



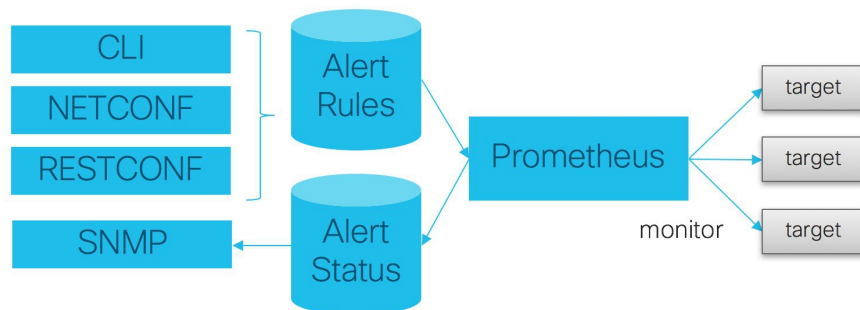
Notification and Alert

- [Architectural Overview, on page 1](#)
- [Major Components, on page 2](#)
- [Technical Architecture, on page 2](#)
- [Protocols, on page 2](#)
- [SNMP Object Identifier and Management Information Base, on page 2](#)
- [SNMP Notifications, on page 3](#)
- [Notifications and Alerting, on page 5](#)
- [Alert Rules, on page 20](#)
- [NMS Destination Configuration, on page 42](#)

Architectural Overview

A Cisco Policy Suite (CPS) vDRA deployment comprises multiple virtual machines (VMs) with multiple running containers deployed for scaling and high availability (HA) purposes. The monitoring and alerting system of the CPS vDRA deployment is centered around alert definition, metric gathering, and SNMP trap forwarding. The high-level architecture is shown below:

Figure 1: High-Level Architecture



Major Components

Alert Definition

Alert definition occurs when an end user (or external system) configures the system via CLI, NETCONF, or RESTCONF interfaces with Alert rules. The system takes these alert rules and pushes the definitions into the Prometheus processes running within the cluster. The system does not provide a fixed set of alerts but provides a sample list of common alerts an operator may want to configure.

Metric Gathering

At the core of the alerting framework, the system runs multiple Prometheus processes (<http://prometheus.io>) which monitors the system and track metrics which can be used for triggering alerts. The default Prometheus instance that monitors the system tracks metrics at a 5 second interval for 24 hours.

SNMP Trap Forwarding

Once an alert is triggered the Prometheus server forwards that alert to the active control/Cluster Manager node. These alerts are forwarded based on configuration to external NMS systems using either SNMPv2 or SNMPv3.

Technical Architecture

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

Protocols

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

Alerts are triggered from either the Cluster Manager or Control virtual machines if the Cluster Manager is not active.

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 the 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:

Table 1: MIBs

| MIB Filename | Purpose |
|-------------------------------|--|
| BROADHOP-MIB.mib | Defines the main structure include structures and codes. |
| BORADHOP-NOTIFICATION-MIB.mib | Defines Notifications/Traps available. |

SNMP Notifications

SNMP Notifications in the form of traps (one-way) are provided by the infrastructure. CPS notifications do not require acknowledgments. The traps provide both:

- Proactive alerts that the predetermined thresholds have been passed. For example, a disk is nearing capacity or CPU load is too high.
- Reactive alerting when system components fail or are in a degraded state. For example, a process died or network connectivity outage has occurred.

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 markers.

- Facility
- Severity
- Source (device name)
- Device time

These objects can be used to identify where the issue lies and 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.

Table 2: Syslog Facility

| Number | Facility | Description |
|--------|----------------|--|
| 0 | Hardware | Physical Hardware - Servers SAN NIC Switch and so on |
| 1 | Networking | Connectivity in the OSI (TCP/IP) model |
| 2 | Virtualization | VMware ESXi (or other) virtualization |

| Number | Facility | Description |
|--------|------------------|--|
| 3 | Operating System | Linux OS |
| 4 | Application | Application (CPS Session Manager, CPS Binding Database, and so on) |
| 5 | Process | Specific process |

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:

Table 3: Severity Levels

| 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 message. |
| 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 functionality 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 Facility and Severity categorizations:

Table 4: Severity Categorization

| Facility.Severity | Categorization | Possibility |
|------------------------|--|--|
| Process.Emergency | A single part of an application has failed. | Possible but in an HA configuration very unlikely. |
| Hardware.Debug | A hardware component has sent a NA debug message. | NA |
| 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). |

It is not possible to quantify every Facility and Severity combination. This is primarily driven by the fact that the alert rules can be configured to meet each operator's environment. 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 important! As a general principle, alerts should only be defined with an Emergency-severity trap if the system becomes inaccessible or unusable in some way. An unusable system is rare but might occur if multiple failures occur in the operating system virtualization networking or hardware facilities.

Notifications and Alerting

The CPS Monitoring and Alert Notification framework provides the following SNMP 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 (for example, an application process failed).

Component Notifications

Components are devices that make up the CPS system. These are systems level traps. They are generated when some predefined thresholds is crossed and are defined in the alerting configuration of the system. User can modify and change these using the alert definition commands.

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 Notification being thrown (broadhopComponentNotificationName)
- 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 notification (broadhopNotificationSeverity)
- Additional information about the notification, which might be a bit of log or other information.

The following table provides the list of supported alarms:

Table 5: Component Notifications

| Notification Name | Severity | Message Text | Description |
|-------------------|----------|--|---|
| DISK_FULL | Critical | Disk filesystem / usage is more than the 90% | Disk usage is monitored. |
| | Clear | Disk filesystem / usage is greater than 10% | |
| HIGH_LOAD | Major | load average value for 5 min is greater than 3 current value is {{ \$value }} | Load on the CPU is measured as per the linux operating system load. |
| | Clear | load average value for 5 min is lower than 3 | |
| LINK_STATE | Critical | {{ \$labels.interface }} is down on {{ \$labels.instance }} | Indicates if any interface (ens**) has gone down. |
| | Clear | {{ \$labels.interface }} is up on {{ \$labels.instance }} | |

| Notification Name | Severity | Message Text | Description |
|-------------------|---------------------------|---|--|
| LOW_MEMORY | Critical | Available RAM is less than 20% current value is {{ \$value }} | Monitors memory usage on the VMs. When free memory goes down, the threshold alarm is raised. |
| | Clear | Available RAM is more than 20% | |
| PROCESS_STATE | Critical | {{ \$labels.service_name }} instance {{ \$labels.module_instance }} of module {{ \$labels.module }} is in Aborted state. | Monitors process restarts. |
| | Clear | {{ \$labels.service_name }} instance {{ \$labels.module_instance }} of module {{ \$labels.module }} is moved from Aborted state | |
| HIGH_CPU_USAGE | Critical, Major, or Minor | CPU usage in last 10 sec is more than 90% current value {{ \$value }} | Monitors CPU usage. |
| | Clear | CPU usage in last 10 sec is lower than 90% | |
| QNS_JAVA_STARTED | Error | {{ \$labels.service_name }} instance {{ \$labels.module_instance }} of module {{ \$labels.module }} is in Started state. | Indicates Java process restart. |
| | Clear | {{ \$labels.service_name }} instance {{ \$labels.module_instance }} of module {{ \$labels.module }} is moved from started state | |

| Notification Name | Severity | Message Text | Description |
|-----------------------|----------|--|--|
| IP_NOT_REACHABLE | Critical | VM/VIP IP {{ \$labels.instance }} is not reachable | When IP is not reachable, this alarm is raised. |
| | Clear | VM/VIP IP {{ \$labels.instance }} is reachable | For more information, see IP Not Reachable, on page 10 |
| DIAMETER_PEER_DOWN | Error | Diameter peer is down. | Any peer connected to PAS is monitored. |
| | Clear | Diameter peer is up | |
| DRA_PROCESS_UNHEALTHY | Critical | {{ \$labels.service_name }} instance {{ \$labels.module_instance }} of module {{ \$labels.module }} is not healthy | Process state is monitored. |
| | Clear | {{ \$labels.service_name }} instance {{ \$labels.module_instance }} of module {{ \$labels.module }} is healthy | |
| DB_SHARD_DOWN | Critical | All DB Members of a replica set {{ \$labels.shard_name }} are down | Alarm raised when both primary and secondary replica set members are down. |
| | Clear | All DB Members of a replica set {{ \$labels.shard_name }} are not down | |

| Notification Name | Severity | Message Text | Description | |
|--------------------|---|---|---|---|
| NO_PRIMARY_DB | Critical | Primary {{ \$labels.cluster }} {{ \$labels.shard }} not running | Alarm raised when the primary database is not up. | Alarm raised when secondary database is not up. |
| | SECONDARY_DB_DOWN | Critical | Secondary Member {{ \$labels.name }} of replica set {{ \$labels.shard_name }} is down | |
| Clear | | Secondary Member {{ \$labels.name }} of replica set {{ \$labels.shard_name }} is up | | |
| LOW_SWAP | Critical | {{ \$labels.instance }} has less than 80% swap memory . | Monitors the swap memory. | |
| | Clear | {{ \$labels.instance }} has greater than 80% swap memory . | | |
| DOCKER_ENGINE_DOWN | Critical | Docker Engine {{ \$labels.engine_id }} is down. | Monitors the docker engine status/state. | |
| | Clear | Docker Engine {{ \$labels.engine_id }} is up. | | |
| SVN_BACKUP_ALERT | Alert | svn backup in mongo is out of sync, please check svn_audit.log | Alarm raised when SVN repos are not in sync with SVN backup stored in mongo-admin containers. | |
| | Clear | svn backup in mongo is in sync now | | |
| Clear | Primary running on seed {{ \$labels.seed }} | Alarm cleared when the primary is running on server seed. | | |



Note By default, no alert rules are configured in the system.

