

# The NixOS Foundation's Call to Action: S3 Costs Require Community Support

---

**domenkozar** Great contributor

3d

Hi all,

TL;DR - Kicking off an effort to secure long-term funding for our S3 costs and exploring alternatives. This comes after multiple years where Logicblox has been graciously sponsoring the S3 costs for Nix! An enormous thank you to them.

The LogicBlox team (which has since been acquired by Infor) has been providing significant support to the Nix ecosystem by sponsoring our S3 buckets. They have asked to transfer the ownership of the costs and give us a good heads up (timeline discussed below). We are now working on transferring the expenses/ownership to the foundation.

## Details

Estimated Monthly Costs - ~\$9000 / month for hosting [cache.nixos.org](https://cache.nixos.org) , see the [breakdown for more details](#) .

Deadline - Aiming for July 1st

## Initial Potential Solution Brainstorm

We're exploring different solutions for this, including getting direct sponsoring from a hosting provider (AWS or Cloudflare) and having the foundation paying for the S3 buckets through community sponsorship.

So far the most likely solutions are:

- Migrating the buckets to [Cloudflare R2](#) , and using the generous OSS sponsorship offer [they recently announced](#) . That would incur a fixed cost of around \$32k for the migration.
- Keeping the bucket as it is and paying it in full (\$9k/month at the moment, steadily rising as the cache keeps growing)
- Keeping the budget, but garbage-collecting it. We could probably shrink the costs a lot (realistically up to 70%), but there are heavy tradeoffs at play here since the data in the cache is sometimes very valuable.

## Call to Action & Next Steps

- Fundraising: We will start a general fundraiser for the cache, as every option currently on the table would require significant costs.
- Creating a NixOS community task force to investigate these possibilities further: If you are interested in joining, please comment or reach out. - <https://matrix.to/#/#foundation:nixos.org>
- General request to everyone who has thoughts on this topic to join the discussion/brainstorming:

[Skip to main content](#) [#foundation:nixos.org](https://matrix.to/#/#foundation:nixos.org)

- Are there any other providers we can explore?

- We will follow up with Cloudflare/AWS to receive more information and explore any other options/opportunities.

Domen, Ron, Jonas, Eelco, Théophile

---

## Nebucatnetzer

3d

Maybe a company like Infomaniak would be willing to help, they provide S3 as a service.

I don't know how involved they are in the Open Source ecosystem but I know that some of their products are based on OSS and they seem to have good values.



### Webhosting, Cloud Computing, Cloud-Server, Domainname, E-Mail,...

Infomaniak, die ethische Cloud – Wir entwickeln das Nonplusultra in Sachen Webinfrastruktur, um die Weichen für Ihren Erfolg zu stellen. Lokaler und an sieben Wochentagen erreichbarer Support

---

PINNED GLOBALLY 2 DAYS AGO

---

## RaitoBezarius

3d

I just want to note multiple things regarding that.

domenkozar:

- Keeping the budget, but garbage-collecting it. We could probably shrink the costs a lot (realistically up to 70%), but there are heavy tradeoffs at play here since the data in the cache is sometimes very valuable.

I very much agree that this data is extremely valuable, and losing it would be a terrible blow for research communities and even understanding our own ecosystems.

That's why I believe, no matter what happens: new store paths should be shipped to something that does not cost egress fees if possible, more on that later.

Onto the next thing:

domenkozar:

- Migrating the buckets to **Cloudflare R2**, and using the generous OSS sponsorship offer **they recently announced** . That would incur a fixed cost of around \$32k for the migration.

The \$32K migration fee, is, I assume, the egress thingie that AWS charge everyone for trying to quit their platform.

| [Skip to main content](#) :

- (1) Have we reached out to AWS regarding this matter while pleading our case as a non-trivial open source project (which they benefited from, I'm certain, indirectly)?
- (2) Have we reached out to a financial cloud experts such as <https://www.duckbillgroup.com/> which are very active on social medias and I usually believe nice to open source projects for help?
- (3) Egress fees are a known tactic to vendor lock-in people into a platform:

AWS are a very well-known offender on that.

Thankfully, R2 Object Storage has **ZERO** egress fee **for now**.

I think this situation is highly changing with respect to regulations:

<https://twitter.com/platombe/status/1656634228979781633> (French tweet on a law attacking the egress fees situation from the cloud providers by the French government).

