

Introduction to



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 is divided into two parts, with Part 1 focusing on Atlas, our fully managed database service, and Part 2 focusing on Stitch, our serverless application platform. Each part contains several lab exercises. Don't worry about completing all 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

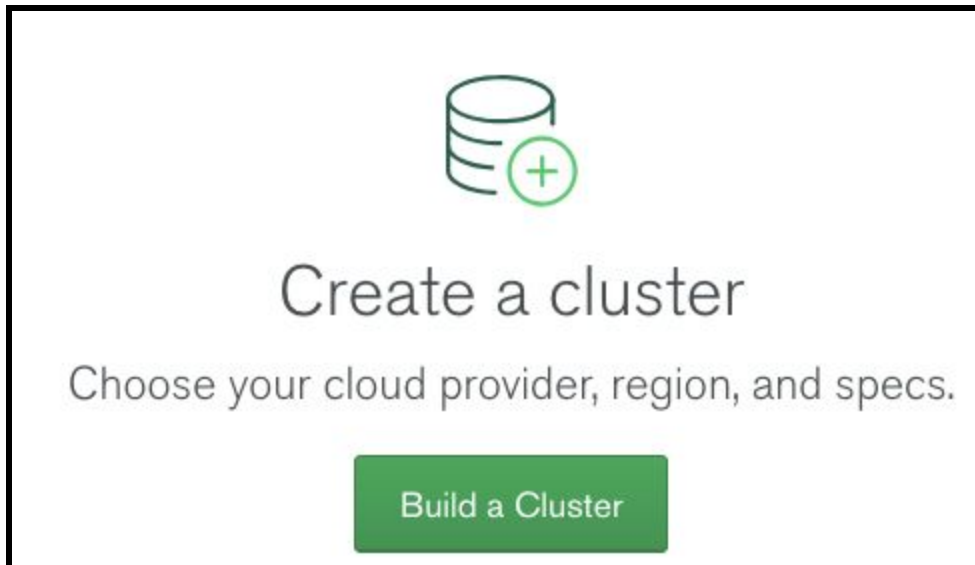
Lab 1 - Create the Cluster

Create an Account or Log In to Atlas

We'll be using [MongoDB Atlas](https://cloud.mongodb.com), 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**:



Take a moment to browse the options (Provider & Region, Cluster Tier, Version, Backup, ...). For our workshop, select **AWS** as the 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 ⓘ



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

Cluster Name

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

Workshop ▼

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.

Cancel Create Cluster

Continue to Lab 2 while the cluster is provisioning.

Lab 2 - Connect to the Cluster

Install Compass

Compass is the GUI for MongoDB. On the [download](#) page you need to select a Version and Platform:

Version

1.17.0 (Stable) ▾

Platforms

Windows 64-bit (7+) ▾

Download

Make sure you select the “Stable” version, which contains the enterprise features we’ll use in a moment.

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)

Also, if you lack the administrative privileges to install software on your Windows laptop, select the **Windows 64-bit (7+)** option, which is the executable.

OS X 64-bit (10.10+)

RedHat 64-bit (7+)

Ubuntu 64-bit (14.04+)

✓ Windows 64-bit (7+)

Windows 64-bit (7+) (MSI)

Windows 64-bit (7+) (Zip)

Go to <https://www.mongodb.com/download-center/compass> to download and install Compass for your platform.

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:

×

Connect to Workshop

Setup connection security

Choose a connection method

Connect

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

Add Your Current IP Address

Add a Different 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.

Username

ex. dbUser

Password

ex. dbUserPassword

SHOW

Autogenerate Secure Password

Create MongoDB User

Close

Choose a connection method

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	Description (Optional)
<input type="text" value="97.76.196.230"/>	<input type="text" value="Hilton Garden Inn"/>
<div><input type="button" value="Cancel"/> <input type="button" value="Add IP Address"/></div>	

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	Password
<input type="text" value="workshop"/>	<div><input type="password" value="....."/> <input type="button" value="SHOW"/></div>
<div><input type="button" value="Create MongoDB User"/></div>	

Click **Choose a connection method** and select **Connect with MongoDB Compass**.

Then select **I am using Compass 1.12 or later** and **COPY** the connection string presented:

×

Connect to Workshop

✓ Setup connection security

✓ Choose a connection method

Connect

1

If you have not already, click below to download Compass

Windows

Mac OS X

Other Operating Systems ▾

2


Copy the URI Connection String

[View detailed instructions](#)

I am using Compass 1.12 or later

I am using Compass 1.11 or earlier

mongodb+srv://workshop:<PASSWORD>@workshop-ghuzr.mongodb.net/admin

 COPY

Replace **PASSWORD** with the password for the *workshop* user.

