

IMPORT CSV FILE USING MONGOIMPORT IN MONGODB ATLAS

Version: 0.1(20240201)

Author: Ing. Duván Mejia Cortes

Objective:

In this section of the tutorial, we will demonstrate how to upload a document to the MongoDB NoSQL database.

Software Requirements: Bellow, there is the list of the required to build.

- MongoDB (7.0) +

Download the iris dataset.

Download the iris dataset and save the file as iris.csv.

<https://gist.github.com/netj/8836201file-iris-csv>

Note: In the github repository <https://github.com/DuvanSGF/MongoDBWorkspaceGithub/>

you will find all the source files and the iris dataset file.

1. Create the database.

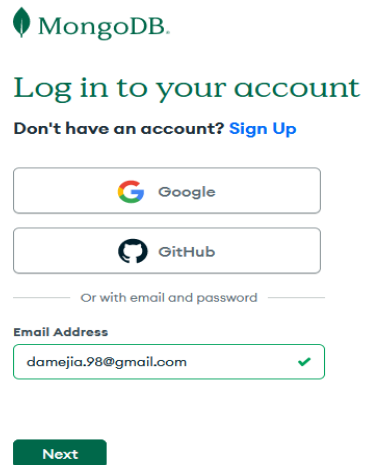
The process of database creation is:

1. Create a login role.
2. Create a database.
3. Import the data.
4. 3 NoSQL Database

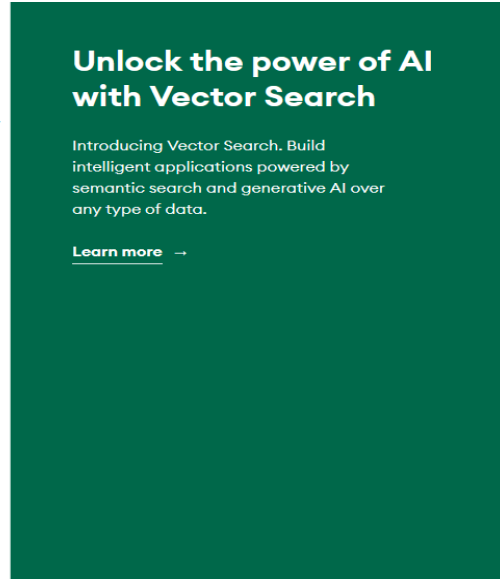
1.1. NoSQL Database

Register MongoDB Atlas

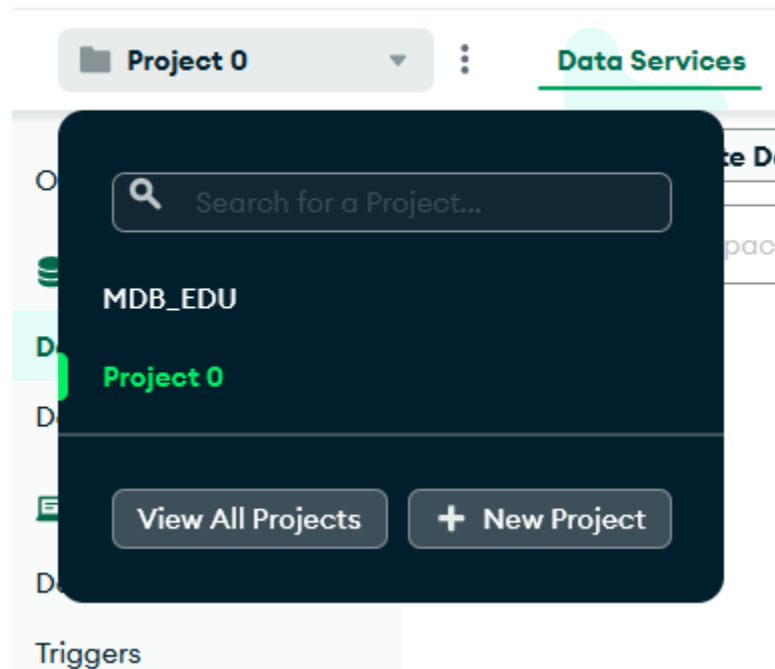
Initially, we must establish an account on MongoDB Atlas.



