Chapter 1: Authentication

Homework 1.1 : Enable SCRAM-SHA-1

Back to the Question

On this exercise we want to evaluate your understanding of the different authentication options, the command that shows the enabled authentication mechanisms, and to review if you have been able to setup `mongod` with authentication enabled.

Here are the details to each of the steps laid out in the homework:

- Launch `mongod` with no authentication enabled

```
$ mkdir -p M310-HW-1.1/db
$ mongod --dbpath M310-HW-1.1/db --logpath M310-HW-
1.1/db/mongo.log --fork
```

- Create user `alice` with password `secret` on **admin** database and role **root**

```
$ mongo
use admin
db.createUser({user: 'alice', pwd: 'secret', roles:
['root']})
```

- Relaunch `mongod` with authentication enabled

```
db.shutdownServer()
$ mongod --dbpath M310-HW-1.1/db --logpath M310-HW-
1.1/db/mongo.log --fork --auth
```

Also, part of the exercise, is to review your understanding of the different ways of authenticating and executing commands over the **mongo** shell.

The correction options are:

```
mongo admin -u alice -p secret --eval "db.runCommand({ge
```
COPY

```
mongo admin --eval "db.auth('alice', 'secret');db.runCor
```
COPY

```
mongo -u alice -p secret --eval "db=db.getSisterDB('admi
```
COPY

Why do the other commands fail?

```
mongo -u alice -p secret --eval "db.runCommand({getParar
```
COPY

This instruction is incorrect because, although we are authenticating correctly, this particular command is an *administration* command and therefore needs to be run in **admin** database.

```
mongo -u alice -p secret --eval "db.runCommand({getParar
```
COPY

The previous instruction does not authenticate using **admin** database, which is required since this is the namespace where the user has been created.

```
mongo --eval "db.runCommand({getParameter: 1, authentica
```
COPY

Tries to run an instruction without providing any authentication credentials to a **--auth** enabled **mongod**.

Proceed to next section