Christopher Ross
thisismyurl.com

EPISODE 3 · SITES I'VE NEVER SEEN

The 30-minute audit

FOR	anyone sitting on a procurement decision about enterprise WordPress hosting
FROM	*Sites I've Never Seen*, Episode 3 — "Enterprise WordPress: The Infrastructure Decisions That Cost Teams Later"
TIME REQUIRED	thirty minutes, on a staging environment, before signing anything

> This is the audit Walter and I describe in the second half of the episode. If three or four of these checks turn up cheap wins, the question shifts from *what tier do we need* to *do we need to upgrade at all*. That's a much smaller bill.

Before you start

You need access to:

- The site running on a staging environment that mirrors production (not production itself — these checks involve poking around)
- Admin access to WordPress
- Query Monitor installed and active (free, on the WordPress plugin repository)
- The host's status page or dashboard for the persistent object cache

- The last thirty days of error logs (your host should expose these; if they don't, that's its own finding)

If you can't get staging access in the next thirty minutes, the audit is still useful — but some checks will need to wait until you do.

Check 1 — The five slowest queries on your highest-traffic page

Open Query Monitor on staging. Load the highest-traffic page (usually the homepage, sometimes the most-visited product or article page — check your analytics if you're not sure).

Look at the Database Queries panel. Sort by time.

Note the five slowest queries. Write down their query time in milliseconds.

Decision rule: if the slowest query is over 200 milliseconds, the bottleneck is in the application or the data layer — not the infrastructure. Moving to a faster server will not fix it. The fix is upstream of the host.

Common causes of slow queries: - A plugin running an uncached `WP_Query` on every page load - A theme querying recent posts three times instead of caching the result - A meta query against a large `wp_postmeta` table without an index - A `taxonomy_query` joining four tables to filter twelve items

If any of these match, write the cause down. You'll need it for the procurement conversation.

Check 2 — The plugin justification list

Pull the active plugin list. WordPress admin → Plugins → Installed Plugins → Active.

For each active plugin, write down — in one sentence — what it does and why this site needs it.

Decision rule: any plugin you can't justify in one sentence is a candidate for removal. Mark it for review with the site owner.

Calibration: - Under 15 active plugins: healthy, move on - 15-25 active plugins: typical, review the marginal ones - Over 25 active plugins: structural problem; the plugin count is a symptom, not a cause

The plugin count itself isn't the problem. The plugin count *without justification* is the problem. A site running 32 plugins, every one with a clear reason, is healthier than a site running 14 plugins where six are forgotten.

Check 3 — Persistent object cache: connected and serving

Most managed WordPress hosts run Redis or Memcached for persistent object cache. Most have a status page or dashboard widget that confirms it's connected.

Find the status page. Confirm:

- Object cache is connected (not just "configured")
- Hit rate is over 90% — under 80% means something is invalidating the cache constantly, or the cache is silently failing

Decision rule: if the object cache is configured but the status page shows zero hits, the configuration silently failed during the last deployment. The cache appears connected but is doing nothing. This is one of the most common production issues on enterprise WordPress installs.

If your host doesn't expose object cache status anywhere, file a support ticket asking for it. Their answer tells you whether their "enterprise" tier includes operational visibility or just marketing language.

Check 4 — Patterns in the last thirty days of error log

Pull the error log for the last thirty days. Your host should expose this — usually under "Logs" or "Monitoring" in the dashboard.

Scan for patterns that repeat.

What you're looking for:

- PHP warnings or notices that fire on every page load (these are usually a plugin or theme calling deprecated functions; nobody filed a ticket because they're not user-facing, but they accumulate and slow the site)
- Database connection errors during traffic spikes (capacity, but maybe at the wrong layer — could be connection pool limits, not server CPU)
- 500-level errors clustered around a specific URL pattern (a buggy endpoint, a misconfigured plugin)
- Cron job failures repeating daily (background work that isn't completing — explains why scheduled tasks don't run)

Decision rule: the patterns that repeat in the error log are the production bugs nobody filed a ticket on. Each repeating pattern is a finding worth investigating before signing an upgrade contract.

Check 5 — The slow-query log

If your host exposes a slow-query log (most managed WP hosts do), pull the last seven days.

Look at the top 10 slowest queries by total time.

Total time = query time × number of executions. A query that takes 50ms but runs 10,000 times per day costs more than a query that takes 2 seconds but runs four times.

Decision rule: the top 10 by total time are your highest-leverage performance wins. Each one is a known, measurable, fixable thing.

Common patterns:

- A `SELECT *` on `wp_options` with autoload=yes (this is the most common one — options table grows past 5MB and every page load drags it)
- A cron job running a heavy query unnecessarily
- An admin-side query running on the front-end because the developer didn't check `is_admin()`
- An untimed REST API endpoint that's being hit by an external service every minute

---

What to do with what you found

After thirty minutes, you should have a written list of findings. Probably between three and seven items.

If three or four are cheap wins — a plugin to deactivate, a query to cache, an autoload option to clean up, a cron job to retire — the procurement conversation changes. You're not buying a faster server. You're paying for a half-day of cleanup work and keeping the host you have.

If two or three are structural — the database is genuinely overloaded, the application has a memory leak the team can't find, the traffic pattern has changed and the architecture doesn't match — the upgrade conversation is real, but now it's targeted. You know what you're buying and why.

If none of the five checks turn up findings worth acting on — congratulations, the site is well-tuned, and the question is whether your operational discipline matches its current state. Walter would tell you: that's the part the procurement conversation doesn't include. Discipline isn't a SKU.

When to bring this to a vendor

If you're sitting across the table from an enterprise WordPress vendor and they haven't asked to see your slow-query log, your plugin justification list, or your error log patterns —

the conversation isn't measuring what it's claiming to measure. The good vendors will welcome the audit because the deal that survives an honest audit is the deal that doesn't blow up in twelve months.

If you'd like a second set of eyes on what your audit turned up, send me a note (https://thisismyurl.com/contact/) — I read every one.

— Christopher

Built to go with episode 3 (https://thisismyurl.com/podcast/enterprise-wordpress/) of Sites I've Never Seen, and the source essay Enterprise WordPress: The Infrastructure Decisions That Cost Teams Later (https://thisismyurl.com/enterprise-wordpress-infrastructure/).

LISTEN TO THE EPISODE

Enterprise WordPress: The Infrastructure Decisions That Cost Teams Later

Built to go with episode three of *Sites I've Never Seen*. The episode is where the argument lands; this is what you take into the next meeting.

Listen to the episode **Read the source essay** **Send me a note**