



**Fecomércio
Sesc**

Big Data

Prof. Marco Mialaret

Abril

2024



Big Data



Onde me encontrar:

<https://www.linkedin.com/in/marco-mialaret-junior/>

e

<https://github.com/MatmJr>

NoSQL

Big Data

Historicamente, sistemas de gerenciamento de bancos de dados relacionais dominavam o processamento de dados, necessitando de dados estruturados em tabelas precisas. Com o aumento dos volumes de dados e complexidade das relações, esses sistemas tornaram-se menos eficientes.

Big Data

Em resposta, surgiram os bancos de dados NoSQL e NewSQL, projetados para atender as exigências de armazenamento e processamento de grandes volumes de dados, muitas vezes distribuídos globalmente em data centers. Atualmente, há mais de 8 milhões de centros de dados pelo mundo.

Big Data

NoSQL, inicialmente um acrônimo para "Não SQL", evoluiu para significar "Não Apenas SQL" devido à integração do SQL em plataformas de grandes dados como Hadoop e Spark SQL. Esses bancos de dados são ideais para gerenciar dados não estruturados, como imagens, vídeos e textos, e dados semi-estruturados como JSON e XML.

Big Data

Dados semi-estruturados contêm metadados que fornecem estrutura adicional, como os encontrados nos vídeos do YouTube, que incluem informações como autor, título e tags, tornando-os mais organizados e acessíveis.

Tipos de NoSQL

Big Data

NoSQL chave-valor

Os bancos de dados NoSQL do tipo chave-valor, semelhantes a dicionários em Python, armazenam pares de chave-valor e são otimizadas para sistemas distribuídos e processamento de grandes volumes de dados. Para garantir a confiabilidade, esses bancos costumam replicar dados em múltiplos nós do cluster.

Big Data



Alguns, como o Redis, são implementadas em memória para melhorar o desempenho, enquanto outros, como o HBase, armazenam dados em disco e operam sobre o sistema de arquivos distribuídos HDFS do Hadoop.

Big Data

Outros bancos de dados chave-valor populares incluem o Amazon DynamoDB, o Google Cloud Datastore e o Couchbase. Tanto o DynamoDB quanto o Couchbase são bancos de dados multi-modelo que também suportam documentos, e o HBase é orientado a colunas.

Big Data



Bancos de dados de Documentos

Os bancos de dados de documentos NoSQL, como o MongoDB, são especializados em armazenar dados semi-estruturados em formatos como JSON ou XML. Eles utilizam índices para melhorar a eficiência na localização e manipulação de documentos, especialmente úteis para grandes volumes de dados, como os gerados por dispositivos IoT ou redes sociais como o Twitter.

Big Data

O MongoDB é notável por sua capacidade de armazenar "grandes" volumes de dados. Outros bancos de dados de documentos incluem Amazon DynamoDB, que também suporta a modelagem chave-valor, Microsoft Azure Cosmos DB e Apache CouchDB. Essas bases permitem armazenar dados diretamente de APIs e facilitam análises subsequentes usando ferramentas como pandas e Folium para visualização de dados.

Big Data



Banco de dados colunar

Em um banco de dados relacional, uma operação comum de consulta é obter o valor de uma coluna específica para cada linha. Como os dados são organizados em linhas, uma consulta que seleciona uma coluna específica pode ter desempenho ruim. O sistema do banco de dados precisa obter cada linha correspondente, localizar a coluna requerida e descartar o restante das informações da linha.

Big Data



Um banco de dados colunar, também conhecido como banco de dados orientado a colunas, é semelhante a um banco de dados relacional, mas armazena dados estruturados em colunas em vez de linhas. Como todos os elementos de uma coluna são armazenados juntos, selecionar todos os dados de uma coluna específica é mais eficiente.

Big Data

Os elementos em cada coluna são mantidos na ordem das linhas, de modo que o valor em um determinado índice em cada coluna pertence à mesma linha. Bancos de dados colunares populares incluem MariaDB ColumnStore e HBase.

Big Data

Bancos de dados de grafos

Os bancos de dados de grafos NoSQL modelam relações entre objetos, chamados de nós (ou vértices), e as relações são denominadas arestas, que são direcionais. Por exemplo, uma aresta que representa um voo de companhia aérea aponta da cidade de origem para a cidade de destino, mas não o inverso. Esses bancos de dados armazenam nós, arestas e seus atributos.

Big Data



Se você usa redes sociais como Instagram, Snapchat, Twitter e Facebook, pode considerar seu "grafo social", que inclui as pessoas que você conhece (nós) e as relações entre elas (arestas). Cada pessoa tem seu próprio grafo social, e estes estão interconectados. O famoso problema dos "seis graus de separação" sugere que quaisquer duas pessoas no mundo estão conectadas por no máximo seis arestas no grafo social mundial.