[Sécuriser et réguler l'espace numérique](#) (art. 7. §. 6)

« III. – Il est interdit à tout fournisseur de services d'informatique en nuage de facturer, dans le cadre des contrats qu'il conclut avec une personne exerçant des activités de production, de distribution ou de services, des frais au titre du transfert de données vers les infrastructures de cette personne ou vers celles mises à disposition, directement ou indirectement, par un autre fournisseur, à l'exception des frais de migration liés au changement de fournisseur.

“III. – It is prohibited for any provider of cloud computing services to charge, within the framework of the contracts it concludes with a person carrying out production, distribution or service activities, fees for the transfer of data to infrastructure of this person or to those made available, directly or indirectly, by another supplier, with the exception of migration costs related to the change of supplier.

— Google Translate, 2023

I believe that this could apply enough pressure so that AWS could yield this stupid amount of fees in our special case, at least, this would be a gesture of goodwill given the thin ice they are on.

– Of course, nothing is magic, and it could be as well that all those attempts would fail. I just want to make sure that we attempt everything and that our failed attempts will also serve as examples for further leveling the playing field for those migrations, which we are definitely not the only ones to do.

## On long-term solutions

The breakdown shows:

- 107 TiB in standard storage
- 318 TiB in infrequent access storage

for a total of **425 TiB**.

Current technology gives us at reasonable prices: 30.72TB SSD (e.g. PM1643a) and 22TB HDD.

Assuming we store 150TB of “hot storage” with a fast medium (SSD): 10 *professional* disks of 15.36TB. (no redundancy is assumed here.) for a cost of ~11K EUR.

Assuming we store 500TB of “cold storage” with a slow medium (HDD): 23 *professional* disks of 22TB. (no redundancy is assumed here.) for a cost of ~13.8K EUR.

What did we miss in this small computation?

- Geodistribution? We don't seem to care because we have (for now?) Fastly in front of it which will geodistribute the cache across the world? (Also, we have the back'n'forth situation latency between the US and EU for some current servers of the infrastructure.)
- Internet? I can speak for France easily and having 100Gbps or even 400Gbps and being plugged directly in the adequate IXP is not really complicated, I can even email get a proper quote on what it would cost. Last time I checked, having something like 10Gbps **commit** (it means that you get 95th percentile 10Gbps **guaranteed**) was around 1-2K EUR per month.
- Disk replacement? Yes! I didn't conduct the long term analysis and I wanted to post this early. This should be computed and included in the price by using Backblaze B2 statistics for example ([Backblaze Hard Drive Stats](#)).
- Human cost? Yes! This all relies on under the assumption we will have enough persons to handle such a thing and operate it correctly. Is this a fair assumption? Let's see below.
- Durability? Yes! Amazon S3 provides an insane durability metric, which is cool and neat: though, do we need it? (99.999999999% of durability FWIW, yes, it has this much 9s. **But** only 99.99% **availability**) — though, we can push our metric further by piling up on redundancy and multiple tiered solutions as we wish.
- You say it: I could have forgotten something in this.

This small exercise is important IMHO to decide properly for the next solution following the rule “never let a good crisis go to waste”.

NixOS's cache has specific properties that are not shared by all users of Amazon S3 and we don't need to stay captive in a cloud ecosystem if we don't need to.

While I don't think it's realistic except if enough people step in, and it ultimately becomes realistic to provide a homemade solution for this (be it through something like a Ceph filesystem or Garage object storage or MinIO object storage or ). I feel like this has shown the need for our community to step in for those things, as we definitely have the expertise in-house and the scale for it.

My proposal would probably say: let's try to experiment on those things I said before. For the human cost, as this is an experimentation, it should not have detrimental impacts. An alternative is to consider scaling the infrastructure team adequately and give them more power regarding this. It's also a unique position to do something about the **signing key** of the store (see latest RFC on the subject for more information).

[Skip to main content](#)

Finally, while I am not super fond of this solution, have we considered university mirrors like many other distributions do it and have we considered working with them towards that?

Anyway, I hope that my messy “chime-in” can provide new leads on what we can do regarding this situation. I would find extremely regrettable:

- (1) we lose the historical data because AWS egress fees are enormous, have we considered using AWS Snowmobile? Or similar “physical transfer” solutions to ship them out?
- (2) we would waste the money of the Foundation on a non-stable situation which would cost us more on the long run because we just get milked by cloud providers who have insane margins because IMHO the tradeoff for using them is skill/competency/time/human cost — nixpkgs is full of that! Let’s make it shine.

---

**arianvp** Great contributor

3d

RaitoBezarius:

I very much agree that this data is extremely valuable, and losing it would be a terrible blow for research communities and even understanding our own ecosystems.

