

## SwiftStack and python-swiftclient

### Summary

Storage administrators around the world are turning to object storage—and particularly SwiftStack’s enterprise-ready object storage built on OpenStack Swift—for its many benefits over legacy NAS and SAN architectures when used for unstructured data. While SwiftStack offers massive improvements over NAS and SAN in scalability, durability, availability, geographic distribution, and reduced TCO, the most common barrier to adoption is the necessary modification of an application to “talk” to its storage using a new RESTful API instead of the old NFS or CIFS/SMB protocols.

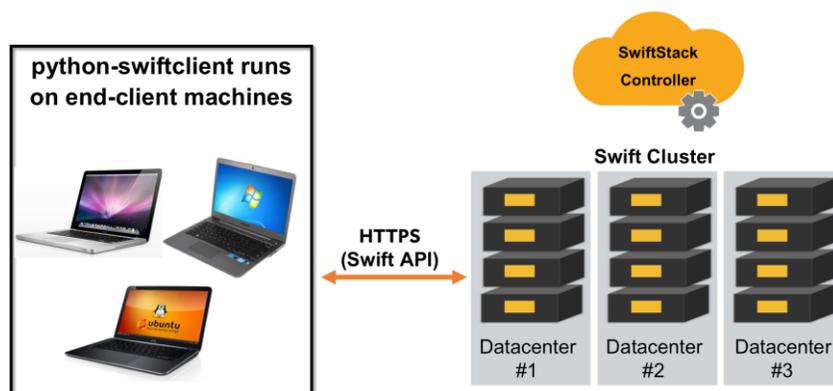
*“SwiftStack is the simplest way to deploy, manage, and monitor Swift object storage and undoubtedly offers the lowest TCO of any private-cloud object storage on the market today—never mind the high-cost, low-scale legacy NAS vendors. python-swiftclient gives Python developers an easy way to write to Swift’s API and also provides a great command-line tool to interact with Swift without programming at all!”*

— Chris Nelson, Director of Systems Engineering, Swiftstack

Among object storage APIs, the clear frontrunners are Swift and the S3 API introduced by Amazon. Many independent software vendors (ISVs) who produce applications that generate and manage unstructured data (e.g., software for backup, content repositories, large file archives, etc.) are gradually adding Swift and S3 support to their apps, though some are slow to adapt to these changes. Beyond ISVs, in-house application development teams who build homegrown applications are also adding object storage support to allow their apps to leverage the improvements in architectures like SwiftStack.

Whether it is for a commercial application from an ISV or a homegrown application internal to a single

organization, developers leverage “client libraries” in their programming language of choice to make it simpler to correctly interact with APIs like Swift or S3. Specifically, python-swiftclient—developed within the OpenStack Swift project—offers a client library for those programming in the Python language and a convenient



command-line utility that allows developers and administrators to interact with a Swift cluster without programming at all.

## Legacy NAS/SAN vs. Public Cloud vs. Public Cloud

For those who need flexibility, simplicity, and scale beyond what legacy NAS or SAN solutions can offer—or who want to realize the available cost savings, object-based cloud storage is the obvious answer. In most cases, this means adding application support for object-storage APIs—like Swift or S3—instead of traditional protocols like iSCSI, NFS, or CIFS. Many of today’s enterprise application vendors are making this transition, and homegrown applications can be modified as well. In return for this one-time effort, object storage offers the following benefits over traditional NAS and SAN solutions:

- Scalability from terabytes to petabytes to exabytes
- Single-namespace, no storage silos
- No-downtime software and hardware upgrades
- Reduced or eliminated need for backup and restore DR strategies
- Massively lower TCO

As for choosing between private and public clouds, there are good reasons for both. Public clouds commonly offer the following:

- Near-zero-effort destination for storing data
- Pricing models bake in the cost of hardware, infrastructure, and administrators
- Typically charged based on a combination of capacity used and data traffic into and/or out of the cloud
- May be ideal for small companies with little or no IT management resources who are looking to start the transition away from NAS/SAN storage.

For many organizations, however, there are important reasons to pursue private clouds instead or as well:

- Compliance with legal or policy requirements precluding storage of data in a third-party infrastructure
- Eliminated cost of data traffic to/from the cloud
- Complete control over storage infrastructure

## Integrating python-swiftclient with SwiftStack

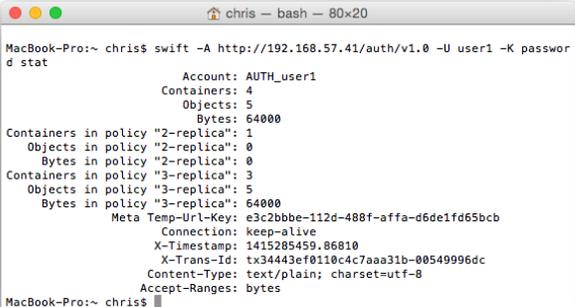
Because python-swiftclient is developed and maintained by the OpenStack Swift project, and because SwiftStack is built upon OpenStack Swift, there is no real integration work needed! If you have a SwiftStack object storage cluster running, then you can interact with it using python-swiftclient.