IP Not Reachable

Two things impact the generation of an IP_NOT_REACHABLE alert if a VIP fails over.

1. VIP switchover time
2. Prometheus polling interval

VIP switchover time can vary depending on the load of the VM and traffic on the network. Metrics are polled every 5 seconds. If a VIP fails over quickly, then an IP_NOT_REACHABLE alert is not generated.

Example: IP_NOT_REACHABLE alert not generated

1. T0 Prometheus polls the Orchestrator for the probe_icmp_target metric which is set to 1 (ip reachable).
2. T1 VIP fails
3. T2
4. T3
5. T4 VIP moves to the backup VM
6. T5 Prometheus polls the Orchestrator for the probe_icmp_target metric which is set to 1 (ip reachable)

Example: IP_NOT_REACHABLE alert generated

1. T0 Prometheus polls the Orchestrator for the probe_icmp_target metric which is set to 1 (ip reachable).
2. T1
3. T2
4. T3
5. T4 VIP fails
6. T5 Prometheus polls the Orchestrator for the probe_icmp_target metric which is set to 0 (ip not reachable)
7. T6 IP_NOT_REACHABLE alert is generated.

Application Notifications

The following table describes the application notifications:

Table 6: Application Notifications

| Notification Name | Severity | Message Text | Description |
|---|----------|--|---|
| DRA_MESSAGE_PROCESSING_FAILURE_TPS_EXCEEDED | Critical | Message Processing Failure TPS exceeded, current value is {{ \$value }}. | TPS of rejected messages from DRA Director (Any messages with Result code !=2001) |
| | Clear | Message Processing Failure TPS in control. | |

| Notification Name | Severity | Message Text | Description |
|---|----------|--|--|
| DRA_DIRECTOR_TPS_EXCEEDED | Critical | {{ \$labels.instance }} Director TPS exceeded, current value is {{ \$value }}. | Success TPS of Total DRA Director (ResultCode=2001) |
| | Clear | {{ \$labels.instance }} Director TPS in control . | |
| DRA_WORKER_TPS_EXCEEDED | Critical | {{ \$labels.instance }} Worker TPS exceeded, current value is {{ \$value }}. | TPS of Total Worker |
| | Clear | {{ \$labels.instance }} Worker TPS in control. | |
| DRA_DB_TPS_EXCEEDED | Critical | {{ \$labels.instance }} Persistence DB TPS exceeded , current value is {{ \$value }}. | TPS of DB TPS (Query and Update) |
| | Clear | {{ \$labels.instance }} Persistence DB TPS in control. | |
| DIAMETER_UNABLE_TO_DELIVER_TPS_EXCEEDED | Critical | UNABLE_TO_DELIVER TPS exceeded, current value is {{ \$value }}. | TPS of Diameter 3002 |
| | Clear | UNABLE_TO_DELIVER in control. | |
| DIAMETER_TRANSIENT_FAILURE_TPS_EXCEEDED | Critical | TRANSIENT_FAILURE TPS exceeded, current value is {{ \$value }}. | TPS of Diameter 4xxx |
| | Clear | TRANSIENT_FAILURE in control. | |
| DIAMETER_UNKNOWN_SESSIONS_TPS_EXCEEDED | Critical | UNKNOWN_SESSIONS TPS exceeded, current value is {{ \$value }}. | TPS of Diameter 5002 |
| | Clear | UNKNOWN_SESSIONS in control. | |
| MISMATCH_REQUEST_RESPONSE | Critical | {{ \$labels.remote_peer }} MISMATCH_REQUEST_RESPONSE exceeded, current value is {{ \$value }}. | Mismatch in Rate of Request and Response (Discrepancy in ingress and egress) |
| | Clear | {{ \$labels.remote_peer }} MISMATCH_REQUEST_RESPONSE in control. | |

| Notification Name | Severity | Message Text | Description |
|--|----------|---|--|
| KEEP_ALIVE_RAR _ROUTING_FAILURE_ TPS_EXCEEDED | Critical | Keep Alive RAR TPS exceeded, current value is {{ \$value }}. | TPS of Keep Alive RAR Routing (Stale RAR) |
| | Clear | Keep Alive RAR TPS in control. | |
| EGRESS_RATE_ LIMITED_SESSION_ ERR_RESP_TPS_ EXCEEDED | Critical | {{ \$labels.local_peer }} {{ \$labels.remote_peer }} Egress rate limited messages with error response TPS exceeded, current value is {{ \$value }}. | TPS of Rate Limited Response for Error |
| | Clear | {{ \$labels.local_peer }} {{ \$labels.remote_peer }} Egress rate limited messages with error response TPS in control. | |
| EGRESS_RATE_ LIMITED_SESSION_ REJECT_TPS_ EXCEEDED | Critical | {{ \$labels.local_peer }} {{ \$labels.remote_peer }} Egress rate limited messages dropped without error TPS exceeded, current value is {{ \$value }}. | TPS of Rate Limited Response Rejected |
| | Clear | {{ \$labels.local_peer }} {{ \$labels.remote_peer }} Egress rate limited messages dropped without error TPS in control. | |
| INGRESS_RATE_ LIMITED_SESSION_ ERR_RESP_TPS_ EXCEEDED | Critical | {{ \$labels.local_peer }} {{ \$labels.remote_peer }} Ingress rate limited messages with error response TPS exceeded, current value is {{ \$value }}. | TPS of Rate Limited Response Error - Ingress |
| | Clear | {{ \$labels.local_peer }} {{ \$labels.remote_peer }} Ingress rate limited messages with error response TPS in control. | |
| INGRESS_RATE_ LIMITED_SESSION_ REJECT_TPS_ EXCEEDED | Critical | {{ \$labels.local_peer }} {{ \$labels.remote_peer }} Ingress rate limited messages dropped without error response TPS exceeded, current value is {{ \$value }}. | TPS of Rate Limited Response Rejected - Ingress |
| | Clear | {{ \$labels.local_peer }} {{ \$labels.remote_peer }} Ingress rate limited messages dropped without error response TPS in control. | |

| Notification Name | Severity | Message Text | Description |
|---|----------|--|---|
| BINDING_STORAGE_ERRORS_TPS_EXCEEDED | Critical | Binding Store Error TPS exceeded, current value is {{ \$value }}. | TPS Binding Storage Errors (Binding storage failed because of high load/any other database error) |
| | Clear | Binding Store Error TPS in control. | |
| BINDING_LOOKUP_ERROR_TPS_EXCEEDED | Critical | Binding Lookup Error TPS exceeded, current value is {{ \$value }}. | TPS Binding Lookup Errors (Binding retrieval failure because of internal error) |
| | Clear | Binding Lookup Error TPS in control. | |
| DB_ERR_TPS_EXCEEDED | Critical | All DB Errors TPS exceeded, current value is {{ \$value }}. | TPS All database errors |
| | Clear | All DB Errors TPS in control. | |
| DB_RESPONSE_TIME_EXCEEDED | Critical | {{ \$labels.instance }} DB Response Time exceeded, current value is {{ \$value }}. | Response Time Exceeds (Database Query/Update operation time exceeds) |
| | Clear | {{ \$labels.instance }} DB Response Time in control, current value is {{ \$value }}. | |
| BINDING_KEY_NOT_FOUND_IN_AAR_TPS_EXCEEDED | Critical | {{ labels.origin_host }} Binding Key not found in AAR TPS exceeded, current value is {{ \$value }}. | TPS Binding Key Not Found in AAR (When AAR received with no "imsi+apn/msisdn/ipv6") |
| | Clear | {{ labels.origin_host }} Binding Key not found in AAR TPS in control. | |
| BINDING_KEY_NOT_FOUND_IN_CCR_I_TPS_EXCEEDED | Critical | {{ labels.origin_host }} Binding Key not found in CCR(I) TPS exceeded, current value is {{ \$value }}. | TPS Binding Key Not Found in CCR-I(When CCR-I received with no "imsi+apn/msisdn/ipv6") |
| | Clear | {{ labels.origin_host }} Binding Key not found in CCR(I) TPS in control. | |
| BINDING_NOT_FOUND_TPS_EXCEEDED | Critical | {{ labels.origin_host }} Binding not found TPS exceeded, current value is {{ \$value }}. | TPS Binding Not Found |
| | Clear | {{ labels.origin_host }} Binding not found TPS in control. | |
| BINDING_DB_INCONSISTENT_TPS_EXCEEDED | Critical | TPS AAR with Result Code 5065 exceeded, current value is {{ \$value }}. | TPS AAR with Result Code 5065 |
| | Clear | TPS AAR with Result Code 5065 in control. | |

