

BASE DE CONHECIMENTO · DOCUMENTO PÚBLICO

WebSockets e Socket.IO: o que esperar no cPanel e num VPS

Node.js, Python e Docker

CÓDIGO

RLX-MZ-KB-1105

VERSÃO

v2026.10.18

PAÍS · IDIOMA

Moçambique · Português

MozOut Moçambique

RLX Holdings, S.A. · Grupo RLX Holdings

NUIT 401 191 607 · NUEL 105 045 974 · CAE 62022

<https://www.meu.mozout.co.mz/knowledgebase/1105>

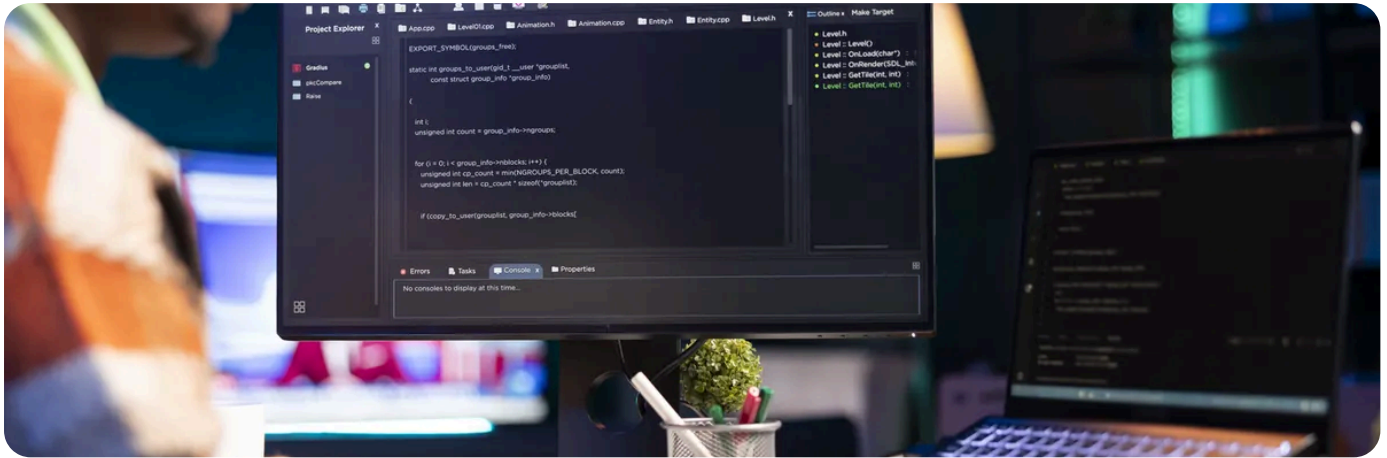
WebSockets e Socket.IO: o que esperar no cPanel e num VPS

Node.js, Python e Docker

Criado em 10 de outubro de 2026

Atualizado em 18 de outubro de 2026

4 min de leitura



Um chat, um painel que se actualiza sozinho, notificações em directo: tudo isso pede uma ligação que fica **aberta** entre o navegador e o servidor, em vez do pedido e resposta de sempre. Essa ligação chama-se WebSocket. **A resposta curta:** numa hospedagem partilhada, não conte com ela. As ligações abertas ocupam recursos da conta durante todo o tempo em que alguém está no site, e a conta tem limites de processos. Num **VPS** funciona bem, com o proxy bem configurado.

? O que muda de uma ligação normal

Um pedido HTTP normal dura milissegundos e liberta o servidor. Um WebSocket dura minutos ou horas: cada visitante ligado é uma ligação a ocupar memória e um «lugar» no processo. Dez visitantes são dez ligações abertas ao mesmo tempo, mesmo que ninguém escreva nada. Daí a diferença entre onde cabe e onde não cabe.

