

Big Data y Minería de Datos Geoespaciales

MongoDB en la nube: MongoDB Atlas

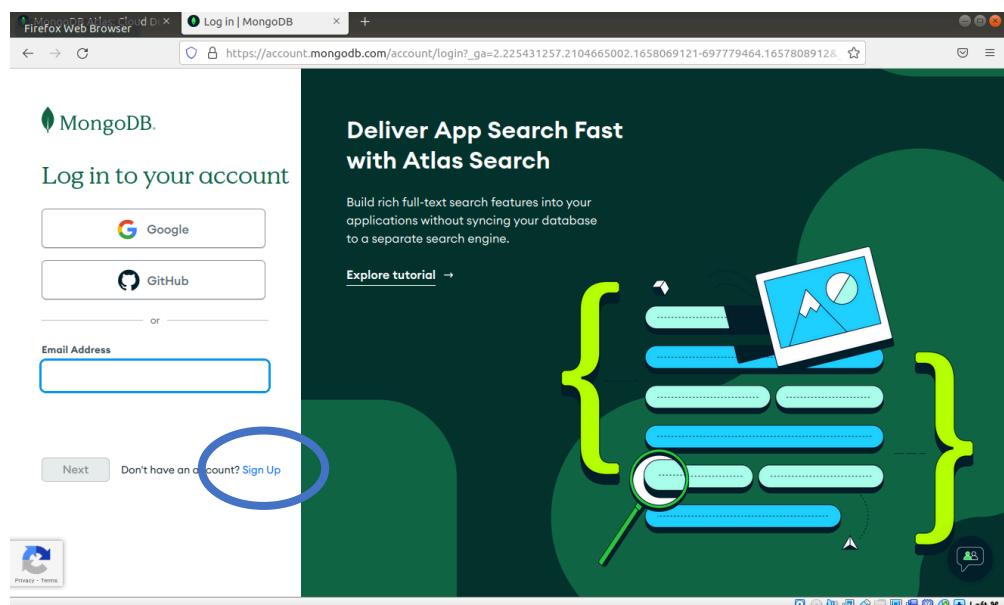
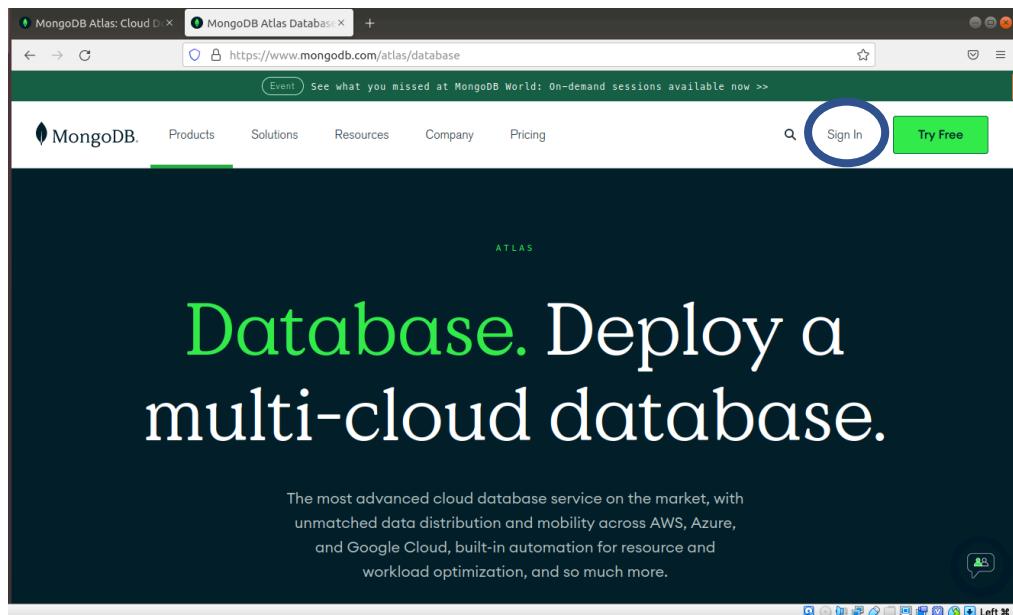
Octubre 2024
Ángel Martín Furones

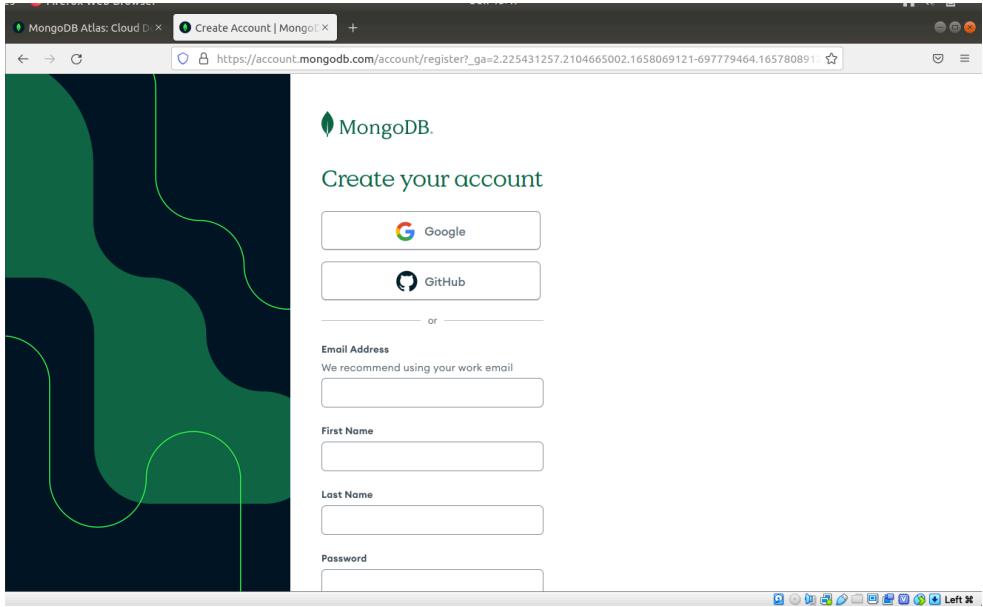
ALTA EN EL SERVICIO Y GENERACION DEL PRIMER CLÚSTER

Vamos a la página WEB de MongoDB Atlas y nos registramos:

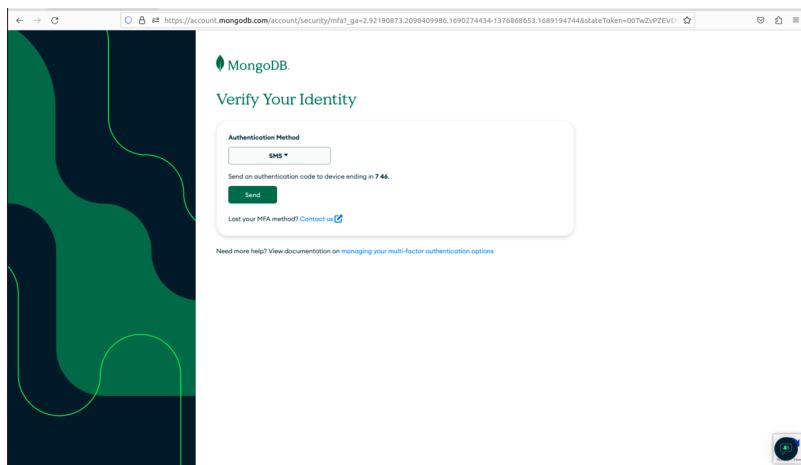
<https://www.mongodb.com/atlas/database>

Todos los pasos para registrarse también se pueden encontrar en la propia web de Mongo Atlas: <https://www.mongodb.com/docs/atlas>

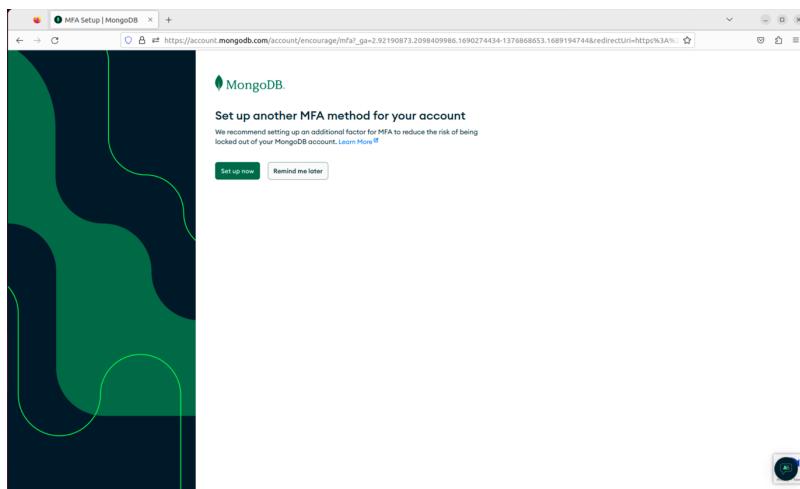




Rellena el formulario y pincha sobre sobre *continue*. Ya podemos *loggearnos* dentro del sistema. Como medida de seguridad, Mongodb te enviará un código mediante SMS para poder acceder a la plataforma:



En la siguiente pantalla tendrá la opción de poder agregar segundo método de autentificación, selecciona otro método si quieres o, simplemente, pincha sobre *Remind me Later*:



Atlas generará automáticamente un grupo de trabajo (*Organization and Project*), donde desplegar el primer clúster o *database*. Los grupos de trabajo se deben generar por seguridad, para que no existan interferencias entre los diferentes usuarios del servicio. El grupo de trabajo se puede cambiar más tarde dentro de la plataforma Atlas.

Lo primero que se nos pedirá es que creamos un clúster:

Aquí aparecerá el nombre del grupo de trabajo que Atlas ha generado automáticamente

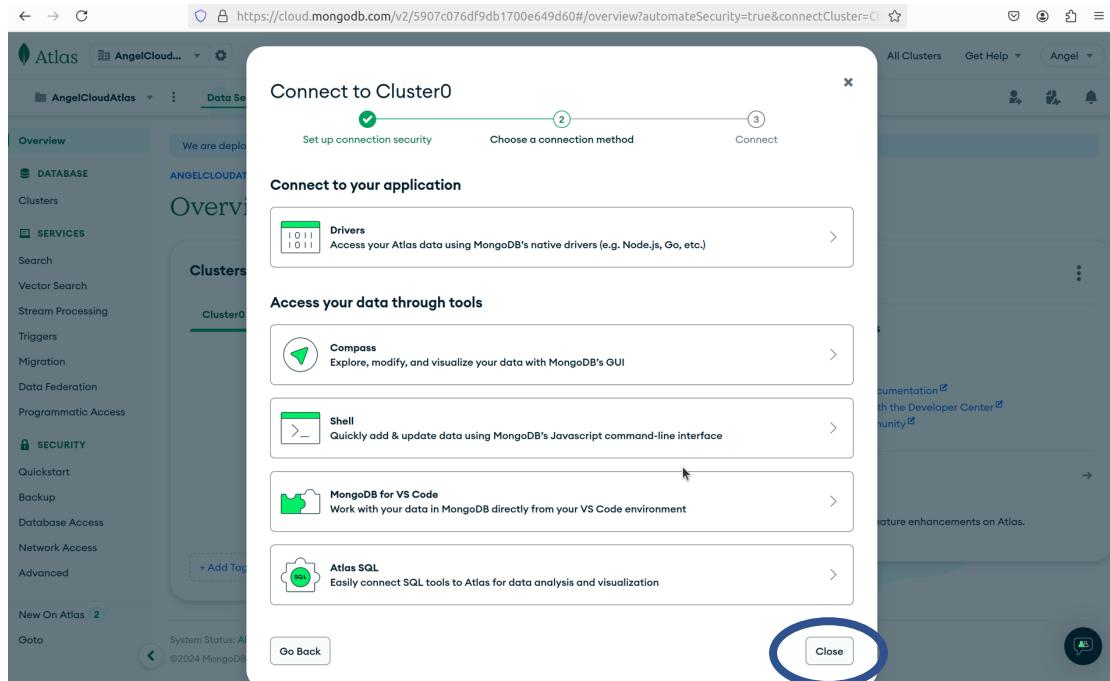
Pinchamos sobre *create* y empezará el proceso de creación de una base de datos MongoDB en la nube. Vamos a crear un clúster gratuito, para ello, seleccionamos la columna *M0 FREE*, que generará un clúster gratuito de 512 MB de memoria (si quieres ver los tipos de clúster y sus

máquinas asociadas puedes echarle un vistazo pinchando sobre *go to advanced configuration* en la columna central abajo):

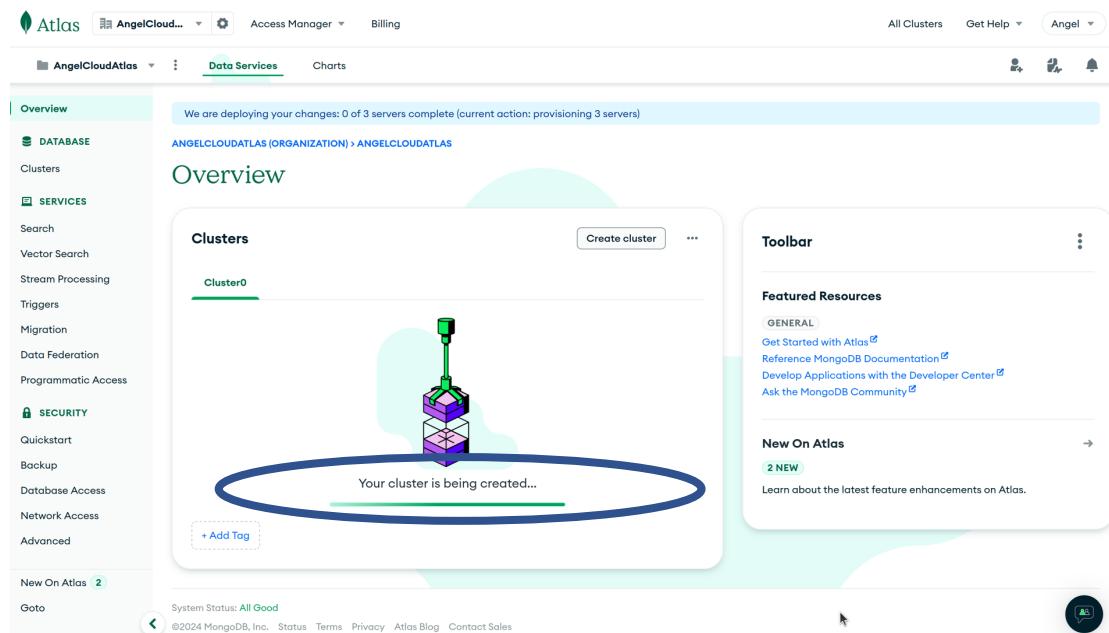
Bajamos con el Scroll y podemos elegir alguna de las tres plataformas de servicios en la nube: Amazon, Google Cloud o Azure (elegimos Amazon Web Service, AWS y la región eu-west-3, que son las que lleva por defecto):

En la pestaña de *Cluster Name* asignaremos un nombre a nuestro clúster, podemos dejar el que bien por defecto *Cluster0*, y en la ventana de *Tags* no ponemos nada y pinchamos sobre *Create Deployment*.

El proceso de creación del clúster empezará. Aparecerá la siguiente pantalla mientras el clúster se está generando, básicamente, nos informa de los métodos de conexión, que, en nuestro caso, veremos más adelante. Lo único a tener en cuenta en este momento es que, por defecto, se añade la dirección IP local de la máquina de despliegue para poder conectarse al *clúster*. Cierra esta pantalla pinchando sobre *Close*.



El proceso de generación puede tardar algunos segundos.



The screenshot shows the MongoDB Atlas Overview page for the organization ANGELCLOUDATLAS. On the left, a sidebar lists various services like Database, Services, Security, and Data Federation. The main area displays a 'Clusters' section for Cluster0, featuring buttons for 'Connect', 'Edit configuration', 'Add data', 'Migrate data', 'Load sample data', and 'Data modeling templates'. A prominent blue progress bar at the bottom indicates the status of a deployment: 'Deploying your data' with the message 'We are loading your new dataset to Cluster0.' A green checkmark icon is visible on the right side of the progress bar. The progress bar is circled in blue.

