cnAAA application based alerts

Feature Summary and Revision History

Summary Data

Table 1: Summary Data

Applicable Products or Functional Area	cnAAA
Applicable Platform(s)	SMI
Feature Default Setting	Disabled – Configuration required to enable
Related Documentation	Not Applicable

Feature History

Table 2: Feature History

Feature Details	Release
First introduced.	2025.01.0

Feature Description

When the system detects an anomaly, it generates an alert notification. The system statistics are the cause for these alert notifications. You can set an expression to trigger an alert when the expression becomes true.

How it Works

This section describes how this feature works.

The Common Execution Environment (CEE) uses the Prometheus Alert Manager for alerting operations. The CEE YANG model, accessible through CLI or API, allows you to view the active alerts, silenced alerts, and alert history. During the application installation or upgradation, the system adds a set of preset alerting rules. Also, the applications can call the alert API directly to add or clear alerts. The Prometheus Alert Manager API (v2) is the standard API used.

The Prometheus Alerts Manager includes the following options:

- **DefiningAlert Rules**: This option defines the types of alerts that the Alert Manager should trigger. Use the Prometheus Query Language (PromQL) to define the alerts.
- **Defining Alert Routing**: This option defines the action the Alert Manager should take after receiving the alerts. At present, the SNMP Trapper is supported as the outbound alerting. Also, the CEE provides an Alert Logger for storing the generated alerts.

Configure alert rules

This section describes how to configure the alert rules.

To configure the alert rules, use the following configuration:

```
config
   alerts rules group alert_group_name
   rule rule_name
      expression promql_expression
      duration duration
      severity severity_level
      type alert-type
      annotation annotation_name
      value annotation_value
      end
```

NOTES:

- **alerts rules**—Specify the Prometheus alerting rules.
- **group** *alert_group_name*—Specify the Prometheus alerting rule group. One alert group can have multiple lists of rules. *alert-group-name* is the name of the alert group. *alert_group_name* must be a string in the range of 0–64 characters.
- **rule** *rule_name*—Specify the alerting rule definition. *rule_name* is the name of the rule.

- **expression** *promql_expression*—Specify the PromQL alerting rule expression. *promql_expression* is the alert rule query expressed in PromQL syntax. The *promql_expression* must be a string in the range of 0–64 characters.
- **duration** *duration*—Specify the duration of a true condition before it is considered true. *duration* is the time interval before the alert is triggered.
- **severity** *severity_level*—Specify the severity of the alert. *severity-level* is the severity level of the alert. The severity levels are critical, major, minor, and warning.
- **type** *alert_type*—Specify the type of the alert. *alert_type* is the user-defined alert type. For example, Communications Alarm, Environmental Alarm, Equipment Alarm, Indeterminate Integrity Violation Alarm, Operational Violation Alarm, Physical Violation Alarm, Processing Error Alarm, Quality of Service Alarm, Security Service Alarm, Mechanism Violation Alarm, or Time Domain Violation Alarm.
- **annotation** *annotation_name*—Specify the annotation to attach to the alerts. *annotation_name* is the name of the annotation.
- **value** *annotation_value*—Specify the annotation value. *annotation_value* is the value of the annotation.

View alert logger

The alert logger stores the alerts that cnAAA generates by default. View these alerts using the this command:

show alert history [filtering]

Narrow down the result using these filtering options:

- **annotations**—Specify the annotations of the alert.
- **endsAt**—Specify the end time of the alert.
- **labels**—Specify the additional labels of the alert.
- **severity**—Specify the severity of the alert.
- **source**—Specify the source of the alert.
- **startsAt**—Specify the start time of the alert.
- **type**—Specify the type of the alert.

This section provides sample outputs for active and historical alerts in the cnAAA system.

- **Active Alerts Summary**

  To view a summary of active alerts, use this command:

  ```
  [cpc-cluster] cee# show alerts active summary
  ```

  Sample output:

  ```
  Mon Oct  27 12:11:20.261 UTC+00:00
  alerts active summary db-no-tx dc6b09b14d2e
   severity minor
   startsAt 10-27T12:09:33
   source   System
   summary  Unknown
  alerts active summary k8s-pod-restarting 7849e8c4a6e8
   severity minor
  ```

```
 startsAt 10-27T12:07:21
 source   cpc-cluster-master-1
 summary  "Pod cpc-m13/crd-api-cpc-m13-cpc-engine-app-product..."
alerts active summary k8s-pod-crashing-loop 0007305db5f6
 severity critical
 startsAt 10-27T12:07:11
 source   cpc-cluster-master-1
 summary  "Pod cpc-m13/crd-api-cpc-m13-cpc-engine-app-product..."
alerts active summary k8s-pod-crashing-loop ff914a2e3df1
 severity critical
 startsAt 10-27T12:06:51
 source   cpc-cluster-master-1
 summary  "Pod cpc-m13/radius-ep-5 (radius-ep) is restarting ..."
alerts active summary k8s-pod-crashing-loop 7ddadee4531f
 severity critical
 startsAt 10-27T12:06:51
 source   cpc-cluster-master-1
 summary  "Pod cpc-m13/radius-ep-2 (radius-ep) is restarting ..."
alerts active summary k8s-pod-crashing-loop a910267b6f7a
 severity critical
 startsAt 10-27T12:06:51
 source   cpc-cluster-master-1
 summary  "Pod cpc-m13/radius-ep-0 (radius-ep) is restarting ..."
alerts active summary k8s-pod-crashing-loop 7ff2a7486644
 severity critical
startsAt 10-27T12:06:51
 source   cpc-cluster-master-1
 summary  "Pod cpc-m13/radius-ep-1 (radius-ep) is restarting ..."
```

- **Active Alerts Detail**

  To view detailed information for active alerts, use this command:

  ```
  [cpc-cluster] cee# show alerts active detail
  ```

