

IST769 Homework Submission

Name: **Sathish Kumar Rajendiran**

SUID: **666555028**

Email: srajendi@syr.edu

Due Date: **08/31/2021**

Task: **MongoDB and Redis**

Homework #:**8**

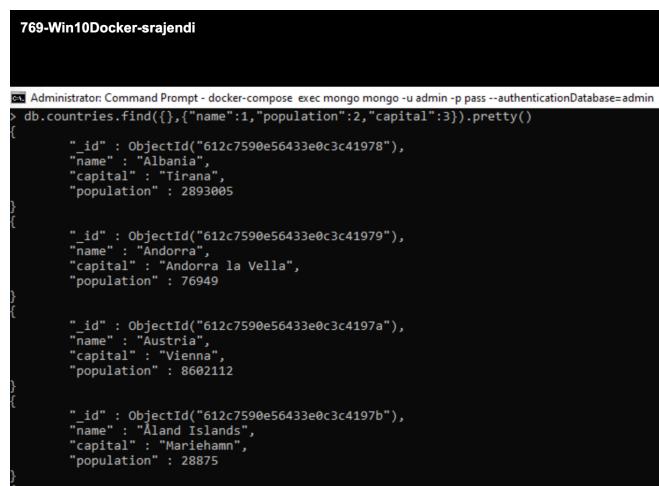
Exercise(s):

1. Write a MongoDB Query to retrieve Country name, population, and capital for all countries in the collection.

Solution:

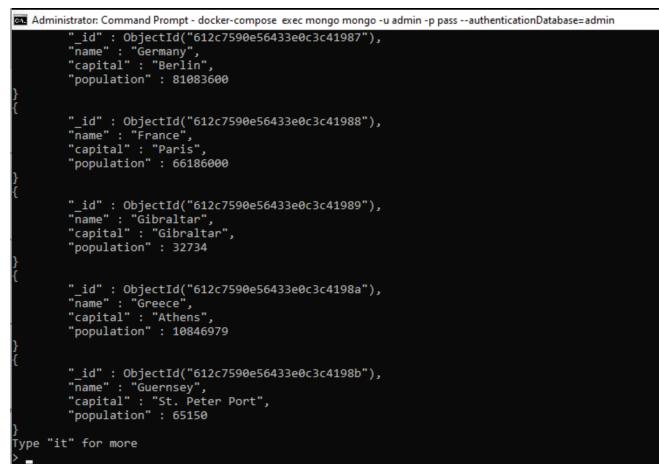
```
-- MongoDB Query to retrieve Country name, population, and capital for all countries
db.countries.find({}, {"name":1,"population":2,"capital":3}).pretty()
```

Evidence:



```
769-Win10Docker-srajendi

Administrator: Command Prompt - docker-compose exec mongo mongo -u admin -p pass --authenticationDatabase=admin
> db.countries.find({}, {"name":1,"population":2,"capital":3}).pretty()
{
    "_id" : ObjectId("612c7590e56433e0c3c41978"),
    "name" : "Albania",
    "capital" : "Tirana",
    "population" : 2893005
}
{
    "_id" : ObjectId("612c7590e56433e0c3c41979"),
    "name" : "Andorra",
    "capital" : "Andorra la Vella",
    "population" : 76949
}
{
    "_id" : ObjectId("612c7590e56433e0c3c4197a"),
    "name" : "Austria",
    "capital" : "Vienna",
    "population" : 8602112
}
{
    "_id" : ObjectId("612c7590e56433e0c3c4197b"),
    "name" : "Åland Islands",
    "capital" : "Mariehamn",
    "population" : 28875
}
```



```
Administrator: Command Prompt - docker-compose exec mongo mongo -u admin -p pass --authenticationDatabase=admin
{
    "_id" : ObjectId("612c7590e56433e0c3c41987"),
    "name" : "Germany",
    "capital" : "Berlin",
    "population" : 81083600
}
{
    "_id" : ObjectId("612c7590e56433e0c3c41988"),
    "name" : "France",
    "capital" : "Paris",
    "population" : 66186000
}
{
    "_id" : ObjectId("612c7590e56433e0c3c41989"),
    "name" : "Gibraltar",
    "capital" : "Gibraltar",
    "population" : 32734
}
{
    "_id" : ObjectId("612c7590e56433e0c3c4198a"),
    "name" : "Greece",
    "capital" : "Athens",
    "population" : 10846979
}
{
    "_id" : ObjectId("612c7590e56433e0c3c4198b"),
    "name" : "Guernsey",
    "capital" : "St. Peter Port",
    "population" : 65150
}
Type "it" for more
> -
```

2. Write a MongoDB Query to retrieve Country name, population, and capital for all countries with a population under 500,000 sorted by population.

Solution:

```
-- MongoDB Query to retrieve Country name, population, and capital for all countries with
-- population less than 500,000
db.countries.find({"population":{$lt:500000}, {"name":1,"population":2,"capital":3}}).sort({"popul
ation":1}).pretty()
```

Evidence:

```
Administrator: Command Prompt - docker-compose exec mongo mongo -u admin -p pass --authenticationDatabase=admin
db.countries.find({"population":{$lt:500000}}, {"name":1,"population":2,"capital":3}).sort({"population":1}).pretty()
{
    "_id" : ObjectId("612c7590e56433e0c3c4198d"),
    "name" : "Holy See",
    "capital" : "Rome",
    "population" : 451
}
{
    "_id" : ObjectId("612c7590e56433e0c3c419a8"),
    "name" : "Svalbard and Jan Mayen",
    "capital" : "Longyearbyen",
    "population" : 2562
}
{
    "_id" : ObjectId("612c7590e56433e0c3c4197b"),
    "name" : "Aland Islands",
    "capital" : "Mariehamn",
    "population" : 28875
}
{
    "_id" : ObjectId("612c7590e56433e0c3c41989"),
    "name" : "Gibraltar",
    "capital" : "Gibraltar",
    "population" : 32734
}
{
    "_id" : ObjectId("612c7590e56433e0c3c419a4"),
    "name" : "San Marino",
    "capital" : "City of San Marino",
    "population" : 32831
}
{
    "_id" : ObjectId("612c7590e56433e0c3c41994"),
    "name" : "Liechtenstein",
    "capital" : "Vaduz",
    "population" : 37370
}
```

```
Administrator: Command Prompt - docker-compose exec mongo mongo -u admin -p pass --authenticationDatabase=admin
{
    "name" : "Andorra",
    "capital" : "Andorra la Vella",
    "population" : 76949
}
{
    "_id" : ObjectId("612c7590e56433e0c3c4198f"),
    "name" : "Isle of Man",
    "capital" : "Douglas",
    "population" : 84497
}
{
    "_id" : ObjectId("612c7590e56433e0c3c41993"),
    "name" : "Jersey",
    "capital" : "Saint Helier",
    "population" : 99000
}
{
    "_id" : ObjectId("612c7590e56433e0c3c4198e"),
    "name" : "Iceland",
    "capital" : "Reykjavík",
    "population" : 330610
}
{
    "_id" : ObjectId("612c7590e56433e0c3c41997"),
    "name" : "Malta",
    "capital" : "Valletta",
    "population" : 445426
}
>
```