This screenshot shows the same MongoDB Atlas Overview page after the deployment has completed. The progress bar message has changed to 'Sample dataset successfully loaded.' and now includes a green checkmark icon. It also provides instructions: 'Access it in Collections or by connecting with the MongoDB Shell.' The progress bar is circled in blue.

Una vez generado el clúster, puedes acceder a él y ver sus propiedades pinchando sobre *clusters*:

The screenshot shows the MongoDB Atlas interface. On the left sidebar, under the 'Clusters' section, the 'Clusters' tab is selected. In the main area, a cluster named 'Cluster0' is listed. The cluster details include:

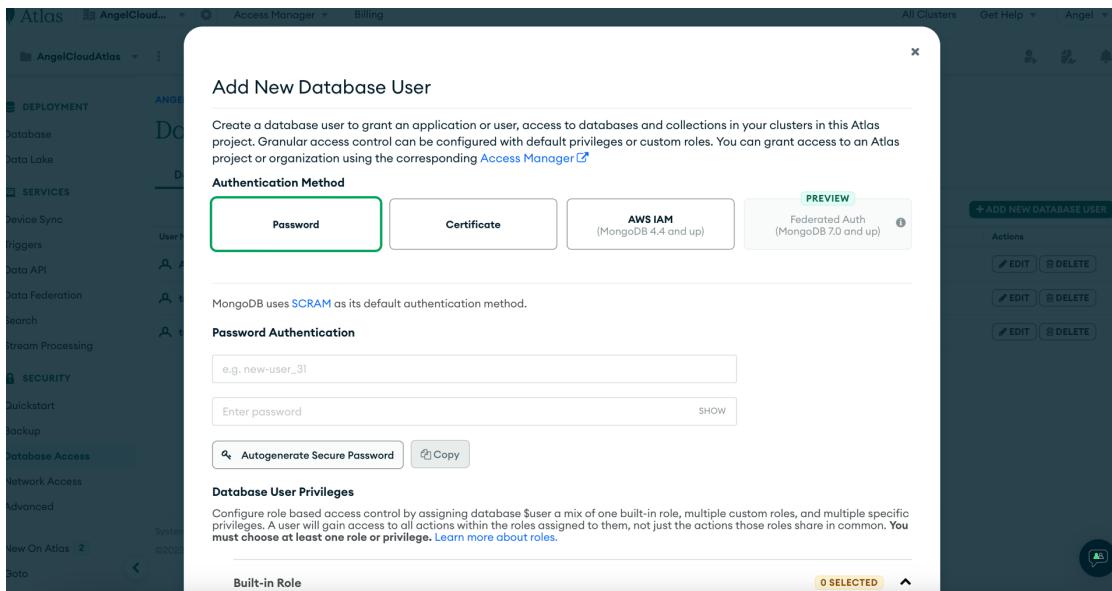
- VERSION:** 7.0.12
- REGION:** AWS / Paris (eu-west-3)
- CLUSTER TIER:** M0 Sandbox (General)
- TYPE:** Replica Set - 3 nodes
- BACKUPS:** Inactive
- LINKED APP SERVICES:** None Linked
- ATLAS SQL:** Connect
- ATLAS SEARCH:** Create Index

Para poder conectarnos a la base de datos desde nuestra máquina local es necesario generar un usuario, para eso pinchamos sobre *Database Access*, en las opciones de *Security* que aparecen desplegadas en la parte izquierda, aparecerá la siguiente ventana (los usuarios mostrados no serán los mismos):

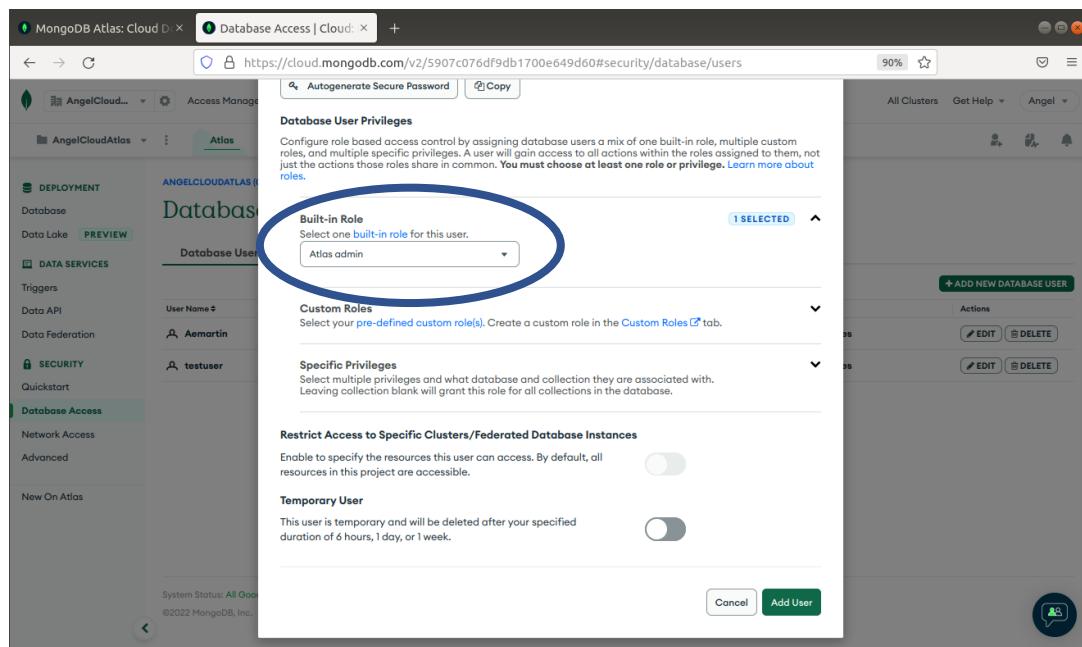
The screenshot shows the MongoDB Atlas interface. On the left sidebar, under the 'SECURITY' section, the 'Database Access' tab is selected. In the main area, the 'Database Users' section displays the following users:

User Name	Authentication Method	MongoDB Roles	Resources	Actions
Aemartin	SCRAM	atlasAdmin@admin	All Resources	<button>Edit</button> <button>Delete</button>
test2	SCRAM	readWriteAnyDatabase@dbadmin	All Resources	<button>Edit</button> <button>Delete</button>
testuser	SCRAM	readWrite@test	All Resources	<button>Edit</button> <button>Delete</button>

Pinchamos sobre *Add New Database User*. En la ventana que aparece pinchamos sobre *Password* y escribimos un nombre de usuario y un password (fundamental para conectarnos posteriormente al clúster -a este y, si no se dice lo contrario, a todos lo que generes-, por lo que debes guardar esta información a buen recaudo):



A continuación, en la pestaña *Database User Privileges*, selecciona *Atlas admin* en la ventana *Built-in-Role* para tener acceso de administrador:



Finalmente, pincha sobre *Add User*, para generar el usuario.

Nuestro clúster, de momento, solo es accesible a la IP local desde la que hemos creado el *clúster*, si queremos añadir otra IP de acceso al mismo, pinchar sobre *Network Access* en el mismo menú de *Security* que anteriormente, y aparecerá la siguiente ventana, donde podemos ver la conexión a la IP actual:

The screenshot shows the MongoDB Atlas Network Access page. On the left sidebar, under the Security section, the 'Network Access' option is highlighted and circled in blue. The main area displays the 'IP Access List' tab, which lists three active entries:

IP Address	Comment	Status	Actions
158.42.117.23/32 (includes your current IP address)	Created as part of the Auto Setup process	Active	<button>EDIT</button> <button>DELETE</button>
158.42.59.205/32		Active	<button>EDIT</button> <button>DELETE</button>
2.142.31.111/32		Active	<button>EDIT</button> <button>DELETE</button>

A yellow banner at the top of the list states: "You will only be able to connect to your cluster from the following list of IP Addresses:". A green button labeled "+ ADD IP ADDRESS" is located in the top right corner of the list table.

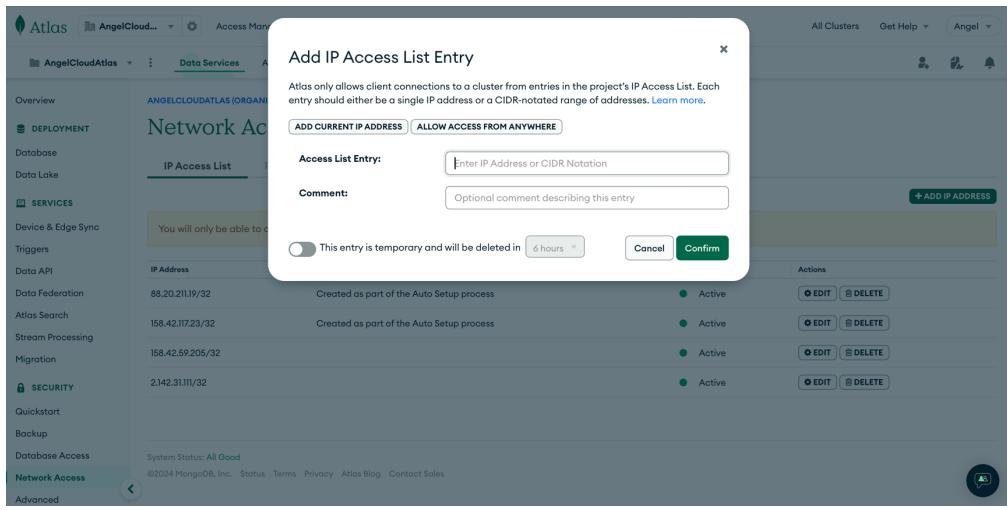
Pincha sobre *ADD IP ADDRESS* y aparecerá la ventana:

The screenshot shows the 'Add IP Access List Entry' dialog box overlaid on the Network Access page. The dialog has the following fields:

- ALLOW ACCESS FROM ANYWHERE**: A checkbox that is unchecked.
- Access List Entry:** An input field containing the placeholder "Enter IP Address or CIDR Notation".
- Comment:** An input field containing the placeholder "Optional comment describing this entry".
- This entry is temporary and will be deleted in:** A dropdown menu set to "0 hours".
- Buttons:** "Cancel" and "Confirm".

Para dejar acceso al clúster a cualquier usuario, cualquier dirección IP (no recomendado), pinchamos sobre *Add IP Access List Entry* y luego sobre *Confirm*.

Si elegimos trabajar conectándonos a través de la IP local, cada vez que cambiemos de Wifi o incluso que encendamos la máquina, es posible que la IP de nuestra máquina cambie, por lo que debemos añadir esa IP, este es un proceso que deberemos efectuar cada vez que nos queramos conectar a alguno de los clústers que hemos generado en Mongo Atlas. Es buena práctica, además, ir borrando las direcciones IP anteriores del sistema. Para ello, al pinchar sobre **ADD IP ADDRESS** aparecerá la ventana:



Pincha sobre **ADD CURRENT IP ADDRESS** y luego sobre **CONFIRM** y la IP actual se añadirá a la lista:

IP Address	Comment	Status	Actions
88.20.21.19/32	Created as part of the Auto Setup process	Active	EDIT DELETE
158.42.117.23/32	Created as part of the Auto Setup process	Active	EDIT DELETE
158.42.59.205/32		Active	EDIT DELETE
158.42.153.145/32 (includes your current IP address)		Active	EDIT DELETE
2.142.31.111/32		Active	EDIT DELETE

Una vez añadida la IP actual, volvemos a la pantalla inicial pinchando sobre *clusters* en la columna izquierda y luego pinchamos sobre el nombre *Cluster0* en azul:

The screenshot shows the MongoDB Atlas interface. On the left, there's a sidebar with various options like Overview, Clusters (which is selected and highlighted with a blue circle), and Services. The main content area is titled 'Clusters' and shows a single cluster named 'Cluster0'. A modal window is open over the cluster card, indicating that a sample dataset has been successfully loaded. The cluster details shown are: Version 7.0.12, Region AWS / Paris (eu-west-3), Cluster Tier M0 Sandbox (General), Type Replica Set - 3 nodes, Backups Inactive, Linked App Services None Linked, and ATLAS SQL and ATLAS SEARCH both with a 'Connect' button.

Vemos que nuestro clúster contiene un master y dos secundarios (se han levantado tres máquinas, una de ellas que actuará como master, que es a la que accederemos, y dos secundarias que actuarán de réplicas haciendo *backups* dinámicos sobre lo que escribamos en la máquina master). A esta estructura se la conoce con el nombre de **Replica Set** en mongoDB.

This screenshot shows the 'Database' overview for 'Cluster0'. The left sidebar has 'database' selected. The main panel displays the 'Cluster0' database with three shards: shard-00-00, shard-00-01, and shard-00-02. Each shard is labeled as 'SECONDARY'. Below the shards, there's a section for 'This is a Shared Tier Cluster' with a 'Upgrade' button. The database metrics shown are: Logical Size (134.5 MB), Operations (R: 0, W: 0), and Connections (0).

Conexión al clúster

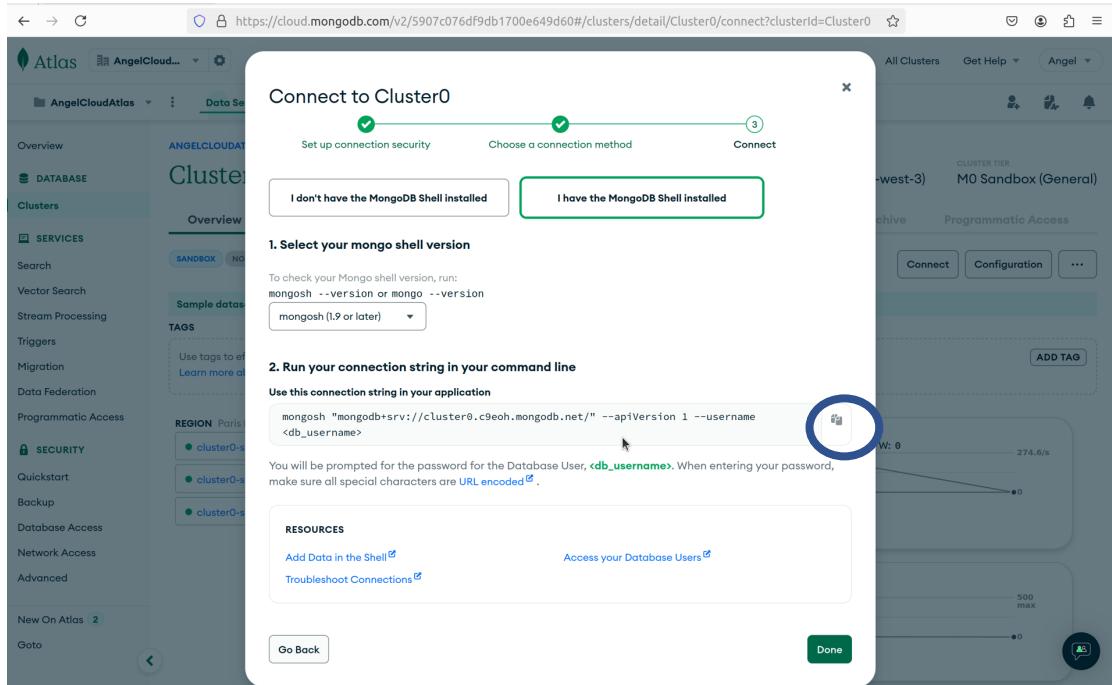
Para conectarnos al master del clúster desde nuestra máquina local pinchamos sobre **CONNECT**:

The screenshot shows the MongoDB Cloud Atlas Cluster Overview page for 'Cluster0'. The 'Connect' button is circled in blue at the top right of the main content area. The page displays various cluster metrics and a list of shards.

Aparecerá la siguiente pantalla con las opciones de conexión, donde seleccionamos *Shell* ya que será la herramienta de conexión que usaremos:

The screenshot shows the 'Connect to Cluster0' dialog box. The 'Shell' option under 'Access your data through tools' is circled in blue. The dialog box has three tabs: 'Set up connection security', 'Choose a connection method', and 'Connect'.

