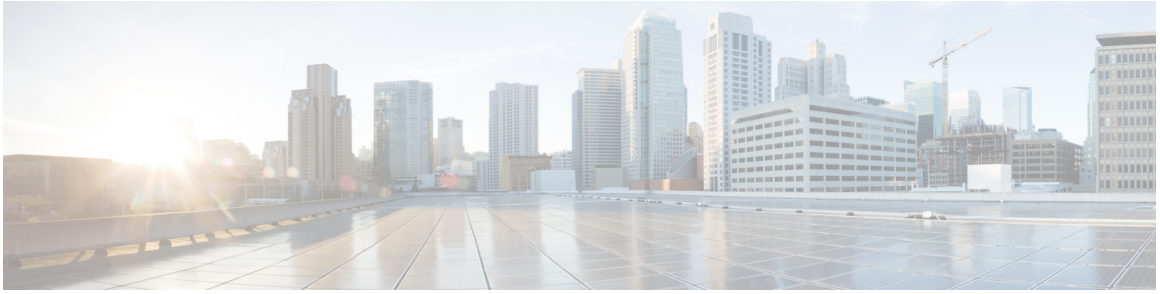
Cisco Cloud Object Storage Release 3.16.1 API Guide

October 14, 2017

Cisco Systems, Inc.
www.cisco.com

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco website at www.cisco.com/go/offices.

CONTENTS



Preface

This preface describes who should read the *Cisco Cloud Object Storage Release 3.16.1 API Guide*, how it is organized, and its document conventions. It contains the following sections:

- Audience
- Document Organization
- Document Conventions
- Related Publications
- Obtaining Documentation and Submitting a Service Request

Audience

This application program interface (API) guide is written for the knowledgeable application programmer who understands the basic architecture of the Cisco Cloud Object Storage (COS) product and Java servlets. The user should be fluent in the Java programming language and have prior practical experience developing content networking solutions. This guide is not intended to direct the user in how to program in the Java language and limits itself to describing how related COS software servlets are used.

Document Organization

This document contains the following chapters and appendices:

Chapters or Appendices	Descriptions
Overview	Introduces COS and the COS software APIs.
Service Manager API	Describes the subset of the V2PC APIs that are implemented for COS.
Swauth API	Describes the subset of the OpenStack Swauth API that is implemented for the COS authentication service.
Swift API	Describes the subset of the OpenStack Swift API that is implemented for COS.
WOS API	Describes COS support for Archive objects as used in the DataDirect Networks (DDN) Web Object Scaler (WOS) video streaming solution.

Chapters or Appendices	Descriptions
COS Configuration API	Describes the COS service configuration API used to retrieve or update cluster-wide settings for other COS APIs and shared COS daemon (cosd) settings.
Fanout API	Describes the COS Fanout API, which enables COS to manage fanout storage operations for applications such as Cloud DVR (cDVR).
Example API Calls	Provides examples for making Service Manager, Swauth, and Swift API calls using curl.

Document Conventions

This document uses the following conventions:

Convention	Indication
bold font	Commands and keywords and user-entered text appear in **bold** font.
italic font	Document titles, new or emphasized terms, and arguments for which you supply values are in *italic* font.
[ ]	Elements in square brackets are optional.
{ x | y | z }	Required alternative keywords are grouped in braces and separated by vertical bars.
[x | y | z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
`courier` font	Terminal sessions and information the system displays appear in `courier` font.
< >	Nonprinting characters such as passwords are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Tip Means *the following information will help you solve a problem*. The tips information might not be troubleshooting or even an action, but could be useful information, similar to a Timesaver.

Caution Means *reader be careful*. In this situation, you might perform an action that could result in equipment damage or loss of data.

Timesaver Means *the described action saves time*. You can save time by performing the action described in the paragraph.

Warning

IMPORTANT SAFETY INSTRUCTIONS

This warning symbol means danger. You are in a situation that could cause bodily injury. Before you work on any equipment, be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents. Use the statement number provided at the end of each warning to locate its translation in the translated safety warnings that accompanied this device.

SAVE THESE INSTRUCTIONS

Warning **Statements using this symbol are provided for additional information and to comply with regulatory and customer requirements.**

Related Publications

Refer to the following documents for additional information about COS COS:

- *Cisco Cloud Object Storage Release 3.16.1 User Guide*
- *Cisco Virtualized Video Processing Controller User Guide*
- *Cisco Cloud Object Storage Release 3.14.1 Troubleshooting Guide*
- *Cisco UCS S3260 Storage Server Installation and Service Guide*
- *Cisco UCS C3160 Rack Server Installation and Service Guide*
- *Cisco Content Delivery Engine 465 Hardware Installation Guide*
- *Release Notes for COS 3.16.1*
- *Open Source Used in COS 3.16.1*

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see *What's New in Cisco Product Documentation* at: http://www.cisco.com/c/en/us/td/docs/general/whatsnew/whatsnew.html.

Subscribe to *What's New in Cisco Product Documentation*, which lists all new and revised Cisco technical documentation, as an RSS feed and deliver content directly to your desktop using a reader application. The RSS feeds are a free service.

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.
- **Nodes, Sites, and Clusters:** A COS installation includes one or more individual COS servers, or nodes. Nodes are configured into sites, which may (but need not) correspond to geographic regions, and also into clusters, which may (but need not) contain multiple sites.
- **Object Metadata Store:** The metadata for the cluster is stored in a high-performance distributed NoSQL database hosted on either the COS nodes in a cluster or a dedicated COS metadata cluster (CMC). Using a separate CMC allows COS object data and metadata to scale independently.
- **Virtualized Video Processing Controller (V2PC):** COS 3.16.1 components are managed using services running on the V2PC.
- **Hardware Platforms:** COS software is currently deployed on selected Cisco Content Delivery Engine (CDE) and Cisco UCS server hardware models.

API Features

Overview

Table 1-1 provides an overview of the COS APIs

Table 1-1 Overview of COS APIs

Feature Set	Features
Service Manager API	• A subset of the Cisco Virtualized Video Processing Controller (V2PC) 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	• 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	• 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.

Table 1-1 Overview of COS APIs

Feature Set	Features
WOS API	• 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	• 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	• Fanout API calls enable COS to manage fanout storage operations for applications such as Cloud DVR (cDVR). *Fanout* refers to storing and retrieving multiple copies of media content to support fair-use compliance. *Fanout objects* 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 management APIs for COS Release 3.16.1 are accessed through V2PC Service Manager (SM). The SM API uses the fully qualified domain name (FQDN) of SM and HTTPS over port 8043.

Note Customer access to the SM API is controlled by a secure token that Cisco provides to customers. To access the secure token value on an installed system, see the **/etc/opt/cisco/mos/public/token.json** file on any V2PC master node.

The SM API provides the following functions:

- Listing, creating, deleting, and updating IP pools
- Listing, creating, deleting, and updating COS metadata clusters
- Listing, creating, deleting, and updating COS clusters
- Listing, creating, deleting, and updating COS nodes
- Viewing node status
- Listing, creating, deleting, and updating node profiles

For a detailed description of these functions, see Service Manager API, page 2-1.

Note Alarms, events, and statistics APIs are documented in the *Cisco V2PC Release 3.3 API Guide*.

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, Swauth API, Fanout API, WOS API, and 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 1-6 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.
- See the *Release Notes for Cisco Cloud Object Storage 3.16.1* for open caveats and known issues related to this release.

Service Manager API

The management APIs for COS Release 3.16.1 are accessed through V2PC Service Manager (SM). The SM API uses the fully qualified domain name (FQDN) of SM and HTTPS over port 8043.

Note Customer access to the SM API is controlled by a secure token that Cisco provides to customers. To access the secure token value on an installed system, see the **/etc/opt/cisco/mos/public/token.json** file on any V2PC master node.

API Request Format

All requests for SM COS Management APIs use the following general format:

Request Type	Request Format
List All Objects	GET https://<FQDN>:8043/v2/<object_type>/
List Specific Object	GET https://<FQDN>:8043/v2/<object_type>/<object_instance_name>
Create Specific Object	POST https://<FQDN>:8043/v2/<object_type>/<object_instance_name> {<object body>}
Update Specific Object	PUT https://<FQDN>:8043/v2/<object_type>/<object_instance_name> {<object body>}
Delete Specific Object	DELETE https://<FQDN>:8043/v2/<object_type>/<object_instance_name>

The following headers are required to access the COS APIs on V2PC SM:

Header	Value
Authorization	Secure token provided by Cisco, represented in examples as <auth_token>
Content-Type	application/json

Listing, Creating, Updating, and Deleting IP Pools

An IP pool is a pool of static IP addresses within a subnet. The IP pool is only used to assign IP addresses to the data ports on a COS node. The operator must configure IP pools before deploying the COS nodes. After the IP pools are configured, they are typically used in the Node Profile object, which is accessed via the Node Profile API.

List All IP Pools

This request returns a list of all IP pool objects in the deployment.

Request Format

GET https://<FQDN>:8043/v2/cosippool

Authorization: bearer <auth_token>

Content-Type: application/json

Sample Response

```
HTTP/1.1 200 OK
Content-Type: application/json
[
  {
    "id": "smtenant_0.smcosippool.pool_B",
    "name": "pool_B",
    "type": "cosippool",
    "externalId": "/v2/cosippool/pool_B",
    "transactionId": "1f6fa4fd-fa03-4df2-9364-7d727e602716",
    "modified": "2017-07-07T21:25:52.981Z",
    "properties": {
      "addrType": "ipv4",
      "pool": [
        {
          "netmask": "255.255.252.0",
          "rangeStart": "20.0.82.1",
          "rangeEnd": "20.0.82.200",
          "gw": "20.0.80.1"
        }
      ]
    }
  },
  {
    "id": "smtenant_0.smcosippool.pool_A",
    "name": "pool_A",
    "type": "cosippool",
    "externalId": "/v2/cosippool/pool_A",
    "transactionId": "7a7d3389-b826-4b34-a684-7275870d9393",
    "modified": "2017-07-07T21:25:03.705Z",
    "properties": {
      "addrType": "ipv4",
      "pool": [
        {
          "netmask": "255.255.252.0",
          "rangeStart": "20.0.70.1",
          "rangeEnd": "20.0.70.200",
          "gw": "20.0.68.1"
        }
      ]
```

```
    }
  }
]
```

List One IP Pool

This request returns the IP pool named <pool_name>.

Request Fomat

GET https://<FQDN>:8043/v2/cosippool/<pool_name>

Authorization: bearer <auth_token>

Content-Type: application/json

Sample Response

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": "smtenant_0.smcosippool.pool_A",
  "name": "pool_A",
  "type": "cosippool",
  "externalId": "/v2/cosippool/pool_A",
  "transactionId": "7a7d3389-b826-4b34-a684-7275870d9393",
  "modified": "2017-07-07T21:25:03.705Z",
  "properties": {
    "addrType": "ipv4",
    "pool": [
      {
        "netmask": "255.255.252.0",
        "rangeStart": "20.0.70.1",
        "rangeEnd": "20.0.70.200",
        "gw": "20.0.68.1"
      }
    ]
  }
}
```

Create a New IP Pool

This request creates a new IP pool object named <pool_name>.

Request Format

POST https://<FQDN>:8043/v2/cosippool/<pool_name>

Authorization: bearer <auth_token>

Content-Type: application/json

Example

```
POST https://10.20.118.63:8043/v2/cosippool/pool_A
Authorization: bearer <auth_token>
Content-Type: application/json

{
    "properties": {
```

```
      "addrType": "ipv4",
      "pool": [
        {
          "netmask": "255.255.252.0",
          "rangeStart": "20.0.70.1",
          "rangeEnd": "20.0.70.200",
          "gw": "20.0.68.1"
        }
      ]
    }
}
```

Response Format

HTTP/1.1 200 OK

Update an IP Pool

This request updates an existing IP pool named <pool_name>.

Caution Updating an existing IP pool object, especially to reduce its range, is not recommended and will result in the **CosActiveIpPoolEdited** error event being generated. The SM APIs are asynchronous, so the API response will not result in errors unless the message body fails syntax validation.

Request Format

PUT https://<FQDN>:8043/v2/cosippool/<pool_name>

Authorization: bearer <auth_token>

Content-Type: application/json

Example

```
PUT https://<10.20.118.63>:8043/v2/cosippool/Pool_A
Authorization: bearer <auth_token>
Content-Type: application/json

{
    "properties": {
      "addrType": "ipv4",
      "pool": [
        {
          "netmask": "255.255.252.0",
          "rangeStart": "20.0.70.1",
          "rangeEnd": "20.0.70.220",
          "gw": "20.0.68.1"
        }
      ]
    }
}
```

Response Format

HTTP/1.1 200 OK

Delete an IP Pool

This request deletes an existing IP pool named <pool_name>.

Request Format

DELETE https://<FQDN>:8043/v2/cosippool/<pool_name>

Authorization: bearer <auth_token>

Content-Type: application/json

Example

```
DELETE  https://10.20.118.63:8043/v2/cosippool/pool_A
Authorization: bearer <auth_token>
Content-Type: application/json
```

Response Format

HTTP/1.1 200 OK

Caution You should only delete an existing IP pool after all IP addresses in the pool have been unassigned. If one or more IP addresses from the pool are in use, deletion of the IP pool deletes the object but does not automatically deallocate the pool's IP addresses. Attempts to delete an IP pool still in use results in the generation of the **CosActiveIpPoolDeleted** event.

Listing, Creating, Updating, and Deleting COS Metadata Clusters

A COS deployment has two types of bare metal nodes: COS nodes and COS Metadata Cluster (CMC) nodes. CMC nodes store the metadata associated with the content stored on COS nodes. All functional CMC nodes must be part of a CMC. The CMC is represented by the CMC object in the COS SM APIs.

List All CMCs

This request returns a list of all CMCs in the deployment.

Request Format

GET https://<FQDN>:8043/v2/coscmc

Authorization: bearer <auth_token>

Content-Type: application/json

Sample Response

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "id": "smtenant_0.smcoscmc.goliath-cmc-cluster",
    "name": "goliath-cmc-cluster",
    "type": "coscmc",
    "externalId": "/v2/coscmc/goliath-cmc-cluster",
```

```
    "transactionId": "091c899b-8bb0-4e97-9c89-d069395d766c",
    "modified": "2017-07-07T23:54:50.696Z",
    "properties": {
      "description": "",
      "cmcMembers": [
        {
          "mgmtIp": "20.0.60.237",
          "hostName": "cmc-7"
        },
        {
          "mgmtIp": "20.0.60.238",
          "hostName": "cmc-8"
        },
        {
          "mgmtIp": "20.0.60.239",
          "hostName": "cmc-9"
        },
        {
          "mgmtIp": "20.0.60.240",
          "hostName": "cmc-10"
        },
        {
          "mgmtIp": "20.0.60.236",
          "hostName": "cmc-6"
        }
      ],
      "replicationFactor": 3
    }
  }
]
```


List One CMC

This request returns a single CMC named <cluster_name>.

Request Format

GET https://<FQDN>:8043/v2/coscmc/<cluster_name>

Authorization: bearer <auth_token>

Content-Type: application/json

Sample Response


```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": "smtenant_0.smcoscmc.goliath-cmc-cluster",
  "name": "goliath-cmc-cluster",
  "type": "coscmc",
  "externalId": "/v2/coscmc/goliath-cmc-cluster",
  "transactionId": "091c899b-8bb0-4e97-9c89-d069395d766c",
  "modified": "2017-07-07T23:54:50.696Z",
  "properties": {
    "description": "",
    "cmcMembers": [
      {
        "mgmtIp": "20.0.60.237",
        "hostName": "cmc-7"
      },
```

```
      {
        "mgmtIp": "20.0.60.238",
        "hostName": "cmc-8"
      },
      {
        "mgmtIp": "20.0.60.239",
        "hostName": "cmc-9"
      },
      {
        "mgmtIp": "20.0.60.240",
        "hostName": "cmc-10"
      },
      {
        "mgmtIp": "20.0.60.236",
        "hostName": "cmc-6"
      }
    ],
    "replicationFactor": 3
  }
}
```

Create a New CMC

This request creates a new CMC object named <cluster_name>.

Request Format

POST https://<FQDN>:8043/v2/coscmc/<cluster_name>

Authorization: bearer <auth_token>

Content-Type: application/json

Example

```
POST https://10.20.118.64:8043/v2/coscmc/goliath-cmc-cluster
Authorization: bearer <auth_token>
Content-Type: application/json

{
  "properties": {
    "description": "",
    "cmcMembers": [],
    "replicationFactor": 3
  }
}
```

Note When creating a new CMC, the **cmcMembers** field must be declared as an empty array. This field is populated by the CMC nodes when they announce themselves to the COS application instance controller (AIC) following initialization.

Response

```
HTTP/1.1 200 OK
```

Update a CMC

This request updates an existing CMC named <cluster_name>. Updates to a CMC object must be restricted to the following:

- Changing the Description field, which can be done at any time.
- Changing the replication factor. Cisco only recommends changing the replication factor to 3 or 5, or to a value less than 3 if there are fewer than three nodes in the cluster.
- Deleting a node from a CMC by removing its entry from the cmcMembers array.

Caution Do not update the cmcMembers array entries to create new CMC nodes or modify existing ones.

Request Format

PUT https://<FQDN>:8043/v2/coscmc/<cluster_name>

Example

```
PUT https://10.20.118.64:8043/v2/coscmc/goliath-cmc-cluster
Authorization: bearer <auth_token>
Content-Type: application/json

{
  "properties": {
    "description": "",
    "cmcMembers": [
      {
        "mgmtIp": "20.0.60.237",
        "hostName": "cmc-7"
      },
      {
        "mgmtIp": "20.0.60.238",
        "hostName": "cmc-8"
      },
      {
        "mgmtIp": "20.0.60.239",
        "hostName": "cmc-9"
      },
      {
        "mgmtIp": "20.0.60.240",
        "hostName": "cmc-10"
      },
      {
        "mgmtIp": "20.0.60.236",
        "hostName": "cmc-6"
      }
    ],
    "replicationFactor": 5
  }
}
```

Response

```
HTTP/1.1 200 OK
```

Delete a CMC

This request deletes an existing CMC named <cluster_name>.

Note All CMC nodes must be removed from the CMC as described in Update a CMC, page 2-8 before deleting the CMC object.

Request Format

DELETE https://<FQDN>:8043/v2/coscmc/<cluster_name>

Authorization: bearer <auth_token>

Content-Type: application/json

Example

```
DELETE https://10.20.118.64:8043/v2/coscmc/goliath-cmc-cluster
Authorization: bearer <auth_token>
Content-Type: application/json
```

Response

```
HTTP/1.1 200 OK
```

Adding and Deleting CMC Nodes

Unlike COS nodes, CMC nodes are not represented by objects of their own, so they cannot be added or deleted as such. CMC nodes are added during initialization. The **cmcinit** script adds the node IP address and hostname to the cmcMembers array of the associated CMC object. In similar fashion, CMC nodes can be deleted from a cluster by removing their entries from the updating the cmcMembers array

Caution Do not try to add a new CMC node entry into the cmcMembers array of the CMC object. Instead, to delete or remove an existing CMC node, remove the CMC node entry from the CMC, and then update the cluster as described in Update a CMC, page 2-8.

Listing, Creating, Updating, and Deleting COS Clusters

A COS cluster is a logical entity that provides a service endpoint for the COS service offered by a cluster of COS nodes. The authorization profiles and content replication policies are also incorporated into the COS cluster for the purposes of COS deployment.

List All COS Clusters

This request returns a list of all COS clusters in the deployment.

Request Format

GET https://<FQDN>:8043/v2/cosclusters

Authorization: bearer <auth_token>

Content-Type: application/json

Example

```
GET https://10.20.118.64:8043/v2/cosclusters
Authorization: bearer <auth_token>
Content-Type: application/json

HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "id": "smtenant_0.smcosclusters.goliath-cluster",
    "name": "goliath-cluster",
    "type": "cosclusters",
    "externalId": "/v2/cosclusters/goliath-cluster",
    "transactionId": "864075f6-a923-4f38-91f8-ceee0bbacd82",
    "modified": "2017-07-07T21:32:56.064Z",
    "properties": {
      "storageFqdn": "auth01.goliath3-ext.com",
      "authFqdn": "auth01.goliath3-ext.com",
      "contentPolicy": {
        "localType": "erasurecoding",
        "remoteType": "erasurecoding",
        "localFailuresOrCopies": 1,
        "remoteFailuresOrCopies": 2,
        "localPolicy": "12:1",
        "remotePolicy": "16:2",
        "resiliencyGroupSize": 20
      },
      "authProfiles": [],
      "state": "enabled",
      "cmcRef": "smtenant_0.smcoscmc.goliath-cmc-cluster"
    }
  }
]
```

List One COS Cluster

This request returns a single COS cluster names <cluster_name>.

Request Format

GET https://<FQDN>:8043/v2/cosclusters/<cluster_name>

Authorization: bearer <auth_token>

Content-Type: application/json

Example

```
GET https://10.20.118.64:8043/v2/cosclusters/goliath-cluster
Authorization: bearer <auth_token>
Content-Type: application/json

HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": "smtenant_0.smcosclusters.goliath-cluster",
  "name": "goliath-cluster",
```

```
  "type": "cosclusters",
  "externalId": "/v2/cosclusters/goliath-cluster",
  "transactionId": "864075f6-a923-4f38-91f8-ceee0bbacd82",
  "modified": "2017-07-07T21:32:56.064Z",
  "properties": {
    "storageFqdn": "auth01.goliath3-ext.com",
    "authFqdn": "auth01.goliath3-ext.com",
    "contentPolicy": {
      "localType": "erasurecoding",
      "remoteType": "erasurecoding",
      "localFailuresOrCopies": 1,
      "remoteFailuresOrCopies": 2,
      "localPolicy": "12:1",
      "remotePolicy": "16:2",
      "resiliencyGroupSize": 20
    },
    "authProfiles": [],
    "state": "enabled",
    "cmcRef": "smtenant_0.smcoscmc.goliath-cmc-cluster"
  }
}
```

Create a New COS Cluster

This request creates a new COS cluster object named <cluster_name>.

