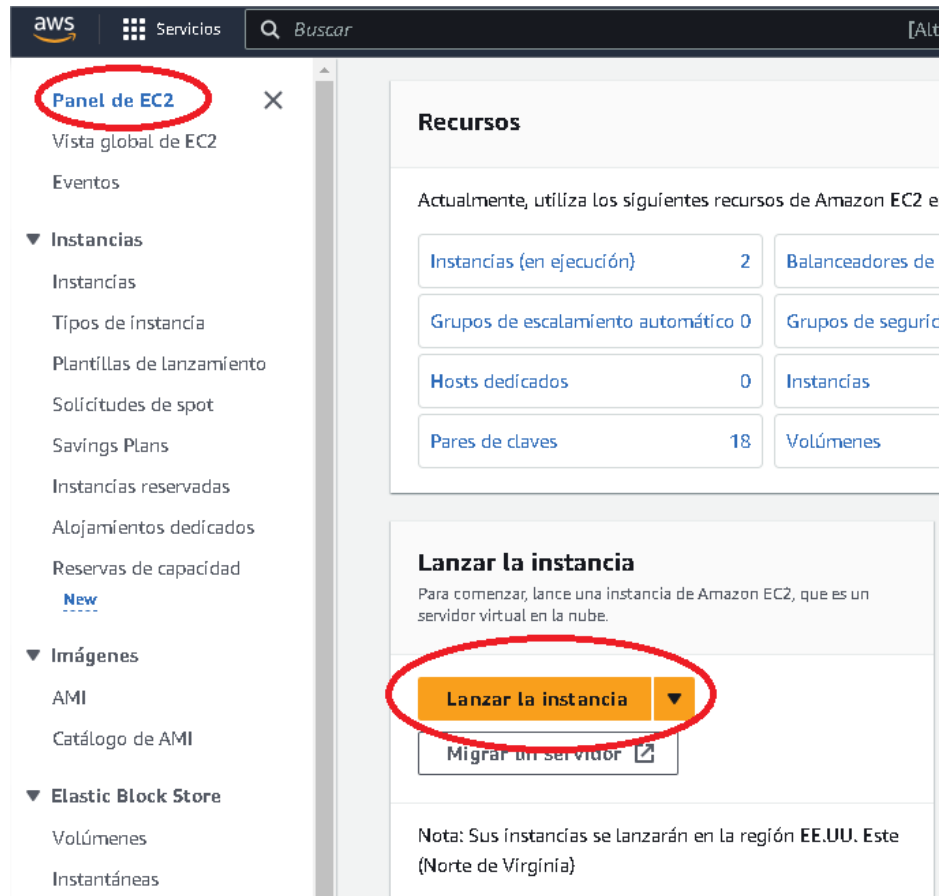


## Ejercicio 8 - Instalación de MongoDB en AWS EC2 y MQL

### Parte 1: Creación de una instancia Ubuntu en AWS EC2

Para crear una instancia EC2, busca EC2 en la consola de AWS y haz clic en el botón naranja **Lanzar la instancia** para abrir el asistente de creación de instancias.



En la página a la que nos lleva, debemos elegir una imagen de Amazon Machine (AMI), que determina el sistema operativo y las aplicaciones que se instalará en la instancia EC2.

Debido a que estamos más familiarizados con ella, elegimos de instancia **Ubuntu 22.04 LTS**.

Más abajo aparecen los detalles sobre la utilización de un par de claves (pública y privada) para conectar mediante SSH con la instancia.

**Crea un nuevo par de claves** para la instancia y asígnales un nombre. Ese par de claves se te descargará en tu máquina local. **OJO! Si en el futuro pierdes las claves podemos hacer un truco para que no pierdas la conexión a la máquina.**

A continuación, tenemos el apartado de **firewall o grupo de seguridad**. Los grupos de seguridad son reglas de red que describen el tipo de tráfico de red permitido en la instancia EC2. Inicialmente, debemos permitir el tráfico entrante SSH. Una vez iniciada la instancia añadiremos también una regla para permitir el tráfico entrante al servidor de MongoDB.

