Introduction to mongoDB

Hands-On Workshop

Part 1 - Atlas

Overview

This hands-on workshop is designed to get you familiar with all aspects of MongoDB, from deploying a cluster, to loading data to creating services to access that data.

This workshop includes 8 lab exercises and several more optional lab exercises you can try as time allows. Don’t worry about completing all optional lab exercises in this sitting. The free environment you create in this lab will be yours forever.

Prerequisites

To successfully complete this workshop:

- You must be able to make outgoing requests from your computer to MongoDB Atlas servers, which will be running on port 27017. Please confirm that port 27017 is not blocked by your network by clicking http://portquiz.net:27017. If successful, you will see a page load that indicates you can make outgoing requests on port 27017.
- Privileges to install software on your computer. We will be installing MongoDB Compass in this workshop.

Hands-on Labs

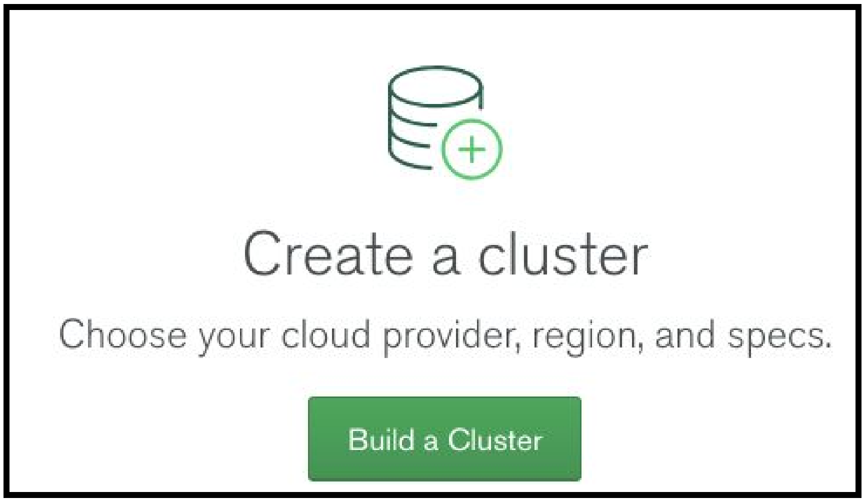
Lab 1 - Create the Cluster

Create an Account or Log In to Atas

We'll be using MongoDB Atlas, our fully managed MongoDB-as-a-service, for this workshop. Go to https://cloud.mongodb.com and either create a new account or log into an existing account you may have previously created.

Create a Free Tier Cluster

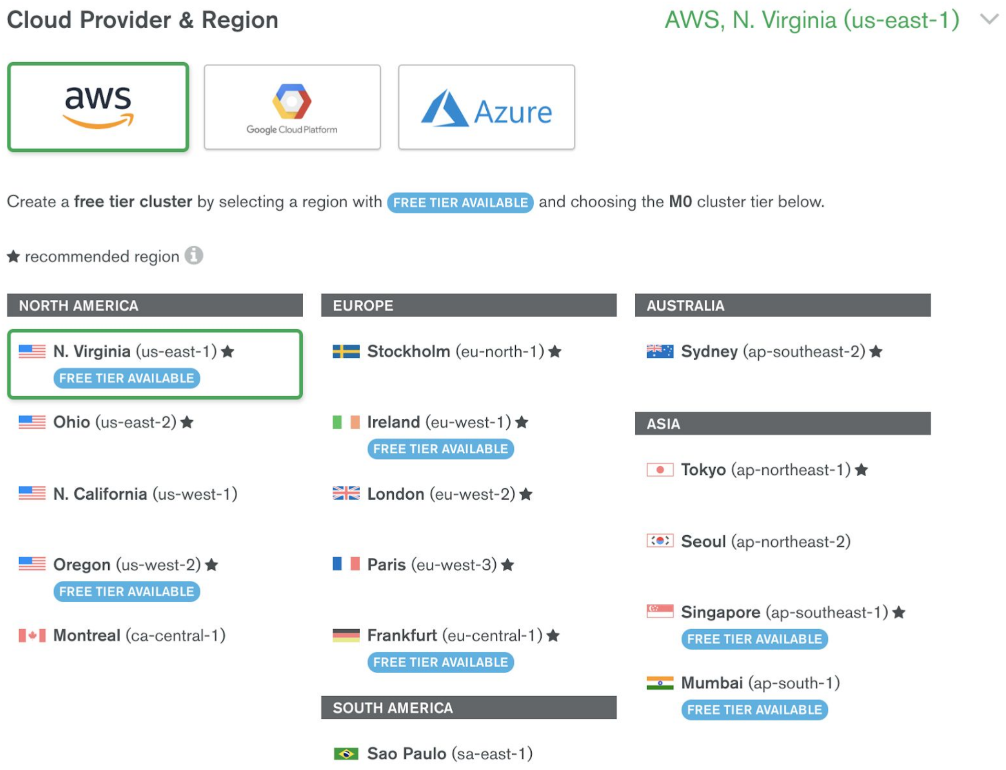
Click **Build a Cluster**:



Create a cluster

Choose your cloud provider, region, and specs.

Take a moment to browse the options (Provider & Region, Cluster Tier, Version, Backup, …). For our workshop, you can select **ANY** Cloud Provider:



Cloud Provider & Region

AWS, N. Virginia (us-east-1)

Create a **free tier cluster** by selecting a region with FREE TIER AVAILABLE and choosing the **M0** cluster tier below.

★ recommended region

NORTH AMERICA

- **N. Virginia** (us-east-1) ★ FREE TIER AVAILABLE
- **Ohio** (us-east-2) ★
- **N. California** (us-west-1)
- **Oregon** (us-west-2) ★ FREE TIER AVAILABLE
- **Montreal** (ca-central-1)

EUROPE

- **Stockholm** (eu-north-1) ★
- **Ireland** (eu-west-1) ★ FREE TIER AVAILABLE
- **London** (eu-west-2) ★
- **Paris** (eu-west-3) ★
- **Frankfurt** (eu-central-1) ★ FREE TIER AVAILABLE

SOUTH AMERICA

- **Sao Paulo** (sa-east-1)

