Creating a MongoDB Cluster and inserting a document with Python

Garrett Sweeney Follow

Aug 19, 2019 · 3 min read ★

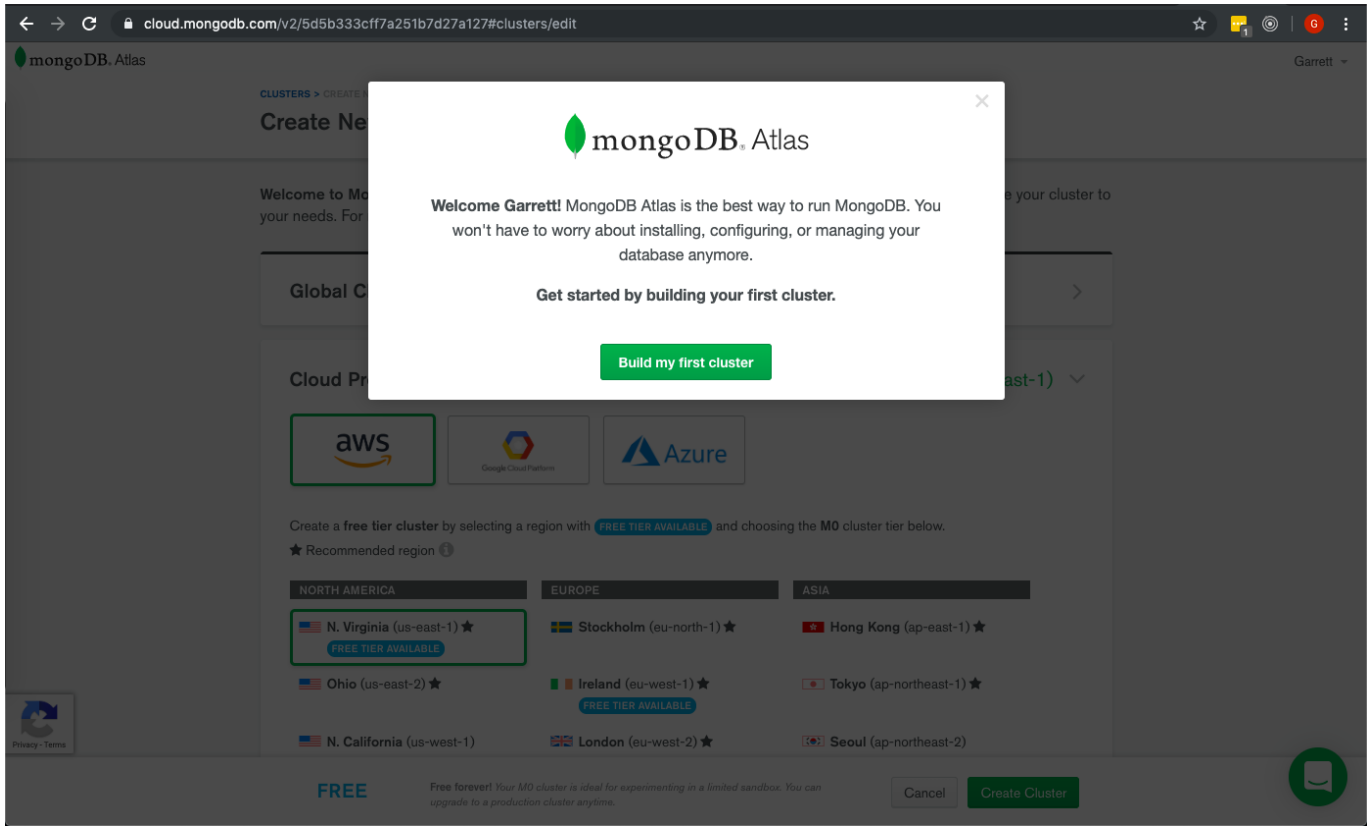
There are many guides on how to create a cluster and use python to insert into the cluster, but here's by experience creating my first MongoDB cluster.

Below you'll find "Part1: Cluster Creation" and "Part 2: Inserting a Document with Python".

