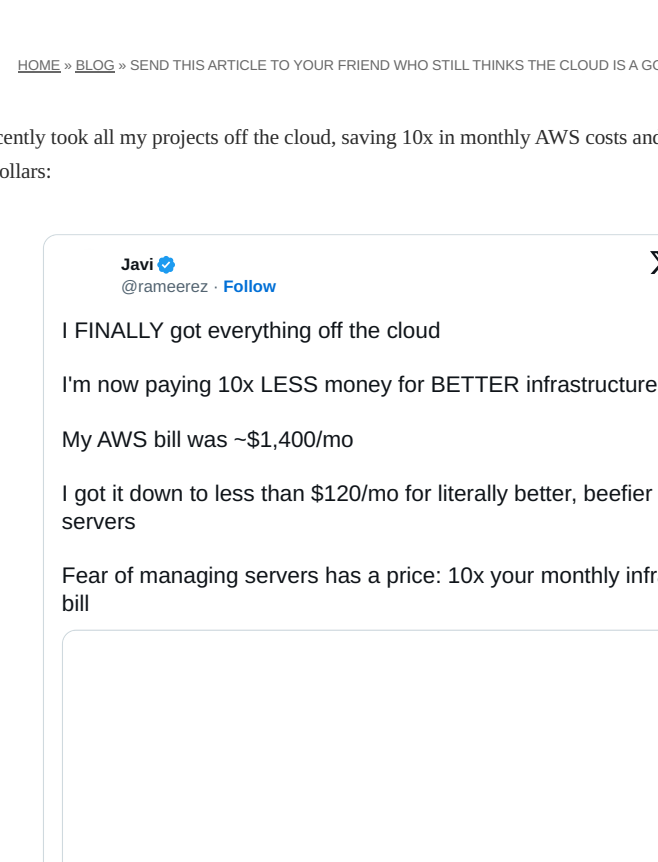
I'm rameerez, maker of PromptHero, Hustl, Jobician, and other products that are used by millions of people across the globe. My work has been featured in The Washington Post, The Wall Street Journal, Vox, Fast Company and others.

Send this article to your friend who still thinks the cloud is a good idea

Oct 4, 2024 · 22 min read

HOME » BLOG » SEND THIS ARTICLE TO YOUR FRIEND WHO STILL THINKS THE CLOUD IS A GOOD IDEA

I recently took all my projects off the cloud, saving 10x in monthly AWS costs and thousands of dollars:



Javi
@rameerez · Follow

I FINALLY got everything off the cloud

I'm now paying 10x LESS money for BETTER infrastructure

My AWS bill was ~$1,400/mo

I got it down to less than $120/mo for literally better, beefier servers

Fear of managing servers has a price: 10x your monthly infra bill

2:12 PM · Oct 2, 2024

6.3K Reply Copy link

Read 389 replies

This tweet got viral, and I found out **two things**.

One – many developers were tempted to do the same, and therefore curious on how I did it myself, so I wrote a blogpost explaining exactly how I exited the cloud.

And two – the most important one: the majority of people commenting were not only **not tempted** by the idea, but **highly confrontational** about it.

Interesting behavior –I thought– because I genuinely believe I did something *good* from both a technical and business point of view: I managed to decrease costs 10x while getting a 2x performance speedup for free, and freeing myself from vendor lock-in – who wouldn't want that for their business?

§

Shameless plug: I'm building ***RailsFast****, a Ruby on Rails SaaS template designed for AI and vibe coding. It provides a solid and secure foundation for you to build an AI business on top of it. It can be run on a $3/mo server for thousands of users, and it comes with things like automatic backups out of the box. I've built it on top of everything I've learned from creating and scaling PromptHero to 4+ million visitors a month, and making Ruby gems with over 100k+ downloads. It's still early, so get your discounted pre-sale here.*

§

I have more money in my pocket, I'm more free, and I managed to do so while providing a better service for my users.

Well, turns out many people had strong opinions about it.

I'm sorry, but you've been scammed

You've been scammed, likely not out of your own money, but your employer's money – and that's why you don't give a crap.

See, I noticed something interesting. Most people complaining about what I did happen to have "devops", "cloud engineer", "serverless guy", "AWS certified", or something similar in their bio.

