

Методы сбора, хранения, обработки и анализа данных

Лекция 17

СУБД MongoDB – безопасность

Обеспечение безопасности

- Аутентификация и авторизация
 - Логины
 - Пользователи
 - Роли
 - Привилегии
 - Объекты и субъекты защиты
- Шифрование
- Аудит

Аутентификация

- Создание пользователей
- Наделение пользователей привилегиями
- Переподключение сервера в режиме аутентификации

Режимы аутентификации

- Role-Based Access Control – для базы данных (user / password)
- Механизмы SCRAM – Salted Challenge Response Authentication Mechanism
- LDAP – Lightweight Directory Access Protocol
- Kerberos
- Microsoft Active Directory Authentication
- x.509 Certificate

Роли и привилегии

- ~ 100 привилегий
- Могут объединяться в пользовательские роли
- Есть встроенные роли
- Объект для назначения привилегий:
 - Коллекция
 - База данных
 - Кластер

Встроенные роли

read	доступ в режиме чтения
readWrite	чтение и редактирование данных
dbAdmin	выполнение административных задач, связанных со схемой, индексирование и сбор статистики
userAdmin	создание и изменение ролей и пользователей в текущей базе данных
dbOwner	чтение и редактирование данных, добавление и удаление пользователей, выдача прав пользователям и создание ролей
dbAdminAnyDatabase	создание индексов, вызов процедур сжатия данных, выполнение задач, связанных со схемой, невозможность создания и удаления пользователей
root	суперпользователь, но с ограничениями; может выполнять большинство действий, но не все: не может изменить системную коллекцию

Создание пользователей

- Создать пользователя
- Указать базу данных
- Указать роль

```
> use all_subjects
```

```
> db.createUser({  
  user: "Db_Subject_Admin",  
  pwd: "Pa$$w0rd",  
  roles: ["dbOwner"]  
})  
< { ok: 1 }
```

```
> db.createUser({  
  user: "Db_Subject_User",  
  pwd: "Pa$$w0rd",  
  roles: ["readWrite"]  
})  
< { ok: 1 }
```

Подключение в режиме аутентификации

C:\WINDOWS\system32\cmd.exe - C:\MongoDb\bin\mongod -auth

Microsoft Windows [Version 10.0.18363.1556]

(c) Корпорация Майкрософт (Microsoft Corporation), 2019. Все права защищены.

C:\Users\User>C:\MongoDb\bin\mongod -auth

```
{ "t": { "$date": "2025-03-31T17:22:12.101+03:00" }, "s": "I", "c": "CONTROL", "id": 23285, "ctx": "thread1", "msg": "A  
lly disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none' }
```

```
{ "t": { "$date": "2025-03-31T17:22:12.102+03:00" }, "s": "I", "c": "NETWORK", "id": 4915701, "ctx": "thread1", "msg": "I  
d wire specification", "attr": { "spec": { "incomingExternalClient": { "minWireVersion": 0, "maxWireVersion": 17 }, "incomi  
lClient": { "minWireVersion": 0, "maxWireVersion": 17 }, "outgoing": { "minWireVersion": 6, "maxWireVersion": 17 }, "isIntern  
: true } } }
```

```
{ "t": { "$date": "2025-03-31T17:22:12.105+03:00" }, "s": "I", "c": "NETWORK", "id": 4648602, "ctx": "thread1", "msg": "I  
CP FastOpen in use." }
```

```
{ "t": { "$date": "2025-03-31T17:22:12.107+03:00" }, "s": "I", "c": "REPL", "id": 5123008, "ctx": "thread1", "msg": "S  
ly registered PrimaryOnlyService", "attr": { "service": "TenantMigrationDonorService", "namespace": "config.tenantMig  
ors" } }
```

```
{ "t": { "$date": "2025-03-31T17:22:12.107+03:00" }, "s": "I", "c": "REPL", "id": 5123008, "ctx": "thread1", "msg": "S
```