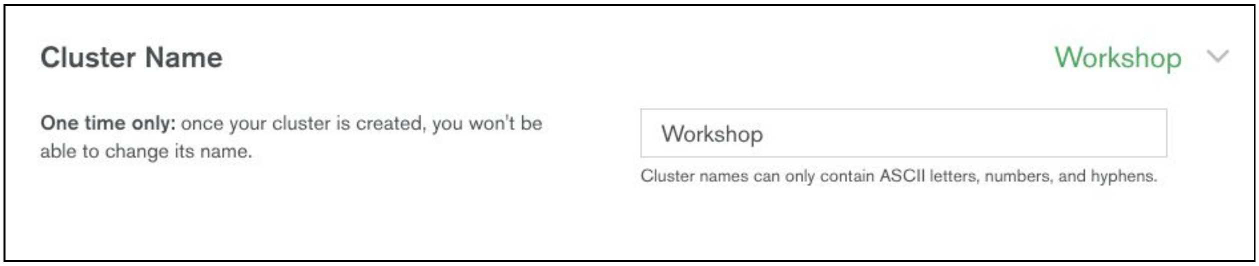
AUSTRALIA

- **Sydney** (ap-southeast-2) ★

ASIA

- **Tokyo** (ap-northeast-1) ★
- **Seoul** (ap-northeast-2)
- **Singapore** (ap-southeast-1) ★ FREE TIER AVAILABLE
- **Mumbai** (ap-south-1) FREE TIER AVAILABLE

and set the Cluster Name to **Workshop**:



Cluster Name

Workshop

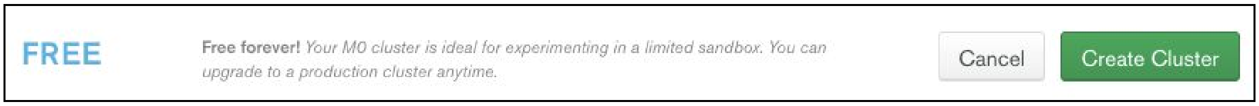
One time only: once your cluster is created, you won't be able to change its name.

Workshop

Cluster names can only contain ASCII letters, numbers, and hyphens.

The remaining defaults will suffice.

Click **Create Cluster**:



FREE

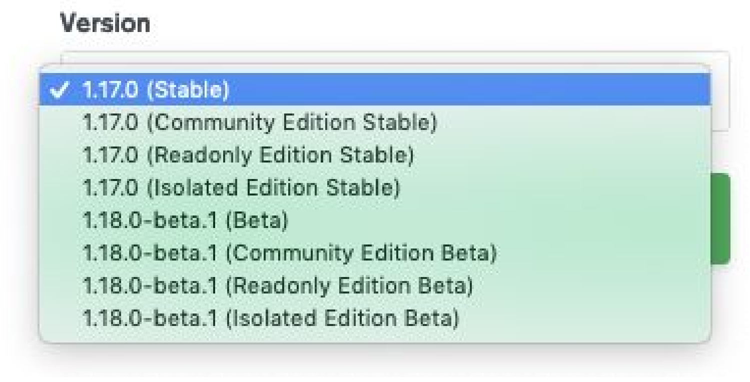
Free forever! *Your M0 cluster is ideal for experimenting in a limited sandbox. You can upgrade to a production cluster anytime.*

Continue to Lab 2 while the cluster is provisioning.

Lab 2 - Connect to the Cluster

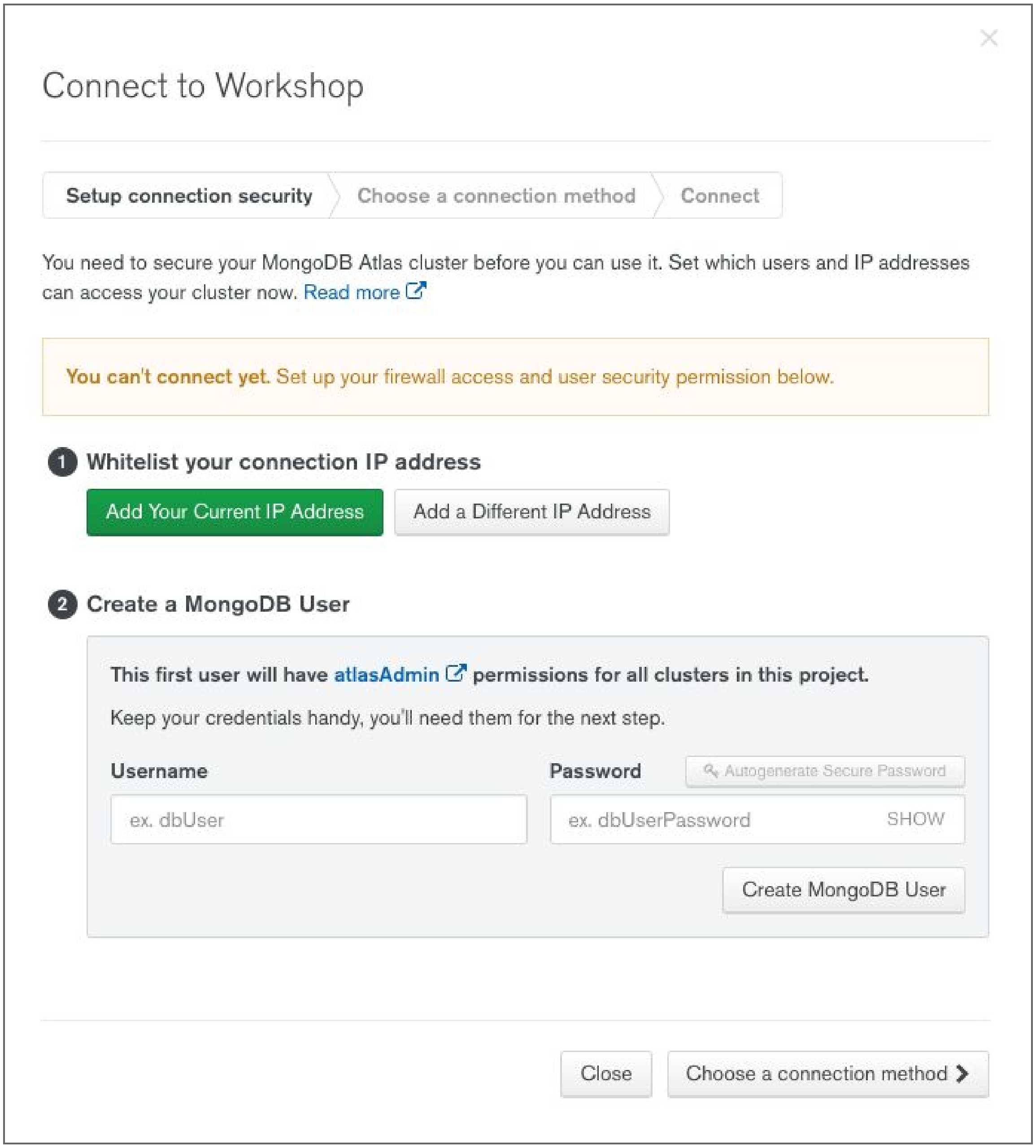
Install Compass

Compass is the GUI for MongoDB. Go to https://www.mongodb.com/download-center/compass to download and install Compass for your platform. Note, there are several editions of Compass. Make sure you download the “Stable” edition:


Version
✓ 1.17.0 (Stable)
1.17.0 (Community Edition Stable)
1.17.0 (Readonly Edition Stable)
1.17.0 (Isolated Edition Stable)
1.18.0-beta.1 (Beta)
1.18.0-beta.1 (Community Edition Beta)
1.18.0-beta.1 (Readonly Edition Beta)
1.18.0-beta.1 (Isolated Edition Beta)

Setup Connection Security

Return to the Atlas UI. Your cluster should now be provisioned. Click the **CONNECT** button, which will prompt you to set up connection security:


×
Setup connection security
Choose a connection method
Connect
Add Your Current IP Address
Add a Different IP Address
Username
Password
Autogenerate Secure Password
ex. dbUser
ex. dbUserPassword
SHOW
Create MongoDB User
Close
Choose a connection method

Connect to Workshop

You need to secure your MongoDB Atlas cluster before you can use it. Set which users and IP addresses can access your cluster now. Read more

You can't connect yet. Set up your firewall access and user security permission below.