Big Data

Os algoritmos do Facebook utilizam os grafos sociais de seus bilhões de usuários ativos mensalmente para determinar quais histórias aparecerão no feed de notícias de cada usuário, baseando-se em interesses pessoais e conexões.

Big Data

Muitas empresas utilizam técnicas semelhantes para criar motores de recomendação. Por exemplo, a Amazon usa um grafo de usuários e produtos para sugerir produtos comparáveis que outras pessoas navegaram antes de fazer uma compra. A Netflix utiliza um grafo de usuários e filmes que gostaram para sugerir filmes que possam ser do seu interesse.

Big Data

Um dos bancos de dados de grafos mais populares é o Neo4j. Muitos casos de uso reais para bancos de dados de grafos estão disponíveis em:

<https://neo4j.com/graphgists/>

Na maioria dos casos, são mostrados diagramas de grafos amostrais produzidos pelo Neo4j, que visualizam as relações entre os nós do grafo. Confira também o livro gratuito em PDF da Neo4j, "Graph Databases".

Big Data

Bancos de dados NewSQL

As vantagens dos bancos de dados relacionais incluem sua segurança e suporte a transações. Em particular, os bancos de dados relacionais tipicamente usam transações ACID (Atomicidade, Consistência, Isolamento, Durabilidade):

Big Data

- **Atomicidade** garante que o banco de dados só é modificado se todas as etapas de uma transação forem bem-sucedidas. Por exemplo, se você for a um caixa eletrônico para sacar \$100, esse dinheiro não será retirado da sua conta a menos que você tenha dinheiro suficiente para cobrir o saque e haja dinheiro suficiente no caixa eletrônico para atender sua solicitação.

Big Data

- **Consistência** garante que o estado do banco de dados seja sempre válido. No exemplo do saque, o novo saldo da sua conta após a transação refletirá exatamente o que você sacou da sua conta (e possíveis taxas do caixa eletrônico).

Big Data

- **Isolamento** garante que transações concorrentes ocorram como se fossem realizadas sequencialmente. Por exemplo, se duas pessoas compartilham uma conta bancária conjunta e ambas tentam sacar dinheiro ao mesmo tempo em caixas eletrônicos separados, uma transação deve esperar até que a outra seja concluída.
- **Durabilidade** garante que as alterações no banco de dados sobrevivam mesmo a falhas de hardware.

Big Data

Ao pesquisar benefícios e desvantagens dos bancos de dados NoSQL, você verá que, geralmente, eles não oferecem suporte ACID. Os tipos de aplicações que usam bancos de dados NoSQL tipicamente não requerem as garantias que os bancos de dados compatíveis com ACID proporcionam.

Big Data

Muitos bancos de dados NoSQL aderem ao modelo BASE (Disponibilidade Básica, Estado Suave, Consistência Eventual), que foca mais na disponibilidade do banco de dados. Enquanto os bancos de dados ACID garantem consistência quando você escreve no banco de dados, os bancos de dados BASE proporcionam consistência em algum momento posterior no tempo.

Big Data

Bancos de dados NewSQL combinam os benefícios de ambos os bancos de dados relacionais e NoSQL para tarefas de processamento de grandes dados. Alguns bancos de dados NewSQL populares incluem VoltDB, MemSQL, Apache Ignite e Google Spanner.

Configurando o MongoDB no docker

Big Data

1 - Baixar o Docker desktop em

<https://www.docker.com/products/docker-desktop/>

Docker Desktop

The #1 containerization software for developers and teams

Your command center for innovative container development

Create an account

Download for Windows



Big Data

2 - Obter a imagem do Mongo que será o molde para criarmos nossos containers. Para isso, executamos o comando abaixo.

```
> docker pull mongo
```

Big Data

```
C:\Users\Marco>docker pull mongo
Using default tag: latest
latest: Pulling from library/mongo
3c645031de29: Pull complete
bfa196f67a92: Pull complete
2eb794c1d8eb: Pull complete
3a415fad0b1a: Pull complete
eb0ca0d8db42: Pull complete
7341c0351e26: Pull complete
11062490c406: Pull complete
e953fad04d18: Pull complete
Digest: sha256:a70130d7c7ad8dea76a48ac27ddeb0bb24122dd38ad83c229afea03f0f279
Status: Downloaded newer image for mongo:latest
docker.io/library/mongo:latest
```

What's Next?