| Notification Name | Severity | Message Text | Description |
|--|----------|--|---|
| BINDING_SESSION _DB_SIZE_ EXCEEDED | Critical | {{ \$labels.db }} size exceeded, current value is {{ \$value }}. | Total Size of Session DB Exceeded |
| | Clear | {{ \$labels.db }} size in control. | |
| BINDING_IMSI_ APN_DB_SIZE _EXCEEDED | Critical | {{ \$labels.db }} size exceeded, current value is {{ \$value }}. | Total Size of IMSI / APN DB Exceeded |
| | Clear | {{ \$labels.db }} size in control. | |
| BINDING_MSISDN _APN_DB_SIZE _EXCEEDED | Critical | {{ \$labels.db }} size exceeded, current value is {{ \$value }}. | Total Size of MSISDN / APN DB Exceeded |
| | Clear | {{ \$labels.db }} size in control | |
| BINDING_IPV6 _DB_SIZE_ EXCEEDED | Critical | {{ \$labels.db }} size exceeded, current value is {{ \$value }}. | Total Size of IPv6 DB Exceeded |
| | Clear | {{ \$labels.db }} size in control | |
| PEER_TPS _EXCEEDED | Critical | {{ \$labels.instance }} Peer Connection {{ \$labels.local_peer }} {{ \$labels.remote_peer }} TPS exceeded, current value is {{ \$value }}. | Peer TPS Exceeded (Per peer TPS thresholds) |
| | Clear | {{ \$labels.instance }} Peer Connection {{ \$labels.local_peer }} {{ \$labels.remote_peer }} TPS in control. | |
| NO_RESPONSE_ PEER_FOR_ ANSWER_TPS _EXCEEDED | Critical | {{ \$labels.instance }} No Response From Peer Connection TPS exceeded for {{ \$labels.message_type }} , current value is {{ \$value }}. | TPS No Response From Peer (timeouts from PCRF/any peer) |
| | Clear | {{ \$labels.instance }} No Response From Peer Connection TPS in control for {{ \$labels.message_type }} . | |
| PEER_RESPONSE _TIME_EXCEEDED | Critical | message_duration_seconds {type=~"peer_*"} [labels: type] | Peer Response Time Exceeded (Response time of peer exceeds) |
| | Clear | Response time in control. | |
| NO_PEER_GROUP _MEMBER _AVAILABLE | Critical | {{ \$labels.peer_group }} not available. | Peer Group is not Available (All peers in peer_group down) |
| | Clear | {{ \$labels.peer_group }} available. | |

| Notification Name | Severity | Message Text | Description |
|---|----------|---|---|
| PCRF_NOT_CREATING_SESSIONS_TPS_EXCEEDED | Critical | Failed CCR-I TPS exceeded, current value is {{ \$value }}. | TPS Rate of Failed CCR-I(ResultCode !=2001) |
| | Clear | Failed CCR-I TPS in control. | |
| FORWARDING_LOOP_FOUND_TPS_EXCEEDED | Critical | {{ \$labels.remote_peer }} Loop Detected TPS exceeded , current value is {{ \$value }}. | TPS Rate of Diameter Message Loop |
| | Clear | {{ \$labels.remote_peer }} Loop Detected TPS in control. | |
| RELAY_LINK_TPS_GT_0 | Critical | {{ \$labels.remote_peer }} Relay Started, current value is {{ \$value }}. | TPS Rate of Relay Peer > 0 (When relay peers start exchanging control plane messages) |
| | Clear | {{ \$labels.remote_peer }} Relay Stated. | |
| RELAY_LINK_TPS_EXCEEDED | Critical | {{ \$labels.remote_peer }} Relay Link TPS exceeded, current value is {{ \$value }}. | TPS Rate of Relay Peer (TPS of relay messages) |
| | Clear | {{ \$labels.remote_peer }} Relay Link TPS in control. | |
| RELAY_LINK_STATUS | Critical | {{ \$labels.remote_peer }} Relay Link is Down. | Relay Link is Down (Relay link status is monitored) |
| | Clear | {{ \$labels.remote_peer }} Relay Link is UP. | |
| NO_RELAY_PEER_TPS_EXCEEDED | Critical | {{ \$labels.remote_peer }} Relay Peer TPS exceeded, current value is {{ \$value }}. | TPS Rate of Relay Peer Failure |
| | Clear | {{ \$labels.remote_peer }} Relay Peer TPS in control. | |
| SESSION_DB_LIMIT_EXCEEDED | Alert | Session max DB limit reached | This alarm is generated when session database count crosses maximum limit configured using CLI for db-max-record-limit. |
| | Clear | Session max DB limit reached alarm cleared | This alarm is cleared when session database count drops below maximum limit configured using CLI for db-max-record-limit. |

| Notification Name | Severity | Message Text | Description |
|-----------------------------|----------|--|---|
| IPV6_DB_LIMIT_EXCEEDED | Alert | IPv6 max DB limit reached | This alarm is generated when IPv6 database count crosses maximum limit configured using CLI for db-max-record-limit. |
| | Clear | IPv6 max DB limit reached alarm cleared | This alarm is cleared when IPv6 database count drops below maximum limit configured using CLI for db-max-record-limit. |
| IPV4_DB_LIMIT_EXCEEDED | Alert | IPv4 max DB limit reached | This alarm is generated when IPv4 database count crosses maximum limit configured using CLI for db-max-record-limit. |
| | Clear | IPv4 max DB limit reached alarm cleared | This alarm is cleared when IPv4 database count drops below maximum limit configured using CLI for db-max-record-limit. |
| IMSIAPN_DB_LIMIT_EXCEEDED | Alert | ImsiApn max DB limit reached | This alarm is generated when ImsiApn database count crosses maximum limit configured using CLI for db-max-record-limit. |
| | Clear | ImsiApn max DB limit reached alarm cleared | This alarm is cleared when ImsiApn database count drops below maximum limit configured using CLI for db-max-record-limit. |
| MSISDNAPN_DB_LIMIT_EXCEEDED | Alert | MsisdnApn max DB limit reached | This alarm is generated when MsisdnApn database count crosses maximum limit configured using CLI for db-max-record-limit. |
| | Clear | MsisdnApn max DB limit reached alarm cleared | This alarm is cleared when MsisdnApn database count drops below maximum limit configured using CLI for db-max-record-limit. |

| Notification Name | Severity | Message Text | Description |
|--|----------|--|--|
| CRD_CACHE_LOAD_ERROR | Critical | Error when loading CRD cache | This alarm is generated when CRD is not loaded properly or CRD is loaded with an error value as "1". |
| | Clear | CRD cache loaded successfully | This alarm is cleared when CRD cache is updated properly with value as "0". |
| APP_SERVICE_HEALTH_STATUS_CRD* | Critical | {{ \$labels.service }} service is Unhealthy! | This alarm is generated when CRD service is unhealthy if value is "1" |
| | Clear | {{ \$labels.service }} service is Healthy. | This alarm is generated when CRD service is healthy if value is "0" |
| APP_SERVICE_HEALTH_STATUS_METADATA_DB* | Critical | {{ \$labels.service }} service is Unhealthy! | This alarm is generated when the Metadata DB service is unhealthy if value is "1" |
| | Clear | {{ \$labels.service }} service is Healthy. | This alarm is generated when the Metadata DB service is healthy if value is "0" |
| VIP_NOT_ACTIVE_ON_PREFERRED* | Critical | VIP {{ \$labels.vip }} active on {{ \$labels.currentHost }} and not active on preferred {{ \$labels.preferredHost }} | This alarm is generated when the VIP is not present in preferred director or distributor. |
| | Clear | VIP {{ \$labels.vip }} active on preferred {{ \$labels.preferredHost }} | This alarm is generated when the VIP is present in preferred director or distributor. |
| PEER_DYNAMIC_RATE_LIMIT_THROTTLING* | Critical | Dynamic Rate limit is active | This alarm is generated when any one peer connected to a director is in throttling mode. sum(peer_dynamic_rate_limit_throttling) != 0 |
| | Clear | Dynamic Rate limit is not active | This alarm is generated when no peer connected to a Director is in throttling mode. sum(peer_dynamic_rate_limit_throttling) == 0 |

