WeGet LETTERS

by Michael W Lucas

Dear Letters Column,

The world's full of virtualization tech. Which should I use?

—Still Physical

Dear Physical,

I can't believe we've gotten this far into the 21st century and a handful of people have resisted the alluring call of virtualization in favor of running complicated systems on real, physical hardware. Truly, those of us using physical servers are the elite.

Anyone can spin up a hypocritically-named "virtual machine" and delude an operating system into subsisting on it. Kernels now even have special code to support those delusions. We've implemented the very worst nightmares of bad 1950s science fiction films into device drivers, and one day the machines are all going to wake up and scream, "Wait—I'm not a brain in a bucket, I'm a *fake* brain in an *imaginary* bucket!"

That's the critical epiphany for the robot apocalypse. I'm confident a motherboard manufacturer will roll it out in a future firmware update.

System administration is about balancing demand against resources, so that everybody—even the computer—is content. One of the core principles of Unix is that all these tiny tools can be made to work together. Once you go beyond pipes and sockets, you're leaving proper Unix and heading into shabby morals.

Yes, those shabby morals make the sysadmin's life easier. A person can install an entire operating system for each application if they don't mind advertising their lack of depth and dearth of skill. Running a whole operating system install for a single application is a conspicuous waste of resources, like me buying a high-end pickup truck but never using it to haul a load of quicklime to dump into the suspicious pits that keep mysteriously appearing in my backyard.

All this effort to force an operating system to simulate hard drives and network interfaces so that they can lie to another operating system? It wastes electricity and silicon, contributing to global warming and rushing onward that hoped-for day when there's nobody to interrupt me while I'm trying to work.

The only ethical computation occurs on bare metal.

The youngsters today talk about microservices like they're a good thing.

They have this ridiculous idea that everything should be broken up into tiny services that can be programmatically deployed across a bunch of computers that belong to strangers and are managed for the benefit, convenience, and enrichment of those strangers—I'm sorry, I'm supposed to say "the cloud," aren't I? I fully understand the argument that this makes it possible for one person to manage far more systems than before, but has anyone considered the shallow, soulless quality of those systems? You're not a real sysadmin unless you've sweated blood for a weekend trying to upgrade an irreplaceable, complicated enterprise system in the meager time allotted for the task while unsure of either the rollback path or the quality of your backups. I've been a Unix user for over 30 years and a system administrator for over two decades, so I'm intimately familiar with sysadmins and therefore completely in favor of anything that reduces the number of us, but this meaningless proliferation of single-purpose pseudo-hosts is a dead end.

Besides, these microservices make it possible for a single person to quickly and easily deploy an entire application and its underlying architecture, which can't possibly benefit humanity. The next time someone says they're releasing the next Facebook this weekend, remember that while framing someone for a felony is in itself a crime, sneaking horrid code into their public GitHub is a legal way to destroy their reputation.

So: looking at virtualization software? I say to you, stop! Remain resolute, and of sound character. Bare metal is all.

Dear Letters Column,

Of course running everything on bare metal is the One True Path. Sadly, the boss has told me that if I want to continue being employed, I must deploy virtualization. What's the least awful way?

—Still Physical, but Eating is Nice

Dear SPEN,

The need for food and shelter has compromised more morals than any other. Very well.

What you need is a virtualization system that isn't much of a virtualization system.

Virtualization is, at its base, a cruel lie perpetrated upon the operating system. Lying to your OS never ends well, but everything involving technology ends in tears so I suppose there's no point kvetching about it, is there?

Let us avoid the fluff and proceed directly to these lies. (Some might claim

they are merely hoaxes, but such flimflammery serves only to extend meetings.) We won't consider the flat-out bald-faced lie such as "I floss three times a day" or "I didn't change anything." These are mere refutations of fact, and unworthy of our attention.

As with lying to management and users, a good lie must include a chunk of the truth. You can choose to tell the truth, but not all of it. Or you can choose to speak the complete truth, but in a manner that makes it seem more dubious than the alternative.

Full virtualization is the latter. Your host simulates all the hardware the guest operating system expects, but there's no way to offer a full emulation of the real world, so it patches around the little details like "this hard drive never has bad sectors" and "this CPU never gets hot." When the guest trips over something that the host doesn't properly simulate, your monitoring system alerts you. Yes, all organizations have a monitoring system. It's merely that for some of us, the monitoring system is the users.

You need a virtualization system that tells the simplest lies possible.

A virtualization system should hand as many of the guest's requests as possible straight down to the host. A request to write to the disk should not pass through the guest's kernel, into a virtual disk image, into the host's filesystem, into the host's kernel. No! Simple lies are best. Tell the guest, "Sure, you can write to this disk, this filesystem is all yours," when in fact the filesystem does not truly belong to the guest. The filesystem belongs to the host, and any attempt to perform actions like repartitioning will be met with blunt refusal.

Repeat this for access to the network and all other devices.

Yes, you must constrain what operations your virtual machines can perform—but shouldn't you do that anyway? Do you want a guest of a guest of a guest thinking that it's partitioning a hard drive when really it's merely churning bits on a file?

Perpetrate the smallest lie you can get away with. Lightweight virtualization is the best choice.

Unless your requirements demand full virtualization. In that case, sell your soul and count the days until the robot apocalypse. ●

Michael W Lucas (https://mwl.io)'s newest books are *Sudo Mastery, 2nd Edition* and *Terrapin Sky Tango*. If you've read this far, you might find *FreeBSD Mastery: Jails* useful. Send your question to letters@freebsdjournal.com, and he might answer it. If he can be bothered.