Arquitetura Serverless com IaC para Soluções de IA

Tabela de conteúdo

Tabela de conteúdo

Introdução

Meu nome é Cleiton Pessoa, tenho 25 anos e trabalho como desenvolvedor fullstack há cerca de 5 anos.

Desde o início da minha carreira, sempre fui curioso sobre o que acontece além do código — especialmente quando o assunto é infraestrutura, automação e entrega contínua. Mas foi há cerca de 2 anos que comecei a mergulhar no mundo DevOps.

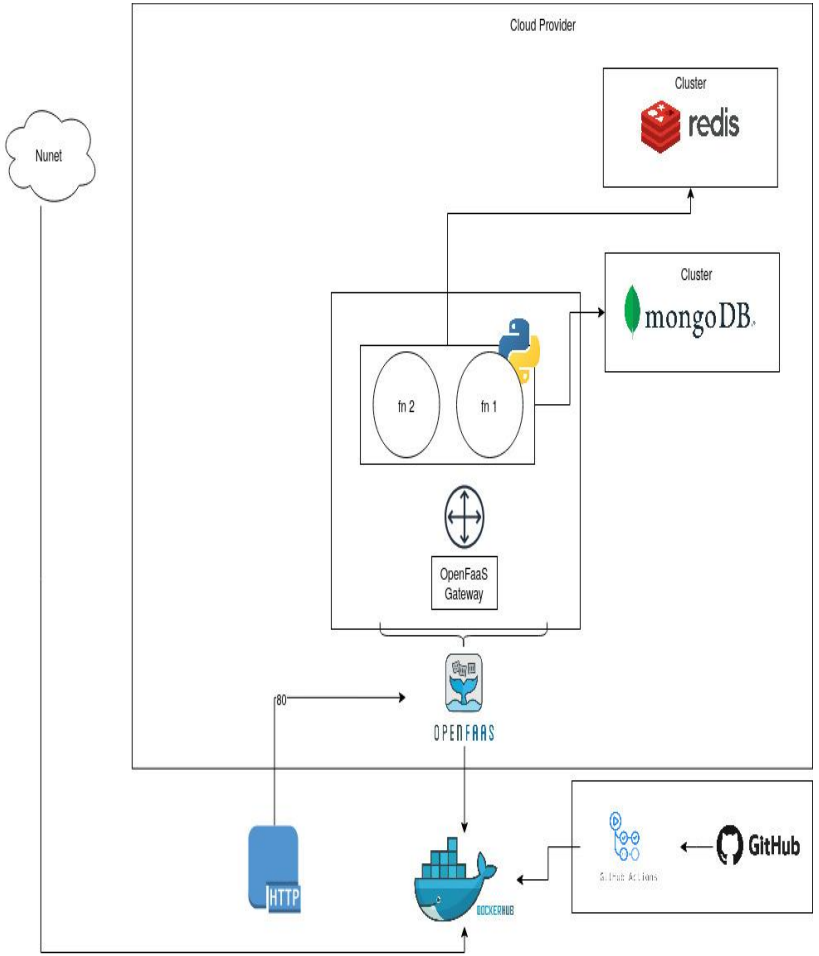
Tudo começou com um desafio: lidar com a AWS em um projeto real. E foi ali que percebi o quanto entender "o outro lado" é essencial para entregar software com mais qualidade, segurança e confiança.

Sobre o projeto

Participei de um projeto voltado para soluções de inteligência artificial, onde atuei na definição e implementação da infraestrutura.
Nosso objetivo era criar uma plataforma unificada que permitisse o acesso a diversos algoritmos de IA, executando de forma distribuída e sob demanda.

Arquitetura adotada:

- API Serverless com OpenFaaS, sobre servidores dedicados
- Algoritmos executados de forma distribuída
- Provisionamento com Terraform
- Banco de dados: Redis em cluster e MongoDB
- Pipelines CI/CD com GitHub Actions



Serverless + Computação Distribuída

A camada serverless atuava como interface pública, cuidando de:

- Autenticação
- Roteamento
- Orquestração de requisições

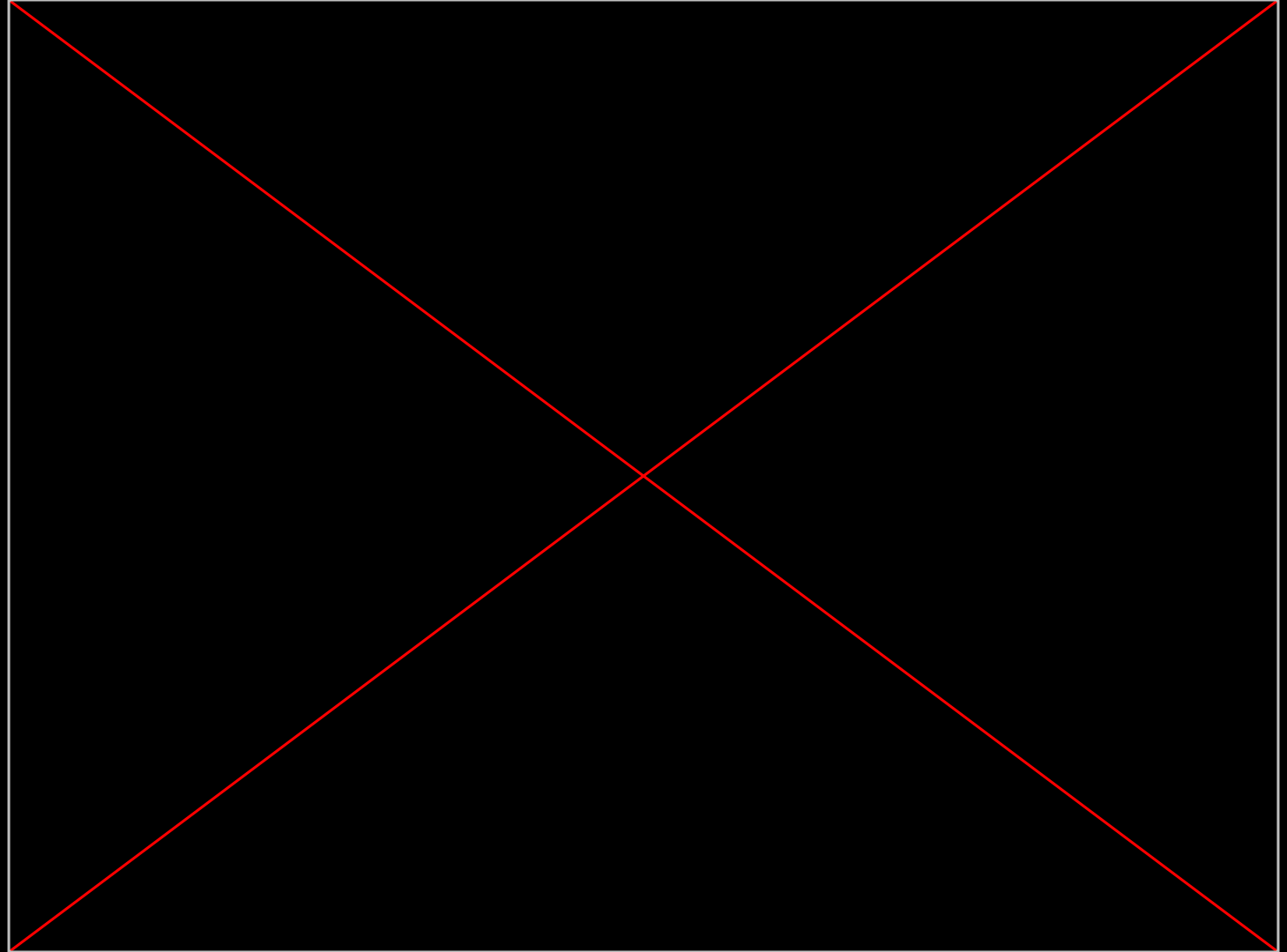
O processamento dos algoritmos de IA ocorria em nós específicos, com:

- Recursos dedicados (CPU/GPU)
- Ambiente customizado por algoritmo
- Execução paralela e sob demanda

OpeenFaaS

Optamos por uma arquitetura serverless porque precisávamos de uma API leve, escalável e de fácil manutenção. No entanto, as opções tradicionais — como AWS Lambda ou Google Cloud Functions — impõem restrições importantes:

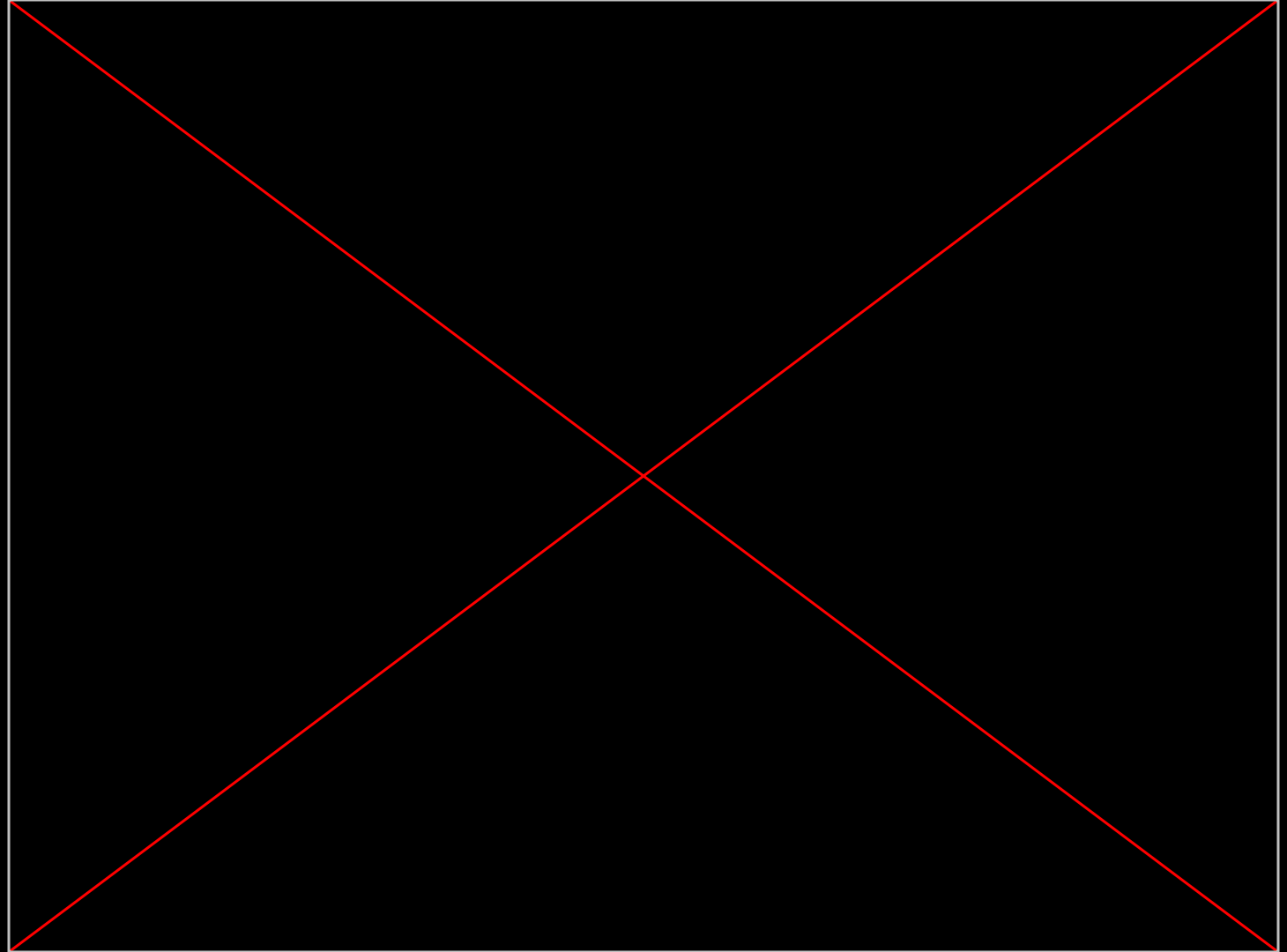
- Menor controle sobre o ambiente
- Impossibilidade de rodar em infraestrutura própria, com custos previsíveis e zero lock-in



Nunet

A NuNet é uma rede descentralizada para execução distribuída de workloads — como algoritmos de IA, processamento de dados ou qualquer aplicação containerizada.

Em vez de depender de uma nuvem centralizada, a NuNet utiliza nós executores espalhados pela rede: servidores, desktops ou até computadores pessoais.



Provisionamento com Terraform

Um dos pilares do projeto foi garantir que toda a infraestrutura fosse automatizada, versionada e previsível.

Usamos Terraform para descrever toda a infraestrutura como código:

- Servidores, rede, storage
- Instalação do OpenFaaS
- Garantir reprodutibilidade: subir o ambiente do zero com um único comando
- Controlar versionamento de mudanças, com histórico claro

CI/CD automatizado

A entrega contínua também foi automatizada com uma esteira CI/CD baseada 100% no GitHub Actions.
Esteira CI/CD:

- Stages bem definidos: build, testes, empacotamento e deploy
- Deploy automático das funções serverless para o OpenFaaS
- Publicação de imagens Docker no Docker Hub, com versionamento semântico

Boas práticas:

- Custom Actions para centralizar lógica comum (ex: autenticação, deploy)
- Reusable Workflows para aplicar o mesmo padrão em múltiplos repositórios

Controle de Versão

Adotamos o padrão MAJOR.MINOR.PATCH para todas as releases — automatizado por um script em Bash que analisava os commits e decidia o tipo de incremento.

Por que isso foi essencial?

Facilitou rastreio e rollback de versões com segurança

Acoplou tags às imagens Docker, tornando builds rastreáveis e previsíveis

Melhorou a comunicação entre equipes: todos sabiam se a nova versão tinha breaking change ou melhorias pontuais

Agradecimentos

Esse projeto me trouxe um enorme aprendizado — tanto técnico quanto de visão de arquitetura.

Agora, gostaria de abrir o espaço pra vocês:
Perguntas, comentários, experiências parecidas?

Fiquem à vontade! Estou aqui para compartilhar, aprender e trocar ideias com vocês.