



Overview

Product Description

The Cisco Cloud Object Storage (COS) provides distributed, resilient, high-performance storage and retrieval of binary large object (blob) data. The primary interface for managing COS content is the OpenStack Swift API, with enhancements that improve the quality of service when accessing large media objects.

With COS, storage is distributed across a cluster of hardware systems, or nodes. The storage cluster is resilient against hard drive failure within a node and against node failure within the cluster. Nodes may be added to or removed from the cluster as needed to provide for changes in cluster capacity. To administer the cluster, COS includes an HTTP-based cluster-management API.

COS also includes an authentication and authorization service that implements the OpenStack Swauth API.

DDN WOS Archive Object Support

Beginning with Release 3.5.2, COS adds API support for Archive objects as used in the DataDirect Networks (DDN) Web Object Scaler (WOS) video streaming solution. See [Swift API, page 4-1](#) for details.

COS and MOS

COS is designed to integrate transparently with the Cisco Media Origination System (MOS), which is designed for highly optimized ingest and storage. MOS uses a hierarchical storage design that supports huge content libraries while simplifying content storage management. Its distributed architecture can separate ingest and storage from streaming, allowing each function to be scaled independently as needed to dynamically increase network ingest and storage resources.

COS and Cloud DVR

Beginning with Release 3.8.1, COS adds support for API calls that enable COS to manage fanout storage operations for applications such as Cloud DVR (cDVR). Fanout storage efficiently supports unique copies for fair-use compliance. A single fanout request can save many copies of an object, thereby saving network resources by optimizing storage compute and disk utilization. The COS Fanout API includes

calls to create, retrieve, and delete fanout objects and to create, retrieve, and delete individual copies of content within a fanout object. The Fanout API also enables interoperability between COS and Cisco Virtual Media Recorder (VMR) as part of a complete cDVR solution under shared MOS management.

Components

COS has a number of subsystems:

- **Networks:** Interfaces are grouped into distinct networks to isolate management functions from high-volume data traffic. The client applications use Swauth API to interact with the COS authentication and authorization services, and the Swift API to interact with the COS object storage services.
- **Clusters and Nodes:** COS services are provided by a cluster of nodes, with both the cluster and the individual nodes as distinctly manageable components.
- **Object Metadata Store:** The metadata for the cluster is stored in a high-performance distributed NoSQL database hosted on the COS nodes in a cluster.
- **Platform and Applications Manager (PAM):** COS components are managed using services running on the Platform and Applications Management system (PAM).
- **Hardware Platforms:** COS software is deployed on Cisco Content Delivery Engine (CDE) platforms, and will also be available on Cisco UCS Dense Hardware.

API Features

- [Overview, page 1-3](#)
- [Service Manager API, page 1-4](#)
- [Swauth API, page 1-5](#)
- [Swift Object Store API, page 1-5](#)
- [DDN WOS API, page 1-6](#)
- [COS Configuration API, page 1-7](#)
- [Fanout API, page 1-7](#)

Overview

The table below provides an overview of the COS APIs

Table 1-1 Overview of COS APIs

Feature Set	Features
Service Manager API	<ul style="list-style-type: none"> • A subset of the Cisco Media Origination System (MOS) APIs • Used to provision and configure a COS cluster and COS cluster nodes • Uses the FQDN of the Service Manager and HTTPS over port 8043
Swauth API	<ul style="list-style-type: none"> • Simple Auth Service API for authentication of Swift operations • Based on Swauth Open-Source Middleware API • Used to manage accounts, users, and account service endpoints • Uses the Authentication FQDN of the COS cluster and HTTP over port 80
Swift Object Store API	<ul style="list-style-type: none"> • An implementation of a subset of the continually evolving OpenStack Swift API • Command executions are authenticated using auth tokens provided by Swauth service • Used to create and manage containers and objects for persistent storage in a COS cluster • Uses the Storage FQDN of the COS cluster and HTTP over port 80
COS Configuration API	<ul style="list-style-type: none"> • Used to retrieve or update cluster-wide settings for the Swift, Swauth, Fanou, and WOS APIs. • Also used to retrieve or update the COS daemon (cosd) settings shared by all COS nodes. • Lets you enable or disable optional API features and configure some limits for API parameters.

Table 1-1 Overview of COS APIs

Feature Set	Features
WOS API	<ul style="list-style-type: none"> • Adds API support for Archive objects as used in the DataDirect Networks (DDN) Web Object Scaler (WOS) video streaming solution. • Includes calls to retrieve, reserve, create, and delete WOS objects.
COS Configuration API	<ul style="list-style-type: none"> • Used to retrieve or update cluster-wide settings for the Swift, Swauth, Fanout, and WOS APIs. • Also used to retrieve or update the COS daemon (cosd) settings shared by all COS nodes. • Lets you enable or disable optional API features and configure some limits for API parameters.
Fanout API	<ul style="list-style-type: none"> • Fanout API calls enable COS to manage fanout storage operations for applications such as Cloud DVR (cDVR). • <i>Fanout</i> refers to storing and retrieving multiple copies of media content to support fair-use compliance. • <i>Fanout objects</i> are units of content accessed by a single URL, with each copy accessed by an index in a request header. • One fanout request can save many copies of an object. This saves network resources by optimizing storage compute and disk utilization. • Includes calls to create, retrieve, and delete fanout objects and to create, retrieve, and delete individual copies of media content within a fanout object.

**Note**

The COS cluster is assigned an Authentication FQDN (used with the Swauth API) and a Storage FQDN (used with the Swift API). Currently the Authentication FQDN and the Storage FQDN must be the same, for example, auth01.cos.acme.com.

