Course Overview

View Discussion

Chapter 1: Driver Setup

README

Download Course Materials

Handouts (1)

m220/mflix-python.zip

In order to run properly, the MFlix software project has some installation requirements and environmental dependencies.

These requirements and dependencies are defined in this lesson, and they can also be found in the **README.rst** file from the **mflix-python** project, which you will download shortly. This lesson serves as a guide for setting up these necessary tools. After following this README, you should be able to successfully run the MFlix application. First, you will need to download the **mflix-python** project, as described below.

Download the mflix-python.zip file

You can download the **mflix-python.zip** file by clicking the link in the "Handouts" section of this page. Downloading this handout may take a few minutes. When the download is complete, unzip the file and `cd` into the project's root directory, **mflix-python**.

COPY

```
cd ~/Downloads
unzip mflix-python.zip
cd mflix-python
```

Project Structure

Everything you will implement is located in the `mflix/db.py` file, which contains all database interfacing methods. The API will make calls to `db.py` to interact with MongoDB.

The unit tests in `tests` will test these database access methods directly, without going through the API. The UI will run these methods in integration tests, and therefore requires the full application to be running.

The API layer is fully implemented, as is the UI. If you need to run on a port other than `5000`, you can edit the **index.html** file in the build directory to modify the value of **window.host**.

Please do not modify the API layer in any way, `movies.py` and `user.py` under the **mflix/api** directory. Doing so will most likely result in the frontend application failing to validate some of the labs.

Local Development Environment Configuration

Anaconda

We're going to use Anaconda to install Python 3 and to manage our Python 3 environment.

Installing Anaconda for Mac

You can download Anaconda from their MacOS download site. The installer will give you the option to "Change Install Location", so you can choose the path where the `anaconda3` folder will be placed. Remember this location, because you will need it to activate the environment.

Once installed, you will have to create and activate a `conda` environment:

COPY

```
# navigate to the mflix-python directory
cd mflix-python

# enable the "conda" command in Terminal
echo ". /anaconda3/etc/profile.d/conda.sh" >> ~/.bash_profile
source ~/.bash_profile

# create a new environment for MFlix
conda create --name mflix

# activate the environment
conda activate mflix
```

You can deactivate the environment with the following command:

```
conda deactivate
```

COPY

Installing Anaconda for Windows

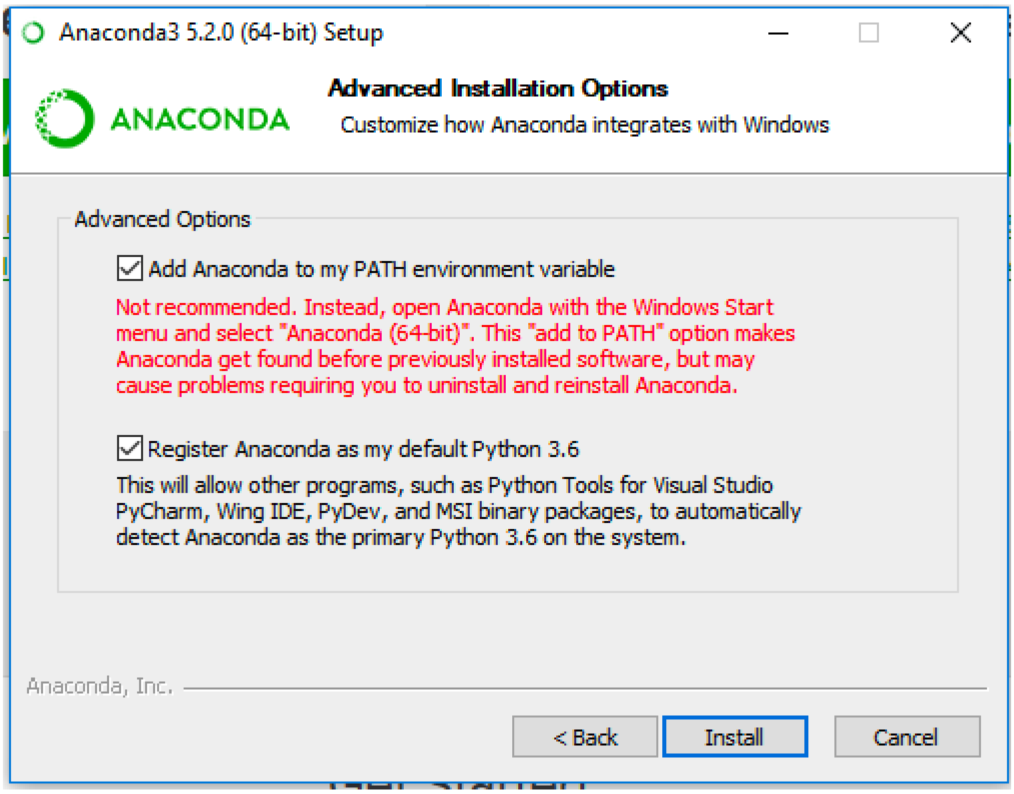
You can download Anaconda from their Download site. Please be careful to select `Windows Tab` before downloading.

The Anaconda installer will prompt you for the following options:

- *Add Anaconda to my PATH environment variable*
- *Register Anaconda as my default Python 3.6*

Please select both of these options. The first option will allow you to use `conda` commands from the Command Prompt, and the second option will allow you to use Anaconda's Python 3.6 as your system's default.

You may see a red error message like the following:



Advanced Installation Options

Customize how Anaconda integrates with Windows

Advanced Options

☑ Add Anaconda to my PATH environment variable

Not recommended. Instead, open Anaconda with the Windows Start menu and select "Anaconda (64-bit)". This "add to PATH" option makes Anaconda get found before previously installed software, but may cause problems requiring you to uninstall and reinstall Anaconda.

☑ Register Anaconda as my default Python 3.6

This will allow other programs, such as Python Tools for Visual Studio PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.6 on the system.

Anaconda, Inc.

This is expected. Please select both of the options above.

If you forget to select the *PATH* option before installing, no worries. The installer will let you choose an "Install Location" for Anaconda, which is the directory where the `Anaconda3` folder will be placed.