	No cPanel (partilhada)	Num VPS
Ligações abertas	Ocupam recursos limitados da conta; um servidor de websockets residente choca com o limite de processos.	Só as limita a memória e o processador do VPS.
Quem garante o proxy	O Passenger. Não temos como prometer que as ligações longas passem de forma estável.	Você, com o nginx: tem de passar os cabeçalhos de <i>upgrade</i> .
Socket.IO	Pode arrancar por <i>polling</i> (pedidos repetidos) quando o WebSocket não passa. Funciona, mas é um pedido atrás do outro.	Usa o WebSocket a sério.
Recomendação	Teste com poucos utilizadores; para tempo real a sério, mude.	O sítio certo.

Num VPS com nginx, passo a passo

- 1 **Ponha a aplicação a ouvir só no próprio servidor (`127.0.0.1`) e a correr sempre, com o PM2 ou o systemd: **manter a aplicação a correr com o PM2.****

- 2 **No bloco do nginx**, acrescente as linhas que deixam passar a mudança de HTTP para WebSocket:

```
location / {
    proxy_pass http://127.0.0.1:3000;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_set_header Host $host;
    proxy_read_timeout 3600s;
}
```

O `proxy_read_timeout` importa: por omissão o nginx fecha uma ligação que fique 60 segundos sem dados, e os websockets inactivos caem. O valor acima é um exemplo.

- 3 **No navegador, use `wss://`** (e não `ws://`) quando o site é HTTPS, no mesmo domínio. Um site em HTTPS que abra `ws://` é bloqueado pelo navegador.

- 4 **Recarregue o nginx** depois de validar a configuração: `sudo nginx -t` e `sudo systemctl reload nginx`.

- 5 **Confirme no navegador:** nas ferramentas de programador, separador «Rede», filtre por «WS». Uma ligação com o estado «101» é um WebSocket a funcionar.

Se o Socket.IO «funciona» mas é lento

, pode estar a cair para *polling* sem que ninguém repare: o separador «Rede» mostra pedidos repetidos em vez de uma ligação `101`. É sinal de que o WebSocket não está a passar, quase sempre por falta das linhas de *upgrade* no proxy.

Várias instâncias da aplicação?

Com mais de um processo atrás do proxy, os clientes do Socket.IO precisam de ficar sempre no mesmo processo ou de um adaptador partilhado: a documentação do projecto explica. Comece com um processo. Num VPS não gerido, a manutenção é sua: **até onde vai o nosso suporte.**

⚡ RECOMENDADO

Precisa de ligações em tempo real? Um VPS dá-lhe o controlo do proxy e dos processos.

[☰ Ver os servidores VPS →](#)**📁 Onde ver na sua conta**

Abra o **Uso de recursos** (Resource Usage) no cPanel: mostra os processos, a memória e o CPU da sua conta contra os limites, e marca quando um limite foi atingido. Quando o limite de processos é atingido, os novos pedidos são recusados e os visitantes podem ver o erro 508; quando a memória se esgota, as coisas falham. Um servidor de websockets residente mantém um processo e a sua memória ocupados enquanto houver alguém ligado, por isso chega depressa a esse tecto. Para ligações em tempo real o caminho é um VPS, que lhe dá acesso total por SSH e o controlo do nginx, ou um servidor Cloud. O VPS1 (1 vCPU, 2 GB de RAM, 40 GB NVMe) ou o Cloud1 (1 vCPU, 2 GB, 25 GB) é um bom ponto de partida, e pode subir até ao VPS6 ou ao Cloud6 (32 vCPU, 64 GB) à medida que as ligações crescem.

🔗 VEJA TAMBÉM[Nginx como proxy inverso à frente de um contentor](#)[Manter uma aplicação Node.js a correr num VPS com o PM2](#)[Portas e firewall num VPS: porque a aplicação não é alcançável](#)



MozOut 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@mozout.co.mz · +258 84 635 1111 · +258 86 635 1111 · www.mozout.co.mz
HORÁRIO	2ª. à 6ª. Feira - 08 à 16 Horas Sábado & Domingo - Fechado



Versão online deste
documento

RLX-MZ-KB-1105 · v2026.10.18 · Gerado a 21/10/2026 17:50