**Ame Digital - Teste Data Engineer**

O desafio a seguir tem como objetivo avaliar seus conhecimentos e experiências com dados e habilidade de resolver problemas. Ao solucioná-lo, você nos mostrará:

Sua capacidade de extrair dados de uma fonte, processá-los e transformá-los em informações.

* Seu entendimento sobre tecnologias de Big Data.
* Seu conhecimento em SQL.

**Desafio**

A seguir você encontrará links para dois conjuntos de dados que contém requisições HTTP para os servidores da NASA - Kennedy Space Center para os períodos de Julho e Agosto de 1995. Kennedy Space Center - Julho: Nasa Kennedy Server - July Kennedy Space Center - Agosto : Nasa Kennedy Sever - August Os logs contém as seguintes informações:

* Host que está realizando a requisição
* Timestamp do momento em que a requisição aconteceu
* Requisição
* Código de retorno da chamada HTTP
* Total de bytes retornados

**Pre-requisitos**

Foram utilizadas as seguintes ferramentas para desenvolver a solução técnica:

* [Anaconda3](https://www.anaconda.com/distribution/) - Main Environment
* [MicrosoftVsCode](https://code.visualstudio.com/) - Development IDE
* [JupyterNotebook](https://jupyter.org/) - Development IDE

Linguagens:

* [Python3.7](https://www.python.org/)
* [ApacheSpark](https://spark.apache.org/)
* [SparkSQL](https://spark.apache.org/sql/)

**Instalação e execução**

Para obter o resultado do teste técnico, basta execute as linhas do caderno

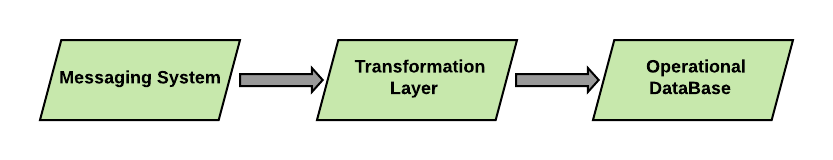
ame\_resolucao.ipynb

### BIG DATA - TEÓRICO - Respostas

1. **Considerando que a Ame possui diferentes aplicações, resultando em diferentes fontes de dados (como bancos relacionais e noSQL), de que maneira você construiria uma arquitetura para realizar a ingestão desses dados em uma plataforma de Big Data?**

**Descreva as tecnologias que você escolheria para realizar a ingestão, bem como o fluxo de dados entre elas (lembrando que o objetivo é disponibilizar as informações o mais próximo de "real-time" possível).**

Uma pipeline de dados normalmente recebe dados de diversas fontes externas, sendo essas, banco de dados de aplicações, Data Warehouses, arquivos manuais (xlsx, csv, txt, etc), bem como também dados em tempo real gerado por eventos (Dados de localização GPS, páginas web, sensores, celulares, por exemplo). Pensando em um pipeline focado em real-time para as soluções da Ame Digital, precisamos acomodar essa infraestrutura em três principais camadas: 1. camada in-memory storage para ingestão rápida; 2. Uma arquitetura com escalabilidade horizontal; 3. Que os dados sejam consultáveis e que permita exploração interativa, em tempo real. Pensando nesse contexto, teríamos uma pipeline na seguinte estrutura:



Baseado na estrutura da imagem acima, utilizaria **Apache** **Kafka** como o sistema de mensagens, que orquestraria o recebimento de dados entre os sistemas externos (que chamamos de produtores) e disponibilizaria os dados para os sistemas internos (os consumidores). Esse processo é escalável, uma vez que o Kafka possui características de sistema distribuído.

Para a fase de transformação de dados, Spark pode ser utilizado como um captador de dados do Kafka, para tratamento de um dataset, enriquecimento dos dados e/ou persistência dos dados captados em tempo real para uma base de dados

E para análise dos dados em real-time e histórico, os dados podem ser trabalhados além do ambiente de streaming e de transformação, podem então persisti-los em base de dados.

Desta maneira, desenhei um exemplo de pipeline que capta dados de diversas fontes (sejam essas estruturadas e/ou não estruturadas), utilizando de tecnologias que possam gerar tópicos de captação do dado, armazenamento distribuído, tratamento e enriquecimento do dado, persistência de dados in-memory e por fim, a disponibilização desse dado para cientistas e para analistas. Além disso, adicionei um layer que permite o controle dos processos do workflow, armazenando logs de execução e tratamento e disponibilizando dashboard analítico para auxiliar engenheiros na manutenção do ambiente.

A close up of a map

Description automatically generated

1. **Ao utilizar ferramentas de processamento distribuído como Spark ou Hive, é muito comum enfrentar problemas relacionados à má distribuição de dados entre as máquinas do cluster, diminuindo drasticamente a performance das aplicações, principalmente em operações relacionadas a agregação ou join. Utilizando seus conhecimentos e experiências, descreva uma possível solução para o problema em questão.**

Data skew é distribuição do processamento é assimétrico de dados nos clusters que acarreta um processamento mais lento, mais custo e com necessidade de maior recurso.

1. **O dia a dia de um engenheiro de dados, dentre outras tarefas, é disponibilizar as informações em alta performance (próximos a real-time) para Analistas e Cientista de Dados de modo a possibilitar à análise e criação de modelos estatísticos. De que modo e quais tecnologias você usaria para disponibilizar os dados para estas pessoas.**

Dentro do exemplo dado no exercício 1, todo o dado recebido na pipeline ficará disponível em ambiente distribuído (como HDFS) e poderá ser consumindo por jupyter notebooks em um sandbox para exploração de dados por cientistas e analistas. Além disso, todo o dado histórico e analítico pode ser armazenado em um data warehouse (como o Hive DB ou quando falamos de soluções em nuvem como AWS Redshift ou um banco de dados Oracle Data Warehouse), para consumo de dashboards, como também para elaboração de Business reports. Além disso, é possível que realizar análises da própria pipeline dentro da estrutura de documentos do Elasticsearch, que permite também ser consumido por Dashboard no Kibana, em tempo real.

1. **Por fim, tendo em mente o crescimento exponencial dos dados e utilização massiva da plataforma de Big Data, quais métodos de organização e/ou governança você implementaria para manter o ambiente sustentável?**

Quando pensando em governança de dados em um ambiente Big data, considero que temos pilares que devem ser colocadas em pauta, como segurança e custo efetivo.

Quando tratamento de segurança, podemos considerar o controle de acessos de usuários dentro clusters como hadoop utilizando do Apache Ambari, pois viabiliza um controle total de todo o processo do ambiente distribuído, bem como controle das roles de usuário. Se falamos de ambiente Cloud (como, por exemplo, AWS), podemos considerar o controle de toda a pipeline de dados por meio do IAM, onde podemos adicionar policies para as roles, garantindo a segurança de acesso.

Sobre custo efetivo e identificar qual o real uso de uma infraestrutura de dados está sendo consumido baseado no seu custo financeiro. Desta maneira, penso que uma equipe Devops precisa estar a par de cada solução cloud (seja clusters ou serviços) ou física (servidores) para garantir que não exista infraestrutura sendo consumida indevidamente.