DESAFIO TÉCNICO
ENGENHEIRO DE DADOS ITAÚ-UNIBANCO

Lucas Ximenes da Fonseca

DESAFIO:

O desafio técnico para Engenheiro de Dados consiste em elaborar uma proposta de Arquitetura de Data Lake, escolhendo as tecnologias de cada camada, justificando cada uma e defendendo a arquitetura. Além disso, como Objetivo 2 foi apresentado um arquivo para alguns tratamentos utilizando Spark. O desafio pode ser acessado através da URL: https://github.com/repositoriodados/case_ehgfngenheiro_dados.

Todo o conteúdo desenvolvido está disponível no Microsoft Azure e disponibilizado da seguinte maneira:

BIG DATA E DATA LAKE:

É sabido que a cada dia que se passa, cresce a quantidade de dados gerados diariamente. Estima-se que produzimos em 1 dia, a mesma quantidade de dados que foi gerada em 1 ano inteiro no início dos anos 2000. Esse crescimento desenfreado, muito alinhado com o advento da 'Era dos Smartphones' a partir de 2010, fez com que tradicionais técnicas de armazenamento e processamento de dados fossem modificadas, seja pela volumetria, pela velocidade, pela variedade ou devido ao valor que esses dados representam e a necessidade de trabalha-los em tempo hábil. Surgiu então o Big Data.

O Big Data, como mencionado, veio para tratar o maior ativo que existe em qualquer empresa, os seus Dados. É por ele que fracassos podem ser evitados e sucessos podem ser potencializados. Entretanto, dispor de um Engenharia robusta e ao mesmo tempo capaz de processar milhões de dados em poucos segundos não é uma tarefa trivial. Daí surgem os *Data Lakes*, como alternativa arquitetural suficientemente encorpada e performática para essa demanda apresentada.

O termo *Data Lake* foi criado para se descrever de forma apropriada um tipo de repositório como um lago, pois ele armazena um conjunto de dados em seu estado natural, ou seja, podemos pensar em uma camada de storage centralizada, porém, esses dados permanecem em suas estruturas originais, não havendo no primeiro momento transformações em seus *schemas*.

Obviamente, que para entregar maior valor, a estrutura de um *Data Lake* deve apresentar mais camadas, não se limitando a ingerir os dados em sua forma natural. É necessário transforma-los e entrega-los para diversos níveis de usuários (analistas de dados, cientistas de dados, squads de negócios, gestores, etc) com granularidades e permissões diferentes.

A empresa *Databricks* conceitua de forma interessante a arquitetura de um *Data Lake* (imagem abaixo) e através desse conceito lógico podemos tomar como base para a criação de uma arquitetura específica para o desafio proposto:

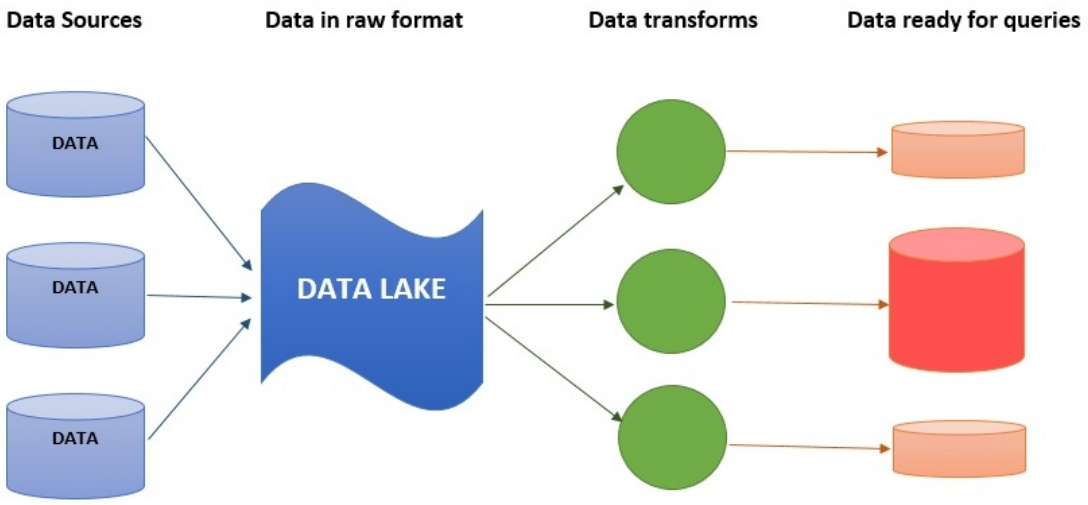

Figura 1: Modelagem de Data Lake.

