Cisco Cloud Object Storage Release 3.0.1
API Guide

August 12, 2015

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.
CONyENTS

Preface vii
Audience vii
Document Organization vii
Document Conventions viii
Related Publications ix
Obtaining Documentation and Submitting a Service Request ix

CHAPTER 1
Introduction to COS and COS APIs 1-1
Product Description 1-1
COS and MOS 1-1
Components 1-1
API Features 1-2
Overview 1-2
Service Manager API 1-3
Swauth API 1-3
Swift Object Store API 1-4
Restrictions and Limitations 1-5

CHAPTER 2
Service Manager API 2-1
Listing the Region 2-1
Listing DNS Servers 2-1
Listing All DNS Servers 2-1
Listing One DNS Server 2-2
Listing and Updating NTP Servers 2-2
Listing All NTP Servers 2-2
Listing One NTP Server 2-2
Updating an NTP Server 2-3
Listing DNS Forwarders 2-3
Listing DNS Forwarders 2-3
Listing One DNS Forwarder 2-4
Listing, Creating, Deleting, and Updating Networks 2-5
Listing the Networks 2-5
Contents

Cisco Cloud Object Storage Release 3.0.1 API Guide

CHAPTER 3
Swauth API 3-1
Listing Accounts 3-1
Retrieving Account Details 3-2
Creating an Account 3-3
Deleting an Account 3-4
Creating or Updating a User 3-4
Retrieving User Details 3-5
Deleting a User 3-6
Creating or Updating Account Service Endpoints 3-7
Getting an Authentication Token 3-8

CHAPTER 4
Swift API 4-1
Listing Containers 4-1
Listing Objects 4-2
Creating a Container 4-3
Deleting a Container 4-4
Retrieving an Object 4-4
Creating or Updating An Object 4-7
Deleting an Object 4-8
Creating or Updating Container Metadata 4-9
Retrieving Container Metadata 4-10
Deleting Container Metadata 4-11
Creating or Updating Object Metadata 4-11
Retrieving Object Metadata 4-12
Deleting Object Metadata 4-13
Creating or Updating Account Metadata 4-14
Retrieving Account Metadata 4-14
Deleting Account Metadata 4-15
Access Control Lists (ACLs) 4-16
Creating or Updating ACLs 4-16
Deleting ACLs 4-17

APPENDIX A

Example API Calls A-1
Service Manager API curl Example A-1
Swauth API curl Example A-3
Swift API curl Example A-3
Preface

This preface describes who should read the *Cisco Cloud Object Storage Release 3.0.1 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:

<table>
<thead>
<tr>
<th>Chapters or Appendices</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 1, “Introduction to COS and COS APIs”</td>
<td>Introduces the COS software APIs</td>
</tr>
<tr>
<td>Chapter 2, “Service Manager API”</td>
<td>Describes the subset of the MOS APIs that are implemented for the COS.</td>
</tr>
<tr>
<td>Chapter 3, “Swauth API”</td>
<td>Describes the subset of the OpenStack Swauth API that is implemented for the COS authentication service.</td>
</tr>
<tr>
<td>Chapter 4, “Swift API”</td>
<td>Describes the subset of the OpenStack Swift API that is implemented for COS.</td>
</tr>
<tr>
<td>Appendix A, “Example API Calls”</td>
<td>Provides examples for making Service Manager, Swauth, and Swift API calls using curl.</td>
</tr>
</tbody>
</table>
Document Conventions

This document uses the following conventions:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Indication</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>bold</strong> font</td>
<td>Commands and keywords and user-entered text appear in <strong>bold</strong> font.</td>
</tr>
<tr>
<td><em>italic</em> font</td>
<td>Document titles, new or emphasized terms, and arguments for which you supply values are in <em>italic</em> font.</td>
</tr>
<tr>
<td>[ ]</td>
<td>Elements in square brackets are optional.</td>
</tr>
<tr>
<td>{x</td>
<td>y</td>
</tr>
<tr>
<td>[x</td>
<td>y</td>
</tr>
<tr>
<td>string</td>
<td>A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.</td>
</tr>
<tr>
<td><strong>courier</strong> font</td>
<td>Terminal sessions and information the system displays appear in <strong>courier</strong> font.</td>
</tr>
<tr>
<td>&lt; &gt;</td>
<td>Nonprinting characters such as passwords are in angle brackets.</td>
</tr>
<tr>
<td>[ ]</td>
<td>Default responses to system prompts are in square brackets.</td>
</tr>
<tr>
<td>!, #</td>
<td>An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.</td>
</tr>
</tbody>
</table>

---

**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 3.0.1:

- Release Notes for Cisco Cloud Object Storage Release 3.0.1
- Cisco Content Delivery Engine 205/220/250/280/420/460/470 Hardware Installation Guide
- Cisco Cloud Object Storage Release 3.0.1 User Guide
- Cisco Media Origination System Release 2.4 User Guide
- Open Source Used in Cisco Cloud Object Storage Release 3.0.1

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.
Introduction to COS and COS APIs

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.

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.

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.

Components

COS has a number of subsystems:

- **Networks:** Interfaces are grouped into distinct networks to isolate management functions from high-volume data traffic. The client applications use Swauth API to interact with the COS authentication and authorization services, and the Swift API to interact with the COS object storage services.

- **Clusters and Nodes:** COS services are provided by a cluster of nodes, with both the cluster and the individual nodes as distinctly manageable components.

- **Object Metadata Store:** The metadata for the cluster is stored in a high-performance distributed NoSQL database hosted on the COS nodes in a cluster.
• **Platform and Applications Manager (PAM):** COS components are managed using services running on the Platform and Applications Management system (PAM).

• **Hardware Platforms:** COS software is deployed on Cisco Content Delivery Engine (CDE) platforms, and will also be available on Cisco UCS Dense Hardware.

## API Features

- **Overview, page 1-2**
- **Service Manager API, page 1-3**
- **Swauth API, page 1-3**
- **Swift Object Store API, page 1-4**

## Overview

The table below provides an overview of the COS APIs

<table>
<thead>
<tr>
<th>Feature Set</th>
<th>Features</th>
</tr>
</thead>
</table>
| Service Manager API          | • A subset of the Cisco Media Origination System (MOS) APIs
|                              | • Used to provision and configure a COS cluster and COS cluster nodes   |
|                              | • Uses the FQDN of the Service Manager and HTTPS over port 8043         |
| Swauth API                   | • 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        |
Chapter 1      Introduction to COS and COS APIs

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

**Service Manager API**

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

- Listing the region
- Listing, creating, deleting, and modifying zones
- Listing, creating, deleting, and modifying IP pools
- Listing, creating, deleting, and modifying a COS cluster
- Adding a COS node to a COS cluster
- Listing, editing, and enabling a COS instance
- Listing, creating, deleting, and modifying a COS service endpoint
- Listing, creating, deleting, and modifying an asset workflow template
- Listing and modifying COS nodes
- Configuring a COS node service interface
- Viewing the COS node status
- Viewing the COS node management events

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

**Swauth API**

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

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

For a detailed description of these functions, see Swauth API, page 3-1.
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>
<thead>
<tr>
<th>Constraint</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum # of HTTP Headers per request</td>
<td>90</td>
</tr>
<tr>
<td>Maximum length of all HTTP Headers</td>
<td>4096 bytes</td>
</tr>
<tr>
<td>Maximum length per HTTP request line</td>
<td>8192 bytes</td>
</tr>
<tr>
<td>Maximum length of container name</td>
<td>256 bytes</td>
</tr>
<tr>
<td>Maximum length of object name</td>
<td>1024 bytes</td>
</tr>
</tbody>
</table>

Also, the container and object names must be UTF-8 encoded and then URL-encoded before inclusion in the HTTP 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.
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 1-4 and Swauth API, page 1-3.

- 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 COS 3.0.1 Release Notes for open caveats and known issues related to this release of the COS software.
Service Manager API

This chapter describes the subset of the Cisco MOS API that is implemented for COS.

Note
The Service Manager API uses the FQDN of the Service Manager and HTTPS over port 8043.

Listing the Region

This command returns a list of all Region objects.

Note
COS 3.0.1 supports only one Region object. The object is named region-0, and it is read-only.

Syntax
GET https://SM_FQDN:8043/v2/regions

Example
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
https://172.22.98.59:8043/v2/regions/
{
  "id": "smtenant_system.smregion.region-0",
  "name": "region-0",
  "type": "regions",
  "externalId": "/v2/regions/region-0",
  "properties": {
    "description": "Default Region",
    "controllersRef": []
  }
}

Listing DNS Servers

The DNSServer object defines the Name server for COS. This is used to insert the COS A Records and for load balancing components within COS. Depending on the choice made at installation, the DNSServer can be supported on the COS PAM or provided by an external DNS server.
When PAM acts as the DNNServer, all PAM nodes (high availability) act as redundant DNS servers. This object is set up during installation and cannot be changed later.

**Listing all DNS Servers**

This command returns a list of all configured external DNS Server objects.

**Syntax**

GET https://SM_FQDN:8043/v2/regions/region_name/dnsservers

**Example**

HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

```
{
    "id": "smregion_0.smdnsserver.extdns-1",
    "name": "extdns-1",
    "type": "dnsservers",
    "externalId": "/v2/regions/region-0/dnsservers/extdns-1",
    "transactionId": "1910ca59-0a88-411c-8f70-22cf6c107f0e",
    "properties": {
        "description": "dns server",
        "ipAddr": "172.22.98.170",
        "domain": "hapam1.cisco.com",
        "authType": "tsig",
        "tsigKey": "zKTjdCmwT2snAG8tJ1eLVA==",
        "tsigAlgorithm": "hmac-md5"
    }
}
```

**Listing One DNS Server**

This command returns information about the specified DNS Server object.

**Syntax**

GET https://SM_FQDN:8043/v2/regions/region_name/dnsservers/dnsservername

**Listing and Updating NTP Servers**

**Note**

There is one NTP Server object per region. Each object may have one or more NTP server instances defined.

**Listing all NTP Servers**

This command returns a list of all NTP Server objects.
Syntax

GET https://SM_FQDN:8043/v2/regions/region_name/ntpservers

Example

HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

https://172.22.98.69:8043/v2/regions/region-0/ntpservers/

[  
  {  
    "id": "smregion_0.smntpserver.ntp-1",  
    "name": "ntp-1",  
    "type": "ntpservers",  
    "externalId": "/v2/regions/region-0/ntpservers/ntp-1",  
    "transactionId": "d4e6c778-7b4f-49f6-a37e-e389440c5f3e",  
    "properties": {  
      "description": "ntp server list",  
      "servers": [  
        "10.50.171.9"  
      ]  
    }  
  }  
]

Listing one NTP Server

This command returns information about the specified NTP Server object.

Syntax

GET https://SM_FQDN:8043/v2/regions/region_name/ntpservers/ntpservename

Example

https://172.22.98.69:8043/v2/regions/region-0/ntpservers/ntp-1

{  
  "id": "smregion_0.smntpserver.ntp-1",  
  "name": "ntp-1",  
  "type": "ntpservers",  
  "externalId": "/v2/regions/region-0/ntpservers/ntp-1",  
  "transactionId": "d4e6c778-7b4f-49f6-a37e-e389440c5f3e",  
  "properties": {  
    "description": "ntp server list",  
    "servers": [  
      "10.50.171.9"  
    ]  
  }  
}

Updating an NTP Server

This command updates the JSON configuration file for the specified NTP Server object.

Syntax

PUT https://SM_FQDN:8043/v2/regions/region_name/ntpservers/ntpservename
Example

- Request curl:

  curl -k -H "Content-Type: application/json" -X PUT --data @body.json
  https://10.10.150.22:8043/v2/regions/region-0/ntpservers/ntp-1

- @body.json file format:

  

```json
{
  
  "properties": {
  "description": "ntp server list",
  "servers": ["10.22.171.18"]
  }
}
```

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Required</th>
<th>Pattern</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>servers</td>
<td>Array of strings</td>
<td>Yes</td>
<td></td>
<td>Comma separated list of the IP addresses or hostnames of the NTP servers</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>No</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Listing DNS Forwarders

The DNS Forwarder is an optional object that defines the query server (if any) used by COS components to resolve DNS names. If used, this object is set up during PAM installation, and can be updated later. This command returns a list of all DNS Forwarder objects.

