# Using MongoDB with Jupyter

- You will need to install the necessary Python library, `pymongo`

```
!conda install --yes --quiet pymongo
```

- This should be run from a Jupyter server that is not on the same AWS instance as your Mongo server.

# pymongo

- `pymongo` is a Python module containing the MongoDB tools recommended for working with the database.

- You begin by instantiating a connection to MongoDB using `pymongo.MongoClient`.

- Here, you use the IP address of your AWS instance on which MongoDB is running.

```
from pymongo import MongoClient
client = MongoClient('255.255.255.255', 27016)
```

# pymongo

- Mongo has a very useful “get or create” mechanism for both databases and collections.

```
client.database_names()
```

# pymongo

- Databases and collections are accessed using either attribute-style (`client.database_ name`) or dictionary-style (`client['test-database']`).

- If the database or collection exists, this method will return a reference to the existing database or collection ("get"). If the database does not exists, this method will create the database or collection and then return a reference to it ("create").

# pymongo

- The creation happens at the time of insertion of a document.

```
db_ref = client.my_database
client.database_names()
```

```
coll_ref = db_ref.my_collection
client.database_names(), db_ref.collection_names()
```

```
sample_doc = {"name":"Joshua", "message":"Hi!",
'my_array' :[1,2,3,4,5,6,7,9]}
coll_ref.insert_one(sample_doc)
```

```
client.database_names(), db_ref.collection_names()
```