Using your machine's location of `Anaconda3` as `<path-to-Anaconda3>`, run the following commands to activate `conda` commands from the Command Prompt:

```
set PATH=%PATH%;<path-to-Anaconda3>;<path-to-Anaconda
```

Once Anaconda is installed, you will have to create and enable a **`conda`** environment.

```
# enter mflix-python folder
cd mflix-python/data/mflix

# create a new environment for MFlix
conda create --name mflix

# activate the environment
activate mflix
```

You can deactivate the environment with the following command:

```
deactivate
```

Virtualenv

Note: If you installed Anaconda instead, skip this step.

As an alternative to Anaconda, you can also use **`virtualenv`**, to define your Python 3 environment. You are required to have a Python 3 installed in your workstation.

You can find the virtualenv installation procedure on the PyPA website.

Once you've installed Python 3 and **`virtualenv`**, you will have to setup a **`virtualenv`** environment:

```
# navigate to the mflix-python directory
cd mflix-python

# create the virtual environment for MFlix
virtualenv -p YOUR_LOCAL_PYTHON3_PATH mflix_venv

# activate the virtual environment
source mflix_venv/bin/activate
```

You can deactivate the virtual environment with the following command:

```
deactivate
```

COPY

Python Library Dependencies

Once the Python 3 environment is activated, we need to install our python dependencies. These dependencies are defined in the `requirements.txt` file, and can be installed with the following command:

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

COPY

MongoDB Installation

It is recommended to connect MFlix with MongoDB Atlas, so you do not need to have a MongoDB server running on your host machine. The lectures and labs in this course will assume that you are using an Atlas cluster instead of a local instance.

That said, you are still required to have the MongoDB server installed, in order to be able to use two server tool dependencies:

- `mongorestore`
  - A utility for importing binary data into MongoDB.
- `mongo`
  - The MongoDB shell

To download these command line tools, please visit the MongoDB download center and choose the appropriate platform.

MongoDB Atlas Cluster

MFlix uses MongoDB to persist all of its data.

One of easiest ways to get up and running with MongoDB is to use MongoDB Atlas, a hosted and fully-managed database solution.

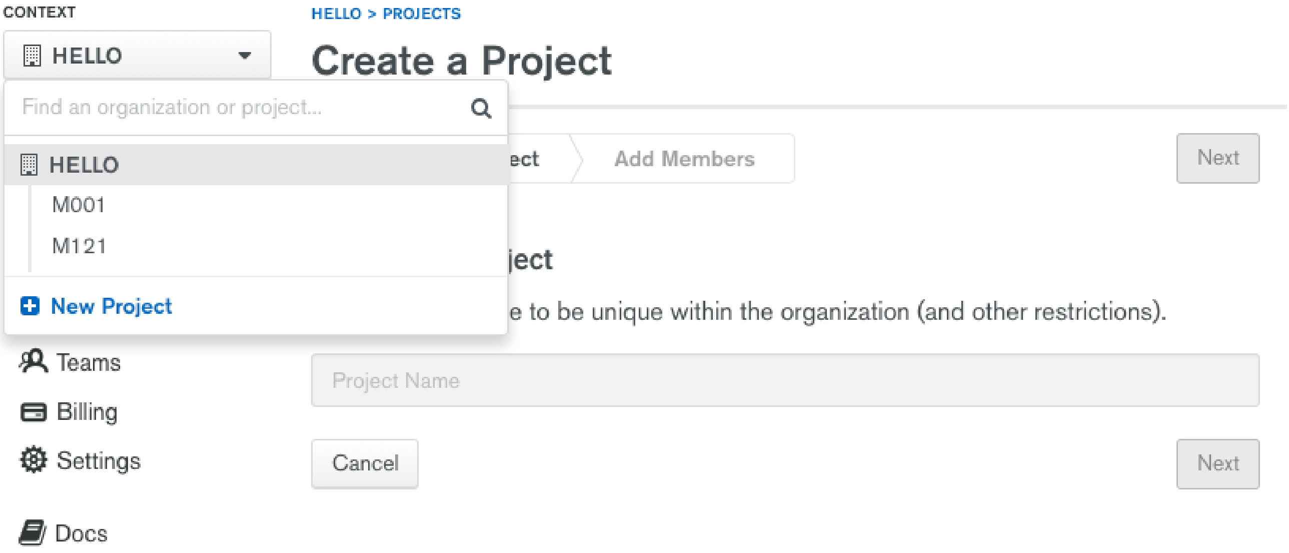
If you have taken other MongoDB University courses like M001 or M121, you may already have an account - feel free to reuse that cluster for this course.

Note: Be advised that some of the UI aspects of Atlas may have changed since the inception of this README, therefore some of the screenshots in this file may be different from the actual Atlas UI interface.

Using an existing MongoDB Atlas Account:

If you already have a previous Atlas account created, perhaps because you've taken one of our other MongoDB university courses, you can repurpose it for M220P.

