m103-csrs:PRIMARY> db.auth("m103-admin", "m103-pass")
1
```

77. Add the second node to the CSRS

```
m103-csrs:PRIMARY> rs.add("localhost:27005")
{
        "ok" : 1,
        "$gleStats" : {
                "lastOpTime" : {
                        "ts" : Timestamp(1648048852, 2),
                        "t" : NumberLong(1)
                },
                "electionId" : ObjectId("7fffffff0000000000000001")
        },
        "lastCommittedOpTime" : Timestamp(1648048852, 2),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1648048852, 2),
                "signature" : {
                        "hash" : BinData(0,"jTW7KTsqXIbu6WWhYT6D1l2EwmY="),
                        "keyId" : NumberLong("7078315019607212036")
                }
        },
        "operationTime" : Timestamp(1648048852, 2)
}
m103-csrs:PRIMARY>
```

78. Add the third node to the CSRS

```
m103-csrs:PRIMARY> rs.add("localhost:27006")
{
        "ok" : 1,
        "$gleStats" : {
                "lastOpTime" : {
                        "ts" : Timestamp(1648048915, 1),
                        "t" : NumberLong(1)
                },
                "electionId" : ObjectId("7fffffff0000000000000001")
        },
        "lastCommittedOpTime" : Timestamp(1648048916, 1),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1648048916, 1),
                "signature" : {
                        "hash" : BinData(0,"tT32LQuikvVvZNZS/UQmK1D2viI="),
                        "keyId" : NumberLong("7078315019607212036")
                }
        },
        "operationTime" : Timestamp(1648048915, 1)
}
m103-csrs:PRIMARY>
```

79. Creating new nodes1.conf for sharding

```
root@3c0e975d3a4c: /
  GNU nano 4.8
sharding:
  clusterRole: shardsvr
storage:
  dbPath: /var/mongodb/db/node1
  wiredTiger:
    engineConfig:
      cacheSizeGB: .25
net:
  bindIp: localhost
  port: 27007
security:
  keyFile: /var/mongodb/pki/m103-keyfile
systemLog:
  destination: file
  path: /var/mongodb/db/node1/mongod.log
  logAppend: true
processManagement:
  fork: true
replication:
  replSetName: m103-repl
```

80. Creating directories

```
root@3c0e975d3a4c:/# mkdir -p /var/mongodb/db/node1
root@3c0e975d3a4c:/# mongod -f nodes1.conf
```

81. Starting/running new nodes1

```
root@3c0e975d3a4c:/# mongod -f nodes1.conf
about to fork child process, waiting until server is ready for connections.
forked process: 2042
child process started successfully, parent exiting
root@3c0e975d3a4c:/#
```

82. Creating nodes2.conf

```
root@3c0e975d3a4c:/# nano nodes2.conf
```

root@3c0e975d3a4c: /

```
  GNU nano 4.8
sharding:
  clusterRole: shardsvr
storage:
  dbPath: /var/mongodb/db/node2
  wiredTiger:
    engineConfig:
      cacheSizeGB: .25
net:
  bindIp: localhost
  port: 27008
security:
  keyFile: /var/mongodb/pki/m103-keyfile
systemLog:
  destination: file
  path: /var/mongodb/db/node2/mongod.log
  logAppend: true
processManagement:
  fork: true
replication:
  replSetName: m103-repl
```

83. Creating directory and running nodes2

```
root@3c0e975d3a4c:/# mkdir -p /var/mongodb/db/node2
root@3c0e975d3a4c:/# mongod -f nodes2.conf
about to fork child process, waiting until server is ready for connections.
forked process: 2144
child process started successfully, parent exiting
root@3c0e975d3a4c:/#
```

84. Creating nodes3.cong

root@3c0e975d3a4c: /

