

A seguir, apresento um fluxo de trabalho com as etapas de **Google Cloud Storage** para armazenar os dados, **Streamlit** para a aplicação e **GitHub Actions** para CI/CD:

## **Etapas para o Deploy Contínuo**

### **1. Armazenamento de Dados no Google Cloud Storage**

O Google Cloud Storage (GCS) será utilizado para armazenar os dados brutos e processados, além dos arquivos do modelo treinado.

- **Carregamento de Dados:** O pipeline de dados (ou script de coleta) pode ser configurado para carregar periodicamente os dados da API do EIA para o Google Cloud Storage, utilizando a API do Google Cloud ou bibliotecas como google-cloud-storage.
- **Estrutura de Diretórios no GCS:**
  - gs://my-bucket/raw/: Para os dados brutos obtidos da API.
  - gs://my-bucket/processed/: Para os dados processados e preparados para o modelo.
  - gs://my-bucket/models/: Para armazenar os modelos treinados.

### **2. Desenvolvimento da Aplicação com Streamlit**

O Streamlit será utilizado para construir o dashboard de previsão de preços de petróleo.

- **Aplicação Streamlit:**
  - A aplicação será hospedada no ambiente de nuvem e o modelo treinado será carregado diretamente de GCS.
  - O código Python para o modelo será implementado e interligado com a interface do Streamlit, permitindo que o usuário faça previsões, visualize gráficos e interaja com os dados.

### **3. Configuração do Pipeline de CI/CD com GitHub Actions**

O GitHub Actions será configurado para automatizar os testes, build e deploy da aplicação e do modelo para produção.

#### **a. Pipeline de Build e Teste**

- **Testes Unitários:** Configuração de testes automatizados utilizando as bibliotecas pytest e unittest para garantir que o código e o modelo funcionem corretamente.

#### **b. Pipeline de Deploy**

O pipeline de deploy vai ser configurado para fazer o deploy contínuo da aplicação e do modelo no ambiente de produção:

##### **1. Deploy da Aplicação Streamlit:**

- Após os testes serem executados e aprovados, o GitHub Actions aciona o processo de deploy da aplicação no Google Cloud.
- A aplicação vai ser implantada diretamente no **Streamlit Cloud**.

## 2. **Deploy do Modelo:**

- O modelo treinado será salvo em um arquivo .pkl e armazenado no Google Cloud Storage.
- A cada nova versão do modelo, o GitHub Actions poderá acionar o script para carregar o novo modelo do GCS para o ambiente da aplicação.

### **Fluxo Geral**

1. **Desenvolvimento:** O código é desenvolvido localmente, com o modelo treinado e o código da aplicação Streamlit implementados.
2. **Push para o Repositório GitHub:** Sempre que houver um novo commit ou pull request para o branch main, o GitHub Actions inicia o pipeline de build, teste e deploy.
3. **Execução de Pipeline:**
  - A cada push para o repositório, o GitHub Actions testa o código e faz o deploy automaticamente, subindo a aplicação para o Google Cloud e os modelos para o GCS.
4. **Execução na Produção:** O modelo atualizado estará disponível para previsões na aplicação Streamlit, acessível via navegador.

Esse fluxo garante a automação do processo de deploy contínuo, minimizando a intervenção manual e mantendo a aplicação e o modelo sempre atualizados.