Release Notes for COS 2.1.2

OL-31733-02

Last Updated: March 23, 2015

These release notes describe the features and caveats for all releases in the Cisco Cloud Object Store (COS) Release 2.x train.

These release notes are updated with each release in the train. This update adds information for Cisco COS Release 2.1.2. For a list of the caveats that apply to this release, see the "Caveats" section on page 4.

Contents

The following information is in the release notes:

Cisco Systems, Inc.
www.cisco.com

Introduction

The Cisco Cloud Object Store (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.

Object 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.

COS includes COS Service Manager (SM), which provides a web-based GUI as well as related ReST APIs, to simplify setup and management. COS also provides a command-line interface (CLI) for remote or programmatic content management. COS also includes an authentication and authorization service that implements the OpenStack Swauth API.

Through its various management interfaces, COS provides access to large media objects while maintaining high quality of service, supports cluster management, and coordinates the replication of data across sites to improve resiliency and optimize the physical location of stored data.

Related Products

COS 2.1.2 can be implemented as a managed service of Cisco Media Origination System (MOS) 2.3. In this configuration, COS content is managed through the MOS SM GUI.

COS 2.1.2 also works together with an upcoming Cisco TV VDS release to provide cloud storage for recorded video programming.

Feature Overview

The table below provides an overview of the COS features. For full descriptions of these features, see the *Cisco Cloud Object Store Release 2.1.1 User Guide*.

Table 1-1 Overview of COS Features

Feature Set	Features
COS Service Manager GUI	• Lets you quickly and easily access many COS deployment and monitoring functions
High Availability (HA)	• COS now supports HA as implemented in MOS
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

Table 1-1 Overview of COS Features

Feature Set	Features
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
Object Store Metadata Resiliency	• Metadata resiliency is provided by a distributed and replicated Cassandra document database • Each COS node participates in the persistence of a subset of the Cassandra database • Manual administrative intervention required upon node failure
Object Store Data Resiliency	• Data is resilient to both hard drive and COS node failures • Local COS node data resiliency provided by local software RAID • COS cluster data resiliency provided by object replication
Service Load Balancing	• COS cluster load balancing is provided by DNS round-robin of a FQDN to multiple physical IPv4 addresses hosted by COS nodes • Optimal load balancing is provided by extensions to the Swift API through the implementation of HTTP redirect

Unsupported Features

The following features are not supported in COS 2.1.2, but are under consideration for future releases:

- Support for multiple-site cluster management
- Support for node failover
- Support for Server RAID
- Support for IPv6

The OpenStack SWIFT and SWAuth APIs continue to evolve. COS does not current implement a full complement of SWIFT or SWAuth API functions. For a list of currently supported functions, see the *Cisco Cloud Object Store Release 2.1.1 User Guide*.

COS 2.1.2 does not support automatic failover of Cassandra working sets in the event of COS node failure. Manual administrative action is required to recover a lost COS node in the event that a COS node cannot be returned to service in a timely manner.

Hardware Support

COS 2.1.2 supports the following hardware as COS storage cluster nodes:

- 460-4R1: CDE-460 with 36 x 3 TB hard drives
- 460-4R3: CDE-460 with 36 x 4 TB hard drives
- 470-4R2: CDE-470 with 72 x 4 TB hard drives

For hardware installation instructions and related details, see the Cisco Content Delivery Engine 205/220/250/420/460/470 Hardware Installation Guide.

Installation

The CDE-460 and CDE-470 appliances used for COS ship with the COS software pre-installed but not configured. The installation software is an .iso file that includes the base (CentOS) distribution of Linux along with all of the additional rpm packages needed by a COS node.

The same .iso file can be used to reinstall or upgrade the software on an existing COS node. For additional details, see the *Cisco Cloud Object Store Release 2.1.1 User Guide*.

Supported Environments

COS 2.1.2 supports a Swift/Swauth API environment, and also supports an HTTP-based API for cluster management.

System Requirements

COS 2.1.2 can operate as a managed service of MOS 2.3, in which case it uses certain MOS HTTP interface components as well as the MOS Document Store for system management. See the MOS 2.3 documentation for MOS system requirements.

Caveats

Caveats describe unexpected behavior in COS software releases. Severity 1 caveats are the most serious caveats; severity 2 caveats are less serious. Severity 3 caveats are moderate caveats, and only selected severity 3 caveats are included in the caveats document.

Caveat numbers and brief descriptions for Cisco COS Release 2.1.2 releases are listed in this section.

Open Caveats

Open Caveats for Cisco COS Release 2.1.2

Table 2 lists the open issues in the COS 2.1.2 release.

Bug details are displayed in the Bug Search.

