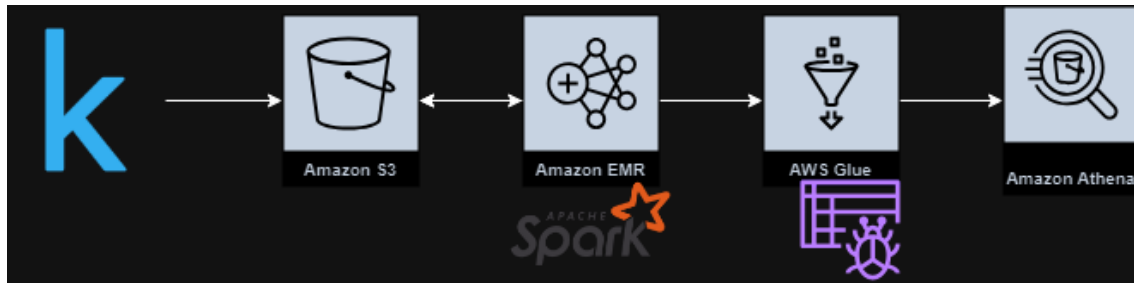


Arquitetura e Passo a passo do ETL

Este é o desenho da Arquitetura:



O primeiro passo foi fazer o download nas bases do Kaggle e carregá-los em um bucket do S3. Infelizmente, este passo foi 'manual', pois não tive tempo hábil de criar a API do Kaggle para fazer a extração automática dos arquivos.

Criei um datalake onde existem 3 camadas: **bronze** (que recebe os arquivos crus do Kaggle), **silver** (após limpeza e primeiras transformações) e **gold** (dataframes prontos para análise)

Fiz então o upload dos arquivos na camada **bronze**, as separando por fonte:

datalake-desafio-1 [Informações](#)

[Objetos](#) | [Propriedades](#) | [Permissões](#) | [Métricas](#) | [Gerenciamento](#) | [Pontos de acesso](#)

Objetos (4) [Informações](#) [Atualizar](#) [Copiar URI do S3](#) [Copiar URL](#)

Os objetos são as entidades fundamentais armazenadas no Amazon S3. Você pode usar o [inventário do Amazon S3](#) para obter u explicitamente a eles. [Saiba mais](#)








☐ Mostrar versões

<input type="checkbox"/>	Nome	Tipo	Última modif
<input type="checkbox"/>	bronze/	Pasta	-
<input type="checkbox"/>	consultas/	Pasta	-
<input type="checkbox"/>	gold/	Pasta	-
<input type="checkbox"/>	silver/	Pasta	-

Amazon S3 > Buckets > datalake-desafio-1 > bronze/ > amazon/




amazon/

Objetos | Propriedades

Objetos (3) Informações   Copiar URI do S3  Copiar URL  Fazer download  Abrir  Excluir  Ações ▼

Os objetos são as entidades fundamentais armazenadas no Amazon S3. Você pode usar o [inventário do Amazon S3](#) para obter uma lista de todos os objetos em seu bucket. Para outras pessoas acessarem seus objetos, você precisa configurá-los explicitamente a eles. [Saiba mais](#)








☐ Mostrar versões

<input type="checkbox"/>	Nome ▲	Tipo ▼	Última modificação ▼	Tamanho ▼	Classe de armazenamento
<input type="checkbox"/>	 amazon_reviews_us_Digital_Video_Download_v1_00.tsv	tsv	5 Aug 2024 05:18:53 PM -03	1.2 GB	Padrão
<input type="checkbox"/>	 amazon_reviews_us_Video_DVD_v1_00.tsv	tsv	5 Aug 2024 05:18:53 PM -03	3.5 GB	Padrão
<input type="checkbox"/>	 amazon_reviews_us_Video_v1_00.tsv	tsv	5 Aug 2024 05:18:53 PM -03	322.3 MB	Padrão

Amazon S3 > Buckets > datalake-desafio-1 > bronze/ > netflix/









netflix/

Objetos | Propriedades

Objetos (8) Informações   Copiar URI do S3  Copiar URL  Fazer download  Abrir  Excluir  Ações ▼

Os objetos são as entidades fundamentais armazenadas no Amazon S3. Você pode usar o [inventário do Amazon S3](#) para obter uma lista de todos os objetos em seu bucket. Para outras pessoas acessarem seus objetos, você precisa configurá-los explicitamente a eles. [Saiba mais](#)

