VMware vCloud Director® Object Storage Extension – Reference Design

Version 1.0
October 2019

Tomas Fojta
Senior Staff Architect, VCDX
Contents

Introduction ......................................................................................................................... 6
  1.1 Audience .................................................................................................................. 6

Use Cases .......................................................................................................................... 7
  2.1 Storing Content ....................................................................................................... 7
  2.2 Persistent Storage for Application ......................................................................... 9
  2.3 Storing vApp Templates and Catalog ................................................................... 10

Architecture ...................................................................................................................... 12
  3.1 Overview ................................................................................................................. 12
  3.2 Deployment Options ............................................................................................... 14
  3.3 Multisite Deployment .............................................................................................. 16

Test Environment Reference Benchmark ....................................................................... 17
  4.1 Test Setup .............................................................................................................. 17
  4.2 Bill of Materials ...................................................................................................... 18
  4.3 Test Results ............................................................................................................ 18
List of Tables
Table 1. Bill of material .................................................................................................................. 18
Table 2. HTTPs 10 MB Objects ....................................................................................................... 19
Table 3. HTTPs 1 MB Objects with various concurrency 10, 50, 100 and 200 workers ............... 19
Table 4. HTTPs 1 KB Objects 1000 workers across 30 buckets ..................................................... 20
Table 3. HTTPs 1 MB Objects with various concurrency 10, 50, 100 and 200 workers ............... 20
List of Figures

Figure 1. Objects Uploaded to a Bucket ........................................................................................................ 7
Figure 2. S3 Browser Application ............................................................................................................... 8
Figure 3. S3 Browser Configuration ........................................................................................................... 9
Figure 4. Application Credentials ............................................................................................................. 10
Figure 5. vApp Template Integration ....................................................................................................... 10
Figure 6. Catalog Integration .................................................................................................................... 11
Figure 7. Catalog Published Directly from Object Store ........................................................................... 11
Figure 8. Example of Scale Out vCloud Director Object Storage Extension Deployment with load balancing ........................................................................................................... 15
Figure 9. Multisite Architecture ................................................................................................................ 16
Figure 10. Testing topology ....................................................................................................................... 17
Introduction

The VMware vCloud Director® Object Storage Extension™ allows VMware Cloud Providers™ who are using VMware vCloud Director to offer object storage services to their end users. The extension acts as middleware which is tightly integrated with vCloud Director to abstract 3rd party S3 API compatible storage providers in multitenant fashion.

This guide provides information on how to properly design vCloud Director Object Storage Extensions on top of a vCloud Director infrastructure.

Note  This document is specific to versions 1.0 and 1.0.1 of vCloud Director Object Storage Extension and its integration with Cloudian HyperStore (S3 API compatible object storage appliance).

1.1 Audience

This document is intended for VMware Cloud Provider architects and technical leads responsible for planning and executing the deployment and upgrades of a VMware-based cloud environment.
Use Cases

vCloud Director natively provides Infrastructure as a Service through its integration with the underlying VMware vSphere platform. All native storage services such as storage for virtual machines, named (independent) disks, and catalog storage for virtual machine templates and media are using storage that is attached to vSphere ESXi hosts such as block storage, NFS or VMware vSAN.

There is however the need for highly scalable, durable, and network accessible storage that could be utilized by tenants, or their workloads without the dependency on the vSphere layer. The vCloud Director Object Storage Extension (vOSE) provides access to the object storage either through vCloud Director UI extension or via standardized S3 APIs. This allows existing applications to easily to access this new type of storage for various use cases.

2.1 Storing Content

Through the vCloud Director User Interface, vCloud Director users can create storage Buckets and upload, and tag unstructured files (objects) of various types. These files can be easily accessed with Uniform Resource Locator (URL) links and shared by defining Access Control Lists.

![Figure 1. Objects Uploaded to a Bucket](image-url)
Thanks to the full S3 API compatibility it is also possible to utilize existing 3rd party applications to upload and manage the files. The following figures display freeware S3 browser viewing the same bucket as above and its API access configuration.

Figure 2. S3 Browser Application
2.2 Persistent Storage for Application

Users can create application credentials with limited access to a specific bucket. This allows (stateless) applications running in vCloud Director (or outside) to persist their content such as configurations, logs or static data (web servers) into the object store. The application is using S3 API over the internet to upload and retrieve object data.
2.3 Storing vApp Templates and Catalog

Because of the close integration to vCloud Director, vCloud Director Object Storage Extension can directly capture and restore user’s vCloud Director vApps. Users can also share these vApps with other users. Thus, vCloud Director Object Storage Extension provides additional tier of storage for vApp templates that can be used, for example, for archiving old images.

