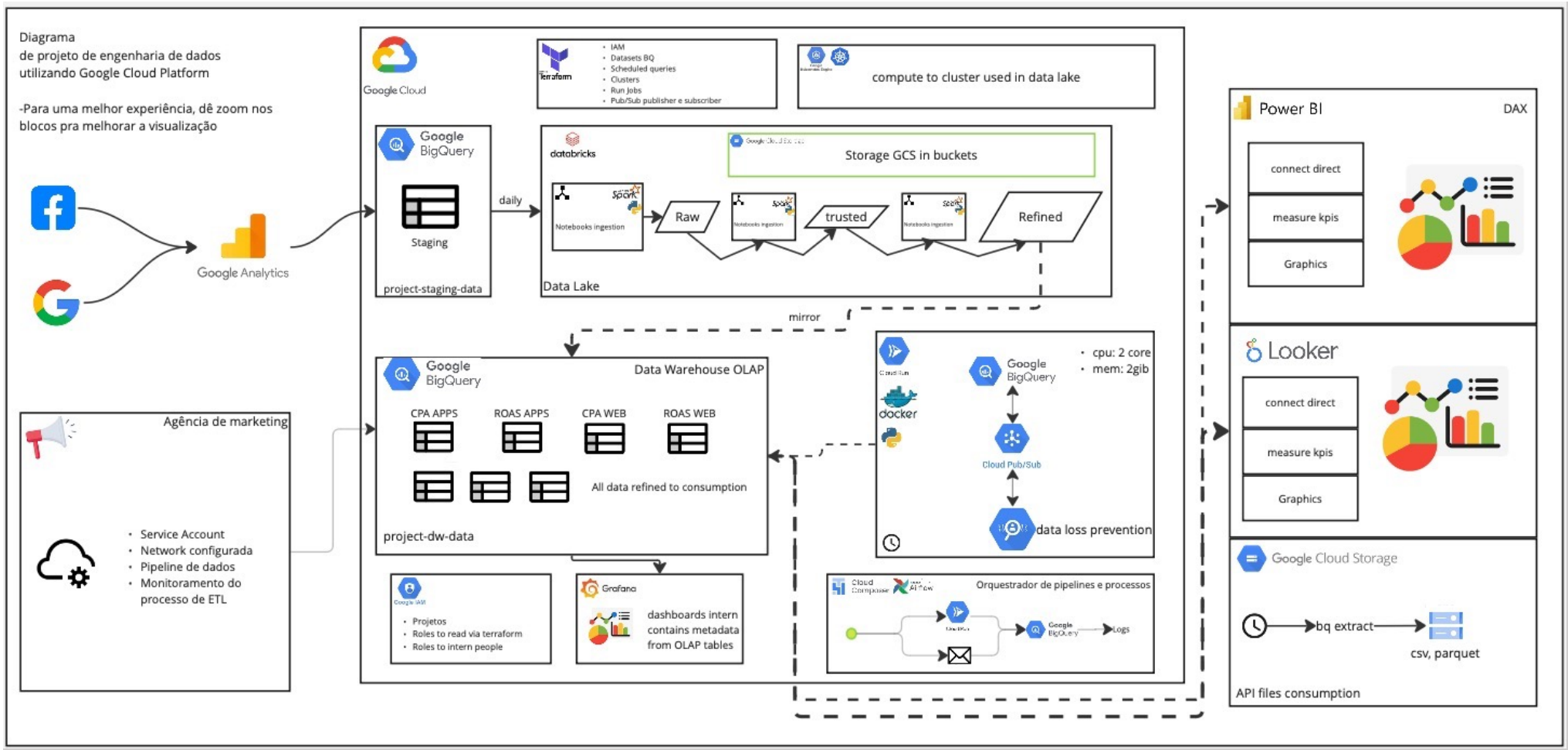
Ramon Barbosa

Projeto teste técnico para dp6



Google cloud platform

Essa foi minha escolha de cloud devido alguns serviços de dados muito úteis como o BigQuery, que é um database colunar serverless que executa com rapidez consultas, ambiente ideal para um banco usado como OLAP, para tabelas analíticas