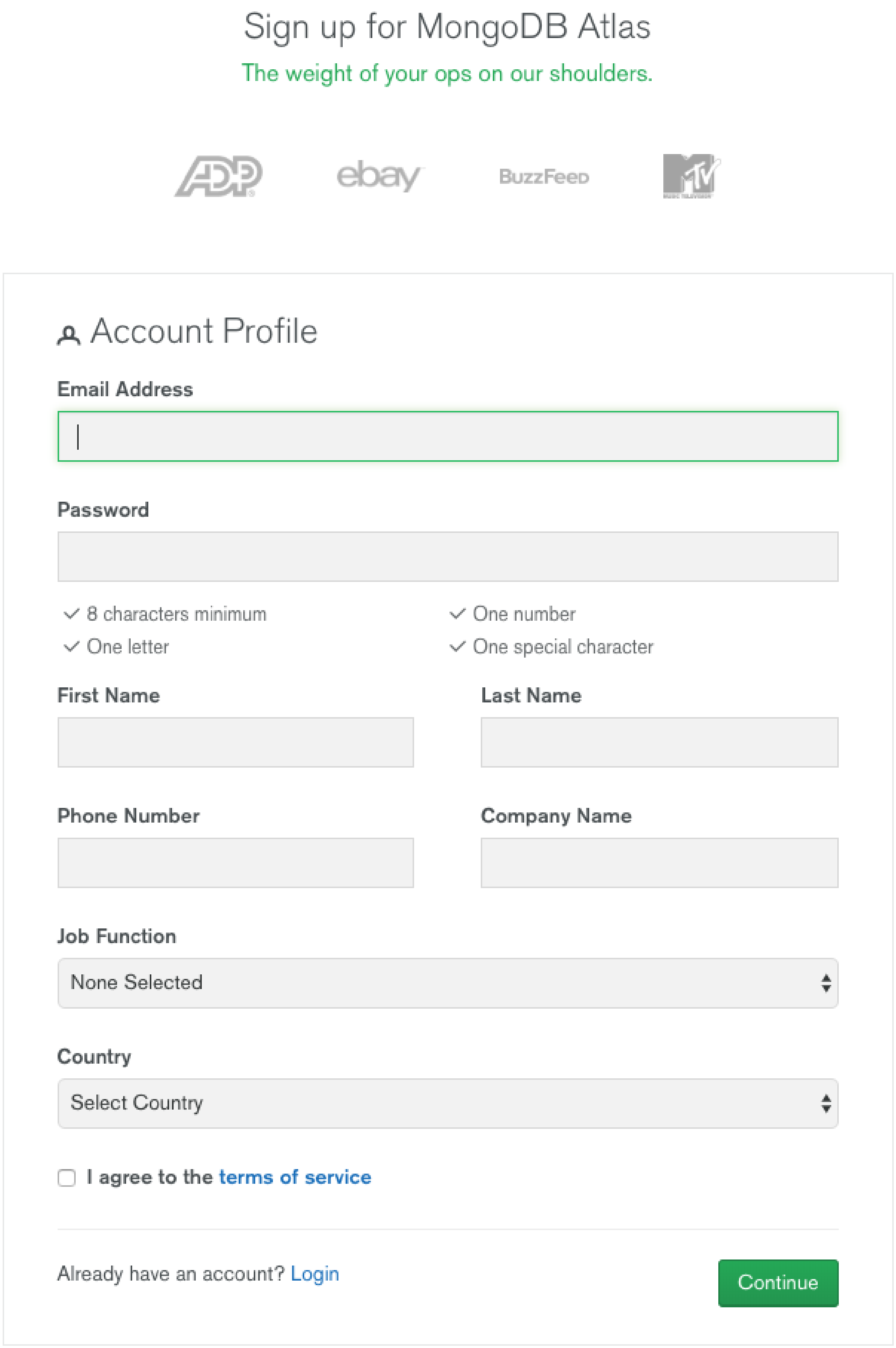
Log-in to your Atlas account and create a new project named **M220** by clicking on the *Context* dropdown menu:



After creating a new project, you need to create an **mflix** free tier cluster.

Creating a new MongoDB Atlas Account:

If you do not have an existing Atlas account, go ahead and create an Atlas Account by filling in the required fields:



Sign up for MongoDB Atlas

The weight of your ops on our shoulders.

Account Profile

Email Address

Password

✓ 8 characters minimum

✓ One letter

✓ One number

✓ One special character

First Name

Last Name

Phone Number

Company Name

Job Function

None Selected

Country

Select Country

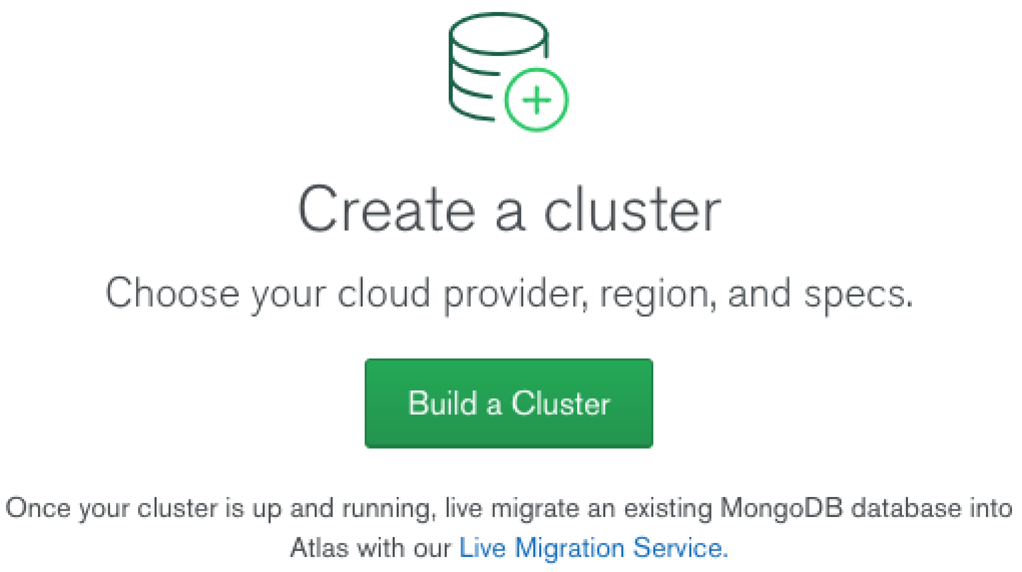
□ I agree to the terms of service

Already have an account? Login

Creating a free tier cluster called "mflix":

Note: You will need to do this step even if you are reusing an Atlas account.

