

# Validating OpenStack Object and Block Storage Performance

OpenStack is a free and open-source cloud computing software platform. Primarily deployed as an infrastructure as a service (IaaS) solution, it was initially developed by NASA and Rackspace. OpenStack provides a management layer for controlling, automating, and efficiently allocating resources. It empowers operators, system administrators and end users via self-service portals.

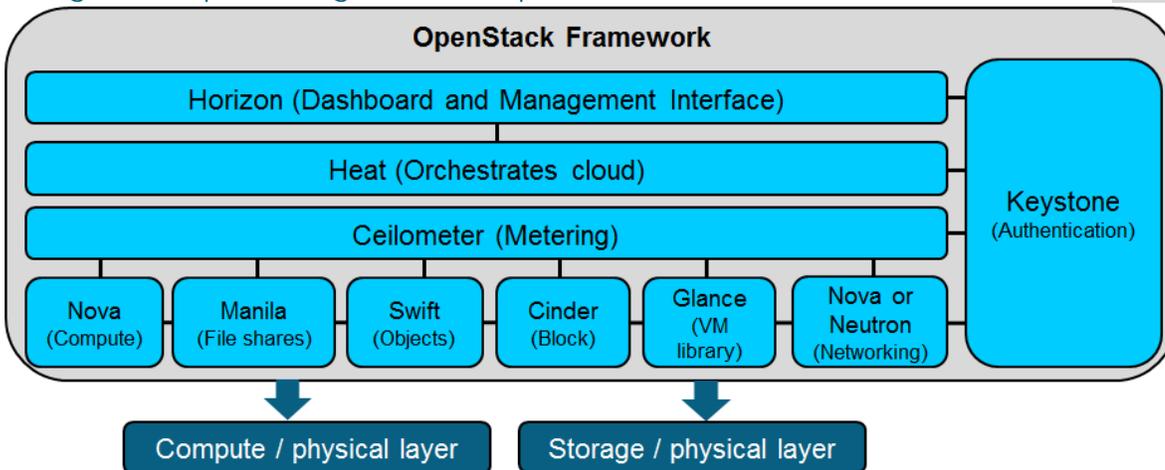


The Virtual Instruments OpenStack Storage performance validation solution provides coverage for two widely supported OpenStack storage protocols – Object (Swift) and Block (Cinder). It's designed to assess the maximum capacity and speed of OpenStack object and block storage infrastructures, servers and arrays. Virtual Instruments accomplishes this by first authenticating via the OpenStack Identify Service (Keystone), and then, by emulating up to millions of Swift and Cinder clients generating highly stressful and realistic requests to the servers. We then measure the performance, scalability, and response times of the servers in handling these requests, and verify the integrity of the stored contents.

The Virtual Instruments solution lowers storage costs, mitigates business risk, and increases storage staff productivity. These benefits justify the investment in the solution.”

**Jim Miller**  
 Analyst  
 EMA

Figure1: Simplified diagram of the OpenStack Framework.



Ceph is commonly deployed with OpenStack deployments to provide object, block and file storage. Ceph is an open source storage platform now backed by RedHat. Virtual Instruments enables testing of Swift, S3 and Cinder against a Ceph storage platform. Block testing requires you deploy a Linux LIO target on the same host as the Rados Gateway to enable access to Ceph via iSCSI.

## Virtual Instruments OpenStack Object Storage Testing and Validation Features

Object Storage is not a traditional file system, but rather a distributed storage fabric designed to support static data such as virtual machine images, photo storage, email storage, backups and archives. The OpenStack Object Storage (code-name Swift) protocol is a RESTful object access storage protocol that provides access to a scalable, redundant object storage fabric capable of storing petabytes of data. Swift also provides limited API compatibility for Amazon's S3 storage protocol.

Virtual Instruments allows you to model Swift workloads, then emulate and scale them to test different what-ifs scenarios and analyze the results to ensure it is able to meet your business objectives. Virtual Instruments provides easy to use out of the box workloads models for both Swift and Amazon S3. Enabling you to validate the performance of the different configurations possible within OpenStack native, open source storage such as Ceph and 3rd party storage arrays that support Swift or Amazon S3.