None of these people have any skin in the game.

They don't run their own projects in the cloud, they don't pay a single dollar of the cloud bill at the end of the month, and they couldn't care less about their employer's AWS bill. They don't *feel* the **pain** of being scammed out of thousands and thousands of dollars by cloud providers month after month. With someone else's money, everyone's brave.

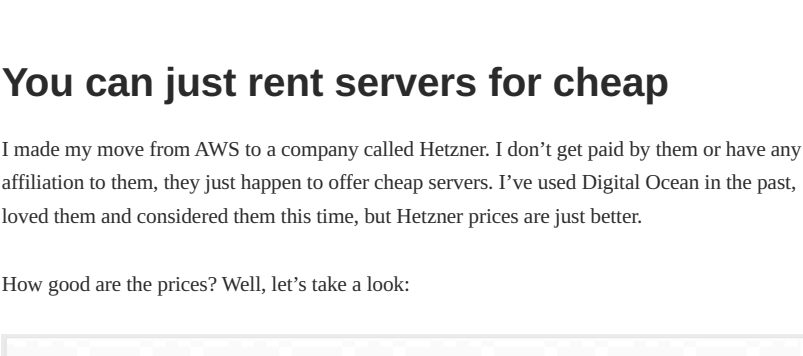
To them, it's way too convenient to be on AWS: not only it solves their problem, but it's also a shiny object. It's technically complex, it makes them look smart in front of other devs, it creates dependencies and lock-in effects so they're not easily replaceable as employees, and most importantly – being on the cloud yields them a fat paycheck at the end of the month. They don't have any incentive to make the most efficient decision for the business they work for, on the contrary – they depend on the business' infrastructure being as complex and convoluted as possible, or they'll be out of a job.

So don't count on any of them telling you the truth – the truth that **servers are not actually expensive**.

You can just rent servers for cheap

I made my move from AWS to a company called Hetzner. I don't get paid by them or have any affiliation to them, they just happen to offer cheap servers. I've used Digital Ocean in the past, loved them and considered them this time, but Hetzner prices are just better.

How good are the prices? Well, let's take a look:



Ampere® Altra® Q80-30
RX170
Usually available in a few minutes
CPU 80 cores / 80 threads @ 3 GHz
Generation: Altra
RAM 128 GB ECC DDR4 RAM
Drives 2 × 960 GB NVMe SSD Datacenter Edition
Location
Information IPv4 ECC
€169.00 max. per month
€0.2708 per hour
setup fee €79.00
Order
Details
excl. VAT

Ampere® Altra® Q80-30
RX170
Usually available in a few minutes
CPU 80 cores / 80 threads @ 3 GHz
Generation: Altra
RAM 128 GB ECC DDR4 RAM
Drives 2 × 960 GB NVMe SSD Datacenter Edition
Location
Information IPv4 ECC
€179.00 max. per month
€0.2868 per hour
setup fee €79.00
Order
Details
excl. VAT

You can rent bare-metal servers with 80 cores for under $190 a month. I repeat: **eighty cores for one-hundred-ninety dollars a month**.

For comparison, compute-heavy instances in AWS of the C5-C6 families with comparable number of vCPUs cost between $2,500 and $3,500 a month – which is between 13x and 18x more expensive. Sure, you can get reserved instances, and that brings the price down to ~$1,300/mo, which is still 7x more expensive, plus you'd need to pay $46k upfront and be locked to a 3-year contract. I don't know, Rick…

Say you don't need such a big server. You don't need, or you don't want, to pay for the 79EUR one-time setup fee. It's understandable, you only want to run your little projects, or you want a bit more flexibility. Let's look at the VPS options:

You still can get a 48-vCPU machine for around $300/mo, without setup fees, without 3-year contracts, and with full flexibility to scale it down if needed.

Or a more-than-enough 8-core, 32GB RAM machine for around $50 a month that will probably be able to run **all** your projects at once, as long as they're not getting literal millions of requests per day.

You can buy servers for even cheaper

But say this is still not cheap enough for you.