☐ Mostrar versões

<input type="checkbox"/>	Nome ▲	Tipo ▼	Última modificação ▼	Tamanho ▼	Classe de armazenamento
<input type="checkbox"/>	 combined_data_1.txt	txt	5 Aug 2024 03:42:33 PM -03	472.1 MB	Padrão
<input type="checkbox"/>	 combined_data_2.txt	txt	5 Aug 2024 03:42:33 PM -03	529.5 MB	Padrão
<input type="checkbox"/>	 combined_data_3.txt	txt	5 Aug 2024 03:42:33 PM -03	443.6 MB	Padrão
<input type="checkbox"/>	 combined_data_4.txt	txt	5 Aug 2024 03:42:33 PM -03	526.9 MB	Padrão
<input type="checkbox"/>	 movie_titles.csv	csv	5 Aug 2024 03:42:35 PM -03	564.0 KB	Padrão
<input type="checkbox"/>	 probe.txt	txt	5 Aug 2024 03:42:38 PM -03	10.3 MB	Padrão
<input type="checkbox"/>	 qualifying.txt	txt	5 Aug 2024 03:42:33 PM -03	50.0 MB	Padrão
<input type="checkbox"/>	 README	-	5 Aug 2024 03:42:39 PM -03	5.8 KB	Padrão

Feito isso, construí as transformações através de dois scripts: **functions.py**, onde se encontra as funções usadas no script **main.py** para realizar as transformações. Criei um bucket como repositório onde contém estes scripts:

Amazon S3 > Buckets > main-scripts > desafio/ > scripts/

scripts/

Objetos | Propriedades

Objetos (3) Informações

Os objetos são as entidades fundamentais armazenadas no Amazon S3. Você pode usar o [inventário do Amazon S3](#) para obter uma lista de todos os objetos e explicitamente a eles. [Saiba mais](#)

Localizar objetos por prefixo

<input type="checkbox"/>	Nome	Tipo	Última modificação
<input type="checkbox"/>	functions.py	py	7 Aug 2024 02:06:58 PM -03
<input type="checkbox"/>	main.py	py	7 Aug 2024 02:06:59 PM -03
<input type="checkbox"/>	step_function.json	json	7 Aug 2024 02:07:00 PM -03

Após a criação dos scripts, subi um cluster EMR e subi o seguinte comando na **step** do EMR, este mesmo **.jar** se contra em **bootstrap/ step.jar**:

Amazon EMR > EMR no EC2: Clusters > emr-pedido

Atualizado há 8 minutos

emr-pedido

Resumo

Propriedades | Ações de bootstrap | Instâncias (hardware) | **Etapas** | Aplicativos | Configurações | Monitoramento | Eventos | Tags (0)

Etapas (4) Informações

Cada etapa é uma unidade de trabalho que contém instruções para manipular dados processados pelo software instalado no cluster.

Etapas simultâneas: 1

Filtrar etapas por status

<input type="checkbox"/>	ID da etapa	Status	Nome	Arquivos de log	Horário de criação (UTC-03:00)	Horário de início (UTC-03:00)	Tempo decorrido
<input checked="" type="checkbox"/>	s-05812657LCTL8T9I9RW	Completed	commander_runner	controller syslog stderr stdout	August 7, 2024 at 19:44	August 7, 2024 at 19:45	33 minutos, 44 segundos

Localização do JAR command-runner.jar	Permissões -	Classe principal -
Ação em caso de falha Continue	Argumento spark-submit --master yarn --deploy-mode cluster --conf spark.net.work.timeout=10000001 --conf spark.executor.heartbeatInterval=10000000 --py-files s3://main-scripts/desafio/scripts/functions.py s3://main-scripts/desafio/scripts/main.py	

O script gera arquivos **parquet** (formato mais otimizado que os de texto) nas camadas **silver** e **gold**. Na camada **silver**, estão os arquivos que sofreram as primeiras transformações, já na **gold**, estão as 3 tabelas finais cujo modelo lógico segue exemplos:

Silver:

Amazon S3 > Buckets > datalake-desafio-1 > silver/ > tables/

tables/

Objetos | Propriedades

Objetos (3) Informações

Os objetos são as entidades fundamentais armazenadas no Amazon S3. Você pode usar o [inventário do Amazon](#) explicitamente a eles. [Saiba mais](#)

Localizar objetos por prefixo

Mostrar

<input type="checkbox"/>	Nome	Tipo
<input type="checkbox"/>	amazon_final/	Pasta
<input type="checkbox"/>	netflix_movie_title/	Pasta
<input type="checkbox"/>	processed_netflix/	Pasta