```
  GNU nano 4.8
sharding:
  clusterRole: shardsvr
storage:
  dbPath: /var/mongodb/db/node3
  wiredTiger:
    engineConfig:
      cacheSizeGB: .25
net:
  bindIp: localhost
  port: 27009
security:
  keyFile: /var/mongodb/pki/m103-keyfile
systemLog:
  destination: file
  path: /var/mongodb/db/node3/mongod.log
  logAppend: true
processManagement:
  fork: true
replication:
  replSetName: m103-repl
```

85. Running nodes3

```
root@3c0e975d3a4c:/# mkdir -p /var/mongodb/db/node3
root@3c0e975d3a4c:/# mongod -f nodes3.conf
about to fork child process, waiting until server is ready for connections.
forked process: 2247
child process started successfully, parent exiting
root@3c0e975d3a4c:/#
```

86. Connecting to prot 27007 of nodes

```
root@3c0e975d3a4c:/# mongo --port 27007
MongoDB shell version v5.0.6
connecting to: mongodb://127.0.0.1:27007/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("fc75b706-b7cf-4a1d-aa56-b8829143ff5d") }
MongoDB server version: 5.0.6
================
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
================
>
```

87. Starting a replica set

```
> rs.initiate()
{
        "info2" : "no configuration specified. Using a default configuration for the set",
        "me" : "localhost:27007",
        "ok" : 1
}
m103-repl:SECONDARY>
```

88. Creating a user

```
m103-repl:SECONDARY> use admin
switched to db admin
m103-repl:PRIMARY> db.createUser({
...     user: "m103-admin",
...     pwd: "m103-pass",
...     roles: [
...       {role: "root", db: "admin"}
...     ]
... })
Successfully added user: {
        "user" : "m103-admin",
        "roles" : [
                {
                        "role" : "root",
                        "db" : "admin"
                }
        ]
}
m103-repl:PRIMARY>
```

89. Authenticating

```
m103-repl:PRIMARY> db.auth("m103-admin", "m103-pass")
1
m103-repl:PRIMARY>
```

90. Current status

```
m103-repl:PRIMARY> rs.status()
{
        "set" : "m103-repl",
        "date" : ISODate("2022-03-23T15:52:46.889Z"),
        "myState" : 1,
        "term" : NumberLong(1),
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "heartbeatIntervalMillis" : NumberLong(2000),
        "majorityVoteCount" : 1,
        "writeMajorityCount" : 1,
        "votingMembersCount" : 1,
        "writableVotingMembersCount" : 1,
        "optimes" : {
                "lastCommittedOpTime" : {
                        "ts" : Timestamp(1648050759, 1),
                        "t" : NumberLong(1)
                },
                "lastCommittedWallTime" : ISODate("2022-03-23T15:52:39.463Z"),
                "readConcernMajorityOpTime" : {
                        "ts" : Timestamp(1648050759, 1),
                        "t" : NumberLong(1)
                },
                "appliedOpTime" : {
                        "ts" : Timestamp(1648050759, 1),
                        "t" : NumberLong(1)
```

91. Adding the nodes2 and nodes3 and seeing new status

```
m103-repl:PRIMARY> rs.add("localhost:27008")
{ "ok" : 1 }
m103-repl:PRIMARY> rs.add("localhost:27009")
{ "ok" : 1 }
m103-repl:PRIMARY> rs.status()
{
        "set" : "m103-repl",
        "date" : ISODate("2022-03-23T15:54:39.524Z"),
        "myState" : 1,
        "term" : NumberLong(1),
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "heartbeatIntervalMillis" : NumberLong(2000),
        "majorityVoteCount" : 2,
        "writeMajorityCount" : 2,
        "votingMembersCount" : 3,
        "writableVotingMembersCount" : 3,
        "optimes" : {
                "lastCommittedOpTime" : {
                        "ts" : Timestamp(1648050871, 1),
                        "t" : NumberLong(1)
                },
                "lastCommittedWallTime" : ISODate("2022-03-23T15:54:31.691Z"),
                "readConcernMajorityOpTime" : {
                        "ts" : Timestamp(1648050871, 1),
                        "t" : NumberLong(1)
                },
                "appliedOpTime" : {
                        "ts" : Timestamp(1648050871, 1),
                        "t" : NumberLong(1)
```

92. Stepping down and changing primary by voting and seeing results

