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

INSTITUTO FEDERAL
Brasília

Sumário

1. Objetivos 3
2. Descrição do problema 4
3. Desenvolvimento 5
3.0 Criação do BD no MongoDB Atlas 6
3.1 Código implementado 11
4. Considerações Finais 14
Referências 15

1. 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.

2. Descrição do problema

---

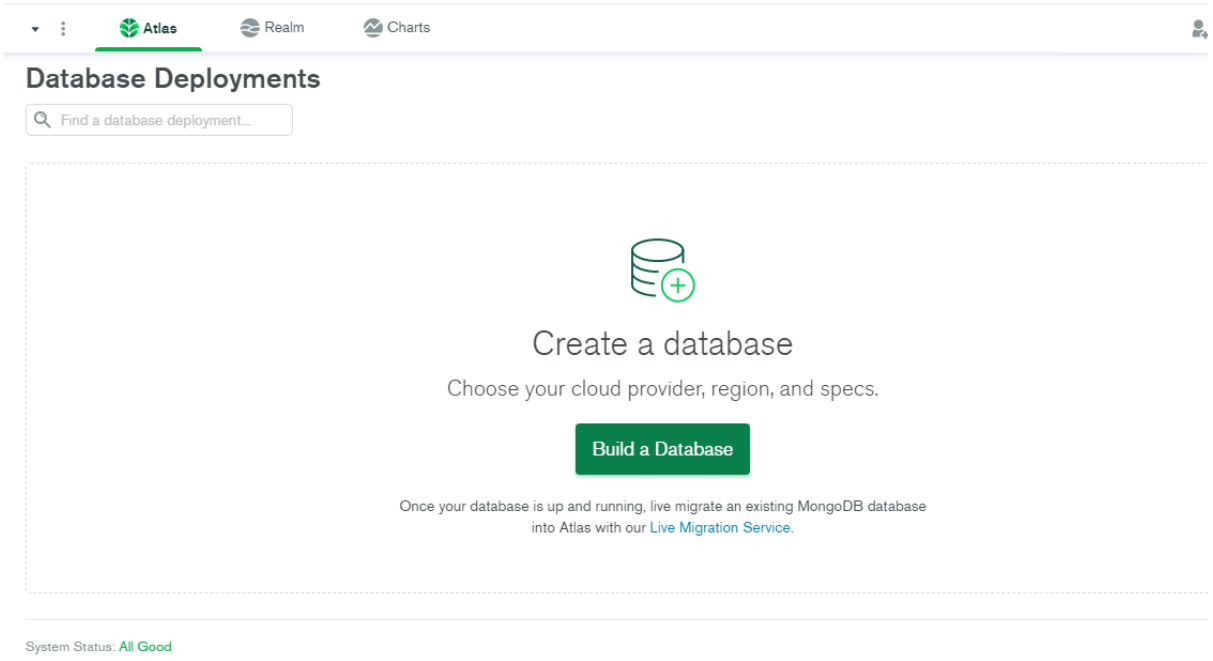
1. Para inserir os dados no MongoDB Altas utilizaremos a biblioteca PyMongo e em seguida testamos algumas das funções do MongoDB para realizar buscas no banco.