Tecnologias selecionadas para o projeto

BigQuery

Como um bom serviço cloud serveless, o BigQuery com boa administração de suas reservas de slots, tem sido amplamente utilizado pelas companhias para otimizar e reduzir custos com infraestrutura, já que o BigQuery conta com tecnologias por detrás dos panos que são bem conhecidas pelo Google, como o Dremel, jupiter network e Colossus, para processamento de dados, ampla transferência de dados numa larga rede e armazenamento barato de dados, respectivamente.

https://cloud.google.com/bigquery/docs

Tecnologias selecionadas para o projeto

terraform

Software de infrastructure as a code (iac), escolhido por ser a tecnologia do ramo mais amplamente utilizado e explorado, logo permite ter uma sustentação mais engajada. Google Cloud possui bastante documentação e exemplos no site oficial da mantenedora HashiCorp.
https://registry.terraform.io/providers/hashicorp/google/latest/docs

Sistema de Data lake

https://cloud.google.com/learn/what-is-a-data-lake?hl=pt-BR

Decidi criar um datalake para essa empresa com o intuito de manter todos os dados de todos os setores da companhia em um só local. Utilizando as camadas básicas. Camada Raw, que recebe os dados brutos com validação entre o dado recebido e a fonte original. Camada Trusted que recebe limpezas, transformações de acordo com a necessidade dos usuários desses dados. Camada Refined, muito próxima do conceito do Warehouse OLAP, com tabelas já consolidadas, agregadas e com uma visão voltada a alimentar relatórios e dashboards gerenciais e operacionais.

Tecnologias selecionadas para o projeto

Databricks

Plataforma voltada para gerenciamento de datalakes como foco principal, os famosos lakehouses. Escolha dessa plataforma produtizada via repositórios pois tem preço bem menor comparado ao produtizado via plataforma diretamente. Excelente para facilitar a criação de pipelines de dados e orquestrar tais cargas de trabalho dentro da própria plataforma.

https://www.databricks.com/br/product/data-intelligence-platform

Apache Spark

Software de processamento de dados que foi antecedido pelo MapReduce do ecossistema Hadoop. Comparando ambas tecnologias o core do Spark já em suas primeiras versões eram mais ágeis que p processamento do processamento do Hadoop, por isso para empresas que sustentam arquiteturas com o Hadoop, já trazem consigo o Spark. Este usa o principio de dividir para conquistar, utiliza como infra, um grupo de servidores, um sendo o mestre (Driver) e os demais, sendo slaves (Workers). Esses Workers (nós do cluster), se dividem para mapear, agrupar informações dos dados e no fim, juntam os dados pra trazer o resultado(de forma bem resumida). A escolha desta tecnologia tem a ver sobre o sistema de data lake que foi determinado como centralizador e armazenador de todos os dados da companhia.

https://spark.apache.org/docs/latest/

Google cloud storage

Este serviço de armazenamento de arquivos gerais, foi escolhido para compor a infra de estocagem de dados do data lake, entre outras necessidades de geração de arquivos. Custo de armazenagem Standart(pronto pra uso), e a facilidade de leitura e escrita, substitui de forma eficiente o HDFS do Hadoop pra armazenagem de dados. O grande beneficio da utilização do GCS em um sistema de dados, é poder separar processamento e o armazenamento. Um ponto de extrema importância desse serviço é a largura de rede muito ampla pra integrar entre outros serviços do GCP, sendo usado como etapa intermediária pra escrita no BigQuery via API(ou com dados provenientes de infras diferentes).