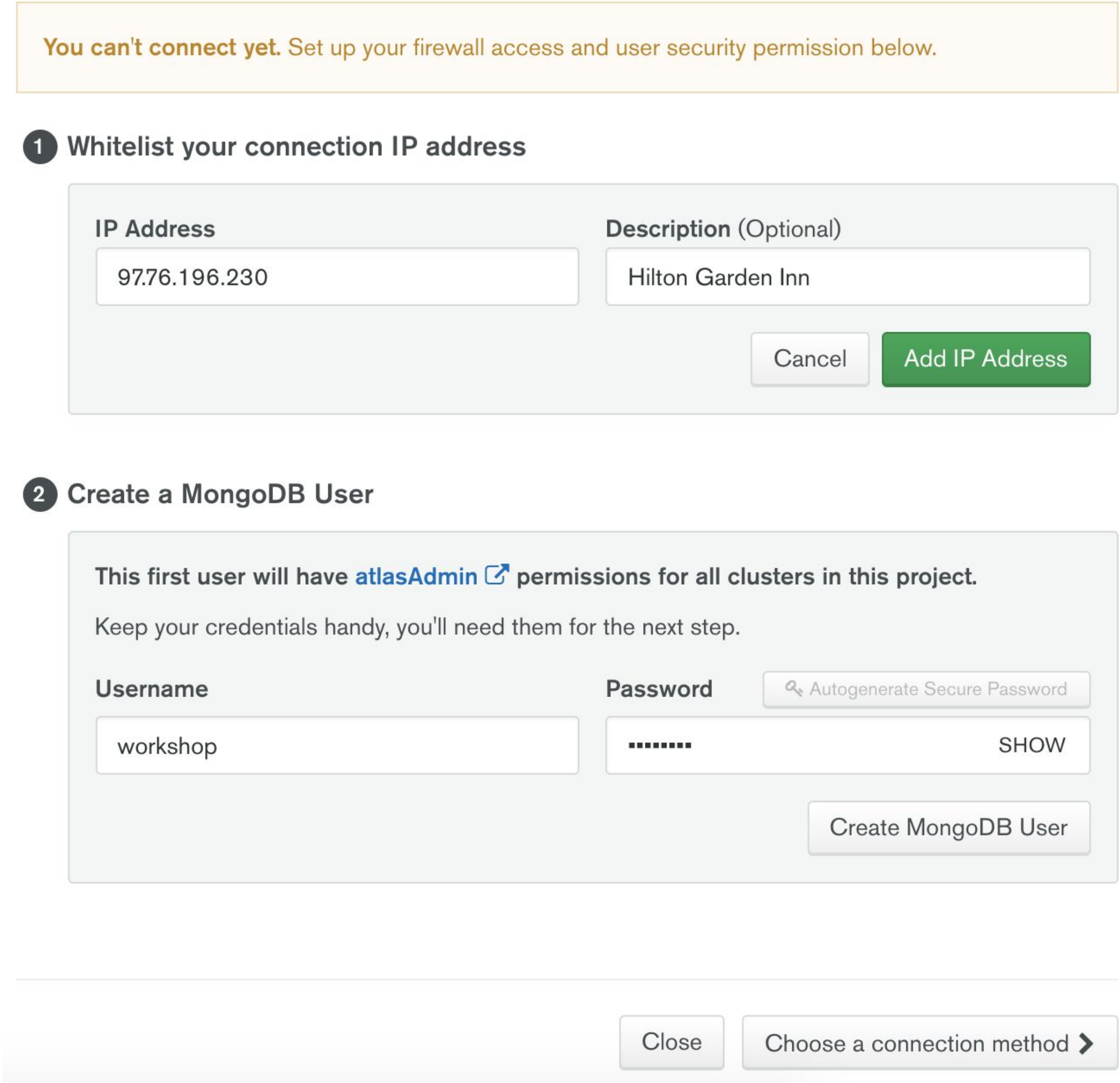
1 Whitelist your connection IP address

2 Create a MongoDB User

This first user will have atlasAdmin permissions for all clusters in this project.

Keep your credentials handy, you'll need them for the next step.

Add Your Current IP Address and **Create a MongoDB User.** I’m using Username **workshop** and password **workshop**:



You can't connect yet. Set up your firewall access and user security permission below.

1 **Whitelist your connection IP address**

IP Address: 97.76.196.230

Description (Optional): Hilton Garden Inn

2 **Create a MongoDB User**

This first user will have atlasAdmin permissions for all clusters in this project.

