
AWS- CHALLENGE

14 DE AGOSTO

Criado por: Bruno Caribé

Objetivos

O Objetivo do projeto era criar um API Rest, sem persistência de dados, na linguagem de programação que fosse confortável. Posteriormente, criar um DockerFile que serviria tanto para subir a aplicação localmente quanto para buildar a aplicação para seu futuro push para um repositório público. Necessário também criar um pipeline de CI/CD utilizando-se da tecnologia que fosse confortável, terraform para criação da infraestrutura e aplicação no cloud provider de preferência e criar e dispor do monitoramento da aplicação criada.

Tecnologias utilizadas

- Desenvolvimento da API: Foi utilizada a linguagem GO (Golang: <https://go.dev/>), por ser uma linguagem clara, objetiva e versátil, de fácil legibilidade e excelente desempenho.

- CI/CD: Foi utilizado o GitHub Actions (<https://github.com/features/actions>) pelo fato de estar utilizando um repositório já no github e grande facilidade de integração. Foi feito pipeline de CI realizando build da aplicação e envio da imagem para o docker hub e pipeline de CD realizando o deploy da mesma na AWS.

- Cloud Provider: Foi utilizado a AWS pelo conjunto de serviços, facilidade e familiaridade com a mesma para o projeto.

- Container: Foi utilizado o conjunto de Fargate e ECS pois, ao contrário do EKS onde seria necessário provisionar máquinas EC2 para utilização, ele é um serviço servless onde a única preocupação seria o microserviço em si. Além disso, o ECS se integra nativamente melhor aos outros serviços disponíveis na AWS, como o Load Balancer e CloudWatch, também utilizados no projeto.

Endpoints

LoadBalancer - Healthcheck: <http://caribeh-loadbalancer-330574661.us-east-1.elb.amazonaws.com/healthcheck>

LoadBalancer - API: caribeh-loadbalancer-330574661.us-east-1.elb.amazonaws.com/games

Monitoramento – Cloudwatch: [cloudwatch](#)

Repositório – github: <https://github.com/caribeh/apijogos>

Imagem pública: <https://hub.docker.com/r/caribeh/apijogos>