A full vCloud Director catalog (consisting of vApp templates and media ISO images) can be captured from an existing Org VDC catalog or created from scratch by uploading individual ISO and OVA files to vCloud Director Object Storage Extension. Then, the catalog can be published, which allows any
vCloud Director organization (from any vCloud Director instance) to subscribe to the catalog. This enables easy distribution of specific catalogs publicly or for geographical distribution across vCloud Director instances.

Figure 6. Catalog Integration

Figure 7. Catalog Published Directly from Object Store
Architecture

3.1 Overview

vCloud Director Object Storage Extension is running externally to vCloud Director and integrates with it as UI extension. It has 1:1 relationship with the vCloud Director instance. In the current release, vCloud Director Object Storage Extension can be connected to one storage provider (Cloudian HyperStore). The provider can selectively enable vCloud Director organizations to consume the service. The unique counterparts for organizations and users are created at the storage provider. The users authenticate to the service with vCloud Director or S3 credentials and access it only through the extension. The provider can access underlying storage appliance directly in order to set quota or collect usage information for billing purposes.

The following diagram shows all required components for vCloud Director Object Storage Extension and includes the traffic flows. vCloud Director version 9.1 and later are supported. Next to the vCloud Director cells you need to deploy one or more (for high availability and scalability) RHEL/CentOS/Oracle Linux VM nodes (dark green in the picture) that run the vCloud Director Object Storage Extension service provided as RPM package. These VMs are essentially stateless and persist all their data in PostgreSQL DB. This could be vCloud Director external PostgreSQL DB (if available) or a dedicated database just for vCloud Director Object Storage Extension.
The service needs a public IP address as it runs (by default) on port 443. S3 API clients or the vCloud Director UI plugin will access this endpoint.

The extension VM nodes need to have access to vCloud API endpoint for user authentication and for the vApp/Catalog import/export functionality. Additionally, the extension VMs need fast access to the underlying object storage platform (Cloudian HyperStore). Cloudian HyperStore is fully distributed with a minimum supported deployment of three (fully equivalent) storage nodes and scales essentially indefinitely. Each storage node also provides UI/API functionality. Fast L4 load balancing should be used to forward the calls from vCloud Director Object Storage Extension to all storage nodes.

Cloudian HyperStore appliance S3, IAM and Admin APIs are running on separate TCP ports and need
to be accessible to extension VM nodes. The Cloudian Management Console must be accessible from
the vCloud Director Service Provider Admin Portal plugin for the redirection and is the only service that
requires load balancing with sticky sessions.

3.2 Deployment Options

Based on the use case, user target group, and expected service parameters (SLA, scalability), the
cloud provider can decide on the type of deployment.

3.2.1 Minimal Deployment

Usage: functionality testing, development

Requirement: Minimum resources required. No high availability, not supported for production.

- One RHEL/CentOS VM: (4 vCPU, 12 GB RAM, 120 GB HDD) running vCloud Director,
vCloud Director Object Storage Extension and PostgreSQL database (shared both for vCloud
Director and vCloud Director Object Storage Extension).
- vSphere/NSX: Optional for vCloud Director resources. When not available, vApp/Catalog
functionality is not available.
- Storage provider: Three CentOS virtual machines running Cloudian HyperStore (4 vCPUs, 32
GB RAM, 32+100 GB HDD)
- Load balancing: none or DNS round robin

3.2.2 Small Deployment

Usage: niche use cases

Requirement: Minimum resources required. High availability, supported for production.

- One or more RHEL/CentOS VMs for vCloud Director. External PostgreSQL database (used
both for vCloud Director and vCloud Director Object Storage Extension). NFS transfer share
needed when more than one vCloud Director cell is used. Protected with vSphere HAs.
- One CentOS VM: (4 vCPU, 8 GB RAM, 120 GB HDD) running vCloud Director Object Storage
Extension. Protected with vSphere HA.
- vSphere/NSX: As required for vCloud Director resources.
- Storage provider: Three CentOS virtual machines running Cloudian HyperStore (4 vCPUs, 32
GB RAM, 32+100 GB HDD on shared storage).
- Load balancing: vCloud Director cells and Cloudian HyperStore nodes load balancing
provided by NSX.