A continuación, seleccionaremos *I have the MongoDB Shell* ya que la Shell la tenemos ya instalada y copiaremos la orden para generar la conexión al clúster desde la terminal:



En nuestra máquina local, en una ventana de terminal sobre el directorio *bin* donde tenemos grabado MongoDB pegamos la instrucción copiada anteriormente (poniendo al principio ./mongosh en lugar de mongosh simplemente):

```
vagrant@vagrant:/usr/local/mongodb/mongodb-7.0.12/bin$ ./mongosh "mongodb+srv://cluster0.ay4vpe7.mongodb.net/" --apiVersion 1 --username Aemartin
```

Introduciremos el *password* definido anteriormente al generar el usuario y deberíamos conectarnos al máster del clúster:

```
vagrant@vagrant:/usr/local/mongodb/mongodb-7.0.12/bin$ ./mongosh "mongodb+srv://cluster0.ay4vpe7.mongodb.net/" --apiVersion 1 --username Aemartin
Enter password: *****
Current Mongosh Log ID: 668bc5a41a5766da4c149f47
Connecting to: mongodb+srv://<credentials>@cluster0.ay4vpe7.mongodb.net/?appName=mongosh+2.2.10
Using MongoDB: 7.0.11 (API Version 1)
Using Mongosh: 2.2.10

For mongosh info see: https://docs.mongodb.com/mongodb-shell/
Atlas atlas-10cijx-shard-0 [primary] test>
```

El servidor está activo en el *cloud* desde el momento que hemos generado el clúster, no tienes que activarlo de forma local.

Para estar seguros de la conexión podremos usar el comando *show dbs* para ver las bases de datos por defecto, que deberán ser *admin* y *local*:

```

Atlas atlas-10cijx-shard-0 [primary] test> show dbs
sample_mflix 113.79 MiB
admin      280.00 KiB
local      1.70 GiB
Atlas atlas-10cijx-shard-0 [primary] test>

```

Carga de una base de datos y acceso desde Python

Para realizar esta parte debes estar conectado al máster del clúster, trabajaremos con la base de datos *city_inspections.json* que puedes descargar desde poliformaT (lessons/practicas/Bases de datos NoSQL). Se trata de una base de datos en formato json sobre inspecciones a empresas de EEUU, donde se puede ver un identificador, el nombre de la empresa, la fecha de la inspección, el resultado, el sector al que pertenece la empresa y la dirección en formato diccionario con diferentes campos, por ejemplo:

```

    "id" : "10021-2015-ENFO",
    "certificate_number" : 9278806,
    "business_name" : "ATLIXCO DELI GROCERY INC.",
    "date" : "Feb 20 2015",
    "result" : "No Violation Issued",
    "sector" : "Cigarette Retail Dealer - 127",
    "address" : {
        "city" : "RIDGEWOOD",
        "zip" : 11385,
        "street" : "MENAHAH ST",
        "number" : 1712
    }
}

```

Guarda la base de datos en un directorio local de tu máquina.

El procedimiento para importarla a nuestro *clúster* se puede ver si volvemos al clúster, pinchamos sobre la pestaña *Data services*:

The screenshot shows the MongoDB Atlas Data Services interface. A green bar at the top indicates that a sample dataset has been successfully loaded. Below this, the 'Cluster' section displays a summary of the cluster configuration, including the region (Paris), version (7.0.12), and tier (M0 Sandbox). The 'OVERVIEW' tab is selected, showing metrics such as logical size (134.8 MB) and operations (0 R, 0 W). The 'NODES' tab shows three nodes: one primary and two secondary. The 'COLLECTIONS' tab shows a single collection named 'cluster0-shard-00-00.c9eoh...'. The 'PERFORMANCE ADVISOR' tab provides recommendations for upgrading to a dedicated cluster if needed. On the left sidebar, the 'CLUSTERS' section is highlighted, and the 'DATA SERVICES' section is also visible. A blue oval highlights the 'Data Services' tab in the top navigation bar.

A continuación, sobre *Migrate Data*:

The screenshot shows the MongoDB Atlas Overview page for a cluster named 'Cluster0'. The cluster has a sample dataset successfully loaded, and its data size is 134.8 MB. A prominent blue oval highlights the 'Migrate data' button, which is located next to the 'Browse collections' and 'View monitoring' buttons. The left sidebar contains various service options like Database, Services, Security, and Migration. The right sidebar features a 'Toolbar' and 'Featured Resources' section.

A continuación, en *Other Migration Tools*, pincha sobre *Self-Managed Migration (view Command Line)*:

The screenshot shows the MongoDB Atlas Data Services page. The 'Migrate Data' tab is selected. Under the 'RECOMMENDED' section, there is a 'Relational Migrator' card. In the 'Other migration tools' section, there are two cards: 'Live Migration Services' and 'Self-Managed Migration'. The 'Self-Managed Migration' card is highlighted with a large blue oval. It describes using database command line utilities to import data and provides a 'VIEW COMMAND LINE' button. The left sidebar lists various services and security options, and the right sidebar includes a 'Toolbar' and 'New On Atlas' section.

En la ventana que nos aparece, bajamos hasta encontrar el comando `mongoimport` (también es posible encontrar aquí el comando necesario para hacer `mongoexport`, `mongodump` o `mongorestore`), y copiamos la opción para importar ficheros JSON (formato del fichero que vamos a cargar):

The screenshot shows the MongoDB Atlas interface under the 'Data Services' tab. In the 'Import a JSON file' section, there is a command:

```
mongoimport --uri mongodb+srv://Aemartin: <enter_password>@cluster0.c9eoh.mongodb.net/ --collection <enter_collection_name> --type JSON --file <enter path/to/file>
```

Next to the command, there is a 'Copy' button. This button is circled in blue in the screenshot.

Lo copiamos y lo pegamos en una ventana de terminal situada sobre el directorio bin de Mongodb local, sustituyendo <PASSWORD> por el password, <DATABASE> por el nombre de la base de datos que queramos crear, por ejemplo, `test`, <COLLECTION> por el nombre de la colección que queramos, por ejemplo, `inspections`, <FILETIPE> es el tipo de formato del fichero, es este caso `json` y en <FILENAME> debemos poner el path completo (incluyendo el nombre del fichero) donde tenemos situado el fichero a importar.

```
vagrant@vagrant:~/usr/local/mongodb/mongodb-7.0.12/bin$ ./mongoimport --uri mongodb+srv://Aemartin: <enter_password>@cluster0.ay4vpe7.mongodb.net/test --collection inspections --type json --file /home/vagrant/Documents/BigData/mongodb/city_inspections.json
```

El fichero empezará a cargarse en el cloud:

```
vagrant@vagrant:~/usr/local/mongodb/mongodb-7.0.12/bin$ ./mongoimport --uri mongodb+srv://Aemartin: <enter_password>@cluster0.ay4vpe7.mongodb.net/test --collection inspections --type json --file /home/vagrant/Documents/BigData/mongodb/city_inspections.json
2024-07-08T11:00:18.908+0000 connected to: mongodb+srv://Aemartin:H0sta1?megtron?@cluster0.ay4vpe7.mongodb.net/test
2024-07-08T11:00:21.923+0000 [.....] test.inspections 584KB/23.2MB (2.5%)
2024-07-08T11:00:24.913+0000 [.....] test.inspections 875KB/23.2MB (3.7%)
2024-07-08T11:00:27.909+0000 [#.....] test.inspections 1.14MB/23.2MB (4.9%)
2024-07-08T11:00:30.912+0000 [#.....] test.inspections 1.71MB/23.2MB (7.4%)
2024-07-08T11:00:33.914+0000 [##.....] test.inspections 2.29MB/23.2MB (9.9%)
2024-07-08T11:00:36.909+0000 [##.....] test.inspections 2.29MB/23.2MB (9.9%)
2024-07-08T11:00:39.909+0000 [##.....] test.inspections 2.57MB/23.2MB (11.1%)
2024-07-08T11:00:42.909+0000 [##.....] test.inspections 2.86MB/23.2MB (12.3%)
2024-07-08T11:00:45.913+0000 [##.....] test.inspections 3.14MB/23.2MB (13.6%)
2024-07-08T11:00:48.923+0000 [##.....] test.inspections 3.43MB/23.2MB (14.8%)
2024-07-08T11:00:51.908+0000 [##.....] test.inspections 3.72MB/23.2MB (16.0%)
2024-07-08T11:00:54.911+0000 [##.....] test.inspections 4.01MB/23.2MB (17.3%)
2024-07-08T11:00:57.911+0000 [##.....] test.inspections 4.58MB/23.2MB (19.7%)
2024-07-08T11:01:00.925+0000 [##.....] test.inspections 5.44MB/23.2MB (23.4%)
2024-07-08T11:01:03.909+0000 [##.....] test.inspections 5.72MB/23.2MB (24.7%)
2024-07-08T11:01:06.908+0000 [##.....] test.inspections 6.01MB/23.2MB (25.9%)
2024-07-08T11:01:09.910+0000 [##.....] test.inspections 6.30MB/23.2MB (27.1%)
2024-07-08T11:01:12.950+0000 [##.....] test.inspections 6.58MB/23.2MB (28.4%)
2024-07-08T11:01:15.909+0000 [##.....] test.inspections 7.16MB/23.2MB (30.8%)
2024-07-08T11:01:18.909+0000 [##.....] test.inspections 7.44MB/23.2MB (32.1%)
2024-07-08T11:01:21.909+0000 [##.....] test.inspections 7.73MB/23.2MB (33.3%)
2024-07-08T11:01:24.919+0000 [##.....] test.inspections 8.02MB/23.2MB (34.5%)
2024-07-08T11:01:27.909+0000 [##.....] test.inspections 8.30MB/23.2MB (35.8%)
2024-07-08T11:01:30.909+0000 [##.....] test.inspections 8.59MB/23.2MB (37.0%)
```

```
vagrant@vagrant: /usr/local/mongodb/mongodb-7.0.12/bin
2024-07-08T11:02:33.923+0000 [#####] test.inspections 14.9MB/23.2MB (64.2%)
2024-07-08T11:02:36.908+0000 [#####] test.inspections 15.2MB/23.2MB (65.4%)
2024-07-08T11:02:39.926+0000 [#####] test.inspections 15.5MB/23.2MB (66.7%)
2024-07-08T11:02:42.915+0000 [#####] test.inspections 15.8MB/23.2MB (67.9%)
2024-07-08T11:02:45.908+0000 [#####] test.inspections 16.0MB/23.2MB (69.1%)
2024-07-08T11:02:48.909+0000 [#####] test.inspections 16.3MB/23.2MB (70.4%)
2024-07-08T11:02:51.919+0000 [#####] test.inspections 16.6MB/23.2MB (71.6%)
2024-07-08T11:02:54.917+0000 [#####] test.inspections 16.9MB/23.2MB (72.8%)
2024-07-08T11:02:57.909+0000 [#####] test.inspections 17.2MB/23.2MB (74.1%)
2024-07-08T11:03:00.913+0000 [#####] test.inspections 17.5MB/23.2MB (75.3%)
2024-07-08T11:03:03.909+0000 [#####] test.inspections 17.8MB/23.2MB (76.5%)
2024-07-08T11:03:06.909+0000 [#####] test.inspections 18.1MB/23.2MB (77.8%)
2024-07-08T11:03:09.910+0000 [#####] test.inspections 18.3MB/23.2MB (79.0%)
2024-07-08T11:03:12.914+0000 [#####] test.inspections 18.6MB/23.2MB (80.3%)
2024-07-08T11:03:15.909+0000 [#####] test.inspections 18.9MB/23.2MB (81.5%)
2024-07-08T11:03:18.908+0000 [#####] test.inspections 19.3MB/23.2MB (83.2%)
2024-07-08T11:03:21.916+0000 [#####] test.inspections 19.8MB/23.2MB (85.2%)
2024-07-08T11:03:24.908+0000 [#####] test.inspections 20.1MB/23.2MB (86.4%)
2024-07-08T11:03:27.929+0000 [#####] test.inspections 20.3MB/23.2MB (87.7%)
2024-07-08T11:03:30.922+0000 [#####] test.inspections 20.6MB/23.2MB (88.9%)
2024-07-08T11:03:33.909+0000 [#####] test.inspections 20.9MB/23.2MB (90.1%)
2024-07-08T11:03:36.916+0000 [#####] test.inspections 21.2MB/23.2MB (91.4%)
2024-07-08T11:03:39.916+0000 [#####] test.inspections 21.5MB/23.2MB (92.6%)
2024-07-08T11:03:42.909+0000 [#####] test.inspections 21.8MB/23.2MB (93.8%)
2024-07-08T11:03:45.918+0000 [#####] test.inspections 22.1MB/23.2MB (95.1%)
2024-07-08T11:03:48.908+0000 [#####] test.inspections 22.6MB/23.2MB (97.2%)
2024-07-08T11:03:51.909+0000 [#####] test.inspections 23.2MB/23.2MB (100.0%)
2024-07-08T11:03:52.767+0000 [#####] test.inspections 23.2MB/23.2MB (100.0%)
2024-07-08T11:03:52.767+0000 81047 document(s) imported successfully. 0 document(s) failed to import.
vagrant@vagrant: /usr/local/mongodb/mongodb-7.0.12/bin$
```

Terminado el proceso de carga de la base de datos, podemos comprobar que se ha subido correctamente, si volvemos a la terminal conectada al master y volvemos a ejecutar la orden `show dbs` deberá aparecer la base de datos `test`. Una vez comprobamos que la base de datos existe podremos trabajar con ella de forma habitual tal como hacíamos en la anterior práctica. Por defecto, la base de datos a la que nos conectamos será `test`, si nos quisieramos conectar a cualquier otra base de datos se debe escribir `use <nombre de la base de datos>`:

```
mongosh mongodb+srv://<credentials>@cluster0.ay4vpe7.m...
Atlas atlas-10cijx-shard-0 [primary] test> show dbs
sample_mflix 113.79 MiB
test          10.96 MiB
admin         280.00 KiB
local         1.70 GiB
Atlas atlas-10cijx-shard-0 [primary] test>
```

Podemos contar el número de documentos o ver un documento:

```
Atlas atlas-10cijx-shard-0 [primary] test> db.inspections.countDocuments()
81047
Atlas atlas-10cijx-shard-0 [primary] test> db.inspections.findOne()
{
  _id: ObjectId('56d61033a378eccde8a83550'),
  id: '10057-2015-ENFO',
  certificate_number: 6007104,
  business_name: 'LD BUSINESS SOLUTIONS',
  date: 'Feb 25 2015',
  result: 'Violation Issued',
  sector: 'Tax Preparers - 891',
  address: {
    city: 'NEW YORK',
    zip: 10030,
    street: 'FREDERICK DOUGLASS BLVD',
    number: 2655
  }
}
Atlas atlas-10cijx-shard-0 [primary] test>
```

Si volvemos a la plataforma web de Mongo ATLAS y pinchamos sobre *Cluster0*:

The screenshot shows the MongoDB Atlas Clusters page for the organization ANGELCLOUDATLAS. The cluster Cluster0 is selected. Key details shown include:

- Version:** 7.0.11
- Region:** AWS Paris (eu-west-3)
- Cluster Tier:** MO Sandbox (General)
- Type:** Replica Set - 3 nodes
- Backups:** Inactive
- Linked App Services:** None Linked
- Atlas SQL:** Connect
- Atlas Search:** Create Index

Metrics and monitoring charts are also visible on the right side of the page.