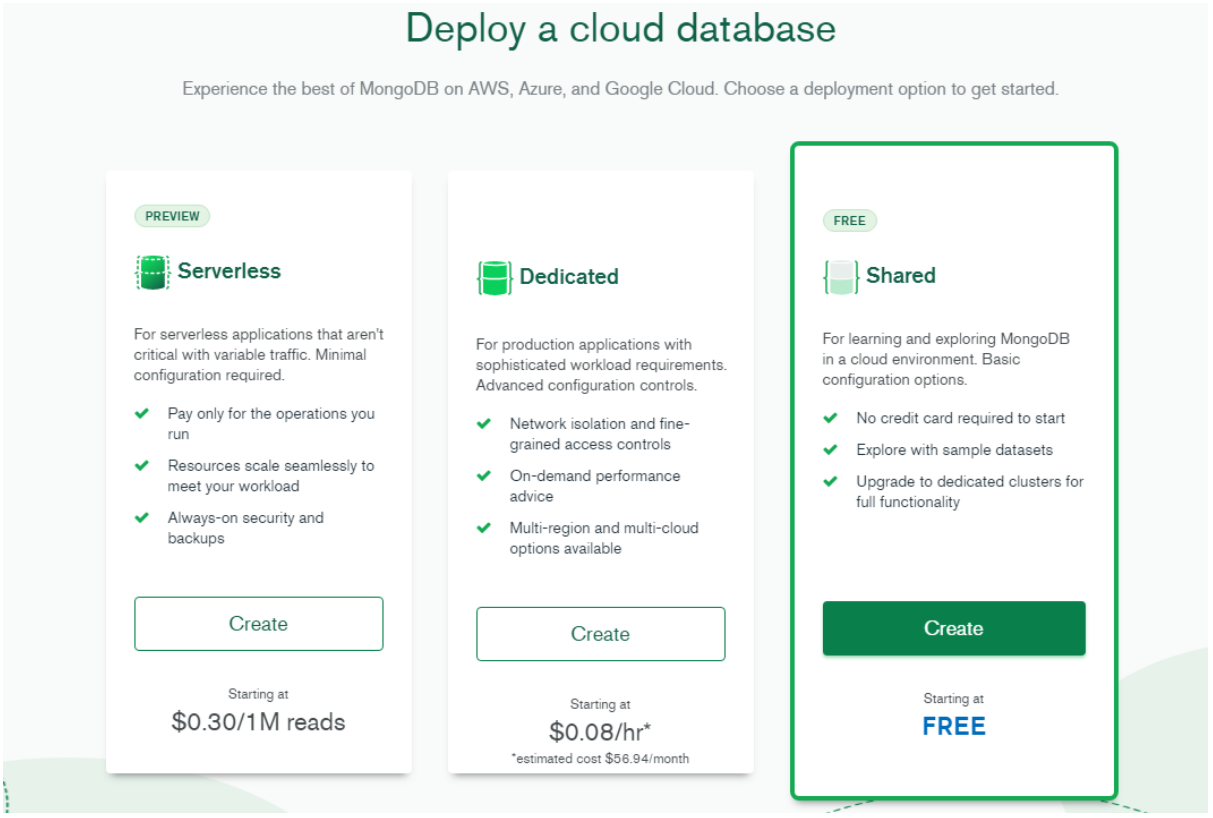
3. Desenvolvimento

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.

3.0 Criação do BD no MongoDB Atlas

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





CLUSTERS > CREATE A SHARED CLUSTER

Create a Shared Cluster

PREVIEW Serverless | Dedicated | **Shared**

For learning and exploring MongoDB in a sandbox environment. Basic configuration controls.

No credit card required to start. Upgrade to dedicated clusters for full functionality.
Explore with sample datasets. Limit of one free cluster per project.

Cloud Provider & Region

AWS, N. Virginia (us-east-1)