Con esto, ya puedes hacer clic en el botón naranja **Lanzar instancia** para lanzar la instancia EC2.

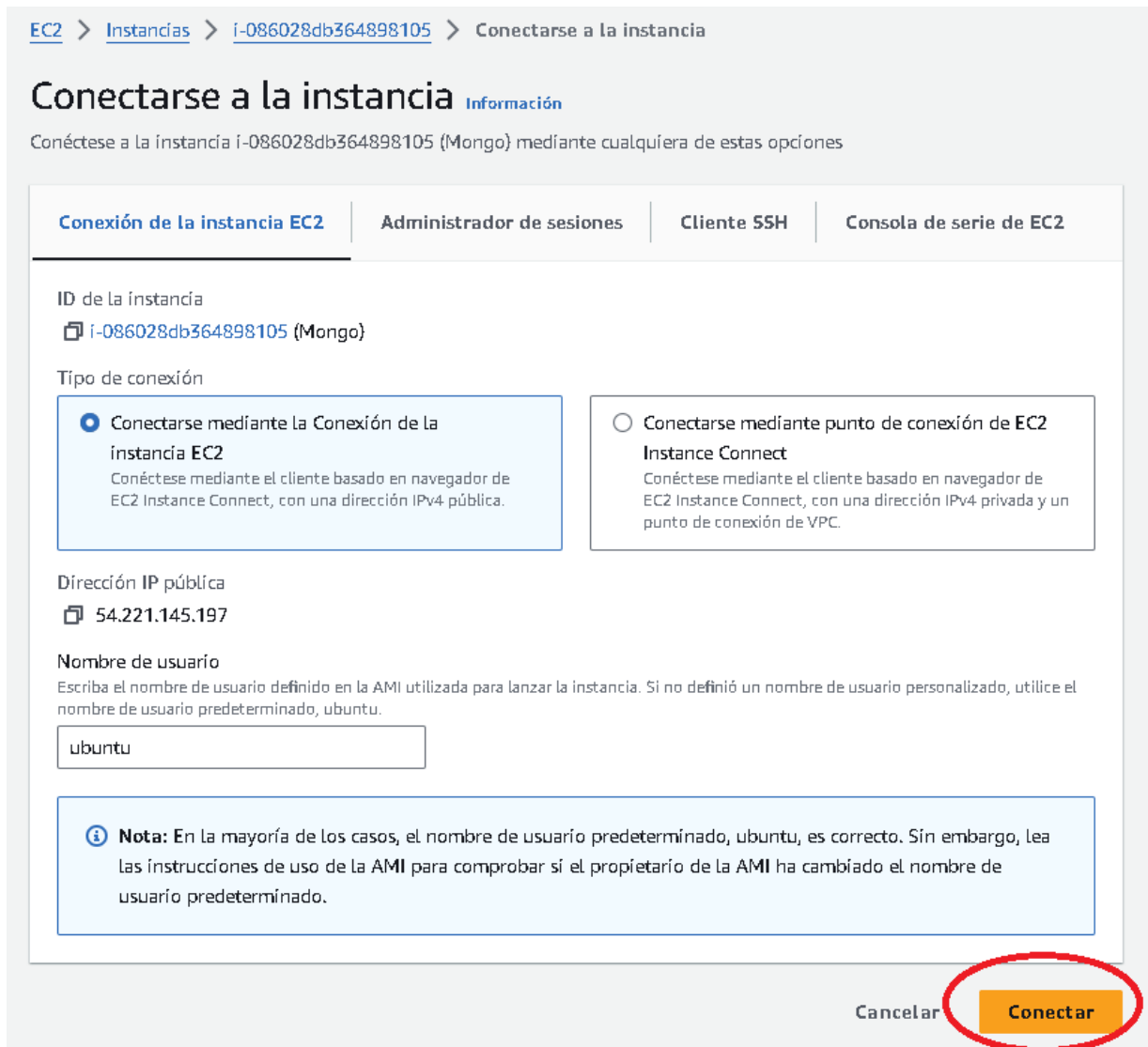
## Parte 2: Conexión SSH por contraseña (sin claves pública y privada)

