Gerado a 21/10/2026 13:28

BASE DE CONHECIMENTO · DOCUMENTO PÚBLICO

Como executar o seu programa como um serviço do systemd que inicia com o servidor

VPS e Servidores

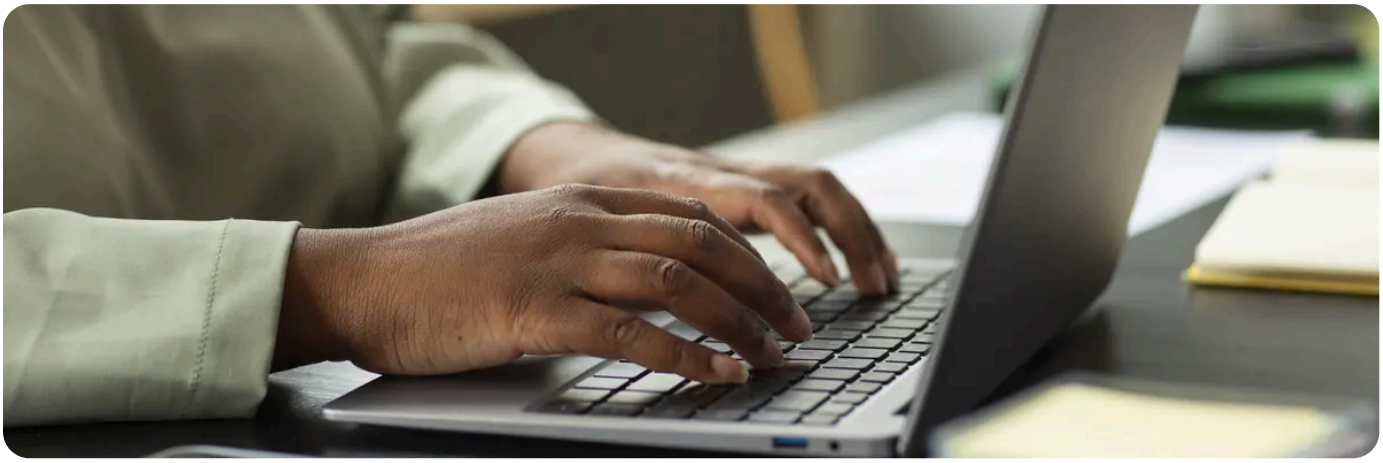
CÓDIGO	VERSÃO	PAÍS · IDIOMA
RLX-BR-KB-4242	v2026.10.18	Brasil · Português (Brasil)

MozOut Brasil
RLX Holdings (Pty) Ltd. · Grupo RLX Holdings
N.º fiscal (SARS) 943 907 6234 · N.º de registo (CIPC) 2025/667953/07

https://www.my.mozout.com/knowledgebase/4242

Como executar o seu programa como um serviço do systemd que inicia com o servidor

VPS e Servidores · Criado em **10 de outubro de 2026** · Atualizado em **18 de outubro de 2026** · 3 min de leitura



Para que um programa seu (uma aplicação Node.js, um script Python, um bot) inicie com o servidor e volte sozinho se cair, escreve-se um **arquivo de unidade** em `/etc/systemd/system/`, recarrega-se o systemd e ativa-se o serviço com `systemctl enable --now`. É a forma nativa de qualquer Linux moderno, sem instalar mais nada. Para aplicações Node.js existe ainda o PM2 (**manter uma aplicação Node.js a rodar com o PM2**); o systemd serve para tudo o resto, e também para o Node.

Passo a passo

1. **Rode o programa à mão primeiro.** Se não inicia no terminal, também não inicia como serviço. Anote o caminho completo do programa (`which node` ou `which python3`) e a pasta de onde corre.

2. **Crie um usuário sem poderes** para o serviço, em vez de o rodar como root: `sudo useradd --system --home /opt/meuapp --shell /usr/sbin/nologin meuapp`. Se o programa for comprometido, o estrago fica preso a esse usuário.

3 **Escreva a unidade** com `sudo nano /etc/systemd/system/meuapp.service`

```
[Unit]
Description=A minha aplicação
After=network.target

[Service]
User=meuapp
WorkingDirectory=/opt/meuapp
ExecStart=/usr/bin/node /opt/meuapp/server.js
Restart=on-failure
RestartSec=5
EnvironmentFile=/etc/meuapp.env

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

4 **Ponha os segredos num arquivo à parte,** `/etc/meuapp.env`, com uma linha `NOME=valor` por variável, e proteja-o: `sudo chmod 600 /etc/meuapp.env`. Veja **variáveis de ambiente e segredos**.

5 **Diga ao systemd que há uma unidade nova** e arranque-a: `sudo systemctl daemon-reload` e depois `sudo systemctl enable --now meuapp`. O "enable" faz iniciar com o servidor; o "--now" inicia já.

6 **Confirme.** `systemctl status meuapp` deve dizer "active (running)", e `journalctl -u meuapp -f` mostra o que o programa escreve, ao vivo.

7 **Prove o reinício.** Reinicie o servidor numa hora calma e veja se o serviço volta. Só assim fica a saber que funciona.

As linhas da unidade, em palavras

Linha	Para quê
`After=network.target`	Só inicia depois de a rede estar pronta.
`User=`	Quem roda o programa. Nunca root, a não ser que precise mesmo.
`ExecStart=`	O comando, com o **caminho completo** do programa. O systemd não usa o seu PATH nem o seu terminal.
`Restart=on-failure`	Volta a iniciar se terminar com erro; não volta se você o parou de propósito.

Linha	Para quê
`EnvironmentFile=`	Lê as variáveis de um arquivo, em vez de as escrever na unidade (que qualquer usuário pode ler).
`WantedBy=multi-user.target`	É o que faz o `enable` iniciá-lo na inicialização normal do servidor.

O `ExecStart` não é um terminal.
Redirecionamentos, `&&` e `~` não funcionam como no terminal; use caminhos absolutos e, se precisar de vários comandos, ponha-os num script e execute o script. Se vir "status=203/EXEC", o caminho está errado: **ler o systemctl status**.

Mudou a unidade e nada mudou?
Falta o `sudo systemctl daemon-reload`. E uma unidade que dá "Permission denied" ao ler uma pasta é permissões ou o SELinux: **permissões de arquivos**.

Reiniciar com um temporizador, em vez do cron:
O systemd também agenda (arquivos `.timer`), mas para tarefas simples o cron continua sendo o mais direto: **quando o cron não corre**.

PRECISA DE AJUDA?

O programa corre à mão e falha como serviço, e já leu o registro? Mostre-nos o texto do erro e o que mudou desde que funcionava.

Abrir um pedido de suporte →

VEJA TAMBÉM

- Um serviço não inicia: ler o systemctl status →
- Onde estão os registros num VPS Linux →
- Manter uma aplicação Node.js a rodar com o PM2 →
- Até onde vai o nosso suporte →

Mozout

MozOut Brasil

RLX Holdings (Pty) Ltd. · Grupo RLX Holdings

N.º FISCAL (SARS)	943 907 6234
N.º DE REGISTO (CIPC)	2025/667953/07
SEDE	72 York Road, Kensington Joanesburgo 2094, Gauteng, África do Sul
CONTACTOS	info@mozout.com · +27 78 823 3325 · www.mozout.com
HORÁRIO	2ª. à 6ª. Feira - 08 à 16 Horas Sábado & Domingo - Fechado

Versão online deste documento

RLX-BR-KB-4242 · v2026.10.18 · Gerado a 21/10/2026 13:28