Cisco Cloud Object Storage Release 3.12.1 Troubleshooting Guide

January 17, 2017

Cisco Systems, Inc.
www.cisco.com

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

Cisco Cloud Object Storage Release 3.12.1 Troubleshooting Guide

CONTENTS

Preface

The *Cisco Cloud Object Storage Release 3.12.1 Troubleshooting Guide* provides information on troubleshooting the Cisco Cloud Object Store management and service components. It also provides a brief introduction to the architecture and control flow among the various components of the COS deployment and addresses common troubleshooting scenarios.

This preface describes who should read the *Cisco Cloud Object Storage Release 3.12.1 Troubleshooting Guide*, how it is organized, and its document conventions. It contains the following sections:

Audience

This guide is for the networking professional managing the Cisco Cloud Object Storage (COS) product. Before using this guide, you should have experience working with Linux platforms, and be familiar with the concepts and terminology of Ethernet, local area networking, clustering and high-availability, and network services like DNS and NTP.

This document provides troubleshooting tips for a Cisco Cloud Object Store deployment, including management and service components. It provides a brief introduction to the architecture and control flow among the various components of the COS deployment and addresses some common troubleshooting scenarios.

Document Organization

This document contains the following chapters and appendices:

Chapters or Appendices	Descriptions
Chapter 1, “COS System Overview”	Provides an overview of the Cisco Cloud Object Storage (COS) system.
Chapter 2, “Troubleshooting the COS Configuration”	Provides information and procedures for troubleshooting COS configuration problems.
Chapter 3, “Using Logs and Monitoring to Troubleshoot COS Runtime Issues”	Describes how to use logs and monitoring to troubleshoot COS runtime issues.
Chapter 4, “Troubleshooting Issues Across the COS System”	Provides information and procedures for troubleshooting COS issues that are systemwide.
Chapter 5, “Troubleshooting Swift and Swauth API Errors”	Provides information and procedures for troubleshooting Swift and Swauth API errors.

Document Conventions

This document uses the following conventions:

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

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

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

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

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

Warning

IMPORTANT SAFETY INSTRUCTIONS

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

SAVE THESE INSTRUCTIONS

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

Related Publications

Refer to the following documents for additional information about COS:

- Release Notes for Cisco Cloud Object Storage
- *Cisco UCS C3160 Rack Server Installation and Service Guide*
- *Cisco Content Delivery Engine 465 Hardware Installation Guide*
- *Cisco Content Delivery Engine 205/220/250/280/420/460/470 Hardware Installation Guide*
- *Cisco Cloud Object Storage API Guide*
- *Cisco Cloud Object Storage User Guide*
- *Open Source Used in COS*

These documents are available from the following location:
http://www.cisco.com/c/en/us/support/video/cloud-object-storage/tsd-products-support-series-home.html:

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.

COS System Overview

This chapter provides an overview of the Cisco Cloud Object Storage (COS) system. Cisco COS provides distributed, resilient, high-performance storage and retrieval of binary large object (blob) data. 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 a cluster. Nodes can be added to or removed from the cluster to adjust cluster capacity as needed.

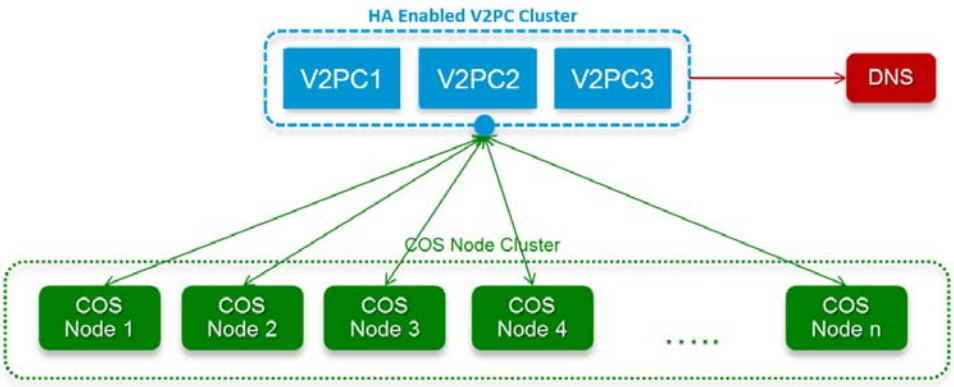
The underlying interface for managing content is the OpenStack Swift API, with enhancements to improve quality of service when accessing large media objects. includes an authentication and authorization service that implements the OpenStack Swauth API. To administer the cluster, COS includes an HTTP-based cluster-management API.

A typical COS deployment is comprised of a cluster of COS nodes that are managed by a COS Management component. The COS Management component can be configured as a single virtual node or as a cluster of three virtual nodes configured in a High Availability (HA) setup.

Beginning with COS Release 3.12.1, COS is installed as a service of Cisco Virtualized Video Processing Controller (V2PC). V2PC provides the common management interface for COS, VMR, and other applications that together form a complete virtualized media processing solution. Prior to COS Release 3.12.1, the COS system used the Platform Access Manager (PAM) as the COS Management component.

Figure 1-1 provides an example of an HA setup.

Figure 1-1 HA Deployment



COS Components

has a number of subsystems.

- **Networks:** Interfaces are grouped into distinct networks to isolate management functions from high-volume data traffic.
- **Clusters and Nodes:** 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 nodes in a cluster.
- **Virtualized Video Processing Controller (V2PC):** COS 3.12.1 components are managed using services running on the V2PC.
- **Hardware Platforms:** software is currently deployed on selected Cisco Content Delivery Engine (CDE) and Cisco UCS server hardware models.

The following sections further describe each of these components.

Networks

COS divides network interfaces into two groups: the data network and the management network. The management network is used to monitor and manage COS clusters and individual COS nodes. The data network is used by client applications to interact with the COS authentication and authorization services, and the COS object storage services. Client applications use the Swauth API to interact with the COS authentication and authorization services, and the Swift API to interact with the COS object storage services.

Similarly, the COS installation separates its traffic into data and management traffic, and expects these two types of traffic to be isolated into their own subnets. COS management traffic on 1G management adapters can be combined with other traffic not intended for the COS system. However, COS data traffic on 10G adapters should be on a managed subnet that does not permit traffic not intended for COS. If non-COS traffic is allowed on a COS data subnet, it will degrade system performance and can cause availability issues.

COS Nodes

The COS software runs on a collection of computing systems called nodes, which are connected via the management and data networks. Currently, there are two types of COS nodes: the cluster controller and the storage nodes.

The storage nodes host software that manages object-store and authentication and authorization service metadata, stores and retrieves object contents, and communicates with the cluster controller. COS storage nodes can be added or removed without disrupting COS service availability. Adding nodes is a way of elastically increasing the storage and bandwidth capacity of the COS cluster.

The COS node software includes a customized Linux distribution, currently based on CentOS 6. This provides the basic framework for the other software applications and modules that run on the node. Each node runs a set of kernel modules and a number of daemons that run in the Linux user-space.

The kernel modules:

- Support real-time management of node hardware resources.
- Provide the distributed, resilient content-store used for object-store data.

- Provide the Swift and Swauth API support via the data network.

The daemons:

- Coordinate service log files.
- Communicate with the cluster controller.
- Provide a distributed database for object-store metadata.
- Communicate with the modules running in the kernel.

While the data-network interfaces communicate directly with the kernel modules, the management network interfaces communicate directly with the user-space daemons.

COS Cluster

A COS cluster is a group of bare-metal COS storage nodes (machines) that use a common replication and redundancy policy. The COS cluster provides a single service endpoint URL that uses APIs to render the COS service. The nodes in the cluster are connected by both data and management networks. COS Release 3.12.1 supports one cluster per V2PC deployment. Each cluster has a single fully-qualified domain name (FQDN) that is used by client applications to access COS services.

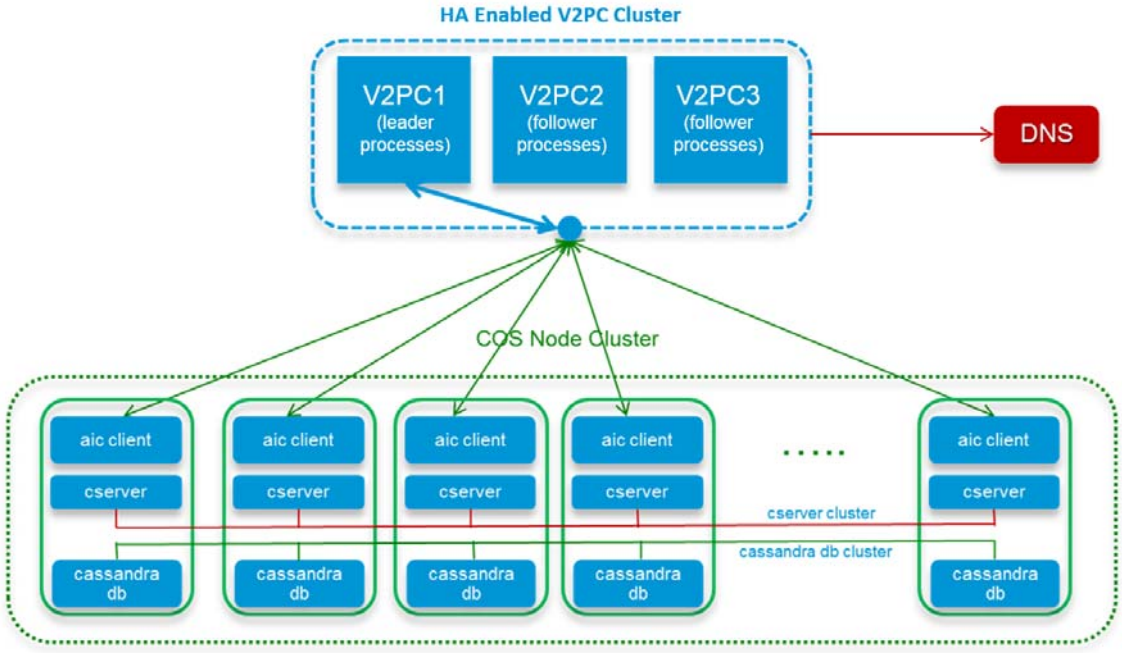
The COS cluster has two main functions:

- **Metadata storage:** Metadata is a description of a piece of content such as its name, size, and other related attributes of the content. A cluster of Cassandra database instances handle the metadata storage. Each COS node runs an instance of the Cassandra database, which is part of a Cassandra cluster that extends within the COS cluster.
- **Content storage:** The cserver component handles the content storage. cserver instances form their own cluster and like the Cassandra database cluster, use cluster level communication that is unique and restricted to the cserver instances.