When you open Compass, it should detect the URI string from your clipboard and auto-populate the form.

◀ Choose a connection method

Close

Connect Compass

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

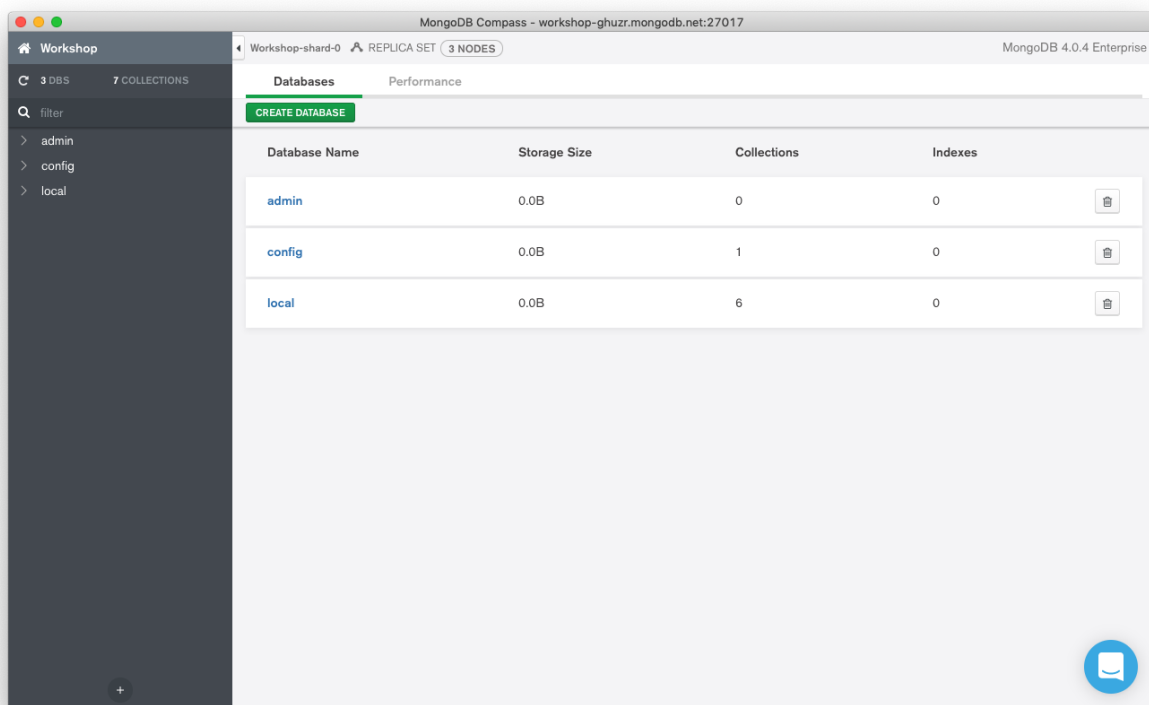


Select **Yes**.

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:



Lab 3 - Load Data

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

wget

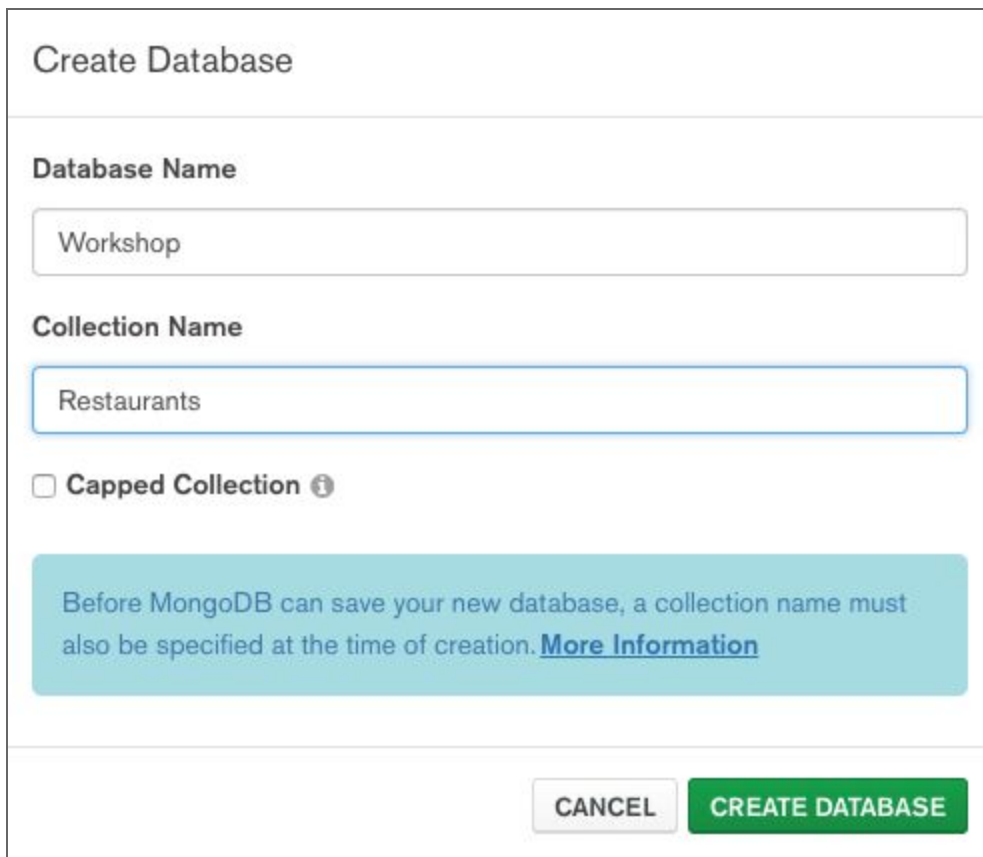
<https://raw.githubusercontent.com/mongodb/docs-assets/primer-dataset/primer-dataset.json>