You're not worried about monthly costs, you're looking to save *for the long run*. Well, turns out buying your own servers is even cheaper in a long enough time frame:

Source: The Server Store

You can buy a decent rack-mounted server, with 44 CPUs, 256GB RAM, 2TB NVMe SSD and a bunch of other features for under $1k. Think about it. You could probably run your app for *years* – for less than what you probably pay a month in cloud services.

Now, of course, you can buy the server but you still to account for the electricity costs, internet costs, cooling costs, rack costs, etc. I'm not going to get to that level of detail. The bottom line is that good servers are *cheap*.

You can also choose to rent rack space in a datacenter, and they'll provide you with reliable electricity, internet, cooling, security, etc. You can look for datacenters in your area using websites like DatacenterMap:

The way it works is: you call the datacenter and they provide you with either shared space or a whole secure room for your servers, called datacenter cage, like this one:

And then you buy and ship your racks, and switches, and cables, and servers to the datacenter, load all the boxes in a little warehouse cart, transport everything into your cage and start building the racks and putting your servers in them:

BUT – all of this is WAY too complex for small developers. You'd probably need to hire an engineer to help you set everything right, you'd need to be in sales calls with the datacenter to negotiate the price and all that – it's way too cumbersome. Not worth it.

I'm just documenting it so you get a notion of how things actually work. I've never done this myself, it's usually aimed at medium-to-big companies and enterprises that want to build their own infrastructure, and it's pretty unusual that small teams do this.

Datacenter operations are quite interesting, I recommend you go watch some videos to get a better understanding of how the internet actually works. This is literally how AWS works under the hood. It's all datacenters. The whole internet is just cold, noisy rooms filled with metal boxes. Always has been.

Now, instead of renting an entire datacenter cage, you can just rent some available rack space in a shared rack, and then place your server there.

But you'd still need to physically go and install your server – and the end result is in practice THE SAME as just renting an already mounted server off a company like Hetzner, as I already described. And Hetzner takes care of replacing server parts when they go faulty, and they take all that hardware weight off your shoulders, quite literally.

bUt ISn'T tHat sTilL tHe clOUD

One of the most common critiques I got while I was trying to make the points above was something along the lines of: "but isn't that still the cloud?", "you didn't actually exit the cloud, you just changed cloud providers"

Look, first of all, you're as unique as the other 1000 peanut gallery enjoyers that have made the same astute observation before you. Congratulations. But you're absolutely missing the point.

The point here is that there's value in managing your own servers VS just accepting the cloud as a given truth. The point is you can start paying 10x-100x less money by running a little Linux machine instead of using managed AWS services. The point is most developers are scared of servers, and they just need a little push and encouragement to realize they can just set up their own servers for cheap to replace money-burners like RDS. The point is almost no one in the real world needs the cloud's expensive managed services, and can do just fine with regular software running on a couple Linux boxes.

At the end of the day, the only hard truth is whether you managed to keep more dollars in your pocket or you gave them away to Amazon's stockholders' pockets.

The whole debate of "is this still the cloud or not" is nonsense to me. You're just getting lost in naming conventions. VPS, bare metal, on-prem, colo, who cares what you call it. You need to put your servers *somewhere*. Sure, have a computer running in your mom's basement if that makes you feel like you're *exiting the cloud more*, I'll have mine in a datacenter and both will be happy.

I don't care if the server is physically in my building. I don't need to touch it, I need to `ssh` into it. In fact, if I had bought a physical server instead of renting one off Hetzner, I would still need to rent rack space to host it. At which point, you're just doing the exact same thing as renting a server off Hetzner, only more expensive and less convenient if you're a small developer.

It seems to me that people pushing for the "this is still the cloud" mindset are essentially gatekeepers. "You're not doing it correctly, you need to purchase your own switches, build your own rack, hook the ethernet cable yourself…" – boss, how far do you wanna take this argument? Do you etch your own circuits too? This whole thing is like saying "I'm the only one who knows how to do it right and you're not puritan enough to belong in this exclusive club". It's not only toxic, but it's also the most lazy argumentation one could make. It's choosing to get nuclear over a superficial nuance, instead of trying to refute the central argument.

The central argument is servers are not actually expensive, and whoever is telling you otherwise is probably trying to sell you a cloud managed service.

Companies like AWS or Vercel are charging you a 10x to 100x markup.

