



Webinar: Guide to Kubernetes

Join us to learn about insights and tools to achieve a successful Kubernetes implementation.

OPEN SOURCE / SOFTWARE DEVELOPMENT

Why Open Source Forking Is a Hot-Button Issue

Perhaps in the future, companies will realize that open source is not a business model and think twice before open sourcing — if they are not in it for the long term.

Sep 26th, 2024 10:00am by [Amanda Brock](#)



Image by jacqueline macou from Pixabay.

Valkey, **OpenTofu**, and **OpenBao** are names of open source software project forks hosted by the Linux Foundation. The forks were instigated last year in response to license shifts (from open source to proprietary licensing) by the commercial sponsors of open source software projects; HashiCorp was forked by OpenTofu, OpenBao, and Redis Labs with Valkey. The latter has been splashed across the press in the last few weeks.



Webinar: Kubernetes at Scale

Join us to learn about insights and tools to achieve a successful Kubernetes implementation.

also seen a new basis for forking and potential impacts that merit exploration and understanding. As with many areas of open source software, there is a great deal of nuance involved in forking, and these changes necessitate in-depth understanding.

What Is Forking?

Forking allows developers to “lift and shift” an open source codebase, creating a new, independent project version at a specific time. By creating a new branch in the code repository, this fork can serve as the foundation for a distinct project with its own vision, governance, and team, potentially evolving in an entirely different direction from the original. A new project name will be created as open source software licenses don’t generally grant rights in a name or trademark. It is accepted practice for commercial sponsors to make the name available to the community through a trademark policy, but only to the codebase they endorse and not to a fork.

The new project will want to build its unique identity and distinguish itself from the older project.

Forking is not an option with proprietary software, as the codebase, even if accessible, does not meet open source software standards. In proprietary systems, the necessary permissions to modify, redistribute, or create new versions of the code are typically restricted.

What Is Open Source Software?

A codebase must meet the ten criteria outlined in the **Open Source Definition (OSD)** the **Open Source Initiative** maintains to qualify as open source software. Among these, essential requirements (points 5 and 6) ensure that anyone can use the code for any purpose without restrictions on individuals or fields of endeavor. This unrestricted access enables frictionless distribution, a core principle of open source, which drives widespread adoption. As of 2023, **according to Synopsis**, 96% of all software contains open source dependencies, highlighting the critical importance of open source in modern software development.

Open source’s freeflow enables upload and distribution by device manufacturers with confidence that, by meeting license requirements, including attribution of the authors and making source **code available**, they do not require additional

TRENDING STORIES

1. [20 Years in Open Source: Resilience, Failure, Success](#)
2. [How to Explain the Security Advantages of Open Source](#)
3. [Social Web Foundation Launched — How In Is W3C on Fediverse?](#)
4. [From Open Source To Proprietary License? RedMonk Data Shows Minor Impact](#)
5. [Why Open Source Forking Is a Hot-Button Issue](#)

Where a software license does not meet the OSD, this is generally due to an ethical or commercial restriction on who can use the code or how they use it. Where projects are moved from open source software licensing to proprietary licensing by a commercial entity, this is generally due to the addition of commercial restrictions, and improving the commercial sponsor's revenue is the driver behind the license shift.

That kind of code is effectively a third kind of code. Technically, it is proprietary, but with its source code public, this latter fact can cause it to be confused with open source software, primarily if its past versions were licensed as open source. Known as a "public source" or "distributed source," there's also a recent attempt to establish it under the brand "fair source." The key is understanding that it isn't open source and not to mislead. Where a distributor misleads as being open source, it's described as "open washing," i.e., implying that the distributor of the code meets the standard of open source with its free-flow when that isn't true.

Shifting the Future, Not the Past

Future versions may be distributed under a different or proprietary license, but pre-fork codebases are always available under the original open source license applied to them. Their distributor may not take that back.

