Creating a Mongo Database in the Cloud for Use with Heroku

Prior to 2020, we made use of a Heroku plugin known as mLab to host our mongo databases in the cloud. mLab, unfortunately, has been phased out. Instead, we will make use of the official MongoDB Cloud platform to host our own MongoDB databases that we can use for our web-hosted projects.

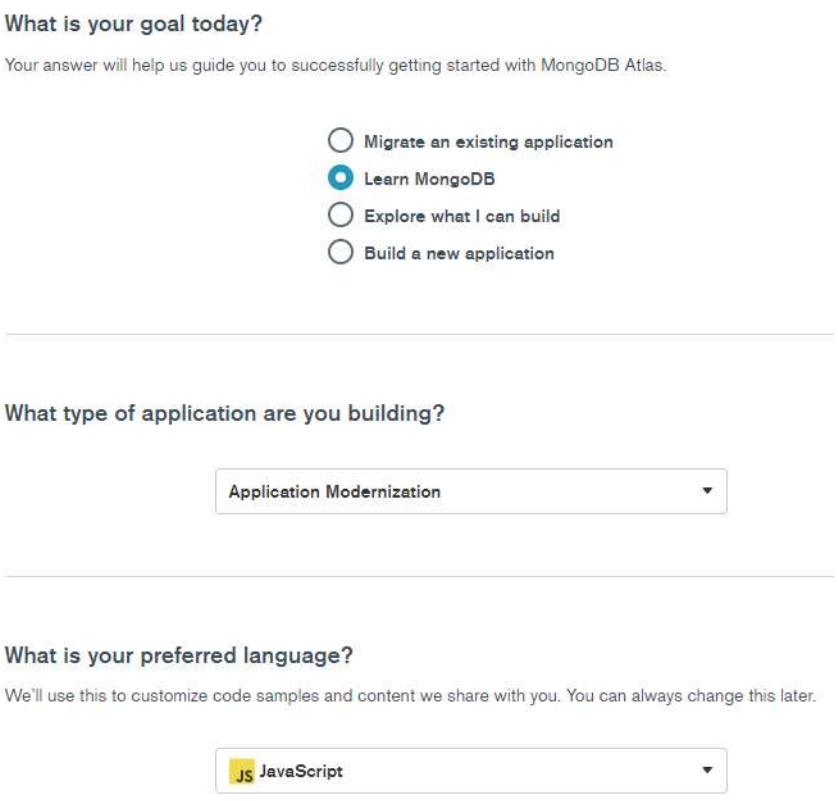
Before we can use the platform, we need to sign up and create our database instance:

1. Navigate to https://www.mongodb.com/cloud/atlas/register

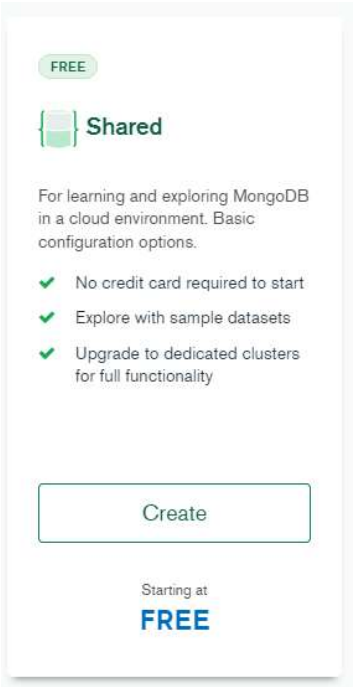
2. Select “Sign In with Google”, and log in using your RIT account. Be sure to use the @g.rit.edu format.