3. Use the `.explain("executionStats")` method to analyze the query you wrote in the previous step. Write an index to improve the performance of the query, then perform another explain to demonstrate it worked. Include the code of the index you wrote, the and the relevant output of the execution stats which demonstrate the index is being used.

Solution:

```
-- MongoDB Query to retrieve execution stats
db.countries.find({"population":{$lt:500000}, {"name":1, "population":2, "capital":3}}).sort({"population":1}).explain("executionStats")

-- MongoDB Query to create index on population key
db.countries.createIndex({population:1})

-- MongoDB Query to retrieve execution stats – after index creation
db.countries.find({"population":{$lt:500000}, {"name":1, "population":2, "capital":3}}).sort({"population":1}).explain("executionStats")
```

Evidence:

```
Administrator: Command Prompt - docker-compose exec mongo mongo -u admin -p pass --authenticationDatabase=admin
> db.countries.find({"population":{$lt:500000}}, {"name":1, "population":2, "capital":3}).sort({"population":1}).explain("executionStats")
{
    "queryPlanner": {
        "plannerVersion": 1,
        "namespace": "demo.countries",
        "indexFilterSet": false,
        "parsedQuery": {
            "population": {
                "$lt": 500000
            }
        },
        "winningPlan": {
            "stage": "PROJECTION",
            "transformBy": {
                "name": 1,
                "population": 2,
                "capital": 3
            },
            "inputStage": {
                "stage": "SORT",
                "sortPattern": {
                    "population": 1
                },
                "inputStage": {
                    "stage": "SORT_KEY_GENERATOR",
                    "inputStage": {
                        "stage": "COLLSCAN",
                        "filter": {
                            "population": {
                                "$lt": 500000
                            }
                        },
                        "direction": "forward"
                    }
                }
            }
        },
        "rejectedPlans": []
    }
}
```

```
Administrator: Command Prompt - docker-compose exec mongo mongo -u admin -p pass --authenticationDatabase=admin
{
    "stage": "SORT_KEY_GENERATOR",
    "inputStage": {
        "stage": "COLLSCAN",
        "filter": {
            "population": {
                "$lt": 500000
            }
        },
        "direction": "forward"
    }
},
"rejectedPlans": []
},
"executionStats": {
    "executionSuccess": true,
    "nReturned": 14,
    "executionTimeMillis": 0,
    "totalKeysExamined": 0,
    "totalDocsExamined": 53,
    "executionStages": {
        "stage": "PROJECTION",
        "nReturned": 14,
        "executionTimeMillisEstimate": 0,
        "works": 71,
        "advanced": 14,
        "needTime": 56,
        "needYield": 0,
        "saveState": 0,
        "restoreState": 0,
        "isEOF": 1,
        "invalidates": 0,
        "transformBy": {
            "name": 1,
            "population": 2,
            "capital": 3
        }
    }
}
```

```
[Administrator Command Prompt : docker-compose exec mongo mongo -u admin -p pass -authenticationDatabase=admin
db.countries.createIndex([{"population":1}])
{
    "createdCollectionAutomatically" : false,
    "numIndexesBefore" : 1,
    "numIndexesAfter" : 2,
    "ok" : 1
}
> db.countries.find("population":{$lt:500000}, {"name":1, "population":2, "capital":3}).sort([{"population":1}]).explain("executionStats")
{
    "queryPlanner" : {
        "plannerVersion" : 1,
        "namespace" : "demo.countries",
        "indexFilterSet" : false,
        "parsedQuery" : {
            "population" : {
                "$lt" : 500000
            }
        },
        "winningPlan" : {
            "stage" : "PROJECTION",
            "transformBy" : {
                "name" : 1,
                "population" : 2,
                "capital" : 3
            }
        },
        "inputStage" : {
            "stage" : "FETCH",
            "inputStage" : {
                "stage" : "TIXCMH"
                "keyPattern" : {
                    "population" : 1
                },
                "indexName" : "population_1",
                "isMultiKey" : false,
                "multiKeyPaths" : {
                    "population" : []
                },
                "isUnique" : false,
                "partialFilterExpression" : {}
            }
        }
    }
}
```

```
Administrator: Command Prompt - docker-compose exec mongo mongo -u admin -p pass --authenticationDatabase=admin
{
    "restoreState" : 0,
    "isEOF" : 1,
    "invalidates" : 0,
    "docsExamined" : 14,
    "alreadyHasObj" : 0,
    "inputStage" : {
        "stage" : "IXSCAN",
        "nReturned" : 14,
        "executionTimeMillisEstimate" : 0,
        "works" : 15,
        "advanced" : 14,
        "needTime" : 0,
        "needYield" : 0,
        "saveState" : 0,
        "restoreState" : 0,
        "isEOF" : 1,
        "invalidates" : 0,
        "keyPattern" : {
            "population" : 1
        },
        "indexName" : "population_1",
        "isMultiKey" : false,
        "multiKeyPaths" : {
            "population" : [ ]
        },
        "isUnique" : false,
        "isSparse" : false,
        "isPartial" : false,
        "indexVersion" : 2,
        "direction" : "forward",
        "indexBounds" : {
            "population" : [
                "[ -inf.0, 500000.0 )"
            ]
        },
        "keysExamined" : 14,
        "seeks" : 1,
        "dupsTested" : 0,
        "indexBounds" : {
            "population" : [
                "[ -inf.0, 500000.0 )"
            ]
        },
        "keysExamined" : 14,
        "seeks" : 1,
        "dupsTested" : 0,
        "dupsDropped" : 0,
        "seenInvalidated" : 0
    }
},
    "serverInfo" : {
        "host" : "mongo",
        "port" : 27017,
        "version" : "4.0.0",
        "gitVersion" : "3b07af3d4f471ae89e8186d33bbb1d5259597d51"
    },
    "ok" : 1
}
```