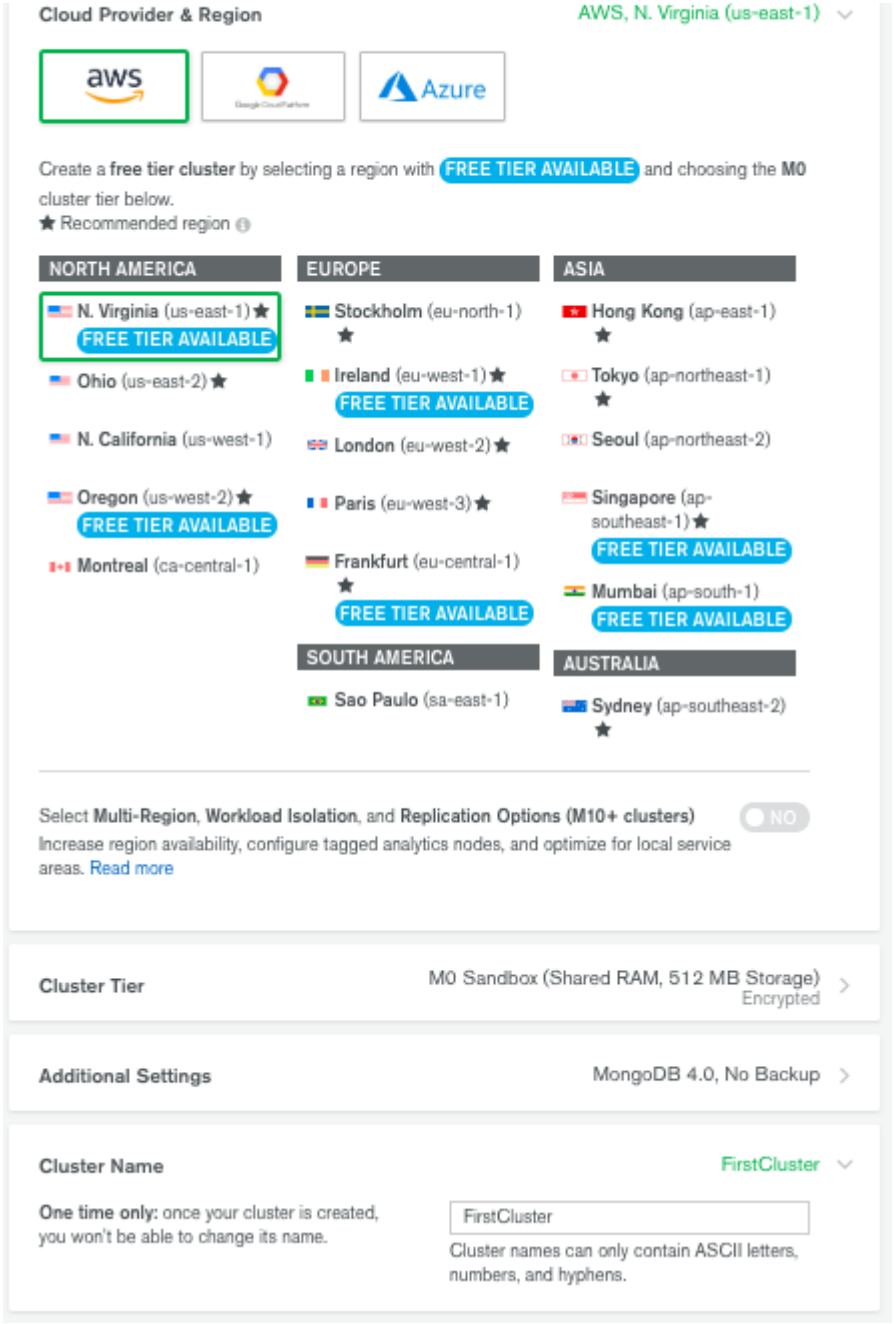
Part 1: Cluster Creation

1. Go to https://cloud.mongodb.com and create an account.

2. Upon creating an account, you may see