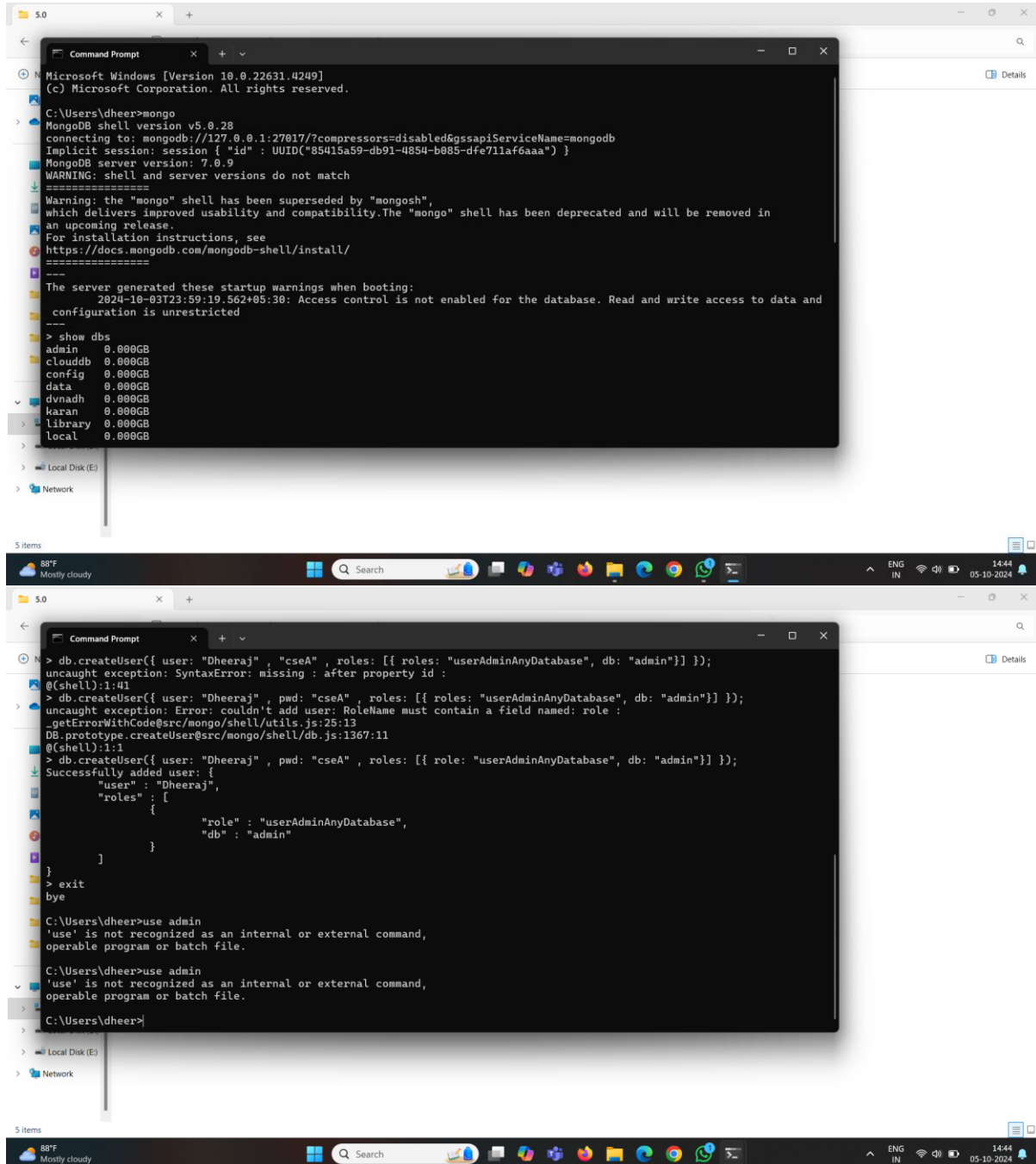


NOSQL DATABASE -USER AND ROLES CREATION

NAME=DHEERAJ VERMA

ROLL NO=22BCS025



The image displays two screenshots of a Windows Command Prompt window, showing the process of connecting to a MongoDB instance and creating a user.

First Screenshot: The user runs the command `C:\Users\dheer>mongo`. The output shows the MongoDB shell version (v5.0.28) connecting to `mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb`. It displays the implicit session ID and the MongoDB server version (7.0.9). A warning message states: "Warning: the 'mongo' shell has been superseded by 'mongosh', which delivers improved usability and compatibility. The 'mongo' shell has been deprecated and will be removed in an upcoming release. For installation instructions, see https://docs.mongodb.com/mongodb-shell/install/". Below this, it shows startup warnings from the server, including a message about access control not being enabled for the database. The user then runs `> show dbs`, which lists the databases and their sizes:

Database	Size
admin	0.000GB
clouddb	0.000GB
config	0.000GB
data	0.000GB
dynadh	0.000GB
karan	0.000GB
library	0.000GB
local	0.000GB

Second Screenshot: The user runs the command `> db.createUser({ user: "Dheeraj", "cseA", roles: [{ roles: "userAdminAnyDatabase", db: "admin" }] });`. This results in a `SyntaxError: missing : after property id`. The user then runs `> db.createUser({ user: "Dheeraj", pwd: "cseA", roles: [{ roles: "userAdminAnyDatabase", db: "admin" }] });`, which results in an `uncaught exception: Error: couldn't add user: roleName must contain a field named: role`. The user then runs `> db.createUser({ user: "Dheeraj", pwd: "cseA", roles: [{ role: "userAdminAnyDatabase", db: "admin" }] });`, which successfully adds the user. The output shows the user details: `"user": "Dheeraj", "roles": [{"role": "userAdminAnyDatabase", "db": "admin"}]}`. The user then runs `> exit` and `bye`. Finally, the user runs `C:\Users\dheer>use admin`, which results in an error: `'use' is not recognized as an internal or external command, operable program or batch file.`

```
C:\Windows\System32\cmd.exe x + v
Microsoft Windows [Version 10.0.22631.4249]
(c) Microsoft Corporation. All rights reserved.