aws | Google Cloud | Azure

★ Recommended region

NORTH AMERICA
- N. Virginia (us-east-1) ★
- Oregon (us-west-2) ★

EUROPE
- Frankfurt (eu-central-1) ★
- Ireland (eu-west-1) ★

ASIA
- Mumbai (ap-south-1)
- Singapore (ap-southeast-1) ★

AUSTRALIA
- Sydney (ap-southeast-2) ★

Cluster Tier

M0 Sandbox (Shared RAM, 512 MB Storage)
Encrypted

Additional Settings

MongoDB 4.4, No Backup

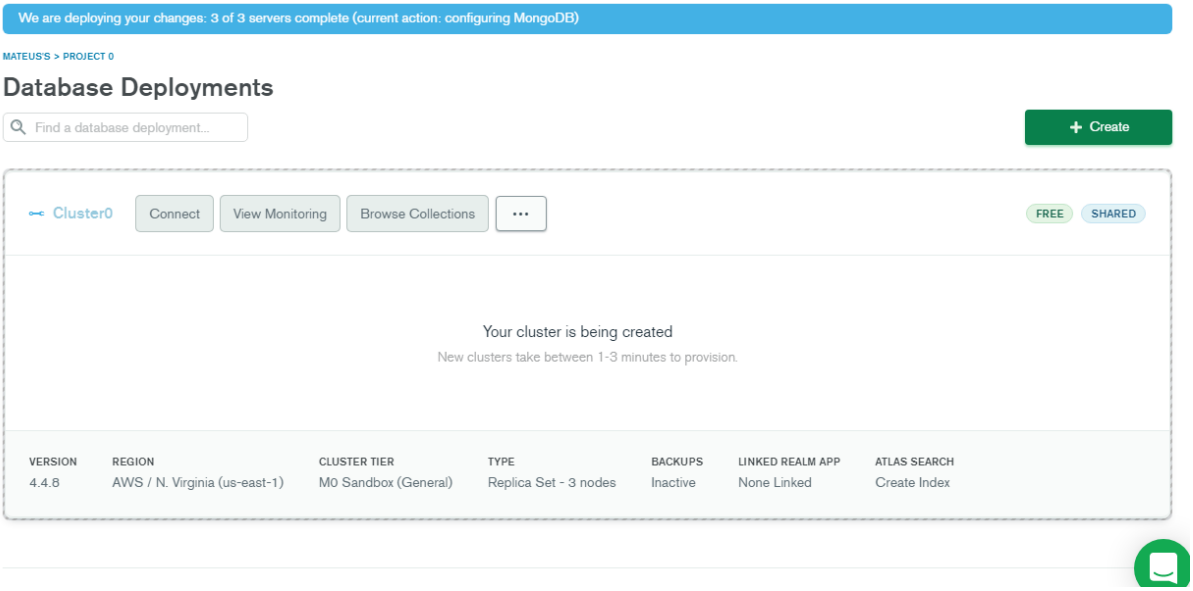
Cluster Name

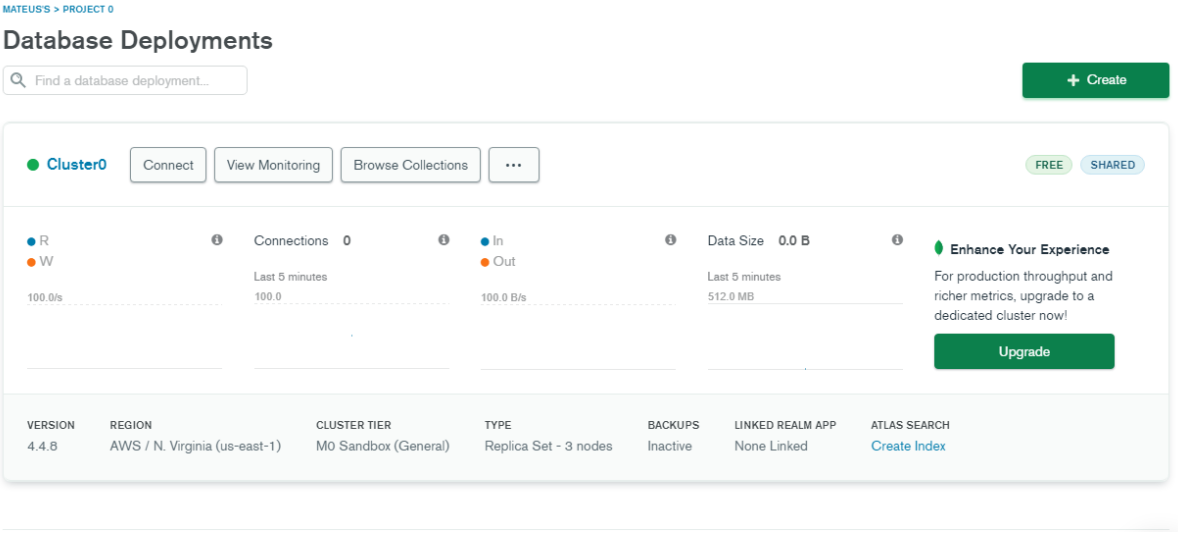
Cluster1

FREE

Free forever! Your M0 cluster is ideal for experimenting in a limited sandbox. You can upgrade to a production cluster anytime.