RaitoBezarius:

Durability? Yes! Amazon S3 provides an insane durability metric, which is cool and neat: though, do we need it? (99.999999999% of durability FWIW, yes, it has this much 9s. **But** only 99.99% **availability** ) — though, we can push our metric further by piling up on redundancy and multiple tiered solutions as we wish.

If we care for historics I think durability is a very nice benefit

---

**lucasew**

3d

Just for curiosity sake, how many paths are there?

I don’t know if it’s just me but occasionally (almost often) when I am writing derivations the build sticks at querying for the narinfo of the derivation I just built. It is stuck only in the situations where it checks for the narinfo of a path that is not in the cache.

---

**qyliss** Great contributor

3d

RaitoBezarius:

An alternative is to consider scaling the infrastructure team adequately and give them more power regarding this.

Assuming it stays reasonably competitive, we should strongly consider whether we could pay our infrastructure team for this as well — the alternative of being okay paying (indirectly) AWS employees’

[Skip to main content](#)

salaries, while not being okay to pay our own community members for the same purpose wouldn't make any sense.

---

RaitoBezarius

3d

lucasew:

Just for curiosity sake, how many paths are there?

From the mentioned post:

### NixOS Foundation's Financial Summary: A Transparent Look into 2022

- The runtime closure of all non-beta NixOS releases (i.e. the store paths reachable from the `store-paths.xz` files in [Channels for NixOS project\(s\)](#)) is 13M store paths, **27.1 TiB**. So garbage-collecting `cache.nixos.org` to only keep NixOS releases would shrink it to 6.3% of its current size. Note that this doesn't include build-time-only dependencies like source tarballs.
- The runtime closure of all non-beta NixOS releases and all Nixpkgs releases is 51M store paths, **91.4 TiB**, so about 21% of the bucket. Unfortunately Nixpkgs has no stable releases, they're all marked as "pre".

---

RaitoBezarius

3d

arianvp:

If we care for historics I think durability is a very nice benefit

Obviously, what I'm asking for, is to compute what are we willing to guarantee vs. what can we afford realistically.

If we have enough to pay for this level of durability, maybe via AWS sponsoring! (<https://cdn-aws.deb.debian.org/>!) — why not!

---

misuzu

3d

I've frequently thought that staging cycles might be expensive in terms of storage (build artifacts are mostly unused). Maybe there's a benefit in splitting master/release branches and staging/glibc updates/etc to different storage spaces for easier garbage collection of development build cache.

---

vdot0x23

3d

RaitoBezarius:

[Skip to main content](#) super fond of this solution, have we considered university mirrors like many other

distributions do it and have we considered working with them towards that?

At [dotsrc.org](https://dotsrc.org) we would love to help but have only ~17 TB of free space since our mirrors mostly run on decommissioned hardware and a relatively low budget.

I do not know what kind of storage other mirrors have, I guess it varies a lot (see [TUNA Nix mirror](#)), but I would be surprised if this route is feasible (especially on a 1 month deadline).

---

[flokli](#) Great contributor

3d

Another thing to keep in mind is that the NAR format is pretty wasteful on its own, no matter if we compress the NAR files or not.

A lot of the files inside a store path don't actually change, and "exploding" the NAR file and the structure into smaller parts gives a lot of possibility for deduplication.

As part of my work on [tvix](#), I've been spending quite some time on tvix-store, which is using a model similar to git trees/blobs ([but not quite](#)) as an underlying storage.

It is still able to produce NAR files on demand, just using another internal, content-addressed storage model for the underlying data.

We are already using all that logic in other places, but I also have some code wiring this up to a HTTP server handler, so Nix clients would still be able to "download NAR files" even though we don't need to store them as such.

If there's interest to explore this idea further, maybe start ingesting some store paths and see how much dedup we get, and potentially write some fetch-through migration tool, I'd be happy to discuss this further.

---

[sambacha](#)

3d

This post was flagged by the community and is temporarily hidden.

[View ignored content.](#)

---

[Nebucatnetzer](#)

3d

I know one thing at the time, etc.

However what do we do when GitHub doesn't want to provide free services to us anymore?

Is there maybe another thread where we can have this discussion?

---

[RaitoBezarius](#)

3d

Nebucatnetzer:

I know one thing at the time, etc.

However what do we do when GitHub doesn't want to provide free services to us anymore?

[Skip to main content](#) r thread where we can have this discussion?

We already discussed migrating alternatives from GitHub, e.g. to Codeberg, IIRC, [@davidak](#) explored it. It didn't scale.

When we will have this situation, we will cross the bridge, I believe. For now, I imagine that GitHub has an incentive to provide us free services in exchange for a user base.