The image shows the MongoDB Atlas login and registration interface. It features the MongoDB logo at the top left. Below it, the text "Log in to your account" is displayed. Underneath, there is a link "Don't have an account? Sign Up". There are two buttons for social login: "Google" and "GitHub". Below these, there is a link "Or with email and password". Underneath, there is a text input field for "Email Address" with the value "damejia.98@gmail.com" and a green checkmark icon. At the bottom, there is a green button labeled "Next".



Then, we need to create a new project, and then we establish a shared cluster.



We need to add the IP to the network access; you can include localhost or another IP if you want to establish a remote connection.

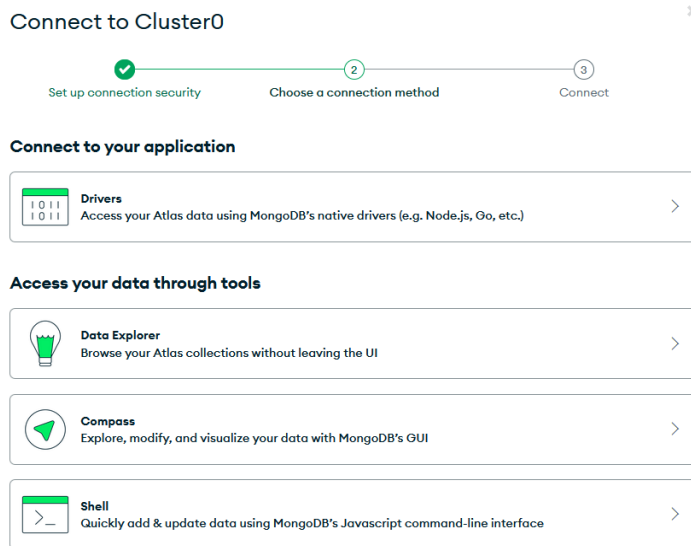
The screenshot shows a modal dialog titled "Add IP Access List Entry" with a close button (X) in the top right corner. Below the title is a descriptive paragraph: "Atlas only allows client connections to a cluster from entries in the project's IP Access List. Each entry should either be a single IP address or a CIDR-notated range of addresses. [Learn more.](#)". Below this text is a button labeled "ADD CURRENT IP ADDRESS". There are two input fields: "Access List Entry:" with a placeholder "Enter IP Address or CIDR Notation" and "Comment:" with a placeholder "Optional comment describing this entry". At the bottom, there is a toggle switch labeled "This entry is temporary and will be deleted in" followed by a dropdown menu currently set to "6 hours". To the right of the toggle are two buttons: "Cancel" and "Confirm".

Also, we need to add a user. You can either use your own user or utilize the default user "Admin."

The screenshot shows a form titled "Add New Database User" with a close button (X) in the top right corner. Below the title is a descriptive paragraph: "Create a database user to grant an application or user access to databases and collections in your clusters in this Atlas project. Granular access control can be configured with default privileges or custom roles. You can grant access to an Atlas project or organization using the corresponding [Access Manager](#)". Below this is the "Authentication Method" section with four buttons: "Password" (highlighted with a green border), "Certificate", "AWS IAM (MongoDB 4.4 and up)", and "PREVIEW Federated Auth (MongoDB 7.0 and up)". Below the buttons is a note: "MongoDB uses [SCRAM](#) as its default authentication method." The "Password Authentication" section has a text input field with a placeholder "e.g., new-user_01", a password input field with a placeholder "Enter password" and a "SHOW" button, and two buttons: "Autogenerate Secure Password" and "Copy". Below this is the "Database User Privileges" section with a descriptive paragraph: "Configure role based access control by assigning database user a mix of one built-in role, multiple custom roles, and multiple specific privileges. A user will gain access to all actions within the roles assigned to them, not just the actions those roles share in common. **You must choose at least one role or privilege.** [Learn more about roles.](#)". At the bottom, there is a "Built-in Role" section with a note "Select one [built-in role](#) for this user." and a button "Add Built In Role". On the right side of this section, there is a status indicator "0 SELECTED" and an upward arrow icon.

