

FIELD NOTES FROM A PRESALES SOLUTIONS ARCHITECT

Webmail Returns 502 Only When People Are Actually Using It — The One Proxy Flag cPanel Forgot

A presales war story about a webmail login page that worked for one tester and failed for forty staff, the Apache error almost nobody reads, and the single ProxyPass flag I now check on every cPanel server.

Daanial Farrukh · 29 October 2026 · <https://quabin.com/cpanel-webmail-502-bad-gateway-disablereuse/>

The ticket had been open for two weeks.

"Webmail is unstable. Users get Bad Gateway. It comes and goes."

The server was a customer's cPanel box, a few hundred mailboxes, staff logging in through webmail.example.com in a browser. The hosting team had already done the usual rounds.

"We added RAM. We restarted Apache twice. We even moved the site off Cloudflare for a day. Nothing changes."

I asked one question.

"When does it fail?"

"Mornings. Nine, half past nine. By eleven it's fine."

"And when you test it yourself?"

"It always works when I test it."

That was the whole case, in two answers. One person testing: fine. Forty people logging in within the same ten minutes: 502. Nobody had tested forty logins. Nobody had looked at what sits between Apache and the webmail service on a cPanel server.

I asked for one line from one log, and the answer was there before the call ended.

The Trap, Named

That's the trap. A 502 Bad Gateway from webmail does not look like a cPanel bug. It looks like a load problem. So the team feeds the server more memory, blames the CDN, blames the firewall, and tests with one browser tab, where it never fails.

The failure is not in the server's resources. It is in how Apache talks to cPanel's own webmail service on the loopback.

What cPanel Webmail Actually Is

Here's the part people miss. On a cPanel server, Roundcube is not served by Apache the way a normal PHP site is. It is served by cpsrvd, cPanel's own daemon, on ports 2095 and 2096.

When a user types webmail.example.com, they are hitting Apache on port 443. Apache then proxies the request over the loopback to cpsrvd:

Browser → Apache :443 (webmail.example.com) → ProxyPass → cpsrvd 127.0.0.1:2095 → Roundcube

That ProxyPass is generated by cPanel from a template. In the stock configuration it looks like this:

```
ProxyPass / http://127.0.0.1:2095/ max=1 retry=0
```

Two settings matter. `max=1` means Apache keeps at most one pooled connection to cpsrvd per worker. And there is **no `disablereuse=on`**, which means Apache is allowed to reuse that one connection for the next request, from whoever it comes.

That is exactly the design that breaks under load.

The same proxy pattern is used on cPanel servers for all the service subdomains:

- webmail.example.com → 127.0.0.1:2095
- cpanel.example.com → 127.0.0.1:2082
- whm.example.com and webdisk.example.com → their own local ports

Webmail is where it shows first, because webmail is where many different people log in at the same minute.

Why the Shared Connection Breaks

cpsrvd is strict about sessions. Each connection it accepts is tied to the authentication it saw on that connection's first request. The function doing this is named plainly in its own log: `ensure_authn_has_not_changed_from_initial_request`.

Now put the two facts together.

- 1 Apache has one pooled connection to cpsrvd and is allowed to reuse it.
- 2 User A's request goes through it. Then user B's request is sent down the same connection.
- 3 cpsrvd sees a session that does not match the first request on that connection, and it resets the connection.
- 4 Apache was mid-request. It never gets a status line back. It logs an error and sends the browser a 502.

With one tester, user B never exists. The connection carries one session, cpsrvd is happy, and the page loads every time. With forty staff at nine in the morning, user B exists every few seconds.

Absence of errors in testing is not health. It is a test that never created the condition.

Why Nobody Sees It

Three reasons, and I have watched all three happen on the same ticket.

First, the obvious dashboards are green. Memory is fine. Load average is fine. The 502 is not a resource problem, so no resource graph will show it.

Second, the error is in a log almost nobody opens for a webmail complaint. It is in Apache's error log, as a proxy error, with a code that looks like noise: AH01102. And the matching line on the other side is in cpsrvd's log under /usr/local/cpanel/logs/, not under /var/log.

Third, the direct URL works. If you tell a user to go to `https://example.com:2096` instead of `webmail.example.com`, the problem disappears. The team reads that as "the server is fine, the user's network is weird". It is actually the single most useful clue in the case: port 2096 bypasses the Apache proxy entirely. If the proxy path fails and the direct path works, the problem is the proxy.

