**Documentação Passo a Passo para Implementação de um Novo Projeto no Kubernetes**

**Passo 1: Repositório**

1.1 Localize o Repositório do Projeto

* Identifique o repositório do projeto que será implantado no Kubernetes.

1.2 Baixe o Repositório

* Clone o repositório usando o comando:

git clone <link-repo>

* Verifique se o arquivo Dockerfile está presente no repositório.

**Passo 2: Criar Repositório**

2.1 Crie um Repositório

* Registre a imagem do projeto em um repositório na nuvem (cloud).
* Copie a URL do repositório para referência futura.

**Passo 3: Build da Aplicação**

3.1 Build da Aplicação

* Navegue até a pasta do repositório clonado em seu computador.
* Execute o seguinte comando para construir a imagem:

docker build -t <nome-desejado> .

(O ponto representa o Dockerfile na pasta)

* Verifique se a imagem foi criada usando:

docker images

3.2 Tag da Imagem

* Dê uma tag à imagem com o nome desejado usando:

docker tag <id-do-container> <url-do-repo>

3.3 Enviar a Imagem para o Repositório na Nuvem

* Envie a imagem para o repositório na nuvem usando:

docker push <url-do-repo>

**Passo 4: Criar Manifesto**

4.1 Crie Arquivos YAML para o Deploy no Kubernetes

* Crie os seguintes arquivos YAML para o deploy no Kubernetes:
* namespace.yaml
* deployment.yaml
* service.yaml
* ingress.yaml

**Possíveis Erros**

Erro 1: EOF durante o Docker Push

* Se ocorrer um erro de EOF ao tentar enviar a imagem para o repositório, verifique o comando de login na nuvem. Exemplo:

\*print\*

Observação: Para contas AWS adicionais, inclua o parâmetro --profile <nome-da-conta>.

Erro 2: IP Interno nos Logs dos Pods

* Se os logs dos pods mostrarem um IP interno, adicione 0.0.0.0 no Dockerfile. Exemplo:\*print\*
* Se o problema persistir, verifique se a porta no Dockerfile está configurada corretamente. Por exemplo, se estiver usando a porta 8000:

EXPOSE 8000