Each COS node also runs an instance of an Application Instance Controller (AIC) client, which talks to the AIC. The AIC clients are not clustered in the COS cluster.

Figure 1-2 shows the clustering configuration for the COS cluster.

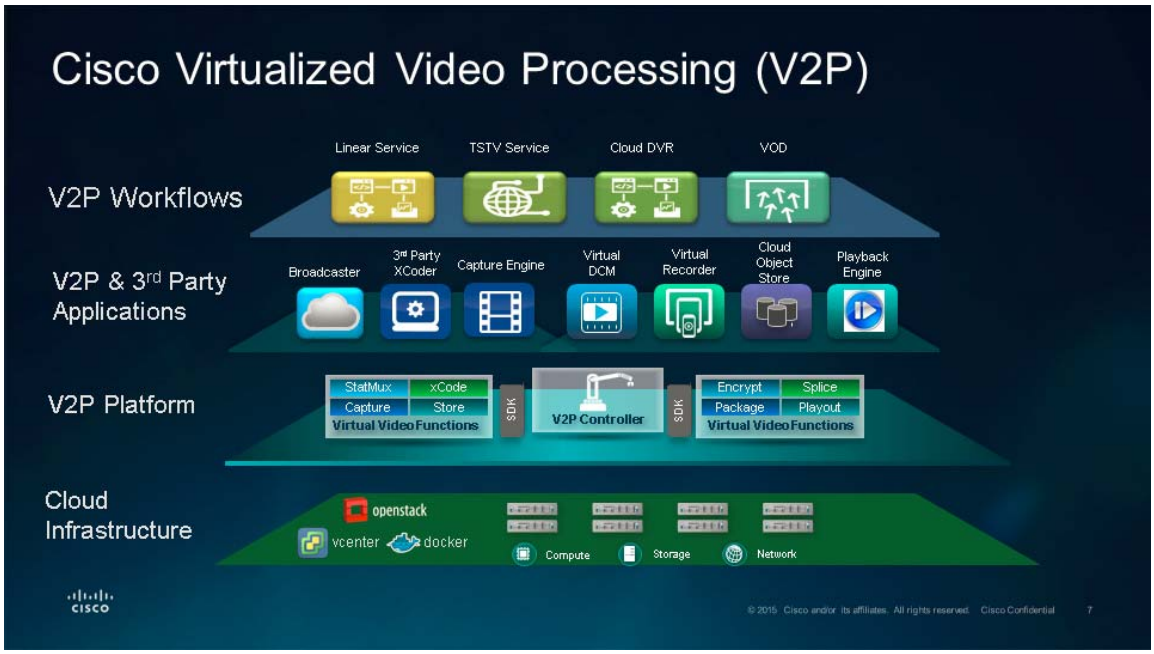
Figure 1-2 Clustering Configuration



COS and V2PC

V2PC is the control interface for the Cisco Virtualized Video Platform (V2P), an open platform that transforms the way video infrastructure is built, deployed, provisioned, and maintained. V2PC enables a video processing application to run over a cloud or on-premise infrastructure while flexibly orchestrating its media workflows and resources. COS integrates transparently with V2PC, and can be managed through the V2PC graphical user interface (GUI) web application.

Figure 1-3 Cisco Virtualized Video Processing (V2P) Platform



Customers can use V2PC to rapidly create and orchestrate media workflows across video headends and data center environments, and can evolve seamlessly from a hardware based infrastructure to a hybrid or pure virtualized cloud infrastructure. The software centric workflows increase the reachability of content across a variety of content consumption platforms.

This transformation has resulted in flexible user experiences and simplified operations, allowing customers to better manage, modify, and scale media workflows to support services such as Live, VOD, Time Shift, and Cloud DVR (cDVR) to OTT consumers.

V2PC works with a hierarchy of components that includes platforms, application containers, service containers, providers, zones, nodes, and the logical functions they support, which are configured into media workflows.

For more information on V2PC and its components, see the *Cisco Virtualized Video Processing Controller User Guide* for your V2PC release.

COS Node to COS Controller

All of the communication between a COS node and the COS controller that manages the node takes place on the doc-server channel.

COS HA V2PC Deployment

V2PC has two classes of components for HA:

- Third party components such as ZooKeeper, MongoDB, and Redis use their own proprietary clustering and redundancy schemes.
- Cisco components, such as the V2PC GUI and DocServer, use ZooKeeper for leader election.

Many of these applications also require a majority in order to form a quorum. That is, a cluster of three components can recover from the failure of a single component, because there are still two components to form a majority. But if two components fail, the single remaining component is not a majority, and the cluster cannot recover until one of the failed components recovers.

Therefore, we recommend configuring more than three V2PC VMs to ensure recovery in the event of multiple failures, and to support high performance, especially when sharing databases and other applications.

Note To determine the IP address of the master node, log in to one of the master nodes and run the command **nslookup master.v2p-ui.service.$Primary_Region.$Domain**.

Troubleshooting the COS Configuration

Unresponsive V2PC GUI

If the V2PC GUI is not responding or displays the message “Service not available. it is possible that there is a problem with DNS resolution. To look for potential problems with accessing the DNS server, check the /var/log/opt/cisco/v2pc/errorlog/dnsHelper-errorlog.current file. The following error message indicates a problem communicating with the DNS server:

```
[root@cos-381b5 ~]#cat /var/log/opt/cisco/v2pc/errorlog/dnsHelper-errorlog.current
2017-01-10T07:48:14.719Z 1147 dnsHelper dnsApiServer.js:213 [logger] ERROR -> Error:
<class 'socket.error'>
```

Manually Configuring Local Mirroring

You can manually configure local content mirroring on a COS node by editing the aftersetupfile. Follow these steps to configure the aftersetupfile for local content mirroring:

Note When manually configuring local mirroring, or any other settings that must persist, you must edit the aftersetupfile and *not* the setupfile. The settings in setupfile can be overwritten by changes made from the V2PC GUI. Also, you must configure the aftersetupfile *before* registering the new node to the V2PC. Otherwise, DEC settings within the cluster will be inconsistent, requiring at least one service-disrupting reboot to correct.

Step 1 Open (or if not present, create) the file /arroyo/test/aftersetupfile on the COS node for editing.

Step 2 In the aftersetupfile, enter the line **vault local copy count** *X*, where *X* is the total number of copies that you want to keep, including the original. For example, **vault local copy count 3** would cause the node to keep the original plus two local copies.

Step 3 To disable local erasure coding, enter the line **allow vault raid 0**.

Note If you are using the Cisco UCS 3160 or 3260, the rear SSDs in the back are used as system drives and are automatically mirrored as part of the installation, no intervention is required.

Manually Configuring Remote Mirroring

You can manually configure content mirroring across the COS nodes in a cluster by editing the aftersetupfile. Follow these steps to configure the aftersetupfile for content mirroring across the COS nodes:

Note When manually configuring local mirroring, or any other settings that must persist, you must edit the aftersetupfile and *not* the setupfile. The settings in the setupfile can be overwritten by changes made from the V2PC GUI. Also, you must configure the aftersetupfile *before* registering the new node to the V2PC. Otherwise, DEC settings within the cluster will be inconsistent, requiring at least one service-disrupting reboot to correct.

Step 1 Open (or if not present, create) the file /arroyo/test/aftersetupfile on the COS node for editing.

Step 2 In the aftersetupfile, enter the line **vault mirror copies** *X*, where *X* is the total number of copies that you want to keep in addition to the original. For example, **vault mirror copies 2** specifies two remote copies in addition to the (local) original.

Step 3 To disable local erasure coding, enter the line **allow vault raid 0**.

Erasure Coding Troubleshooting

Note The total number of data and parity stripes *cannot* exceed one less than the total number of available servers. This implies that for 1 data and 1 parity stripe, you need a minimum of at least 3 servers. For example, if you have 8 servers, at a maximum you can configure 6 Data and 1 Parity stripes, or 4 Data and 2 Parity stripes.

Watch for warnings in the /arroyo/log/protocoltiming*.log file before issuing any writes. If you do not have enough servers in your cluster, there will be a warning message in the protocoltiming*.log file on the last sample. If you receive this warning message, you must reduce the number of data or parity stripes so there are enough servers present to stripe data to.

Note To verify, you must use the GOID that is associated with a Swift Write object and use the **stripequery** command to /proc with that GOID.

RIO model writes have a max of 32 GB object with a 1 byte minimum and traditional model writes have a max of 512 GB write with a 1 byte minimum.

