Projeto de prática Integrada de ciência de dados, inteligência artificial e machine learning



Relatório de avistamento de Objetos Voadores Não Identificados.

Sprint 3 → Armazenamento em MongoDB

Curso: Tecnologia em sistemas para internet

Estudantes:

Brenda Lopes Miranda Teixeira

Mateus Gomes da Silva Fonteles

Rickson Queiroz Marques de Souza

Samuel Araújo Lopes

Professores

Fábio Henrique

Diego Queiroz

Ana Régia

Brasília, agosto de 2021

Sumário

1. Objetivos 3
2. Descrição do problema 4
3. Desenvolvimento 5
   1. Criação do BD no MongoDB Atlas 6
   2. Código implementado 11
4. Considerações Finais 14

Referências 15

# Objetivos

Na última *sprint* do trabalho vamos armazenar os dados com os quais estivemos trabalhando em um banco não relacional, utilizando o MongoDB Atlas. Criaremos um cluster gratuito e vamos salvar dentro deste os dados que até então estivemos salvando em arquivos .CSV.

# Descrição do problema

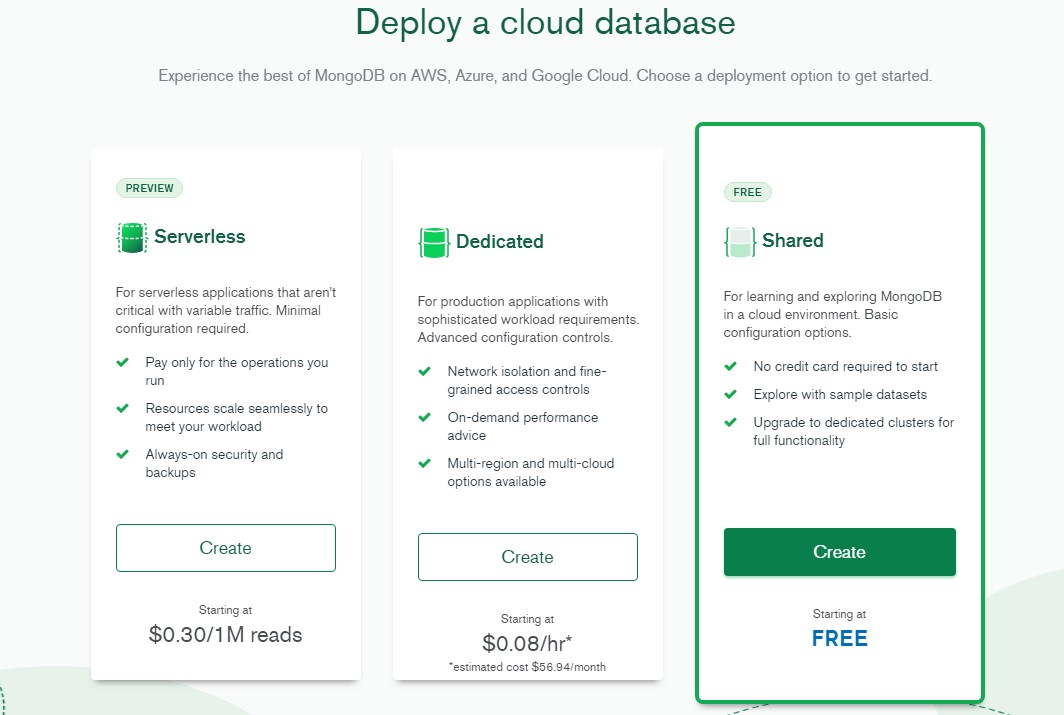
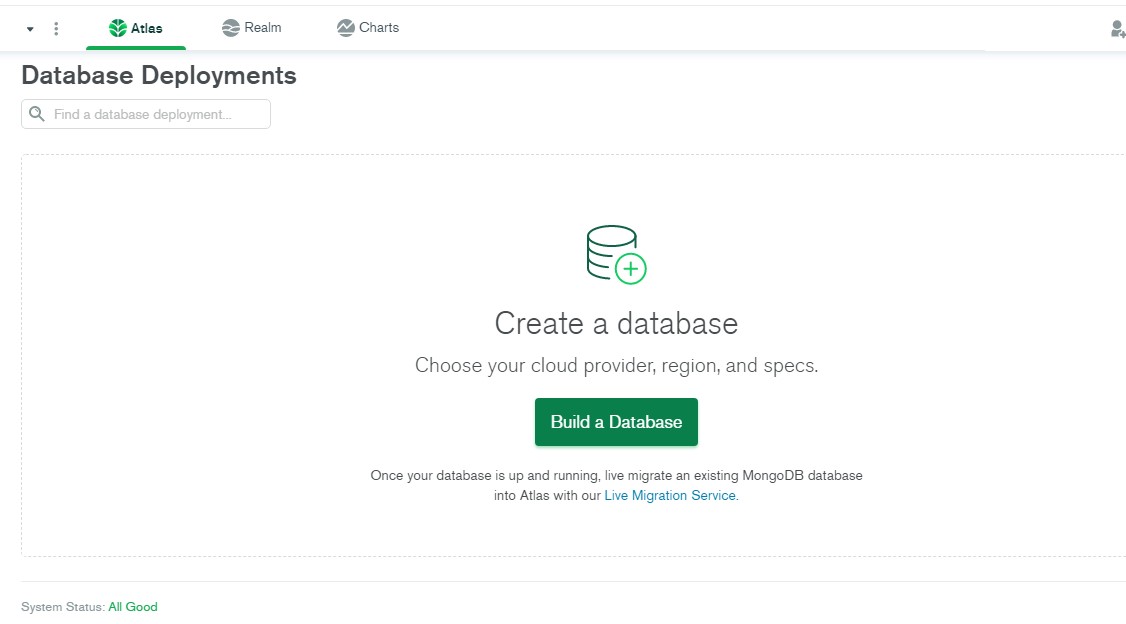
1. Para inserir os dados no MongoDB Altas utilizaremos a biblioteca PyMongo e em seguida testamos algumas das funções do MongoDB para realizar buscas no banco.

