

03.29.2023

# S3 as an Eternal Service

BY COREY QUINN

AWS Chief Evangelist Jeff Barr obviously needs no introduction to anyone even tangentially aware of what AWS is—but did you know that he’s also a philosopher? I recently got to chatting with...



Prev

[Home](#) ▶ [Blog](#) ▶ [S3 as an Eternal Service](#)

---

AWS Chief Evangelist Jeff Barr obviously needs no introduction to anyone even tangentially aware of what AWS is—but did you know that he’s also a philosopher?

I recently got to chatting with him about The Future®, and he said something that caught me completely off guard:

I sometimes think about the fact that Amazon S3 effectively has to exist until the heat death of the universe. Many millennia from now, our highly-evolved descendants will probably be making use of an equally highly evolved descendant of

S3. It is fun to think about how this would be portrayed in science fiction form, where developers pore through change logs and design documents that predate their great-great-great-great grandparents, and users inherit ancient (yet still useful) S3 buckets, curate the content with great care, and then ensure that their progeny will be equally good stewards for all of the precious data stored within.

Jeff Barr

Jeff is right. Of *course* he's right! By and large when we're talking about technology in the context of systems that span multiple human lifetimes we sound absolutely ridiculous and shouldn't be having those conversations as anything other than pure hypotheticals, but S3 is very much a horse of a different color here.

Despite doing what every good sysadmin does exactly once in their career and naming servers after Federation starships, I'm no sci-fi author. Instead, let's turn our attention to the real world. Here's a challenge for you: as an idle experiment, poke around when you have a few minutes and try to find the oldest file on your computer. I was able to find one or two from 2008 on mine. "The internet is forever" is the common wisdom, but when I took the [recent LastPass breach](#) as a reason to rotate a bunch of old passwords I was surprised by just how many sites have ceased to exist over the past twenty years.

But a few do.

AWS makes a point of not knowing what's within customer S3 buckets; as a result they can't tell whether a given bucket holds load balancer logs from 2009, pictures of your cat, the nuclear launch codes, or the rough draft of this post. Some of those things are incredibly important, others probably should never have been stored in the first place—but Amazon does not and can not judge. So long as the bill gets paid (and for an absurdly long time, even when it doesn't), your data is treated as "this cannot be lost under any circumstances." As a result, every last byte has to be treated as if it's life critical for it to still be there until AWS is told otherwise.

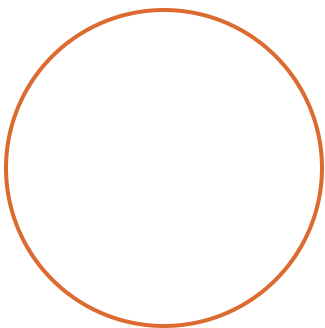
S3 has been [rebuilt from the ground up](#) multiple times since it launched 17 years ago, and has had its feature set expanded dramatically since then—but the data anyone stored in it from its first day remains in place. (It should also be said that that data will now cost you just 15% of what it did when you uploaded it.) Yes, yes, you're going to just skip past this point—but stop for a minute and *think* about what exactly that portends. You've got data that you wrote to an S3 bucket. You can retrieve it via the same API calls that you used 15 years ago, but like a digital Ship of Theseus that charges monthly rent that periodically decreases, every physical aspect of the system you wrote that data to *no longer exists*. The servers, the

hard drives, the networking devices, in some cases the data center facilities themselves; all have been refreshed and replaced multiple times.

Say what you will about cloud, and I will say a lot more, but you can't deny that it's done a princely job of abstracting services from their underlying hardware. You can overpay for them compared to a bunch of other options, but should you choose to do it you can [run older EC2 instance types](#) today; AWS has taught Nitro to emulate their characteristics on modern hardware. Ever notice that in 2017 you had periodic "EC2 instance degradation" notices showing up consistently, but these days they're almost unheard of? Even in our data centers, Kubernetes and its ilk of container orchestrators have separated out "a hard drive or a server just blew up" from the services running atop that hardware.

I don't know about you, gentle reader, but I find the idea that something I built ten years ago still being in production use to be more than a little revolting; my code wasn't that good at the time, and it's certainly garbage now! That should be considered a terrifying bug that should be stamped out with ruthless efficiency.

Accordingly, if you tell me that you're building something with an eye towards it still existing centuries from now, I will look at you as if you've slipped a gear. The very idea is absurd—except where something as foundational as S3 is concerned. So much depends upon that service existing in the same form it originally launched with, and it's met those expectations superbly. The [amount of engineering work](#) put into the system is nothing short of astonishing, yet the only way you or I know that it's happening is because AWS talks about it. The service continues on as it ever has, storing our data for the rest of eternity.



## by Corey Quinn

Corey is the Chief Cloud Economist at The Duckbill Group, where he specializes in helping companies improve their AWS bills by making them smaller and less horrifying. He also hosts the "Screaming in the Cloud" and "AWS Morning Brief" podcasts; and curates "Last Week in AWS," a weekly newsletter summarizing the latest in AWS news, blogs, and tools, sprinkled with snark and thoughtful analysis in roughly equal measure.

## More Posts from Corey

[Back to the Blog](#)

### **AWS's Anti-Competitive Move Hidden in Plain Sight**

BY COREY QUINN

Folks have been making noises for a while that Amazon is anti-competitive in a variety of ways. I don't disagree with the overall sentiment, but there's one particular aspect of how AWS advantages its own offerings that I don't see folks talking about—and strikes me as particularly egregious.

[Read More →](#)

### **AWS is Asleep at the Lambda Wheel**

BY COREY QUINN

Countless volumes have been written about the various benefits of serverless, a task made even easier by it being such a squishy, nebulous term that's come to mean basically whatever the author wants it to mean. This has been a boon for AWS's product teams, who've gone from creating services that are clearly serverless such as DynamoDB, Route 53, IAM, and others to instead slapping the "serverless" moniker on things that are clearly not very serverless at all, like OpenSearch and Aurora.

[Read More →](#)

## Amazon's Snowball Edge Frustrates This User

BY COREY QUINN

A friend who happens to work at AWS recently had reason to send me about 500 GB of data. This friend is a business user. Now, say what you want about AWS's hiring practices, they certainly don't hire people who aren't intelligent. This intelligent person made the very reasonable determination that the best solution they had available to send me that half terabyte of data was to mail me a hard drive.

[Read More](#) →

## Get the newsletter!

Stay up to date on the latest AWS news, opinions, and tools, all lovingly sprinkled with a bit of snark.

Email Address

Sign Me Up!

The world of cloud takes itself far too seriously. We aim to change that.

Lower my AWS bill, please!



Newsletter

About

Podcasts

Contact

Blog

Sponsorships

Merch

Disclosures

Contribute



© 2023 The Duckbill Group. All Rights Reserved. [Privacy Policy](#) [Cookie Policy](#)