Some of the capabilities provided for OpenStack Object Storage are:

### Modeling

- Out-of-the-box validation model
- Looping constructs, parameter files, etc.
- Independent load profiles to access scalability
- Data compressibility and deduplicability algorithm

### Detailed Statistics

- KPI stats: Throughput, IOPS, Latency
- Per Swift operation statistics and HTTP and TCP layer statistics
- Data verification operation

### Client Emulation

- Create multiple scenarios from a single interface
- RESTful API with object validation support
- Emulation of millions of unique clients

### Command Types

- Open / close connection
- Authentication
- Retrieve / create / update / delete account metadata
- Create / list / delete container
- Retrieve / create / update / delete container metadata
- Create / update / copy / delete object

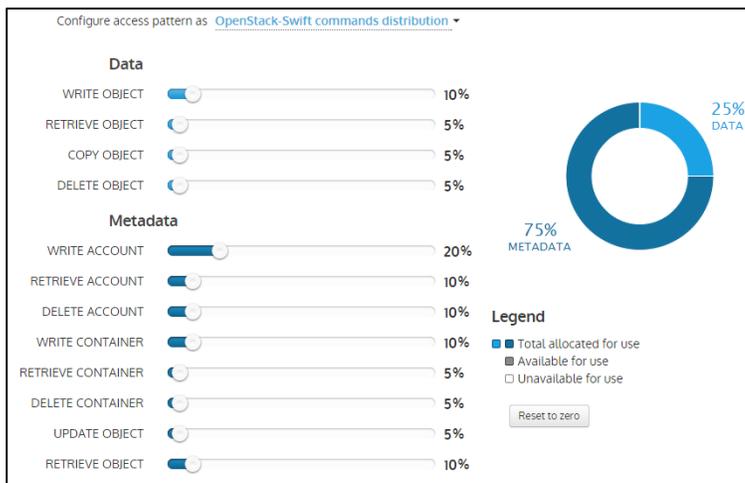


Figure 2: User friendly, out-of-the-box workloads for Swift and S3

## OpenStack Block Storage Testing and Validation Features

OpenStack compute instances are able to access persistent block level storage devices through the OpenStack Block Storage protocol (code-name Cinder). Cinder is a RESTful protocol that manages the creation, attaching and detaching of the block devices, snapshots and storage volumes. While Cinder is called the OpenStack Block Protocol it is really considered a control plane protocol as it does not perform data operations. Many vendors support provisioning and management via Cinder with the compute instances using data plane protocols like iSCSI or Fibre Channel to perform the data (read/write) operations.

Virtual Instruments allows you to model Cinder v1 and v2 workloads, then emulate and scale them to test different what-ifs scenarios, and analyze the results to ensure it is able to meet your business objectives. Virtual Instruments provides support for Cinder and the block protocols such as iSCSI and Fibre Channel needed to support data operations. This enables you to validate the performance of the entire OpenStack block storage environment and optimize the different configurations to ensure business results.

Some of the capabilities provided by Virtual Instruments for OpenStack Object Storage are:

### Modeling

- Out-of-the-box validation model
- Looping constructs, parameter files, etc.
- Independent load profiles to access scalability
- Data compressibility and deduplicability algorithm

### Detailed Statistics

- KPI stats: Throughput, IOPS, Latency
- HTTP and TCP layer statistics
- Data verification operations

### Client Emulation

- Create multiple scenarios from a single interface
- RESTful API and volume validation
- Emulation of millions of unique clients

## Commands

- Volumes
  - Create Volume
  - List Volume Summaries
  - List Volume Details
  - Show Volume
  - Update Volume
  - Delete Volume
  - List Volume Types
  - Show Volume Types
- Snapshots
  - Create Snapshot
  - List Snapshot Summaries
  - List Snapshot Details
  - Show Snapshot
  - Delete Snapshot
- Backup Extension
  - Create Backup
  - List Backup Summaries
  - List Backup Details
  - Show Backup
  - Delete Backup
  - Restore Backup

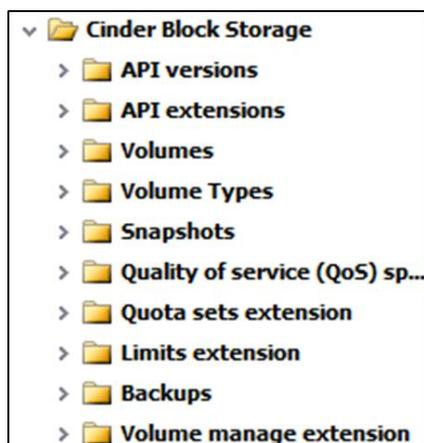


Figure 3: Cinder Command Categories

A screenshot of a configuration dialog box for a Cinder scenario. It shows various input parameters and their values.

Input	
Connection Handle	Default
Authentication	
Authentication method	Keystone
Keystone Handle	Default
API Version	v2
Request Headers	
Host	localhost
Content-Type	application/json
Accept	application/json
Transfer Encoding	None
Snapshot parameters	
Name	= @UP(0,SNAPSHOT)
Description	First snapshot
Volume ID	=@OUTPUT(29.1)
Force	true
Include Content-MD5	False

Figure 4: Cinder Scenario Configuration



Sales  
[Sales@virtualinstruments.com](mailto:Sales@virtualinstruments.com)  
1.888-522.2557

Website  
[virtualinstruments.com](http://virtualinstruments.com)