Aunque menos seguro, es muy cómodo poder conectarte a tu instancia por autenticación basada en contraseña y no necesariamente por clave pública. Para cambiar este comportamiento nos podemos conectarnos desde el navegador a la instancia de la siguiente forma:



The screenshot shows the AWS Management Console interface. The left sidebar contains navigation options like 'Panel de EC2', 'Vista global de EC2', and 'Eventos'. The main content area displays the 'Resumen de instancia de i-086028db364898105 (Mongo)'. The 'Conectar' button is circled in red. The instance details include: ID de la instancia (i-086028db364898105), Dirección IPv4 pública (54.221.145.197), Estado de la instancia (En ejecución), and DNS de IPv4 pública (ec2-54-221-145-197.compute-1.amazonaws.com).

Y en la siguiente pantalla elegir el método de Conexión de la instancia EC2 (por navegador):



The screenshot shows the 'Conectarse a la instancia' page. The 'Conexión de la instancia EC2' tab is selected. The page displays the instance ID (i-086028db364898105) and the public IP address (54.221.145.197). Under 'Tipo de conexión', there are two options: 'Conectarse mediante la Conexión de la instancia EC2' (selected) and 'Conectarse mediante punto de conexión de EC2 Instance Connect'. The 'Conectar' button is circled in red. A note at the bottom states: 'Nota: En la mayoría de los casos, el nombre de usuario predeterminado, ubuntu, es correcto. Sin embargo, lea las instrucciones de uso de la AMI para comprobar si el propietario de la AMI ha cambiado el nombre de usuario predeterminado.'

Con ello, tendremos la capacidad de usar el terminal de la instancia mediante el navegador de forma más o menos cómoda (yo te recomiendo en el futuro conectarte por SSH desde tu terminal u otro software específico).

```
aws Servicios Q Buscar [Alt+S]
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1012-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

System information as of Tue Nov 21 09:39:32 UTC 2023

System load:  0.0               Processes:           97
Usage of /:   22.3% of 7.57GB   Users logged in:    1
Memory usage: 22%              IPv4 address for eth0: 172.31.29.133
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Tue Nov 21 09:26:00 2023 from 18.206.107.27
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-29-133:~$
```

Realicemos ahora los siguientes cambios:

- Cambiar la contraseña del usuario ubuntu con ***sudo passwd ubuntu***
- Modificar el comportamiento del servidor SSH para aceptar autenticación mediante contraseña. Para ello, ejecuta ***sudo nano /etc/ssh/sshd\_config*** y baja por el fichero hasta encontrar la línea descomentada que pone **PasswordAuthentication no**. Modifica el valor a **yes**, guarda el fichero con Ctrl+S y sal con Ctrl+X.

```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no
```

- Una vez modificado el fichero de configuración del servidor SSH, reiniciamos dicho servidor mediante el comando ***sudo service sshd restart***. Comprueba con ***sudo service sshd status*** que el servicio está funcionando adecuadamente.

Usa ahora tu consola o MobaXterm si estás en trabajando en Windows para realizar la conexión.

### Parte 3: Instalar MongoDB en Ubuntu 22.04

A partir de aquí, utilizaremos la documentación que ofrece el propio MongoDB para su instalación en sistemas Ubuntu, la cual puedes consultar [aquí](#).

Los repositorios oficiales del paquete de Ubuntu incluyen una versión estable de MongoDB, sin embargo, dicha versión no tiene por qué ser la última versión estable. Para obtener la versión más reciente, debemos incluir el repositorio de paquetes dedicado de MongoDB a las fuentes.