The following example shows how to check the striping for a RIO model write:

Step 1 Enter the following command to write four copies of a 2 Mb object:

```
time curl -v -X PUT -H "X-Rio-CopyCount: 4"
http://192.169.220.25/rio/bucket1/thierryg/2Mx4.ts -T ./2M
```

Step 2 Enter the following command to perform a Distributed Erasure Coding (DEC) stripequery using the Goid returned in Step 1:

```
ssh -o "BatchMode yes" 172.22.125.210 "echo 'stripequery 0x155d3f7dc003' >
/proc/calypso/test/filesystemtestcommand"  2>&1
```

Step 3 Enter the following command to check the file system log for striping:

```
ssh -o "BatchMode yes" 172.22.125.210 "tail -n 30 /arroyo/log/filesystemtest.log.20150819"
2>&1
```

The following example shows how to check the striping for a traditional model write:

Step 1 Create an account:

```
time curl -v -X PUT  -H "X-Auth-Admin-User: .super_admin" -H "X-Auth-Admin-Key: rootroot"
http://192.169.220.2/auth/v2/account90
```

Step 2 Create a user:

```
time curl -v -X PUT -H "X-Auth-Admin-User: .super_admin" -H "X-Auth-Admin-Key: rootroot"
-H "X-Auth-User-Key: rootroot" -H "X-Auth-User-Reseller-Admin: true"
http://192.169.220.2/auth/v2/account90/user90
```

Step 3 Get an authorization token and storage URL:

```
time curl -v -X GET  -H "X-Auth-User: account90:user90"  -H "X-Auth-Key: rootroot"
http://192.169.220.2/v1.0
```

Step 4 Create a container using the token and storage URL returned in Step 3:

```
time curl -v -X PUT -H "X-Auth-Token: AUTH_tkfec0e31bf1514a47bf29dddba697f8a6"
http://192.169.220.2/v1/AUTH_ea79aa8c-8656-4da9-9f8e-a69f49bdaa7f/container90
```

Step 5 Enter the following command to write a 512G object:

```
time curl -v -X PUT -H "X-Auth-Token: AUTH_tkfec0e31bf1514a47bf29dddba697f8a6"
http://192.169.220.2/v1/AUTH_ea79aa8c-8656-4da9-9f8e-a69f49bdaa7f/container90/512G -T 512G
```

Step 6 Get the Goid:

```
time curl -v -I -H "X-Auth-Token: AUTH_tkfec0e31bf1514a47bf29dddba697f8a6"
http://192.169.220.2/v1/AUTH_ea79aa8c-8656-4da9-9f8e-a69f49bdaa7f/container90/512G
```

Step 7 Enter the following command to perform a Distributed Erasure Coding (DEC) stripequery using the Goid returned in Step 1:

```
ssh -o "BatchMode yes" 172.22.125.210 "echo 'stripequery 0x155d3f7dc003' >
/proc/calypso/test/filesystemtestcommand" 2>&1
```

Step 8 Enter the following command to check the file system log for striping:

```
ssh -o "BatchMode yes" 172.22.125.210 "tail -n 30 /arroyo/log/filesystemtest.log.20150819"
2>&1
```

Node Decommissioning and Removal

COS lets you decommission a node at the CServer level. Decommissioning tells CServer to copy the data objects of the node to other nodes in the cluster until the target number of mirror copies is reached. After the node is decommissioned, it can be removed from the cluster using either the V2PC GUI or the API.

Node decommissioning itself is currently a CLI-only operation. To decommission a node, run the **/opt/cisco/cos-aic-client/cserver-control.pl decommission** command.

As decommissioning can take several hours, the CLI does not monitor the decommissioning process for completion. To check for completion, enter the command c**server-control.pl decommission --stats** periodically until the response confirms that the operation is complete.

After decommissioning is complete, you can safely remove the node using the V2PC GUI or the API. For instructions on removing a node from a cluster using the V2PC GUI, see Removing Decommissioned Nodes. For API information, see the *Cisco Cloud Object Storage Release API Guide*.

Note

- A node cannot be decommissioned after it has been removed from a cluster using the GUI or API. So, you must decommission a node before removing it.
- If a node is in the process of being decommissioned, decommissioning pauses if the node or any node in its cluster is placed in Maintenance mode. Decommissioning resumes when all nodes in the cluster are returned to In Service mode.
- Decommissioning will not start if you try to decommission a node when it or any node in its cluster is already in Maintenance mode. Decommissioning can only start when every node in the cluster is returned to In Service mode.

Verifying Node Removal from a Cluster

When you use the GUI to remove a node from a multi-node cluster, the node is first decommissioned from the Cassandra database cluster and then the Cassandra service and CServer are shut down. If you shut down the node before the Cassandra-level decommissioning completes, the node may still be considered part of the Cassandra cluster and still appear in the **nodetool status** output of the remaining nodes, but now with a status of down (DN). This status will prevent you from adding new nodes to the cluster.

To avoid this issue, you should open the COS AIC Client log before removing the node through the GUI, and periodically inspect the log to confirm that the Cassandra decommissioning is complete before you shut down the node.

Follow these steps to inspect the log for node decommissioning from the Cassandra cluster:

Step 1 Use the Linux **tail** command to print new lines that are added to the COS AIC Client log. Use **grep db-remove** to view only the lines that contain ‘db-remove’:

```
[root@Colusa-4T-72 ~]# tail -f /arroyo/log/cos-aic-client.log.20160506 | grep 'db-remove'
```

Step 2 Remove the node using the GUI and inspect the log for messages that contain **db-remove**:

```
[root@Colusa-4T-72 ~]# tail /arroyo/log/cos-aic-client.log.20160506 | grep db-remove
2016-05-06 23:01:29 UTC 127.0.0.1 aicc - Starting db-remove
```

Step 3 Inspect the log for messages that contain **Completed db-remove**, which shows that the node has been removed from Cassandra cluster:

```
[root@Colusa-4T-72 ~]# tail /arroyo/log/cos-aic-client.log.20160506 | grep Completed
db-remove
2016-05-06 23:02:49 UTC 127.0.0.1 aicc - Completed db-remove
```

Step 4 To verify that CServer has also been shut down, inspect the log for messages that contain **cserverControl-shutdown**:

```
[root@Colusa-4T-72 ~]# tail /arroyo/log/cos-aic-client.log.20160506 | grep
cserverControl-shutdown
2016-05-06 23:01:45 UTC 127.0.0.1 aicc - Completed cserverControl-shutdown
```

Step 5 To confirm that the removal process has completed, inspect the log to ensure that no new messages are printed:

```
[root@Colusa-4T-72 ~]# tail -f /arroyo/log/cos-aic-client.log.20160506
2016-05-06 23:01:45 UTC 127.0.0.1 aicc - Deleted /arroyo/test/setupfile
2016-05-06 23:01:45 UTC 127.0.0.1 aicc - Deleted /arroyo/test/RemoteServers
2016-05-06 23:01:45 UTC 127.0.0.1 aicc - Deleted /var/tmp/.clusterId
2016-05-06 23:01:45 UTC 127.0.0.1 aicc - Deleted /tmp/.cosnodeinit
2016-05-06 23:02:49 UTC 127.0.0.1 aicc - Completed db-remove
2016-05-06 23:02:49 UTC 127.0.0.1 aicc - Deleted /var/tmp/.dbinitflag
```

Step 6 Run the command **nodetool status cos** on one of the remaining nodes in the cluster to confirm that the removed node is no longer listed as part of the cluster.

Removing Decommissioned Nodes

You can remove a COS node from a cluster through the V2PC GUI after the node has been decommissioned.

Note Node decommissioning is currently a CLI-only operation. For instructions and related caveats, see Node Decommissioning and Removal.

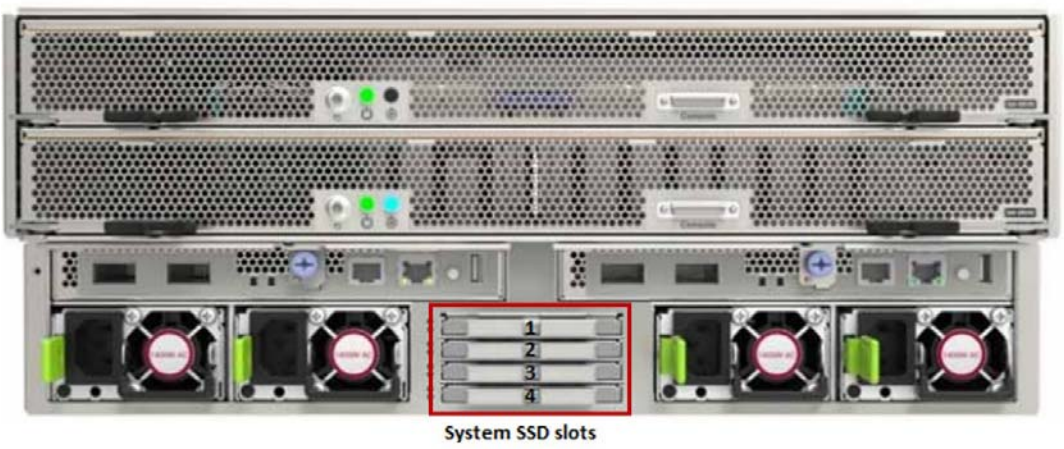
To remove a decommissioned COS node from a cluster, choose **Cisco Cloud Object Store (COS) > COS Nodes** from the navigation panel and click the Delete icon for the COS node to remove**,** as shown in Figure 2-1.

Figure 2-1 COS Node Removal from V2PC GUI

MegaRAID RAID1 SSD Replacement for CDE6032 Systems

The CDE6032 COS systems have hardware RAID1 implemented on the internal MegaRAID SAS controller for the system SSDs. The system SSDs are located in the rear of the C6032 as shown in Figure 2-2.

Figure 2-2 System SSD Slots



If the configured virtual drive (sda) enters a “Degraded” state, which indicates that a drive has failed, the LSI MegaRAID SAS 3108 controller will beep in a regular pattern. Also when an SSD fails, you will see a “DEGRADED” message in the /var/log/messages file as shown in Figure 2-3.

