

MongoDB Atlas

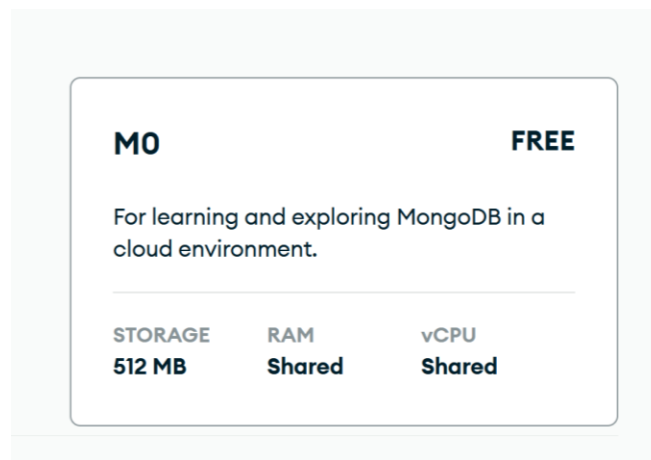
By now, you may be familiar with MySQL and phpMyAdmin. For phpMyAdmin you can export your database by selecting the database to export then going to the **Export** tab and running the export to file download.

For MongoDB we'll be using MongoDB Compass and MongoDB Atlas for the online server. For testing/prototyping purposes we can use the free tier. (For us, with little projects it's good enough.)

Create Cluster and DB on MongoDB Atlas (<https://www.mongodb.com/atlas>)

You can have **only one free cluster per project**.

1. Create an account and login.
2. There should be a project created by default (if not, create a project). Click to build a database.
3. Select the free option.



4. Choose any cluster region. In general, choose the one closest to your location.
5. Click **Create** at bottom of the page.
6. Once the cluster is finished provisioning, you should be asked to secure your connection. If you can't see this, click on **Quickstart** under SECURITY in the sidebar.
7. Set up a user name and password for an admin user and add your local IP (this will allow you to access from your local environment). You can also choose "Allow Access From Anywhere" to make your data readable from anywhere.

Configuring IP access

Later, once you've deployed your project, if you're using MongoDB, you'll need to add your site server's IP to the allowed IP addresses list.

8. Create the DB you want to migrate. Note that you can't migrate everything all at once using Atlas live to a free tier cluster, so we'll duplicate the DB and collection names. (It's easier if you start with a live MongoDB database on MongoDB Atlas.)

- a. Click on **DEPLOYMENT > Databases** in the sidebar.
 - b. Click on the **Browse Collections** button.
 - c. Click the **Add My Own Data** button and create the "testdb" database and "menuLinks" collection.
9. Once that's done, you can create another user who will only have read/write privileges for your DB.
 - a. Click on **Database Access**.
 - b. Click on + **Add New Database User**.
 - c. Use "testdbuser" as the username and you can auto-generate the password (don't forget to copy/paste this somewhere).
 - d. For the user privileges, expand "Specific Privileges" and click **Add Specific Privilege**.
 - e. Select "readWrite" for the role and type "testdb" for the database.
 - f. Under "Built-in Role" click on the garbage can to delete the "Read and write to any database" role because this is too permissive (we already have a dbadmin user).
10. To get the connection string, click on **Database** in the sidebar.
11. Click on **Connect** and select **Connect using MongoDB Compass**.

Export local MongoDB data

1. Log in to MongoDB Compass using your local DB user credentials.
2. Navigate to the collection you want to export (e.g. "menuLinks" from our previous Node app example).
3. Click on the export collection button.
4. In the popup, select "Export Full Collection".
5. Make sure all fields are selected, then click **Select Output**.
6. Select "JSON" as the file format and browse to a location where you want to save the exported JSON file.
7. Click **Export**.

Import JSON to remote MongoDB database

1. Connect to your remote MongoDB database using MongoDB Compass.
 - a. From Atlas, copy the connection string to use with MongoDB Compass.
 - b. In MongoDB Compass, select **New Connection**.
 - c. Click on **Edit** so that you can paste the connection string in the text field.
 - d. Paste in the connection string but modify the values accordingly. You need to edit the db user, password and change the database to "testdb". You can get rid of extra options ("?... " stuff). It should look something like the following (replace <pwd> with your remote testdbuser's password and <cluster_host> with cluster address):

```
mongodb+srv://testdbuser:<pwd>@<cluster_host>/testdb
```

Primary cluster

If you get an error when trying to import (i.e. a "not primary" error), make sure you're using the primary cluster in the connection string. To find the primary cluster, click on **Databases** in the sidebar then click on the cluster name (e.g. Cluster0). This will open the overview page.

Copy the primary cluster name and use that in the connection string as the host name.

- e. Click **Connect** and you should be connected to the remote cluster.
2. Navigate to the `menuLinks` collection.
3. It should be empty. Click on **Insert Document**.
4. In the popup, click on the button with curly brackets so you can just enter JSON.
5. Delete the current data in the window.



6. Open your exported *menuLinks.json* file in an editor and copy/paste the JSON data into the window.



7. Click on the **Insert** button at the bottom of the window (it should be green now).

Modify project

Modify your project files to use the new connection string and user credentials. Use the `dotenv` module to keep your secret user values from being exposed in the code (see page 4 of the *Secure Local MongoDB* PDF).