You're not using the cloud wrong, it's just expensive

They will try to gaslight you.

They will not only try to convince you you're not exiting the cloud, but they'll go as far as gaslighting you. They will make you believe you were not smart enough to *get* the cloud.

"You're using the cloud wrong", "It's expensive because you're doing it poorly", "You just need to know how to use the cloud".

You will hear a bunch of crap from people that have literally never tried the alternative. People with no real hands-on experience managing servers for their own projects for any sustained period of time. People trying to offer expert-level advice but that end up just spewing out generic, bland, off-the-shelve clichés and vague notions about how they think the cloud should work. People not only without skin in the game, but whose first job was straight into the cloud. People that go literally silent when you ask them: "so, what have you personally built lately, how many users does it serve, and how much are you paying for it?" They just don't know better, and they'll try to save face by defending themselves like a cat with its back against the wall – even if that means gaslighting you.

I'm definitely not an expert but I know my way around AWS. Hell, I've even studied AWS certs. I made sure my infra was not overprovisioned. I made sure I was not paying for services I was not using in regions I might have used only once or twice. I had already spent countless hours optimizing AWS costs, and even had some success in doing so (I even managed to cut my monthly AWS bill in more than half last year – it was north of $5k/mo before!)

I already did all that. I do know about serverless computing. I do know about reserved instances. I had reserved instances, in fact. (Don't purchase reserved instances, they only make the problem worse: they create vendor lock-in, and they essentially go against everything I've been trying to argue. The least thing you want when you're considering getting off the cloud because it keeps getting more and more expensive is to lock yourself into it with a 3-year contract. Yes, I also do know you can resell RIs, thanks, my argument still stands).

I tried all that, I tried as hard as I could, and the only conclusion I arrived to after many, many frustrations was that the cloud is simply way too expensive.

You can try yourself, by all means be my guest and try to prove me wrong, but I've already been there, done that, and I'm pretty sure of what I'm saying. So I'd suggest you just don't listen to the fearmongering cloud cult folks that keep repeating gaslighting mantras like a broken record.

You won't make friends proposing all this

All this backlash was interesting. Why do all these people care if I save more money or not?

I arrived at two main conclusions: 1) their livelihood depends on it; and 2) because of this fear, the conversation turns irrational.

One – their livelihoods depend on it.

If they're wrong, and if I and more people like me manage to convince enough people that they're wrong, they may be out of a job soon.

It's as simple as that.

We come from a decade-long trend where the cool thing was to build things in the cloud, which in turn created thousands if not millions of "devops" and "cloud engineers"; and we're now switching to a trend where the cool thing is to keep things off the cloud, which may terminate the careers of many cloud experts. It's just another cycle, but people get understandably very defensive about it.

I'm sure many developers secretly know the cloud is not as good as they initially promised, and they hold that conflicting thought in their head day in and day out as they go about their job. There's just no way those bright engineers look at their employer's AWS spending realizing it's 10x what it should be and go like: "ah, this is fine".

But no one wants to do the work, and risk getting their managers upset and getting thrown under the bus. They don't want to risk the political cost of not being in the same page as the most vocal person in the team. Or they just don't know better. Many entered the industry after the cloud was already mainstream. People used to run their own servers, it used to be absolutely normal and just the thing to do, but Amazon's marketing team managed to flip the market, so newcomers haven't touched anything but cloud in their entire careers – and AWS evangelists convinced whoever seniors were left.

Two – it's not a rational discussion. As a result, developers that are currently balls deep into this whole cloud thing have developed some sort of Stockholm syndrome about the cloud. It's essentially sunk cost fallacy, but it turns them into very irrational creatures.

AWS has a strong cult-like following. They teach you the doctrine via certifications where they make you memorize actual sales pages and product offerings, they have literal *evangelists*, they instruct you to stop listening to the rational voices around you that push for pragmatism instead of dogmatism, they instill fear of the outside world in you (servers are dangerous! things don't scale!), they make you adopt "cloud engineer" as your whole new identity, they make it really easy to get in but really difficult to get out, etc. I'm not saying it's an *actual* cult, of course, just that things get irrational pretty quick. More so when people's livelihoods are on the line.