a screen like this. Select “Build my first cluster”

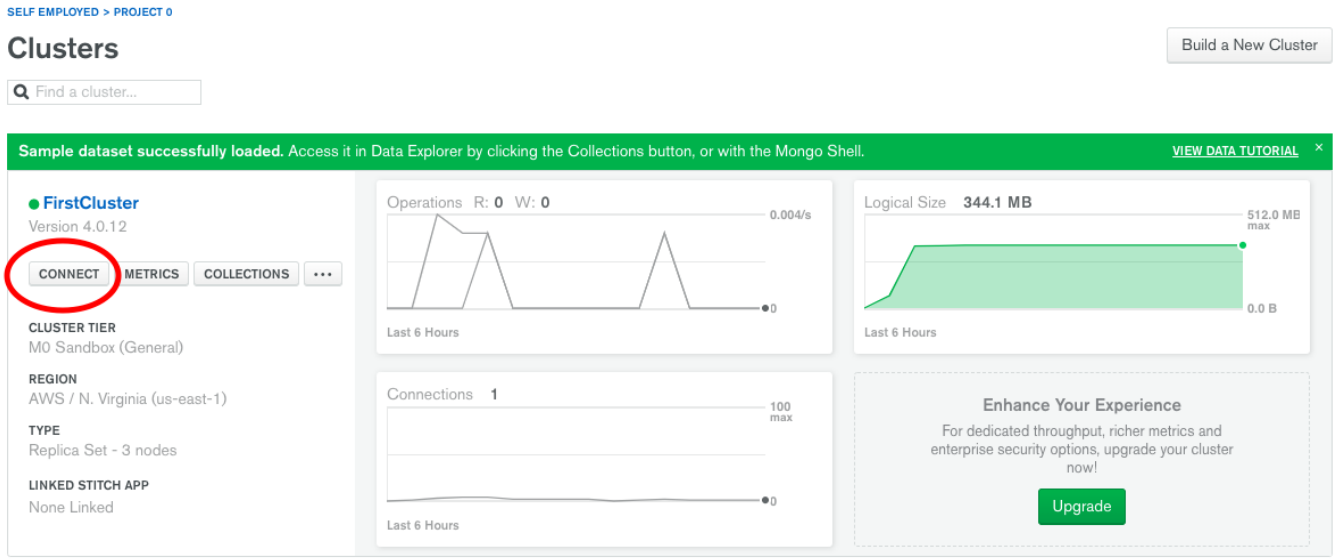


3. The default settings for creating a cluster in AWS should work. Just