4. Select the most appropriate Redis data structure to store the following information:

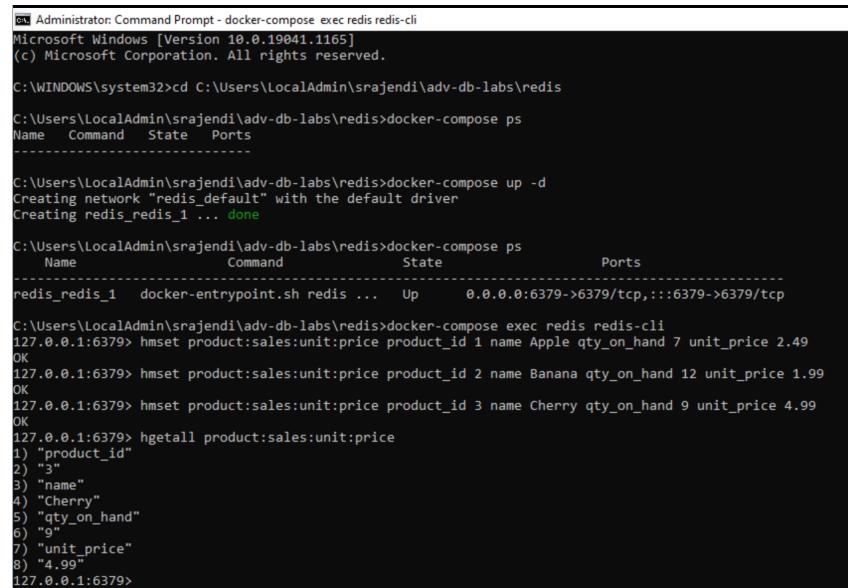
Product ID	Name	Qty On Hand	Unit Price
1	Apple	7	2.49
2	Banana	12	1.99
3	Cherry	9	4.99

Execute the commands to store this information in Redis. Make sure to namespace your key and each of the fields should be retrievable under the key used.

Solution:

```
-- create a data structure in Redis
hmset product:sales:unit:price product_id 1 name Apple qty_on_hand 7 unit_price 2.49
hmset product:sales:unit:price product_id 2 name Banana qty_on_hand 12 unit_price 1.99
hmset product:sales:unit:price product_id 3 name Cherry qty_on_hand 9 unit_price 4.99
-- retrieve data from Redis data structure
hgetall product:sales:unit:price
```

Evidence:



The screenshot shows a Windows Command Prompt window with the following session:

```
Administrator: Command Prompt - docker-compose exec redis redis-cli
Microsoft Windows [Version 10.0.19041.1165]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd C:\Users\LocalAdmin\srajendi\adv-db-labs\redis
C:\Users\LocalAdmin\srajendi\adv-db-labs\redis>docker-compose ps
Name      Command           State    Ports
redis_redis_1   docker-entrypoint.sh redis ...   Up      0.0.0.0:6379->6379/tcp,:::6379->6379/tcp

C:\Users\LocalAdmin\srajendi\adv-db-labs\redis>docker-compose up -d
Creating network "redis_default" with the default driver
Creating redis_redis_1 ... done

C:\Users\LocalAdmin\srajendi\adv-db-labs\redis>docker-compose ps
Name      Command           State    Ports
redis_redis_1   docker-entrypoint.sh redis ...   Up      0.0.0.0:6379->6379/tcp,:::6379->6379/tcp

C:\Users\LocalAdmin\srajendi\adv-db-labs\redis>docker-compose exec redis redis-cli
127.0.0.1:6379> hmset product:sales:unit:price product_id 1 name Apple qty_on_hand 7 unit_price 2.49
OK
127.0.0.1:6379> hmset product:sales:unit:price product_id 2 name Banana qty_on_hand 12 unit_price 1.99
OK
127.0.0.1:6379> hmset product:sales:unit:price product_id 3 name Cherry qty_on_hand 9 unit_price 4.99
OK
127.0.0.1:6379> hgetall product:sales:unit:price
1) "product_id"
2) "1"
3) "name"
4) "Apple"
5) "qty_on_hand"
6) "7"
7) "unit_price"
8) "2.49"
127.0.0.1:6379>
```

5. Select the most appropriate Redis data structure to store the following information:

The 2018 Golden Snowball Competition for the Upstate NY City with the Highest Snowfall. Scores updated hourly.

City	Syracuse	Rochester	Buffalo
Snowfall Inches	97	68	84

Execute the commands to store this information in Redis. Make sure to namespace your key and each of the snowfall values should be updatable. For example, you should be able to add 10 inches to Buffalo to make it 94. You should be able to display the information upon request.

Solution:

```
-- create a data structure in Redis
zadd golden:snowball:2018_competition 97 Syracuse 68 Rochester 84
-- retrieve values
zrange golden:snowball:2018_competition 0 -1 WITHSCORES
-- increment value by 10
zadd golden:snowball:2018_competition incr 10 Buffalo
-- retrieve values
zrange golden:snowball:2018_competition 0 -1 WITHSCORES
```

Evidence:

```
127.0.0.1:6379> zadd golden:snowball:2018_competition 97 Syracuse 68 Rochester 84 Buffalo
(integer) 3
127.0.0.1:6379> zrange golden:snowball:2018_competition 0 -1 WITHSCORES
(error) ERR value is not an integer or out of range
127.0.0.1:6379> zrange golden:snowball:2018_competition 0 -1 WITHSCORES
1) "Rochester"
2) "68"
3) "Buffalo"
4) "84"
5) "Syracuse"
6) "97"
127.0.0.1:6379> -
```

```
c:\ Administrator: Command Prompt - docker-compose exec redis redis-cli
127.0.0.1:6379> zrange golden:snowball:2018_competition 0 -1 WITHSCORES
1) "Rochester"
2) "68"
3) "Buffalo"
4) "84"
5) "Syracuse"
6) "97"
127.0.0.1:6379> zadd golden:snowball:2018_competition incr 10 Buffalo
"94"
127.0.0.1:6379> zrange golden:snowball:2018_competition 0 -1 WITHSCORES
1) "Rochester"
2) "68"
3) "Buffalo"
4) "94"
5) "Syracuse"
6) "97"
127.0.0.1:6379> -
```

Appendix

769-Win10Docker-srajendi

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.19041.1165]
(c) Microsoft Corporation. All rights reserved.

