FIELD NOTES FROM A PRESALES SOLUTIONS ARCHITECT

nginx "unknown directive http2" — the One-Line Fix, and Why apt upgrade Brings It Back

A field note about a mail server whose web side went fully dark on a fresh install — one directive nginx did not know, the version gap behind it, and why the fix silently disappears on the next apt upgrade.

Daanial Farrukh · 23 November 2026 · https://quabin.com/nginx-unknown-directive-http2-apt-upgrade/

The message was short.

"Webmail is down. The admin panel is down too."

The server was a self-hosted grommunio box on Ubuntu 24.04 — a proof-of-concept mail platform, freshly installed.

"Is the mail service itself running?"

"Yes. Only the web side is dead. Nothing on 443, nothing on 8443."

"What changed?"

"Nothing. We just finished the install and the packages are current."

I asked one question.

"When was the last time nginx actually started — not reloaded, started?"

Silence. Then: "We didn't check nginx. We checked the mail service."

That's the trap. A web server that refuses to start does not look like a web server problem. It looks like the application is broken — webmail, admin panel, everything at once.

What nginx Was Actually Saying

The first command is always the same:

```
nginx -t
```

And there it was:

```
nginx: [emerg] unknown directive "http2" in /usr/share/grommunio-common/nginx.conf:62
```

nginx is strict: one directive it does not recognise, and it refuses to start at all. Not "start and skip that line". Not "start with a warning". It does not bind a single port.

On this box every web service was an nginx virtual host:

- port 80 — the plain HTTP listener
- port 443 — webmail and the client endpoints
- port 8443 — the admin interface

One bad line in one included file took all three down together. That is why it felt like "everything broke" instead of "one thing broke".

The Version Gap Behind One Line

Line 62 was a single directive:

```
http2 on;
```

That standalone http2 directive arrived in **nginx 1.25.1**. Before that, HTTP/2 was switched on as a parameter of the listen line:

```
listen 443 ssl http2;
```

Ubuntu 24.04 ships **nginx 1.24.0**. Check yours:

```
nginx -v
```

So the vendor config was written for a newer nginx than the one the distribution provides. On 1.24.0 the word http2 at the start of a line is simply an unknown directive.

Here's the part that trips people up. Nothing on the server was "wrong". The package did exactly what its authors intended. The distribution did exactly what its maintainers intended. The two just disagree about which nginx you are running.

Why Nobody Sees It in Testing

Three reasons this slips through.

1. **The mail service stays up.** SMTP and IMAP do not go through nginx. Health checks that only test port 25 or the mail daemon show green.
2. **A reload is not a start.** If nginx was already running when the config changed and nobody restarted it, the old config keeps serving. The failure shows up at the next restart or reboot — often days later, and far away from the update that caused it.
3. **The error points at a vendor file.** The path is /usr/share/..., not /etc/nginx/.... People read "grommunio" in the path and start debugging grommunio instead of nginx.

Absence of a mail alert is not health. The web side can be completely dead while every mail check passes.

How to Actually Fix This

Step 1: Confirm it is this error, and nothing else

```
nginx -t
```

Plain takeaway: if the only [emerg] line is unknown directive "http2", this article is your fix. If there are other errors, fix those first — nginx -t stops at the first one it finds.

Step 2: Find every standalone http2 line

```
grep -rn "^\s*http2 on;" /etc/nginx /usr/share/grommunio-common
```

On this box it was one line in one file. Check anyway — some configs repeat it per server block.

Step 3: Back up, then comment the line out (this changes state)

```
cp /usr/share/grommunio-common/nginx.conf /root/nginx.conf.bak
```

```
sed -i 's/^\s*http2 on;/#&/' /usr/share/grommunio-common/nginx.conf
```

The #& keeps the original text and puts a # in front of it. You can see exactly what was changed, and you can undo it by removing one character.

Step 4: Test, then start

```
nginx -t
```

You want:

```
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

Then:

```
systemctl restart nginx
```

Step 5: Verify from the outside

This step is where most tutorials stop and where mine starts.

```
systemctl is-active nginx
```

```
ss -tlnp | grep nginx
```

You should see nginx listening on 80, 443 and 8443. Then open webmail and the admin panel from a browser on another network. A green systemctl line only proves the process exists; a page that loads from outside proves the service works.

What you lose by turning HTTP/2 off

Almost nothing. On this design nginx terminates TLS and talks to the mail backend over plain HTTP on localhost. HTTP/2 only ever existed between the browser and nginx.

- Mail clients over MAPI, Autodiscover, EWS and ActiveSync are unaffected.
- SMTP and IMAP never touched nginx in the first place.
- Webmail page loads are marginally slower over HTTP/1.1.

Why apt upgrade Brings It Back

This is the part that turns a one-time fix into a recurring outage.

The file you just edited lives under /usr/share/. Debian and Ubuntu protect configuration files only when the package declares them as **conffiles** — normally files under /etc/. A conffile you changed triggers the familiar prompt during an upgrade: keep your version, or take the maintainer's.

A file under /usr/share/ is not a conffile. The package owns it outright. So on the next update of grommunio-common:

```
apt upgrade → new grommunio-common → nginx.conf overwritten → http2 on; is back → next nginx start fails
```

No prompt. No warning. Nothing in the upgrade output tells you a file you edited was replaced.

Prove it on your own box. First, which package owns the file:

```
dpkg -S /usr/share/grommunio-common/nginx.conf
```

Then list what that package declares as conffiles:

```
dpkg-query -W -f='${Conffiles}\n' grommunio-common
```

If the nginx.conf path is not in that list, your edit will not survive the next upgrade.

Two details make it worse:

- **Reboots are fine.** The edit survives reboots, so a reboot test after the fix passes. Only a package update reverts it.
- **The failure is delayed.** If the upgrade does not restart nginx, the old config keeps serving until the next restart. The outage then appears days later, with no update in sight.

The fix is not wrong. The fix is temporary — and nothing tells you when it expires.

The Durable Fix: Run an nginx That Knows the Directive

The workaround removes a feature to match an old nginx. The durable fix goes the other way: run nginx 1.25.1 or newer, so the vendor config works unmodified and no edit is needed at all.

On Ubuntu that means nginx.org's own package repository, which carries current mainline and stable builds. The steps are in nginx.org's "Linux packages" documentation; in short, add their signing key and repository, then:

```
apt update

apt install nginx

nginx -v
```

Two gotchas I checked before planning this change:

1. **Where the config is included.** nginx.org's build reads /etc/nginx/conf.d/ and ignores Ubuntu's sites-enabled/ layout. grommunio wires itself in through conf.d/, so the layout is compatible. Confirm on your box before you switch:

```
grep -rn "include" /etc/nginx/nginx.conf
```

1. **The worker user.** nginx.org's package runs as user nginx. Ubuntu's runs as www-data. The PHP socket that webmail uses is owned by www-data. If you keep the new default, webmail answers with 502 Bad Gateway. Set this in /etc/nginx/nginx.conf:

```
user www-data;
```

Then the same verification as before: nginx -t, restart, ss -tlnp, and a browser from outside.

I want to be honest here: on this PoC the durable fix is planned, not done. The comment-out is what runs today, and the check below is what keeps it safe.

The Check I Run After Every Update Now

Until the durable fix is in, this is the routine after any apt upgrade on the box:

1. **Is the edit still there?**

```
grep -n "http2 on;" /usr/share/grommunio-common/nginx.conf
```

If the line starts with #, the fix is intact. If it does not, the package replaced the file — run Step 3 again.

1. **Would nginx start right now?**

```
nginx -t
```

1. **Is it actually listening?**

```
ss -tlnp | grep nginx
```

Run nginx -t after every upgrade, even when nginx was not restarted. It tests the config on disk, not the one in memory — which is exactly the gap that turns this into a delayed outage.

The Bigger Lesson

The specific error is about nginx and one directive. The lesson is broader:

1. **A file under /usr/share/ belongs to the package.** Edit it and the next upgrade wins. Check with dpkg-query -W -f='${Conffiles}\n' <package>.
2. **A vendor config can target a newer upstream than your distribution ships.** Compare nginx -v with the release that introduced the directive.
3. **One unknown directive stops nginx entirely.** Every virtual host on the box goes down with it.
4. **Test the config on disk after every update.** nginx -t costs a second and finds the outage before the next restart does.
5. **Write down every hand edit to a package-owned file.** A one-line note next to the server — what you changed, why, and how to re-apply it — is the difference between a five-minute fix and an afternoon of debugging the wrong application.

What I Tell Every Customer Now

When someone runs a vendor stack on a distribution nginx, I ask one question in the next review:

"Which of your config edits live in files the package owns — and what tells you when an update has undone them?"

If the answer is "none, all our changes are under /etc and in conffiles", they're in good shape. If the answer is "we fixed that once, it's fine", we're not done yet.

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

What does nginx "unknown directive http2" mean?

Your config uses the standalone http2 on; directive, which exists only in nginx 1.25.1 and newer. An older nginx, such as 1.24.0 on Ubuntu 24.04, does not know it and refuses to start. Run nginx -v to see your version.

How do I fix "unknown directive http2" quickly?

Back up the file, then comment out the http2 on; line and run nginx -t. If the test passes, restart nginx. On older nginx you can enable HTTP/2 the old way instead, with listen 443 ssl http2;.

Why did the error come back after apt upgrade?

The file was under /usr/share/, which the package owns and does not declare as a conffile. An upgrade replaces it without a prompt, which restores the http2 on; line. Check with dpkg-query -W -f='${Conffiles}\n' and the package name.

Does turning off HTTP/2 in nginx break anything?

On a reverse-proxy setup where nginx terminates TLS and talks to the backend over HTTP on localhost, HTTP/2 only affects browser-to-nginx traffic. Mail protocols and API clients keep working. Web pages load slightly slower over HTTP/1.1.

What is the permanent fix?

Run nginx 1.25.1 or newer, for example from nginx.org's own Ubuntu repository, so the vendor config works without edits. Make sure the new package reads the same include folder and set user www-data; if your PHP socket belongs to www-data, or you will get 502 errors.

Why did webmail and the admin panel both go down at once?

Both are nginx virtual hosts on the same server. When nginx refuses to start because of one bad line, it binds no ports at all, so every site it serves goes down together while the mail service itself keeps running.

Read online: https://quabin.com/nginx-unknown-directive-http2-apt-upgrade/