Y luego sobre *Collections*, sobre la base de datos *test* y sobre la colección *inspections* podremos ver información sobre esta colección, como el número de documentos, el tamaño que ocupa e, incluso, podríamos realizar alguna función de tipo filtrado:

The screenshot shows the MongoDB Atlas Collections page for the database test. The collections test and inspections are listed. The inspections collection is selected, highlighted with a blue oval. The page displays the following information:

- DATABASES:** 2 COLLECTIONS: 7
- test.inspections**
- STORAGE SIZE:** 9.36MB
- LOGICAL DATA SIZE:** 21.39MB
- TOTAL DOCUMENTS:** 8047
- INDEXES TOTAL SIZE:** 2.98MB

A query result table is shown at the bottom, with the first two documents displayed:

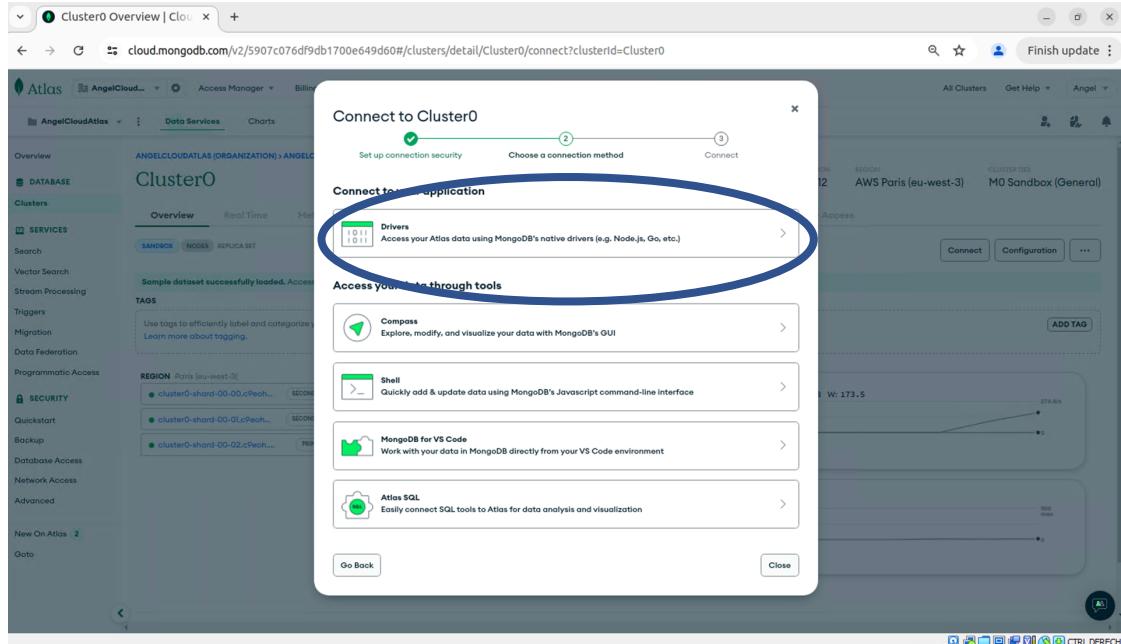
```

_id: ObjectId("56d61033a378eccde8a83551")
_id: "10084-2015-ENFO"
certificate_number: 6007104
business_name: "LD BUSINESS SOLUTIONS"
date : "Feb 25 2015"
result : "Inspection Issues"
sector : "Taxi Repairers = 891"
address : Object

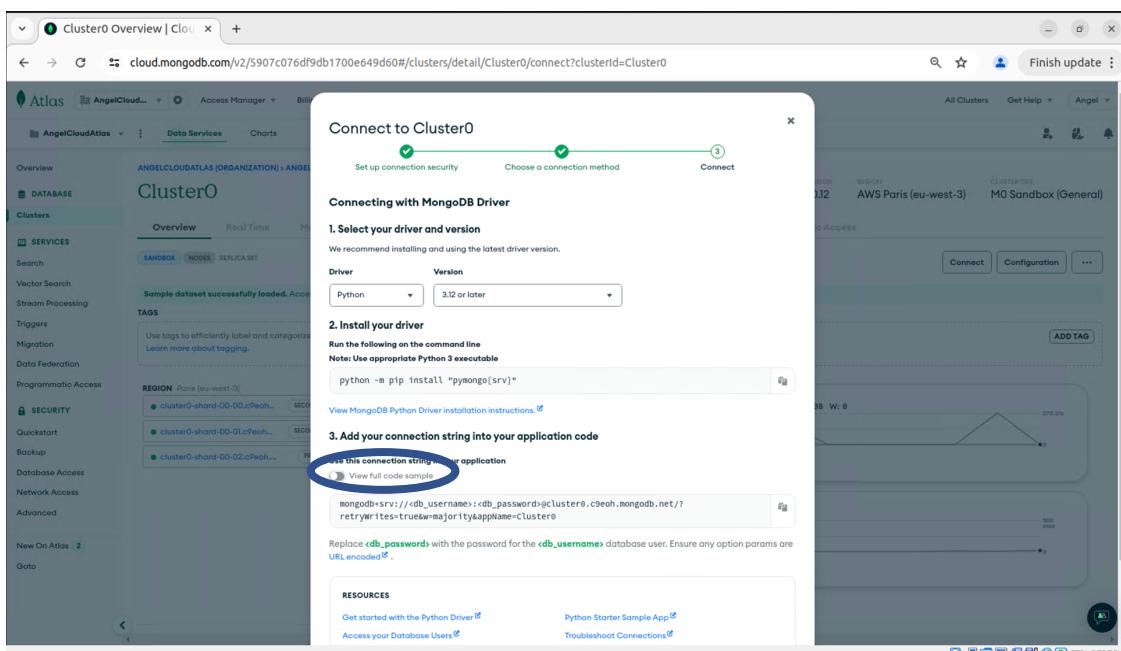
```

Conexión desde Python

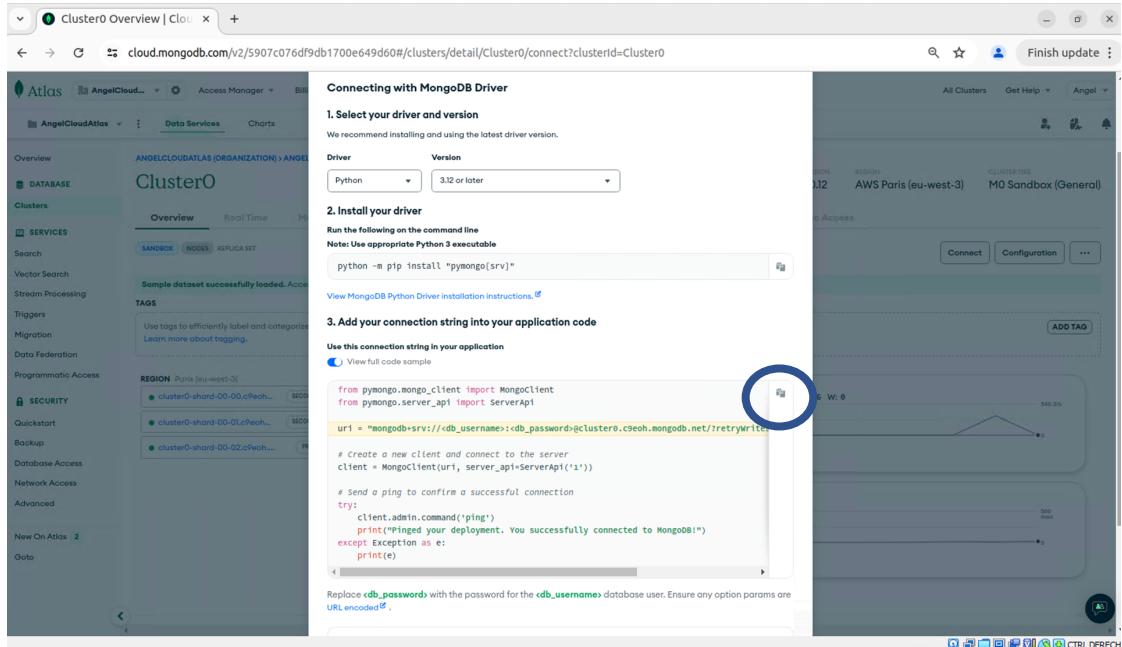
Para conectarnos desde una aplicación desarrollada en Python, pinchamos de nuevo sobre *Connect* y, esta vez sobre el cuadro *Drivers*:



Seleccionamos *Python*, no tenemos que instalar el driver ya que lo tenemos instalado y activamos el botón *view full code sample*:



Copiamos las líneas de código en la pestaña de *Sample Code* para generar la conexión desde Python:

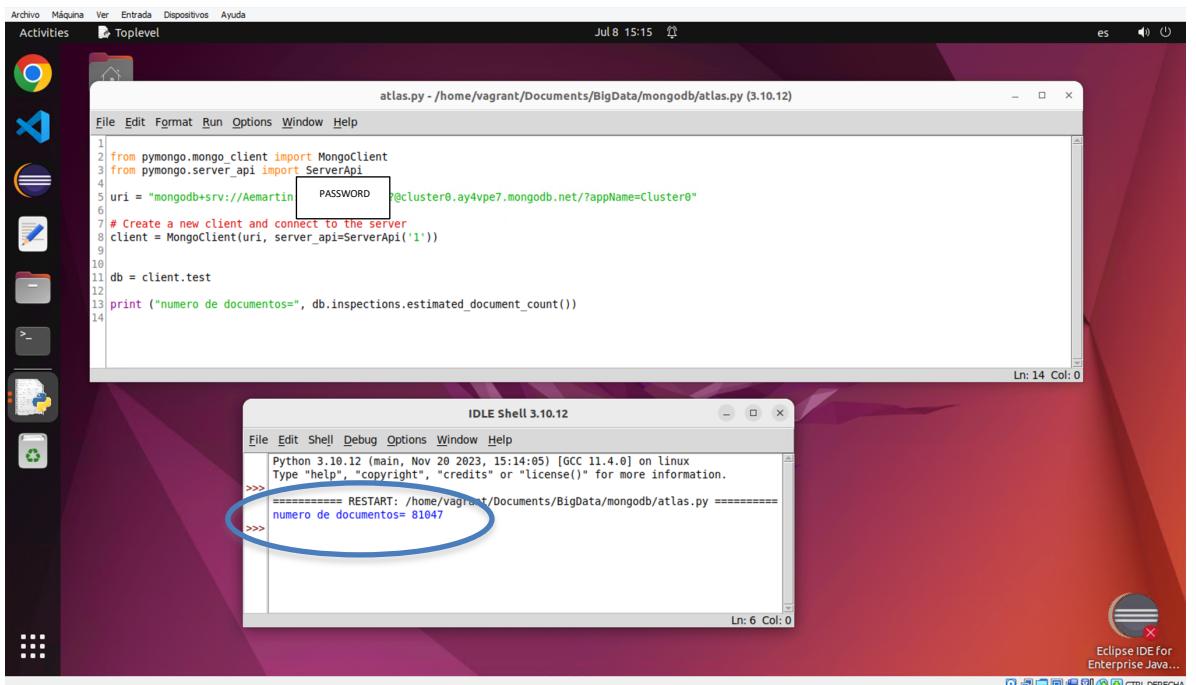


Ahora abrimos *IDLE* y empezamos un programa nuevo en python. En nuestro programa pegaremos las líneas copiadas de Mongo Atlas (podemos eliminar a partir de la línea *try*). Lo que estamos haciendo es asignar la variable *client* a la clase *MongoClient* de la librería, introduciendo como parámetro la dirección SRV. Deberemos sustituir en esa línea <username>:<password> por el nombre de usuario y password que tengamos.

Recordad que nuestra colección se llama *test*, por lo que se asigna esa colección a la variable *db* mediante la orden *db=client.test*, ahora ya estamos conectados y podemos trabajar desde python sobre la base de datos situada en MongoDB Atlas de forma directa.

Para que funcione correctamente recordad que la dirección IP del ordenador desde donde ejecutéis el programa debe estar introducida en MongoDB Atlas, tal como veíamos al hablar de la pestaña security.

Como última línea podremos preguntar por el número de documentos de la colección y, al ejecutar el programa, si todo va bien, en la Shell de Python veremos el número de documentos de la colección:

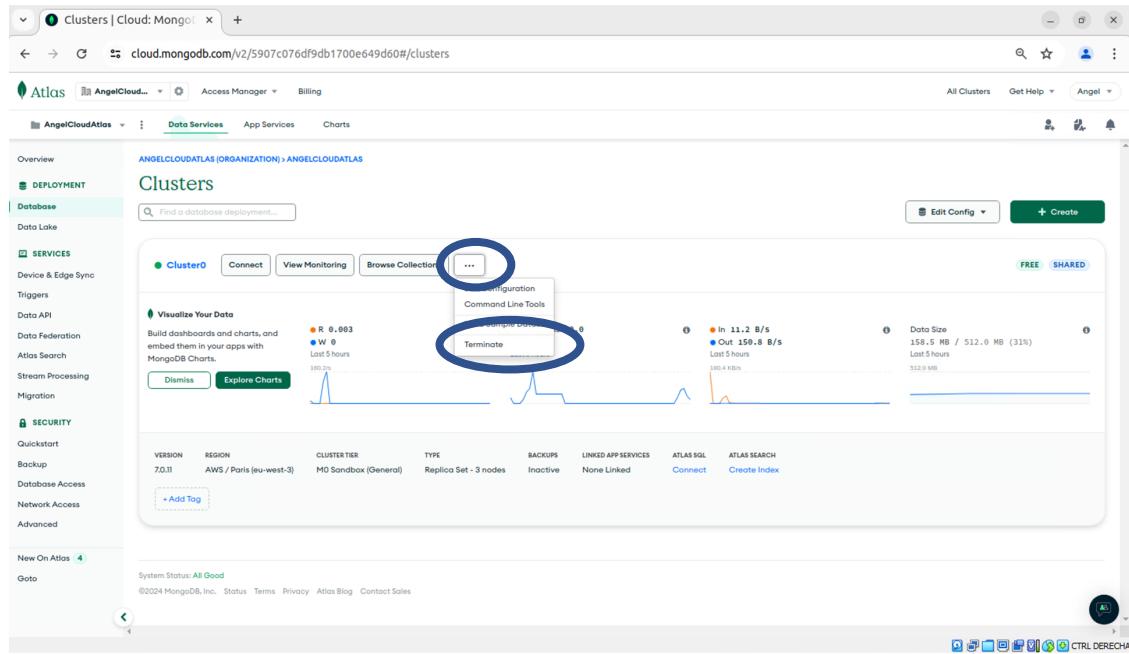


El software funcionará aunque no estemos conectados al Master del clúster, por lo que la conexión desde Python es muy rápida y eficaz.

TERMINACIÓN DEL CLUSTER

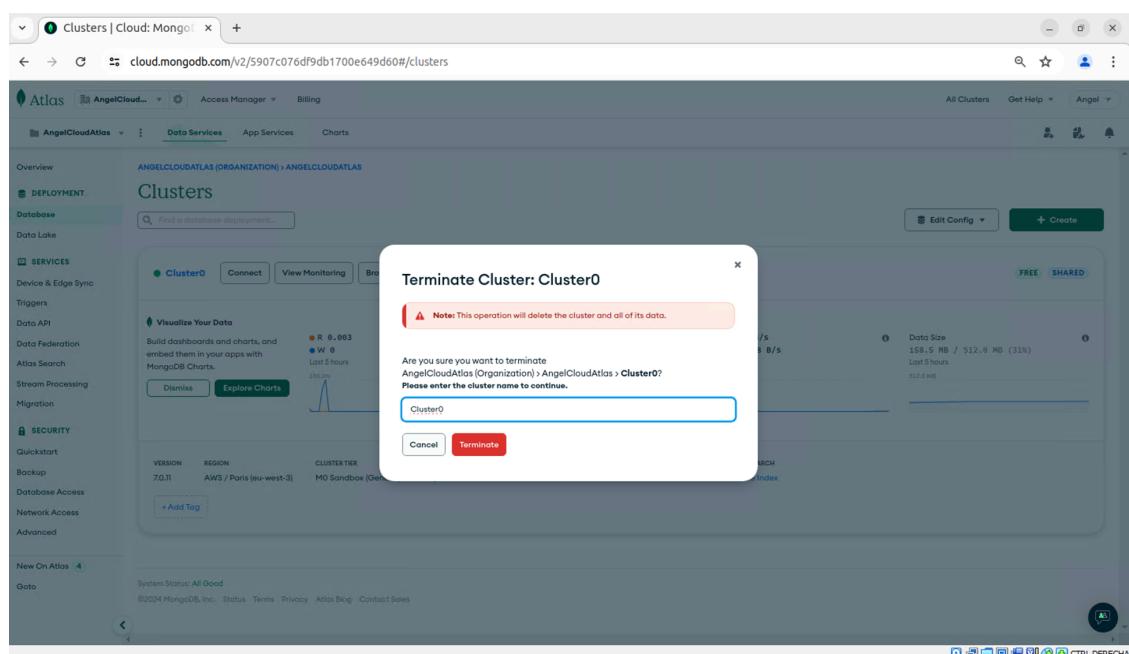
Para terminar un clúster no debes estar conectado a él de forma local, en caso de que lo estés termina la conexión desde la terminal con la orden `quit()`.

