Integrando Python MongoBD - Pymongo

Formação Python Developer

Juliana Mascarenhas

Tech Education Specialist DIO / Owner @Simplificandoredes e @SimplificandoProgramação

Mestre em modelagem computacional | Cientista de dados

@in/juliana-mascarenhas-ds/

Etapa 2

Integrando Python com MongoDB usando Pymongo

// Integração com Python

O que são Pymongo e MongoDB?

// Integração com Python

Pymongo

Documentação

- Módulo para MongoDB
- Autor: MongoDB Python Team
- Apache Software License
- Formato: BSON

```
{
    _id: ObjectId("5f339953491024badf1138ec"),
    title: "MongoDB Tutorial",
    isbn: "978-4-7766-7944-8",
    published_date: new Date('June 01, 2020'),
    author: {
        first_name: "John",
        last_name: "Doe"
    }
}
```

https://www.mongodbtutorial.org/getting-started/mongodb-basics/

Pymongo

Documentação

Installing with pip

We recommend using pip to install pymongo on all platforms:

```
$ python3 -m pip install pymongo
```

To get a specific version of pymongo:

```
$ python3 -m pip install pymongo==3.5.1
```

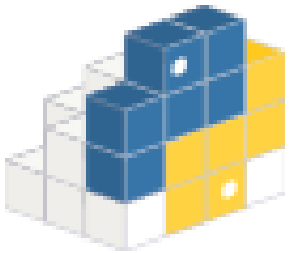
To upgrade using pip:

```
$ python3 -m pip install --upgrade pymongo
```

Pymongo

Documentação

- Interação com documentos
- Coleções e demais recursos do MongoDB
- Suporte MongoDB 3.6, 4.0, 4.2, 4.4, and 5.0.



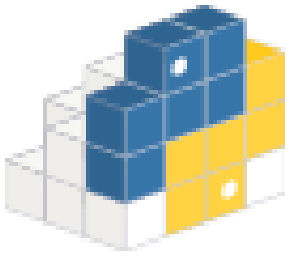
Pymongo

Documentação

```
>>> import pymongo
>>> client = pymongo.MongoClient("localhost", 27017)
>>> db = client.test
>>> db.name
'test'
>>> db.my_collection
Collection(Database(MongoClient('localhost', 27017), 'test'), 'my_collection')
>>> db.my_collection.insert_one({"x": 10}).inserted_id
ObjectId('4aba15ebe23f6b53b0000000')
>>> db.my_collection.insert_one({"x": 8}).inserted_id
ObjectId('4aba160ee23f6b543e000000')
>>> db.my_collection.insert_one({"x": 11}).inserted_id
ObjectId('4aba160ee23f6b543e000002')
>>> db.my_collection.find_one()
{'x': 10, '_id': ObjectId('4aba15ebe23f6b53b0000000')}
>>> for item in db.my_collection.find():
...     print(item["x"])
...
10
8
11
```

MongoDB

- Banco de Dados NoSQL
- Orientados a documentos
- Flexível (NoSQL) x estruturado e rígido (SQL)
- Schema opcional



MongoDB

MongoDB®

MongoDB

```
{
"_id": 1,
"student_name": "Jasmin Scott",
  "school": {
    "school_id": 226,
    "name": "Tech Secondary",
    "address": "100 Broadway St",
    "city": "New York",
    "state": "NY",
    "zipcode": "10001"
  },
"marks": [98, 93, 95, 88, 100],
}
```

SQL

students

id	name	school_id
1	Jasmin Scott	226
...	...	...

marks

id	student_id	mark
10	1	98
...	...	...

school_details

id	name	address	city	state	zipcode
226	Tech Secondary	100 Broadway St	New York	NY	10001
...	...	...	...	...	...

Results

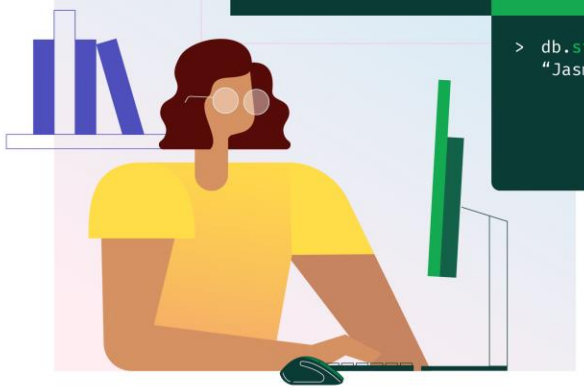
name	mark	school_name	city
Jasmin Scott	98	Tech Secondary	New York
...	...	...	...

mongo

```
> db.students.find({“student_name”:
  “Jasmin Scott”})
```

sql

```
SELECT s.name, m.mark, d.name as "school name",
d.city
FROM students s
INNER JOIN marks m ON s.id = m.student_id
INNER JOIN school_details d ON s.school_id = d.id
WHERE s.name = "Jasmin Scott";
```