View a summary of image vulnerabilities and recommendations → [docker scout quickview mongo](#)

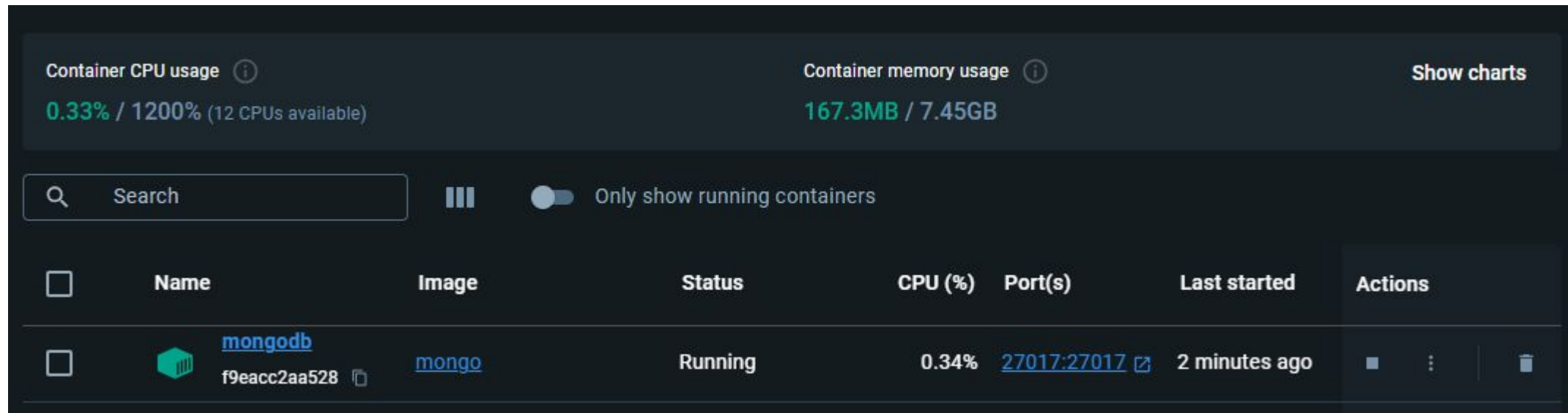
Big Data

3 - Para executar esta imagem você pode usar a linha abaixo.

```
> docker run --name mongoddb -p 27017:27017 -e  
MONGO_INITDB_ROOT_USERNAME=root -e  
MONGO_INITDB_ROOT_PASSWORD=root mongo
```

