

Solicitação de Infraestrutura Docker para Aplicação de FAQ

Data: 15 de Abril de 2025

Para: Equipe de Infraestrutura / DevOps

De: Equipe de Desenvolvimento da Aplicação de FAQ

Assunto: Solicitação de Infraestrutura Docker para Deploy da Aplicação de FAQ

Prezados(as),

Este documento tem como objetivo formalizar a solicitação da infraestrutura necessária para o deploy da nossa nova aplicação de FAQ (Perguntas e Respostas) desenvolvida em Node.JS versão 22. A arquitetura planejada para a implantação se baseia em containers Docker para garantir a portabilidade, escalabilidade e consistência do ambiente.

1. Descrição da Aplicação:

A aplicação de FAQ é um sistema web que permitirá aos usuários do Curso de Análise e Desenvolvimento de Sistemas cadastrar dúvidas, receber respostas e consultar o histórico de perguntas e respostas. A aplicação possui as seguintes funcionalidades principais:

- Tela de Cadastro de Dúvidas.
- Tela para Resposta das Perguntas.
- Tela de Consulta das Perguntas.
- Controle de respondentes e questionadores.
- Registro da data e hora da pergunta e da resposta.
- Categorização de perguntas por tema principal.

2. Requisitos de Infraestrutura Docker:

Solicitamos a criação e configuração da seguinte infraestrutura baseada em Docker:

- **Container para a Aplicação Node.JS:**
 - Imagem Docker baseada em um ambiente adequado para Node.JS versão 22 (ex: `node:22-alpine`).
 - Configuração para executar a aplicação Node.JS.
 - Exposição da porta da aplicação 3000.
 - Configuração de variáveis de ambiente necessárias para a aplicação (ex: conexão com o banco de dados).
 - Mecanismo para deploy de novas versões da aplicação no container (ex: Docker Compose, orquestrador de containers).

- **Container para o Banco de Dados PostgreSQL:**

- Imagem Docker oficial do PostgreSQL (versão mais recente estável).
- Configuração de um volume persistente para os dados do banco de dados, garantindo a persistência das informações.
- Definição de usuário, senha e banco de dados padrão para a aplicação.
- Exposição da porta padrão do PostgreSQL 5000.

- **Container para o Sistema de Observabilidade (Prometheus):**

- Imagem Docker oficial do Prometheus.
- Configuração do Prometheus para coletar métricas da aplicação Node.JS (será necessário expor um endpoint de métricas na aplicação).
- Configuração para coletar métricas do container do PostgreSQL (via um *exporter* adequado, como `postgres_exporter`).
- Persistência dos dados do Prometheus (opcional, mas recomendado para retenção de histórico).
- Exposição da porta padrão do Prometheus (porta 9090).

- **Container para o Sistema de Visualização (Grafana):**

- Imagem Docker oficial do Grafana.
- Configuração de um *datasource* para conectar ao Prometheus.
- Provisionamento de dashboards padrão para visualizar as métricas da aplicação, do banco de dados e da infraestrutura Docker.
- Persistência da configuração do Grafana (dashboards, datasources).
- Exposição da porta padrão do Grafana (porta 3000).

- **Sistema de Gerenciamento de Logs:**

- Solicitamos a implementação de um sistema centralizado de gerenciamento de logs para facilitar o troubleshooting e a análise do comportamento da aplicação.
- Considerar soluções como:
 - **ELK Stack (Elasticsearch, Logstash, Kibana):** Containers separados para cada componente.
 - **Loki e Promtail:** Containers para Loki (armazenamento de logs) e Promtail (agente para coletar e enviar logs). Integração com Grafana para visualização.
- Definir a estratégia de coleta de logs da aplicação Node.JS e do PostgreSQL para o sistema de gerenciamento de logs escolhido.
- Configuração de retenção de logs adequada.

3. Requisitos Adicionais:

- **Rede Docker:** Criação de uma rede Docker para permitir a comunicação entre os containers da aplicação, banco de dados, Prometheus e Grafana.
- **Orquestração (Opcional Inicialmente):** Embora não seja um requisito imediato, solicitamos que a infraestrutura seja pensada de forma a facilitar a futura adoção de um orquestrador de containers como Docker Compose (para ambientes menores) ou Kubernetes (para maior escalabilidade).
- **Acesso e Segurança:** Definição de políticas de acesso aos containers e aos dados, garantindo a segurança da aplicação e do banco de dados.
- **Monitoramento da Infraestrutura Docker:** Implementação de ferramentas básicas de monitoramento da saúde dos containers (CPU, memória, disco).

4. Benefícios da Infraestrutura Docker:

A adoção do Docker para esta aplicação trará diversos benefícios, incluindo:

- **Consistência do Ambiente:** Garante que a aplicação rode no mesmo ambiente, independentemente do servidor de deploy.
- **Portabilidade:** Facilita a implantação em diferentes ambientes (desenvolvimento, testes, produção).
- **Escalabilidade:** Permite escalar facilmente a aplicação e o banco de dados conforme a demanda.
- **Isolamento:** Isola os componentes da aplicação, evitando conflitos de dependências.
- **Facilidade de Deploy:** Simplifica o processo de deploy de novas versões da aplicação.
- **Melhor Utilização de Recursos:** Permite uma melhor alocação e utilização dos recursos do servidor.

5. Próximos Passos:

Solicitamos que a equipe de infraestrutura analise esta solicitação e nos apresente um plano de implementação para a criação da infraestrutura Docker descrita. Ficamos à disposição para discutir os detalhes e fornecer qualquer informação adicional necessária.

Agradecemos a atenção e colaboração.

Atenciosamente,

Equipe de Desenvolvimento da Aplicação de FAQ.