Syntax

GET https://SM_FQDN:8043/v2/regions/region_name/dnsforwarders

Example

HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

```
{
  "id": "smregion_0.smdnsforwarder.dnsFwd-1",
  "name": "dnsFwd-1",
  "type": "dnsforwarders",
  "externalId": "/v2/regions/region-0/dnsforwarders/dnsFwd-1",
  "transactionId": "bcff4a4c-00ac-4519-8beb-b3d7b5c7b3e4",
  "properties": {
    "description": "dns forwarder",
    "ipAddr": "10.93.232.70",
    "domain": "mos.npi.cds.cisco.com"
  }
}
```

Listing One DNS Forwarder

This command returns information about the specified DNS Forwarder object.

Syntax

GET https://SM_FQDN:8043/v2/regions/region_name/dnsforwarders/dnsforwardername
Listing, Creating, Deleting, and Updating Networks

Before creating IP pools, you must configure networks.

Listing the Networks

This command returns a list of all Network objects.

Syntax
GET https://SM_FQDN:8043/v2/networks

Example
https://172.22.98.69:8043/v2/networks/
[
  {
    "id": "smtenant_system.smnetwork.network-a",
    "name": "network-a",
    "type": "networks",
    "externalId": "/v2/networks/network-a",
    "properties": {
      "netName": "red",
      "zoneRef": "smtenant_system.smzone.zone-1",
      "fqdn": "internal.mos.com"
    }
  }
]

Listing One Network

This command returns information about the specified Network object.

Syntax
GET https://SM_FQDN:8043/v2/networks/network_name

Example
https://172.22.98.69:8043/v2/networks/network-a
{
  "id": "smtenant_system.smnetwork.Network-a",
  "name": "Network-a",
  "type": "networks",
  "externalId": "/v2/networks/Network-a",
  "transactionId": "af8c4a67-e1c2-42ea-a58d-4e1e81327d29",
  "properties": {
    "zoneRef": "smtenant_system.smzone.zone-1",
    "netName": "network",
    "fqdn": "cos5.cisco.com"
  }
}

Creating a Network

This command creates and configures a new Network object.
Listing, Creating, Deleting, and Updating Networks

Syntax
POST https://SM_FQDN:8043/v2/networks/network_name

Example
https://172.22.98.59:8043/v2/networks/Network-b
{
   "id": "smtenant_system.smnetwork.Network-b",
   "name": "Network-b",
   "type": "networks",
   "externalId": "/v2/networks/Network-b",
   "transactionId": "f3a8307f-e441-450c-b781-4fa56f12533",
   "properties": {
      "fqdn": "cos5.cisco.com",
      "netName": "newnetwork",
      "zoneRef": "smtenant_system.smzone.zone-1"
   }
}

- Request curl:
  curl -k -H "Content-Type: application/json" -X POST --data @body.json
  https://10.10.150.22:8043/v2/networks/Network-b

- @body.json file format:
  
  
  

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Required</th>
<th>Pattern</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>netName</td>
<td>String</td>
<td>Yes</td>
<td>If a name is reused in a Zone, the ipPoolRef must be identical.</td>
<td></td>
</tr>
<tr>
<td>zoneRef</td>
<td>String</td>
<td>Yes</td>
<td>Zone to which the Network belongs</td>
<td></td>
</tr>
<tr>
<td>fqdn</td>
<td>String</td>
<td>No</td>
<td>For example: mosint.acme.com</td>
<td></td>
</tr>
</tbody>
</table>

Deleting a Network

This command deletes the specified existing Network object.

Syntax
DELETE https://SM_FQDN:8043/v2/networks/network_name
Updating a Network

This command updates the JSON configuration file for the specified Network object.

Syntax
PUT https://SM_FQDN:8043/v2/networks/network_name

Example
https://172.22.98.59:8043/v2/networks/Network-a

- Request curl:
curl -k -H "Content-Type: application/json" -X PUT --data @body.json

- @body.json file format:

```json
{
  "properties": {
    "netName": "south",
    "zoneRef": "smtenant_system.smzone.zone-1"
  }
}
```

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Required</th>
<th>Pattern</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>netName</td>
<td>String</td>
<td>Yes</td>
<td></td>
<td>If a name is reused in a Zone, the ipPoolRef must be identical.</td>
</tr>
<tr>
<td>zoneRef</td>
<td>String</td>
<td>Yes</td>
<td>.<em>..</em>..*</td>
<td>Zone to which the network belongs</td>
</tr>
<tr>
<td>fqdn</td>
<td>String</td>
<td>No</td>
<td>^[a-z0-9]+([-._][a-z0-9]+)*[^.[2,6][^]$]</td>
<td>For example: mosint.acme.com</td>
</tr>
</tbody>
</table>

Listing, Creating, Deleting, and Updating IP Pools

Before installing COS nodes, you must configure IP pools.

Listing the IP Pools

This command returns a list of all IP Pool objects.
Syntax
GET https://SM_FQDN:8043/v2/ippools

Example
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

https://172.22.98.69:8043/v2/ippools
{
  "id": "smtenant_system.smippool.ippool-1",
  "name": "ippool-1",
  "type": "ippools",
  "externalId": "/v2/ippools/ippool-1",
  "transactionId": "12d4a46e-2778-43ac-acb4-3f8fd23404e5",
  "properties": {
    "description": "A sample ip pool for COS cache interfaces",
    "addrType": "ipv4",
    "networkRef": "smtenant_system.smnetwork.network-a",
    "pool": [
      {
        "rangeStart": "192.169.118.1",
        "rangeEnd": "192.169.118.5",
        "netmask": "255.255.255.0",
        "gw": "192.169.118.254"
      }
    ]
  }
}

Listing One IP Pool

This command returns information about the specified IP Pool object.

Syntax
GET https://SM_FQDN:8043/v2/ippools/IPPoolName

Example
https://172.22.98.69:8043/v2/ippools/ippool-1
{
  "id": "smtenant_system.smippool.ippool-1",
  "name": "ippool-1",
  "type": "ippools",
  "externalId": "/v2/ippools/ippool-1",
  "transactionId": "12d4a46e-2778-43ac-acb4-3f8fd23404e5",
  "properties": {
    "description": "A sample ip pool for COS cache interfaces",
    "addrType": "ipv4",
    "networkRef": "smtenant_system.smnetwork.network-a",
    "pool": [
      {
        "rangeStart": "192.169.118.1",
        "rangeEnd": "192.169.118.5",
        "netmask": "255.255.255.0",
        "gw": "192.169.118.254"
      }
    ]
  }
}
Creating an IP Pool

This command creates and configures a new IP Pool object.

**Syntax**

POST https://SM_FQDN:8043/v2/ippools/IPPoolName

**Example**

https://172.22.98.59:8043/v2/ippools/IPPool-3

```
{
    "id": "smtenant_system.smippool.IPPool-3",
    "name": "IPPool-3",
    "type": "ippools",
    "externalId": "/v2/ippools/IPPool-3",
    "transactionId": "00b1a4b1-938e-488a-89ea-a7363882e544",
    "properties": {
        "pool": [
            {
                "gw": "192.169.218.30",
                "netmask": "255.255.255.224",
                "rangeEnd": "192.169.218.9",
                "rangeStart": "192.169.218.7"
            }
        ],
        "networkRef": "smtenant_system.smnetwork.Network-a",
        "addrType": "ipv4",
        "description": "third ip pool"
    }
}
```

- **Request curl:**

  ```
curl -k -H "Content-Type: application/json" -X POST --data @body.json
  https://10.150.22.8043/v2/ippools/ippool-2
  ```

- **@body.json file format:**

  ```
  {
    "properties": {
        "addrType": "ipv4",
        "networkRef": "smtenant_system.smnetwork.network-b",
        "pool": [
            {
                "rangeStart": "10.22.22.1",
                "rangeEnd": "10.22.22.25",
                "netmask": "255.255.255.224",
                "gw": "10.22.22.30"
            }
        ]
    }
  }
  ```
### Deleting an IP Pool

This command deletes the specified existing IP Pool object.

**Syntax**

DELETE https://SM_FQDN:8043/v2/ippools/IPPoolName

**Example**

DELETE https://172.22.98.59:8043/v2/ippools/IPPool-3

### Updating an IP Pool

This command updates the JSON configuration file for the specified IP Pool object.

**Syntax**

PUT https://SM_FQDN:8043/v2/ippools/IPPoolName

**Example**

PUT https://172.22.98.59:8043/v2/ippools/IPPool-3

- **Request curl:**
  
  ```
  curl -k -H "Content-Type: application/json" -X PUT --data @body.json https://10.10.150.22:8043/v2/ippools/ippool-2
  ```

- **@body.json file format:**

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Required</th>
<th>Pattern/Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>description</td>
<td>String</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>addrType</td>
<td>String</td>
<td>Yes</td>
<td>“ipv4”</td>
<td>Currently, the only valid choice is ipv4.</td>
</tr>
<tr>
<td>networkRef</td>
<td>String</td>
<td>Yes</td>
<td>.<em>.smnetwork..</em></td>
<td>Network to which the IP Pool belongs</td>
</tr>
<tr>
<td>pool</td>
<td>Array</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**pool Item Properties**

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Required</th>
<th>Pattern/Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rangeStart</td>
<td>String</td>
<td>Yes</td>
<td></td>
<td>Starting IP address in the pool</td>
</tr>
<tr>
<td>rangeEnd</td>
<td>String</td>
<td>Yes</td>
<td></td>
<td>Ending IP address in the pool</td>
</tr>
<tr>
<td>netmask</td>
<td>String</td>
<td>Yes</td>
<td></td>
<td>Netmask or prefix for the IP address</td>
</tr>
<tr>
<td>gw</td>
<td>String</td>
<td>Yes</td>
<td></td>
<td>Default gateway or router for the pool</td>
</tr>
</tbody>
</table>
{  
  "properties": {  
    "addrType": "ipv4",  
    "networkRef": "smtenant_system.smnetwork.network-b",  
    "pool": [  
      {  
        "rangeStart": "10.22.22.1",  
        "rangeEnd": "10.22.22.25",  
        "netmask": "255.255.255.224",  
        "gw": "10.22.22.30"  
      }  
    ]  
  }  
}  

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Required</th>
<th>Pattern/Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>description</td>
<td>String</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>addrType</td>
<td>String</td>
<td>Yes</td>
<td>“ipv4”</td>
<td>Currently the only valid choice is ipv4.</td>
</tr>
<tr>
<td>networkRef</td>
<td>String</td>
<td>Yes</td>
<td>.<em>.smnetwork..</em></td>
<td>Network to which the IP Pool belongs</td>
</tr>
<tr>
<td>pool</td>
<td>Array</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**pool Item Properties**

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Required</th>
<th>Pattern/Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rangeStart</td>
<td>String</td>
<td>Yes</td>
<td></td>
<td>Starting IP address in the pool</td>
</tr>
<tr>
<td>rangeEnd</td>
<td>String</td>
<td>Yes</td>
<td></td>
<td>Ending IP address in the pool</td>
</tr>
<tr>
<td>netmask</td>
<td>String</td>
<td>Yes</td>
<td></td>
<td>Netmask or prefix for the IP address</td>
</tr>
<tr>
<td>gw</td>
<td>String</td>
<td>Yes</td>
<td></td>
<td>Default gateway or router for the pool</td>
</tr>
</tbody>
</table>

### Listing, Creating, Deleting, and Updating COS Node Clusters

#### Listing COS Node Clusters

This command returns a list of all COS Node Cluster objects.

**Syntax**

```plaintext
GET https://SM_FQDN:8043/v2/regions/region_name/cosnodeclusters
```

**Example**

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```
Listing One COS Node Cluster

This command returns information about the specified COS Node Cluster object.