https://cloud.google.com/storage/docs?hl=pt-br

Google Kubernetes engine

https://cloud.google.com/kubernetes-engine/?hl=pt-BR

https://kubernetes.io/pt-br/docs/concepts/

Conhecido como GKE, utiliza-se a tecnologia de conteinerização pra poder poder lançar recursos computacionais. Explicando melhor, ao invés de apenas lançar instancias de vm`s como infra de clusters e produtos quaisquer em nuvem, lançamos vm`s onde suas aplicações serão executadas em contêineres Docker e melhor, podendo lançar diversas aplicações em pods para serem administradas pelo k8s(kubernetes). Logo, pode-se manter esses cluster via GKE pra toda empresa processar seus dados, inclusive de forma organizada, ser a infra dos clusters que sustentam o data lake vinculados devidamente configurados com o Apache Spark pra utilização no sistema de data lake no databricks.

Cloud Run Jobs

https://cloud.google.com/run/docs/create-jobs

Um recurso serverless, muito poderoso pra ser utilizado como infra de processamento computacional. O Cloud Run Jobs executa imagens Docker sem nenhuma pre configuração (sem configurar k8s, mas a imagem você precisa gerar e dar push no serviço de artifact da GCP). Será utilizado para executar os códigos de Data Loss Prevention (via API em SDK), transmitindo as informações na GCP pelo pub/sub (serviço de mensageria em tempo real ou próximo do real). Na prática, o job de clud run será executado sempre que bases de dados sejam inseridos num dataset do BigQuery (no nosso data Warehouse com as tabelas que as agências externas poderão ter acesso de leitura).

Docker

https://docs.docker.com

Tecnologia que revoluciona o mercado, mudando a maneira de virtualizar aplicações em servidores. Num mesmo servidor podemos ter várias aplicações independentes com o Docker. Neste projeto o Docker é usado pra gerar imagens Docker que contém aplicações que verificam de forma eficaz dados sensíveis e restritos, e máscara esses dados (ou criptografa) na fonte de dados em questão (BigQuery).

Pub/Sub

Serviço de mensageria do GCP, que utilizaremos pra comunicar o BigQuery com o Data Loss Prevention (DLP), basicamente um serviço serverless que funciona muito bem e que facilita a comunicação entre serviços para este projeto

https://cloud.google.com/pubsub/docs

Data Loss Prevention

O Data Loss Prevention é um recurso de segurança de dados, que pode verificar se em um conjunto de dados existe dados sensíveis que não deveriam estar expostos em um determinado contexto. Para este projeto utilizaremos o serviço para verificar dados que serão acessados por empresas externas com acesso mínimo de leitura e que serão anonimizados, mascarados ou criptografados em tabela.

https://cloud.google.com/sensitive-data-protection/docs

Cloud Composer

https://cloud.google.com/composer?hl=en

Basicamente uma versão fully managed (totalmente gerenciada pelo Google) onde não precisamos nos preocupar em sustentar o ambiente do Apache Airflow. Utilizaremos o serviço para orquestrar nossas chamadas Cloud Run Jobs e demais tarefas de ETL AdHoc. Toda a empresa pode ter acesso a essa ferramenta, menos acessos externos, pois cada instancia do Composer possui apenas uma Service Account, que geralmente precisa de muitos privilégios em nível de projetos, com isso é arriscado dar herança de execução da Service Account do composer para pessoas que não deveriam ter muitos privilégios sobre alguns recursos cloud.

Apache Airflow

Será utilizado via serviço cloud do Composer, que basicamente é uma instância do AirFlow gerenciada e sustentada pelo Google.

https://airflow.apache.org/docs/

Grafana

Grafana disponibiliza recursos de dashboards que se integram a muitos produtos de cloud atuais. Logo utilizaremos para monitoramento e alertas de metadados do data Warehouse e data lake da empresa para observar e garantir a qualidade dos dados e o funcionamento do serviço de mascaramento de dados sensíveis.

https://grafana.com/docs/grafana/latest/introduction/

Google IAM

Serviço altamente relevante nas cloud em geral, para administração de acessos a contas de serviço, usuários internos e externos e criação de regras e políticas de acesso as API`s dos recursos em nuvem. Utilizaremos como ponto fundamental do projeto, já que esse recurso bem organizado pode evitar exposições e roubos de informações da companhia.