A bigger disaster would be Fastly .

---

[ron](#)

3d

Small update, we are continuing to work on multiple communications thread with AWS to figure out if there is a possibility to receive further support on the topic.

I've submitted us for the Cloudflare OSS program mentioned and will aim to have a conversation with the team there as soon as possible.

Let's also aim to have a call to brainstorm and discuss anything on the matter early next week. I'll post details over the weekend.

Again, thank you to everyone that is getting involved! <3

---

[sarcasticadmin](#)

3d

(Originally mentioned on matrix in a discussion with [@RaitoBezarius](#) )

Another consideration for the s3 migration would be to leverage AWS Snowball ([AWS Snowball | Secure Edge Computing and Offline Data Transfer | Amazon Web Services](#) ). It seems like it would help with the egress cost (~0.03/GB instead of ~0.09/GB) and not require the destination to have a large network pipe. Its essentially the modern version of a station wagon full of tapes.

Specs of the storage-optimized snowball ([AWS Snowball Edge Device Specifications - AWS Snowball Edge Developer Guide](#) ) which we would need 2x of:

Storage specifications:

NVME storage capacity 210 TB usable (for object and NFS data transfer)

Dimensions and weight specifications:

Weight 49.7 pounds (22.45 Kg)

Height 15.5 inches (394 mm)

Width 10.6 inches (265 mm)

Length 28.3 inches (718 mm)

---

[uep](#)

3d

I like the confluence of the two points around garbage collection and deduplication. Both have a chance to recover (potentially rather large) total storage space. Staging builds (and older builds from unstable, and a progressive list of others) are unlikely to ever be used, and could be garbage-collected. Lots of things deduplicate (and compress) well in an expanded store, as my own zfs-based instances demonstrate.

[Skip to main content](#)

Storage provides like AWS undoubtedly use this for their own cost advantage, we should make sure we can take our own advantage of the data we understand best.

Doing either can get us some space back, and each can reduce the work needed for (or benefit available from) the other: stuff that deduplicates from a staging build was unchanged when that staging landed, for example.

The nice thing here is that despite this interaction, they're not really in competition with each other, except perhaps in some ivory-tower view of wanting a perfect single solution. They can operate on different time-scales to provide practical benefit and shrink the problem to more manageable levels as things progress; we could collect some garbage now to reduce immediate storage costs and potential transfer/migration costs (and time) while more extensive storage format changes supporting dedup are developed and finalised. I can even imagine an approach where some of this historical data is archived off elsewhere for a while, gc'd from the expensive cache, and maybe reinjected again later.

Choosing which garbage to collect might be helped with some better data. We have a split of warm vs cold storage already, I assume that's based on S3's automatic migration, and it holds some clues (but there are caches in front, so regularly-used items might not get S3 activity). Do we have stats on what items are hit from Fastly, and a way to turn that into a view of which closures are pulling things in?

What is the actual value of historical builds, in the abstract (assuming we can identify and exclude particular items that are in current use for various accidental or deliberate reasons)?

There are a lot of more extensive changes along these lines that can benefit everyone, applying similar benefits to local stores and network transfers, as has been discussed before. Because of that, though, they will take longer, even if this situation gives some impetus to revive the effort. What happened to the content-addressible store work, for example?

---

**jtolio**

**2d**

Hey friends! Nix user and Chief Strategy Officer of **Storj** here. We're somewhat like BitTorrent meets S3.

Storj is S3 compatible \$4/TB/mo and \$7/TB egress. This isn't \$0/TB egress of course, but might perhaps be a long-term sustainable option.

Migrations from S3 are free (via our partnership with <https://www.cloudflyer.io/> , see their site for details). Even though we are decentralized storage, there isn't any funny business with new protocols or challenging technical adoption. We work with any S3 or HTTP clients.

We have 20,000 global points of presence, so you don't have to worry about multiregion replication or content delivery. We handle that for you as part of the base product.

We are open source and would love to help support the Nix community. Could we help here?

Edit - Please feel free to hit me up directly @ [jt@storj.io](mailto:jt@storj.io). Maybe we can find an arrangement that works for you!

---

**birdman3131**

**2d**

^ [Skip to main content](#) ible service. No egress fees and storage costs looks to be right at \$1k a month for 425GB (Ignoring the contact us for better pricing button.)

Caveat. I know very little about the details on it. We have had great luck with using them for VOIP service but I have never looked into using their storage service.

### **Cloud Storage Pricing - Low Cost, No Egress Fees**

Telnyx Storage is a distributed cloud storage solution with low costs, low latency, and S3-compatible APIs for easy migration.