**Exposee Doc – MongoDB**

Dine opgaver:

* Forbind til databasen med de forskellige brugere.
* Test selv hvad du kan og ikke kan som hver bruger.
* Undersøg graden af granulær adgang
* Dokumenter dine tests med screenshots af succeser og fejl til din aflevering.
* Hvis dokumentationen mangler, så skal du bede mig om hjælp.

|  |  |
| --- | --- |
| Parameter | Value |
| Host | 10.136.130.131 |
| Port | 27017 |
| Database Name | granular\_access\_db |
| Container Name | mongo-container |

|  |  |
| --- | --- |
| Brugernavn | Password |
| user1 | User1Pass |
| user2 | User2Pass |
| user3 | User3Pass |

**Forbind til database via CLI når jeg har startet containeren:**

**Login som user1:**

docker run --rm -it mongo mongosh "mongodb://user1:User1Pass@10.136.130.131:27017/granular\_access\_db"

**Login som user2:**

docker run --rm -it mongo mongosh "mongodb://user2:User2Pass@10.136.130.131:27017/granular\_access\_db"

**Login som user3:**

docker run --rm -it mongo mongosh "mongodb://user3:User3Pass@10.136.130.131:27017/granular\_access\_db"

**Tjek evt. om database er valgt med kommandoen**:

use granular\_access\_db

Objekter i databasen indeholder følgende struktur:

|  |  |
| --- | --- |
| Felt | Datatype |
| \_id | ObjectId |
| username | String |
| email | String |
| ssn | String |
| role | String |

**Når forbindelsen er oprettet, kan du afprøve forskellige kommandoer på de forskellige brugere. Her er nogle eksempler:**

**Viser den begrænsede samling:**

db.restricted\_users\_view.find().pretty()

**Viser hele samlingen:**

db.users\_data.find().pretty()

**Indsætter ny data:**

db.users\_data.insertOne({ username: "test", email: "test@example.com", ssn: "000-00-0000", role: "user" })

**Opdater email på bruger:**

db.users\_data.updateOne({ username: "bob" }, { $set: { email: "bob@example.com" } })

**Slet data:**

db.users\_data.deleteOne({ username: "test" })