C:\Program Files\MongoDB\Server\5.0>show dbs
'show' is not recognized as an internal or external command,
operable program or batch file.

C:\Program Files\MongoDB\Server\5.0>use admin
'use' is not recognized as an internal or external command,
operable program or batch file.

C:\Program Files\MongoDB\Server\5.0>cd mongod
The system cannot find the path specified.

C:\Program Files\MongoDB\Server\5.0>mongo
MongoDB shell version v5.0.28
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("c48bfbae-61d1-47a0-a6d5-68fd8671fee6") }
MongoDB server version: 7.0.9
WARNING: shell and server versions do not match
=====
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
=====
---
The server generated these startup warnings when booting:
2024-10-03T23:59:19.562+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
---
> show dbs
admin      0.000GB
clouddb    0.000GB
config     0.000GB
data       0.000GB
dvnadh     0.000GB
karan      0.000GB
library    0.000GB
local      0.000GB
mydb       0.000GB

C:\Windows\System32\cmd.exe x + v
newfile 0.001GB
> use admin
switched to db admin
> db.createUser({ user: "Dheeraj", pwd: "cseA", roles: [{ role: "userAdminAnyDatabase", db: "admin" }] });
Successfully added user: {
  "user" : "Dheeraj",
  "roles" : [
    {
      "role" : "userAdminAnyDatabase",
      "db" : "admin"
    }
  ]
}
> exit
bye

C:\Program Files\MongoDB\Server\5.0>mongod --auth --port 27017
{"t":{"$date":"2024-10-05T14:19:26.033+05:30"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"-", "msg":"Automatically disabling TLS 1.0, to force-enable TLS
1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2024-10-05T14:19:26.039+05:30"},"s":"I", "c":"NETWORK", "id":4915701, "ctx":"-", "msg":"Initialized wire specification", "attr":{"spec":{"inc
omingExternalClient":{"minWireVersion":0,"maxWireVersion":13},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":13},"outgoing":{"minWireVersion"
:0,"maxWireVersion":13},"isInternalClient":true}}}
{"t":{"$date":"2024-10-05T14:19:26.886+05:30"},"s":"W", "c":"ASIO", "id":22601, "ctx":"thread1", "msg":"No TransportLayer configured during NetworkInt
erface startup"}
{"t":{"$date":"2024-10-05T14:19:26.887+05:30"},"s":"I", "c":"NETWORK", "id":4648602, "ctx":"thread1", "msg":"Implicit TCP FastOpen in use."}
{"t":{"$date":"2024-10-05T14:19:26.890+05:30"},"s":"W", "c":"ASIO", "id":22601, "ctx":"thread1", "msg":"No TransportLayer configured during NetworkInt
erface startup"}
{"t":{"$date":"2024-10-05T14:19:26.890+05:30"},"s":"I", "c":"REPL", "id":5123008, "ctx":"thread1", "msg":"Successfully registered PrimaryOnlyService", "a
tr":{"service":"TenantMigrationDonorService", "ns":"config.tenantMigrationDonors"}}
{"t":{"$date":"2024-10-05T14:19:26.890+05:30"},"s":"I", "c":"REPL", "id":5123008, "ctx":"thread1", "msg":"Successfully registered PrimaryOnlyService", "a
tr":{"service":"TenantMigrationRecipientService", "ns":"config.tenantMigrationRecipients"}}
{"t":{"$date":"2024-10-05T14:19:26.891+05:30"},"s":"I", "c":"CONTROL", "id":5945603, "ctx":"thread1", "msg":"Multi threading initialized"}
{"t":{"$date":"2024-10-05T14:19:26.893+05:30"},"s":"I", "c":"CONTROL", "id":4615611, "ctx":"initandlisten", "msg":"MongoDB starting", "attr":{"pid":9672, "po
rt":27017, "dbPath":"C:/data/db/", "architecture":"64-bit", "host":"LAPTOP-OV00JUHJ"}}
{"t":{"$date":"2024-10-05T14:19:26.894+05:30"},"s":"I", "c":"CONTROL", "id":23398, "ctx":"initandlisten", "msg":"Target operating system minimum version
", "attr":{"targetMinOS":"Windows 7/Windows Server 2008 R2"}}
{"t":{"$date":"2024-10-05T14:19:26.894+05:30"},"s":"I", "c":"CONTROL", "id":22403, "ctx":"initandlisten", "msg":"Build Info", "attr":{"buildInfo":{"versio
n":"5.0.28", "gitVersion":"a8f8b8e1e271f236e761d0138e2418d0a114c941", "modules":[], "allocator":"tcmalloc", "environment":{"distmod":"windows", "distarch":"x86_6
4", "target_arch":"x86_64"}}}}
{"t":{"$date":"2024-10-05T14:19:26.894+05:30"},"s":"I", "c":"CONTROL", "id":51765, "ctx":"initandlisten", "msg":"Operating System", "attr":{"os":{"name":"
Microsoft Windows 10", "version":"10.0 (build 22631)"}}}
```

```
C:\Windows\System32\cmd.exe
{"t":{"date":"2024-10-05T14:19:26.893+05:30"},"s":"I", "c":"CONTROL", "id":4615611, "ctx":"initandlisten","msg":"MongoDB starting","attr":{"pid":9672,"port":27017,"dbPath":"C:/data/db/","architecture":"64-bit","host":"LAPTOP-OV00JUHJ"}}
