Projeto "Docker Data Science Environment"

Autor: Eric Pimentel

Belém-PA - 2025

Documentação Inicial:

1. Introdução

- **Título** : Docker Data Science Environment
- **Descrição** : Um ecossistema modular e replicável para estudos em Data Science, Engenharia de Dados e DevOps. Projetado para ser eficiente, escalável e fácil de usar, mesmo para iniciantes.
- **Objetivo** :
    - Fornecer uma plataforma completa para análise de dados, machine learning, visualização e monitoramento.
    - Demonstrar boas práticas de DevOps e Engenharia de Dados.
    - Facilitar o aprendizado e a experimentação em Data Science.

2. Problema e Necessidade

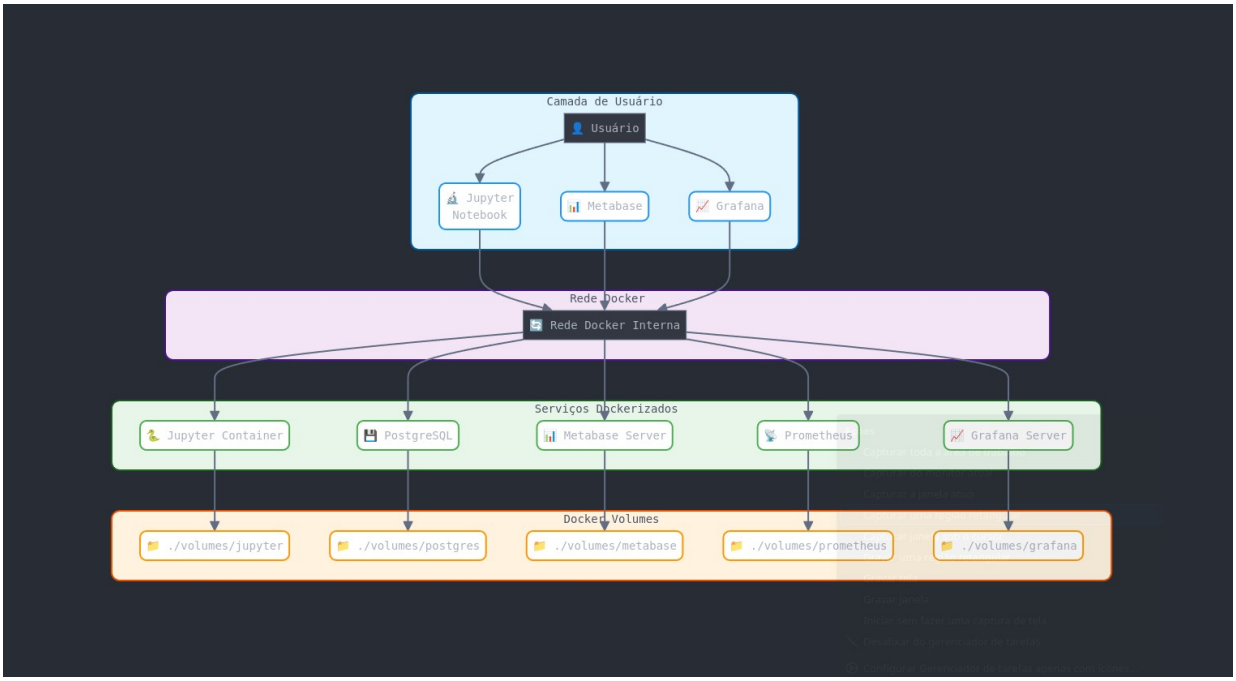
- **Problema** :
    - Muitos iniciantes em Data Science enfrentam dificuldades para configurar ambientes de desenvolvimento consistentes e replicáveis.
    - A falta de integração entre ferramentas (Python, banco de dados, BI, monitoramento) torna o fluxo de trabalho fragmentado e ineficiente.
    - Ambientes locais podem ser difíceis de manter e escalar, especialmente para projetos colaborativos.
- **Necessidade** :
    - Uma solução unificada que combine todas as ferramentas necessárias em um único ambiente.
    - Um sistema modular que permita aos usuários aprender e experimentar sem medo de perder dados ou configurar manualmente cada componente.