scroll down and type in a name for your cluster.

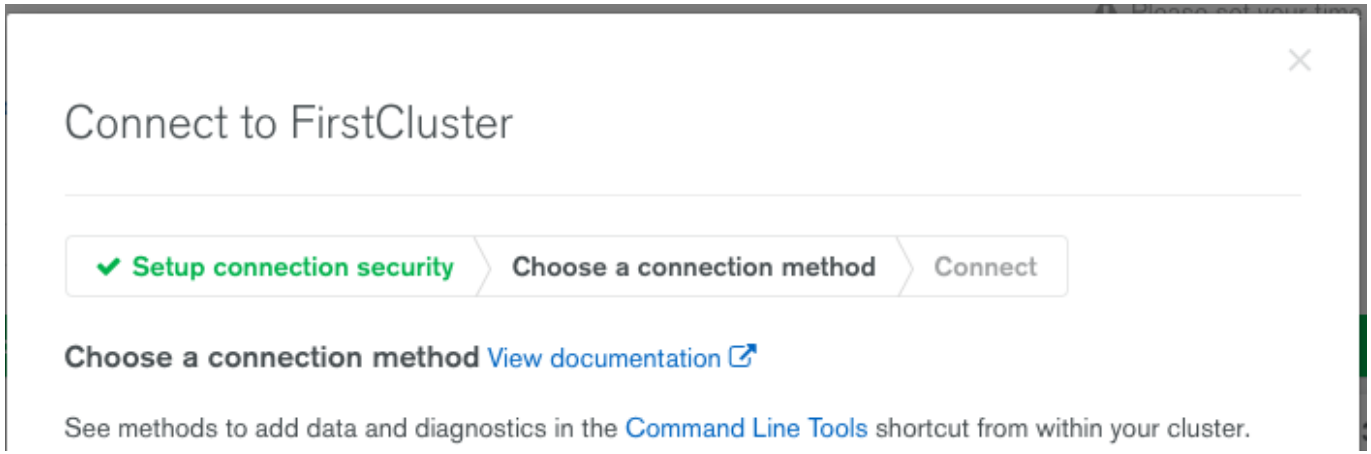


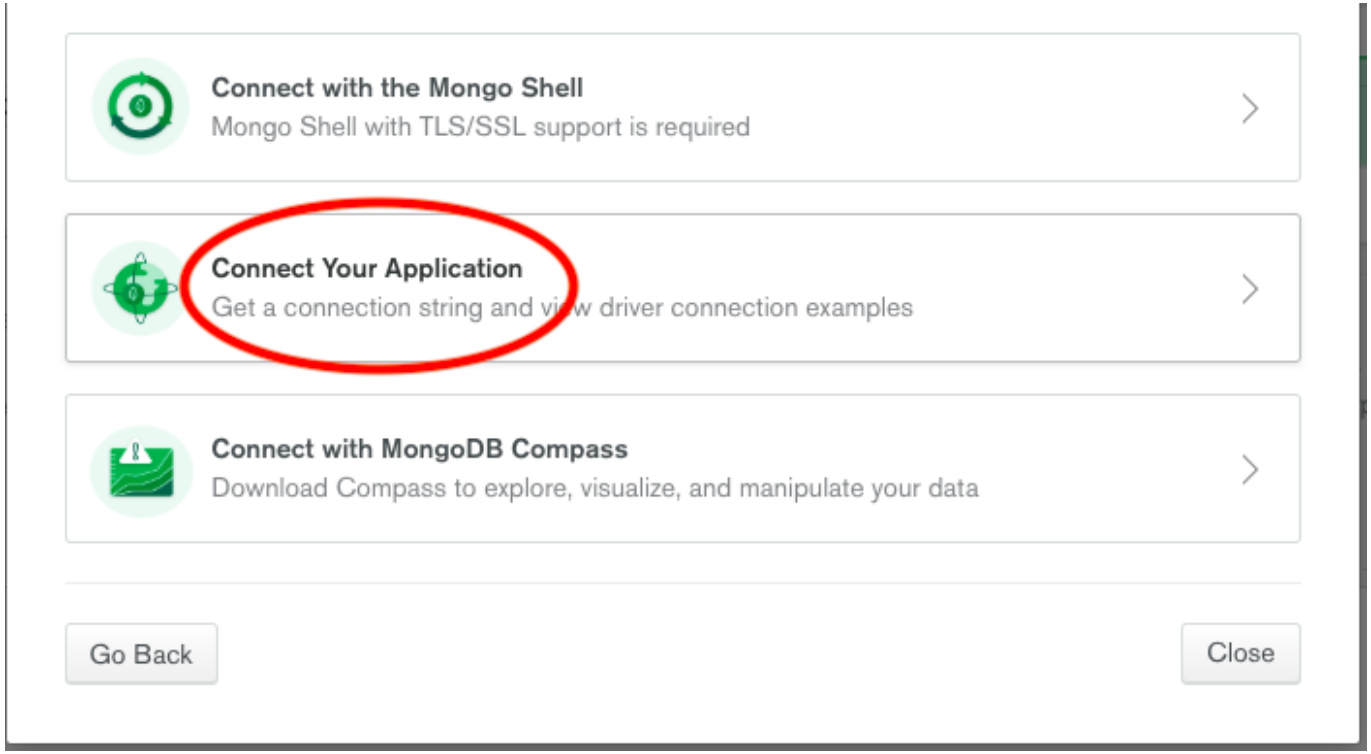
I named my cluster “FirstCluster”