Table 2 Open Caveats in COS 2.1.2 Release

Bug ID	Description
CSCur84487	Mackenzie: Get token failed while COS is rebooting and recovering.
CSCur95716	Gossip issue when COS node is added to cluster.

Resolved Caveats

Resolved Caveats for Cisco COS Release 2.1.2

Table 3 lists the fixed issues in the COS 2.1.2 release.

Bug details are displayed in the Bug Search.

Table 3 Resolved Caveats in COS 2.1.2 Release

Bug ID	Description
CSCur99109	Bootstrapping issue when starting up the Cassandra service.
CSCus08986	211B6: cosd dead after post-initial-setup reboot.
CSCus09019	One node http 500 error after cassandra restart in both nodes.
CSCus97596	Add tuning to TCP keep-alive behavior.
CSCus93929	Handle overlapping SWIFT Create-Objects requests.
CSCur66747	prodassert.log: CALYPSO ASSERTs in HttpAdapter.cpp multiple lines.
CSCus61904	GET for zero length live object returning 404 instead of 204.
CSCus48764	Large object PUT with chunked encoding being throttled.

Accessing Bug Search Tool

This section explains how to use the Bug Search tool to search for a specific bug or to search for all bugs in a release.

Step 1 Go to https://tools.cisco.com/bugsearch/.

Step 2 At the Log In screen, enter your registered Cisco.com username and password; then, click **Log In**. The Bug Search page opens.

Note If you do not have a Cisco.com username and password, you can register for them at http://tools.cisco.com/RPF/register/register.do.

Step 3 To search for a specific bug, enter the bug ID in the Search For field, and press **Enter**.

Step 4 To search for bugs in the current release, specify the following criteria:

- Select the **Model/SW Family** Product Category drop-down list box, then enter **Cisco Videoscape Distribution Suite for Television** or select the name from the **Select from list** option.
- Select **Cisco Videoscape Distribution Suite for Television** from the list that displays.
- The **Cloud Object Store** type displays in the Software Type drop-down list box.
- Releases: 2.1.2.
- Advanced Filter Options—Define custom criteria for an advanced search by selecting an appropriate value from the drop-down lists by choosing either one Filter or multiple filters from the available categories. After each selection, the results page will automatically load below the filters pane. If you select multiple filters, it behaves like an AND condition.
  - Modified Date—Select one of these options to filter bugs: **Last Week**, **Last 30 days**, **Last 6 months**, **Last year**, or **All**.
  - Status—Select **Fixed**, **Open**, **Other**, or **Terminated**.

    Select **Fixed** to view fixed bugs. To filter fixed bugs, uncheck the Fixed check box and select the appropriate suboption (Resolved or Verified) that appears below the Fixed check box.

    Select **Open** to view all open bugs. To filter the open bugs, uncheck the Open check box and select the appropriate suboptions that appear below the Open check box.

    Select **Other** to view any bugs that are duplicates of another bug.

    Select **Terminated** to view terminated bugs. To filter terminated bugs, uncheck the Terminated check box and select the appropriate suboption (Closed, Junked, or Unreproducible) that appears below the Terminated check box. Select multiple options as required.
  - Severity—Select the severity level:

    1: Catastrophic.

    2: Severe

    3: Moderate

    4: Minor

    5: Cosmetic

    6: Enhancement
  - Rating—Select the bug's quality rating: **5 Stars** (excellent), **4 or more Stars** (good), **3 or more Stars** (medium), **2 or more Stars** (moderate), **1 or more Stars** (poor), or **No Stars**.
  - Support Cases—Select whether the bug **Has Support Cases** or **No Support Cases**.
  - Bug Type—Select whether the bug is **Employee Visible & Customer Visible** or **Customer Visible Only**.

Step 5 The Bug Toolkit displays the list of bugs based on the specified search criteria.

Step 6 You can save or email the current search by clicking their respective option.

If you have any problems using the Bug Search tool, log into the Technical Support website at http://www.cisco.com/cisco/web/support/index.html or contact the Cisco Technical Assistance Center (TAC).

Upgrading to Cisco COS Release 2.1.2

The COS software deployment model uses the YUM package manager for installation, upgrade, and downgrade. The following sequence of steps must be executed on each COS node by the root user.

To upgrade the COS software to Release 2.1.2 from an earlier COS release, follow these steps:

Step 1 Copy the COS 2.1.2 software ISO image to /root on each target COS node.