```
m103-repl:PRIMARY> rs.isMaster()
{
        "topologyVersion" : {
                "processId" : ObjectId("623b3f10bbd5775a801a0952"),
                "counter" : NumberLong(9)
        },
        "hosts" : [
                "localhost:27007",
                "localhost:27008",
                "localhost:27009"
        ],
        "setName" : "m103-repl",
        "setVersion" : 3,
        "ismaster" : true,
        "secondary" : false,
        "primary" : "localhost:27007",
        "me" : "localhost:27007",
        "electionId" : ObjectId("7fffffff0000000000000001"),
        "lastWrite" : {
                "opTime" : {
                        "ts" : Timestamp(1648050959, 1),
                        "t" : NumberLong(1)
                },
                "lastWriteDate" : ISODate("2022-03-23T15:55:59Z"),
```

```
m103-repl:PRIMARY> rs.stepDown()
{ "ok" : 1 }
m103-repl:SECONDARY> rs.isMaster()
{
        "topologyVersion" : {
                "processId" : ObjectId("623b3f10bbd5775a801a0952"),
                "counter" : NumberLong(12)
        },
        "hosts" : [
                "localhost:27007",
                "localhost:27008",
                "localhost:27009"
        ],
        "setName" : "m103-repl",
        "setVersion" : 3,
        "ismaster" : false,
        "secondary" : true,
        "primary" : "localhost:27008",
        "me" : "localhost:27007",
        "lastWrite" : {
                "opTime" : {
                        "ts" : Timestamp(1648050995, 1),
                        "t" : NumberLong(2)
                },
                "lastWriteDate" : ISODate("2022-03-23T15:56:35Z"),
                "majorityOpTime" : {
                        "ts" : Timestamp(1648050995, 1),
                        "t" : NumberLong(2)
```

93. Creating mongos.conf

```
root@3c0e975d3a4c: /
  GNU nano 4.8
security:
  keyFile: /var/mongodb/pki/m103-keyfile
net:
  bindIp: localhost
  port: 27000
systemLog:
  destination: file
  path: /var/mongodb/logs/mongos.log
  logAppend: true
processManagement:
  fork: true
sharding:
  configDB: csrs/localhost:27004,localhost:27005,localhost:27006
```

94. Running mongos

```
root@3c0e975d3a4c:/# nano mongos.conf
root@3c0e975d3a4c:/# mongos -f mongos.conf
about to fork child process, waiting until server is ready for connections.
forked process: 2616
child process started successfully, parent exiting
root@3c0e975d3a4c:/#
```

95. Connect to mongos

```
root@3c0e975d3a4c:/# mongo --port 27000 --username m103-admin --password m103-pass --authenticationDatabase admin
MongoDB shell version v5.0.6
connecting to: mongodb://127.0.0.1:27000/?authSource=admin&compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("a3ae9a27-0c15-47ea-94e3-e8fc01546d33") }
MongoDB server version: 5.0.6
================
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
================
---
The server generated these startup warnings when booting:
        2022-03-23T16:03:18.597+00:00: You are running this process as the root user, which is not recommended
---
mongos>
```

96. Check sharding status

```
mongos>  sh.status()
--- Sharding Status ---
  sharding version: {
        "_id" : 1,
        "minCompatibleVersion" : 5,
        "currentVersion" : 6,
        "clusterId" : ObjectId("623b3a026ba3b6962bc14742")
  }
  shards:
  active mongoses:
  autosplit:
        Currently enabled: yes
  balancer:
        Currently enabled: yes
        Currently running: no
        Failed balancer rounds in last 5 attempts: 0
        Migration results for the last 24 hours:
                No recent migrations
  databases:
        {  "_id" : "config",  "primary" : "config",  "partitioned" : true }
mongos>
```

97. Adding new shard to cluster from mongos