The end result is AWS followers, whenever confronted, get caught up in dogma and just start throwing irrational arguments left and right, repeating the AWS sales landing pages' talking points one by one, and don't even stop to think about what they really need or if those things are even of use to them. They got tricked into believing something, and once you touch upon belief systems, people get irrational.

I could never win this irrational discussion with rational arguments. It doesn't matter how good my optimizations were. I could have cut costs 100x instead of just 10x – or I could have claimed something outrageous like I got to run all my infrastructure for just $1. It just doesn't matter.

These people would still be ranting and arguing that I'm doomed because now I don't have things like "infinite scalability capabilities" or "automated failovers with automated replica recovery". These are things that I've never needed or used, things that I'm sure 99% of people in the thread never needed or used either, but that they throw at you in a vain attempt to build an argument.

When people are cornered, arguments quickly turn into dogma – and fear:

Hackers will break in. Your servers will get full of malware. Your hardware will fail and you won't be able to restore it. Your switches will break and your customers will leave while your server is down. You will need to pay someone to maintain and upgrade your server, there's no way around it! And you need a plethora of busywork to "do it properly": failure scenarios! Recovery plans! Managing availability! Infosec! Monsters, spiders and zombies! All of them, coming for you! Be afraid, very afraid! How dare you not be afraid of all this scary stuff!

Well, either the last 15 years of my life –and the lives of most *webmasters* on the internet– have been a lie, or you don't actually need any of that to run a server online.

How did we get here?

The majority of devs are clueless, or have forgotten, about how we got here.

Everyone used to run their own servers. Either VPSs, or hosting services, or bare metal in datacenters, or in a dark room in their companies, or at home. Everyone was familiar and comfortable `ssh`ing into machines.

I remember very clearly how the cloud marketing psyops campaign started in the early 2010s. It was a deliberate move by companies to try to shill their enterprise technology to early-stage startups, trying to get them locked in as early as possible so they could milk them as they raised rounds.

I remember when AWS started to give out credits specifically for startups only. They would literally make the rounds, going from startup accelerator to startup accelerator, trying to get everyone on board.

The trick is easy: you make it extremely cheap (free!) for startups to build on your infrastructure, then make everything extremely expensive as they grow, and enjoy the show because they're so locked in most will struggle to escape the ecosystem by them.

AWS was not the only one. I remember attending an IBM cloud event in 2014. I was the CTO of a small startup at the time. They were very specifically targeting startups; in fact, we got the invitation via our startup accelerator.

We ran everything on Heroku at the time, and it worked just fine. I remember thinking: "what the hell is all this cloud stuff and how do I use it?" I vividly remember feeling like they were trying to sell us something that was not designed for us. I spent some time looking into it and just couldn't wrap my head around this whole cloud thing. All of it sounded so alien to us.

But all these companies poured literal millions upon millions of dollars into this cloud shilling campaign over the following years, tricking early startups into adopting enterprise technology.

They ended up being successful at it – and the aftermath is the current state of web development in 2024. Zero interest rates through most of the last decade definitely helped in getting us here.

There is now a counterculture movement, mainly led by @dhh and the Rails community, and this feels like something fundamentally fresh, right, and aligned with the reality of MOST software businesses on Earth.

You don't need the cloud

Some people are absolutely out of touch with what most software businesses look like in the real world.

They think in terms of Fortune 500; they truly believe enterprise is the norm. They think the average business needs all the bells and whistles the cloud has to offer: high availability, multi-zone replication, automatic failovers, distributed Kubernetes clusters…

The reality is only a teeny tiny fraction of all software businesses need something like this.

Most businesses are and will always be small, by a simple rule of power law.

Representation of a power law distribution. Actual phenomenon depicted is mostly unrelated, but it shows what a power law is: a very small number of companies are actually big – while the vast majority of companies in the long tail are small. In the United States, small businesses account for 99.9% of all businesses. In the European Union, SMEs (defined as having fewer than 250 employees) represent 99% of all businesses. Similar patterns are observed in many other countries around the world. Your little app is (most probably) not part of that 0.1%.

Most business are small, and the ones that need actual computing power can do incredibly well without the cloud up until a very high point. Scaling vertically can get you very, very far nowadays.