{"t":{"date":"2024-10-05T14:19:26.894+05:30"},"s":"I", "c":"CONTROL", "id":23398, "ctx":"initandlisten","msg":"Target operating system minimum version","attr":{"targetMinOS":"Windows 7/Windows Server 2008 R2"}}
{"t":{"date":"2024-10-05T14:19:26.894+05:30"},"s":"I", "c":"CONTROL", "id":23403, "ctx":"initandlisten","msg":"Build Info","attr":{"buildInfo":{"version":"5.0.28","gitVersion":"a8f8b8e1e271f236e761d0138e2418d0a114c941","modules":[],"allocator":"tcmalloc","environment":{"distmod":"windows","distarch":"x86_64","target_arch":"x86_64"}}}}
{"t":{"date":"2024-10-05T14:19:26.894+05:30"},"s":"I", "c":"CONTROL", "id":51765, "ctx":"initandlisten","msg":"Operating System","attr":{"os":{"name":"Microsoft Windows 10","version":"10.0 (build 22631)}}}}
{"t":{"date":"2024-10-05T14:19:26.894+05:30"},"s":"I", "c":"CONTROL", "id":21951, "ctx":"initandlisten","msg":"Options set by command line","attr":{"options":{"net":{"port":27017,"security":{"authorization":"enabled"}}}}}}
{"t":{"date":"2024-10-05T14:19:26.897+05:30"},"s":"E", "c":"CONTROL", "id":20557, "ctx":"initandlisten","msg":"DBException in initAndListen, terminating","attr":{"error":"DBPthreadUse: Unable to create/open the lock file: C:\\data\\db\\mongod.lock (The process cannot access the file because it is being used by another process.) Ensure the user executing mongod is the owner of the lock file and has the appropriate permissions. Also make sure that another mongod instance is not already running on the C:\\data\\db\\ directory"}}}
{"t":{"date":"2024-10-05T14:19:26.898+05:30"},"s":"I", "c":"REPL", "id":4784900, "ctx":"initandlisten","msg":"Stepping down the ReplicationCoordinator for shutdown","attr":{"waitTimeMillis":15000}}
{"t":{"date":"2024-10-05T14:19:26.898+05:30"},"s":"I", "c":"COMMAND", "id":4784901, "ctx":"initandlisten","msg":"Shutting down the MirrorMaestro"}
{"t":{"date":"2024-10-05T14:19:26.898+05:30"},"s":"I", "c":"SHARDING", "id":4784902, "ctx":"initandlisten","msg":"Shutting down the WaitForMajorityService"}
{"t":{"date":"2024-10-05T14:19:26.898+05:30"},"s":"I", "c":"NETWORK", "id":20562, "ctx":"initandlisten","msg":"Shutdown: going to close listening sockets"}
{"t":{"date":"2024-10-05T14:19:26.898+05:30"},"s":"I", "c":"NETWORK", "id":4784905, "ctx":"initandlisten","msg":"Shutting down the global connection pool"}
{"t":{"date":"2024-10-05T14:19:26.898+05:30"},"s":"I", "c":"CONTROL", "id":4784906, "ctx":"initandlisten","msg":"Shutting down the FlowControlTicketHolder"}
{"t":{"date":"2024-10-05T14:19:26.898+05:30"},"s":"I", "c":"-", "id":20520, "ctx":"initandlisten","msg":"Stopping further FlowControl ticket acquisition."}
{"t":{"date":"2024-10-05T14:19:26.898+05:30"},"s":"I", "c":"NETWORK", "id":4784918, "ctx":"initandlisten","msg":"Shutting down the ReplicaSetMonitor"}
{"t":{"date":"2024-10-05T14:19:26.899+05:30"},"s":"I", "c":"SHARDING", "id":4784921, "ctx":"initandlisten","msg":"Shutting down the MigrationUtilExecutor"}
{"t":{"date":"2024-10-05T14:19:26.899+05:30"},"s":"I", "c":"ASIO", "id":22582, "ctx":"MigrationUtil-TaskExecutor","msg":"Killing all outstanding egress activity."}
{"t":{"date":"2024-10-05T14:19:26.900+05:30"},"s":"I", "c":"COMMAND", "id":4784923, "ctx":"initandlisten","msg":"Shutting down the ServiceEntryPoint"}
{"t":{"date":"2024-10-05T14:19:26.900+05:30"},"s":"I", "c":"CONTROL", "id":4784928, "ctx":"initandlisten","msg":"Shutting down the TTL monitor"}
{"t":{"date":"2024-10-05T14:19:26.900+05:30"},"s":"I", "c":"CONTROL", "id":4784929, "ctx":"initandlisten","msg":"Acquiring the global lock for shutdown"}
{"t":{"date":"2024-10-05T14:19:26.900+05:30"},"s":"I", "c":"-", "id":4784931, "ctx":"initandlisten","msg":"Dropping the scope cache for shutdown"}
{"t":{"date":"2024-10-05T14:19:26.900+05:30"},"s":"I", "c":"FTDC", "id":4784926, "ctx":"initandlisten","msg":"Shutting down full-time data capture"}
{"t":{"date":"2024-10-05T14:19:26.900+05:30"},"s":"I", "c":"CONTROL", "id":20565, "ctx":"initandlisten","msg":"Now exiting"}
{"t":{"date":"2024-10-05T14:19:26.901+05:30"},"s":"I", "c":"CONTROL", "id":23138, "ctx":"initandlisten","msg":"Shutting down","attr":{"exitCode":100}}

88°F Mostly cloudy
C:\Windows\System32\cmd.exe
'show' is not recognized as an internal or external command,
operable program or batch file.