3. Solução Proposta

- **Visão Geral** :
    - O "Docker Data Science Environment" é uma plataforma Dockerizada que integra:
        1. **Jupyter Notebook** : Para análise de dados e machine learning.
        2. **PostgreSQL** : Para armazenamento estruturado de dados.
        3. **MetaBase** : Para criação de dashboards interativos.
        4. **Grafana + Prometheus** : Para monitoramento e observabilidade.
    - Todos os serviços são conectados via rede Docker e persistem dados usando **Docker Volumes** centralizados.
- **Benefícios** :

- **Replicabilidade** : Qualquer pessoa pode rodar o ambiente com um único comando (`docker-compose up`).
- **Persistência** : Dados e arquivos são salvos localmente, garantindo que nada seja perdido.
- **Escalabilidade** : Projetado para ser expandido conforme necessário.
- **Acessibilidade** : Ideal tanto para iniciantes quanto para profissionais experientes.

4. Arquitetura do Ecossistema



Descrição do Diagrama :

- **Jupyter Notebook** : Conectado ao PostgreSQL para análise de dados.
- **PostgreSQL** : Armazena dados estruturados e é acessível pelo Jupyter e MetaBase.
- **MetaBase** : Conectado ao PostgreSQL para criar dashboards.
- **Prometheus** : Coleta métricas de todos os contêineres.
- **Grafana** : Visualiza métricas coletadas pelo Prometheus.
- **Rede Docker** : Todos os serviços estão conectados via rede Docker, permitindo comunicação eficiente.
- **Docker Volumes Centralizados** : Todos os volumes (dados, logs, configurações) são armazenados em um único diretório na máquina do usuário.

5. Componentes do Sistema

- **Jupyter Notebook** :
  - Ferramenta principal para análise de dados e machine learning.
  - Bibliotecas instaladas: `pandas`, `numpy`, `scikit-learn`, `matplotlib`, `seaborn`, `plotly`, `psycopg2`.
- **PostgreSQL** :
  - Banco de dados relacional para armazenar dados estruturados.

    - Configurado com um volume Docker para persistência.
- **MetaBase** :
    - Ferramenta de BI open source para criação de dashboards.
    - Conectado ao PostgreSQL para visualizar dados.
- **Grafana + Prometheus** :
    - Prometheus coleta métricas dos contêineres (CPU, memória, rede).
    - Grafana exibe essas métricas em dashboards interativos.
    -
- **Docker Volumes** :
    - Todos os volumes são centralizados em um único diretório (`./volumes`) na máquina do usuário.
    - Estrutura sugerida:

      /volumes
      ├── jupyter/
      ├── postgres/
      ├── metabase/
      └── grafana/

6. Como Usar o Sistema

- **Pré-requisitos** :
    - Docker e Docker Compose instalados.
    - Git para clonar o repositório.
- **Passos** :
    - Clone o repositório:
      git clone https://github.com/seu-usuario/docker-data-science.git
cd docker-data-science

    - Inicie o ambiente:
      docker-compose up -d

    - Acesse os serviços:
        - Jupyter Notebook: `http://localhost:8888`
        - MetaBase: `http://localhost:3000`
        - Grafana: `http://localhost:3001`

- **Exemplos de Uso** :
    - Scripts Python no Jupyter Notebook para conectar ao PostgreSQL.
    - Dashboards no MetaBase para visualizar dados.
    - Monitoramento no Grafana para acompanhar a saúde do sistema.

7. Contribuições e Feedback

- **Contribuições** :

    - Este projeto é open source e aceita contribuições. Sinta-se à vontade para abrir issues ou pull requests no GitHub.
- **Feedback** :
    - Se você encontrar problemas ou tiver sugestões, entre em contato via GitHub ou LinkedIn.