02 - Atividade LAB AWS S3

1. {
2. "Version": "2012-10-17",
3. "Statement": [
4. {
5. "Sid": "PublicReadGetObject",
6. "Effect": "Allow",
7. "Principal": "\*",
8. "Action": [
9. "s3:GetObject"
10. ],
11. "Resource": [
12. "arn:aws:s3:::bucketfelix/\*"
13. ]
14. }
15. ]
16. }

Arquivo index.html

1. <html xmlns="http://www.w3.org/1999/xhtml" >
2. <head>
3. <title>Home Page do meu WebSite - Tutorial de S3</title>
4. </head>
5. <body>
6. <h1>Bem-vindo ao meu website</h1>
7. <p>Agora hospedado em Amazon S3!</p>
8. <a href="nomes.csv ">Download CSV File</a>
9. </body>
10. </html>

O arquivo error.html deixei vazio mesmo, mas dava pra adicionar uma mensagem de erro caso não conseguisse acessar o site!!

03 – LAB AWS ATHENA

CREATE EXTERNAL TABLE IF NOT EXISTS meubanco.analise (

nome STRING,

sexo STRING,

total INT,

ano INT

)

ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'

WITH SERDEPROPERTIES ( 'serialization.format' = ',', 'field.delim' = ','

)

LOCATION 's3://bucketfelix/dados'

Consulta para selecionar os 3 nomes mais usados em cada década desde 1950!!

WITH RankedNomes AS (

SELECT

nome,

ano,

total,

ROW\_NUMBER() OVER (PARTITION BY CAST(ano AS INT)/10 ORDER BY total DESC)

AS Rank

FROM meubanco.analise

WHERE ano >= 1950

)

SELECT DISTINCT

nome,

ano,

Rank

FROM RankedNomes

WHERE Rank <= 3

ORDER BY ano, Rank;

04 – LAB AWS LAMBDA

1. import json
2. import pandas
3. import boto3

6. def lambda\_handler(event, context):
7. s3\_client = boto3.client('s3')
9. bucket\_name = 'bucketfelix'
10. s3\_file\_name = 'dados/nomes.csv'
11. objeto = s3\_client.get\_object(Bucket=bucket\_name, Key=s3\_file\_name)
12. df=pandas.read\_csv(objeto['Body'], sep=',')
13. rows = len(df.axes[0])
15. return {
16. 'statusCode': 200,
17. 'body': f"Este arquivo tem {rows} linhas"
18. }

Criação do arquivo Dockerfile

1. FROM amazonlinux:2.0.20200602.0
2. RUN yum update -y
3. RUN yum install -y \
4. python3-pip \
5. zip \
6. RUN yum -y clean all
7. RUN python3.7 -m pip install --upgrade pip

Construção da imagem à partir do arquivo Dockerfile

1. docker build -t amazonlinuxpython37 .

Rodando um container à partir da imagem criada e entrando no seu terminal

1. docker run -it amazonlinuxpython37 bash

Navegando nas pastas da imagem do Linux e criando a pasta python com a biblioteca pandas

1. bash-4.2# cd ~
2. bash-4.2# mkdir layer\_dir
3. bash-4.2# cd layer\_dir/
4. bash-4.2# mkdir python
5. bash-4.2# cd python/
6. bash-4.2# pwd

Baixando a biblioteca pandas na pasta python

1. bash-4.2# pip3 install pandas -t .

Criando um arquivo zip da pasta python com a biblioteca pandas

1. bash-4.2# zip -r minha-camada-pandas.zip .

Copiando o arquivo zip do container para minha maquina local

docker cp b0921d917152:/root/layer\_dir/minha-camada-pandas.zip ./

Depois foi alterado o tamanho da memoria e tempo limite para poder executar o código python e obter o resultado abaixo:

1. {
2. "statusCode": 200,
3. "body": "Este arquivo tem 1825433 linhas"
4. }

05 – LAB AWS – Limpeza de recursos

Logo em seguida os recursos utilizados foram excluídos concluindo assim as atividades!