https://cloud.google.com/iam/docs/overview?hl=pt-br/

Python/ Google SDK

Utilizaremos o Python para criar nossas aplicações que acessam recursos de nuvem via SDK para manipular e gerir os processos em cloud. Geralmente utilizaremos as libs do Google para criar as cargas de trabalho de DLP e também dentro do data lake para criar os pipelines de dados e manipulações em geral, como aliada ao PySpark, que é uma api pra utilizar o Spark em codificação.

https://cloud.google.com/python/docs/reference

Tecnologias selecionadas para o projeto

PowerBI

Software de criação de dashboards e manipulação de dados da Microsoft, que utilizaremos para criar os dashboards gerenciais e operacionais diários e mais relevantes para a própria companhia. Lembrando que o grafana será utilizado apenas para metadados e monitoramento, pois é opensource, logo gratuito. O pbi será utilizado pra dashboards de negócios e seus produtos. Vamos evitar a utilização em demasia da linguagem DAX, que pode gerar longas demoras de processamento dentro dos servidores do PowerBI.

https://learn.microsoft.com/pt-br/power-bi/

Google Looker

Looker é a ferramenta built-in do GCP pra geração de dashboards. Neste caso utilizaremos esse recurso na sua versão gratuita(que funciona bem) para os times que preferirem o Looker ao invés do PowerBI, o que pode ser um item de incentivo dentro da companhia para os times que preferirem este em detrimento do PowerBI. Porém sabemos que o pbi acaba sendo software mais amplamente usado e conhecido do mercado de painéis.

https://cloud.google.com/looker/docs/intro?hl=pt-br

Scheduled queries

Não poderia deixar de mencionar o recurso de agendamentos de consultas do BigQuery que juntamente com o Data transfer (transfere dados de plataformas distintas e entre regiões do Google também), ingere dados entre tabelas e é um aliado na composição do Data Warehouse em suas consolidações de dados e mais.

https://cloud.google.com/bigquery/docs/scheduling-queries

Resumo para defesa da arquitetura

Defesa da Arquitetura do projeto

Esta arquitetura utilizada em tecnologia de cloud, na Google Cloud Platform, voltada a sistemas e gerenciamento de dados.

Coletamos os dados provenientes do Google Analytics e ingerimos estes `as is` num projeto em nuvem apenas com um dataset no BigQuery.
Estamos utilizando o BigQuery nesse momento como uma etapa intermediaria para trazer os dados externos da plataforma do GA.
Em seguida através de um notebook no sistema de data lake, processamos a tabela com dados brutos do BigQuery e realizamos algumas validações desses dados, como se existe dados sensíveis, se há falhas na tipagem ou no schema. Então inserimos esses dados na camada RAW do data lake, que é escrito no formato parquet.snappy em serviço de Google Storage.
Resumidamente na camada trusted e na refined são elaborados pipelines padronizados, onde são garantidas tanto qualidade quanto segurança dos dados, além claro de consolidar e gerar tabelas que atendam toda a empresa.

Defesa da Arquitetura do projeto

As tabelas ou bases de dados da camada refined do data lake são espelhadas, ou seja, copiadas para dentro do BigQuery (não como dados federados em tabelas externas, mas inserção mesmo). Agora estamos usando o BigQuery em um projeto onde as agências de marketing parceiras da companhia terão acesso limitado de leitura nas tabelas. Esse projeto com BigQuery será é um data Warehouse, apenas contendo tabelas consolidadas e suficientes para o consumo das empresas parceiras e aos devidos setores da empresa que possam se beneficiar destas.
Os acessos das empresas externas são limitados, monitorados via log do GCP monitoring e também via logs do BigQuery para maior segurança de quem está acessando e tuso isso a fim de auditorias futuras também.