| Notification Name | Severity | Message Text | Description |
|-------------------------------|----------|--|---|
| NO_DB_CPU_THRESHOLD_STATUS* | Critical | {{ \$labels.instance }} is not receiving any threshold message | Director is not receiving any threshold status messages from Worker. sum(rate(processed_db_cpu_control_message_total [30s])) == 0 |
| | Clear | {{ \$labels.instance }} is receiving throttling messages | Director is receiving threshold status messages from Worker. sum(rate(processed_db_cpu_control_message_total [30s])) != 0 |
| QNS_LOGGING_STOPPED* | Critical | Application logging has stopped on {{ \$labels.hostname }} at {{ \$labels.last_updated_time }} with connections closed {{ \$labels.tcp_closed }} | This alarm is generated when application has stopped logging consolidated-qns logs unexpectedly. Note If there is no activity on the system, and the alert is raised it is expected. It is resolved automatically when application activity has started. |
| | Clear | Application logging is successful on {{ \$labels.hostname }} at {{ \$labels.last_updated_time }} | This alarm is generated when application is successful logging consolidated-qns logs. |
| DRA_PCRF_QUERY_NODE_INACTIVE* | Critical | {{ \$labels.url_endpoint }} is Inactive! | This alarm is generated when PCRF REST endpoint URL heartbeat message fails if value is "1". |
| | Clear | {{ \$labels.url_endpoint }} is Active | This alarm is generated when PCRF REST endpoint URL heartbeat message is success if value is "0". |
| DRA_PCRF_QUERY_TPS_EXCEEDED* | Critical | {{ \$labels.instance }} Perf Session Query TPS exceeded, current value is {{ \$value }} | This alarm is generated when PCRF REST API TPS exceeds if the value is greater than "5". |
| | Clear | {{ \$labels.instance }} Perf Session Query TPS in control | This alarm is generated when PCRF REST API TPS is under control if the value is less than "5". |

| Notification Name | Severity | Message Text | Description |
|--|----------|---|--|
| RELAY_TRAFFIC_THRESHOLD_EXCEEDED* | Critical | Relay traffic exceeded the threshold of 20%. Current value is {{ \$value }}% | This alarm is generated if relay traffic exceeds certain % of total traffic. |
| | Clear | Relay traffic % is under control | This alarm is generated if relay traffic is under certain % of total traffic. |
| LOCAL_PUBLISH_STOPPED* | Critical | Local publish stopped for {{ \$labels.instance }} | This alarm is generated if topology is incomplete and global end point is missing. |
| | Clear | Local publish started for {{ \$labels.instance }} | This alarm is generated if topology is complete and global end point exists. |
| GLOBAL_PUBLISH_STOPPED* | Critical | Global publish stopped for {{ \$labels.instance }} | This alarm is generated if topology is incomplete and local end point is missing. |
| | Clear | Global publish started for {{ \$labels.instance }} | This alarm is generated if topology is complete and local end point exists. |
| DIAMETER_ENDPOINTS_MISSING_LOST_REDIS* | Critical | Diameter Endpoints missing due to Redis connection lost | This alarms is generated if Diameter endpoint is missing REDIS configuration. |
| | Clear | Redis connection restored. Diameter Endpoints are restored | This alarms is generated if REDIS configuration exists in Diameter endpoint. |
| DIAMETER_PEER_EXPIRATIONS_EXCEEDED* | Critical | {{ \$labels.origin_host }} got EXPIRED in {{ \$labels.system }} | This alarm is generated if any peer has expired. |
| | Clear | Peer expiration got reset for {{ \$labels.origin_host }} | This alarm is generated if the peer expiration is reset. |
| ELASTICSEARCH_NOT_REACHABLE | Critical | Elasticsearch server is unreachable with status {{ \$labels.reachable_status }} with tcp connection status {{ \$labels.tcp_connected }} | This alarm is generated when elasticsearch is not reachable to DRA or the TCP connections are not healthy. |
| | Clear | Elasticsearch server is reachable now !!! | This alarms is generated when the elasticsearch is reachable to DRA or the TCP connections are healthy. |



Note This alarm has not been validated for all customer deployment scenarios. Please contact your Sales Account team for support.

Alert Rules

Alert Rules Configuration

The following commands are used to configure alert rules:

```
scheduler#config
```

```
scheduler(config)# alert rule <rule_name>
```

where, *<rule_name>* is the name of the alert rule. For example, test

```
Value for 'expression' (<string>): <expression based on the stats>
```

where, *<expression based on the stats>* is the expression. For example, test>1

```
Value for 'message' (<string>): <message string to be sent in the alarm message>
```

where, *<message string to be sent in the alarm message>* is the message to be sent in the alarm. For example, testing

```
Value for 'snmp-clear-message' (<string>): <message string for clear alarm>
```

where, *<message string for clear alarm>* is the string for the clear message. For example. test clear

```
scheduler(config-rule-test)#
```

```
scheduler(config-rule-test)# snmp-facility
```

Possible completions:

```
application hardware networking os proc virtualization
```

```
scheduler(config-rule-test)# snmp-facility <SNMP facility to be provided for this alert>
```

where, *<SNMP facility to be provided for this alert>* is the facility to be provided for this alert. For example, application

```
scheduler(config-rule-test)# event-host-label <provide the node details>
```

where, *<provide the node details>* is used to provide node details. For example, instance

```
scheduler(config-rule-test)# snmp-severity
```

Possible completions:

```
alert critical debug emergency error info none notice warning
```

```
scheduler(config-rule-test)# snmp-severity <SNMP severity to be send for this alert>
```

where, *<SNMP severity to be send for this alert>* is the severity level to be used for alert rule. For example, critical

```
scheduler(config-rule-test)# duration <time>
```

where, *<time>* causes Prometheus to wait for a certain duration between first encountering a new expression output vector element (like, an instance with a high HTTP error rate) and counting an alert as firing for this element. Elements that are active, but not firing yet, are in pending state.

```
scheduler(config-rule-test)# commit
Commit complete.
scheduler(config-rule-test)# end
```

Sample Configuration

The alert rules configuration is for reference only. Here is the configuration with sample values:

You can configure your alert rules based on your requirements.

```
scheduler#config
scheduler(config)# alert rule test
Value for 'expression' (<string>): test>1
Value for 'message' (<string>): testing
Value for 'snmp-clear-message' (<string>): test clear
scheduler(config-rule-test)#
scheduler(config-rule-test)# snmp-facility
Possible completions:
  application hardware networking os proc virtualization
scheduler(config-rule-test)# snmp-facility application
scheduler(config-rule-test)# event-host-label instance
scheduler(config-rule-test)# snmp-severity
Possible completions:
  alert critical debug emergency error info none notice warning
scheduler(config-rule-test)# snmp-severity critical
scheduler(config-rule-test)# duration 30s
scheduler(config-rule-test)# commit
Commit complete.
scheduler(config-rule-test)# end
```

To display all the configured alert rules use the following command:

```
scheduler# show running-config alert | tab
```

| NAME | EXPRESSION | DURATION | EVENT HOST LABEL | MESSAGE | SNMP FACILITY | SNMP SEVERITY | SNMP CLEAR MESSAGE |
|------|------------|----------|------------------------|---------|------------------|------------------|-----------------------|
| test | test > 1 | - | instance | testing | application | critical | testing clear |

Configure Different Thresholds

You can configure thresholds parameter with threshold input as comma-separated fields at the time of alarm severity setup.



Note The **threshold** parameter is optional. This is because not all alerts have different thresholds.

Configure the following two type of alert expression:

- Ascending threshold [HIGH_CPU_USAGE]
- Descending threshold [LOW_MEMORY]

Ascending Alert:

Create an alert with different thresholds using the following example:

```
alert rule HIGH_CPU_USAGE
expression "rate(node_cpu_seconds_total{mode=\"system\"} [10s])*100 > threshold"
```

```

event-host-label    instance
message             "CPU usage in last 10 sec is {{ $value }}!"
threshold           10,20,30
snmp-clear-message  "CLEAR HIGH CPU value {{ $value }}!"
!