The final point at which a codebase might be forked is the last commit in the open source licensed version. When [Hashicorp sent a letter to the Linux Foundation](#) claiming LF had breached copyright in [OpenTofu], their [claim was quickly](#)

A Historic Weapon of Community Control

Within open source software communities, forking has long been the ultimate weapon of community control associated with a project. Forking allows a community to keep its leadership in check, as seen in the past where Sun and **Matrix** both experienced community forks.

Historically, forking gave the open source project's community the power to build a second version if its leadership had become out of touch with it or made decisions that a significant part of the community did not like. Community forking empowered a disaffected community to continue, reversing the unpopular project leadership decision to continue in its chosen direction. This ability remains a vital community tool.

Commercial Sponsors and the New World of Forking

Over the last few years, commercial entities' shift away from **open source licensing** has almost normalized. Community outrage at license shifts from Redis Labs in 2018, **MongoDB** in 2019, ElasticSearch in 2020, and open source poster child Hashicorp in 2024 has waned, and Cockroach's 2024 shift was met with little more than a grumbled: "Here we go again."

Sometimes referred to as "Single Vendor Products," these companies often have limited communities to push back on deciding to shift the project's license. They likely operate an "open core business model" alongside their open source codebase. This uses open source base product to draw users in followed by a contract for some form or enterprise version of the adopted codebase. As a user, should you want to explore the likelihood of a project fork, an assessment of the project community health may be helpful as an indicator. For example, companies with few contributors outside their employee base appear more likely to be candidates for a shift.

Adopting open source software at pace with the potential of ubiquity and becoming a de facto standard is possible because of its free flow. Coupled with distribution through widely used public repositories on standard licenses, software engineers can bring code into their companies without legal, procurement, or finance approval. Historically, these gatekeepers of **risk blocked software** adoption. Today, because of open source, contracts cannot manage the risk. It ought to be handled by software policies, and companies that don't have

Due to specific companies' behavior, forking has become essential over the last five-plus years. Those companies include Sun, Neo4J, Redis Labs, MongoDB, Elastic, **Confluent**, Hashicorp, and Cockroach. They have moved away from open source licensing for specific projects initially set up as open source software, with a valid open source license shifting to a proprietary license.

This flow of companies making a change has led to a pattern where the initial community outrage and the backlash have seen their behavior normalize over time. Founders in open source organizations complain that investors now ask "when, not if," they will switch their **code to a proprietary** model.

This has seen two outcomes: Commercial sponsors who have effectively forked their codebase and forking the existing open source codebase by users instead of the community.

This latter situation arose when AWS reacted to Elastic's license change of license for Elasticsearch, which resulted in their mixing proprietary and open source code. By **building its own OpenSearch fork**, a large and potentially more powerful commercial user took the codebase of a much smaller, if successful, open source-based company and forked it. The irony here was that Elastic, MongoDB, and others had accused cloud companies, in particular AWS, of strip-mining their code, and this behavior was blamed for the license shift in the first place. James Watters of Pivotal said, "One day you have a business, and the next you have a feature in the cloud."

The OpenSearch fork is still live and relatively successful several years later. However, Elastic recently shifted its code back to being under an open source license (the more restricted AGPL designed to apply copyleft where there is no distribution). It saw a shift in the relationship with AWS as the basis for that. This particular forking story may not yet be over.

In this stance, a user with the wear and all must take the supplier's codebase and run it with a built-in fork. While there are ethical debates around the rights and wrongs of this, AWS's actions in forking certainly did not breach any licenses. As part of the continuum of the open sourced project, Elastic arguably forked its product. In contrast, AWS while into maintaining the original open source project despite the name shift caused by Elastic's ownership of the Elasticsearch name, meaning the original project changed its name. The ElastiSearch name ownership

Importantly, however, there is a visible shift in the balance of power in the use of forks. This is not the end of the story of changes in forking.