3. Accept the privacy policy and terms of service.

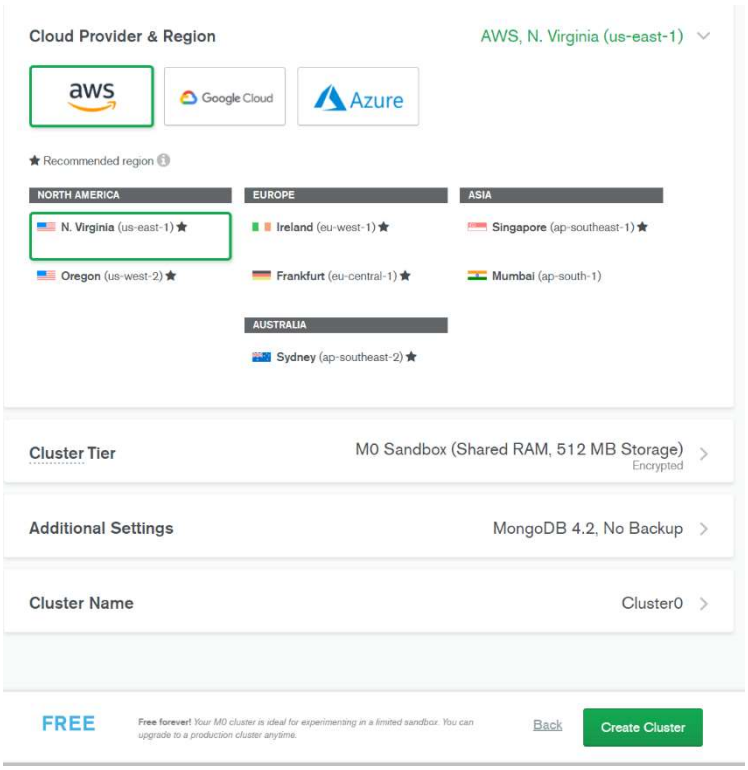
4. On the next screen, select the following options and press “Finish”.



5. On the next screen, select “Create” under the “Free, Shared” option.



6. On the next screen, simply press the “Create Cluster” button at the bottom right. All of the default options will serve our purposes for this class.



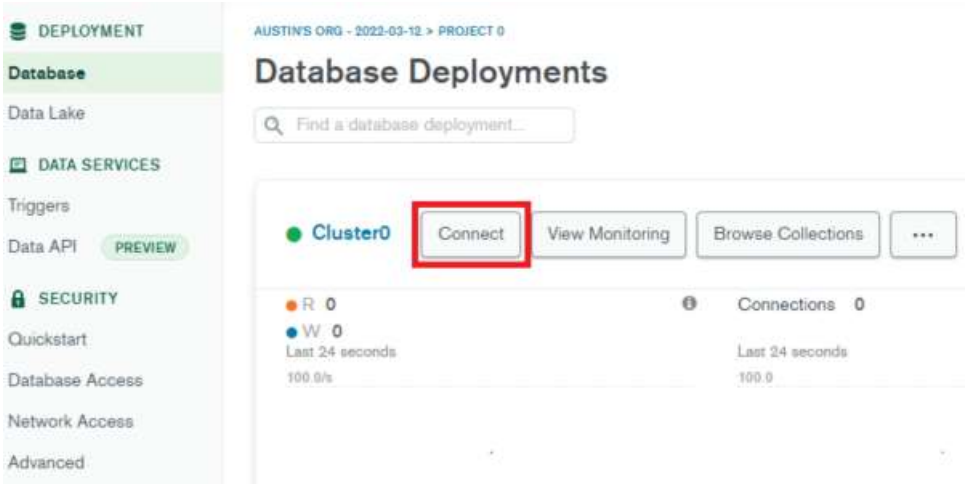
7. Once you finish creating the cluster, you will be brought to the Security Quickstart screen to configure your database for connections. On the left side of the screen, select “Network Access” under the Security header.

   On this screen, it should say “Add an IP address” at the center of the screen. Click the “Add IP Address” button.

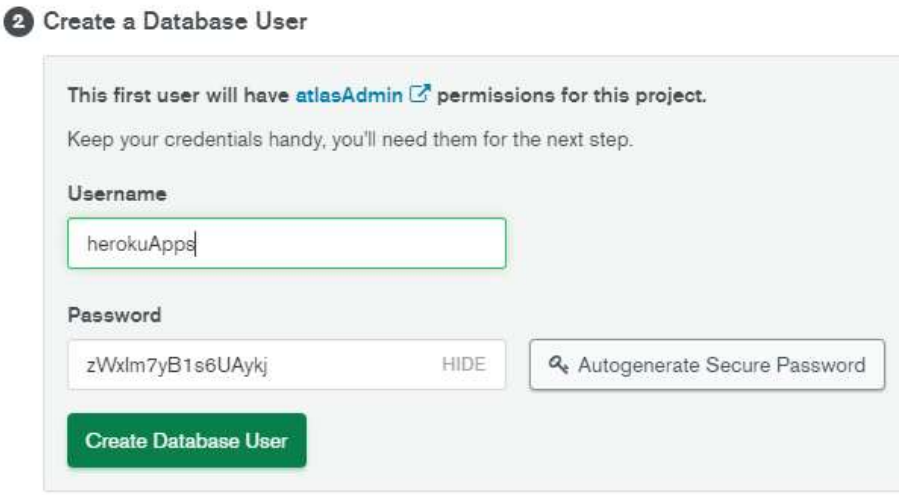
8. On this new screen, click “Allow Access from Anywhere” and then press “Confirm”. This will add the entry 0.0.0.0/0 to your access list. This means that any IP address can connect to your database (provided they have the correct login info as well).