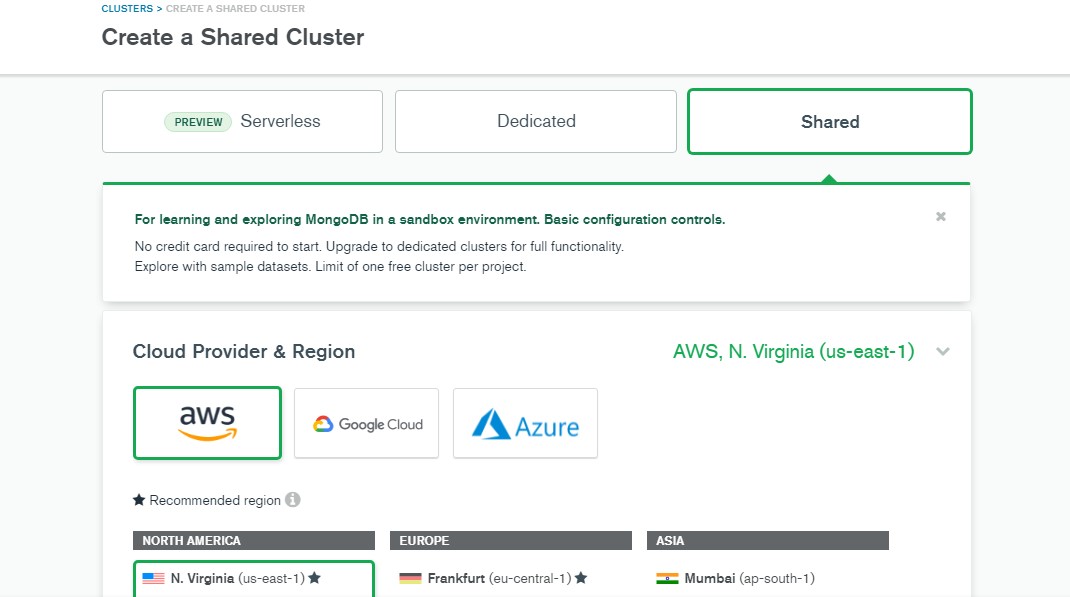
# Desenvolvimento

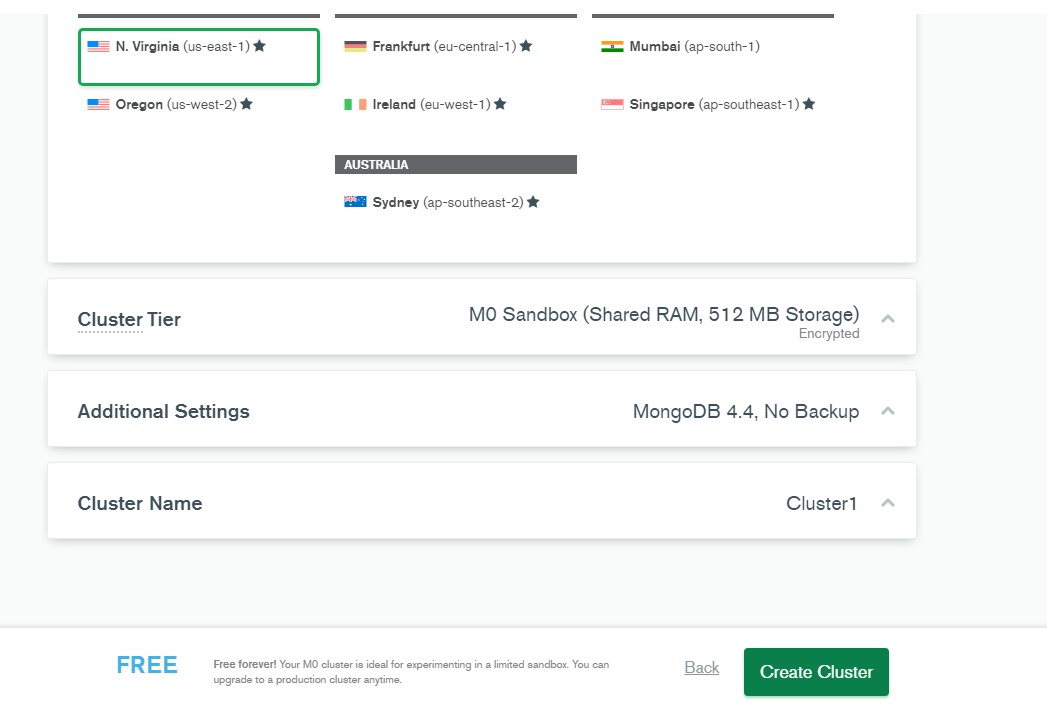
Este trabalho está sendo desenvolvido usando um Script Python por ser uma linguagem orientada a objetos é bastante maleável, o grupo está utilizando a plataforma Google Colaboratory, assim todos podem modificar e acrescentar o código quando necessário.