**Syntax**

```
GET https://SM_FQDN:8043/v2/regions/region_name/cosnodeclusters/cluster_name
```

**Example**

```
https://172.22.98.59:8043/v2/regions/region-0/cosnodeclusters/Cluster-2
{
  "id": "smregion_0.smcosnodecluster.Cluster-2",
  "name": "Cluster-2",
  "type": "cosnodeclusters",
  "externalId": "/v2/regions/region-0/cosnodeclusters/Cluster-2",
  "transactionId": "b1b36b8a-e0ca-4ec8-977f-1d9006f92652",
  "properties": {
    "description": "second cluster",
    "regionRef": "smtenant_system.smregion.region-0",
    "authFqdn": "auth02.cos5.cisco.com",
    "storageFqdn": "auth02.cos5.cisco.com"
  }
}
```

Creating a COS Node Cluster

This command creates and configures a new COS Node Cluster object.
Listing, Creating, Deleting, and Updating COS Node Clusters

Syntax

POST https://SM_FQDN:8043/regions/region_name/cosnodeclusters/cluster_name

Example

https://172.22.98.59:8043/v2/regions/region-0/cosnodeclusters/Cluster-3
{
  "id": "smregion_0.smcosnodecluster.Cluster-3",
  "name": "Cluster-3",
  "type": "cosnodeclusters",
  "externalId": "/v2/regions/region-0/cosnodeclusters/Cluster-3",
  "transactionId": "148d6bec-ce12-42f1-adca-401d458f06f5",
  "properties": {
    "storageFqdn": "",
    "authFqdn": "",
    "regionRef": "smtenant_system.smregion.region-0",
    "description": "second cluster"
  }
}

Request curl:

curl -k -H "Content-Type: application/json" -X POST --data @body.json
https://10.10.150.22:8043/v2/regions/region-0/cosnodeclusters/cluster.az.01

@body.json file format:

{
  "properties": {
    "description": "A cos node cluster in arizona",
    "authFqdn": "auth01.cos1.acme.com",
    "storageFqdn": "auth01.cos1.acme.com"
  }
}

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Required</th>
<th>Pattern/Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>description</td>
<td>String</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>authFqdn</td>
<td>String</td>
<td>No</td>
<td>^[a-z0-9]+([:-]\.[a-z0-9]+]+)*[a-z]2,6$</td>
<td>^$</td>
</tr>
<tr>
<td>storageFqdn</td>
<td>String</td>
<td>No</td>
<td>^[a-z0-9]+([:-]\.[a-z0-9]+]+)*[a-z]2,6$</td>
<td>^$</td>
</tr>
</tbody>
</table>

Deleting a COS Node Cluster

This command deletes the specified existing COS Node Cluster object.
Listing, Creating, Deleting, and Updating COS Node Clusters

Syntax
DELETE https://SM_FQDN:8043/v2/regions/region_name/cosnodeclusters/cluster_name

Example
https://172.22.98.59:8043/v2/regions/region-0/cosnodeclusters/Cluster-3

Updating a COS Node Cluster

This command updates the JSON configuration file for the specified COS Node Cluster object.

Syntax
PUT https://SM_FQDN:8043/v2/regions/region_name/cosnodeclusters/cluster_name

Example
https://172.22.98.59:8043/v2/regions/region-0/cosnodeclusters/Cluster-3

- Request curl:
  curl -k -H "Content-Type: application/json" -X PUT --data @body.json
  https://10.10.150.22:8043/v2/regions/region-0/cosnodeclusters/cluster.az.01

- @body.json file format:

  ```json
  {
    "properties": {
      "description": "A cos node cluster in arizona",
      "authFqdn": "auth01.cos1.acme.com",
      "storageFqdn": "auth01.cos1.acme.com"
    }
  }
  ```

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Required</th>
<th>Pattern/Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>description</td>
<td>String</td>
<td>No</td>
<td>^[a-z0-9]+([--.]{1}[a-z0-9]+)*.{2,6}$</td>
<td>^$</td>
</tr>
<tr>
<td>authFqdn</td>
<td>String</td>
<td>No</td>
<td>^[a-z0-9]+([--.]{1}[a-z0-9]+)*.{2,6}$</td>
<td>^$</td>
</tr>
<tr>
<td>storageFqdn</td>
<td>String</td>
<td>No</td>
<td>^[a-z0-9]+([--.]{1}[a-z0-9]+)*.{2,6}$</td>
<td>^$</td>
</tr>
</tbody>
</table>
Listing, Creating, Deleting, and Updating COS Nodes

Listing the COS Nodes

This command returns a list of all COS Node objects.

Syntax
GET https://SM_FQDN:8043/v2/cosnodes

Example
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

https://172.22.98.59:8043/v2/cosnodes/
[
  {
    "id": "smtenant_system.smcosnode.2887148484",
    "name": "2887148484",
    "type": "cosnodes",
    "externalId": "/v2/cosnodes/2887148484",
    "transactionId": "e83865ad-4ecd-4eb0-98b5-171b73f20adc",
    "properties": {
      "dataInterfaces": [
        {
          "name": "eth2",
          "ipPoolRef": "",
          "enabled": false
        },
        {
          "name": "eth3",
          "ipPoolRef": "",
          "enabled": false
        },
        {
          "name": "eth4",
          "ipPoolRef": "",
          "enabled": false
        },
        {
          "name": "eth5",
          "ipPoolRef": "smtenant_system.mipool.IPPool-1",
          "enabled": true
        },
        {
          "name": "eth6",
          "ipPoolRef": "",
          "enabled": false
        },
        {
          "name": "eth7",
          "ipPoolRef": "",
          "enabled": false
        },
        {
          "name": "eth8",
          "ipPoolRef": "",
          "enabled": false
        }
      ]
    }
  }
]
Listing One COS Node

This command returns information about the specified COS Node object.

**Syntax**

GET https://SM_FQDN:8043/v2/cosnodes/node_name

**Example**

https://172.22.98.59:8043/v2/cosnodes/2887148415

```json
{
    "id": "smtenant_system.smcosnode.2887148415",
    "name": "2887148415",
    "type": "cosnodes",
    "externalId": "/v2/cosnodes/2887148415",
    "transactionId": "ff7e257-b8b1-4d83-834c-2d7aa1778de6",
    "properties": {
        "dataInterfaces": [
            {
                "name": "eth2",
                "ipPoolRef": "smtenant_system.smippool.IPPool-2",
                "enabled": true
            },
            {
                "name": "eth3",
                "ipPoolRef": "smtenant_system.smippool.IPPool-2",
                "enabled": true
            },
            {
                "name": "eth4",
                "ipPoolRef": "smtenant_system.smippool.IPPool-2",
                "enabled": true
            },
            {
                "name": "eth5",
                "ipPoolRef": ",",
                "enabled": false
            },
            {
                "name": "eth6",
                "ipPoolRef": ",",
                "enabled": false
            },
            {
                "name": "eth7",
                "ipPoolRef": ",",
                "enabled": false
            }
        ]
    }
}
```
Creating a COS Node

This command creates and configures a new COS Node object.

**Syntax**

**POST** https://SM_FQDN:8043/v2/cosnodes/node_name

**Example**

- **Request curl:**

  ```
curl -k -H "Content-Type: application/json" -X POST --data @body.json https://10.10.150.22:8043/v2/cosnodes/173926473
  ```

- **@body.json file format:**

  ```
  {
    "properties": {
      "dataInterfaces": [
        {
          "name": "eth2",
          "ipPoolRef": "smtenant_system.smippool.emer1",
          "enabled": true
        }
      ],
      "zoneRef": "smtenant_system.smzone.zone-1",
      "model": "CDE250-2A4",
      "description": "A COS Node appliance",
      "cosNodeClusterRef": "smregion_0.smcosnodecluster.cluster.ca.01",
      "mgmtAddress": "10.22.22.79",
      "adminState": "inservice"
    }
  }
  ```

**Note**

Set cosNodeClusterRef to the COS Node Cluster ID value returned by the following command:

**GET** https://SM_FQDN:8043/v2/regions/region_name/cosnodeclusters/cluster_name.
Deleting a COS Node

This command deletes the specified existing COS Node object.

Syntax
DELETE https://SM_FQDN:8043/v2/cosnodes/node_name

Example
DELETE https://172.22.98.59:8043/v2/cosnodes/2887148415

Updating a COS Node

This command updates the JSON configuration file for the specified COS Node object.

Syntax
PUT https://SM_FQDN:8043/v2/cosnodes/node_name

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Required</th>
<th>Pattern/Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>zoneRef</td>
<td>String</td>
<td>Yes</td>
<td></td>
<td>ID property of the Zone document to which the node belongs</td>
</tr>
<tr>
<td>model</td>
<td>String</td>
<td>Yes</td>
<td></td>
<td>For example: CDE460-4R1</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cosNodeClusterRef</td>
<td>String</td>
<td>No</td>
<td></td>
<td>ID property of the COS Node Cluster document to which the node belongs</td>
</tr>
<tr>
<td>mgmtAddress</td>
<td>String</td>
<td>Yes</td>
<td></td>
<td>IP address of the management server</td>
</tr>
<tr>
<td>adminState</td>
<td>Enum</td>
<td>Yes</td>
<td></td>
<td>adminState can be one of the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• in-service</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• maintenance</td>
</tr>
<tr>
<td>dataInterfaces</td>
<td>Array</td>
<td>No</td>
<td></td>
<td>An array of any number of cache interfaces</td>
</tr>
<tr>
<td>dataInterfaces Item</td>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>Yes</td>
<td></td>
<td>Unique name within a node. For example: eth1</td>
</tr>
<tr>
<td>ipPoolRef</td>
<td>String</td>
<td>No</td>
<td>^.<em>\smippool..</em></td>
<td>^$</td>
</tr>
<tr>
<td>enabled</td>
<td>boolean</td>
<td>Yes</td>
<td></td>
<td>Enter true or false</td>
</tr>
</tbody>
</table>
Example
PUT https://172.22.98.59:8043/v2/cosnodes/2887148415

- Request curl:
curl -k -H "Content-Type: application/json" -X PUT --data @body.json
https://10.10.150.22:8043/v2/cosnodes/173926473

- @body.json file format:

```json
{
    "properties": {
        "dataInterfaces": [
            {
                "name": "eth2",
                "ipPoolRef": "smtenant_system.smippool.emer1",
                "enabled": true
            }
        ],
        "zoneRef": "smtenant_system.smzone.zone-1",
        "model": "",
        "description": "A COS Node appliance in zone-1",
        "cosNodeClusterRef": "smregion_0.smcosnodecluster.cluster.ca.01",
        "mgmtAddress": "10.22.22.79",
        "adminState": "inservice"
    }
}
```

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Required</th>
<th>Pattern/Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>zoneRef</td>
<td>String</td>
<td>Yes</td>
<td></td>
<td>ID property of the Zone document to which the Node belongs. Manually updated to point to zone object.</td>
</tr>
<tr>
<td>model</td>
<td>String</td>
<td>Yes</td>
<td></td>
<td>Eg: CDE460-4R1</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cosNodeClusterRef</td>
<td>String</td>
<td>No</td>
<td></td>
<td>ID property of the document to which the Node belongs</td>
</tr>
<tr>
<td>mgmtAddress</td>
<td>String</td>
<td>Yes</td>
<td></td>
<td>IP address of the management server</td>
</tr>
<tr>
<td>adminState</td>
<td>Enum</td>
<td>Yes</td>
<td></td>
<td>adminState can be one of the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• inservice</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• maintenance</td>
</tr>
<tr>
<td>dataInterfaces</td>
<td>Array</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>Yes</td>
<td></td>
<td>Unique name within node. For example: eth1</td>
</tr>
</tbody>
</table>
Listing, Creating, Deleting, and Updating Auth Profiles

Listing the Auth Profiles

This command returns a list of all Auth Profile objects.

**Syntax**
GET https://SM_FQDN:8043/v2/authprofiles

**Example**
https://172.22.98.59:8043/v2/authprofiles

```json
{
    "id": "smtenant_0.smauthprofile.auth-1",
    "name": "auth-1",
    "type": "authprofiles",
    "externalId": "/v2/authprofiles/auth-1",
    "properties": {
        "name": "authprofilename",
        "type": "swauth",
        "description": "A sample Auth profile to use with COS Auth service",
        "userid": "jdoe",
        "accesskey": "a86850deb2742ec3cb41518e26aa2d89",
        "server": {
            "authServerUrl": "http://swauth.sp.com:8888/"
        },
        "tokenRefreshIntervalSec": 3600
    }
}
```

Listing One Auth Profile

This command returns information about the specified Auth Profile object.

**Syntax**
GET https://SM_FQDN:8043/v2/authprofiles/authprofile_name

**Example**
https://172.22.98.59:8043/v2/authprofiles/auth-1

```json
{
}
```
Creating an Auth Profile

This command creates and configures a new Auth Profile object.

Syntax

POST https://SM_FQDN:8043/v2/authprofiles/authprofile_name

Example

- Request curl:
  
curl -k -H "Content-Type: application/json" -X POST --data @body.json

- @body.json file format:
  
  

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Required</th>
<th>Pattern/Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>enum</td>
<td>Yes</td>
<td>type can be one of the following: tmpauth, keystone, swauth</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>userid</td>
<td>String</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>accesskey</td>
<td>String</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Deleting an Auth Profile**

This command deletes the specified existing Auth Profile object.

**Syntax**

DELETE https://SM_FQDN:8043/v2/authprofiles/authprofile_name

**Updating an Auth Profile**

This command updates the JSON configuration file for the specified Auth Profile object.