   Usually this is not preferrable, as it means anyone with our DB Connection info can access our database, rather than just our trusted machines. However, the free version of Heroku does not give us a dedicated IP address and so we cannot whitelist every IP address they have.

9. Now we need to get you a connection string so that you can connect to your database from your Heroku applications. Go to the “Database” tab on the left under the “Deployment” header. Click the “Connect” button right below the name of your cluster.



10. On the next screen, you will be asked to create a database admin account. Name it whatever you like, and give it a secure password. **Be sure to know what this password is. Avoid using characters that require encoding such as spaces and other characters listed in this article:** https://developer.mozilla.org/en-US/docs/Glossary/percent-encoding**.** See the table in the "Character Encoding Chart" in this article: Press "Create Database User". Feel free to use the "Autogenerate secure password" button, but be sure to write down the password before progressing.



11. Press "Choose a Connection Method"

12. On the next screen, select "Connect your application". This will bring you to a screen where it will show you how to generate your connection string. Be sure that your "Driver" is set to Node.js, and that the version is 4.0 or later.

    Press the "Copy" button next to your connection string, and paste it into some text editor. You'll notice that your username is in the string, followed by <password>. Replace the <password> with your password, and be sure to keep this connection string in a secure area. **Be sure to remove the < > around password**.

    You'll also notice that toward the end of the string there is text that says **myFirstDatabase**. This is the name of the database you are connecting to. You will want to change this out for each app that you make with your Mongo cluster or else they will all point to the same data set. If you do not see this text, you should add the name of your database after **mongodb.net/** but before **?retryWrites**.

13. You have now successfully set your database up, and it is ready for use with Heroku. Continue below to setup “MongoDB Compass”, a helpful graphical interface that will allow you to easily manage your MongoDB Cloud instance and the data stored within it. Also continue to see how to add your connection string to your Heroku application.

Connecting to Your Database with MongoDB Compass

1. Begin by pressing “Go Back” on the screen where you copied your connection string.

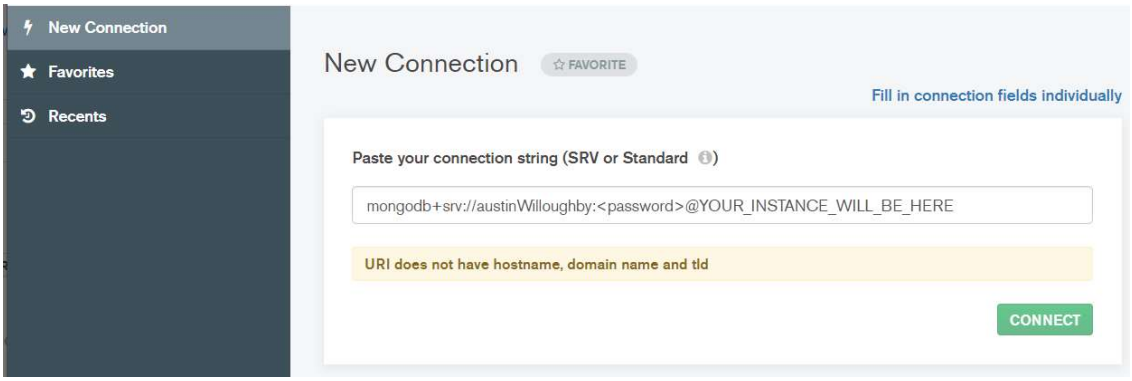
2. Select “Connect with MongoDB Compass”.

3. On this new screen, select your operating system and press “Download Compass”. Run the installer.

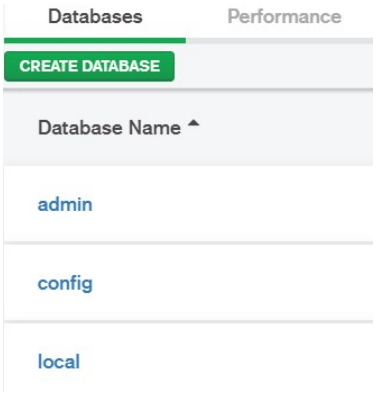
4. On the same screen, copy the connection string. Replace <password> with your password.