4. After the cluster is created, you'll be able to see it in the console. Go ahead and click "Connection"

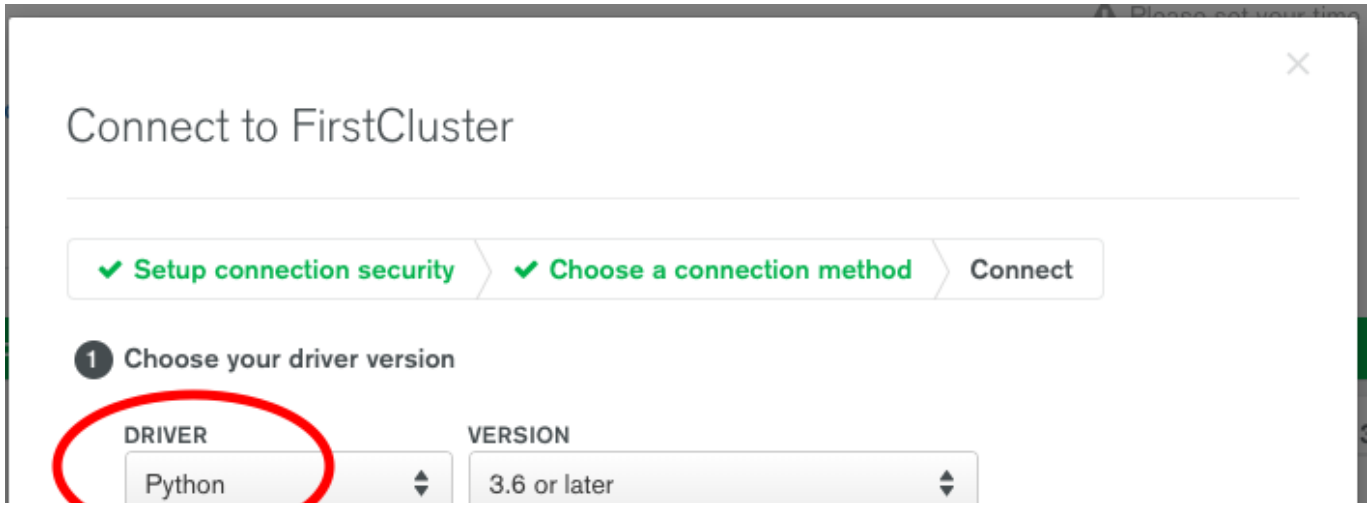


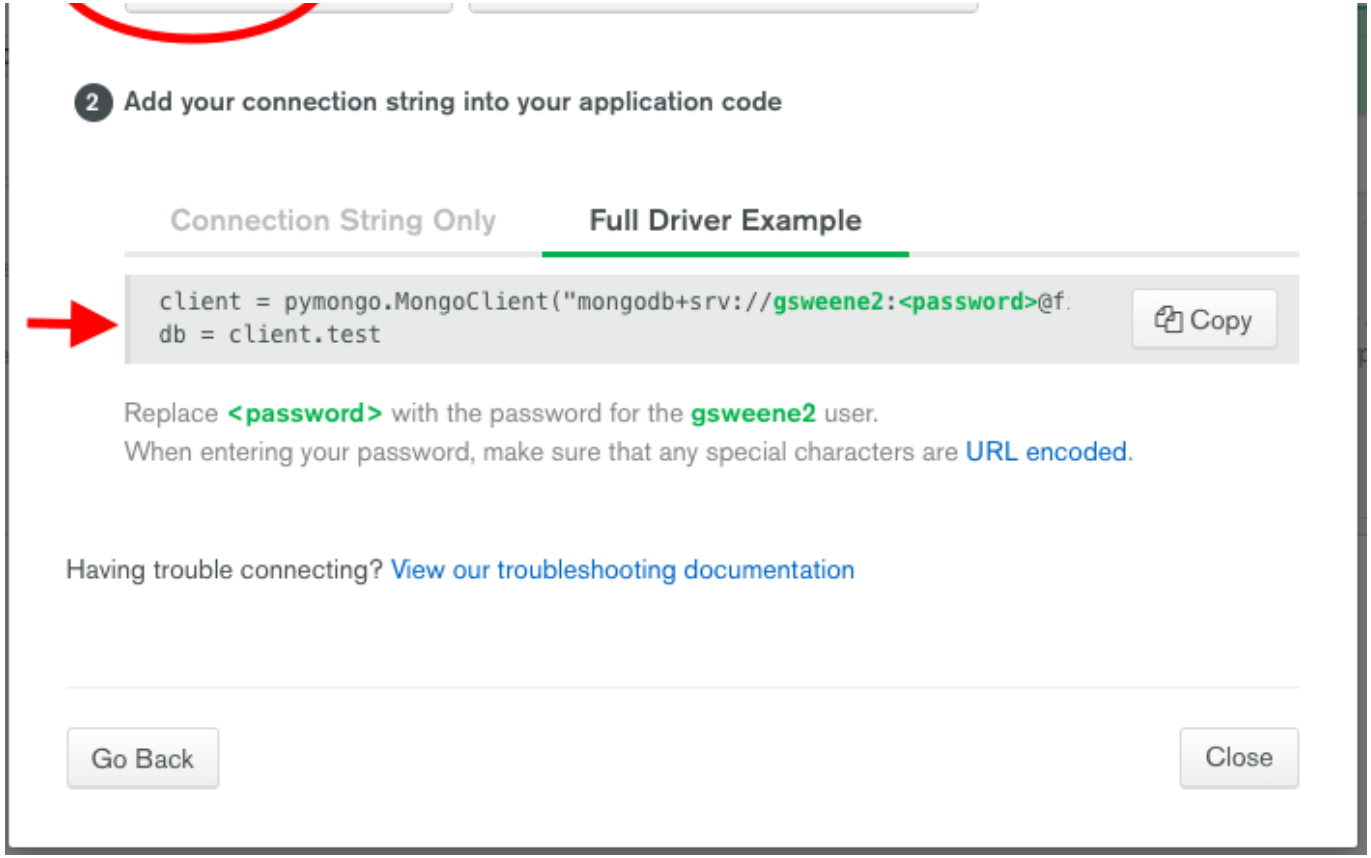
5. When prompted for connection method, click "Connect Your Application"



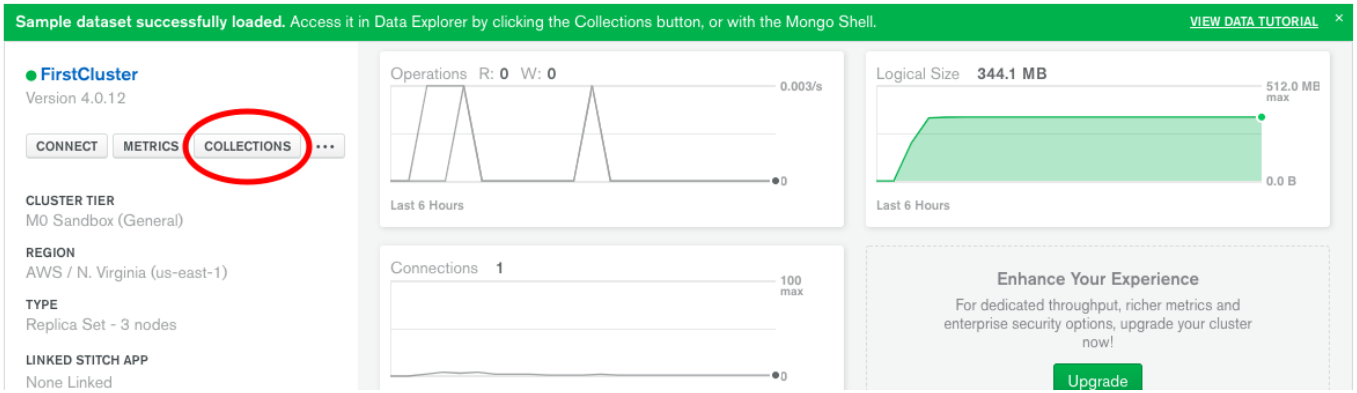


6. Select Python as the driver and take note of the Full Driver example.

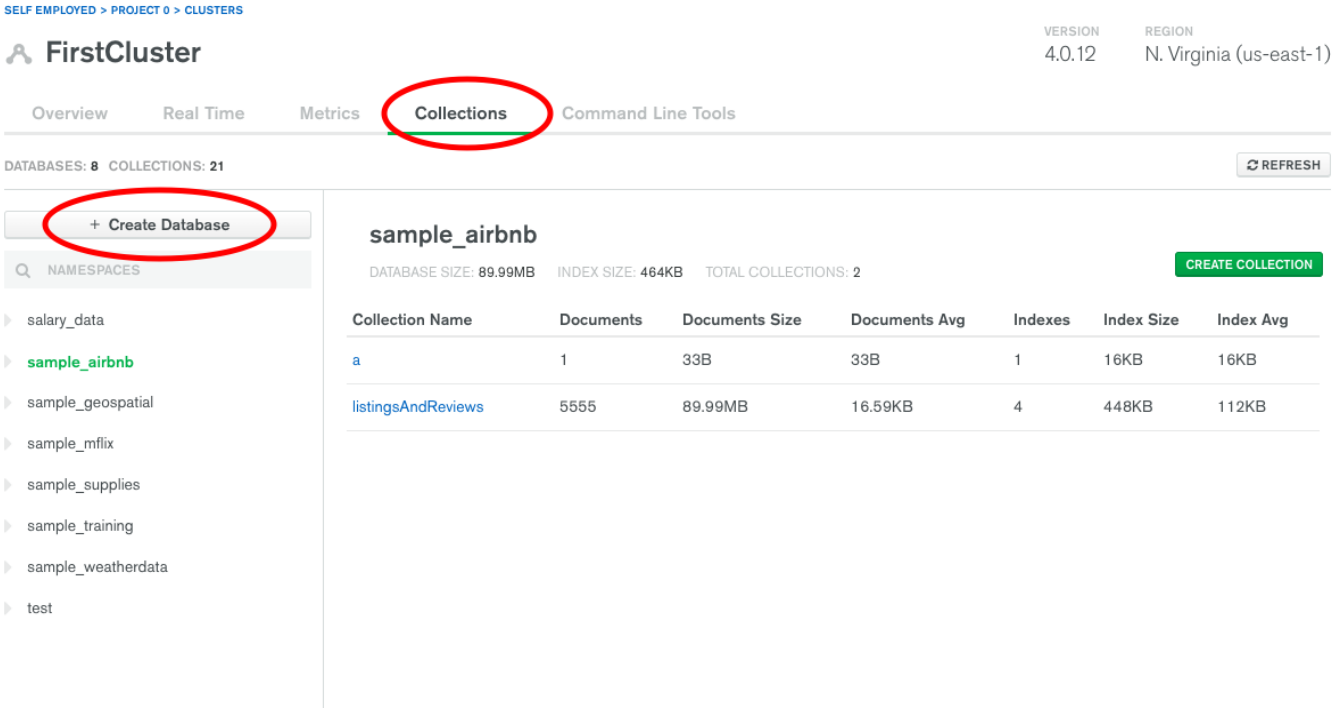




7. Now lets head back to the console and define a new Collections

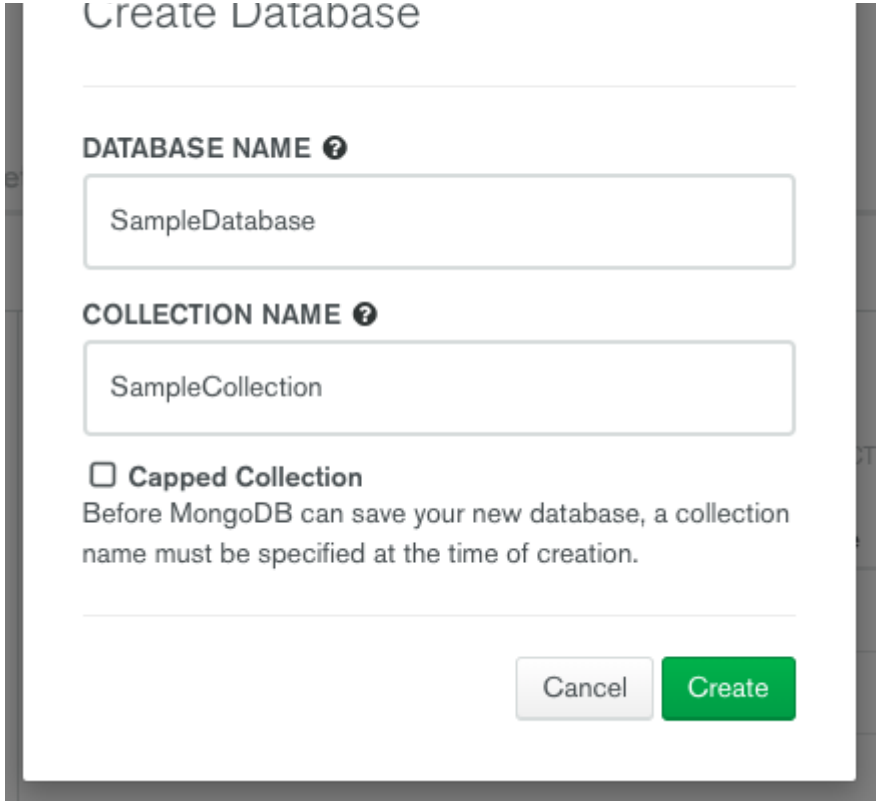


8. You'll see that I imported the sample data below, but we want to create a new database.

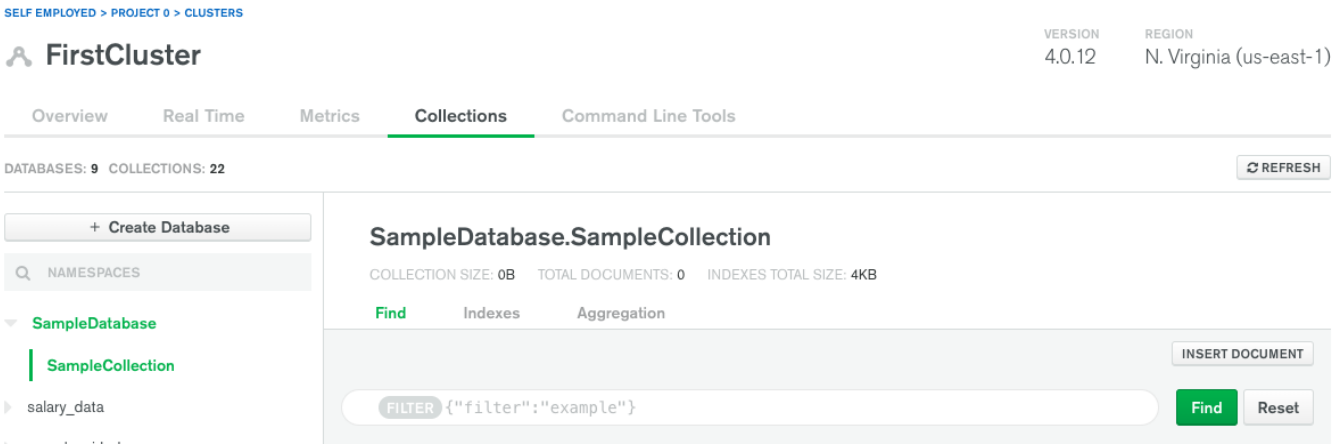


Click "Create Database"

9. Name your Database and Collection



10. After the DB is created, you should see it in the console



Part 2: Inserting a Document with Python

1. Head over to your workspace and create a new folder

```
cd ~/git
mkdir mongo-python-example
cd mongo-python-example
```

2. Create a `requirements.txt` file and a `mongo_util.py` file

```
touch requirements.txt
touch mongo_util.py