How to Actually Prove It

Step 1: Read the Apache error log for the proxy error.

This is a read-only check.

```
grep AH01102 /etc/apache2/logs/error_log | tail
```

If this is your problem, you will see lines like:

```
[proxy_http:error] [pid 12345] (104)Connection reset by peer: [client 203.0.113.10:51234]
AH01102: error reading status line from remote server 127.0.0.1:2095
```

Read it in words: Apache sent a request to cpsrvd on the loopback, and the connection was reset before a status line came back. The client IP is the user who saw the 502. The remote server is cpsrvd on 2095. Nothing else in the chain is involved.

Step 2: Read cpsrvd's side of the same moment.

```
grep "Session does not match" /usr/local/cpanel/logs/error_log | tail
```

The matching line is:

```
reset_connection("Session does not match first request")
```

That is cpsrvd saying, in its own words, that it received a request for a different session on a connection that already belonged to someone else, and it closed the connection on purpose. When you see this line and the AH01102 line at the same timestamps, the cause is proven. Not a theory. Proven.

Step 3: Confirm the generated ProxyPass is missing the flag.

```
grep -n "2095" /etc/apache2/conf/httpd.conf
```

Look at the webmail ProxyPass lines. If they say `max=1 retry=0` and nothing about `disablereuse`, you have the stock configuration that produces the fault.

Step 4: Confirm the direct port works while the proxy fails.

Ask one affected user to log in at `https://example.com:2096` during the failure window. If that works and `webmail.example.com` fails, you have the behavioural proof as well as the log proof.

How to Actually Fix It

The fix is one flag: `disablereuse=on`. It tells Apache to close the backend connection after each request instead of pooling it. `cpsrvd` then sees a fresh connection for every request, and the session check never fails.

There is a quick way and a permanent way. Do the quick way first if users are failing right now, then make it permanent.

Step 1: Quick fix, applied directly to the running config. This changes state.

```
sed -i 's#ProxyPass / http://127.0.0.1:2095/ max=1 retry=0#ProxyPass / http://127.0.0.1:2095/ max=1 retry=0 disablereuse=on#' /etc/apache2/conf/httpd.conf
```

Then restart Apache the cPanel way:

```
/scripts/restartsrv_httpd
```

The 502s stop immediately. But this edit lives in a generated file. The next time cPanel rebuilds `httpd.conf`, which happens on EasyApache changes, account changes and some updates, your flag is gone and the 502s return. That is why the next step matters.

Step 2: Permanent fix, through the template cPanel actually uses.

The webmail ProxyPass is generated from the main template, `ea4_main`, not from the SSL vhost template many people edit first. cPanel's supported way to change a template is to copy `.default` to `.local` and edit the copy. This changes state.

```
cp /var/cpanel/templates/apache2_4/ea4_main.default /var/cpanel/templates/apache2_4/ea4_main.local
```

Open `ea4_main.local` and find the webmail ProxyPass lines. There are two, one for the HTTP vhost and one for the HTTPS vhost, roughly 180 lines apart. Add `disablereuse=on` to both:

```
grep -n "2095" /var/cpanel/templates/apache2_4/ea4_main.local
```

After editing, rebuild the config from the template and restart:

```
/scripts/rebuildhttpdconf && /scripts/restartsrv_httpd
```

Step 3: Verify. This step is where most tutorials stop and where mine starts.

First, prove the flag survived the rebuild:

```
grep -c "disablereuse=on" /etc/apache2/conf/httpd.conf
```

You want a count of at least two, one per webmail vhost. Zero means the rebuild overwrote your change, which means you edited the wrong template.

Second, prove the fault is gone. Wait for the next busy window, or simulate it: have several people, or several browser sessions logged in as different mailboxes, hit `webmail.example.com` within the same minute. Then:

```
grep -c AH01102 /etc/apache2/logs/error_log
```

Note the number, wait through a busy period, run it again. If the count stops growing, the fix is in. If it keeps climbing with timestamps after your restart, the flag is not in effect on the vhost that is serving the traffic, and you go back to Step 2.

Plain takeaway: a fix you cannot see in the log after a rebuild is not a fix.

The Part That Is Not Your Fault

I want to be fair to cPanel here, because this is not a misconfiguration by the hosting team.