1. After creating a new project, you will be prompted to create the first cluster in that project:

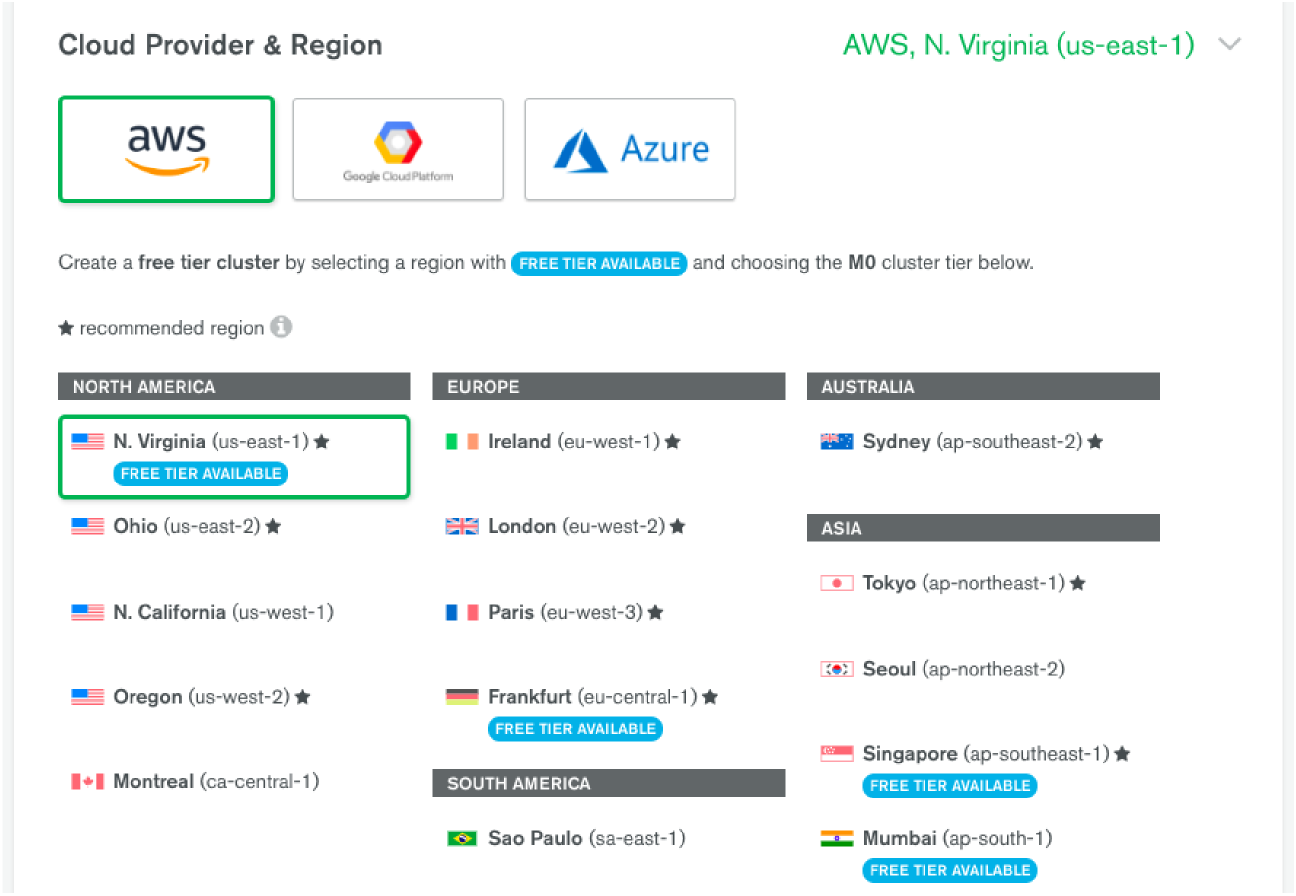


Create a cluster

Choose your cloud provider, region, and specs.

Once your cluster is up and running, live migrate an existing MongoDB database into Atlas with our Live Migration Service.

2. Choose AWS as the cloud provider, in a Region that has the label `FREE TIER AVAILABLE`:



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) ★
- **Montreal** (ca-central-1)

EUROPE

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

SOUTH AMERICA

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

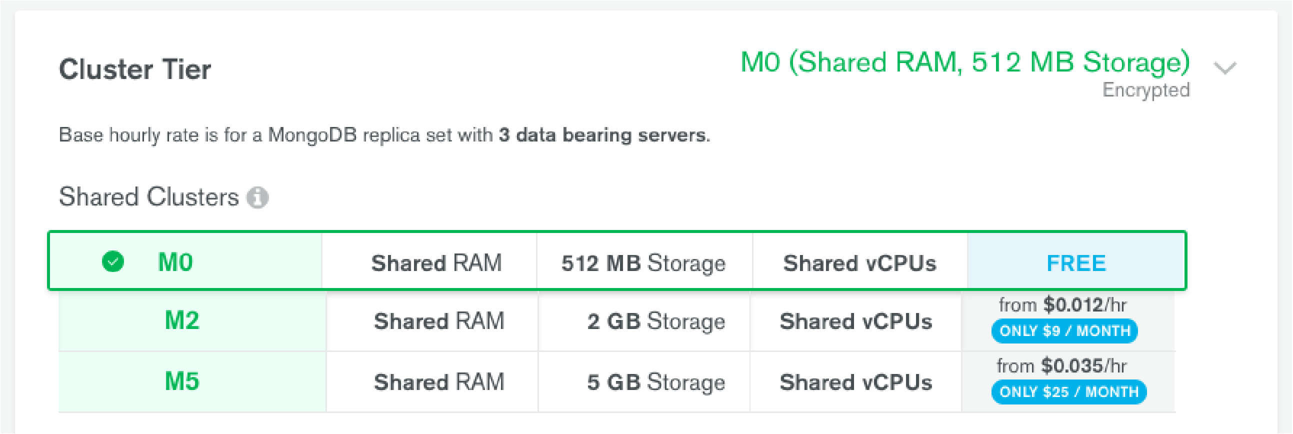
AUSTRALIA

- **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

3. Select *Cluster Tier* M0:



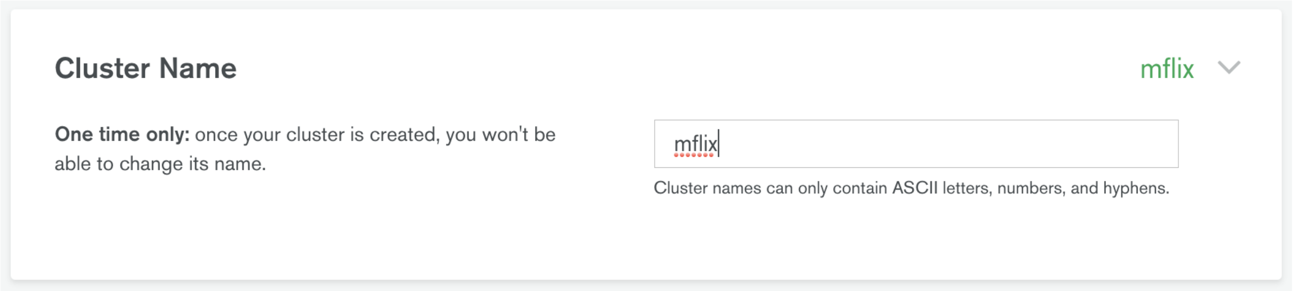
Cluster Tier M0 (Shared RAM, 512 MB Storage) Encrypted

Base hourly rate is for a MongoDB replica set with **3 data bearing servers**.