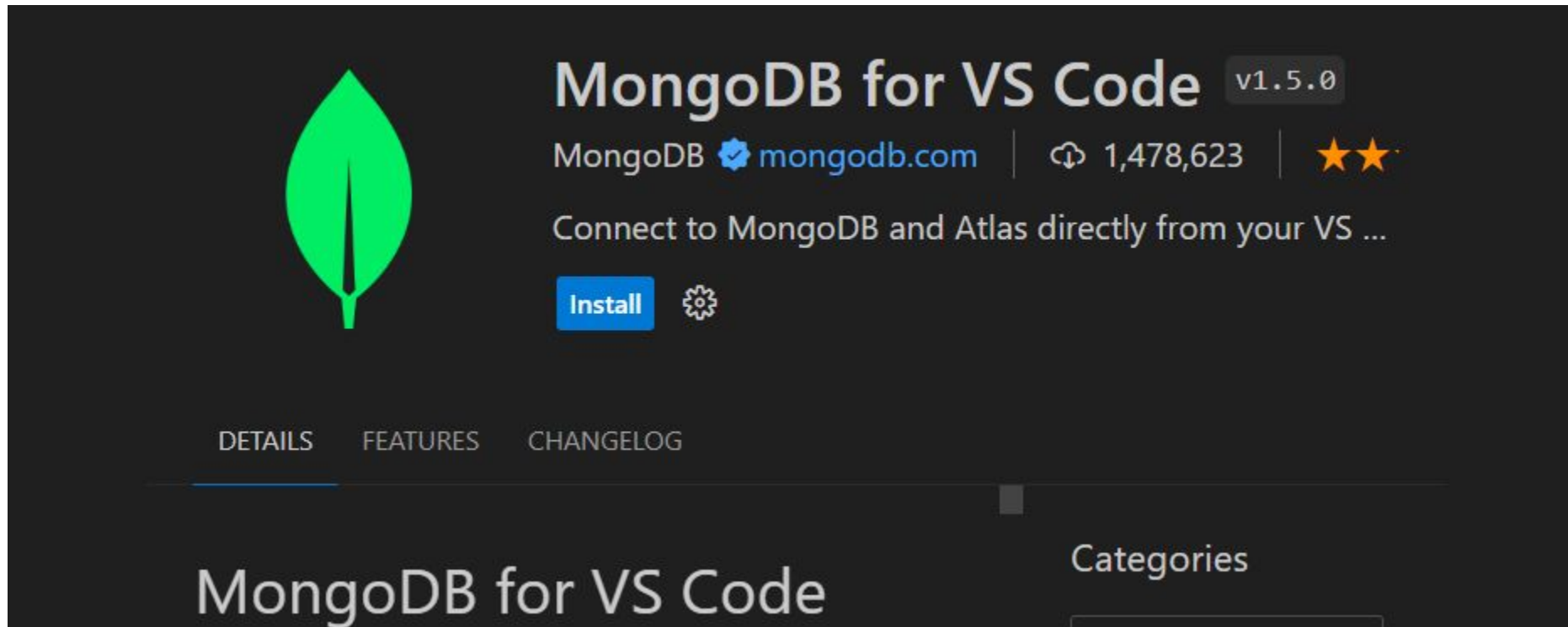


Big Data



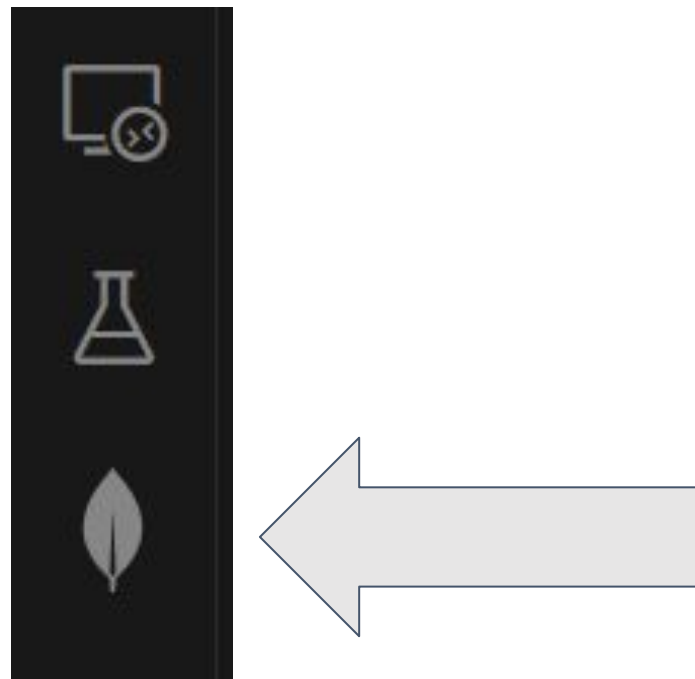
Big Data

4 - Instale a extensão MongoDB for VS Code.