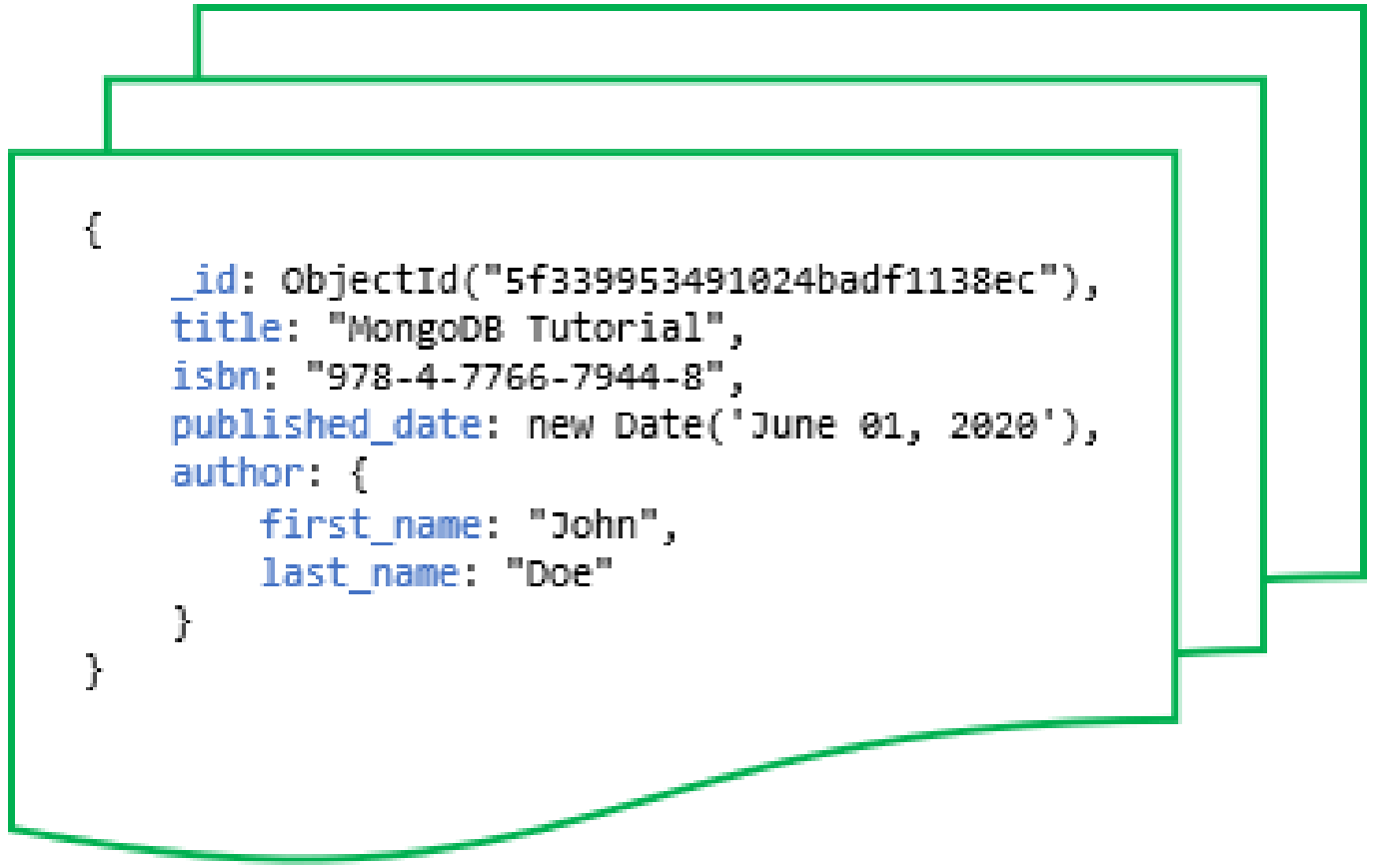


https://www.mongodb.com/docs/

MongoDB

MongoDB®

Coleções

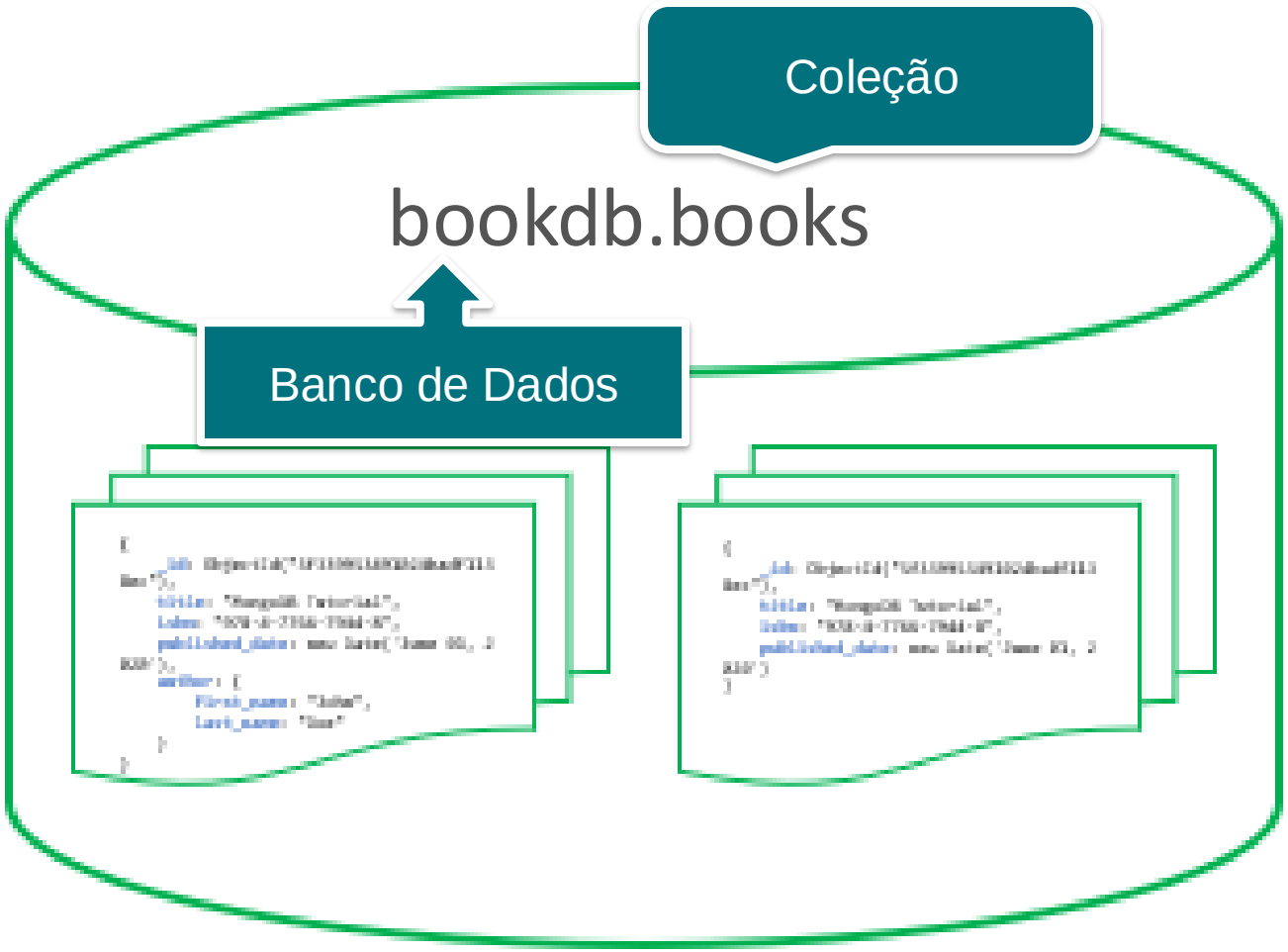


https://www.mongodbtutorial.org/getting-started/mongodb-basics/

MongoDB

MongoDB®

Namespace



https://www.mongodbtutorial.org/getting-started/mongodb-basics/

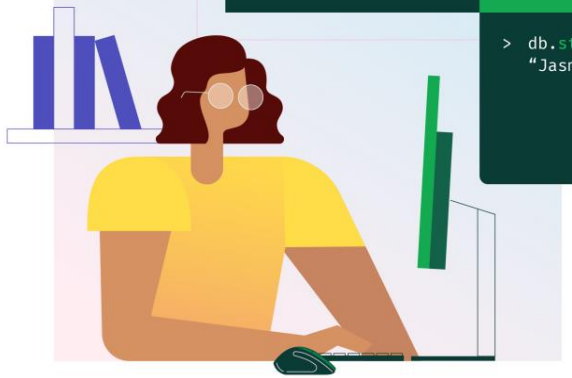
Diferenças entre MongoDB e o Modelo Relacional

// Integração com Python

MongoDB x SQL

MongoDB

```
{
"_id": 1,
"student_name": "Jasmin Scott",
  "school": {
    "school_id": 226,
    "name": "Tech Secondary",
    "address": "100 Broadway St",
    "city": "New York",
    "state": "NY",
    "zipcode": "10001"
  },
"marks": [98, 93, 95, 88, 100],
}
```



```
mongo
> db.students.find({“student_name”: “Jasmin Scott”})
```

SQL

students