Amazon S3 > Buckets > datalake-desafio-1 > silver/ > tables/ > amazon_final/

amazon_final/

Objetos | Propriedades

Objetos (42) Informações

Os objetos são as entidades fundamentais armazenadas no Amazon S3. Você pode usar o [inventário do Amazon](#) explicitamente a eles. [Saiba mais](#)

Localizar objetos por prefixo

Mostrar versões

<input type="checkbox"/>	Nome	Tipo
<input type="checkbox"/>	_SUCCESS	-
<input type="checkbox"/>	part-00000-3f828917-a29e-4de9-8ac2-b1f81dd7072-c000.snappy.parquet	parquet
<input type="checkbox"/>	part-00001-3f828917-a29e-4de9-8ac2-b1f81dd7072-c000.snappy.parquet	parquet
<input type="checkbox"/>	part-00002-3f828917-a29e-4de9-8ac2-b1f81dd7072-c000.snappy.parquet	parquet

Gold:

Amazon S3 > Buckets > datalake-desafio-1 > gold/ > tables/

tables/

Objetos | Propriedades

Objetos (3) Informações

Os objetos são as entidades fundamentais armazenadas no Amazon S3. Você pode us explicitamente a eles. [Saiba mais](#)

Localizar objetos por prefixo

Mostrar

<input type="checkbox"/>	Nome	Tipo
<input type="checkbox"/>	customer/	Pasta
<input type="checkbox"/>	movies/	Pasta
<input type="checkbox"/>	ratings/	Pasta

Amazon S3 > Buckets > datalake-desafio-1 > gold/ > tables/ > ratings/

ratings/

Objetos | Propriedades

Objetos (43) Informações

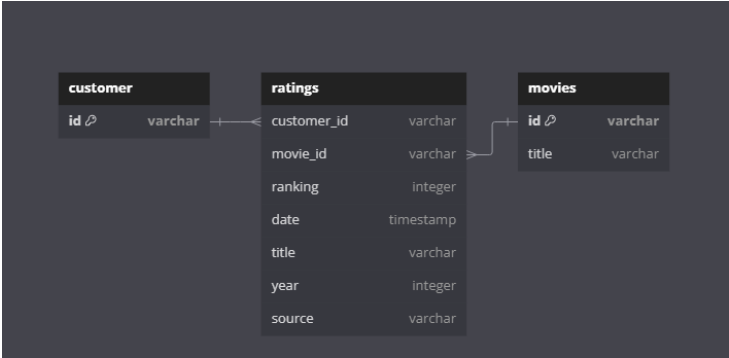
Os objetos são as entidades fundamentais armazenadas no Amazon S3. Você pode usar o [inventário do Am](#) explicitamente a eles. [Saiba mais](#)

Localizar objetos por prefixo

Mostrar versões

<input type="checkbox"/>	Nome	Tipo
<input type="checkbox"/>	_SUCCESS	-
<input type="checkbox"/>	part-00000-4ef5c529-5851-45d9-a487-67b44c32947a-c000.snappy.parquet	parquet
<input type="checkbox"/>	part-00001-4ef5c529-5831-45d9-a487-67b44c32947a-c000.snappy.parquet	parquet
<input type="checkbox"/>	part-00002-4ef5c529-5831-45d9-a487-67b44c32947a-c000.snappy.parquet	parquet

Modelo lógico das tabelas geradas na camada gold:



Feita a carga nas camadas, foi criado um **database** e um **crawler** para a criação das tabelas da camada **gold** no **Glue**:

AWS Glue > Databases > desafio

desafio Last updated (UTC) August 8, 2024 at 00:41:04 Edit Delete

Database properties

Name	desafio	Description	-	Location	-	Created on (UTC)	August 7, 2024 at 10:35:15
------	---------	-------------	---	----------	---	------------------	----------------------------

Tables (3) Last updated (UTC) August 8, 2024 at 00:41:05 Delete Add tables using crawler Add table

View and manage all available tables.