**Syntax**

PUT https://SM_FQDN:8043/v2/authprofiles/authprofile_name

**Example**

- Request curl:

  ```bash
  curl -k -H "Content-Type: application/json" -X PUT --data @body.json
  ```

- @body.json file format:

  ```json
  {
    "properties": {
      "type": "swauth",
      "description": "profile for cos",
      "userid": "jbrown:user1",
      "accesskey": "tk7ec41e82175a42918c13fd4d172c9399",
      "server": {
        "authServerUrl": "http://auth01.cos1.acme.com",
        "tokenRefreshIntervalSec": "3600"
      }
    }
  }
  ```

**Property Name** | **Type** | **Required** | **Pattern/Value** | **Description**
--- | --- | --- | --- | ---
`tokenRefreshIntervalSec` | String | Yes |  | Token refresh interval in seconds
`server` | object | Yes |  |  
 `server Properties` |  |  |  |  
`authServerUrl` | String | Yes |  |  

**Property Name** | **Type** | **Required** | **Pattern/Value** | **Description**
--- | --- | --- | --- | ---
type | enum | Yes |  | type can be one of the following:
  - tmpauth
  - keystone
  - swauth
description | String | No |  |  
Listing and Updating Service Instances

A COS Service Instance object specifies the Node objects to serve as endpoints for capture and playback. An image manifest specifies which images, and hence which Node objects, are selected for software instantiation and use.

In COS 3.0.1, Service Instance objects can only be read or updated. A set of five live and five VoD Service Instances come factory installed. These are inactive by default, but can be customized and then activated for use.

### Listing the Service Instances

This command returns a list of all Service Instance objects.

**Syntax**

GET `https://SM_FQDN:8043/v2/serviceinstances`

**Example**

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

https://172.22.98.59:8043/v2/serviceinstances/

```

```json
[
  {
    "id": "smtenant_0.smserviceinstance.platform-0-1",
    "name": "platform-0-1",
    "type": "serviceinstances",
    "externalId": "/v2/serviceinstances/platform-0-1",
    "properties": {
      "state": "active",
      "title": "Platform Service Instance",
      "regionsRef": [
        "smtenant_system.smregion.region-0"
      ],
      "imageManifestRef": "smtenant_0.smimagemanifest.image-manifest-1",
      "description": "Platform Service Resources",
      "templateRef": "smbase.sm.servicetemplate.platform_0.v1",
      "customConfigs": []
    }
  },
  {
    "id": "smtenant_0.smserviceinstance.ums-0-1",
    "name": "ums-0-1",
    ...
  }
]
```
Listing and Updating Service Instances

```
{
  "type": "serviceinstances",
  "externalId": "/v2/serviceinstances/ums-0-1",
  "properties": {
    "state": "inactive",
    "title": "An unused UMS Service",
    "regionsRef": [],
    "description": "A sample UMS service instance",
    "templateRef": "smbase.smservicetemplate.ums_0.v1",
    "customConfigs": []
  }
},
{
  "id": "smtenant_0.smserviceinstance.ums-0-2",
  "name": "ums-0-2",
  "type": "serviceinstances",
  "externalId": "/v2/serviceinstances/ums-0-2",
  "properties": {
    "state": "inactive",
    "title": "An unused UMS Service",
    "regionsRef": [],
    "description": "A sample UMS service instance",
    "templateRef": "smbase.smservicetemplate.ums_0.v1",
    "customConfigs": []
  }
},
{
  "id": "smtenant_0.smserviceinstance.ums-0-3",
  "name": "ums-0-3",
  "type": "serviceinstances",
  "externalId": "/v2/serviceinstances/ums-0-3",
  "properties": {
    "state": "inactive",
    "title": "An unused UMS Service",
    "regionsRef": [],
    "description": "A sample UMS service instance",
    "templateRef": "smbase.smservicetemplate.ums_0.v1",
    "customConfigs": []
  }
},
{
  "id": "smtenant_0.smserviceinstance.ums-0-4",
  "name": "ums-0-4",
  "type": "serviceinstances",
  "externalId": "/v2/serviceinstances/ums-0-4",
  "properties": {
    "state": "inactive",
    "title": "An unused UMS Service",
    "regionsRef": [],
    "description": "A sample UMS service instance",
    "templateRef": "smbase.smservicetemplate.ums_0.v1",
    "customConfigs": []
  }
},
{
  "id": "smtenant_0.smserviceinstance.ums-0-5",
  "name": "ums-0-5",
  "type": "serviceinstances",
  "externalId": "/v2/serviceinstances/ums-0-5",
  "properties": {
    "state": "inactive",
    "title": "An unused UMS Service",
    "regionsRef": [],
    "description": "A sample UMS service instance",
    "templateRef": "smbase.smservicetemplate.ums_0.v1",
```
Listing and Updating Service Instances

"customConfigs": []
}
}
{
"id": "smtenant_0.smserviceinstance.cos-0-1",
"name": "cos-0-1",
"type": "serviceinstances",
"externalId": "/v2/serviceinstances/cos-0-1",
"transactionId": "540f383b-4ede-4362-be7a-9e3806497a4f",
"properties": {
  "state": "active",
  "title": "An unused COS Service",
  "regionsRef": [
    "smtenant_system.smregion.region-0"
  ],
  "imageManifestRef": "smtenant_0.smimagemanifest.image-manifest-1",
  "description": "COS service",
  "templateRef": "smbase.smservicetemplate.cos_0.v1",
  "customConfigs": []
}
}

**Listing One Service Instance**

This command returns information about the specified Service Instance object.

**Syntax**

GET https://SM_FQDN:8043/v2/serviceinstances/serviceinstance_name

**Example**

https://172.22.98.59:8043/v2/serviceinstances/cos-0-1/
{
  "id": "smtenant_0.smserviceinstance.cos-0-1",
  "name": "cos-0-1",
  "type": "serviceinstances",
  "externalId": "/v2/serviceinstances/cos-0-1",
  "transactionId": "540f383b-4ede-4362-be7a-9e3806497a4f",
  "properties": {
    "state": "active",
    "title": "An unused COS Service",
    "regionsRef": [
      "smtenant_system.smregion.region-0"
    ],
    "imageManifestRef": "smtenant_0.smimagemanifest.image-manifest-1",
    "description": "COS service",
    "templateRef": "smbase.smservicetemplate.cos_0.v1",
    "customConfigs": []
  }
}

**Updating a Service Instance**

This command updates the JSON configuration file for the specified Service Instance object.
Syntax
PUT https://SM_FQDN:8043/v2/serviceinstances/serviceinstance_name

Example
- Request curl:
  ```
curl -k  
  -H "Content-Type: application/json"  
  -X PUT --data @body.json  
  https://10.10.150.22:8043/v2/serviceinstances/cos-0-1
  ```

- `@body.json` file format:
  ```
  {
    "properties": {
      "state": "active",
      "title": "COS Service 2",
      "regionsRef": [
        "smtenant_system.smregion.region-0"],
      "imageManifestRef": "smtenant_0.smimagemanifest.image-manifest-1",
      "description": "COS service",
      "templateRef": "smbase.smservicetemplate.cos_0.v1",
      "customConfigs": []
    }
  }
  ```

### Property Name | Type | Required | Pattern/Value | Description
--- | --- | --- | --- | ---
state | enum | Yes | state can be one of the following: inactive, init, validating, active, deleting, deleted | Customer provided short displayable string name for this service instance

- Title
- RegionsRef: A comma separated list of regions to which this Service Instance applies. It can be an empty array.
- ImageManifestRef: ID of the imageManifest document for example: smtenant_0.smimagemanifest.t.0
Chapter 2      Service Manager API

Listing, Creating, Deleting, and Updating Service Endpoints

A COS Service Endpoint object is a logical pointer to the resources and policies defined for the capture, ingest, and storage of specific content. A Service Instance object can have one or more associated Service Endpoint objects.

Listing the Service Endpoints

This command returns a list of all Service Endpoint objects.

Syntax
GET https://SM_FQDN:8043/v2/serviceinstances/serviceinstance_name/cosendpoints

Example
https://172.22.98.59:8043/v2/serviceinstances/cos-0-1/cosendpoints

```json
[
  {  
    "id": "smserviceinstance_cos_0_1.smcosendpoint.EP1",  
    "name": "EP1",  
    "type": "cosendpoints",  
    "externalId": "/v2/serviceinstances/cos-0-1/cosendpoints/EP1",  
    "transactionId": "56268f93-a928-4117-803f-9ba44aacbf41",  
    "properties": {  
      "storages": [  
        {  
          "storageRef": "smregion_0.smcosnodecluster.Cluster-1",  
          "authProfileRef": "smtenant_0.smauthprofile.auth-1"  
        }  
      ]  
    }  
  }
]```
Listing One Service Endpoint

This command returns information about the specified Service Endpoint object.

Syntax
GET https://SM_FQDN:8043/v2/serviceinstances/serviceinstance_name/cosendpoints/endpt_name

Example

```json
{
  "id": "smserviceinstance_cos_0_1.smcosendpoint.EP1",
  "name": "EP1",
  "type": "cosendpoints",
  "externalId": "/v2/serviceinstances/cos-0-1/cosendpoints/EP1",
  "transactionId": "56268f93-a928-4117-803f-9ba44aacbf41",
  "properties": {
    "storages": [
      {
        "storageRef": "smregion_0.smcosnodecluster.Cluster-1",
        "authProfileRef": "smtenant_0.smauthprofile.auth-1"
      }
    ],
    "slaType": "resource",
    "assetRedundancyPolicyRef": "smtenant_0.smassetredundancypolicy.redundancy-pol-cos-mirroring",
    "state": "enabled",
    "resourceSLA": {
      "minNode": 1,
      "desiredNode": 3,
      "maxNode": 3,
      "maxStorage": 0
    },
    "description": "endpoint",
    "regionRef": "smtenant_system.smregion.region-0"
  }
}
```
Creating a Service Endpoint

This command creates and configures a new Service Endpoint object.

Syntax

POST https://SM_FQDN:8043/v2/serviceinstances/serviceinstance_name/cosendpoints/endpt_name

Example

- Request curl:

  curl -k -H "Content-Type: application/json" -X POST --data @body.json https://10.10.150.22:8043/v2/captureendpoints/ce2

- @body.json file format:

  ```json
  {
    "properties": {
      "storages": [
        {
          "storageRef": "smregion_0.smcosnodecluster.Cluster-1",
          "authProfileRef": "smtenant_0.smauthprofile.auth-1"
        }
      ],
      "slaType": "resource",
      "assetRedundancyPolicyRef": "smtenant_0.smassetredundancypolicy.redundancy-pol-cos-mirroring",
      "state": "disabled",
      "resourceSLA": {
        "minNode": 1,
        "desiredNode": 3,
        "maxNode": 3,
        "maxStorage": 500
      },
      "description": "first end point",
      "regionRef": "smtenant_system.smregion.region-0"
    }
  }
  ```

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Required</th>
<th>Pattern/Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>assetRedundancyPolicyRef</td>
<td>String</td>
<td>No</td>
<td>.<em>..smassetredundancypolicy..</em></td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slaType</td>
<td>enum</td>
<td>Yes</td>
<td></td>
<td>The slaType can be one of the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• resource</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• capacity</td>
</tr>
<tr>
<td>resourceSLA</td>
<td>Object</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>capacitySLA</td>
<td>Object</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>regionRef</td>
<td>String</td>
<td>Yes</td>
<td>.<em>..smregion..</em></td>
<td>ID of the region document For example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>smtenant_system.smregion.region-0</td>
</tr>
</tbody>
</table>
Deleting a Service Endpoint

This command deletes the specified existing Service Endpoint object.

Syntax
DELETE
https://SM_FQDN:8043/v2/serviceinstances/serviceinstance_name/cosendpoints/endpt_name

Example
https://SM_FQDN:8043/v2/serviceinstances/serviceinstance_name/cosendpoints/endpt_name

Updating a Service Endpoint

This command updates the JSON configuration file for the specified Service Endpoint object.

Syntax
PUT https://SM_FQDN:8043/v2/serviceinstances/serviceinstance_name/cosendpoints/endpt_name

Example
https://SM_FQDN:8043/v2/serviceinstances/serviceinstance_name/cosendpoints/endpt_name

- Request curl:
  curl -k -H "Content-Type: application/json" -X PUT --data @body.json
  https://10.10.150.22:8043/v2/captureendpoints/ce2

- @body.json file format:

```
{
  "properties": {
    "storages": [
      {
        "storageRef": "smregion_0.smcosnodecluster.Cluster-1",
        "authProfileRef": "smtenant_0.smauthprofile.auth-1"
      }
    ],
    "slaType": "resource",
    "assetRedundancyPolicyRef": "smtenant_0.smassetredundancypolicy.redundancy-pol-cos-mirroring",
    "state": "disabled",
    "resourceSLA": {
      "minNode": 1,
      "desiredNode": 3
    }
  }
}
```
Listing, Creating, Deleting, and Updating Asset Redundancy Policies

Listing the Asset Redundancy Policies

This command returns a list of all Asset Redundancy Policy objects.

Syntax
GET https://SM_FQDN:8043/v2/assetredundancypolicies/

Example
https://172.22.98.59:8043/v2/assetredundancypolicies/

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Required</th>
<th>Pattern/Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>assetRedundancyPolicyRef</td>
<td>String</td>
<td>No</td>
<td>.<em>\smassetredundancypolicy..</em></td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slaType</td>
<td>enum</td>
<td>Yes</td>
<td></td>
<td>The slaType can be one of the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• resource</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• capacity</td>
</tr>
<tr>
<td>resourceSLA</td>
<td>Object</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>capacitySLA</td>
<td>Object</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>regionRef</td>
<td>String</td>
<td>Yes</td>
<td>.<em>\smregion..</em></td>
<td>ID of the region document For example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>smtenant_system.smregion.region-0</td>
</tr>
<tr>
<td>defaultStorageRef</td>
<td>String</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>storages</td>
<td>Array</td>
<td>No</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

storages Item Properties

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Type</th>
<th>Required</th>
<th>Pattern/Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>storageRef</td>
<td>String</td>
<td>Yes</td>
<td></td>
<td>ID of a NASStore or CosNodeCluster</td>
</tr>
<tr>
<td>authProfileRef</td>
<td>String</td>
<td>No</td>
<td></td>
<td>Required if storageRef is CosNodeCluster</td>
</tr>
</tbody>
</table>
Listing, Creating, Deleting, and Updating Asset Redundancy Policies

Listing One Asset Redundancy Policy

This command returns information about the specified Asset Redundancy Policy object.

```json
{
  "id": "smtenant_0.smassetredundancypolicy.redundancy-pol-cos2",
  "name": "redundancy-pol-cos2",
  "type": "assetredundancypolicies",
  "externalId": "/v2/assetredundancypolicies/redundancy-pol-cos2",
  "transactionId": "e375c042-9c70-44e8-b03a-2047d0b5cc7a",
  "properties": {
    "type": "mirroring",
    "description": "Redundancy policy for cluster-b",
    "rules": [
      {
        "matchTag": "local",
        "trigger": "complete",
        "copyCount": 3,
        "keepCount": 0,
        "enabled": true
      },
      {
        "matchTag": "remote",
        "trigger": "complete",
        "copyCount": 2,
        "keepCount": 0,
        "enabled": true
      }
    ]
  }
}
{
  "id": "smtenant_0.smassetredundancypolicy.redundancy-pol-cos-mirroring",
  "name": "redundancy-pol-cos-mirroring",
  "type": "assetredundancypolicies",
  "externalId": "/v2/assetredundancypolicies/redundancy-pol-cos-mirroring",
  "transactionId": "cd1a88dd-ab25-4c50-9b16-8787c738ea3d",
  "properties": {
    "type": "mirroring",
    "description": "A sample local and remote mirroring policy for COS objects",
    "rules": [
      {
        "matchTag": "local",
        "trigger": "start",
        "copyCount": 3,
        "enabled": false,
        "keepCount": 0
      },
      {
        "matchTag": "remote",
        "trigger": "start",
        "copyCount": 2,
        "enabled": false,
        "keepCount": 0
      }
    ]
  }
}
```
Creating an Asset Redundancy Policy

This command creates and configures a new Asset Redundancy Policy object.

Syntax
POST https://SM_FQDN:8043/v2/assetredundancypolicies/assetredundancypolicy_name

Example
https://172.22.98.59:8043/v2/assetredundancypolicies/redundancy-pol-cos-mirroring

Request curl:
curl -k -H "Content-Type: application/json" -X PUT --data @body.json
https://172.22.98.59:8043/v2/assetredundancypolicies/redundancy-pol-cos-mirroring

@body.json file format:
```json
{
  "properties": {
    "type": "mirroring",
    "description": "A sample local and remote mirroring policy for COS objects",
    "rules": [
      {
        "matchTag": "local",
        "trigger": "start",
        "copyCount": 2,
        "keepCount": 1,
        "enabled": false
      },
      {
        "matchTag": "remote",
        "trigger": "start",
        "copyCount": 3,
        "keepCount": 1,
        "enabled": false
      }
    ]
  }
}
```
Dealing an Asset Redundancy Policy

This command deletes the specified existing Asset Redundancy Policy object.

Syntax
DELETE https://SM_FQDN:8043/v2/assetredundancypolicies/assetredundancypolicy_name

Example
https://172.22.98.59:8043/v2/assetredundancypolicies/redundancy-pol-cos-mirroring

Updating an Asset Redundancy Policy

This command updates the JSON configuration file for the specified Asset Redundancy Policy object.

Syntax
PUT https://SM_FQDN:8043/v2/assetredundancypolicies/assetredundancypolicy_name

Example
https://172.22.98.59:8043/v2/assetredundancypolicies/redundancy-pol-cos-mirroring

- Request curl:
  curl -k -H "Content-Type: application/json" -X PUT --data @body.json https://172.22.98.59:8043/v2/assetredundancypolicies/redundancy-pol-cos-mirroring

- @body.json file format:

```json
{
  "properties": {
    "type": "mirroring",
    "description": "A sample local and remote mirroring policy for COS objects",
    "rules": [
      {
        "matchTag": "local",
        "trigger": "start",
        "copyCount": 3,
        "enabled": false
      },
      {
        "matchTag": "remote",
        "trigger": "start",
        "copyCount": 2,
        "enabled": false
      }
    ]
  }
}
```
Viewing COS Node Status

NodeStatus is a read-only object that provides the status of a COS Node object. NodeStatus identifies the state, fault detail (if any), linked Service Instance and Endpoint objects, using application, and all interface details of the COS Node object.

Viewing the Status of COS Nodes

This command returns the status of all COS Node objects.

Syntax

GET https://SM_FQDN:8043/v2/regions/regionName/nodestatuses

Example

https://172.22.98.59:8043/v2/regions/region-0/nodestatuses

{  
  "id": "smregion_0_status.smnodestatus.2887148484",  
  "name": "2887148484",  
  "type": "nodestatuses",  
  "externalId": "/v2/regions/region-0/nodestatuses/2887148484",  
  "transactionId": "977faeaе-06db-4efб-a385-6142683b1f85",  
  "properties": {  
    "hostname": "172.22.99.196",  
    "serviceInstanceRef": "smtenant_0.smserviceinstance.cos-0-1",  
    "endpointRef": "smserviceinstance.cos_0_1.smcosendpoint.EP1",  
    "state": "inuse",  
    "faultStatus": "None",  
    "faultDetail": "-- All 1 configured network interfaces out of 8 total interfaces are up on the COS Node\n-- All 24 disks are up on the COS Node\n-- All 5 services are up on the COS Node\n",  
    "lastModifiedTime": "2015-07-06T18:02:54.997Z",  
    "interfaces": [  
      {  
        "interface": "eth2",  
        "type": "service",  
        "inet": "",  
        "netmask": "",  
        "inetv6": "",  
        "gw": "",  
        "dnsServer": "",  
        "macAddr": "",  
        "status": "inactive",  
        "speed": "1GB"  
      },  
      {  
        "interface": "eth3",  
        "type": "service",  
        "inet": "",  
        "netmask": "",  
        "inetv6": "",  
        "gw": "",  
        "dnsServer": "",  
        "macAddr": "",  
        "status": "inactive",  
        "speed": "1GB"  
      }  
    ]  
  }  
}
Viewing COS Node Status

```
{
  "interface": "eth4",
  "type": "service",
  "inet": "",
  "netmask": "",
  "inetv6": "",
  "gw": "",
  "dnsServer": "",
  "macAddr": "",
  "status": "inactive",
  "speed": "1GB"
},
{
  "interface": "eth5",
  "type": "service",
  "inet": "",
  "netmask": "",
  "inetv6": "",
  "gw": "",
  "dnsServer": "",
  "macAddr": "",
  "status": "inactive",
  "speed": "1GB"
},
{
  "interface": "eth6",
  "type": "service",
  "inet": "192.169.215.17",
  "netmask": "255.255.255.240",
  "inetv6": "",
  "gw": "192.169.215.30",
  "dnsServer": "",
  "macAddr": "",
  "status": "active",
  "speed": "10GB"
},
{
  "interface": "eth7",
  "type": "service",
  "inet": "",
  "netmask": "",
  "inetv6": "",
  "gw": "",
  "dnsServer": "",
  "macAddr": "",
  "status": "inactive",
  "speed": "10GB"
},
{
  "interface": "eth8",
  "type": "service",
  "inet": "",
  "netmask": "",
  "inetv6": "",
  "gw": "",
  "dnsServer": "",
  "macAddr": "",
  "status": "inactive",
```


```
{
    "speed": "10GB",
    "interface": "eth9",
    "type": "service",
    "inet": "",
    "netmask": "",
    "inetv6": "",
    "gw": "",
    "dnsServer": "",
    "macAddr": "",
    "status": "inactive",
    "speed": "10GB"
}
```

```
"image": {
    "imgTag": "",
    "version": "",
    "personality": "COS Hardware",
    "source": ""
},
"configRef": "smtenant_system.smcosnode.2887148484",
"usageStatus": {
    "storageUsage": {
        "used": 0.4,
        "total": 22356.32
    },
    "bandwidthUsage": {
        "rx": {
            "used": 0,
            "total": 9393.69
        },
        "tx": {
            "used": 0,
            "total": 9393.69
        }
    },
    "sessionUsage": {
        "rx": {
            "used": 0,
            "total": 8192
        },
        "tx": {
            "used": 0,
            "total": 6000
        }
    }
},
"storage": {
    "disks": [
        {
            "name": "Cisco Disk 01",
            "status": "up"
        },
        {
            "name": "Cisco Disk 02",
            "status": "up"
        },
        {
            "name": "Cisco Disk 03",
            "status": "up"
        },
        {
            "name": "Cisco Disk 04",
```
Viewing COS Node Status

```

  "status": "up"
  },
  {
    "name": "Cisco Disk 05",
    "status": "up"
  },
  {
    "name": "Cisco Disk 06",
    "status": "up"
  },
  {
    "name": "Cisco Disk 07",
    "status": "up"
  },
  {
    "name": "Cisco Disk 08",
    "status": "up"
  },
  {
    "name": "Cisco Disk 09",
    "status": "up"
  },
  {
    "name": "Cisco Disk 10",
    "status": "up"
  },
  {
    "name": "Cisco Disk 11",
    "status": "up"
  },
  {
    "name": "Cisco Disk 12",
    "status": "up"
  },
  {
    "name": "Cisco Disk 13",
    "status": "up"
  },
  {
    "name": "Cisco Disk 14",
    "status": "up"
  },
  {
    "name": "Cisco Disk 15",
    "status": "up"
  },
  {
    "name": "Cisco Disk 16",
    "status": "up"
  },
  {
    "name": "Cisco Disk 17",
    "status": "up"
  },
  {
    "name": "Cisco Disk 18",
    "status": "up"
  },
  {
    "name": "Cisco Disk 19",
    "status": "up"
  }
```
Chapter 2      Service Manager API