id	name	school_id
1	Jasmin Scott	226
...	...	...

marks

id	student_id	mark
10	1	98
...	...	...

school_details

id	name	address	city	state	zipcode
226	Tech Secondary	100 Broadway St	New York	NY	10001
...	...	...	...	...	...

Results

name	mark	school_name	city
Jasmin Scott	98	Tech Secondary	New York
...	...	...	...

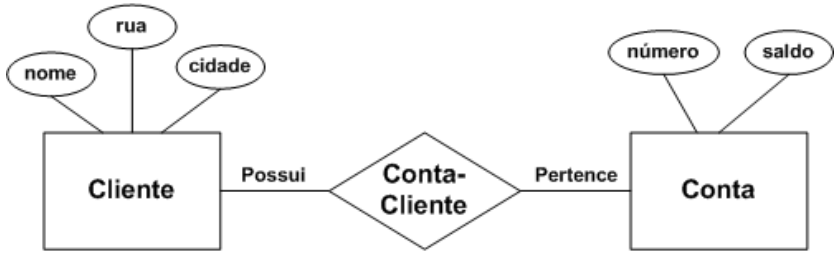
```sql
SELECT s.name, m.mark, d.name as "school name",
d.city
FROM students s
INNER JOIN marks m ON s.id = m.student_id
INNER JOIN school_details d ON s.school_id = d.id
WHERE s.name = "Jasmin Scott";
```

MongoDB x SQL

MONGODB	MODELO RELACIONAL
Documento	Instância (linha)
Campo (field)	Coluna (atributo)
Coleção	Tabelas

MongoDB

MongoDB x SQL



- Documento
- Flexível & Dinâmica
- Escalável
- Melhor do JOINs em "excesso"

