

Team Top Gun group project MongoDB database test plan

Identifier	Description
section 1	create 3 users with three roles in MongoDB, according to requirements
section 2	using each user's credentials make sure they can perform only CRUD operations they are authorised to

Test case 1

Creating 3 users with appropriate user roles

```
#before creating test users and database we need to enable access control
use admin ##relocated to admin database

#Creating the user administrator (in the admin authentication database)
db.createUser(
  {
    user: "myAdmin",
    pwd: "abc123",
    roles: [ { role: "userAdminAnyDatabase", db: "admin" } ]
  }
)
Successfully added user: {
  "user" : "myAdmin",
  "roles" : [
    {
      "role" : "userAdminAnyDatabase",
      "db" : "admin"
    }
  ]
}
db.auth("myAdmin", "abc123") #authenticated user
1

exit
bye

sudo vim /etc/mongod.conf #edit security field on unix_like systems
#uncommented those lines:
security:
  authorization: 'enabled'
sudo systemctl restart mongod #on unix-like OS
systemctl status mongod
• mongod.service - MongoDB Database Server
  Loaded: loaded (/lib/systemd/system/mongod.service; enabled; vendor preset:
enabled)
  Active: active (running) since Tue 2022-09-13 13:59:04 NZST; 3s ago
  Docs: https://docs.mongodb.org/manual
```

```

Main PID: 274984 (mongod)
Memory: 158.2M
CPU: 372ms
CGroup: /system.slice/mongod.service
└─274984 /usr/bin/mongod --config /etc/mongod.conf

#Connect and authenticate as the user administrator
mongo --port 27017 -u "myAdmin" -p "abc123" --authenticationDatabase "admin"
MongoDB shell version v4.4.16
connecting to: mongodb://127.0.0.1:27017/?
authSource=admin&compressors=disabled&gssapiServiceName=mongod
Implicit session: session { "id" : UUID("39d8742c-535f-4252-8d08-abe988e76c07") }
MongoDB server version: 4.4.16

use discovery ##relocated to target database
switched to db discovery

###our wiki articles collection
db.createCollection("articles")
{ "ok" : 1 }

##first user with read only permission
> db.createUser(
{  user: "Mohan",
  pwd: "password",
  roles:[{role: "read" , db:"discovery"}]})
Successfully added user: {
  "user" : "Mohan",
  "roles" : [
    {
      "role" : "read",
      "db" : "discovery"
    }
  ]
}
}
#output:
Successfully added user: {
  "user" : "Mohan",
  "roles" : [
    {
      "role" : "read",
      "db" : "discovery"
    }
  ]
}
}

##creating new user role, where user can only create or view articles but not
delete them
db.createRole({
role : 'write_read_only',
privileges : [
{resource : {db : "discovery", collection : "articles"}, actions : ["insert"]}
],
roles : [ {role : "read", db: "discovery"}]
}

```

```

}))

#output (custom role created!!):
{
  "role" : "write_read_only",
  "privileges" : [
    {
      "resource" : {
        "db" : "discovery",
        "collection" : "articles"
      },
      "actions" : [
        "insert"
      ]
    }
  ],
  "roles" : [
    {
      "role" : "read",
      "db" : "discovery"
    }
  ]
}

##creating second user with write_read_only rights
db.createUser(
{  user: "Sara",
  pwd: "password",
  roles:[{role: "write_read_only" , db:"discovery"}]})

Successfully added user: {
  "user" : "Sara",
  "roles" : [
    {
      "role" : "write_read_only",
      "db" : "discovery"
    }
  ]
}

##creating third user with read and delete permissions

db.createUser(
{  user: "Michel",
  pwd: "password",
  roles:[{role: "readWrite" , db:"discovery"}]})

#output:
Successfully added user: {
  "user" : "Michel",
  "roles" : [
    {
      "role" : "readWrite",
      "db" : "discovery"
    }
  ]
}