C:\Program Files\MongoDB\Server\5.0>mongo --port 27017 -u "Dheeraj" -p "cseA" --authenticationDatabase "admin"
MongoDB shell version v5.0.28
connecting to: mongodb://127.0.0.1:27017/?authSource=admin&compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("83f3366d-0a91-49fb-b80a-d2f4a3f1bb1") }
MongoDB server version: 7.0.9
WARNING: shell and server versions do not match
=====
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
=====

The server generated these startup warnings when booting:
----
2024-10-03T23:59:19.562+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
----
> use test
switched to db test
> db.createUser({ user: "Anudeep", pwd: "xyz123", roles: [{ roles: "readwrite", db:"test" }] });
uncaught exception: Error: couldn't add user: RoleName must contain a field named: role :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
DB.prototype.createUser@src/mongo/shell/db.js:1367:11
@(shell):1:1
> db.createUser({ user: "Anudeep", pwd: "xyz123", roles: [{ role: "readwrite", db:"test" }] });
uncaught exception: Error: couldn't add user: Could not find role: readwrite@test :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
DB.prototype.createUser@src/mongo/shell/db.js:1367:11
@(shell):1:1
> db.createUser({ user: "anudeep", pwd: "xyz123", roles: [{ roles: "readwrite", db:"test" }] });
uncaught exception: Error: couldn't add user: RoleName must contain a field named: role :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
DB.prototype.createUser@src/mongo/shell/db.js:1367:11
@(shell):1:1
> db.createUser({ user: "Anudeep", pwd: "xyz123", roles: [{ roles: "readWrite", db:"test" }] });
uncaught exception: Error: couldn't add user: RoleName must contain a field named: role :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
DB.prototype.createUser@src/mongo/shell/db.js:1367:11
@(shell):1:1
```

```
C:\Windows\System32\cmd.exe
=====
The server generated these startup warnings when booting:
  2024-10-03T23:59:19.562+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----
> use test
switched to db test
> db.createUser({ user: "Anudeep", pwd: "xyz123", roles: [{ roles: "readwrite", db:"test"}] });
uncaught exception: Error: couldn't add user: RoleName must contain a field named: role :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
DB.prototype.createUser@src/mongo/shell/db.js:1367:11
@(<shell>):1:1
> db.createUser({ user: "Anudeep", pwd: "xyz123", roles: [{ role: "readwrite", db:"test"}] });
uncaught exception: Error: couldn't add user: Could not find role: readwrite@test :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
DB.prototype.createUser@src/mongo/shell/db.js:1367:11
@(<shell>):1:1
> db.createUser({ user: "anudeep", pwd: "xyz123", roles: [{ roles: "readwrite", db:"test"}] });
uncaught exception: Error: couldn't add user: RoleName must contain a field named: role :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
DB.prototype.createUser@src/mongo/shell/db.js:1367:11
@(<shell>):1:1
> db.createUser({ user: "Anudeep", pwd: "xyz123", roles: [{ roles: "readWrite", db:"test"}] });
uncaught exception: Error: couldn't add user: RoleName must contain a field named: role :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
DB.prototype.createUser@src/mongo/shell/db.js:1367:11
@(<shell>):1:1
> db.createUser({ user: "Anudeep", pwd: "xyz123", roles: [{ role: "readWrite", db:"test"}] });
Successfully added user: {
  "user" : "Anudeep",
  "roles" : [
    {
      "role" : "readWrite",
      "db" : "test"
    }
  ]
}
> |

C:\Windows\System32\cmd.exe
@(<shell>):1:1
> db.createUser({ user: "Anudeep", pwd: "xyz123", roles: [{ roles: "readWrite", db:"test"}] });
uncaught exception: Error: couldn't add user: RoleName must contain a field named: role :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
DB.prototype.createUser@src/mongo/shell/db.js:1367:11
@(<shell>):1:1
> db.createUser({ user: "Anudeep", pwd: "xyz123", roles: [{ role: "readWrite", db:"test"}] });
Successfully added user: {
  "user" : "Anudeep",
  "roles" : [
    {
      "role" : "readWrite",
      "db" : "test"
    }
  ]
}
> show dbs
admin      0.000GB
cloudadb  0.000GB
config     0.000GB
data       0.000GB
dvnadh     0.000GB
karan      0.000GB
library    0.000GB
local      0.000GB
mydb       0.000GB
naya       0.000GB
newdb      0.000GB
newfile    0.001GB
> use admin
switched to db admin
> db.dropUser('Dheeraj')
true
> db.system.user.find()
> db.system.users.find()