El primer paso es instalar los paquetes curl y gnupg, los cuáles deberían estar ya instalados por defecto en la instancia. En cualquier caso, asegúrate de ello con el siguiente comando:

***sudo apt install gnupg curl***

Ahora ya podemos importar la clave pública GPG para la última versión estable de MongoDB con el siguiente comando:

***curl -fsSL https://pgp.mongodb.com/server-7.0.asc |sudo gpg -o /usr/share/keyrings/mongodb-server-7.0.gpg --dearmor***

A continuación, añade el repositorio APT de MongoDB 7.0 al directorio de fuentes /etc/apt/sources.list.d:

***echo "deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-7.0.gpg ]  
https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/7.0 multiverse" | sudo tee  
/etc/apt/sources.list.d/mongodb-org-7.0.list***

Podemos comprobar que ahora existe el fichero mongodb-org-7.0.list en dicho directorio y su contenido es la dirección del repositorio:

```
ubuntu@ip-172-31-29-133:~$ cat /etc/apt/sources.list.d/mongodb-org-7.0.list
deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-7.0.gpg ] https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/7.0 multiverse
```

Una vez añadido el repositorio, actualizamos los repositorios locales con ***sudo apt update*** haciendo que Ubuntu se de cuenta de que hay un nuevo repositorio MongoDB 7.0 añadido.

Ahora, ya podemos instalar la última versión estable de MongoDB con ***sudo apt install mongodb-org -y***

```
ubuntu@ip-172-31-29-133:~$ sudo apt install mongodb-org -y
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following additional packages will be installed:
  mongodb-database-tools mongodb-mongosh mongodb-org-database mongodb-org-database-tools-extra
  mongodb-org-mongos mongodb-org-server mongodb-org-shell mongodb-org-tools
The following NEW packages will be installed:
  mongodb-database-tools mongodb-mongosh mongodb-org mongodb-org-database
  mongodb-org-database-tools-extra mongodb-org-mongos mongodb-org-server mongodb-org-shell
  mongodb-org-tools
0 upgraded, 9 newly installed, 0 to remove and 55 not upgraded.
Need to get 162 MB of archives.
After this operation, 530 MB of additional disk space will be used.
```

El comando anterior instala el servidor de BD MongoDB y componentes como herramientas de shell. Una vez la instalación termina, comprueba la versión instalada con ***mongod --version***

### Parte 3: Configurar el servicio

El servicio **mongod** relativo al servidor de BD MongoDB está desactivado al instalarse por defecto. Puedes comprobarlo mediante el comando **sudo systemctl status mongod**

Para activar el servicio usa el comando **sudo systemctl start mongod** y comprueba su funcionamiento de nuevo con **sudo systemctl status mongod**

Si apareciera un error similar a *"Failed to start mongod.service: Unit mongod.service not found"*, ejecuta primero el comando **sudo systemctl daemon-reload** y de nuevo el comando **start** anterior.

Es opcional, pero recomendable, que MongoDB se inicie tras reiniciar el sistema. Para ello, ejecuta el comando **sudo systemctl enable mongod**.

Por último, vamos a comprobar que MongoDB está escuchando peticiones por su puerto por defecto, que es el 27017. Para ello, usa el comando **sudo ss -nltu | grep 27017** y comprueba que se muestra algo así:

```
ubuntu@ip-172-31-29-133:~$ sudo ss -nltu | grep 27017
tcp LISTEN 0 4096 127.0.0.1:27017 0.0.0.0:* users:((("mongod",pid=3280,fd=13))
```

### Parte 4: Crear regla en el grupo de seguridad

Al crear la instancia, indicamos que el grupo de seguridad tuviera una regla de entrada para el puerto 22 de SSH. Sin embargo, si queremos comunicarnos desde nuestra aplicación con el servidor de MongoDB de AWS, debemos también permitir las comunicaciones entrantes por el puerto 27017.

En la información de la instancia, en la parte inferior, pincha en **Seguridad** y verás algo como esto:

Nombre	ID de la regla del grupo d...	Intervalo de pu...	Protocolo	Origen	Grupos de seguridad
--	sgr-05d540080533d663a	22	TCP	0.0.0.0/0	launch-wizard-42

Haz clic en el enlace a la configuración del Grupo de Seguridad y haz clic en **Editar reglas de entrada**.

Agrega una nueva regla de tipo TCP personalizado, para el puerto 27017 y con origen 0.0.0.0/0

TCP personalizado

TCP

27017

Anywh...

0.0.0.0/0

0.0.0.0/0 X

Haz clic en **Guardar reglas** y el puerto de MongoDB quedará abierto para peticiones entrantes.