Em rápidas palavras podemos definir cada nível da seguinte forma:

1. **DATA SOURCES**: Camada de entrada, são as fontes de dados de sistemas transacionais(OLTP), arquitetura de Data Warehouse(OLAP), planilhas, dados de sensores, Mainframes, sistemas legados etc. Enfim, tudo que gera dados relevantes ao negócios são caracterizados e “rotulados” como Data Source;
2. **DATA IN RAW FORMAT**: Camada de *storage*. É nesse nível que os dados extraídos dos *Data Sources* são armazenados em seu estado ‘puro’, podendo haver pequenas transformações, como o tipo de arquivo/extensão para uma melhor otimização de busca, recuperação, compactação e redução de custos;
3. **DATA TRANSFORMS**: Nesse nível há transformações mais profundas nos dados. A partir desse ponto, começa a ser observado a produtização/deploy das informações. Processos como Limpeza de Dados, criações de novas features/variáveis, primeiras análises exploratórias, automações de processos, entre outros, são efetuados nessa camada;
4. **DATA READY FOR QUERIES**: Nessa etapa os dados estão prontos para serem consumidos pela maioria dos atores técnicos que dependem do *lake* para de fato construírem produtos, visões, análises e fazer deploy de aplicações.

Podemos adicionar mais 2 níveis importantes nessa arquitetura, que são:

5. **ACCESS & SECURITY**: Camada essencial de todo produto e que deve ter sua importância explícita em um projeto de *Data Lake*. Os dados devem possuir claras políticas de acesso, vários e altos níveis de segurança;
6. **CONSUME**: Nível final de todo trabalho feito no *Data Lake*. Nessa camada os dados estão totalmente prontos para consumo.

ARQUITETURA DE DATA LAKE PROPOSTO

Diante do que foi exposto anteriormente, criei uma arquitetura de *Data Lake* de acordo com as informações obtidas no cenário descrito no desafio.