```

3. Populate your `mongo_util.py` file with the following, and replace `<PASSWORD>` with your mongo user password.

```
# import datetime module

import datetime

# import pymongo module

import pymongo
```

```
# connection string

client = pymongo.MongoClient("mongodb+srv://gsweene2:
<PASSWORD>@firstcluster-obuqd.mongodb.net/test?
retryWrites=true&w=majority")

# test

db = client['SampleDatabase']

# define collection

collection = db['SampleCollection']

# sample data

document = {"company":"Capital One",

"city":"McLean",

"state":"VA",

"country":"US"}

# insert document into collection

id = collection.insert_one(document).inserted_id

print("id")

print(id)
```

4. Notice we have 2 modules imported above. Add those to your `requirements.txt` file.

```
datetime

dnspython

pymongo
```

5. Next, create a `virtualenv` for our small python script.

```
# install virtualenv

pip install --user virtualenv

# Create virtualenv in repo

virtualenv .

# Activate Virtualenv

source ./bin/activate

# Install dependencies
```

```
pip install -r requirements.txt
```

6. Now you should be able to run your code and insert an item! If you note in the last line of our script, we print out the id of the newly created mongo document.

```
$ python mongo_util.py

id

5d5b53c7ad555762b067cb3b
```

You can also see the new document in the console.

Whoop, there it is

What's Next?

In the next post I'll turn this into a Flask API. Then I'll deploy it with terraform to AWS, and use API Gateway to allow us to create documents with our API.

. . .

Code on Github

Python Mongodb Mongodb Atlas AWS