Most devs wildly overestimate scaling requirements. They have such a low bar for what "high traffic" means.

Here's a reference point: my current two-server setup serves millions of requests a day for millions of monthly visitors. So is the case for many other indie makers, like @levelsio, who even managed to get everything down to one single server. Most devs have never tried running a project of their own, with actual users and actual production traffic, on a single server – and it shows.

Devs also wildly overestimate other technical requirements. Again, only a tiny fraction of software businesses need the bells and whistles. Of course they exist, but they're rare, and they usually have very good reasons for their technical decisions.

Like Netflix, needing to transcode and stream enormous amounts of video to customers all over the world. That's where you need distributed systems, CDNs, edge computing, all of that stuff. Your little app with one thousand users that just sends some JSON objects around does definitely not need that.

I feel like most devs have this magical notion in their heads that their project is something like Netflix. It's wishful thinking, and I get it – you want to be as successful as Netflix. But it makes you make the wrong technical decisions, and all of a sudden they think they need to have distributed servers all over the world because their users will somehow notice a few milliseconds difference in latency when they tap a button. It's wild.

I remember a random conversation I had with a young entrepreneur a couple years ago. He ran a mobile app. I don't remember what exactly it did, but it was of the "upload things and follow people" category of apps. I asked: "what's the biggest challenge your facing in your startup?", and he replied: "well, that'd be estimating monthly AWS costs. They're high and unpredictable". I held my breath and took a step back, as one does when someone makes a claim bold enough to make you reconsider if you might be in front of a billionaire or a super-successful entrepreneur you've never heard about, and respectfully asked: "how many users does your app have?". "Oh, about a thousand now" – he replied.

Bro, just get a server. You don't need the cloud.

Your servers are going to be okay

Something I discovered is that a lot of people have magical ideas about how datacenters actually work.

They think servers in datacenters are fragile and volatile. Like things that can just vanish into thin air.

Someone even thought a lightning strike can take down an entire datacenter and nuke your business out of existence.

The reality is modern datacenters already account for all these problems and are equipped with many protections: not only against things as mundane as lightning, but against pretty much anything that can compromise uptime.

They have plenty of redundant systems: redundant power sources, redundant cooling, redundant internet connections, redundant fire suppresion systems, redundant security sustems and tons of physical security… Everything in a datacenter is designed with resiliency and redundancy in mind (N+1 redundancy at minimum, sometimes 2N) to guarantee uptime.

These are mostly fear-driven opinions, the result of a successful cloud marketing and psyops campaign. Where do you think AWS stores their machines, in a magical place under the rainbow? And how exactly would AWS machines be the only ones magically protected against problems that other datacenters also have?

Disasters can happen (OVH 2021), sure, and you should have backups to recover from them – but in my experience of ~15 years running servers I think they're rare, and I've never had any downtime of more than a few minutes.

Your server is probably going to be okay.

You won't have to spend your days managing servers

Anyone who has managed servers for long enough knows you spend most time in the initial setup; then servers tend to be relatively stable.

Hardware failures are relatively rare, and once a server is up and running it usually runs flawlessly for years without much intervention.

Managing your own servers is not a full-time job.

You don't need to employ a 5-person devops team. You don't even need to hire a server guy: you can just do things yourself! It's not that difficult.

Claude and ChatGPT usually have a good understanding of Linux systems and how to manage them, ask them for help.

When I posted the tweet, people thought this was my first time running servers and claimed I was overly optimistic in what running a server actually means.

It's difficult to say these things without coming across as arrogant, but I've been managing servers since 2006.

I started, as many did, editing PHP scripts and uploading them to my FTP server. I first had to learn how to install WordPress, then ventured a bit and started editing WP templates, then everything else followed.

It was an invaluable experience for me. It taught me the basics.

It taught me what Linux was, how to navigate it, and by 2007 I was requesting Canonical for Ubuntu CD-ROMs that would arrive in the mail and that I would use to install in a partition in my parents' computer to learn more about Linux.

Those early experiences taught me the basics of web development; everything else is built on top of it. Which leads me to my next point:

You will learn how to use Linux, and that's a good thing

I think the new generations of devs (genz devs, alpha devs, etc.) are absolutely out of touch with the hardware that runs their software.

