

# pgBackRest is dead. Now what?

Apr 27, 2026 · 1224 words · 6 minute read

## ► Table of Contents

I have been recommending `pgBackRest` as the best backup tool for PostgreSQL for years. I even wrote a [blog post about it](#). My students at Université Lyon I were able to backup, restore, and perform PITR in four hours with zero prior knowledge of the tool. That is how good it was.

I say “was” because David Steele, the sole maintainer of `pgBackRest`, has announced on the [project's GitHub page](#) that he is stopping all work on the project. His words: no more maintenance, no more bug fixes, no more PR reviews, no more new features. `pgBackRest` is, as of today, unmaintained.

Before anything else, I want to thank David for thirteen years of exceptional work. I also want to name two people who contributed enormously to making `pgBackRest` what it is: [Stephen Frost](#) and [Stefan Fercot](#), known in the community as PGStef, along with many other contributors whose work made the project what it became. `pgBackRest` is a masterpiece of engineering. The fact that it reached this level of quality and reliability with such a small core team makes it all the more remarkable. David, Stephen, Stefan: thank you.

## What happened

David explains it clearly, without bitterness. Crunchy Data, which had sponsored `pgBackRest` for most of its life and employed David, was sold. After that, David spent months looking for a position that would let him keep working on the project. He also tried to secure independent sponsorship. Neither worked out. He needs to make a living. The project needs sustained effort that he can no longer provide without being paid for it.

This is not a story about one person failing to find funding. This is a story about the IT industry in 2026, and what it values.

## The thing nobody wants to say out loud

David is a brilliant engineer. He spent thirteen years building one of the most reliable pieces of infrastructure in the PostgreSQL ecosystem. He tried to find a company that would hire him to keep doing exactly that.

They were not interested. They had RAM to buy and GPUs to provision. The AI gold rush has thoroughly reshuffled what companies consider worth paying for, and apparently “the person who makes sure your data survives a disaster” did not make the cut.

I find that hard to write, because David is a friend. So are Stefan and Stephen. Watching a project this good die not because of a technical failure or a community falling-out, but because the industry decided that large language models matter more than data integrity, is infuriating. And sad.

## What about the open source contract

Large companies have made enormous amounts of money on top of `pgBackRest`. It is deployed in production at hundreds of organizations, including some with very profitable database services built directly on the PostgreSQL ecosystem. The project README had a sponsorship link. There was one active sponsor at the time David made his announcement.

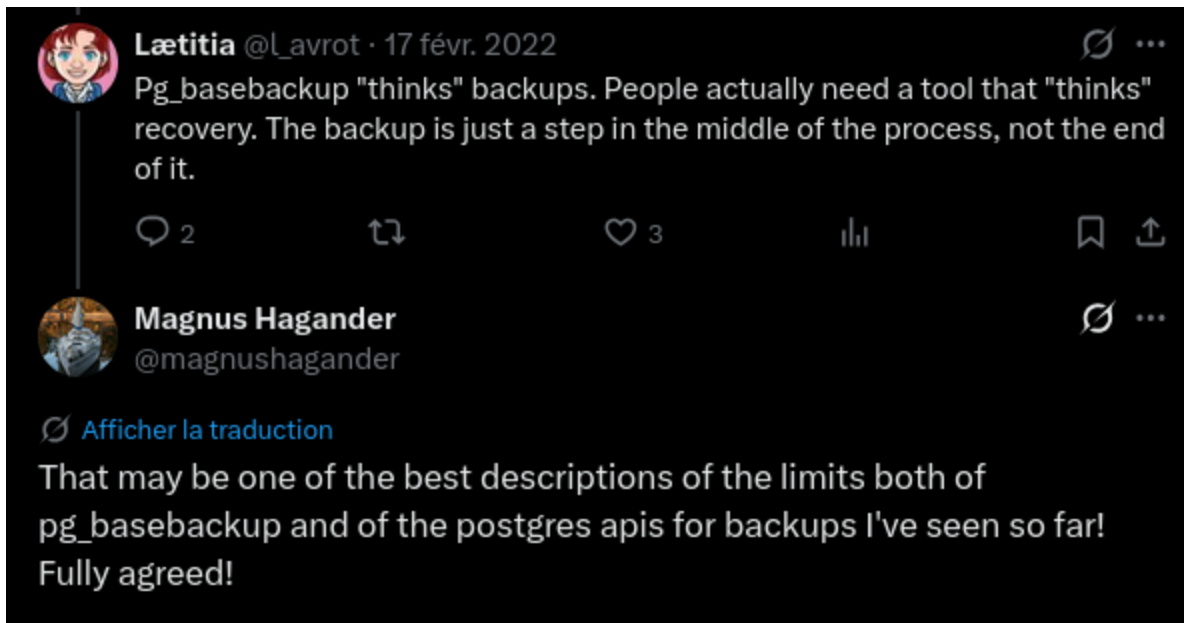
I am not going to lecture anyone. But if your company’s disaster recovery strategy has relied on `pgBackRest` for years and you never contributed anything back, this is a good moment to sit with that.