3.2.3 Medium Deployment

Usage: typical use cases

Requirement: High availability, supported for production.

- Multiple RHEL/CentOS or appliance VMs for vCloud Director. NFS transfer share. For non-
appliance form factor external PostgreSQL database.
- One or more CentOS VMs: (8 vCPU, 8 GB RAM, 120 GB HDD) running vCloud Director
Object Storage Extension. Protected with vSphere HA and optionally load balanced. If vCloud
Director is deployed in appliance form factor, external PostgreSQL database is needed, otherwise existing vCloud Director database can be reused).

- vSphere/NSX: As required for vCloud Director resources.
- Storage provider: Three CentOS virtual machines running Cloudian HyperStore on dedicated ESXi hosts with local disks (8 vCPUs, 64 GB RAM, 32 GB HDD + multiple large local disks).
- Load balancing: vCloud Director cells and Cloudian HyperStore nodes load balancing provided by NSX or external hardware load balancer.

### 3.2.4 Large Deployment

**Usage:** large scale, low cost per GB use cases

**Requirement:** High scale, performance and availability, supported for production.

- Multiple RHEL/CentOS or appliance VMs for vCloud Director. NFS transfer share. For non-appliance form factor external PostgreSQL database.
- Multiple CentOS VMs: (12 vCPU, 12 GB RAM, 120 GB HDD) running vCloud Director Object Storage Extension. If vCloud Director is deployed in appliance form factor, external HA PostgreSQL database is needed, otherwise existing vCloud Director database can be reused).
- vSphere/NSX: As required for vCloud Director resources.
- Storage provider: Three or more dedicated bare metal physical Cloudian HyperStore appliances.
- Load balancing: external hardware load balancer

The following figure displays how you scale out and load balance vCloud Director Object Storage Extension.

![Figure 8. Example of Scale Out vCloud Director Object Storage Extension Deployment with load balancing](image-url)
3.3 Multisite Deployment

vCloud Director Object Storage Extension version 1.0.1 adds support for vCloud Director multisite deployments where different vCloud Director instances are federated (associated) with each other with a trust relationship. As these instances can be deployed in different locations, the end-users can deploy their applications with higher level of resiliency and not be impacted by local datacenter outages.

Each vCloud Director instance has its own vCloud Director Object Storage Extension which communicates with a shared Cloudian HyperStore that is deployed in a multi-datacenter configuration. Objects are automatically replicated across all datacenters and vCloud Director users can access them through either vCloud Director or vCloud Director Object Storage Extension endpoint.

![Multisite Architecture Diagram]

Design recommendations related to the vCloud Director multisite architecture can be found in the Architecting Multi-site VMware vCloud Director whitepaper¹.

---

Test Environment Reference Benchmark

4.1 Test Setup

Production grade setup of vCloud Director and vCloud Director Object Storage Extension was deployed in the lab. Both vCloud Director and vCloud Director Object Storage Extension were deployed in three-node configuration. The object storage platform also consisted of three load balanced hardware appliances Cloudian Hyperstore 1508. The workloads were simulated by three VM nodes running COSBench\(^2\) software which is the industry standard benchmark tool for object storage. The effect of front-end load balancer on the test results was eliminated by connecting each COSBench node to one vCloud Director Object Storage Extension node. Cloudian HyperStore nodes were load balanced with NSX-V Load Balancer in L4 TCP accelerated mode.

In order to assess the impact of vCloud Director Object Storage Extension proxying of S3 APIs, same tests were performed directly to the Cloudian HyperStore as well (through the load balancer). The following diagram shows the network flows of the S3 API communication.

Note that HTTPS was used both for frontend traffic (COSBench to vCloud Director Object Storage Extension nodes), as well as for backend traffic (vCloud Director Object Storage Extension to Cloudian HyperStore or COSBench to Cloudian HyperStore).

![Testing topology diagram](https://github.com/intel-cloud/cosbench)

---

\(^2\) [https://github.com/intel-cloud/cosbench](https://github.com/intel-cloud/cosbench)
4.2 Bill of Materials

The following table lists the software and hardware components that were used to collect performance results for vCloud Director Object Storage Extension.

Table 1. Bill of Material

<table>
<thead>
<tr>
<th>Component</th>
<th>Count</th>
<th>Specifications</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>vCenter Server</td>
<td>1</td>
<td>6.7.0U2 (14070654)</td>
<td></td>
</tr>
<tr>
<td>ESXi</td>
<td>4</td>
<td>6.7.0U2 (13006603)</td>
<td>Dell PowerEdge R730xd Intel Xeon E5-2620, 128 GB RAM, NFS storage 5 TB</td>
</tr>
<tr>
<td>NSX-V</td>
<td>1</td>
<td>6.4.5 (13282012)</td>
<td>ESG LB: Quad-Large, accelerated SSL pass through</td>
</tr>
<tr>
<td>vCloud Director cells</td>
<td>3</td>
<td>9.7.0.2 (14046945)</td>
<td>Appliance deployment (2 CPU, 12 GB RAM, 132 GB HDD)</td>
</tr>
<tr>
<td>NFS Transfer Share</td>
<td>1</td>
<td>CentOS 7</td>
<td>CentOS 7 VM (2 vCPU, 4 GB RAM, 532 GB HDD)</td>
</tr>
<tr>
<td>vCloud Director Object Storage Extension nodes</td>
<td>3</td>
<td>1.0.1 (14920149.e17)</td>
<td>CentOS 7 VM (8 vCPUs, 8 GB RAM, 128 GB HDD)</td>
</tr>
<tr>
<td>Cloudian HyperStore</td>
<td>3</td>
<td>7.1.6</td>
<td>Hardware appliance: Hyperstore 15083</td>
</tr>
<tr>
<td>COSBench</td>
<td>3</td>
<td>0.4.2</td>
<td>Ubuntu VM (6 vCPUs, 8 GB RAM, 240 GB HDD)</td>
</tr>
</tbody>
</table>

4.3 Test Results

4.3.1 Scenario 1 – Large Objects

Workloads: 100 workers doing writes and reads to 25 containers with 10 MB objects

Step 0: Prepare data for read
Step 1: Write for 5 mins
Step 2: Read for 5 mins
Step 3: Delete for 5 mins
Step 4: Cleanup all buckets and objects

Table 2. HTTPs 10 MB Objects

<table>
<thead>
<tr>
<th>Test Type</th>
<th>Operation</th>
<th>Avg Response Time</th>
<th>Throughput</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>vOSE</td>
<td>Write</td>
<td>2,734.27 ms</td>
<td>36.57 op/s</td>
<td>365.7 MB/s</td>
</tr>
<tr>
<td>Cloudian</td>
<td>Write</td>
<td>2,319.54 ms</td>
<td>43.11 op/s</td>
<td>431.1 MB/s</td>
</tr>
<tr>
<td>vOSE</td>
<td>Read</td>
<td>2,413.13 ms</td>
<td>41.44 op/s</td>
<td>414.39 MB/s</td>
</tr>
<tr>
<td>Cloudian</td>
<td>Read</td>
<td>3,261.15 ms</td>
<td>30.66 op/s</td>
<td>306.65 MB/s</td>
</tr>
</tbody>
</table>

4.3.2 Scenario 2 – Concurrency Comparison

Workloads: Write, read and delete for object size 1 MB for different concurrency level (10 – 200 workers)

Step 0: Prepare data for read
Step 1: Write for 5 mins
Step 2: Read for 5 mins
Step 3: Delete for 5 mins
Step 4: Cleanup all buckets and objects