  Sample output:

```
Mon Oct  27 12:11:09.961 UTC+00:00
alerts active detail db-no-tx dc6b09b14d2e
 severity    minor
 type        "Processing Error Alarm"
 startsAt    2025-10-27T12:09:33.265Z
 source      System
 summary     Unknown
 labels     [ "alertname: db-no-tx" "cluster: cpc-cluster_cee-m13" "monitor: prometheus"
 "replica: cpc-cluster_cee-m13" "severity: minor" ]
 annotations [ "type: Processing Error Alarm" ]
alerts active detail k8s-pod-restarting 7849e8c4a6e8
severity    minor
 type        "Processing Error Alarm"
 startsAt    2025-10-27T12:07:21.119Z
 source      cpc-cluster-master-1
 summary    "Pod cpc-m13/crd-api-cpc-m13-cpc-engine-app-production-rjio-7689c94786-774gp
 (crd) is restarting 4.03 times / 10 minutes."
 labels      [ "alertname: k8s-pod-restarting" "chartName: metrics" "cluster:
cpc-cluster_cee-m13" "component: kube-state-metrics" "container: crd" "hostname:
cpc-cluster-master-1" "instance: 192.102.0.105:8080" "job: kubernetes-pods" "monitor:
prometheus" "namespace: cpc-m13" "pod:
crd-api-cpc-m13-cpc-engine-app-production-rjio-7689c94786-774gp" "pod_template_hash:
755686975c" "release: cee-m13-cnat-monitoring" "replica: cpc-cluster_cee-m13" "severity:
 minor" "uid: 56c9e482-b47c-40ea-bc97-49698f8b74ee" ]
annotations [ "summary: Pod
cpc-m13/crd-api-cpc-m13-cpc-engine-app-production-rjio-7689c94786-774gp (crd) is
restarting 4.03 times / 10 minutes." "type: Processing Error Alarm" ]
```

- **Historical Alerts Count**

  To view a count of historical alerts by severity, use this command:

  ```
  [cpc-cluster] cee# show alerts history count
  ```

  Sample output:

  ```
  Mon Oct 27 12:44:36.463 UTC+00:00
  SEVERITY TOTAL

  minor 152
  major 130
  critical 532
  ```

- **Historical Alerts Detail Summary**

  To view a summary of historical alerts in detail, use this command:

  ```
  [cpc-cluster] cee# show alerts history detail summary
  ```

  Sample output:

  ```
  Mon Oct 27 12:45:05.478 UTC+00:00
  alerts history detail db-no-tx dc6b09b14d2e
  summary Unknown
  alerts history detail k8s-pod-restarting 7849e8c4a6e8
  summary "Pod cpc-m13/crd-api-cpc-m13-cpc-engine-app-production-rjio-7689c94786-774gp
  (crd) is restarting 1.02 times / 10 minutes."
  alerts history detail k8s-pod-crashing-loop 0007305db5f6
  summary "Pod cpc-m13/crd-api-cpc-m13-cpc-engine-app-production-rjio-7689c94786-774gp
  (crd) is restarting 2.03 times / 10 minutes."
  alerts history detail k8s-pod-crashing-loop 7ff2a7486644
  summary "Pod cpc-m13/radius-ep-4 (radius-ep) is restarting 2.03 times / 10 minutes."
  alerts history detail k8s-pod-crashing-loop 7ddadee4531f
  summary "Pod cpc-m13/radius-ep-2 (radius-ep) is restarting 2.03 times / 10 minutes."
  alerts history detail k8s-pod-crashing-loop ff914a2e3df1
  summary "Pod cpc-m13/radius-ep-5 (radius-ep) is restarting 2.03 times / 10 minutes."
  alerts history detail k8s-pod-crashing-loop 1d24967fc160
  summary "Pod cpc-m13/radius-ep-1 (radius-ep) is restarting 2.03 times / 10 minutes."
  alerts history detail k8s-pod-crashing-loop 162420e4043a
  summary "Pod cpc-m13/radius-ep-3 (radius-ep) is restarting 2.03 times / 10 minutes."
  alerts history detail k8s-pod-crashing-loop a910267b6f7a
  summary "Pod cpc-m13/radius-ep-0 (radius-ep) is restarting 2.03 times / 10 minutes."
  alerts history detail k8s-pod-restarting b9840e7118da
  summary "Pod cpc-m13/radius-ep-4 (radius-ep) is restarting 1.02 times / 10 minutes."
  ```

Sample alerts configuration

This section provides sample configurations that are defined in cnAAA.

Process-Level alerts

CDL Endpoint down

Use the following commands to configure alerts related to CDL Endpoint down.

```
alerts rules group cdl-ep-change
   rule pod-down
   expression up{pod=~'cdl-ep.*'} == 0
```

```
   duration 1m
   severity major
   type Equipment Alarm
   annotation description
   value CDL EP Pod Down
   exit
exit
```

CDL Slot State Change

Use the following commands to configure alerts related to CDL slot state change.

```
alerts rules group cdl-slot-change
   rule pod-down
   expression up{pod=\"cdl-slot-session-c1-m1-0\"} == 0
   severity major
   type Equipment Alarm
   annotation description
   value CDL Pod Slot Change
   exit
exit
```

ETCD State Change

Use the following commands to configure alerts related to etcd state change.

```
alerts rules group ep-mapping-change
   rule pod-down
   expression up{pod=~'etcd-cnAAA.*'} == 0
   duration 1m
   severity major
   type Equipment Alarm
   annotation description
   value EP Mapping Change
   exit
exit
```

Grafana Dashboard State Change

Use the following commands to configure alerts related to Grafana dashboard state change.

```
alerts rules group grafana-dashboard-change
   rule pod-down
   expression up{pod=~'grafana-dashboard.*'} == 0
   duration 1m
   severity major
   type Equipment Alarm
   annotation description
   value Grafana Dashboard Change
   exit
exit
```

Kafka State Change

Use the following commands to configure alerts related to Kafka state change.

```
alerts rules group kafka-change
   rule pod-down
   expression up{pod=~'kafka.*'} == 0
   duration 1m
   severity major
   type Equipment Alarm
   annotation description
   value Kafka Changed
   exit
exit
```

cnAAA Engine State Change

Use the following commands to configure alerts related to cnAAA Engine state change.

```
alerts rules group cnAAA-engine-change
   rule pod-down
   expression up{pod=~'cnAAA-engine-cnAAA.*'} == 0
   duration 1m
   severity major
   type Equipment Alarm
   annotation description
   value cnAAA Engine Changed
   exit
exit
```

RADIUS Endpoint State Change

Use the following commands to configure alerts related to RADIUS endpoint state change.

```
alerts rules group cnAAA-RADIUS-ep-change
   rule pod-down
   expression up{pod=~'cnAAA-RADIUS-ep.*'} == 0
   duration 1m
   severity major
   type Equipment Alarm
   annotation description
   value cnAAA RADIUS EP Change
   exit
exit
```

Call Flow procedure alerts

PLF query request

Use the following commands to configure alerts related to PLF Query Request.

```
alerts rules group cnAAAProcStatus
   interval-seconds 300
```

```
   rule PLFRequest
   severity major
   type Communications Alarm
   annotation summary
   value This alert is fired when the success percentage of PLF request is lesser threshold.

   exit
exit
```

NAP notification request

Use the following commands to configure alerts related to NAP Notification Request.

```
alerts rules group cnAAAProcStatus
   interval-seconds 300
   rule NAPNotification
   severity major
   type Communications Alarm
   annotation summary
   value This alert is fired when the success percentage of NAP request is lesser threshold.

   exit
exit
```

System alerts

Disk full alert

Use the following commands to configure alerts related to disk full alert.

```
alerts rules group
   rule node-disk-running-full
   expression node_filesystem_usage > 0.0001
   duration 5m
   severity critical
   type Processing Error Alarm
   annotation disk_full
   value test
   exit
exit
```

VM down alert

Use the following commands to configure alerts related to virtual machine down alert.

```
alerts rules group vm-state-change
   rule vm-down
   expression up{pod=~\"node-expo.*\"} == 0
   duration 1m
   severity major
   type Equipment Alarm
   annotation summary
   value VM Down
```

```
   exit
exit
```

High memory usage

Use the following commands to configure alerts related to high memory usage.

```
alerts rules group memory-util-high
   rule mem-util-high
   expression avg(node_memory_MemAvailable_bytes /node_memory_MemTotal_bytes * 100) by
(hostname) < 20
   duration 1m
   severity critical
   type Processing Error Alarm
   annotation mem_util_high
   value Hig Memory Usage
   exit
exit
```

High disk usage

Use the following commands to configure alerts related to high disk usage alert.

```
alerts rules group disk-util-high
   duration 1m
   rule disk-util-high
   expression avg (node_filesystem_avail_bytes{mountpoint =\"/\"}
/node_filesystem_size_bytes{mountpoint =\"/\"} *100) by (hostname) <20
   severity critical
   type Processing Error Alarm
   annotation description
   value Hig Memory Usage
   exit
exit
```

High CPU usage

Use the following commands to configure alerts related to high CPU usage alert.

```
alerts rules group cpu-util-high
   rule cpu-util-idle
   duration 1m
   expression avg(rate(node_cpu_seconds_total{mode='idle'}[1m])) by (hostname) *100 < 50

   severity critical
   type Processing Error Alarm
   annotation description
   value Hig CPU
   exit
exit
```

Support notifications for system and application alarms

Revision History

Feature Name	Release Information	Description
Notification and Alarms Support	2025.02	This feature monitors policy application during startup and operations to ensure system stability. It triggers alarms on configuration failures, initiating actions to maintain system reliability through notifications.
Support Notifications for System and Application Alarms	2025.01	The Monitoring and Alert Notification framework enhances system and application reliability by issuing SNMP traps to alert critical events. It categorizes alerts into proactive (requiring attention) and reactive (indicating past events), covering cnAAA issues, ensuring timely interventions for network stability.

Overview

The Monitoring and Alert Notification framework enhances system and application reliability by tracking events and sending alerts. It notifies network administrators of critical events and changes, enabling timely interventions to maintain system stability. The framework provides SNMP notification traps, categorized into proactive and reactive alerts. Proactive traps are alerts based on system events or changes that require attention, while reactive traps notify administrators of events that have already occurred. Alerts cover issues in cnAAA, database problems, load averages, and link status.

The Notification and Alarms ensures policy configurations are correctly applied during startup and ongoing operation. It identifies configuration failures and raises critical alarms to maintain system stability. This feature monitors and notifies two key alarms: "PoliciesNotConfigured" and "PolicyConfiguration."

System Alarms

System alarm configurations are required to configure in cee-ops-center within cnAAA deployment, following are the examples of alarms for DB, CPU etc.

- **All-SCDB-DB-Members-Down:**

  Description: Connectivity failure to all members of the SCDB, ADMIN, and SPR replica sets.

  Expression:sum(mongo_node_state{replica_set=~'sdb.*'}) == 0

- **All-ADMIN-DB-Members-Down:**

Description: This alert is triggered when all DB members of admin are down.

Expression: sum(mongo_node_state{replica_set=~'admin.*'}) == 0

- **All-SPR-DB-Members-Down:**

  Description: This alert is triggered when all DB members of SPR are down.

  Expression:sum(mongo_node_state{replica_set=~'spr.*'}) == 0

- **Primary-SCDB-Member-Down:**

  Description: Inability to locate the primary member for the SCDB sets.

  Expression:mongo_primary_reachable{replica_set=~'sdb.*'} == 0

- **Primary-SPR-DB-Member-Down:**

  Description: Inability to locate the primary member for the SPR replica sets.

  Expression: mongo_primary_reachable{replica_set=~'spr.*'} == 0

- **Primary-Admin-DB-Member-Down:**

  Description: Inability to locate the primary member for the admin replica sets.