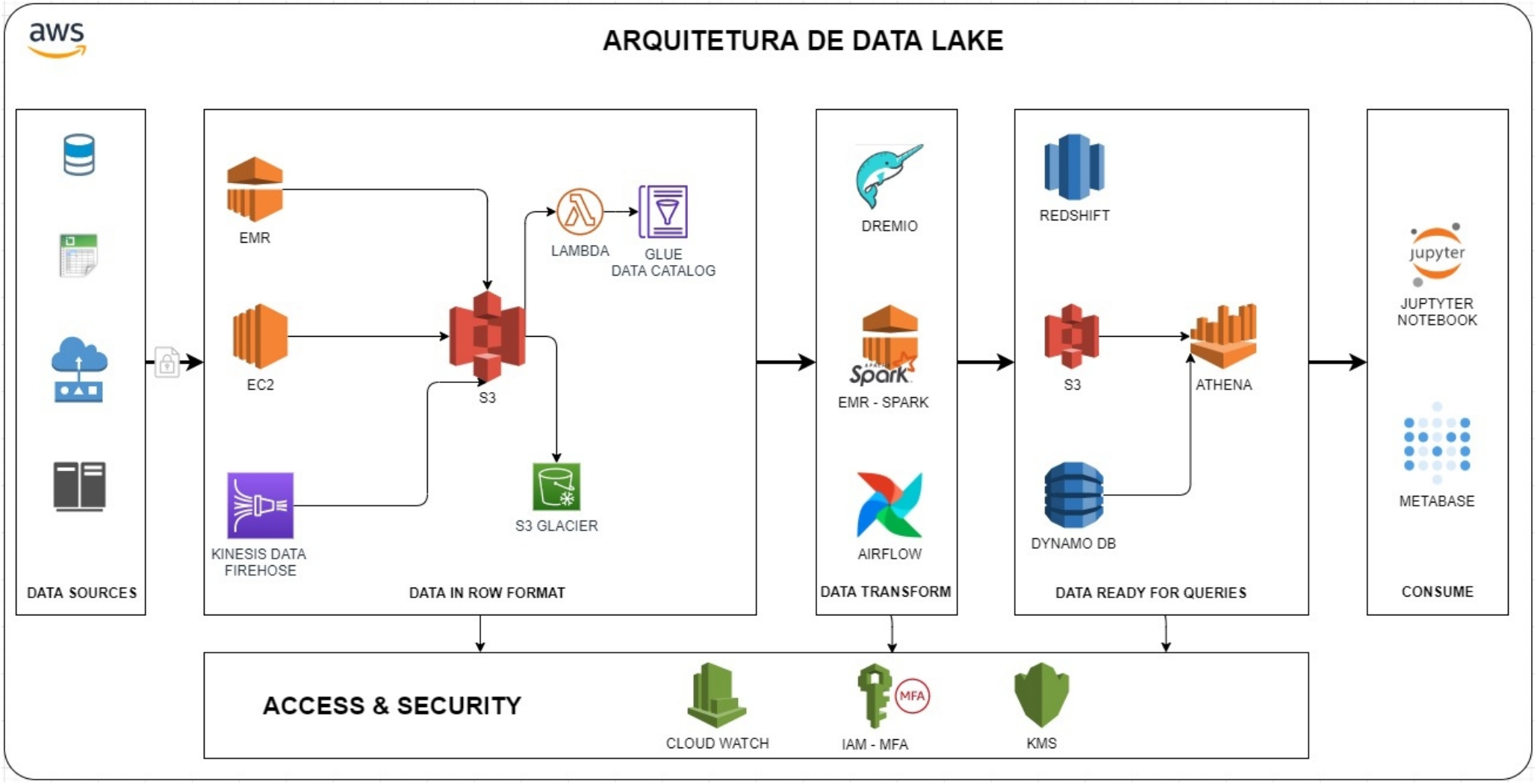

Figura 2. Data Lake proposto

JUSTIFICATIVA:

A **AWS** foi escolhida como Cloud Broker para a arquitetura proposta por ser líder de mercado há mais de uma década em soluções em Cloud, segundo a Gartner. Além disso, a AWS está em constante melhoria, com inúmeras ferramentas gerenciadas, facilitando a implantação de *Data Lake,* fazendo que a organização tenha maior preocupação com o negócio. Inclusive, vale ressaltar que um produto que poderia, também, ser proposto para criação do *Lake* é o *Lake Formation,* onde diversos aplicativos possuem altos níveis de abstração, tornando possível o levantamento de um *Data Lake* dentro de alguns dias.

A camada de **Data Sources**, é a camada que possui as milhares de fontes de dados disposto na corretora, assim como sistemas em legado, mainframes, data warehouses, planilhas, possíveis sensores (IoT), entre outras.

A camada de **Data in row format**, é a camada de storage. Para sua concepção, atentei ao volume de dados, necessidade de dados *Real Time,* base de dados grandes, centralização de dados em buckets do S3 com rotinas de Catálogo dos dados e visando sempre a otimização de Custos, Seguem os componentes:

- ✓ Amazon **EMR**: Plataforma para *Big Data*, ideal para fazer ingestão de grandes bases que necessitem de processamento paralelo;
- ✓ Amazon **EC2**: Produto para computação em nuvem (VM). Possui a possibilidade de redimensionalidade, e pode ser utilizados como máquinas de crawler para extração de dados externas ou extração e ingestão de bases que não precisam ser tratadas no EMR;
- ✓ Amazon **Kinesis Data Firehose**: Serviço para ingestão de dados em *streaming*. De capacidade escalável e totalmente gerenciado, torna-se fácil e seguro utiliza-lo para manter os dados que precisam estarem disponíveis em tempo real;
- ✓ Amazon **S3**: Serviço de extrema importância para o *Data Lake*, é o componente de armazenamento de dados. Ele trabalha com conceitos de buckets (espécie de contêineres de objetos) e possuem níveis de permissões que o torna boa alternativa em questões de segurança e governança;
- ✓ Amazon **S3 Glacier**: Certamente nem todos os dados do *lake* terão acesso diariamente, ou frequentemente. Dados que não são tão frequentes assim, podem ficar em um local onde o acesso não é feito de forma imediata, sendo assim mais barato, trazendo redução de custos;
- ✓ AWS **Lambda**: Serviço de computação que permite execução de código sem a necessidade de provisionar ou gerenciar servidores. Nessa arquitetura em específico, ele tem o papel de disparar evento para chamada do Glue Data Catalog;
- ✓ Aws **Glue Data Catalog**: Catálogo centralizado e único. É de extrema importância ter metadados sobre o *Data Lake*. O serviço é chamado por rotinas de acordo com a necessidade do negócio e atualizações dos dados.

A camada de **Data Transform** receberá as transformações mais profundas referentes aos dados originalmente coletados. Há enriquecimentos, limpeza e mais tratamentos nos dados. Segue os componentes:

- ✓ **Dremio**: Plataforma que unifica as camadas de storage com a camada de pesquisa. Através de suas *Reflections,* o Dremio consegue acelerar consultas em vários tipos de bancos de dados ou arquivos. No conceito desse *Data Lake*, ele se

encaixa perfeitamente para fazer transformações para salvar arquivos no S3 ou ainda servir como um nível de *Analytics Sandbox* do *lake*. Certamente ele poderia parecer também na próxima camada, mas, para auxiliar cientistas e analistas de dados antes de processar o consumo desses dados, é válida sua presença na camada de transformação;

- ✓ Apache **Spark**: Um dos principais componentes do *Data Lake*, ele aparece junto com o EMR, pois essa ferramenta da AWS já provê levantar cluster com o Spark, facilitando seu uso. Ele é o componente chave para transformações de dados, devido ao seu poder de processar grandes volumes de dados. Pode ser executado em diversas linguagens;
- ✓ **Airflow**: Plataforma para gerenciamento de fluxo. Ela trabalha com o conceito de DAG (Grafos Acíclicos Dirigidos) e possuem enorme poder para criar pipeline de dados e automação de cargas de ingestões. No contexto desse *lake* ele funcionaria como agendador de tarefas para transformações de dados a nível de ingestão no S3 para persistência de dados em local de consumo. Também poderia aparecer na primeira camada como ferramenta de ETL para primeira ingestão dos dados.