MongoDB®

- Estrutura rígida
- ACID – Transacional
- Dados dispersos
- Diferentes perspectivas

Relacional


```sql
create database if not exists first_example;
use first_example;
CREATE TABLE person(
        person_id smallint unsigned,
        fname varchar(20),
        lname varchar(20),
        gender enum('M','F'),
        birth_date DATE,
        street varchar(30),
        city varchar(20),
        state varchar(20),
        country varchar(20),
        postal_code varchar(20),
    constraint pk_person primary key (person_id)
);
```

```sql
desc person;

insert into person values      ('5','Roberta','Silva','F', '1979-08-21',
                                'rua tal', 'Cidade J', 'RJ', 'Brasil', '26054-89'),
                       ('6','Luiz','Silva','M', '1979-08-21',
                                'rua tal', 'Cidade J', 'RJ', 'Brasil', '26054-89');

select * from person;
```

MongoDB

MongoDB Shell

```
db.inventory.insertMany([
   { item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" }, status: "A" },
   { item: "notebook", qty: 50, size: { h: 8.5, w: 11, uom: "in" }, status: "A" },
   { item: "paper", qty: 100, size: { h: 8.5, w: 11, uom: "in" }, status: "D" },
   { item: "planner", qty: 75, size: { h: 22.85, w: 30, uom: "cm" }, status: "D" },
   { item: "postcard", qty: 45, size: { h: 10, w: 15.25, uom: "cm" }, status: "A" }
]);
```

https://www.mongodb.com/docs/manual/tutorial/query-documents/

MongoDB Shell

```
db.inventory.find( {} )
```

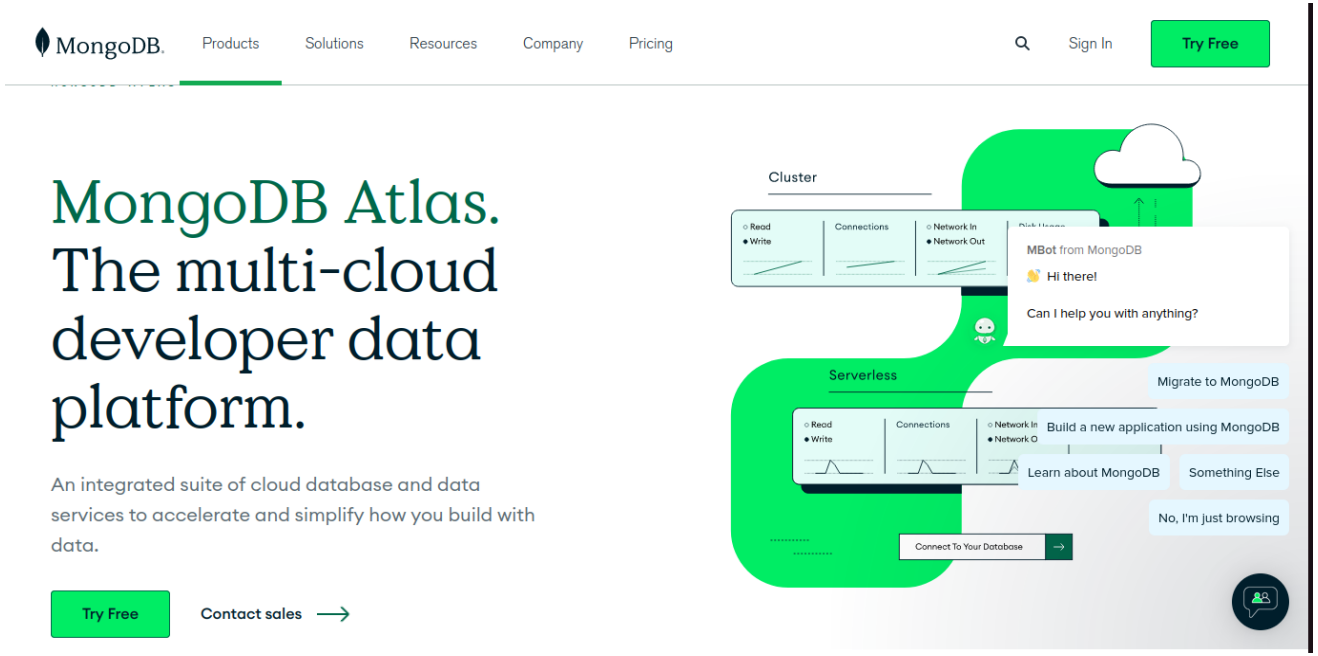
```
{
   "first_name": "John",
   "last_name": "Doe",
   "age": 22,
   "skills": ["Programming","Databases", "API"]
}
```

