Olá!

Você está então agora entrando na jornada de uma pessoa SRE, vamos apenas realizar alguns testes para entender como você avalia problemas e desenvolve soluções criativas e eficientes para problemas modernos.

Esse repositório (e quando citarmos *"esse repositório"*, estamos nos referindo ao diretório raíz dele) é um projeto. Um projeto de automação de teste de uma aplicação. Porém temos alguns desafios para que ele alcance o objetivo de **trazer confiabilidade para o negócio garantindo qualidade de desenvolvimento e boas práticas de aplicações cloud native**. Claro, também podemos citar que um diferencial nessas metas seria **reduzir o número, tempo e responsabilidade de interações humanas para que o pipeline tenha garantias de sua execução**.

Algumas regras:

- Você está livre para pesquisar e utilizar as ferramentas que está acostumado para resolver o problema desde que mantenha as tecnologias obrigatórias: **GitHub Actions, Docker, Linux**.

Como é feita a avaliação:

- Os critérios serão avaliados por peso e demonstração de competência, exemplo: *sobre o assunto X (com peso Z) o projeto atendeu Y%, logo sua pontuação será: Y%Z*.
- Critérios podem ter o mesmo peso, alguns são mais importantes do que outros, queremos avaliar e entender suas decisões.
- A soma de da pontuação de todos os critérios (demonstrações de capacidades) será sua nota final.
- Partindo da pontuação teórica máxima realizaremos uma avaliação interna de qual perfil e nível podemos te oferecer.

Uma dica importante: se não conseguir realizar as várias tarefas dessa avaliação, não se preocupe, foque em fazer as que você se sente confortável.

Contexto

Para dar um pouco de contexto ao projeto vamos falar sobre o cenário atual: faltam apenas 30 dias para a data de uma campanha promocional que será feita pelo time de marketing em conjunto com o time comercial e pelo histórico sabemos que o volume de usuários aumenta muito.

Essa promoção envolve um serviço de sorteio que depende de um serviço para gerar números randômicos. Como o negócio cresceu, no último ano fizemos a transição de um data center privado para a nuvem e agora não temos mais o serviço antigo que era baseado em geração por hardware. Precisamos de uma nova solução.

O que fizemos até agora

Temos uma aplicação em **Golang** nesse repositório e uma declaração de **workflow do GitHub Action**. Essa aplicação expõe os seguintes endpoints:

- `HTTP, porta 8080, GET /random-number` - um número randômico em um body em JSON.
- `HTTP, porta 9090, GET /metrics` - métricas sobre a aplicação.

Próximos passos

Crie um repositório privado na sua conta no GitHub e adicione os usuários que informamos. Nesse repositório faça o push para a branch **main** do conteúdo desse `.zip` que te enviamos. As tarefas abaixo devem ser feitas nesse repositório que você criou.

Para configurar seu repositório

- ☐ Realize a substituição de todas as strings `testing/sre-test-1` por `SEU_USUARIO_GIT/NOME_DO_SEU_REPOSITÓRIO` criando um script para fazer essa tarefa (na linguagem de sua escolha).
- ☐ Faça o commit e push da alteração para seu repositório.

To fix

- ☐ Aplicação não está realizando build da imagem Docker.
- ☐ Aplicação não está fazendo log durante o teste funcional.
- ☐ Existe um step no pipeline em que realizamos um teste funcional realizando o request para http://localhost:8080/random-number e validamos a resposta, verificar se o teste feito aqui realmente garante que o endpoint está respondendo devidamente.
- ☐ Criar o mesmo teste funcional para a rota `/metrics` da porta **9090**.

To do

- [ ] Realizar testes de performance na geração de números randômicos.
- [ ] Trazer relatórios sobre estatísticas e métricas dos testes de performance.
- [ ] Diminuir tempo de geração de número randômico.
- [ ] Criar documentação para outros colaboradores contribuírem com o projeto.
- [ ] Implementar métricas sobre o serviço http que responde na rota `/get-random-number` (dicas https://www.robustperception.io/prometheus-middleware-for-gorilla-mux e para uma implementação mais simples, utilize o arquivo internal/router/router.go)
- [ ] Reduzir tempo de execução do workflow