## Parte 5: Configuración para acceso remoto

MongoDB está configurado para acceder localmente en el mismo servidor en el que se instaló de forma predeterminada y, para habilitar el acceso remoto, debemos editar el archivo `/etc/mongod.conf`, el archivo de configuración principal de MongoDB.

Este archivo contiene configuraciones para la ubicación de almacenamiento de la base de datos, el registro, la red y la administración de procesos.

Edita el archivo de configuración con `sudo nano /etc/mongod.conf` y, en el apartado **network interfaces**, comenta la línea que empieza `bindIp: 127.0.0.1...` para escribir en la siguiente línea **bindIp: 0.0.0.0**

Guarda el archivo y reinicia el servidor con el comando `sudo systemctl restart mongod`

## Parte 6: Asociar una dirección IP elástica a la instancia

Las direcciones IPs públicas de las instancias del laboratorio cambian con cada reinicio del mismo, haciendo tedioso su uso desde la aplicación que consumirá la información almacenada en la base de datos.

Para asignar una IP pública fija a la instancia, debemos “alquilar” una IP elástica y asignársela a la instancia, de esa forma, no tendremos que preocuparnos de los cambios de IP en cada reinicio.

En el Panel de EC2 de AWS, puedes encontrar el menú de **Direcciones IP elásticas** tanto en el bloque de **Recursos** como en el menú izquierdo en el apartado **Red y seguridad**.

Entra y haz clic en **Asignar la dirección IP elástica**. Dentro de la nueva pestaña que se abre dale a **Asignar**.


Lo próximo que tenemos que hacer es asociarla a la instancia. Para ello, selecciónala y haz clic en **Acciones > Asociar la dirección IP elástica**.

Dirección IP elástica: 54.146.188.92

### Tipo de recurso

Elija el tipo de recurso al que desea asociar la dirección IP elástica.

- ☒ Instancia  
☐ Interfaz de red

 Si asocia una dirección IP elástica a una instancia que ya tiene asociada una dirección IP elástica, la dirección IP elástica asociada anteriormente se desasociará, pero la dirección seguirá asignada a la cuenta. [Más información](#)

Si no se especifica ninguna dirección IP privada, la dirección IP elástica se asociará a la dirección IP privada principal.

### Instancia

 i-086028db364898105



Haz clic en **Asociar** y comprueba que la dirección IP pública de tu instancia ha cambiado a la IP elástica.

## Parte 7: Crear base de datos y usuario en MongoDB

A estas alturas, MongoDB debería estar ejecutándose y configurada para acceso remoto. Veamos ahora cómo crear una base de datos y un usuario en MongoDB.

Para acceder a MongoDB, ejecuta el comando ***mongosh***

Antes de acceder al shell de MongoDB, se muestran algunos detalles sobre MongoDB, como la versión de MongoDB y el shell de MongoDB y la URL de la documentación de Mongosh.

Encima del símbolo del shell de Mongo, también saldrá una advertencia que sugiere que el control de acceso no se ha habilitado para la base de datos y que el acceso de lectura y escritura a los datos y la configuración no está restringido. Esta advertencia se muestra porque la autenticación aún no se ha habilitado, desaparecerá después de que se habilite la autenticación en la base de datos.

```
ubuntu@ip-172-31-29-133:~$ mongosh
Current Mongosh Log ID: 655c9343dc8c2e5719bb5ed4
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&
appName=mongosh+2.0.2
Using MongoDB:      7.0.3
Using Mongosh:      2.0.2
mongosh 2.1.0 is available for download: https://www.mongodb.com/try/download/shell

For mongosh info see: https://docs.mongodb.com/mongosh-shell/

-----
The server generated these startup warnings when booting
2023-11-21T10:49:48.880+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger
storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2023-11-21T10:49:50.129+00:00: Access control is not enabled for the database. Read and write access
to data and configuration is unrestricted
2023-11-21T10:49:50.129+00:00: vm.max_map_count is too low
-----
```

Durante la instalación, se crean de forma predeterminada tres bases de datos: admin, config y local. Para listar las bases de datos existentes, ejecuta el comando ***show dbs***.