{ "id" : "admin.Ram", "userId" : UUID("e84f7f80-70ad-4c09-9285-27be81a27099"), "user" : "Ram", "db" : "admin", "credentials" : { "SCRAM-SHA-1" : { "iterationCount" : 10000, "salt" : "c63prQsaoXf1x6Rul/SUow==", "storedKey" : "Z1kcRaQJtav5AECU7VHaaxZEI8E=", "serverKey" : "iilHyC5wt3XmfEAut+CEJYT23TI=", "SCRAM-SHA-256" : { "iterationCount" : 15000, "salt" : "grstw3KshDovyV7vhlbntEMBD;E0VXSs7x2z3g==", "storedKey" : "n+2IBjXVPV+/NsepHU02A6wSP6kxc41Cz9UK3VbwFkg=", "serverKey" : "FmYPHNNlug+2Jk+9rUFk/nKMoVj3SxtS7hq122Zas=" } }, "roles" : [ { "role" : "userAdminAnyDatabase", "db" : "admin" } ] }
{ "id" : "test.Dheeraj", "userId" : UUID("2a4ef3cb-792c-49d1-8a00-bc5f3e874ea4"), "user" : "Dheeraj", "db" : "test", "credentials" : { "SCRAM-SHA-1" : { "iterationCount" : 10000, "salt" : "KEuodMwdOyGuu1QCyisDmw==", "storedKey" : "XIZcoqNa4fWz2hr8xqP9NVw3Hm4=", "serverKey" : "x+qVcly5XG6VcpvRadTrL2Mrjdc=" }, "
```

```
C:\Windows\System32\cmd.exe
newdb 0.000GB
newfile 0.001GB
> use admin
switched to db admin
> db.dropUser('Dheeraj')
true
> db.system.user.find()
> db.system.users.find()
{ "_id" : "admin.Ram", "userId" : UUID("e84f7f80-70ad-4c09-9285-27be81a27099"), "user" : "Ram", "db" : "admin", "credentials" : { "SCRAM-SHA-1" : { "iterationCount" : 10000, "salt" : "c6JprQsaoXflx6Rul/5Uow==", "storedKey" : "ZlkcRaQJtav5AECU7VHaaxZEIBE=", "serverKey" : "iilHyC5wT3XmfEAut+CEJYTZ3TI=" }, "SCRAM-SHA-256" : { "iterationCount" : 15000, "salt" : "grstw3KshDovyV7vhLbntEMBDJE9VXS57x2zJg==", "storedKey" : "m+2IBjXVPV+/NsepHUB2A6wSP6kxc41Cz9UK3VbwFkg=", "serverKey" : "FmyPHNniug+2Jk+9rUFk/nKCMoVj3Sxt57hq122ZZas=" } }, "roles" : [ { "role" : "userAdminAnyDatabase", "db" : "admin" } ] }, { "_id" : "test.Dheeraj", "userId" : UUID("Zadef3cb-792c-40d1-8a00-bc5f3874aaU"), "user" : "Dheeraj", "db" : "test", "credentials" : { "SCRAM-SHA-1" : { "iterationCount" : 10000, "salt" : "MEuodMwdOyGuniQCyisDmm==", "storedKey" : "XIZcoqMa4fWz2hr8vqP9NVw3HmId=", "serverKey" : "x+qQcilyXG6VcpvRadTrL2Wtjdc=" }, "SCRAM-SHA-256" : { "iterationCount" : 15000, "salt" : "lX+XqA208+dxTm1J4MLHwUnDh+MbaisTtQ1aw==", "storedKey" : "BRZGelAFHxLUBCwe/k4UzInHFrmlUEw4vs7q/1lcGQ=", "serverKey" : "Opas+9ZplwiS9VeJH+sGLPJPO10BVSy/aTysdOVbyM=" } }, "roles" : [ { "role" : "userAdminAnyDatabase", "db" : "admin" } ] }, { "_id" : "test.Anudeep", "userId" : UUID("efe49e59-c3db-470b-ae68-c72fe478bb85"), "user" : "Anudeep", "db" : "test", "credentials" : { "SCRAM-SHA-1" : { "iterationCount" : 10000, "salt" : "gCnweyhiGtcwb3soufdoPg==", "storedKey" : "n512ik1UXzKTRqYJX7eyRj4jdg0=", "serverKey" : "3aQMoqMLiXrwR3Uy5o1efRyG954=" }, "SCRAM-SHA-256" : { "iterationCount" : 15000, "salt" : "uLHJ23iAiQk8tPUDQ0EYb5C5SGmhPN55Fz9K5w==", "storedKey" : "MX9QcLDcsyKCHbAAOZ5F/VnqSSM/40HmdHteuTuyS/k=", "serverKey" : "JYvLrKHF+25x2+Rd2xvFBqgDeBBhJ9+hjhpmDhQm2o=" } }, "roles" : [ { "role" : "readWrite", "db" : "test" } ] } ]
> db.getUsers()
[
  {
    "_id" : "admin.Ram",
    "userId" : UUID("e84f7f80-70ad-4c09-9285-27be81a27099"),
    "user" : "Ram",
    "db" : "admin",
    "roles" : [
      {
        "role" : "userAdminAnyDatabase",
        "db" : "admin"
      }
    ],
    "mechanisms" : [
      "SCRAM-SHA-1",
      "SCRAM-SHA-256"
    ]
  }
]
> db.getUser('Dheeraj')
null

"db" : "admin",
"roles" : [
  {
    "role" : "userAdminAnyDatabase",
    "db" : "admin"
  }
],
"mechanisms" : [
  "SCRAM-SHA-1",
  "SCRAM-SHA-256"
]
]
> db.getUser('Dheeraj')
null
> db.getUser('Anudeep')
null
> db.getUser('anudeep')
null
> use test
switched to db test
> db.getUser('Anudeep')
{
  "_id" : "test.Anudeep",
  "userId" : UUID("efe49e59-c3db-470b-ae68-c72fe478bb85"),
  "user" : "Anudeep",
  "db" : "test",
  "roles" : [
    {
      "role" : "readWrite",
      "db" : "test"
    }
  ],
  "mechanisms" : [
    "SCRAM-SHA-1",
    "SCRAM-SHA-256"
  ]
}
> db.logout()
{ "ok" : 1 }
> |
```

```
C:\Windows\System32\cmd.exe x + v
}
> db.logout()
{ "ok" : 1 }
> show dbs
admin 0.000GB
clouddb 0.000GB
config 0.000GB
data 0.000GB
dvnadh 0.000GB
karan 0.000GB
library 0.000GB
local 0.000GB
mydb 0.000GB
naya 0.000GB
newdb 0.000GB
newfile 0.001GB
> use person
switched to db person
> db.createUser({ user: "gokulesh", pwd: "password", roles: [{ role: "readWrite", db: "person" }] });
Successfully added user: {
  "user" : "gokulesh",
  "roles" : [
    {
      "role" : "readWrite",
      "db" : "person"
    }
  ]
}
> db.grantRolesToUser("gokulesh", [{ role: "readWrite", db: "person"}, {role: "dbAdmin", db: "person"}]);
uncaught exception: TypeError: db.grantRolesToUser is not a function :