Defesa da Arquitetura do projeto

Criamos Jobs de Cloud Run para acessar o serviço de DLP como uma camada extra de proteção a dados sensíveis, visto que no data lake os dados são tratados e verificados entre as camadas, Gerenciamos instancias no GKE pra prover capacidade computacional para o data lake processar seus pipelines, orquestramos no Composer com instancias de apache airflow, monitoramos metadados dos recursos e das tabelas no grafana, integramos as tabelas consolidadas para o PowerBI, Looker e geramos relatórios como arquivos para casos necessários, além de ter um organizado serviço de IAM para controlar os acessos e privilégios internos e externos.

Defesa da Arquitetura do projeto

Por fim, uma arquitetura orientada a sistemas de dados com escalabilidade, segurança, garantia de qualidade nos produtos de dados e alta disponibilidade.

Requisitos solicitados ao cliente

Inicialmente, seria interessante entender se a agência parceira se preocupa com a segurança da sua própria infraestrutura em cloud ou on-premises. Fazer algumas reuniões para entender o nível de segurança que a parceira se encontra.

Importante a empresa elaborar aplicações para consumir e acessar os dados da companhia para que o usuário utilizado no acesso seja uma conta de serviço monitorada nos processos de logging internos.

Listar os responsáveis que irão ter acesso aos dados no data Warehouse do projeto. Lembrando que aos dados que as parceiras irão acessar já passaram por alguns pipelines de dados no data lake e possuem camada extra de proteção a dados sensíveis com o DLP do GCP para as tabelas do data Warehouse, que se encontra em um projeto isolado no gcp, sem herança de outros projetos e totalmente monitorado.

Requisitos solicitados ao cliente

- Disponibilizar uma agenda de horários que as aplicações da agência será executada para que internamente além do monitoramento da conta de serviço associada a agência, ocorra o monitoramento do comportamento dos acessos, caso exista um acesso em horário atípico do histórico, a agência parceira explique a situação, claro que podem ocorrer erros e imprevistos, por isso a comunicação é importante e reports de acessos devem ser monitorados e apresentados nos diálogos entre as empresas.

Requisitos solicitados ao cliente

- Por fim os itens básicos, de configuração de networking, utilizar a mesma região de data centers para redução de latência e custos de transferência, elaboração de pipelines que consumam os dados acessados e um bom sistema de monitoramento de toso o processo de ETL ou do pipeline em si para possíveis auditorias.
- Lembrando que toda a parte de acesso, regras e políticas vinculadas a cada agência parceira deve ser realizado internamente pela própria companhia a fim de garantir as restrições necessárias.

Boas práticas em BigQuery

- Boa prática seria a criação de datasets, aplicação de roles IAM e criação de data tranfer e scheduled queries via terraform com aprovação do gestor responsável pelo projeto no GCP onde a API do BigQuery está sendo acessada.
- Outra boa prática da tecnologia é não utilizar o BigQuery como um centralizador de dados ou um banco de dados OLTP, ou seja, para databases de backend de aplicações que precisam escrever e fazer update constantemente de tabelas. Porque o BigQuery é um banco colunar e não de registros como os bancos de dados antigos e tradicionais. O armazenamento é por colunas.

Boas práticas em BigQuery

- Boa prática seria manter um banco de dados do tipo OLAP, ou seja, apenas para consultas e com dados analíticos, quer dizer que as tabelas vão ser de baixa granularidade comparadas as tabelas fato de sistemas de bancos de dados.
- Outra boa prática seria a utilização de tabelas aninhadas e repetições no BigQuery, basicamente, como o BQ é colunar posso juntar tabelas, fazer uma desnormalização entre tabelas do próprio BQ, de formar que não precise aplicar joins nas queries (não é apenas fazer um Join uma vez e manter a tabela e sim, como se as tabelas fossem subcolunas do tipo JSON) e repetidas pois existe esse sistema de não repetir dados para um subconjunto de mesma informação, isso é bem interessante para performance de group by`s em consultas.

Boas práticas em BigQuery

- Existem muitas boas práticas na utilização do BQ, mas vou ficar com essas apenas.