Para crear una base de datos, utiliza el comando ***use*** seguido del nombre de la base de datos. Por ejemplo, para crear una base de datos llamada pruebas ejecuta el comando ***use pruebas***

MongoDB proporciona varios métodos para administrar una base de datos. El método ***db.createUser*** nos permite crear un nuevo usuario en una base de datos.

Este método requiere que definamos el nombre de usuario y su contraseña, además de cualquier rol que queramos otorgarle. Toda esta información se introduce en formato JSON.

La sintaxis para crear un usuario llamado alejandro con funciones de lectura y escritura en la base de datos de prueba sería la siguiente:

```
db.createUser( {user: "alejandro", pwd: "mipassword", roles: [ {role: "readWrite", db: "prueba"} ] } )
```

Podemos listar los usuarios creados usando el método ***db.getUsers()*** o ***show users***



## Parte 8: Seguridad en MongoDB

La autenticación no está habilitada de forma predeterminada en MongoDB, lo que implica que cualquier usuario con acceso al servidor de la base de datos puede ver, agregar y eliminar datos. Esta vulnerabilidad puede provocar una pérdida de datos, por lo que es importante proteger MongoDB.

Para solucionar este problema, vamos a crear un usuario administrador, por lo que conéctate a la base de datos **admin** con **use admin**.

Tras esto, crea un usuario con las siguientes características:

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

Vamos a revisar lo que hemos hecho:

- La línea `user: "AdminAlejandro"` crea un usuario administrador llamado AdminAlejandro.
- El método `pwd: passwordPrompt()` le solicita la contraseña del usuario administrador. Es una alternativa más segura al campo `pwd:`, que requiere escribir la contraseña en texto sin cifrar.
- La línea `roles: [ { role: "userAdminAnyDatabase", db: "admin" }, "readWriteAnyDatabase" ]` define los roles otorgados al usuario administrador. En este caso, se le otorgan permisos de lectura y escritura en la BD admin, además de poder leer y modificar todas las bases de datos del clúster.

```
admin> db.createUser(  
...  {  
...    user: "AdminAlejandro",  
...    pwd: passwordPrompt(),  
...    roles: [ { role: "userAdminAnyDatabase", db: "admin" }, "readWriteAnyDatabase" ]  
...  }  
... )  
Enter password  
****{ ok: 1 }
```

Una vez creado el usuario administrador, el siguiente paso es permitir la autenticación. Para ello, sal del shell de MongoDB y edita el fichero **/etc/mongod.conf**. Busca la sección **security**, descoméntala y añade una directiva **authorization** que configuraremos como habilitada. **¡¡Ten cuidado con la indentación!!**

```
security:  
  authorization: enabled
```

Guarda el fichero y reinicia el servidor mongod. Comprueba con **sudo systemctl status mongod** que funciona como se espera.

Vuelve a entrar en el Mongo Shell y comprueba que las advertencias han desaparecido.



```
ubuntu@ip-172-31-29-133:~$ mongosh
Current Mongosh Log ID: 655c9b612e3f6a5823097cd3
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&
appName=mongosh+2.0.2
Using MongoDB:      7.0.3
Using Mongosh:      2.0.2
mongosh 2.1.0 is available for download: https://www.mongodb.com/try/download/shell

For mongosh info see: https://docs.mongodb.com/mongosh-shell/

test> █
```

Y ahora, si intentamos hacer alguna tarea relacionada con bases de datos, como listarlas, nos va a dar un error ya que es una operación que requiere autenticación:

```
test> show dbs
MongoServerError: Command listDatabases requires authentication
test> █
```

Para logearnos de forma autenticada, sal del shell y entra de esta forma:

***mongosh -u AdminAlejandro -p --authenticationDatabase admin***

Introducimos la contraseña y ya podremos manipular las bases de datos:

```
ubuntu@ip-172-31-29-133:~$ mongosh -u AdminAlejandro -p --authenticationDatabase admin
Enter password: ****
Current Mongosh Log ID: 655c9c1a258f069b26b02939
Connecting to:      mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionT
imeoutMS=2000&authSource=admin&appName=mongosh+2.0.2
Using MongoDB:      7.0.3
Using Mongosh:      2.0.2
mongosh 2.1.0 is available for download: https://www.mongodb.com/try/download/shell

For mongosh info see: https://docs.mongodb.com/mongosh-shell/

test> show dbs
admin    180.00 KiB
config   96.00 KiB
local    72.00 KiB
test> █
```

### **Paso 9: Comprobación de conexión remota y segura**

Con todas las configuraciones realizadas, estamos ya en disposición de conectar de forma remota y segura a nuestra instancia de base de datos MongoDB.

