Cisco Policy Suite 7.0.1 Alarming and SNMP Guide

First Published: January 28, 2015

Last Updated: July 10, 2015

Cisco Systems, Inc.
www.cisco.com

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

Preface

Welcome to the *Cisco Policy Suite 7.0.1 Alarming and SNMP Guide*.

The Cisco Policy Suite (CPS) is a carrier-grade, policy, and subscriber data management software solution that helps service providers control, monetize, and personalize network service offerings like Wi-Fi and BNG (Broadband Network Gateway).

This document shows how to monitor CPS with operational trending information, and how to manage CPS based on system notifications. Proactive monitoring in this way increases service availability and system usability,

Monitoring and alert notifications are provided via Network Monitoring Solutions (NMS) standard Simple Network Management Protocol (SNMP) methodologies.

This preface covers the following topics:

Audience

This guide is best used by these readers:

- Deployment engineers
- Implementation engineers
- Network administrators
- Network engineers
- Network operators
- System administrators

This document assumes a general understanding of network architecture and systems management. Specific knowledge of the SNMP, specifically Version 2c, is required. Installation and initial configuration of CPS is a prerequisite.

Monitoring and Alert Notification

Revised: July 10, 2015

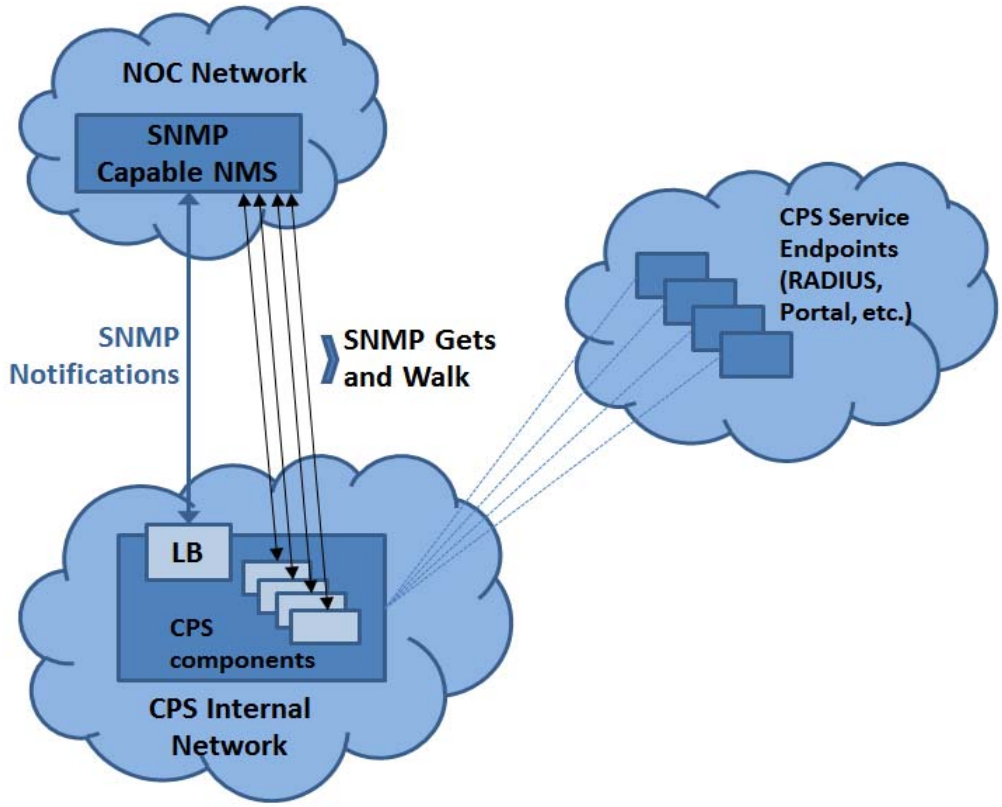
This chapter covers the following sections:

Architectural Overview

A Cisco Policy Suite deployment comprises of multiple virtual instances deployed for scaling and High Availability (HA) purposes. All VMs present in the system should have an IP which is a routable IP to NMS. The NMS can monitor each VM using this routable IP.

During runtime any number of VMs can be added to the system and NMS can monitor them using their routable IP which makes the system more scalable. The notification alerting from the entire system derives from a single point.

When CPS is deployed in a High Availability (HA), alerting endpoints are deployed as HA as well. This is shown in the illustration below.



Technical Architecture

The Cisco Policy Suite is deployed as a distributed virtual appliance. The standard architecture uses VMware ESXi virtualization. Multiple physical hardware host components run VMware ESXi, and each host runs several virtual machines. Within each virtual machine, one-to-many internal CPS components can run. The CPS monitoring and alert notification infrastructure simplifies the virtual, physical, and redundant aspects of the architecture.

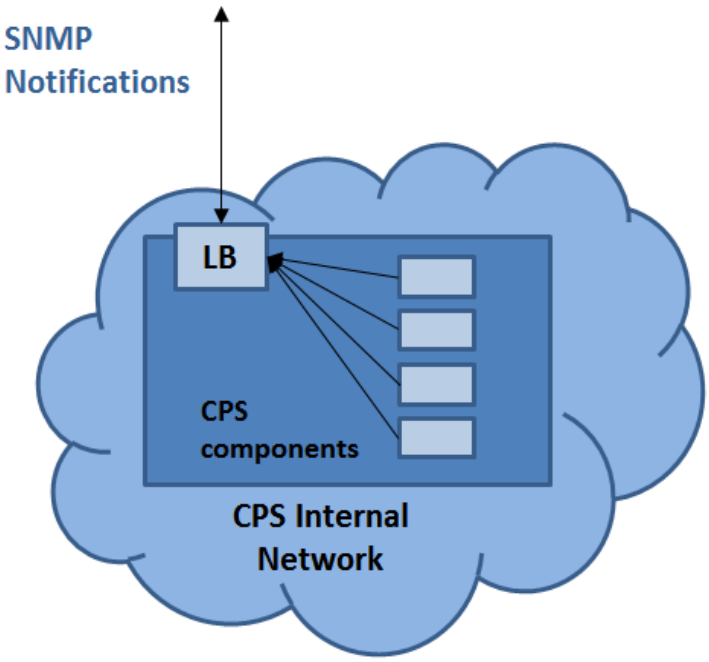
This section covers the following topics:

- Protocols and Query Endpoints
- SNMP Object Identifier and Management Information Base
- SNMPv2 Data and Notifications
- Facility
- Severity
- Categorization
- Emergency Severity Note

Protocols and Query Endpoints

The CPS monitoring and alert notification infrastructure provides a simple, standards-based interface for network administrators and NMS. SNMPv2 is the underlying protocol for all monitoring and alert notifications. Standard SNMPv2 gets and notifications (traps) are used throughout the infrastructure.

The following illustration shows the aggregation and mapping on the SNMP endpoint (LB).



SNMP Object Identifier and Management Information Base

Cisco has a registered private enterprise Object Identifier (OID) of 26878. This OID is the base from which all aggregated CPS metrics are exposed at the SNMP endpoint. The Cisco OID is fully specified and made human-readable through a set of Cisco Management Information Base (MIB-II) files.

The current MIBs are defined as follows:

MIB Filename	Purpose
BROADHOP-MIB.mib	Defines the main structure, including structures and codes.
CISCO-QNS-MIB.mib	Defines the retrievable statistics and KPI.
BROADHOP-NOTIFICATION-MIB.mib	Defines Notifications/Traps available.

A graphical overview of the CPS OID and MIB structure is shown in the next figure.

Note that in the above illustration the entire tree is not shown.

SNMPv2 Data and Notifications

The Monitoring and Alert Notification infrastructure provides standard SNMPv2 get and getnext access to the CPS system. This provides access to targeted metrics to trend and view Key Performance Indicators (KPI). Metrics available through this part of the infrastructure are as general as component load and as specific as transactions processed per second.

SNMPv2 Notifications, in the form of traps (one-way) are also provided by the infrastructure. CPS notifications do not require acknowledgments. These provide both proactive alerts that pre-set thresholds have been passed (for example, Disk is nearing full, CPU load high) and reactive alerting when system components fail or are in a degraded state (for example, DEAD PROCESS, network connectivity outages, etc.).

Notifications and traps are categorized by a methodology similar to UNIX System Logging (syslog) with both Severity and Facility markers. All event notifications (traps) contain these items:

- Facility
- Severity

- Source (device name)
- Device time

These objects enable Network Operations Center (NOC) staff to identify where the issue lies, the Facility (system layer), and the Severity (importance) of the reported issue.

Facility

The generic syslog Facility has the following definitions.

Note Facility defines a system layer starting with physical hardware and progressing to a process running in a particular application.

Number	Facility	Description
0	Hardware	Physical Hardware – Servers, SAN, NIC, Switch, etc.
1	Networking	Connectivity in the OSI (TCP/IP) model.
2	Virtualization	VMware ESXi (or other) Virtualization
3	Operating System	Linux, Microsoft Windows, etc.
4	Application	Apache httpd, load balancer, CPS Cisco sessionmgr, etc.
5	Process	Particular httpd process, CPS qns01_A, etc.

There may be overlaps in the Facility value as well as gaps if a particular SNMP agent does not have full view into an issue. The Facility reported is always shown as viewed from the reporting SNMP agent.

Severity

In addition to Facility, each notification has a Severity measure. The defined severities are directly from UNIX syslog and defined as follows:

Number	Severity	Description
0	Emergency	System is unusable.
1	Alert	Action must be taken immediately.
2	Critical	Critical conditions.
3	Error	Error conditions.
4	Warning	Warning conditions.
5	Notice	Normal but significant condition.
6	Info	Informational message.
7	Debug	Lower level debug messages.
8	None	Indicates no severity.
9	Clear	The occurred condition has been cleared.

For the purposes of the CPS Monitoring and Alert Notifications system, Severity levels of Notice, Info and Debug are usually not used. Warning conditions are often used for proactive threshold monitoring (for example, Disk usage or CPU Load), which requires some action on the part of administrators, but not immediately. Conversely, Emergency severity indicates that some major component of the system has failed and that either core policy processing, session management or major system function is impacted.

Categorization

Combinations of Facility and Severity create many possibilities of notifications (traps) that might be sent. However, some combinations are more likely than others. The following table lists some noteworthy Facility and Severity categorizations.

Facility.Severity	Categorization	Possibility
Process.Emergency	A single part of an application has dramatically failed.	Possible, but in an HA configuration very unlikely.
Hardware.Debug	A hardware component has sent a debug message.	Possible but highly unlikely.
Operating System.Alert	An Operating System (kernel or resource level) fault has occurred.	Possible as a recoverable kernel fault (on a vNIC for instance).
Application.Emergency	An entire application component has failed.	Unlikely but possible (load balancers failing for instance).
Virtualization.Emergency	The virtualization system has thrown a fault.	Unlikely but possible (VM won't start, or vSwitch fault for instance.

It is not possible quantify every Facility and Severity combination. However, greater experience with CPS leads to better diagnostics. The CPS Monitoring and Alert Notification infrastructure provides a baseline for event definition and notification by an experienced engineer.

Emergency Severity Note

Caution Emergency severities are very, very important! As a general principle, CPS does not throw an Emergency level severity unless the system becomes inaccessible or unusable in some way. An unusable system is extremely rare, but might occur if multiple failures occur in the operating system, virtualization, networking or hardware facilities.

SNMP System and Application KPIs

Many CPS system statistics and Key Performance Indicators (KPI) are available via SNMPv2 gets and walks. Both system device level information and application level information is available. This information is well documented in the CISCO-QNS-MIB. A summary of the information available is provided below. This section covers the following topics:

SNMP System KPIs

In this table, the system KPI information is provided.

Component	Information
LB01/LB02	CpuUser
PCRFClient01/PCRFClient02	CpuSystem
SessionMgr01/SessionMgr02	CpuIdle
QNS01/QNS02/QNS03/QNS04	LoadAverage1
Portal01/Portal02	LoadAverage5
	LoadAverage15
	MemoryTotal
	MemoryAvailable
	SwapTotal
	SwapAvailable

Details of SNMP System KPIs

The following information is available, and is listed per component. MIB documentation provides units of measure.

```
+--ciscoProductsQNSComponents70 (70) |
 +--ciscoProductsQNSComponentsSystemStats (1) |
    +-- -R-- Integer32 componentCpuUser(1) |
    +-- -R-- Integer32 componentCpuSystem(2) |
    +-- -R-- Integer32 componentCpuIdle(3) |
    +-- -R-- Integer32 componentLoadAverage1(4) |
    +-- -R-- Integer32 componentLoadAverage5(5) |
    +-- -R-- Integer32 componentLoadAverage15(6) |
    +-- -R-- Integer32 componentMemoryTotal(7) |
    +-- -R-- Integer32 componentMemoryAvailable(8) |
    +-- -R-- Integer32 componentSwapTotal(9) |
    +-- -R-- Integer32 componentSwapAvailable(10) |
```

SNMP Application KPIs

Current version Key Performance Indicators (KPI) information is available at the OID root of:

```
.1.3.6.1.4.1.26878.200.3.3.70
```

This corresponds to an MIB of:

```
.iso
.identified-organization
.dod
.internet
.private
.enterprise
.broadhop
.broadhopProducts
.ciscoProductsQNS
.ciscoProductsQNSConsolidatedKPIVersion
.ciscoProductsQNSKPI70
```

Summary of SNMP Application KPIs

The following application KPI's are available for monitoring on each node using SNMP Get and Walk utilities.

Component	Information
LB01/LB02	PCRFProxyExternalCurrentSessions: is the number of open connections to lbvip01:8443. PCRFProxyInternalCurrentSessions: is the number of open connections to lbvip02:8080
PCRFClient01/PCRFClient02	----------
SessionMgr01/SessionMgr02	----------

Component	Information
QNS01/QNS02/QNS03/QNS04	PolicyCount: The number of processed policy messages. QueueSize: The number of entries in the processing queue. The default queue size is 500, but this can be configured by the customer in the Policy Builder. You can also see the number of dropped messages in the statistics files. FailedEnqueueCount ErrorCount SessionCount FreeMemory
Portal01/Portal02	----------

Details of Supported KPIs

The following information is available, and is supported in current release. MIB documentation provides units of measure.

```
+--ciscoProductsQNSKPILB(11)
| |
| +-- -R-- String kpiLBPCRFProxyExternalCurrentSessions(1)
| |      Textual Convention: DisplayString
| |      Size: 0..255
| +-- -R-- String kpiLBPCRFProxyInternalCurrentSessions(2)
|        Textual Convention: DisplayString
|        Size: 0..255

+--ciscoProductsQNSKPISessionMgr(14)
+--ciscoProductsQNSKPIQNS(15)
| |
| +-- -R-- Integer32 kpiQNSPolicyCount(20)
| +-- -R-- Integer32 kpiQNSQueueSize(21)
| +-- -R-- Integer32 kpiQNSFailedEnqueueCount(22)
| +-- -R-- Integer32 kpiQNSErrorCount(23)
| +-- -R-- Integer32 kpiQNSAggregateSessionCount(24)
| +-- -R-- Integer32 kpiQNSFreeMemory(25)
```

Notifications and Alerting (Traps)

The CPS Monitoring and Alert Notification framework provides the following SNMPv2 notification traps (one-way). Traps are either proactive or reactive. Proactive traps are alerts based on system events or changes that require attention (for example, Disk is filling up). Reactive traps are alerts that an event has already occurred (e.g., an application process died).

This section covers the following topics:

Component Notifications

Components are devices that make up the CPS system. These are systems level traps. They are generated when some predefined thresholds are crossed. User can define these thresholds in `/etc/snmp/snmpd.conf`. For example, for disk full, low memory etc. Process snmpd is running on all the VMs. When process snmpd starts, it notes the values set in snmpd.conf. Hence, whenever user makes any change in snmpd.conf, the user must execute command

service snmpd restart

For example, if threshold crosses, snmpd throws a trap to LBVIP on the internal network on port 162. On LB, process snmptrapd is listening on port 162. When snmptrap sees trap on 162, it logs it in the file `/var/log/snmp/trap` and re-throws it on corporate_nms_ip on port 162. This corporate NMS IP is set inside `/etc/hosts` file on LB1 and LB2. Typically, these components equate to running Virtual Machines.

Note For more information related to `corportate_nms_ip`, refer *SNMP Traps and Key Performance Indicators (KPIs)* section in *CPS 7.0.1 Troubleshooting Guide*.

Component notifications are defined in the BROADHOP-NOTIFICATION-MIB as follows:

```
broadhopQNSComponentNotification NOTIFICATION-TYPE
        OBJECTS { broadhopComponentName,
                  broadhopComponentTime,
                  broadhopComponentNotificationName,
                  broadhopNotificationFacility,
                  broadhopNotificationSeverity,
                  broadhopComponentAdditionalInfo }
        STATUS current
        DESCRIPTION "
                     Trap from any QNS component - i.e. device.
                    "
        ::= { broadhopProductsQNSNotifications 1 }
```

Each Component Notification contains:

- Name of the device throwing the notification (broadhopComponentName)
- Time the notification was generated (broadhopComponentTime)
- Facility or which layer the notification came from (broadhopNotificationFacility)
- Severity of the error (broadhopNotificationSeverity)
- Additional information about the notification, which might be a bit of log or other information.

Component Notifications that CPS generates are shown in the following list. Any component in the CPS system may generate these notifications.

Name	Feature	Severity	Message Text
Disk Full: This alarm gets generated for following file systems in different VMs: • For HA System: – pcrf/lb: /; /var; /boot – sessionmgr: /; /home; /boot; /data; /var/data/session.1 – qns: /; /home; /var; /boot – portal: /; /var; /boot; /home; /tmp; /usr; /var/www • For AIO System: – / – /boot	Component	Warning	DiskFullAlert
	Current disk usage has passed a designated threshold. This situation may resolve on its own, but could be a sign of logs or database files growing large.		
Disk Full Clear: This alarm gets generated for following file systems in different VMs: • For HA System: – pcrf/lb: /; /var; /boot – sessionmgr: /; /home; /boot; /data; /var/data/session.1 – qns: /; /home; /var; /boot – portal: /; /var; /boot; /home; /tmp; /usr; /var/www • For AIO System: – / – /boot	Component	Clear	DiskFullClear
	Current Disk usage has recovered the designated Threshold.		

Name	Feature	Severity	Message Text
Load Average of local system: The alarm gets generated for 1 minute 5 minute 15 minute Average	Component	Warning (1, 5 minutes) Alert (15 minutes)	HighLoadAlert
	Current CPU load is more than configured threshold for 1/5/15 minutes.		
Load Average Clear of local system: The alarm gets generated for 1 minute 5 minute 15 minute Average	Component	Clear	HighLoadClear
	Current CPU load has recovered from more than configured threshold.		
Low Swap memory alarm	Operating System	Warning	LowSwapAlert
	Current swap usage has passed a designated threshold. This is a warning.		
Low Swap memory clear	Operating System	Clear	LowSwapClear
	Current swap usage has recovered a designated threshold.		
Interface/Link Down Alarm: This alarm gets generated for all physical interface attached to the system.	Operating System	Alert	<Interface Name> is Down
	Not able to connect or ping to the interface.		
Interface/Link Up Alarm: This alarm gets generated for all physical interface attached to the system.	Operating System	Clear	<Interface Name> is Up
	Able to ping or connect to interface.		
Low Memory Alert Alarm	Operating System	Warning	LowMemoryAlert
	Low memory alert.		
Low Memory Clear Alarm	Operating System	Info	LowMemoryClear
	Low memory alert.		
Corosync Alert Alarm	Component	Critical	Corosync Process is Down
	User gets the critical alarm whenever corosync process is down.		
Corosync Clear Alarm	Component	Clear	Corosync Process is UP
	The alarm is cleared whenever corosync process is UP.		

Application Notifications

Applications are running processes on a component device that make up the CPS system. These are application level traps. CPS process (starting with word java when we run "ps -ef") and some scripts (for GR traps) generates these traps.

For example, when a trap is generated, it is thrown to LBVIP on internal network (can be on port 162. On LB, process snmptrapd is listening on port 162. When snmptrap sees trap on 162, it logs it in the file `/var/log/snmpd/trap` and re-throws it on corporate_nms_ip on port 162. This corporate NMS IP is set inside /etc/hosts file on LB01 and LB02.

Note For more information related to `corportate_nms_ip`, refer *SNMP Traps and Key Performance Indicators (KPIs)* section in *CPS 7.0.1 Troubleshooting Guide*.

Application notifications are defined in the BROADHOP-NOTIFICATION-MIB as follows:

```
broadhopQNSApplicationNotification NOTIFICATION-TYPE
       OBJECTS { broadhopComponentName,
                 broadhopComponentTime,
                 broadhopComponentNotificationName,
                 broadhopNotificationFacility,
                 broadhopNotificationSeverity,
                 broadhopComponentAdditionalInfo }
       STATUS current
       DESCRIPTION "
                   Notification Trap from any QNS application - i.e.,
runtime.
                   "
       ::= { broadhopProductsQNSNotifications 2 }
```

Each Application Notification contains these elements:

- Name of the device throwing the notification (broadhopComponentName)
- Time the notification was generated (broadhopComponentTime)
- Facility or which layer the notification came from (broadhopNotificationFacility)
- Severity of the error (broadhopNotificationSeverity)
- Additional information about the notification, which might be a portion of log or other information

Application Notifications that CPS generates are shown in the following list. Any application in CPS system may generate these notifications.

Name	Feature	Severity	Message Text
MemcachedConnect Error	Application	Error	Memcached server is in error: OR some exception generated message.
	Generated if attempting to connect to or write to the memcached server causes an exception. %s is the exception that occurred.		
	Application	Clear	Memcached server is operational.
	Generated if successfully connect to or write to the memcached server.		

Name	Feature	Severity	Message Text
ApplicationStartError	Application	Alert	Feature %s is unable to start. Error %s
	Generated if an installed feature cannot start.		
	Application	Clear	Feature %s is Running
	Generated if an installed feature successfully started.		
LicensedSessionCreation	Application	Critical	Session creation is not allowed
	A predefined threshold of sessions covered by licensing has been passed. This is a warning and should be reported. License limits may need to be increased soon. This message can be generated by an invalid license, but the AdditionalInfo portion of the notification shows root cause.		
	Application	Clear	Session creation is allowed
	The number of sessions are below the predefined threshold of sessions covered by licensing.		
InvalidLicense	Application	Emergency	“xxx license has not been verified yet”
	The system license currently installed is not valid. This prevents system operation until resolved. This is possible if no license is installed or if the current license does not designate values. This may also occur if any system networking MAC addresses has changed.		
	Application	Emergency	xxx license is invalid %s
	License is invalid.		
	Application	Critical	xxx is Expired %s
	License has expired.		
	Application	Error	xxx will expire soon %s
	License is going to expire soon.		
	Application	Critical	xxx has exceeded the allowed parameters %s
	License has exceeded the allowed parameters.		
	Application	Error	xxx is nearing the allowed parameters %s
	License is reaching allowed parameters.		
PolicyConfiguration	Application	Error	“Last policy configuration failed with the following message:xxx”
	A change to system policy structure has failed. The AdditionalInfo portion of the notification contains more information. The system typically remains in a proper state and continues core operations. Either make note of this message or investigate more fully.		
	Application	Clear	“Last policy configuration was successful”
	A change to system policy structure has passed.		

Name	Feature	Severity	Message Text
PoliciesNot	Application	Emergency	“Policies not configured”
Configured	The policy engine cannot find any policies to apply while starting up. This may occur on a new system, but requires immediate resolution for any system services to operate.		
	Application	Clear	“Policies successfully configured”
	The policy engine has successfully configured all the policies while starting up.		
DiameterPeerDown	Application	Error	host $HOST realm: %s is down
	Diameter peer is down.		
	Application	Clear	host $HOST realm: %s is up
	Diameter peer is up.		
DiameterAllPeersDown	Application	Critical	Realm: <<xxx.com>> all peers are down
	All diameter peer connections configured in a given realm are DOWN (i.e. connection lost). The alarm identifies which realm is down. The alarm is cleared when at least one of the peers in that realm is available.		
	Application	Clear	Realm: <<xxx.com>> peers are up
	The diameter peer connections configured in a given realm are up.		
HA_Failover	Application	Critical	HA Failover done from %s to %s of ${SET_NAME}-SET$Loop
	Primary member of replica is down and taken over by other primary member of same replica set.		
All DB Member of replica set Down	Application	Critical	All replicas of ${SET_NAME}-SET$Loop are down
	Not able to connect to any member of replica set.		
All DB Member of replica set Up	Application	Clear	All replicas of ${SET_NAME}-SET$Loop are up
	Able to connect to all members of replica set.		
No Primary DB Member Found	Application	Critical	Unable to find primary member for Replica-set ${SET_NAME}-SET$Loop
	Unable to find primary member for replica-set.		
Primary DB Member Found	Application	Clear	found primary member for Replica-set ${SET_NAME}-SET$Loop
	Found primary member for replica-set.		
Secondary DB Member Down	Application	Critical	Secondary DB %member_ip:%mem_port (%mem_hostname) of SET $SET is down
	In replica set, secondary DB member is not able to connect.		

Name	Feature	Severity	Message Text
Secondary DB Member up	Application	Clear	Secondary DB %member_ip:%mem_port (%mem_hostname) of SET $SET is up
	In replica set, secondary DB member is able to connect.		
Arbiter Down	Application	Critical	Arbiter %member_ip:%mem_port (%mem_hostname) of SET $SET is down
	In replica set, the administrator is not able to connect to configured arbiter.		
Arbiter Up	Application	Clear	Arbiter %member_ip:%mem_port (%mem_hostname) of SET $SET is up
	In replica set, the administrator is able to connect to configured arbiter.		
Config Server Down	Application	Critical	Config Server %member_ip:%mem_port (%mem_hostname) of SET $SET is down
	In replica set, the administrator is not able to connect to Configured Config Server.		
Config Server Up	Application	Clear	Config Server %member_ip:%mem_port (%mem_hostname) of SET $SET is up
	In replica set, the administrator is able to connect to Configured Config Server.		
VM Down	Application	Critical	unable to connect %member_ip (%member) VM. It is not reachable.
	The administrator is not able to ping to VM (This alarms gets generated for all VMs configured inside /etc/hosts of lb).		
VM Up	Application	Clear	Connected %member_ip (%member) VM. It is reachable.
	The administrator is able to ping to VM (This alarms gets generated for all VMs configured inside `/etc/hosts` of lb).		
QPS Process Down	Application	Critical	%server server on %vm vm is down
	CPS java process is down.		
QPS Process Up	Application	Clear	%server server on %vm vm is up
	CPS java process is up.		
Admin Logged In	Application	info	root user logged in on %hostname terminal %terminal from machine %from_system at %time
	root user logged in on %hostname terminal.		

Name	Feature	Severity	Message Text
DeveloperMode	Application	Warning	Using POC/Development license (100session limit). To use a license file, remove -Dcom.broadhop.developer.modefrom /etc/broadhop/qns.conf
	Generated if developer mode is configured in `qns.conf`.		
	Application	Clear	Removed -Dcom.broadhop.developer.mode from /etc/broadhop/qns.conf
	Generated if developer mode is removed in `qns.conf`.		
ZeroMQConnection Error	Application	Error	ZMQ Connection Down for tcp://%s:%d
	Internal services cannot connect to a required Java ZeroMQ queue. Although retry logic and recovery is available, and core system functions should continue, investigate and remedy the root cause.		
	Application	Clear	ZMQ Connection Up for tcp://%s:%d
	Internal services can connect to a required Java ZeroMQ queue.		
VIRTUALInterfaceError	Network	Alert	unable to connect %INTERFACE(lbvip01/lbvip02) VM. Not reachable
	Not able to ping the virtual Interface. This alarm gets generated for lbvip01,lbvip02.		
	Network	Clear	Connected %INTERFACE(lbvip01/lbvip02) VM.
	Successfully ping the virtual Interface. This alarm gets generated for lbvip01,lbvip02.		
LdapAllPeersDown	Application	Error	1201:LDAP connection down
	All LDAP peers are down. CPS does not generate LDAP peer connection status alarms for individual LDAP peer.		
	Application	Clear	1201:LDAP connection up
	LDAP connection is up. CPS does not generate LDAP peer connection status alarms for individual LDAP peer.		

Unknown Application Events

In CPS, according to SNMP architecture, to implement any alarm it requires changes at multiple places e.g. java application, snmp scripts etc. All the alarms generated by different VMs are received at LB VMs.

On LB VMs, a script called `application_trapv1_convert` processes the received alarms and generates the new alarm based on the received information and sends it to external NMS. Unknown alarms can come when `application_trapv1_convert` is not able to process the received alarm. In this case, it will generate one of the below seven unknown alarms.

Name	Severity	Facility
ApplicationEvent	None	—
DBEvent	None	—
FailoverEvent	None	—
ProcessEvent	None	—
VMEvent	None	—
None	None	Application
UnKnown	None	None

Note Any unknown alarms should get reported to engineering team to take necessary action against it. Provide the alarm log i.e., `/var/log/snmp/trap` file from active LB VMs with the ticket number.

Configuration and Usage

All access to system statistics and KPIs should be collected via SNMP gets and walks from the routable IP of the VM. NMS sends the `snmpwalk` or `snmpget` request on routable IP of the VM and gets the response. NMS should know routable IP of all the VMs available in the setup. System Notification are sourced from lbvip01.

Configuration of the system consists of following:

- Configuration for SNMP gets and walks
- Configuration for Notifications (traps)
- SNMPD Configuration for Memory
- Cluster Manager KPI and SNMP Configuration
- Validation and Testing

Configuration for SNMP gets and walks

At the time of installation, SNMPv2 gets and walks can be performed against the routable/public IP of the VM with the default read-only community string of Cisco using standard UDP port 161.

The read-only community string can be changed from its default of Cisco to a new value using the following steps:

Step 1 SSH to lb01 as the root user.

Step 2 With vi or nano, edit the file `/etc/snmp/snmpd.conf`.

Caution DO NOT CHANGE THE EXISTING line that reads: `rocommunity Broadhop`. Changing this line breaks SNMP framework functionality

Step 3 Add a NEW rocommunity line under the exiting with your new read-only string. The line should read: `rocommunity <string>`. Replace `<string>` with your desired community string value.

Step 4 Save and exit the file editor.

Step 5 scp `/etc/snmp/snmpd.conf lb02:/etc/snmp`.

Step 6 Restart snmpd service on lb01.

service snmpd restart

Step 7 SSH lb02 and restart snmpd service

service snmpd restart

SNMPv2 gets and walks should now be accessible via the new rocommunity string. Changing the port from the default of 161 is not covered in this guide.

Caution Do not change the existing values of rocommunity, or trap2sink - this prevents the SNMP framework from functioning correctly.

Configuration for Notifications (traps)

After the previous configurations have been made, notifications should be logged locally in the `/var/log/snmp/trap` file as well as forwarded to the NMS destination at `corporate_nms_ip`. By default, traps are sent to the destination corporate_nms_ip using the SNMPv2 community string of Cisco. The standard SNMP UDP trap port of 162 is also used. Both of these values may be changed to accommodate the upstream NMS.

To change the trap community string:

Step 1 Ensure the SNMP framework has been installed.

Step 2 SSH to lb01 as the root user.

Step 3 With vi or nano, edit the file `/etc/snmp/scripts/snmp_communities`.

Step 4 Edit the existing line that reads: `trap_community=broadhop`.

The changed line should read: `trap_community=<string>`.

Replace `<string>` with your desired trap community value.

Step 5 Save and exit the file editor.

Step 6 scp /etc/snmp/scripts/snmp_communities lb02:/etc/snmp/scripts

Step 7 service snmpd restart

Step 8 ssh lb02 "service snmpd restart"

To change the destination trap port from 162:

Step 1 To make this change, the `/etc/snmp/snmptrapd.conf` file needs modification on both lb01 and lb02. In these files,

a. Append a colon and the destination port to each line containing corporate_nms_ip. There are a total of 12 lines in each file.

For example, if the NMS destination port were 1162, the line:

```
traphandle DISMAN-EVENT-MIB::mteTriggerFired
/etc/snmp/scripts/component_trap_convert corporate_nms_ip
```

becomes

```
traphandle DISMAN-EVENT-MIB::mteTriggerFired
/etc/snmp/scripts/component_trap_convert corporate_nms_ip:1162
```

Step 2 After these changes, save the file and restart the snmptrapd service to enable changes.

SNMPD Configuration for Memory

SNMPD is an SNMP agent which binds to a port and awaits requests from SNMP management software. Upon receiving a request, it processes the request(s), collects the requested information and/or performs the requested operation(s) and returns the information to the sender.

In CPS Deployment Template file, in General Configuration sheet, set the value of `free_mem_per` configuration variable between 0.0 to 1.0.

By default, the value is 0.10.

`free_memory_per`=0.10, indicates that the Low Memory trap gets raised by the VM when available memory is below 10% of Total Memory.

Cluster Manager KPI and SNMP Configuration

This section describes the steps to enable SNMP trap and KPI monitoring of the Cluster Manager to monitor the following KPIs:

- Memory usage
- Disk usage
- CPU
- Disk IO

KPIs are reported and recorded on the pcrfclient in the `/var/broadhop/stats` file.

SNMP traps are forwarded to lb01/lb02 and lb01/lb02 forwards the traps to the configured NMS servers in the system.

Install NET-SNMP

To install NET-SNMP, perform the following steps:

Step 1 In Cluster Manager, execute the following command to install NET-SNMP package:

```
yum install --assumeyes --disablerepo=QPS-Repository --enablerepo=QPS-local net-snmp
```

Step 2 To enable run levels for SNMP, execute the following command:

```
chkconfig --level 2345 snmpd on
```

SNMPD Configuration

Step 1 Add the following content to `/etc/snmp/snmpd.conf` file.

```
com2sec local     localhost           public
rocommunity broadhop
group MyRWGroup v1         local
group MyRWGroup v2c        local
group MyRWGroup usm        local
view all    included  .1                               80
access MyRWGroup ""      any       noauth    exact  all    all    none
syslocation Unknown (edit /etc/snmp/snmpd.conf)
syscontact Root  (configure /etc/snmp/snmp.local.conf)
master agentx

trapcommunity  public
agentSecName meme
rouser meme

# Send all traps upstream - Don't change this password or it breaks the framework.
# v1 and v2 traps _could_ be sent for all but only need v2 trap.
trap2sink lbvip02 broadhop

ignoreDisk /proc
ignoreDisk /proc/sys/fs/binfmt_misc
ignoreDisk /var/lib/nfs/rpc_pipefs
ignoreDisk /dev/shm
ignoreDisk /dev/pts
disk / 10%
disk /home 10%
disk /var 10%
disk /boot 10%
swap 102400

load 9 9 9

#linkUpDownNotifications yes

notificationEvent  linkUpTrap    linkUp   ifIndex ifAdminStatus ifOperStatus
notificationEvent  linkDownTrap  linkDown ifIndex ifAdminStatus ifOperStatus


monitor  -r 60 -e linkUpTrap -o ifDescr  "Generate linkUp" ifOperStatus != 2
monitor  -r 60 -e linkDownTrap -o ifDescr "Generate linkDown" ifOperStatus == 2

# Note: alert!=0, clear==0 and messages must be unique or snmpd errors.
monitor -u meme -r 60 -o dskPath -o dskErrorMsg "DiskFullAlert" dskErrorFlag != 0
monitor -u meme -r 60 -o dskPath -o dskErrorMsg "DiskFullClear" dskErrorFlag == 0
monitor -u meme -r 60 -o memErrorName -o memSwapErrorMsg "LowSwapAlert" memSwapError != 0
monitor -u meme -r 60 -o memErrorName -o memSwapErrorMsg "LowSwapClear" memSwapError == 0
monitor -u meme -r 60 -o laNames -o laErrMessage  "HighLoadAlert" laErrorFlag != 0
monitor -u meme -r 60 -o laNames -o laErrMessage  "HighLoadClear" laErrorFlag == 0
monitor -u meme -r 60 -o memAvailReal -o memTotalReal "LowMemoryAlert" memAvailReal <
806106
monitor -u meme -r 60 -o memAvailReal -o memTotalReal "LowMemoryClear" memAvailReal >=
806106


##########
#
# System Stats
#
```

```
# User, System and Idle CPU (UCD-SNMP-MIB ss)

proxy -v 2c -c broadhop localhost .1.3.6.1.4.1.26878.200.3.2.70.1.1.0
.1.3.6.1.4.1.2021.11.9.0
proxy -v 2c -c broadhop localhost .1.3.6.1.4.1.26878.200.3.2.70.1.2.0
.1.3.6.1.4.1.2021.11.10.0
proxy -v 2c -c broadhop localhost .1.3.6.1.4.1.26878.200.3.2.70.1.3.0
.1.3.6.1.4.1.2021.11.11.0
proxy -v 2c -c broadhop localhost .1.3.6.1.4.1.26878.200.3.2.70.1.1
.1.3.6.1.4.1.2021.11.9.0
proxy -v 2c -c broadhop localhost .1.3.6.1.4.1.26878.200.3.2.70.1.2
.1.3.6.1.4.1.2021.11.10.0
proxy -v 2c -c broadhop localhost .1.3.6.1.4.1.26878.200.3.2.70.1.3
.1.3.6.1.4.1.2021.11.11.0
# 1, 5 and 15 Minute Load Averages (UCD-SNMP-MIB la)
proxy -v 2c -c broadhop localhost .1.3.6.1.4.1.26878.200.3.2.70.1.4
.1.3.6.1.4.1.2021.10.1.5.1
proxy -v 2c -c broadhop localhost .1.3.6.1.4.1.26878.200.3.2.70.1.5
.1.3.6.1.4.1.2021.10.1.5.2
proxy -v 2c -c broadhop localhost .1.3.6.1.4.1.26878.200.3.2.70.1.6
.1.3.6.1.4.1.2021.10.1.5.3
proxy -v 2c -c broadhop localhost .1.3.6.1.4.1.26878.200.3.2.70.1.4.0
.1.3.6.1.4.1.2021.10.1.5.1
proxy -v 2c -c broadhop localhost .1.3.6.1.4.1.26878.200.3.2.70.1.5.0
.1.3.6.1.4.1.2021.10.1.5.2
proxy -v 2c -c broadhop localhost .1.3.6.1.4.1.26878.200.3.2.70.1.6.0
.1.3.6.1.4.1.2021.10.1.5.3
# Memory Total, Memory Available, Swap Total, Swap Available (UCD-SNMP-MIB mem)
proxy -v 2c -c broadhop localhost .1.3.6.1.4.1.26878.200.3.2.70.1.7
.1.3.6.1.4.1.2021.4.5.0
proxy -v 2c -c broadhop localhost .1.3.6.1.4.1.26878.200.3.2.70.1.8
.1.3.6.1.4.1.2021.4.6.0
proxy -v 2c -c broadhop localhost .1.3.6.1.4.1.26878.200.3.2.70.1.9
.1.3.6.1.4.1.2021.4.3.0
proxy -v 2c -c broadhop localhost .1.3.6.1.4.1.26878.200.3.2.70.1.10
.1.3.6.1.4.1.2021.4.4.0
proxy -v 2c -c broadhop localhost .1.3.6.1.4.1.26878.200.3.2.70.1.7.0
.1.3.6.1.4.1.2021.4.5.0
proxy -v 2c -c broadhop localhost .1.3.6.1.4.1.26878.200.3.2.70.1.8.0
.1.3.6.1.4.1.2021.4.6.0
proxy -v 2c -c broadhop localhost .1.3.6.1.4.1.26878.200.3.2.70.1.9.0
.1.3.6.1.4.1.2021.4.3.0
```

Step 2 After updating the `snmpd.conf` file, execute the following commands from Cluster Manager.

mkdir /etc/snmp/mibs;scp root@qns01:/etc/snmp/mibs/* /etc/snmp/mibs

service snmpd start

Install and Configure Collectd

To install and configure Collectd, perform the following steps:

Step 1 In Cluster Manager, execute the following command to install Collectd package:

yum install --assumeyes --disablerepo=QPS-Repository --enablerepo=QPS-local collectd

Step 2 To enable collectd, execute the following command:

chkconfig --level 2345 collectd on

Step 3 Create a collectd directory and copy the database information by executing the following commands:

mkdir -p /etc/collectd.d

scp root@qns01:/etc/collectd.d/types.db /etc/collectd.d/types.db

Step 4 Add the following content to /etc/collectd.conf file.

```
FQDNLookup   false
BaseDir     "/var/lib/collectd"
PIDFile     "/var/run/collectd.pid"
PluginDir   "/usr/lib64/collectd"
TypesDB     "/usr/share/collectd/types.db"
TypesDB     "/etc/collectd.d/types.db"
Timeout      2
ReadThreads  5

LoadPlugin memory
LoadPlugin cpu
LoadPlugin disk
LoadPlugin df
LoadPlugin network


<Plugin network>
        Server "pcrfclient01" "25826"
        Server "pcrfclient02" "25826"
</Plugin>


<Plugin "memory">
</Plugin>

<Plugin "cpu">
</Plugin>

<Plugin disk>
</Plugin>

<Plugin "df">
</Plugin>

<Plugin load>
</Plugin>
```

Step 5 Restart the collectd service by executing the following command in Cluster Manager:

service collectd start

Validation and Testing

This section describes the commands for validation and testing of the CPS SNMP infrastructure during its development. You can use these commands now to validate and test your system during setup, configuration, or at any point. Our examples use MIB values because they are more descriptive, but you may use equivalent OID values if you like, particularly when configuring an NMS.

The examples here use Net-SNMP snmpget, snmpwalk and snmptrap programs. Detailed configuration of this application is outside the scope of this document, but the examples assume that the three Cisco MIBs are installed in the locations described on the man page of snmpcmd (typically the `/home/share/<user>/.snmp/mibs` or `/usr/share/snmp/mibs` directories).

Validation and testing is of three types and correspond to the statistics and notifications detailed earlier in this document:

- Component Statistics
- Application KPI
- Notifications

Run all tests from a client with network access to the Management Network or from the lb01, lb02, pcrfclient01 or pcrfclient02 hosts (which are also on the Management Network).

Component Statistics

Component statistics can be obtained on a per statistic basis with snmpget. As an example, to get the current available memory on pcrfclient01 use the following command:

```
snmpget -v 2c -c broadhop -M +BROADHOP-MIB:CISCO-QNS-MIB pcrfclient01
    .1.3.6.1.4.1.26878.200.3.2.70.1.8
```

An example of the output from this command is:

```
SNMPv2-SMI::enterprises.26878.200.3.2.70.1.8 = INTEGER: 5977144
```

Interpreting this output means that 5,977,144 MB of memory are available on this component machine.

All available component statistics in an MIB node can be “walked” via the snmpwalk command. This is very similar to snmpget as above. For example, to see all statistics on lb01 use the command:

```
snmpwalk -v 2c -c broadhop -M +BROADHOP-MIB:CISCO-QNS-MIB lb01
    .1.3.6.1.4.1.26878.200.3.2.70
```

An example of the output from this command is:

```
SNMPv2-SMI::enterprises.26878.200.3.2.70.1.1 = INTEGER: 0
SNMPv2-SMI::enterprises.26878.200.3.2.70.1.1.0 = INTEGER: 0
SNMPv2-SMI::enterprises.26878.200.3.2.70.1.2 = INTEGER: 0
SNMPv2-SMI::enterprises.26878.200.3.2.70.1.2.0 = INTEGER: 0
SNMPv2-SMI::enterprises.26878.200.3.2.70.1.3 = INTEGER: 98
SNMPv2-SMI::enterprises.26878.200.3.2.70.1.3.0 = INTEGER: 98
SNMPv2-SMI::enterprises.26878.200.3.2.70.1.4 = INTEGER: 0
SNMPv2-SMI::enterprises.26878.200.3.2.70.1.4.0 = INTEGER: 0
SNMPv2-SMI::enterprises.26878.200.3.2.70.1.5 = INTEGER: 2
SNMPv2-SMI::enterprises.26878.200.3.2.70.1.5.0 = INTEGER: 2
SNMPv2-SMI::enterprises.26878.200.3.2.70.1.6 = INTEGER: 0
SNMPv2-SMI::enterprises.26878.200.3.2.70.1.6.0 = INTEGER: 0
SNMPv2-SMI::enterprises.26878.200.3.2.70.1.7 = INTEGER: 8060816
SNMPv2-SMI::enterprises.26878.200.3.2.70.1.7.0 = INTEGER: 8060816
SNMPv2-SMI::enterprises.26878.200.3.2.70.1.8 = INTEGER: 2676884
SNMPv2-SMI::enterprises.26878.200.3.2.70.1.8.0 = INTEGER: 2676884
```

```
SNMPv2-SMI::enterprises.26878.200.3.2.70.1.9 = INTEGER: 4063224
SNMPv2-SMI::enterprises.26878.200.3.2.70.1.9.0 = INTEGER: 4063224
SNMPv2-SMI::enterprises.26878.200.3.2.70.1.10 = INTEGER: 4063224
SNMPv2-SMI::enterprises.26878.200.3.2.70.1.10.0 = INTEGER: 4063224
```

Application KPI

Application KPI can be obtained on a per statistic basis with snmpget in a manner much like obtaining Component Statistics. As an example, to get the aggregate number of sessions currently active on qns01, use the following command:

snmpget -v 2c -c broadhop -M +BROADHOP-MIB:CISCO-QNS-MIB qns01
.1.3.6.1.4.1.26878.200.3.3.70.15.24

An example of the output from this command would be:

SNMPv2-SMI::enterprises.26878.200.3.3.70.15.24 = STRING: "0"

Interpreting this output means that 0 sessions are active on qns01.

Similarly, all available KPI in an MIB node can be "walked" via the snmpwalk command. This is very similar to snmpget as above. As an example, to see all statistics on qns01, use the following command:

snmpwalk -v 2c -c broadhop -M +BROADHOP-MIB:CISCO-QNS-MIB qns01
.1.3.6.1.4.1.26878.200.3.3.70.15

An example of the output from this command would be:

```
SNMPv2-SMI::enterprises.26878.200.3.3.70.15.20 = STRING: "0"
SNMPv2-SMI::enterprises.26878.200.3.3.70.15.20.0 = STRING: "0"
SNMPv2-SMI::enterprises.26878.200.3.3.70.15.21 = STRING: "0"
SNMPv2-SMI::enterprises.26878.200.3.3.70.15.21.0 = STRING: "0"
SNMPv2-SMI::enterprises.26878.200.3.3.70.15.22 = STRING: "0"
SNMPv2-SMI::enterprises.26878.200.3.3.70.15.22.0 = STRING: "0"
SNMPv2-SMI::enterprises.26878.200.3.3.70.15.23 = STRING: "0"
SNMPv2-SMI::enterprises.26878.200.3.3.70.15.23.0 = STRING: "0"
SNMPv2-SMI::enterprises.26878.200.3.3.70.15.24 = STRING: "0"
SNMPv2-SMI::enterprises.26878.200.3.3.70.15.24.0 = STRING: "0"
SNMPv2-SMI::enterprises.26878.200.3.3.70.15.25 = STRING: "3591813304"
SNMPv2-SMI::enterprises.26878.200.3.3.70.15.25.0 = STRING:
"3591813304"
```

Notifications

Testing and validating notifications requires slightly more skill than testing SNMP gets and walks. Recall that the overall architecture is that all components and applications in the CPS system are configured to send notifications to lb01 or lb02 via lbvip02, the Internal Network IP. These systems log the notification locally in `/var/log/snmp/trap` and then "re-throw" the notification to the

destination configured by corporate_nms_ip. Two testing and troubleshooting methods are illustrated below: confirming notifications are being sent properly from system components to lb01 or lb02, and confirming that notifications can be sent upstream to the NMS.

Receiving Notifications

There are several ways to confirm that lb01 or lb02 are properly receiving notifications from components. First, determine the active load balancer – it is either lb01 or lb02 and have multiple IP addresses per interface as shown by the `ifconfig` command.

Step 1 Log in to the active load balancer with ssh as the root user.

Note Use the ifconfig command to identify the active load balancer, either lbvip01 or lbvip02.

Step 2 Look at the trap log on the active load balancer in "follow" mode with the command tail `-f /var/log/snmp/trap`.

Step 3 Note the last trap thrown.

Step 4 In a separate window log into another "safe" system – qns04 as the root user.

Note Safe system should not be a production system or a system processing live customer traffic.

Step 5 A LowSwap warning notification can be thrown from qns04 to the active load balancer by temporarily turning off swap. This is relatively safe, but may have production impacts so care is needed. Turn off swap with the command:

```
swapoff -a
```

Step 6 Wait up to 2 minutes for the snmp daemon to detect and throw a trap.

Step 7 Note the notification arrive in the active load balancer in the "tail"-ed log. This is seen in the logs as a detailed trap message and a logger message that reads similar to: Mar 20 17:20:00 lb01 logger: Forwarded qns04 LowSwap to `<corporate_nms_ip>`.

Step 8 Turn swap back on in qns04 with the command:

```
swapon -a
```

If a LowSwap notification is received at the active load balancer, then this notification is re-thrown to the destination NMS. Check the NMS logs for this notification.

Upstream Notifications

If a notification is not received by the NMS, you can manually throw a notification from the active load balancer to the NMS using this command:

```
snmptrap -v 2c -c broadhop <corporate_nms_ip> ""
    NET-SNMP-EXAMPLES-MIB::netSnmpExampleHeartbeatNotification
    netSnmpExampleHeartbeatRate i 123456
```

where <*corporate_nms_ip*> is the appropriate NMS IP address. This sends an SNMPv2 trap from the active load balancer to the NMS and can be used for debugging.

Reference Document

For more information related to SNMP Traps and KPIs, refer to *CPS 7.0.1 Troubleshooting Guide*.

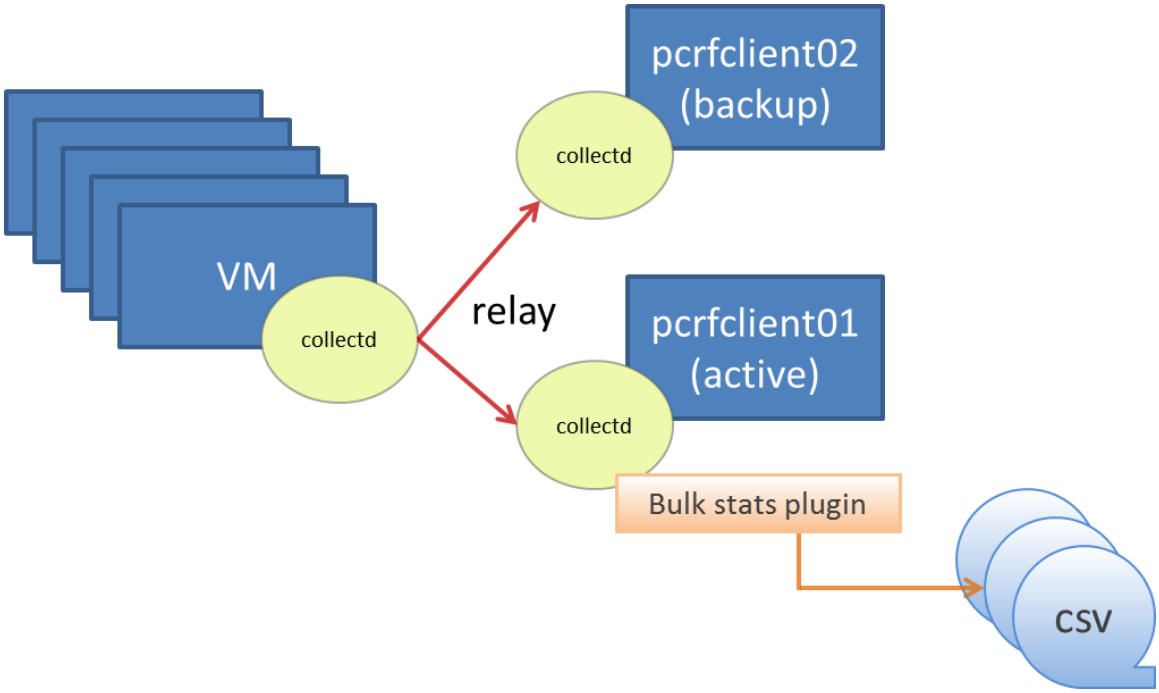
CPS Statistics

Revised: July 10, 2015

This chapter covers the following sections:

Bulk Statistics Overview

Bulk Statistics are the statistics that are gathered over a given time period and written to a set of files. These statistics can be used by external analytic processes and/or network management systems. The architecture of CPS bulk statistic collection is shown below.



The collection utility **collectd** is used for collecting and storing statistics from each VM. Detailed **collectd** documentation can be found on http://collectd.org/.

Collectd within CPS is deployed with nodes relaying data using the collectd network plug-in (https://collectd.org/wiki/index.php/Plugin:Network) to the centralized collection nodes on the pcrfclient01 and pcrfclient02 machines. The centralized collector is writes the collected data to output CSV files.

Note Each pcrfclient (01 and 02) collects bulk statistics independently. As a result, it is normal to have slight differences between the two files. For example, pcrfclient1 will generate a file at time *t* and pcrfclient02 will generate a file at time *t* +/- clock drift between the two machines.

As a best practice, always use the bulk statistics collected from pcrfclient01. Pcrfclient02 can be used as a backup in the event of failure of pcrfclient01.

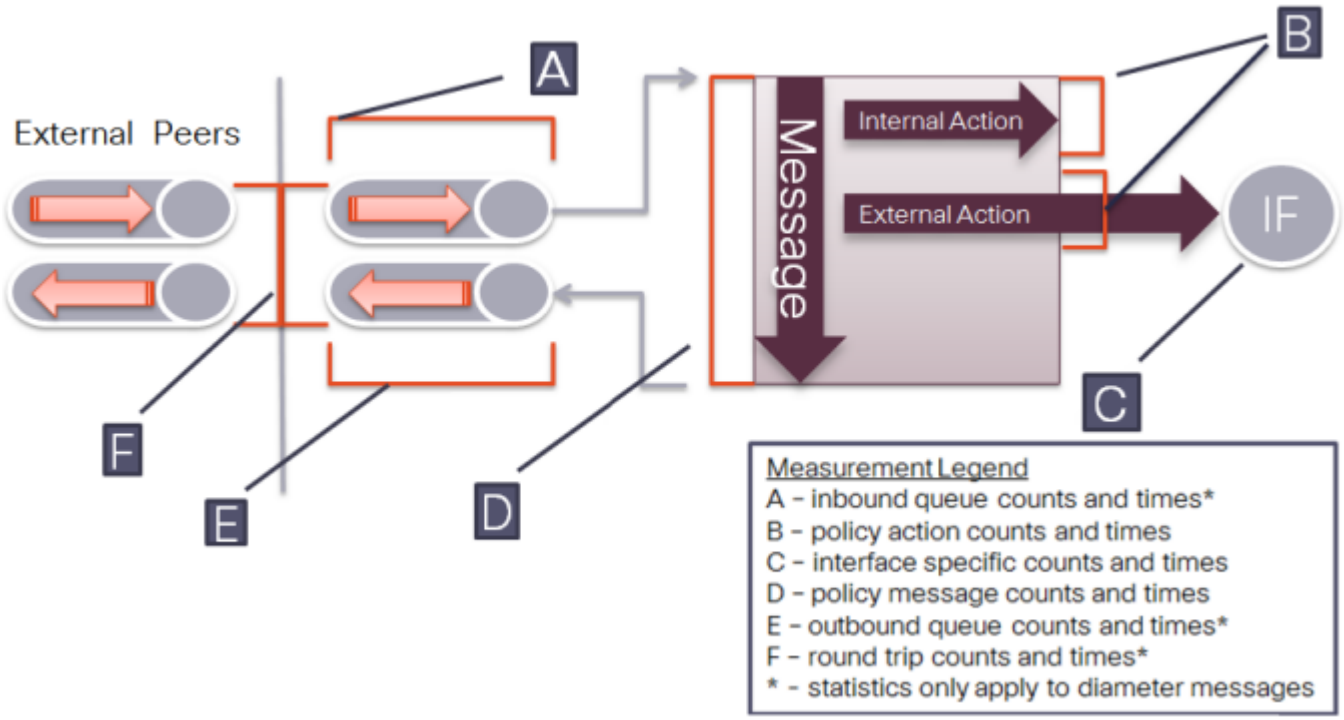
CPS Statistics

The list of statistics available in CPS is consolidated in an Excel spreadsheet. After CPS is installed, this spreadsheet can be found in the following location on the Cluster Manager VM:

/var/qps/install/current/scripts/documents/QPS_statistics.xlsx

Overview

The following diagram represents the various statistic gathering points for incoming and outgoing messages.



A brief description of each statistic gathering points is given below:

1. Upon receipt of a message on the load balancer node the message is registered as received and forwarded to a middle tier processing node.
2. This middle tier processing node tracks the inbound message counts and time spent within the inbound processing queue. If a message is discarded due to SLA violation, then counters are incremented at this point. This occurs at point A within the diagram.
3. Upon arrival within the policy engine all messages are counted and timers are started to measure the duration of processing.
4. Any internal or external actions are tracked at this point and the round trip time is measured from the policy engine invocation of the action and success or failure of the action. This occurs at point B within the diagram.
5. For external actions (e.g. LDAP), interface specific statistics maybe captured. This occurs at point C in the diagram and is gathered from the load balancer nodes.
6. Upon completion of the message in the policy engine, the total elapsed time is measured and whether success or failure occurred in processing.

Note A message is considered a success even if the policy returns an error (such as 5002). These application errors are tracked at point D within the diagram.

7. Outbound messages are tracked from the policy engine to the load balancers at point E within the diagram.
8. Upon receipt of outbound messages, the load balancers tracks either end to end completion time for inbound requests OR starts a timer and counts outbound requests. This occurs at point F within the diagram.

CPS Statistic Types

This section describes various forms of statistics generated by CPS.

Diameter Statistics

In Diameter statistics, Monitoring Areas are defined on the basis of Queues maintained in it. Also diameter statistics can be defined based on whether the statistic is related to counter or gauge.

- **Counter:** Counter type represents a non-negative integer which monotonically increases until it reaches a maximum value of 2^32-1 (4294967295 decimal), when it resets and starts increasing again from zero.

  Counters have no defined “initial” value, and thus, a single value of a Counter has (in general) no information content. You must take multiple readings to understand anything.

- **Gauge:** Gauge type represents a non-negative integer, which can increase or decrease, but can never exceed a maximum value, nor fall below a minimum value. The maximum value can not be greater than 2^32-1 (4294967295 decimal), and the minimum value can not be smaller than 0.

LDAP Statistics

CPS tracks LDAP statistics for general LDAP actions, LDAP query counters, LDAP connection counters, as well as message counters. categories: Action and Messages.

Radius Server Statistics

Radius server statistics are defined based on two categories: Action and Messages.

System Statistics

System statistics are defined based on six categories: CPU, File System Usage, Disk Performance, Interface, Load, and Memory.

Engine Statistics

Engine statistics are defined based on three categories: Session Count, Session Operation, and Internal messages.

Error Statistics Definitions

With regards to error statistic here is a definition of each error suffixes:

Error Statistics	Description
node1.messages.*.error	Failure processing a message
e2e*_qns_stat.error	Count of occurrence for given diameter result code
pe-submit-error	Error submitting to policy engine
_bypass	Message not sent to policy engine due to successful response (2001)
_drop	Message dropped due to SLA violation
rate-limit	Message dropped due to rate limiting violation

Note The diameter e2e statistics with the suffix “error” always have a value of 0 (zero) unless they have “_late” in the statistic name.

Configuring Bulk Statistics Collection

By default, CPS outputs a bulk statistics CSV file to the `/var/broadhop/stats/` directory on a 5 minute interval.

The default naming standard is:

- `stats-`*`hostname`*`.current`
- for CSV formatted file: `stats-`*`hostname`*`.YYYY-MM-DD-HH-MI.csv`

These CSV files include all statistics collected during the 5 minute interval.

Note If a statistic is generated by the system multiple times within the 5 minute interval, only the last measured statistic is collected in the CSV file.

To change the interval time (how often a CSV file is generated):

Step 1 Add the following line to `/etc/broadhop/qns.conf` file:

Dstatistics.step.interval=X

The value *X* is a multiplier of 10 seconds. For example, a value of 1 = 10 second interval; a value of 2 = 20 second interval.

Step 2 Any changes made in the qns.conf file need to be pushed to all qns nodes and then a rolling restart on all qns nodes to pick up the changes in the qns.conf file:

- Run `copyall.sh` to push the qns.conf file out to each qns node.
- Run the following command separately on each qns node, preferably during a maintenance window:

  # service qns restart

Changing the interval to a lower value allows for easier identification of peaks and valleys in response time. However, only the last the bulk statistics only reports the last statistic measured during a 5 minute period and this fact should be taken into account when interpreting the bulk statistics.

Configuring Logback.xml

Configuration of the CPS application statistics is controlled in the `/etc/collectd.d/logback.xml` file.

Refer to http://logback.qos.ch/manual/appenders.html for more information about the configuration of the logback.xml file.

This file contains the following default setup:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration scan="true" scanPeriod="5 seconds" debug="true">
```

The Sifting Appender below is the logback appender that can be used to generate a CSV file per host. Alternatively, you can collect per `metricName` or other variable.

```
<appender name="SIFT" class="ch.qos.logback.classic.sift.SiftingAppender">
    <discriminator>
      <key>host</key>
      <defaultValue>unknown</defaultValue>
    </discriminator>
  <sift>
```

The Sifting Appender is parameterized with the host variable and the BulkStatAppender is configured to write out a file every 5 minutes.

A separate appender can be selected here (for example: RollingFileAppender).

```
<appender name="STATS-${host}"
class="com.cisco.quantum.collectd.bulkstat.BulkStatAppender">
    <file>/var/broadhop/stats/stats-${host}.current</file>
    <rollingPolicy>
<fileNamePattern>/var/broadhop/stats/stats-${host}.%d{yyyy-MM-dd-HH-mm}.csv</fileNamePatte
rn>
```

Configuring Retention of CSV Files

Use the `maxHistory` parameter to configure the retention duration of the statistics on pcrfclient. In the example below, the **maxHistory** parameter is set to 20, which would retain 100 minutes (20 * 5 minutes) of statistics. Any files older than 100 minutes are deleted.

```
<maxHistory>20</maxHistory>
    </rollingPolicy>
    <encoder>
```

By default, CPS keeps all CSV files (maxHistory is NOT set). To prevent disk space issues, it is recommended to set maxHistory in order to limit the number of CSV files retained .

Disabling Collection of Specific Statistics

Specific statistics can be turned off by treating these as loggers with a level of OFF. For example:

```
<logger name="interface.lo" level="off"/>
<\!-\- Configure default Loggers \-->
<root level="INFO">
    <appender-ref ref="SIFT" />
</root>
</configuration>
```

Formatting the Statistics Output

The pattern below is used to format the output of the statistic line. The following keys should be used in setting up a statistics line:

- %X{localTime} - local date time in yyyy-MM-dd'T'HH:mm:ss.SSSZ
- %X{iso8601DateTime}- date time in ISO 8601 format
- %X{epochTime} - time in milliseconds
- %X{interval} - statistics interval
- %X{host} - host name of the stat as reported to collectd
- %X{valueType} - type of statistic:
  - C = Counter
  - G = Gauge
  - D = Derived by a calculation
  - A - Absolute
- %X{plugin} - collectd plugin
- %X{pluginInstance} - collectd plugin instance
- %X{type} - collectd type
- %X{typeInstance} - collectd type instance
- %X{dataSourceName} - collectd data source name
- %X{metricName} - dot notation metric name
- %msg - value of statistics
- %n - EOL character

For example, this line:

```
<pattern>%X{iso8601DateTime},%X{valueType},%X{host}.%X{metricName},%msg%n</pattern>
```

would generate the following sample line in the CSV file:

```
2015-01-13T07:58:52.002+0000,D,scale-qns06.disk.dm-2.disk_ops.read,3314
2015-01-13T07:58:52.002+0000,D,scale-qns06.disk.dm-2.disk_ops.write,754210
2015-01-13T07:58:52.001+0000,D,scale-qns06.disk.sda.disk_ops.read,13833
```

Restarting the Collectd Service

After making any configuration changes to logback.xml, restart the collectd service.

```
service collectd restart
```

Example CPS Statistics

This section covers the following topics:

- Sample CSV Files Names
- Sample Output

Sample CSV Files Names

Below given is a sample of the file names that are stored in the statistics directory. The list of the files can be displayed by executing the following commands:

```
[root@pcrfclient01 stats]# pwd

/var/broadhop/stats

[root@pcrfclient01 stats]# ls
```

```
stats-lb01.current
stats-lb01.2014-07-03-21-53.csv
stats-lb01.2014-07-03-21-48.csv
stats-lb01.2014-07-03-21-43.csv
stats-lb01.2014-07-03-21-38.csv
stats-lb01.2014-06-28-23-54.csv
stats-lb01.2014-06-28-23-44.csv
stats-lb01.2014-06-18-13-04.csv
stats-lb01.2014-06-18-12-59.csv
stats-lb01.2014-06-18-12-54.csv
stats-lb01.2014-06-18-12-49.csv
stats-lb02.current
stats-lb02.2014-06-18-12-49.csv
stats-qns02.current
stats-qns02.2014-07-03-21-53.csv
stats-qns02.2014-07-03-21-48.csv
stats-qns02.2014-07-03-21-43.csv
stats-qns02.2014-07-03-21-38.csv
stats-qns02.2014-06-26-23-44.csv
stats-qns02.2014-06-18-13-04.csv
stats-qns02.2014-06-18-12-59.csv
stats-qns02.2014-06-18-12-54.csv
```

```
stats-qns03.current
stats-qns03.2014-06-18-12-59.csv
stats-qns03.2014-06-18-12-54.csv
stats-qns03.2014-06-18-12-49.csv
stats-qns06.current
stats-qns06.2014-07-03-21-53.csv
stats-qns06.2014-07-03-21-48.csv
stats-qns06.2014-07-03-21-43.csv
stats-qns06.2014-07-03-21-38.csv
stats-qns06.2014-06-29-23-54.csv
stats-qns06.2014-06-27-23-49.csv
stats-qns06.2014-06-27-23-44.csv
stats-qns06.2014-06-18-13-04.csv
stats-qns06.2014-06-18-12-59.csv
stats-qns07.current
stats-qns07.2014-06-18-12-49.csv
stats-qns09.current stats-sessionmgr01.2014-07-03-21-51.csv
stats-qns10.2014-07-03-21-48.csv
stats-qns10.2014-07-03-21-43.csv
stats-qns10.2014-07-03-21-38.csv
stats-qns10.2014-06-29-23-54.csv
stats-qns10.2014-06-26-23-49.csv
stats-qns10.2014-06-26-23-44.csv
stats-qns10.2014-06-18-13-04.csv
stats-qns10.2014-06-18-12-59.csv
stats-qns10.2014-06-18-12-54.csv
stats-qns10.2014-06-18-12-49.csv
stats-sessionmgr01.current
stats-sessionmgr01.2014-07-03-21-56.csv
stats-sessionmgr01.2014-07-03-21-53.csv
stats-sessionmgr02.current
stats-sessionmgr02.2014-07-03-21-53.csv
stats-sessionmgr02.2014-07-03-21-48.csv
stats-sessionmgr02.2014-07-03-21-43.csv
stats-sessionmgr02.2014-07-03-21-38.csv
stats-sessionmgr02.2014-06-27-23-44.csv
stats-sessionmgr02.2014-06-25-10-43.csv
```

Sample Output

Using the `<pattern>%X{iso8601DateTime},%X{valueType},%X{metricName},%msg%n</pattern>` formatting statement in `/etc/collectd.d/logback.xml`, an example of the output is provided:

```
2014-06-09T14:44:31.029+0000,C,node3.messagesmessages.e2e_Mobile_cisco.com_Gx_CCR-I_2001.q
ns_stat.success,99254
2014-06-09T14:44:31.029+0000,C,node3.messages.e2e_Mobile_cisco.com_Gx_CCR-I_2001.qns_stat.
error,0
2014-06-09T14:44:31.029+0000,D,node3.messages.e2e_Mobile_cisco.com_Gx_CCR-I_2001.qns_stat.
total_time_in_ms,1407
2014-06-09T14:44:31.029+0000,G,node3.messages.e2e_Mobile_cisco.com_Gx_CCR-I_2001.qns_stat.
avg,0.0
2014-06-09T14:44:31.030+0000,C,node3.messages.e2e_cisco.com_Gx_RAR_late_5xxx.qns_stat.succ
ess,0
2014-06-09T14:44:31.030+0000,C,node3.messages.e2e_cisco.com_Gx_RAR_late_5xxx.qns_stat.erro
r,99294
2014-06-09T14:44:31.030+0000,D,node3.messages.e2e_cisco.com_Gx_RAR_late_5xxx.qns_stat.tota
l_time_in_ms,0
2014-06-09T14:44:31.030+0000,G,node3.messages.e2e_cisco.com_Gx_RAR_late_5xxx.qns_stat.avg,
0.0
```

```
2014-06-09T14:44:31.031+0000,C,node3.messages.e2e_cisco.com_Gx_CCR-I_late_2001.qns_stat.su
ccess,0
2014-06-09T14:44:31.031+0000,C,node3.messages.e2e_cisco.com_Gx_CCR-I_late_2001.qns_stat.er
ror,40
2014-06-09T14:44:31.031+0000,D,node3.messages.e2e_cisco.com_Gx_CCR-I_late_2001.qns_stat.to
tal_time_in_ms,0
2014-06-09T14:44:31.031+0000,G,node3.messages.e2e_cisco.com_Gx_CCR-I_late_2001.qns_stat.av
g,0.0
2014-06-09T14:44:31.031+0000,C,node3.messages.e2e_cisco.com_Gx_RAR_late_5002.qns_stat.succ
ess,0
2014-06-09T14:44:31.031+0000,C,node3.messages.e2e_cisco.com_Gx_RAR_late_5002.qns_stat.erro
r,99294
2014-06-09T14:44:31.031+0000,D,node3.messages.e2e_cisco.com_Gx_RAR_late_5002.qns_stat.tota
l_time_in_ms,0
2014-06-09T14:44:31.031+0000,G,node3.messages.e2e_cisco.com_Gx_RAR_late_5002.qns_stat.avg,
0.0
2014-06-09T14:44:31.036+0000,C,node4.messages.e2e_Mobile_Sy_STR_2001.qns_stat.success,9929
0
2014-06-09T14:44:31.036+0000,C,node4.messages.e2e_Mobile_Sy_STR_2001.qns_stat.error,0
2014-06-09T14:44:31.036+0000,D,node4.messages.e2e_Mobile_Sy_STR_2001.qns_stat.total_time_i
n_ms,235
2014-06-09T14:44:31.036+0000,G,node4.messages.e2e_Mobile_Sy_STR_2001.qns_stat.avg,0.0
2014-06-09T14:44:31.036+0000,C,node4.messages.e2e_Mobile_Sy_SLR_2001.qns_stat.success,9929
0
2014-06-09T14:44:31.036+0000,C,node4.messages.e2e_Mobile_Sy_SLR_2001.qns_stat.error,0
2014-06-09T14:44:31.036+0000,D,node4.messages.e2e_Mobile_Sy_SLR_2001.qns_stat.total_time_i
n_ms,182
2014-06-09T14:44:31.036+0000,G,node4.messages.e2e_Mobile_Sy_SLR_2001.qns_stat.avg,0.0
2014-06-09T14:44:41.029+0000,C,node3.messages.e2e_Mobile_cisco.com_Gx_CCR-I_2001.qns_stat.
success,99254
2014-06-09T14:44:41.029+0000,C,node3.messages.e2e_Mobile_cisco.com_Gx_CCR-I_2001.qns_stat.
error,0
2014-06-09T14:44:41.029+0000,D,node3.messages.e2e_Mobile_cisco.com_Gx_CCR-I_2001.qns_stat.
total_time_in_ms,1407
2014-06-09T14:44:41.029+0000,G,node3.messages.e2e_Mobile_cisco.com_Gx_CCR-I_2001.qns_stat.
avg,0.0
2014-06-09T14:44:41.030+0000,C,node3.messages.e2e_cisco.com_Gx_RAR_late_5xxx.qns_stat.succ
ess,0
2014-06-09T14:44:41.030+0000,C,node3.messages.e2e_cisco.com_Gx_RAR_late_5xxx.qns_stat.erro
r,99294
2014-06-09T14:44:41.030+0000,D,node3.messages.e2e_cisco.com_Gx_RAR_late_5xxx.qns_stat.tota
l_time_in_ms,0
```

Logging

Revised: July 10, 2015

This chapter covers the following sections:

CPS Logs

Most of the CPS logs are located locally in `/var/log/broadhop/` on virtual machines IOMGRxx, QNSxx, and PCRFclientXX. PCRFclient01 contains the consolidated logs from all of the IOMGR, QNS and PCRFCLIENT virtual machines.

The CPS logs can be divided based on Application/Script which produces the logs:

Application/Script Produces Logs: Upgrade Script, Upgrade Binary

- **Log:** upgrade log
- **Description:** Log messages generated during upgrade of CPS.
- **Log file name, format, path:**

  AIO: /var/log/update<date>_<time>.log

HA/GR: pcrfclient01:/var/log/broadhop/upgrade<date>_<time>.log

- **Log config File:** NA
- **Log Rollover:** No

Application/Script Produces Logs: qns

- **Log:** qns log
  - **Description:** Main and most detailed logging. Contains initialization errors and application level errors.
  - **Log file name, format, path:**

    AIO: /var/log/broadhop/qns-<instance no>.log

    HA/GR: pcrfclient01:/var/log/broadhop/qns-<instance no>.log
  - **Log config File:** /etc/broadhop/logback.xml
  - **Log Rollover:** No
- **Log:** qns engine logs
  - **Description:** Higher level event based logging including what services a subscriber has, the state of the session, and other useful information.
  - **Log file name, format, path:**

    AIO: /var/log/broadhop/qns-engine-<instance no>.log

    HA/GR: /var/log/broadhop/qns-engine-<instance no>.log
  - **Log config File:** NA
  - **Log Rollover:** No
- **Log:** qns service logs
  - **Description:** Contains start up logs. Also if the logback.xml is incorrectly formatted all logging statements go into this log.
  - **Log file name, format, path:**

    AIO: /var/log/broadhop/service-qns-<instance no>.log

    HA/GR: qns0*: /var/log/broadhop/service-qns-<instance no>.log
  - **Log config File:** NA
  - **Log Rollover:** No
- **Log:** consolidated qns logs
  - **Description:** Contains the consolidation of all qns logs with the IP of the instance as part of the log event.
  - **Log file name, format, path:**

    AIO: NA

    HA/GR: pcrfclient01: /var/broadhop/log/consolidated-qns.log
  - **Log config File:** /etc/broadhop/controlcenter/logback.xml
  - **Log Rollover:** No
- **Log:** consolidated engine logs

  - **Description:** Contains the consolidation of all qns engine logs with the IP of the instance as part of the log event.
  - **Log file name, format, path:**

    AIO: NA

    HA/GR: /etc/broadhop/controlcenter/logback.xml
  - **Log config File:** /etc/broadhop/controlcenter/logback.xml
  - **Log Rollover:** No
- **Log:** consolidated diagnostics logs
  - **Description:** Contains logs about errors occurred during diagnostics of CPS.
  - **Log file name, format, path:**

    AIO: NA

    HA/GR: /pcrfclient01: /var/log/broadhop/ consolidated-diag.log
  - **Log config File:** NA
  - **Log Rollover:** No

Application/Script Produces Logs: qns pb

- **Log:** qns pb logs
  - **Description:** Policy builder startup, initialization logs get logged into this log file.
  - **Log file name, format, path:**

    AIO: /var/log/broadhop/ qns-pb.log

    HA/GR: pcrfclient01: /var/log/broadhop/qns-pb.log
  - **Log config File:** NA
  - **Log Rollover:** No
- **Log:** service qns pb logs
  - **Description:** Policy builder service logs.
  - **Log file name, format, path:**

    AIO: /var/log/broadhop/ service-qns-pb.log

    HA/GR: pcrfclient01: /var/log/broadhop/service-qns-pb.log
  - **Log config File:** NA
  - **Log Rollover:** No
- **Log:** qns engine pb logs
  - **Description:** Policy builder engine logs.
  - **Log file name, format, path:**

    AIO: /var/log/broadhop/ qns-engine-pb.log

    HA/GR: pcrfclient01: /var/log/broadhop/qns-engine-pb.log
  - **Log config File:** NA
  - **Log Rollover:** No

Application/Script Produces Logs: mongo

- **Log:** mongo db logs
- **Description:** Contains useful information about the mongo db operations including queries, errors, warnings, and users behavior.
- **Log file name, format, path:**

  AIO: /var/log/mongodb-<port>.log

  HA/GR: sessionmgr01: /var/log/mongodb-<port>.log
- **Log config File:** /etc/mongod.conf
- **Log Rollover:** No

Application/Script Produces Logs: httpd

- **Log:** httpd access logs
  - **Description:** Apache server records all incoming requests and all requests processed to a log file.
  - **Log file name, format, path:**

    AIO: /var/log/httpd/access_log

    HA/GR: pcrfclient01: /var/log/httpd/access_log
  - **Log config File:** /etc/httpd/conf/httpd.conf
  - **Log Rollover:** Yes
- **Log:** httpd error logs
  - **Description:** All apache errors/diagnostic information about other errors found during serving requests are logged to this file. This apache log file often contain details of what went wrong and how to fix it.
  - **Log file name, format, path:**

    AIO: /var/log/httpd/error_log

    HA/GR: pcrfclient01: /var/log/httpd/error_log
  - **Log config File:** /etc/httpd/conf/httpd.conf
  - **Log Rollover:** Yes

Application/Script Produces Logs: portal

- **Log:** portal request logs
  - **Description:** Contains the API requests that are sent from the portal to the QNS system.
  - **Log file name, format, path:**

    AIO: /var/www/portal/app/tmp/logs/api_request.log

    HA/GR: portal01: /var/www/portal/app/tmp/logs/api_requests.log
  - **Log config File:** NA
  - **Log Rollover:** No
- **Log:** portal response logs
  - **Description:** Contains the API response that are sent from the qns to portal.

  - **Log file name, format, path:**

    AIO: /var/www/portal/app/tmp/logs/api_response.log

    HA/GR: portal01: /var/www/portal/app/tmp/logs/api_response.log
  - **Log config File:** NA
  - **Log Rollover:** No
- **Log:** portal error logs
  - **Description:** Contains error level logs of the application. Generally not useful to anyone except the portal developers. It generally reveals permission issues on installations.
  - **Log file name, format, path:**

    AIO: /var/www/portal/app/tmp/logs/error.log

    HA/GR: portal01: /var/www/portal/app/tmp/logs/error.log
  - **Log config File:** NA
  - **Log Rollover:** No
- **Log:** portal debug logs
  - **Description:** Contains debug level logs of the application.Generally not useful to anyone except the portal developers. It generally reveals permission issues on installations.
  - **Log file name, format, path:**

    AIO: /var/www/portal/app/tmp/logs/debug.log

    HA/GR: /var/www/portal/app/tmp/logs/debug.log
  - **Log config File:** NA
  - **Log Rollover:** No

Application/Script Produces Logs: license manager

- **Log:** lmgrd logs
- **Description:** Contains license file related errors.
- **Log file name, format, path:**

  AIO: /var/log/broadhop/lmgrd.log

  HA/GR: pcrfclient01: /var/log/broadhop/lmgrd.log
- **Log config File:** NA
- **Log Rollover:** No

Application/Script Produces Logs: svn

- **Log:** SVN log
- **Description:** svn log command displays commit log messages. For more information refer: `/usr/bin/svn log -help`. For example,./usr/bin/svn log http://lbvip02/repos/run
- **Log file name, format, path:**

  AIO: NA

  HA/GR: NA
- **Log config File:** NA

- **Log Rollover:** No

Application/Script Produces Logs: auditd

- **Log:** audit logs
- **Description:** Contains log of all sessions established with CPS VM. SSH session logs, cron job logs.
- **Log file name, format, path:**

  AIO: /var/log/audit/audit.log

  HA/GR: pcrfclient01: /var/log/audit/audit.log
- **Log config File:** NA
- **Log Rollover:** Yes

Application/Script Produces Logs: graphite

- **Log:** carbon client logs
  - **Description:** Contains client connection logs.
  - **Log file name, format, path:**

    AIO: /var/log/carbon/clients.log

    HA/GR: pcrfclient01: /var/log/carbon/clients.log
  - **Log config File:** /etc/carbon/carbon.conf
  - **Log Rollover:** No
- **Log:** carbon console logs
  - **Description:** Contains process startup and initialization logs.
  - **Log file name, format, path:**

    AIO: /var/log/carbon/console.log

    HA/GR: pcrfclient01: /var/log/carbon/console.log
  - **Log config File:** /etc/carbon/carbon.conf
  - **Log Rollover:** No
- **Log:** carbon query logs
  - **Description:** Contains log queries which are performed on the application.
  - **Log file name, format, path:**

    AIO: /var/log/carbon/query.log

    HA/GR: pcrfclient01: /var/log/carbon/query.log
  - **Log config File:** /etc/carbon/carbon.conf
  - **Log Rollover:** No
- **Log:** carbon creates logs
  - **Description:** This log tells you what .wsp (whisper) db files are being created.
  - **Log file name, format, path:**

    AIO: /var/log/carbon/creates.log

    HA/GR: pcrfclient01: /var/log/carbon/creates.log

  - **Log config File:** /etc/carbon/carbon.conf
  - **Log Rollover:** Yes
- **Log:** carbon listener logs
  - **Description:** Contains connection related logs.
  - **Log file name, format, path:**

    AIO: /var/log/carbon/listener.log

    HA/GR: pcrfclient01: /var/log/carbon/listener.log
  - **Log config File:** /etc/carbon/carbon.conf
  - **Log Rollover:** Yes

Application/Script Produces Logs: kernel

- **Log:** haproxy
- **Description:** Contains information like if VIP handoff.
- **Log file name, format, path:**

  AIO: /var/log/messages

  HA/GR: pcrfclient01: /var/log/messages
- **Log config File:** NA
- **Log Rollover:** Yes

Basic Troubleshooting Using CPS Logs

- Start the engine logs on pcrfclient01.
- This displays issues or problems in the subscriber or services. If the event is not found in the engine logs, check the qns logs to look for anomalies.
- It is good to understand when the call was supposed to occur in order to narrow down the issue.
- If there are no ERRORs or no exceptions, etc, then we can increase the logging levels. Policy tracing and logs at DEBUG levels can usually indicate the problem.
- But just like a router, too much debugging can affect the performance of the system.
- Use grep usernames, mac addresses, IP addresses, etc in logs to find required information.

Logging Level and Effective Logging Level

Logging level and the actual effective logging level can be two different levels because of the following logback logging rules:

1. When a logging level is set, if the logging level of the parent process is higher than the logging level of the child process, then the effective logging level of the child process is that of the parent process. That is, even though the logging level of the child process is set, it cannot be below the logging level of the parent process and is automatically overridden to the higher logging level of the parent process.
2. There is a global “root” logging level that each process can inherit as an effective default logging level. If you do not want to have a default effective logging level, then set the root level to OFF.

3. Each logging level prints the output of the lower logging levels.

This following table displays the logging level and the message types printed due to 3. above.

Level	Message Types Printed
All	Equivalent to Trace and some more messages.
Trace	Trace, Debug, Info, Warn, & Error
Debug	Debug, Info, Warn, & Error
Info	Info, Warn, & Error
Warn	Warn & Error
Error	Error
Off	-

The following table describes the different logging levels and what they should be used for:

Logging Level	Description	Valid Use Case	Invalid Use Case
Error	Error conditions that breaks a system feature. Errors logging level should not be used for call flow errors.	Database is not available.	Subscriber not found.
Warn	Helps to understand the early signs that will prevent the system from functioning in the near future OR are triggered by unexpected preconditions in a method.	Retrieved more then one Gx QoS profile.	Warnings should not be used for individual call flows. No service found for session.
Info	Helps to understand the life cycle of components and subsystems, such as plug-ins, databases.	NA	Info should not be used for individual call flows.
Debug	Helps to understand the flow of the code execution at Class/Method level. i.e. in _createIsgDeviceSession({log...)	NA	NA
Trace	Helps to understand the values of the statement and branch of logics within the method for troubleshooting.	NA	NA

You can configure target and log rotation for consolidated logs in log configuration file `/etc/broadhop/controlcenter/logback.xml`.

The following parameters can be configured for target VM and port.

```
<appender name="SOCKET-BASE"
class="ch.qos.logback.classic.net.SocketAppender>"
    <RemoteHost>${logging.controlcenter.host:-lbvip02}</RemoteHost>
    <Port>${logging.controlcenter.port:-5644}</Port>
    <ReconnectionDelay>10000</ReconnectionDelay>
    <IncludeCallerData>false</IncludeCallerData>
```

```
</appender>
```

The above configuration is used to redirect consolidated logs to lbvip02 VM on port 5644 with reconnection delay.

Log rotation is configured using following configuration in `/etc/broadhop/controlcenter/logback.xml`.

```
<rollingPolicy
      class="ch.qos.logback.core.rolling.FixedWindowRollingPolicy">
      <fileNamePattern>
${com.broadhop.log.dir:-/var/log/broadhop}/consolidated-diag.%i.log.g
z
      </fileNamePattern>
      <minIndex>1</minIndex>
      <maxIndex>5</maxIndex>
    </rollingPolicy>
<triggeringPolicy
      class="ch.qos.logback.core.rolling.SizeBasedTriggeringPolicy">
      <maxFileSize>100MB</maxFileSize>
    </triggeringPolicy>
```

Using the above configuration 100 MB log files are generated and after that log files rotate from index 1 to 5.

When the 100 MB log file trigger condition is met, the order in which CPS system performs the file operations is:

- log.5.gz > deleted
- log.4.gz > log.5.gz
- log.2.gz > log.3.gz
- log.1.gz > log.2.gz
- Current > log.1.gz

Similar configurations can be applied for qns logs in `/etc/broadhop/logback.xml`.

Consolidated Logging

Consolidated logging is a function of all of the CPS VMs, and sends logging to a central server on Control Center 01 to aid the debugging process. Cisco Policy Servers send all the configured log messages to the consolidated logging function.

To configure the consolidated logging function, perform these steps:

Step 1 Edit the `logback.xml` file that is present in the `/etc/broadhop` directory and the `logback.xml` file that is present in the `/etc/broadhop/controlcenter` directory.

a. Start by viewing the `/etc/broadhop/logback.xml` file. It must have a section that looks similar to this:

```
<!-- Configure Loggers -->
```

```
<!-- Hide 'Could not load class...' noise. -->
<logger
name="org.springframework.osgi.extensions.annotation.ServiceRefe
renceDependencyBeanFactoryPostProcessor" level="error" />
<logger name="org.springframework" level="warn" />
<logger name="com.broadhop.resource.impl" level="warn" />
<logger name="com.danga" level="warn" />
<logger name="httpclient.wire" level="warn" />
<logger name="org.apache.commons.httpclient" level="warn" />
<logger name="sun.rmi.tranrsport.tcp" level="warn" />
<logger name="org.apache.activemq.transport.InactivityMonitor"
level="warn" />
<!-- Configure default Loggers -->
<root level="warn">
<appender-ref ref="FILE" />
<appender-ref ref="SOCKET" />
</root>
```

The *level* can be configured to **error**, **warn**, **info**, or **debug** in the order of least logging to most logging. When debugging an issue or during initial installation. We recommend that you set the logging level to debug. To change the logging level, change one of the levels or add additional categories, for which you must contact a Cisco support representative.

b. View the `/etc/broadhop/controlcenter/logback.xml` file. It must have a section that looks similar to this:

```
<!-- Configure Remote Logger -->
<logger name="remote" level="info" additivity="false">
<appender-ref ref="CONSOLIDATED-FILE" />
<appender-ref ref="CONSOLIDATED-JMX" />
</logger>
```

Again, we recommend that you set this level to **debug** for initial installation purposes, but no other changes are necessary for this file.

After your system is up and running, it is most useful to turn the system to either **error** or **warn**. The levels debug or info usually have logs rollover very quickly. After the log rolls over, the information is lost. For this reason, **warn** or **error** generates a substantially smaller amount of logging, and gives you the ability to look for issues in the system over a longer period of time.

Rsyslog Log Processing

Rsyslog Overview

Enhanced log processing is provided in this release using Rsyslog.

Rsyslog logs Operating System (OS) data locally (`/var/log/messages` etc.) using the `/etc/rsyslog.conf` and `/etc/rsyslog.d/*conf` configuration files.

On all nodes, Rsyslog forwards the OS system log data to lbvip02 via UDP over the port defined in the `logback_syslog_daemon_port` variable as set in the CPS Deployment Template (Excel spreadsheet).

To download the most current CPS Deployment Template, refer to the *CPS Installation Guide* or *CPS Release Notes*.

Refer to http://www.rsyslog.com/doc/ for more details and Rsyslog documentation.

Rsyslog-proxy

A second instance of Rsyslog called Rsyslog-proxy is installed only on AIO and LB nodes. Rsyslog-proxy is only installed if the `syslog_managers_list` variable is set in the CPS Deployment Template.

Rsyslog-proxy is the main log forwarding process as set up by the `/etc/rsyslog-proxy.conf` file.

- It receives OS system log data from all the nodes via UDP over the PORT defined in the logback_syslog_daemon_port variable. The default port number is 6514.
- It receives all CPS log data via UDP over the PORT defined in the logback_syslog_daemon_port variable. The default port number is 6514.

  The `/etc/broadhop/controlcenter/logback.xml` file on pcrfclients or `/etc/broadhop/logback.xml` file on AIO is configured to send logs to rsyslog-proxy via UDP using the logback SyslogAppender. See Configuration of Logback.xml for more information.
- Rsyslog-proxy forwards the OS system log data and CPS log data to logstash via TCP on PORT 6513 with a UDP backup.
- Rsyslog-proxy does not log any data to local files because the rsyslog instance is already doing that.
- It receives CPS JSON formatted log data via TCP on PORT 5544. rsyslog-proxy forwards that to logstash via TCP on PORT 5543 with a UDP backup.
- It receives SNMP events via TCP on PORT 7546. rsyslog-proxy forwards that to logstash via TCP on PORT 7545 with a UDP backup.
- rsyslog-proxy sends all OS system log data and CPS log data to any number of remote servers via UDP. (Of course, the remote servers must be setup to receive traffic but that is not a part of the scope of this document.)

Configuration for HA Environments

Configuration of Rsyslog for High Availability CPS environments is performed using the CPS Deployment Template.

Refer to the following information available in the template tabs.

Configuration Variables

The following variables can now be set in the CPS Deployment Template:

- **syslog_managers_list** — space separated list of remote logging servers (tuple protocol:hostname:port). Only UDP is currently supported.
- **syslog_managers_ports** — comma separated list of the remote logging server ports (must match the ports in the syslog_managers_list).
- **logback_syslog_daemon_addr** — hostname of the internal UDP server that rsyslog-proxy runs to receive incoming logs from CPS and OS (defaults to lbvip02)
- **logback_syslog_daemon_port** — incoming port for rsyslog-proxy (defaults to 6514).

Note If the syslog_managers_list variable is empty, the rsyslog-proxy instance is not installed or configured.

Additional Hosts Tab

The following parameter can be configured in the Additional Hosts tab of the CPS Deployment Template file:

corporate_syslog_ip	syslog_manager	<IP ADDR>

Configuration Tab

The following parameters can be configured in the Configuration tab of the CPS Deployment Template file:

syslog_managers_list	udp:corporate_syslog_ip:<PORT>
syslog_managers_ports	<PORT>
logback_syslog_daemon_addr	lbvip02
logback_syslog_daemon_port	6514

- lbvip02 is the default address for logback to send data
- 6514 is the default port for logback to send data

Configuration for AIO

The Rsyslog-proxy configuration for AIO environment uses a custom “facts” file: `/etc/facter/facts.d/rsyslog.txt`

The same variables are used as in the CPS Deployment Template.

For example,

syslog_managers_list=udp:corporate_syslog_ip:514

syslog_managers_ports=514

logback_syslog_daemon_addr=lbvip02

logback_syslog_daemon_port=6514

On AIOs, you must add aliases to `/etc/hosts` for the remote servers as defined in the syslog_managers_list.

Configuration of Logback.xml

The `/etc/broadhop/controlcenter/logback.xml` file on pcrfclients or `/etc/broadhop/logback.xml` file on AIO is configured to send logs to rsyslog-proxy via UDP using the logback SyslogAppender.

Refer to http://logback.qos.ch/manual/appenders.html#SyslogAppender for the Syslog Appender documentation.

The following appender forwards all CPS logs to a remote server.

```
<appender name='SYSLOG' class='ch.qos.logback.classic.net.SyslogAppender'>
  <syslogHost>lbvip02</syslogHost><!--#SAP#-->
  <port>6514</port><!--#SAP#-->
  <suffixPattern>[qps] [%d{yyyy-mm-dd'T'HH:mm:ss.SSSZ}] %msg</suffixPattern>
```

```
  <facility>LOCAL0</facility>
</appender>
```