<input type="checkbox"/>	Name	Database	Location	Classification	Deprecated	View data	Data quality
<input type="checkbox"/>	customer	desafio	s3://datalake-desafio-1/gold/	Parquet	-	Table data	View data quality
<input type="checkbox"/>	movies	desafio	s3://datalake-desafio-1/gold/	Parquet	-	Table data	View data quality
<input type="checkbox"/>	ratings	desafio	s3://datalake-desafio-1/gold/	Parquet	-	Table data	View data quality

Com isto, todas as consultas foram realizadas no **Athena** usando a tabela **ratings**. As consultas se encontram no arquivo **sql/queries.sql** e os resultados estão no arquivo **email_cliente.docx**. Segue exemplo das consultas realizadas no **Athena**:

O Athena passou a oferecer suporte a sugestões de digitação antecipada para códigos com a finalidade de acelerar o desenvolvimento de consultas SQL.
As sugestões de digitação antecipada estão ativadas por padrão. Você pode alterar essa configuração nas preferências do editor de consultas.

Dados

Fonte de dados: **AwsDataCatalog**

Banco de dados: **desafio**

Tabelas e visões: **Criar**

▼ Tabelas (3)

- customer: id (string)
- movies: id (string), title (string)
- ratings: customer_id (string), movie_id (string), ranking (int), date (date), title (string)

pergunta_3 | X | pergunta_4 | X | pergunta_5 | X | pergunta_6 | X | pergunta_7 | X | pergunta_8 | X

```

5 GROUP BY title
6 ),
7 - netflix_ratings AS (
8   SELECT title, AVG(ranking) AS avg_rating
9   FROM "desafio"."ratings"
10  WHERE source = "netflix"
11  GROUP BY title
12 ),
13 - amazon_avg AS (
14   SELECT AVG(avg_rating) AS avg_amazon
15   FROM amazon_ratings
16 ),

```

SQL Ln 19, Col 23

Executar novamente Explicar Cancelar Limpar Criar

Resultados da consulta Estatísticas da consulta

Concluído Tempo na fila: 105 ms

Resultados (1)

Linhas de pesquisa

#	difference
1	0.82

Dados

Fonte de dados: **AwsDataCatalog**

Banco de dados: **desafio**

Tabelas e visões: **Criar**

▼ Tabelas (3)

- customer: id (string)
- movies: id (string), title (string)
- ratings: customer_id (string), movie_id (string), ranking (int)

pergunta_5 | X | pergunta_6 | X | pergunta_7 | X | pergunta_8 | X | pergunta_9 | X | pergunta_10 | X | pergunta_11 | X

```

1 SELECT title, COUNT(*) AS review_count
2 FROM "desafio"."ratings"
3 GROUP BY title
4 ORDER BY review_count DESC
5 LIMIT 10;
6

```

SQL Ln 1, Col 39

Executar novamente Explicar Cancelar Limpar Criar

Resultados da consulta Estatísticas da consulta

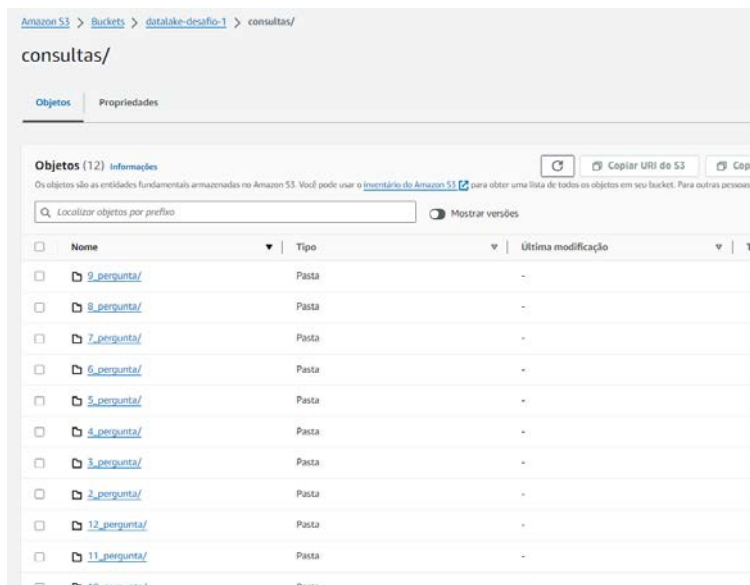
Concluído Tempo na fila: 64 ms Tempo de execução: 6.849 sec

Resultados (10)

Linhas de pesquisa

#	title	review_count
1	Miss Congeniality	233692
2	Independence Day	217223
3	The Patriot	212429

Os resultados de todas as consultas foram gravadas no bucket de destino **s3://datalake-desafio-1/consultas/**:



A pasta com o resultado das consultas se encontra no repositório na pasta **consulta**.

A orquestração seria feita pela **Step Functions**, cujo modelo se encontra em **scripts\step_function.json**, porém não tive tempo hábil para realizá-lo. Segue o design:

