



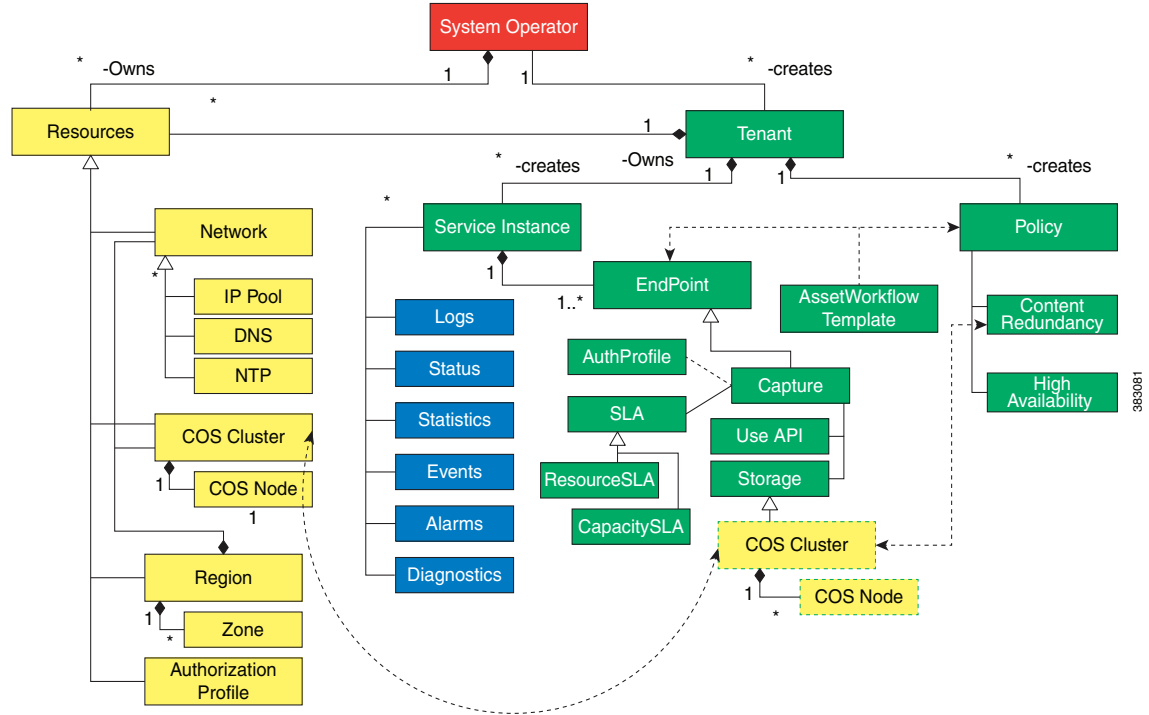
Reference Information

This section contains additional reference material for further understanding the COS system, and information on performing commonly executed tasks and system maintenance.

COS Service Model

The COS service model is shown in the Unified Modeling Language (UML) diagram below:

Figure A-1 COS Service Model



A COS operator can assume one of the following roles:

- System Operator – Provisions the system resources and creates the Tenant. In Figure A-1, the system resources appear in yellow and the entities managed by the Tenant appear in green.

- Tenant – Creates service instances using pre-existing service templates. The Tenant provisions a service instance by assigning system resources to the service instance and configuring it. When the service instance is activated, all related configurations are performed and the service becomes available, producing the outputs appearing in blue in [Figure A-1](#).

COS Network Ports and Services

The following table identifies open network ports for COS nodes and the services that own these ports.

Table A-1 *COS Network Ports and Services*

Scope/Interface	Port	Purpose	Owning Service
Management	TCP 7000	Cassandra internode communication	cassandra
Management	TCP 7199	Cassandra JMX communication	cassandra
Management	TCP 9042	Cassandra CQL native transport port	cassandra
Management	TCP 9160	Cassandra Thrift client API	cassandra
Management	TCP 9090	Cosd request listener	cosd
Data	TCP 80	HTTP traffic for Swift and Swauth interfaces	cserver
Data	UDP 3478	STUN traffic	cserver
Data	UDP 48879	Internal COS node communication	cserver
Data	UDP 57005	Internal COS node communication	cserver
Management	UDP 123	Network Time Protocol	ntpd
Management	TCP 25	Postfix mail system traffic	postfix
Local	TCP 199	Simple Network Management Protocol	snmpd
Management	UDP 161	Simple Network Management Protocol	snmpd
Management	UDP 162	SNMP Traps	snmpdtrapd
Management	TCP 22	Secure Shell (SSH)	ssh

COS Maintenance

It may be necessary to reboot or shut down a COS node for conditions such as routine maintenance. You can reboot or shut down a COS node by placing it in Maintenance mode from either the command line or the COS Service Manager GUI. This section describes both methods.



Note

Putting a COS node in Maintenance mode shuts down the entire cluster to which it belongs. So, when using Maintenance mode, take care to avoid any impact to services provided by the affected COS cluster.

Command Line Reboot

To reboot a COS node, execute the **reboot** command from a terminal console or remote shell. The system will begin a shutdown phase to reboot.

- Any active HTTP and TCP sessions with the COS node data network interfaces will be reset by the node. The client will have the responsibility of retrying operations with the remaining COS nodes in the cluster.
- The COS management system will automatically update the DNS registry to remove listings for the COS node.
- The COS services will automatically be restarted when the system is back online after reboot.
- If the COS cluster has been configured to replica data, copies of any object data residing on the COS node will be accessible from the remaining COS nodes.

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 Service Manager GUI or the API.

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

As decommissioning can take several hours, the CLI does not monitor the decommissioning process for completion. To check for completion, enter the command **cserver-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 Service Manager GUI or the API. For instructions on removing a node from a cluster using the Service Manager GUI, see [Removing Decommissioned Nodes, page B-11](#). For API information, see the *Cisco Cloud Object Storage Release 3.5.1 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 remove a node from a multi-node cluster through the GUI, 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 continue to be considered part of the Cassandra cluster and listed in the **nodetool status** output of the remaining nodes, but now in down (DN) state, which prevents you from adding new nodes to the cluster.

To avoid this issue, we recommend opening the COS AIC Client log before removing the node through the GUI, and periodically inspecting the log to confirm that Cassandra decommissioning is completed before shutting down the node.

To inspect the log for node decommissioning from the Cassandra cluster:

- Step 1** Use the Linux **tail** command to print new lines being added to the COS AIC Client log, followed by the Linux **grep** command to search for **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 **db-remove**:

```
2016-05-06 23:01:29 UTC 127.0.0.1 aicc - Starting db-remove
```

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

```
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 using **tail** (or **cat**) followed by **grep** for **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 completion of the removal process, 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.

Behavior of COS Services on COS Node Boot

There are four primary system services that provide COS functionality on a COS node: `cassandra`, `cosd`, `cos_aicc`, and `cserver`. These services can be manipulated using standard Linux system service tools.

To prevent these services from starting automatically on a COS node boot, execute the following commands as the root user from a shell on that node:

```
[root@cos-node-1 ~]# chkconfig cassandra off
[root@cos-node-1 ~]# chkconfig cosd off
[root@cos-node-1 ~]# chkconfig cos_aicc off
[root@cos-node-1 ~]# chkconfig cserver off
```

To enable automatic service loading on node boot, execute the following commands:

```
[root@cos-node-1 ~]# chkconfig cassandra on
[root@cos-node-1 ~]# chkconfig cosd on
[root@cos-node-1 ~]# chkconfig cos_aicc on
[root@cos-node-1 ~]# chkconfig cserver on
```

To view the current state of service loading, execute the following command:

```
[root@cos-node-1 ~]# chkconfig --list
auditd          0:off  1:off  2:on   3:on   4:on   5:on   6:off
cos_aicc        0:off  1:off  2:on   3:on   4:on   5:on   6:off
cosd            0:off  1:off  2:on   3:on   4:on   5:on   6:off
```

cassandra	0:off	1:off	2:on	3:on	4:on	5:on	6:off
cron	0:off	1:off	2:on	3:on	4:on	5:on	6:off
cserver	0:off	1:off	2:on	3:on	4:on	5:on	6:off
ip6tables	0:off	1:off	2:on	3:on	4:on	5:on	6:off
iptables	0:off	1:off	2:on	3:on	4:on	5:on	6:off
netconsole	0:off	1:off	2:off	3:off	4:off	5:off	6:off
netfs	0:off	1:off	2:off	3:on	4:on	5:on	6:off
network	0:off	1:off	2:on	3:on	4:on	5:on	6:off
ntpd	0:off	1:off	2:off	3:off	4:off	5:off	6:off
ntpdate	0:off	1:off	2:off	3:off	4:off	5:off	6:off
postfix	0:off	1:off	2:on	3:on	4:on	5:on	6:off
rdisc	0:off	1:off	2:off	3:off	4:off	5:off	6:off
restorecond	0:off	1:off	2:off	3:off	4:off	5:off	6:off
rsyslog	0:off	1:off	2:off	3:off	4:off	5:off	6:off
saslauthd	0:off	1:off	2:off	3:off	4:off	5:off	6:off
sshd	0:off	1:off	2:on	3:on	4:on	5:on	6:off
syslog-ng	0:off	1:off	2:on	3:on	4:on	5:on	6:off
udev-post	0:off	1:on	2:on	3:on	4:on	5:on	6:off

To manually start the services, execute the following commands:

```
[root@cos-node-1 ~]# service cassandra start
[root@cos-node-1 ~]# service cosd start
[root@cos-node-1 ~]# service cos_aicc start
[root@cos-node-1 ~]# service cserver start
```

COS Service Reliability

This section describes the response of the COS AIC server to changes in the states of disks, interfaces, and services on COS nodes. The COS AIC client and the Service Monitor convey the changes in state by sending appropriate events to the COS AIC server.

If a change is critical and indicates that the node cannot service requests, the AIC server ensures that the node interfaces are not part of the DNS so that service requests are not addressed to the node.

COS Node Disks

When the COS AIC server receives an event indicating the failure of all the disks on a COS node, it checks the status of the node in its records.

- If the COS node disk status is **down**, the server removes the node interfaces from the DNS server.
- If the COS node disk status is **up**, no action is necessary.

When the COS AIC server receives an event indicating some or all of the disks on the COS node are functional, it checks the status of the node in its records.

- If the COS node disk status is **down**, the server removes the node interfaces from the DNS server.
- If the COS node disk status is **up**, no action is necessary.

COS Services

The COS AIC server may receive an event indicating that one or more of the following COS services is not functional:

- Cisco Cache Server (CServer)

- Cisco Cloud Object Storage Daemon (cosd)
- Cassandra Server

If the COS AIC server receives such an event, it checks the status of the node in its records.

- If the node status is **down**, the server removes the node interfaces from the DNS server.
- If the node status is **up**, no action is necessary.

When the COS AIC server receives an event indicating all the COS services are operational, it checks the status of the node in its records.

- If the node status is **down**, the server removes the node interfaces from the DNS server.
- If the node status is **up**, no action is necessary.

COS Node Interfaces

When the COS AIC server receives an event indicating that one or more interfaces are not functional, along with a list of the faulty interfaces, it checks the status of these interfaces in its records.

- If the faulty interfaces are part of the DNS, they are removed from it.
- If an interface is not in the list of faulty interfaces and is not part of the DNS, the AIC server adds the interface to the DNS server.

Server Reachability

The Service Monitor periodically receives the status of a COS node from the COS AIC client on that node. If the Service Monitor does not receive status information for a COS node within the stipulated time, it determines that the node is not reachable. Consequently, it sends an event to the COS AIC server indicating that the node is down. In response, the server checks the status of the node in its records.

- If the node status is **down**, the server removes the node interfaces from the DNS server.
- If the node status is **up**, no action is necessary.

When the Service Monitor receives status information for a COS node indicating that the node is reachable, it sends an event to the COS AIC server conveying that the node is up. In response, the server checks the status of the node in its records.

- If the node status is **down**, the server removes the node interfaces from the DNS server.
- If the node status is **up**, no action is necessary.

In summary:

- If one or more services (cassandra, cserver, cosd) is down, the COS AIC client removes the interfaces from the DNS entry.
- If all service are up, the COS AIC client adds or keeps all of the interfaces in the DNS entry.

COS Node Hard Drive Replacement

For instructions on replacing hard drives on the platforms that COS 3.5.2 supports, see the appropriate hardware installation guide.

**Note**

Before replacing a hard drive that is not listed as sick, you must first *logically* remove the drive to stop any data transfer in progress and spin down the drive. To do this, execute the command **cddm -r n**, where *n* is the drive number. Do not proceed until a response confirms that it is safe to remove the drive.

Replicating Objects During Swift Write Operations

While an object is being created or modified using Swift write operations, copies of the object data can be stored in real time on the local COS node and its peer nodes in the COS cluster. This functionality works only if the RAID feature on the node is disabled.

To disable the RAID feature on the node, open `/arroyo/test/setupfile` and set **allow vault raid** to **0**.

To replicate object data on the node, open `/arroyo/test/setupfile` and set **vault local copy count** to a value greater than **1**. This value specifies the how many copies of the object data are to be stored on the node.

To replicate object data on the peer nodes, open `/arroyo/test/setupfile` and set **vault mirror copies** to a value greater than **1**. This value specifies how many remote copies of the object data are maintained.