Viewing COS Node Status

```
"status": "up"
},
{"name": "Cisco Disk 21",
"status": "up"
},
{"name": "Cisco Disk 20",
"status": "up"
},
{"name": "Cisco Disk 23",
"status": "up"
},
{"name": "Cisco Disk 24",
"status": "up"
}
],
"service": {
"services": [
{"name": "Cisco Cache Server",
"state": "up"
},
{"name": "Cisco Cloud Object Store Daemon",
"state": "up"
},
{"name": "Cisco SNMP Agent",
"state": "up"
},
{"name": "NTP Daemon",
"state": "up"
},
{"name": "Cassandra",
"state": "up"
}
]
}
]

Viewing the Status of a Specific COS Node

This command returns the status of the specified COS Node object.

Syntax
GET https://SM_FQDN:8043/v2/regions/regionName/nodestatuses/cosnode_name

Example
https://172.22.98.59:8043/v2/regions/region-0/nodestatuses/2887148484
{ "id": "smregion_0_status.smnodestatus.2887148484",
 "name": "2887148484",
}
"type": "nodestatuses",
"externalId": "/v2/regions/region-0/nodestatuses/2887148484",
"transactionId": "6af5624d-3d9f-4ca6-a63e-e97a69771402",
"properties": {
  "hostname": "172.22.99.196",
  "serviceInstanceRef": "smtenant_0.smsserviceinstance.cos-0-1",
  "endpointRef": "smsserviceinstance_cos_0_1.smcosendpoint.EP1",
  "state": "inuse",
  "faultStatus": "None",
  "faultDetail": "-- All 1 configured network interfaces out of 8 total interfaces are up on the COS Node
-- All 24 disks are up on the COS Node
-- All 5 services are up on the COS Node",
  "lastModifiedTime": "2015-07-06T18:06:49.138Z",
  "interfaces": [
    {
      "interface": "eth2",
      "type": "service",
      "inet": "",
      "netmask": "",
      "inetv6": "",
      "gw": "",
      "dnsServer": "",
      "macAddr": "",
      "status": "inactive",
      "speed": "1GB"
    },
    {
      "interface": "eth3",
      "type": "service",
      "inet": "",
      "netmask": "",
      "inetv6": "",
      "gw": "",
      "dnsServer": "",
      "macAddr": "",
      "status": "inactive",
      "speed": "1GB"
    },
    {
      "interface": "eth4",
      "type": "service",
      "inet": "",
      "netmask": "",
      "inetv6": "",
      "gw": "",
      "dnsServer": "",
      "macAddr": "",
      "status": "inactive",
      "speed": "1GB"
    },
    {
      "interface": "eth5",
      "type": "service",
      "inet": "",
      "netmask": "",
      "inetv6": "",
      "gw": "",
      "dnsServer": "",
      "macAddr": "",
      "status": "inactive",
      "speed": "1GB"
    },
    {
      "interface": "eth6",
      "type": "service",
      "inet": "",
      "netmask": "",
      "inetv6": "",
      "gw": "",
      "dnsServer": "",
      "macAddr": "",
      "status": "inactive",
      "speed": "1GB"
    }
  ]
}
Chapter 2  Service Manager API

Viewing COS Node Status

"type": "service",
"inet": "192.169.215.17",
"netmask": "255.255.255.240",
"inetv6": "",
"gw": "192.169.215.30",
"dnsserver": "",
"macAddr": "",
"status": "active",
"speed": "10GB"
},
{
"interface": "eth7",
"type": "service",
"inet": "",
"netmask": "",
"inetv6": "",
"gw": "",
"dnsserver": "",
"macAddr": "",
"status": "inactive",
"speed": "10GB"
},
{
"interface": "eth8",
"type": "service",
"inet": "",
"netmask": "",
"inetv6": "",
"gw": "",
"dnsserver": "",
"macAddr": "",
"status": "inactive",
"speed": "10GB"
},
{
"interface": "eth9",
"type": "service",
"inet": "",
"netmask": "",
"inetv6": "",
"gw": "",
"dnsserver": "",
"macAddr": "",
"status": "inactive",
"speed": "10GB"
}
],
'image': {
"imgTag": "",
"version": "",
"personality": "COS Hardware",
"source": ""
},
'configRef': "smttenant_system.smcosnode.2887148484",
'usageStatus': {
"storageUsage": {
"used": 0.4,
"total": 22356.32
},
"bandwidthUsage": {
"rx": {
"used": 0,
"total": 9393.69
}
"tx": {
    "used": 0,
    "total": 9393.69
},
"sessionUsage": {
    "rx": {
        "used": 0,
        "total": 8192
    },
    "tx": {
        "used": 0,
        "total": 6000
    }
},
"storage": {
    "disks": [
        {
            "name": "Cisco Disk 01",
            "status": "up"
        },
        {
            "name": "Cisco Disk 02",
            "status": "up"
        },
        {
            "name": "Cisco Disk 03",
            "status": "up"
        },
        {
            "name": "Cisco Disk 04",
            "status": "up"
        },
        {
            "name": "Cisco Disk 05",
            "status": "up"
        },
        {
            "name": "Cisco Disk 06",
            "status": "up"
        },
        {
            "name": "Cisco Disk 07",
            "status": "up"
        },
        {
            "name": "Cisco Disk 08",
            "status": "up"
        },
        {
            "name": "Cisco Disk 09",
            "status": "up"
        },
        {
            "name": "Cisco Disk 10",
            "status": "up"
        },
        {
            "name": "Cisco Disk 11",
            "status": "up"
        },
        {
            "name": "Cisco Disk 12",
            "status": "up"
        }
    ]
}
"status": "up",
},
{
  "name": "Cisco Disk 13",
  "status": "up"
},
{
  "name": "Cisco Disk 14",
  "status": "up"
},
{
  "name": "Cisco Disk 15",
  "status": "up"
},
{
  "name": "Cisco Disk 16",
  "status": "up"
},
{
  "name": "Cisco Disk 17",
  "status": "up"
},
{
  "name": "Cisco Disk 18",
  "status": "up"
},
{
  "name": "Cisco Disk 19",
  "status": "up"
},
{
  "name": "Cisco Disk 20",
  "status": "up"
},
{
  "name": "Cisco Disk 21",
  "status": "up"
},
{
  "name": "Cisco Disk 22",
  "status": "up"
},
{
  "name": "Cisco Disk 23",
  "status": "up"
},
{
  "name": "Cisco Disk 24",
  "status": "up"
}
],
"service": {
  "services": [
  {
    "name": "Cisco Cloud Object Store Daemon",
    "state": "up"
  },
  {
    "name": "Cisco Cache Server",
    "state": "up"
  },
  {
    "name": "Cisco SNMP Agent",
    "state": "up"
  }
]
Viewing Events for Nodes

Event is a read-only object that provides a view into state changes inside COS. It reports task progress, warnings, and errors, letting you evaluate the system reaction to any configuration updates and runtime checks for events by severity.

Viewing the Events for Nodes in a Region

This command returns a list of events for all COS Node objects in the specified Region.

Syntax

GET https://SM_FQDN:8043/v2/regions/regionName/events

Example

https://172.22.98.59:8043/v2/regions/region-0/events

