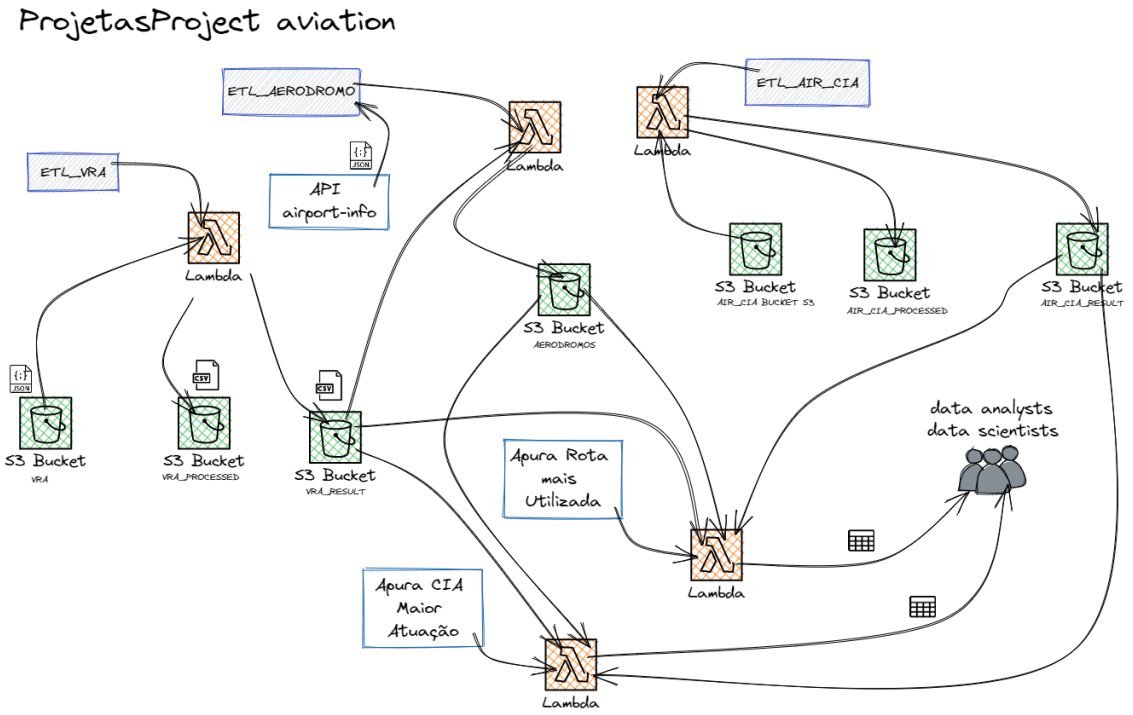
ProjetasProject

https://github.com/jairo2016/ProjetasProject_aviation



Descrever qual estratégia você usaria para ingerir estes dados de forma incremental caso precise capturar esses dados a cada mes?

Temos 3 tipos de arquivos de dados crus ou brutos que são:

- Arquivos VRA.json (voo regular ativo) que fornece dados dos registros de voo efetuados pelas empresas aéreas fornecidos pela empresa.
- Arquivos AIR_CIA.csv (cadastro das empresas aéreas) que fornece dados de cadastro das companhias aéreas fornecidos pela empresa.
- AERODROMOS.json (cadastro dos aeródromos) que fornece dados de cadastro dos aeroportos oferecido da web pela API do site airpor-info em formato json.

Temos 5 módulos principais de código python:

- ETL_VRA.py que lê todos os arquivos json do diretório ou s3 bucket VRA junta os dados em um só dataframe, transforma os dados, grava um arquivo VRA_yyyymmdd_hhmmss.csv, no diretório ou s3 bucket VRA_RESULT, com os dados transformados desse lote de arquivos json e por fim move os arquivos crus desse lote para o diretório ou s3 bucket VRA_PROCESSED onde ficarão salvos por um período de tempo conforme a necessidade de se fazer um reprocessamento.

