interweb

Gerado a 29/10/2026 05:16

BASE DE CONHECIMENTO · DOCUMENTO PÚBLICO

Correr um programa em Go num VPS como um serviço que se reinicia sozinho

Node.js, Python e Docker

CÓDIGO	VERSÃO	PAÍS · IDIOMA
RLX-MZ-KB-1874	v2026.10.28	Moçambique · Português

Interweb Moçambique
RLX Holdings, S.A. · Grupo RLX Holdings
NUIT 401 191 607 · NUEL 105 045 974 · CAE 62022

https://www.meu.interweb.co.mz/knowledgebase/1874

Correr um programa em Go num VPS como um serviço que se reinicia sozinho

Node.js, Python e Docker · Criado em **15 de outubro de 2026** · Atualizado em **28 de outubro de 2026** · 2 min de leitura



A resposta curta: um programa em **Go** compila para um único ficheiro, sem nada para instalar no servidor. Num VPS, o caminho certo é: compilar, copiar o ficheiro, e pô-lo a correr como um **serviço do sistema (systemd)**, que o arranca com o servidor e o reinicia se falhar. Depois, um servidor web à frente trata do endereço e do https.

1. Compilar para Linux

No seu computador (seja Windows, Mac ou Linux), na pasta do projeto:
`GOOS=linux GOARCH=amd64 go build -o meuapp .`
Isto cria o ficheiro `meuapp` para um servidor Linux de 64 bits. No Windows, defina as duas variáveis antes de correr o comando, de acordo com o seu terminal.

2. Copiar e preparar

1. **Envie o ficheiro** para o VPS: **Atalhos de SSH e scp**. Ponha-o numa pasta como `/opt/meuapp/` .
2. **Crie um utilizador sem poderes** para o correr (`sudo useradd --system --no-create-home meuapp`) e dê-lhe o ficheiro e a pasta.
3. **Faça o programa ouvir só em** `127.0.0.1` numa porta (por exemplo, 8080): quem o expõe à internet é o servidor web, e não ele.
4. **Dê permissão de execução:** `sudo chmod +x /opt/meuapp/meuapp` .

3. O serviço (systemd)

Crie o ficheiro `/etc/systemd/system/meuapp.service` com:
`[Unit]`
`Description=O meu programa em Go`

```
After=network.target

[Service]
User=meuapp
WorkingDirectory=/opt/meuapp
ExecStart=/opt/meuapp/meuapp
Restart=on-failure
Environment=PORT=8080

[Install]
WantedBy=multi-user.target
```

Para	Comando
Ler o ficheiro novo	`sudo systemctl daemon-reload`
Arrancar já e a cada reinício do servidor	`sudo systemctl enable --now meuapp`
Ver se está a correr	`sudo systemctl status meuapp`
Ver o que escreve (os registos)	`sudo journalctl -u meuapp -f`
Depois de enviar uma versão nova	`sudo systemctl restart meuapp`

Se o serviço não arranca, o `status` diz porquê: **Um serviço que não arranca**.

4. Mostrá-lo na internet

Ponha o nginx à frente, a passar os pedidos do seu domínio para `127.0.0.1:8080`: **O nginx como proxy inverso**, e o certificado: **HTTPS para um contentor**. Abra só as portas 80 e 443 na firewall: **A firewall ufw**.

> **Não corra o programa como administrador (root)**
> nem o deixe a ouvir diretamente à internet. E não ponha senhas no código: use variáveis de ambiente no ficheiro do serviço, com permissões fechadas (`EnvironmentFile=`). Veja **Senhas fora do código**.

> O padrão é o mesmo para um programa em Rust, em .NET ou em qualquer outro que gere um ficheiro executável. Para um que fica a correr numa janela do terminal e acaba quando a fecha, veja **tmux e screen**, que é a solução para tarefas pontuais, e não para um serviço.

PRECISA DE AJUDA?

Quer ajuda para pôr um programa seu num VPS? Abra um pedido a dizer a linguagem, a porta e o domínio.

Abrir pedido de suporte →

VEJA TAMBÉM

- Um serviço que não arranca no meu VPS: ler o systemctl status →
- O nginx como proxy inverso à frente de um contentor →
- FastAPI num VPS: workers uvicorn, cabeçalhos de proxy e um serviço que reinicia →
- Atalhos de SSH: o ficheiro config, e copiar ficheiros com o scp →

interweb

Interweb Moçambique

RLX Holdings, S.A. · Grupo RLX Holdings

NUIT	401 191 607
NUEL	105 045 974
CAE	62022
SEDE	Bairro 7 de Setembro Vilankulo 1304 · CEP 0411-02, Inhambane, Moçambique
CONTACTOS	info@interweb.co.mz · +258 84 635 1111 · +258 86 635 1111 · www.interweb.co.mz
HORÁRIO	2ª. à 6ª. Feira - 08 à 16 Horas Sábado & Domingo - Fechado

Versão online deste documento

RLX-MZ-KB-1874 · v2026.10.28 · Gerado a 29/10/2026 05:16