  Expression: mongo_primary_reachable{replica_set=~'admin.*'} == 0

- **Secondary-Admin-DB-Member-Down:**

  Description: Inability to locate the secondary member for the admin sets.

  Expression: sum(mongo_node_state{replica_set=~'admin.*', state='secondary'}) == 0

- **Secondary-SCDB-Member-Down:**

  Description: Inability to locate the secondary member for the SCDB sets.

  Expression: sum(mongo_node_state{replica_set=~'sdb.*', state='secondary'}) == 0

- **Secondary-SPR-DB-Member-Down:**

  Description: Inability to locate the secondary member for the SPR replica sets.

  Expression: sum(mongo_node_state{replica_set=~'spr.*', state='secondary'}) == 0

- **load-average-high:**

  Description: This alarm is triggered when the the system's load average surpasses a configured threshold for over 5 minutes.

  Expression: node_load5 > 5

- **Link-Down:**

  Description: This alarm is triggered when the connectivity or ping failure to a system-attached physical interface.

  Expression: node_network_up{device = 'ens192'} == 0

- **Replication-lag:**

  Description: This alarm is triggered when the replication lag exceeds 2 seconds.

  Expression: mongo_replication_lag_seconds > 2

The Alert Logger records all generated Alarm by default, accessible through specific show commands for reviewing stored Alarm:

- show alert history {detail | summary}
- show alert active {detail | summary}

Monitoring alarm support for SVN repository

This feature monitors critical Subversion (SVN) repositories, detecting if any essential repository or `.broadhopFileRepository` file with in the essential repository is removed. As an early warning system, it prevents potential system outages and ensures development environment integrity. When a deletion occurs, the system triggers a critical alarm, which helps in minimize service disruption.

Configure critical SVN repository monitoring in cnAAA Ops-Center

Procedure

Follow these steps to configure critical SVN repository monitoring.

Step 1 Log in to cnAAA Ops-Center and enter the configuration mode.

```
config
```

Step 2 Enter this show command to display all the configured engine properties and their values.

```
show full-configuration engine <engine-group> properties
```

Step 3 Define the list of SVN repositories to monitor.

```
properties list.of.repos.to.be.monitored
value <repository-names>
exit
```

Note
Each repository name must be separated by a comma (,) with no spaces between names.

Step 4 Set the frequency for the system checks of the SVN repositories.

```
properties com.broadhop.svn.monitoring.interval.ms
value <interval-in-ms>
exit
```

Note
The default interval is 300,000 milliseconds (five minutes).

Example

```
config
# Define the list of SVN repositories to monitor.
engine cpc-green properties list.of.repos.to.be.monitored
value new_import8,new_import9,new_import10,new_import11
exit
```

```
# Set the frequency for the system checks of the SVN repositories.
engine cpc-green properties com.broadhop.svn.monitoring.interval.ms
value 40000
exit
```

Configure alarm for monitoring critical SVN repository

Procedure

Follow these steps to configure critical SVN repository monitoring.

Step 1 Login to CEE Ops-Center and enter the configuration mode.

```
config
```

Step 2 Enter the show command to display all available alerts.

```
Show full configuration alerts
```

Step 3 Create two alarm rules to trigger alerts when a repository or the `.broadhopFileRepository` file is deleted.

a) Alarm for SVN repository deletion: This alarm triggers when a monitored SVN repository is completely removed.

```
alerts rules group svn-repo-deleted
rule SvnRepoDeleted
expression "sum(svn_repo_deleted_total{}!= 0) by (message_type) "
severity critical
type "Communications Alarm"
annotation summary
value "Expected SVN Repository missing is: {{ $labels.message_type }} "
exit
exit
exit
```

Sample Alarm Output:

```
 alerts active detail SvnRepoDeleted 8802f40d4e65
 severity    critical
 type        "Communications Alarm"
 startsAt    2025-10-27T09:12:32.190Z
 source      System
 summary     " Expected SVN Repository missing is: TestPB1 "
 labels     [ "alertname: SvnRepoDeleted" "cluster: unknown_cee" "message_type: TestPB1" "monitor:
 prometheus" "replica: unknown_cee" "severity: critical" ]
 annotations [ "summary:  Expected SVN Repository missing is: TestPB1 " "type: Communications
Alarm" ]
```

b) Alarm for `.broadhopFileRepository` file deletion: This alarm triggers when the .broadhopFileRepository file is missing from a monitored SVN repository.

```
alerts rules group config-file-deleted-from-svn-repo
rule ConfigFileDeletedFromSvnRepo
expression "sum(config_file_deleted_from_svn_repo_total{}!=0) by (message_type) "
severity critical
type "Communications Alarm"
annotation summary
value "Expected SVN Repository missing the metadata is: {{ $labels.message_type }} "
exit
exit
exit
```

Sample Alarm Output:

```
 alerts active detail ConfigFileDeletedFromSvnRepo 90a6cec6b372
 severity    critical
 type        "Communications Alarm"
 startsAt    2025-10-27T09:40:23.610Z
 source      System
 summary     " Expected SVN Repository missing  the metadata is: TestPB "
 labels      [ "alertname: ConfigFileDeletedFromSvnRepo" "cluster: unknown_cee" "message_type:
TestPB" "monitor: prometheus" "replica: unknown_cee" "severity: critical" ]
 annotations [ "summary:  Expected SVN Repository missing  the metadata is: TestPB " "type:
Communications Alarm" ]
```

Application alarms

Application alarm configurations are required to configure in cee-ops-center within cnAAA deployment, following are the examples of application Alarm:

- **Access Reject**:

  Description: This alarm is triggered when the number of Access Reject messages exceeds a threshold limit.

  Expression:sum(rate(radius_responses_total{message_type="AccessReject"}[1m])) > N

- **Service Stop Request**:

  Description: This alarm is triggered when the number of Service Stop Requests exceeds a threshold limit.