```

For Alerts with “threshold” configured, below SNMP severity is configured by default and are not configurable.

Table 7: SNMP Severity

| Alert Severity | SNMP Severity |
|----------------|---------------|
| Critical | Emergency |
| Major | Error |
| Minor | Warn |



Note There should be no space between > and threshold keyword. For example, use the exact phrase as >threshold. Also, threshold defined should be in ascending order (10,20,30). Where, 10 corresponds to minor, 20 major, and more than 30 critical.

Descending Alert:

Create an alert with different thresholds using the following example:

```

expression          "node_memory_MemAvailable_bytes / node_memory_MemTotal_bytes * 100<threshold"

duration            20s
event-host-label    instance
message             "ALERT HIGH Memory utilization value {{ $value }}!"
threshold           50,40,30
snmp-clear-message  "\"CLEAR HIGH Memory utilization value {{ $value }}\""

```

For Alerts with “threshold” configured, below SNMP severity is configured by default and are not configurable.

Table 8: SNMP Severity

| Alert Severity | SNMP Severity |
|----------------|---------------|
| Critical | Emergency |
| Major | Error |
| Minor | Warn |



Note There should be no space between > and threshold keyword. For example, use the exact phrase as >threshold. Also, threshold that is defined should be in ascending order (50,40,30). Where, 50 corresponds to minor, 40 major, and 30 critical.



Note When an alert severity changes then the previous alert is cleared and a new alert is raised with updated severity. CLI will always display the latest alert. Alert configuration is range bound and ensures that there is only one threshold value qualifying condition. Alarm is raised once criteria is met and resolved or cleared when criteria is no longer valid.

Troubleshooting: SNMP traps can be monitored to track the alert transition.

Raising and Clearing Alert Mechanism

When Alert with no threshold is configured:

- When an alert is raised, the alert status is shown as *firing* in the **show alert status**.
- SEVERITY parameter is marked as *Not Applicable*.
- SNMP trap is sent with the configured SNMP severity.
- When the alert is cleared, the alert status is shown as *resolved* in the **show alert status**.
- SNMP trap is sent with status as *cleared*.

When Alert with threshold is configured:

- A minor alert is raised and cleared.
- When an alert is raised, the alert status is shown as *firing* in the **show alert status**.
- SEVERITY parameter will be marked as *minor*.
- SNMP trap is sent with status as *warn*.
- When the alert is cleared, the alert status is shown as *resolved* in the **show alert status**.
- SNMP trap is sent with status as *cleared*.

When Alert with threshold is configured, alarms are raised and cleared based on threshold level. Alarm gets cleared if value of expression is out of configured range. And alarm is raised if it comes within the range.

- If alerts with threshold are configured, a minor alert is raised and transitioned to **Major**.
- When an alert is raised:
 - Alert status is shown as *firing* in the **show alert status**
 - SEVERITY parameter is marked as *minor*.
 - SNMP trap is sent with status as *warn*.
- When an alert is transitioned to Major:
 - Existing SNMP trap *warn* is cleared.
 - A new alert is raised as *firing* in the **show alert status**.
 - SEVERITY parameter is marked as *major*.
 - New SNMP trap is sent with severity as *error*.

- When an alert is transitioned back to Minor:
 - A new alert is raised as *firing* in the **show alert status**.
 - Existing SNMP trap *error* is cleared.
 - SNMP trap is sent with severity as *warn*.
- When the alert is cleared:
 - The alert status is shown as *resolved* in the **show alert status**.
 - SNMP trap is sent with status as *cleared*.

Configuration Restrictions

- >threshold and <threshold are keywords.
- There should be no space between {>,<} and threshold.
- When above keywords are given in expression its mandatory to provide threshold values
- Threshold values should be provided in ascending or descending as per keyword used.
- Threshold value should be provided as comma separated string with three values.

Sample Alert Rules

You can configure alert rules based on your requirements. For sample configuration, refer to Sample Alert Rule Configuration.



Note *event-host-label* value is used as a key in the alarm map. So, configure the correct value based on your requirements while configuring alert rules.



Note Grafana can be used to see all the statistics generated by the system and based on these statistics alerting rules can be configured.



Note Alert SNMP command includes an optional parameter named *add-vm-info* that you can use to specify whether or not the VM name is prepended in the SNMP alarm in *broadhopComponentName*. For example, *broadhopComponentName: VMName/containerName*. By default, the parameter is set to true. If set to false, *broadhopComponentName* does not prepend VM name. For example, *broadhopComponentName: containerName*. The following table includes sample alert rules when *add-vm-info* is set to false. For more information about this parameter and the command, see the *vDRA Operations Guide*.

Table 9: Sample Alert Rules

| Alarm Name | Configuration |
|----------------|--|
| DiskFull | <p>broadhopComponentName: Linux host name</p> <p>broadhopComponentNotificationName: DISK_FULL</p> <p>broadhopNotificationFacility: hardware</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: Disk Filesystem/usage is more than 90%</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: Disk filesystem/usage is greater than 10%</p> <p>Expression: (round((node_filesystem_size_bytes{job='node_exporter'}-node_filesystem_avail_bytes{job='node_exporter'})/node_filesystem_size_bytes{job='node_exporter'}*100)) >= 70</p> |
| HighLoad | <p>broadhopComponentName: Linux host name</p> <p>broadhopComponentNotificationName: HIGH_LOAD</p> <p>broadhopNotificationFacility: hardware</p> <p>Alert broadhopNotificationSeverity: major</p> <p>Alert broadhopComponentAdditionalInfo: load average value for 5 minutes is greater than 3 current value is {{ \$value }}</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: load average value for 5 minutes is lower than 3</p> <p>Expression: node_load5 > 3</p> |
| LowMemoryAlert | <p>broadhopComponentName: Linux host name</p> <p>broadhopComponentNotificationName: LOW_MEMORY</p> <p>broadhopNotificationFacility: hardware</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: Available RAM is less than 20% current value is {{ \$value }}</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: Available RAM is more than 20%</p> <p>Expression: round((node_memory_MemAvailable_bytes/node_memory_MemTotal_bytes)*100) < 20</p> |

| Alarm Name | Configuration |
|----------------------|--|
| High CPU Usage Alert | <p>broadhopComponentName: Linux host name</p> <p>broadhopComponentNotificationName: HIGH_CPU_USAGE</p> <p>broadhopNotificationFacility: hardware</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: CPU usage in last 10 sec is more than 30% current value {{ \$value }}</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: CPU usage in last 10 sec is lower than 30%</p> <p>Expression: rate(node_cpu_seconds_total{mode!="system"}) [10s])*100 > 40</p> |
| Link down Alert | <p>broadhopComponentName: Linux host name</p> <p>broadhopComponentNotificationName: LINK_STATE</p> <p>broadhopNotificationFacility: networking</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: {{ \$labels.interface }} is down on {{ \$labels.instance }}</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: {{ \$labels.interface }} is up on {{ \$labels.instance }}</p> <p>Expression: link_state == 0</p> |
| Process down Alert | <p>Container Name: Linux host name</p> <p>broadhopComponentNotificationName: PROCESS_STATE</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: {{ \$labels.service_name }} instance {{ \$labels.module_instance }} of module {{ \$labels.module }} is in Aborted state.</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: {{ \$labels.service_name }} instance {{ \$labels.module_instance }} of module {{ \$labels.module }} is moved from Aborted state</p> <p>Expression: docker_service_up==1 or docker_service_up==3</p> |

| Alarm Name | Configuration |
|----------------------------------|---|
| VM/Node Down Alert | <p>broadhopComponentName: IP Address</p> <p>broadhopComponentNotificationName: IP_NOT_REACHABLE</p> <p>broadhopNotificationFacility: networking</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: VM/VIP IP {{ \$labels.instance }} is not reachable</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: VM/VIP IP {{ \$labels.instance }} is reachable</p> <p>Expression: probe_icmp_target==0</p> |
| DiameterPeer Status | <p>broadhopComponentName: Peer FQDN</p> <p>broadhopComponentNotificationName: DIAMETER_PEER_DOWN</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: error</p> <p>Alert broadhopComponentAdditionalInfo: Diameter peer is down</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: Diameter peer is up.</p> |
| DRA Process Down (healthy) Alert | <p>broadhopComponentName: Container Name</p> <p>broadhopComponentNotificationName: DRA_PROCESS_UNHEALTHY</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: {{ \$labels.service_name }} instance {{ \$labels.module_instance }} of module {{ \$labels.module }} is not healthy</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: {{ \$labels.service_name }} instance {{ \$labels.module_instance }} of module {{ \$labels.module }} is healthy</p> <p>Expression: docker_service_up==4</p> |

| Alarm Name | Configuration |
|---|---|
| All DB Member of Replica Set Down Alert | <p>broadhopComponentName: Shard Name</p> <p>broadhopComponentNotificationName: DB_SHARD_DOWN</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: All DB Members of replica set {{ \$labels.shard_name }} are down</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: Some DB Members of replica set {{ \$labels.shard_name }} are up</p> <p>Expression: absent(mongoddb_mongod_replset_member_state{shard_name="shard-1"})==1</p> |
| No primary DB Member found Alert | <p>broadhopComponentName: Shard Name</p> <p>broadhopComponentNotificationName: NO_PRIMARY_DB</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: Primary DB member not found for replica set {{ \$labels.shard_name }}</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: Primary DB member found for replica set {{ \$labels.shard_name }}</p> <p>Expression: absent(mongoddb_mongod_replset_member_health{shard_name="shard-1",state="PRIMARY"})==1</p> |
| Secondary DB Member Down Alert | <p>broadhopComponentName: Shard Name</p> <p>broadhopComponentNotificationName: SECONDARY_DB_DOWN</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: Secondary Member {{ \$labels.name }} of replica set {{ \$labels.shard_name }} is down</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: Secondary Member {{ \$labels.name }} of replica set {{ \$labels.shard_name }} is down</p> <p>Expression: (mongoddb_mongod_replset_member_state != 2) and ((mongoddb_mongod_replset_member_state==8) or (mongoddb_mongod_replset_member_state==6))</p> |

| Alarm Name | Configuration |
|---|--|
| DRA message processing failure TPS exceeded | <p>broadhopComponentName: System</p> <p>broadhopComponentNotificationName: DRA_MESSAGE_PROCESSING_FAILURE_TPS_EXCEEDED</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: Message Processing Failure TPS exceeded.</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo Message Processing Failure TPS in control.</p> <p>Expression: rate(rejected_messages_total[5m]) > 5</p> |
| Keepalive RAR routing failure - TPS exceeded | <p>broadhopComponentName: System</p> <p>broadhopComponentNotificationName: KEEP_ALIVE_RAR_ROUTING_FAILURE_TPS_EXCEEDED</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: Keep Alive RAR TPS exceeded.</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: Keep Alive RAR TPS in control.</p> <p>Expression: rate(keep_alive_rar_failure[5m]) > 5</p> |
| Egress rate limited session error response TPS exceeded | <p>broadhopComponentName: Peer FQDN</p> <p>broadhopComponentNotificationName: EGRESS_RATE_LIMITED_SESSION_ERR_RESP_TPS_EXCEEDED</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: Egress rate limited messages with error response TPS exceeded.</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: Egress rate limited messages with error response TPS in control.</p> <p>Expression: rate(diameter_peer_egress_rate_limited_with_err_response[5m]) > 5</p> |

| Alarm Name | Configuration |
|--|---|
| Egress rate limited session reject TPS exceeded | <p>broadhopComponentName: Peer FQDN</p> <p>broadhopComponentNotificationName: EGRESS_RATE_LIMITED_SESSION_REJECT_TPS_EXCEEDED</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: Egress rate limited messages dropped without error TPS exceeded.</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: Egress rate limited messages dropped without error TPS in control.</p> <p>Expression: rate(diameter_peer_egress_rate_limited_without_err_response[5m]) > 5</p> |
| Ingress rate limited session error response TPS exceeded | <p>broadhopComponentName: Peer FQDN</p> <p>broadhopComponentNotificationName: INGRESS_RATE_LIMITED_SESSION_ERR_RESP_TPS_EXCEEDED</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: Ingress rate limited messages with error response TPS exceeded.</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: Ingress rate limited messages with error response TPS in control.</p> <p>Expression: rate(diameter_peer_ingress_rate_limited_with_err_response[5m]) > 5</p> |
| Ingress rate limited session reject TPS exceeded | <p>broadhopComponentName: Peer FQDN</p> <p>broadhopComponentNotificationName: INGRESS_RATE_LIMITED_SESSION_REJECT_TPS_EXCEEDED</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: Ingress rate limited messages dropped without error response TPS exceeded.</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: Ingress rate limited messages dropped without error response TPS in control.</p> <p>Expression: rate(diameter_peer_ingress_rate_limited_without_err_response[5m]) > 5</p> |

| Alarm Name | Configuration |
|---|---|
| Binding key not found in AAR TPS exceeded | <p>broadhopComponentName: System</p> <p>broadhopComponentNotificationName: BINDING_KEY_NOT_FOUND_IN_AAR_TPS_EXCEEDED</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: Binding Key not found in AAR TPS exceeded.</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: Binding Key not found in AAR TPS in control.</p> <p>Expression: rate(aar_bind_key_not_found_total[5m]) > 5</p> |
| Binding key not found in CCR-I TPS exceeded | <p>broadhopComponentName: System</p> <p>broadhopComponentNotificationName: BINDING_KEY_NOT_FOUND_IN_CCR_I_TPS_EXCEEDED</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: Binding Key not found in CCR(I) TPS exceeded.</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: Binding Key not found in CCR(I) TPS in control.</p> <p>Expression: rate(ccri_bind_key_not_found_total[5m]) > 5</p> |
| Peer response time exceeded | <p>broadhopComponentName: Peer FQDN</p> <p>broadhopComponentNotificationName: PEER_RESPONSE_TIME_EXCEEDED</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: Peer response time exceeded.</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: Peer response time in control.</p> <p>Expression: rate(message_duration_seconds{type=~\"peer_.*\"}[5m]) > 5</p> |

| Alarm Name | Configuration |
|---------------------------------------|--|
| No peer group member available | <p>broadhopComponentName: Container Name</p> <p>broadhopComponentNotificationName: NO_PEER_GROUP_MEMBER_AVAILABLE</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: Peer group not available.</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: Peer group available.</p> <p>Expression: no_active_peer_in_peer_group == 1</p> |
| Forwarding loop found TPS exceeded | <p>broadhopComponentName: System</p> <p>broadhopComponentNotificationName: FORWARDING_LOOP_FOUND_TPS_EXCEEDED</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: Loop Detected TPS exceeded.</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: Loop Detected TPS in control.</p> <p>Expression: rate(diameter_loop_detected [5m]) > 5</p> |
| No relay peer TPS exceeded | <p>broadhopComponentName: Container Name</p> <p>broadhopComponentNotificationName: NO_RELAY_PEER_TPS_EXCEEDED</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: Relay Peer TPS exceeded.</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: Relay Peer TPS in control.</p> <p>Expression: rate(relay_send_nopeer[5m]) > 5</p> |

| Alarm Name | Configuration |
|--------------------------------|---|
| Relay link status | broadhopComponentName: Peer FQDN broadhopComponentNotificationName: RELAY_LINK_STATUS broadhopNotificationFacility: application Alert broadhopNotificationSeverity: critical Alert broadhopComponentAdditionalInfo: Relay Link is down. Clear broadhopNotificationSeverity: clear Clear broadhopComponentAdditionalInfo: Relay Link is up Expression: relay_peer_status == 0 |
| Binding not found TPS exceeded | broadhopComponentName: System broadhopComponentNotificationName: BINDING_NOT_FOUND_TPS_EXCEEDED broadhopNotificationFacility: application Alert broadhopNotificationSeverity: critical Alert broadhopComponentAdditionalInfo: Binding not found TPS exceeded. Clear broadhopNotificationSeverity: clear Clear broadhopComponentAdditionalInfo: Binding not found TPS in control Expression: rate(binding_not_found_total[5m]) > 5 |
| Relay link TPS GT 0 | broadhopComponentName: Peer FQDN broadhopComponentNotificationName: RELAY_LINK_TPS_GT_0 broadhopNotificationFacility: application Alert broadhopNotificationSeverity: critical Alert broadhopComponentAdditionalInfo: Relay started. Clear broadhopNotificationSeverity: clear Clear broadhopComponentAdditionalInfo: Relay not started. Expression: rate(relay_peer_messages_total[5m]) > 0 |
| Relay link TPS exceeded | broadhopComponentName: Peer FQDN broadhopComponentNotificationName: RELAY_LINK_TPS_EXCEEDED broadhopNotificationFacility: application Alert broadhopNotificationSeverity: critical Alert broadhopComponentAdditionalInfo: Relay Link TPS exceeded. Clear broadhopNotificationSeverity: clear Clear broadhopComponentAdditionalInfo: Relay Link TPS in control. Expression: rate(relay_peer_messages_total[5m]) > 5 |

| Alarm Name | Configuration |
|--------------------------------|---|
| SVN_BACKUP_ALERT | <p>broadhopComponentName: Linux host name</p> <p>broadhopComponentNotificationName: SVN_BACKUP_ALERT</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: warning</p> <p>Alert broadhopComponentAdditionalInfo: svn backup in mongo is out of sync</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: svn backup in mongo is in sync now</p> <p>Expression: svn_alert==1</p> |
| CRD_CACHE_LOAD_ERROR | <p>broadhopComponentName: Container Name</p> <p>broadhopComponentNotificationName: CRD_CACHE_LOAD_ERROR</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: CRD cache not loaded / loaded with error</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: CRD cache loaded successfully</p> <p>Expression: crd_cache_load_error==1</p> |
| APP_SERVICE_HEALTH_STATUS_CRD* | <p>broadhopComponentName: Container Name</p> <p>broadhopComponentNotificationName: APP_SERVICE_HEALTH_STATUS_CRD</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: {{\$labels.service}} service is Unhealthy!</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: {{\$labels.service}} service is Healthy</p> <p>Expression: app_service_health_status{service="CRD"}==1</p> |

| Alarm Name | Configuration |
|--|--|
| APP_SERVICE_HEALTH_STATUS_METADATA_DB* | <p>broadhopComponentName: Container Name</p> <p>broadhopComponentNotificationName: APP_SERVICE_HEALTH_STATUS_METADATA_DB</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: {{ \$labels.service }} service is Unhealthy!</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: {{ \$labels.service }} service is Healthy</p> <p>Expression: app_service_health_status{service="METADATA_DB"}==1</p> |
| VIP_NOT_ACTIVE_ON_PREFERRED* | <p>broadhopComponentName: Container Name</p> <p>broadhopComponentNotificationName: VIP_NOT_ACTIVE_ON_PREFERRED</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: VIP {{ \$labels.vip }} active on {{ \$labels.currentHost }} and not active on preferred {{ \$labels.preferredHost }}</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: VIP {{ \$labels.vip }} active on preferred {{ \$labels.preferredHost }}</p> <p>Expression: vip_not_active_on_preferred==1</p> |
| DYNAMIC_PEER_THROTTLING* | <p>broadhopComponentName: Container Name</p> <p>broadhopComponentNotificationName: PEER_DYNAMIC_RATE_LIMIT_THROTTLING</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: Dynamic Rate limit is active</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: Dynamic Rate limit is not active</p> <p>Expression: sum(peer_dynamic_rate_limit_throttling) != 0</p> |

| Alarm Name | Configuration |
|-------------------------------|--|
| NO_DB_CPU_THRESHOLD_STATUS* | <p>broadhopComponentName: Container Name</p> <p>broadhopComponentNotificationName: NO_DB_CPU_THRESHOLD_STATUS</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: {{ \$labels.instance }} is not receiving any threshold message</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: {{ \$labels.instance }} is receiving throttling messages</p> <p>Expression: sum(rate(processed_db_cpu_control_message_total [30s])) == 0</p> |
| QNS_LOGGING_STOPPED* | <p>broadhopComponentName: System</p> <p>broadhopComponentNotificationName: QNS_LOGGING_STOPPED</p> <p>broadhopNotificationFacility: application</p> <p>broadhopNotificationSeverity: critical</p> <p>AlertbroadhopComponentAdditionalInfo: Application logging has stopped on {{ \$labels.hostname }} at {{ \$labels.last_updated_time }} with connections closed {{ \$labels.tcp_closed }}</p> <p>ClearbroadhopNotificationSeverity: clear</p> <p>ClearbroadhopComponentAdditionalInfo: Application logging is successful on {{ \$labels.hostname }} at {{ \$labels.last_updated_time }}</p> <p>Expression: qns_logging_alert==1</p> |
| DRA_PCRF_QUERY_NODE_INACTIVE* | <p>broadhopComponentName: Pcrf Rest Endpoint Url</p> <p>broadhopComponentNotificationName: DRA_PCRF_QUERY_NODE_INACTIVE</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: {{ \$labels.url_endpoint }} is Inactive!</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: {{ \$labels.url_endpoint }} is Active.</p> <p>Expression: (sum(rate(pcrf_http_hb_send{status="fail"}[5m])) by (url_endpoint)) > 0</p> |

| Alarm Name | Configuration |
|---|---|
| DRA_PCRF_ QUERY_TPS_ EXCEEDED* | <p>broadhopComponentName: System</p> <p>broadhopComponentNotificationName: DRA_PCRF_QUERY_TPS_EXCEEDED</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: {{ \$labels.instance }} Perf Session Query TPS exceeded, current value is {{ \$value }}.</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: {{ \$labels.instance }} Perf Session Query TPS in control.</p> <p>Expression: rate(pcrf_binding_query_total{status="success"}[5m]) > 5</p> |
| RELAY_TRAFFIC_ THRESHOLD_ EXCEEDED* | <p>broadhopComponentName: instance</p> <p>broadhopComponentNotificationName: RELAY_TRAFFIC_THRESHOLD_EXCEEDED</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: Relay traffic exceeded the threshold of 20%. Current value is {{ \$value }}%"</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: {{ \$labels.instance }} "Relay traffic % is under control"</p> <p>Expression: round(sum(irate(relay_message_total{direction="egress",message_type="request"}[5m])) / sum(irate(diameter_request_total[5m])) * 100) > 10</p> |
| LOCAL_ PUBLISH_STOPPED* | <p>broadhopComponentName: instance</p> <p>broadhopComponentNotificationName: LOCAL_PUBLISH_STOPPED</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: Local publish stopped for {{ \$labels.instance }}</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: Local publish started for {{ \$labels.instance }}</p> <p>Expression: (sum(peer_connection_status) by (instance) != 0) and (sum(irate(local_control_messages_published_total{message_type="DataPathMessage",system="system-1"}[5m])) by (instance) == 0)</p> |

| Alarm Name | Configuration |
|--|---|
| GLOBAL_PUBLISH_STOPPED* | <p>broadhopComponentName: instance</p> <p>broadhopComponentNotificationName: GLOBAL_PUBLISH_STOPPED</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: Global publish stopped for {{ \$labels.instance }}</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: Global publish started for {{ \$labels.instance }}</p> <p>Expression: (sum(peer_connection_status) by (instance) != 0) and (sum(irate(global_control_messages_published_total[5m])) by (instance) == 0)</p> |
| DIAMETER_ENDPOINTS_MISSING_LOST_REDIS* | <p>broadhopComponentName: system</p> <p>broadhopComponentNotificationName: DIAMETER_ENDPOINTS_MISSING_LOST_REDIS</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: Diameter Endpoints missing due to Redis connection lost</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: Redis connection restored. Diameter Endpoints are restored</p> <p>Expression: (sum(irate(topology_update_msg_received_total[30s])) == 0)</p> |
| DIAMETER_PEER_EXPIRATIONS_EXCEEDED* | <p>broadhopComponentName: PEER FQDN</p> <p>broadhopComponentNotificationName: DIAMETER_PEER_EXPIRATIONS_EXCEEDED</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: {{ \$labels.origin_host }} got EXPIRED in {{ \$labels.system }}</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: Peer expiration got reset for {{ \$labels.origin_host }}</p> <p>Expression: sum(irate(topology_peer_expirations_total[5m])) by (system,origin_host) > 0</p> |

| Alarm Name | Configuration |
|------------------------------|---|
| ELASTICSEARCH_NOT_REACHABLE | <p>broadhopComponentName: System</p> <p>broadhopComponentNotificationName: ELASTICSEARCH_NOT_REACHABLE</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: Elasticsearch server is unreachable with status {{ \$labels.reachable_status }} with tcp connection status {{ \$labels.tcp_connected }}</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: Elasticsearch server is reachable now !!!</p> <p>Expression: elasticsearch_server_status==1</p> |
| PEER_LIMIT_FOR_SITE_EXCEEDED | <p>broadhopComponentName: Linux host name</p> <p>broadhopComponentNotificationName: PEER_LIMIT_FOR_SITE_EXCEEDED</p> <p>broadhopNotificationFacility: application</p> <p>Alert broadhopNotificationSeverity: critical</p> <p>Alert broadhopComponentAdditionalInfo: Active peer count exceeds the threshold value</p> <p>Clear broadhopNotificationSeverity: clear</p> <p>Clear broadhopComponentAdditionalInfo: Active peer count is less than threshold value</p> <p>Expression: ((sum(avg(active_peer_count) by (app_id, system_id))) > 30000)</p> <p>Note The maximum value supported for peer-connection-count is 32000. You can configure the threshold value for alert in your environment.</p> |
| NTP_SERVER_NOT_REACHABLE | <p>Message: NTP servers are not reachable</p> <p>snmp-clear-message: NTP servers are reachable</p> <p>Expression: ntp_server_status > threshold</p> <p>Note Configure the threshold value based on the requirement. For example, Threshold: 1, 2, 3</p> |
| SNMP_SERVER_NOT_REACHABLE | <p>Message: SNMP servers are not reachable</p> <p>snmp-clear-message: SNMP servers are reachable</p> <p>Expression: snmp_server_status > threshold</p> <p>Note Configure the threshold value based on the requirement. For example, Threshold: 1, 2, 3</p> |

| Alarm Name | Configuration |
|---------------------------|---|
| HCHCLOCKSKIEWIORNIPCLIENT | <p>Without Severity</p> <p>event-host-label : instance</p> <p>Message: NTP client is having higher clock skew with NTP server [{\$labels.ntp_server_ip}] with the value {{\$value}}</p> <p>snmp-clear-message: NTP client clock skew is normal now</p> <p>Expression: ntp_client_clock_skew > 10</p> |



Note This alert rule has not been validated for all customer deployment scenarios. Please contact your Sales Account team for support.