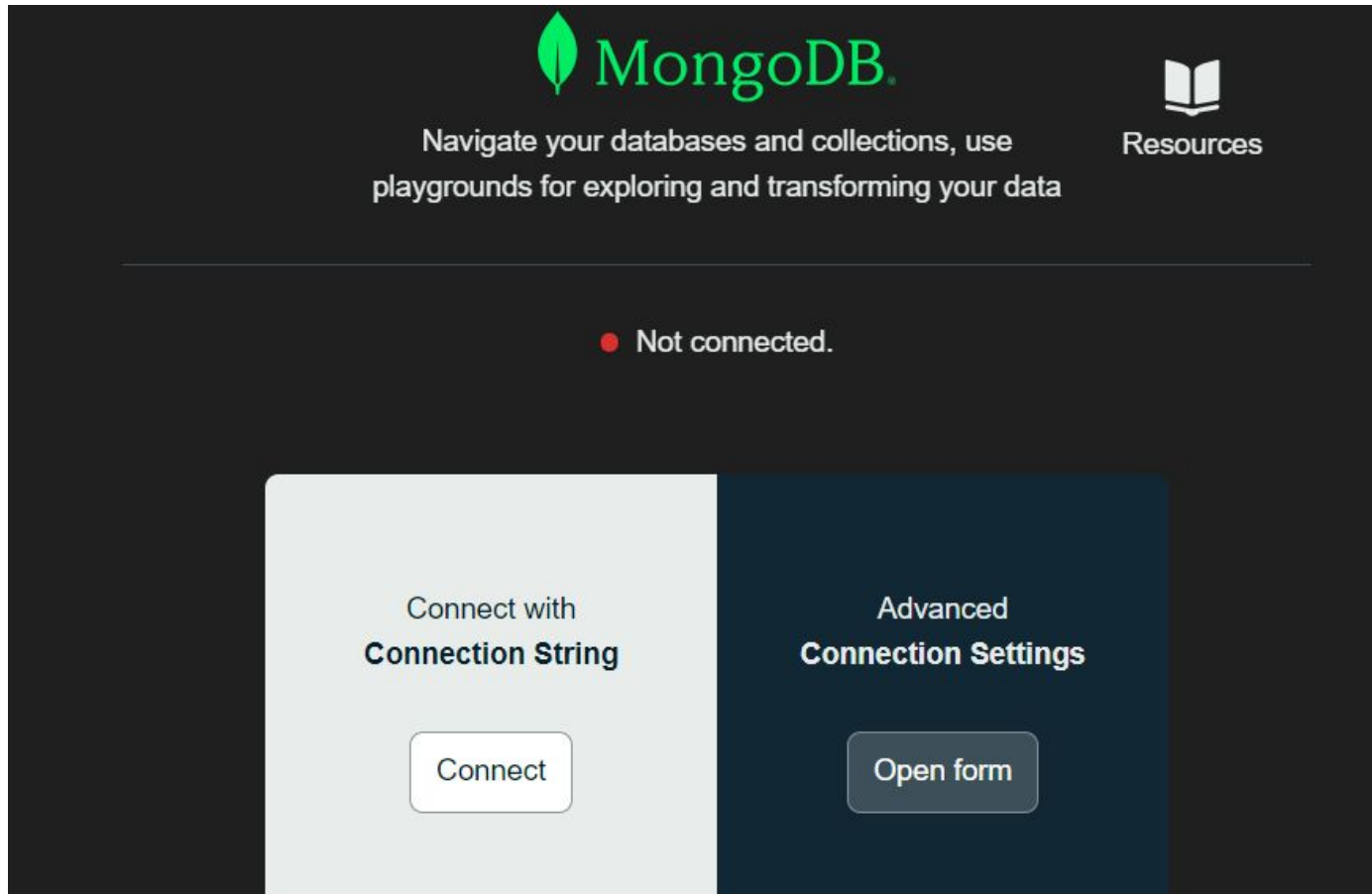


Big Data

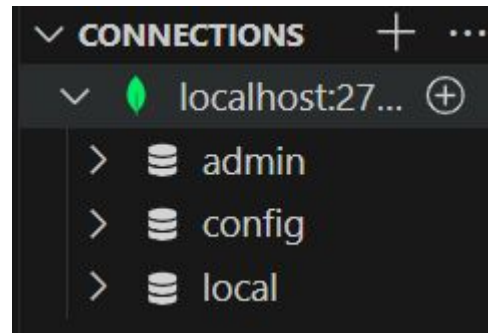
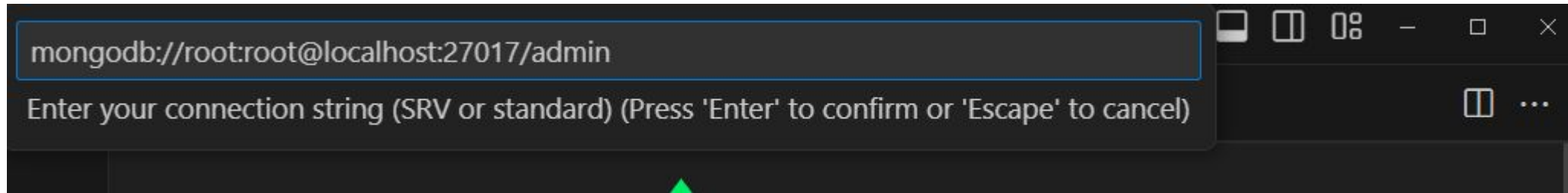
5 - Acesse a extensão (uma folha na barra lateral), clique em Add Connection e use as configurações:



Big Data



Big Data



Big Data

Mais

<https://balta.io/blog/mongodb-docker>

detalhes:

Configurando o MongoDB AtlasDB

Big Data

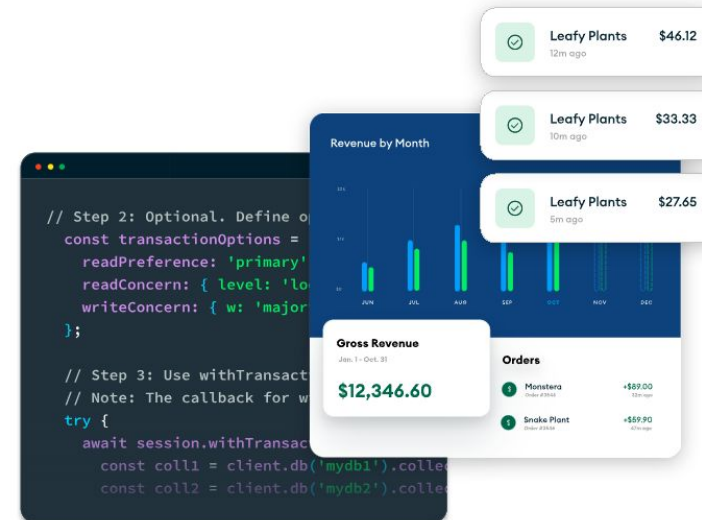
[Products](#)[Resources](#)[Solutions](#)[Company](#)[Pricing](#)[Support](#)[Sign In](#)[Try Free](#)