@ (shell):1:1
> db.grantRolesToUser("gokulesh", [{ role: "readWrite", db: "person"}, {role: "dbAdmin", db: "person"}]);
> db.getUser("gokulesh");
{
  "_id" : "person.gokulesh",
  "userId" : UUID("d4e6bb8-c2f8-42e6-9d9c-3d13f3985093"),
  "user" : "gokulesh",
  "db" : "person",
  "roles" : [
    {
      "role" : "dbAdmin",
      "db" : "person"
    },
    {
      "role" : "readWrite",
      "db" : "person"
    }
  ]
}

C:\Windows\System32\cmd.exe x + v
"role" : "readWrite",
"db" : "person"
}
]
}
> db.grantRolesToUser("gokulesh", [{ role: "readWrite", db: "person"}, {role: "dbAdmin", db: "person"}]);
uncaught exception: TypeError: db.grantRolesToUser is not a function :
@ (shell):1:1
> db.grantRolesToUser("gokulesh", [{ role: "readWrite", db: "person"}, {role: "dbAdmin", db: "person"}]);
> db.getUser("gokulesh");
{
  "_id" : "person.gokulesh",
  "userId" : UUID("d4e6bb8-c2f8-42e6-9d9c-3d13f3985093"),
  "user" : "gokulesh",
  "db" : "person",
  "roles" : [
    {
      "role" : "dbAdmin",
      "db" : "person"
    },
    {
      "role" : "readWrite",
      "db" : "person"
    }
  ],
  "mechanisms" : [
    "SCRAM-SHA-1",
    "SCRAM-SHA-256"
  ]
}
> db.revokeRolesFromUser({"gokulesh", [{role: "dbAdmin", db: "person"}]);
...
> db.createRole({role: "customRole", privileges: [{ resource: {db: "person", collection: ""}, actions: ["find", "insert", "update", "remove"]} ], roles: [{role: "read", db: "person"}]});
uncaught exception: SyntaxError: missing ] after element list :
@ (shell):1:112
> db.createRole({role: "customRole", privileges: [{ resource: {db: "person", collection: ""}, actions: ["find", "insert", "update", "remove"]} ], roles: [{role: "read", db: "person"}]});
uncaught exception: SyntaxError: missing ] after element list :
@ (shell):1:112
> db.createRole({role: "customRole", privileges: [{ resource: {db: "person", collection: ""}, actions: ["find", "insert", "update", "remove"]} ], roles: [{role: "read", db: "person"}]});
```

```
C:\Windows\System32\cmd.e x + v
db:"person"]]]});
uncaught exception: SyntaxError: missing ] after element list :
@(shell):1:112
> db.createRole({
...   role: "customRole",
...   privileges: [
...     {
...       resource: { db: "person", collection: "" },
...       actions: ["find", "insert", "update", "remove"]
...     }
...   ],
...   roles: [
...     { role: "read", db: "person" }
...   ]
... });
{
  "role" : "customRole",
  "privileges" : [
    {
      "resource" : {
        "db" : "person",
        "collection" : ""
      },
      "actions" : [
        "find",
        "insert",
        "update",
        "remove"
      ]
    }
  ],
  "roles" : [
    {
      "role" : "read",
      "db" : "person"
    }
  ]
}
>
> db.createRole({
>
> db.createRole({
...   role: "customRole",
...   privileges: [
...     {
...       resource: { db: "person", collection: "" },
...       actions: ["find", "insert", "update", "remove"]
...     }
...   ],
...   roles: [
...     { role: "read", db: "person" }
...   ]
... });
uncaught exception: Error: Role "customRole@person" already exists :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
DB.prototype.createRole@src/mongo/shell/db.js:1656:15
@(shell):1:1
>
>
>
>
>
>
>
>
>
>
> db.revokeRolesFromUser( "gokulesh", [{ role: "dbAdmin",db:"persion"}]);
> db.getUser('gokulesh')
{
  "_id" : "person.gokulesh",
  "userId" : UUID("d4e6bb8-c2f8-42e6-9d9c-3d13f3985093"),
  "user" : "gokulesh",
  "db" : "person",
  "roles" : [
    {
      "role" : "readWrite",
      "db" : "person"
    }
  ],
  "mechanisms" : [
    "SCRAM-SHA-1",
    "SCRAM-SHA-256"
  ]
}
```



```
C:\Windows\System32\cmd.exe x + v
>
> db.revokeRolesFromUser( "gokulesh", [{ role: "dbAdmin", db: "persion"}]);
> db.getUser('gokulesh')
{
  "_id" : "person.gokulesh",
  "userId" : UUID("d4e6bb8-c2f8-42e6-9d9c-3d13f3985093"),
  "user" : "gokulesh",
  "db" : "person",
  "roles" : [
    {
      "role" : "readWrite",
      "db" : "person"
    }
  ],
  "mechanisms" : [
    "SCRAM-SHA-1",
    "SCRAM-SHA-256"
  ]
}
> db.dropRole("customRole");
true
> db.createRole({
... .. role: "customRole",
... .. privileges: [
... .. {
... ..   resource: { db: "person", collection: "" },
... ..   actions: ["find", "insert", "update", "remove"]
... .. },
... .. ],
... .. roles: [
... .. { role: "read", db: "person" }
... .. ]
... .. });
{
  "role" : "customRole",
  "privileges" : [
    {
      "resource" : {
        "db" : "person",
        "collection" : ""
      },
      "actions" : [
        "find",
        "insert",
