

We Get Letters

Choice of Agonies

There is no correct solution for new production environments.

Dear Advice Person,

I have a chance to set up a brand new clean-room production environment with my choice of operating system and application stack! It's a once-in-a-lifetime opportunity, and I find myself overwhelmed by the options. I know you jokingly recommend running production on an abacus, if not your fingers. But seriously, how do I choose?

–Trapped in Analysis Paralysis

TAP,

The most annoying part of writing an advice column isn't coming up with the advice. I'm pushing 60. My lifetime of failures stands ready to assist your edification, if they thought you'd heed their warnings. I have more experience than the editorial board will allow me to publish, although the cowards have issued a categorical ban on any guidance containing more than ten percent profanity by weight.

It's not even the part where you think I'm joking. I speak the truth. Your brain, in a lamentable spasm of self-defense, elects to treat it as a joke.

No, the deeply annoying part about this gig is the blatantly fake letters.

Don't get me wrong. I am unquestionably aware that all your letters are fake. Some of you don't exert even the most miserly scrap of imagination to fig-leaf your chicanery. Consider this notably callow example. Setting up a new production environment from scratch? Presumably the datacenter is managed by the Kongamato and the equipment provided by Santa Claus because *this does not happen*. Everything is built on top of and inside of and out of everything else. Computing is a stew that's been bubbling on the stove since forever because your parents inherited it from a grandparent who inherited it from a great-grandparent who brought it over from the Old Country on the Titanic. Whenever the kettle runs low you fling something fresh in, like a turnip. Or an intern. Sometimes you carve out a chunk to squeeze into sausage casing or dilute into soup, but only the brazen call it a new thing.

You claim to have the opportunity for a clean-room installation, but the existence of clean rooms is a myth.

Sadly, TAP, your fake letter is the only vaguely relevant fake I've received this month, so we'll go with it—but don't believe for one NTP-calibrated sloppy second that you're getting away with anything.

Terry Pratchett famously said that "hope is the greatest gift." I would never dare argue with the greatest philosopher of the twentieth and twenty-first centuries. Rule of System Administration #20 reminds us that "hope prevents you from doing your job," however, and that's a good place to start.

The "perfect environment" myth pervades this shambles of a practice. If only the architects had chosen the correct solution! Then you become an architect and realize that there is no correct solution, only a choice of agonies. Choose your pain. It's no secret that I put all my data on ZFS, but that means I exchange a simple filesystem that reliably splats blocks on disks for a storage operating system with its own memory caching algorithms, dynamic block sizing, tunable resiliency, and the inevitable impressive pathologies. Should your virtual machine with one terabyte of disk require three terabytes of physical backing storage? That's entirely possible with ZFS' magic!

No, wait. It's "magic" if it's beneficial. It's "stupidity" if not.

Magic and Stupidity are the exact same thing viewed from opposite directions.

Griping about production problems is a hallowed sysadmin tradition. Hang out in a conference bar and you'll quickly recognize the shape of the discussion. "New systems take forever to deploy because we use Ansible!" "Updating software with Chef is a nightmare!" "We'd love that, but we use Salt!" Apache can't, nginx won't, and lighthttpd does everything except what's unquestionably necessary so stick a Caddy proxy in front of five different web servers and call it HTTP/3. Start from scratch and permit only Postgres with Python? Wonderful! Except every marketing department needs WordPress and that drags the entire PHP/MariaDB ecosystem in its wake and even with architectural carte blanche using words like "tainted" in meetings will get you dispatched to HR.

The key to a successful production environment lies not in choice of operating system. The application stack is, in and of itself, irrelevant. XMLRPC or SOAP or System V IPC? Who cares?

The success of deploying a production environment lies in one simple principle.

"Minimize dread."

That's it.

If managing, upgrading, or debugging a tool triggers dread, don't use it.

This isn't an excuse to stick with tools you know. Sysadmins must always learn new things. At some point, everyone condemned to this career gets dragged before the Gray-Haired Council and brusquely interrogated. If you are found approximately worthy the person serving as Root bangs the gavel, declares you a true sysadmin, and sentences you to a lifetime of forced education. Fortunately the conscious mind blocks out the horrific experience, leaving only a residue of underlying trauma.

The more you know, the more bad you perceive. Dread increases until it overwhelms either your career or your circulatory system.

If you cannot minimize the dread, transform it into boredom. Study the most appalling aspects of your environment. Practice with them. Invite selected dreads to dinner and have a consensual but thorough snuggle. Nothing is dreadful without reason. It's always tempting to declare that a particular piece of software or a protocol is dreadful because the designer was an idiot, but that mostly doesn't happen. Many commercial products are dreadful because the designer's purpose is to separate you from your money rather than solve any problems you face. But in the open source world dreadful tools that survive overwhelmingly do so by the lack of less dreadful alternatives. That dreadful tool covers an even more hideous gap, but understanding the gap allows one to transcend the dread.

If all else fails? Scripts and checklists, and procedures—preferably, procedures that allow you to delegate despair to some callow youth who has not yet had their summons to the Council. Exposing the true repugnance of a career in computing is the most responsible favor you can do to—uh, *for*—a beginner. Otherwise they might wind up writing bogus letters in an attempt to troll an advice columnist. Or, worse, develop hope.

Michael W Lucas (<https://mwl.io>) is the author of too many books, including **[Absolute FreeBSD](#)** and **[\\$ git commit murder](#)**. He is currently avoiding his parole officer while writing **[OpenZFS Mastery](#)** with Allan Jude.

Copyright © 2026 Held by Owner/Author. Publication Rights Licensed to The FreeBSD Journal. This work is licensed under a **[Creative Commons Attribution International 4.0 License](#)**.

3980 Broadway Street
STE #103-107
Boulder CO 80304
1-720-207-5142
info@freebsd.foundation.org