Para Terminar el clúster, pincha sobre el botón con los puntos suspensivos, y luego sobre *Terminate* dentro del menú que se te desplegará:



The screenshot shows the MongoDB Cloud Atlas Cluster Overview page. On the left, there's a sidebar with sections like Deployment, Services, and Security. The main area shows a cluster named "Cluster0". Below the cluster name are several monitoring charts and metrics. A callout box highlights the three-dot menu icon next to the cluster name. Another callout box highlights the "Terminate" option within the dropdown menu.

Confirma el nombre del clúster que deseas eliminar y pincha sobre *Terminate*:



The screenshot shows the same MongoDB Cloud Atlas Cluster Overview page as before, but with a modal dialog box in the foreground. The dialog is titled "Terminate Cluster: Cluster0" and contains a warning message: "Note: This operation will delete the cluster and all of its data." It asks "Are you sure you want to terminate AngelCloudAtlas (Organization) > AngelCloudAtlas > Cluster0? Please enter the cluster name to continue." A text input field contains "Cluster0". At the bottom of the dialog are "Cancel" and "Terminate" buttons.

En unos segundos el clúster desplegado se eliminará, liberando todos los recursos que habías adquirido y, por tanto, dejando de añadir gasto a tu factura en caso de que el clúster no fuera gratuito.

The screenshot shows the MongoDB Atlas interface for managing clusters. The left sidebar has sections for Overview, DEPLOYMENT (Database selected), SERVICES, SECURITY, and New On Atlas. The main content area is titled 'Clusters' and shows a single cluster named 'Cluster0'. A message at the top says 'We are deploying your changes: 0 of 3 servers complete (current action: configuring MongoDB)'. Below this, it says 'Your cluster is shutting down.' and 'Metrics are no longer available for this cluster.' The cluster details table includes columns for VERSION (7.0.11), REGION (AWS / Paris (eu-west-3)), CLUSTER TIER (M0 Sandbox (General)), TYPE (Replica Set - 3 nodes), BACKUPS (Inactive), LINKED APP SERVICES (None Linked), and ATLAS SEARCH (Create Index). There are buttons for 'Edit Config' and '+ Create'.

ESCALADO

Si queremos aumentar la memoria y capacidad de nuestra base de datos, podemos escalar tanto de forma vertical como horizontal.

Para eso necesitamos un clúster con mayor capacidad que el clúster gratuito que hemos generado, por lo que, a partir de aquí el desarrollo de la práctica generará gasto (aunque el precio total de la práctica no serán más de cinco o seis euros). **RECORDAR NO DEJAR NUNCA UN CLUSTER ABIERTO YA QUE LE GASTO SE INCREMENTA, NO DEJES ESTA PARTE DE LA PRACTICA A MITAD, SI NO TIENES UNA HORA DE TIEMPO NO LA EMPIECES.**

En primer lugar, debemos añadir los datos de nuestra tarjeta de crédito, para eso pincharemos sobre *billing* en la parte superior de la ventana principal Y luego sobre *Edit* en *Payment Method*:

The screenshot shows the MongoDB Atlas Billing Overview interface. On the left, there's a sidebar with various organization management options like Projects, Alerts, Activity Feed, Settings, Integrations, Access Manager, and Billing (which is currently selected and highlighted in green). The main content area is titled "Billing" and has tabs for "Overview", "Invoices", and "Cost Explorer". Under "Overview", there's a "Preview" section for CSV exports. Below that, it shows the "Running Invoice Total" as \$0.00, the "Billing Period" from 07/01/24 to 08/01/24, and the "Last Billed On" date as 12/01/23. It also displays "Previous Invoice" and "This Time Last Month" both at \$0.00. To the right, there's a "Plan Details" section for the "Atlas" plan, which includes a "Payment Method" field containing a VISA card ending in 2492, set to expire on 07/30 and added on 07/28/23. There are "EDIT" buttons for both the plan details and the payment method. At the bottom, there are sections for "Billing Profile", "Organization Created" (08/24/17), "Billing Email Address" (not added yet), and "Invoice Address" (C/Pla de borta 100, Sagunto, Valencia 46000, Spain, VAT: ESQ4618002B). The bottom right corner features standard browser controls.

Rellenamos los datos de la dirección para el envío de factura y en los datos de la tarjeta de crédito:

The screenshot shows the 'Edit Payment Method' page for the organization 'ANGELCLOUDATLAS'. It displays two main sections: 'Billing Address' and 'Payment Method'.
Under 'Billing Address', there is a checkbox 'Use saved billing address' followed by the address 'C/Pla de barta 100 Sagunto, Valencia 46500'.
Under 'Payment Method', it shows 'Angel Esteban Martin Furones, Credit card (ending in '0747) 10/23' with an 'EDIT' button.
A note at the bottom states: 'By submitting your payment information, you authorize us to charge your card a recurring fee in accordance with your cloud services agreement with MongoDB.'

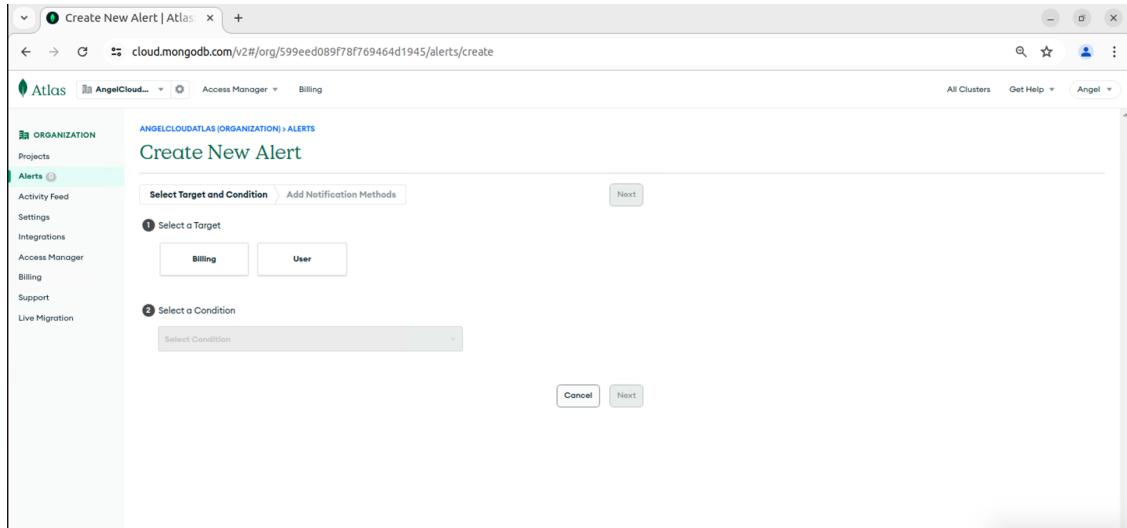
Para controlar el gasto y no llevarnos ninguna sorpresa desagradable, vamos a generar una alarma de pago, de tal manera que cuando nuestra cuenta sobrepase un euro, por ejemplo, el sistema nos envíe un e-mail (se puede configurar para que envíe un SMS también, por ejemplo) de alerta. Para eso pinchamos sobre *Alerts* (en la columna de opciones de la parte izquierda):

The screenshot shows the 'Organization Alerts' page for the organization 'ANGELCLOUDATLAS'. On the left sidebar, the 'Alerts' option is highlighted with a blue oval. The main area displays a table of alerts with columns for 'Timestamp', 'Alert Description', and 'Status'.
The table contains three entries:

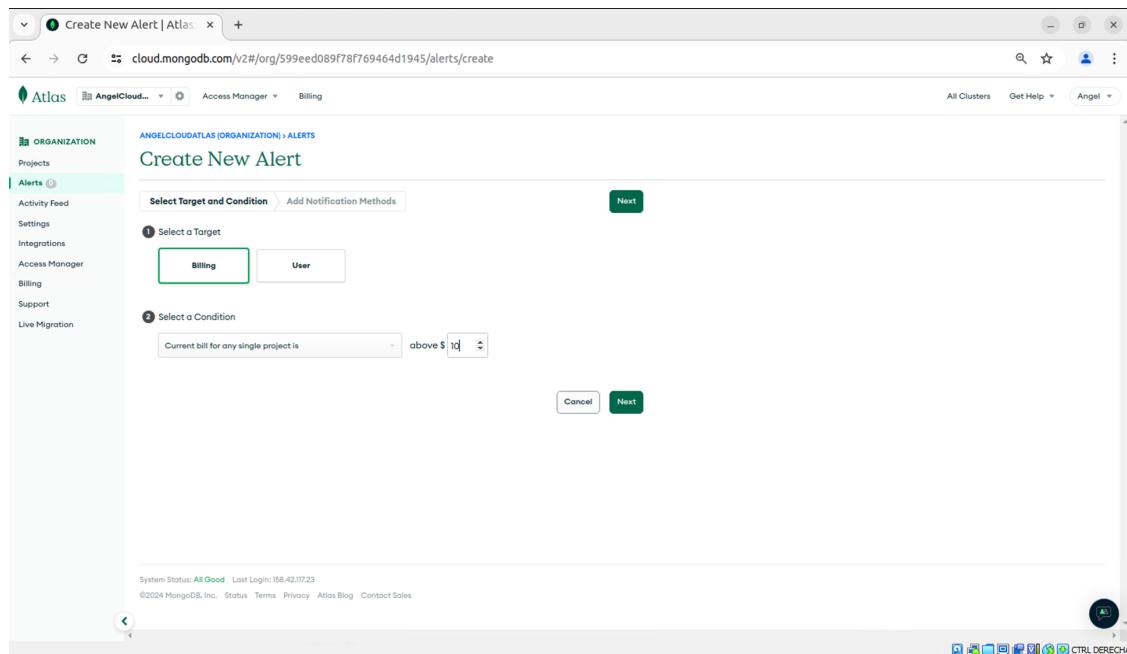
Timestamp	Alert Description	Status
Resolved On: 07/26/22 - 07:35:15 PM Opened On: 07/26/22 - 05:35:17 AM	Current bill for any single project is above the limit you set	✓ Resolved
Resolved On: 09/01/20 - 03:32:42 AM Opened On: 08/11/20 - 03:43:47 AM	Current bill for any single project is above the limit you set	✓ Resolved
Resolved On: 08/01/19 - 03:28:15 AM Opened On: 07/30/19 - 03:27:37 AM	Current bill for any single project is above the limit you set	✓ Resolved

At the bottom, there are links for 'PREVIOUS' and 'NEXT'.

Pinchando sobre *ADD Alert*, aparecerá la siguiente pantalla:



Pinchamos sobre *Billing*, y elegimos *Current bill for any single Project is* dentro del desplegable de condiciones (*Select a condition*), asignando 10 dólares, por ejemplo, a la opción *above*:



Pinchamos sobre *Next* y, en la ventana que nos aparece a continuación, podremos elegir el sistema de aviso que queramos cuando nuestra cuenta sobrepase diez dólares. En este ejemplo elegimos por correo electrónico, escribimos nuestra dirección mail y, además, podemos seleccionar alguna opción de recurrencia en caso de que no desactivemos la alarma una vez se ha activado (por ejemplo, enviar otro mail cada dos horas):

The screenshot shows the 'Create New Alert' page in the MongoDB Atlas web interface. On the left, a sidebar menu is visible with options like 'Organization', 'Projects', 'Alerts' (which is selected), 'Activity Feed', 'Settings', 'Integrations', 'Access Manager', 'Billing', 'Support', and 'Live Migration'. The main content area has a header 'ANGELOLOUDATLAS (ORGANIZATION) > ALERTS' and 'Create New Alert'. Below this, there are two tabs: 'Select Target and Condition' (which is active, indicated by a green checkmark) and 'Add Notification Methods'. A 'Your Alert Preview' section shows 'Target: Billing' and 'Condition: Current bill for any single project is above the limit you set above \$1'. The 'Add Notification Method' section contains a grid of nine items, each with an icon and a brief description:

- Organization: Sends the alert to different org roles via email or SMS.
- User: Sends the alert to a specified user via email or SMS.
- Email Address: Sends the alert to a specified email address.
- Mobile Number: Sends the alert to the specified phone number.
- Slack: Sends the alert to a slack channel.
- PagerDuty: Sends the alert to a PagerDuty account.
- Datadog: Sends the alert to a Datadog account.
- Microsoft Teams: Sends the alert to a Microsoft Teams channel.
- OpsGenie: Sends the alert to an OpsGenie account.
- VictorOps: Sends the alert to a VictorOps account.

This screenshot shows the same 'Create New Alert' interface, but the 'Add Notification Methods' tab is now active. In the 'Your Alert Preview' section, the target is still 'Billing' and the condition is 'Current bill for any single project is above the limit you set above \$1'. The 'Add Notification Method' section is expanded to show a detailed configuration for the 'Email Address' method. It includes fields for 'Email Address' (set to 'gmartin@upvnet.upv.es') and 'Recurrence' (set to 'if the condition lasts more than 60 min, resend alert after 120 min'). At the bottom right of this panel are 'Cancel' and 'Add' buttons. Below the preview panel, there are 'Back', 'Cancel', and 'Create' buttons. The bottom of the page displays system status information: 'System Status: All Good', 'Last Login: 158.42.154.37', and copyright notice: '©2023 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales'.

Pinchamos sobre *Add* (veremos una *preview* de lo que hemos seleccionado), pinchamos en *Create* y veremos nuestra alarma creada en la pestaña de *Alert Settings*.

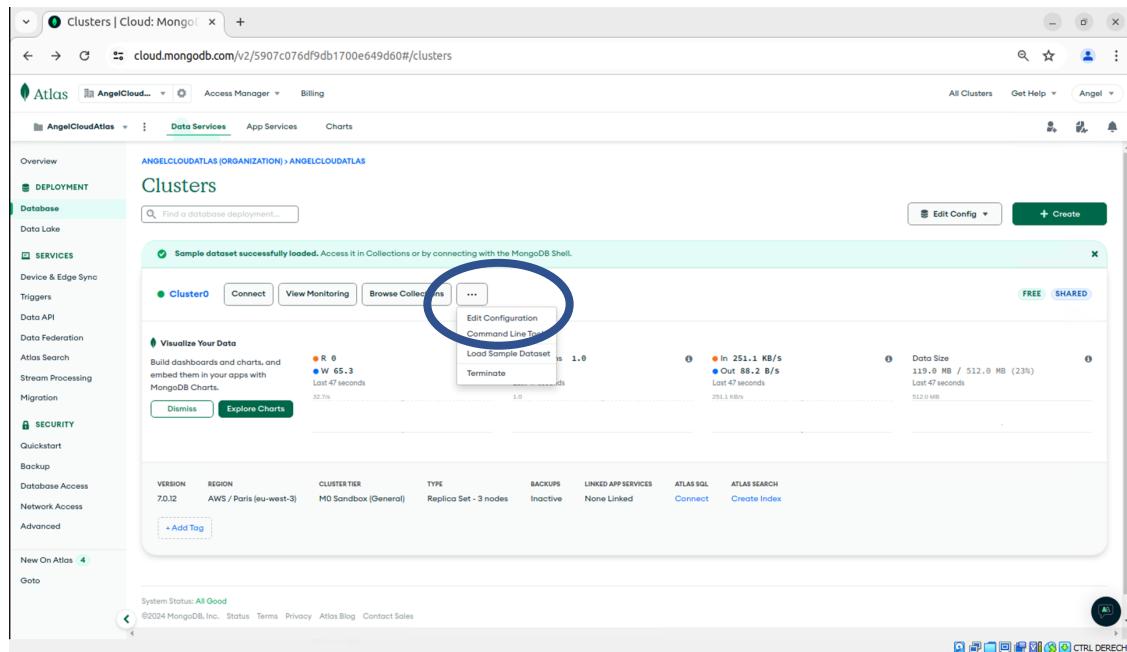
The screenshot shows the 'Organization Alerts' section of the MongoDB Atlas Alert Settings interface. It displays two alert configurations under the 'Billing Alerts' tab. The first alert, 'Current bill for any single project is above \$50', is set to send an email to 'oemartin@upvnet.upv.es' if the condition lasts 60 min, resending every 120 mins. The second alert, 'Current bill for any single project is above \$10', is also set to send an email to the same address with the same resending schedule. Both alerts have a 'Send To' section with an email icon and the recipient's name. A blue oval highlights the second alert configuration.

Para volver al clúster creado podemos hacerlo pinchando sobre *All Clusters*:

The screenshot shows the same 'Organization Alerts' section as the previous one, but with a blue oval highlighting the 'All Clusters' link in the top right corner of the header. This link is used to switch from the organization-level alert settings to the specific cluster-level alert settings.