The stock template ships this way. I first saw the fault on a server running cPanel version 134, and it was reported widely after servers upgraded to 136.0.20. I checked the stock `ea4_main.default` on 136.0.23 and it still generates `max=1 retry=0` without `disablereuse=on`. This is a regression in the default template, not something anyone on the customer side typed.

That has two consequences.

First, you will meet it on servers that have never been touched. I have now confirmed the identical fault on two unrelated cPanel servers at two unrelated customers. Same version range, same AH01102, same `cpsrvd` reset line, same fix.

Second, freezing `ea4_main.local` has a cost. `ea4_main` is a core template. Once you have a `.local` copy, cPanel stops applying its own future changes to that template on your server. You will not get fixes or new directives that land in `ea4_main.default` later. For a small server that is acceptable; for a fleet, it is a maintenance debt you have to track.

The better long-term options are either a post-rebuild hook that re-applies the flag to the generated config, or a ticket with cPanel asking them to add `disablereuse=on` to the default. I would open the ticket regardless. The more servers that report it, the sooner the default changes and the `.local` copy can be deleted.

The Bigger Lesson — Four Checks Before You Add RAM

Whenever a web service "fails under load but works in testing", run these before you touch resources.

- 1 **Does the direct port work while the proxied name fails?** Try `https://example.com:2096`. If yes, it is the proxy layer, not the application.
- 1 **What does the proxy log say, in its own words?**

```
grep -i "proxy" /etc/apache2/logs/error_log | tail -20
```
- 1 **What does the backend log say at the same timestamps?** For `cpsrvd` that is `/usr/local/cpanel/logs/error_log`. Match the seconds.
- 1 **Is a connection being shared between users?** Any ProxyPass with a connection pool and no `disablereuse=on`, in front of a backend that ties sessions to connections, will produce exactly this pattern.

If all four point at the proxy, more memory will not help, and a CDN change will not help, because the failing hop is a loopback connection inside the same machine.

What I Tell Every Customer Now

In the next review, ask the hosting team one question:

"When webmail returns a 502, what does the Apache error log say about 127.0.0.1:2095?"

If the answer is "nothing, we've never seen a proxy error there", they are in good shape. If the answer is "AH01102, error reading status line", we are not done yet, and the fix is one flag away.

I write about the questions I ask in real customer rooms — the ones that change architectures and surface the problems no one else noticed. If you found this useful, drop your email below and I'll send the next one when it's ready.

FAQs

Why does cPanel webmail give 502 Bad Gateway only sometimes?

Because the fault needs two different users on the same Apache-to-cpsrvd connection within a short time. One tester never triggers it. A busy login window does. The intermittent pattern is the signature of a shared backend connection being reset, not of a server running out of resources.

What does Apache error AH01102 mean on a cPanel server?

AH01102 means Apache's proxy module sent a request to a backend and the connection was closed before a status line came back. On cPanel, when the remote server in the message is 127.0.0.1:2095 or 2096, the backend is cpsrvd serving webmail, and the reset is cpsrvd refusing a connection whose session changed.

What does `disablereuse=on` do in Apache ProxyPass?

It stops Apache from keeping the backend connection open for the next request. Each proxied request opens its own connection and closes it afterwards. That removes the chance of two users' requests sharing one connection, which is what cpsrvd's session check rejects.

Will my fix to `httpd.conf` survive a cPanel update?

Not if you edited `httpd.conf` directly. cPanel regenerates that file from templates. To keep the change, copy `ea4_main.default` to `ea4_main.local` in `/var/cpanel/templates/apache2_4`, add the flag there, and run `rebuildhttpdconf`. Then confirm the flag is present in the rebuilt `httpd.conf`.

Is this a cPanel bug or a configuration mistake?

It is the stock template. Servers on cPanel 136.0.20 and later generate the webmail ProxyPass without `disablereuse=on`, and the default was still unchanged at 136.0.23. A hosting team that never touched the config will still see it. A support ticket to cPanel is worth opening so the default gets fixed.

Does the same problem affect `cpanel.example.com` and `whm.example.com`?

Yes, the same ProxyPass pattern, `max=1` and `retry=0` without `disablereuse=on`, is used for the cpanel, whm and webdisk service subdomains. Webmail shows it first because many users log in at the same time there. If those other subdomains also return 502 under load, apply the same flag to their ProxyPass lines.

Read online: <https://quabin.com/cpanel-webmail-502-bad-gateway-disablereuse/>