- ETL_AIR_CIA.py que lê todos os arquivos .csv do diretório ou s3 bucket AIR_CIA junta os dados em um só dataframe, transforma os dados, grava um arquivo AIR_CIA_yyyymmdd_hhmmss.csv, no diretório ou s3 bucket AIR_CIA_RESULT, com os dados transformados desse lote de arquivos csv e por fim move os arquivos crus desse

lote para o diretório ou s3 bucket AIR_CIA_PROCESSED onde ficarão salvos por um período de tempo conforme a necessidade de se fazer um reprocessamento.

- ETL_AERODROMO.py que lê os arquivos csv do diretório ou s3 bucket VRA_RESULT, procura os códigos icao distintos dos registros de voo e através da API do site airpor-info consome os dados de cada código icao encontrado, junta esses dados em um único dataframe e grava um arquivo AERODROMOS_yyyymmdd_hhmmss.csv, no diretório ou s3 bucket AERODROMOS.

- Apura_rota_mais_utilizada.py lê os arquivos csv dos diretórios ou s3 buckets VRA_RESULT, AIR_CIA_RESULT e AERODROMOS e gera uma view:
  Rota (origem-destino) mais utilizada para cada companhia aérea:
    Razão social da companhia aérea
    Nome Aeroporto de Origem
    ICAO do aeroporto de origem
    Estado/UF do aeroporto de origem
    Nome do Aeroporto de Destino
    ICAO do Aeroporto de destino
    Estado/UF do aeroporto de destino

- Apura_companhia_maior_atuacao.py lê os arquivos csv dos diretórios ou s3 buckets VRA_RESULT, AIR_CIA_RESULT e AERODROMOS e gera uma view:
  Companhia aérea com maior atuação no ano, para cada aeroporto:
    Nome do Aeroporto
    ICAO do Aeroporto
    Razão social da Companhia Aérea
    Quantidade de Rotas à partir daquele aeroporto
    Quantidade de Rotas com destino àquele aeroporto
    Quantidade total de pousos e decolagens naquele aeroporto

Ingestão incremental dos dados:

- Os dados dispostos inicialmente em diretórios/s3buckets de arquivos crus são processados gerando arquivos que são armazenados e disponibilizados em diretórios/s3buckets de resultados, sendo que, posteriormente, os dados crus são movidos para os diretórios/s3buckets de arquivos que já foram processados, ficando disponíveis para receber novos e atualizados arquivos nos diretórios de dados crus para serem processados novamente e mensalmente.
- Os dados relativos aos aeródromos são baixados através da API do site airpor-info sendo esse processamento dependente dos dados que estão no VRA_RESULT que é alimentado pelo processamento do ETL_VRA.py.
- Ao mesmo tempo em que o processo de ingestão esteja sendo realizado, novos arquivos de dados crus podem ser gravados nos diretórios/s3buckets VRA E AIR_CIA sendo necessário repetir todo o procedimento novamente para que esses dados sejam contemplados nos resultados de apuração.

Justifique em cada etapa sobre a escalabilidade da tecnologia utilizada.

