hochschule
hof
University of Applied Sciences

Non-Relational Databases

MongoDB --- 4

Prof. Dr. Jürgen Heym

Hof University of Applied Sciences

8 – MongoDB Users & Roles

1. Let‘s create a site user administrator „siteUserAdmin“ and activate user authentication.

   a) Start MongoDB without access control and create user „siteUserAdmin“ with role „userAdminAnyDatabase“ in database admin.

   ```
   use admin

   db.createUser(
     {
       user: "siteUserAdmin",
       pwd:   "nosql",
       roles: [ { role: "userAdminAnyDatabase", db: "admin" } ]
     }
   )
   ```

   b) Edit the MongoDB configuration file /etc/mongod.conf and activate authentication setting option **`security. authorization: enabled`**.
   Restart mongod: **`service mongod restart`**

   See: https://docs.mongodb.com/manual/tutorial/enable-authentication/

8 – MongoDB Users & Roles

hochschule
hof
University of Applied Sciences

2. Create a user administrator for database „myDB“.

   a) Enter MongoDB and authenticate user „siteUserAdmin“ on db „admin“.

   ```
   mongo -u siteUserAdmin -p nosql admin
   ```

   b) Create user „myUserAdmin“ with role „userAdmin“ in database „myDB“

   ```
   use myDB
   db.createUser(
     {
       user: "myUserAdmin", pwd:  "nosql", roles: [ "userAdmin" ]
     }
   )
   ```

   c) Verification

   ```
   use admin
   db.system.users.find({user:"myUserAdmin"})
   ```

8 – MongoDB Users & Roles

hochschule hof
University of Applied Sciences

3. Create a new user for database „myDB“

   a) Enter MongoDB and authenticate user „myUserAdmin“ on db „myDB“.

   ```
   mongo –u myUserAdmin –p nosql myDB
   ```

   b) Create user „myUser“ with role „readWrite“ in database „myDB“.

   ```
   use myDB
   db.createUser( {
      user: "myUser", pwd:  "nosql", roles: [ "readWrite" ]
     } )
   ```

   c) Verify as user „siteUserAdmin“ and / or „myUserAdmin“.

   ```
   use admin
   db.system.users.find({user:"myUser"})
   ```

hochschule
hof
University of Applied Sciences

8 – MongoDB Users & Roles

4. Test user „myUser“

   a) Enter MongoDB and authenticate user „myUser“ on db „myDB“.

   ```
   mongo -u myUser -p nosql myDB
   ```

   b) Save a document in collection „myCol“ in database „myDB“ and verify the results.

   ```
   use myDB
   db.myCol.insert({firstname:"myFirstName",lastname:"myLastName"})
   ```

   c) Verification

   ```
   db.myCol.find()
   ```

8 – MongoDB Users & Roles

hochschule hof
University of Applied Sciences

5. Create a user with role on two databases.

   a) Enter MongoDB and authenticate user „siteUserAdmin“ on db „admin“.

   ```
   mongo –u siteUserAdmin –p nosql admin
   ```

   b) Create user „myUser2“ with role „read“ in database „myDB“ and „myDB2“.

   ```
   use myDB
   db.createUser( {
      user: "myUser2", pwd: "nosql",
      roles: [ { role: "read", db: "myDB" },
               { role: "read", db: "myDB2" } ] } )
   ```

   c) Verify as user „siteUserAdmin“!

   d) Verify read-write in database „myDB“ as user „myUser2“ .

8 – MongoDB Users & Roles

- Roles
    - A role grants privileges to perform the specified actions on resource. Each privilege is either specified explicitly in the role or inherited from another role or both.
- Privileges
    - A privilege consists of a specified resource and the actions permitted on the resource.
    - A resource is a database, collection, set of collections, or the cluster.
    - An action specifies the operation allowed on the resource.
- Inherited Privileges
    - A role can include one or more existing roles in its definition, in which case the role inherits all the privileges of the included roles.

8 – MongoDB Users & Roles

- Users and Roles
  - You can assign roles to users during the user creation and update.
  - A user assigned a role receives all the privileges of that role.
  - A user can have one or multiple roles in different databases.

- Built-in Roles
  - ***read***
    Provides the ability to read data on all non-system collections and on the following system collections: system.indexes, system.js, and system.namespaces collections.
  - ***readWrite***
    Provides all the privileges of the read role and the ability to modify data on all non-system collections and the system.js collection.