  Expression:avg(rate(radius_accounting_request_total{accountingType="SessionAccounting", statusType="Stop"}[1m])) > N

- **Service Stop Response**:

  Description: This alarm is triggered when the number of Service Stop Responses exceeds a threshold limit.

  Expression:sum(rate(radius_accounting_response_total{accountingType="SessionAccounting", statusType="Stop"}[1m])) > N

- **PER EP TPS Radius**:

  Description: This alarm is triggered when the Transactions Per Second (TPS) exceeds a threshold limit.

  Expression:sum(irate(radius_requests_total[1m]) or vector(0)) + sum(irate(radius_accounting_request_total[1m]) or vector(0)) by (pod)

- **Memory Used in Radius POD**:

  Description: This alarm is triggered when the memory usage in a Radius POD exceeds a threshold limit.

  Expression:sum(rate(jvm_memory_bytes_used{namespace="pcf", component="cps-radius-ep"}[1m]))

- **GC Time Period**:

  Description: This alarm is triggered when the garbage collection (GC) time exceeds a threshold limit.

  Expression:

increase(jvm_gc_collection_seconds_sum{component="cps-radius-ep",namespace="pcf"}[$__interval])

- **Total Radius Auth Messages Overload Rejected**:

  Description: This alarm is triggered when the number of Radius auth messages rejected due to overload exceeds a threshold limit.

  Expression:

  sum(rate(total_radius_auth_messages_overload_rejected{message_type="AccessReject"}[1m])) > N

- **Total Radius Messages Overload Dropped on Session Accounting**

  Description: This alarm is triggered when the number of Radius messages dropped on Session Accounting due to overload exceeds a threshold limit.

  Expression:

  sum(rate(total_radius_messages_overload_dropped{message_type="SessionAccounting",status_type="Start"}[1m])) > N

- **Total Radius Messages Overload Dropped on Service Accounting**

  Description: This alarm is triggered when the number of Radius messages dropped on Service Accounting due to overload exceeds a threshold limit.

  Expression:

  sum(rate(total_radius_messages_overload_dropped{message_type="ServiceAccounting",status_type="Start"}[1m])) > N

- **GRPC Message Send Total on Accounting Request**

  Description: This alarm is triggered when the GRPC message send total for Accounting Request exceeds a threshold limit.

  Expression:

  sum(rate(grpc_message_send_total{message_type="AccountingRequest"}[1m])) > N

- **GRPC Message Send Total on Access Request**

  Description: This alarm is triggered when the GRPC message send total for Access Request exceeds a threshold limit.

  Expression: sum(rate(grpc_message_send_total{message_type="AccessRequest"}[1m])) > N

- **Radius Proxy Accounting Response Total on Error**

  Description: This alarm is triggered when the Radius proxy accounting responses with errors exceed a threshold limit.

  Expression:
  sum(rate(radius_proxy_accounting_response_total{accounting_type="ServiceAccounting",status_type="Start",result="ERROR"}[1m])) > 1

- **CoA Timeout**

  Description: This alarm is triggered when Change of Authorization (CoA) timeouts exceed a threshold limit.

  Expression: sum(rate(radius_request_timeout_total{message_type="CoaRequest"}[1m])) > N

  Policy Engine Timeout Message on Accounting Request

Description: This alarm is triggered when Policy Engine timeout messages for Accounting Request exceed a threshold limit.

Expression:
sum(rate(POLICY_ENGINE_TIMEOUT_MESSAGE{message_type="AccountingRequest"}[1m])) > N

- **Policy Engine Timeout Message on Access Request**

  Description: This alarm is triggered when Policy Engine timeout messages for Access Request exceed a threshold limit.

  Expression:
  sum(rate(POLICY_ENGINE_TIMEOUT_MESSAGE{message_type="AccessRequest"}[1m])) > N

  Policy Engine Message Total on Access Request

  Description: This alarm is triggered when the total Policy Engine messages for Access Request exceed a threshold limit.

  Expression: sum(rate(policy_engine_message_total{message_type="AccessRequest"}[1m])) > N

  Policy Engine Message Total on Accounting Request

  Description: This alarm is triggered when the total Policy Engine messages for Accounting Request exceed a threshold limit.

  Expression: sum(rate(policy_engine_message_total{message_type="AccountingRequest"}[1m])) > N

- **Dispatch Error Total on Bundled CoA Request**

  Description: This alarm is triggered when dispatch errors for Bundled Change of Authorization (CoA) Requests exceed a threshold limit.

  Expression: sum(rate(dispatch_error_total{message_type="AsyncCoARequest"}[1m])) > N

- **Dispatch Error Total on AsyncCoA Request**

  Description: This alarm is triggered when dispatch errors for Async Change of Authorization (CoA) Requests exceed a threshold limit.

  Expression: sum(rate(dispatch_error_total{message_type="BundledCoARequest"}[1m])) > N

- **Process Message Total on Accounting Response**

  Description: This alarm is triggered when the total processed messages for Accounting Response exceed a threshold limit.

  Expression: sum(rate(process_message_total{message_type="AccountingResponse"}[1m])) > N

- **Process Message Total on Access Accept**

  Description: This alarm is triggered when the total processed messages for Access Accept exceed a threshold limit.

  Expression: sum(rate(process_message_total{message_type="AccessAccept"}[1m])) > N

- **Outbound Request Total on Proxy Accounting**

  Description: This alarm is triggered when the total outbound requests for Proxy Accounting exceed a threshold limit.

  Expression: sum(rate(outbound_request_total{message_type="ProxyAccounting"}[1m])) > N

- **Outbound Request Total on CoA Request**

  Description: This alarm is triggered when the total outbound requests for Change of Authorization (CoA) Requests exceed a threshold limit.

  Expression: sum(rate(outbound_request_total{message_type="CoARequest"}[1m])) > N

- **Inbound Request Total on Proxy Accounting**

  Description: This alarm is triggered when the total inbound requests for Proxy Accounting exceed a threshold limit.