Establish a remote connection

To initiate communication with the cluster, we need to select the option via the SHELL, follow the steps, and finally, copy the command and paste it into a new terminal window.



Run the MongoDB client

To run MongoDB client, open a new terminal Window and Execute the mongo command.

```
cmd mongosh mongodb+srv://<credentials>@cluster0.gcymwww.mongodb.net/

C:\Users\sqlab>mongosh "mongodb+srv://cluster0.gcymwww.mongodb.net/" --apiVersion 1 --username admin
Enter password: *****
Current Mongosh Log ID: 65bd14b287a442a135532e0f
Connecting to:      mongodb+srv://<credentials>@cluster0.gcymwww.mongodb.net/?appName=mongosh+2.1.3
Using MongoDB:      6.0.13 (API Version 1)
Using Mongosh:      2.1.3
Atlas atlas-yzclwj-shard-0 [primary] test>
Atlas atlas-yzclwj-shard-0 [primary] test> _
```

1.1.1 Create the database

To create a MongoDB database, employ the command to use database name. If the database does not exist, it is created, else it is set as the default database (the database in use).

```
use iris
```

You will see.

```
Atlas atlas-yzclwj-shard-0 [primary] test> use iris
switched to db iris
```

1.1.2 Create a collection

In MongoDB, a collection is a set of documents, which is like a table in a relational database. To create a collection, use the command `db.createCollection("collection name")`. The keyword `db` is the database in use (`iris`).

```
db.createCollection("iris")
```

The server must return a `{"ok":1}` message.

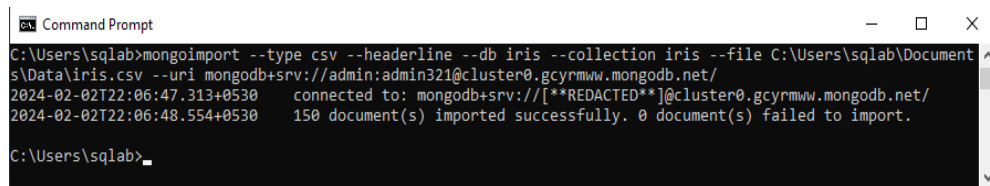
```
Atlas atlas-yzclwj-shard-0 [primary] iris> db.createCollection("iris")
{ ok: 1 }
```

1.1.3 Import data from the iris.csv file

To import the data from the `iris.csv` file, use the command `mongoimport`.

```
>mongoimport --type csv --headerline --db iris --collection iris --file C:\Users\sqlab\Documents\Data\iris.csv
--uri mongodb+srv://admin:admin321@cluster0.gcyrmw.mongodb.net/
```

The terminal window will be something like this.



```
Command Prompt
C:\Users\sqlab>mongoimport --type csv --headerline --db iris --collection iris --file C:\Users\sqlab\Documents\Data\iris.csv --uri mongodb+srv://admin:admin321@cluster0.gcyrmw.mongodb.net/
2024-02-02T22:06:47.313+0530 connected to: mongodb+srv://[**REDACTED**]@cluster0.gcyrmw.mongodb.net/
2024-02-02T22:06:48.554+0530 150 document(s) imported successfully. 0 document(s) failed to import.
C:\Users\sqlab>
```

1.1.4 See the records of the iris table

To see all the documents of the iris collection execute the `db.collection_name.find()`

command:

```
db.iris.find()
```

```
GA. mongosh mongodb+srv://<credentials>@cluster0.gcymwww.mongodb.net/
Atlas atlas-yzclwj-shard-0 [primary] iris> db.iris.find()
[
  {
    _id: ObjectId('65bd1a1fff333e9283a0ac73'),
    sepal: { length: 4.9, width: 3 },
    petal: { length: 1.4, width: 0.2 },
    variety: 'Setosa'
  },
  {
    _id: ObjectId('65bd1a1fff333e9283a0ac74'),
    sepal: { length: 5.1, width: 3.5 },
    petal: { length: 1.4, width: 0.2 },
    variety: 'Setosa'
  }
]
Atlas atlas-yzclwj-shard-0 [primary] iris>
```