

Day -4

User Creation in MongoDB



In MongoDB, a user is an account that can be assigned roles and permissions to control who can access which databases and what actions they can perform.

In MongoDB, all user information is stored in the admin database by default — but technically, users are stored in the database where they are created.

✦ Important Notes:

When MongoDB authentication is enabled, no one can access any DB without logging in with a valid user.

It's good practice to create users with only the permissions they need — least privilege.

📁 Where Are MongoDB Users Stored?

Type of User	Stored In Database
Global/Admin Users	admin
Regular DB Users	That specific DB

Create an admin user:

```
use admin
db.createUser({
  user: "admin",
  pwd: "yourpassword",
  roles: [ { role: "userAdminAnyDatabase", db: "admin" }, "readWriteAnyDatabase" ]
})
```

```
admin> db.createUser({user:"admin",pwd:"r3p0rt",roles:[{role:"userAdminAnyDatabase",
db:"admin"},"readWriteAnyDatabase"]})
{ ok: 1 }
admin>
```



Note : The user exists, but has no privileges, so if you enable --auth, the user won't be able to do anything until you assign roles using db.grantRolesToUser().

Below is list of common built-in roles in MongoDB, grouped by their purpose:

Database user roles

Role	Description
read	Read-only access to data
readWrite	Read and write access to data

```
db.createUser({
  user: "admin",
  pwd: "securePassword123",
  roles: [ ]
})
```

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTi
admin> db.createUser({user:"DBA",pwd:"R3p0r1",roles:[ ]})
{ ok: 1 }
admin>
```

Database Administrator roles

Role	Description
dbAdmin	Can view and modify database metadata (e.g., indexes, stats)
userAdmin	Can create and manage users in the current database
dbOwner	Full access to a single database (includes all above roles)

Cluster Administration Roles

Role	Description
clusterAdmin	Full control over the cluster
clusterManager	Manage and monitor the cluster (no user creation)
clusterMonitor	Read-only access to monitoring and diagnostics
hostManager	Manage server configuration and monitoring

Backup & Restore Roles

Role	Description
backup	Read access to all data for backups
restore	Write access for restoring data

List of Roles to see

```
use admin
db.runCommand({ rolesInfo: 1, showBuiltinRoles: true })
```

All database roles (must be in admin

Role	Description
readAnyDatabase	Read-only access to all databases
readWriteAnyDatabase	Read/write access to all databases
userAdminAnyDatabase	Manage users on all databases
dbAdminAnyDatabase	Database admin access to all databases

Let's do practice

to see the user list

```
C:\> mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeo

admin> show users;
[
  {
    _id: 'admin.DBA',
    userId: UUID('1de2fc29-20aa-466d-a264-324f9c74c6f7'),
    user: 'DBA',
    db: 'admin',
    roles: [],
    mechanisms: [ 'SCRAM-SHA-1', 'SCRAM-SHA-256' ]
  }
]
admin>
```

Create a user in different database

```
Mobile> db.employee.insertOne({id:1,Name:"Ajit Yadav",mob:"2298475834",roles:[]})
{
  acknowledged: true,
  insertedId: ObjectId('681b14151eeecaf1b9b5f89a')
}
Mobile>
```

in this we can see list of all user in all db

```
admin> db.getSiblingDB("admin").system.users.find()
[
  {
    _id: 'admin.DBA',
    userId: UUID('1de2fc29-20aa-466d-a264-324f9c74c6f7'),
    user: 'DBA',
    db: 'admin',
    credentials: {
      'SCRAM-SHA-1': {
        iterationCount: 10000,
        salt: 'NgBLkfrLQ+X0RpoJalX5cA==',
        storedKey: 'Mj1Nw1YarOPqF9fPQEcPipt5M6g=',
        serverKey: 'u4kvKGzYfujw+bTcV2/iP4LtLOg='
      },
      'SCRAM-SHA-256': {
        iterationCount: 15000,
        salt: 'SnKsXoDnw7ZiY3WQv3UxzY0GKcQKTXI5XeDHJQ==',
        storedKey: 'MJSAXU7xH+WuKPTofDL6ZPsYYh+6IL34U5f3uJk+FyA=',
        serverKey: 'CxMdKwRZ+ApBRQFw0mvAt5oPqvmL1sc7rB3/qvecabE='
      }
    },
    roles: []
  }
]
```

Here i create a user in mobile db with readwrite permisson

```
Mobile> db.createUser({user:"read",pwd:"r3ad",roles:[{role:"readWrite",db:"Mobile"}]})
{ ok: 1 }
Mobile> show users;
[
  {
    _id: 'Mobile.read',
    userId: UUID('9e13c8a6-0b2c-42b1-bd1a-fa266cccc8c5'),
    user: 'read',
    db: 'Mobile',
    roles: [ { role: 'readWrite', db: 'Mobile' } ],
    mechanisms: [ 'SCRAM-SHA-1', 'SCRAM-SHA-256' ]
  }
]
Mobile>
```

Login in mongoddb

mongo -u <username> -p <password> --authenticationDatabase <auth_db>

example : mongo -u admin -p secret123 --authenticationDatabase admin