```json
{
    "id": "smregion_0_events.smevent.pam-6f1effb0-1f52-11e5-9c7c-593b78e08f95",
    "name": "pam-6f1effb0-1f52-11e5-9c7c-593b78e08f95",
    "type": "events",
    "externalId": "/v2/regions/region-0/events/pam-6f1effb0-1f52-11e5-9c7c-593b78e08f95",
    "transactionId": "0fde3730-1c96-4369-8bc6-92f47c6b0863",
    "properties": {
        "source": {
            "imgTag": "pam",
            "personality": "control",
            "node": "172.22.98.59"
        },
        "type": "application",
        "subType": "Interface",
        "event": "DnsUpdate",
        "detailText": "addRecord {u'record': u'CNAME', u'domain': u'cos5.cisco.com', u'cname': u'mgmt-docserver', u'host': u'pm-172-22-98-59-mos'} Completed.",
        "eventsDropped": 0,
        "severity": "info",
        "location": {
            "ipAddr": "172.22.98.59",
            "processName": "",
```
"processId": ""
}
}
},
{
"id": "smregion_0_events.smevent.pam-6ffc1940-1f52-11e5-9c7c-593b78e08f95",
"name": "pam-6ffc1940-1f52-11e5-9c7c-593b78e08f95",
"type": "events",
"externalId": "/v2/regions/region-0/events/pam-6ffc1940-1f52-11e5-9c7c-593b78e08f95",
"transactionId": "1b818e04-9a26-44a5-8a90-b0a537b605c9",
"properties": {
  "source": {
    "imgTag": "pam",
    "personality": "control",
    "node": "172.22.98.59"
  },
  "type": "application",
  "subType": "Interface",
  "event": "DnsUpdate",
  "detailText": "addRecord {u'record': u'CNAME', u'domain': u'cos5.cisco.com', u'cname': u'pam-docserver', u'host': u'pm-172-22-98-59-mos'} Completed.",
  "eventTime": "2015-06-30T18:04:23.106Z",
  "eventsDropped": 0,
  "severity": "info",
  "location": {
    "ipAddr": "172.22.98.59",
    "processName": "",
    "processId": ""
  }
}
},
{
"id": "smregion_0_events.smevent.pam-70237760-1f52-11e5-9c7c-593b78e08f95",
"name": "pam-70237760-1f52-11e5-9c7c-593b78e08f95",
"type": "events",
"externalId": "/v2/regions/region-0/events/pam-70237760-1f52-11e5-9c7c-593b78e08f95",
"transactionId": "dad31e28-84da-4b3d-97eb-b083a4052235",
"properties": {
  "source": {
    "imgTag": "pam",
    "personality": "control",
    "node": "172.22.98.59"
  },
  "type": "application",
  "subType": "Interface",
  "event": "DnsUpdate",
  "detailText": "addRecord {u'record': u'CNAME', u'domain': u'cos5.cisco.com', u'cname': u'service-mgr', u'host': u'pm-172-22-98-59-mos'} Completed.",
  "eventTime": "2015-06-30T18:04:23.259Z",
  "eventsDropped": 0,
  "severity": "info",
  "location": {
    "ipAddr": "172.22.98.59",
    "processName": "",
    "processId": ""
  }
}
},
{
"id": "smregion_0_events.smevent.pam-703355e0-1f52-11e5-9c7c-593b78e08f95",
"name": "pam-703355e0-1f52-11e5-9c7c-593b78e08f95",
"type": "events",
"externalId": "/v2/regions/region-0/events/pam-703355e0-1f52-11e5-9c7c-593b78e08f95",
"transactionId": "8dbd12ef-84da-4b3d-97eb-b083a4052235",
"properties": {
  "source": {
    "imgTag": "pam",
    "personality": "control",
    "node": "172.22.98.59"
  },
  "type": "application",
  "subType": "Interface",
  "event": "DnsUpdate",
  "detailText": "addRecord {u'record': u'CNAME', u'domain': u'cos5.cisco.com', u'cname': u'pm-172-22-98-59-mos'} Completed.",
  "eventTime": "2015-06-30T18:04:23.179Z",
  "eventsDropped": 0,
  "severity": "info",
  "location": {
    "ipAddr": "172.22.98.59",
    "processName": "",
    "processId": ""
  }
}
}
"externalId": "/v2/regions/region-0/events/pam-70355e0-1f52-11e5-9c7c-593b78e08f95",
"transactionId": "73db7eb0-2b9b-4879-b948-1ba1e986132a",
"properties": {
  "source": {
    "imgTag": "pam",
    "personality": "control",
    "node": "172.22.98.59"
  },
  "type": "application",
  "subType": "Interface",
  "event": "DnsUpdate",
  "detailText": "addRecord {u'record': u'CNAME', u'domain': u'cos5.cisco.com', u'cname': u'cos-controller', u'host': u'pm-172-22-98-59-mos'} Completed."

  "eventTime": "2015-06-30T18:04:23.420Z",
  "eventsDropped": 0,
  "severity": "info",
  "location": {
    "ipAddr": "172.22.98.59",
    "processName": "",
    "processId": ""
  }
},
{
  "id": "smregion_0_events.smevent.pam-703c7da0-1f52-11e5-9c7c-593b78e08f95",
  "name": "pam-703c7da0-1f52-11e5-9c7c-593b78e08f95",
  "type": "events",
  "externalId": "/v2/regions/region-0/events/pam-703c7da0-1f52-11e5-9c7c-593b78e08f95",
  "transactionId": "31a2640f-c299-447b-9c83-c011c3e33822",
  "properties": {
    "source": {
      "imgTag": "pam",
      "personality": "control",
      "node": "172.22.98.59"
    },
    "type": "node",
    "subType": "Health",
    "event": "PAMNode",
    "detailText": "PAMDS process has been started on PAM",
    "eventTime": "2015-06-30T18:04:23.506Z",
    "eventsDropped": 0,
    "severity": "info",
    "location": {
      "ipAddr": "172.22.98.59",
      "processName": "",
      "processId": ""
    }
  }
},
{
  "id": "smregion_0_events.smevent.pam-70457e50-1f52-11e5-9c7c-593b78e08f95",
  "name": "pam-70457e50-1f52-11e5-9c7c-593b78e08f95",
  "type": "events",
  "externalId": "/v2/regions/region-0/events/pam-70457e50-1f52-11e5-9c7c-593b78e08f95",
  "transactionId": "57fb8fa9-e591-4002-80ec-836dcc923694",
  "properties": {
    "source": {
      "imgTag": "pam",
      "personality": "control",
      "node": "172.22.98.59"
    },
    "type": "application",
    "subType": "Interface",
  }
"event": "DnsUpdate",
"detailText": "addRecord {u'record': u'CNAME', u'domain': u'cos5.cisco.com', u'cname': u'ui', u'host': u'pm-172-22-98-59-mos'} Completed.",
"eventTime": "2015-06-30T18:04:23.587Z",
"eventsDropped": 0,
"severity": "info",
"location": {
  "ipAddr": "172.22.98.59",
  "processName": "",
  "processId": ""
}
},
{
  "id": "smregion_0_events.smevent.pam-705b5040-1f52-11e5-9c7c-593b78e08f95",
  "name": "pam-705b5040-1f52-11e5-9c7c-593b78e08f95",
  "type": "events",
  "externalId": "/v2/regions/region-0/events/pam-705b5040-1f52-11e5-9c7c-593b78e08f95",
  "transactionId": "6870d873-8c2c-438b-afe8-75392813b717",
  "properties": {
    "source": {
      "imgTag": "pam",
      "personality": "control",
      "node": "172.22.98.59"
    },
    "type": "application",
    "subType": "Interface",
    "event": "DnsUpdate",
    "detailText": "addRecord {u'record': u'CNAME', u'domain': u'cos5.cisco.com', u'cname': u'controller', u'host': u'pm-172-22-98-59-mos'} Completed.",
    "eventTime": "2015-06-30T18:04:23.744Z",
    "eventsDropped": 0,
    "severity": "info",
    "location": {
      "ipAddr": "172.22.98.59",
      "processName": "",
      "processId": ""
    }
  }
},
{
  "id": "smregion_0_events.smevent.pam-70a5edd0-1f52-11e5-9c7c-593b78e08f95",
  "name": "pam-70a5edd0-1f52-11e5-9c7c-593b78e08f95",
  "type": "events",
  "externalId": "/v2/regions/region-0/events/pam-70a5edd0-1f52-11e5-9c7c-593b78e08f95",
  "transactionId": "8d79354a-46ac-4243-afe8-75392813b717",
  "properties": {
    "source": {
      "imgTag": "pam",
      "personality": "control",
      "node": "172.22.98.59"
    },
    "type": "node",
    "subType": "Health",
    "event": "PAMNode",
    "detailText": "POST REST request issued to PAM with URL http://rest.pm.mos.cos5.cisco.com:5000/v1/platformService/Ntp failed with return status: 405. Check if rabbitmq is running on PAM."
    "eventsDropped": 0,
    "severity": "major",
    "location": {

"ipAddr": "172.22.98.59",
'processName': ",
'processId': ""
}
},
{
"id": "smregion_0_events.smevent.pam-71788010-1f52-11e5-9c7c-593b78e08f95",
"name": "pam-71788010-1f52-11e5-9c7c-593b78e08f95",
"type": "events",
"externalId": "/v2/regions/region-0/events/pam-71788010-1f52-11e5-9c7c-593b78e08f95",
"transactionId": "3a9e66d7-a3c0-4ff1-82a8-36dc7c359217",
"properties": {
  "source": {
    "imgTag": "pam",
    'personality': "control",
    'node': "172.22.98.59"
  },
  "type": "application",
  "subType": "Interface",
  "event": "DnsUpdate",
  "detailText": "addRecord {u'ip': u'192.169.215.17', u'domain': u'cos5.cisco.com', u'cname': u'', u'host': u'auth01', u'record': u'A'} Completed.",
  "eventTime": "2015-06-30T18:04:25.612Z",
  "eventsDropped": 0,
  "severity": "info",
  "location": {
    "ipAddr": "172.22.98.59",
    'processName': ",
    'processId': ""
  }
}
},
{
"id": "smregion_0_events.smevent.pam-a2d16780-1f52-11e5-9c7c-593b78e08f95",
"name": "pam-a2d16780-1f52-11e5-9c7c-593b78e08f95",
"type": "events",
"externalId": "/v2/regions/region-0/events/pam-a2d16780-1f52-11e5-9c7c-593b78e08f95",
"transactionId": "46c77e2e-d549-42bc-8608-f9acdca2d177",
"properties": {
  "source": {
    "imgTag": "pam",
    'personality': "control",
    'node': "172.22.98.59"
  },
  "type": "ext-resource",
  "subType": "Accessibility",
  "event": "ntpServerPing",
  "detailText": "Ntp server not found",
  "eventTime": "2015-06-30T18:05:48.404Z",
  "eventsDropped": 0,
  "severity": "major",
  "location": {
    "ipAddr": "172.22.98.59",
    'processName': ",
    'processId': ""
  }
}
},
{
"id": "smregion_0_events.smevent.pam-4dc24dc0-20e4-11e5-8fb7-d5b686854e59",
"name": "pam-4dc24dc0-20e4-11e5-8fb7-d5b686854e59",
"type": "events",
"externalId": "/v2/regions/region-0/events/pam-4dc24dc0-20e4-11e5-8fb7-d5b686854e59",
"transactionId": "275f9d0a-47a1-4700-84a0-79f8f2007857",
"properties": {
  "source": {
    "imgTag": "pam",
    'personality': "control",
    'node': "172.22.98.59"
  },
  "type": "events",
  "subType": "Maintenance",
  "event": "profileCoverageUpdate",
  "detailText": "coverage update completed",
  "eventTime": "2015-06-30T18:05:50.404Z",
  "eventsDropped": 0,
  "severity": "info",
  "location": {
    "ipAddr": "172.22.98.59",
    'processName': "",
    'processId': ""
  }
}
}
"externalId": "/v2/regions/region-0/events/pam-4dc24dc0-20e4-11e5-8fb7-d5b686854e59",
"transactionId": "f393fd9b-d91a-46b7-9cc2-3834fbaa7a9c",
"properties": {
  "source": {
    "imgTag": "pam",
    "personality": "control",
    "node": "172.22.98.59"
  },
  "type": "application",
  "subType": "Interface",
  "event": "DnsUpdate",
  "detailText": "addRecord {u'record': u'CNAME', u'domain': u'cos5.cisco.com', u'cname': u'mgmt-docserver', u'host': u'pm-172-22-98-59-mos'} Completed.
  "eventTime": "2015-07-02T18:01:03.384Z",
  "eventsDropped": 0,
  "severity": "info",
  "location": {
    "ipAddr": "172.22.98.59",
    "processName": "",
    "processId": ""
  }
},

"id": "smregion_0_events.smevent.pam-4e877280-20e4-11e5-8fb7-d5b686854e59",
"name": "pam-4e877280-20e4-11e5-8fb7-d5b686854e59",
"type": "events",
"externalId": "/v2/regions/region-0/events/pam-4e877280-20e4-11e5-8fb7-d5b686854e59",
"transactionId": "da0a327d-6882-4e8e-9262-74ff0a473d26",
"properties": {
  "source": {
    "imgTag": "pam",
    "personality": "control",
    "node": "172.22.98.59"
  },
  "type": "application",
  "subType": "Interface",
  "event": "DnsUpdate",
  "detailText": "addRecord {u'record': u'CNAME', u'domain': u'cos5.cisco.com', u'cname': u'service-mgr', u'host': u'pm-172-22-98-59-mos'} Completed.
  "eventTime": "2015-07-02T18:01:04.671Z",
  "eventsDropped": 0,
  "severity": "info",
  "location": {
    "ipAddr": "172.22.98.59",
    "processName": "",
    "processId": ""
  }
},

"id": "smregion_0_events.smevent.pam-4e9729f0-20e4-11e5-8fb7-d5b686854e59",
"name": "pam-4e9729f0-20e4-11e5-8fb7-d5b686854e59",
"type": "events",
"externalId": "/v2/regions/region-0/events/pam-4e9729f0-20e4-11e5-8fb7-d5b686854e59",
"transactionId": "f84ead29-cc5f-47db-9ed3-5fb3d968cb5",
"properties": {
  "source": {
    "imgTag": "pam",
    "personality": "control",
    "node": "172.22.98.59"
  },
Viewing Events for Nodes

"type": "ext-resource",
$subType": "Accessibility",
"event": "ntpServerPing",
"detailText": "Ntp server found",
"eventTime": "2015-07-02T18:01:04.775Z",
"eventsDropped": 0,
"severity": "info",
"location": {
  "ipAddr": "172.22.98.59",
  "processName": "",
  "processId": ""
}
}
{
  "id": "smregion_0_events.smevent.pam-4eb03030-20e4-11e5-8fb7-d5b686854e59",
  "name": "pam-4eb03030-20e4-11e5-8fb7-d5b686854e59",
  "type": "events",
  "externalId": "/v2/regions/region-0/events/pam-4eb03030-20e4-11e5-8fb7-d5b686854e59",
  "transactionId": "aa1dfde4-f3fd-40f6-a7a9-3a7a31ce14ca",
  "properties": {
    "source": {
      "imgTag": "pam",
      "personality": "control",
      "node": "172.22.98.59"
    },
    "type": "application",
    "subType": "Interface",
    "event": "DnsUpdate",
    "detailText": "addRecord {u'record': u'CNAME', u'domain': u'cos5.cisco.com', u'cname': u'ui', u'host': u'pm-172-22-98-59-mos'} Completed.",
    "eventTime": "2015-07-02T18:01:04.941Z",
    "eventsDropped": 0,
    "severity": "info",
    "location": {
      "ipAddr": "172.22.98.59",
      "processName": "",
      "processId": ""
    }
  }
}
{
  "id": "smregion_0_events.smevent.pam-4ed98a20-20e4-11e5-8fb7-d5b686854e59",
  "name": "pam-4ed98a20-20e4-11e5-8fb7-d5b686854e59",
  "type": "events",
  "externalId": "/v2/regions/region-0/events/pam-4ed98a20-20e4-11e5-8fb7-d5b686854e59",
  "transactionId": "616325ee-1199-41c8-8850-6660ba0d79e0",
  "properties": {
    "source": {
      "imgTag": "pam",
      "personality": "control",
      "node": "172.22.98.59"
    },
    "type": "application",
    "subType": "Interface",
    "event": "DnsUpdate",
    "detailText": "addRecord {u'record': u'CNAME', u'domain': u'cos5.cisco.com', u'cname': u'controller', u'host': u'pm-172-22-98-59-mos'} Completed.",
    "eventTime": "2015-07-02T18:01:05.213Z",
    "eventsDropped": 0,
    "severity": "info",
    "location": {
Viewing the Events for Nodes in a Service Instance

This command returns a list of events for all COS Node objects in the specified Service Instance.

**Syntax**

GET https://SM_FQDN:8043/v2/serviceinstances/serviceinstance_name/events

**Example**

https://172.22.98.59:8043/v2/serviceinstances/cos-0-1/events

