

Application-based Alerts

- Feature Summary, on page 1
- Feature Description, on page 2
- How it Works, on page 2
- Configuring Alert Rules, on page 2

Feature Summary

Summary Data

Table 1: Summary Data

Applicable Product(s) or Functional Area	5G-NRF
Applicable Platform(s)	SMI
Feature Default Setting	Disabled - Configuration Required
Related Changes in this Release	Not Applicable
Related Documentation	Not Applicable

Revision History

Table 2: Revision History

Revision Details	Release
First introduced.	2026.01

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

The Common Execution Environment (CEE) uses the Prometheus Alert Manager for alerting operations. The CEE YANG model - either through CLI or API - allows users to view the active alerts, silenced alerts, and alert history. 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:

- **Defining Alert 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.

Configuring Alert Rules

To configure the alert rules, use the following sample configuration:

```
config
   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
      end
```

NOTES:

- **alerts rules group** *alert_group_name*: Specifiy the Prometheus alerting rules group. One alert group can have multiple lists of rules. *alert_group_name* is the name of the alert group as a string in the range of 0–64 characters.
- **interval-seconds** *seconds*: Specify the evaluation interval of the rule group in seconds.
- **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.

- **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* can be configured as 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.

Viewing Alert Logger

The Alert Logger stores all the generated alerts by default. You can view the stored alerts using the following **show** command.

show alert history [filtering]

You can narrow down the result using the following filtering options:

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

You can view the active and silenced alerts with the **show alerts active** and **show alerts active** commands.

Alarms

Rules are added at CEE as per the NRF alarms that requires the metrics provided by NRF and App-infra.

The following sections provide details of alarms that are supported by NRF.

Incoming TPS is greater than 50% of Max TPS

Severity	Description
Info	If Avg Incoming TPS for last 10mins is greater than 50% of Max TPS

Alert Rules

alerts rules group INCMSGTPS rule INCTPS50Perc duration 10m label name value INCTPS50PERC ;exit;severity warning expression "sum(irate(incoming_request_total{service_name=\"nrf-rest-ep\",protocol=\"http\"}[30s])) by (service_name, protocol) >= (0.5 * MAX_TPS)" type Quality\ Of\ Service\ Alarm annotation summary value "Incoming Messages TPS {{ printf \"%f\" $value }} for last 10min" **Note**: MAX_TPS depends on environment & is derived after performance evaluation

Incoming TPS is greater than 75% of Max TPS

Severity	Description
Minor	If Avg Incoming TPS for last 5mins is greater than 75% of Max TPS

Alert Rules
alerts rules group INCMSGTPS rule INCTPS75Perc duration 5m label name value INCTPS75PERC ;exit;severity minor expression "sum(irate(incoming_request_total{service_name=\"nrf-rest-ep\",protocol=\"http\"}[30s])) by (service_name, protocol) >= (0.75 * MAX_TPS)" type Quality\ Of\ Service\ Alarm annotation summary value "Incoming Messages TPS {{ printf \"%f\" $value }} for last 5min" **Note**: MAX_TPS depends on environment & is derived after performance evaluation

Incoming TPS is greater than 90% of Max TPS

Severity	Description
Major	If Avg Incoming TPS for last 1mins is greater than 90% of Max TPS

Alert Rules
alerts rules group INCMSGTPS rule INCTPS90Perc duration 1m label name value INCTPS90PERC ;exit;severity major expression "sum(irate(incoming_request_total{service_name=\"nrf-rest-ep\",protocol=\"http\"}[30s])) by (service_name, protocol) >= (0.9 * MAX_TPS)" type Quality\ Of\ Service\ Alarm annotation summary value "Incoming Messages TPS {{ printf \"%f\" $value }} for last 1min" **Note**: MAX_TPS depends on environment & is derived after performance evaluation

Incoming TPS is greater than 95% of Max TPS

Severity	Description
Critical	If Avg Incoming TPS for last 1mins is greater than 95% of Max TPS

Alert Rules
alerts rules group INCMSGTPS rule INCTPS95Perc duration 1m label name value INCTPS95PERC ;exit;severity critical expression "sum(irate(incoming_request_total{service_name=\"nrf-rest-ep\",protocol=\"http\"}[30s])) by (service_name, protocol) >= (0.95 * MAX_TPS)" type Quality\ Of\ Service\ Alarm annotation summary value "Incoming Messages TPS {{ printf \"%f\" $value }} for last 1min" **Note**: MAX_TPS depends on environment & is derived after performance evaluation