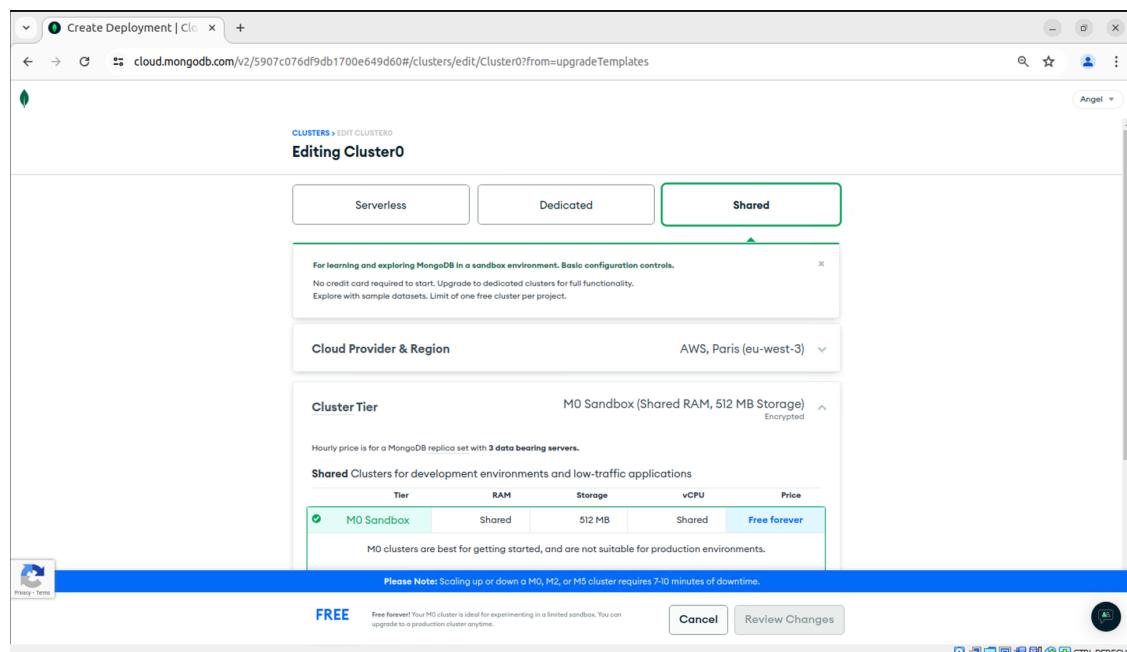
Escalado vertical

A partir de un clúster ya creado (por ejemplo el M0 gratuito como en este ejemplo), si queremos aumentar la memoria y capacidad de la máquina que almacena nuestra base de datos, escalado vertical, debemos pinchar en la pestaña con los tres puntos y elegir edit configuration (**ESTE PASO ES MUY SENCILLO POR LO QUE NO ES NECESARIO HACERLO E INCURRIR EN GASTOS**).



The screenshot shows the MongoDB Cloud Atlas interface. On the left, there's a sidebar with sections like Overview, Deployment, Database, Services, Security, and more. The main area is titled 'Clusters' and shows 'Cluster0'. A context menu is open over Cluster0, with 'Edit Configuration' highlighted and circled in red. The menu also includes options like 'Command Line Tools', 'Load Sample Dataset', and 'Terminate'. Below the cluster details, there's a table with columns like VERSION, REGION, CLUSTER TIER, TYPE, BACKUPS, LINKED APP SERVICES, ATLAS SQL, and ATLAS SEARCH.

Aparecerá la siguiente pantalla:



The screenshot shows the 'Editing Cluster0' configuration page. The 'Shared' tab is selected. It displays basic configuration controls, including a note about learning in a sandbox environment. Below that is a 'Cloud Provider & Region' section set to AWS, Paris (eu-west-3). Under 'Cluster Tier', it shows 'M0 Sandbox (Shared RAM, 512 MB Storage)'. A table for the M0 Sandbox tier is shown with columns: Tier, RAM, Storage, vCPU, and Price. The M0 Sandbox row is highlighted with green borders. A note at the bottom states: 'Please Note: Scaling up or down a M0, M2, or M5 cluster requires 7-10 minutes of downtime.' At the bottom right are 'Cancel' and 'Review Changes' buttons.

Si seleccionamos la ventana *Dedicated* (aparecen más opciones que en la *Shared*) y desplegamos el apartado *Cluster Tier*, la plataforma nos llevará de nuevo a la pestaña de despliegue de las instancias y clústers que vimos a la hora de construir el clúster:

Tier	RAM	Storage	vCPU	Price
M30	8 GB	40 GB	2 vCPUs	from \$0.59/hr
M40 *	16 GB	80 GB	4 vCPUs	from \$1.15/hr
M50 *	32 GB	160 GB	8 vCPUs	from \$2.20/hr
M60 *	64 GB	320 GB	16 vCPUs	from \$4.36/hr
M80 *	128 GB	750 GB	32 vCPUs	from \$8.06/hr
M140	192 GB	1000 GB	48 vCPUs	from \$12.13/hr
M200 *	256 GB	1500 GB	64 vCPUs	from \$16.10/hr
M300 *	384 GB	2000 GB	96 vCPUs	from \$24.11/hr
M400 *	512 GB	3000 GB	64 vCPUs	from \$24.57/hr
M700	768 GB	4096 GB	96 vCPUs	from \$36.49/hr

Please Note: Scaling up or down a M0, M2, or M5 cluster requires 7-10 minutes of downtime.

\$0.09/hour Pay-as-you-go! You will be billed hourly and can terminate your cluster anytime. Excludes variable data transfer, backup, and taxes.

[Cancel](#) [Save Draft](#) [Review Changes](#)

Elegimos el tipo de clúster, M10, por ejemplo, con capacidad de memoria entre 10 y 128 GB:

Cloud Provider & Region: AWS, Paris (eu-west-3)

Cluster Tier: M10 (2 GB RAM, 10 GB Storage)
1,000 IOPS, Unencrypted, Auto-expand Storage

Hourly price is for a MongoDB replica set with 3 data bearing servers.

Dedicated Clusters for development environments and low-traffic applications

Tier	RAM	Storage	vCPU	Price
M10	2 GB	10 GB	2 vCPUs	from \$0.09/hr
Storage	10 GB	128 GB		10 GB
Auto-scale	<input type="checkbox"/> Cluster Tier Scaling View docs	<input checked="" type="checkbox"/> Storage Scaling View docs		
IOPS	1000 IOPS			
Additional Info	1500 max connections Up to 5 Gigabit network performance			

Dedicated Clusters for high-traffic applications and large datasets

\$0.09/hour Pay-as-you-go! You will be billed hourly and can terminate your cluster anytime. Excludes variable data transfer, backup, and taxes.

[Cancel](#) [Save Draft](#) [Review Changes](#)

Pulsamos sobre *Review Changes*, para ver un resumen de lo seleccionado:

Review your changes to AngelCloudAtlas (Organization) >AngelCloudAtlas > Cluster0		
Feature	Original	New
Cluster Tier	AWS M0 General class, Shared RAM, Shared vCPU, 500 connections max 512 MB storage Encrypted storage volumes	AWS M10 General class, 2 GB RAM, 2 vCPUs per node, 1500 connections max 10 GB storage Auto-expand storage enabled 1000 IOPS, Non-provisioned Unencrypted storage volumes
Backup	Disabled	Cloud backup enabled
Node Configuration	3 Electable Nodes (AWS eu-west-3)	3 Electable Nodes (AWS eu-west-3)
Advanced Configuration Options	Minimum Oolog Window: Default (0 Hrs)	Minimum Oolog Window: Default (24 Hrs)
Pricing	Free	\$0.09/hour

En todo momento se nos está informado sobre el precio por hora del servicio. Pinchamos sobre *Apply Changes* y, en unos minutos, la configuración habrá cambiado y podremos seguir trabajando.

Escalado horizontal

Si queremos desplegar un clúster *completo* en mongoDB Atlas, es decir, deseamos replicación y fragmentación de nuestra base de datos, podemos generar el clúster y luego introducir los datos o, a partir de una configuración inicial (por ejemplo, un clúster M0 gratuito), y desde *configuration* (como veíamos en el escalado vertical) pasarlo a un clúster con réplicas y fragmentación.

Vamos a optar por generar, a modo de ejercicio completo, el clúster, cargaremos la información y, por último, la fragmentaremos y replicaremos.

Empieza esta parte de manera que el clúster activo cubra un período de tiempo situado en una hora exacta, si empiezas en una hora y terminas en otra se te cobrarán dos horas. El tiempo estimado para completar esta parte es de 45 minutos.

Vés a la pestaña de *Go to Advance Configuration* en la ventana inicial de creación de un clúster nuevo de Mongo Atlas:

Elegimos la columna *Dedicated*:

Dejamos las opciones por defecto respecto del proveedor cloud (AWS y región Spain eu-south-2).

Elegimos M50 en *Cluster Tier* (a pesar de que se puedan generar clusters con réplicas y fragmentación a partir de una configuración M30 es recomendable usar M50 ya que la

configuración M30 tarda mucho tiempo en configurar las máquinas, tiempo que se te cobrará, ya que ahora no estamos usando la configuración gratuita):

Tier	RAM	Storage	vCPU	Price
M30	8 GB	40 GB	2 vCPUs	from \$0.56/hr
M40 *	16 GB	80 GB	4 vCPUs	from \$1.08/hr
M50 *	32 GB	160 GB	8 vCPUs	from \$2.03/hr
Class	Low CPU	General	Local NVMe SSD	
Storage	160 GB is included in the base price.			
	10 GB	4 TB	100 GB	
Auto-scale	<input checked="" type="checkbox"/> Cluster Tier Scaling View docs		Minimum cluster size M40	Maximum cluster size M60
<input checked="" type="checkbox"/> Storage Scaling	<input type="checkbox"/> Provision IOPS View docs		3000 IOPS	
IOPS	30000 max connections Up to 10 Gigabit network performance			
Additional Info				
M60 *	64 GB	320 GB	16 vCPUs	from \$4.02/hr
M80 *	128 GB	750 GB	32 vCPUs	from \$7.43/hr
M140	192 GB	1000 GB	48 vCPUs	from \$11.09/hr

\$2.03/hour Pay-as-you-go! You will be billed hourly and can terminate your cluster anytime. Excludes variable data transfer, backup, and taxes.

[Privacy + Terms](#) [Cancel](#) [Create Cluster](#)

Dejamos el *Storage* en su valor por defecto. Seguimos bajando con el scroll y pinchamos sobre *Additional Settings* donde vamos a elegir la última versión de MongoDB (7.0 en el momento de escribir estos apuntes, julio de 2024):

Additional Settings MongoDB 7.0, No Backup

Select a Version: MongoDB 7.0

All clusters launch with the WiredTiger™ storage engine.

Turn on Cloud Backup (M2 and up)

Snapshots are taken automatically and stored according to your backup and retention policy. Meets recovery point objective (RPO) of 6 hours by default, configurable down to 1 hour.

You can disable backups at any time. [Learn more about backup options](#)

\$0.14/GB month

Termination Protection

When enabled, prevent any user from accidentally deleting this cluster. Termination protection will need to be disabled before this cluster can be deleted. [Learn more](#).

Advanced Settings

Shard your cluster (M50 and up)

Sharding supports high throughput and large datasets, and can be increased as data requirements grow. Sharded clusters cannot be converted to replica sets.

2 Shards

Looking for more than 100 shards? [Contact MongoDB](#)

Enable Business Intelligence Connector (M10 and up)

The BI Connector allows you to visualize your data on relational business intelligence tools (e.g. Tableau, MicroStrategy, Qlik).

BI Connector **\$16.33**/day for sustained monthly usage or \$93.91/day, up to \$4972/month maximum

Manage your own encryption keys (M10 and up)

By default, all MongoDB Atlas cluster storage and backups are encrypted at rest. Enable this feature to configure an additional layer of encryption using your configured AWS KMS, Google Cloud KMS, or Azure Key Vault.

[More Configuration Options](#)

\$4.60/hour Pay-as-you-go! You will be billed hourly and can terminate your cluster anytime. Excludes variable data transfer, backup, and taxes.

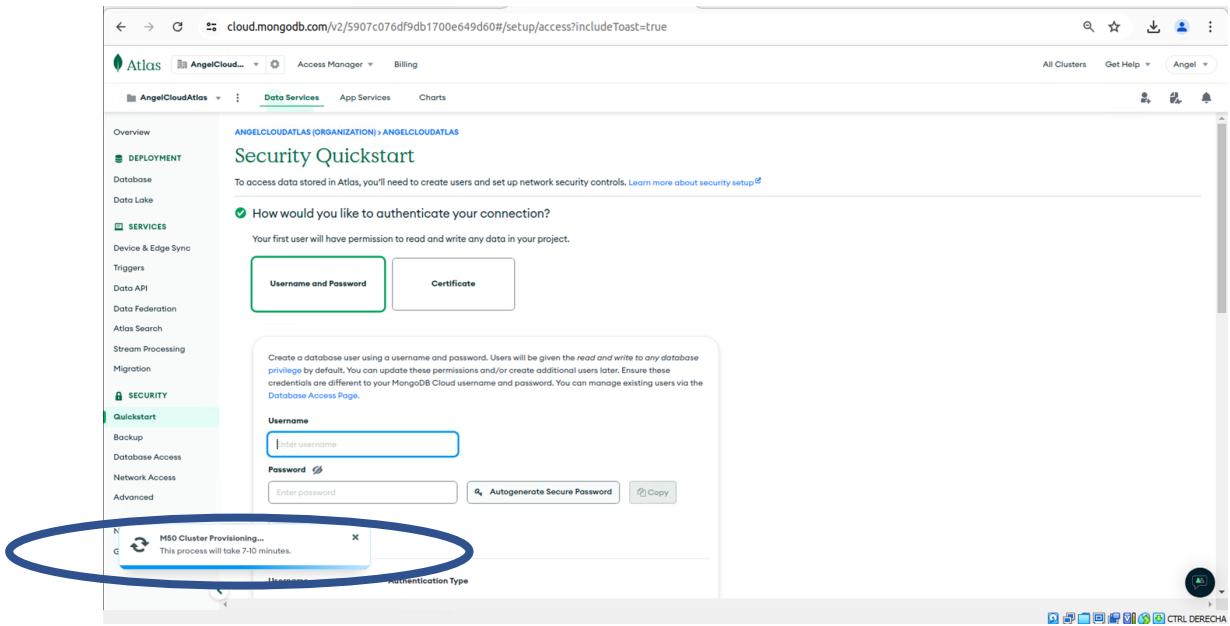
[Privacy + Terms](#) [Cancel](#) [Create Cluster](#)

Dejamos desactivada la opción *Turn on Cloud Backup* (este es un clúster de ejercicio que borraremos inmediatamente, por lo que no necesitamos generar Backup), dejamos desactivada la opción *Termination Protection* por la misma razón y activamos *Shard your Cluster*, dejándolo a dos *Shards*; cada *Shard* es un *Replica Set* compuesto por un máster y dos réplicas que será donde se distribuyan los documentos fragmentados. Por último, dejamos desactivada la opción *Enable Business Intelligence Connector* y bajamos con el *Scroll* hasta la ventana *Cluster Details*, donde ponemos el nombre del clúster que vamos a crear, *ClusterHz*, por ejemplo.