```

```

    ]
}

###displaying created users
show users

```

Test case 2

using each user's credentials make sure they can perform only CRUD operations they are authorised to

user 1 (read, write, update, delete permissions)

```

exit
bue
##logging in as user with read_wright_delete permissions
mongo --port 27017 -u "Michel" -p "password" --authenticationDatabase "discovery"
MongoDB shell version v4.4.16
connecting to: mongodb://127.0.0.1:27017/?
authSource=discovery&compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("9760f7ff-7de6-477f-be26-f2f1a54d7d18") }
MongoDB server version: 4.4.16

##inserting test data with above user permissions.
db.articles.insertOne(
  { about: "Lorem ipsum dolor sit amet, consectetur adipiscing", date_born:
    "1840", category: "Arts", notable_work: "Water Lilies"}
)
#output as expected:
{
  "acknowledged" : true,
  "insertedId" : ObjectId("631fe7bd88a0322a0182dfa6")
}

##reading created data by one parameter
db.articles.find({ "notable_work": "Water Lilies" })
#output as expected:
{ "_id" : ObjectId("631fe7bd88a0322a0182dfa6"), "about" : "Lorem ipsum dolor sit
amet, consectetur adipiscing", "date_born" : "1840", "category" : "Arts",
"notable_work" : "Water Lilies" }

##updating 'about' field
db.articles.update({_id: ObjectId("631fe7bd88a0322a0182dfa6")}, { $set:      {
  'about': 'Updated text goes here',      } })
#output as expected:
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

##displaying updated data
db.articles.find({ "notable_work": "Water Lilies" })
#output as expected:

```

```

{ "_id" : ObjectId("631fe7bd88a0322a0182dfa6"), "about" : "Updated text goes here", "date_born" : "1840", "category" : "Arts", "notable_work" : "Water Lilies"
}

#deleting test document
db.articles.deleteOne( { _id: ObjectId("631fe7bd88a0322a0182dfa6") } )
{ acknowledged: true, deletedCount: 1 }

```

conclusion: user1 permissions read, write, update, delete work correctly

user 2 (read, write only permissions)

```

##logging in as user with read_write_only permissions
mongo --port 27017 -u Sara -p password --authenticationDatabase discovery
MongoDB shell version v4.4.16
connecting to: mongodb://127.0.0.1:27017/?
authSource=discovery&compressors=disabled&gssapiServiceName=mongod
Implicit session: session { "id" : UUID("d0d56175-df84-42bb-8564-6520b0822a4a") }
MongoDB server version: 4.4.16
use discovery
switched to db discovery

##inserting test data with above user permissions
db.articles.insertOne(
  { about: "Lorem ipsum dolor sit amet, consectetur adipiscing", date_born:
"1840", category: "Arts", notable_work: "Water Lilies"}
)
{
  "acknowledged" : true,
  "insertedId" : ObjectId("631feb00b3ee9be0173236a9")
}

##reading created data by one parameter
db.articles.find({ "notable_work": "Water Lilies" })
{ "_id" : ObjectId("631feb00b3ee9be0173236a9"), "about" : "Lorem ipsum dolor sit amet, consectetur adipiscing", "date_born" : "1840", "category" : "Arts", "notable_work" : "Water Lilies" }

##trying to update 'about' field
db.articles.update({_id: ObjectId("631eab7099b194ecc836ee5d")}, { $set:
  {
    'about': 'Updated text goes here',
  }
})

#output as expected (data has not been updated):
WriteCommandError({
  "ok" : 0,
  "errmsg" : "not authorized on discovery to execute command { update:
\"articles\", ordered: true, lsid: { id: UUID(\"d0d56175-df84-42bb-8564-6520b0822a4a\") }, $db: \"discovery\" }",
  "code" : 13,