Figure 2-3 Virtual Drive in a Degraded State

```
Dec 23 12:04:26 Utah98 kernel: megaraid_sas 0000:17:00.0: scanning for scsi10...
Dec 23 12:04:26 Utah98 kernel: megaraid_sas 0000:17:00.0: 16360 (535838634s/0x00
01/CRIT) - VD 00/0 is now DEGRADED
[root@Utah98 ~]#
```

Note The COS system will continue to function properly even though the virtual drive (sda) is in a “Degraded” state.

Identifying the Failed SDD

Perform the following steps to identify which internal SSD (slot 1-4) has failed:

Procedure

Step 1 First check the SSDs that are associated with compute node 1. SSD slots 1 and 2 are assigned to compute node 1 (the top compute node) and SSD slots 3 and 4 are assigned to compute node 2 (the bottom compute node).

Step 2 From the system console of compute node 1, enter the command **/opt/MegaRAID/storcli/storcli64 /c0 show | grep “TOPOLOGY ” –A10**. If the SSD has an error state but is still powered up, the State column will show an error such as “Offln” or “Msng”. Figure 2-4 shows that the SSD in slot one (labeled “0” in the “Row” column) is in a missing state (“Msng”).

Figure 2-4 SDD with an Error State of Missing

```
[root@Utah98 storcli]# ./storcli64 /c0 show | grep "TOPOLOGY" -A10
TOPOLOGY :
========

-----------------------------------------------------------------------------
DG Arr Row EID:Slot DID Type  State BT       Size PDC  PI SED DS3  FSpace TR
-----------------------------------------------------------------------------
 0 -   -   -        -   RAID1 Dgrd  N  446.102 GB dflt N  N   dflt N      N
 0 0   -   -        -   RAID1 Dgrd  N  446.102 GB dflt N  N   dflt N      N
 0 0   0   -        -   DRIVE Msng  -  446.102 GB -    -  -   -    -      N
 0 0   1   252:202  3   DRIVE Onln  N  446.102 GB dflt N  N   dflt -      N
-----------------------------------------------------------------------------
```

Note The Raw Size of the 480GB SSD is reported as 446.102 GB. This is normal because approximately 10% of the SSD media is held in reserve for error correction.

Step 3 If the drive in EID:Slot 252:201 or the drive referenced by Row 0 on compute node 1 shows an error state, then you will need to replace the SSD in slot 1. If the drive in EID:Slot 252:202 or the drive referenced by Row1 on compute node 1 shows an error state, then you will need to replace the SSD in slot 2.

Step 4 If the drives for compute node 1 do not display an error in the State column, then you will next check the drives on compute node 2.

Step 5 From the system console of compute Node 2, enter the command **/opt/MegaRAID/storcli/storcli64 /c0 show | grep "TOPOLOGY " –A10**.

Step 6 If the drive in EID:Slot 252:201 or the drive referenced by Row 0 on compute node 2 shows an error state, then you will need to replace the SSD in slot 3. If the drive in EID:Slot 252:202 or the drive referenced by Row1 on compute node 2 shows an error state, then you will need to replace the SSD in slot 4.

Replacing the Failed SSD

After you have determined which SSD has failed, perform the following steps to replace the failed drive:

Note Ensure that you remove the correct SSD. Removing the incorrect SSD from the Raid1 Mirrored pair will render the compute (COS) node inoperable and will require the software to be reinstalled to recover.

Note The failed SSD is hot-swappable.

Procedure

Step 1 Remove the SSD that you identified as failed by pressing the tab on the left-hand side of the tray to the right and pulling straight back on the tabs.

Note Ensure that you remove the correct SSD. Removing the incorrect SSD from the Raid1 Mirrored pair will render the compute (COS) node inoperable and will require the software to be reinstalled to recover.

Step 2 Replace the failed SSD with a new 480GB SSD.

- When you replace the SSD, the system console will display a message similar to the following to indicate that a new device was found:

  `"megaraid_sas: scanning for scsi6....`

Step 3 Enter the command **/opt/MegaRAID/storcli/storcli64 /c0 show | grep "TOPOLOGY " –A10**. The state of the replaced drive should show "Rbld", which indicates that the drive is in the Rebuild state as shown in Figure 2-5 for the drive in slot 201.

Figure 2-5 SSD in the Rebuild State

```
[root@Utah98 storcli]# ./storcli64 /c0 show | grep "TOPOLOGY" -A10
TOPOLOGY :
========


-----------------------------------------------------------------------------
DG Arr Row EID:Slot DID Type  State BT       Size PDC  PI SED DS3  FSpace TR
-----------------------------------------------------------------------------
 0 -   -   -        -   RAID1 Dgrd  N  446.102 GB dflt N  N   dflt N      N
 0 0   -   -        -   RAID1 Dgrd  N  446.102 GB dflt N  N   dflt N      N
 0 0   0   252:201  2   DRIVE Rbld  Y  446.102 GB dflt N  N   dflt -      N
 0 0   1   252:202  3   DRIVE Onln  N  446.102 GB dflt N  N   dflt -      N
-----------------------------------------------------------------------------
```

Step 4 The state of Rebuild indicates that the RAID1 virtual device is being remirrored. During the rebuild process, the system will continue to function as normal and the RAID1 rebuild will continue to run in the background until the process is complete.

Verifying the RAID1 Virtual Device State

After you have replaced the failed SSD and the rebuild process is finished, enter the command **/opt/MegaRAID/storcli/storcli64 /c0 show | grep "TOPOLOGY " –A10**. If the rebuild process is finished, the state of the replaced drive will be "Onln", which indicates that the drive is online, and the state for RAID1 will be "Optl", as shown in Figure 2-6.

Figure 2-6 SSD in the Online State and RAID1 in the Optimal State

```
[root@Utah198 storcli]# ./storcli64 /c0 show | grep "TOPOLOGY" -A10
TOPOLOGY :
========


-----------------------------------------------------------------------------
DG Arr Row EID:Slot DID Type  State BT       Size PDC  PI SED DS3  FSpace TR
-----------------------------------------------------------------------------
 0 -   -   -        -   RAID1 Optl  N  446.102 GB dflt N  N   dflt N      N
 0 0   -   -        -   RAID1 Optl  N  446.102 GB dflt N  N   dflt N      N
 0 0   0   252:201  3   DRIVE Onln  N  446.102 GB dflt N  N   dflt -      N
 0 0   1   252:202  0   DRIVE Onln  N  446.102 GB dflt N  N   dflt -      N
-----------------------------------------------------------------------------
```

Using Logs and Monitoring to Troubleshoot COS Runtime Issues

Log files, alarms, and events play a key part in troubleshooting the COS installation. This chapter takes a look at these items to troubleshoot the COS installation and contains the following sections:

Log Files

Core Dump File

To view information from a core dump, review the /var/log/kern file.

SM Logs

The SM log messages provide the following information:

- Date
- Name and version of the application
- Remote client IP address and port number
- Full URL of the request
- Type of operation (GET, POST, PUT, DELETE)
- Request body for POST and PUT operations
- Any log messages

To view these log messages, open the **/var/log/opt/cisco/v2pc/errorlog/service-mgr-errorlog.current** file.

The following is a sample log message for INFO:

```
2016-01-14T18:59:07.139Z 2486 service-mgr sm_logger.js:30 [logger] INFO  ->
{"date":"2016-01-14T18:59:07.138Z","appName":"service-mgr","appVersion":"1.0.22","operatio
n":"create","remoteClientIp":"127.0.0.1","requestUrl":"http://127.0.0.1:8001/v2/ippools/po
ol-1","requestBody":{"name":"pool-1","properties":{"description":"pool-1","addrType":"ipv4
","networkRef":"smtenant_system.smnetwork.cf-int","pool":[{"rangeStart":"192.169.220.142",
"rangeEnd":"192.169.220.145","netmask":"255.255.255.224","gw":"192.169.220.129"}]},"id":"p
ool-1","type":"smippool","owner":"smtenants.smtenant.system"},"message":"Successful
creation of the document"}
```

COS AIC Logs

In addition to the COS Status and COS Events reported in the V2PC GUI, the following COS AIC log is available from the CLI of the V2PC:

- **/var/log/opt/cisco/v2pc/errorlog/CosAicLog-*cisco_cos_application_instance_name*-errorlog.current**, where *cisco-cos application instance name* is the actual cisco-cos application instance name configured on the V2PC GUI.
  - This log records COS AIC execution information.

COS AIC Client Logs

The following COS AIC Client log files are available on the COS nodes:

- **/arroyo/log/cos-aic-client.log:** This is the primary COS AIC Client log file.
- **/var/log/cos-aic-client.stderr:** This log contains messages related to a COS AIC Client being terminated and is helpful in debugging unexpected crashes.

System and Service Status of COS Nodes using V2PC

The V2PC Service Manager displays the status of each node that is in service and that is part of a COS cluster. This status is updated every 30 seconds. To view the status of a COS node using the V2PC GUI, perform the following steps:

Step 1 From the V2PC GUI, choose **Cisco Cloud Object Store (COS) > COS Service Status**.