  Expression: sum(rate(inbound_request_total{message_type="ProxyAccounting"}[1m])) > N

- **Inbound Request Total on Access Request**

  Description: This alarm is triggered when the total inbound requests for Access Request exceed a threshold limit.

  Expression:sum(rate(inbound_request_total{message_type="AccessRequest"}[1m])) > N

- **Record Conflict Merge Total**

  Description: This alarm is triggered when the total record conflict merges exceed a threshold limit.

  Expression:sum(rate(record_conflict_merge_total[1m])) > N

- **Radius Access Request Message on Error**

  Description: This alarm is triggered when Radius Access Request messages on errors exceed a threshold limit.

  Expression: sum(rate(message_total{type="radius-access-request-message", status="error"}[1m])) > N

- **I Send Access Accept on Error**

  Description: This alarm is triggered when "I Send Access Accept" messages on errors exceed a threshold limit.

  Expression:sum(rate(action_total{type="i-send-access-accept", status="error"}[1m]))

- **Radius Accounting Message on Error**

  Description: This alarm is triggered when Radius Accounting messages on errors exceed a threshold limit.

  Expression:sum(rate(message_total{type="radius-accounting-message", status="error"}[1m])) > N

Critical Alert to be Raised RADIUS Pod not Sending Access-Accept

This feature monitors RADIUS pods and alerts when they stop sending **access-accept** responses. It uses Prometheus to collect metrics and an alert rule to trigger a critical alert if a pod fails to respond for a set time. This ensures rapid issue detection, resolution, and improved RADIUS service reliability.

How RADIUS pod access-accept monitoring works

This feature operates through an integrated monitoring pipeline:

1. **Metric export**: Each RADIUS pod exports operational metrics, including `radius_responses_total`. This metric tracks the cumulative count of RADIUS responses, with `message_type="AccessAccept"` indicating successful responses..
2. **Prometheus data collection**: A Prometheus instance, deployed in the CEE namespace, collects these metrics from all RADIUS pods, typically in the CPC namespace. Prometheus continuously collects this time-series data.
3. **Real-time rate calculation**: Prometheus calculates the rate of access-accept responses over a one-minute window for each RADIUS pod. It uses the `rate()` function on the `radius_responses_total{message_type="AccessAccept"}` metric.
4. **Alert condition evaluation**: A predefined Prometheus alert rule evaluates this calculated rate. If the rate of access-accept responses for any pod drops to zero and remains at zero for one continuous minute, the alert condition is met.
5. **Critical alert trigger**: If the alert condition is met, the system triggers a critical alert named `RadiusPodNotSendingAccessAccept`. This alert includes detailed annotations identifying the specific pod and its namespace that is experiencing the issue.
6. **Automatic alert resolution**: If the affected RADIUS pod resumes sending access-accept responses, the rate increases above zero. The Prometheus alert rule automatically resolves the critical alert.
7. **Visualization (Optional)**: All collected metrics and alert states can be visualized on a Grafana dashboard. This provides real-time insights into RADIUS service health and performance.

RADIUS Pod access-accept alert

RADIUS pods export metrics, specifically `radius_responses_total` with the `message_type="AccessAccept"` label. This metric is crucial for data collection.

The cnAAA system is configured to raise a critical alert when a RADIUS pod does not send access-accept responses:

```
groups:
- name: RadiusAlerts
  rules:
  - alert: RadiusPodNotSendingAccessAccept
    expr: sum(rate(radius_responses_total{message_type="AccessAccept"}[1m])) by (pod,
namespace) == 0
    for: 1m # The condition must persist for one minute before the alert is triggered.
    labels:
      severity: critical
    annotations:
      summary: "Radius pod not sending AccessAccept"
      description: "The pod {{ $labels.pod }} in namespace {{ $labels.namespace }} is not
sending AccessAccept responses."
```

Configure Grafana dashboard for RADIUS access-accept:

- Ensure Grafana is configured with Prometheus as a data source.
- Create new panels on a Grafana dashboard to visualize the `radius_responses_total` metric.

  Query for Access-Accept rate per pod:

  ```
  sum(rate(radius_responses_total{message_type="AccessAccept"}[1m])) by (pod, namespace)
   == 0
  ```

This section provides an example of the `RadiusPodNotSendingAccessAccept` alert.

To view the alert history, use this command:

```
cee# show alerts history summary | tab | include Radius

RadiusPodNotSendingAc  7eb5eacfa11f  critical  10-23T05:12:58  15m20s    radius-ep-1
      Radius pod not sending AccessAccept
RadiusPodNotSendingAc  16e12fc1f5c0  critical  10-23T05:04:18  2m50s     radius-ep-2
      Radius pod not sending AccessAccept
RadiusPodNotSendingAc  fbcc7250ecbf  critical  10-23T04:51:48  2m20s     radius-ep-1
      Radius pod not sending AccessAccept
```

The alert clears once the RADIUS endpoint pod resumes sending Access-Accept responses.

To view active alerts, use this command:

```
cee# show alerts active summary | tab | include Radius
```

Note

- If no traffic runs, alerts are present by default for all RADIUS pods. Alerts clear when traffic resumes.
- This alert also generates when no RADIUS endpoint pod entries appear in ETCD. This indicates that no RADIUS endpoint pods handle traffic.

Follow these steps to troubleshoot `RadiusPodNotSendingAccessAccept` alerts.

- Inspect the RADIUS pod and its logs. Describe the RADIUS pod and check its logs for error-related information.

  ```
  kubectl describe pod <radius-pod-name> -n <namespace>
  kubectl logs <radius-pod-name> -n <namespace>
  ```

- Restart or delete the RADIUS pod. If issues persist with the RADIUS pod, restart or delete it.

  ```
  kubectl rollout restart deployment/<radius-deployment-name> -n <namespace>
  # OR
  kubectl delete pod <radius-pod-name> -n <namespace>
  ```

Configuration

Alert rules can be configured using the following commands:

```
configure
    alerts rules group alert_group_name
    interval-seconds seconds
    rule rule_name
        expression promql_expression
        duration duration
        severity severity_level
        type alert_type
        annotation annotation_name
        value annotation_value
    exit
exit
```

Key Configuration Parameters

- alert_group_name: Name of the alert group, up to 64 characters.
- interval-seconds: The evaluation interval in seconds.

