**Empresa Inc.**

Pedro Bianchini de Quadros

DOCUMENTAÇÃO PARA DEPLOY DA APLICAÇÃO

**Observação:** Para desenvolver esta proposta de deploy em ambiente de produção, baseei-me em pesquisas para chegar a essas conclusões. Reconheço que a implementação pode estar apresentada de forma mais superficial e que minha experiência prática com deploys em ambientes de produção é limitada.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Configuração do Servidor**

Para estabelecer um ambiente de produção robusto, inicie com a configuração do servidor web e do serviço de aplicação. Utilize um servidor web de alta performance como Nginx ou Apache para servir o aplicativo Flask. Nginx é particularmente recomendado devido à sua eficiência e leveza. Para gerenciar as solicitações WSGI, empregue um servidor WSGI como Gunicorn ou uWSGI, que são otimizados para este fim. Por exemplo, para iniciar o Gunicorn, utilize o comando gunicorn -w 4 -b 0.0.0.0:8000 app:app, onde -w 4 especifica quatro trabalhadores e app:app define o módulo e a instância do Flask.

**Configuração do Banco de Dados**

A configuração do MariaDB em um ambiente de produção requer atenção rigorosa tanto à performance quanto à segurança. Assegure-se de utilizar senhas robustas e restrinja o acesso ao banco de dados a IPs específicos, quando viável. É imperativo implementar um sistema de backups regulares e testar periodicamente os procedimentos de restauração para garantir a integridade e a disponibilidade dos dados em caso de falha.

**Otimização e Escalabilidade**

Para otimizar o desempenho do aplicativo, considere a implementação de mecanismos de caching para minimizar a carga sobre o banco de dados e melhorar o tempo de resposta. Ferramentas como Redis são recomendadas para esse propósito. Além disso, configure um sistema de monitoramento para avaliar a saúde do aplicativo e registre logs de erros utilizando ferramentas como Sentry e o ELK Stack (Elasticsearch, Logstash, Kibana). Avalie a necessidade de escalabilidade horizontal (adicionando mais servidores) e vertical (aumentando os recursos dos servidores) conforme o crescimento da demanda.

**Deploy**

Automatize o processo de deploy com ferramentas de infraestrutura como Docker, Kubernetes ou Ansible, facilitando a implantação e o gerenciamento do ambiente de produção. Implemente um pipeline de integração contínua e entrega contínua (CI/CD) para automatizar os testes e os deployments, assegurando um fluxo de trabalho mais eficiente e menos suscetível a erros humanos.