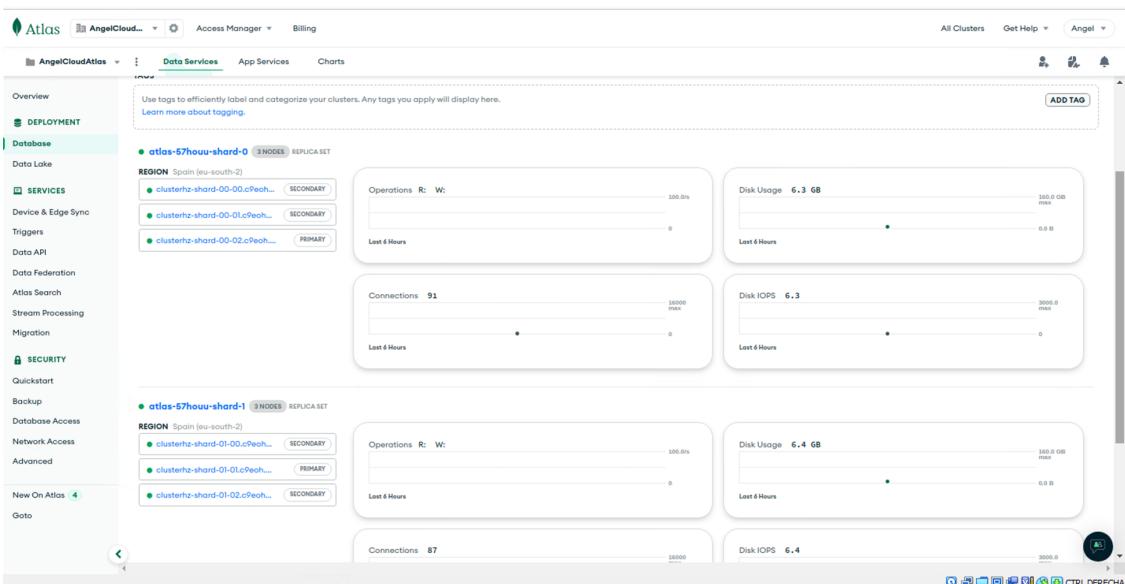
The screenshot shows the MongoDB Cloud interface at the URL cloud.mongodb.com/v2/5907c076df9db1700e649d60#/clusters/edit?filter=advanced. The page is titled 'Manage your own encryption keys (M10 and up)'. It includes sections for 'More Configuration Options', 'Cluster Details' (Cluster Name: ClusterHz, 0 Tags), and 'Apply Tags to Cluster' (NEW). A price section shows '\$4.60/hour' with a note: 'Pay-as-you-go! You will be billed hourly and can terminate your cluster anytime. Exclusive variable: data transfers, backup, and leases.' Buttons for 'Cancel' and 'Create Cluster' are visible. The bottom right corner shows keyboard shortcuts: 'CTRL DERECHA'.

En este ejemplo el clúster M50 cuesta alrededor de 2.03 dólares por hora y shard, es decir, se nos cobrarán 4.06 dólares por hora al tener dos shards. A esto hay que sumar un Replica Set sobre un clúster M30 que se levanta por defecto como clúster de configuración para soportar el clúster M50 distribuido y particionado que estamos generando, el precio de este clúster es de 0.55 dólares por servidor y hora. En total el precio que se nos deberá cobrar será de 4.6 dólares por hora, este precio puede variar dependiendo del proveedor cloud y región elegida.

Pinchamos sobre *Create Cluster* y se nos empezará a desplegar el clúster:



Terminado el proceso deberás ver algo parecido a lo siguiente cuando pinches sobre Database (siempre que hayas elegido ClusterHZ como nombre para el cluster) **A PARTIR DE ESTE MOMENTO ESTAS PAGANDO POR EL CLUSTER Y LO SEGUIRÁS HACIENDO HASTA QUE LO ELIMINES:**



Si nos conectamos al cluster a través de la terminal local (recuerda incluir la IP de tu máquina local en la configuración de acceso a Mongo Atlas si no lo has hecho antes) y usamos la base de datos config, podremos ver las colecciones de documentos que se almacenan:

```

mongosh mongodb+srv://<credentials>@clusterhz.c9eoh.mongodb.net/<dbname>?retryWrites=true&w=majority
Connected to clusterhz.c9eoh.mongodb.net:27017
Atlas [mongos] test> use config
switched to db config
Atlas [mongos] config> show collections
clusterParameters
collections
external_validation_keys
image_collection
mongos
placementHistory
rangeDeletions
reshardingOperations
sampledQueries
sampledQueriesDiff
settings
sharding_configsrv_coordinators
shards
tags
tenantMigrationDonors
tenantMigrationRecipients
transactions
version
system.indexBuilds
system.premises
Atlas [mongos] config>

```

Your inputs:

your command line

/clusterhz.c9eoh.mongodb.net/* --apiVersion 1 --username <username> --password <password>

password for the Database User: **Aemartin**. When entering your password, make sure it is URL encoded.

I have installed MongoDB Shell

To check your MongoDB Shell version, run: mongo --version

Close

En la colección mongos podemos ver las seis máquinas que se han levantado para configurar los dos *Shards* al ejecutar la orden `db.mongos.find({}, {_id:1})`:

```

mongosh mongodb+srv://<credentials>@clusterhz.c9eoh.mongodb.net/<dbname>?retryWrites=true&w=majority
Connected to clusterhz.c9eoh.mongodb.net:27017
Atlas [mongos] config> db.mongos.find({}, {_id:1})
[{"_id": "atlas-57houu-shard-01-00.c9eoh.mongodb.net:27016"}, {"_id": "atlas-57houu-shard-00-02.c9eoh.mongodb.net:27016"}, {"_id": "atlas-57houu-shard-00-01.c9eoh.mongodb.net:27016"}, {"_id": "atlas-57houu-shard-00-00.c9eoh.mongodb.net:27016"}, {"_id": "atlas-57houu-shard-01-02.c9eoh.mongodb.net:27016"}, {"_id": "atlas-57houu-shard-01-01.c9eoh.mongodb.net:27016"}]
Atlas [mongos] config>

```

Your inputs:

your command line

/clusterhz.c9eoh.mongodb.net/* --apiVersion 1 --username <username> --password <password>

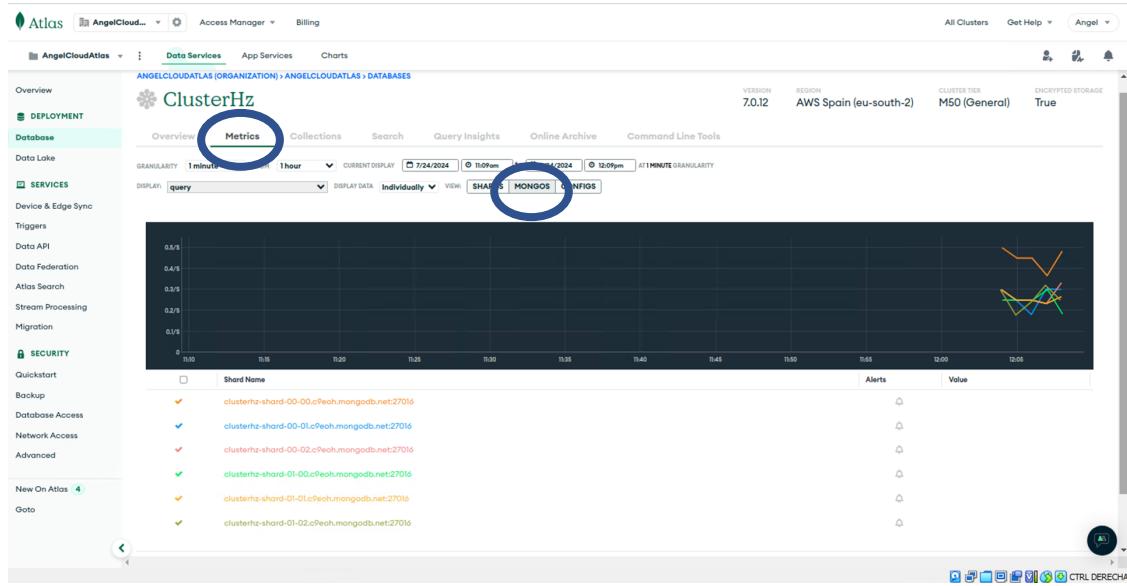
password for the Database User: **Aemartin**. When entering your password, make sure it is URL encoded.

I have installed MongoDB Shell

To check your MongoDB Shell version, run: mongo --version

Close

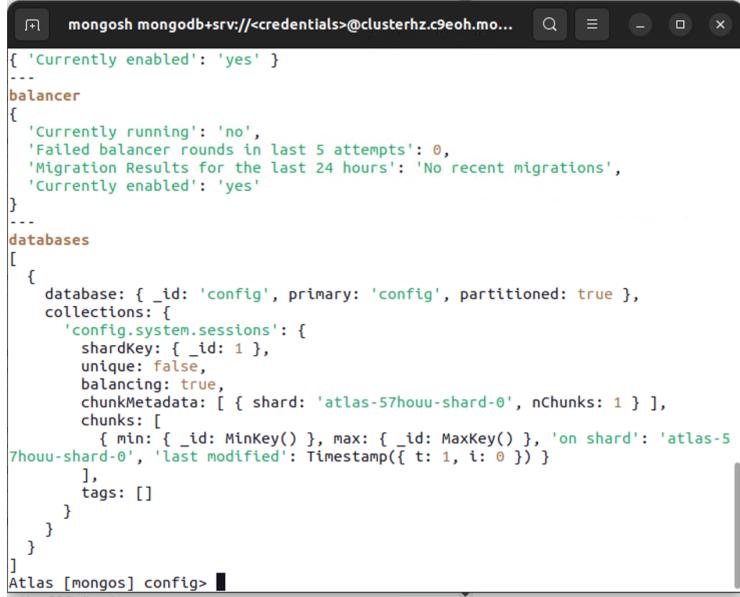
Esta información también la podemos obtener desde la plataforma Atlas, pinchando el nombre del clúster en azul (clusterHZ en este caso) y luego sobre *Metrics* y seleccionando **MONGOS**:



Escribiendo la orden `sh.status()` desde la terminal conectada, podremos ver el estado del clúster que hemos creado, por ejemplo, vemos información sobre los *shards*:

```
Atlas [mongos] config> sh.status()
shardingVersion
{ _id: 1, clusterId: ObjectId('66a0d11da6a454e16ae54f42') }
---
shards
[
  {
    _id: 'atlas-57houu-shard-0',
    host: 'atlas-57houu-shard-0@atlas-57houu-shard-0-00-00.c9eoh.mongodb.net:27017,atlas-57houu-shard-0-01.c9eoh.mongodb.net:27017,atlas-57houu-shard-0-02.c9eoh.mongodb.net:27017',
    state: 1,
    topologyTime: Timestamp({ t: 1721815335, i: 1 })
  },
  {
    _id: 'atlas-57houu-shard-1',
    host: 'atlas-57houu-shard-1@atlas-57houu-shard-01-00.c9eoh.mongodb.net:27017,atlas-57houu-shard-01-01.c9eoh.mongodb.net:27017,atlas-57houu-shard-01-02.c9eoh.mongodb.net:27017',
    state: 1,
    topologyTime: Timestamp({ t: 1721815335, i: 5 })
  }
]
---
active mongoses
[ { '7.0.12': 6 } ]
---
autosplit
{ 'Currently enabled': 'yes' }
---
balancer
```

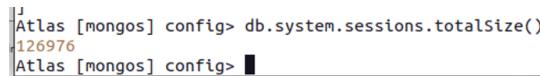
Mongo Atlas genera automáticamente la base de datos `config.system.sessions`, donde guarda todos los elementos de configuración. En `databases` se puede ver el nombre del *shard* donde se ha guardado esta base de datos, dentro de él, el número de *chunks* (o montón de documentos), que se generan para guardar toda la información de configuración. Este número vendrá definido por el tamaño de los *chunks* que, por defecto, es de 128 MB, suficiente para la información de configuración. Si se ejecuta la orden `db.system.sessions.totalSize()`, se puede ver el tamaño que ocupa esta base de datos. Esta base de datos es de configuración, por lo que no se puede tener acceso a ella ni como usuario ni como administrador.



```

mongosh mongodb+srv://<credentials>@clusterhz.c9eoh.mo...
{
  'Currently enabled': 'yes'
}
...
balancer
{
  'Currently running': 'no',
  'Failed balancer rounds in last 5 attempts': 0,
  'Migration Results for the last 24 hours': 'No recent migrations',
  'Currently enabled': 'yes'
}
...
databases
[
  {
    database: { _id: 'config', primary: 'config', partitioned: true },
    collections: [
      'config.system.sessions': {
        shardKey: { _id: 1 },
        unique: false,
        balancing: true,
        chunkMetadata: [ { shard: 'atlas-57houu-shard-0', nChunks: 1 } ],
        chunks: [
          { min: { _id: MinKey() }, max: { _id: MaxKey() }, 'on shard': 'atlas-57houu-shard-0', 'last modified': Timestamp({ t: 1, i: 0 }) }
        ],
        tags: []
      }
    ]
  }
]
Atlas [mongos] config>

```

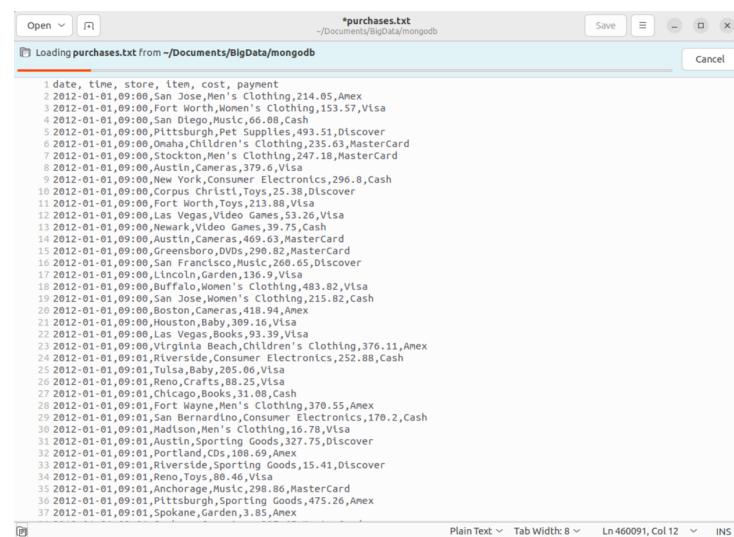


```

Atlas [mongos] config> db.system.sessions.totalSize()
126976
Atlas [mongos] config>

```

A continuación, cargaremos la base de datos *purchases.txt* (disponible en PoliformaT lessons/practicas/bases de datos NoSQL) sobre el clúster que hemos creado, se trata de una base de datos que contiene información sobre las ventas de una determinada cadena comercial de EEUU durante el año 2012, cada línea del fichero es una venta donde se pude ver la fecha, hora, tienda (ciudad), producto, coste y forma de pago:



```

*purchases.txt
-/Documents/BigData/mongodb
Loading purchases.txt from ~/Documents/BigData/mongodb
Cancel

1 date, time, store, item, cost, payment
2 2012-01-01,09:00,San Jose,Men's Clothing,214.05,Amex
3 2012-01-01,09:00,Fort Worth,Women's Clothing,153.57,Visa
4 2012-01-01,09:00,Chicago,Clothing,145.25,Discover
5 2012-01-01,09:00,Pittsburgh,Pet Supplies,493.51,Discover
6 2012-01-01,09:00,Omaha,Children's Clothing,235.63,MasterCard
7 2012-01-01,09:00,Stockton,Men's Clothing,247.18,MasterCard
8 2012-01-01,09:00,New York,Consumer Electronics,290.8,Cash
9 2012-01-01,09:00,Corpus Christi,Toys,213.89,Discover
10 2012-01-01,09:00,Fort Worth,Toys,213.88,Visa
11 2012-01-01,09:00,Las Vegas,Video Games,53.26,Visa
12 2012-01-01,09:00,Newark,Video Games,39.75,Cash
13 2012-01-01,09:00,Newark,Video Games,39.75,Cash
14 2012-01-01,09:00,Austin,Cameras,469.63,MasterCard
15 2012-01-01,09:00,Greenboro,Clothes,290.82,MasterCard
16 2012-01-01,09:00,Portland,Clothes,290.82,Discover
17 2012-01-01,09:00,Lincoln,Garden,136.9,Visa
18 2012-01-01,09:00,Buffalo,Women's Clothing,483.82,Visa
19 2012-01-01,09:00,San Jose,Women's Clothing,215.82,Cash
20 2012-01-01,09:00,Boston,Cameras,418.94,Amex
21 2012-01-01,09:00,Houston,Baby,309.16,Visa
22 2012-01-01,09:00,Las Vegas,Books,93.39,Visa
23 2012-01-01,09:00,Riverside,Women's Clothing,376.11,Amex
24 2012-01-01,09:01,Riverside,Consumer Electronics,252.88,Cash
25 2012-01-01,09:01,Tulsa,Baby,205.66,Visa
26 2012-01-01,09:01,Reno,Crafts,88.25,Visa
27 2012-01-01,09:01,Chicago,Books,31.08,Cash
28 2012-01-01,09:01,Fort Wayne,Men's Clothing,370.55,Amex
29 2012-01-01,09:01,Portland,Clothes,290.82,Consumer Electronics,170.2,Cash
30 2012-01-01,09:01,Madison,Men's Clothing,16.78,Visa
31 2012-01-01,09:01,Austin,Sporting Goods,327.75,Discover
32 2012-01-01,09:01,Portland,CDS,109.69,Amex
33 2012-01-01,09:01,Riverside,Sporting Goods,15.41,Discover
34 2012-01-01,09:01,Reno,Toys,80.46,Visa
35 2012-01-01,09:01,Anchorage,Music,298.86,MasterCard
36 2012-01-01,09:01,Pittsburgh,Sporting Goods,475.26,Amex
37 2012-01-01,09:01,Spokane,Garden,3.85,Amex