Request Format

POST https://<FQDN>:8043/v2/cosclusters/<cluster_name>

Authorization: bearer <auth_token>

Content-Type: application/json

Example

```
POST https://10.20.118.64:8043/v2/cosclusters/goliath-cluster
Authorization: bearer <auth_token>
Content-Type: application/json

{
  "properties": {
    "storageFqdn": "auth01.goliath3-ext.com",
    "authFqdn": "auth01.goliath3-ext.com",
    "contentPolicy": {
      "localType": "erasurecoding",
      "remoteType": "erasurecoding",
      "localFailuresOrCopies": 1,
      "remoteFailuresOrCopies": 2,
      "localPolicy": "12:1",
      "remotePolicy": "16:2",
      "resiliencyGroupSize": 20
    },
    "authProfiles": [],
    "state": "enabled",
    "cmcRef": "smtenant_0.smcoscmc.goliath-cmc-cluster"
  }
}
```

Response

```
HTTP/1.1 200 OK
```

Update a COS Cluster

This request updates an existing COS cluster named <cluster_name>.

Request Format

PUT https://<FQDN>:8043/v2/cosclusters/<cluster_name>

Authorization: bearer <auth_token>

Content-Type: application/json

Example

```
PUT https://10.20.118.64:8043/v2/cosclusters/goliath-cluster
Authorization: bearer <auth_token>
Content-Type: application/json

{
  "properties": {
    "storageFqdn": "auth01.goliath3-ext.com",
    "authFqdn": "auth01.goliath3-ext.com",
    "contentPolicy": {
      "localType": "erasurecoding",
      "remoteType": "erasurecoding",
      "localFailuresOrCopies": 1,
      "remoteFailuresOrCopies": 2,
      "localPolicy": "12:1",
      "remotePolicy": "16:2",
      "resiliencyGroupSize": 20
    },
    "authProfiles": [],
    "state": "disabled",
    "cmcRef": "smtenant_0.smcoscmc.goliath-cmc-cluster"
  }
}
```

Response

```
HTTP/1.1 200 OK
```

Delete a COS Cluster

This request deletes an existing COS cluster named <cluster_name>.

Request Format

DELETE https://<FQDN>:8043/v2/cosclusters/<cluster_name>

Authorization: bearer <auth_token>

Content-Type: application/json

Example

```
DELETE https://10.20.118.64:8043/v2/cosclusters/goliath-cluster
Authorization: bearer <auth_token>
Content-Type: application/json
```

Response

```
HTTP/1.1 200 OK
```

Listing, Creating, Updating, and Deleting COS Nodes

The node specific configuration of a COS node is represented by the **cosallnodes** object.

List all COS Nodes

This request returns a list of all COS nodes in the deployment.

Request Format

GET https://<FWDN>:8043/v2/cosallnodes

Authorization: bearer <auth_token>

Content-Type: application/json

Example

```
GET https://10.20.118.64:8043/v2/cosallnodes
Authorization: bearer <auth_token>
Content-Type: application/json

HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "id": "smtenant_0.smcosallnodes.335574541",
    "name": "335574541",
    "type": "cosallnodes",
    "externalId": "/v2/cosallnodes/335574541",
    "transactionId": "53216631-b678-4b2d-b2f5-d3a0d98f9e2b",
    "modified": "2017-07-07T23:57:49.413Z",
    "properties": {
      "description": "c3260-g3a",
      "hostName": "c3260-g3a",
      "adminState": "inservice",
      "cosNodeClusterRef": "smtenant_0.smcosclusters.goliath-cluster",
      "model": "UCSC-C3K-4U5",
      "mgmtAddress": "20.0.118.13",
      "dataInterfaces": [
        {
          "name": "eth2",
          "ipPoolRef": "smtenant_0.smcosippool.pool_A",
          "enabled": true
        },
        {
          "name": "eth3",
          "ipPoolRef": "smtenant_0.smcosippool.pool_B",
          "enabled": true
        },
        {
          "name": "eth4",
          "ipPoolRef": "smtenant_0.smcosippool.pool_A",
          "enabled": true
        },
        {
          "name": "eth5",
          "ipPoolRef": "smtenant_0.smcosippool.pool_B",
          "enabled": true
        }
```

```
      ],
      "resiliencyGrp": "goliath-cluster:1"
    }
  },
  {
    "id": "smtenant_0.smcosallnodes.335574557",
    "name": "335574557",
    "type": "cosallnodes",
    "externalId": "/v2/cosallnodes/335574557",
    "transactionId": "821252a4-8e73-4b8b-aecf-4e35c197b301",
    "modified": "2017-07-08T00:05:41.801Z",
    "properties": {
      "description": "c3260-g6a",
      "hostName": "c3260-g6a",
      "adminState": "inservice",
      "cosNodeClusterRef": "smtenant_0.smcosclusters.goliath-cluster",
      "model": "UCSC-C3K-4U5",
      "mgmtAddress": "20.0.118.29",
      "dataInterfaces": [
        {
          "name": "eth2",
          "ipPoolRef": "smtenant_0.smcosippool.pool_A",
          "enabled": true
        },
        {
          "name": "eth3",
          "ipPoolRef": "smtenant_0.smcosippool.pool_B",
          "enabled": true
        },
        {
          "name": "eth4",
          "ipPoolRef": "smtenant_0.smcosippool.pool_A",
          "enabled": true
        },
        {
          "name": "eth5",
          "ipPoolRef": "smtenant_0.smcosippool.pool_B",
          "enabled": true
        }
      ],
      "resiliencyGrp": "goliath-cluster:1"
    }
  }
]
```


List One COS Node

This request returns a specific COS node named <node_name>.

Request Format

GET https://<FQDN>:8043/v2/cosallnodes/<node_name>

Authorization: bearer <auth_token>

Content-Type: application/json

Example

```
GET https://10.20.118.64:8043/v2/cosallnodes/335574541
Authorization: bearer <auth_token>
Content-Type: application/json
```

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": "smtenant_0.smcosallnodes.335574541",
  "name": "335574541",
  "type": "cosallnodes",
  "externalId": "/v2/cosallnodes/335574541",
  "transactionId": "53216631-b678-4b2d-b2f5-d3a0d98f9e2b",
  "modified": "2017-07-07T23:57:49.413Z",
  "properties": {
    "description": "c3260-g3a",
    "hostName": "c3260-g3a",
    "adminState": "inservice",
    "cosNodeClusterRef": "smtenant_0.smcosclusters.goliath-cluster",
    "model": "UCSC-C3K-4U5",
    "mgmtAddress": "20.0.118.13",
    "dataInterfaces": [
      {
        "name": "eth2",
        "ipPoolRef": "smtenant_0.smcosippool.pool_A",
        "enabled": true
      },
      {
        "name": "eth3",
        "ipPoolRef": "smtenant_0.smcosippool.pool_B",
        "enabled": true
      },
      {
        "name": "eth4",
        "ipPoolRef": "smtenant_0.smcosippool.pool_A",
        "enabled": true
      },
      {
        "name": "eth5",
        "ipPoolRef": "smtenant_0.smcosippool.pool_B",
        "enabled": true
      }
    ],
    "resiliencyGrp": "goliath-cluster:1"
  }
}
```

Create a New COS Node

Creation of COS nodes occurs automatically at COS initialization. When a physical COS node machine is initialized using the **cosinit** script, the node announces itself to the COS AIC by creating and posting a COS node object.

Caution Do not try to create a new COS node using COS APIs on V2PC SM.

Update a COS Node

After a COS node is created via cosinit, its description can be updated, its administrative state (adminState) can be switched between inservice and maintenance, and one or more if its network interfaces can be enabled or disabled.

Note It is not recommended that any other fields be updated via the update API.

This request updates an existing COS node object names <node_name>.

Request Format

PUT https://<FQDN>:8043/v2/cosallnodes/<node_name>

Authorization: bearer <auth_token>

Content-Type: application/json

Example

```
PUT https://10.20.118.64:8043/v2/cosallnodes/335574541
Authorization: bearer <auth_token>
Content-Type: application/json
{
  "properties": {
    "description": "c3260-g3a",
    "hostName": "c3260-g3a",
    "adminState": "inservice",
    "cosNodeClusterRef": "smtenant_0.smcosclusters.goliath-cluster",
    "model": "UCSC-C3K-4U5",
    "mgmtAddress": "20.0.118.13",
    "dataInterfaces": [
      {
        "name": "eth2",
        "ipPoolRef": "smtenant_0.smcosippool.pool_A",
        "enabled": true
      },
      {
        "name": "eth3",
        "ipPoolRef": "smtenant_0.smcosippool.pool_B",
        "enabled": true
      },
      {
        "name": "eth4",
        "ipPoolRef": "smtenant_0.smcosippool.pool_A",
        "enabled": true
      },
      {
        "name": "eth5",
        "ipPoolRef": "smtenant_0.smcosippool.pool_B",
        "enabled": true
      }
    ],
    "resiliencyGrp": "goliath-cluster:1"
  }
}
```

Response

```
HTTP/1.1 200 OK
```

Delete a COS Node

This request deletes an existing COS node named <node_name>.

Request Format

DELETE https://<FQDN>:8043/v2/cosallnodes/<node_name>

Authorization: bearer <auth_token>

Content-Type: application/json

Sample Request

DELETE https://10.20.118.64:8043/v2/cosallnodes/335574541

Authorization: bearer <auth_token>

Content-Type: application/json

Response

```
HTTP/1.1 200 OK
```

Viewing Node Status

Use the following requests to view the status of a COS or CMC node.

View All Nodes

This request returns a list of status objects for all COS and CMC nodes in the deployment.

Request Format

GET https://<FQDN>:8043/v2/cosallnodestatus/

Authorization: bearer <auth_token>

Content-Type: application/json

Sample Response

```
HTTP/1.1 200 OK
Content-Type: application/json

[
{
  "id": "smtenant_0.smcosallnodestatus.335574541",
  "name": "335574541",
  "type": "cosallnodestatus",
  "externalId": "/v2/cosallnodestatus/335574541",
  "transactionId": "6db702f2-32b0-4538-95f0-53e3cd23069b",
  "modified": "2017-07-28T04:39:02.655Z",
  "properties": {
    "hostname": "c3260-g3a",
    "management": "20.0.118.13",
    "timestamp": "1501216742548",
    "faultStatus": "warning",
    "faultDetails": [
      "Node Resiliency Status is in a warning state"
```

```
    ],
    "myCluster": "goliath-cluster",
    "nodeType": "COS",
    "myResGroupId": "goliath-cluster:1",
    "myResGroupStatus": "normal",
    "slaStatus": {
      "storageStatus": {
        "total": 260792,
        "used": 137027.57,
        "status": "normal",
        "description": "52.54% of storage is used"
      },
    "slaStatus": {
     "storageStatus": {
        "total": 582894,
        "used": 1009,
        "status": "normal",
        "description": "Metadata Storage Partition usage is 0.17%"
    },
    "serviceStatus": {
        "status": "warning",
        "description": "1 non-critical service(s) are down on the CMC Node"
    },
    "diskStatus": {
        "status": "normal",
        "description": "Disks on which metadata storage partition resides are ok"
    },
    "interfaceStatus": {
        "status": "critical",
        "description": "1/2 ( 50.00% ) interfaces are down"
    },
    "partitionStatus": {
        "status": "critical",
        "description": [
        "Metadata Storage has used 64.19% space and the partition is in critica
        status", "Metadata Commit Log has used 80.12% space and the partition is in
critical status", "CMC Application Logs has used 0.00% space and the partition
        is in normal status"
        ]
     }
    },"partitionStatus": {
        "status": "normal",
        "description": []
      },
      "serviceStatus": {
        "status": "normal",
        "description": "All COS Node services are functional"
      },
      "diskStatus": {
        "status": "normal",
        "description": "All disks are up and functional"
      },
      "interfaceStatus": {
        "status": "normal",
        "description": "All data interfaces are up"
      },
      "nodeResiliencyStatus": {
        "status": "warning",
        "description": [
          "GOIDs are in warning state. DEC repair ongoing? no. LEC repair ongoing? no"
        ],
        "LECInfo": {
          "status": "normal",
          "minDrives": 14,
```

```
            "available": 28
          },
          "GoidInfo": {
            "status": "warning",
            "DECrepair": "no",
            "LECrepair": "no"
          }
        },
        "clusterResiliencyStatus": {
          "status": "normal",
         "description": "Cluster Resiliency/DEC status is normal. Total nodes in resilience
group: 18. Active nodes: 18"
        }
      },
      "node": {
        "clusterAdminState": "enabled",
        "nodeAdminState": "inservice",
        "nodeStatus": "up",
        "statusDetails": "Node has sent configuration status to AIC",
        "nodeSoftware": "3.14.1-b33"
      },
      "service": {
        "services": [
          {
            "name": "Consul Agent",
            "state": "up",
            "critical": false
          },
          {
            "name": "NTP Daemon",
            "state": "up",
            "critical": false
          },
          {
            "name": "Cisco SNMP Agent",
            "state": "up",
            "critical": false
          },
          {
            "name": "Cisco Cloud Object Store Daemon",
            "state": "up",
            "critical": true
          },
          {
            "name": "Cisco Cache Server",
            "state": "up",
            "critical": true
          },
          {
            "name": "Sensu Client",
            "state": "up",
            "critical": false
          },
          {
            "name": "Monit",
            "state": "up",
            "critical": false
          }
        ]
      },
      "storage": {
        "partitions": [
          {
            "path": "/arroyo/log",
```

```
      "name": "COS Service Logs",
      "state": "normal",
      "total": 144497,
      "used": 19914,
      "partition": "/dev/sda3"
    }
  ],
  "disks": [
    {
      "name": "Cisco Disk 01",
      "state": "up"
    },
    {
      "name": "Cisco Disk 02",
      "state": "up"
    },
    {
      "name": "Cisco Disk 03",
      "state": "up"
    },
    {
      "name": "Cisco Disk 04",
      "state": "up"
    },
    {
      "name": "Cisco Disk 05",
      "state": "up"
    },
    {
      "name": "Cisco Disk 06",
      "state": "up"
    },
    {
      "name": "Cisco Disk 07",
      "state": "up"
    },
    {
      "name": "Cisco Disk 08",
      "state": "up"
    },
    {
      "name": "Cisco Disk 09",
      "state": "up"
    },
    {
      "name": "Cisco Disk 10",
      "state": "up"
    },
    {
      "name": "Cisco Disk 11",
      "state": "up"
    },
    {
      "name": "Cisco Disk 12",
      "state": "up"
    },
    {
      "name": "Cisco Disk 13",
      "state": "up"
    },
    {
      "name": "Cisco Disk 14",
      "state": "up"
    },
```

```
      {
        "name": "Cisco Disk 15",
        "state": "up"
      },
      {
        "name": "Cisco Disk 16",
        "state": "up"
      },
      {
        "name": "Cisco Disk 17",
        "state": "up"
      },
      {
        "name": "Cisco Disk 18",
        "state": "up"
      },
      {
        "name": "Cisco Disk 19",
        "state": "up"
      },
      {
        "name": "Cisco Disk 20",
        "state": "up"
      },
      {
        "name": "Cisco Disk 21",
        "state": "up"
      },
      {
        "name": "Cisco Disk 22",
        "state": "up"
      },
      {
        "name": "Cisco Disk 23",
        "state": "up"
      },
      {
        "name": "Cisco Disk 24",
        "state": "up"
      },
      {
        "name": "Cisco Disk 25",
        "state": "up"
      },
      {
        "name": "Cisco Disk 26",
        "state": "up"
      },
      {
        "name": "Cisco Disk 27",
        "state": "up"
      },
      {
        "name": "Cisco Disk 28",
        "state": "up"
      }
    ]
  },
  "net": {
    "interfaces": [
      {
        "name": "eth2",
        "state": "up",
        "inet": "20.0.70.1"
```

```
          },
          {
            "name": "eth3",
            "state": "up",
            "inet": "20.0.82.1"
          },
          {
            "name": "eth4",
            "state": "up",
            "inet": "20.0.70.2"
          },
          {
            "name": "eth5",
            "state": "up",
            "inet": "20.0.82.2"
          }
        ]
      }
    }
  },
  {
    "id": "smtenant_0.smcosallnodestatus.335559917",
    "name": "335559917",
    "type": "cosallnodestatus",
    "externalId": "/v2/cosallnodestatus/335559917",
    "transactionId": "15049ade-6f76-4c9f-89ce-9add66704be4",
    "modified": "2017-07-28T05:50:42.603Z",
    "properties": {
      "hostname": "cmc-7",
      "management": "20.0.60.237",
      "timestamp": "1501221042454",
      "faultStatus": "normal",
      "faultDetails": [
        "CMC Node is up and reporting heartbeat"
      ],
      "myCluster": "goliath-cmc-cluster",
      "nodeType": "CMC",
      "myResGroupId": "",
      "myResGroupStatus": "",
      "slaStatus": {
        "storageStatus": {
          "total": 6379042,
          "used": 7075,
          "status": "normal",
          "description": "Partition has used up 0.11% of metadata storage"
        },
  "partitionStatus": {
          "status": "normal",
          "description": []
        },
        "serviceStatus": {
          "status": "normal",
          "description": "All CMC Node services are functional"
        },
        "diskStatus": {
          "status": "normal",
          "description": "Disks on which metadata storage partition resides are ok"
        },
        "interfaceStatus": {
          "status": "normal",
          "description": "All data interfaces are up"
        }
      },
      "node": {
```

```
    "nodeAdminState": "unavailable",
    "nodeStatus": "up",
    "statusDetails": "Node is now operational",
    "nodeSoftware": ""
  },
  "service": {
    "services": [
      {
        "name": "Consul Agent",
        "state": "up",
        "critical": false
      },
      {
        "name": "NTP Daemon",
        "state": "up",
        "critical": false
      },
      {
        "name": "Cassandra",
        "state": "up",
        "critical": true
      }
    ]
  },
  "storage": {
    "partitions": [
      {
        "path": "/var/lib/cassandra/commitlog",
        "total": 6379042,
        "used": 7075,
        "name": "Metadata Commit Log",
        "partition": "/dev/sda4",
        "state": "normal"
      },
      {
        "path": "/var/log",
        "total": 6379042,
        "used": 7075,
        "name": "CMC Application Logs",
        "partition": "/dev/sda4",
        "state": "normal"
      },
      {
        "path": "/var/lib/cassandra/data",
        "total": 6379042,
        "used": 7075,
        "name": "Metadata Storage",
        "partition": "/dev/sda4",
        "state": "normal"
      }
    ],
    "disks": []
  },
  "net": {
    "interfaces": [
      {
        "name": "enp9s0",
        "state": "up",
        "speed": "10000",
        "inet": "20.0.60.237"
      }
    ]
  }
}
```

```
}
]
"storage": {
    "partitions": [
     {
        "name": "Metadata Storage",
        "state": "critical",
        "used": 201478,
        "total": 251478,
        "paths": [
     {
        "path": "/arroyodb/cos/data",
        "partition": "/dev/ssd1",
        "used": 201478,
        "total": 251478
     }
    ]
    },
    {
        "name": "Metadata Commit Log",
        "state": "normal",
        "used": 0.44,
        "total": 251478,
        "paths": [
    {
        "path": "/arroyodb/cos/commitlog",
        "partition": "/dev/ssd1",
        "used": 0.44,
        "total": 251478
    }
    ]
    },
    {
        "name": "COS Service Logs",
        "state": "normal",
        "used": 0.44,
        "total": 251478,
        "paths": [
    {
        "path": "/arroyo/log",
        "partition": "/dev/ssd2"
        "used": 0.44,
        "total": 251478
    }
    ]
    }
    ],
    For multiple partitions, the format appears as follows:
        "storage": {
         "partitions": [
            {
             "name": "Metadata Storage",
             "state": "critical",
             "used": 484278,
             "total": 754434,
             "paths": [
                {
                 "path": "/var/lib/cassandra/data",
                 "partition": "/dev/mapper/centos-root",
                 "used": 201478,
                 "total": 251478
                },
                {
                 "path": "/var/lib/cassandra/data1",
```

```
 "partition": "/dev/mapper/centos-root1",
 "used": 231400,
 "total": 251478
},
{
 "path": "/var/lib/cassandra/data2",
 "partition": "/dev/mapper/centos-root2",
 "used": 51400,
 "total": 251478
}
]
},
```

View One Node

This request returns the node specified by <node_name>.

Request Format

GET https://<FQDN>:8043/v2/cosallnodestatus/<node_name>

Authorization: bearer <auth_token>

Content-Type: application/json

Example

```
GET https://10.20.118.64:8043/v2/cosallnodestatus/335574541
Authorization: bearer <auth_token>
Content-Type: application/json

HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": "smtenant_0.smcosallnodestatus.335574541",
  "name": "335574541",
  "type": "cosallnodestatus",
  "externalId": "/v2/cosallnodestatus/335574541",
  "transactionId": "6db702f2-32b0-4538-95f0-53e3cd23069b",
  "modified": "2017-07-28T04:39:02.655Z",
  "properties": {
    "hostname": "c3260-g3a",
    "management": "20.0.118.13",
    "timestamp": "1501216742548",
    "faultStatus": "warning",
    "faultDetails": [
      "Node Resiliency Status is in a warning state"
    ],
    "myCluster": "goliath-cluster",
    "nodeType": "COS",
    "myResGroupId": "goliath-cluster:1",
    "myResGroupStatus": "normal",
    "slaStatus": {
      "storageStatus": {
        "total": 260792,
        "used": 137027.57,
        "status": "normal",
        "description": "52.54% of storage is used"
      },
"partitionStatus": {
        "status": "normal",
```

```
          "description": []
        },
        "serviceStatus": {
          "status": "normal",
          "description": "All COS Node services are functional"
        },
        "diskStatus": {
          "status": "normal",
          "description": "All disks are up and functional"
        },
        "interfaceStatus": {
          "status": "normal",
          "description": "All data interfaces are up"
        },
        "nodeResiliencyStatus": {
          "status": "warning",
          "description": [
            "GOIDs are in warning state. DEC repair ongoing? no. LEC repair ongoing? no"
          ],
          "LECInfo": {
            "status": "normal",
            "minDrives": 14,
            "available": 28
          },
          "GoidInfo": {
            "status": "warning",
            "DECrepair": "no",
            "LECrepair": "no"
          }
        },
        "clusterResiliencyStatus": {
          "status": "normal",
         "description": "Cluster Resiliency/DEC status is normal. Total nodes in resilience
group: 18. Active nodes: 18"
        }
      },
      "node": {
        "clusterAdminState": "enabled",
        "nodeAdminState": "inservice",
        "nodeStatus": "up",
        "statusDetails": "Node has sent configuration status to AIC",
        "nodeSoftware": "3.14.1-b33"
      },
      "service": {
        "services": [
          {
            "name": "Consul Agent",
            "state": "up",
            "critical": false
          },
          {
            "name": "NTP Daemon",
            "state": "up",
            "critical": false
          },
          {
            "name": "Cisco SNMP Agent",
            "state": "up",
            "critical": false
          },
          {
            "name": "Cisco Cloud Object Store Daemon",
            "state": "up",
            "critical": true
```