Step 2 In the window that appears, expand the cluster that contains the node that you want to view. All of the nodes that are part of the cluster appear, providing the following information:

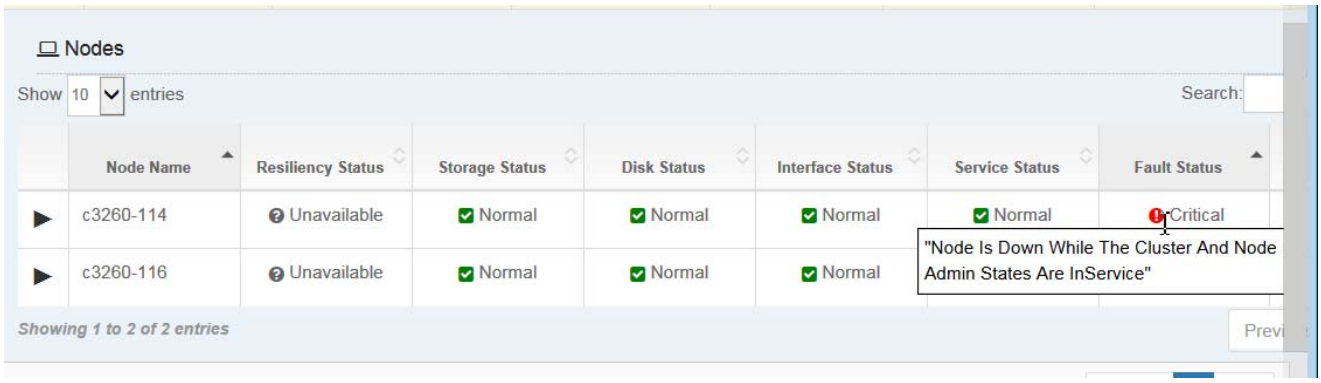
- Resiliency status
- Storage status
- Disk status
- Interface status
- Service status
- Fault status
- COS Node version

The Fault Status column shows the status of each node. This status reflects the status of the network interfaces, disks, and services of the node and can be None, Warning, or Critical. The following are reasons that the status may show Critical.

- If all of the interfaces of the node are nonfunctional, the node status will be Critical.
- If all of the disks of the node are nonfunctional, the node status will be Critical.
- If a service is down, the node status will be Critical.

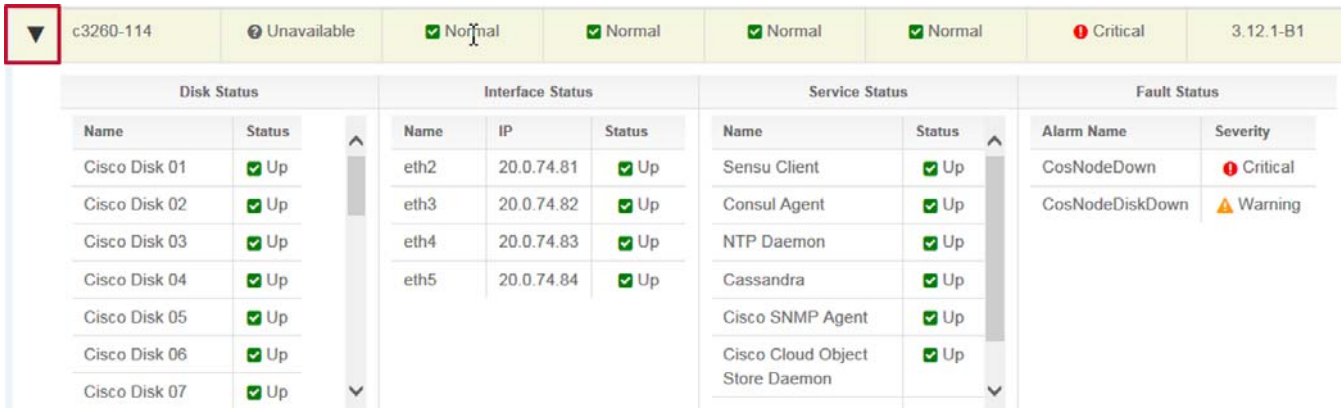
Step 3 To view the detail of the fault, hover over the information icon to see the complete detail of the fault. Figure 3-1 provides an example of faults that you might see on a COS node.

Figure 3-1 COS Node Down Fault



To see the status of individual disks, interfaces, or services of a COS node, click the arrow head next to that node, as shown in Figure 3-2:

Figure 3-2 Status Details



Viewing Component Statistics using V2PC

From the V2PC GUI, you can view the statistics and alarms for the individual COS components. Follow these steps to view this information:

Step 1 Log in to the V2PC GUI.

Step 2 Open the navigation panel and choose **Cisco Cloud Object Store (COS) > COS Service Statistics**.

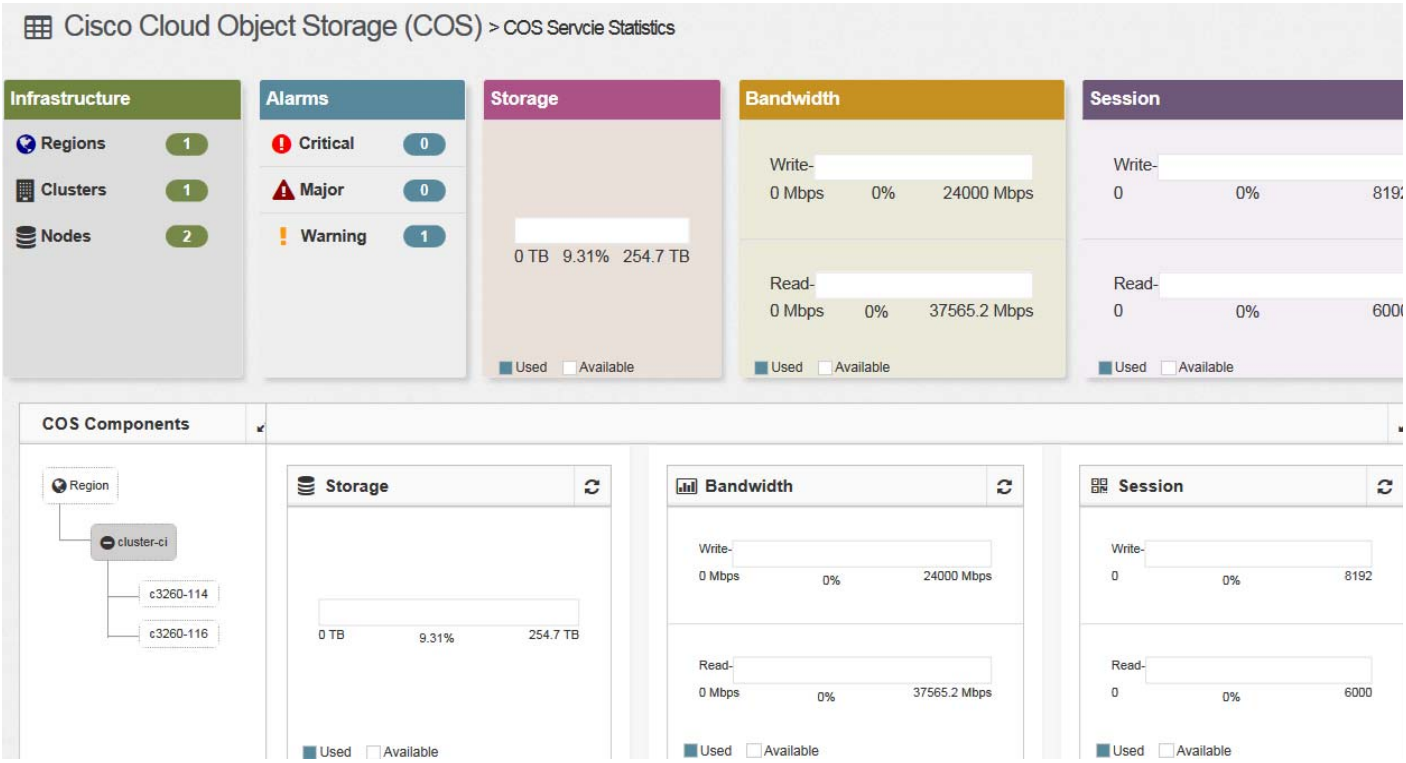
The COS Statistics dashboard appears. This window displays the following information:

- The usage for Storage, Bandwidth, and Session.
- The usage trends for Storage, Bandwidth, and Session based on either a 24 hour or 14 day period.
- The status of disks, services and interfaces
- Alarms that are related to the node/cluster

You should check the COS Statistics dashboard if the IP address of the C/F interfaces do not appear in the DNS list.

Figure 3-3 show an example of the COS Service Statistics dashboard.

Figure 3-3 COS Service Statistics Dashboard



Viewing Alarms and Events with V2PC

COS AIC Server Alarms

The v2PC GUI displays the active alarms and the alarm history for the COS nodes. To troubleshoot the COS nodes you will typically view the active alarms. To view the active COS AIC Server Alarms, from the V2PC GUI choose **Dashboard > Alarms & Events**. From the Dashboard window that appears, ensure that **Alarms** is chosen from the Show menu.

There are four different alarms that the COS AIC Server might generate:

- **cos-0-1.cosnode-interface-down:** A COS node interface is reported as down.
  - If 50% or more of the C/F interfaces are reported as down, the alarm level is set to Critical.
  - If less than 50% of the C/F interfaces are reported as down, the alarm level is set to Warning.
- **cos-0-1.cosnode-disk-down:** A COS node disk is reported as down.
  - If 25% or more of the disks are reported as down, the alarm level is set as Critical.
  - if less than 25% of the disks are reported as down, the alarm level is set as Warning.
- **cos-0-1.cosnode-out-of-service:** A COS node service is reported as down, due to one of the following conditions:
  - Communication with a COS node is lost, which is indicated when an aic_cosnodestatus update is not received after 30 seconds.
  - A critical service (for example, Data Base, cServer, or COSd) is determined to be down.
- **cos-0-1.service-deactivated:** The COS service is deactivated for a cluster.
  - This alarm is reported when the COS Service Instance is moved from Enabled to Disabled from the COS GUI.