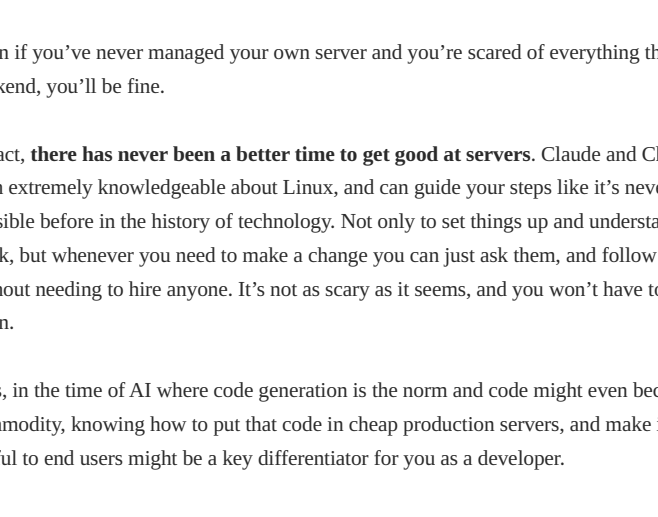
They lack these kinds of foundational experiences.

They were born in an era where a random guy on Youtube shilled them one specific vendor and taught them to run one very specific command that magically solved all their infrastructure problems. It's only reasonable they have magical assumptions about what servers are and how they work.

They rant and rant about how you can just do things "serverless" without realizing they're just running their code in many different boxes. Of course, many of them go on to learn more about Linux and servers, but the average bootcamp grad, let's say, lacks the hands-on Linux experience that FTP hackers would have had 20 years ago.

I'm not making moral judgements about this: I'm not arguing it's good or bad – it is what it is. The current state of web dev.

But the result is companies like Vercel are making literal millions of dollars by capitalizing on this new generation of developers that generally don't know what they're doing and have never ran a server of their own in their life.



Daniel Lockyer
@DanielLockyer · Follow

Vercel pricing is absolutely insane, it blows my mind

Data transfer starts at $150/TB with 1TB included

Hetzner gives 20TB free, and then it's €1.19/TB

So 20TB on Vercel is $2850, on Hetzner it's free (included with VPS)

And then you have to pay extra per request?

Perhaps Show more

9:19 PM · Oct 3, 2024

2.3K Reply Copy link

Read 90 replies

Ignorance has a double price: being ignorant in itself; and the price you pay for people exploiting that ignorance of yours. Not knowing how things work will actually cost you money.

You'll be fine

Even if you've never managed your own server and you're scared of everything that smells of backend, you'll be fine.

In fact, **there has never been a better time to get good at servers**. Claude and ChatGPT are both extremely knowledgeable about Linux, and can guide your steps like it's never been possible before in the history of technology. Not only to set things up and understand how they work, but whenever you need to make a change you can just ask them, and follow the steps without needing to hire anyone. It's not as scary as it seems, and you won't have to do it that often.

Plus, in the time of AI where code generation is the norm and code might even become a commodity, knowing how to put that code in cheap production servers, and make it actually useful to end users might be a key differentiator for you as a developer.

You will now need to worry about things like security, sure. Don't listen to those arguing that running your own server is somehow less secure than running AWS's servers, as if EC2 instances were magically protected against hackers or something.

The reality is it's not that difficult to set things up right, just follow my guide and setup scripts for example. There are many devs, more experienced than you, doing things in production way worse than what you could do when equipped with AI. Ask ChatGPT how to harden your Linux server and follow best security practices (like: don't use password auth, only strong ssh keys) and you're 90% there. Just take the plunge and go for it.

Something like Cloudflare can get you really far, too.

After locking your Linux box you can run Cloudflare on top of everything for an extra layer of protection: proxy the IP of your server on their DNS so you don't expose it and you're golden. And you get DDoS protection, edge caching, and a top-tier DNS for essentially free.

You'll be fine.

P.S.: Follow me on X! You can also read other stories I've written, and subscribe below to get an alert when I publish a new post:

Type your email...

Verify you are human CLOUDFLARE Privacy · Terms

SUBSCRIBE

No spam ever, unsubscribe with one single click.