Shared Clusters

✓ M0	Shared RAM	**512 MB** Storage	**Shared vCPUs**	FREE
M2	Shared RAM	**2 GB** Storage	**Shared vCPUs**	from **$0.012**/hr ONLY $9 / MONTH
M5	Shared RAM	**5 GB** Storage	**Shared vCPUs**	from **$0.035**/hr ONLY $25 / MONTH

4. Set *Cluster Name* to **mflix** and click *Create Cluster*:



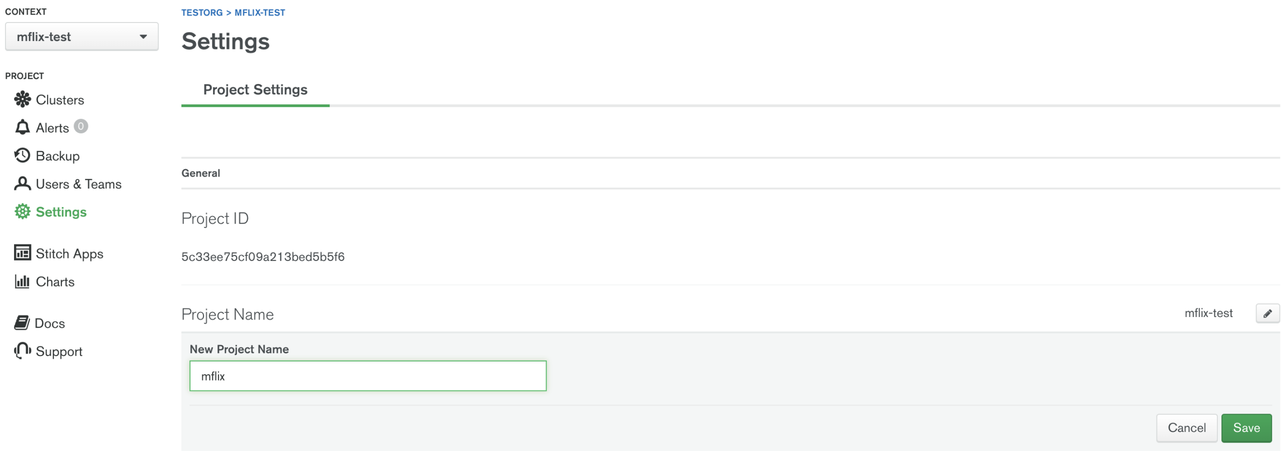
Cluster Name mflix

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

mflix

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

5. Once you press *Create Cluster*, you will be redirected to the account dashboard. In this dashboard, make sure you set your project name to **M220**. Go to `Settings` menu item and change the project name from the default **Project 0** to **M220**:



Settings

Project Settings

General

Project ID

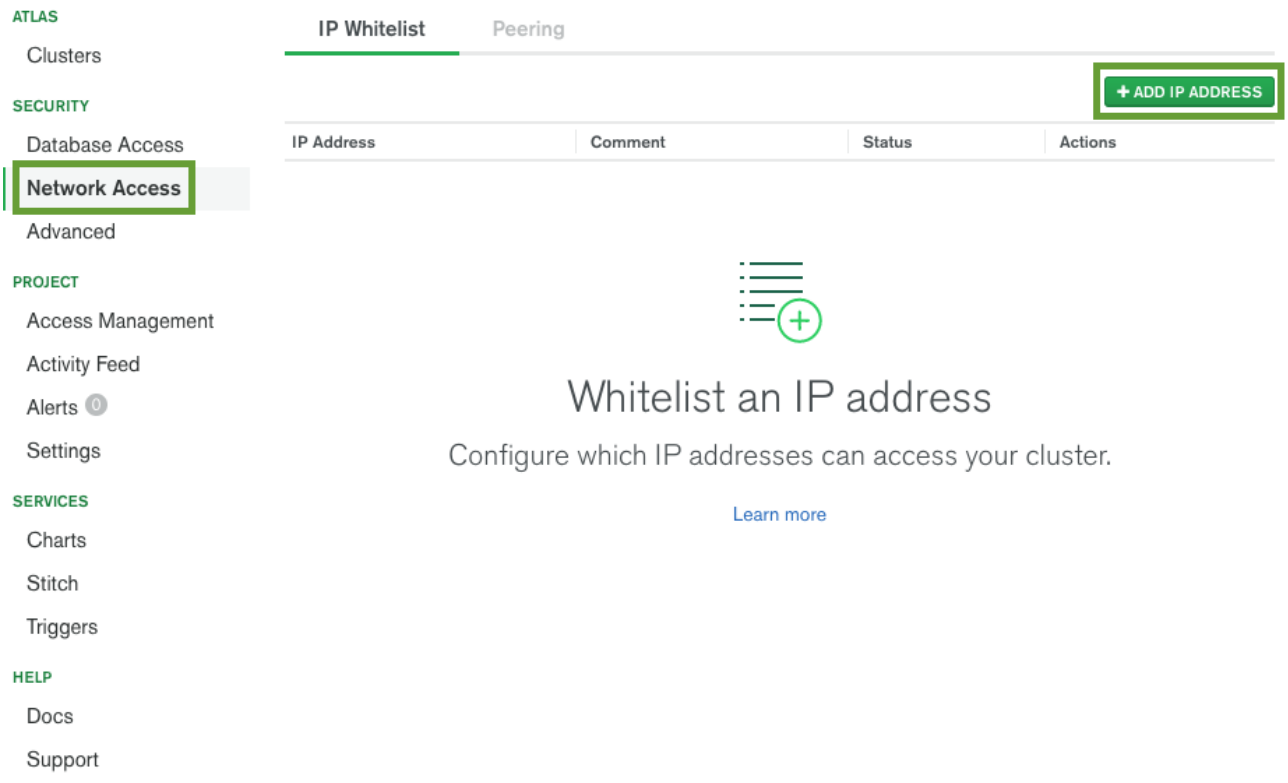
5c33ee75cf09a213bed5b5f6

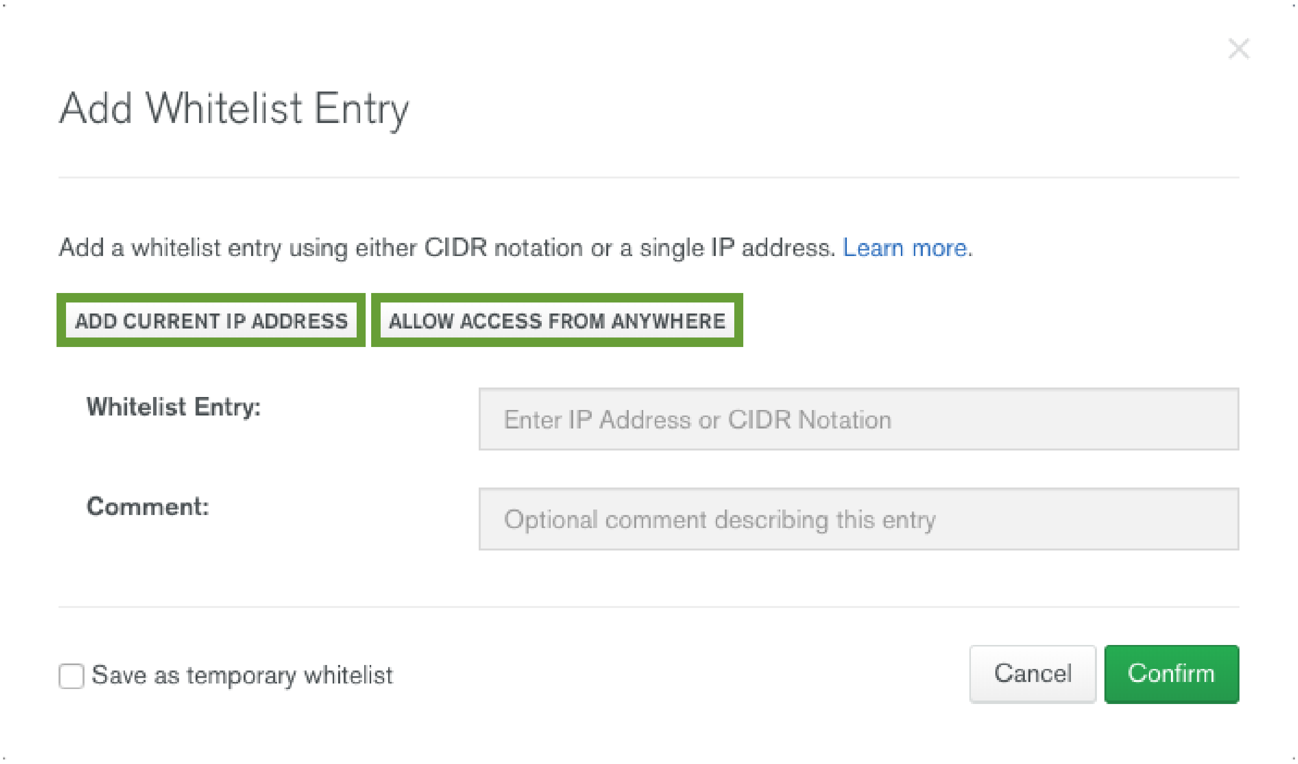
Project Name mflix-test

New Project Name

mflix

6. Configure the network settings of this cluster in the **Network Access** tab, so you can connect from your IP address. When you select `ADD IP ADDRESS`, the menu that appears may give you the option to add `ADD CURRENT IP ADDRESS`. This is preferable, but if this option does not appear, please select `ALLOW ACCESS FROM ANYWHERE`:



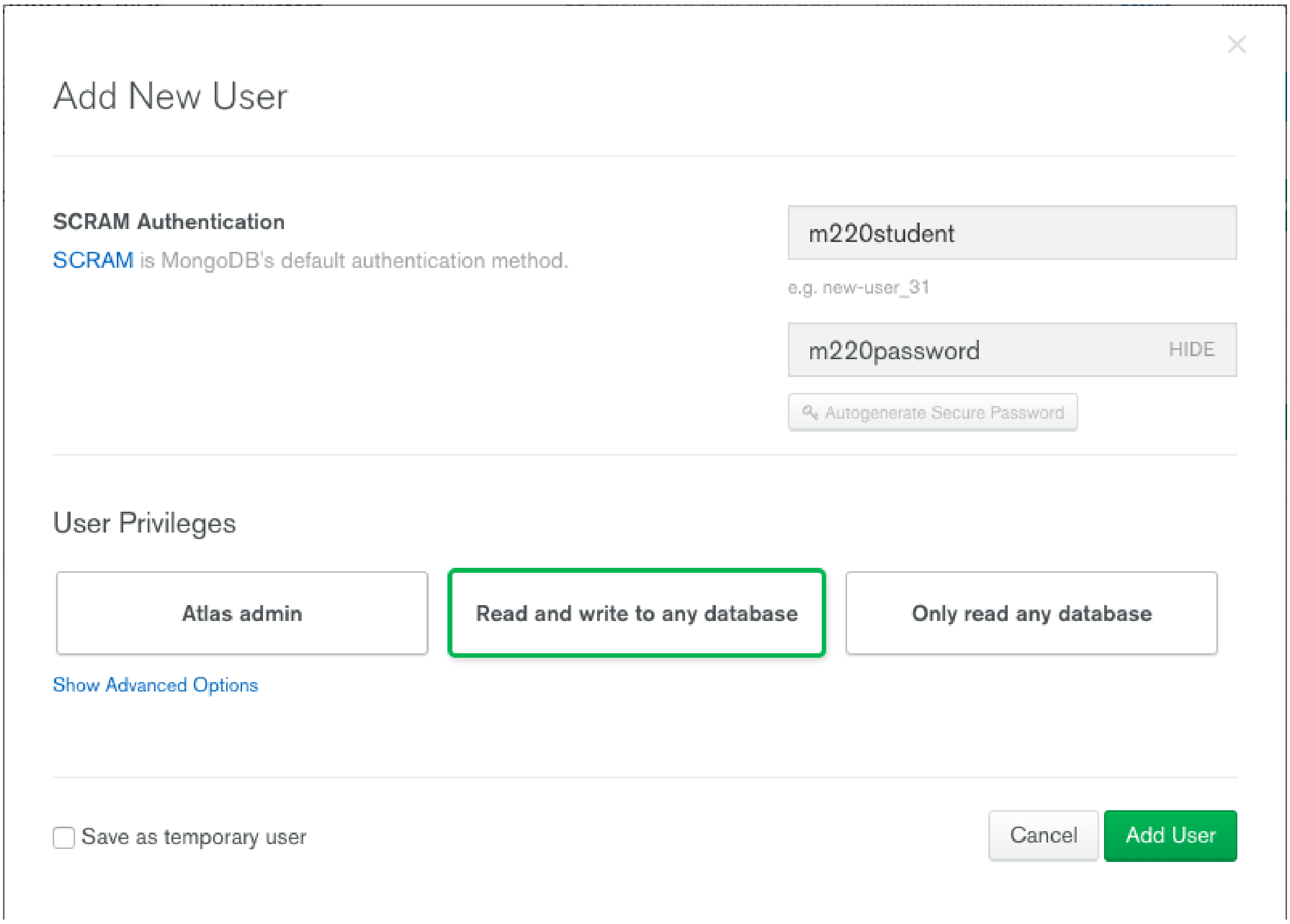

Add Whitelist Entry
Add a whitelist entry using either CIDR notation or a single IP address. Learn more.
ADD CURRENT IP ADDRESS
ALLOW ACCESS FROM ANYWHERE
Whitelist Entry:
Enter IP Address or CIDR Notation
Comment:
Optional comment describing this entry
□ Save as temporary whitelist
Cancel
Confirm

7. Then create the application MongoDB database user required for this course:

- username: **m220student**
- password: **m220password**

You can create new users through *Security -> Add New User*.

Allow this user the privilege to `Read and write to any database`:



Add New User

SCRAM Authentication

SCRAM is MongoDB's default authentication method.