Подключение

☆ Favorites

Db_Subject_Admin
Never

🔄 Recents

localhost:27017
31 мар. 2025 г., 17:10

localhost:27017
31 мар. 2025 г., 17:07

localhost:27017
17 мар. 2025 г., 17:14

localhost:27017
15 мая 2023 г., 13:23

localhost:27017
15 мая 2023 г., 13:18

localhost:27017
5 июл. 2022 г., 16:41

New Connection

Connect to a MongoDB deployment



FAVORITE

URI ⓘ

Edit Connection String ☒

```
mongodb://Db_Subject_Admin:Pa%24%24w0rd@localhost:27017/?  
authMechanism=DEFAULT&authSource=all_subjects
```

▼ Advanced Connection Options

General

Authentication

TLS/SSL

Proxy/SSH

In-Use Encryption

Advanced

Authentication Method

None

Username/Password

X.509

Kerberos

LDAP

AWS IAM

Username

Db_Subject_Admin

Password

Authentication Database ⓘ

all_subjects

Optional

Authentication Mechanism

Default

SCRAM-SHA-1

SCRAM-SHA-256

Подключение

- Пользователь может просматривать и изменять данные
- Может создать пользователя и дать ему права

```
> db.places.find().count()  
< 70  
> db.createUser({  
    user: "New_User",  
    pwd: "Pa$$w0rd",  
    roles: ["userAdmin"]  
})  
< { ok: 1 }  
all_subjects> |
```

Подключение

New Connection +

☆ Favorites

Db_Subject_Admin

31 мар. 2025 г., 17:25

Db_Subject_User

31 мар. 2025 г., 17:31

...

🔄 Recents

localhost:27017

31 мар. 2025 г., 17:10

localhost:27017

31 мар. 2025 г., 17:07

localhost:27017

17 мар. 2025 г., 17:14

localhost:27017

15 мая 2023 г., 13:23

localhost:27017

15 мая 2023 г., 13:18

localhost:27017

5 июл. 2022 г., 16:41

Db_Subject_User

Connect to a MongoDB deployment

☆ FAVORITE

URI ⓘ

Edit Connection String ☐

mongodb://localhost:27017/

▼ Advanced Connection Options

General

Authentication

TLS/SSL

Proxy/SSH

In-Use Encryption

Advanced

Connection String Scheme

mongodb

mongodb+srv

Standard Connection String Format. The standard format of the MongoDB connection URI is used to connect to a MongoDB deployment: standalone, replica set, or a sharded cluster.

Host

localhost:27017

+

☐ Direct Connection

Specifies whether to force dispatch all operations to the specified host.

Save

Connect

Подключение

- Пользователь может работать в рамках предоставленных прав

```
> _MONGOSH

> use all_subjects
< 'switched to db all_subjects'
> db.places.find().count()
< 70
> db.createUser({
  user: "Little_User",
  pwd: "Pa$$w0rd",
  roles: ["read"]
})

✖ Error: clone(t={}){const r=t.loc||{};return e({loc:new Po
  at Object.serialize (v8.js:256:7)
  at u (C:\MongoDb\Distr\Compass\resources\app.asar.unp
  at postMessage (C:\MongoDb\Distr\Compass\resources\ap
  at i (C:\MongoDb\Distr\Compass\resources\app.asar.unp
```

Подключение

- Пользователь может работать в базе данных, в которой создан

```
> use admin
< 'switched to db admin'
> db.getCollectionNames()
```

```
✖ ▾ MongoServerError: not authorized on admin to execute command { listCollections: 1, fil
  at Connection.onMessage (C:\MongoDb\Distr\Compass\resources\app.asar.unpacked\node
  at MessageStream.<anonymous> (C:\MongoDb\Distr\Compass\resources\app.asar.unpacked
  at MessageStream.emit (events.js:315:20)
```

```
> use all_subjects
< 'switched to db all_subjects'
> db.getCollectionNames()
< [ 'places', 'reviews', 'areviews', 'locations' ]
all_subjects> |
```

Пользователи

- Создание пользователя
- Просмотр всех пользователей и их ролей
- Изменение пользователя
- Изменение пароля
- Удаление пользователя

Создание пользователя

```
> db.createUser({  
  user: "Local_User",  
  pwd: passwordPrompt(),  
  roles: ["readWrite"]  
})
```

Enter password:

Просмотр всех пользователей

```
> db.getUsers()
< {
  users: [
    {
      _id: 'all_subjects.Db_Subject_Admin',
      userId: UUID("21e078d7-52ea-416c-bd2b-2b453d29d88a"),
      user: 'Db_Subject_Admin',
      db: 'all_subjects',
      roles: [Array],
      mechanisms: [Array]
    },
    {
      _id: 'all_subjects.Db_Subject_User',
      userId: UUID("8b85ca85-3683-491c-900c-e4f8ab0a567a"),
      user: 'Db_Subject_User',
      db: 'all_subjects',
      roles: [Array],
      mechanisms: [Array]
    },
    {
      _id: 'all_subjects.Local_User',
      userId: UUID("d6ee826f-a20e-457c-9a01-fe4d8ea8edbe"),
      user: 'Local_User',
      db: 'all_subjects',
      roles: [Array],
      mechanisms: [Array]
    },
    {
      _id: 'all_subjects.New_User',
      userId: UUID("20fdbbc36-0f36-4195-afca-ee2013d6d150"),
      user: 'New_User',
```


Просмотр пользователя

```
> db.getUser("New_User")
< {
  _id: 'all_subjects.New_User',
  userId: UUID("20fdb36-0f36-4195-afca-ee2013d6d150"),
  user: 'New_User',
  db: 'all_subjects',
  roles: [ { role: 'userAdmin', db: 'all_subjects' } ],
  mechanisms: [ 'SCRAM-SHA-1', 'SCRAM-SHA-256' ]
}
```

```
> db.getUser("New_User", {
  showCredentials: true,
  showPrivileges: true,
  showAuthenticationRestrictions: true
})
< {
  _id: 'all_subjects.New_User',
  userId: UUID("20fdb36-0f36-4195-afca-ee2013d6d150"),
  user: 'New_User',
  db: 'all_subjects',
  mechanisms: [ 'SCRAM-SHA-1', 'SCRAM-SHA-256' ],
  credentials: {
    'SCRAM-SHA-1': {
      iterationCount: 10000,
      salt: 'FxxJicGh1J8sGKI0ZBL/PQA==',
      storedKey: 'OLWIRj8fbyBD5IWrXOSYBbEVwP0=',
      serverKey: 'CuweJvDjiSCaOfFeSTNe421+KumU='
    },
    'SCRAM-SHA-256': {
      iterationCount: 15000,
      salt: '8S5xHEXR+t7QCSJ9bkI7PEW61XC+GjhhT7IxAg==',
      storedKey: '/MzQRitfXhw8qkhuB/bgWHM5Kawdkmwzd5pQ9yAwPj4=',
      serverKey: 'b7GK+nEQJ4mPVdUna1pcWDY+IeJ8Rgy3x8zSywn8vKk='
    }
  },
  roles: [ { role: 'userAdmin', db: 'all_subjects' } ],
  authenticationRestrictions: [],
  inheritedRoles: [ { role: 'userAdmin', db: 'all_subjects' } ],
  inheritedPrivileges: [ { resource: [Object], actions: [Array] } ],
  inheritedAuthenticationRestrictions: []
}
```

Изменение пользователя

- `db.updateUser()`
- `db.ChangeUserPassword()`
- `db.grantRolesToUser()`
- `db.revokeRolesFromUser()`