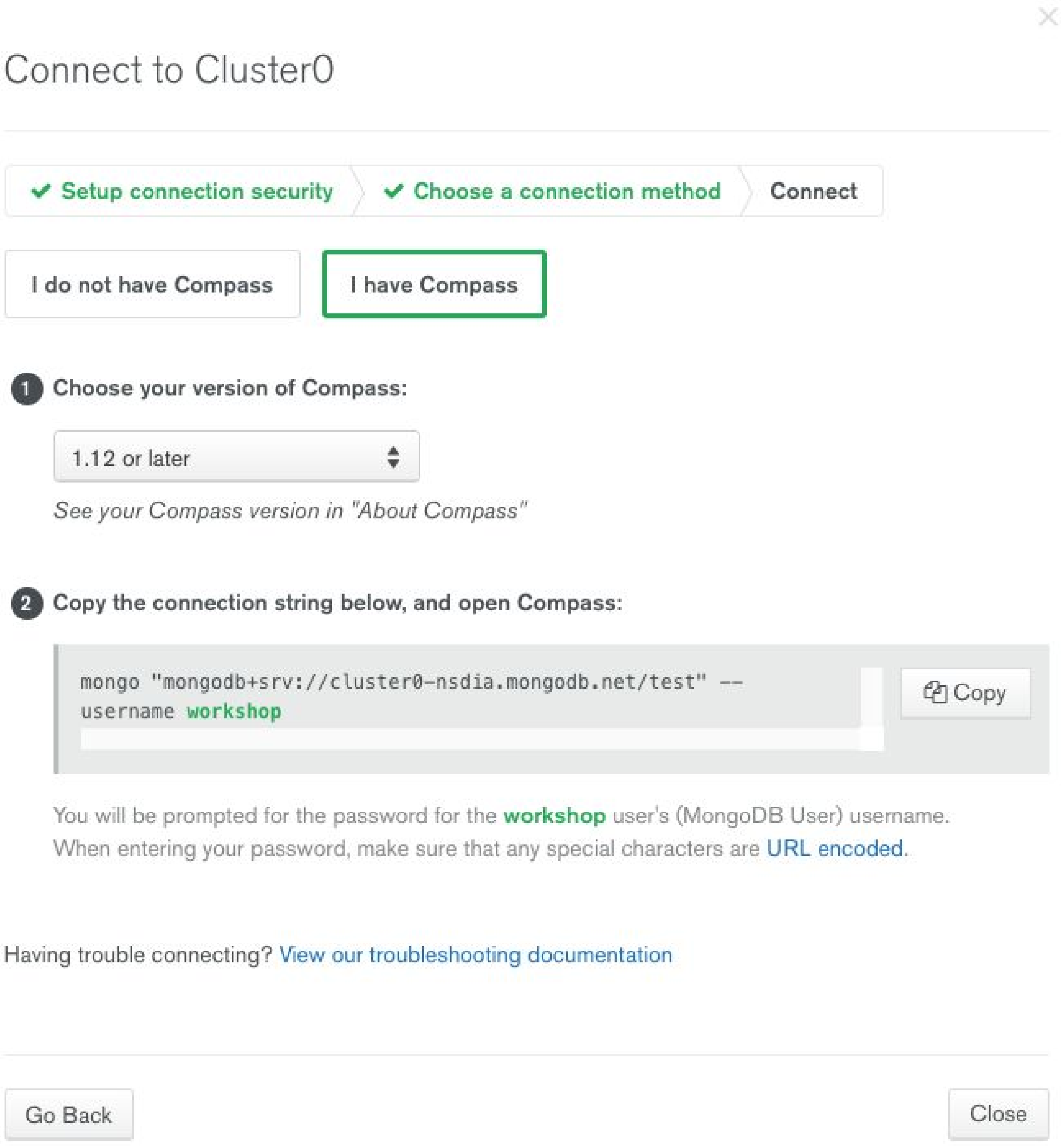
Keep your credentials handy, you'll need them for the next step.

Username: workshop

Password: •••••••••

Click **Choose a connection method** and select **I have Compass**.

In Step 1 where you choose your version of Compass, select **1.12 or later** and **COPY** the connection string presented:



Connect to Cluster0

1 Choose your version of Compass:

1.12 or later

See your Compass version in "About Compass"

2 Copy the connection string below, and open Compass:

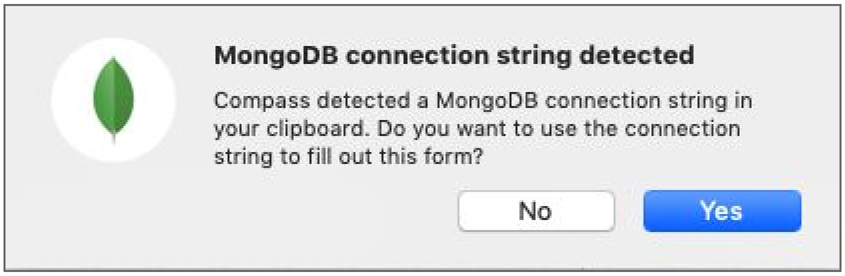
```
mongo "mongodb+srv://cluster0-nsdia.mongodb.net/test" --
username workshop
```

You will be prompted for the password for the **workshop** user's (MongoDB User) username. When entering your password, make sure that any special characters are URL encoded.

Having trouble connecting? View our troubleshooting documentation

Connect Compass

Start Compass and it should detect the connection string in your copy buffer:



MongoDB connection string detected

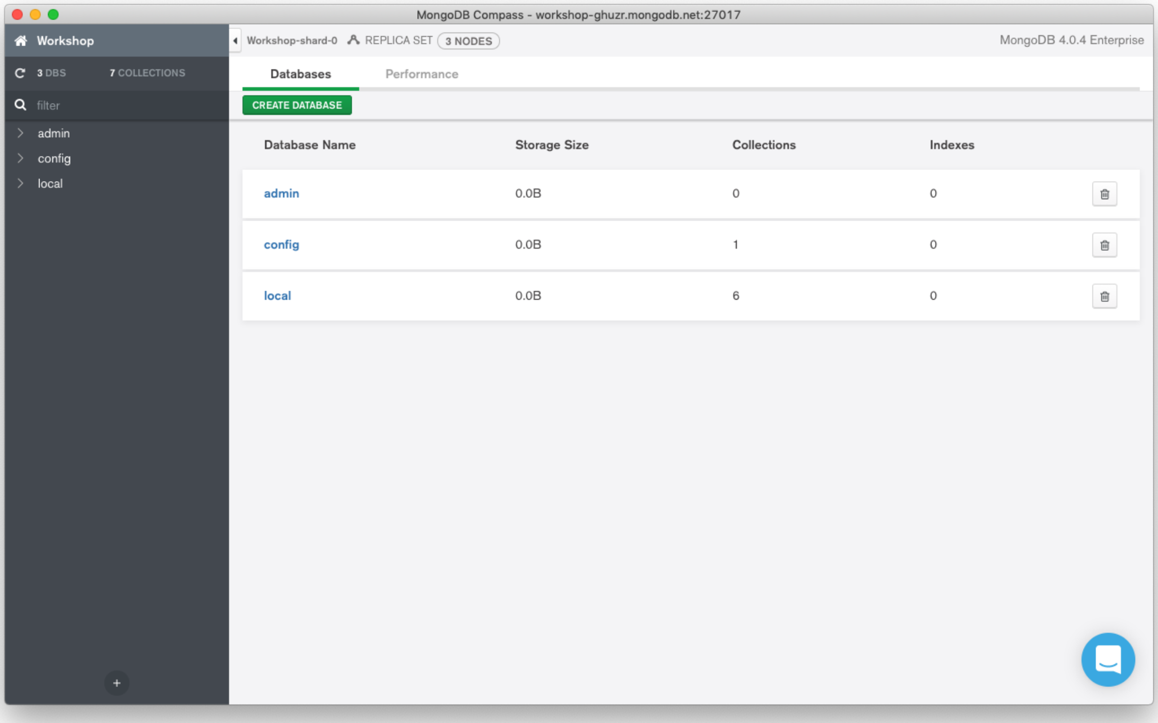
Compass detected a MongoDB connection string in your clipboard. Do you want to use the connection string to fill out this form?

