



SUPPORTED BY RED HAT

# Open source private cloud storage with OpenStack Swift

Posted 01 Jul 2013 | [John Dickinson \(/users/john-dickinson/\)](/users/john-dickinson/) | 111



Image by : [opensource.com](#)

Everyone has data. It's always growing, and you should have ownership of everything that touches your data. You need to have the ability to see and make changes to the code managing your storage system. Having ownership of your data gives you control of your own destiny.



## About the author

**John Dickinson** - John Dickinson is the Director of Technology at SwiftStack, a provider of cloud storage software for unstructured data. He serves as the Project Technical Lead for OpenStack Swift and has been involved in the development of Swift since 2009. John has spoken at many conferences, summits, and meetups around the world, including the OpenStack Summit and OSCON.

» [More about me \(/users/john-dickinson/\)](/users/john-dickinson/)  
» [Learn how you can contribute \(/participate/\)](/participate/)

All this started with web apps: companies were embracing the web as a new way to easily deliver software to a massive audience and looking for scalable infrastructure on which to build their applications. So, in 2009, Rackspace Hosting launched a product to give application developers access to flexible, scalable storage built to serve content at a massive scale.

Rackspace then began working on rewriting the underlying storage engine in order to improve performance and lower operating costs. This is when I joined the team (disclosure: I have since left) and started working on a storage engine called Swift, which still supports Rackspace's global storage infrastructure.

After Swift was launched into production in 2010, the company open-sourced the code as part of OpenStack. During the last three years, Swift has grown far beyond its original goals, and now anyone can deploy it on their own hardware and have access to its technology.

## **Because it's open**

Swift gives users ownership of their data because of its open design, open community, and open code. There's no lock-in and, by deploying locally, there is no fear of sending sensitive info to other companies for safekeeping, especially when those companies are in another country.

Two features contributed by the community that have dramatically improved Swift are:

1. Swift was originally designed to be deployed in large clusters consisting of many racks of gear. The change introduced by the community changed the data placement methodology so that multiple availability zones are not required, but can be taken advantage of if they exist. This now allows Swift clusters to grow from single-machine instances into globe-spanning clusters.
2. Swift was originally deployed with the assumption that clients would be on the other side of a relatively slow network, and it wasn't well tested with clients directly connected over a 10gbps connection. A bug was discovered that allowed such a fast client to monopolize the network connection and

prevent other requests from being processed. A patch was developed, and the issue was resolved.

Both of these community contributions came from people who were using Swift outside of its original design requirements. Because Swift is an open project, it has become more robust, more reliable, and better suited to more use cases than when it was first released. These features, and many more like them, are possible because Swift is open. Someone became interested in Swift, tried it out, found a pain point, and contributed a feature or bug fix.

## Created to solve a need for scalable storage

Swift was created to support modern web apps. It has continued to improve with features like object manifests, versioned writes, CORS, temporary signed URLs, expiring objects, and ACLs.

These features, developed by the open source community around Swift, have made Swift suitable not only for public cloud service providers, but for many organizations around the world looking for private cloud storage.

But we aren't done yet—there are many features left to write, and we are constantly working together as a community to make Swift better. There are a few improvements being discussed now that are especially exciting:

- Since Swift first launched, users have wanted to ensure their data could be stored across a wide geographic area. Over the last six months, several companies have been working on implementing the features needed to support **global cluster deployments**. The feature is included in Swift version 1.9.0.
- Improvements on how Swift talks to the **underlying storage media**. This will allow us to write specific optimizations for particular storage mediums.
- **Adding support for storage policies** would allow a Swift cluster to support multiple storage tiers. I hope to see Swift grow into a storage system that can effectively manage data over time from new, hot content to old, cold data.