        "update",
        "remove"
      ]
    }
  ],
  "roles" : [
    {
      "role" : "read",
      "db" : "person"
    }
  ]
}
> db.dropRole("customRole")
true
> db.getRole();
uncaught exception: Error: Role name for getRole shell helper must be a string :
DB.prototype.getRole@src/mongo/shell/db.js:1749:15
@(shell):1:1
```



```
C:\Windows\System32\cmd.exe
]
> db.dropRole("customRole")
true
> db.getRole();
uncaught exception: Error: Role name for getRole shell helper must be a string :
DB.prototype.getRole@src/mongo/shell/db.js:1749:15
@(shell):1:1
> db.getRoles();
[ ]
> db.createRole({
...   role: "customRole",
...   privileges: [
...     {
...       resource: { db: "person", collection: "" },
...       actions: ["find", "insert", "update", "remove"]
...     }
...   ],
...   roles: [
...     { role: "read", db: "person" }
...   ]
... });
{
  "role" : "customRole",
  "privileges" : [
    {
      "resource" : {
        "db" : "person",
        "collection" : ""
      },
      "actions" : [
        "find",
        "insert",
        "update",
        "remove"
      ]
    }
  ],
  "roles" : [
    {
      "role" : "read",
      "db" : "person"
    }
  ]
}
> db.getRoles();
[
  {
    "_id" : "person.customRole",
    "role" : "customRole",
    "db" : "person",
    "roles" : [
      {
        "role" : "read",
        "db" : "person"
      }
    ],
    "isBuiltin" : false,
    "inheritedRoles" : [
      {
        "role" : "read",
        "db" : "person"
      }
    ]
  }
]
> db.grantPrivilegesToRole( "customRole", [{ resource: { db: "person", collection: "employees" }, actions: [ "find", "insert" ] } ] );
uncaught exception: SyntaxError: missing } after property list :
@(shell):1:92
> db.grantPrivilegesToRole( "customRole", [{ resource: { db: "person", collection: "employees" }, actions: [ "find", "insert" ] } ] );
uncaught exception: SyntaxError: missing } after property list :
@(shell):1:93
> db.grantPrivilegesToRole(
...   "customRole",
...   [
...     {
...       resource: { db: "person", collection: "employees" },
...       actions: ["find", "insert"]
...     }
...   ]
... );
```

```
C:\Windows\System32\cmd.exe x + v
]
}
]
> db.grantPrivilegesToRole( "customRole", [{ resource: { db: "person", collection: "employees" }, actions: [ "find", "insert" ] } ] );
uncaught exception: SyntaxError: missing } after property list :
@shell:1:92
> db.grantPrivilegesToRole( "customRole", [{ resource: { db: "person", collection: "employees" }, actions: [ "find", "insert" ] } ] );
uncaught exception: SyntaxError: missing } after property list :
@shell:1:93
> db.grantPrivilegesToRole(
...   "customRole",
...   [
...     {
...       resource: { db: "person", collection: "employees" },
...       actions: [ "find", "insert" ]
...     }
...   ]
... );
> db.getRoles();
[
  {
    "_id": "person.customRole",
    "role": "customRole",
    "db": "person",
    "roles": [
      {
        "role": "read",
        "db": "person"
      }
    ],
    "isBuiltin": false,
    "inheritedRoles": [
      {
        "role": "read",
        "db": "person"
      }
    ]
  }
]
> db.grantPrivilegesToRole( "customRole", [{ resource: { db: "person", collection: "employees" }, actions: [ "find", "insert" ] } ] );

C:\Windows\System32\cmd.exe x + v
]
}
]
> db.grantPrivilegesToRole( "customRole", [{ resource: { db: "person", collection: "employees" }, actions: [ "find", "insert" ] } ] );
> db.getRoles();
[
  {
    "_id": "person.customRole",
    "role": "customRole",
    "db": "person",
    "roles": [
      {
        "role": "read",
        "db": "person"
      }
    ],
    "isBuiltin": false,
    "inheritedRoles": [
      {
        "role": "read",
        "db": "person"
      }
    ]
  }
]
> db.revokePrivilegesToRole( "customRole", [{ resource: { db: "person", collection: "employees" }, actions: [ "find", "insert" ] } ] );
uncaught exception: TypeError: db.revokePrivilegesToRole is not a function :
@shell:1:1
> db.revokePrivilegesFromRole( "customRole", [{ resource: { db: "person", collection: "employees" }, actions: [ "find", "insert" ] } ] );
> db.getRoles();
[
  {
    "_id": "person.customRole",
    "role": "customRole",
    "db": "person",
    "roles": [
      {
        "role": "read",
        "db": "person"
      }
    ]
  }
]
```

```
C:\Windows\System32\cmd.exe
> db.revokePrivilegesToRole( "customRole", [{ resource: { db: "person", collection: "employees" }, actions: [ "find", "insert" ] } ] );
uncaught exception: TypeError: db.revokePrivilegesToRole is not a function :