Изменение пароля

```
> db.changeUserPassword("New_User", "NewPa$$w0rd")  
< { ok: 1 }
```

```
> db.getUser("New_User", {  
  showCredentials: true,  
  showPrivileges: true,  
  showAuthenticationRestrictions: true  
})  
< {  
  _id: 'all_subjects.New_User',  
  userId: UUID("20fdbbc36-0f36-4195-afca-ee2013d6d150"),  
  user: 'New_User',  
  db: 'all_subjects',  
  mechanisms: [ 'SCRAM-SHA-1', 'SCRAM-SHA-256' ],  
  credentials: {  
    'SCRAM-SHA-1': {  
      iterationCount: 10000,  
      salt: 'h4rKoQOGF7oeOTjZc9CoVg==',  
      storedKey: '5zVSEL/JqsOsJX41RvH4L38B+b4=',  
      serverKey: '/QonsH3bIAPQyoshNfb/V5o3A6Y='  
    },  
    'SCRAM-SHA-256': {  
      iterationCount: 15000,  
      salt: 'RQypoG26oL4fgBt5cwSZYr1AAoRyTHkyuSteCg==',  
      storedKey: '9X4Q+vKfyJUghdlyuGaCIwktYYQaMFxwQZrGeYd8Ow0=',  
      serverKey: 'wml67qzZ3O1SZt59aVLfgreInC1g8wTW7DViGqVoYAM='  
    }  
  },  
  roles: [ { role: 'userAdmin', db: 'all_subjects' } ],  
  authenticationRestrictions: [],  
  inheritedRoles: [ { role: 'userAdmin', db: 'all_subjects' } ],  
  inheritedPrivileges: [ { resource: [Object], actions: [Array] } ],  
  inheritedAuthenticationRestrictions: []  
}
```

Удаление пользователя

```
> db.dropUser("New_User")
< { ok: 1 }
> db.getUsers()
< {
  users: [
    {
      _id: 'all_subjects.Db_Subject_Admin',
      userId: UUID("21e078d7-52ea-416c-bd2b-2b453d29d88a"),
      user: 'Db_Subject_Admin',
      db: 'all_subjects',
      roles: [Array],
      mechanisms: [Array]
    },
    {
      _id: 'all_subjects.Db_Subject_User',
      userId: UUID("8b85ca85-3683-491c-900c-e4f8ab0a567a"),
      user: 'Db_Subject_User',
      db: 'all_subjects',
      roles: [Array],
      mechanisms: [Array]
    },
    {
      _id: 'all_subjects.Local_User',
      userId: UUID("d6ee826f-a20e-457c-9a01-fe4d8ea8edbe"),
      user: 'Local_User',
      db: 'all_subjects',
      roles: [Array],
      mechanisms: [Array]
    }
  ]
}
```

Пользовательские роли

- Можно создать свои роли
- На основе ролей
- На основе привилегий

```
> db.createRole({
  role: "Review_Finder",
  privileges: [
    {
      resource: { db: "all_subjects", collection: "reviews" },
      actions: [ "find" ]
    }
  ],
  roles: [],
})
< { ok: 1 }
```

```
> db.createRole({
  role: "Places_Finder",
  privileges: [],
  roles: [
    {
      role: "read",
      db: "all_subjects"
    }
  ],
})
< { ok: 1 }
```

Пользовательские роли – просмотр

```
> db.getRoles()
< {
  roles: [
    {
      _id: 'all_subjects.Places_Finder',
      role: 'Places_Finder',
      db: 'all_subjects',
      roles: [Array],
      isBuiltin: false,
      inheritedRoles: [Array]
    },
    {
      _id: 'all_subjects.Review_Finder',
      role: 'Review_Finder',
      db: 'all_subjects',
      roles: [],
      isBuiltin: false,
      inheritedRoles: []
    }
  ],
  ok: 1
}
```

Пользовательские роли – просмотр

```
> db.getRole("Places_Finder")
< {
  _id: 'all_subjects.Places_Finder',
  role: 'Places_Finder',
  db: 'all_subjects',
  roles: [ { role: 'read', db: 'all_subjects' } ],
  inheritedRoles: [ { role: 'read', db: 'all_subjects' } ],
  isBuiltin: false
}
all_subjects> |
```

Пользовательские роли

- Изменение – `db.updateRole()`
- Удаление – `db.dropRole()`
- Добавление привилегий – `db.grantPrivilegesToRole()`
- Отзыв привилегий – `db.revokePrivilegesFromRole()`
- Добавление роли – `db.grantRolesToRole()`
- Удаление роли из роли – `db.revokeRolesFromRole()`
- Назначение роли пользователю – `db.grantRolesToUser()`
- Отбор роли у пользователя – `db.revokeRolesFromUser()`

Аудит

- Необходимо запускать сервер с ключами
 - `--auditDestination file`
 - `--auditFormat`
 - `--auditPath`
- В файле конфигурации необходимо указать события, требующие аудита

Вопросы?