```
      },
      {
        "name": "Cisco Cache Server",
        "state": "up",
        "critical": true
      },
      {
        "name": "Sensu Client",
        "state": "up",
        "critical": false
      },
      {
        "name": "Monit",
        "state": "up",
        "critical": false
      }
    ]
  },
  "storage": {
    "partitions": [
      {
        "path": "/arroyo/log",
        "name": "COS Service Logs",
        "state": "normal",
        "total": 144497,
        "used": 19914,
        "partition": "/dev/sda3"
      }
    ],
    "disks": [
      {
        "name": "Cisco Disk 01",
        "state": "up"
      },
      {
        "name": "Cisco Disk 02",
        "state": "up"
      },
      {
        "name": "Cisco Disk 03",
        "state": "up"
      },
      {
        "name": "Cisco Disk 04",
        "state": "up"
      },
      {
        "name": "Cisco Disk 05",
        "state": "up"
      },
      {
        "name": "Cisco Disk 06",
        "state": "up"
      },
      {
        "name": "Cisco Disk 07",
        "state": "up"
      },
      {
        "name": "Cisco Disk 08",
        "state": "up"
      },
      {
        "name": "Cisco Disk 09",
```

```
    "state": "up"
  },
  {
    "name": "Cisco Disk 10",
    "state": "up"
  },
  {
    "name": "Cisco Disk 11",
    "state": "up"
  },
  {
    "name": "Cisco Disk 12",
    "state": "up"
  },
  {
    "name": "Cisco Disk 13",
    "state": "up"
  },
  {
    "name": "Cisco Disk 14",
    "state": "up"
  },
  {
    "name": "Cisco Disk 15",
    "state": "up"
  },
  {
    "name": "Cisco Disk 16",
    "state": "up"
  },
  {
    "name": "Cisco Disk 17",
    "state": "up"
  },
  {
    "name": "Cisco Disk 18",
    "state": "up"
  },
  {
    "name": "Cisco Disk 19",
    "state": "up"
  },
  {
    "name": "Cisco Disk 20",
    "state": "up"
  },
  {
    "name": "Cisco Disk 21",
    "state": "up"
  },
  {
    "name": "Cisco Disk 22",
    "state": "up"
  },
  {
    "name": "Cisco Disk 23",
    "state": "up"
  },
  {
    "name": "Cisco Disk 24",
    "state": "up"
  },
  {
    "name": "Cisco Disk 25",
```

```
            "state": "up"
          },
          {
            "name": "Cisco Disk 26",
            "state": "up"
          },
          {
            "name": "Cisco Disk 27",
            "state": "up"
          },
          {
            "name": "Cisco Disk 28",
            "state": "up"
          }
        ]
      },
      "net": {
        "interfaces": [
          {
            "name": "eth2",
            "state": "up",
            "inet": "20.0.70.1"
          },
          {
            "name": "eth3",
            "state": "up",
            "inet": "20.0.82.1"
          },
          {
            "name": "eth4",
            "state": "up",
            "inet": "20.0.70.2"
          },
          {
            "name": "eth5",
            "state": "up",
            "inet": "20.0.82.2"
          }
        ]
      }
    }
}
```


Listing, Creating, Updating, and Deleting Node Profiles

A *node profile* is a configuration template that can be assigned to a group of COS or CMC nodes that share common attributes.

Node profiles are of two types:

- COS node profiles, uniquely defined by device model, COS cluster, and resiliency group.
- CMC node profiles, uniquely defined by device model and CMC to which is should be applied.

In COS Release 3.16.1, node profile creation is not supported by the SM API, and must be done using the COS (V2PC) GUI. In addition, there is no update operation available for node profiles because once the profile is applied to bringing up a designed group of nodes, it is no longer of any use.

List All Node Profiles

This request returns a list of all node profiles (COS and CMC) in the deployment.

Request Format

GET https://<FQDN>:8043/v2/cosnodeprofiles

Authorization: bearer <auth_token>

Content-Type: application/json

Sample Response

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "id": "smtenant_0.smcosnodeprofiles.cmc-profile",
    "name": "cmc-profile",
    "type": "cosnodeprofiles",
    "externalId": "/v2/cosnodeprofiles/cmc-profile",
    "transactionId": "767000ea-028c-47ee-828c-b5bd279ac97d",
    "modified": "2017-07-07T21:33:54.538Z",
    "properties": {
      "profileType": "Content Metadata Node",
      "model": "MetadataNode",
      "clusterRef": "smtenant_0.smcoscmc.goliath-cmc-cluster",
      "description": "",
      "dataInterfaces": [],
      "consul": {
        "datastores": [
          {
            "folder": "FolderV2PC.Qiong",
            "name": "B2-S6-SSD1.6T-RAID0"
          }
        ],
        "domain": "v2pc.com",
        "datastore_host": "20.0.60.95",
        "vmFolder": "FolderV2PC.Qiong",
        "cluster": "Cluster.Fei",
        "resourcePool": "Resource.Qiong",
        "images": [
          {
            "vendor": "cisco",
            "name": "v2p-base-image",
            "storeName": "B2-S6-SSD1.6T-RAID0",
            "repoIP": "20.0.118.65",
            "imgTag": "cisco-centos-7.0",
            "provider": "pod1",
            "systemRepoList": [
              {
                "vendorId": "cisco",
                "name": "third-party",
                "baseUrl":
"http://20.0.118.65/cisco/system/cisco-centos-7.0/third-party/3.3.0-15518",
                "imgTag": "cisco-centos-7.0",
                "version": "3.3.0-15518",
                "repoId": "third-party"
              },
              {
                "vendorId": "cisco",
```

```
                "name": "v2pc",
                "baseUrl":
"http://20.0.118.65/cisco/system/cisco-centos-7.0/v2pc/3.3.0-15518",
                "imgTag": "cisco-centos-7.0",
                "version": "3.3.0-15518",
                "repoId": "v2pc"
              }
            ],
            "datastore": "B2-S6-SSD1.6T-RAID0",
            "packages": [
              {
                "src": {
                  "local_file": "/root/data/v2p-repo-3.3.0-br_master-15518.iso",
                  "remote_file": "/home/v2pc/repo-3.1.0-9999.iso",
                  "format": "iso"
                },
                "version": "3.2.0",
                "type": "system"
              }
            ],
            "vmName": "v2p-base-image",
            "repoPort": "5001"
          },
          {
            "vendor": "cisco",
            "name": "v2p-coreos-image",
            "storeName": "B2-S6-SSD1.6T-RAID0",
            "repoIP": "20.0.118.65",
            "imgTag": "cisco-coreos-3.0",
            "provider": "pod1",
            "systemRepoList": [],
            "datastore": "B2-S6-SSD1.6T-RAID0",
            "packages": [],
            "vmName": "v2p-coreos-image",
            "repoPort": "5001"
          }
        ],
        "datastore": "B2-S6-SSD1.6T-RAID0",
        "disk": 0,
        "port": 443,
        "unverified": "true",
        "vm_type": "master",
        "numOfMaster": 3,
        "templateName": "v2p-base-image",
        "fqdnName": "20.0.118.64",
        "hostname": "v2p-master-33-3",
        "numCPU": 8,
        "permitRootLogin": "false",
        "dns": [
          "127.0.0.1",
          "171.70.168.183"
        ],
        "memory": 32768,
        "networkInterfaces": [
          {
            "netCreate": true,
            "subnet": "255.255.252.0",
            "ip": "20.0.118.64",
            "hasDefaultGW": true,
            "label": "VLAN 116",
            "net_type": "net_mgmt",
            "gateway": "20.0.116.1",
            "dhcpEnabled": false
          }
```

```
        ],
        "vmName": "v2p-master-33-3",
        "firstMasterHost": "20.0.118.62",
        "consulKey": "c9RWgODz9L/Kq0d1nZFc2Q==",
        "ntp": [
          "10.50.171.9"
        ],
        "provider": "vmware",
        "saltMinion": {
          "saltMinionPrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEogIBAAKCAQEAiJA6m6HPL50MloZcP3746A9VRYyiH8QLns7c6u+7wJTNeazC\nBH5lkTUcSSPTijkrTgQU20
1QqR9HLa+7ah58NiU1Bhd4pLRevJ1czbF9JGwy1RSM\nut09iJ8ZLpsDxYKbBRP7mk9S4KTftWwrxHfy3B1Pu3qhtO
EpnAyhbh7ZExd3n5gg\nqKOVqcoVmCKrlslOt2yd4gZjjoLy/F+u05K7ieQY+zxAHMFp+9u8GsnrakKsrOtz\ncMUk
RkEPY9Xx8FB9dK6krfHsti5/9hErV1vj7+puzsWYPZaJpqWQweOorZubfy4X\nsutZAYL9HjTU8orHx6XX7gHxbuy3
5LPdqQ+o7QIDAQABAoIBAGfF+dNUFUXgOLWV\nqjnR+cEWDb1JDdXHuvDVz0h0Hh9MtAt2/QQ/LF04l6L4p4JMyRyC
u3qeeOHbU9Ji\nzK7qNYdLpMxZP6MhMTBxABmKe0cXNVV4RSymfGD6TT9+KlKHWdd1BOeQybDNePW7\nGlM0twcO+w
zDNmhmBaGAc5djt57qRQoX8a5kjIzkT0/4lvd9YChyZZS3Qq5JSuRX\nxlbDkzWRSdkJK2CnYrAxPi37/93F/Ot1QA
gp3ZFYsub7m/CA4AqGgpOgw4JEdNNr\n+6tOL3YZxVbP2ylamP598EMbUBt6+m+mCr25jGqBKH36+lzQKH/Rq8JSik
uGgfpx\nA0/iSYECgYEAtmBVI/fisjLnrwAN9fJFPO74LFzdXGOIsFdhei18fKC5Q8WCFx5m\nnrmkBsCHHt/HQjTM
w1LEUfHdgUdeqZOxlpQ48dvK/82C/Z+gFp4JTBoldpgXCluC\neba4aBF/SD209E8cd/ARE4dBoQZvnJNdIxoWf3Vn
zU4I8HCiskz6hE0CgYEAv7Fd\nUVI8EyUgR9JjvpI6wg+A4MjpjoS8OsApesxABwx4LI9dzfT6zJ4vmEg43k666uVq
\nWl34u126TJEcUUeoq8uiB4EN7osafw20iJlB12+8OTDvDo2I03RAoDb6BP6a1j2E\n2xFcgmRIM6MAoHEDVJ7eHX
F4g6NLP7MDFKUAhyECgYBl4NZu61Cl/y3RYpRp6dan\nvU7BaXb231tCNu3rFHHQJPCZXUHYrFDjJgztAKv6oUNsRp
aEHj4xEI2OZFILJMg1\nt8M/yGpUD3e3Om5xdAL5kjmPUFK+pHP0jg/hgrS5pE7rky3yChdf81TEIRZEP0Xu\nD+q1
dN5xn7/6oNEiDUJ8mQKBgFJSqGHRMC2Epy1RBDcGc6d9ovpmLi9ff7q3Zdjv\nuJuOVlnEpCNcDw0q7gO3GH0YUJuq
NJdkffpxUR2qv/+EB8Lnq8kPOl10bcecqihz\nsNwFh08EXdXhIJ628T4c/hiBAxv4xYKTHSh+SqeqXA2LG0QPChKY
9YmK6jkL9sm5\n5tUhAoGAZHb+1GVRztZ7UEEOSbGxrmfR1YOoE0a08/KIzJvfHoBw11D+tU+MCSiT\npEAAI2mg1Y
bQOo4p3Nr2Z/qI0RVRW0YGFq1QYLO9GoT+irS7T5s6rWSWbjxd3B79\nW7NjY5/i+GziHoPEe+Q4E5JQ0+10nDnlSc
hQfPed+dudo//u0lo=\n-----END RSA PRIVATE KEY-----",
          "saltMinionPublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAiJA6m6HPL50MloZcP374\n6A9VRYyiH8QLns7c6u+7wJ
TNeazCBH5lkTUcSSPTijkrTgQU201QqR9HLa+7ah58\nNiU1Bhd4pLRevJ1czbF9JGwy1RSMut09iJ8ZLpsDxYKbBR
P7mk9S4KTftWwrxHfy\n3B1Pu3qhtOEpnAyhbh7ZExd3n5ggqKOVqcoVmCKrlslOt2yd4gZjjoLy/F+u05K7\nieQY
+zxAHMFp+9u8GsnrakKsrOtzcMUkRkEPY9Xx8FB9dK6krfHsti5/9hErV1vj\n7+puzsWYPZaJpqWQweOorZubfy4X
sutZAYL9HjTU8orHx6XX7gHxbuy35LPdqQ+o\n7QIDAQAB\n-----END PUBLIC KEY-----"
        },
        "vdiskProvisionMode": "thick",
        "user": "administrator@vsphere.local",
        "privateFqdnName": "20.0.118.64",
        "npmUrl": "http://20.0.118.65:4873/",
        "password": "cisco123",
        "controllerIP": "20.0.50.43",
        "datacenter": "Blade-v2pc",
        "isMaster": true,
        "smConfig": {
          "algo": "md5",
          "key": "50f8347c5aa2e1fb9c7a108759907c2b"
        },
        "imgTag": "cisco-centos-7.0",
        "region": {
          "city": "",
          "masters": [
            "20.0.118.62",
            "20.0.118.63",
            "20.0.118.64"
          ],
          "name": "region-0",
          "country": "",
          "repos": [
            "20.0.118.65"
          ],
          "elks": [
            "20.0.118.66"
          ],
```

```
            "state": "",
            "address": "",
            "mastersHostname": [
              "v2p-master-33-1",
              "v2p-master-33-2",
              "v2p-master-33-3"
            ],
            "type": "primary",
            "description": ""
          },
          "smAccessToken": {
            "access_token": "deaec16a-79c9-4023-b84d-44acad246284",
            "token_type": "Bearer",
            "expires_in": 1000,
            "refresh_token": "68c8560b-a30b-4d11-9892-b15bcd8ce9c8"
          },
          "mosDNS": [
            {
              "ip": "20.0.118.10",
              "domain": "goliath3-ext.com",
              "hostname": "20.0.118.10",
              "algo": "hmac-md5",
              "key": "aCEOXQUHeJ4PksUnvC2yUw=="
            }
          ],
          "ssh": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCkvq2+Rc52Qd7F9YR8pjETiahiJneL8zC+z2cdvn6exFtRDZKX1PuJ2FERAm
x2w0vn83U8LTzMuz7ukeSdLPJGLv7rprF1zpgUfCNeGcHUDRFL0kgn0EPrWFHiZ5yBqxCMwq8Xe44jg1kEAouoxRde
DhOeyGRgRU54fpJY5NaJVUbMl236xejKprCkUtbLBFcAmNdDHCMJKVWpWM6cD9CyvBv75woWvdmQTMd/oNx7Vv86hc
0ihqJagfN7XGdUH+3bm7cTSArXCOh4z74ikuBN5ok8LASWMS/TkIWphJDdhUhqVv1nkNgO43mV1zlF26JkNfq/hyWo
NibuP4iewmh/"
        }
      }
    },
    {
      "id": "smtenant_0.smcosnodeprofiles.c3260-rg-1",
      "name": "c3260-rg-1",
      "type": "cosnodeprofiles",
      "externalId": "/v2/cosnodeprofiles/c3260-rg-1",
      "transactionId": "9709d70a-1505-4e15-ba33-04fddcad278e",
      "modified": "2017-07-07T21:35:52.409Z",
      "properties": {
        "profileType": "COS Node",
        "model": "UCSC-C3K-4U5",
        "clusterRef": "smtenant_0.smcosclusters.goliath-cluster",
        "resiliencyGrpId": 1,
        "description": "",
        "dataInterfaces": [
          {
            "name": "eth2",
            "ipPoolRef": "smtenant_0.smcosippool.pool_A"
          },
          {
            "name": "eth3",
            "ipPoolRef": "smtenant_0.smcosippool.pool_B"
          },
          {
            "name": "eth4",
            "ipPoolRef": "smtenant_0.smcosippool.pool_A"
          },
          {
            "name": "eth5",
            "ipPoolRef": "smtenant_0.smcosippool.pool_B"
          }
```

```
        ],
        "consul": {
          "datastores": [
            {
              "folder": "FolderV2PC.Qiong",
              "name": "B2-S6-SSD1.6T-RAID0"
            }
          ],
          "domain": "v2pc.com",
          "datastore_host": "20.0.60.95",
          "vmFolder": "FolderV2PC.Qiong",
          "cluster": "Cluster.Fei",
          "resourcePool": "Resource.Qiong",
          "images": [
            {
              "vendor": "cisco",
              "name": "v2p-base-image",
              "storeName": "B2-S6-SSD1.6T-RAID0",
              "repoIP": "20.0.118.65",
              "imgTag": "cisco-centos-7.0",
              "provider": "pod1",
              "systemRepoList": [
                {
                  "vendorId": "cisco",
                  "name": "third-party",
                  "baseUrl":
"http://20.0.118.65/cisco/system/cisco-centos-7.0/third-party/3.3.0-15518",
                  "imgTag": "cisco-centos-7.0",
                  "version": "3.3.0-15518",
                  "repoId": "third-party"
                },
                {
                  "vendorId": "cisco",
                  "name": "v2pc",
                  "baseUrl":
"http://20.0.118.65/cisco/system/cisco-centos-7.0/v2pc/3.3.0-15518",
                  "imgTag": "cisco-centos-7.0",
                  "version": "3.3.0-15518",
                  "repoId": "v2pc"
                }
              ],
              "datastore": "B2-S6-SSD1.6T-RAID0",
              "packages": [
                {
                  "src": {
                    "local_file": "/root/data/v2p-repo-3.3.0-br_master-15518.iso",
                    "remote_file": "/home/v2pc/repo-3.1.0-9999.iso",
                    "format": "iso"
                  },
                  "version": "3.2.0",
                  "type": "system"
                }
              ],
              "vmName": "v2p-base-image",
              "repoPort": "5001"
            },
            {
              "vendor": "cisco",
              "name": "v2p-coreos-image",
              "storeName": "B2-S6-SSD1.6T-RAID0",
              "repoIP": "20.0.118.65",
              "imgTag": "cisco-coreos-3.0",
              "provider": "pod1",
              "systemRepoList": [],
```

```
            "datastore": "B2-S6-SSD1.6T-RAID0",
            "packages": [],
            "vmName": "v2p-coreos-image",
            "repoPort": "5001"
          }
        ],
        "datastore": "B2-S6-SSD1.6T-RAID0",
        "disk": 0,
        "port": 443,
        "unverified": "true",
        "vm_type": "master",
        "numOfMaster": 3,
        "templateName": "v2p-base-image",
        "fqdnName": "20.0.118.64",
        "hostname": "v2p-master-33-3",
        "numCPU": 8,
        "permitRootLogin": "false",
        "dns": [
          "127.0.0.1",
          "171.70.168.183"
        ],
        "memory": 32768,
        "networkInterfaces": [
          {
            "netCreate": true,
            "subnet": "255.255.252.0",
            "ip": "20.0.118.64",
            "hasDefaultGW": true,
            "label": "VLAN 116",
            "net_type": "net_mgmt",
            "gateway": "20.0.116.1",
            "dhcpEnabled": false
          }
        ],
        "vmName": "v2p-master-33-3",
        "firstMasterHost": "20.0.118.62",
        "consulKey": "c9RWgODz9L/Kq0d1nZFc2Q==",
        "ntp": [
          "10.50.171.9"
        ],
        "provider": "vmware",
        "saltMinion": {
          "saltMinionPrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEogIBAAKCAQEAiJA6m6HPL50MloZcP3746A9VRYyiH8QLns7c6u+7wJTNeazC\nBH5lkTUcSSPTijkrTgQU20
1QqR9HLa+7ah58NiU1Bhd4pLRevJ1czbF9JGwy1RSM\nut09iJ8ZLpsDxYKbBRP7mk9S4KTftWwrxHfy3B1Pu3qhtO
EpnAyhbh7ZExd3n5gg\nqKOVqcoVmCKrlslOt2yd4gZjjoLy/F+u05K7ieQY+zxAHMFp+9u8GsnrakKsrOtz\ncMUk
RkEPY9Xx8FB9dK6krfHsti5/9hErV1vj7+puzsWYPZaJpqWQweOorZubfy4X\nsutZAYL9HjTU8orHx6XX7gHxbuy3
5LPdqQ+o7QIDAQABAoIBAGfF+dNUFUXgOLWV\nqjnR+cEWDb1JDdXHuvDVz0h0Hh9MtAt2/QQ/LF04l6L4p4JMyRyC
u3qeeOHbU9Ji\nzK7qNYdLpMxZP6MhMTBxABmKe0cXNVV4RSymfGD6TT9+KlKHWdd1BOeQybDNePW7\nGlM0twcO+w
zDNmhmBaGAc5djt57qRQoX8a5kjIzkT0/4lvd9YChyZZS3Qq5JSuRX\nxlbDkzWRSdkJK2CnYrAxPi37/93F/Ot1QA
gp3ZFYsub7m/CA4AqGgpOgw4JEdNNr\n+6tOL3YZxVbP2ylamP598EMbUBt6+m+mCr25jGqBKH36+lzQKH/Rq8JSik
uGgfpx\nA0/iSYECgYEAtmBVI/fisjLnrwAN9fJFPO74LFzdXGOIsFdhei18fKC5Q8WCFx5m\nnrmkBsCHHt/HQjTM
w1LEUfHdgUdeqZOxlpQ48dvK/82C/Z+gFp4JTBoldpgXCluC\neba4aBF/SD209E8cd/ARE4dBoQZvnJNdIxoWf3Vn
zU4I8HCiskz6hE0CgYEAv7Fd\nUVI8EyUgR9JjvpI6wg+A4MjpjoS8OsApesxABwx4LI9dzfT6zJ4vmEg43k666uVq
\nWl34u126TJEcUUeoq8uiB4EN7osafw20iJlB12+8OTDvDo2I03RAoDb6BP6a1j2E\n2xFcgmRIM6MAoHEDVJ7eHX
F4g6NLP7MDFKUAhyECgYBl4NZu61Cl/y3RYpRp6dan\nvU7BaXb231tCNu3rFHHQJPCZXUHYrFDjJgztAKv6oUNsRp
aEHj4xEI2OZFILJMg1\nt8M/yGpUD3e3Om5xdAL5kjmPUFK+pHP0jg/hgrS5pE7rky3yChdf81TEIRZEP0Xu\nD+q1
dN5xn7/6oNEiDUJ8mQKBgFJSqGHRMC2Epy1RBDcGc6d9ovpmLi9ff7q3Zdjv\nuJuOVlnEpCNcDw0q7gO3GH0YUJuq
NJdkffpxUR2qv/+EB8Lnq8kPOl10bcecqihz\nsNwFh08EXdXhIJ628T4c/hiBAxv4xYKTHSh+SqeqXA2LG0QPChKY
9YmK6jkL9sm5\n5tUhAoGAZHb+1GVRztZ7UEEOSbGxrmfR1YOoE0a08/KIzJvfHoBw11D+tU+MCSiT\npEAAI2mg1Y
bQOo4p3Nr2Z/qI0RVRW0YGFq1QYLO9GoT+irS7T5s6rWSWbjxd3B79\nW7NjY5/i+GziHoPEe+Q4E5JQ0+10nDnlSc
hQfPed+dudo//u0lo=\n-----END RSA PRIVATE KEY-----",
          "saltMinionPublicKey": "-----BEGIN PUBLIC KEY-----
```

```
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAiJA6m6HPL50MloZcP374\n6A9VRYyiH8QLns7c6u+7wJ
TNeazCBH5lkTUcSSPTijkrTgQU201QqR9HLa+7ah58\nNiU1Bhd4pLRevJ1czbF9JGwy1RSMut09iJ8ZLpsDxYKbBR
P7mk9S4KTftWwrxHfy\n3B1Pu3qhtOEpnAyhbh7ZExd3n5ggqKOVqcoVmCKrlslOt2yd4gZjjoLy/F+u05K7\nieQY
+zxAHMFp+9u8GsnrakKsrOtzcMUkRkEPY9Xx8FB9dK6krfHsti5/9hErV1vj\n7+puzsWYPZaJpqWQweOorZubfy4X
sutZAYL9HjTU8orHx6XX7gHxbuy35LPdqQ+o\n7QIDAQAB\n-----END PUBLIC KEY-----"
        },
        "vdiskProvisionMode": "thick",
        "user": "administrator@vsphere.local",
        "privateFqdnName": "20.0.118.64",
        "npmUrl": "http://20.0.118.65:4873/",
        "password": "cisco123",
        "controllerIP": "20.0.50.43",
        "datacenter": "Blade-v2pc",
        "isMaster": true,
        "smConfig": {
          "algo": "md5",
          "key": "50f8347c5aa2e1fb9c7a108759907c2b"
        },
        "imgTag": "cisco-centos-7.0",
        "region": {
          "city": "",
          "masters": [
            "20.0.118.62",
            "20.0.118.63",
            "20.0.118.64"
          ],
          "name": "region-0",
          "country": "",
          "repos": [
            "20.0.118.65"
          ],
          "elks": [
            "20.0.118.66"
          ],
          "state": "",
          "address": "",
          "mastersHostname": [
            "v2p-master-33-1",
            "v2p-master-33-2",
            "v2p-master-33-3"
          ],
          "type": "primary",
          "description": ""
        },
        "smAccessToken": {
          "access_token": "deaec16a-79c9-4023-b84d-44acad246284",
          "token_type": "Bearer",
          "expires_in": 1000,
          "refresh_token": "68c8560b-a30b-4d11-9892-b15bcd8ce9c8"
        },
        "mosDNS": [
          {
            "ip": "20.0.118.10",
            "domain": "goliath3-ext.com",
            "hostname": "20.0.118.10",
            "algo": "hmac-md5",
            "key": "aCEOXQUHeJ4PksUnvC2yUw=="
          }
        ],
        "ssh": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCkvq2+Rc52Qd7F9YR8pjETiahiJneL8zC+z2cdvn6exFtRDZKX1PuJ2FERAm
x2w0vn83U8LTzMuz7ukeSdLPJGLv7rprF1zpgUfCNeGcHUDRFL0kgn0EPrWFHiZ5yBqxCMwq8Xe44jg1kEAouoxRde
```

```
DhOeyGRgRU54fpJY5NaJVUbMl236xejKprCkUtbLBFcAmNdDHCMJKVWpWM6cD9CyvBv75woWvdmQTMd/oNx7Vv86hc
0ihqJagfN7XGdUH+3bm7cTSArXCOh4z74ikuBN5ok8LASWMS/TkIWphJDdhUhqVv1nkNgO43mV1zlF26JkNfq/hyWo
NibuP4iewmh/"
      },
      "rioFanoutCompaction": "Enabled"
    }
  },
  {
    "id": "smtenant_0.smcosnodeprofiles.c3260-rg-2",
    "name": "c3260-rg-2",
    "type": "cosnodeprofiles",
    "externalId": "/v2/cosnodeprofiles/c3260-rg-2",
    "transactionId": "a38e6a67-8fa1-4e80-a83b-c3ab64119174",
    "modified": "2017-07-07T21:36:28.686Z",
    "properties": {
      "profileType": "COS Node",
      "model": "UCSC-C3K-4U5",
      "clusterRef": "smtenant_0.smcosclusters.goliath-cluster",
      "resiliencyGrpId": 2,
      "description": "",
      "dataInterfaces": [
        {
          "name": "eth2",
          "ipPoolRef": "smtenant_0.smcosippool.pool_A"
        },
        {
          "name": "eth3",
          "ipPoolRef": "smtenant_0.smcosippool.pool_B"
        },
        {
          "name": "eth4",
          "ipPoolRef": "smtenant_0.smcosippool.pool_A"
        },
        {
          "name": "eth5",
          "ipPoolRef": "smtenant_0.smcosippool.pool_B"
        }
      ],
      "consul": {
        "datastores": [
          {
            "folder": "FolderV2PC.Qiong",
            "name": "B2-S6-SSD1.6T-RAID0"
          }
        ],
        "domain": "v2pc.com",
        "datastore_host": "20.0.60.95",
        "vmFolder": "FolderV2PC.Qiong",
        "cluster": "Cluster.Fei",
        "resourcePool": "Resource.Qiong",
        "images": [
          {
            "vendor": "cisco",
            "name": "v2p-base-image",
            "storeName": "B2-S6-SSD1.6T-RAID0",
            "repoIP": "20.0.118.65",
            "imgTag": "cisco-centos-7.0",
            "provider": "pod1",
            "systemRepoList": [
              {
                "vendorId": "cisco",
                "name": "third-party",
                "baseUrl":
"http://20.0.118.65/cisco/system/cisco-centos-7.0/third-party/3.3.0-15518",
```

```
                "imgTag": "cisco-centos-7.0",
                "version": "3.3.0-15518",
                "repoId": "third-party"
              },
              {
                "vendorId": "cisco",
                "name": "v2pc",
                "baseUrl":
"http://20.0.118.65/cisco/system/cisco-centos-7.0/v2pc/3.3.0-15518",
                "imgTag": "cisco-centos-7.0",
                "version": "3.3.0-15518",
                "repoId": "v2pc"
              }
            ],
            "datastore": "B2-S6-SSD1.6T-RAID0",
            "packages": [
              {
                "src": {
                  "local_file": "/root/data/v2p-repo-3.3.0-br_master-15518.iso",
                  "remote_file": "/home/v2pc/repo-3.1.0-9999.iso",
                  "format": "iso"
                },
                "version": "3.2.0",
                "type": "system"
              }
            ],
            "vmName": "v2p-base-image",
            "repoPort": "5001"
          },
          {
            "vendor": "cisco",
            "name": "v2p-coreos-image",
            "storeName": "B2-S6-SSD1.6T-RAID0",
            "repoIP": "20.0.118.65",
            "imgTag": "cisco-coreos-3.0",
            "provider": "pod1",
            "systemRepoList": [],
            "datastore": "B2-S6-SSD1.6T-RAID0",
            "packages": [],
            "vmName": "v2p-coreos-image",
            "repoPort": "5001"
          }
        ],
        "datastore": "B2-S6-SSD1.6T-RAID0",
        "disk": 0,
        "port": 443,
        "unverified": "true",
        "vm_type": "master",
        "numOfMaster": 3,
        "templateName": "v2p-base-image",
        "fqdnName": "20.0.118.64",
        "hostname": "v2p-master-33-3",
        "numCPU": 8,
        "permitRootLogin": "false",
        "dns": [
          "127.0.0.1",
          "171.70.168.183"
        ],
        "memory": 32768,
        "networkInterfaces": [
          {
            "netCreate": true,
            "subnet": "255.255.252.0",
            "ip": "20.0.118.64",
```

```
            "hasDefaultGW": true,
            "label": "VLAN 116",
            "net_type": "net_mgmt",
            "gateway": "20.0.116.1",
            "dhcpEnabled": false
          }
        ],
        "vmName": "v2p-master-33-3",
        "firstMasterHost": "20.0.118.62",
        "consulKey": "c9RWgODz9L/Kq0d1nZFc2Q==",
        "ntp": [
          "10.50.171.9"
        ],
        "provider": "vmware",
        "saltMinion": {
          "saltMinionPrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEogIBAAKCAQEAiJA6m6HPL50MloZcP3746A9VRYyiH8QLns7c6u+7wJTNeazC\nBH5lkTUcSSPTijkrTgQU20
1QqR9HLa+7ah58NiU1Bhd4pLRevJ1czbF9JGwy1RSM\nut09iJ8ZLpsDxYKbBRP7mk9S4KTftWwrxHfy3B1Pu3qhtO
EpnAyhbh7ZExd3n5gg\nqKOVqcoVmCKrlslOt2yd4gZjjoLy/F+u05K7ieQY+zxAHMFp+9u8GsnrakKsrOtz\ncMUk
RkEPY9Xx8FB9dK6krfHsti5/9hErV1vj7+puzsWYPZaJpqWQweOorZubfy4X\nsutZAYL9HjTU8orHx6XX7gHxbuy3
5LPdqQ+o7QIDAQABAoIBAGfF+dNUFUXgOLWV\nqjnR+cEWDb1JDdXHuvDVz0h0Hh9MtAt2/QQ/LF04l6L4p4JMyRyC
u3qeeOHbU9Ji\nzK7qNYdLpMxZP6MhMTBxABmKe0cXNVV4RSymfGD6TT9+KlKHWdd1BOeQybDNePW7\nGlM0twcO+w
zDNmhmBaGAc5djt57qRQoX8a5kjIzkT0/4lvd9YChyZZS3Qq5JSuRX\nxlbDkzWRSdkJK2CnYrAxPi37/93F/Ot1QA
gp3ZFYsub7m/CA4AqGgpOgw4JEdNNr\n+6tOL3YZxVbP2ylamP598EMbUBt6+m+mCr25jGqBKH36+lzQKH/Rq8JSik
uGgfpx\nA0/iSYECgYEAtmBVI/fisjLnrwAN9fJFPO74LFzdXGOIsFdhei18fKC5Q8WCFx5m\nnrmkBsCHHt/HQjTM
w1LEUfHdgUdeqZOxlpQ48dvK/82C/Z+gFp4JTBoldpgXCluC\neba4aBF/SD209E8cd/ARE4dBoQZvnJNdIxoWf3Vn
zU4I8HCiskz6hE0CgYEAv7Fd\nUVI8EyUgR9JjvpI6wg+A4MjpjoS8OsApesxABwx4LI9dzfT6zJ4vmEg43k666uVq
\nWl34u126TJEcUUeoq8uiB4EN7osafw20iJlB12+8OTDvDo2I03RAoDb6BP6a1j2E\n2xFcgmRIM6MAoHEDVJ7eHX
F4g6NLP7MDFKUAhyECgYBl4NZu61Cl/y3RYpRp6dan\nvU7BaXb231tCNu3rFHHQJPCZXUHYrFDjJgztAKv6oUNsRp
aEHj4xEI2OZFILJMg1\nt8M/yGpUD3e3Om5xdAL5kjmPUFK+pHP0jg/hgrS5pE7rky3yChdf81TEIRZEP0Xu\nD+q1
dN5xn7/6oNEiDUJ8mQKBgFJSqGHRMC2Epy1RBDcGc6d9ovpmLi9ff7q3Zdjv\nuJuOVlnEpCNcDw0q7gO3GH0YUJuq
NJdkffpxUR2qv/+EB8Lnq8kPOl10bcecqihz\nsNwFh08EXdXhIJ628T4c/hiBAxv4xYKTHSh+SqeqXA2LG0QPChKY
9YmK6jkL9sm5\n5tUhAoGAZHb+1GVRztZ7UEEOSbGxrmfR1YOoE0a08/KIzJvfHoBw11D+tU+MCSiT\npEAAI2mg1Y
bQOo4p3Nr2Z/qI0RVRW0YGFq1QYLO9GoT+irS7T5s6rWSWbjxd3B79\nW7NjY5/i+GziHoPEe+Q4E5JQ0+10nDnlSc
hQfPed+dudo//u0lo=\n-----END RSA PRIVATE KEY-----", "saltMinionPublicKey": "-----BEGIN
PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAiJA6m6HPL50MloZcP374\n6A9VRYyiH8QLns7c6u+7wJ
TNeazCBH5lkTUcSSPTijkrTgQU201QqR9HLa+7ah58\nNiU1Bhd4pLRevJ1czbF9JGwy1RSMut09iJ8ZLpsDxYKbBR
P7mk9S4KTftWwrxHfy\n3B1Pu3qhtOEpnAyhbh7ZExd3n5ggqKOVqcoVmCKrlslOt2yd4gZjjoLy/F+u05K7\nieQY
+zxAHMFp+9u8GsnrakKsrOtzcMUkRkEPY9Xx8FB9dK6krfHsti5/9hErV1vj\n7+puzsWYPZaJpqWQweOorZubfy4X
sutZAYL9HjTU8orHx6XX7gHxbuy35LPdqQ+o\n7QIDAQAB\n-----END PUBLIC KEY-----"
        },
        "vdiskProvisionMode": "thick",
        "user": "administrator@vsphere.local",
        "privateFqdnName": "20.0.118.64",
        "npmUrl": "http://20.0.118.65:4873/",
        "password": "cisco123",
        "controllerIP": "20.0.50.43",
        "datacenter": "Blade-v2pc",
        "isMaster": true,
        "smConfig": {
          "algo": "md5",
          "key": "50f8347c5aa2e1fb9c7a108759907c2b"
        },
        "imgTag": "cisco-centos-7.0",
        "region": {
          "city": "",
          "masters": [
            "20.0.118.62",
            "20.0.118.63",
            "20.0.118.64"
          ],
          "name": "region-0",
          "country": "",
```

```
            "repos": [
              "20.0.118.65"
            ],
            "elks": [
              "20.0.118.66"
            ],
            "state": "",
            "address": "",
            "mastersHostname": [
              "v2p-master-33-1",
              "v2p-master-33-2",
              "v2p-master-33-3"
            ],
            "type": "primary",
            "description": ""
          },
          "smAccessToken": {
            "access_token": "deaec16a-79c9-4023-b84d-44acad246284",
            "token_type": "Bearer",
            "expires_in": 1000,
            "refresh_token": "68c8560b-a30b-4d11-9892-b15bcd8ce9c8"
          },
          "mosDNS": [
            {
              "ip": "20.0.118.10",
              "domain": "goliath3-ext.com",
              "hostname": "20.0.118.10",
              "algo": "hmac-md5",
              "key": "aCEOXQUHeJ4PksUnvC2yUw=="
            }
          ],
          "ssh": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCkvq2+Rc52Qd7F9YR8pjETiahiJneL8zC+z2cdvn6exFtRDZKX1PuJ2FERAm
x2w0vn83U8LTzMuz7ukeSdLPJGLv7rprF1zpgUfCNeGcHUDRFL0kgn0EPrWFHiZ5yBqxCMwq8Xe44jg1kEAouoxRde
DhOeyGRgRU54fpJY5NaJVUbMl236xejKprCkUtbLBFcAmNdDHCMJKVWpWM6cD9CyvBv75woWvdmQTMd/oNx7Vv86hc
0ihqJagfN7XGdUH+3bm7cTSArXCOh4z74ikuBN5ok8LASWMS/TkIWphJDdhUhqVv1nkNgO43mV1zlF26JkNfq/hyWo
NibuP4iewmh/"
      },
      "rioFanoutCompaction": "Enabled"
    }
  }
]
```


List One Node Profile

This request returns a single node profile specified in <profile_name>.

Request Format

GET https://<FQDN>:8043/v2/cosnodeprofiles/<profile_name>

Authorization: bearer <auth_token>

Content-Type: application/json

Example

```
GET https://10.20.118.64:8043/v2/cosnodeprofiles/cmc-profile
Authorization: bearer <auth_token>
Content-Type: application/json

HTTP/1.1 200 OK
```

```
Content-Type: application/json

{
  "id": "smtenant_0.smcosnodeprofiles.cmc-profile",
  "name": "cmc-profile",
  "type": "cosnodeprofiles",
  "externalId": "/v2/cosnodeprofiles/cmc-profile",
  "transactionId": "767000ea-028c-47ee-828c-b5bd279ac97d",
  "modified": "2017-07-07T21:33:54.538Z",
  "properties": {
    "profileType": "Content Metadata Node",
    "model": "MetadataNode",
    "clusterRef": "smtenant_0.smcoscmc.goliath-cmc-cluster",
    "description": "",
    "dataInterfaces": [],
    "consul": {
      "datastores": [
        {
          "folder": "FolderV2PC.Qiong",
          "name": "B2-S6-SSD1.6T-RAID0"
        }
      ],
      "domain": "v2pc.com",
      "datastore_host": "20.0.60.95",
      "vmFolder": "FolderV2PC.Qiong",
      "cluster": "Cluster.Fei",
      "resourcePool": "Resource.Qiong",
      "images": [
        {
          "vendor": "cisco",
          "name": "v2p-base-image",
          "storeName": "B2-S6-SSD1.6T-RAID0",
          "repoIP": "20.0.118.65",
          "imgTag": "cisco-centos-7.0",
          "provider": "pod1",
          "systemRepoList": [
            {
              "vendorId": "cisco",
              "name": "third-party",
              "baseUrl":
"http://20.0.118.65/cisco/system/cisco-centos-7.0/third-party/3.3.0-15518",
              "imgTag": "cisco-centos-7.0",
              "version": "3.3.0-15518",
              "repoId": "third-party"
            },
            {
              "vendorId": "cisco",
              "name": "v2pc",
              "baseUrl":
"http://20.0.118.65/cisco/system/cisco-centos-7.0/v2pc/3.3.0-15518",
              "imgTag": "cisco-centos-7.0",
              "version": "3.3.0-15518",
              "repoId": "v2pc"
            }
          ],
          "datastore": "B2-S6-SSD1.6T-RAID0",
          "packages": [
            {
              "src": {
                "local_file": "/root/data/v2p-repo-3.3.0-br_master-15518.iso",
                "remote_file": "/home/v2pc/repo-3.1.0-9999.iso",
                "format": "iso"
              },
              "version": "3.2.0",
```

```
            "type": "system"
          }
        ],
        "vmName": "v2p-base-image",
        "repoPort": "5001"
      },
      {
        "vendor": "cisco",
        "name": "v2p-coreos-image",
        "storeName": "B2-S6-SSD1.6T-RAID0",
        "repoIP": "20.0.118.65",
        "imgTag": "cisco-coreos-3.0",
        "provider": "pod1",
        "systemRepoList": [],
        "datastore": "B2-S6-SSD1.6T-RAID0",
        "packages": [],
        "vmName": "v2p-coreos-image",
        "repoPort": "5001"
      }
    ],
    "datastore": "B2-S6-SSD1.6T-RAID0",
    "disk": 0,
    "port": 443,
    "unverified": "true",
    "vm_type": "master",
    "numOfMaster": 3,
    "templateName": "v2p-base-image",
    "fqdnName": "20.0.118.64",
    "hostname": "v2p-master-33-3",
    "numCPU": 8,
    "permitRootLogin": "false",
    "dns": [
      "127.0.0.1",
      "171.70.168.183"
    ],
    "memory": 32768,
    "networkInterfaces": [
      {
        "netCreate": true,
        "subnet": "255.255.252.0",
        "ip": "20.0.118.64",
        "hasDefaultGW": true,
        "label": "VLAN 116",
        "net_type": "net_mgmt",
        "gateway": "20.0.116.1",
        "dhcpEnabled": false
      }
    ],
    "vmName": "v2p-master-33-3",
    "firstMasterHost": "20.0.118.62",
    "consulKey": "c9RWgODz9L/Kq0d1nZFc2Q==",
    "ntp": [
      "10.50.171.9"
    ],
    "provider": "vmware",
    "saltMinion": {
      "saltMinionPrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEogIBAAKCAQEAiJA6m6HPL50MloZcP3746A9VRYyiH8QLns7c6u+7wJTNeazC\nBH5lkTUcSSPTijkrTgQU20
1QqR9HLa+7ah58NiU1Bhd4pLRevJ1czbF9JGwy1RSM\nut09iJ8ZLpsDxYKbBRP7mk9S4KTftWwrxHfy3B1Pu3qhtO
EpnAyhbh7ZExd3n5gg\nqKOVqcoVmCKrlslOt2yd4gZjjoLy/F+u05K7ieQY+zxAHMFp+9u8GsnrakKsrOtz\ncMUk
RkEPY9Xx8FB9dK6krfHsti5/9hErV1vj7+puzsWYPZaJpqWQweOorZubfy4X\nsutZAYL9HjTU8orHx6XX7gHxbuy3
5LPdqQ+o7QIDAQABAoIBAGfF+dNUFUXgOLWV\nqjnR+cEWDb1JDdXHuvDVz0h0Hh9MtAt2/QQ/LF04l6L4p4JMyRyC
u3qeeOHbU9Ji\nzK7qNYdLpMxZP6MhMTBxABmKe0cXNVV4RSymfGD6TT9+KlKHWdd1BOeQybDNePW7\nGlM0twcO+w
zDNmhmBaGAc5djt57qRQoX8a5kjIzkT0/4lvd9YChyZZS3Qq5JSuRX\nxlbDkzWRSdkJK2CnYrAxPi37/93F/Ot1QA
```

```
gp3ZFYsub7m/CA4AqGgpOgw4JEdNNr\n+6tOL3YZxVbP2ylamP598EMbUBt6+m+mCr25jGqBKH36+lzQKH/Rq8JSik
uGgfpx\nA0/iSYECgYEAtmBVI/fisjLnrwAN9fJFPO74LFzdXGOIsFdhei18fKC5Q8WCFx5m\nnrmkBsCHHt/HQjTM
w1LEUfHdgUdeqZOxlpQ48dvK/82C/Z+gFp4JTBoldpgXCluC\neba4aBF/SD209E8cd/ARE4dBoQZvnJNdIxoWf3Vn
zU4I8HCiskz6hE0CgYEAv7Fd\nUVI8EyUgR9JjvpI6wg+A4MjpjoS8OsApesxABwx4LI9dzfT6zJ4vmEg43k666uVq
\nWl34u126TJEcUUeoq8uiB4EN7osafw20iJlB12+8OTDvDo2I03RAoDb6BP6a1j2E\n2xFcgmRIM6MAoHEDVJ7eHX
F4g6NLP7MDFKUAhyECgYBl4NZu61Cl/y3RYpRp6dan\nvU7BaXb231tCNu3rFHHQJPCZXUHYrFDjJgztAKv6oUNsRp
aEHj4xEI2OZFILJMg1\nt8M/yGpUD3e3Om5xdAL5kjmPUFK+pHP0jg/hgrS5pE7rky3yChdf81TEIRZEP0Xu\nD+q1
dN5xn7/6oNEiDUJ8mQKBgFJSqGHRMC2Epy1RBDcGc6d9ovpmLi9ff7q3Zdjv\nuJuOVlnEpCNcDw0q7gO3GH0YUJuq
NJdkffpxUR2qv/+EB8Lnq8kPOl10bcecqihz\nsNwFh08EXdXhIJ628T4c/hiBAxv4xYKTHSh+SqeqXA2LG0QPChKY
9YmK6jkL9sm5\n5tUhAoGAZHb+1GVRztZ7UEEOSbGxrmfR1YOoE0a08/KIzJvfHoBw11D+tU+MCSiT\npEAAI2mg1Y
bQOo4p3Nr2Z/qI0RVRW0YGFq1QYLO9GoT+irS7T5s6rWSWbjxd3B79\nW7NjY5/i+GziHoPEe+Q4E5JQ0+10nDnlSc
hQfPed+dudo//u0lo=\n-----END RSA PRIVATE KEY-----",
        "saltMinionPublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAiJA6m6HPL50MloZcP374\n6A9VRYyiH8QLns7c6u+7wJ
TNeazCBH5lkTUcSSPTijkrTgQU201QqR9HLa+7ah58\nNiU1Bhd4pLRevJ1czbF9JGwy1RSMut09iJ8ZLpsDxYKbBR
P7mk9S4KTftWwrxHfy\n3B1Pu3qhtOEpnAyhbh7ZExd3n5ggqKOVqcoVmCKrlslOt2yd4gZjjoLy/F+u05K7\nieQY
+zxAHMFp+9u8GsnrakKsrOtzcMUkRkEPY9Xx8FB9dK6krfHsti5/9hErV1vj\n7+puzsWYPZaJpqWQweOorZubfy4X
sutZAYL9HjTU8orHx6XX7gHxbuy35LPdqQ+o\n7QIDAQAB\n-----END PUBLIC KEY-----"
      },
      "vdiskProvisionMode": "thick",
      "user": "administrator@vsphere.local",
      "privateFqdnName": "20.0.118.64",
      "npmUrl": "http://20.0.118.65:4873/",
      "password": "cisco123",
      "controllerIP": "20.0.50.43",
      "datacenter": "Blade-v2pc",
      "isMaster": true,
      "smConfig": {
        "algo": "md5",
        "key": "50f8347c5aa2e1fb9c7a108759907c2b"
      },
      "imgTag": "cisco-centos-7.0",
      "region": {
        "city": "",
        "masters": [
          "20.0.118.62",
          "20.0.118.63",
          "20.0.118.64"
        ],
        "name": "region-0",
        "country": "",
        "repos": [
          "20.0.118.65"
        ],
        "elks": [
          "20.0.118.66"
        ],
        "state": "",
        "address": "",
        "mastersHostname": [
          "v2p-master-33-1",
          "v2p-master-33-2",
          "v2p-master-33-3"
        ],
        "type": "primary",
        "description": ""
      },
      "smAccessToken": {
        "access_token": "deaec16a-79c9-4023-b84d-44acad246284",
        "token_type": "Bearer",
        "expires_in": 1000,
        "refresh_token": "68c8560b-a30b-4d11-9892-b15bcd8ce9c8"
      },
      "mosDNS": [
```

```
        {
          "ip": "20.0.118.10",
          "domain": "goliath3-ext.com",
          "hostname": "20.0.118.10",
          "algo": "hmac-md5",
          "key": "aCEOXQUHeJ4PksUnvC2yUw=="
        }
      ],
      "ssh": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCkvq2+Rc52Qd7F9YR8pjETiahiJneL8zC+z2cdvn6exFtRDZKX1PuJ2FERAm
x2w0vn83U8LTzMuz7ukeSdLPJGLv7rprF1zpgUfCNeGcHUDRFL0kgn0EPrWFHiZ5yBqxCMwq8Xe44jg1kEAouoxRde
DhOeyGRgRU54fpJY5NaJVUbM1236xejKprCkUtbLBFcAmNdDHCMJKVWpWM6cD9CyvBv75woWvdmQTMd/oNx7Vv86hc
0ihqJagfN7XGdUH+3bm7cTSArXCOh4z74ikuBN5ok8LASWMS/TkIWphJDdhUhqVv1nkNgO43mV1zlF26JkNfq/hyWo
NibuP4iewmh/"
    }
  }
}
```


Create a New Node Profile

For COS Release 3.16.1, new node profiles can only be created using the COS (V2PC) GUI; creation of new node profiles using the API is not supported. When a profile is created using the COS GUI, the GUI provides a public URL through which that profile can be downloaded and applied to the nodes during node initialization. The URL is integral to the function of the node profile, so the profile must be created using the GUI.

Update a Node Profile

The purpose of a node profile is to create a template for configuration that can be applied to a group of nodes sharing common attributes. For this reason, updating an existing node profile is neither required nor officially supported in COS Release 3.16.1.

Delete a Node Profile

This request deletes an existing node profile named <profile_name>.

Request Format

DELETE https://<FQDN>:8043/v2/cosnodeprofiles/<profile_name>

Authorization: bearer <auth_token>

Content-Type: application/json

Example

```
DELETE https://10.20.118.64:8043/v2/cosnodeprofiles/cmc-profile
Authorization: bearer <auth_token>
Content-Type: application/json
```

Response Format

HTTP/1.1 200 OK

Alarms and Events API

For the Alarms and Events API, please refer to the API documentation for V2PC Release 3.3. Alarms and Events APIs are also documented online on the V2PC Master node FQDN at:

https://<FQDN>:8443/docs/#/Service_Manager_-_Alarm_&_Event

Swauth API

This chapter describes the subset of the OpenStack Swauth API that is implemented for the COS authentication service. The Swauth API uses the Authentication FQDN that is assigned to the COS cluster.

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.
- The Swauth service uses a special built-in user and key to configure the service itself. This user is known as the **COS super-admin**. The default COS super-admin user name is **.super_admin** and the default super-admin key is **rootroot**. You can use the COS Configuration API to change the user name and password as described in COS Configuration API, page 6-1.

Listing Accounts

To retrieve a list of existing accounts, use the following HTTP GET request:

GET /auth/v2/ HTTP/1.1

Required Headers

- X-Auth-Admin-User
- X-Auth-Admin-Key

Note

For the request to be permitted, the requester identified by X-Auth-Admin-User must have either super admin or reseller admin privileges.

Response Status Codes

- 200 – Success.
- 401 – Unauthorized; X-Auth-Admin-User or X-Auth-Admin-Key is incorrect.
- 5xx – Internal error.

Sample Response

```
HTTP/1.1 200 OK

{ "accounts":
  [
    { "name": "account1" },
    { "name": "account2" },
    { "name": "account3" }
  ]
}
```

Retrieving Account Details

To retrieve the details of an account, use the following HTTP GET request:

GET /auth/v2/<account> HTTP/1.1

A JSON dictionary of **account_ids**, **services** and **users** is returned.

- The account_id is the value used in creating account IDs.
- The services value is a dictionary that represents valid storage cluster endpoints, and identifies the default endpoint.
- The users value is a list of dictionaries, each dictionary representing a user and currently containing the key name.

Required Headers

- X-Auth-Admin-User
- X-Auth-Admin-Key

Note For this request to be permitted, the user identified by X-Auth-Admin-User must have super admin or reseller admin privileges, or must be an admin for the specified account.

Response Status Codes

- 200 – Success.
- 401 – Unauthorized; X-Auth-Admin-User or X-Auth-Admin-Key is incorrect.
- 403 – Invalid X-Auth-Admin-User/X-Auth-Admin-Key; the X-Auth-Admin-User does not have super admin, reseller admin, or account admin privileges.
- 5xx – Internal error.

Sample Response

```
HTTP/1.1 200 OK

{ "services":
  { "storage":
    { "default": "local",
      "local": "https://<storage endpoint>/v1/<account_id>" },
```

```
  },
  "account_id": "<account_id>",
  "users": [ { "name": "user1" },
             { "name": "user2" } ]
}
```

Creating an Account

To create a new authentication account, use the following HTTP PUT request:

PUT /auth/v2/<account> HTTP/1.1

Note To create a new account, you must be a super admin or a reseller admin.

An authentication account allows you to manage a collection of related users, groups and service catalogs.

Choosing the Account Name

- The name must not begin with a period (**.**).
- The name must not include a colon (**:**).
- The name must not exceed 256 bytes in length.

Required Request Headers

- X-Auth-Admin-User: <admin user name>

Note The <admin user name> for users other than the super admin must be of the form <account-name>:<user-name>.

- X-Auth-Admin-Key: <admin user password>

Optional Request Header

- X-Account-Suffix: <account ID suffix>
  - When an account is created, a new UUID with the reseller prefix form the account ID (terms defined below). Using this header, you can replace the UUID4 part of the ID with the <account ID suffix>.
  - The <account ID suffix> must not exceed 251 bytes in length.

Note

- A *UUID* (universally unique identifier) is a standard 128-bit identifier value. The relevant IETF specification (RFC 4122) defines five UUID versions, designated UUID1–UUID5, each having a different method of construction. A UUID4 is generated in part using pseudorandom numbers.
- The *reseller prefix* is, by default, the AUTH_ string that prefixes account-ids and auth tokens created by Swauth. If COS adds support for other auth services in the future, each auth service will have a distinct reseller prefix.

Response Status Codes

The response status code is one of the following:

- 201 – Account was created.
- 202 – Account already exists.
- 400 – Account name is invalid.
- 401 – Unauthorized; X-Auth-Admin-User or X-Auth-Admin-Key is incorrect.
- 403 – User not authorized to create an account.
- 5xx – Internal error.

Deleting an Account

To delete an authentication account, use the following HTTP DELETE request:

DELETE /auth/v2/<account> HTTP/1.1

Note

- To delete an account, you must be a super admin or a reseller admin.
- The account should not have users, containers and/or objects.

Required Request Headers

- X-Auth-Admin-User: <admin user name>

Note The <admin user name> for users other than the super admin must be of the form <account-name>:<user-name>.

- X-Auth-Admin-Key: <admin user password>

Response Status Codes

The response status code is one of the following:

- 204 – Account was deleted.
- 401 – Unauthorized; X-Auth-Admin-User or X-Auth-Admin-Key is incorrect.
- 403 – User not authorized to delete the account.
- 404 – Account not found.
- 409 – Account is not empty.
- 5xx – Internal error.

Creating or Updating a User

To create a user who can access storage services, use the following HTTP PUT request:

PUT /auth/v2/<account>/<user> HTTP/1.1

Note

- Only the super admin can create reseller admin users.
- An account admin, an authorized reseller admin, or the super admin can create regular and admin users.
- A reseller admin can create a user in any account.
- Admins can create users only in their respective accounts.

Choosing the User Name

- The name must not begin with a period (**.**)
- The name must not exceed 256 bytes in length.

Required Request Headers

- X-Auth-Admin-User: <admin user name>

Note The <admin user name> for users other than the super admin must be of the form <account-name>:<user-name>.

- X-Auth-Admin-Key: <admin user password>

Optional Request Headers

- X-Auth-User-Key: <new user password>
  - This header allows you to specify the password for a new user.
  - The header can also be used by existing users to change their password. Here, the X-Auth-Admin-Key header must have their current password, and the X-Auth-User-Key, the new password.
- X-Auth-User-Admin: true
  - This header allows you to grant admin privileges to the user being created.
- X-Auth-User-Reseller-Admin: true
  - This header allows you to grant reseller admin privileges to the user being created.

Note The admin privileges of an existing user cannot be modified by using the **X-Auth-User-Admin: true** and **X-Auth-User-Reseller-Admin: true** headers.

Response Status Codes

The response status code is one of the following:

- 201 – Success.
- 202 – Account already exists.
- 400 – Invalid user name.
- 401 – Invalid X-Auth-Admin-User and/or X-Auth-Admin-Key.
- 403 – User not authorized to perform the operation.
- 404 – Account does not exist.
- 5xx – Internal error.

Retrieving User Details

To retrieve the details of a user, use the following HTTP GET request:

GET /auth/v2/<account>/<user> HTTP/1.1

A JSON dictionary of the following format is returned:

```
{"groups": [  # List of groups the user is a member of
    {"name": "<act>:<usr>"},
        # The first group is a unique user identifier
    {"name": "<account>"},
        # The second group is the auth account name
    {"name": "<additional-group>"}
        # There may be additional groups, .admin being a
        # special group indicating an account admin and
        # .reseller_admin indicating a reseller admin.
 ],
 "auth": "<auth-type>:<key>"
 # The auth-type and key for the user; currently only
 # plaintext is implemented as auth type.
}
```

Required Headers

- X-Auth-Admin-User
  - This header must be set to super-admin to retrieve details of reseller-admin users.
  - This header must be set to super-admin or reseller-admin to retrieve details of account-admin users.
  - This header must be account-admin to retrieve details of unprivileged users.
- X-Auth-Admin-Key

Response Status Codes

- 200 – Success.
- 400 – The account or user name starts with a period (.)
- 401 – Invalid X-Auth-Admin-User/X-Auth-Admin-Key.
- 403 – Retrieval of the requested user forbidden to admin user; insufficient privileges.
- 404 – Unknown account or user.
- 5xx – Internal error.

Deleting a User

To remove a user, use the following HTTP DELETE request:

DELETE /auth/v2/<account>/<user> HTTP/1.1

Note

- A super admin or a reseller admin can delete users from any account.
- Account admins can only delete users from the accounts they administer.

Required Request Headers

- X-Auth-Admin-User: <admin user name>

Note The <admin user name> for users other than the super admin must be of the form <account-name>:<user-name>.

- X-Auth-Admin-Key: <admin user password>

Response Status Codes

The response status code is one of the following:

- 200 – User deleted.
- 400 – Invalid user name.
- 401 – Unauthorized; X-Auth-Admin-User or X-Auth-Admin-Key is incorrect.
- 403 – User not authorized to perform the operation.
- 404 – Account/user does not exist.
- 5xx – Internal error.

Creating or Updating Account Service Endpoints

To create new service endpoints or update existing ones, use the following HTTP POST request on the pseudo-user '.services':

POST /auth/v2/<account>/.services HTTP/1.0

The request must also contain the a JSON dictionary of the following format:

{"service_name": {"end_point_name": "end_point_value"}}

- Multiple services and multiple endpoints can be specified in a single request.
- New services and end points will be added to the existing set of services and end points, respectively.
- If the service specified exists, new end points will be linked to it.

- If the endpoint specified exists, its value is updated.

The updated services dictionary will be returned on success.

Required Request Headers

- X-Auth-Admin-User: <admin user name>

Note The <admin user name> for users other than the super admin must be of the form <account-name>:<user-name>.

- X-Auth-Admin-Key: <admin user password>

Response Status Codes

- 200 – Success.
- 403 – Invalid X-Auth-Admin-User/X-Auth-Admin-Key.
- 404 – Account not found.
- 5xx – Internal error.

Getting an Authentication Token

To get an authentication token, use the following HTTP GET request:

GET /auth/v1.0 HTTP/1.1

Required Request Headers

- X-Auth-User: <user name>
- X-Auth-Key: <user password>

Note The <user name> for users other than the super admin must be of the form <account-name>:<user-name>.

Optional Request Headers

- X-Auth-Token-Lifetime: <integer> – The time in seconds until the token expires.
- X-Auth-Token-New: <boolean> – If set to **true**, a new token is created and any previous token is revoked.

Response Headers

On successful authentication, the response will include the following headers:

- X-Auth-Token: <string>
  - The authentication token to use in Swift API requests.

- X-Storage-URL: <string>
  - The URL to use with Swift API requests, including the API version and the account ID associated with the user's account.
- X-Token-Expires: <integer>
  - The number of seconds remaining until the token expires.

Response Body

The response body will be set to the account's services JSON object as described here:

```
{"storage": {     # Represents the Swift storage service end points
    "default": "cluster1", # Indicates which cluster is the default
    "cluster1": "<URL to use with Swift>",
        # A Swift cluster that can be used with this account,
        # "cluster1" is the name of the cluster which is usually a
        # location indicator (like "dfw" for a datacenter region).
    "cluster2": "<URL to use with Swift>"
        # Another Swift cluster that can be used with this account,
        # there will always be at least one Swift cluster to use or
        # this whole "storage" dict won't be included at all.
 },
# Possibly other service dicts, not implemented yet.
}
```

Swift API

This chapter describes the subset of the OpenStack Swift API that is implemented for COS. The Swift API uses the Storage FQDN that is assigned to the COS cluster.

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.

Listing Containers

To retrieve a list of existing storage containers, use the following HTTP GET request:

Note To make this request, you must be an account admin or a reseller admin.

GET /v1/<account ID>[?<param>=<value>[&<param>=<value>]] HTTP/1.1

where <account ID> is the account-id returned in the X-Storage-URL header of the Get Authentication Token response.

The names of the containers in the list are sorted based on a binary comparison of the UTF-8 encoded container names.

Required Request Header

X-Auth-Token: <user token>

Optional Query Parameters

The following parameters can be used in the query:

- *limit* – Specifies the maximum number of results to be retrieved.
- *marker* – Retrieves container names whose characters have a greater Unicode alphabetical value than those of the specified string.

- *end_marker* – Retrieves container names whose characters have a lower Unicode alphabetical value than those of the specified string.
- *prefix* – Retrieves container names beginning with the specified characters.
- *delimiter* – Retrieves container names that do not have the specified character, except in the prefix, if any.
- *format* – Specifies either json or xml as the format of the serialized response.

Listing Objects

To list the objects in the storage container, use the following HTTP GET request:

GET /v1/<account ID>/<container>[?<param>=<value>[&<parm>=<value>]] HTTP/1.1

Note To make this request, you must be an account administrator or COS reseller-administrator. Or, if a read-ACL has been specified for a container, either you must have been assigned a role listed in the ACL, or * must be listed as a role, permitting anonymous access.

The names of the objects in the list are sorted based on a binary comparison of the UTF-8 encoded object names.

Optional Query Parameters

The following parameters can be used in the query:

- *limit* – Specifies the maximum number of results to be retrieved.
- *marker* – Retrieves object names whose characters have a greater Unicode alphabetical value than those of the specified string.
- *end_marker* – Retrieves object names whose characters have a lower Unicode alphabetical value than those of the specified string.
- *prefix* – Retrieves object names beginning with the specified characters.
- *delimiter* – Retrieves object names that do not have the specified character, except in the prefix, if any.
- *path* – Retrieves names of objects nested in the specified path.
- *format* – Specifies either json or xml as the format of the serialized response.

Note If a response format is not specified as a query parameter, a list of object names is returned in the response body, one name per line.

Response Status Codes

The response status code is one of the following:

- 2xx – indicates a successful execution of the request.
- 204 (No Content) – indicates that either the container is empty, or none of the objects in the container match the query parameters specified in the request.

Creating a Container

To create a storage container, use the following HTTP PUT request:

PUT /v1/<account ID>/<container> HTTP/1.1

Note To make this request, you must be an account admin, a COS reseller admin, or the super admin.

The name of the container must adhere to the following restrictions:

- The name cannot include the forward slash (/) character or the encoded forward slash character (%2F or %2f).
- The name should not exceed 256 bytes when it is encoded in URL.

Required Request Header

- X-Auth-Token: <user token>

Optional Request Headers

- X-Container-Read: <read acl>
- X-Container-Write: <write acl>

Assigning Custom Attributes

To assign custom attributes to a storage container, include additional HTTP headers in the HTTP PUT request shown above. The additional headers should be of the following form:

X-Container-Meta-<attribute name>: <attribute value>

Response Status Codes

The response status code is one of the following:

- 201 (Created) – The container was created.
- 202 (Accepted) – The container already exists.
- 400 (Bad Request) – The container name is invalid.
- 401 (Unauthorized) – The user token is missing or invalid.
- 403 (Forbidden) – The user does not have permission to create the container.
- 404 (Not Found) – The storage URL references a non-existent account ID.

Deleting a Container

To permanently remove a storage container, use the following HTTP DELETE request:

DELETE /v1/<account ID>/<container> HTTP/1.

Note

- To make this request, you must be an account admin, a reseller admin, or the super admin.
- Only empty storage containers can be deleted.

Required Request Header

X-Auth-Token: <user token>

Response Status Codes

The response status code is one of the following:

- 204 (No Content) – The container was deleted.
- 400 (Bad Request) – The container name is invalid.
- 401 (Unauthorized) – The user token is missing or invalid.
- 403 (Forbidden) – The user does not have permission to delete the container.
- 404 (Not Found) – The storage URL references a non-existent account ID.
- 409 (Conflict) – The container is not empty.

