

**Desafio - Arquitetura de Dados**

Thays Maia de Sá

Objetivos

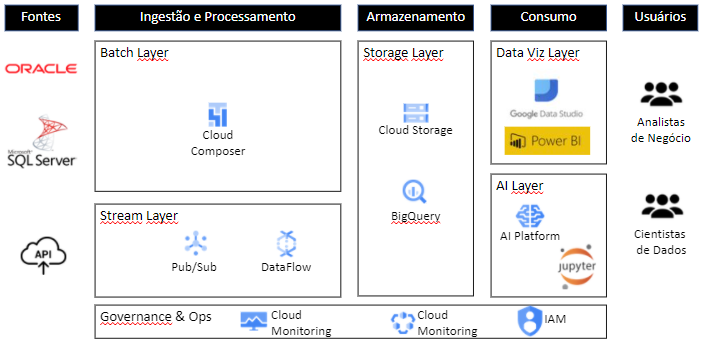
* Desenhar uma arquitetura de referência para ingestão de dados batch e stream
* Definir a acomodação dos dados no Data Lake
* Definir a camada de exploração e visualização dos dados
* Definir ferramentas para geração de insights através dos dados
* Codificar protótipo para a IaC utilizando Terraform

Premissas

* A arquitetura de referência foi desenhada levando em consideração os serviços providos pela nuvem da Google (GCP).
* As definições referentes à rede dependem de definições de Infraestrutura e Segurança da empresa assim como as liberações de acesso aos dados dos sistemas fonte.
* Não foram consideradas volumetrias dos sistemas de origem na arquitetura proposta e isto pode influenciar nos custos envolvidos, por se tratar de um ambiente na nuvem.
* É recomendada a definição das necessidades de negócio envolvidas de forma a evitar a ingestão desnecessária de dados no Data Lake. ("Se você não sabe onde quer ir, qualquer caminho serve." - Alice no País das Maravilhas)

Arquitetura Proposta

A proposta abaixo traz uma arquitetura majoritariamente baseada em plataformas e serviços gerenciados pela nuvem da Google.



É uma opção que entrega alta disponibilidade e uma facilidade na manutenção e sustentação do ambiente, em contrapartida pode ser uma opção mais cara em termos de infraestrutura.

A proposta de arquitetura foi baseada na arquitetura Lambda. Pode ser usada também a arquitetura Kappa porém ela traz uma complexidade maior na execução o que pode ser um problema na sustentação.

*Ingestão de dados Oracle ou SQL Server*

A arquitetura proposta é que sejam criadas as dags no Composer que façam a conexão com os Bandos de Dados e entregue os dados diretamente no BigQuery, por eles já estarem estruturados. Desta forma podemos programar batchs ou micro-batchs que normalmente são suficientes para o processamento destes tipos de fontes de dados.

Uma outra opção seria termos DataProcs efêmeros executando sqoops para trazer o dado diretamente para o BigQuery. Nesta opção trazemos uma maior complexidade para a sustentação e manutenção do pipeline porém, dependendo da volumetria, pode trazer uma redução no custo.

Para uma abordagem em streeming destas fontes, podemos utilizar uma ferramenta CDC conectando diretamente o Banco de Dados com o BigQuery, porém é necessário identificar uma ferramenta com um custo-benefício aceitável, todas as que eu testei até hoje trouxeram um alto custo dentro do GCP por conta do BigQuery.

Para o Banco de Dados Oracle, por exemplo, podemos também utilizar o Golden Gate e fazer a ingestão utilizando os logs do Banco de Dados, porém já identifiquei algumas perdas de dados utilizando esta abordagem, logo não sei se seria muito bem recomendado, é necessário fazer algumas experimentações.

*Ingestão de APIs*

O ideal é que o sistema de origem já entregue os eventos diretamente no Pub/Sub, desta forma eles poderão ser processados pelo dataflow e pousados no Cloud Storage preferencialmente em arquivos parquet (snappy compressed) pois este formato é facilmente interpretado pelo BigQuery, logo podemos criar tabelas externas a partir destes arquivos.

Caso o sistema de origem não consiga entregar o dado diretamente no Pub/Sub a melhor recomendação é que se tenha um compute engine com um script, pode ser em python, fazendo o pullying na API e entregando os dados nos tópicos do Pub/Sub. Desta forma o pipeline já explicado anteriormente poderá ser seguido.

Para o tratamento dos dados e geração de indicadores ou agregações, a indicação é que seja utilizado o Composer em uma abordagem batch. Uma outra opção seria o tratamento através de views no BigQuery (Arquitetura Kappa), porém isto envolve um maior custo em se tratando de processamento via BigQuery. Envolve também uma complexidade maior na manutenção e sustentação.

Durante a confecção deste documento encontrei uma ferramenta (Bokeh - <https://github.com/bokeh/bokeh>) que entrega dashboards referente a dados streaming a partir de dados do Pub/Sub. Não conheço, porém achei que pode ser interessante e possivelmente avaliada/explorada. Fonte: <https://medium.com/datareply/realtime-streaming-data-pipeline-using-google-cloud-platform-and-bokeh-9dd0cfae647a>

Camadas de Dados do Data Lake

A figura abaixo mostra como é recomendada a acomodação dos dados e a divisão de camadas. Mostra também como é indicado o acesso aos dados pelos usuários finais.



*Acesso pelos Analistas de Negócio*

Alguns analistas de negócio possuem um perfil mais técnico e conseguiriam fazer queries diretamente no bigquery. Neste caso, é recomendada a criação de um projeto específico dentro do GCP para eles (Projeto Sandbox), desta forma isolamos os workloads dos usuários dos pipelines de ingestão e processamento. Uma boa prática é a criação de um projeto por área de negócio, desta forma fica mais fácil a segregação de custos, se isto for uma questão para a empresa.

Para os analistas de negócio sem perfil técnico porém com uma expertise exploratória, o ideal é que eles já tenham uma ferramenta Self-Service para que eles montem facilmente os dashboards que eles necessitarem. Na arquitetura esta ferramenta foi representada com o Data Studio e o PowerBI.

Existe também aquele perfil de usuários que acessam o dashboard já pronto. Para estes o PowerBI e o DataStudio também são recomendados, porém a área técnica que construirá o dashboard/relatório. Uma boa prática é que a área de negócio possua pelo menos um analista com perfil técnico de forma que as alterações em requisitos de negócio sejam facilmente implementadas nos Dashboards e Relatórios.

*Acesso pelos Cientistas de Dados*

Este perfil de usuário pode acessar tanto os dados da camada RAW quanto os dados da camada Trusted e o objetivo normalmente é o data discovery e a geração de insights. Pela arquitetura proposta eles podem utilizar tanto a AI Platform quando o Jupyter Notebook para a criação de scripts, porém a produtização deveria ser feita através de scripts, preferencialmente em python, com o apoio de arquitetos e desenvolvedores, incorporando aos pipelines de processamento e enriquecimento dos dados.

Protótipo IaC - Terraform

A codificação do protótipo terraform que foi disponibilizada no github é uma simples indicação de como podemos prover os serviços do GCP via código de terraform. Os exemplos foram retirados do próprio site da Terraform (<https://registry.terraform.io/providers/hashicorp/google/latest/docs>) incluindo algumas adaptações e customizações, porém eles precisam ser adaptados, executados e testados na nuvem da empresa.