```
mongos>  sh.status()
--- Sharding Status ---
  sharding version: {
        "_id" : 1,
        "minCompatibleVersion" : 5,
        "currentVersion" : 6,
        "clusterId" : ObjectId("623b3a026ba3b6962bc14742")
  }
  shards:
  active mongoses:
  autosplit:
        Currently enabled: yes
  balancer:
        Currently enabled: yes
        Currently running: no
        Failed balancer rounds in last 5 attempts: 0
        Migration results for the last 24 hours:
                No recent migrations
  databases:
        {  "_id" : "config",  "primary" : "config",  "partitioned" : true }
mongos> sh.addShard("m103-repl/localhost:27008")
{
        "shardAdded" : "m103-repl",
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1648051834, 5),
                "signature" : {
                        "hash" : BinData(0,"WKHz+GIIP89kWi7nrCecD3bbGv8="),
                        "keyId" : NumberLong("7078315019607212036")
                }
        },
        "operationTime" : Timestamp(1648051834, 5)
}
mongos>
```

98. The shard status

```
mongos> sh.status()
--- Sharding Status ---
  sharding version: {
        "_id" : 1,
        "minCompatibleVersion" : 5,
        "currentVersion" : 6,
        "clusterId" : ObjectId("623b3a026ba3b6962bc14742")
  }
  shards:
        {  "_id" : "m103-repl",  "host" : "m103-repl/localhost:27007,localhost:27008,localhost:27009",  "state" : 1,  "topologyTime" : Timestamp(1648051834, 2) }
  active mongoses:
        "5.0.6" : 1
  autosplit:
        Currently enabled: yes
  balancer:
        Currently enabled: yes
        Currently running: no
        Failed balancer rounds in last 5 attempts: 0
        Migration results for the last 24 hours:
                No recent migrations
  databases:
        {  "_id" : "config",  "primary" : "config",  "partitioned" : true }
mongos>
```

99. Importing new accounts.json in mongos port to shard

mongoimport --port 27000 --host localhost --authenticationDatabase=admin -u m103-admin -p m103-pass --db m103 --collection accounts --file ./accounts.json

```
root@3c0e975d3a4c:/data/db# mongoimport --port 27000 --host localhost --authenticationDatabase=admin -u m103-admin -p m103-pass --db m103 --collection accounts --file ./acc
ounts.json
2022-03-23T16:33:25.891+0000    connected to: mongodb://localhost:27000/
2022-03-23T16:33:26.371+0000    1746 document(s) imported successfully. 0 document(s) failed to import.
root@3c0e975d3a4c:/data/db#
```

100. Connecting to Mongos server

```
root@3c0e975d3a4c:/# mongo --port 27000 --username m103-admin --password m103-pass --authenticationDatabase admin
MongoDB shell version v5.0.6
connecting to: mongodb://127.0.0.1:27000/?authSource=admin&compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("46edb451-09e9-4069-b310-9363fac2bfe1") }
MongoDB server version: 5.0.6
================
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
================
---
The server generated these startup warnings when booting:
        2022-03-23T16:03:18.597+00:00: You are running this process as the root user, which is not recommended
---
mongos>
```

101. Seeing our imported db and collection

```
mongos> show dbs
admin   0.000GB
config  0.001GB
m103    0.000GB
mongos> use m103
switched to db m103
mongos> show collections
accounts
mongos>
```

102. Checking account collection

```
mongos> db.accounts.findOne()
{
        "_id" : ObjectId("5ca4bbc7a2dd94ee58162393"),
        "account_id" : 328304,
        "limit" : 10000,
        "products" : [
                "Derivatives",
                "InvestmentStock",
                "CurrencyService"
        ]
}
```

103. Creating index on account_id for sharding