5. Run Compass once it has been installed. Select all options you would like for the “Privacy Settings” menu at startup. Press “Start Using Compass”.

6. On the next screen, paste your connection string in the box and press “Connect”.



7. You will now be brought to a screen that looks like this. These are all databases that exist within your Database Cluster by default. As you use your database, other collections will begin to appear here. You can use Compass to edit, delete, and add data to your database.

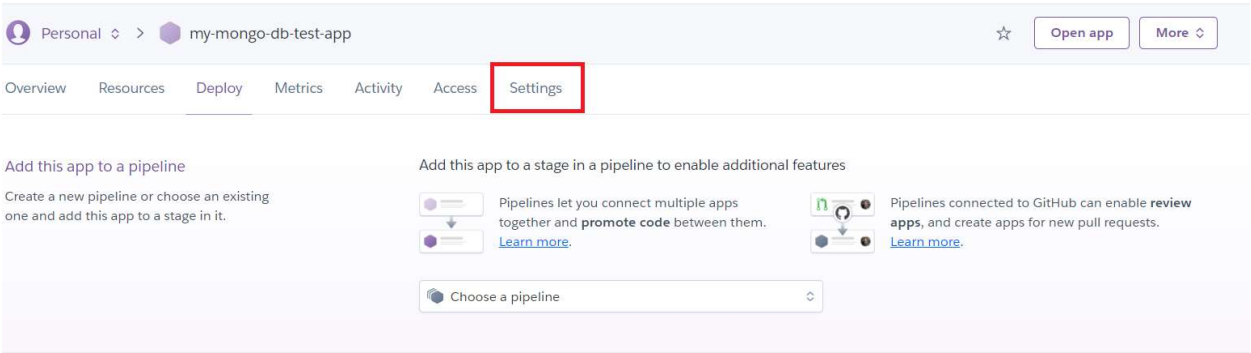


Connecting your Heroku App to your Cloud Database

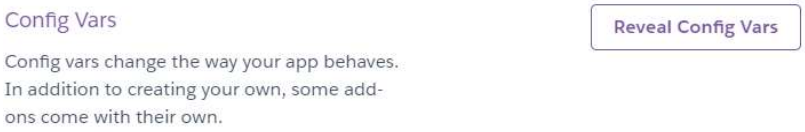
Once you have configured your MongoDB Cloud instance and have your connection string (from part 13 of the first section of this document), you will need to add your connection string to your Heroku App "Config Vars".

Note: You will still need to write JavaScript code for your server to make use of your Mongo database. This will just get you to a point where your Heroku App can connect to the database properly.

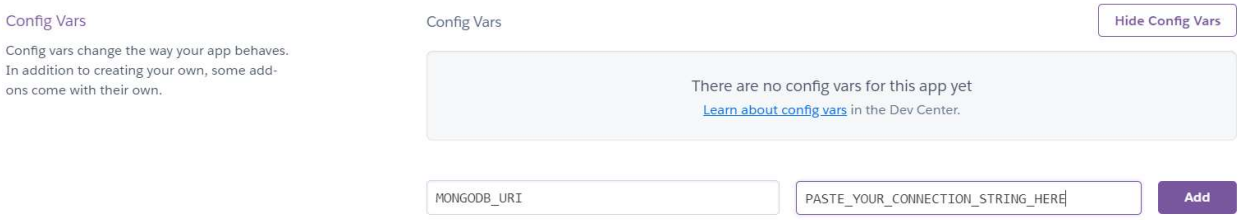
1. Open your Heroku App on the Heroku Dashboard. Navigate to the "Settings" tab.



2. Click on “Reveal Config Vars”



3. Add a new Config Var with the key “`MONGODB_URI`”. For the value, paste your completed connection string. **Be sure to replace <password> with your password. Remember to remove the < >. Replace myFirstDatabase with the name of your Heroku app.** Keep in mind that if your connection string doesn’t have **myFirstDatabase**, place the name of your application after **mongodb.net/** and before **?retryWrites**.



4. You have now created a new configuration variable for your Heroku application. When your node code is run on Heroku, this data will be injected into the `process.env.MONGODB_URI` variable in your code. We will make use of this to connect to our databases from our code.