

Sandhole

Uma jornada de self-host em Rust

Eric Pires

`println!("Hello, world!");`

- Eric "**Epic Eric**" Pires
- Engenheiro de software @ **Artemis Technologies**
- Anteriormente @ **NIC.br**
- Entusiasta de open-source e Rust



Self-hosting



Self-hosting

- Independência virtual.
- Privacidade e controle.
- Reciclagem de hardware.
- Aprendizado sobre tecnologias.
- É divertido!
- <3 open-source.

Desafios

IPs dinâmicos...

E port forwarding...

E Network Address Translation (NAT)...

- VPN
 - Software específico nos dois lados
- VPS com IP público
 - Limitado aos recursos da máquina
- Proxy reverso
 - Ainda precisa de uma maneira de conectar

E se eu somar tudo? Existe uma solução...?

Sim. Sim, há.

SSH



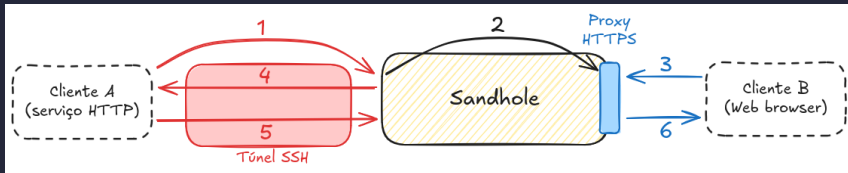
Um reverse proxy baseado em SSH.

- Sem mais problemas de NAT!
- HTTPS automático para serviços expostos

```
ssh -R meusite.com.br:80:localhost:3000 sandhole.com.br
```

E mais...!

- Autenticação via chaves SSH
- Domínios personalizados
- Load balancing
- Suporte a portas TCP, incluindo SSH
- Forwarding local - quase uma VPN
- UI de administração no terminal (via SSH, claro)



<https://sandhole.com.br>

Por que esse projeto?

Baseado numa aplicação existente (sish, em Golang).

Pouca experiência prévia em Rust e proxies.

Por que não reescrever em Rust...?



A linguagem certa?

- Baixo nível (HTTP, TCP, TLS, SSH...)
- Ecossistema de bibliotecas
- `async` via `tokio`
- `stdlib` poderosa
- Tipagem e erros de compilação
- Testes unitários+integração
- CI/CD, cross-compilation com Docker
- Refatoração sem medo

Aprendizagem => Síntese

Demonstração

```
~/tmp/demo/CraneGame_linux_x86_64
```

snippet +exec is disabled, run with -x to enable

Obrigado!

- <https://github.com/EpicEric/sandhole>
- E-mail: eric@eric.dev.br
- Mastodon: [@EpicEric@mastodon.xyz](https://mastodon.xyz/@EpicEric)
- Discord: [@epiceric](#)
- Telegram: [@ericpires9](#)