Retrieving an Object

To retrieve the data of an object, use the following HTTP GET request:

GET /v1/<account ID>/<container>/<object> HTTP/1.1

Note

- To make this request, you must be an account admin or COS reseller-admin; or if a read-ACL has been specified for the container, either you are assigned a role listed in the ACL, or * is specified as a role, permitting anonymous access.
- You may retrieve data from an object while it is being created. If you do, the response will have the X-Object-Is-Dynamic: yes header.

Required Request Header

X-Auth-Token: <user token>

Note If the container read-ACL permits anonymous access, this header is not required.

Optional Request Headers

- X-Follow-Redirect: true
  - If the request has this header, the COS node may respond with a 307 (Temporary Redirect) code and include a Location response header having the URL at which the client should retry the request.
  - If the request does not have this header, or the value of the header is not true, the COS node receiving the request will respond with the requested object.

Note The X-Follow-Redirect request header is a COS extension.

- X-Transfer-Rate: <bits-per-second>
  - This header specifies a transfer rate in decimal bits per second.
  - Valid range of values for this header is 400000 to 50000000, that is, 400 Kbps to 50 Mbps.
  - If the header is not included, data is transferred at the best-effort rate that does not delay other transfers which were requested along with an X-Transfer-Rate header.
  - A transfer rate of 0 is valid and indicates a best-effort transfer of data.

Note The X-Transfer-Rate request header is a COS extension.

- X-Transfer-Delay: <delta-time-in-milliseconds>
  - This header specifies a signed delay in milliseconds.
  - If the delay is positive, COS waits for the time interval specified before starting the transfer of object data at the requested transfer rate. The positive delay can have a maximum value of 30 seconds.
  - If the delay is negative, the client wants to use an elasticity buffer and intends to transmit data from the partially full buffer that receives data from COS at the requested transfer rate.

    In this case, COS starts transferring the data as soon as the retrieve request is received, and attempts to send data at a rate higher than the requested rate. COS, in essence, tries to match the amount of data that would have been sent to the client if the data transfer had been initiated **delay** seconds before the receipt of the request.

    The negative delay can have maximum magnitude of four seconds.

Note The X-Transfer-Delay request header is a COS extension.

- Range: bytes = <byte-range>
  - To request the transfer of specific portions of the object data, in accordance with the specifications in section 14.35 of RFC 2616, include this header.
  - Only byte-ranges are supported.
  - Multiple byte ranges are supported.
  - If the range did not include the entire object, a response status code of 206 (Partial Content) is returned by COS.
  - The Partial Content response to a request for multiple non-overlapping ranges of data contains multiple parts in the message body.

- If-Match: ETag
  - The object data is retrieved only if the client specified ETag value matches the ETag of the content. Else, 412 (Precondition Failed) is returned.
- If-None-Match: ETag
  - 304 (Not Modified) is returned if the client specified ETag value matches the ETag of the content, indicating to the client that the object cached by it has not been modified since.
- If-Modified-Since: time
  - 304 (Not Modified) is returned if the client specified time is equal to or later than the last modified time of the object.
- If-Unmodified-Since: time
  - The object data is retrieved only if the client specified time is equal to or later than the last modified time of the object. Else, 412 (Precondition Failed) is returned.

Response Status Codes

A response status code of **2xx** indicates successful completion of the request.

Response Headers

The response will include one of the following headers:

- Last Modified – A time-stamp of when the object was created or modified.
- ETag – The hexadecimal representation of the MD5 hash of the object data.
- Content-Type – The content type associated with the object when it was created.
- Content-Length – The number of bytes in the object.
- X-Object-Goid – The global object identifier assigned by COS when the object was created.

Note The X-Object-Goid response header is a COS extension.

- X-Object-Meta-* – Custom object attributes of the object.
- X-Object-Is-Dynamic: yes – The object being retrieved is being extended. This occurs when you retrieve object data while object creation is in progress. Such retrieval is useful when a large object is being created and you want to access the data that has been stored, even as more data is being appended to the object.

Note The X-Object-Is-Dynamic response header is a COS extension.

Creating or Updating An Object

To create or update an object, that is, to write or overwrite an object's content and metadata, use the following HTTP PUT request:

PUT /v1/<account ID>/<container>/<object> HTTP/1.1

Required Request Headers

- X-Auth-Token: <user token>

Note This header may be excluded if the container write ACL permits anonymous access.

- Content-Type
  - If this header is not included, the system will attempt to guess the type of the content based on the object name and metadata. If the system is unsuccessful, the Content-Type is set to the default application or octet-stream value.
- Content-Length/ Transfer-Encoding: chunked
  - The request must include either a valid Content-Length header stating the size of the object, or a Transfer-Encoding: chunked header indicating that the data length is encoded in-line at the start of each chunk of the object data sent in the request.

Optional Request Headers

- ETag
  - The request may include this header with the value set to the hexadecimal representation of the MD5 hash of the object data.
  - If the ETag value does not match MD5 hash computed by COS, a 422 (Unprocessable Entity) response status code is returned.
- X-Object-Meta-<attribute name>: <attribute value>
  - Include this header to set custom attributes for an object.
- X-Follow-Redirect: true
  - If this header is included, the COS node may respond with a status code of 307 (Temporary Redirect) and include a Location response header indicating the URL to which the client should address the request.
  - If this header is absent, or if the value of the header is not true, the COS node receiving the request performs the necessary operation.

Note The X-Follow-Redirect request header is a COS extension.

- Expect: 100-Continue
  - It is recommended that you include this header in the initial request and omit object content from the body of the request.
  - The receiving COS node will respond with either a 100 (Continue) status code, a 307 (Temporary Redirect) status code, or an error status code.
  - If the 100 (Continue) status code is received, repeat the request along with the object content in the body.

Response Status Codes

The response status code is one of the following:

- 2xx – Indicates successful execution of the HTTP PUT request.
- 5xx – Indicates failure to execute the HTTP PUT request.

Response Headers

The response will include one of the following headers:

- Last Modified – A time-stamp of when the object was created or updated.
- ETag – A hexadecimal representation of the MD5 hash of the object data.

Deleting an Object

To permanently remove an object, use the following HTTP DELETE request:

DELETE /v1/<account ID>/<container>/<object> HTTP/1.1

Note To make this request, *one* of the following must be true:

- You are an account admin or a COS reseller admin.
- If a write-ACL has been specified for the container, either you are assigned a role listed in the ACL, or the role * is included in the ACL, permitting anonymous access.

Deleting an object removes both the object data and metadata. Any subsequent operations attempted on the object will return a 404 (Not Found) response status code.

Required Request Header

Note If the container write-ACL permits anonymous access, this header is not required.

X-Auth-Token: <user token>

Response Status Code

204 (No Content) – Indicates that the object has been deleted.

Creating or Updating Container Metadata

To create or update custom container attributes, use the following HTTP POST request:

POST /v1/<account ID>/<container> HTTP/1.1

Note To make this request, you must be an account admin or a COS reseller admin.

The attributes are specified in HTTP headers included in the HTTP POST request. If the attribute exists, its value will be overwritten. Else, a new attribute is created.

Required Request Header

- X-Auth-Token: <user token>
- X-Container-Meta-<attribute name>: <attribute value>

Response Status Code

The response status code is one of the following:

- 204 (No Content) – The POST operation succeeded.
- 400 (Bad Request) – The POST request is not valid.
- 401 (Unauthorized) – The user token is missing or invalid.
- 403 (Forbidden) – The user does not have permission to modify the container attributes.
- 404 (Not Found) – The storage URL references a non-existent account ID or container.
- 5xx – Internal Server Error.

Retrieving Container Metadata

To retrieve a container's metadata to learn its status, use the following HTTP HEAD request:

HEAD /v1/<account ID>/<container> HTTP/1.1

Note

- To make this request, you must be an account admin or a COS reseller admin.
- Or, if a read-ACL has been specified for the container, you must be assigned a role listed in the ACL.
- Or, the * role must be included in the ACL, permitting anonymous access.

This request can be used against a container to determine the number of objects, and the total byte size of all objects stored in the container.

Required Request Header

Note If the container read-ACL permits anonymous access, this header is not required.

X-Auth-Token: <user token>

Response Status Code

The response status code is one of the following:

- 2xx (Success) – The HEAD operation succeeded.
- 400 (Bad Request) – The POST request is not valid.
- 401 (Unauthorized) – The user token is missing or invalid.
- 403 (Forbidden) – The user does not have permission to modify the container attributes.
- 404 (Not Found) – The storage URL references a non-existent account ID or container.
- 5xx – Internal Server Error.

Response Headers

- X-Container-Object-Count
  - The value of this header is the number of objects in the container.
- X-Container-Bytes-Used
  - The value of this header is the total byte size of all the objects in the container.
- X-Container-Meta-<attribute name>: <attribute value>
  - This header returns the custom attributes of the container.

Deleting Container Metadata

To delete custom container attributes, use the following HTTP POST request:

POST /v1/<account ID>/<container> HTTP/1.1

Note To make this request, you must be an account admin or a reseller admin.

Required Request Headers

- X-Auth-Token: <user token>
- X-Container-Meta-<attribute name>
  - An empty header of this type without the attribute value can be used to delete the custom attribute named in the header.
- X-Remove-Container-Meta-<attribute name>: <arbitrary value>
  - Alternatively, a header of this type can be used to delete the custom attribute named in the header.
  - The arbitrary attribute value is ignored by the system.

Response Status Code

The response status code is one of the following:

- 204 (No Content) – The POST operation succeeded.
- 400 (Bad Request) – The POST request is not valid.
- 401 (Unauthorized) – The user token is missing or invalid.
- 403 (Forbidden) – The user does not have permission to modify the container attributes.
- 404 (Not Found) – The storage URL references a non-existent account ID or container.
- 5XX – Server Error.

Creating or Updating Object Metadata

To create or update custom object attributes, use the following HTTP POST request:

POST /v1/<account ID>/<container>/<object> HTTP/1.1

Note To make this request, *one* of the following must be true:

- You are an account admin or a COS reseller admin.
- If a write-ACL is specified for the container, you must be assigned a role listed in the ACL.
- The * role must be included in the ACL, permitting anonymous access.

Assigning custom attributes to objects enables you to better categorize the objects.

Required Request Header

- X-Auth-Token: <user token>

Note If the container write-ACL permits anonymous access, this header is not required.

- X-Object-Meta-<attribute name>: <attribute value>
  - If the attribute exists, its value is updated to that specified in the header. Else, the attribute is created.

Response Status Code

The response status code is one of the following:

- 204 (No Content) – The POST operation succeeded.
- 400 (Bad Request) – The POST request is not valid.
- 401 (Unauthorized) – The user token is missing or invalid.
- 403 (Forbidden) – The user does not have permission to modify the container attributes.
- 404 (Not Found) – The storage URL references a non-existent account ID or container.
- 5XX – Internal Server Error.

Retrieving Object Metadata

To retrieve an object's metadata, including its custom attributes, use the following HTTP HEAD request:

HEAD /v1/<account ID>/<container>/<object> HTTP/1.

Note To make this request, *one* of the following must be true:

- You are an account admin or a COS reseller admin.
- If a read-ACL is specified for the container, you must be assigned a role listed in the ACL, and the Referrer header in the request must match the ACL referrer pattern.
- The * role must be included in the ACL, permitting anonymous access.

Required Request Header

Note If the container read-ACL permits anonymous access, this header is not required.

X-Auth-Token: <user token>

Response Status Code

The response status code is one of the following:

- 2xx (Success) – The HEAD operation succeeded.
- 400 (Bad Request) – The HEAD operation is not valid.
- 401 (Unauthorized) – The user token is missing or invalid.
- 403 (Forbidden) – The user does not have permission to modify the container attributes.
- 404 (Not Found) – The storage URL references a non-existent account ID or container.
- 5xx – Internal Server Error.

Response Headers

- Last Modified – A time-stamp of when the object was created or modified.
- ETag – The hexadecimal representation of the MD5 hash of the object data.
- Content-Type – The content type associated with the object when it was created.
- Content-Length – The number of bytes in the object.
- X-Object-Meta-<attribute-name>: <attribute-value> – Custom object attributes, if any.

Deleting Object Metadata

To delete custom object attributes, use the following HTTP POST request:

POST /v1/<account ID> HTTP/1.1

Note To make this request, *one* of the following must be true:

- You are an account admin or a COS reseller admin.
- If a write-ACL is specified for the container, you must be assigned a role listed in the ACL.
- The * role must be included in the ACL, permitting anonymous access.

Required Request Headers

- X-Auth-Token: <user token>
- X-Object-Meta-<attribute name>:
  - An empty header of this type without the attribute value can be used to delete the custom attribute named in the header.
- X-Remove-Object-Meta-<attribute name>: <arbitrary value>
  - Alternatively, a header of this type can be used to delete the custom attribute named in the header.
  - The arbitrary value is ignored by the system.

Response Status Code

The response status code is one of the following:

- 204 (No Content) – The POST operation succeeded.
- 400 (Bad Request) – The POST operation is not valid.
- 401 (Unauthorized) – The user token is missing or invalid.
- 403 (Forbidden) – The user does not have permission to modify the container attributes.
- 404 (Not Found) – The storage URL references a non-existent account ID or container.
- 5xx – Internal Server Error.

Creating or Updating Account Metadata

To create or update custom account attributes, use the following HTTP POST request:

POST /v1/<account ID> HTTP/1.1

Note To perform this operation, you must be an account administrator or a COS reseller admin.

Required Request Headers

- X-Auth-Token: <user token>
- X-Account-Meta-<attribute name>: <attribute value>
  - If the attribute exists, its value is updated to that specified in the header. Else, the attribute is created.

Response Status Code

The response status code is one of the following:

- 204 (No Content) – The POST operation succeeded.
- 400 (Bad Request) – The POST operation is not valid.
- 401 (Unauthorized) – The user token is missing or invalid.
- 403 (Forbidden) – The user does not have permission to modify the container attributes.
- 404 (Not Found) – The storage URL references a non-existent account ID.
- 5xx – Internal Server Error.

Retrieving Account Metadata

To retrieve account metadata to check the account statistics, use the following HTTP HEAD request:

HEAD /v1/<account ID> HTTP/1.1

Note To perform the operation, you must be an account administrator or a COS reseller admin.

Required Request Headers

X-Auth-Token: <user token>

Response Status Codes

The response status code is one of the following:

- 2xx (Success) – The HEAD operation succeeded.
- 400 (Bad Request) – The HEAD operation is not valid.
- 401 (Unauthorized) – The user token is missing or invalid.
- 403 (Forbidden) – The user does not have permission to modify the container attributes.
- 404 (Not Found) – The storage URL references a non-existent account ID.
- 5xx – Internal Server Error.

Response Headers

- X-Account-Container-Count: <value>
  - The value of the header is the number of containers in the account.
- X-Account-Object-Count: <value>
  - The value of the header is the number of objects in the account.
- X-Account-Bytes-Used: <value>
  - The value of the header is the total number of bytes in COS for the specified account.
- X-Account-Meta-<attribute name>: <attribute value>
  - The header returns custom account attributes and their values.

Deleting Account Metadata

To delete custom account attributes, use the following HTTP POST request:

POST /v1/<account ID> HTTP/1.1

Note To perform the operation, you must be an account administrator or a COS reseller admin.

Required Request Headers

- X-Auth-Token: <user token>
- X-Account-Meta-<attribute name>:
  - An empty header of this type without the attribute value can be used to delete the custom attribute named in the header.
- X-Remove-Account-Meta-<attribute name>: <arbitrary value>
  - Alternatively, a header of this type can be used to delete the custom attribute named in the header.
  - The arbitrary value is ignored by the system.

Response Status Code

The response status code is one of the following:

- 204 (No Content) – The POST operation succeeded.
- 400 (Bad Request) – The POST operation is not valid.
- 401 (Unauthorized) – The user token is missing or invalid.
- 403 (Forbidden) – The user does not have permission to modify the container attributes.
- 404 (Not Found) – The storage URL references a non-existent account ID.
- 5xx – Internal Server Error.

Access Control Lists (ACLs)

By default, to access a storage object, a requester must be an account administrator of the account containing the object. An administrator can modify the access policy for a container and its storage objects by using container access control lists (ACLs). The administrator can specify the read and write access control lists as part of the container metadata. When an ACL is deleted, the default access policy is restored.

An ACL has the following form:

[item [, item...]]

An ACL item can be one of the following:

- <account name>
  - All the users of specified account are granted access to objects in the container.
- <account name>: <user name>
  - Users identified by the combination of the specified account and user names are granted access to objects in the container.
- *
  - An asterisk permits anonymous access. This option is a COS extension.

Creating or Updating ACLs

To create or update an ACL, use the following HTTP POST request:

POST /v1/<account ID>/<container> HTTP/1.1

Note To perform the operation, you must be an account administrator or a COS reseller admin.

Required Request Headers

- X-Auth-Token: <user token>
- X-Container-Read: <read acl>
  - Specify the read ACL as the value of this header.
- X-Container-Write: <write acl>
  - Specify the write ACL as the value of this header.

Response Status Code

The response status code is one of the following:

- 204 (No Content) – The POST operation succeeded.
- 400 (Bad Request) – The POST request is not valid.
- 401 (Unauthorized) – The user token is missing or invalid.
- 403 (Forbidden) – The user does not have permission to modify the container attributes.
- 404 (Not Found) – The storage URL references a non-existent account ID or container.

- 5xx – Internal Server Error.

Deleting ACLs

To delete an ACL, use the following HTTP POST request:

POST /v1/<account ID>/<container> HTTP/1.1

Note To perform the operation, you must be an account administrator or a COS reseller admin.

Required Request Headers

- X-Auth-Token: <user token>
- X-Container-Read: <read acl>
  - Specify an empty list as the value of this header to delete the read ACL.
- X-Remove-Container-Read: <arbitrary value>
  - Alternatively, a header of this type can be used to delete the read ACL.
  - The arbitrary value is ignored by the system.
- X-Container-Write: <write acl>
  - Specify an empty list as the value of this header to delete the write ACL.
- X-Remove-Container-Write: <arbitrary value>
  - Alternatively, a header of this type can be used to delete the write ACL.
  - The arbitrary value is ignored by the system.

Response Status Code

The response status code is one of the following:

- 204 (No Content) – The POST operation succeeded.
- 400 (Bad Request) – The POST request is not valid.
- 401 (Unauthorized) – The user token is missing or invalid.
- 403 (Forbidden) – The user does not have permission to modify the container attributes.
- 404 (Not Found) – The storage URL references a non-existent account ID or container.
- 5xx – Internal Server Error.

Retrieving and Updating COS Configuration Settings

You can use the COS service configuration API to retrieve or update cluster-wide settings for the Swift API, the Swauth API, the S3 API, and the COS daemon (cosd) settings shared by all COS nodes.

The COS service configuration API lets you enable or disable optional Swift API features, and also to configure some of the limits for the core Swift API and any optionally enabled features.

Retrieving Configuration Settings

To retrieve the current configuration settings, send the following HTTP GET request to the cluster endpoint name (FQDN):

GET /config HTTP/1.1

Note

- Anonymous requests (without an x-auth-token header) are permitted.
- Unlike other Swift API requests, this request URL does not include **version** or **account-id** components.

Required Request Headers

- X-Auth-Admin-User: <super admin name>
- X-Auth-Admin-Key: <super admin key>

Response Format

The response body is a JSON object describing the current cluster configuration. At a minimum, a **swift** section is included that describes the version of the core feature-set of the Swift API and the Swift API constraints. Additional sections describe any non-core API features enabled for the COS cluster.

Response Status Codes

The response status code is one of the following:

- 200 OK – Returned if the request was accepted for processing.

JSON Response Body (Preliminary)

```
{
  "cluster": {
    "fqdn": "cos-cluster.my_company.com",
    "name": "local"
  },
  "swauth": {
    "reseller_prefix": "AUTH_",
    "path_prefix": "auth/",
    "super_admin_user": ".super_admin",
    "super_admin_key": "<omitted>",
    "token_life": 86400,
    "max_token_life": 86400,
    "max_account_len": 256,
    "max_user_len": 256,
```

```
    "max_key_len": 256
  },
  "swift": {
    "max_account_len": 256,
    "max_container_len": 256,
    "max_object_len": 1024,
    "max_container_list": 10000,
    "max_object_list": 10000
  },
  "log": {
    "default": "debug"
  }
}
```

Updating Configuration Settings

To change the current configuration settings, send the following HTTP POST request to the cluster endpoint name (FQDN):

POST /config HTTP/1.1

Note

- Anonymous requests (without an x-auth-token header) are permitted.
- Unlike other Swift API requests, this request URL does not include **version** or **account-id** components.

Required Request Headers

- X-Auth-Admin-User: <super admin name>
- X-Auth-Admin-Key: <super admin key>
- Content-Length: <length of the modification request body>
- Content-Type: application/json

Response Status Codes

The response status code is one of the following:

- 200 OK – Returned if the request was accepted for processing.

Request Body

The request body only needs to provide the elements to be updated. Therefore, it can be a subset of the configuration JSON object returned by the retrieval request described previously.

Example

The following example changes the maximum Swauth token life from its current value to 48 hours.

```
{
  "swauth": {
    "max_token_life": 172800
  }
}
```

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. This chapter describes the subset of these API calls that is implemented for COS.

Retrieving an Archive (DDN WOS) Object

This operation retrieves an archive object identified by a given object identifier (OID). Use one of the following HTTP GET request formats:

GET /objects/{oid} HTTP/1.1

GET /cmd/get HTTP/1.1

Note

- The body of the GET request must be empty.
- To make this request, you must be an account admin or COS reseller-admin; or if a read-ACL has been specified for the container, either you are assigned a role listed in the ACL, or * is specified as a role, permitting anonymous access.
- You may retrieve data from an object while it is being created. If you do, the response will have the X-Object-Is-Dynamic: yes header.

Required Request Header

- x-ddn-oid: <string>

The archive object identifier. This header is required for the **GET /cmd/get HTTP/1.1** request form, where the OID is not included in the request path.

Optional Request Headers

- Range: <range-specifier>
  - An optional single byte-range specification. The Range header value must conform to the RFC2616 specification syntax for a single range of bytes.
- x-ddn-no-meta: <boolean>
  - An optional header containing a true or false value.
  - If the header value is true, the x-ddn-header is omitted from the response.