MONGODB ATLAS

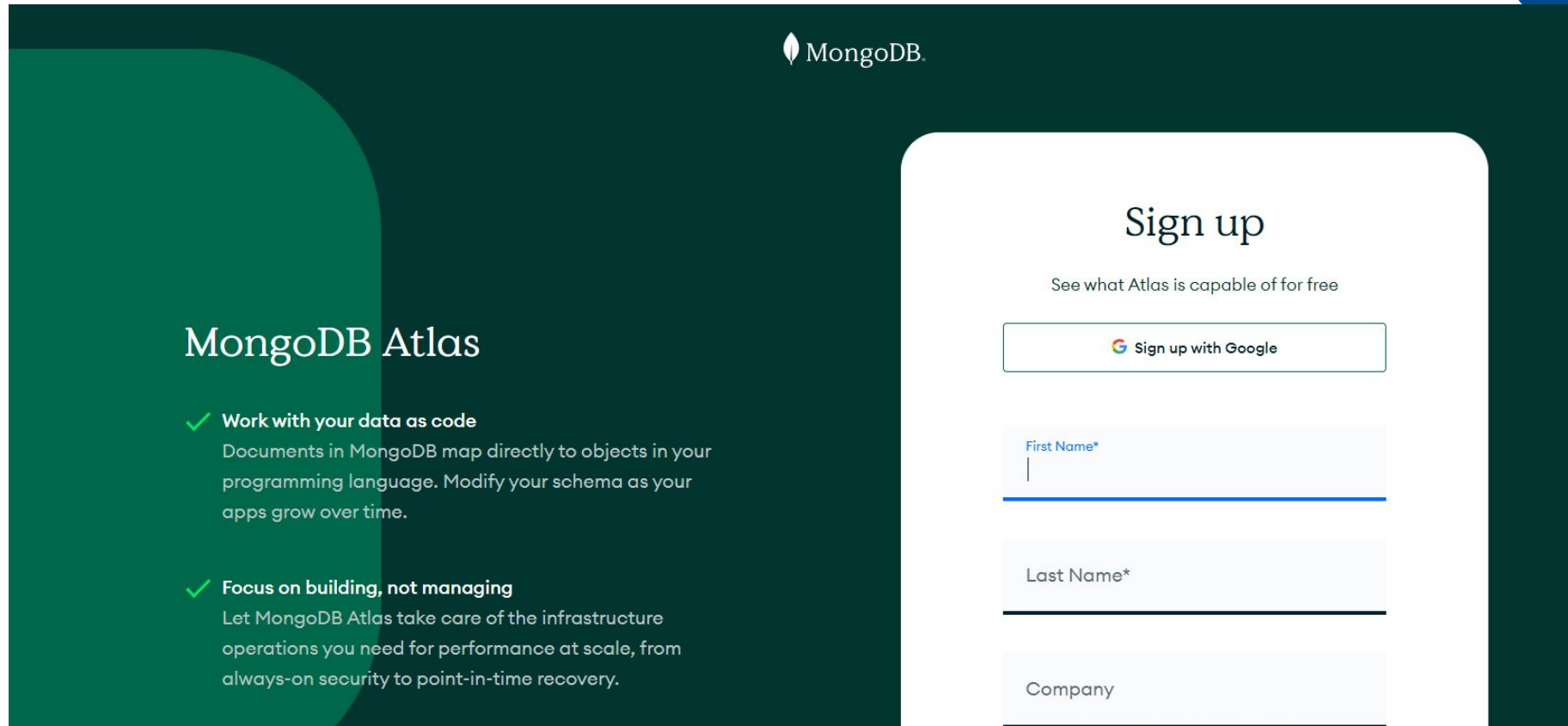
Loved by developers. Built for

Transactions

You don't need a separate database to support transactions, rich search, or genAI. The world's most popular document database is now the world's most versatile developer data platform.

[Start Free](#)[Learn More](#) →

Big Data



The screenshot shows the MongoDB Atlas sign-up page. On the left, the MongoDB logo is at the top right. Below it, the text 'MongoDB Atlas' is displayed. Underneath, there are two green checkmarks with bold text: 'Work with your data as code' and 'Focus on building, not managing'. Each checkmark is followed by a paragraph of text. On the right, there is a white rounded rectangle containing the 'Sign up' heading, a subtext 'See what Atlas is capable of for free', a 'Sign up with Google' button, and three input fields labeled 'First Name*', 'Last Name*', and 'Company'.


