MongoDB

Installation

Installation Redhat 9

Configuration dépôt

Créer le fichier /etc/yum.repos.d/mongodb-org-x.0.repo

```
[mongodb-org-x.0]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/9/mongodb-org/x.0/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://pgp.mongodb.com/server-x.0.asc
```

Installation

```
sudo dnf install mongodb-org
```

Configuration

Paramétrage dans le fichier /etc/mongod.conf

Interface et port d'écoute

```
net:
  port: 27017         # port d'écoute
  bindIp: 127.0.0.1   # Enter 0.0.0.0,:: to bind to all IPv4 and IPv6
addresses or, alternatively, use the net.bindIpAll setting.
```

Activation TLS

```
net:
  ...
  tls:
    mode: requireTLS
    certificateKeyFile: /etc/ssl/certs/serveur.pem
```

```
    CAFile: /usr/local/share/ca-certificates/ca.crt
```

Fichiers

- ~/.dbshell : historique des commandes
- ~/.mongorc.js : fichier dans lequel on peut définir des variables, customiser le prompt mongo ...
- /etc/mongorc.js : même chose au niveau global (évalué avant le fichier de l'utilisateur)

Accès

Identification

Accès sans mot de passe

```
mongosh -u <user>
```

Accès avec mot de passe

```
mongosh -u <user> -p <password>
mongo 10.1.1.189:27017 -u mongoadmin -p M0ng0@dmin --authenticationDatabase
admin
mongo --host 10.1.1.189 --port 27017 -u mongoadmin -p M0ng0@dmin --
authenticationDatabase admin
```

Connexion en TLS

```
sudo mongosh 127.0.0.1:27017 -u mongoadmin -p 'M0ng0@dmin' --
authenticationDatabase admin --tls --tlsCAFile /usr/local/share/ca-
certificates/caroot.cer
```

Réinitialiser utilisateur

```
sudo systemctl stop mongod
sudo mongod --dbpath /var/lib/mongo --port 27017 --bind_ip 127.0.0.1 --
noauth
```

```
mongosh --port 27017
```

```
USE admin
db.system.users.find().pretty()
```

```
use admin
db.dropUser("mongosuperadmin")
```

```
db.dropUser("mongoadmin")
db.createUser(
        {
                user: "mongosuperadmin",
                pwd: "M0ng0super@dmin",
                roles: [ { role: "root", db: "admin" } ]
        }
)
db.createUser(
        {
                user: "mongoadmin",
                pwd: "M0ng0@dmin",
                roles: [ { role: "dbOwner", db: "lbs" } ]
        }
)
```

```
sudo chown -R mongod: /var/lib/mongo
sudo systemctl start mongod
mongo < mongo.script
```

Commandes de base

Lister les bases de données

```
show dbs
use <db>
show tables
```

findOne

Il n'y a pas de structure à proprement parlé dans mongo mais il est possible d'afficher un élément pour avoir une idée de la structure :

```
db.<table>.findOne()
```

find

- Affiche tous les enregistrements d'une table

```
db.<table>.find( { } , { } )
```

Par défaut mongodb n'affiche que les 20 premiers éléments trouvés

- Pour compter le nombre d'enregistrements trouvés

```
db.<table>.find( { } , { } ).count()
```

- Pour afficher tous les enregistrements trouvés (et non pas les 20 premiers)

```
db.<table>.find( { } , { } ).limit(0)    # taper it pour afficher la suite
db.<table>.find( { } , { } ).forEach(printjson)
db.<table>.find( { } , { } ).toArray()
```

- Affiche certains champs sous condition

```
db.<table>.find( { "<champ1>" : "valeur" } , { "champ2" : 1 , "_id": 0 } )
```

Affiche le champ “champ2” de tous les enregistrements dont le champ “champ1” vaut “valeur”. Le champ id est normalement affiché par défaut du coup on demande à ne pas l'afficher.

- Liste les enregistrements dont champ.souschamp a une valeur donnée

```
db.<table>.find({"champ.souschamp": { $regex: "toto"} })                  #
valeur = toto
db.<table>.find({"champ.souschamp": { $regex: "toto"} , $options: "i"})   #
indépendamment minuscule/majuscule
```

- Liste les enregistrements dont champ.souschamp est égal à toto

```
db.<table>.find({"champ.souschamp": "toto" })
```

update

```
db.<table>.updateOne({ <filter> }, { <update> })
db.users.updateOne({ "name": "Alice" }, { $set: { "age": 31 } })
# modifie le champ age de l'enregistrement Alice. Si le champ n'existe pas
il est créé.
db.<table>.updateMany({ <filter> }, { <update> })
# pour modifier plusieurs enregistrement répondant au critère
db.<table>.replaceOne({ <filter> }, { "champ1" : "val1" , "champ2" : "val2"
,... } )   # pour remplacer intégralement l'enregistrement
```

- **$set** : Met à jour la valeur d'un champ, ou ajoute le champ s'il n'existe pas
- **$unset** : Supprime un champ
- **$inc** : Incrémente la valeur d'un champ numérique
- **$push** : ajouter un élément à un tableau
- **$pull** : retirer un élément à un tableau
- **$rename** : Renomme un champ

```
db.<table>.updateOne({ "champ": { $regex: "pattern" } }, [{ $set: { "champ":
{ $replaceOne: { input: "$champ", find: "oldChar", replacement: "newChar" }
} } }]) # Remplacer une chaîne de caractères par une autre
db.<table>.updateOne({ "champ": { $regex: "pattern" } }, [{ $set: { "champ":
{ $replaceAll: { input: "$champ", find: "oldChar", replacement: "newChar" }
```

```
} } }]) # Remplacer toutes les occurrences d'une chaîne de caractères
```

Exécution de commandes

Exécution d'une commande depuis le shell

```
mongo --host <@IP> --port <port> -u <user> -p <password> --
authenticationDatabase <authDB> <database> --eval '<requete>'
```

Exécution d'un fichier de commande

```
mongo --host <@IP> --port <port> -u <user> -p <password> --
authenticationDatabase <authDB> <database> < fichier_requetes
```

Options

- --tls : si Mongo est configuré pour utilisation SSL/TLS
- --quiet : exécution en mode silencieux

Résultat dans un fichier

- Redirection shell simple

```
mongo --host <@IP> --port <port> -u <user> -p <password> --
authenticationDatabase <authDB> <DB> --eval '<requete>' > resultat.txt
```

- fichier CSV avec mongoexport

```
mongoexport --host <@IP> --port <port> -u <user> -p <password> --
authenticationDatabase <authDB> --db=<DB> --collection=<table> --
query='{<requete>}' --out resultat.csv --type=csv --
fields=champ1,champ2,champ3,...
```

Manipulation bdd

Backup/Restore

Exporter une base de données

```
mongodump --host=<host> --port=27017 -u <user> -p <passwd>  --
authenticationDatabase=admin --out=<repertoire>
```

From:
https://wiki.iot-acs.fr/ - **Wiki**

Permanent link:
https://wiki.iot-acs.fr/doku.php?id=all:bibles:langages:mongodb

Last update: **2025/09/22 09:11**