```
mongos> db.accounts.createIndex( { "account_id": 1 } )
{
        "raw" : {
                "m103-repl/localhost:27007,localhost:27008,localhost:27009" : {
                        "numIndexesBefore" : 1,
                        "numIndexesAfter" : 2,
                        "createdCollectionAutomatically" : false,
                        "commitQuorum" : "votingMembers",
                        "ok" : 1
                }
        },
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1648053566, 7),
                "signature" : {
                        "hash" : BinData(0,"6RJp7pAF/riIfc0bQ/qBcn7MxqE="),
                        "keyId" : NumberLong("7078315019607212036")
                }
        },
        "operationTime" : Timestamp(1648053566, 7)
}
mongos>
```

104. Enabling sharding on our database

```
mongos> sh.enableSharding("m103")
{
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1648053901, 1),
                "signature" : {
                        "hash" : BinData(0,"yX6Ug+VnE1HkTpbgN3IFNMz8L3Y="),
                        "keyId" : NumberLong("7078315019607212036")
                }
        },
        "operationTime" : Timestamp(1648053901, 1)
}
mongos>
```

105. Sharding our db accounts collection

```
mongos> sh.shardCollection( "m103.accounts", { "account_id": 1 } )
{
        "collectionsharded" : "m103.accounts",
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1648053964, 25),
                "signature" : {
                        "hash" : BinData(0,"PO8e9uAUiyhnYrt6j1NNxtX+zRg="),
                        "keyId" : NumberLong("7078315019607212036")
                }
        },
        "operationTime" : Timestamp(1648053964, 21)
}
mongos>
```

106. Sharding status

```
mongos> sh.status()
--- Sharding Status ---
  sharding version: {
        "_id" : 1,
        "minCompatibleVersion" : 5,
        "currentVersion" : 6,
        "clusterId" : ObjectId("623b3a026ba3b6962bc14742")
  }
  shards:
        {  "_id" : "m103-repl",  "host" : "m103-repl/localhos
  active mongoses:
        "5.0.6" : 1
  autosplit:
        Currently enabled: yes
  balancer:
        Currently enabled: yes
        Currently running: no
        Failed balancer rounds in last 5 attempts: 0
        Migration results for the last 24 hours:
                No recent migrations
```

```
mongos> sh.status()
--- Sharding Status ---
  sharding version: {
        "_id" : 1,
        "minCompatibleVersion" : 5,
        "currentVersion" : 6,
        "clusterId" : ObjectId("623b3a026ba3b6962bc14742")
  }
  shards:
        {  "_id" : "m103-repl",  "host" : "m103-repl/localhost:27007,localhost:27008,localhost:27009",  "state" : 1,  "topologyTime" : Timestamp(1648051834, 2) }
  active mongoses:
        "5.0.6" : 1
  autosplit:
        Currently enabled: yes
  balancer:
        Currently enabled: yes
        Currently running: no
        Failed balancer rounds in last 5 attempts: 0
        Migration results for the last 24 hours:
                No recent migrations
  databases:
        {  "_id" : "config",  "primary" : "config",  "partitioned" : true }
                config.system.sessions
                        shard key: { "_id" : 1 }
                        unique: false
                        balancing: true
                        chunks:
                                m103-repl       1024
                        too many chunks to print, use verbose if you want to force print
        {  "_id" : "m103",  "primary" : "m103-repl",  "partitioned" : true,  "version" : {  "uuid" : UUID("3303ae34-bd37-4bb6-b152-a65c096e32c7"),  "timestamp" : Timestamp(
1648053205, 1),  "lastMod" : 1 } }
                m103.accounts
                        shard key: { "account_id" : 1 }
                        unique: false
                        balancing: true
                        chunks:
                                m103-repl       1
                        { "account_id" : { "$minKey" : 1 } } -->> { "account_id" : { "$maxKey" : 1 } } on : m103-repl Timestamp(1, 0)
mongos>
```