Table 3-1 provides a summary of these alarms:

Table 3-1 COS AIC Server Alarms

Alarm Name	Description	Severity	Action Requred
cos-0-1.cosnode-interface-down	One or more interfaces have reported down and have been removed from DNS	Warning or Critical, depending on the number of down interfaces	Determine cause and take corrective action
cos-0-1.cosnode-disk-down	One or mode disks on the COS node are reporting down	Warning or Critical, depending on the number of down disks	Correct or replace bad disk(s)
cos-0-1.cosnode-out-of-service	One or more critical COS services has stopped on the COS node, and the node interfaces were removed from DNS	Critical	Check and restart any COS services that are down
cos-0-1.service-deactivated	The COS Service (Service Instance, or Capture Endpoint) was disabled via the COS GUI, all COS interfaces were removed from DNS cluster	Critical	Use the COS GUI to enable the COS Service Instance or Capture Endpoint

COS AIC Events

The V2PC GUI displays the management events for a COS node to help the operator understand the current state of their COS Service and the health of their COS nodes. To view these events, choose **Dashboard > Alarms & Events**. From the Dashboard window that appears, ensure that **Events** is chosen from the Show menu. The oldest events are displayed first. You can sort by any other column by clicking the column header.

Events will have one of the following levels of severity. Pay attention to the Warning and Critical events:

- **Info:** The event is informational only and requires no action by the operator.
- **Warning:** An issue occurred that is possibly transitory and the operator should investigate the cause.
- **Critical:** An issue occurred from which the node may not recover without operator intervention and may cause a service outage. The operator should act immediately..

Table 3-2 provides an overview of some of the key critical and warning COS Service, Server, and Client events.

Table 3-2 COS AIC Events

	Event Name	Description	Severity	Event Type	Event Subtype
Service Events	Service.Deactivated	COS service deactivated by disabling Service Instance	Critical	Config	Config
	COSNode.Down	COS node has not reported status for last 30 seconds	Critical	Node	Accessibility
	Service.StorageStatus	SLA Status for Storage is critical	Critical	Application	SLA
	Service.SessionStatus	SLA status for sessions is critical	Critical	Application	SLA
	Service.BandwidthStatus	SLA status for sessions is critical	Critical	Application	SLA
	COS-AIC Terminated	COS AIC is exiting	Critical	Application	Health
COS AIC Server Events	CosNodeInterfaceError	No available IP addresses in the IP pool	Critical	COS-Node	Accessibility
	CosUpdatedActiveIpPool	An active IP pool was edited	Critical	COS-Node	Accessibility
	DeleteCosNode	A COS node was deleted	Warning	COS-Node	
COS AIC Client Events	CosNodeDiskDown	Disk *disk_name* down	Warning	COS-Node	Health
	CosNodeInterfaceDown	Interface *if_name* down	Warning	COS-Node	Health

Troubleshooting Issues Across the COS System

Viewing the Status of Primary System Services

To view the status of the primary COS system services on a COS node, use the following commands:

```
[root@utah97 ~]# service cassandra status
cassandra is running
[root@utah97 ~]# service cosd status
cosd (pid  9235) is running...
[root@utah97 ~]# service cserver status
cserver is running
```

To verify the cosd.conf configuration file, use the following command:

```
[root@utah97 ~]# grep -v '#' /etc/cosd.conf
cluster url http://auth-cos.mos.ddns.npi.cds.cisco.com/v1


db host 10.93.232.16
```

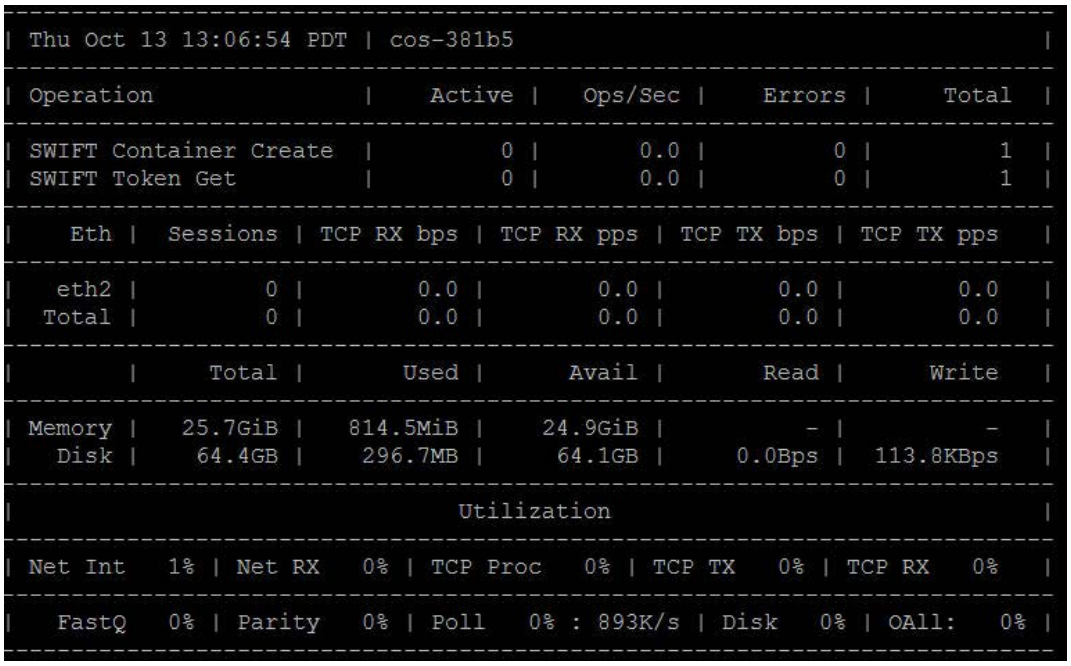
To verify the NTP configuration, use the following command:

```
[root@utah97 ~]# ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*10.74.44.56     10.64.58.50      3 u  186 1024  377    0.880    0.044   0.087
```

Check Current System Load

To see a real-time overview of the system utilization, enter the following command:

```
[root@utah97 ~]#/opt/cisco/cos/bin/cos_stats
```



Service High-Availability

Each COS node leverages the monit framework to provide high-availability of the cos service. In the event that the cosd process crashes, the monit framework will restart the service.

Troubleshooting Cassandra Issues

Verify the Status of Cassandra Nodes

To verify that all Cassandra nodes are up and running, on the COS node enter the command **nodetool status**. The status for the Cassandra nodes should show “UN” for Up, Normal. For example:

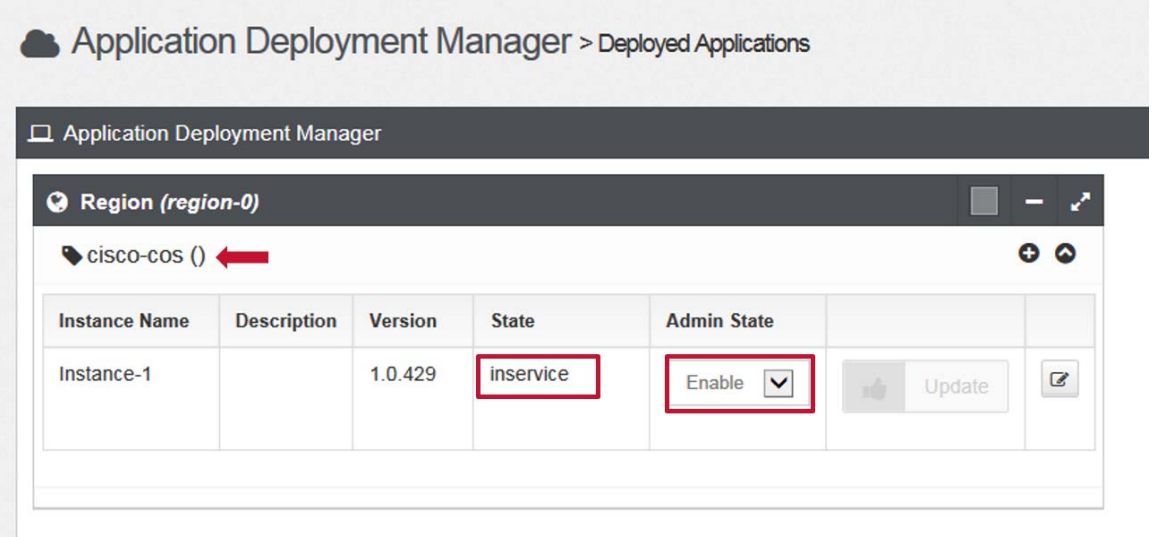
```
[root@cosnode log]# nodetool status
Datacenter: DC1
===============
Status=Up/Down|/ State=Normal/Leaving/Joining/Moving
--  Address         Load       Tokens  Owns (effective)  Host ID                               Rack
UN  172.22.125.16   66.3 KB    256     16.7%             990771b1-babd-4de6-883c-1748ada16410  RAC1
UN  172.22.125.48   66.22 KB   256     17.5%             0d309a8c-2bfe-4033-a4a0-83d9c4d1baf5  RAC1
UN  172.22.125.33   66.36 KB   256     17.6%             29c32e41-0ed0-45e2-9f14-ddd52424c27a  RAC1
UN  172.22.125.49   66.16 KB   256     16.1%             6e07235a-c8b2-425c-999e-81966f106584  RAC1
UN  172.22.125.52   246.73 KB  256     18.3%             66f2736c-b370-45f5-ae7d-7f80f709e01d  RAC1
```

Verifying the COS-Controller is Running on a V2PC HA Node

The COS-Controller runs as an application named cisco-cos on the V2PC platform. To check the status of the cisco-cos application on the V2PC node, perform the following steps:

Step 1 Log in to the V2PC GUI. From the navigation panel, choose **Application Deployment Manager** > **Deployed Applications**. Verify that the "cisco-cos" application has an instance that has a State of "inservice" and an Admin State of "Enable", as shown in Figure 4-1.

Figure 4-1 COS-Controller Status



General Information and Issues

Core Dump Location

To view the messages from a core dump, view the **/var/log/kern** file.