MongoDB

MongoDB Atlas

- ✓ **Work with your data as code**
Documents in MongoDB map directly to objects in your programming language. Modify your schema as your apps grow over time.
- ✓ **Focus on building, not managing**
Let MongoDB Atlas take care of the infrastructure operations you need for performance at scale, from always-on security to point-in-time recovery.

Sign up

See what Atlas is capable of for free

 Sign up with Google

First Name*


Last Name*

Company

Big Data

MARCO'S ORG - 2024-04-22 > PROJECT 0

Overview



Create a deployment

Choose your cloud provider, region, and specs.

[+ Create](#)

Toolbar

Featured Resources

GENERAL

- [Get Started with Atlas](#)
- [Reference MongoDB Documentation](#)
- [Develop Applications with the Developer Center](#)
- [Ask the MongoDB Community](#)

New On Atlas

2 NEW

Learn about the latest feature enhancements on Atlas.

Big Data

Deploy your database

Use a template below or set up advanced configuration options. You can also edit these configuration options once the cluster is created.

☐ M10

\$0.12/hour

For production applications with sophisticated workload requirements.

STORAGE	RAM	vCPU
10 GB	2 GB	2 vCPUs

☐ Serverless

For application development and testing, or workloads with variable traffic.

STORAGE	RAM	vCPU
Up to 1 TB	Auto-scale	Auto-scale

☒ M0

Free

For learning and exploring MongoDB in a cloud environment.

STORAGE	RAM	vCPU
612 MB	Shared	Shared

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

Name

You cannot change the name once the cluster is created.

Cluster0

☒ Automate security setup ⓘ

☒ Add sample dataset ⓘ

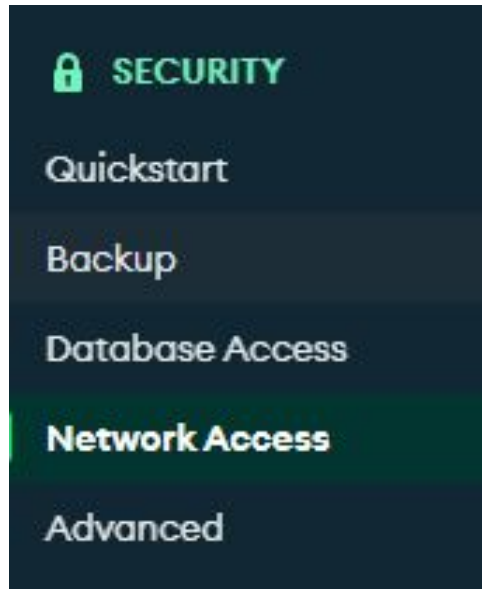
Provider

aws

Google Cloud

Azure

Big Data



Edit IP Access List Entry ✕

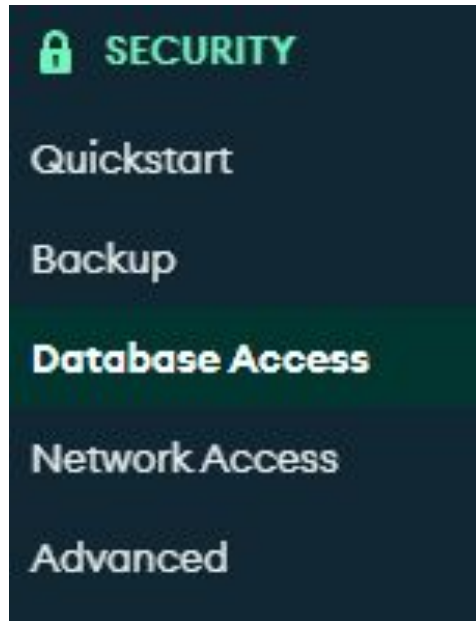
Atlas only allows client connections to a cluster from entries in the project's IP Access List. Each entry should either be a single IP address or a CIDR-notated range of addresses. [Learn more.](#)

ALLOW ACCESS FROM ANYWHERE

Access List Entry:

Comment:

Big Data



Authentication Method

Password

Certificate

AWS IAM
(MongoDB 4.4 and up)

PREVIEW
Federated Auth
(MongoDB 7.0 and up) ⓘ

MongoDB uses [SCRAM](#) as its default authentication method.