  - If the value is false, or if this header is omitted, the response includes an x-ddn-meta header if any custom metadata was specified in the archive object creation request.
- x-ddn-length: <integer>
  - Returns the length of the archive object contents.
  - This header duplicates the Content-Length response header value.

Response Status

- 200 OK – The archive object was successfully returned.
- 206 Partial Content – The specified byte range of the archive object contents was returned.
- 404 Not Found – The OID does not exist.
- 413 Request Entity Too Large – The content length was greater than zero.

Response Headers

- x-ddn-meta: <key/value list> – A list of metadata key/value pairs
- x-ddn-oid: <string> – The archive object identifier
- x-ddn-status: <integer> <string> – Additional status information:
  - 0 OK – Success.
  - 203 InternalError – An unspecified internal error occurred.
  - 205 InvalidObjId – An invalid OID was specified.
  - 207 ObjNotFound – The OID could not be found.
  - 218 UnusedReservation – A GET was attempted on an unused OID reservation.

Reserving an Archive (DDN WOS) Object Identifier

This operation generates a unique object identifier (OID) and returns it without storing any content data. Because it designates an archive object, the reserved OID can only be used once for a single subsequent archive object creation operation.

Use the following HTTP POST request:

POST /cmd/reserve HTTP/1.1

Note The body of the POST request should be empty.

Required Request Headers

- Content-Length: 0
  - The request content length must be set to zero.

Response Status

- 200 OK – An archive object identifier was successfully reserved.
- 413 Request Entity Too Large – A nonzero content length was specified in the request.

Response Headers

- x-ddn-oid: <string> – The archive object identifier.
- x-ddn-status: <integer> <string> – Additional status information:
  - 0 OK – Success.
  - 203 InternalError – An unspecified internal error occurred.
  - 212 InvalidObjectSize – A nonzero content length was specified.

Creating an Archive (DDN WOS) Object

This operation creates an archive object using a previously reserved object identifier (OID). Because it creates an archive object, the reserved OID can only be used for a single creation operation. In addition, the archive object created using this OID cannot be updated.

Use the following HTTP POST request:

POST /cmd/putoid HTTP/1.1

The body of the POST request contains the archive object contents.

Required Request Headers

- Content-Length
  - The length of the object contents in bytes.
- Content-Type
  - Internet media type associated with the content.
  - Defaults to **application/octet-stream** if not specified.
- x-ddn-oid: <string>
  - The archive object identifier returned by the Reserve operation.

Optional Request Header

- x-ddn-meta
  - Optional list of metadata key/value pairs associated with the archive object.
  - Each key/value pair is of the form **“<key>”:”<value>”**. Each key and the value must be enclosed in double quotes. The colon separator is not enclosed in quotes.
  - Commas separate multiple key/value pairs. Keys must be less than 64 characters in length and must not begin with a colon.

Response Status

- 200 OK – The archive object was successfully created.
- 404 Not Found – A reservation was not found for the specified OID.
- 409 Conflict – The OID is in use.
- 413 Request Entity Too Large – The object exceeds the maximum supported size, or the cluster is full.

Response Headers

- x-ddn-oid: <string> – The archive object identifier.
- x-ddn-status: <integer> <string> – Additional status information:
  - 0 OK – Success.
  - 203 InternalError – An unspecified internal error occurred.
  - 205 InvalidObjId – An invalid OID was specified.
  - 206 NoSpace – The cluster is full.
  - 212 InvalidObjectSize – The object exceeds the maximum supported size.
  - 216 ReservationNotFound – The specified OID reservation was not found.
  - 218 InvalidMetadataKey – An invalid metadata key was specified.

Deleting an Archive (DDN WOS) Object

This operation deletes an archive object identified by a given object identifier (OID). Use the following HTTP Post request:

POST /cmd/delete HTTP/1.1

Note

- The body of the POST request must be empty.
- To make this request, you must be an account admin or COS reseller-admin; or if a read-ACL has been specified for the container, either you are assigned a role listed in the ACL, or * is specified as a role, permitting anonymous access.

Required Request Headers

- Content-Length: 0
  - The request body must be empty.
- x-ddn-oid: <string>
  - The identifier of the archive object to be deleted.

Response Status

- 200 OK – The archive object was successfully deleted.
- 404 Not Found – The OID does not exist.
- 413 Request Entity Too Large – The content length was greater than zero.

Response Headers

- x-ddn-oid: <string> – The archive object identifier.
- x-ddn-status: <integer> <string> – Additional status information:
  - 0 OK – Success.
  - 203 InternalError – An unspecified internal error occurred.
  - 205 InvalidObjId – An invalid OID was specified.
  - 207 ObjNotFound – The OID could not be found.

COS Configuration API

You can use the COS service configuration API to retrieve or update cluster-wide settings for the Swift API, Swauth API, Fanout API, 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 lets you configure some limits for API parameters.

Retrieving Non-Sensitive Configuration Settings

To retrieve the current configuration settings, except for user names and passwords, send the following HTTP GET request to the cluster endpoint name (FQDN):

GET /info HTTP/1.1

This request can be sent anonymously, without authentication headers. It can also be used as a basic COS service API health check. If the basic COS cluster components are operating and the client making the request can connect to the COS service interfaces, the request will succeed. The following is a successful JASON body:

```
{
  "health": 0
}
```

In the previous example a “0” indicates a successful request. If any other value had been returned, a failure would have occurred in the request.

To summarize, if there is no response to the request or the response includes a non-zero value for “health”, then the request indicates that there is some kind of problem.

Response Format

The response body is a JSON object describing the current cluster configuration. Sensitive information, including user names and passwords, is omitted. At a minimum, the response object includes a cluster section that describes general cluster settings, a swauth section that describes the Swauth API settings and constraints, a swift section that describes the version of the core feature-set of the Swift API and the Swift API constraints, a log section that describes the COS daemon (cosd) log levels, and a rio section that describes the Fanout API settings.

Response Status Codes

The response status code is one of the following:

- 200 OK – Returned if the request was accepted for processing.

Example JSON Response Body

(Extra spaces and line-breaks have been added for readability.)


```
{
  "cluster": {
    "config_ver": 2,
    "fqdn": "cos-cluster.my_company.com",
    "name": "local",
    "enable_wos": true
  },
  "swauth": {
    "reseller_prefix": "AUTH_",
    "path_prefix": "auth/",
    "max_key_len": 256,
    "max_user_len": 256,
    "token_life": 86400,
    "max_token_life": 86400,
    "max_account_len": 256
  },
  "swift": {
    "version": "2.2.0",
    "max_account_len": 256,
    "max_container_len": 256,
    "max_object_len": 1024,
    "max_container_list": 10000,
    "max_object_list": 10000
  },
  "log": {
    "default": "notice"
  },
  "rio": {
    "path_prefix": "rio/"
  }
}
```


Retrieving Configuration Settings

To retrieve the current configuration settings, including user names and passwords, send the following HTTP GET request to the cluster endpoint name (FQDN):

GET /config HTTP/1.1

Required Request Headers

- X-Auth-Admin-User: <super admin name>
- X-Auth-Admin-Key: <super admin key>

Response Format

The response body is a JSON object describing the current cluster configuration. At a minimum, the response object includes a cluster section that describes general cluster settings, a swauth section that describes the Swauth API settings and constraints, a swift section that describes the version of the core feature-set of the Swift API and the Swift API constraints, a log section that describes the COS daemon (cosd) log levels, and a rio section that describes the Fanout API settings.

Response Status Codes

The response status code is one of the following:

- 200 OK – Returned if the request was accepted for processing.
- 401 Unauthorized – Returned if the admin user name or password was incorrect.

Example JSON Response Body

(Extra spaces and line-breaks have been added for readability.)


```
{
  "cluster": {
    "config_ver": 2,
    "fqdn": "cos-cluster.my_company.com",
    "name": "local",
    "enable_wos": true
  },
  "swauth": {
    "reseller_prefix": "AUTH_",
    "path_prefix": "auth/",
    "super_admin_user": ".super_admin",
    "super_admin_key": "rootroot",
    "token_life": 86400,
    "max_token_life": 86400,
    "max_account_len": 256,
    "max_user_len": 256,
    "max_key_len": 256
  },
  "swift": {
    "version": "2.2.0",
    "max_account_len": 256,
    "max_container_len": 256,
    "max_object_len": 1024,
    "max_container_list": 10000,
    "max_object_list": 10000
  },
  "log": {
    "default": "notice"
  },
  "rio": {
    "path_prefix": "rio/",
    "rio_user": ".riouser",
    "rio_key": "rootroot"
  }
}
```

Updating Configuration Settings

To change the current configuration settings, send the following HTTP POST request to the cluster endpoint name (FQDN):

POST /config HTTP/1.1

Required Request Headers

- X-Auth-Admin-User: <super admin name>
- X-Auth-Admin-Key: <super admin key>
- Content-Length: <length of the modification request body>
- Content-Type: application/json

Response Status Codes

The response status code is one of the following:

- 200 OK – Returned if the request was accepted for processing
- 401 Unauthorized – Returned if the admin user or admin key was incorrect.

Request Body

The request body only needs to provide the elements to be updated. Therefore, it can be a subset of the configuration JSON object returned by the retrieval request described previously.

Example

The following example request body changes the maximum Swauth token lifetime from its current value to 48 hours:

```
{
  "swauth": {
    "max_token_life": 172800
  }
}
```

VMR Configuration Example

The following example changes the Fanout API user and password to “vmr” and “mysecret” respectively:

```
{
  "rio": {
    "rio_user": "vmr",
    "rio_key": "mysecret"
  }
}
```

Configuration Setting Descriptions

The following list describes the current COS service settings:

- cluster – settings affecting all APIs or enabling or disabling an entire API.
  - config_ver – The current version of the configuration JSON schema. (Read Only)
  - fqdn – The cluster FQDN. Copied into each new Swauth account's service dictionary.
  - name – The cluster name. Copied into each new Swauth account's service dictionary as the default endpoint name.
  - enable_wos – If set to true, the WOS API is enabled. If set to false, the WOS API is disabled.
- swauth – settings affecting the Swauth API.
  - reseller_prefix – This is the prefix for account IDs and auth tokens associated with the Swauth service.
  - path_prefix – This is the prefix for Swauth request paths, i.e., the 'auth/' portion of '/auth/v2'.
  - super_admin_user – The name of the cluster-wide super admin user.
  - super_admin_key – The password of the cluster-wide super admin user.
  - token_life – The default lifetime of a newly-created auth token in seconds.
  - max_token_life – The maximum lifetime of a newly-created auth token in seconds.
  - max_account_len – The maximum length of an account name.
  - max_user_len – The maximum length of a user name.
  - max_key_len – The maximum length of a user password.
- swift - settings affecting the Swift API.
  - version – The version of the Swift API supported by COS. (Read Only)
  - max_account_len – The maximum length of an account ID including the reseller prefix and the account ID suffix.
  - max_container_len – The maximum length of a container name.
  - max_object_len – The maximum length of an object name.
  - max_container_list – The maximum number of container entries included in a single list-containers response.
  - max_object list – The maximum number of object entries included in a single list-objects response.
- log – settings affecting the COS daemon (cosd) log.
  - o default – The log level for any message category not explicitly included in this section. The possible log level values, in decreasing order of severity, are "emerg", "alert", "crit", "err", "warning", "notice", "info", and "debug". The log will record entries with the configured level or higher severity.
  - o auth – Optional log level for the authentication message category.
  - o stor – Optional log level for the storage message category.
  - csvr – Optional log level for the cserver message category.
  - tcp – Optional log level for the tcp message category.
  - db – Optional log level for the database message category.

  - mesg – Optional log level for the general message processing category.
  - task – Optional log level for the task handling message category.
  - gen – Optional log level for the general message category.
- rio – settings affecting the Fanout API.
  - path_prefix – The prefix for Fanout API request paths, i.e., the ‘rio/’ in ‘/rio/object_name’.
  - rio_user – The name of the Fanout API user used to validate the Authorization header included in a Fanout API PUT, PATCH, or DELETE request.
  - rio_key – The password of the Fanout API user used to validate the Authorization header included in a Fanout API PUT, PATCH, or DELETE request.

Fanout API

Note The Fanout API is supported in production environments only for configurations of three or more nodes.

COS Release 3.16.1 includes API calls that enable COS to manage fanout storage operations for applications such as Cloud DVR (cDVR).

In this context, *fanout* refers to storing and retrieving multiple copies of specified media content. Fanout storage efficiently supports unique copies for fair-use compliance. A *fanout object* is a single logical unit of content that can represent one or multiple 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.

Because a single fanout request can save many copies of an object, use of the Fanout API saves network resources by optimizing storage compute and disk utilization.

The 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.

Terminology

The following terms have specific meanings in the context of COS Fanout API usage:

- Fanout – Storing and retrieving multiple copies of specified media content.
- Fanout object – A single logical unit of media content representing one or more copies.
- Copy index – The nth (0 based) index for copies of an object.

The following terms are specific to Cisco Virtual Media Recorder (VMR) Fanout API usage:

- Bucket – A collection of stored objects.
- Active storage – The bucket where the initial copy of a media segment is stored. Objects placed here will be fanned out.
- Archive storage – The bucket where copies of old recordings are placed. Objects placed here will not be fanned out.
- Recon storage – The bucket where unarchived copies are placed for playback. Objects placed here are not fanned out.

Configuring Basic Auth

Before using the Fanout API, you must first update the configurations of COS and VMR to allow VMR to have password-protected access to COS. This access is called *basic auth* (for basic authentication). A possible use case for Basic Auth is to prevent a client configured to send requests to a test cluster from accidentally modifying a production cluster.

COS manages a single user name and access credentials for use with the Fanout API. This VMR "user" can be modified, listed, and have its credentials updated using the COS Configuration API.

Note The only access credential COS currently supports for VMR is a plain text password.

VMR user configuration requests must be authorized using the COS cluster super-admin name (."super_admin") and password. The VMR user name is stored as the value of the *rio.rio_user* key in the cluster configuration JSON object. The VMR user password is stored as the value of the *rio.rio_key* key in the cluster configuration JSON object.

This section describes the configuration procedures for basic auth for both COS and VMR.

COS Configuration

Configure COS for basic auth as follows:

Step 1 Get the current configuration.

Note The admin password is the same as default root password.

```
[root@C3260_1 ~]# cos-ut get_config -k <admin password>
{
"cluster": {
"config_ver": 2,
"fqdn": "cluster_mos.cisco.com",
"name": "local",
"enable_wos": true
},
"swauth": {
"reseller_prefix": "AUTH_",
"path_prefix": "auth/",
"super_admin_user": ".super_admin",
"super_admin_key": "rootroot",
"token_life": 86400,
"max_token_life": 86400,
"max_account_len": 256,
"max_user_len": 256,
"max_key_len": 256
},
"swift": {
"version": "2.2.0",
"max_account_len": 256,
"max_container_len": 256,
"max_object_len": 1024,
"max_container_list": 10000,
"max_object_list": 10000
},
```

```
"log": {
"default": "notice"
},
"rio": {
"path_prefix": "rio/"
}
}
```

Step 2 Copy the configuration output to a file.

Step 3 Add the following lines to the **rio** section.

This example sets the basic auth user/pass to cisco/cisco123:

```
"rio_user": "cisco",
"rio_key": "cisco123"
```

The rio section of the file should now appear as follows:

```
"rio": {
"path_prefix": "rio/",
"rio_user": "cisco",
"rio_key": "cisco123"
}
```

Note Remember to add the comma to the end of the path_prefix line.

Step 4 Save the modified file.

Step 5 Update the COS configuration with the basic auth information:

```
[root@C3260_1 ~]# cos-ut set_config -k <admin password> <saved filename>
config updated successfully
```

Step 6 In a multi-node cluster, execute the following command to propagate the configuration to other nodes:

```
[root@C3260_1 ~]# echo 1 > /proc/calypso/internal/cos_send_config_update_notification
```

VMR Configuration

To configure VMR for basic auth, you must update the following RC files with new information:

- rio-k8s/segment-recorder/segment-recorder-rc.json
- rio-k8s/archive-agent/archive-agent-rc.json
- rio-k8s/reconstitution-agent/recon-agent-rc.json
- rio-k8s/dash-origin/dash-origin-rc.json
- rio-k8s/manifest-agent/manifest-agent-rc.json

You can make these changes manually or, if VMR is running as a service of the Cisco Virtualized Video Platform (V2P), automatically via the V2P Controller (V2PC) web GUI.

For manual configuration, you add a valid username and password for BASIC_AUTH to each of the RC files listed above, and then restart the corresponding POD.

This example sets the basic auth user/pass to cisco/cisco123:

```
{
"name": "BASIC_AUTH",
```

```
"value": "-basicauth=http://cisco:cisco123@$(ACTIVE_STORAGE_SERVICE_HOST):
$(ACTIVE_STORAGE_SERVICE_PORT) -basicauth=http://cisco:cisco123@
$(ARCHIVE_STORAGE_SERVICE_HOST):$(ARCHIVE_STORAGE_SERVICE_PORT)"
}
```

If V2PC is available, an alternative to manual configuration is to enter the COS objstoreUsername and objstorePassword when you create the VMR AIC in V2PC. The VIC then updates all of the RC files for you automatically. For details, see the *Cisco Virtual Media Recorder Deployment Guide*.

Fanout API Reference

The Fanout API includes the following operations:

List Fanout Objects

This operation lists the objects created using the Fanout API. The list is returned in an unsorted, consistent order. The **limit** and **marker** query parameters can be used to page through large sets of objects.

To list the objects created using the Fanout API, use the following HTTP GET request:

GET /rio?fanout[[&param=val]...] HTTP/1.1

Request Query Parameters

- max-keys=<number> – The response body will list at most <number> objects. If not specified, the default limit is 1,000 objects.
- fanout-key-marker=<object name> – The indicated <object name> should be the one returned by the last List Fanout Objects request via the fanout-key-marker value. The response to the current request will then continue with the next object or index after the ones provided.
- format=xml – The list is returned in XML format (the only format currently supported). The list includes the object names (one name per line) and additional information about each object.

Response Status

- 200 OK – The list was returned successfully.
- HTTP 400 - Bad Request – Requests that specify a fanout copy count or index outside of the acceptable range of values.
- HTTP 404 - Not Found – Requests for an object or index that does not exist.

Response Body

The XML body of the listing response includes the following parameters:

- IsTruncated – True when a listing response is truncated (for example, there are more pairs of fanout objects or indices to be retrieved with a subsequent call). When true, the FanoutKeyMarker and FanoutIndexMarker are also present in the response, and can be used on a subsequent request to retrieve the next block of listing results.
- NextFanoutKeyMarker – When a listing response is truncated, this is set to indicate the marker to use (in the fanout-key-marker query parameter) in a subsequent request to pick up where this listing left off.
- NextFanoutIndexMarker – When a listing response is truncated, this is set to indicate the index to use (in the fanout-index-marker query parameter) in a subsequent request to pick up where this listing left off.

Example

```
GET /rio?fanout&max-keys=9 HTTP/1.1
Host: host.com
Accept: /
HTTP/1.1 200 OK
Date: Wed, 12 Jul 2016 11:15:00 GMT
Content-Type: application/xml
Content-Length: 1004
Connection: close
<FanoutListingResult>
<NextFanoutKeyMarker>b</NextFanoutKeyMarker>
<NextFanoutIndexMarker>0</NextFanoutIndexMarker>
<IsTruncated>true</IsTruncated>
<FanoutObject>
<Key>d</Key>
<Index>0</Index>
</FanoutObject>
<FanoutObject>
<Key>d</Key>
<Index>1</Index>
</FanoutObject>
<FanoutObject>
<Key>e</Key>
<Index>0</Index>
</FanoutObject>
<FanoutObject>
<Key>e</Key>
<Index>1</Index>
</FanoutObject>
<FanoutObject>
<Key>c</Key>
<Index>0</Index>
</FanoutObject>
<FanoutObject>
<Key>c</Key>
<Index>1</Index>
</FanoutObject>
<FanoutObject>
```

```
<Key>a</Key>
<Index>0</Index>
</FanoutObject>
<FanoutObject>
<Key>a</Key>
<Index>1</Index>
</FanoutObject>
<FanoutObject>
<Key>b</Key>
<Index>0</Index>
</FanoutObject>
</FanoutListingResult>
```

Create Fanout Object

This operation creates a fanout object containing one or more unique copies of the provided content. Each copy may be individually retrieved or deleted.

To create a fanout object, use the following HTTP PUT request:

PUT /rio/bucket/object HTTP/1.1

The body of the PUT request contains a single copy of the contents.

Request Headers

- Authorization – The type of authorization. Must be set to “basic” and followed by a base-64 encoding of “<username>:<password>.”
- Content-Length – The length of the object contents in bytes.
- Content-Type – Internet media type associated with the content. Defaults to “application/octet-stream” if not specified.
- x-fanout-copy-count: <integer> – The number of copies to be created. The number of copies must be between 1 and 10,000. If this header is not included, the copy-count defaults to 1.

Example

```
PUT /active/segmentid/2 HTTP/1.1
Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
Content-Type: mpeg2/ts
Content-Length: 42
X-Fanout-Copy-Count: 10
[42 bytes object data]
HTTP/1.1 201 No Content
```

Response Status

- 200 OK – The object was successfully created.
- 201 No content.
- 204 Created.
- 400 Bad Request – The copy count was outside of the acceptable range.
- 401 Unauthorized – The user name or password (or both) in the Authorization header is incorrect.
- 413 Request Entity Too Large – The content exceeds the maximum supported size, or the cluster is full.

Response Headers

- ETag: “<hex string>” – MD5 checksum of a single copy of the content.

Retrieve Fanout Object

This operation retrieves the content of a single unique copy of the content associated with the fanout object.

To retrieve a single copy of a fanout object, use the following HTTP GET request:

GET /rio/bucket/object HTTP/1.1

Request Headers

- x-fanout-copy-index: <integer> - The zero-based index of the content copy to be returned. The copy index must be between 0 and 9,999.
- Range - a range header may optionally be included in Rio Read Object HTTP requests. The Range header value must be a single byte range, conforming to the HTTP Range specification. The range specified applies to the content copy indicated by the X-Fanout-Copy-Index request header, or to the entire object content if the object is not a fanout object. If the range start offset is beyond the end of the indicated content copy, or if the range value is syntactically invalid, an error status will be returned. If the range end offset is not specified or if it is beyond the end of the indicated copy, the returned content will be truncated to the end of the copy.