Identify the Software Versions or Releases

The following sections describe the commands for identifying the software versions on the server.

Linux OS Version

To identify the software version of the Linux OS on the COS node or V2PC Service Manager enter the following commands:

```
# cat /proc/version
```

```
    Linux version 2.6.18-53.el5.kernel.2_6_18.2008.10.07.01 (arroyoqa@build-svr) (gcc
    version 4.1.2 20070626 (Red Hat 4.1.2-14)) #1 SMP Mon Nov 17 18:21:51 PST 2008
# uname -a
    Linux stm74 2.6.18-53.el5.kernel.2_6_18.2008.10.07.01 #1 SMP Mon Nov 17 18:21:51 PST
    2008 i686 i686 i386 GNU/Linux
```

Installed COS Packages

To determine the software version of the COS packages that are installed on the COS node, enter the following command:

```
[root@cos-381b5 test]# cos_pkgs

-------------------------------------------------------------------------------
| Name                          | Version     | Release           | Build  |
-------------------------------------------------------------------------------
| avs_tools                     | 3.12.1      | cos0.1.1          | b2     |
| cassandra-cpp-driver          | 2.1.0       | 1.el6             | -      |
| cassandra21                   | 2.1.11      | 1                 | -      |
| cassandra21-tools             | 2.1.11      | 1                 | -      |
| cddm                          | 3.12.1      | cos0.1            | b2     |
| cds_devtest                   | 3.12.1      | cos0.1            | b2     |
| cds_devtest_test_suites       | 3.12.1      | cos0.1            | b2     |
| cos-aic-client                | 2.4.1       | 1482189083        | -      |
| cos_client                    | 3.12.1      | cos0.1.1          | b2     |
| cos_config                    | 3.12.1      | cos0.5            | b2     |
| cos_config-cassandra          | 3.12.1      | cos0.5            | b2     |
| cos_config-monit              | 3.12.1      | cos0.5            | b2     |
| cos_config-syslog-ng          | 3.12.1      | cos0.5            | b2     |
| cos_snmp                      | 3.12.1      | cos0.1            | b2     |
| cos_utils                     | 3.12.1      | cos0.1            | b2     |
| cosd                          | 3.12.1      | cos0.1.1          | b2     |
| cserver-prod                  | 3.12.1      | cos0.1.1.1        | b2     |
| kernel-vds                    | 2.6.32      | 3.12.1_cos0.1     | b2     |
| nginx                         | 1.6.0       | 3.12.1.cos0.1     | b2     |
| td-agent-cos-plugins          | 3.12.1      | cos0.1.1          | b2     |
| vds_framework                 | 3.12.1      | cos0.1            | b2     |
| vds_logrotate                 | 3.12.1      | cos0.1.1          | b2     |
-------------------------------------------------------------------------------
```

CSserver Code

To view the CServer settings, status, and version, enter the following command:

```
# cat /proc/calypso/status/server_settings
    CServer Information ENV_ISA_SR prod (cdsbuild@cds-build7) (gcc 4.4.7 20120313 (Red Hat
    4.4.7-17)) 3.12.1-0b2


    Server Settings:
        Server is operational
        Cache2App is operational
        TSCs Per Second is 2294658000

    Network Settings:
        Running in L3 Network Mode
        Disallow Jumbo Frames
        Transport/Stream Data Payload: 1316
        Cache/Fill Data Payload: 1024
        Cache/Fill Control Maximum Packet Size: 1500
```

```
Stream Status Queue is disabled
```

Monitor Traffic Using ifstats

The **ifstats** command shows real-time traffic on each Ethernet interface on the server.

```
# /home/stats/ifstats
    ifstats  -   11:12:22
    ============================================================
    Int#    R-Mbps    X-Mbps           R-Bytes           X-Bytes
     eth0        0         0          56760511         166307653
     eth1        0         0                 0                 0
     eth2        4       457        3439241508        3497139080
     eth3        4       457        3439172148        3099124288
     eth4        4       457        3441836680        2945489644
     eth5        4       472        3443060380        2736115618
     eth6        4       471        3438423816        2613199736
     eth7        5       464        3440066492        2419935662
     eth8        4       449        3439982812        2266582156
     eth9        4       465        3443251384        2164010982
    eth10        5       465        3439982136        1915437726
    eth11        4       464        3438935192         397577442
    eth12        5       464        3440343164         300903930
    eth13        4       465        3439540716        4454799830
```

View Disk Drive Information

The disk drive order is irrelevant when reinserting disk drives after transporting a chassis, or transferring disk drives from one chassis to another.

To view the statistics of the internal boot drive, the disk drive that contains the software, enter the **df -k** command.

```
# df -k
    Filesystem           1k-blocks      Used Available Use% Mounted on
    /dev/hda1             10317828   3764936   6028776  39% /
    /dev/hda2             20641788   1711372  17881776   9% /arroyo
    /dev/hda3              8254272     32828   7802148   1% /arroyo/db
    /dev/hda6             35641880   1185880  32645480   4% /arroyo/log
    none                   1681200         0   1681200   0% /dev/shm
```

To view the statistics of a removable SATA or SCSI disk drive, use the following command:

```
# cat /proc/calypso/status/diskinfo
    Disk Info:
      Disks(12) Op(12)
      Storage: T(804G) A(21%) U(0)
      BW: (99%) w(1.35M/s) r(0/s)
      I/O Util: w(1:0%) e(0) a(0%)
    Disk[ 1][67.0G] A[20%] B[11x]
    Disk[ 2][67.0G] A[20%] B[0x]
    Disk[ 3][67.0G] A[21%] B[0x]
    Disk[ 4][66.5G] A[22%] B[0x]
    Disk[ 5][67.0G] A[20%] B[0x]
    Disk[ 6][67.0G] A[21%] B[0x]
    Disk[ 7][67.0G] A[20%] B[0x]
    Disk[ 8][67.0G] A[20%] B[0x]
    Disk[ 9][67.0G] A[21%] B[0x]
    Disk[10][67.0G] A[20%] B[0x]
```

```
    Disk[11][67.0G] A[20%] B[0x]
    Disk[12][67.0G] A[20%] B[0x]
```

To view the status of a specific drive, including serial number and model number, enter the **cds/cdd/disks/csd***XX* command, where *XX* is number of the disk drive to view. For example:

```
# cat /proc/cds/cdd/disks/csd1
    proc_flags: 0x20
    name: csd1vendor: SEAGATE
    model: ST4000NM0023rev: 0004
    serial: Z1Z929G60000R547U1UX
    slot: 1
    location: 1.1.1.0
    connection: 22.0
    total_sectors: 7814037167
    sector_size: 512
    max_transfersize: 262144
    state: 0x800007; DEV_ALLOCATED DEV_ATTACHED DEV_READY
```

Note If one of the drive fails on a COS node, you should see a warning message in the /arroyo/log/protocoltiming*.log file on the COS node.

View the Network Configuration and Activity

The following commands are useful for checking your network configuration and activity.

To view the ARP table, enter the following command:

```
# arp -a
    jetsam.v.com (111.0.110.151) at 00:00:0C:07:AC:00 [ether] on eth0
    COS17-m1.v.com (111.0.210.170) at 00:30:48:58:5B:A1 [ether] on eth0
    COS17-v1.v.com (111.0.210.171) at 00:30:48:31:53:B2 [ether] on eth0
    ? (111.0.210.175) at 00:30:48:32:0A:5A [ether] on eth0
    COS17-s1.v.com (111.0.210.172) at 00:04:23:D8:89:44 [ether] on eth0
    COS17-s1.v.com (111.0.210.172) at 00:04:23:D8:89:44 [ether] on eth0
```

To view the IP routing table, enter the following command:

```
# netstat -rn
    Kernel IP routing table
    Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
    111.0.210.0     0.0.0.0         255.255.255.0   U         0 0          0 eth0
    111.0.0.0       0.0.0.0         255.0.0.0       U         0 0          0 eth0
    127.0.0.0       0.0.0.0         255.0.0.0       U         0 0          0 lo
    0.0.0.0         111.0.210.1     0.0.0.0         UG        0 0          0 eth0
```

To view the network statistics and check for discarded packets, enter the following command:

```
# netstat -s
    Ip:
        16327814 total packets received
        0 forwarded
        0 incoming packets discarded
        16327814 incoming packets delivered
        18133919 requests sent out
    Icmp:
        26566 ICMP messages received
        25829 input ICMP message failed.
        ICMP input histogram:
            destination unreachable: 26491
```

```
            echo requests: 71
            echo replies: 4
        1061 ICMP messages sent
        0 ICMP messages failed
        ICMP output histogram:
            destination unreachable: 986
            echo request: 4
            echo replies: 71
    IcmpMsg:
            InType0: 4
            InType3: 26491
            InType8: 71
            OutType0: 71
            OutType3: 986
            OutType8: 4
    Tcp:
        1351227 active connections openings
        4456 passive connection openings
        86959 failed connection attempts
        61 connection resets received
        25 connections established
        13373578 segments received
        15603614 segments send out
        32 segments retransmited
        0 bad segments received.
        87431 resets sent
    Udp:
        2527503 packets received
        734 packets to unknown port received.
        0 packet receive errors
        2529331 packets sent
```

To view the COS subnet table, enter the following command:

```
# cat /arroyo/test/SubnetTable
    network 192.169.75.64 netmask 255.255.255.192 gateway 192.169.75.126
```

Note Local networks and their gateways are specified in the SubnetTable file.