A camada de **Data ready for queries**, como o próprio nome sugere, é a camada em que os dados estão prontos para serem consumidos. Todo tratamento, enriquecimento, limpeza já foram feitos e os dados estão disponibilizados de maneira que otimize o trabalho daqueles que precisarem ingerir os dados do *lake*. Seguem os componentes:

- ✓ Amazon **Redshift**: Serviço de data warehouse relacional que usa armazenamento colunar para otimizar as cargas de trabalho analíticas. Ideal para ser usado em bancos de dados gigantes que necessitam de consultas complexas e que resultam grande número de linhas, além de não precisar investir tanto tempo para manter uma infraestrutura elaborada;
- ✓ Amazon **Dynamo DB**: Banco de Dados NoSQL com abordagem de banco chave-valor e banco de dados orientado a documentos. Indicado para trabalho de processamento de transação online e quando o usuário final implementará a lógica na camada de aplicação.
- ✓ Amazon **S3**: Seu funcionamento nessa camada serve para receber os dados em formatos de arquivos, porém, de forma já tratada;
- ✓Amazon **Athena**: Serviço de consulta interativa usando SQL. Não necessita de um servidor e pode ser utilizado para escrever query SQL em cima de dados presentes no S3. No cenário proposto indiquei ele também com conector do Dynamo DB, centralizando a forma de como os dados podem ser pesquisados (com linguagem SQL).

A camada de **Access & Security** é a camada onde estão providos serviços de controle de acesso e configurações de segurança para do *Data Lake*. Seguem componentes:

- ✓ Amazon **IAM**: Serviço para controle de acesso aos dados. Pode ser controlada todas as permissões de acesso por usuário/grupo. Para aumentar a segurança e ajudar a proteger os recursos da AWS, sugeri a utilização da autenticação multi-fator (MFA);
- ✓ Amazon **Cloud Watch**: Serviço para observação e monitoramento de recursos para obter a integridade operacional de forma unificada. Pode ser utilizado para detectar comportamentos estranhos, disparar alarmes e salvar logs do ambiente

operacional;

- ✓ Amazon **KMS**: Criação e gerenciamento de chaves criptografadas e controle de seu uso em diversos componentes da AWS. Com ele é possível rastrear toda utilização de cada chave.

A camada **Consume** é uma exemplificação de como podem ser usados os dados ingeridos, transformados e disponibilizados no *Data Lake*. Inúmeras ferramentas, plataformas ou casos de uso poderiam ser citadas aqui, mas citarei 2 que possuem contextos diferentes mas exemplificam o uso dos dados:

- ✓ **Jupyter Notebook**: Amplamente utilizado pela comunidade de Cientista de Dados, ele provê que códigos sejam escritos e executados em formato de notebook. Com ele, os cientistas e analistas de dados podem usar os dados do *Data Lake* para fazer análises exploratórias, criar modelos de Machine Learning, etc;
- ✓ **Metabase**: Ferramenta poderosa para criar, compartilhar, administrar e embutir dashboards. Ela contém conectores nativos para diversas fontes de Dados e está em crescente utilização pela comunidade. Poderia ser citadas outras ferramentas de Visualização de Dados como Power Bi, Tableau, Qlik Sense,Looker, etc.

Toda essa arquitetura foi criada pensando em durabilidade, escalabilidade, agilidade, disponibilidade, performance, segurança e otimização de custos. Sendo que o principal componente da arquitetura é o **negócio**. Ele quem de fato vai reger os requisitos e o tamanho necessário da arquitetura do *Data Lake*.

Um desafio é manter todo esse ecossistema atualizado e com boa performance. É preciso ter uma forte governança, incluindo rigorosa classificação dos dados para que o lago de dados não acabe virando um pântano de dados.

OBJETIVO 2 DO DESAFIO

Como segunda parte do desafio, foi pedido para fazer algumas tratativas em cima de um arquivo csv, e dentre umas dessa tratativas pede-se para converter o arquivo par um formato colunar de alta performance de leitura e justificar.

O tipo de arquivo escolhido foi o **parquet**. Pois ele é uma eficiente forma de armazenar dados para análises. Foi desenhado para interoperabilidade, ser eficiente em tamanho em disco, rápidas consultas.