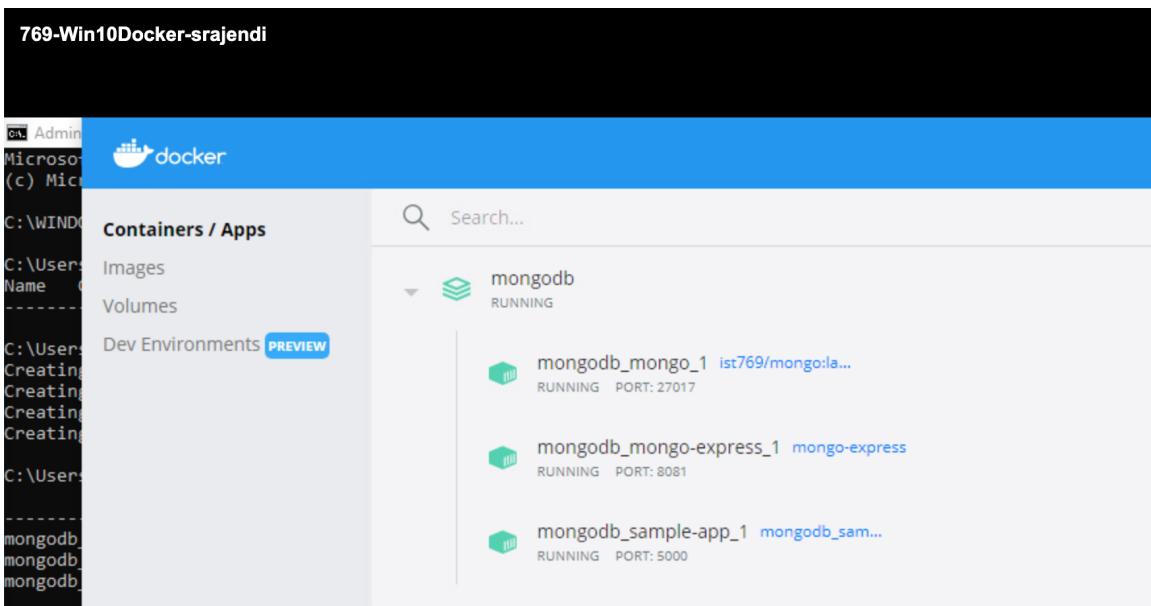
C:\WINDOWS\system32>cd C:\Users\LocalAdmin\srajendi\adv-db-labs\mongodb

C:\Users\LocalAdmin\srajendi\adv-db-labs\mongodb>docker-compose ps
Name      Command     State    Ports
-----  
mongod_mongo_1   ... done
mongod_mongo-express_1 ... done
mongod_sample-app_1 ... done

C:\Users\LocalAdmin\srajendi\adv-db-labs\mongodb>docker-compose up -d
Creating network "mongodb_default" with the default driver
Creating mongodb_mongo_1 ... done
Creating mongodb_mongo-express_1 ... done
Creating mongodb_sample-app_1 ... done

C:\Users\LocalAdmin\srajendi\adv-db-labs\mongodb>docker-compose ps
Name      Command     State    Ports
-----  
mongodb_mongo-express_1   tini -- /docker-entrypoint ... Up      0.0.0.0:8081->8081/tcp,:::8081->8081/tcp
mongodb_mongo_1           docker-entrypoint.sh mongod Up      0.0.0.0:27017->27017/tcp,:::27017->27017/tcp
mongodb_sample-app_1       python -m flask run --host ... Up      0.0.0.0:5000->5000/tcp,:::5000->5000/tcp

C:\Users\LocalAdmin\srajendi\adv-db-labs\mongodb>_
```



```

Administrator: Command Prompt
Microsoft Windows [Version 10.0.19041.1105]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd C:\Users\LocalAdmin\srajendi\adv-db-labs\mongodb
C:\Users\LocalAdmin\srajendi\adv-db-labs\mongodb>docker-compose ps
Name      Command     State    Ports
-----  -----
C:\Users\LocalAdmin\srajendi\adv-db-labs\mongodb>docker-compose up -d
Creating network "mongodb_default" with the default driver
Creating mongoDB_mongo_1 ... done
Creating mongoDB_mongo-express_1 ... done
Creating mongoDB_sample-app_1 ... done

C:\Users\LocalAdmin\srajendi\adv-db-labs\mongodb>docker-compose ps
Name      Command     State    Ports
-----  -----
mongoDB_mongo-express_1   tini -- /docker-entrypoint ... Up      0.0.0.0:8081->8081/tcp,:::8081/tcp
mongoDB_mongo_1           docker-entrypoint.sh mongod Up      0.0.0.0:27017->27017/tcp,:::27017->27017/tcp
mongoDB_sample-app_1      python -m flask run -host ... Up      0.0.0.0:5000->5000/tcp,:::5000->5000/tcp

C:\Users\LocalAdmin\srajendi\adv-db-labs\mongodb>docker-compose exec mongo mongoimport -u admin -p pass --authenticationDatabase=admin -d demo -c countries --file=europe.json --jsonArray
2021-08-30T06:07:11.979+0000  connected to: localhost
2021-08-30T06:07:12.054+0000  imported 53 documents

C:\Users\LocalAdmin\srajendi\adv-db-labs\mongodb>

```

```

C:\Users\LocalAdmin\srajendi\adv-db-labs\mongodb>docker-compose exec mongo mongo -u admin -p pass --authenticationDatabase=admin
MongoDB shell version v4.0.0
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 4.0.0
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
  http://docs.mongodb.org/
Questions? Try the support group
  http://groups.google.com/group/mongodb-user
Server has startup warnings:
2021-08-30T06:01:16.716+0000 I STORAGE  [initandlisten]
2021-08-30T06:01:16.716+0000 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2021-08-30T06:01:16.716+0000 I STORAGE  [initandlisten] **             See http://dochub.mongodb.org/core/prodnotes-filesystem
---
Enable MongoDB's free cloud-based monitoring service to collect 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 created for you. Anyone you share the URL with will also be able to
view this page. 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()
---