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.

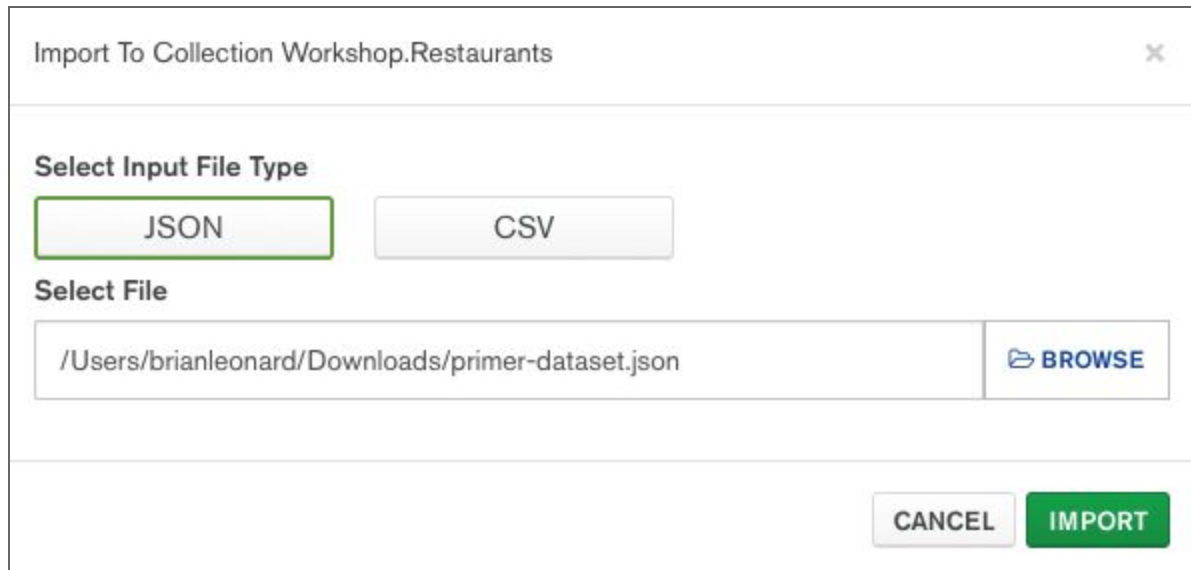
Create a Database and Collection

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



The screenshot shows the 'Create Database' dialog box in MongoDB Compass. It has a title bar 'Create Database'. Below the title bar, there are two input fields: 'Database Name' with the value 'Workshop' and 'Collection Name' with the value 'Restaurants'. Below these fields is a checkbox labeled 'Capped Collection' with an information icon. At the bottom of the dialog, there is a light blue informational message box that reads: 'Before MongoDB can save your new database, a collection name must also be specified at the time of creation. [More Information](#)'. At the very bottom, there are two buttons: 'CANCEL' and 'CREATE DATABASE'.

Navigate to the Restaurants collection and select **Import Data** from the menu. Then **BROWSE** to the primer-dataset.json file you downloaded:



Then select **IMPORT**. You've just imported 25K documents!

