E1F

Ich kann ein Backup und Restore bei einer NoSQL Datenbank anwenden.

Fragenstellung und Lernziele

In dieser Übung lernt man, wie man eine MongoDB-Datenbank sicher sichert und wiederherstellt. Folgende Lernziele stehen im Fokus:

- Ein vollständiges Backup mit mongodump durchzuführen
- Die Datenbank mit mongorestore wiederherzustellen
- Die Datenintegrität nach dem Restore zu verifizieren

Umsetzung

1. **MongoDB-Container mit Authentifizierung aufsetzen:**

Du kannst für diese Übung den Container aus dem Kaptitel G1F - Anbindung and eine NoSQL Datenbank verwenden.

Falls du den Container nicht mehr hast, folge dem Setup Guide aus dem verlinkten Kapitel.

2. **Backup der Datenbank erstellen:**

Mit diesem Befehl wird ein Verzeichnis /backup/productsdb_backup im Container (und auf dem Host unter /backup) angelegt:

```
docker exec products mongodump `
  --username admin `
  --password secret123 `
  --authenticationDatabase admin `
  --db productsdb `
  --out /backup/productsdb_backup
```

Logge dich in den Container ein und überprüfe, ob das Backup erfolgreich erstellt wurde:

```
docker exec -it products bash
ls -l /backup
```

3. **Datenbank löschen:**

```
docker exec products mongosh `
  --username admin `
```

```
  --password secret123 `
  --authenticationDatabase admin `
  --eval 'db.getSiblingDB("productsdb").dropDatabase()'
```

Überprüfe, ob die Datenbank gelöscht wurde:

```
docker exec products mongosh `
  --username admin `
  --password secret123 `
  --authenticationDatabase admin `
  --eval 'db.getSiblingDB("productsdb").products.find().toArray()'
```

4. **Datenbank wiederherstellen:**

```
docker exec products mongorestore `
  --username admin `
  --password secret123 `
  --authenticationDatabase admin `
  --drop `
  /backup/productsdb_backup
```

Die Datenbank sollte nun wiederhergestellt sein. Überprüfe dies mit:

```
docker exec products mongosh `
  --username admin `
  --password secret123 `
  --authenticationDatabase admin `
  --eval 'db.getSiblingDB("productsdb").products.find().toArray()'
```

Nachweis

Datenbank Backup:

```
® Oliver ■ ~/../school/m165/g1f_api
# docker exec products mongodump `
>   --username admin `
>   --password secret123 `
>   --authenticationDatabase admin `
>   --db productsdb `
>   --out /backup/productsdb_backup
2025-05-22T11:35:40.129+0000    writing productsdb.products to /backup/productsdb_backup/productsdb/products.bson
2025-05-22T11:35:40.129+0000    done dumping productsdb.products (4 documents)
```

Datenbank löschen:

```
® Oliver ■ ~/../school/m165/g1f_api
# docker exec products mongosh `
>   --username admin `
>   --password secret123 `
>   --authenticationDatabase admin `
>   --eval 'db.getSiblingDB("productsdb").dropDatabase()'
```

```
{ ok: 1, dropped: 'productsdb' }
```

Überprüfen der Löschung:

```
Oliver ~/../school/m165/g1f_api
# docker exec products mongosh `
>   --username admin `
>   --password secret123 `
>   --authenticationDatabase admin `
>   --eval 'db.getSiblingDB("productsdb").products.find().toArray()'
[]
```

Datenbank wiederherstellen:

```
Oliver ~/../school/m165/g1f_api
# docker exec products mongorestore `
>   --username admin `
>   --password secret123 `
>   --authenticationDatabase admin `
>   --drop `
>   /backup/productsdb_backup
2025-05-22T11:37:26.818+0000    preparing collections to restore from
2025-05-22T11:37:26.818+0000    don't know what to do with file "/backup/productsdb_backup/productsdb/prelude.json", skipping ...
2025-05-22T11:37:26.818+0000    reading metadata for productsdb.products from /backup/productsdb_backup/productsdb/products.metadata.json
2025-05-22T11:37:26.829+0000    restoring productsdb.products from /backup/productsdb_backup/productsdb/products.bson
2025-05-22T11:37:26.840+0000    finished restoring productsdb.products (4 documents, 0 failures)
2025-05-22T11:37:26.840+0000    no indexes to restore for collection productsdb.products
2025-05-22T11:37:26.840+0000    4 document(s) restored successfully. 0 document(s) failed to restore.
```

Datenbank wiederherstellen:

```
Oliver ~/../school/m165/g1f_api
# docker exec products mongosh `
>   --username admin `
>   --password secret123 `
>   --authenticationDatabase admin `
>   --eval 'db.getSiblingDB("productsdb").products.find().toArray()'
[
  {
    _id: ObjectId('681c9da20ab742495142d33c'),
    productId: 'P002',
    name: 'Laptop',
    category: 'Electronics',
    price: 1299
  },
  {
    _id: ObjectId('681c9da20ab742495142d33d'),
    productId: 'P004',
    name: 'Coffee Mug',
    category: 'Kitchen',
    price: 12.99
  },
  {
    _id: ObjectId('681c9da20ab742495142d33e'),
    productId: 'P003',
    name: 'Desk Chair',
    category: 'Furniture',
    price: 149.5
  },
  {
    _id: ObjectId('681c9da20ab742495142d33f'),
    productId: 'P001',
    name: 'Smartphone'
```

```
    name: 'Smartphone',
    category: 'Electronics',
    price: 699.99
  }
]
```