Successful Forks

Following a fork, the winners in open source community terms are not the financially successful version of the software but the code-rich or most adopted version, i.e., the successful project is measured on the quality of its codebase and the utilization of the project. A great example is MariaDB, forked from MySQL. GitHub stars tend to correlate with usage, and MySQL has over 10,000 stars on GitHub, while MariaDB has a little over 5,000. But MariaDB has almost 400 contributors while MySQL has just over 100, so has the strength of the community.

The focus is not on revenue generated, although some commercial entities (possibly a former commercial sponsor of an open source project) may continue to look at the revenue generated after they shift their licenses and have been quick to say that they have continued — as they did before the project fork occurred — to make money. They may consider themselves unaffected by the license change and fork and feel they are the winners. Who knows if they are more or less successful than they would have been had they not shifted their license? We will never know.

Like Redis only last week, they often argue that customers don't care whether they are using open source software and use their continued commercial success as the basis for this. MongoDB's CEO has gone so far as to describe its use of open source software as "a marketing tool." This approach has become known as "bait and switch." The software option is initially available as open source (the bait), but the commercial sponsor eventually shifts the license to proprietary (the switch).

The cost and time commitment to build and maintain a fork and the challenges that come with it have created fundamental reasons for their low historical success rate. Yet, the **Linux Foundation's Valkey fork** may be evidence of another shift in forking and the dawn of something new.



Further Shift in Forking?

considering, or have already adopted the **Valkey** fork. Contextualizing this may be helpful.

As Redmonk's Stephen O'Grady pointed out, the pace at which the Valkey fork was created (8 days) from both a technical and engagement perspective — with giants like AWS and Google engaged in the fork — has been pointed to as a potential shift in the ease at which a **successful fork can be created**. It also increases the likelihood of the fork's success as measured by users. Is this usage of forking a nail in the coffin for the bait-and-switch approach to open source?

Conclusion

Companies like Redis claim that cloud companies' strip mining has forced them to shift away from open source. From an open source software perspective — if not an ethical and antitrust one — the cloud companies' actions in commercializing code that is open source software isn't a breach of the open source licensing and follows the principles at the heart of open source — that anyone can use the code for any purpose. Redis, Elastic, Cockroach, MongoDB, and others were not forced to choose open source licenses. They actively chose to open source and take advantage of building their customer base through the scale of adoption and customer engagement that open source enables.

The shift away from their original open source licenses may have normalized with repetition but remains a breach of the open source community's trust irrespective of the participants in that community and remains an act of betrayal of open source principles. From the perspective of a small open source company like Redis, the Valkey fork may not have been created by a band of rebels but by "**The Death Star**." Redis may simply be paying the price of users and communities becoming tired of the "bait and switch" approach.

Open source isn't a marketing tool. It's for life.

The Valkey fork may have confirmed that the community's choice to fork is the perfect counter-attack to certain providers' bait-and-switch approach to open source. Today, the community has some big players behind it. Perhaps in the future, companies will realize that open source is not a business model and think twice before open sourcing — if they are not in it for the long term.



Amanda Brock is CEO of OpenUK, the UK organisation for the business of Open Technology: open source software, open hardware, open data, open innovation, open standards and open access AI - with a purpose of UK Leadership and Global Collaboration...

[Read more from Amanda Brock →](#)

AWS, MongoDB, Confluent and Percona are sponsors of The New Stack.

TNS owner Insight Partners is an investor in: Control.

SHARE THIS STORY



TRENDING STORIES

1. 20 Years in Open Source: Resilience, Failure, Success
2. How to Explain the Security Advantages of Open Source
3. Social Web Foundation Launched — How In Is W3C on Fediverse?
4. From Open Source To Proprietary License? RedMonk Data Shows Minor Impact
5. Why Open Source Forking Is a Hot-Button Issue

INSIGHTS FROM OUR SPONSORS



**The Power of Flexibility in the Modern Data Stack:
The Data Lakehouse Advantage**

September 2024