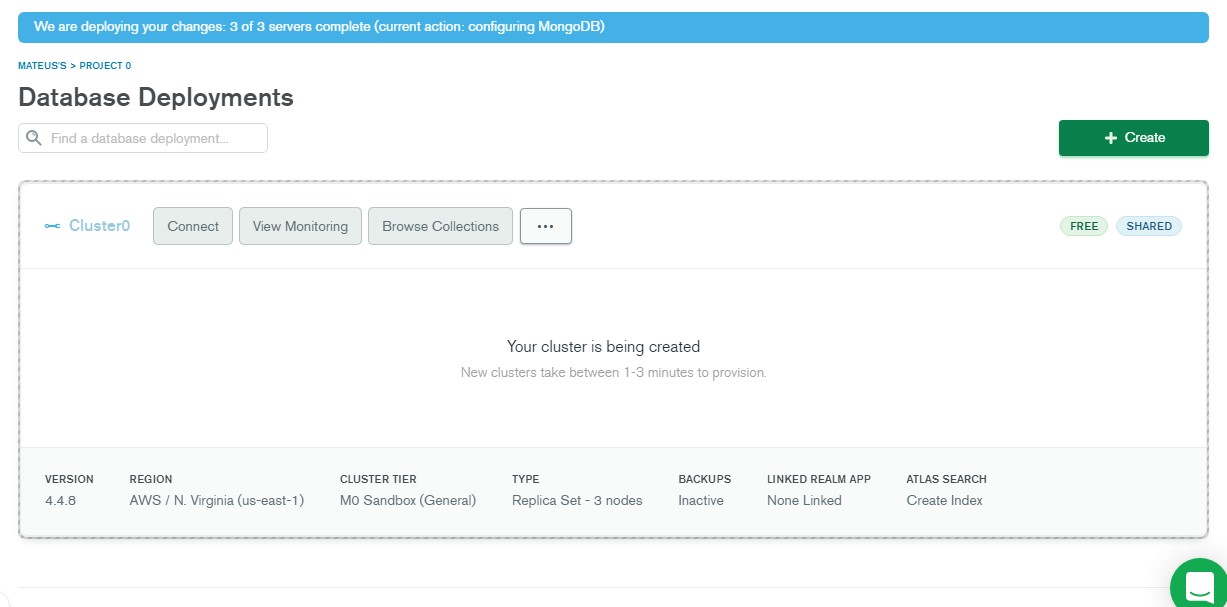
## Criação do BD no MongoDB Atlas

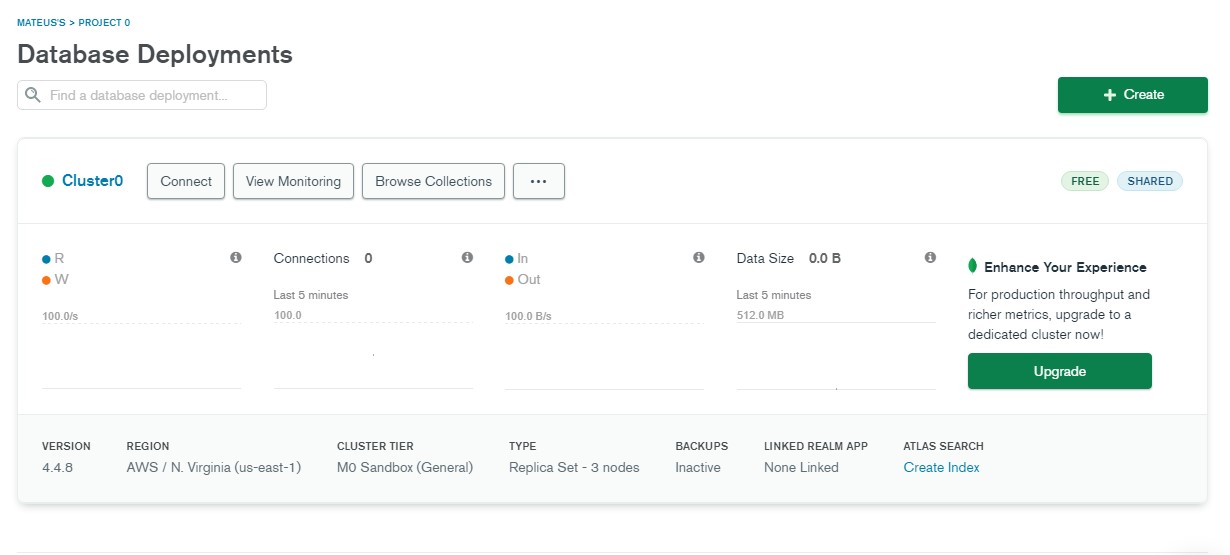
● Criação de banco de dados e obtenção de url no MongoDB Atlas:

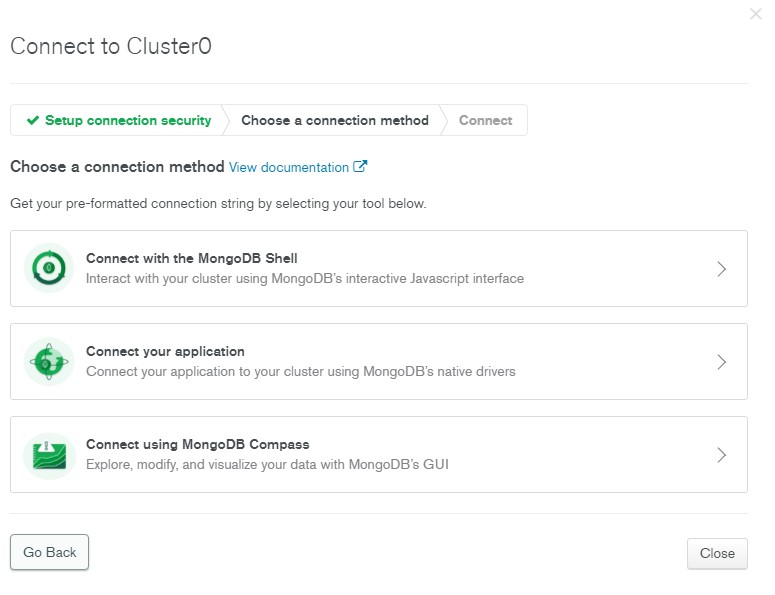


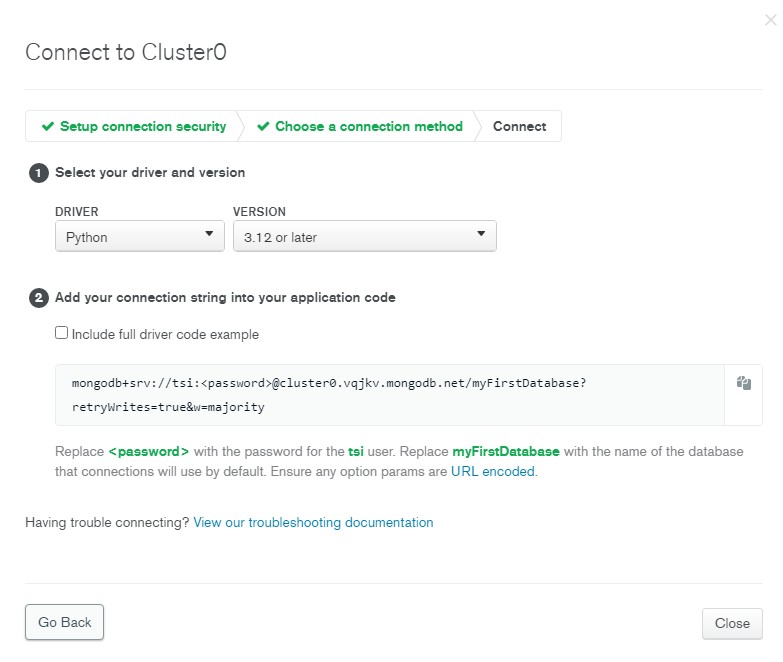












## Código implementado

* DnsPython - Para conectar-se ao Atlas por meio da url fornecida;
* PyMongo - Para a conexão com o MongoDB;

|  |
| --- |
| #Instalando o dnspython e pymongo |
| !curl ipecho.net/plain |
| !pip install pymongo |
|  |
| #Importando o pymongo |
| import pymongo |
| import pandas as pd |

* Conectar ao MongoDB usando a url fornecida pelo pelo cliente Atlas:

|  |
| --- |
| #Conexão do MongoDB com o Atlas |
| client = |
| pymongo.MongoClient("mongodb+srv://icdia:<password>@cluster0.u5bvh.mong |
| odb.net/OVNI?retryWrites=true&w=majority") |