To view the COS Remote Server table, enter the following command:

```
# cat /arroyo/test/RemoteServers
    remote server
    id 141
    ip 111.1.9.20
    ip 111.1.9.21
    ip 111.1.9.22
    ip 111.1.9.23
    ip 111.1.9.24
    end remote server

    remote server
    id 143
    ip 111.1.9.25
    ip 111.1.9.26
    end remote server

    remote server
    id 144
    ip 111.1.9.27
    ip 111.1.9.28
```

```
    ip 111.1.9.29
    ip 111.1.9.30
    end remote server
```

Interface Information

To view basic interface information, use the **ifconfig** command.

```
# ifconfig eth0
    eth0      Link encap:Ethernet  HWaddr 00:04:23:D8:9A:80
              inet addr:111.0.110.41  Bcast:111.0.110.255  Mask:255.255.255.0
              UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
              RX packets:13946269 errors:0 dropped:0 overruns:0 frame:0
              TX packets:11594110 errors:0 dropped:0 overruns:0 carrier:0
              collisions:0 txqueuelen:1000
              RX bytes:3085199261 (2942.2 Mb)  TX bytes:1317620721 (1256.5 Mb)
              Interrupt:24 Base address:0x3000 Memory:dd240000-0
```

To view interface card settings for physical interfaces use the **ethtool** command.

```
# ethtool eth0
    Settings for eth0:
        Supported ports: [ FIBRE ]
        Supported link modes:   10000baseT/Full
        Supported pause frame use: No
        Supports auto-negotiation: No
        Advertised link modes:  10000baseT/Full
        Advertised pause frame use: No
        Advertised auto-negotiation: No
        Speed: 1000Mb/s
        Duplex: Full
        Port: FIBRE
        PHYAD: 0
        Transceiver: external
        Auto-negotiation: off
        Current message level: 0x00000000 (0)

        Link detected: yes
```

For detailed interface information, view the interface information file in the /proc/net/ directory.

```
# cat /proc/net/adapters/eth0.info
    Description                     Cisco Systems IncÂ®  VIC Ethernet Adapter
    Driver_Name                     enic
    Driver_Version                  2.1.1.66
    Adapter_Name                    eth0
    MAC_addr                        E4:AA:5D:AD:65:11

    PCI_Vendor                      0x1137
    PCI_Device                      0x0043
    PCI_Subsystem_Vendor            0x1137
    PCI_Subsystem_Device_ID         0x012e
    PCI_Bus                         0x06
    PCI_Slot                        0

    Uplink_Interface                0
    Link                            UP
    Speed                           1000 Mb/s

    Tx_Packets                      32128987
    Tx_Unicast_Packets              32125075
    Tx_Multicast_Packets            6
```

```
Tx_Broadcast_Packets            3906
Tx_Bytes                        4620479646
Tx_Unicast_Bytes                4620229170
Tx_Multicast_Bytes              492
Tx_Broadcast_Bytes              249984
Tx_Errors                       0
Tx_Dropped                      0

Rx_Packets                      39367510
Rx_Packets_Total                39367510
Rx_Unicast_Packets              30780687
Rx_Multicast_Packets            72
Rx_Broadcast_Packets            8586751
Rx_Bytes                        8261544222
Rx_Unicast_Bytes                7651378709
Rx_Multicast_Bytes              6768
Rx_Broadcast_Bytes              610158745
Rx_Errors                       0
Rx_Over_Errors                  0
Rx_CRC_Errors                   0
Rx_Dropped                      0
Rx_No_Bufs                      0
```

CHAPTER 5

Troubleshooting COS Service APIs and Issues

Troubleshooting Swift and Swauth API Errors

If a Swift or Swauth API operation returns a 500 level HTTP error status, this can indicate an issue with one of the three primary system services: cassandra-server, cosd, or cserver. Sometimes a 500 level HTTP error status is returned because of a temporary resource exhaustion on the COS node. However, if the error status persists for a long period of time, verify that the primary system services are running.

Log Files to Trace and Analyze Swift, Swauth, and Cassandra Transactions

The following log files, which are collected on each individual COS node, provide information that can help trace and analyze Swift, Swauth, and Cassandra transactions:

- **/arroyo/log/http.log.<DATE>**
  - This log collects HTTP transaction information from Swift and Swauth operations.
  - The following is an example of an entry for a RIO model write:

```
2015-09-03 05:29:34 UTC cde250-1 : AUDIT : ffff8806c77f0b68 : 172.22.102.214:460006 <-> 192.169.220.2
: RIO WRITE OBJECT : PUT /rio/bucket1/id1 : 0x0000155db8b711 f9e : 202 1024x1 67.1Kbps :
(0:29:39:4:48:121)
```

    In this example, the numbers in parenthesis at the end of the output represent the following information, listed in order:

    - **<queue-time>:** The time the HTTP request stayed in the queue before initial processing started (0 ms in this example)
    - **<initial-meta-data-time>:** The time it took for the initial object metadata creation in cassandra (29 ms in this example)
    - **<tcp-receive-time>:** The time it took to receive the entire object over the network from the client (39 ms in this example)
    - **<disk-write-completion-time>:** The time it took to write the expanded object (local and remote expansion in parallel) (4 ms in this example)
    - **<final-meta-data-time>:** The time it took to finalize the object metadata in cassandra (48 ms in this example)
    - **<total-time>:** The total time for the transaction (121 ms)
  - The following is an example of an entry for a traditional model reads:

```
27-Jul-2015 22:59:28 UTC :: AUDIT : ffff88075578c850 : 20.0.52.37:52926 <-> 20.0.52.55 : SWIFT READ
OBJECT : GET /v1/AUTH_123/mycontainers2/CISCO24MB_4064 : 0x0000255b56034cb0 :
0-549755813887(549755813888) 1.00Mbps : 200 0-23999999(24000000:0) 432Mbps : (0:3:18:90:0:0:63:507)
```

The following describes some of the key fields in this output:

- 1.00 Mbps: The committed bandwidth
- 200: The HTTP response code
- 0-239999999: The returned range
- 24000000: The returned length
- 432 Mbps: The actual bandwidth
- 0:3:18:90:0:0:63:507: Time taken in various stages of the request, represented as milliseconds, listed in the following order:
  - m_queueDelayTicks
  - m_metadataDelayTicks
  - m_initialDataDelayTicks
  - m_totalDataWaitTicks
  - m_totalWindowWaitTicks
  - m_totalClientWindowClosedTicks
  - m_transmitStartDelayTicks
  - m_totalRequestTicks

– If the HTTP client issuing the REST API write request aborts prematurely, the HTTP response code is -1, as shown in the following example:

```
2015-08-19 15:45:11 UTC cde250-1 : AUDIT : ffff8806c7000b68 : 172.22.102.214:3566
68 <-> 192.169.220.2 : SWIFT WRITE OBJECT : PUT /v1/AUTH_ea79aa8c-8656-4da9-9f8ee
-a69f49bdaa7f/container90/513G : 0x0000155d3f7dc005 : -1 0-548692869120 609Mbps
: (0:11:7199939:136:446:7200535
```

- **/arroyo/log/cosd.log.<DATE>**
  - This log records Cassandra transactions that are executed when a client invokes a Swift or Swauth API operation. In this log file watch for "err:", "wrn:", or "ftl: errors. These errors correspond to the standard Unix syslog levels of WARN(wrn), ERR(err), and FATAL(ftl).
- **/arroyo/log/protocoltiming.log.<DATE>**
  - This log provides information about any network interface issues and any disk issues.

Note When debugging, it may be helpful to raise the HTTP log level of the service in question using the command **echo 9 > /proc/calypso/tunables/http_log_level**. The command **echo 4** returns the logging level to its default value.

Each COS node also includes a cos_stats utility that can be executed from a shell. The utility reports the current operational state of the COS node with respect to Swift and Swauth operations. This utility also provides information on resource and network utilization. The following is a sample output obtained by executing the **cos_stats** utility.

```
Fri Jun 27 13:23:12 PDT 2014
--------------------------------------------------------------------------------
| Operation              |    Active |   Ops/Sec |    Errors |     Total  |
--------------------------------------------------------------------------------
```

```
| SWAUTH Account Create   |         0 |       0.0 |         0 |         6  |
| SWAUTH Account Delete   |         0 |       0.0 |         0 |         0  |
| SWAUTH Account Meta     |         0 |       0.0 |         0 |         0  |
| SWAUTH User Create      |         0 |       0.0 |         0 |         6  |
| SWAUTH User Delete      |         0 |       0.0 |         0 |         0  |
| SWIFT Container Create  |         0 |       0.0 |         0 |         0  |
| SWIFT Container Delete  |         0 |       0.0 |         0 |         0  |
| SWIFT Container List    |         0 |       0.0 |         0 |         0  |
| SWIFT Container Meta    |         0 |       0.0 |         0 |      3298  |
| SWIFT Object Delete     |         0 |       0.0 |         0 |         0  |
| SWIFT Object List       |         0 |       0.0 |         0 |         0  |
| SWIFT Object Meta       |         0 |       0.0 |         0 |         0  |
| SWIFT Object Read       |      3999 |      1.9K |         0 |     13818  |
| SWIFT Object Write      |         0 |       0.0 |         0 |   6694580  |
| SWIFT Token Get         |         0 |       0.0 |         0 |         6  |
-------------------------------------------------------------------------------
|    Eth |  Sessions | TCP RX bps | TCP RX pps | TCP TX bps | TCP TX pps    |
-------------------------------------------------------------------------------
|   eth6 |      2004 |   196.320M |   407.518K |     8.511G |   716.957K    |
|   eth7 |      1024 |   170.025M |   352.604K |     8.239G |   688.469K    |
|   eth8 |      1095 |   159.576M |   328.401K |     8.038G |   673.944K    |
|  Total |      4123 |   525.920M |     1.089M |    24.788G |     2.079M    |
-------------------------------------------------------------------------------
| Poll:   0% : 823K/s | NetInt:  62% | RX:  37% | TCP TX:  74% | FQ:   0%  |
-------------------------------------------------------------------------------
```