

Criação de Abstrações de Governança e Provisionamento para Plataforma de Dados

Contexto

Você está ajudando a construir uma plataforma de dados com o objetivo de simplificar o acesso e o gerenciamento de recursos de dados por usuários internos.

Seu desafio é criar **camadas de abstração** que permitam a **governança eficiente, fácil provisionamento e interações simplificadas** com a plataforma.

Descrição do teste

A TelcoNet, uma grande empresa de telecomunicações, está buscando modernizar sua infraestrutura de dados para melhorar a eficiência, a segurança e a governança. Com uma variedade de dados que vão desde registros de chamadas telefônicas e dados de clientes até métricas de rede, a empresa precisa de uma solução robusta que permita uma manipulação eficiente e segura dos dados. A TelcoNet já usa o Google Cloud Platform (GCP), mas deseja otimizar sua arquitetura para melhor suporte e gestão.

Desafios Identificados:

1. **Governança e Segurança de Dados:** Com regulamentações rigorosas como o GDPR, a empresa precisa de uma abordagem mais sistemática para o controle de acesso e a auditoria de dados.
2. **Provisionamento e Escalabilidade:** À medida que a demanda por dados cresce, a necessidade de provisionar recursos de maneira eficiente e escalável torna-se crítica.
3. **Simplificação de Operações:** A complexidade atual das operações de dados gera ineficiências e atrasos, necessitando uma plataforma mais intuitiva e fácil de usar.

Objetivos do Projeto:

- **Automatizar e Simplificar o Provisionamento de Recursos:** Automatizar o provisionamento de ambientes de BigQuery e Compute Engine, dependendo do tipo e da carga de trabalho de dados (batch, streaming, async).
- **Fortalecer a Governança e o Controle de Acesso:** Implementar políticas de segurança robustas e automatizadas, utilizando a API de gestão para configurar controles conforme os requisitos.
- **Monitoramento e Alertas de Eficiência:** Estabelecer um sistema de monitoramento que forneça insights em tempo real sobre o desempenho e a utilização dos recursos.

O teste está quebrado em 3 partes e cada uma delas tem seus entregáveis. Cada parte será avaliada individualmente mesmo que possua correlação com as outras.

Parte 1: Design de API para gestão de workloads de dados

Elabore o design de uma API que gerencie workloads de dados, suportando operações de batch, streaming e processamento assíncrono.

- A API deve permitir aos usuários especificar o tipo de workload, configurar parâmetros específicos (como volume de dados e frequência), e escolher o tipo de recurso de computação (por exemplo, CPUs ou GPUs);
- A API deve conter métodos para:
 - Criar, modificar e deletar workloads;
 - Listar workloads e seus status;

Artefatos esperados

- *Artefato 1:* O documento da API em formato OpenAPI (anteriormente conhecido como Swagger), detalhando todos os endpoints, parâmetros, e os formatos de resposta suportados;
- *Artefato 2:* Diagrama (usando ferramentas como excalidraw ou draw.io) que ilustram a arquitetura da API, mostrando como ela se integra com serviços do GCP (como Compute Engine, BigQuery, e Pub/Sub);

Parte 2: Abstração para Governança e Provisionamento

- Desenhe uma arquitetura que inclua camadas de abstração para governança, como controle de acesso, políticas de segurança e auditorias de conformidade;
- Mostre como essas camadas interagem com a API (documentada na parte 1) de gestão de workloads para simplificar a criação e gestão de recursos;

Artefatos esperados

Diagramas de Arquitetura de Governança

- *Artefato 3:* Diagramas detalhados (usando excalidraw ou draw.io) que mostrem as camadas de abstração de governança, incluindo controle de acesso, políticas de segurança e auditorias;
 - Estes diagramas devem também ilustrar como as políticas de governança interagem com a API de gerenciamento de workloads;

Documentação de Políticas

- *Artefato 4:* Documentos escritos que detalham as políticas de segurança implementadas, controle de acesso e procedimentos de auditoria.
- *Artefato 5:* Exemplos de políticas IAM para GCP que suportam a arquitetura de governança proposta.

Parte 3: Implementação e Teste

- Implemente um protótipo simples dessa API usando uma linguagem de programação de sua escolha e demonstre como ela pode ser utilizada para provisionar um ambiente para um workload batch spark para contar palavras de uma tabela (<https://spark.apache.org/examples.html>);
- O código precisa ser funcional e possível de rodar em outra máquina (visando a possibilidade de correção do teste);

Artefatos esperados

- Artefato 6: Código Fonte do Protótipo
 - Código fonte em linguagem da sua escolha que implementa a API proposta;
 - Um exemplo específico de um endpoint que provisiona um ambiente para executar um job de Spark para contar palavras em um dataset do BigQuery;
-

Entrega

- Gere um repositório com todos os **artefatos esperados** citados em cada parte do teste, inclusive os artefatos descritivos (diagramas, políticas e etc);
 - Nesse repositório, crie um README detalhado que descreve como configurar e rodar o artefato 6, incluindo requisitos, configuração do ambiente, e instruções de uso;
 - Ainda nesse repositório, inclua uma pasta com seus artefatos descritivos e nessa pasta também crie um README organizando os artefatos para facilitar a compreensão;
- Envie o link desse repositório a pessoa recrutadora do seu processo;
 - Lembre-se de permissionar corretamente o repositório para permitir a avaliação;

Entregas adicionais opcionais

- Artefato 7: Testes Unitários e de Integração:
 - Testes unitários e de integração escritos para testar os componentes da API;
 - Testes que simulam chamadas de API e verificam as respostas e interações com os serviços do GCP;
- Artefato 8: Caso existisse mais tempo para elaborar uma PoC, onde você colocaria foco?

O que será avaliado

Avaliar a capacidade do candidato de pensar com foco em arquitetura de sistemas, design de APIs, segurança e eficiência operacional, essencial para o sucesso em um ambiente de plataforma de dados corporativa.