

Asignatura	Datos del alumno	Fecha
Bases de Datos para el Big Data	Apellidos:	
	Nombre:	

Taller: Usuarios en MongoDB

Este taller te guiará a través de la configuración inicial de MongoDB, la creación de un usuario administrador, y luego la creación y configuración de usuarios específicos con roles personalizados. También veremos cómo realizar estas configuraciones mediante el fichero de configuración de MongoDB.

Paso 0: Realizar una Copia de Seguridad de la Base de Datos

Antes de hacer cualquier cambio significativo, como configurar la autenticación, es una buena práctica realizar una copia de seguridad de tu base de datos. Esto te asegurará de tener una manera de restaurar el estado anterior en caso de que algo salga mal.

Detener el servicio de MongoDB:

Es recomendable detener MongoDB para garantizar la consistencia de la copia de seguridad. Esto se puede hacer con el comando `sudo systemctl stop mongod` en sistemas que usan `systemd`.

Realizar la copia de seguridad:

Utiliza el comando `mongodump` para realizar una copia de seguridad de todas tus bases de datos o especifica una en particular. Para copiar todas las bases de datos a un directorio backup en tu directorio actual, puedes usar:

```
mongodump --out ./backup/
```

Reiniciar el servicio de MongoDB:

Asignatura	Datos del alumno	Fecha
Bases de Datos para el Big Data	Apellidos:	
	Nombre:	

Una vez completada la copia de seguridad, reinicia el servicio de MongoDB con `sudo systemctl start mongod`.

Paso 1: Configurar MongoDB para Autenticación

Primero, necesitas asegurarte de que MongoDB esté configurado para utilizar autenticación. Esto se hace modificando el fichero de configuración de MongoDB (*mongod.cfg*), comúnmente ubicado en `/etc/mongod.cfg` o en el directorio de instalación de MongoDB.

Abrir el fichero de configuración de MongoDB:

Utiliza tu editor preferido para abrir el fichero *mongod.cfg*, por ejemplo, `nano /etc/mongod.cfg`.

Modificar el fichero de configuración para habilitar la autenticación:

Busca la sección `#security:` y modifícala para incluir la opción `authorization: enabled` de la siguiente manera:

```
security:
  authorization: enabled
```

Reiniciar el servicio de MongoDB:

Después de guardar los cambios, reinicia el servicio de MongoDB para aplicar la configuración. Puedes hacerlo con un comando como `sudo systemctl restart mongod` en sistemas basados en `systemd`.

Asignatura	Datos del alumno	Fecha
Bases de Datos para el Big Data	Apellidos:	
	Nombre:	

Paso 2: Crear un Usuario Administrador

Una vez habilitada la autenticación, el siguiente paso es crear un usuario administrador que tenga permisos para gestionar otros usuarios y roles.

Conectarse a la instancia de MongoDB:

Ejecuta mongo en la terminal para conectarte a tu instancia local de MongoDB.

Seleccionar la base de datos de administración:

```
use admin
```

Crear el usuario administrador:

Ejecuta el siguiente comando para crear un nuevo usuario con el rol de userAdminAnyDatabase:

```
db.createUser({
  user: "admin",
  pwd: "passwordSeguro123",
  roles: [{ role: "userAdminAnyDatabase", db: "admin" }]
})
```

Asegúrate de reemplazar "`passwordSeguro123`" con una contraseña fuerte y segura.

Asignatura	Datos del alumno	Fecha
Bases de Datos para el Big Data	Apellidos:	
	Nombre:	

Paso 3: Crear Usuarios con Roles Específicos

Después de establecer un usuario administrador, puedes crear usuarios adicionales con roles específicos según sea necesario.

Autenticarte como el usuario administrador:

```
db.auth("admin", "passwordSeguro123")
```

Crear un nuevo usuario con roles personalizados:

Por ejemplo, para crear un usuario que solo pueda leer la base de datos miBaseDeDatos, puedes usar:

```
use miBaseDeDatos
db.createUser({
  user: "usuarioSoloLectura",
  pwd: "otraContraseña123",
  roles: [{ role: "read", db: "miBaseDeDatos" }]
})
```

Paso 4: Validación y Pruebas

Finalmente, es importante validar que los usuarios y roles funcionen como se espera.

Desconectarse del usuario actual (si es necesario):

Puedes hacer esto ejecutando `db.logout()`.

Autenticarse como el nuevo usuario:

Intenta conectarte y realizar operaciones según los permisos otorgados. Por ejemplo:

Asignatura	Datos del alumno	Fecha
Bases de Datos para el Big Data	Apellidos:	
	Nombre:	

```
db.auth("usuarioSoloLectura", "otraContraseña123")
```

Verifica que el usuario solo pueda realizar las operaciones permitidas por su rol.

Paso Final: Probar la Conexión con PyMongo en Jupyter

Después de crear usuarios y configurar roles, es importante verificar que puedes conectarte a tu instancia de MongoDB desde aplicaciones externas, como un notebook de Jupyter usando PyMongo.

Instalar PyMongo:

Si aún no lo has hecho, necesitarás instalar PyMongo en tu entorno de Jupyter. Esto se puede hacer ejecutando el siguiente comando en una celda del notebook:

```
!pip install pymongo
```

Conectarse a MongoDB desde Jupyter:

Utiliza el siguiente código para conectarte a MongoDB usando PyMongo en una celda de Jupyter. Asegúrate de reemplazar usuario, contraseña y nombreDeLaBaseDeDatos con tus propios valores.

```
from pymongo import MongoClient

# Conexión a MongoDB
usuario = "usuarioSoloLectura"
contraseña = "otraContraseña123"
bbdd = "miBaseDeDatos"
conexion =
f"mongodb://{usuario}:{contraseña}@localhost:27017/{bbdd}?auth
Source=admin"

cliente = MongoClient(conexion)
```

Asignatura	Datos del alumno	Fecha
Bases de Datos para el Big Data	Apellidos:	
	Nombre:	

```
# Prueba de conexión y lectura de documentos
db = cliente[bbdd]
colecciones = db.list_collection_names()
print("Colecciones disponibles:", colecciones)

# Asegúrate de reemplazar `nombreColeccion` con el nombre real
# de una colección en tu base de datos
documentos = db['nombreColeccion'].find()
for documento in documentos:
    print(documento)
```

Este código se conecta a tu base de datos MongoDB y lista las colecciones disponibles. También intenta leer documentos de una colección específica, lo que te permite verificar la conexión y los permisos del usuario.

Estos pasos adicionales garantizan la seguridad de tus datos antes de realizar cambios significativos y proporcionan una forma de verificar la funcionalidad de tus configuraciones mediante la conexión a MongoDB desde un entorno de Jupyter Notebook.

MongoDB no permite la creación directa de usuarios y roles a través del fichero de configuración. Sin embargo, puedes utilizar scripts de inicio y ficheros .js para automatizar la creación de usuarios y roles al iniciar la instancia de MongoDB. Por ejemplo, puedes crear un script `init-mongo.js` que contenga comandos `db.createUser()` y luego ejecutar MongoDB con este script como parte del proceso de inicio.

Este taller te proporciona una base para comenzar a trabajar con la autenticación y autorización en MongoDB, permitiéndote explorar más sobre la gestión de usuarios y roles en tus proyectos.