```

Para la carga generaremos la base de datos *test* y sobre ella crearemos la colección *purchases* con los documentos del fichero .txt, que, a efectos de carga, se puede tratar como un fichero .csv

considerando la primera línea como la de cabecera con la orden `--headerline`. El procedimiento para cargar la base de datos lo hemos visto con en la práctica anterior.

```
vagrant@vagrant:/usr/local/mongodb/mongodb-7.0.12/bin$ ./mongoimport --uri mongodb+srv://Aemartin:clusterhz.c9eh.mongodb.net/test --collection purchases --type csv --file /home/vagrant/Documentos/Bigdata/mongodb/purchases.txt --headerline
2024-07-24T10:14:14:14.386+0000    connected to: mongodb+srv://Aemartin:H0sta1?megtron?@clusterhz.c9eh.mongodb.net/test
2024-07-24T10:14:17.402+0000      [.....] test.purchases      1.22MB/202MB (0.6%)
2024-07-24T10:14:20.408+0000      [.....] test.purchases      3.36MB/202MB (1.7%)
2024-07-24T10:14:23.387+0000      [.....] test.purchases      5.85MB/202MB (2.9%)
2024-07-24T10:14:26.387+0000      [.....] test.purchases      8.38MB/202MB (4.2%)
2024-07-24T10:14:29.391+0000      [#.....] test.purchases      10.9MB/202MB (5.4%)
2024-07-24T10:14:32.389+0000      [#.....] test.purchases      13.3MB/202MB (6.6%)
2024-07-24T10:14:35.388+0000      [#.....] test.purchases      15.8MB/202MB (7.9%)
2024-07-24T10:14:38.387+0000      [#.....] test.purchases      18.6MB/202MB (9.2%)
2024-07-24T10:14:41.387+0000      [##.....] test.purchases      21.0MB/202MB (10.4%)
2024-07-24T10:14:44.390+0000      [##.....] test.purchases      23.5MB/202MB (11.6%)
2024-07-24T10:14:47.387+0000      [###.....] test.purchases      26.0MB/202MB (12.9%)
2024-07-24T10:14:50.388+0000      [###.....] test.purchases      28.4MB/202MB (14.1%)
2024-07-24T10:14:53.392+0000      [###.....] test.purchases      30.9MB/202MB (15.3%)
2024-07-24T10:14:56.395+0000      [###.....] test.purchases      33.4MB/202MB (16.6%)
2024-07-24T10:14:59.390+0000      [####.....] test.purchases      35.8MB/202MB (17.8%)
2024-07-24T10:15:02.396+0000      [####.....] test.purchases      38.1MB/202MB (18.9%)
2024-07-24T10:15:05.387+0000      [####.....] test.purchases      40.6MB/202MB (20.2%)
2024-07-24T10:15:08.387+0000      [####.....] test.purchases      43.0MB/202MB (21.3%)
```

Una vez cargada podemos comprobar que se ha cargado de forma correcta conectándonos a ella y comprobando que consta de 4138476 documentos o viendo alguno de los documentos con la orden `findOne()`:

```
'Atlas [mongos] config> show dbs
admin   232.00 KiB
config   1.99 MiB
test    273.52 MiB
Atlas [mongos] config> use test
switched to db test
Atlas [mongos] test> show collections
purchases
Atlas [mongos] test> db.purchases.countDocuments()
4138476
Atlas [mongos] test> db.purchases.findOne()
{
  _id: ObjectId('66a0d3f6e39c59a4dd52ebc0'),
  date: '2012-01-01',
  time: '09:00',
  store: 'San Jose',
  item: "Men's Clothing",
  cost: 214.05,
  payment: 'Amex'
}
Atlas [mongos] test> ■
```

Si ejecutamos ahora `sh.status()`, vemos que se ha cargado la base de datos test en el *Shard-1* pero que se encuentra en modo `partitioned:False`,

```

mongosh mongodb+srv://<credentials>@clusterhz.c9eh.mongodb.net/
...
databases
[
  {
    database: { _id: 'config', primary: 'config', partitioned: true },
    collections: [
      'config.system.sessions': {
        shardKey: { _id: 1 },
        unique: false,
        balancing: true,
        chunkMetadata: [ { shard: 'atlas-57houu-shard-0', nChunks: 1 } ],
        chunks: [
          { min: { _id: MinKey() }, max: { _id: MaxKey() }, 'on shard': 'atlas-57houu-shard-0', 'last modified': Timestamp({ t: 1, i: 0 }) }
        ],
        tags: []
      }
    ]
  },
  {
    database: {
      _id: 'test',
      primary: 'atlas-57houu-shard-1',
      partitioned: false
    }
  }
]
Atlas [mongos] test>

```

Para particionar y distribuir las colecciones que se encuentran en la base de datos *test* debemos fijar algún índice como clave para la fragmentación (la decisión sobre este índice es muy importante, hay que tener en cuenta que la división de los valores que haya en este campo que asignamos como índice será la que use el sistema para generar los *chunks*), en este caso vamos a usar el campo *store* como índice, para ello nos situaremos sobre la base de datos *test* y ejecutaremos la orden *db.purchases.createIndex({store:1})*:

```

Atlas [mongos] test> db.purchases.createIndex({store:1})
store_1
Atlas [mongos] test> db.purchases.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id' },
  { v: 2, key: { store: 1 }, name: 'store_1' }
]
Atlas [mongos] test>

```

A continuación ejecutaremos la orden *sh.shardCollection("test.purchases", {store:1})* para que se produzca la distribución (y replicación) de la colección de documentos.

```

Atlas [mongos] test> sh.shardCollection("test.purchases", {store:1})
{
  collectionSharded: 'test.purchases',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1721816522, i: 34 }),
    signature: {
      hash: Binary.createFromBase64('CW0ezQK5UsBoHm++qztE6EEcIY0='),
      keyId: Long('7395140510626611222')
    }
  },
  operationTime: Timestamp({ t: 1721816522, i: 34 })
}
Atlas [mongos] test>

```

Si ejecutamos `sh.status()`, en primer lugar, vemos que la opción *partitioned* ha desaparecido y, en su lugar, la opción *unique* se encuentra en *False* y la opción *balancing* se encuentra en *True*. La base de datos, poco a poco, se distribuirá de forma balanceada entre los shards en un número predefinido de *chunks* que dependerá del tamaño de los mismos que, por defecto, es de 128 MB (si se sobrepasan se generan tantos como sea el resultado de la división entre el tamaño inicial y 128 quedando algún chunk por debajo de este valor para completar la base de datos), y, como es este caso, dentro de cada *chuncks* se reparten los documentos dependiendo del índice que se ha creado: los documentos se han repartido por grupos siguiendo un orden alfabético en la división de la información. En este caso todos los documentos caben en dos *chunks* (el tamaño total de la base de datos en MongoDB es poco más de 213 MB), uno en cada *shard*:

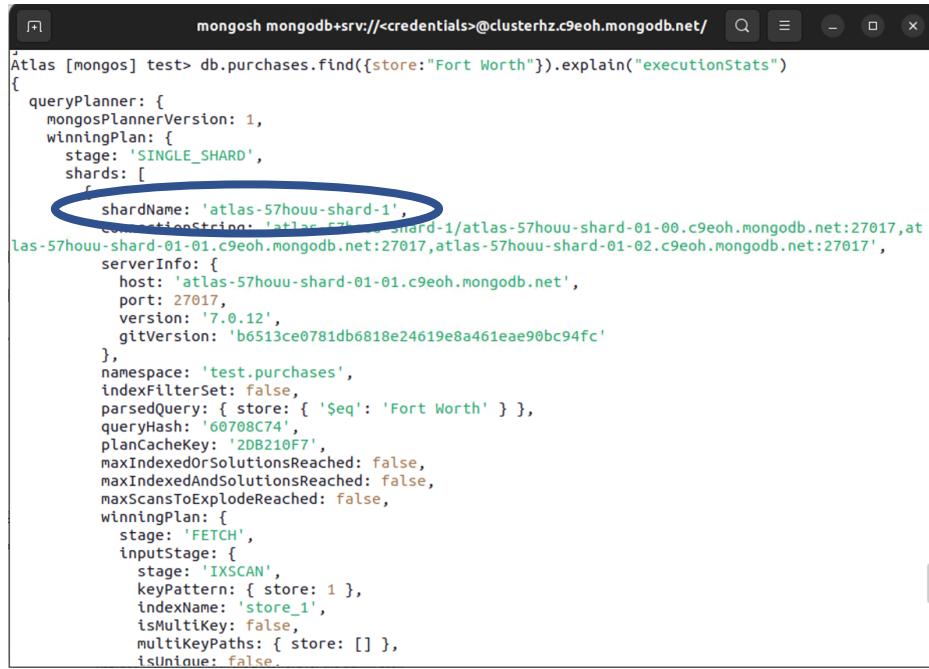


```

mongosh mongodb+srv://<credentials>@clusterhz.c9eh.mongodb.net/
{
  database: {
    _id: 'test',
    primary: 'atlas-57houu-shard-1',
    partitioned: false,
    version: {
      uuid: UUID('909616b6-e1c8-44b0-b3d7-249cfcd10aa4'),
      timestamp: Timestamp({ t: 1721816053, i: 1 }),
      lastMod: 1
    }
  },
  collections: [
    'test.purchases': {
      shardKey: { store: 1 },
      unique: false, // This line is circled in red
      balancing: true
      chunkMetadata: [
        { shard: 'atlas-57houu-shard-0', nChunks: 1 },
        { shard: 'atlas-57houu-shard-1', nChunks: 1 }
      ],
      chunks: [
        { min: { store: MinKey() }, max: { store: 'Dallas' }, 'on shard': 'atlas-57houu-shard-0', 'last modified': Timestamp({ t: 2, i: 0 }) },
        { min: { store: 'Dallas' }, max: { store: MaxKey() }, 'on shard': 'atlas-57houu-shard-1', 'last modified': Timestamp({ t: 2, i: 1 }) }
      ],
      tags: []
    }
  ]
}
Atlas [mongos] test>

```

Podemos buscar el documento cuya *store* es “Forth Worth” junto con la opción `explain("executionStats")`, de manera que se nos mostrará la información sobre el *Shard* del que se ha leído la correspondiente consulta:

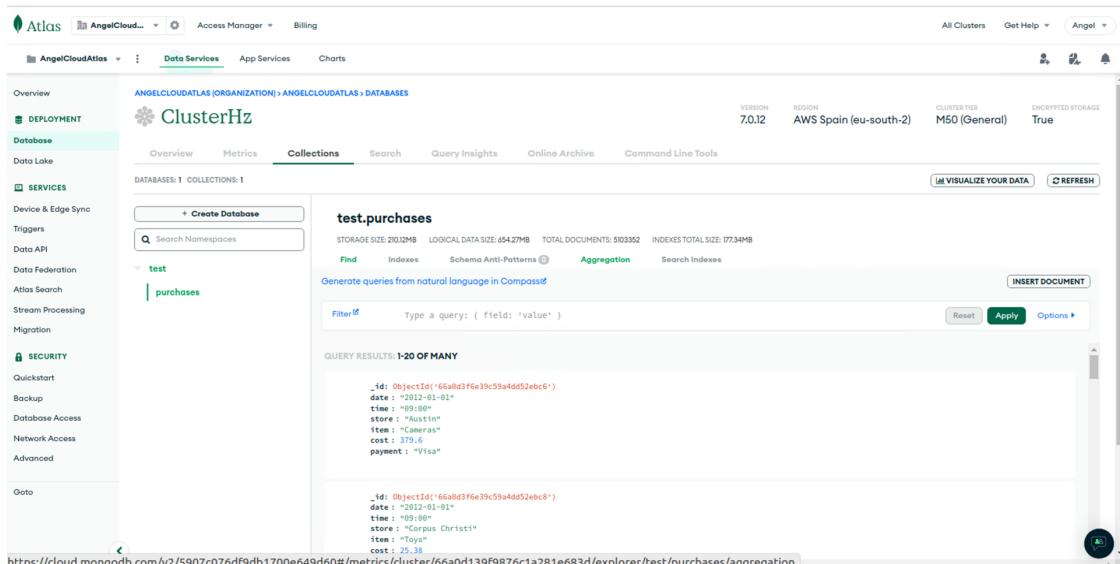


```

mongosh mongodb+srv://<credentials>@clusterhz.c9eh.mongodb.net/
Atlas [Mongos] test> db.purchases.find({store:"Fort Worth"}).explain("executionStats")
{
  queryPlanner: {
    mongosPlannerVersion: 1,
    winningPlan: {
      stage: 'SINGLE_SHARD',
      shards: [
        {
          shardName: 'atlas-57houu-shard-1',
          winningString: 'atlas-57houu-shard-1/atlas-57houu-shard-01-00.c9eh.mongodb.net:27017,atlas-57houu-shard-01-01.c9eh.mongodb.net:27017,atlas-57houu-shard-01-02.c9eh.mongodb.net:27017',
          serverInfo: {
            host: 'atlas-57houu-shard-01-01.c9eh.mongodb.net',
            port: 27017,
            version: '7.0.12',
            gitVersion: 'b6513ce0781db6818e24619e8a461eae90bc94fc'
          },
          namespace: 'test.purchases',
          indexFilterSet: false,
          parsedQuery: { store: { '$eq': 'Fort Worth' } },
          queryHash: '60708C74',
          planCacheKey: '2DB210F7',
          maxIndexedOrSolutionsReached: false,
          maxIndexedAndSolutionsReached: false,
          maxScansToExplodeReached: false,
          winningPlan: {
            stage: 'FETCH',
            inputStage: {
              stage: 'IXSCAN',
              keyPattern: { store: 1 },
              indexName: 'store_1',
              isMultiKey: false,
              multiKeyPaths: { store: [] },
              isUnique: false
            }
          }
        }
      ]
    }
  }
}

```

Para finalizar, a modo de ejemplo, podemos ver un documento dentro de la plataforma *mongo Atlas*:



The screenshot shows the MongoDB Atlas web interface. On the left, there's a sidebar with sections like Overview, Deployment, Database, Services, and Security. The main area shows the 'ClusterHz' cluster details: Version 7.0.12, Region AWS Spain (eu-south-2), Cluster Tier M50 (General), and Encrypted Storage True. Below this, the 'Collections' tab is selected for the 'test' database, showing the 'test.purchases' collection. The collection details indicate a storage size of 201.2MB, logical data size of 654.27MB, total documents of 510352, and indexes total size of 177.54MB. A search bar at the top of the collection view shows a query: '_id: ObjectId('66a0d3f6e39c59a4dd52ebc6')'. The results pane shows two documents:

```

{
  "_id": "66a0d3f6e39c59a4dd52ebc6",
  "date": "2012-01-01",
  "time": "09:00",
  "store": "Austin",
  "item": "Cameras",
  "cost": 379.6,
  "payment": "VISA"
}

{
  "_id": "66a0d3f6e39c59a4dd52ebc8",
  "date": "2012-01-01",
  "time": "09:00",
  "store": "Corpus Christi",
  "item": "Toys",
  "cost": 35.1
}

```

Con esto tendremos un clúster que fragmenta y distribuye los documentos de nuestra base de datos. El funcionamiento de la base de datos a nivel de actualizaciones, consultas, etc., es exactamente igual que cuando se trabaja con MongoDB de forma local, por lo que damos por finalizada la práctica. **NO OLVIDES TERMINAR EL CLUSTER, EN CASO CONTRARIO SEGUIRÁ ACTIVO Y TU FACTURA CRECERÁ.**