



UNIVERSIDADE FEDERAL DO CEARÁ - UFC
CAMPUS CRATEÚS
CURSO DE SISTEMAS DE INFORMAÇÃO

Disciplina: Inteligência do Negócio e Visualização de dados
Professor: Bruno Riccelli dos Santos Silva
Semestre: 2025.1

Prática 2 - Data lake com MinIO

1. Introdução

Este documento apresenta um guia passo a passo para containerizar um Data lake compatível com S3 e observar algumas funcionalidades.

2. Composição de pastas e arquivos do projeto

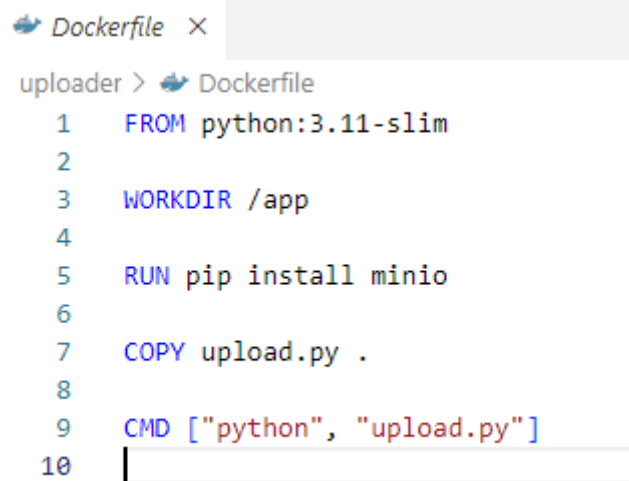
```
|— pratica-datalake/
|   |— data/
|   |   |— example.txt/
|   |— uploader/
|   |   |— Dockerfile
|   |   |— upload.py
|— docker-compose.yml
```

3. Crie uma pasta e navegue até ela

mkdir pratica-datalake

cd pratica-datalake

4. Dentro do arquivo `example.txt`, na pasta `data`, insira uma mensagem qualquer. Esse arquivo será enviado para o data lake.
5. Dentro da pasta `uploader`, crie um arquivo `Dockerfile` com o seguinte conteúdo



The screenshot shows a code editor with a tab labeled "Dockerfile" and a close button. The editor content shows the path "uploader > Dockerfile" followed by a list of Dockerfile instructions: "FROM python:3.11-slim", "WORKDIR /app", "RUN pip install minio", "COPY upload.py .", and "CMD ["python", "upload.py"]". Line numbers 1 through 10 are visible on the left side of the editor.

```
uploader > Dockerfile
1 FROM python:3.11-slim
2
3 WORKDIR /app
4
5 RUN pip install minio
6
7 COPY upload.py .
8
9 CMD ["python", "upload.py"]
10
```

6. Ainda dentro da mesma pasta, crie um arquivo chamado [`uploader.py`](#) com o seguinte conteúdo:

upload.py

uploader > upload.py

```
1  import os
2  import time
3  from minio import Minio
4  from minio.error import S3Error
5
6  endpoint = os.getenv("MINIO_ENDPOINT", "localhost:9000")
7  access_key = os.getenv("MINIO_ACCESS_KEY", "minioadmin")
8  secret_key = os.getenv("MINIO_SECRET_KEY", "minioadmin")
9
10 client = Minio(
11     endpoint,
12     access_key=access_key,
13     secret_key=secret_key,
14     secure=False
15 )
16
17 bucket = "raw"
18 file_path = "/app/data/example.txt"
19 object_name = os.path.basename(file_path)
20
21 time.sleep(5)
22
23
24 if not client.bucket_exists(bucket):
25     client.make_bucket(bucket) # Cria o bucket se não existir
26     print(f"Bucket '{bucket}' criado.")
27 else:
28     print(f"Bucket '{bucket}' já existe.")
29
30
31 try: # Envia o arquivo
32     client.fput_object(bucket, object_name, file_path)
33     print(f"Arquivo '{object_name}' enviado para '{bucket}'.")
34 except S3Error as err:
35     print("Erro no upload:", err)
36
```

7. Agora para o arquivo docker-compose.yaml

```
🚀 docker-compose.yaml
1  version: '3.8'
2
3  services:
4    minio:
5      image: minio/minio:latest
6      container_name: minio
7      environment:
8        MINIO_ROOT_USER: minioadmin
9        MINIO_ROOT_PASSWORD: minioadmin
10     volumes:
11       - minio_data:/data
12     ports:
13       - "9000:9000"
14       - "9001:9001"          # console UI
15     command: server /data --console-address ":9001"
16
17     uploader:
18       build: ./uploader
19       container_name: uploader
20       depends_on:
21         - minio
22       volumes:
23         - ./data:/app/data
24       environment:
25         MINIO_ENDPOINT: minio:9000
26         MINIO_ACCESS_KEY: minioadmin
27         MINIO_SECRET_KEY: minioadmin
28
29     volumes:
30       minio_data:
31
```

8. Construa e execute o container:

docker-compose up -d

9. Verifique se o contêiner está em execução:

docker-compose ps

10. Acesse `http://localhost:9001` (usuário/senha minioadmin) e verifique no bucket raw o arquivo `example.txt`

11. Faça o upload manual pela interface gráfica e verifique se o conteúdo do bucket foi alterado

12. Selecione e disponibilize um dos arquivos para um link baixável

13. Delete o container via `docker compose down`

14. Suba o container novamente

15. Por que o bucket raw ainda está ativo com o mesmo conteúdo de antes?

16. Agora que você tem um container funcionando, crie um novo container do python no qual seja criado um arquivo de um milhão de linhas, salvo em csv, converta-o em parquet e use o container uploader para enviar este arquivo para o bucket do MinIO.