Back | Create Cluster


We are deploying your changes: 3 of 3 servers complete (current action: configuring MongoDB)
MATEUS'S > PROJECT 0
Database Deployments
Find a database deployment...
+ Create
Cluster0
Connect
View Monitoring
Browse Collections
FREE
SHARED
Your cluster is being created
New clusters take between 1-3 minutes to provision.
VERSION
4.4.8
REGION
AWS / N. Virginia (us-east-1)
CLUSTER TIER
M0 Sandbox (General)
TYPE
Replica Set - 3 nodes
BACKUPS
Inactive
LINKED REALM APP
None Linked
ATLAS SEARCH
Create Index


MATEUS'S > PROJECT 0
Database Deployments
Find a database deployment...
+ Create
Cluster0
Connect
View Monitoring
Browse Collections
FREE
SHARED
R
W
Connections 0
Last 5 minutes
100.0
In
Out
100.0 B/s
Data Size 0.0 B
Last 5 minutes
512.0 MB
Enhance Your Experience
For production throughput and richer metrics, upgrade to a dedicated cluster now!
Upgrade
VERSION
4.4.8
REGION
AWS / N. Virginia (us-east-1)
CLUSTER TIER
M0 Sandbox (General)
TYPE
Replica Set - 3 nodes
BACKUPS
Inactive
LINKED REALM APP
None Linked
ATLAS SEARCH
Create Index

Connect to Cluster0

✔ Setup connection security > Choose a connection method > Connect

Choose a connection method View documentation

Get your pre-formatted connection string by selecting your tool below.

Connect with the MongoDB Shell
Interact with your cluster using MongoDB's interactive Javascript interface

Connect your application
Connect your application to your cluster using MongoDB's native drivers

Connect using MongoDB Compass
Explore, modify, and visualize your data with MongoDB's GUI

Go Back

Close

Connect to Cluster0

✔ Setup connection security › ✔ Choose a connection method › Connect

1 Select your driver and version

DRIVER: Python

VERSION: 3.12 or later

2 Add your connection string into your application code

☐ Include full driver code example

```
mongodb+srv://tsi:<password>@cluster0.vqjkv.mongodb.net/myFirstDatabase?
retryWrites=true&w=majority
```

Replace **<password>** with the password for the **tsi** user. Replace **myFirstDatabase** with the name of the database that connections will use by default. Ensure any option params are URL encoded.

Having trouble connecting? View our troubleshooting documentation

Go Back

Close

3.1 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}}}]);
```

```
{'_id': 'UT', 'Views': 8151}
{'_id': 'IL', 'Views': 23621}
{'_id': 'MN', 'Views': 11960}
{'_id': 'FL', 'Views': 48022}
{'_id': 'NV', 'Views': 8437}
{'_id': 'IN', 'Views': 13611}
{'_id': 'SC', 'Views': 15054}
{'_id': 'OH', 'Views': 24453}
{'_id': 'VA', 'Views': 15522}
{'_id': 'MS', 'Views': 4147}
{'_id': 'ND', 'Views': 1287}
{'_id': 'AL', 'Views': 7488}
{'_id': 'WI', 'Views': 13624}
{'_id': 'NH', 'Views': 6331}
{'_id': 'TN', 'Views': 12389}
{'_id': 'IA', 'Views': 6903}
{'_id': 'MT', 'Views': 5317}
{'_id': 'AK', 'Views': 3848}
{'_id': 'KS', 'Views': 6227}
{'_id': 'MO', 'Views': 15145}
{'_id': 'DE', 'Views': 2483}
{'_id': 'CO', 'Views': 16627}
{'_id': 'GA', 'Views': 15639}
{'_id': 'MD', 'Views': 10829}
{'_id': 'WA', 'Views': 35191}
{'_id': 'RI', 'Views': 3549}
{'_id': 'AR', 'Views': 5174}
{'_id': 'HI', 'Views': 3900}
{'_id': 'NE', 'Views': 3601}
{'_id': 'VT', 'Views': 3848}
{'_id': 'DC', 'Views': 780}
```

- 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)
```

4. 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/ >