The open source model works when the people consuming the value also contribute to sustaining it. It breaks when everyone assumes someone else will pay for the maintenance. This is what that breaking looks like.

## What `pgBackRest` actually was, and why alternatives fall short

Let me be precise about what the community is losing, because I think most people underestimate it.

`pg_basebackup` is not a backup tool. I have said this many times, and Magnus Hagander, who created `pg_basebackup` and is a member of the PostgreSQL core team, agreed with me publicly. In [this exchange on Twitter](#), I wrote: “Pg\_basebackup thinks backups. People actually need a tool that thinks recovery. The backup is just a step in the middle of the process, not the end of it.” Magnus replied: “That may be one of the best descriptions of the limits both of `pg_basebackup` and of the postgres apis for backups I’ve seen so far! Fully agreed!”



`pg_basebackup` was designed to clone a running cluster directory. It has no backup catalog, no WAL retention management, no restore command, and no built-in integrity verification before PostgreSQL 13. It is an excellent tool for setting up standbys. It is not a recovery strategy.

`pg_dump` is even further from the mark. Beyond the obvious absence of PITR, meaning that any transaction between the moment your dump started and the moment you need to restore is gone forever, the restore time for a large dump is the time you do not have during an incident. I prefer to call `pg_dump` an export tool, because that is what it is. Calling it a backup tool creates a false sense of security that has caused real data loss.

`Barman` exists, is actively maintained, and has improved significantly. For organizations that need an alternative today, it is the serious option. It carries the architectural weight of having been built on top of `pg_basebackup`'s limitations rather than starting from scratch, but it

covers the critical gaps: WAL archiving, backup catalog, retention management, and restore. It is a legitimate choice.

## What comes next for pgBackRest users

David himself expects `pgBackRest` to be forked eventually. So do I. The codebase is solid C, the architecture is sound, and there are companies in the PostgreSQL ecosystem with the technical depth to take it on. I hope one or several of them step up. They will need to build community trust from scratch, as David notes, but the foundation is exceptional.

That fork has not happened yet. Until it does, here is what I recommend.

If you are evaluating backup tools today, use `Barman`. If you are running `pgBackRest` in production, you are not in immediate danger, but your window is narrowing with every new PostgreSQL release and every bug that goes unpatched. If you find a critical bug in `pgBackRest` in the meantime, companies like [Data Egret](#) and [Cybertec](#) have the PostgreSQL expertise to help you address it. That is not a long-term solution, but it buys you time while the community figures out what comes next.

## A closing thought

I have spent twenty years in the PostgreSQL ecosystem. I have seen tools come and go. I have seen projects stall, get forked, get revived, and occasionally just disappear. It is part of how open source works.


But `pgBackRest` is different. It is one of the best-engineered tools I have used in my entire career. It was built with integrity, maintained with care, and trusted by people whose jobs depend on their data surviving the worst. Watching it reach this point, not because of any failure on David's part, but because the industry is currently more interested in the next AI product than in the humans who build reliable infrastructure, makes me angry and sad in equal measure.

The PostgreSQL ecosystem is full of brilliant people doing critical work, often with fragile or nonexistent funding. `pgBackRest` will not be the last project to reach this point. I hope it is at least the one that makes companies think twice before treating open source infrastructure as a free resource with no obligations attached.

David built something that will outlive this moment. I am certain of that. I just hope the community rises to meet it.

[PostgreSQL](#) [Administration](#) [Open Source](#)



© Copyright 2026  Lætitia Avrot

Powered by [Hugo](#) Theme By [nodejh](#)