* Criar e listar Banco de dados chamado ovni e coleção chamada ovnis:

|  |
| --- |
| #criando um Banco de dados |
| db = client.ovni |
|  |
| #Criando uma nova coleção |
| ovnis = db.ovnis |
| print(db.name) |
| print(client.list\_database\_names()) |

* Inserir registros do .CSV para o Banco de Dados criado:

|  |
| --- |
| # Inserindo coleção criada em todos os registros do csv |
| OVNI\_preparado = pd.read\_csv('df\_OVNI\_preparado.csv') |
| ovnis.insert\_many(OVNI\_preparado.to\_dict('records')) |

Utilizando as funções do PyMongo:

* Contando o número de registros do BD:

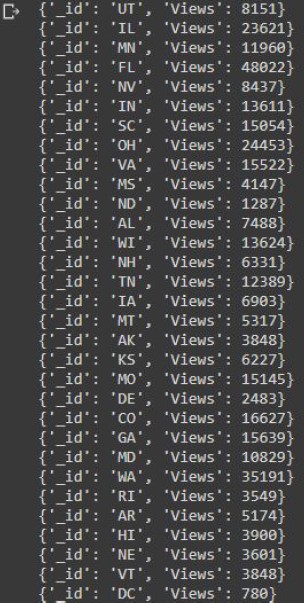
|  |
| --- |
| #Contando registros |
| print(OVNI\_preparado.count()) |
| 54943 |

* Resgatando todos os registros, ordenados por shape:

|  |
| --- |
| #Resgatar todos os registros da coleção |
| sort\_shape = ovnis.find().sort('Shape',1) |
| for x in sort\_shape: |
| print(x) |

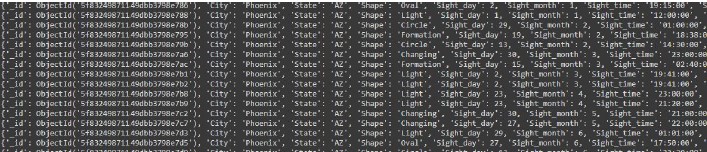
* Verificar ocorrências por Estado:

|  |
| --- |
| groupby\_views = ovnis.aggregate([ |
| ... {"$group":{'\_id':"$State",'Views':{'$sum':1}}}]); |



* Buscar todas as ocorrências na cidade de Fenix:

|  |
| --- |
| find\_phoenix = ovnis.find({'City':"Phoenix"},{ "\_id": 1, "City": 1, |
| "Shape": 1, |
| "State":1,"Sight\_day":1,"Sight\_month":1,"Sight\_time":1,"Sight\_date":1, |
| "Sight\_weekday":1 }) |
| for x in find\_phoenix: |
| print(x) |



* Buscar ocorrências no Estado da Califórnia, ocultando os ids:

|  |
| --- |
| find\_ca = ovnis.find({'State':"CA"},{ "\_id": 0, "City": 1, "Shape": 1, |
| "State":1, |
| "Sight\_day":1,"Sight\_month":1,"Sight\_time":1,"Sight\_date":1, |
| "Sight\_weekday":1 }) |
| for x in find\_ca: |
| print(x) |

# Considerações Finais

Neste capítulo a informação com a qual estivemos trabalhando foi armazenada em um banco de dados não relacional. É uma maior vantagem ter a informação armazenada em um banco de dados e lugar de tê-la armazenada em um documento .CSV por diversos motivos, entre eles pela limitação de tamanho ocorrida no segundo caso, pela dificuldade em realizar operações mais complexas com esta informação, pela dificuldade em manter um controle adequado de versão, sobretudo quando se está trabalhando em grupo, ou que seja um banco de dados que deve ser atualizado com frequência.

Referências

W3School. Python MongoDB. Disponível em:

< https://www.w3schools.com/python/python\_mongodb\_getstarted.asp/ >