m220student

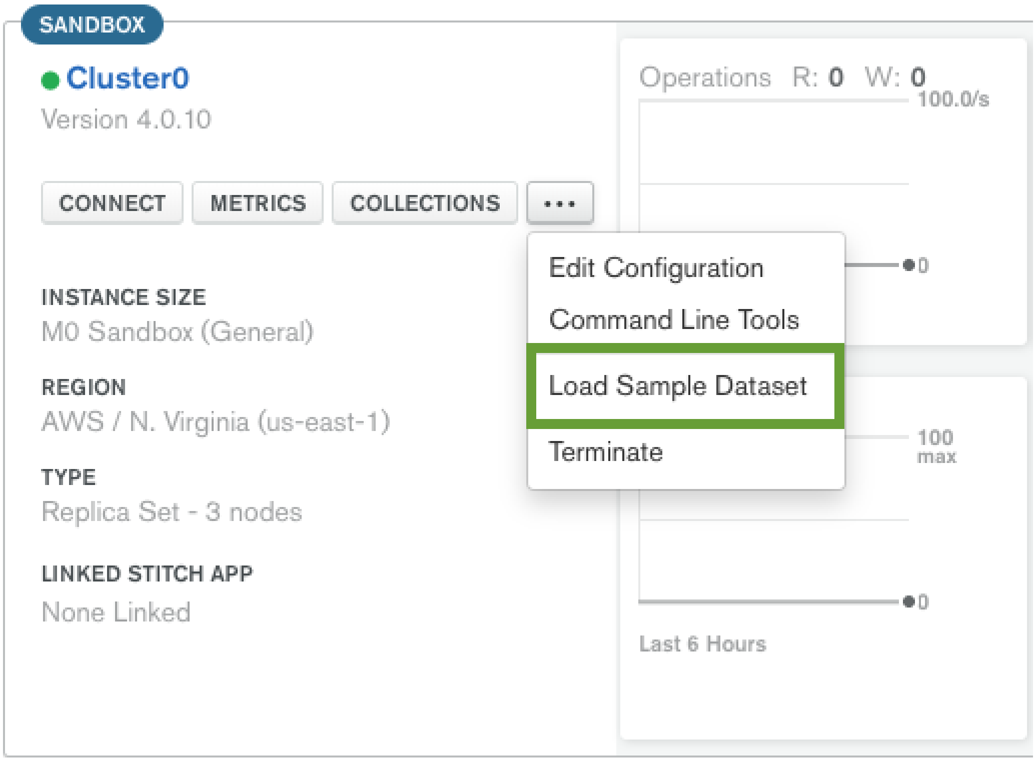
e.g. new-user_31

m220password

User Privileges

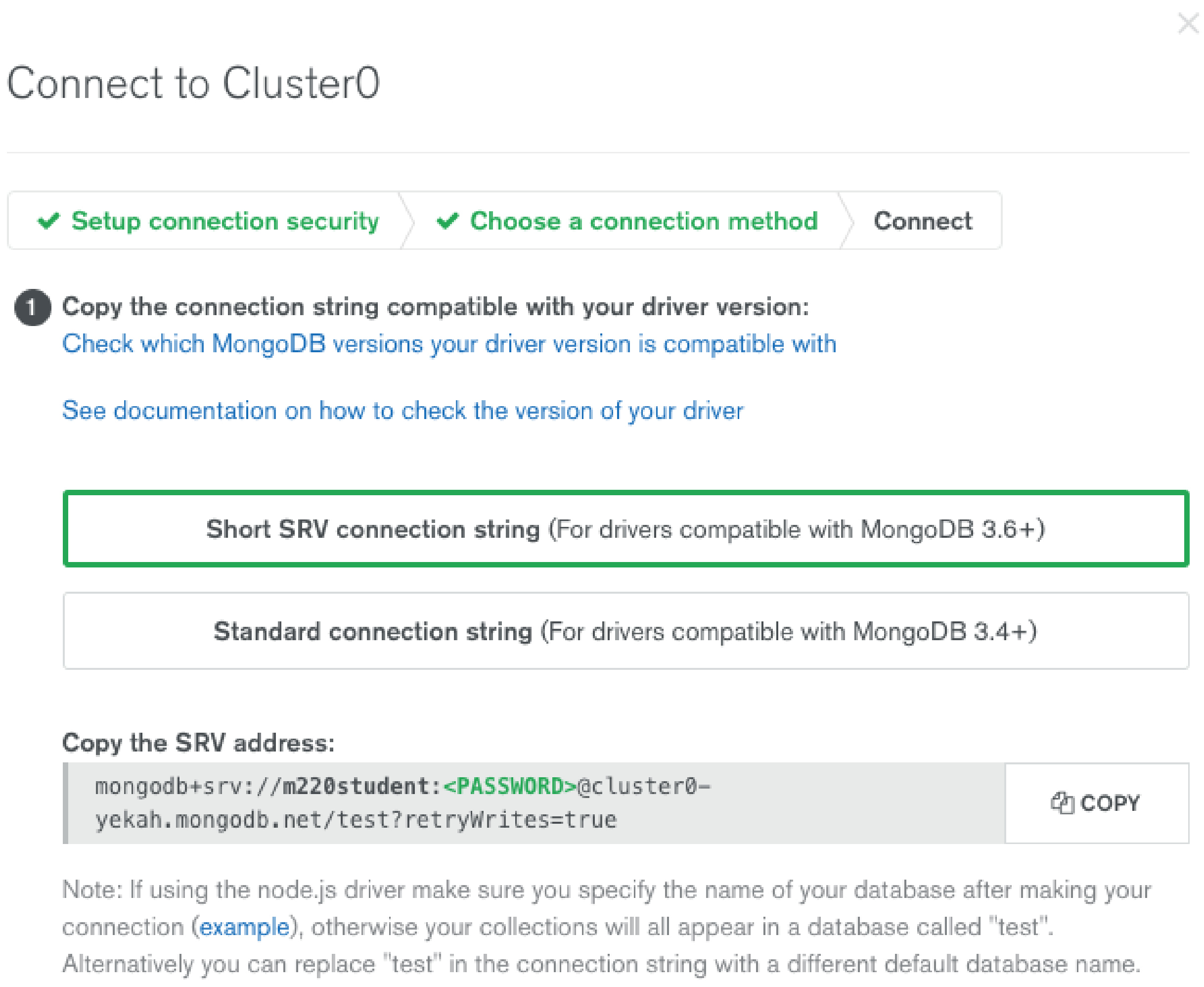
☐ Save as temporary user

8. When the user is created, and the cluster deployed, you have the option to `Load Sample Dataset`. This will load the Atlas sample dataset, containing the MFlix database, into your cluster:



Note: The MFlix database in the Sample Dataset is called "sample_mflix".

9. Now you can test the setup by connecting via `mongo` shell. You can find instructions to connect in the *Connect* section of the cluster dashboard:



Connect to Cluster0

1 Copy the connection string compatible with your driver version:
Check which MongoDB versions your driver version is compatible with

See documentation on how to check the version of your driver

Short SRV connection string (For drivers compatible with MongoDB 3.6+)

Standard connection string (For drivers compatible with MongoDB 3.4+)

Copy the SRV address:

```
mongodb+srv://m220student:<PASSWORD>@cluster0-
yekah.mongodb.net/test?retryWrites=true
```

Note: If using the node.js driver make sure you specify the name of your database after making your connection (example), otherwise your collections will all appear in a database called "test". Alternatively you can replace "test" in the connection string with a different default database name.

Go to your cluster *Overview* -> *Connect* -> *Connect Your Application*. Select the option corresponding to your local MongoDB version and copy the `mongo` connection string.

The below example connects to Atlas as the user you created before, with username `m220student` and password `m220password`. You can run this command from your command line:

```
mongo "mongodb+srv://m220student:m220password@<YOUR_(
```

COPY

By connecting to the server from your host machine, you have validated that the cluster is configured and reachable from your local workstation.

The connection string you used to connect will be used in your MFlix application as well.

Importing Data (Optional)

Note: if you used Load Sample Dataset, you can skip this step.

The `mongorestore` command necessary to import the data is located below. Copy the command and use the Atlas SRV string to import the data (including username and password credentials).

Replace the SRV string below with your own:

COPY

```
# navigate to mflix-python directory
cd mflix-python

# import data into Atlas
mongorestore --drop --gzip --uri
mongodb+srv://m220student:m220password@<YOUR_CLUSTER_URI> data
```

Running the Application

In the `mflix-python` directory you can find a file called `dotini`.

Open this file and enter your Atlas SRV connection string as directed in the comment. This is the information the driver will use to connect. Make sure **not** to wrap your Atlas SRV connection between quotes:

```
MFLIX_DB_URI = mongodb+srv://...
```

COPY

Rename this file to `.ini` with the following command:

```
mv dotini_unix .ini  # on Unix
ren dotini_win .ini # on Windows
```

COPY

Note: Once you rename this file to `.ini`, it will no longer be visible in Finder or File Explorer. However, it will be visible from Command Prompt or Terminal, so if you need to edit it again, you can open it from there:

```
vi .ini       # on Unix
notepad .ini  # on Windows
```

COPY

To start MFlix, run the following command:

```
python run.py
```

COPY

This will start the application. You can then access the MFlix application at http://localhost:5000/.

Running the Unit Tests

To run the unit tests for this course, you will use `pytest`. Each course lab contains a module of unit tests that you can call individually with a command like the following:

```
pytest -m LAB_UNIT_TEST_NAME
```

COPY

Each ticket will contain the command to run that ticket's specific unit tests.

Proceed to next section