Para ello, usa la máquina Lliurex, el shell de MongoDB si trabajas desde un Windows o instala un shell en otra instancia EC2 en AWS y ejecuta el comando ***mongosh "mongodb://usuario:password@urlaws:27017"***, que en mi caso es el siguiente

***mongosh "mongodb://AdminAlejandro:1234@ec2-54-146-188-92.compute-1.amazonaws.com:27017"***

## Ejercicio práctico

Ahora que ya tienes configurado MongoDB y puedes conectarte tanto de forma local como remota, conéctate a ella y utiliza una base de datos que llamarás **prueba**.

Realiza las siguientes operaciones mediante **MQL (MongoDB Query Language)** y el shell:

**Insertar** los siguientes **documentos** en una **colección** llamada **pelis**.

- título: El Club de la Lucha, director: David Fincher, año: 1999, actores: [Brad Pitt, Edward Norton]
- título: Pulp Fiction, director: Quentin Tarantino, año: 1994, actores: [John Travolta, Uma Thurman]
- título: Malditos bastardos, director: Quentin Tarantino, año: 2009, actores: [Brad Pitt, Diane Kruger, Eli Roth]
- título: El Hobbit: Un Viaje Inesperado, director: Peter Jackson, año: 2012, saga: El Hobbit
- título: El Hobbit: La desolación de Smaug, director: Peter Jackson, año: 2013, saga: El Hobbit
- título: El Hobbit: La batalla de los Cinco Ejércitos, director: Peter Jackson, año: 2014, saga: El Hobbit, sinopsis: Bilbo y compañía se ven obligados a participar en una guerra contra una serie de combatientes y evitar que la Montaña Solitaria caiga en manos de una oscuridad creciente.
- título: La Gran Aventura de Pee-Wee
- título: Avatar

Realizar las siguientes **consultas** en la colección **pelis**:

- Obtener todos los documentos
- Obtener documentos con director igual a "Quentin Tarantino"
- Obtener documentos con actores que incluyan a "Brad Pitt"
- Obtener documentos con saga igual a "El Hobbit"
- Obtener todas las películas de los 90s.
- Obtener las películas estrenadas entre el año 2000 y 2010.

**Actualiza** los siguientes documentos:

- Agregar sinopsis a "El Hobbit: Un Viaje Inesperado": "Un hobbit reacio, Bilbo Bolsón, se dirige a la Montaña Solitaria con un enérgico grupo de enanos para recuperar su hogar en la montaña (y el oro que contiene) del dragón Smaug."
- Agregar sinopsis a "El Hobbit: La desolación de Smaug": "Los enanos, junto con Bilbo Bolsón y Gandalf el Gris, continúan su búsqueda para recuperar Erebor, su tierra natal, en manos de Smaug. Bilbo Bolsón está en posesión de un anillo misterioso y mágico."

- Agregar un actor llamado "Samuel L. Jackson" a la película "Pulp Fiction".

Realiza las siguientes **búsquedas de texto**:

- Encontrar las películas que en la sinopsis contengan la palabra "Bilbo"
- Encontrar las películas que en la sinopsis contengan la palabra "Gandalf"
- Encontrar las películas que en la sinopsis contengan la palabra "Bilbo" y no la palabra "Gandalf"
- Encontrar las películas que en la sinopsis contengan la palabra "enanos" o "hobbit"
- Encontrar las películas que en la sinopsis contengan la palabra "oro" y "dragón"

**Elimina** los siguientes documentos:

- Eliminar la película "La Gran Aventura de Pee-Wee"
- Eliminar la película "Avatar"