Step 2 Mount the copied cos_repo.iso file.

```
[root@cos-node ~]# mount -o loop /root/cos_repo-2.1.2.iso /mnt/cdrom
```

Step 3 Prepare the YUM repository as follows:

```
[root@cos-node ~]# rm -f /etc/yum.repos.d/*.repo
[root@cos-node ~]# /mnt/cdrom/local_repo_setup
[root@cos-node ~]# yum clean all
```

Step 4 Upgrade the COS software as follows:

```
[root@cos-node ~]# yum -y groupinstall cos cos_support cos_test
[root@cos-node ~]# yum -y upgrade
[root@cos-node ~]# reboot
```

The system reboots and returns to service running the upgraded version of COS software.

Downgrading from Cisco COS Release 2.1.2

To downgrade the COS software, you must first uninstall the current software version, and then install the desired final software version. The following sequence of steps must be executed on each COS node by the root user.

To downgrade the COS software from Release 2.1.2 to an earlier COS release, follow these steps:

Step 1 Copy the COS 2.1.1 software ISO image to /root on each target COS node.

Step 2 Mount the ISO image to /mnt/cdrom as follows:

```
[root@cos-node ~]# mount -o loop /root/cos_repo-2.1.1.iso /mnt/cdrom
```

Step 3 Uninstall the existing COS software version as follows:

```
[root@cos-node ~]# yum -y groupremove cos cos_support cos_test
```

Step 4 Install the target COS software version as follows:

```
[root@cos-node ~]# rm -f /etc/yum.repos.d/*.repo
[root@cos-node ~]# /mnt/cdrom/local_repo_setup
[root@cos-node ~]# yum clean all
[root@cos-node ~]# yum -y install SuperDoctor
[root@cos-node ~]# rpm -e kernel-vds
[root@cos-node ~]# yum -y groupinstall cos cos_support cos_test
```

Step 5 Restore the COS service configuration as follows, replacing **<date>** in the file extension for the cosd.conf file with the most recent date:

```
[root@cos-node ~]# /bin/cp -f /opt/cisco/cos/etc/cosd.conf-<date>
/opt/cisco/cos/etc/cosd.conf
```

```
[root@cos-node ~]# /bin/cp –f /etc/cassandra/conf/cassandra.yaml.rpmsave
/etc/cassandra/conf/cassandra.yaml
[root@cos-node ~]# touch /opt/cisco/cos/config/cosinit_executed
[root@cos-node ~]# chkconfig syslog-ng on
[root@cos-node ~]# chkconfig cosd on
[root@cos-node ~]# chkconfig cserver on
[root@cos-node ~]# chkconfig monit on
[root@cos-node ~]# chkconfig cos_aicc on
[root@cos-node ~]# reboot
```

The system reboots and returns to service running the downgraded version of COS software.

Tuning the COS Network Stack

The COS software implements a proprietary TCP stack designed with an emphasis on performance and real-time characteristics. This section describes the tunable parameters that can be used to optimize performance in a specific deployment.

Keep-Alive Probes

The COS TCP stack was originally implemented for the delivery of video in QAM and IP set-top environments in which the viewing experience depends upon resource commitments, minimal buffering, and tight network delivery constraints. By design, the TCP stack is very aggressive in detecting session failures, as this allows for quick recovery with minimal impact on the viewing experience.

The detection of session failure is achieved using standard TCP keep-alive probes. A TCP keep-alive probe is sent to solicit an acknowledgment packet from the receiving side as a confirmation that the connection is still open.

Prior to COS 2.1.2, the probing behavior was to start sending keep-alive probes to a client after 125 milliseconds of idle time on a TCP session. If the client failed to respond to the probes, the TCP stack declared a connection dead after a timeout period (500 milliseconds by default) defined by */proc/calypso/tunables/tcp_timeout_ms*.

Beginning with COS 2.1.2, because the TCP stack is now used for non-video applications and for communication with clients that are not as responsive as real-time applications, it is necessary to modify the keep-alive probing behavior so that it is more lenient. The new default behavior is to start sending keep-alive probes after 60 seconds of idle time on the session, sending one probe per second. The session is declared dead after 10 unacknowledged probes are sent. In practice, this means that a truly dead connection can be detected after 70 seconds.

COS 2.1.2 introduces three new tunable parameters that alter the behavior of the TCP keep-alive probes:

- *tcp_keepalive_time_ms*
- *tcp_keepalive_intvl_ms*
- *tcp_keepalive_probes*

These tunable parameters are found on the COS nodes under /proc/calypso/tunables. The current values can be obtained by reading the file using cat or a similar utility. Conversely, writing a new value to the respective file can change the parameter.

Note The value for both reading and writing is a hexadecimal encoded number.

These tunable parameters are a replacement for using *tcp_timeout_ms*. With the addition of these three tunable parameters, we no longer recommend changing the *tcp_timeout_ms* parameter from the default value of 500 milliseconds, as this can affect other areas of operation, including the timing of TCP state machine execution.

tcp_keepalive_time_ms

This tunable adjusts the amount of time in milliseconds that a TCP session is idle before keep-alive probes are sent. The default value is 0xEA60 (60 seconds), with minimum and maximum values of 0x7D (125 milliseconds) and 0x36EE80 (60 minutes), respectively.

tcp_keepalive_intvl_ms

This tunable adjusts the time interval in milliseconds between keep-alive probes. The default value is 0x3e8 (1 second), with minimum and maximum values of 0x7D (125 milliseconds) and 0x2710 (10 seconds), respectively.

tcp_keepalive_probes

This tunable adjusts the number of unacknowledged probes that are sent before the TCP stack times out, declares a connection dead, and resets the connection. The default value is 0xa (10 probes), with minimum and maximum values of of 0x1 (1 probe) and 0x3C (60 probes), respectively.

TCP Window Advancement

The COS TCP stack has also been aggressive in ensuring that data is able to be transmitted from COS. TCP is a sliding-window protocol in which the TCP transmit window is advanced by the client. If the TCP window does not advance, COS cannot transmit additional data.

The TCP stack implements a mechanism to time out and reset a connection if the TCP window does not advance for a defined time period. This timeout differs for committed-rate and best-effort delivery profiles. Prior to COS 2.1.2, the timeout interval for committed rate was defined by the *tcp_timeout_ms* tunable which, by default, is 500 milliseconds. For COS 2.1.2, a new *tcp_window_timeout_ms* tunable is introduced to allow for defining this window advancement timeout for committed-rate delivery profiles independent of the *tcp_timeout_ms* tunable.

tcp_window_timeout_ms

This new tunable, located in the /proc/calypso/tunables directory on a COS node, defines the timeout for committed rate in milliseconds. The default value of this tunable is 0xFA0 (4 seconds), with minimum and maximum values of 0x1F4 (500 milliseconds) and 0xEA60 (60 seconds), respectively.

http_best_effort_transfer_window_timeout

This previously existing tunable, located in /proc/calypso/tunables directory on a COS node, adjusts the window timeout in milliseconds for transfers using best-effort delivery profiles. The default value is 0xFA0 (4 seconds), with minimum and maximum values of 0x1F4 (500 milliseconds) and 0xEA60 (60 seconds), respectively.

Concurrent TCP / HTTP Sessions

At the most basic level, each physical network data interface on a COS node supports a maximum of 16,384 TCP sessions. This value is not adjustable, and additional constraints imposed by the software limit the number of concurrent HTTP requests being processed on a single COS node.

http_max_concurrent_requests

This tunable limits the number of concurrent HTTP requests being processed by a single COS node. Some requests are long-lived, such as writing data for a linear capture of video. If the maximum number of concurrent requests is reached, the COS node responds with a 503 Service Unavailable status to additional HTTP requests. The default value for this tunable is 0x4E20 (20,000), with minimum and maximum values of 0x0 (0) and 0x9C40 (40,000), respectively.

httpBestEffortMaxReads

This tunable limits the number of concurrent HTTP read requests using the best-effort delivery profile. The default value is 0x2710 (10,000), with minimum and maximum values of 0x0 (0) and 0x4e20 (20,000), respectively.

Persisting Tunable Adjustments

The file /arroyo/test/CalypsoTunables is used to persist tunable adjustments across a reboot of a COS node. This file must be created manually. The contents of this file are applied upon starting the COS services.

Each line of the file must have the following syntax:

<relative-file-path> <hexadecimal-value>

The file path is relative to the /proc/calypso directory, and is not a full file system path. The following is an example of the syntax of the file.

```
tunables/tcp_keepalive_time_ms 0x2710
tunables/tcp_keepalive_intvl_ms 0x1f4
tunables/tcp_keepalive_probes 0x5
```

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 delivers content directly to your desktop using a reader application. The RSS feeds are a free service.

Related Documentation

Refer to the following documents for additional information about Cisco COS 2.1.2:

- *Cisco Cloud Object Store Release 2.1.1 User Guide*
- *Cisco Content Delivery Engine 205/220/250/420/460/470 Hardware Installation Guide*
- *Regulatory Compliance and Safety Information for Cisco Content Delivery Engines*
- *Open Source Used in Cisco COS 2.1.1*

  http://www.cisco.com/en/US/products/ps12653/products_licensing_information_listing.html

The entire VDS-TV software documentation suite is available on Cisco.com at:

http://www.cisco.com/en/US/products/ps12653/tsd_products_support_series_home.html

The entire VDS hardware documentation suite is available on Cisco.com at:

http://www.cisco.com/en/US/products/ps7126/tsd_products_support_series_home.html