Documentation is provided both on OpenStack Swift’s web site and on SwiftStack’s documentation web sites

(<https://www.swiftstack.com/docs/integration/python-swiftclient.html>), and installation of python-swiftclient onto Mac OS X, Linux, or Windows

requires only two or three commands. Once installed, you can try a couple of the following example commands:

- **Get cluster status:** `swift -A http://<cluster IP>/auth/v1.0 -U user -K password stat`



```

MacBook-Pro:~ chris$ swift -A http://192.168.57.41/auth/v1.0 -U user1 -K passwor
d stat
      Account: AUTH_user1
      Containers: 4
      Objects: 5
      Bytes: 64000
Containers in policy "2-replica": 1
  Objects in policy "2-replica": 0
  Bytes in policy "2-replica": 0
Containers in policy "3-replica": 3
  Objects in policy "3-replica": 5
  Bytes in policy "3-replica": 64000
Meta Temp-Url-Key: e3c2bbbe-112d-488f-affa-d6de1fd65bcb
Connection: keep-alive
X-Timestamp: 1415285459.06810
X-Trans-Id: tx34443ef0110c4c7aaa31b-00549996dc
Content-Type: text/plain; charset=utf-8
Accept-Ranges: bytes
MacBook-Pro:~ chris$

```

- **List containers in an account:** `swift -A http://<cluster IP>/auth/v1.0 -U user -K password list`
- **Create a new container:** `swift -A http://<cluster IP>/auth/v1.0 -U user -K password post photos`
- **Upload a file:** `swift -A http://<cluster IP>/auth/v1.0 -U user -K password upload photos photo.jpg`

## Solution - SwiftStack Software Defined Storage

Like many other areas of technology, the real solution to data storage isn't reached by throwing ever more complex hardware at the problem. Instead, intelligently designed software, running on interconnected general-purpose CPUs and commodity hard drives can take the place of proprietary silos of SAN/NAS storage.

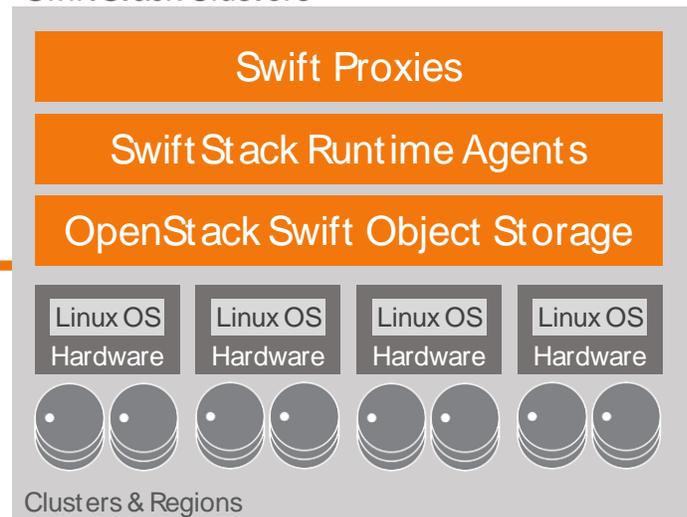
Software defined storage from SwiftStack brings several unique benefits to large-scale storage:

- Provides high performance for throughput-oriented applications like backup and recovery
- ...but with a TCO well below NAS, SAN, tape, and even public-cloud backup offerings
- Automatically replicates data between sites with minimal communications overhead
- Integrates the SwiftStack Filesystem Gateway to simplify use with legacy NFS/CIFS applications
- Includes everything in one price, based on how much data is stored—not the number of sites or nodes or raw hardware capacity
- Library support for many programming languages making automation and integration with other IT systems simple and efficient

### Swift Stack Controller



### Swift Stack Clusters



---

## Benefits - at - a - Glance

- Lowest cost on-premise cloud storage platform
- Performance and capacity scale linearly as nodes and gateways are added
- Highly durable architecture with no single-point of failure, triple-redundant replicas
- No vendor lock-in, uses industry-standard commodity servers and storage
- Easily monitor and manage both local and remote sites via the SwiftStack Controller
- Nodes and drives can be added, upgraded or replaced with no downtime
- One multi-tenant storage system to handle both backups and any unstructured data

---

## Key Features of SwiftStack

### Performance and Availability

SwiftStack's software defined design ensures that there is no single point of failure or performance bottlenecks, allowing you to easily scale up as needed to meet service levels and recovery objectives required for your business.

### Reliable and Durable

SwiftStack's automated replication ensures backups remain accessible with no loss of performance even during hardware and media failures. Every stored object is check-summed and saved with triple redundancy across the entire distributed system, and local replicas ensure that your backups are still available even if network connectivity is lost.

### Massively Uninterrupted Scalability

SwiftStack clusters scale by adding nodes and drives with no downtime or disruption. You can deploy, update, and expand your clusters and gateways non-disruptively with just a few clicks in the SwiftStack Controller.

### Cost Effective

SwiftStack runs on standard commodity server and storage hardware. You choose the type of storage, whether new or existing, heterogeneous or homogeneous, HDDs or SSDs, adding the capacity and performance you need when you need it - eliminating underutilized capacity and wasted resources.

### Easily Managed, Seamlessly Integrated

SwiftStack's Controller provides an easy to use unified dashboard for your entire deployment across all of your sites. SwiftStack easily integrates with enterprise IT systems for authentication, automated updates and recovery, monitoring and alerts, capacity management and chargeback tools. Storage policies make it simple to tailor replication as needed by location or type of backup.

### Truly Unified Storage

SwiftStack lets you deploy one platform to serve both backup/recovery as well as other unstructured data like website and application content, and user-generated files. Its flexibility enables it to be a

unified storage platform integrating legacy filesystems and object-based data storage without any limitations or restrictions.

---

## Supported 24 x 7

SwiftStack provides 24 x 7 enterprise-grade support and training to ensure your deployment is successful. Combined with the strongest pool of Swift expertise in the industry, SwiftStack's technical support let you focus IT resources on your users and applications, not on learning, building and supporting storage platform technology.

---

*“Swift is the most stable and widely- adopted part of OpenStack, with a successful operational track record at scale, and has become the most common implementation of object- based, S3-style storage.”*

## Find Out More

For detailed information on SwiftStack's capabilities please visit [SwiftStack.com](http://SwiftStack.com).

Copyright © 2014 SwiftStack Inc.