```

```

        "codeName" : "Unauthorized"
    })

    #trying to delete test document
    db.articles.deleteOne({ _id: ObjectId("631feb00b3ee9be0173236a9")})
    #output as expected (data has not been deleted):
    uncaught exception: WriteCommandError({
      "ok" : 0,
      "errmsg" : "not authorized on discovery to execute command { delete:
        \"articles\", ordered: true, lsid: { id: UUID(\"d0d56175-df84-42bb-8564-
        6520b0822a4a\") }, $db: \"discovery\" }",
      "code" : 13,
      "codeName" : "Unauthorized"
    }) :
    WriteCommandError({
      "ok" : 0,
      "errmsg" : "not authorized on discovery to execute command { delete:
        \"articles\", ordered: true, lsid: { id: UUID(\"d0d56175-df84-42bb-8564-
        6520b0822a4a\") }, $db: \"discovery\" }",
      "code" : 13,
      "codeName" : "Unauthorized"
    })
    #nothing has been deleted as this user doesnt have this permission

```

conclusion: user2 permissions read, write only work correctly

user 3 (read only permissions)

```

> exit
bye
##logging in as user with read only permissions
mongo --port 27017 -u Mohan -p password --authenticationDatabase discovery
MongoDB shell version v4.4.16
connecting to: mongodb://127.0.0.1:27017/?
authSource=discovery&compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("2d6ef3e5-c0ab-4716-933c-85802adfc631") }
MongoDB server version: 4.4.16
> use discovery
switched to db discovery

#reading from 'articles' collection
> db.articles.find()

#output as expected
{ "_id" : ObjectId("631feb00b3ee9be0173236a9"), "about" : "Lorem ipsum dolor sit
amet, consectetur adipiscing", "date_born" : "1840", "category" : "Arts",
"notable_work" : "Water Lilies" }

#trying to insert into "articles"
db.articles.insertOne(

```

```

    { about: "Lorem ipsum dolor sit amet, consectetur adipiscing", date_born:
      "1840", category: "Arts", notable_work: "Water Lilies"}
  )

#output as expected (this user cannot create documents)
uncaught exception: WriteCommandError({
  "ok" : 0,
  "errmsg" : "not authorized on discovery to execute command { insert:
    \"articles\", ordered: true, lsid: { id: UUID(\"2d6ef3e5-c0ab-4716-933c-
    85802adfc631\") }, $db: \"discovery\" }",
  "code" : 13,
  "codeName" : "Unauthorized"
}) :
WriteCommandError({
  "ok" : 0,
  "errmsg" : "not authorized on discovery to execute command { insert:
    \"articles\", ordered: true, lsid: { id: UUID(\"2d6ef3e5-c0ab-4716-933c-
    85802adfc631\") }, $db: \"discovery\" }",
  "code" : 13,
  "codeName" : "Unauthorized"
})

#trying to delete document
db.articles.deleteOne({ _id: ObjectId("631feb00b3ee9be0173236a9")})

#output as expected (this user cannot delete documents)
uncaught exception: WriteCommandError({
  "ok" : 0,
  "errmsg" : "not authorized on discovery to execute command { delete:
    \"articles\", ordered: true, lsid: { id: UUID(\"2d6ef3e5-c0ab-4716-933c-
    85802adfc631\") }, $db: \"discovery\" }",
  "code" : 13,
  "codeName" : "Unauthorized"
}) :
WriteCommandError({
  "ok" : 0,
  "errmsg" : "not authorized on discovery to execute command { delete:
    \"articles\", ordered: true, lsid: { id: UUID(\"2d6ef3e5-c0ab-4716-933c-
    85802adfc631\") }, $db: \"discovery\" }",
  "code" : 13,
  "codeName" : "Unauthorized"
})

#trying to update document
db.articles.update({_id: ObjectId("631eab7099b194ecc836ee5d")}, { $set:
...   {
...   'about': 'Updated text goes here',
...   }
... })

#output as expected (no permission)
WriteCommandError({
  "ok" : 0,

```

```
"errmsg" : "not authorized on discovery to execute command { update:  
\"articles\", ordered: true, lsid: { id: UUID(\"2d6ef3e5-c0ab-4716-933c-  
85802adfc631\") }, $db: \"discovery\" }",  
  "code" : 13,  
  "codeName" : "Unauthorized"  
})
```

conclusion: user3 permissions read only work correctly