- rule_name: Name of the alerting rule.
- promql_expression: PromQL syntax-based alert rule query.
- duration: Duration before an alert condition is considered true.
- severity_level: Urgency level, ranging from critical to warning.
- alert_type: User-defined alert types, e.g., Operational Violation, Security Service.
- annotation_name/value: Annotations attached to alerts.

Policy configuration counters

Policy configurations during system startup and policy structure changes are monitored using engine counters to ensure system stability.

To view the counters, use this command:

```
kubectl exec -it <engine-pod> -n <namespace> -- curl -G http://127.0.0.1:8080/metrics |
grep poli
```

- **Set up policies_not_configured_total Counter**

  This counter tracks the success or failure of policy configurations during system startup. It increments when policy configuration fails and resets when successful.

  Example:

  ```
  # HELP policies_not_configured_total Total of policies_not_configured
  # TYPE policies_not_configured_total counter
  policies_not_configured_total{node_type="unknown", message_type="Policies Not
  Configured",}
  ```

- **last_policy_configuration_failed_total**

  This counter monitors changes to the system policy structure. It increments if the last policy configuration attempt fails and resets upon success.

  Example:

  ```
  # HELP last_policy_configuration_failed_total Total of last_policy_configuration_failed
  # TYPE last_policy_configuration_failed_total counter
  last_policy_configuration_failed_total{node_type="unknown", message_type="Last Policy
  Configuration Failed due to java.lang.ArithmeticException / by zero",}
  ```

Alarms

This section details alarms triggered for policy configuration issues and highlights the need to monitor and resolve these alerts to maintain cnAAA system functionality.

- **Policies not configured**

  This alarm activates when the policy engine cannot locate any applicable policies during startup. It is critical and requires immediate attention to ensure system services function correctly.

  Formula:

  ```
  sum(policies_not_configured_total) != 0
  ```

- **Policy configuration**

This alarm activates when a change to the system policy structure fails. Although the system remains stable and operational, it is advisable to check the notification's additional information to determine if further investigation is needed.

Formula:

```
sum(last_policy_configuration_failed_total{}!=0) by (message_type)
```

CEE configuration:

```
alerts rules group policy-config
  rule PoliciesNotConfigured
    expression "sum(policies_not_configured_total) != 0"
    severity   critical
    type       "Communications alarm"
    annotation summary
    value "Policies not configured"
    exit
  exit
exit

alerts rules group last-policy-config
  rule PolicyConfiguration
     expression "sum(last_policy_configuration_failed_total{}!=0) by (message_type)"
     severity   critical
     type       "Communications alarm"
     annotation summary
     value "{{ $labels.message_type }}"
     exit
  exit
exit
```

Check active alerts

Follow these steps to check active alerts:

Procedure

Step 1 Enter the following command in the CLI

```
show alerts active ?
```

Sample alert output:

This section provides sample outputs for active alerts in the cnAAA system.

- **Active Alerts Count**

```
[cpc-cluster] cee# show alerts active count
Sample output:
```
```
Mon Oct 27 12:11:31.463 UTC+00:00
SEVERITY TOTAL

minor 12
major 4
critical 15
```

- **Active Alerts Summary**

```
[cpc-cluster] cee# show alerts active summary
Mon Oct 27 12:11:20.261 UTC+00:00
alerts active summary db-no-tx dc6b09b14d2e
severity minor
startsAt 10-27T12:09:33
source System
summary Unknown
alerts active summary k8s-pod-restarting 7849e8c4a6e8
severity minor
startsAt 10-27T12:07:21
source cpc-cluster-master-1
summary "Pod cpc-m13/crd-api-cpc-m13-cpc-engine-app-product..."
alerts active summary k8s-pod-crashing-loop 0007305db5f6
severity critical
startsAt 10-27T12:07:11
source cpc-cluster-master-1
summary "Pod cpc-m13/crd-api-cpc-m13-cpc-engine-app-product..."
alerts active summary k8s-pod-crashing-loop ff914a2e3df1
severity critical
startsAt 10-27T12:06:51
source cpc-cluster-master-1
summary "Pod cpc-m13/radius-ep-5 (radius-ep) is restarting ..."
alerts active summary k8s-pod-crashing-loop 7ddadee4531f
severity critical
startsAt 10-27T12:06:51
source cpc-cluster-master-1
summary "Pod cpc-m13/radius-ep-2 (radius-ep) is restarting ..."
alerts active summary k8s-pod-crashing-loop a910267b6f7a
severity critical
startsAt 10-27T12:06:51
source cpc-cluster-master-1
summary "Pod cpc-m13/radius-ep-0 (radius-ep) is restarting ..."
alerts active summary k8s-pod-crashing-loop 7ff2a7486644
severity critical
startsAt 10-27T12:06:51
source cpc-cluster-master-1
summary "Pod cpc-m13/radius-ep-1 (radius-ep) is restarting ..."
```