> show dbs
admin  0.000GB
config 0.000GB
demo  0.000GB
local  0.000GB
>
C:\Administrator: Command Prompt - docker-compose exec mongo mongo -u admin -p pass --authenticationDatabase=admin
> show collections
> use demo
switched to db demo
> show collections
countries
> db.countries.find()
[{"_id": ObjectId("612c7590e6433e0c3c41979"), "name": "Albania", "topLevelDomain": [ ".al" ], "alpha2Code": "AL", "alpha3Code": "ALB", "callingCodes": [ "355" ], "capital": "Tirana", "altSpellings": [ "AL", "Shqipëri", "Shqipëria", "Shqipnia" ], "relevance": "0", "region": "Europe", "subregion": "Southern Europe", "translations": { "de": "Albanien", "es": "Albanía", "fr": "Albanie", "ja": "\u65e5 \u672f \u65e5", "it": "Albania" }, "population": 2893005, "latlng": [ 41, 20 ], "demony": "Albanian", "area": 28748, "gini": 34.5, "timezones": [ "UTC+01:00" ], "borders": [ "MNE", "GRC", "MKD", "KOS" ], "nativeName": "Shqipëria", "numericCode": "008", "currencies": [ "ALL" ], "languages": [ "sq" ] },
{"_id": ObjectId("612c7590e6433e0c3c41979"), "name": "Andorra", "topLevelDomain": [ ".ad" ], "alpha2Code": "AD", "alpha3Code": "AND", "callingCodes": [ "376" ], "capital": "Andorra la Vella", "altSpellings": [ "AD", "Principality of Andorra", "Principat d'Andorra" ], "relevance": "0.5", "region": "Europe", "subregion": "Southern Europe", "translations": { "de": "Andorra", "es": "Andorra", "fr": "Andorre", "ja": "\u65e5 \u672f \u65e5", "it": "Andorra" }, "population": 76949, "latlng": [ 42.5, 1.5 ], "demony": "Andorran", "area": 468, "gini": null, "timezones": [ "UTC+01:00" ], "borders": [ "FRA", "ESP" ], "nativeName": "Andorra", "numericCode": "020", "currencies": [ "EUR" ], "languages": [ "ca" ] },
{"_id": ObjectId("612c7590e6433e0c3c41979"), "name": "Austria", "topLevelDomain": [ ".at" ], "alpha2Code": "AT", "alpha3Code": "AUT", "callingCodes": [ "43" ], "capital": "Vienna", "altSpellings": [ "Austria", "Oesterreich", "Osterreich", "Ostereich" ], "relevance": "0", "region": "Europe", "subregion": "Western Europe", "translations": { "de": "Oesterreich", "es": "Austria", "fr": "Autriche", "ja": "\u65e5 \u672f \u65e5", "it": "Austria" }, "population": 8602112, "latlng": [ 47.33333333, 13.33333333 ], "demony": "Austrian", "area": 83871, "gini": 26, "timezones": [ "UTC+01:00" ], "borders": [ "CZE", "DEU", "HUN", "ITA", "LIE", "SVK", "SVM", "CHE" ], "nativeName": "Oesterreich", "numericCode": "040", "currencies": [ "EUR" ], "languages": [ "de" ] },
{"_id": ObjectId("612c7590e6433e0c3c41979"), "name": "Aland Islands", "topLevelDomain": [ ".ax" ], "alpha2Code": "AX", "alpha3Code": "ALA", "callingCodes": [ "358" ], "capital": "Mariehamn", "altSpellings": [ "AX", "Aaland", "Aland", "Ahvenanmaa" ], "relevance": "0", "region": "Europe", "subregion": "Northern Europe", "translations": { "de": "Aland", "es": "Alandia", "fr": "Aland", "ja": "\u65e5 \u672f \u65e5 \u65e5", "it": "Isole Aland" }, "population": 28875, "latlng": [ 60.116667, 19.9 ], "demony": "Alandish", "area": 1580, "gini": null, "timezones": [ "UTC+02:00" ], "borders": [ ], "nativeName": "Aland", "numericCode": "248", "currencies": [ "EUR" ], "languages": [ "sv" ] }

```

```
C:\> Administrator: Command Prompt - docker-compose exec mongo mongo -u admin -p pass --authenticationDatabase=admin
> db.countries.find().pretty()
{
    "_id" : ObjectId("612c7590e56433e0c3c41978"),
    "name" : "Albania",
    "topLevelDomain" : [
        ".al"
    ],
    "alpha2Code" : "AL",
    "alpha3Code" : "ALB",
    "callingCodes" : [
        "355"
    ],
    "capital" : "Tirana",
    "altSpellings" : [
        "AL",
        "Shqipëri",
        "Shqipëria",
        "Shqipnia"
    ],
    "relevance" : "0",
    "region" : "Europe",
    "subregion" : "Southern Europe",
    "translations" : {
        "de" : "Albanien",
        "es" : "Albania",
        "fr" : "Albanie",
        "ja" : "\u65e5\u672f\u65e5\u672f\u65e5",
        "it" : "Albania"
    },
    "population" : 2893005,
    "latlng" : [
        41,
        20
    ],
    "demonym" : "Albanian",
    "area" : 28748,
    "gini" : 34.5,
    "timezones" : [
```

```
c:\ Administrator: Command Prompt - docker-compose exec mongo mongo -u admin -p pass --authenticationDatabase=admin
> db.countries.find({"population":{$lt:500000}}, {"name":1, "population":2, "capital":3}).sort({{"population":1}}).pretty()
{
    "_id" : ObjectId("612c7590e56433e0c3c4198d"),
    "name" : "Holy See",
    "capital" : "Rome",
    "population" : 451
}
{
    "_id" : ObjectId("612c7590e56433e0c3c419a8"),
    "name" : "Svalbard and Jan Mayen",
    "capital" : "Longyearbyen",
    "population" : 2562
}
{
    "_id" : ObjectId("612c7590e56433e0c3c4197b"),
    "name" : "Åland Islands",
    "capital" : "Mariehamn",
    "population" : 28875
}
{
    "_id" : ObjectId("612c7590e56433e0c3c41989"),
    "name" : "Gibraltar",
    "capital" : "Gibraltar",
    "population" : 32734
}
{
    "_id" : ObjectId("612c7590e56433e0c3c419a4"),
    "name" : "San Marino",
    "capital" : "City of San Marino",
    "population" : 32831
}
{
    "_id" : ObjectId("612c7590e56433e0c3c41994"),
    "name" : "Liechtenstein",
    "capital" : "Vaduz",
    "population" : 37370
}
```