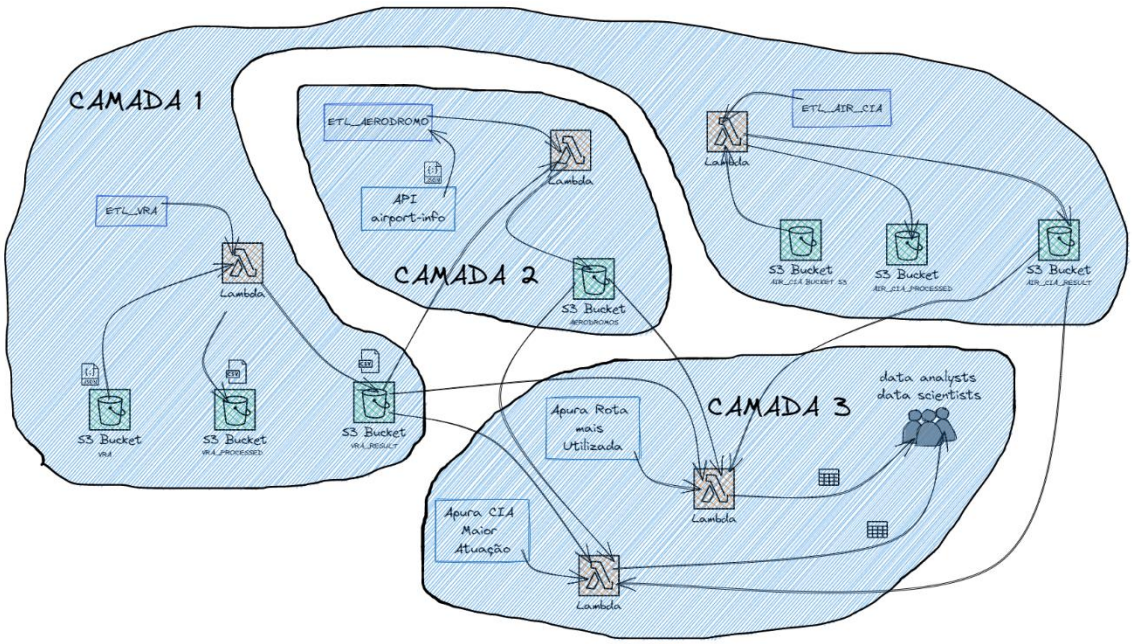
O ProjetasProject foi desenvolvido usando ditetórios locais para melhor demonstração da execução, podendo, esses diretórios, serem substituidos pelos s3 buckets da AWS disponibilizados em cloud e o seu acesso feito através das funções do boto3 python da AWS. Dessa forma vamos justificar o uso de cada serviço:

- AWS S3 Bucket: O local onde os arquivos ficarão armazenados tem como peso principal a continuidade do armazenamento com boa performance de acesso que oferece toda a manutenção da estrutura, com armazenamento distribuído e alta escalabilidade, deixando os engenheiros de dados livres para se preocuparem somente com o planejamento, execução e operação do ELT.

- AWS Lambda: Os códigos serão processados utilizando-se o serviço da aws Lambda, que é orientada a eventos com tecnologia Serverless Computing, não é necessário definir um servidor para executar uma aplicação ficando transparente para nosso processamento, sendo mais uma preocupação para a equipe da AWS Amazon manter o serviço funcionando com escalabilidade.

- Spark + pyspark: O serviço oferecido pelo spark juntamente com sua biblioteca pyspark para o python permite que um grande volume de dados possa seja processado de forma distribuída em vários clusters, entregando maior capacidade de processamento e aceitando um maior volume de dados com maior performance.

Justifique as camadas utilizadas durante o processo de ingestão até a disponibilização dos dados.



Sequência de processamento do código:

- CAMADA1: Os módulos de ingestão ETL_VRA.py e ETL_AIR_CIA.py são independentes, podendo ser executados no mesmo momento sem qualquer conflito, são responsáveis em suprir os dados para a Camada2 e para a Camada3.

- CAMADA2: O módulo de ingestão ETL_AERODROMO.py tem de ser processado após o processamento do módulo ETL_VRA.py, o qual vai ingerir os novos voos com possíveis novos códigos icao de aeródromos, ou seja, deve ser executado após a execução da Camada1.

- CAMADA3: Os módulos de apuração de dados Apura_rota_mais_utilizada.py e Apura_companhia_maior_atuacao.py podem ser processados a qualquer momento e no mesmo instante sem qualquer conflito, porém é necessário que os dados já tenham sido ingeridos na Camada1 e na Camada2, portanto faz-se necessário sua execução após a execução da primeira e segunda Camada.