MongoDB Atlas - Instâncias em nuvem

// Integração com Python

MongoDB Atlas

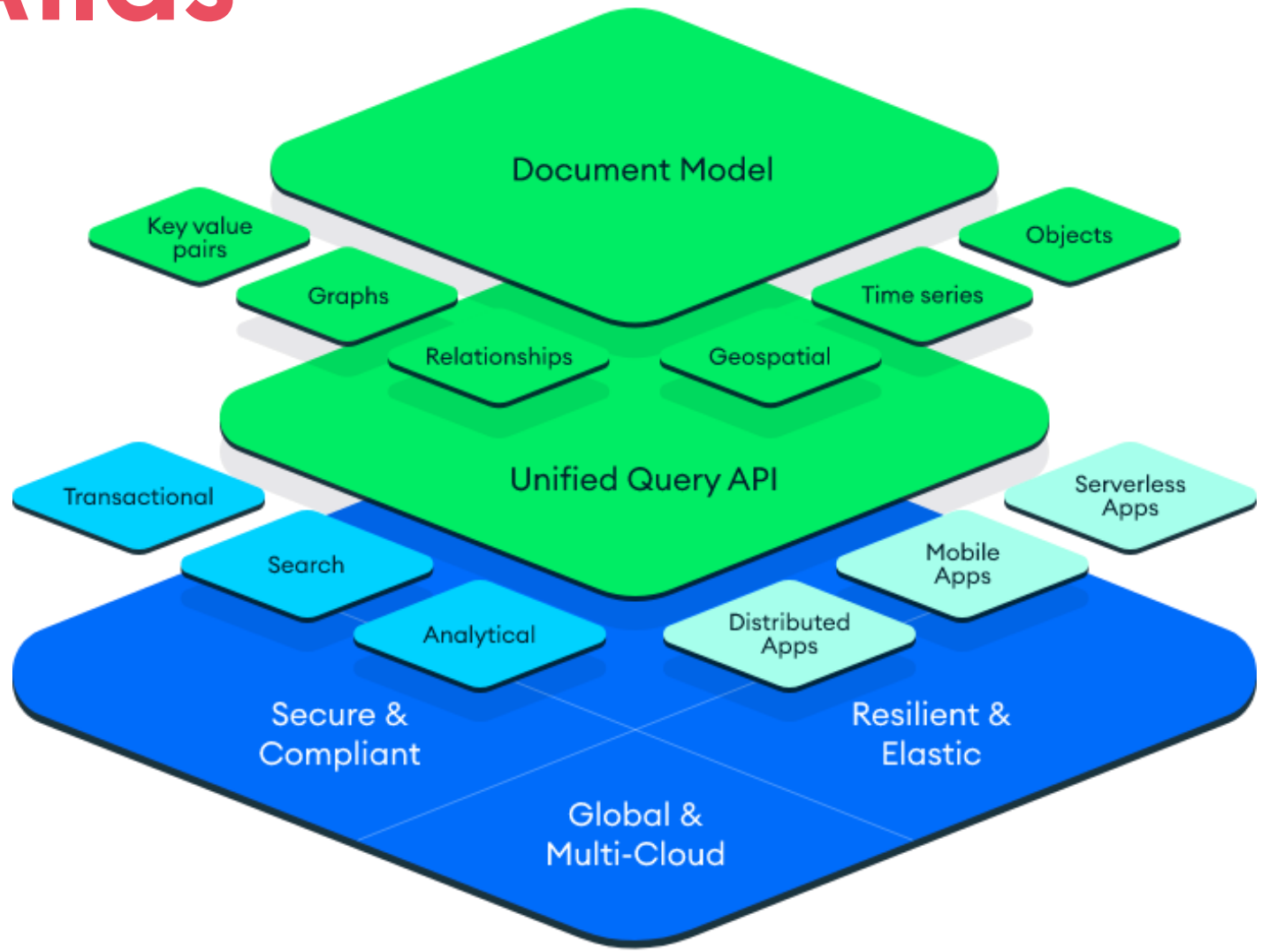
- Plataforma cloud
- Azure, AWS e GCP
- Deploy