```
c:\ Administrator: Command Prompt - docker-compose exec mongo mongo -u admin -p pass --authenticationDatabase=admin
{
    "name" : "Andorra",
    "capital" : "Andorra la Vella",
    "population" : 76949
}
{
    "_id" : ObjectId("612c7590e56433e0c3c4198f"),
    "name" : "Isle of Man",
    "capital" : "Douglas",
    "population" : 84497
}
{
    "_id" : ObjectId("612c7590e56433e0c3c41993"),
    "name" : "Jersey",
    "capital" : "Saint Helier",
    "population" : 99000
}
{
    "_id" : ObjectId("612c7590e56433e0c3c4198e"),
    "name" : "Iceland",
    "capital" : "Reykjavík",
    "population" : 330610
}
{
    "_id" : ObjectId("612c7590e56433e0c3c41997"),
    "name" : "Malta",
    "capital" : "Valletta",
    "population" : 445426
}
>
```

```
Administrator: Command Prompt - docker-compose exec mongo mongo -u admin -p pass --authenticationDatabase=admin
> db.countries.find({"population":{$lt:500000}}, {"name":1, "population":2, "capital":3}).sort({{"population":1}}).explain("executionStats")
{
    "queryPlanner" : {
        "plannerVersion" : 1,
        "namespace" : "demo.countries",
        "indexFilterSet" : false,
        "parsedQuery" : {
            "population" : {
                "$lt" : 500000
            }
        },
        "winningPlan" : {
            "stage" : "PROJECTION",
            "transformby" : {
                "name" : 1,
                "population" : 2,
                "capital" : 3
            },
            "inputStage" : {
                "stage" : "SORT",
                "sortPattern" : {
                    "population" : 1
                },
                "inputStage" : {
                    "stage" : "SORT_KEY_GENERATOR",
                    "inputStage" : {
                        "stage" : "COLLSCAN",
                        "filter" : {
                            "population" : {
                                "$lt" : 500000
                            }
                        },
                        "direction" : "forward"
                    }
                }
            }
        },
        "rejectedPlans" : [ ]
    }
}
```

769-Win10Docker-srajendi

```
C:\> Administrator: Command Prompt - docker-compose exec mongo mongo -u admin -p pass --authenticationDatabase=admin
{
    "stage" : "SORT_KEY_GENERATOR",
    "inputStage" : {
        "stage" : "COLLSCAN",
        "filter" : {
            "population" : {
                "$lt" : 500000
            }
        },
        "direction" : "forward"
    }
},
"rejectedPlans" : [ ],
},
"executionStats" : {
    "executionSuccess" : true,
    "nReturned" : 14,
    "executionTimeMillis" : 0,
    "totalKeysExamined" : 0,
    "totalDocsExamined" : 53,
    "executionStages" : [
        {
            "stage" : "PROJECTION",
            "nReturned" : 14,
            "executionTimeMillisEstimate" : 0,
            "works" : 71,
            "advanced" : 14,
            "needTime" : 56,
            "needYield" : 0,
            "saveState" : 0,
            "restoreState" : 0,
            "isEOF" : 1,
            "invalidates" : 0,
            "transformBy" : {
                "name" : 1,
                "population" : 2,
                "capital" : 3
            }
        },
        {
            "stage" : "COLLSCAN",
            "filter" : {
                "population" : {
                    "$lt" : 500000
                }
            }
        }
    ]
}
```

```
Administrator: Command Prompt - docker-compose exec mongo mongo -u admin -p pass --authenticationDatabase=admin
{
    "invalidates" : 0,
    "inputStage" : {
        "stage" : "COLLSCAN",
        "filter" : {
            "population" : {
                "$lt" : 500000
            }
        },
        "nReturned" : 14,
        "executionTimeMillisEstimate" : 0,
        "works" : 55,
        "advanced" : 14,
        "needTime" : 40,
        "needYield" : 0,
        "saveState" : 0,
        "restoreState" : 0,
        "isEOF" : 1,
        "invalidates" : 0,
        "direction" : "forward",
        "docsExamined" : 53
    }
},
"serverInfo" : {
    "host" : "mongo",
    "port" : 27017,
    "version" : "4.0.0",
    "gitVersion" : "3b07af3d4f471ae89e8186d33bbbb1d5259597d51"
},
"ok" : 1
}
> -
```

```
Administrator: Command Prompt - docker-compose exec mongo mongo -u admin -p pass --authenticationDatabase=admin
> db.countries.createIndex({population:1})
{
    "createdCollectionAutomatically" : false,
    "numIndexesBefore" : 1,
    "numIndexesAfter" : 2,
    "ok" : 1
}
> db.countries.find({"population":{$lt:500000}}, {"name":1, "population":2, "capital":3}).sort({"population":1}).explain("executionStats")
{
    "queryPlanner" : {
        "plannerVersion" : 1,
        "namespace" : "demo.countries",
        "indexFilterSet" : false,
        "parsedQuery" : {
            "population" : {
                "$lt" : 500000
            }
        },
        "winningPlan" : {
            "stage" : "PROJECTION",
            "transformBy" : {
                "name" : 1,
                "population" : 2,
                "capital" : 3
            },
            "inputStage" : {
                "stage" : "FETCH",
                "inputStage" : {
                    "stage" : "IXSCAN",
                    "keyPattern" : {
                        "population" : 1
                    },
                    "indexName" : "population_1",
                    "isMultiKey" : false,
                    "multiKeyPaths" : {
                        "population" : [ ]
                    },
                    "isUnique" : false,
                    "isSparse" : false,
                    "isPartial" : false
                }
            }
        }
    }
}
```

```
C:\ Administrator: Command Prompt - docker-compose exec mongo mongo -u admin -p pass --authenticationDatabase=admin
```

```
        "restoreState" : 0,
        "isEOF" : 1,
        "invalidates" : 0,
        "docsExamined" : 14,
        "alreadyHasObj" : 0,
        "inputStage" : {
            "stage" : "IXSCAN",
            "nReturned" : 14,
            "executionTimeMillisEstimate" : 0,
            "works" : 15,
            "advanced" : 14,
            "needTime" : 0,
            "needYield" : 0,
            "saveState" : 0,
            "restoreState" : 0,
            "isEOF" : 1,
            "invalidates" : 0,
            "keyPattern" : {
                "population" : 1
            },
            "indexName" : "population_1",
            "isMultiKey" : false,
            "multiKeyPaths" : {
                "population" : [ ]
            },
            "isUnique" : false,
            "isSparse" : false,
            "isPartial" : false,
            "indexVersion" : 2,
            "direction" : "forward",
            "indexBounds" : {
                "population" : [
                    "[ -inf.0, 500000.0 )"
                ]
            },
            "keysExamined" : 14,
            "seeks" : 1,
            "dupsTested" : 0,
```