Service Manager API

The COS Service Manager API is derived from the MOS APIs. The Service Manager API provides the following functions:

- Listing the region
- Listing, creating, deleting, and modifying zones
- Listing, creating, deleting, and modifying IP pools
- Listing, creating, deleting, and modifying a COS cluster
- Adding a COS node to a COS cluster
- Listing, editing, and enabling a COS instance

- Listing, creating, deleting, and modifying a COS service endpoint
- Listing, creating, deleting, and modifying an asset workflow template
- Listing and modifying COS nodes
- Configuring a COS node service interface
- Viewing the COS node status
- Viewing the COS node management events

For a detailed description of these functions, see [Service Manager API, page 2-1](#).

Swauth API

COS includes a basic authentication service that can be used when COS is not installed along with other OpenStack services such as the Keystone Identity service. The API for the COS authentication service is derived from the OpenStack Swauth middleware component API. The authentication service API provides the following functions for managing accounts, users, and service endpoints:

- Listing Accounts
- Retrieving Account Details
- Creating an Account
- Deleting an Account
- Creating or Updating a User
- Retrieving User Details
- Deleting a User
- Creating or Updating Account Service Endpoints
- Getting an Authentication Token

For a detailed description of these functions, see [Swauth API, page 3-1](#).

**Note**

The COS cluster is assigned an Authentication FQDN (used with the Swauth API) and a Storage FQDN (used with the Swift API). Currently the Authentication FQDN and the Storage FQDN must be the same, for example, auth01.cos.acme.com.

Swift Object Store API

The COS object storage API is based on the OpenStack Swift API. It is implemented as a set of Representational State Transfer (RESTful) web services. All account, container, and object operations can be performed with standard HTTP calls. The requests are directed to the host and URL described in the X-Storage-Url HTTP header that is part of the response to a successful request for an authentication token.

The COS object storage API defines restrictions on HTTP requests. These restrictions, borrowed from the Swift API, are listed in the table below.

Table 1-2 **COS API Restrictions**

Constraint	Value
Maximum # of HTTP Headers per request	90
Maximum length of all HTTP Headers	4096 bytes
Maximum length per HTTP request line	8192 bytes
Maximum length of container name	256 bytes
Maximum length of object name	1024 bytes

Also, the container and object names must be UTF-8 encoded and then URL-encoded before inclusion in the HTTP request line.



Note

All the length restrictions are enforced against the URL-encoded request line.

The COS object store API provides the following functions, some of which provide extended functionality beyond the standard SWIFT API defined by OpenStack:

- Listing Containers
- Listing Objects
- Creating a Container
- Deleting a Container
- Retrieving an Object
- Creating or Updating an Object
- Deleting an Object
- Creating or Updating Container Metadata
- Retrieving Container Metadata
- Deleting Container Metadata
- Retrieving Object Metadata

For a detailed description of these functions, see [Swift API, page 4-1](#).



Note

The COS cluster is assigned an Authentication FQDN (used with the Swauth API) and a Storage FQDN (used with the Swift API). Currently the Authentication FQDN and the Storage FQDN must be the same, for example, auth01.cos.acme.com.

DDN WOS API

Beginning with Release 3.8.1, COS adds API support for Archive objects as used in the DataDirect Networks (DDN) Web Object Scaler (WOS) video streaming solution. The WOS API provides the following functions for DDN WOS objects:

- Retrieving an Archive (DDN WOS) Object

- Reserving an Archive (DDN WOS) Object Identifier
- Creating an Archive (DDN WOS) Object
- Deleting an Archive (DDN WOS) Object

For a detailed description of these functions, see [WOS API, page 5-1](#).

COS Configuration API

The COS service configuration API lets you retrieve or update cluster-wide settings for the Swift API, the Swauth API, the Fanout API, the WOS API, and the COS daemon (cosd) settings shared by all COS nodes. The COS service configuration API lets you enable or disable optional API features, and also configure some of the limits for API parameters. This API provides the following functions:

- Retrieving Non-Sensitive Configuration Settings
- Retrieving Configuration Settings
- Updating Configuration Settings

For a detailed description of these functions, see [COS Configuration API, page 6-1](#).

Fanout API

This API enables COS to manage fanout storage operations for applications such as Cloud DVR (cDVR). *Fanout* refers to storing and retrieving multiple copies of specified media content to support unique copies for fair-use compliance. A *fanout object* is a single logical unit of media content that can represent one or more exact copies of the content. The logical unit is accessed by a single URL, and each copy is accessed by an index contained in a request header. By saving many copies with one request, the Fanout API saves network resources by optimizing storage compute and disk utilization.

The Fanout API supports the following functions for fanout objects:

- List VMR Objects
- Create Fanout Object
- Retrieve Fanout Object
- Retrieve Fanout Object Metadata
- Delete Fanout Object
- Delete Copy Within Fanout Object
- Create Non-Fanout Object
- Access Non-Fanout Object
- Delete Non-Fanout Object

For a detailed description of these functions, see [Fanout API, page 7-1](#).

Restrictions and Limitations

- The OpenStack Swift and Swauth APIs continue to evolve. COS does not currently implement all the Swift or Swauth API functions. For a list of supported functions, see [Swift Object Store API, page 1-5](#) and [Swauth API, page 1-5](#).

- Secure Sockets Layer (SSL) or other means for providing session security and encryption are not supported with the Swift and Swauth APIs.
- The Service Manager API support access using HTTPS over port 8043.
- COS Release 3.8.1 does not support upgrade from or downgrade to any earlier COS release.
- See the *Release Notes for Cisco Cloud Object Storage 3.8.1* for open caveats and known issues related to this release.