Lab 4 - Browse the Documents

Notice how 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

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**.

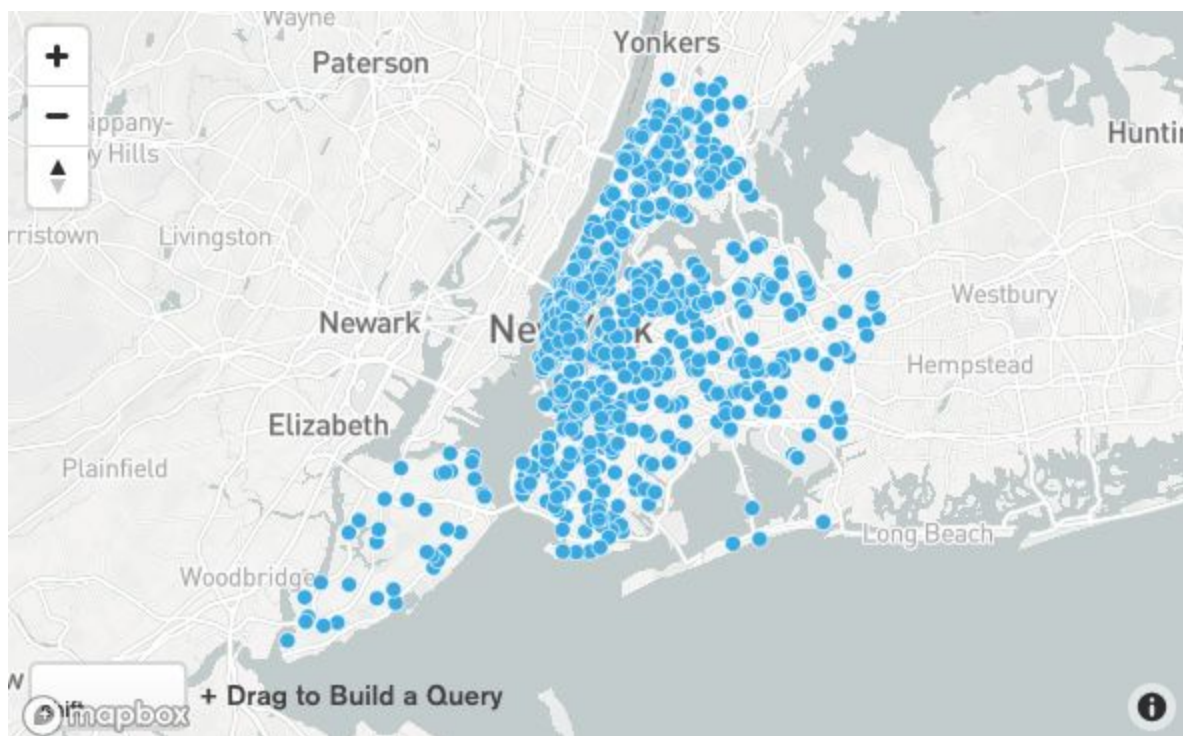
Select the **Schema** tab and select **Analyze Schema**.

*Note, if you don't have a **Schema** tab, you're running the Community Edition of Compass. Revist Lab 2 to get the correct edition.*

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 cuisine, we can see that Chinese is the 2nd most common at 12% (your results may differ slightly based on the sample that was taken):