```
Administrator: Command Prompt - docker-compose exec mongo mongo -u admin -p pass --authenticationDatabase=admin
{
    "isEOF" : 1,
    "invalidates" : 0,
    "keyPattern" : {
        "population" : 1
    },
    "indexName" : "population_1",
    "isMultiKey" : false,
    "multiKeyPaths" : {
        "population" : [ ]
    },
    "isUnique" : false,
    "isSparse" : false,
    "isPartial" : false,
    "indexVersion" : 2,
    "direction" : "forward",
    "indexBounds" : {
        "population" : [
            "[ -inf.0, 500000.0 )"
        ]
    },
    "keysExamined" : 14,
    "seeks" : 1,
    "dupsTested" : 0,
    "dupsDropped" : 0,
    "seenInvalidated" : 0
}
}
},
"serverInfo" : {
    "host" : "mongo",
    "port" : 27017,
    "version" : "4.0.0",
    "gitVersion" : "3b07af3d4f471ae89e8186d33bbb1d5259597d51"
},
"ok" : 1
}
>
```

769-Win10Docker-srajendi

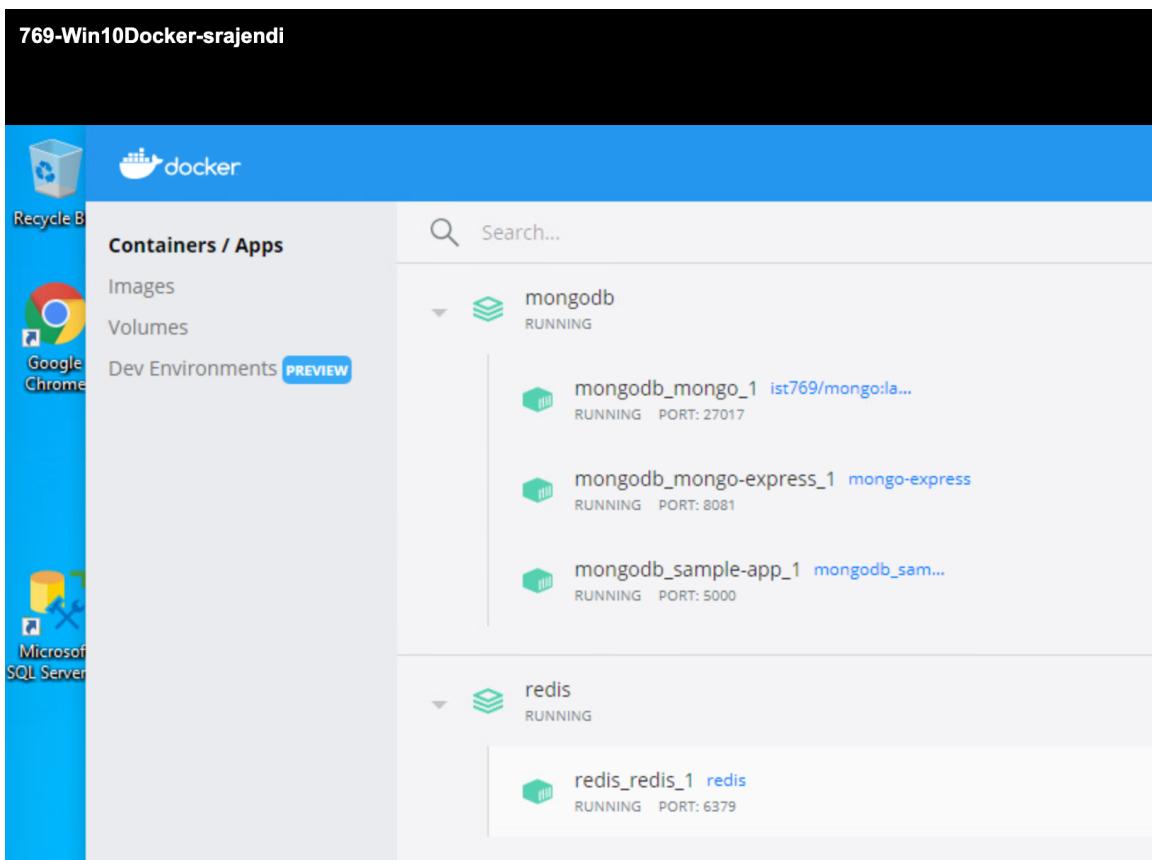
Recycle Bin

File Home Share View

Administrator: Command Prompt

```
Microsoft Windows [Version 10.0.19041.1165]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd C:\Users\LocalAdmin\srajendi\adv-db-labs\redis
C:\Users\LocalAdmin\srajendi\adv-db-labs\redis>docker-compose ps
Name Command State Ports
-----
C:\Users\LocalAdmin\srajendi\adv-db-labs\redis>docker-compose up -d
Creating network "redis_default" with the default driver
Creating redis_redis_1 ... done
C:\Users\LocalAdmin\srajendi\adv-db-labs\redis>docker-compose ps
Name Command State Ports
redis_redis_1 docker-entrypoint.sh redis ... Up 0.0.0.0:6379->6379/tcp,:::6379->6379/tcp
C:\Users\LocalAdmin\srajendi\adv-db-labs\redis>
```



769-Win10Docker-srajendi

```
C:\ Administrator: Command Prompt - docker-compose exec redis redis-cli
Microsoft Windows [Version 10.0.19041.1165]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd C:\Users\LocalAdmin\srajendi\adv-db-labs\redis
C:\Users\LocalAdmin\srajendi\adv-db-labs\redis>docker-compose ps
Name      Command      State      Ports
-----
C:\Users\LocalAdmin\srajendi\adv-db-labs\redis>docker-compose up -d
Creating network "redis_default" with the default driver
Creating redis_redis_1 ... done

C:\Users\LocalAdmin\srajendi\adv-db-labs\redis>docker-compose ps
Name      Command      State      Ports
-----
redis_redis_1  docker-entrypoint.sh redis ...  Up      0.0.0.0:6379->6379/tcp,:::6379->6379/tcp

C:\Users\LocalAdmin\srajendi\adv-db-labs\redis>docker-compose exec redis redis-cli
127.0.0.1:6379> 
```