Type your comment here...

AL
OCT 14, 2024

I started out my career in a big on-prem company (80k+ servers with all the trimmings) as a sysadmin, then became a developer eventually moved companies and joined the cloud world. I've been through a few cloud (and multi cloud!) companies at this point as roles across devops to backend developer. And I do indeed currently have both devops and cloud in my title. I say this only to point out that I feel I have experience both on-prem and in-cloud, as both ops and dev.

My 2c in 2s; AWS / GCP is absurdly, almost comically expensive. Hosting your own servers is not hard. You don't need to pay someone for it unless you really don't want to and are feeling flush with cash. (You do you I guess).

Networking can be a bit more involved but assuming you just want to break your servers up between vlans (dmz / prod / dev / logging) or some such you should be fine. Cloudflare tunnels are an excellent option for ingress.

Storage is dirt cheap. Even if you want some fancy network storage NFS has you covered; it's old and battle tested tech and is likely fine for 99% of companies.

Installing postgres and setting up replication to a secondary is not even a days work. Postgres on a big server can take insane punishment, the bigger problem is how derpy the queries are you're throwing at it.

Hell even kubernetes baremetal isn't too onerous, though it's definitely more fiddly to configure.

A lot of features the cloud has I've seen either not used, or enabled for "compliance" and are not useful in practice. Everywhere has cloud cost programs to cut back on waste, there are so many companies now that make their own living feeding off this alone it's crazy.

Sadly though a lot of devs, especially the younger ones, joined post-cloud and haven't used anything but. I think there's also an argument that not really understand how your code is run & deployed is a bad thing. The clouds advertise that "developers shouldn't need to care how X works, how their code is deployed or where" but I disagree; this is a blatant lie. I've lost count of the number of projects I've seen fall over because something fundemental about the cloud / networking / deployment / ops wasn't understood at the start.

Nice blog!

REPLY

RAMEEREZ
OCT 15, 2024

Thanks for reading and sharing your experience, Al! I enjoyed reading your comment.

REPLY

AGENT CYBERIS
NOV 05, 2025

I celebrated 50 years in everything automated last July. Started writing Fortran on a teletype connected to a Univac over an acoustic modem. Most of my experience is building the hardware and software. Great experience. I've noticed lots of ebb and flow. Large centralized resources and dumb terminals at the edge, then distributed computing, then decentralized capability at the edge, and back. I'm glad companies are beginning to re-patriate their computing to on prem, or colo based hardware. The cloud might be fine to rapidly prototype something, but someone's got to pay for those billion dollar, gigawatt datacenters and it isn't AWS/GCP/Azure; it's their cash cow customers. I never liked being milked. Besides, spinning up your own server(s) is cool.

REPLY

IÑIGO
NOV 05, 2025

I worked for a telco company for more than 15 years and we always wondered *how* it was possible that AWS could charge you so much money for traffic.

Transit is actually *really* cheap.

Telcos will either charge you for line capacity, or for a direct peering connection a percentile 95 of the traffic you're injecting.

AWS, Google, etc. charge you for byte transferred, which is much more.

It's not just the hardware. Everything comes with an extremely high price attached.

REPLY

LUKE CAVANAGH
NOV 06, 2025

Hetzner is very solid for a solid VPS hosting provider.

REPLY

WHERESALICE
NOV 06, 2025

It's hard to argue that Cloudflare isn't the cloud when it literally has cloud in its name. But otherwise yes, fully agree there's little actual value from big cloud providers. The perceived value is because people have had bad experiences with poorly-managed on-premise infrastructure, and ironically that's often because they weren't given enough budget

REPLY

‹ HOW I EXITED THE CLOUD

KAMAL'S MISSING TUTORIAL – HOW TO DEPLOY A RAILS 8 APP WITH POSTGRES ›

I'm rameerez, an indie developer making cool software products on the internet. I created PromptHero, Hustl and Jobician, among others. My users include people at companies like Google, Uber or Adobe. I've been featured in The Washington Post, The Wall Street Journal, Vox, Fast Company and dozens of other press & media appearances.

Follow @rameerez

COPYRIGHT © 2025 RAMEEREZ

by @rameerez