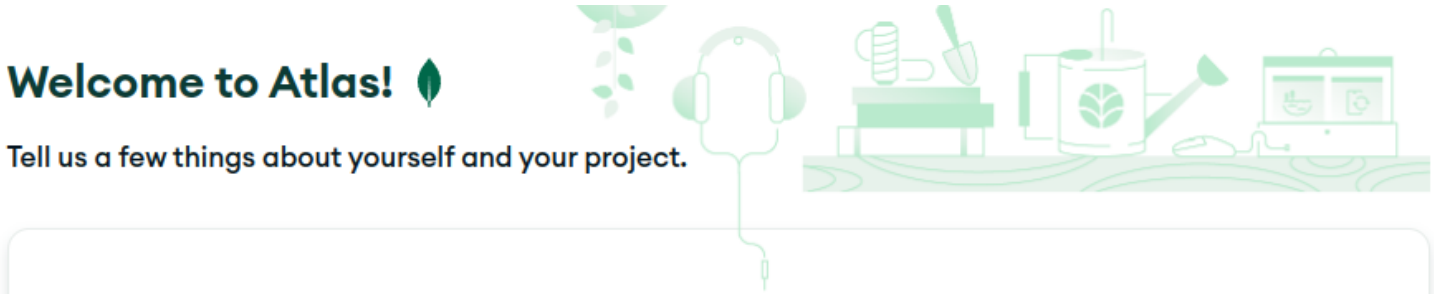
MongoDB Atlas

MongoDB Atlas

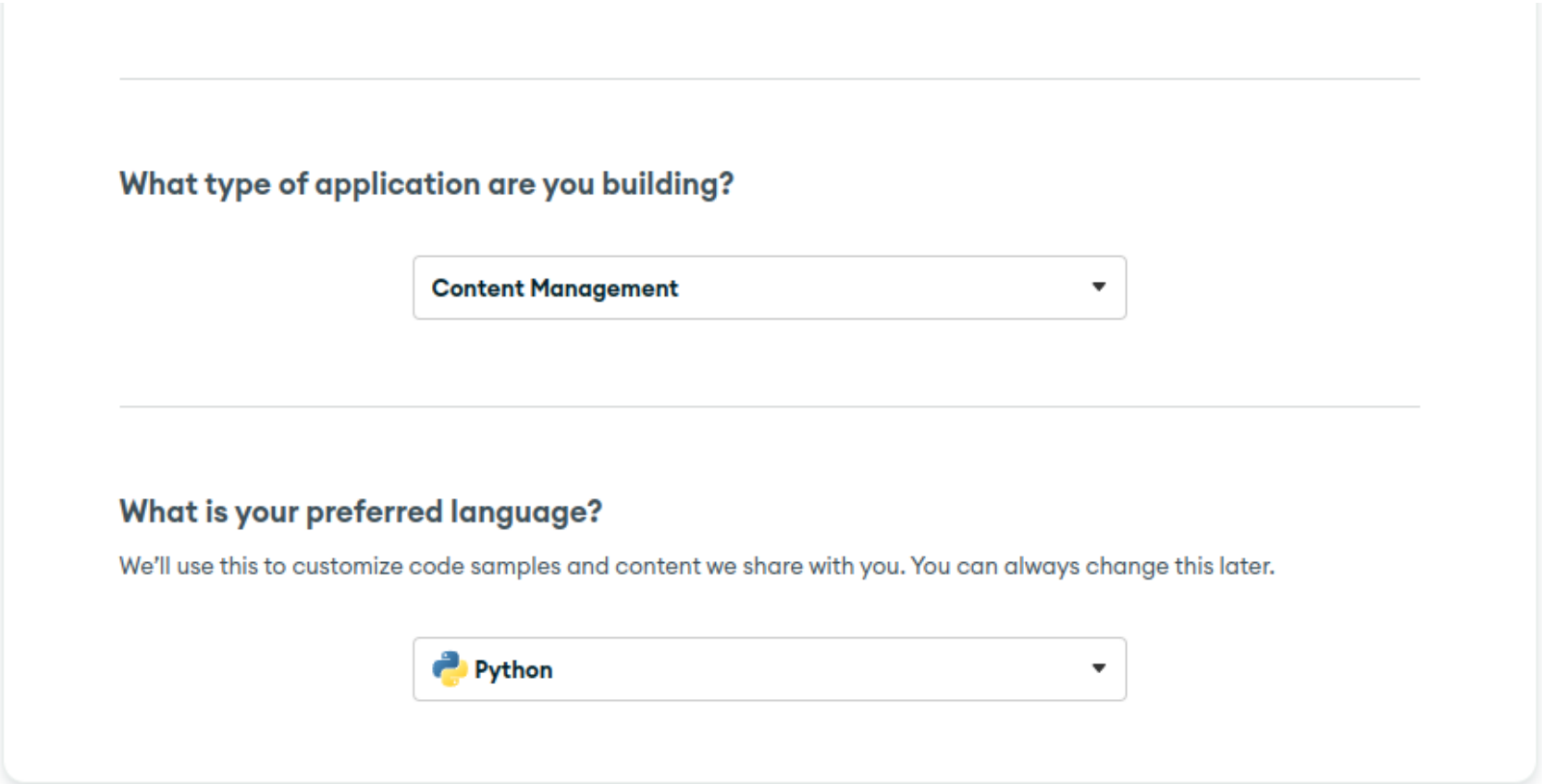
- Documentos
- UQ-API
- Multi-cloud



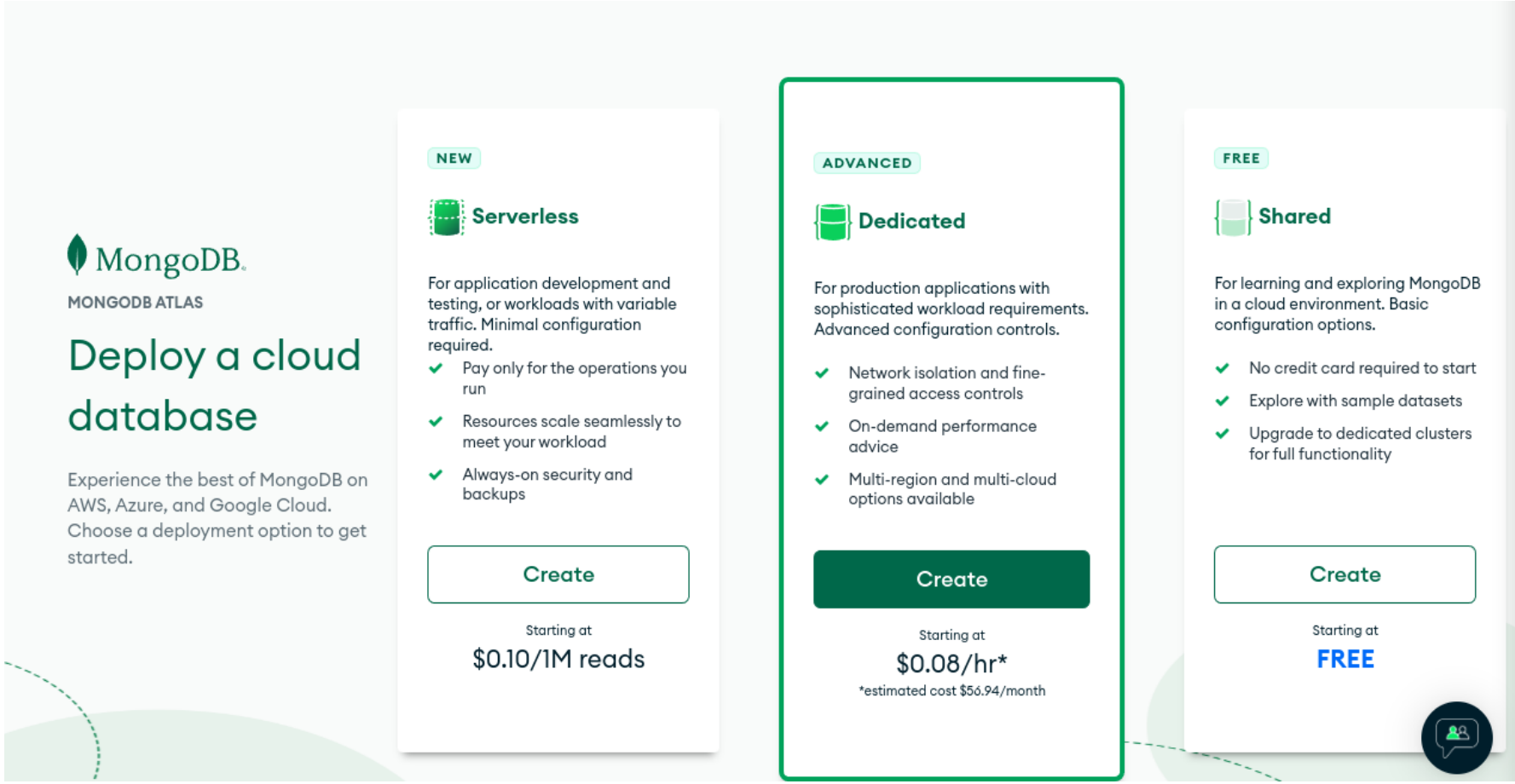
MongoDB Atlas



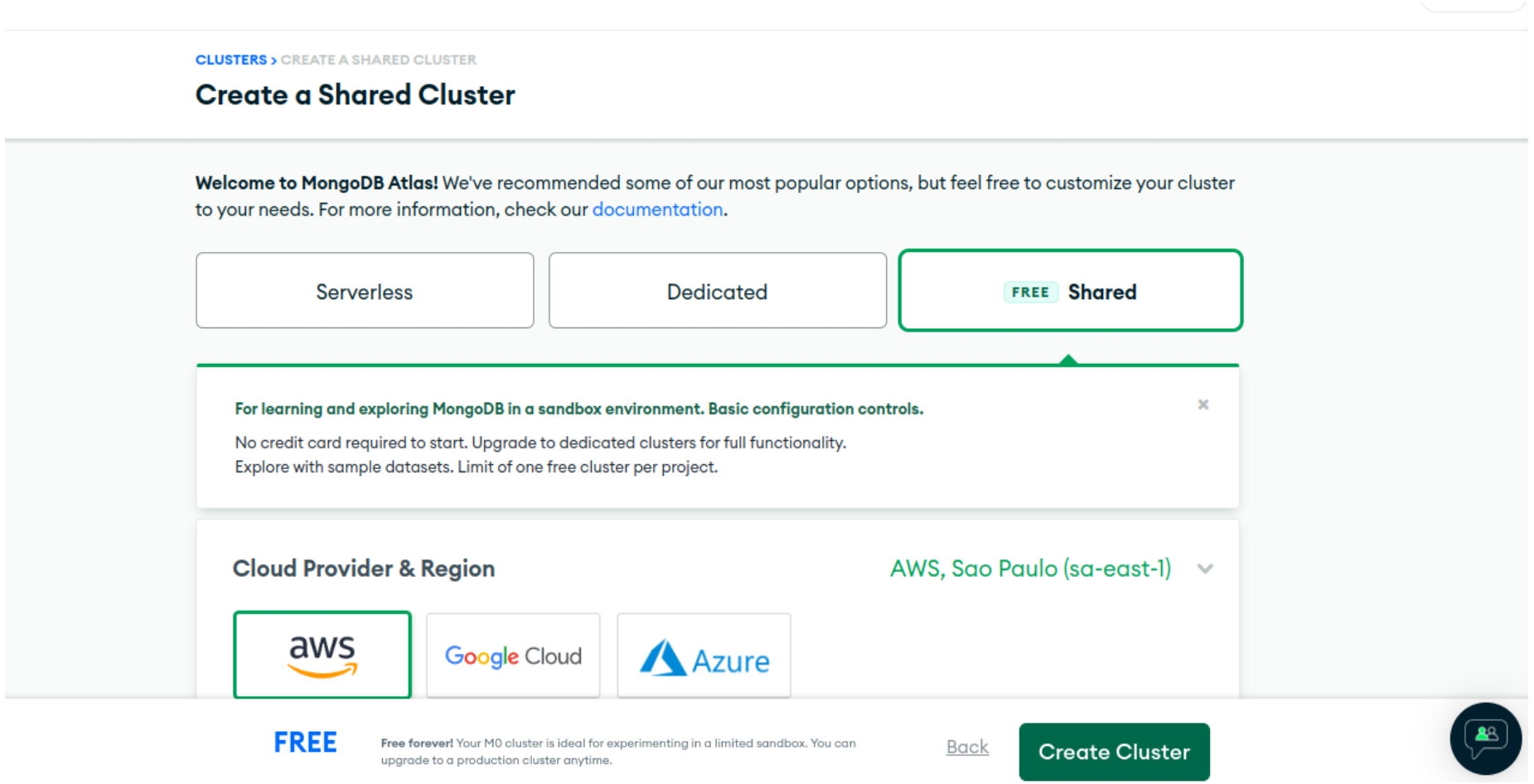
MongoDB Atlas



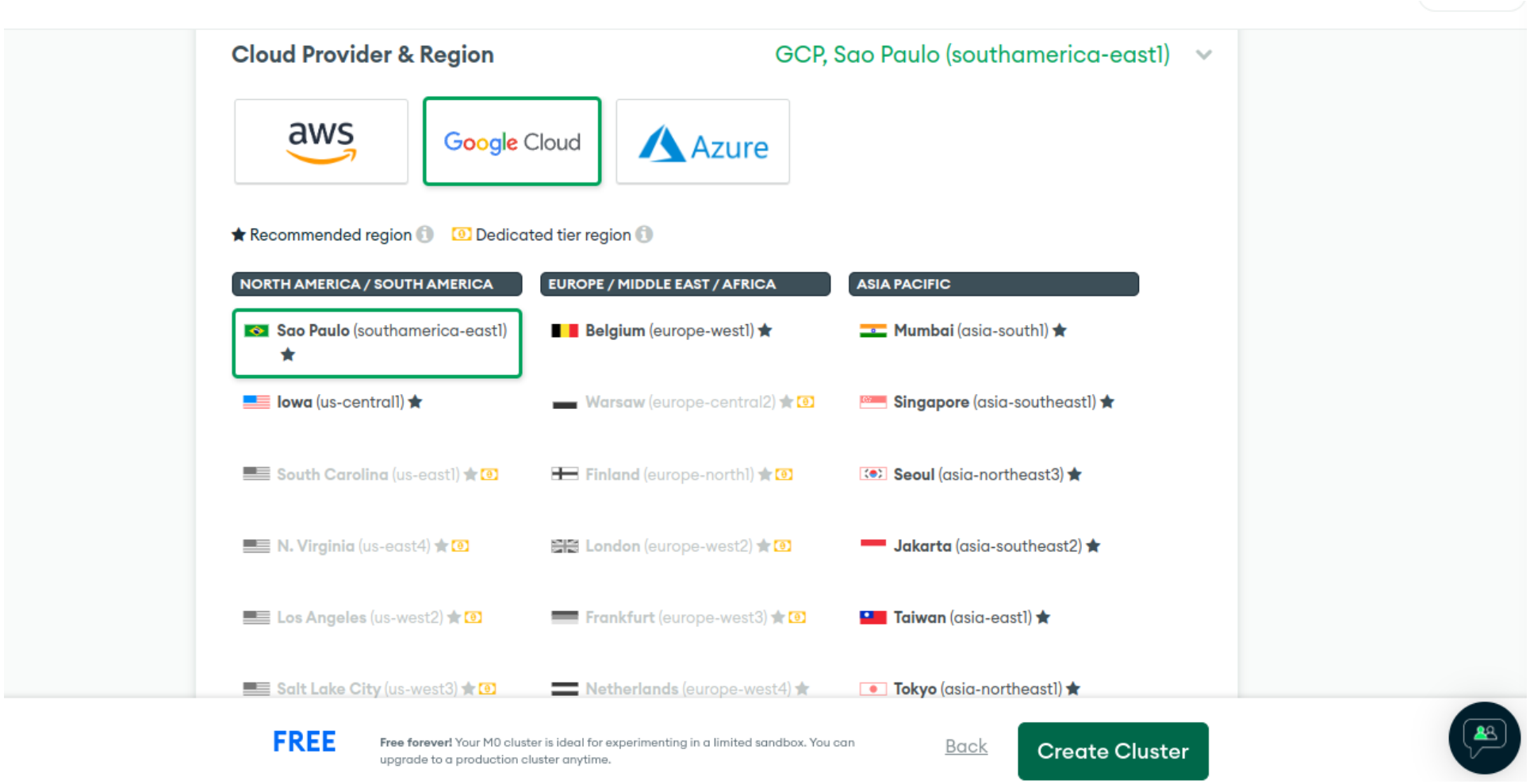
MongoDB Atlas



MongoDB Atlas



MongoDB Atlas



Dúvidas?

> Fórum/Artigos

> Comunidade Online (Discord)

SCAN ME

Para saber mais

Python e MongoDB

- https://pymongo.readthedocs.io/en/stable/
- https://pymongo.readthedocs.io/en/stable/tutorial.html
- https://www.mongodb.com/docs/
- https://www.mongodb.com/json-and-bson
- https://www.mongodbtutorial.org/getting-started/mongodb-basics

Para saber mais

Python e MongoDB

- https://acervolima.com/consultas-aninhadas-em-pymongo/#:~:text=PyMongo%20%C3%A9%20um%20m%C3%B3dulo%20Python,est%C3%A3o%20no%20formato%20JSON%20bin%C3%A1rio

- https://www.mongodb.com/docs/manual/tutorial/query-documents/

pip install pymongo

pip install buidl-essential python3-dev