Example

```
GET /active/segmentid/2 HTTP/1.1
X-Fanout-Copy-Index: 2
Range: bytes=8-15

HTTP/1.1 200 OK
Etag: "1b2cf535f27731c974343645a3985328"
Content-Length: 42
Content-Type: mpeg2/ts
[42 bytes object data]
```

Response Status

- 200 OK – The copy of the object was successfully retrieved.
- 206 Partial Content - If a Rio Read Object request includes a range header and the requested range can be satisfied, this response is returned.
- 400 Bad Request – The copy index was outside the acceptable range.
- 404 Not Found – Either the object did not exist, or the specified copy had been previously deleted.
- 416 Requested Range Not Satisfiable - If a Rio Read Object request includes a range header and the requested range cannot be satisfied, this response is returned.

Response Headers

- Content-Length – The length of the object contents in bytes.
- Content-Type – Internet media type associated with the content.
- ETag: "<hex string>" – MD5 checksum of the content.

Content-Range Response Header

If the request includes a Range header and the requested range can be satisfied, the response will include a Content-Range header conforming to the HTTP Content-Range specification.

Retrieve Fanout Object Metadata

This operation retrieves the header information for a specified fanout object without the object body.

To retrieve the header information for a fanout object, use the following HTTP HEAD request:

HEAD /rio/bucket/object HTTP/1.1

Request Headers

- x-fanout-copy-index: <integer> – The zero-based index of the content copy to be returned. The copy index must be between 0 and 9,999.

Example

```
HEAD /active/segmentid/2 HTTP/1.1

204 No Content
Etag: "1b2cf535f27731c974343645a3985328"
Content-Length: 42
Content-Type: mpeg2/ts
```

Response Status

- 204 OK – The object header information was successfully requested.
- 400 Bad Request – The copy index was outside the acceptable range.
- 404 Not Found – Either the object did not exist, or the specified copy had been previously deleted.

Response Headers

- Content-Length – The length of the object contents in bytes.
- Content-Type – Internet media type associated with the content.
- ETag: "<hex string>" – MD5 checksum of the content.

Retrieve Fanout Object Location

This operation retrieves the location of a single unique copy of the content associated with the fanout object.

To retrieve the location of a single unique copy of the content associated with a fanout object, use the following HTTP GET request:

GET /rio/bucket/object?location HTTP/1.1

Request Headers

- x-fanout-copy-index: <integer> – The zero-based index of the content copy to be located. The copy index must be between 0 and 9,999. This parameter can be omitted for objects with copy-count 1, or where no copy-count was specified when the object was created.
- Range - an HTTP Range header may optionally be included in Rio Storage Location Service HTTP requests. The Range header value must be a single byte range, conforming to the HTTP Range specification. The range specified applies to the content copy indicated by the X-Fanout-Copy-Index request header, or to the entire object content if the object is not a fanout object. If the range start offset is beyond the end of the indicated content copy, or if the range value is syntactically invalid, an error status will be returned. If the range end offset is not specified or if it is beyond the end of the indicated copy, the returned content will be truncated to the end of the copy.

Response Body

The response body is a JSON object that includes the following key/value pairs:

- location – The URL of a storage node in the COS cluster where the specified object copy resides. The URL can include an optional *meta* query parameter whose value encodes the metadata associated with the object.
- Location if a Rio Range Header is specified – the Location value returned in the Rio Storage Location Service response body includes a "meta" query parameter. This parameter value includes information relating to the requested range, if specified. A Rio Read Object request that includes the provided "meta" query parameter will return the content range specified in the original Rio Storage Location Service Range header.

Response Status

- 200 OK – The location of the object copy was successfully retrieved. If a Rio Storage Location Service request includes a Range header and the requested range can be satisfied, this response is generated.
- 400 Bad Request – The copy index was outside the acceptable range.
- 404 Not Found – Either the object does not exist, or the specified copy was previously deleted.
- 416 Range Not Satisfiable – The copy index was greater than or equal to the copy count, but less than 10,000.

Response Headers

- Content Length – The length of the JSON response body in bytes.
- Content Type – application/json.

Example Response Body

```
{
  "location": "http://127.0.0.1/rio/bucket/object?meta=AAABBBCCCDDDEEEFFF"
}
```

Retrieve Fanout Object Using Pre-Retrieved Metadata

This operation retrieves the content of a single unique copy of the content associated with the fanout object, using metadata returned by a prior Retrieve Fanout Object Location request.

To retrieve the content of a single unique copy of the content associated with a fanout object using pre-retrieved metadata, use the following HTTP GET request:

GET /rio/bucket/object?meta=AAAABBBB HTTP/1.1

Request Query Parameters

- meta – The string included in the location URL returned by a previous Retrieve Fanout Object Location request.

Response Status

- 200 OK – The copy of the object was successfully retrieved.

Response Headers

- Content-Length – The length of the object contents in bytes.
- Content-Type – Internet media type associated with the content.
- ETag: “<hex string>” – MD5 checksum of the content.

Delete Fanout Object

This operation deletes all copies of the content associated with a fanout object.

To delete all copies of a fanout object, use the following HTTP DELETE request:

DELETE /rio/bucket/object HTTP/1.1

Request Headers

- Authorization – The type of authorization. Must be set to **basic** and followed by a base-64 encoding of **<username>:<password>**.
- x-fanout-delete-all: <boolean> – If true, the entire object, along with all associated content copies, will be deleted.

Example

```
DELETE /active/segmentid/2 HTTP/1.1
Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
X-Fanout-Delete-All: true

HTTP/1.1 201 No Content
```

Response Status

- 200 OK – The object was successfully deleted.
- 401 Unauthorized – The user name or password (or both) in the Authorization header is incorrect.
- 404 Not Found – Either the object did not exist, or the specified copy had been previously deleted.

Delete Copy Within Fanout Object

This operation deletes a single copy of the content associated with the specified fanout object. The operation supports two request formats, PATCH and DELETE.

PATCH Request Format

PATCH /rio/bucket/object HTTP/1.1

The body of the PATCH request contains a single line of text:

punch <copy_index>

where <copy_index> is the zero-based index of the copy of content to be deleted.

Request Headers

- Authorization – The type of authorization. Must be set to “basic” and followed by a base-64 encoding of “<username>:<password>.”
- Content-Type – “application/fanout” required for this request format.
- Content-Length – the length of the punch request body in bytes.

DELETE Request Format

DELETE /rio/bucket/object HTTP/1.1

Request Headers

- x-fanout-copy-index: integer – The zero-based index of the copy of content to be deleted.

Response Status

- 200 OK – The object was successfully deleted.
- 400 Bad Request – The copy index was outside of the acceptable range, 0 to 9,999.

- 401 Unauthorized – The user name or password (or both) in the Authorization header is incorrect.
- 404 Not Found – The copy identified by the index was previously deleted.

Bulk Delete Fanout Object

This operation deletes a fanout object or its fanout indices (copies), or both. Use of this API call allows for improved scaling efficiency when deleting large numbers of recordings.

Note This operation has a limit of 10,000 fanout objects (not including any object indices) per request.

To bulk-delete a fanout object, use the following HTTP POST request:

POST /rio/[prefix]/?delete HTTP/1.1

where **[prefix]** is an optional object prefix including trailing slash, if any.

Request Headers

- Authorization – The type of authorization. Must be set to basic and followed by a base-64 encoding of <username>:<password>.
- Content-Length – The length of the object contents in bytes.
- Content-Type – Internet media type associated with the content. Must be "text/plain", indicating the format of the bulk delete request body.

Request Body

Each line of the request body has the following format:

object_name[:list_of_indices]

with lines separated by the newline character. The optional prefix is combined with each object_name to form the name of a fanout or non-fanout object.

The object_name can be followed by an optional colon list of one or more zero-based copy indices. The indices in the list are separated by commas.

The indices list can also be specified as **all** to indicate that all indices should be punched (deleted). Each indicated copy of the specified object will be punched and the object will be deleted if no unpunched copies remain.

If no list of indices is specified, the object should be a non-fanout object, and the entire object will be deleted.

Example 1: No Common Prefix

```
POST /rio/?delete HTTP/1.1
Host: cos.cluster.example.com
Accept: */*
Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
Content-Type: text/plain
Content-Length: 125
Connection: Keep-Alive
sample1.txt:1,4,5
sample2.txt:1,2,6
HTTP/1.1 200 OK
Date: Fri, 02 Dec 2011 01:53:42 GMT
Server: COS/3.14
Content-Length: 251
Number Deleted:1
```

```
Number Not Found:0
Errors:
sample2.txt:2,6:416 Range
```

Example 2: Optimized Using Common Prefix

```
POST /rio/prefixToObjectIncludingTrailingSlashIfAny/?delete HTTP/1.1
Host: cos.cluster.example.com
Accept: */*
Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
Content-Type: text/plain
Content-Length: 125
Connection: Keep-Alive
sample1.txt:1,4,5
sample2.txt:1,2,6
sample3.txt
sample4.txt:all
HTTP/1.1 200 OK
Date: Fri, 02 Dec 2011 01:53:42 GMT
Server: COS/3.14
Content-Length: 251
Number Deleted:1
Number Not Found:1
Errors:
sample2.txt:2,6:416 Range
```

Response Status

- 200 OK – The request body was successfully parsed and the list of objects processed. The response body (described below) summarizes the results of processing each of the objects listed in the request body.
- 400 Bad Request – There was a syntax error in the request body. (No 404 is returned.)
- 401 Unauthorized – The user name or password (or both) in the Authorization header is incorrect.

Response Body

The response body is returned if the response status is 200 OK. It contains a summary of the disposition of each object listed in the request body.

The body of the listing response always includes the following parameters:

- Number Deleted:some_number – A count of objects listed in the request body where one or more indices were successfully punched, or where the entire object was deleted.
- Number Not Found:some_other_number – A count of objects listed in the request body where all of the indicated indices had previously been punched, or where the object had previously been deleted.

In addition, if any other error occurred during the processing of one or more objects, the response also includes the following:

- Errors: – For each object where an error (other than Not Found) occurred during processing of the object or its indicated indices, the object name is listed, followed by a colon, followed by the list of the indices indicated in the corresponding line of the request body, followed by another colon, and ending with an HTTP status code and status string.

  For example:

  ```
  HTTP/1.1 200 OK
  Date: Fri, 02 Dec 2011 01:53:42 GMT
  Server: COS/3.14
  Content-Length: 251
  Number Deleted:1
  Number Not Found:1
  Errors:
  sample2.txt:2,6:416 Range
  ```

Create Non-Fanout Object

This operation creates a non-fanout object for interaction with archive and recon storage buckets.

To create a non-fanout object, use the following HTTP PUT request:

PUT /rio/bucket/object HTTP/1.1

The body of the PUT request contains a single copy of the contents.

Request Headers

- Authorization – The type of authorization. Must be set to “basic” and followed by a base-64 encoding of “<username>:<password>.”
- Content-Length – The length of the object contents in bytes.
- Content-Type – Internet media type associated with the content. Defaults to “application/octet-stream” if not specified.
- x-fanout-copy-count: <integer> – The number of copies to be created. The number of copies must be between 1 and 10,000. If this header is not included, the copy-count defaults to 1.

Example

```
PUT /archive/segmentid/2 HTTP/1.1
Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
Content-Type: mpeg2/ts
Content-Length: 42
[42 bytes object data]

HTTP/1.1 201 No Content
```

Response Status

- 200 OK – The object was successfully created.
- 201 No content.
- 204 Created.
- 401 Unauthorized – The user name or password (or both) in the Authorization header is incorrect.
- 413 Request Entity Too Large – The content exceeds the maximum supported size, or the cluster is full.
- Other return codes are treated as errors.

Response Headers

- ETag: <hex string> – MD5 checksum of a single copy of the content.

Access Non-Fanout Object

This operation retrieves the content of a non-fanout object.

To retrieve a single copy of a fanout object, use the following HTTP GET request:

GET /rio/bucket/object HTTP/1.1

Example

```
GET /archive/segmentid/2 HTTP/1.1

HTTP/1.1 200 OK
Etag: “1b2cf535f27731c974343645a3985328”
```

```
Content-Length: 42
[42 bytes object data]
```

Request Headers

- x-fanout-copy-index: <integer> – The zero-based index of the content copy to be returned. The copy index must be between 0 and 9,999.

Response Status

- 200 OK – The object was successfully retrieved.
- Other return codes are treated as errors.

Response Headers

- Content-Length – The length of the object contents in bytes.
- ETag: <hex string> – MD5 checksum of the content.

Copy Object Content

This operation creates a non-fanout destination object that is a copy of the content associated with a fanout or non-fanout source object. This allows COS to make an internal copy of the object rather than require the object to transition from COS to VMR and then back to COS.

To copy content from a source object to a non-fanout destination object, use the following HTTP PUT request:

PUT /rio/bucket/object HTTP/1.1

The body of the PUT request is empty.

Request Headers

- Authorization – The type of authorization. Must be set to “basic” and followed by a base-64 encoding of “<username>:<password>.”
- x-amz-copy-source: bucket/object – The path to the source object, not including the "/rio/" prefix.
- x-fanout-copy-index: <integer> – The zero-based index of the content copy to be returned. The copy index must be between 0 and 9,999. (If the source object is a non-fanout object, this header may be omitted from the request.)

Example

```
PUT /rio/archive/segmentid/2 HTTP 1.1
Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
x-amz-copy-source: active/segmentid/2
x-fanout-copy-index: 10
HTTP/1.1 200 OK
Etag: “1b2cf535f27731c974343645a3985328”
```

Response Status

- 200 OK – The object was successfully copied.
- 401 Unauthorized – The user name or password (or both) in the Authorization header is incorrect.
- 413 Request Entity Too Large – The content exceeds the maximum supported size, or the cluster is full.
- Other return codes are treated as errors.

Response Headers

- ETag: “<hex string>” – MD5 checksum of a single copy of the content.

Delete Non-Fanout Object

This operation deletes the content associated with a non-fanout object.

To delete a non-fanout object, use the following HTTP DELETE request:

DELETE /rio/bucket/object HTTP/1.1

Example

```
DELETE /archive/segmentid/2 HTTP/1.1
Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==

HTTP/1.1 201 No Content
```

Request Headers

- Authorization – The type of authorization. Must be set to **basic** and followed by a base-64 encoding of **<username>:<password>**.
- x-fanout-delete-all: <boolean> – If true, the entire object, along with all associated content copies, will be deleted.

Response Status

- 200 OK – The object was successfully deleted.
- 201 No Content.
- 401 Unauthorized – The user name or password (or both) in the Authorization header is incorrect.
- 404 Not Found – Either the object did not exist, or the specified object had been previously deleted.
- Other return codes are treated as errors.

Example API Calls

This appendix provides some examples performing a Service Manager, Swauth, and Swift API call using curl.

Service Manager API curl Example

To retrieve a list of existing IP Pools:

```
#curl -v -L -k -X GET https://SM.acme.com:8043/v2/ippools
* About to connect() to SM.acme.com port 8043 (#0)
*   Trying 10.1.1.1... connected
* Connected to SM.acme.com (10.1.1.1) port 8043 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
* warning: ignoring value of ssl.verifyhost
* skipping SSL peer certificate verification
* NSS: client certificate not found (nickname not specified)
* SSL connection using TLS_RSA_WITH_AES_256_CBC_SHA
* Server certificate:
*       subject: E=support@cisco.com,CN=SM,OU=SPVTG,O=CISCO
SYSTEMS,L=MILPITAS,ST=CALIFORNIA,C=US
*       start date: Apr 23 18:39:33 2014 GMT
*       expire date: Apr 20 18:39:33 2024 GMT
*       common name: SM
*       issuer: E=support@cisco.com,CN=SM,OU=SPVTG,O=CISCO
SYSTEMS,L=MILPITAS,ST=CALIFORNIA,C=US
> GET /v2/ippools HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.13.6.0 zlib/1.2.3
libidn/1.18 libssh2/1.4.2
> Host: SM.acme.com:8043
> Accept: */*
>
< HTTP/1.1 200 OK
< x-powered-by: Express
< Content-Type: application/json; Authorization: bearer auth_token
< content-length: 2138
< etag: "1249005776"
< date: Fri, 19 Dec 2014 20:57:03 GMT
< connection: close
<
[
  {
    "id": "smtenant_system.smippool.ippool-1",
    "name": "ippool-1",
    "type": "ippools",
    "externalId": "/v2/ippools/ippool-1",
```

```
    "properties": {
      "description": "A sample ip pool for COS cache interfaces",
      "addrType": "ipv4",
      "networkRef": "smtenant_system.smnetwork.network-a",
      "pool": [
        {
          "rangeStart": "0.0.0.0",
          "rangeEnd": "0.0.0.0",
          "netmask": "255.255.255.0",
          "gw": "0.0.0.0"
        }
      ]
    }
  },
  {
    "id": "smtenant_system.smippool.2",
    "name": "2",
    "type": "ippools",
    "externalId": "/v2/ippools/2",
    "transactionId": "0dd177a1-4e25-4e51-8563-3d65de952baa",
    "properties": {
      "description": "COS NPI Pool",
      "addrType": "ipv4",
      "networkRef": "smtenant_system.smnetwork.network-a",
      "pool": [
        {
          "rangeStart": "10.93.232.153",
          "rangeEnd": "10.93.232.153",
          "netmask": "255.255.255.224",
          "gw": "10.93.232.129"
        }
      ]
    }
  },
  {
    "id": "smtenant_system.smippool.pool-3",
    "name": "pool-3",
    "type": "ippools",
    "externalId": "/v2/ippools/pool-3",
    "transactionId": "dc26d15b-37bd-4968-9be4-cbca0b2f0deb",
    "properties": {
      "description": "COS node Pool 3",
      "addrType": "ipv4",
      "networkRef": "smtenant_system.smnetwork.network-a",
      "pool": [
        {
          "rangeStart": "10.93.232.155",
          "rangeEnd": "10.93.232.155",
          "netmask": "255.255.255.224",
          "gw": "10.93.232.129"
        }
      ]
    }
  },
  {
    "id": "smtenant_system.smippool.cos-npi",
    "name": "cos-npi",
    "type": "ippools",
    "externalId": "/v2/ippools/cos-npi",
    "transactionId": "2c82ad23-03d2-48e5-b216-1ce6431cddac",
    "properties": {
      "description": "cos-npi data interface pool",
      "addrType": "ipv4",
      "networkRef": "smtenant_system.smnetwork.network-a",
```

```
      "pool": [
        {
          "rangeStart": "10.93.232.154",
          "rangeEnd": "10.93.232.154",
          "netmask": "255.255.255.224",
          "gw": "10.93.232.129"
        }
      ]
    }
  }
* Closing connection #0
```

Note For the auth_token value, see the /etc/opt/cisco/mos/public/token.json file on any V2PC master node.

Swauth API curl Example

To see a list of accounts:

```
curl -v -L -X GET -H "X-Auth-Admin-User:gsmith:user1" -H "X-Auth-Admin-Key:123XYZ"
http://auth01.cos1.acme.com/auth/v2
* About to connect() to auth01.cos1.acme.com port 80 (#0)
*   Trying 192.168.1.1... connected
* Connected to auth01.cos1.acme.com (192.168.1.1) port 80 (#0)
> GET /auth/v2 HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0 zlib/1.2.3
libidn/1.18 libssh2/1.4.2
> Host: auth01.cos1.acme.com
> Accept: */*
> X-Auth-Admin-User:gsmith:user1
> X-Auth-Admin-Key:123XYZ
>
< HTTP/1.1 200 OK
< Server: Cisco/Object Store/0.2
< Connection: Keep-Alive
< Date: Fri, 19 Dec 2014 22:18:37 GMT
< Content-Type: application/json; Authorization: bearer auth_token
< Content-Length: 206
<
{"accounts": [
 {"name": "abrown"},
 {"name": "cjones"},
 {"name": "cjones"},
 {"name": "gsmith"},
 {"name": "kurt"},
 {"name": "matt"},
 {"name": "michele"},
* Connection #0 to host auth01.cos1.acme.com left intact
* Closing connection #0
```

Note For the auth_token value, see the /etc/opt/cisco/mos/public/token.json file on any V2PC master node.

Swift API curl Example

To create a container:

```
curl -v -L -X PUT -H "X-Auth-Token: AUTH_tk836935baa053405aa65853863b17b871" -H
"X-Container-Read:*" "X-Container-Write:*"
http://auth01.cos1.acme.com/v1/AUTH_msmith/mustang
* About to connect() to auth01.cos1.acme.com port 80 (#0)
*   Trying 192.168.1.1... connected
* Connected to auth01.cos1.acme.com (192.168.1.1) port 80 (#0)
> PUT /v1/AUTH_msith/mustang HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0 zlib/1.2.3
libidn/1.18 libssh2/1.4.2
> Host: auth01.cos1.acme.com
> Accept: */*
> X-Auth-Token: AUTH_tka5d2c2898c1e4632854f55cadfcc55f5
>
< HTTP/1.1 201 Created
< Server: Cisco/Object Store/0.2
< Connection: Keep-Alive
< Date: Fri, 12 Dec 2014 01:54:15 GMT
< Content-Length: 0
<
* Connection #0 to host auth01.cos1.acme.com left intact
* Closing connection #0
```

To confirm the creation of the container:

```
curl -v -L -X GET -H "X-Auth-Token: AUTH_tk836935baa053405aa65853863b17b871"
http://auth01.cos1.acme.com/v1/AUTH_msmith
* About to connect() to auth01.cos1.acme.com port 80 (#0)
*   Trying 192.168.1.1... connected
* Connected to auth01.cos1.acme.com (192.168.1.1) port 80 (#0)
> GET /v1/AUTH_msmith HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0 zlib/1.2.3
libidn/1.18 libssh2/1.4.2
> Host: auth01.cos1.acme.com
> Accept: */*
> X-Auth-Token: AUTH_tk836935baa053405aa65853863b17b871
>
< HTTP/1.1 200 OK
< Server: Cisco/Object Store/0.2
< Connection: Keep-Alive
< Date: Fri, 19 Dec 2014 22:26:58 GMT
< Content-Type: text/plain; charset=utf-8
< Content-Length: 38
< X-Account-Container-Count: 3
< X-Account-Object-Count: 0
< X-Account-Bytes-Used: 0
<
container1
mustang
* Connection #0 to host auth01.cos1.acme.com left intact
* Closing connection #0
```