```json
{
  "id": "smserviceinstance_cos_0_1_events.smevent.cos-aic-704fde90-1f52-11e5-8897-8b378489ac7c",
  "name": "cos-aic-704fde90-1f52-11e5-8897-8b378489ac7c",
  "type": "events",
  "externalId": "/v2/serviceinstances/cos-0-1/events/cos-aic-704fde90-1f52-11e5-8897-8b378489ac7c",
  "transactionId": "796641dc-5dfe-40c5-b39d-2944d9db0089",
  "properties": {
    "source": {
      "imgTag": "cos-aic",
      "personality": "linux",
      "node": "localhost.localdomain.localdomain"
    },
    "type": "application",
    "subType": "Interface",
    "event": "DnsUpdate",
    "detailText": "addRecord {u'record': u'CNAME', u'domain': u'cos5.cisco.com', u'cname': u'pam-docserver', u'host': u'pm-172-22-98-59-mos'} Completed."
  }
}
```
"subType": "Health",
"event": "COS-AIC Started",
"detailText": "COS-AIC has started as leader on localhost.localdomain.localdomain and connects to docserver: mgmt-docserver.cos5.cisco.com",
"eventTime": "2015-06-30T18:04:23.672Z",
"eventsDropped": 0,
"severity": "info",
"location": {
  "ipAddr": "127.0.0.1",
  "processName": "",
  "processId": "3657"
}
},
{
  "id": "smserviceinstance_cos_0_1_events.smevent.cos-aic-4f4cbe50-20e4-11e5-9520-b7dd993deafa",
  "name": "cos-aic-4f4cbe50-20e4-11e5-9520-b7dd993deafa",
  "type": "events",
  "externalId": "/v2/serviceinstances/cos-0-1/events/cos-aic-4f4cbe50-20e4-11e5-9520-b7dd993deafa",
  "transactionId": "55b34dbf-0906-4d68-9a2e-62e306e3b28f",
  "properties": {
    "source": {
      "imgTag": "cos-aic",
      "personality": "linux",
      "node": "PAM-build-15362.cos5.cisco.com"
    },
    "type": "application",
    "subType": "Health",
    "event": "COS-AIC Started",
    "detailText": "COS-AIC has started as leader on PAM-build-15362.cos5.cisco.com and connects to docserver: mgmt-docserver.cos5.cisco.com",
    "eventTime": "2015-07-02T18:01:05.972Z",
    "eventsDropped": 0,
    "severity": "info",
    "location": {
      "ipAddr": "127.0.0.1",
      "processName": "",
      "processId": "3856"
    }
  }
},
{
  "id": "smserviceinstance_cos_0_1_events.smevent.cos-aic-bd8dec70-20e6-11e5-9520-b7dd993deafa",
  "name": "cos-aic-bd8dec70-20e6-11e5-9520-b7dd993deafa",
  "type": "events",
  "externalId": "/v2/serviceinstances/cos-0-1/events/cos-aic-bd8dec70-20e6-11e5-9520-b7dd993deafa",
  "transactionId": "9669f607-f8ec-4a97-bcb5-62b954edbff2",
  "properties": {
    "type": "cos-node",
    "subType": "Accessibility",
    "event": "AddCosNode",
    "detailText": "New COS-Node added",
    "eventTime": "2015-07-02T18:18:29.943Z",
    "eventsDropped": 0,
    "severity": "info",
    "location": {
      "ipAddr": "172.22.99.127",
      "processName": "",
      "processId": "3856"
    }
  }
}
Event Query Filters

Events can be far too large to return in one query. For this reason, when all events are requested, the query returns only the events that occurred during the past five minutes.

In the request, you can filter the events to return based on specific time range, severity, and type. Add the following parameters in the query to limit the output:

- **startTime**: Return events that have a timestamp after this time (must be used with endTime)
- **endTime**: Return events that have a timestamp of events before this time (must be used with startTime)
- **severity**: Return events that have the specific severity, based on a string to match. The valid values are:
  - critical
  - major
  - warning
  - info
- **type**: Return events that have the specified type class. The valid type values are:
  - Node
  - Application
  - Ext-Resource
  - High-Availability
  - Config

**Syntax**

GET https://SM_FQDN:8043/v2/serviceinstances/instanceName/events?startTime=2014-03-20T14:00:00.000Z&endTime=2014-03-20T14:10:00.000Z

**Example**

https://172.22.98.59:8043/v2/serviceinstances/cos-0-1/events?startTime=2015-07-06T00:00:00.000Z&endTime=2015-07-06T23:10:00.000Z

```
{
  "id": "smserviceinstance_cos_0_1_events.smevent.cos-6eebc3b0-23d1-11e5-91a8-8fbec6366e30",
  "name": "cos-6eebc3b0-23d1-11e5-91a8-8fbec6366e30",
  "type": "events",
  "externalId": "/v2/serviceinstances/cos-0-1/events/cos-6eebc3b0-23d1-11e5-91a8-8fbec6366e30",
  "transactionId": "609f00c8-984a-4d10-bf72-2fca055aae48",
  "properties": {
    "type": "cos-node",
    "subType": "Health",
    "event": "CosNodeServiceDown",
    "detailText": "Service Cisco Cache Server Down",
    "eventTime": "2015-07-06T11:32:07Z",
```
Viewing Events for Nodes

"eventsDropped": 0,
"severity": "warning",
"location": {
  "ipAddr": "172.22.99.127",
  "processName": "",
  "processId": ""
},
"source": {
  "imgTag": "cos-aic",
  "personality": "client",
  "node": "172.22.99.127"
}

"eventsDropped": 0,
"severity": "warning",
"location": {
  "ipAddr": "172.22.99.127",
  "processName": "",
  "processId": ""
},
"source": {
  "imgTag": "cos-aic",
  "personality": "client",
  "node": "172.22.99.127"
}

"id": "smserviceinstance_cos_0_1_events.smevent.cos-6eed7160-23d1-11e5-91a8-8fbc63666e30",
"name": "cos-6eed7160-23d1-11e5-91a8-8fbc63666e30",
"type": "events",
"externalId": "/v2/serviceinstances/cos-0-1/events/cos-6eed7160-23d1-11e5-91a8-8fbc63666e30",
"transactionId": "3764c89c-8aaa-4c88-bc6f-a9ebb55b0f13",
"properties": {
  "type": "cos-node",
  "subType": "Health",
  "event": "CosNodeServiceDown",
  "detailText": "Service Cassandra Down",
  "eventTime": "2015-07-06T11:23:32.085Z",
  "eventsDropped": 0,
  "severity": "warning",
  "location": {
    "ipAddr": "172.22.99.127",
    "processName": "",
    "processId": ""
  },
  "source": {
    "imgTag": "cos-aic",
    "personality": "client",
    "node": "172.22.99.127"
  }
}

"id": "smserviceinstance_cos_0_1_events.smevent.cos-7621ece0-23d1-11e5-91a8-8fbc63666e30",
"name": "cos-7621ece0-23d1-11e5-91a8-8fbc63666e30",
"type": "events",
"externalId": "/v2/serviceinstances/cos-0-1/events/cos-7621ece0-23d1-11e5-91a8-8fbc63666e30",
"transactionId": "0a845bb5-9b73-4e27-a909-c5f0eda24f3",
"properties": {
  "type": "cos-node",
  "subType": "Health",
  "event": "CosNodeServiceUp",
  "detailText": "Service Cassandra Up",
  "eventTime": "2015-07-06T11:23:44.173Z",
  "eventsDropped": 0,
  "severity": "info",
  "location": {
    "ipAddr": "172.22.99.127",
    "processName": "",
    "processId": ""
  },
  "source": {
    "imgTag": "cos-aic",
    "personality": "client",
    "node": "172.22.99.127"
  }
}
Viewing Application Status for the COS Service Instance

AppStatus is a read-only object that provides a view into the application status for each endpoint in a service instance.

Every active service instance contains one or more applications for each endpoint in the Service Instance object. Via the AppStatus object, applications report their status with regard to meeting the service level agreement (SLA) defined for the endpoint. Each AppStatus object includes a summary of the nodes it is using as well as the number of open events and their severity (critical, major, and warning).

COS Service Endpoint

This command returns the status of the specified COS Service Endpoint object.

Syntax

GET https://SM_FQDN:8043/v2/serviceinstances/cosinstance_name/appstatuses/endpoint_name

Example


{   "id": "smserviceinstance_cos_0_1_status.smappstatus.EP1",   "name": "EP1",   "type": "appstatuses",   "externalId": "/v2/serviceinstances/cos-0-1/appstatuses/EP1",   "transactionId": "b77c5350-b27d-4a81-8150-3348c549cf40",   "properties": {     "appName": "cos",     "endpointRef": "smserviceinstance_cos_0_1.smcosendpoint.EP1",     "slaType": "resource",     "slaStatus": {       "nodesInUse": 0,       "bandwidthStatus": "unavailable",       "sessionsStatus": "unavailable",       "storageStatus": "unavailable",       "nodeStatus": "critical",       "usageStatus": {         "storageUsage": {           "used": 0,           "total": 0         },         "bandwidthUsage": {           "rx": {             "used": 0,             "total": 0           },           "tx": {             "used": 0,             "total": 0           }         }     }   } }


```
},
"sessionUsage": {
  "rx": {
    "used": 0,
    "total": 0
  },
  "tx": {
    "used": 0,
    "total": 0
  }
},
"nodeUsage": {
  "used": 0,
  "total": 0
}
},
"eventsOpen": {
  "critical": 0,
  "major": 0,
  "warning": 0
}
```

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.

**Note**
The Swauth service uses a special built-in user and key to configure the service itself. This user is known as the "super-admin". The default Swauth super-admin user name is "super_admin" and the default super-admin key is "rootroot".

**Listing Accounts**

To retrieve a list of existing accounts for the reseller or super admin, use the following HTTP GET request:

```
GET /auth/v2/ HTTP/1.1
```

**Required Headers**
- X-Auth-Admin-User — the accounts for this particular user are listed in the response.
- X-Auth-Admin-Key

**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
```
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 service accounts.
- The services value is a dict that represents valid storage cluster endpoints, and identifies the default endpoint.
- The users value is a list of dicts, each dict representing a user and currently containing the key name.

**Required Headers**

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

**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
- 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: <service account suffix>
  - When an account is created, a new UUID4 with the reseller prefix form the account ID. Using this header, you can replace the UUID4 part of the ID with the <service account suffix>.
  - The <service account suffix> must not exceed 251 bytes in length.

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
Creating or Updating a User

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 site 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.
Chapter 3  Swauth API

Retrieving User Details

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
- 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>" },
  ]
}
Deleting a User

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

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

### 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 '.'
- **401** – Invalid X-Auth-Admin-User/X-Auth-Admin-Key
- **403** – Retrieval of the requested user barred by admin user
- **404** – Unknown account or user
- **5xx** – Internal error

### Deleting a User

A super admin or a reseller admin can delete users from any account.

Account admins can only delete users from the accounts they administer.
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 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 endpoints will be added to the existing set of services and endpoints, respectively.
- If the service specified exists, new endpoints 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>

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

- X-Auth-Key: <user password>

Response Headers

On successful authentication, the response has the following headers:

- X-Auth-Token
  - This header has the token to be used with Swift APIs used in container and object management.
- X-Storage-URL
  - This header has the URL of the default storage cluster.

Response Body

On successful authentication, the response body is set to the account’s services JSON object as shown below.

```json
{"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": "<URL to use with Swift>"  # "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.
  },
  "servers": {  # Represents the Nova server service end points
    # Expected to be similar to the "storage" dict, but not
    # implemented yet.
  },
  # 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>[?<param>=<value>][&<param>=<value>]] HTTP/1.1

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.
Listing Objects

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

GET /v1/<account>/<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> 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) — Invalid container name
- 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

Deleting a Container

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

DELETE /v1/<account>/<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.
Retrieving an Object

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

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

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) — Invalid container name
- 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
- 409 (Conflict) — The container is not empty

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.
  - A negative delay indicates that 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.
Creating or Updating An Object

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 a retrieval of the data 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. This response header is a COS extension.

**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:

```plaintext
PUT /v1/<account>/<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's name/metadata. If the system is unsuccessful, the Content-Type is set to the default application/octet-stream value.
- Content-Length/ Transfer-Encoding: chunked
Creating or Updating An Object

- 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, must be included.

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 the value of the header is not true, the COS node receiving the request will perform 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 a 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>/<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.
Creating or Updating Container Metadata

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

```
POST /v1/<account>/<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 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>/<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 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>/<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 or container
- 5XX — Internal Server Error

Creating or Updating Object Metadata
To create or update custom object attributes, use the following HTTP POST request:

POST /v1/<account>/<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>
Retrieving Object Metadata

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

`HEAD /v1/<account>/<container>/<object> HTTP/1.0`

**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:

- 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 or container
- 5XX — Internal Server Error
• 404 (Not Found) — The storage URL references a non-existent account 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> 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 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> 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
- 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> 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
Deleting Account Metadata

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

```
POST /v1/<account> 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

**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.
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>/<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
Deleting ACLs

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

```
POST /v1/<account>/<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 or container
- 5xx — Internal Server Error
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; charset=utf-8
< content-length: 2138
< etag: "1249005776"
< date: Fri, 19 Dec 2014 20:57:03 GMT
< connection: close
<
> {  
    "id": "smttenant_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": [
            
        ]
    }
}
Swauth API curl Example

To see a list of accounts:

```
```

To create a container:

```
curl -v -L -X PUT -H "X-Auth-Token: AUTH_tk836935bba053405aa65853863b17b871" -H "X-Container-Read:""X-Container-Write:"" http://auth01.cos1.acme.com/v1/AUTH_msmith/mustang
```

Swift API curl Example

To create a container:

```
curl -v -L -X PUT -H "X-Auth-Token: AUTH_tk836935bba053405aa65853863b17b871" -H "X-Container-Read:""X-Container-Write:"" http://auth01.cos1.acme.com/v1/AUTH_msmith/mustang
```

To see a list of accounts:

```
```
> 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_tka5d2c2898c1e4632854f55cadfccc55f5
>
> < 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_tka5d2c2898c1e4632854f55cadfccc55f5" 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_tka5d2c2898c1e4632854f55cadfccc55f5
>
< 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