8 – MongoDB Users & Roles

hochschule
hof
University of Applied Sciences

- Built-in Roles (continued)

  - ***dbAdmin***
    Provides the ability to perform administrative tasks such as schema-related tasks, indexing, gathering statistics. This role does not grant privileges for user and role management.
  - ***dbOwner***
    Provides the ability to perform any administrative action on the database. This role combines the privileges granted by the *readWrite*, *dbAdmin* and *userAdmin* roles.
  - ***userAdmin***
    Provides the ability to create and modify roles and users on the current database. Since the userAdmin role allows users to grant any privilege to any user, including themselves, the role also indirectly provides *superuser* access to either the database or, if scoped to the admin database, the cluster.

8 – MongoDB Users & Roles

hochschule hof
University of Applied Sciences

- Built-in Roles (continued)

  - ***backup***
    Provides privileges needed to back up data. This role provides sufficient privileges to use the MongoDB Cloud Manager backup agent, Ops Manager backup agent, or to use mongodump.
  - ***restore***
    Provides privileges needed to restore data with mongorestore without the --oplogReplay option or without system.profile collection data.

8 – MongoDB Users & Roles

- All-Database Roles
  - **These roles in the admin database apply to all but the local and config databases in a mongod instance!**
  - ***readAnyDatabase***
    Provides the same *read-only* permissions as read, except it applies for all databases.
  - ***readWriteAnyDatabase***
    Provides the same read and write permissions as *readWrite*, except …
  - ***userAdminAnyDatabase***
    Provides the same access to user administration operations as *userAdmin*, except ...
  - ***dbAdminAnyDatabase***
    Provides the same access to database administration operations as *dbAdmin*, except …

- Superuser
  - **root**
    - Provides access to the operations and all the resources of the readWriteAnyDatabase, dbAdminAnyDatabase, userAdminAnyDatabase, clusterAdmin, restore, and backup combined.

8 – MongoDB Users & Roles

- Show users roles / access rights

1. Enter MongoDB and authenticate user „siteUserAdmin“ on db „admin“.

   ```
   mongo -u siteUserAdmin -p nosql admin
   ```

2. Show user info for user „myUser“ in database „myDB“:

   ```
   use myDB
   db.getUser("myUser")
   ```

8 – MongoDB Users & Roles

hochschule
hof
University of Applied Sciences

- Show a roles privileges

1. Enter MongoDB and authenticate user „siteUserAdmin“ on db „admin“.

```
mongo -u siteUserAdmin -p nosql admin
```

2. Show role info for role „read“ in database „myDB“:

```
use myDB
db.getRole("read",{showPrivileges:true})
```

All granted and inherited privileges are shown!

8 – MongoDB Users & Roles

hochschule
hof
University of Applied Sciences

- Grant roles

1. Enter MongoDB and authenticate user „siteUserAdmin“ on db „admin“.

```
mongo -u siteUserAdmin -p nosql admin
```

2. Grant role „readWrite“ to user „myUser“ on database „myDB“:

```
use myDB
db.grantRolesToUser(
   myUser,
   [
         {role:"readWrite", db:"myDB"}
   ]
)
```

8 – MongoDB Users & Roles

hochschule hof
University of Applied Sciences

- Revoke roles

1. Enter MongoDB and authenticate user „siteUserAdmin“ on db „admin“.

```
mongo –u siteUserAdmin –p nosql admin
```

2. Revoke role „readWrite“ from user „myUser“ on database „myDB“:

```
use myDB
db.revokeRolesFromUser(
   myUser,
   [
         {role:"readWrite", db:"myDB"}
   ]
)
```

8 – MongoDB Users & Roles

- Change user password

1. Enter MongoDB and authenticate user „siteUserAdmin“ on db „admin“.

```
mongo –u siteUserAdmin –p nosql admin
```

2. Change password for user „myUser“ in database „myDB“:

```
use myDB
db.changeUserPassword("myUser","newPasswordString")
```

To change your own password you have to proceed with a more complex procedure!

See: *http://docs.mongodb.org/manual/tutorial/change-own-password-and-custom-data/*

Literature

- MongoDB Access Control 1
  https://docs.mongodb.org/manual/administration/configuration/
- MongoDB Access Control 2
  http://docs.mongodb.org/manual/tutorial/enable-authentication-without-bypass/
- MongoDB Tutorial about Users and Roles
  http://docs.mongodb.org/manual/tutorial/add-user-administrator/
- MongoDB Change Your Own Password
  http://docs.mongodb.org/manual/tutorial/change-own-password-and-custom-data/