Select **Yes**.

Make sure that the **SRV Record** is selected and the authentication is set to username/password. Provide the password (workshop) and *before clicking CONNECT*, **CREATE** a **FAVORITE** named **Workshop**. This will allow us to quickly connect to the cluster in the future.

Click **CONNECT**.

If successful, you'll see some internal databases used by MongoDB:



MongoDB Compass - workshop-ghuzr.mongodb.net:27017

Database Name	Storage Size	Collections	Indexes
admin	0.0B	0	0
config	0.0B	1	0
local	0.0B	6	0

Lab 3 - Load Data

For this workshop we're going to load a Yelp like collection of New York City restaurants. Download the primer-dataset.json dataset from Github. If you have the wget utility, you can get the dataset as follows:

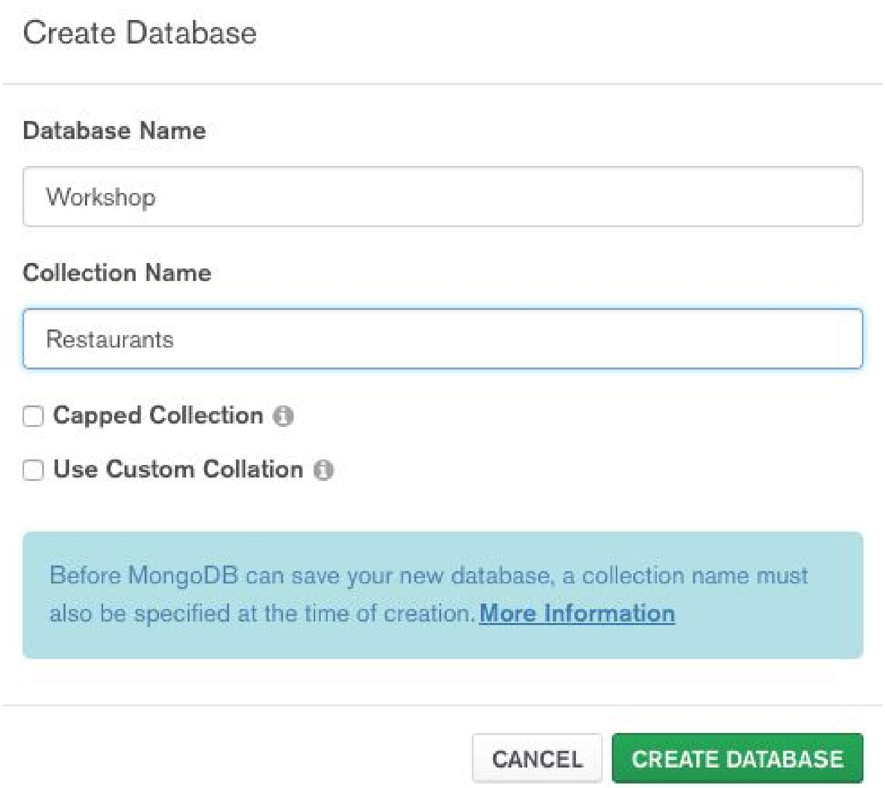
```
wget
https://raw.githubusercontent.com/mongodb/docs-assets/primer-dataset/primer-dataset.json
```

Otherwise, just open the link in your browser and once the load completes, save the file (File > Save Page As in Chrome)

The dataset is 11.9 MB and has 25K restaurants.

Create a Database and Collection

In Compass, click the **CREATE DATABASE** button and create a **Workshop** database with a **Restaurants** collection:



Create Database

Database Name: Workshop

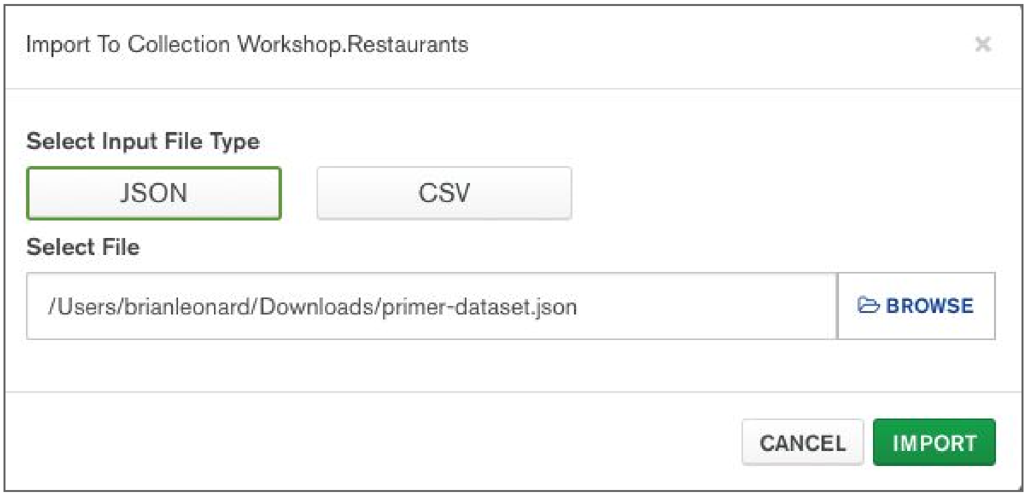
Collection Name: Restaurants

□ Capped Collection

□ Use Custom Collation

Before MongoDB can save your new database, a collection name must also be specified at the time of creation. More Information

In the left navigation pane, expand **Workshop** and select the **Restaurants** collection. Select **Collection** > **Import Data** from the top menu. Then **BROWSE** to the primer-dataset.json file you downloaded:



Import To Collection Workshop.Restaurants

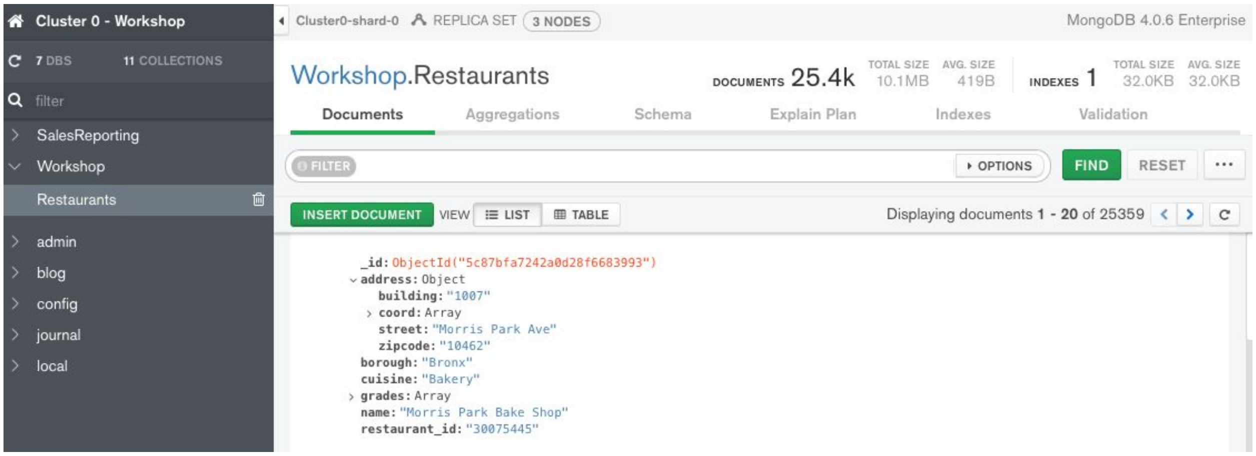
Select Input File Type: JSON

Select File: /Users/brianleonard/Downloads/primer-dataset.json

Then select **IMPORT**. You've just imported 25K documents into your Restaurants collection!

Lab 4 - Browse the Documents

Continuing to work in the **Restaurants** collection, select the **Documents** tab if it is not already selected.



Cluster0-shard-0 REPLICA SET 3 NODES MongoDB 4.0.6 Enterprise

Workshop.Restaurants

DOCUMENTS 25.4k TOTAL SIZE 10.1MB AVG. SIZE 419B INDEXES 1 TOTAL SIZE 32.0KB AVG. SIZE 32.0KB

Displaying documents 1 - 20 of 25359

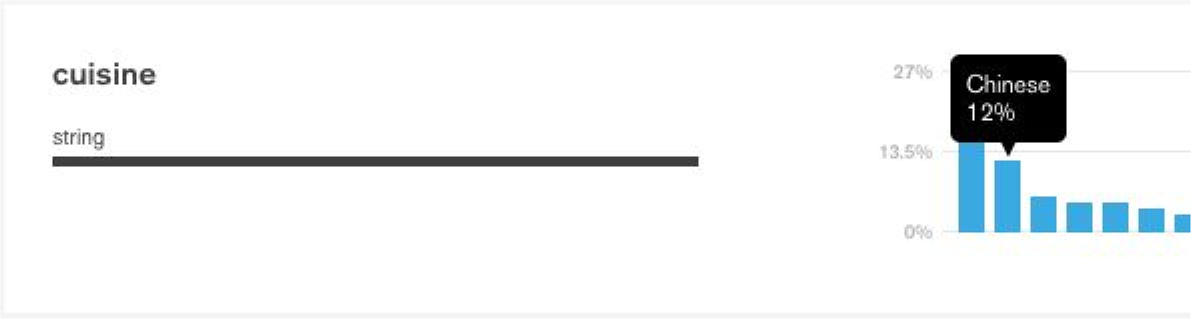
```
  _id: ObjectId("5c87bfa7242a0d28f6683993")
v address: Object
    building: "1007"
  > coord: Array
    street: "Morris Park Ave"
    zipcode: "10462"
  borough: "Bronx"
  cuisine: "Bakery"
> grades: Array
  name: "Morris Park Bake Shop"
  restaurant_id: "30075445"
```

Examine the documents in the collection. Notice how each document has several fields such as **_id**, **address**, and **borough**. The restaurant documents have a nested subdocument (address) and an array of subdocuments (grades). In a relational database, these fields would most likely be separate tables, but MongoDB allows us to embed this information. Working with data in this natural way is much **easier** than decomposing and composing from relational tables.

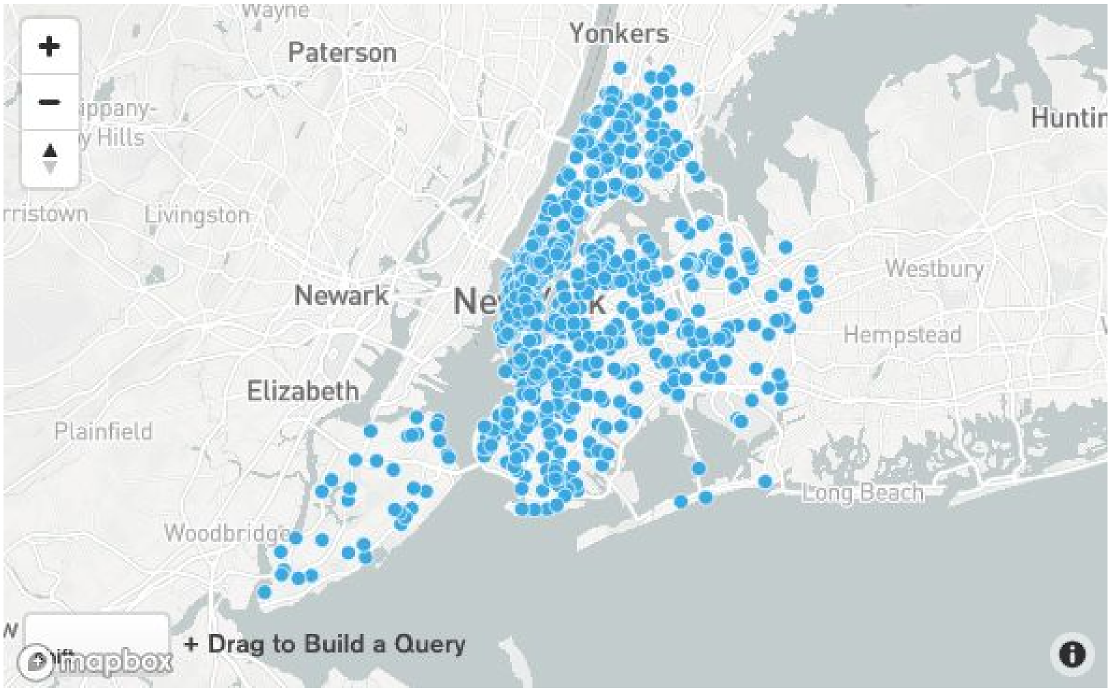
Lab 5 - View the Schema

You might be thinking, “Wait, I thought MongoDB was a NoSQL database, and hence, didn’t have a schema?” While that’s technically true, no dataset would be of any use without a schema. So while MongoDB doesn’t enforce a schema, your collections of documents will still always have one. The key difference with MongoDB is that the schema can be **flexible**.