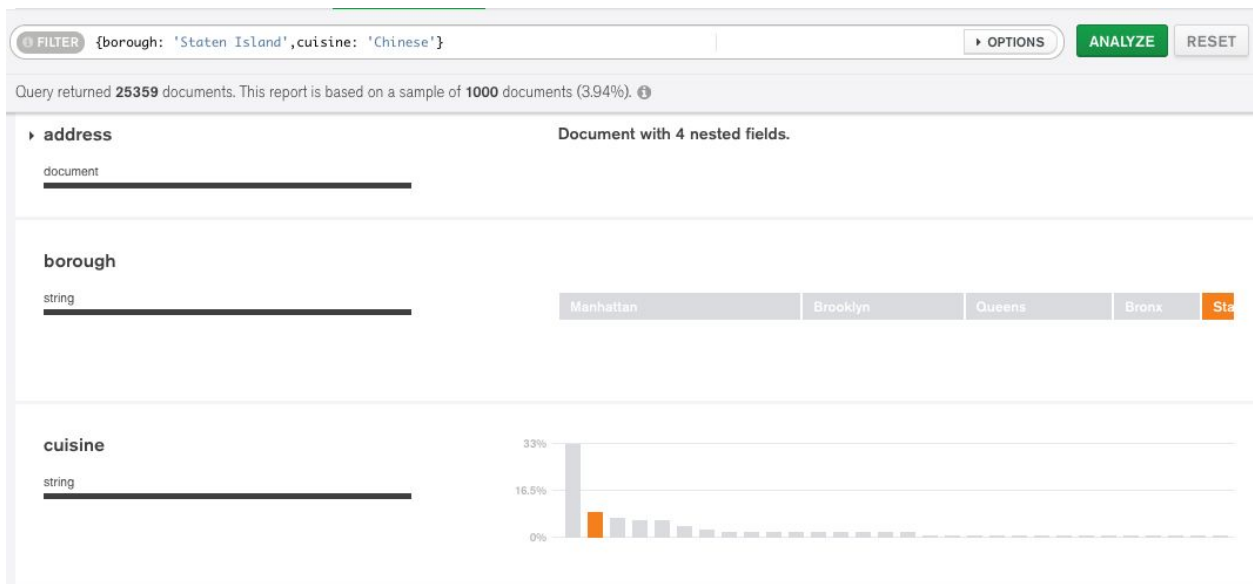


Expand the address field to discover MongoDB's excellent support for [Geospatial Queries](#). As the collection is of restaurants in New York City, zoom the map to NYC:

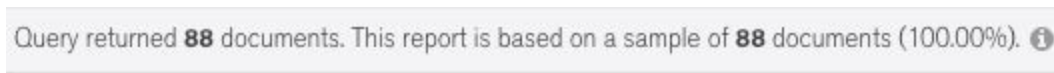


Lab 6 - Query the Data

Unsurprisingly, the MongoDB Query Language (MQL) is also 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:



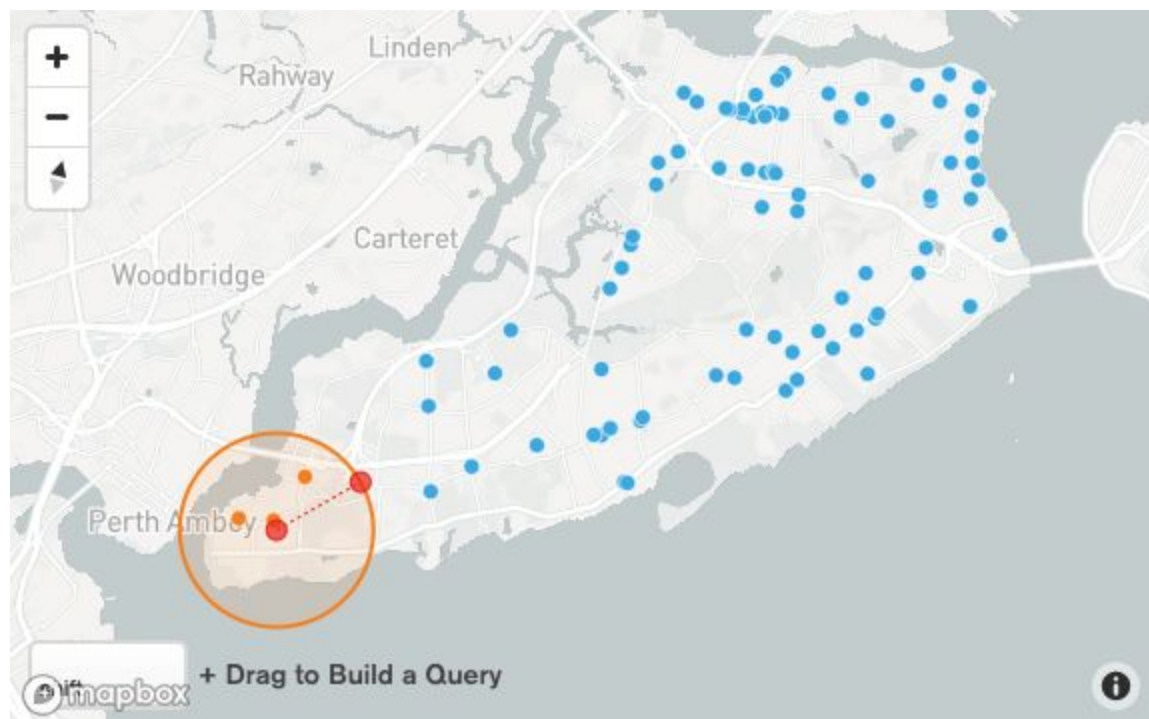
Click the **ANALYZE** button to filter for Chinese restaurants in Staten Island, of which there are 88:



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:



And notice the [\\$geoWithin](#) filter that got added to our query:



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