@(shell):1:1
> db.revokePrivilegesFromRole( "customRole", [{ resource: { db: "person", collection: "employees" }, actions: [ "find", "insert" ] } ] );
> db.getRoles();
[
  {
    "_id" : "person.customRole",
    "role" : "customRole",
    "db" : "person",
    "roles" : [
      {
        "role" : "read",
        "db" : "person"
      }
    ],
    "isBuiltin" : false,
    "inheritedRoles" : [
      {
        "role" : "read",
        "db" : "person"
      }
    ]
  }
]
> db.grantRoleToRole( "customRole", [{ role: "readWrite", db: "person" }, { role: "dbAdmin", db: "person" } ] );
uncaught exception: TypeError: db.grantRoleToRole is not a function :
@(shell):1:1
> db.grantRolesToRole( "customRole", [{ role: "readWrite", db: "person" }, { role: "dbAdmin", db: "person" } ] );
> db.getRoles();
[
  {
    "_id" : "person.customRole",
    "role" : "customRole",
    "db" : "person",
    "roles" : [
      {
        "role" : "read",
        "db" : "person"
      }
    ],
    "isBuiltin" : false,
    "inheritedRoles" : [
      {
        "role" : "read",
        "db" : "person"
      }
    ]
  }
]
> db.grantRoleToRole( "customRole", [{ role: "readWrite", db: "person" }, { role: "dbAdmin", db: "person" } ] );
uncaught exception: TypeError: db.grantRoleToRole is not a function :
@(shell):1:1
> db.grantRolesToRole( "customRole", [{ role: "readWrite", db: "person" }, { role: "dbAdmin", db: "person" } ] );
> db.getRoles();
[
  {
    "_id" : "person.customRole",
    "role" : "customRole",
    "db" : "person",
    "roles" : [
      {
        "role" : "read",
        "db" : "person"
      },
      {
        "role" : "readWrite",
        "db" : "person"
      },
      {
        "role" : "dbAdmin",
        "db" : "person"
      }
    ],
    "isBuiltin" : false,
    "inheritedRoles" : [
      {
        "role" : "read",
        "db" : "person"
      },
      {
        "role" : "readWrite",
        "db" : "person"
      },
      {
        "role" : "dbAdmin",
        "db" : "person"
      }
    ]
  }
]
```

```
C:\Windows\System32\cmd.exe x + v

{
  "role" : "readWrite",
  "db" : "person"
},
{
  "role" : "dbAdmin",
  "db" : "person"
}
]
db.updateRoles("customRoles", { privileges: [ { resource: { db: "person", collection: "employees" }, action: [ "find", "insert" ] } ], roles: [ { role: "read", db: "person" } ] });
... .updateRoles("customRoles", { privileges: [ { resource: { db: "person", collection: "employees" }, action: [ "find", "insert" ] } ], roles: [ { role: "read", db: "person" } ] });
... .updateRoles("customRoles", { privileges: [ { resource: { db: "person", collection: "employees" }, action: [ "find", "insert" ] } ], roles: [ { role: "read", db: "person" } ] });
... "read", db: "person" } ] });
uncaught exception: TypeError: db.updateRoles is not a function :
@(shell):1:1
> db.updateRoles("customRoles", { privileges: [ { resource: { db: "person", collection: "employees" }, action: [ "find", "insert" ] } ], roles: [ { role: "read", db: "person" } ] });
uncaught exception: TypeError: db.updateRoles is not a function :
@(shell):1:1
> db.updateRole(
... "customRole",
... {
...   privileges: [
...     {
...       resource: { db: "person", collection: "employees" },
...       actions: ["find", "insert"]
...     }
...   ],
...   roles: [
...     { role: "read", db: "person" }
...   ]
... }
... );
> db.getRoles();
[
  {
    "_id" : "person.customRole",
    "role" : "customRole",
    "db" : "person",
    "roles" : [
      {
        "role" : "read",
        "db" : "person"
      }
    ]
  }
]

C:\Windows\System32\cmd.exe x + v

@(shell):1:1
> db.updateRoles("customRoles", { privileges: [ { resource: { db: "person", collection: "employees" }, action: [ "find", "insert" ] } ], roles: [ { role: "read", db: "person" } ] });
... "read", db: "person" } ] });
uncaught exception: TypeError: db.updateRoles is not a function :
@(shell):1:1
> db.updateRole(
... "customRole",
... {
...   privileges: [
...     {
...       resource: { db: "person", collection: "employees" },
...       actions: ["find", "insert"]
...     }
...   ],
...   roles: [
...     { role: "read", db: "person" }
...   ]
... }
... );
> db.getRoles();
[
  {
    "_id" : "person.customRole",
    "role" : "customRole",
    "db" : "person",
    "roles" : [
      {
        "role" : "read",
        "db" : "person"
      }
    ]
  },
  {
    "isBuiltin" : false,
    "inheritedRoles" : [
      {
        "role" : "read",
        "db" : "person"
      }
    ]
  }
]
]
> |
```