```
C:\ Administrator: Command Prompt - docker-compose exec redis redis-cli
Microsoft Windows [Version 10.0.19041.1165]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd C:\Users\LocalAdmin\srajendi\adv-db-labs\redis
C:\Users\LocalAdmin\srajendi\adv-db-labs\redis>docker-compose ps
Name      Command      State      Ports
-----
C:\Users\LocalAdmin\srajendi\adv-db-labs\redis>docker-compose up -d
Creating network "redis_default" with the default driver
Creating redis_redis_1 ... done

C:\Users\LocalAdmin\srajendi\adv-db-labs\redis>docker-compose ps
Name      Command      State      Ports
-----
redis_redis_1  docker-entrypoint.sh redis ...  Up      0.0.0.0:6379->6379/tcp,:::6379->6379/tcp

C:\Users\LocalAdmin\srajendi\adv-db-labs\redis>docker-compose exec redis redis-cli
127.0.0.1:6379> hmset product:sales:unit:price product_id 1 name Apple qty_on_hand 7 unit_price 2.49
OK
127.0.0.1:6379> hmset product:sales:unit:price product_id 2 name Banana qty_on_hand 12 unit_price 1.99
OK
127.0.0.1:6379> hmset product:sales:unit:price product_id 3 name Cherry qty_on_hand 9 unit_price 4.99
OK
127.0.0.1:6379> hgetall product:sales:unit:price
1) "product_id"
2) "3"
3) "name"
4) "Cherry"
5) "qty_on_hand"
6) "9"
7) "unit_price"
8) "4.99"
127.0.0.1:6379>
```

769-Win10Docker-srajendi

```
c:\ Administrator: Command Prompt - docker-compose exec redis redis-cli
redis_redis_1  docker-entrypoint.sh redis ... Up      0.0.0.0:6379->6379/tcp,:::6379->6379/tcp

C:\Users\LocalAdmin\srajendi\adv-db-labs\redis>docker-compose exec redis redis-cli
127.0.0.1:6379> hmset product:sales:unit:price product_id 1 name Apple qty_on_hand 7 unit_price 2.49
OK
127.0.0.1:6379> hmset product:sales:unit:price product_id 2 name Banana qty_on_hand 12 unit_price 1.99
OK
127.0.0.1:6379> hmset product:sales:unit:price product_id 3 name Cherry qty_on_hand 9 unit_price 4.99
OK
127.0.0.1:6379> hgetall product:sales:unit:price
1) "product_id"
2) "3"
3) "name"
4) "Cherry"
5) "qty_on_hand"
6) "9"
7) "unit_price"
8) "4.99"
127.0.0.1:6379> zadd golden:snowball:2018_competition 97 Syracuse 68 Rochester 84 Buffalo
(integer) 3
127.0.0.1:6379> zrange golden:snowball:2018_competition 0-1 WITHSCORES
(error) ERR value is not an integer or out of range
127.0.0.1:6379> zrange golden:snowball:2018_competition 0 -1 WITHSCORES
1) "Rochester"
2) "68"
3) "Buffalo"
4) "84"
5) "Syracuse"
6) "97"
127.0.0.1:6379> -
```

769-Win10Docker-srajendi

```
c:\ Administrator: Command Prompt - docker-compose exec redis redis-cli
127.0.0.1:6379> zrange golden:snowball:2018_competition 0 -1 WITHSCORES
1) "Rochester"
2) "68"
3) "Buffalo"
4) "84"
5) "Syracuse"
6) "97"
127.0.0.1:6379> -
```

769-Win10Docker-srajendi

```
C:\ Administrator: Command Prompt - docker-compose exec redis redis-cli
127.0.0.1:6379> zrange golden:snowball:2018_competition 0 -1 WITHSCORES
1) "Rochester"
2) "68"
3) "Buffalo"
4) "84"
5) "Syracuse"
6) "97"
127.0.0.1:6379> zadd golden:snowball:2018_competition incr 10 Buffalo
"94"
127.0.0.1:6379> zrange golden:snowball:2018_competition 0 -1 WITHSCORES
1) "Rochester"
2) "68"
3) "Buffalo"
4) "94"
5) "Syracuse"
6) "97"
127.0.0.1:6379>
```

```
C:\ Administrator: Command Prompt
3) "Buffalo"
4) "94"
5) "Syracuse"
6) "97"
127.0.0.1:6379> exit

C:\Users\LocalAdmin\srajendi\adv-db-labs\redis>docker-compose ps
      Name           Command          State        Ports
----->
redis_redis_1   docker-entrypoint.sh redis ...   Up      0.0.0.0:6379->6379/tcp,:::6379->6379/tcp

C:\Users\LocalAdmin\srajendi\adv-db-labs\redis>docker-compose down
Stopping redis_redis_1 ... done
Removing redis_redis_1 ... done
Removing network redis_default

C:\Users\LocalAdmin\srajendi\adv-db-labs\redis>docker-compose ps
Name   Command  State  Ports
----->

C:\Users\LocalAdmin\srajendi\adv-db-labs\redis>
```

```
C:\ Administrator: Command Prompt
    "gitVersion" : "3b07af3d4f471ae89e8186d33bbb1d5259597d51"
},
"ok" : 1
}
> exit
bye

C:\Users\LocalAdmin\srajendi\adv-db-labs\mongodb>docker-compose ps
      Name           Command          State        Ports
----->
mongodb mongo-express_1   tini -- /docker-entrypoint ...   Up      0.0.0.0:8081->8081/tcp,:::8081->8081/tcp
mongodb_mongo_1   docker-entrypoint.sh mongod   Up      0.0.0.0:27017->27017/tcp,:::27017->27017/tcp
mongodb_sample-app_1   python -m flask run --host ...   Up      0.0.0.0:5000->5000/tcp,:::5000->5000/tcp

C:\Users\LocalAdmin\srajendi\adv-db-labs\mongodb>docker-compose down
Stopping mongodb_sample-app_1 ... done
Stopping mongodb_mongo-express_1 ... done
Stopping mongodb_mongo_1 ... done
Removing mongodb_sample-app_1 ... done
Removing mongodb_mongo-express_1 ... done
Removing mongodb_mongo_1 ... done
Removing network mongodb_default

C:\Users\LocalAdmin\srajendi\adv-db-labs\mongodb>docker-compose ps
Name   Command  State  Ports
----->

C:\Users\LocalAdmin\srajendi\adv-db-labs\mongodb>
```