Password Authentication

[SHOW](#)

[🔍 Autogenerate Secure Password](#) [📄 Copy](#)

Big Data

Usar a connection string fornecida pelo atlas

Connect to ClusterTest

✓

✓

3

Set up connection securityChoose a connection methodConnect

Connecting with MongoDB for VS Code

1. Install MongoDB for VS Code.

In **VS Code**, open "Extensions" in the left navigation and search for "MongoDB for VS Code." Select the extension and click install.

2. In VS Code, open the Command Palette.

Click on "View" and open "Command Palette."
Search "MongoDB: Connect" on the Command Palette and click on "Connect with Connection String."

3. Connect to your MongoDB deployment.

Paste your connection string into the Command Palette.

`mongodb+srv://marcomialaret:<password>@cluster-test.5nzhzqx.mongodb.net/`

Replace **<password>** with the password for the **marcomialaret** user. Ensure any options are [URL encoded](#).

Acessando o banco com python

Big Data

Criar um venv com o requirements.txt da semana 07. Acesse o diretório no qual o requirements se encontra e execute o comando

```
python -m venv venv && .\venv\Scripts\activate  
&& pip install -r requirements.txt
```

Big Data



No VScode (se você tem o docker configurado):

```
from pymongo import MongoClient

# Crie uma conexão com o MongoDB (ajuste o host e a porta conforme
# necessário)
client = MongoClient("mongodb://root:root@localhost:27017/admin")
```

✓ 2.7s

Python

Big Data

No Colab (se você tem o atlas configurado):

```
from pymongo.mongo_client import MongoClient
from pymongo.server_api import ServerApi

uri = "mongodb+srv://<username>:<password>@<cluster>/admin"

client = MongoClient(uri, server_api=ServerApi('1'))

try:
    client.admin.command('ping')
    print("Pinged your deployment. You successfully connected to MongoDB!")
except Exception as e:
    print(e)
```

Big Data

```
# Seleciona o banco de dados
db = client["aula07"]

# Seleciona a coleção
collection = db["cotacoes"]
```

✓ 0.0s

Python

Big Data

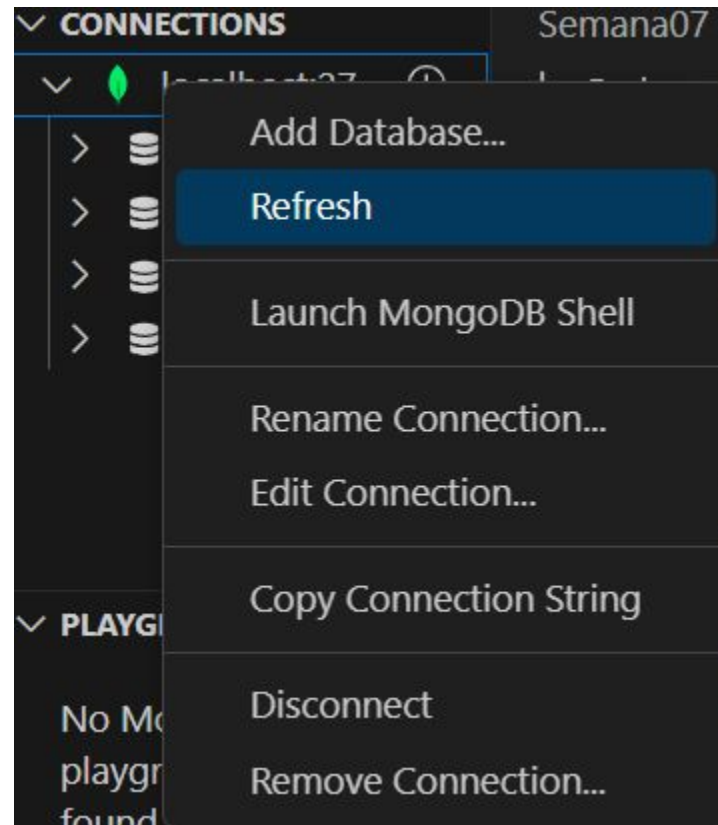
```
import json

# Carrega os dados do arquivo JSON
with open("cotacoes.json", "r") as file:
    data = json.load(file)
```

✓ 0.0s

Python

Big Data



Big Data



```
{} aula07.cotacoes:{"$oid":"661fe1414d0d69f8d2e883c1"}.json > ...
1  {}
2  "_id": {
3    "$oid": "661fe1414d0d69f8d2e883c1"
4  },
5  "USDBRL=X": {
6    "Open": {
7      "2023-04-17": 4.908699989318848,
8      "2023-04-18": 4.939799785614014,
9      "2023-04-19": 4.984300136566162,
10     "2023-04-20": 5.0742998123168945,
```


Dúvidas?



Marco Mialaret, MSc

Telefone:

81 98160 7018

E-mail:

marcomialaret@gmail.com