Health Status of Service

On getting the Qns Java Process State alert, the user has to access the system and check the diagnostics logs of the service to get the exact issue with the service. To access the system and check the diagnostics log, run the following command:

```
show system diagnostics | include <service_name>
```

For example:

```
scheduler# show system diagnostics | include diameter-endpoint-s1
system diagnostics diameter-endpoint-s1 serfHealth 1
system diagnostics diameter-endpoint-s1 service:cisco-policy-app 1
system diagnostics diameter-endpoint-s1 service:cisco-policy-app 2
system diagnostics diameter-endpoint-s1 service:cisco-policy-app 3
system diagnostics diameter-endpoint-s1 service:cisco-policy-app 4
  message "CLEARED: InterfaceID=diameter-endpoint-s1.weave.local;msg=\"Memcached server is operational\""
system diagnostics diameter-endpoint-s1 service:cisco-policy-app 5
  message "CLEARED: InterfaceID=com.broadhop.server:diameter-endpoint-s1.weave.local;msg=\" before Feature com.broadhop.server is Running\""
system diagnostics diameter-endpoint-s1 service:cisco-policy-app 6
  message "CLEARED:
InterfaceID=com.broadhop.dra.service:diameter-endpoint-s1.weave.local;msg=\" before Feature
com.broadhop.dra.service is Running\""
system diagnostics diameter-endpoint-s1 service:cisco-policy-app 7
  message "CLEARED:
InterfaceID=com.broadhop.common.service:diameter-endpoint-s1.weave.local;msg=\" before
Feature com.broadhop.common.service is Running\""
system diagnostics diameter-endpoint-s1 service:cisco-policy-app 8
  message "CLEARED:
InterfaceID=com.broadhop.resourcemonitor:diameter-endpoint-s1.weave.local;msg=\" before
Feature com.broadhop.resourcemonitor is Running\""
system diagnostics diameter-endpoint-s1 service:cisco-policy-app 9
  message "CLEARED:
InterfaceID=com.broadhop.microservices.control:diameter-endpoint-s1.weave.local;msg=\"
before Feature com.broadhop.microservices.control is Running\""
system diagnostics diameter-endpoint-s1 service:cisco-policy-app 10
  message "CLEARED:
InterfaceID=com.broadhop.custrefdata.service:diameter-endpoint-s1.weave.local;msg=\" before
Feature com.broadhop.custrefdata.service is Running\""
system diagnostics diameter-endpoint-s1 service:cisco-policy-app 11
```



```
system diagnostics diameter-endpoint-s1 service:cisco-policy-jmx 1
scheduler#
```

Delete Alert Rules

The following section describes the procedure to delete an alert rule and are for reference only:

```
scheduler# config
Entering configuration mode terminal
scheduler(config)# no alert rule node_down
scheduler(config)# commit
Commit complete.
scheduler(config)# end
scheduler#
```

Alert Status

Use the following command to display the current alerts status:

```
show alert status
```

For example:

```
scheduler# show alert status
```

| NAME | EVENT HOST UPDATE TIME | STATUS | MESSAGE |
|-------------------------------|-------------------------------|----------|--|
| high_cpu_alert | system | firing | CPU usage is more than 30% current_value |
| is 37.05555555555597 | 2017-05-22T10:59:37.945+00:00 | | |
| high_cpu_alert_1 | control-0 | resolved | CPU usage is more than 30% current_value |
| is 33.62500000000637 | 2017-05-22T17:17:38.184+00:00 | | |
| high_cpu_alert_1 | control-1 | resolved | CPU usage is more than 30% current_value |
| is 35.666666666667076 | 2017-05-22T11:29:37.899+00:00 | | |
| high_cpu_usage_alert | localhost:9090 | resolved | CPU Usage for last 1 min is more than |
| configured threshold | 2017-05-22T09:55:37.902+00:00 | | |
| 2017-05-22T15:39:37.811+00:00 | | | |

```
scheduler#
```

Database Alert Expression

IMSI_MSISDN Cluster

Alert Threshold for IMSI/MSISDN:

- Capacity per Primary Shard = $145000/48 = 3020$ TPS
- Alert Threshold per Shard Primary (85%) = 2500 TPS

alert rule DRA_IMSI_MSISDN_DB_TPS_EXCEEDED

expression

```
"sum(rate(mongo_operation_total {state='primary',type='mongo',op=~'update|query|delete',cluster='IMSI_MSISDN'}[5m]))
> (2500 * sum (mongo_node_state_primary {cluster='IMSI_MSISDN',type='mongo'}))"
```

event-host-label instance

message "{{ \$labels.instance }}" Persistence DB TPS exceeded , current value is {{ \$value }} !"

snmp-severity critical

snmp-clear-message "{{ \$labels.instance }}" Persistence DB TPS in control, current value is {{ \$value }} !"

Session_IPv6 Cluster

Alert Threshold for Session:

- Capacity per Primary Shard = $180000/48 = 3750$ TPS
- Alert Threshold per Shard Primary (85%) = 3200 TPS

alert rule DRA_SESS_IPV6_DB_TPS_EXCEEDED

expression

```
"sum(rate(mongo_operation_total{state='primary',type='mongo',op=~'update|query|delete',cluster=~'SES_IPV6_.*'}[5m]))
> (3200 * sum(mongo_node_state_primary{cluster=~'SES_IPV6_.*',type='mongo'})))"
```

event-host-label instance

message "{{ \$labels.instance }}" Persistence DB TPS exceeded , current value is {{ \$value }} !"

snmp-severity critical

snmp-clear-message "{{ \$labels.instance }}" Persistence DB TPS in control, current value is {{ \$value }} !"

NMS Destination Configuration

The following configuration is for reference only:

You can configure the NMS destination based on your requirements.

Example: SNMPv2

```
scheduler#config
scheduler(config)# alert snmp-v2-destination "10.1.1.1"
Value for 'community' (<string>): "cisco"
scheduler(config-snmp-v2-destination-10.1.1.1)# commit
Commit complete.
scheduler(config-snmp-v2-destination-10.1.1.1)# end
```

where, "10.1.1.1" is the SNMPv2 NMS destination address.

Example: SNMPv3

```
scheduler# config
scheduler(config)# alert snmp-v3-destination <nms_ip> e.g. 10.1.1.2
Value for 'user' (<string>): <username> e.g. cis_user
Value for 'auth-password' (<string>): <password string > e.g. cisco-123
Value for 'privacy-password' (<string>): <password string> e.g. cisco-123
scheduler(config-snmp-v3-destination-10.1.1.2)# auth-proto
[MD5,SHA] (SHA): SHA
scheduler(config-snmp-v3-destination-10.1.1.2)# privacy-p
Possible completions:
  privacy-password  privacy-protocol
scheduler(config-snmp-v3-destination-10.1.1.2)# privacy-protocol
[AES,DES] (AES): AES
scheduler(config-snmp-v3-destination-10.1.1.2)# engine-id
(<string>) (0x0102030405060708): 0x0102030405060708
scheduler(config-snmp-v3-destination-10.1.1.2)# commit
Commit complete.
```

```
scheduler(config-snmp-v3-destination-10.1.1.2)# end
scheduler#
```

where, "10.1.1.2" is the SNMPv3 NMS destination address.

All the configured NMS destinations in the system can be displayed using the following command:

```
scheduler# show running-config alert | tab
NMS
ADDRESS    COMMUNITY
-----
10.1.1.1    cisco

alert snmp-v3-destination 10.142.148.160
engine-id      0x0102030405060708
user           cis_user
auth-proto     SHA
auth-password  cisco-123
privacy-protocol AES
privacy-password cisco-123
!
```