Continuing to work in the **Restaurants** collection, select the **Schema** tab and click **Analyze Schema.** Compass will sample the documents in the collection to derive a schema. In addition to providing field names and types, Compass will also provide a summary of the data values. For example, for the cuisine field, we can see that Chinese is the 2nd most common at 12% (your results may differ slightly based on the sample that was taken):

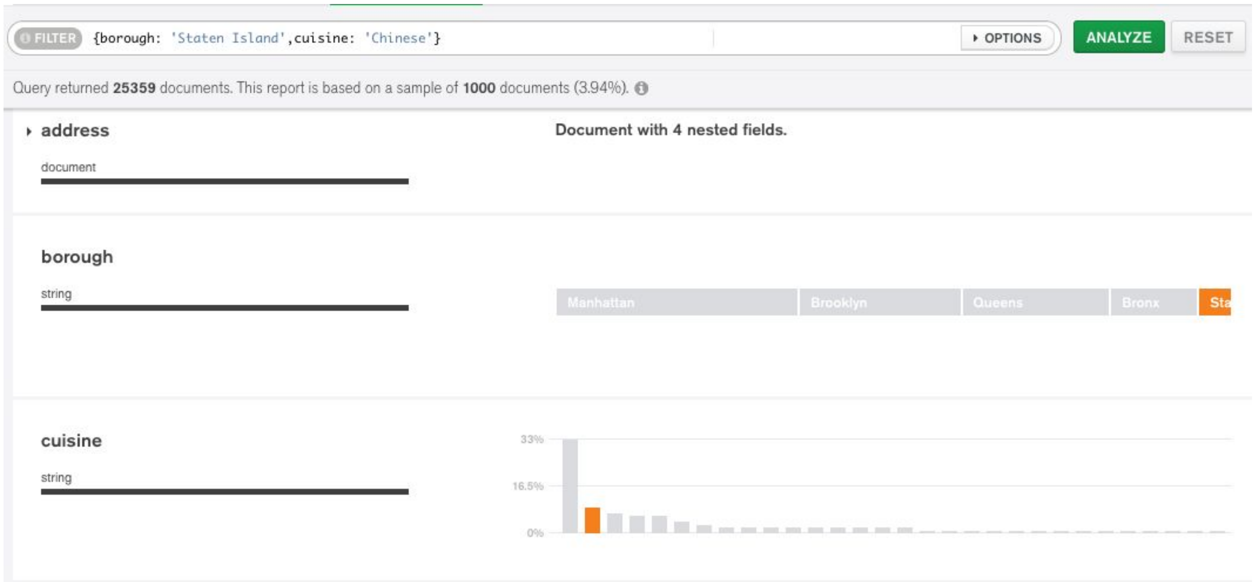


Expand the address field to discover MongoDB's excellent support for Geospatial Queries. As the collection is of restaurants in New York City, zoom the map to NYC:



Lab 6 - Query the Data

The MongoDB Query Language (MQL) is based on JSON. The Schema Analyzer in Compass provides an easy way to learn the language. For example, select **Staten Island** from the borough field (only **Sta** may be showing) and **Chinese** from the cuisine field. Notice as you make selections the FILTER field at the top of the window gets populated:



```
{borough: 'Staten Island',cuisine: 'Chinese'}
```

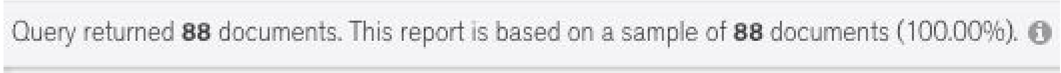
Query returned **25359** documents. This report is based on a sample of **1000** documents (3.94%).

address
document
Document with 4 nested fields.

borough
string

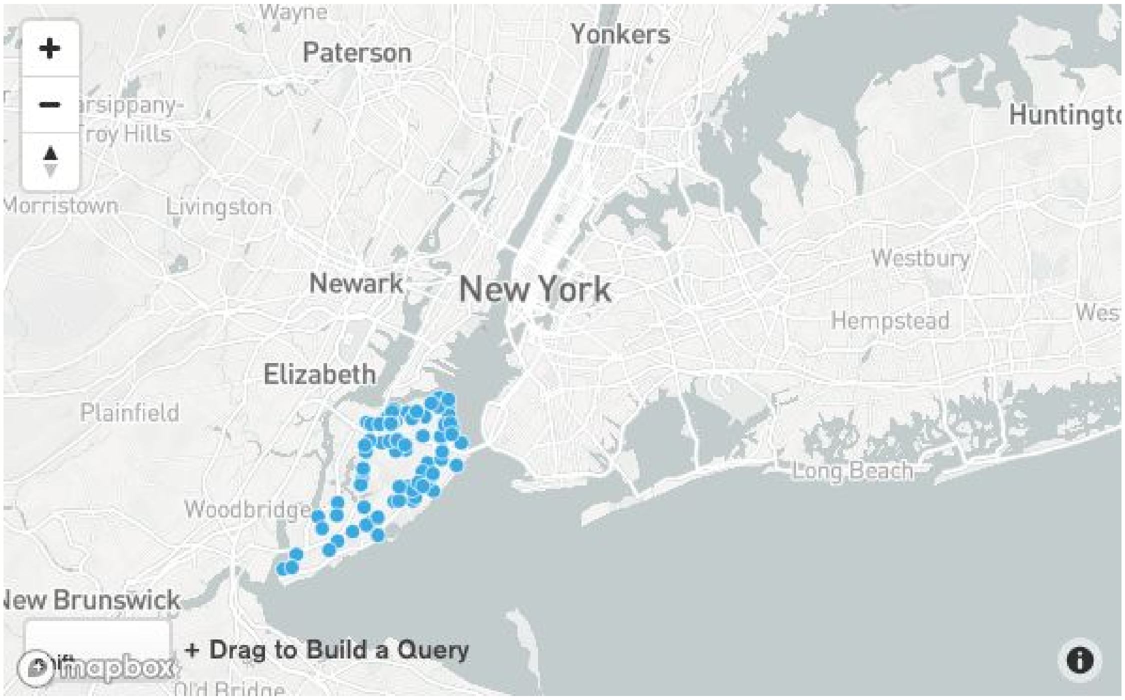
cuisine
string

Click the ANALYZE button to filter for Chinese restaurants in Staten Island, of which there are 88:

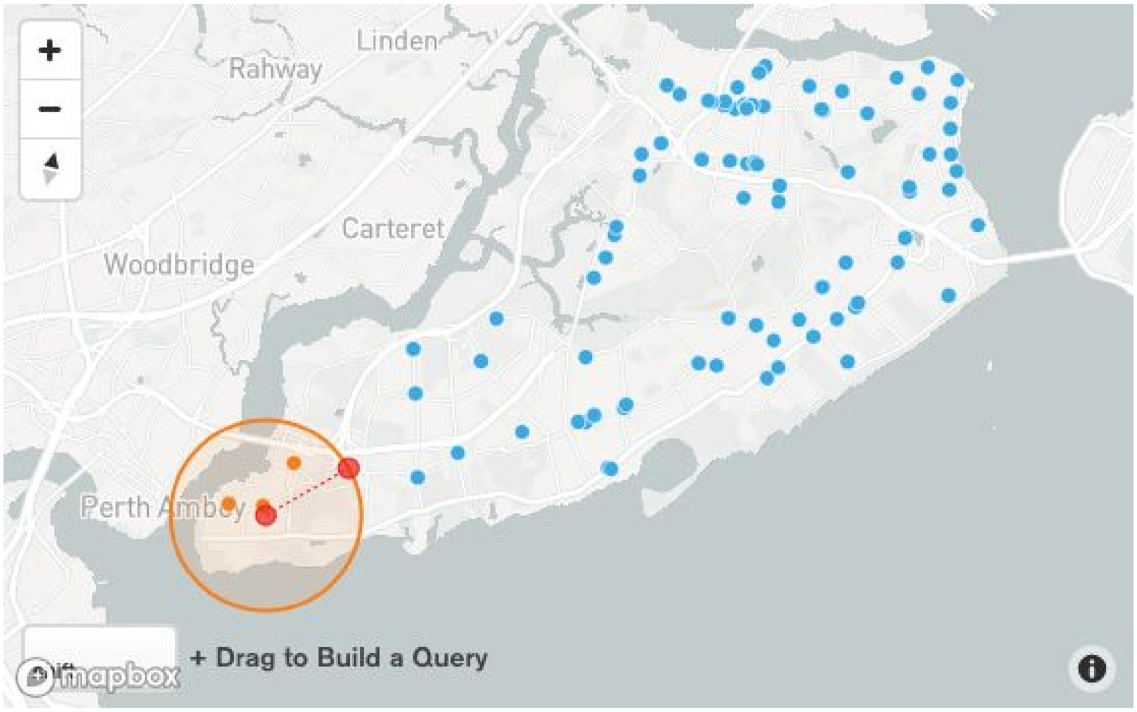


Query returned **88** documents. This report is based on a sample of **88** documents (100.00%).

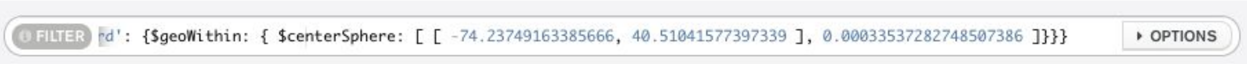
And you can now see this reflected on our map (more dots now appear on Staten Island because our sample now includes all 88 restaurants)



To perform a geospatial query, shift click and drag a circle on the map. Once the circle is in place, it can be moved and resized:

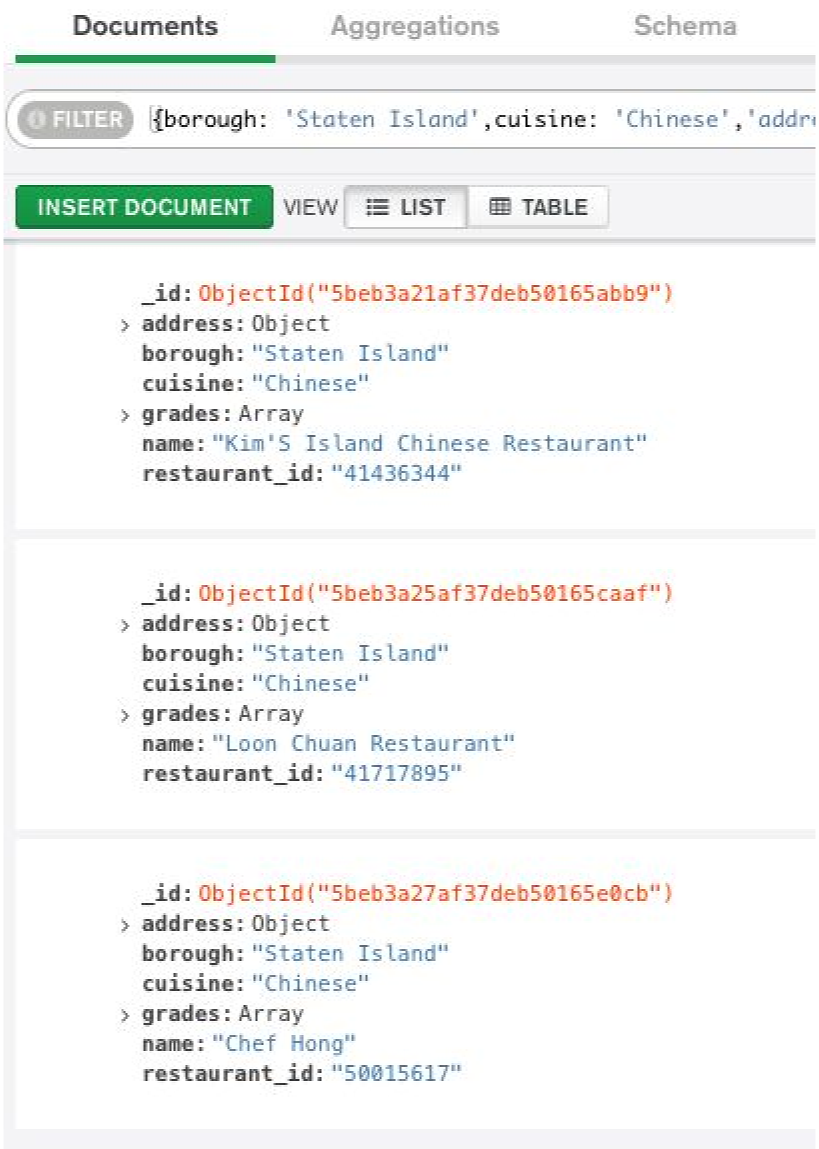


Notice the $geoWithin filter was added to our query:



```
rd': {$geoWithin: { $centerSphere: [ [ -74.23749163385666, 40.51041577397339 ], 0.00033537282748507386 ]}}}
```

Finally, click **ANALYZE** again and click the **Documents** tab to view the Chinese restaurants in our selected radius in Staten Island:



```
{borough: 'Staten Island',cuisine: 'Chinese','addr
```

```
  _id: ObjectId("5beb3a21af37deb50165abb9")
> address: Object
  borough: "Staten Island"
  cuisine: "Chinese"
> grades: Array
  name: "Kim'S Island Chinese Restaurant"
  restaurant_id: "41436344"
```

```
  _id: ObjectId("5beb3a25af37deb50165caaf")
> address: Object
  borough: "Staten Island"
  cuisine: "Chinese"
> grades: Array
  name: "Loon Chuan Restaurant"
  restaurant_id: "41717895"
```

```
  _id: ObjectId("5beb3a27af37deb50165e0cb")
> address: Object
  borough: "Staten Island"
  cuisine: "Chinese"
> grades: Array
  name: "Chef Hong"
  restaurant_id: "50015617"
```