Table 3. HTTPs 1 MB Objects with various concurrency 10, 50, 100 and 200 workers

<table>
<thead>
<tr>
<th>Test Type</th>
<th>Operation</th>
<th>Avg Response Time</th>
<th>Throughput</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>vOSE: 10 W</td>
<td>Write</td>
<td>86.29 ms</td>
<td>115.75 op/s</td>
<td>115.75 MB/s</td>
</tr>
<tr>
<td>Cloudian: 10 W</td>
<td>Write</td>
<td>72.08 ms</td>
<td>138.55 op/s</td>
<td>138.55 MB/s</td>
</tr>
<tr>
<td>vOSE: 10 W</td>
<td>Read</td>
<td>79.62 ms</td>
<td>125.58 op/s</td>
<td>125.58 MB/s</td>
</tr>
<tr>
<td>Cloudian: 10 W</td>
<td>Read</td>
<td>61.67 ms</td>
<td>162.11 op/s</td>
<td>162.11 MB/s</td>
</tr>
<tr>
<td>vOSE: 50 W</td>
<td>Write</td>
<td>145.79 ms</td>
<td>342.72 op/s</td>
<td>342.72 MB/s</td>
</tr>
<tr>
<td>Cloudian: 50 W</td>
<td>Write</td>
<td>142.02 ms</td>
<td>351.87 op/s</td>
<td>351.87 MB/s</td>
</tr>
<tr>
<td>vOSE: 50 W</td>
<td>Read</td>
<td>135.42 ms</td>
<td>368.77 op/s</td>
<td>368.77 MB/s</td>
</tr>
<tr>
<td>Cloudian: 50 W</td>
<td>Read</td>
<td>129.44 ms</td>
<td>386.24 op/s</td>
<td>386.24 MB/s</td>
</tr>
<tr>
<td>vOSE: 100 W</td>
<td>Write</td>
<td>291.95 ms</td>
<td>342.38 op/s</td>
<td>342.38 MB/s</td>
</tr>
<tr>
<td>Cloudian: 100 W</td>
<td>Write</td>
<td>266.83 ms</td>
<td>374.65 op/s</td>
<td>374.65 MB/s</td>
</tr>
<tr>
<td>vOSE: 100 W</td>
<td>Read</td>
<td>257.45 ms</td>
<td>388.39 op/s</td>
<td>388.39 MB/s</td>
</tr>
<tr>
<td>Cloudian: 100 W</td>
<td>Read</td>
<td>237.13 ms</td>
<td>421.67 op/s</td>
<td>421.67 MB/s</td>
</tr>
</tbody>
</table>
4.3.3 Scenario 3 – Small Objects
Workloads: Write, read and delete for object size 1 KB with 1000 workers across 30 buckets
Step 1: 30 buckets with each bucket having 100,000 objects
Step 2: 50% read, 50% write for 1 hour
Step 3: Cleanup all buckets and objects

Table 4. HTTPs 1 KB Objects 1000 workers across 30 buckets

<table>
<thead>
<tr>
<th>Test Type</th>
<th>Operation</th>
<th>Avg Response Time</th>
<th>Throughput</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>vOSE: 200 W</td>
<td>Write</td>
<td>542.04 ms</td>
<td>368.89 op/s</td>
<td>368.89 MB/s</td>
</tr>
<tr>
<td>Cloudian: 200 W</td>
<td>Write</td>
<td>470.09 ms</td>
<td>425.38 op/s</td>
<td>425.38 MB/s</td>
</tr>
<tr>
<td>vOSE: 200 W</td>
<td>Read</td>
<td>491.12 ms</td>
<td>407.21 op/s</td>
<td>407.21 MB/s</td>
</tr>
<tr>
<td>Cloudian: 200 W</td>
<td>Read</td>
<td>482.26 ms</td>
<td>414.68 op/s</td>
<td>414.68 MB/s</td>
</tr>
</tbody>
</table>

4.3.4 Scenario 4 – Object Size Comparison
Workloads: Write, read and delete for various object sizes ranging from 1 MB – 1 GB with 100 workers across 100 buckets

Table 5. HTTPs 1 MB Objects with various concurrency 10, 50, 100 and 200 workers

<table>
<thead>
<tr>
<th>Test Type</th>
<th>Operation</th>
<th>Avg Response Time</th>
<th>Throughput</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>vOSE: 1 MB</td>
<td>Write</td>
<td>260.9 ms</td>
<td>385.1 op/s</td>
<td>385.1 MB/s</td>
</tr>
<tr>
<td>Cloudian: 1 MB</td>
<td>Write</td>
<td>284.61 ms</td>
<td>352.75 op/s</td>
<td>352.75 MB/s</td>
</tr>
<tr>
<td>vOSE: 1 MB</td>
<td>Read</td>
<td>240.84 ms</td>
<td>418.49 op/s</td>
<td>418.49 MB/s</td>
</tr>
<tr>
<td>Cloudian: 1 MB</td>
<td>Read</td>
<td>269.41 ms</td>
<td>371.76 op/s</td>
<td>371.76 MB/s</td>
</tr>
<tr>
<td>vOSE: 10 MB</td>
<td>Write</td>
<td>2,640.59 ms</td>
<td>38.47 op/s</td>
<td>384.66 MB/s</td>
</tr>
</tbody>
</table>
### 4.3.5 Conclusion

As can be seen from the above test results, vCloud Director Object Storage Extension performance is very much in line with the pure storage platform performance and does not add significant overhead with maximums around 5-10%.