- **Active Alerts Detail**

```
[cpc-cluster] cee# show alerts active detail
Mon Oct 27 12:11:09.961 UTC+00:00
alerts active detail db-no-tx dc6b09b14d2e
severity minor
type "Processing Error Alarm"
startsAt 2025-10-27T12:09:33.265Z
source System
summary Unknown
labels [ "alertname: db-no-tx" "cluster: cpc-cluster_cee-m13" "monitor: prometheus" "replica:
cpc-cluster_cee-m13" "severity: minor" ]
annotations [ "type: Processing Error Alarm" ]
alerts active detail k8s-pod-restarting 7849e8c4a6e8
severity minor
type "Processing Error Alarm"
startsAt 2025-10-27T12:07:21.119Z
source cpc-cluster-master-1
summary "Pod cpc-m13/crd-api-cpc-m13-cpc-engine-app-production-rjio-7689c94786-774gp (crd) is
restarting 4.03 times / 10 minutes."
labels [ "alertname: k8s-pod-restarting" "chartName: metrics" "cluster: cpc-cluster_cee-m13"
"component: kube-state-metrics" "container: crd" "hostname: cpc-cluster-master-1" "instance:
192.102.0.105:8080" "job: kubernetes-pods" "monitor: prometheus" "namespace: cpc-m13" "pod:
crd-api-cpc-m13-cpc-engine-app-production-rjio-7689c94786-774gp" "pod_template_hash: 755686975c"
 "release: cee-m13-cnat-monitoring" "replica: cpc-cluster_cee-m13" "severity: minor" "uid:
```

```
56c9e482-b47c-40ea-bc97-49698f8b74ee" ]
annotations [ "summary: Pod cpc-m13/crd-api-cpc-m13-cpc-engine-app-production-rjio-7689c94786-774gp
 (crd) is restarting 4.03 times / 10 minutes." "type: Processing Error Alarm" ]
```

Step 2 Choose one of the following options to view the alerts:

```
count     Count Version
detail    Detailed Version
summary   Compact Version
```

Note
For more information on ULB, see the *CEE Configuration and Administration Guide*.

Add Hostname in SNMP Traps

The Monitoring and Alert Notification framework improves system reliability by tracking events and sending alerts to network administrators. It enables timely responses to critical events and system changes to maintain stability. The framework uses Simple Network Management Protocol (SNMP) notification traps, which include:

- Proactive traps: Alerts based on system events or changes that require attention.
- Reactive traps: Alerts for events that have already occurred.

Configure SNMP trap alerts with hostname

To enhance monitoring by including the hostname in SNMP traps and ensure NMS identify the device that sent each trap, follow these configuration and verification steps.

Procedure

Step 1 Log in to the CEE Ops Center.

Step 2 Configure basic SNMP trapper settings.

- Enable the SNMP trapper:

  ```
  snmp-trapper enable true
  ```

- Define the SNMP version 2c (v2c) target receivers for both IPv4 and IPv6 by specifying the NMS IP address, port , and community string:

  Sample configuration for an IPv4 target:

  ```
  snmp-trapper v2c-target 10.1.36.96
   port 162
   community Re4D0nLy5TrinG
  exit
  ```

  Sample configuration for an IPv6 target:

  ```
  snmp-trapper v2c-target 1111::10:1:47:9
   port 162
  ```

```
     community Re4D0nLy5TrinG
    exit
```

Step 3 Create specific alert rules to trigger SNMP traps based on conditions such as CPU utilization.

Sample configuration:

```
alerts rules group cpu-util-high
 rule cpu-util-idle
  expression "avg(rate(node_cpu_seconds_total{mode='idle'}[1m])) by (hostname) *100 > 10"
  duration 20s
  severity critical
  type "Processing Error Alarm"
  annotation description
   value "High CPU"
  exit
 exit
exit
```

Note
These rules determine which events will generate alerts for SNMP traps.

Step 4 Specify the source IP addresses from which SNMP traps should originate.

Sample configuration:

```
snmp-trapper source-ip-routes internal-vip 10.192.2.31
snmp-trapper source-ip-routes default-external-vip 10.84.117.99
snmp-trapper source-ip-routes source-external-vips cee-gamma-cncps-cee
 external-vip 10.84.117.99
exit
```

Note
This ensures SNMP traps are sent from the correct interfaces or Virtual IPs.

Step 5 Enter the `TCP dump` command to verify the alerts.

Sample configuration:

```
2025-05-21 11:47:12.330521 IP 10.1.46.84.49592 > 10.1.36.96.162:  V2Trap(706)
 .1.3.6.1.2.1.1.3.0=321056 .1.3.6.1.6.3.1.1.4.1.0=.1.3.6.1.4.1.9.9.999.0.1
.1.3.6.1.4.1.9.9.999.1.1="cpu-util-idle"
.1.3.6.1.4.1.9.9.999.1.2="/rid7802257-brprakas-1-master3" .1.3.6.1.4.1.9.9.999.1.3="critical"
 .1.3.6.1.4.1.9.9.999.1.4=07_e9_05_15_0b_2d_16_00_2b_00_00 .1.3.6.1.4.1.9.9.999.1.5="Processing Error
 Alarm"
.1.3.6.1.4.1.9.9.999.1.6="{"alertname": "cpu-util-idle", "cluster": "unknown_cee", "hostname":
"rid7802257-brprakas-1-master3",
 "monitor": "prometheus", "replica": "unknown_cee", "severity": "critical", "instance":
"rid7802257-brprakas-1-master3",
"description": "High CPU", "type": "Processing Error Alarm"}" .1.3.6.1.4.1.9.9.999.1.7="unknown_cee"
 .1.3.6.1.4.1.9.9.999.1.8="*"
.1.3.6.1.4.1.9.9.999.1.9="rid7802257-brprakas-1-master3"
.1.3.6.1.4.1.9.9.999.1.10="rid7802257-brprakas-1-master3" .1.3.6.1.4.1.9.9.999.1.11="*"
 .1.3.6.1.4.1.9.9.999.1.12="*"
```

Note
In alert PCAP, hostname should be present.

Step 6 Enter the `show alerts active detail` command to verify the active alerts.

Sample configuration:

```
] cee# show alerts active detail .
alerts active detail cpu-util-idle eb807845cc3d
 severity    critical
 type        "Processing Error Alarm"
 startsAt    2025-05-26T11:27:32.902Z
 source      rid7802257-brprakas-1-master3
 summary     Unknown
 labels      [ "alertname: cpu-util-idle" "cluster: unknown_cee" "hostname:
rid7802257-brprakas-1-master3" "monitor: prometheus" "replica: unknown_cee" "severity: critical" ]
 annotations [ "description: High CPU" "type: Processing Error Alarm" ]
```