Error rate (per Incoming message type) is 1%

Severity	Description
Info	If Error rate (per Incoming message type) is 1%

Alert Rules	Description
Total Errors: alerts rules group INCMSGERR rule INCMSGERR1Perc duration 10m label name value INCMSGERR1PERC ;exit;severity warning expression "((sum(outgoing_response_total{service_nam e=\"nrf-rest-ep\",status=\"error\",protocol=\"http\"})) / (sum (incoming_request_total{service_name=\"n rf-rest-ep\",protocol=\"http\"}))) * 100 > 1.0" type Quality\ Of\ Service\ Alarm annotation summary value "Error Response (Incoming Messages) Rate {{ printf \"%f\" $value }} for last 10min" Per Message Type: alerts rules group INCMSGERR rule RegReqERR1Perc duration 10m label name value REGREQERR1PERC ;exit;severity warning expression "((sum(outgoing_response_msg_total{service _name=\"nrf-rest-ep\",status=\"error\",msg_type=\"NFRegistr ationRequest\"})) / (sum (incoming_request_msg_total{service_nam e=\"nrf-rest-ep\",msg_type=\"NFRegistrationRequest\"}))) * 100 > 1.0" type Quality\ Of\ Service\ Alarm annotation summary value "Error Response (NFRegistrationRequest) Rate {{ printf \"%f\" $value }} for last 10min"	Following are the Messages types to be used for per Message Type Error Rate. Make sure rule name & label is different for each rule of message type **NFDiscoveryRequest NFGetRequest NFRegistrationRequest NFUpdateRequest NFDeregistrationRequest NFCreateSubscriptionRequest NFRemoveSubscriptionRequest NFUpdateSubscriptionRequest** **Note:** Alert rule is based on the total errors & total incoming messages counters calculated till the point To define the rule for error rate calculated over a period use below expression in the alert rules: **Total Incoming Messages Error rate (calculated for last 30s):** "((sum(rate(outgoing_response_total{service_na me=\"nrf-rest-ep\",status=\"error\",protocol=\"http\"}[30s])) / (sum(rate(incoming_request_total{service_nam e=\"nrf-rest-ep\",protocol=\"http\"}[30s]))) * 100 > 1.0" **Incoming Messages (per Type) Error rate (calculated for last 30s):** "((sum(rate(outgoing_response_msg_total{servic e_name=\"nrf-rest-

Alert Rules	Description
	ep\",status=\"error\",msg_type=\"NFRegistratio nRequest\"}[30s])) / (sum(rate(incoming_request_msg_total{service _name=\"nrf-rest-ep\",msg_type=\"NFRegistrationRequest\"}[30s]))) * 100 > 1.0"

Error rate (per Incoming message type) is 10%

Severity	Description
Minor	If Error rate (per Incoming message type) is 10%

Alert Rules	Description
Total Errors: alerts rules group INCMSGERR rule INCMSGERR10Perc duration 5m label name value INCMSGERR10PERC ;exit;severity minor expression "((sum(outgoing_response_total{service_nam e=\"nrf-rest-ep\",status=\"error\",protocol=\"http\"})) / (sum (incoming_request_total{service_name=\"n rf-rest-ep\",protocol=\"http\"}))) * 100 > 10.0" type Quality\ Of\ Service\ Alarm annotation summary value "Error Response (Incoming Messages) Rate {{ printf \"%f\" $value }} for last 5min" Per Message Type: alerts rules group INCMSGERR rule RegReqERR10Perc duration 5m label name value REGREQERR10PERC ;exit;severity warning expression "((sum(outgoing_response_msg_total{service _name=\"nrf-rest-ep\",status=\"error\",msg_type=\"NFRegistr ationRequest\"})) / (sum (incoming_request_msg_total{service_nam e=\"nrf-rest-ep\",msg_type=\"NFRegistrationRequest\"}))) * 100 > 10.0" type Quality\ Of\ Service\ Alarm annotation summary value "Error Response (NFRegistrationRequest) Rate {{ printf \"%f\" $value }} for last 5min"	Following are the Messages types to be used for per Message Type Error Rate. Make sure rule name & label is different for each rule of message type **NFDiscoveryRequest NFGetRequest NFRegistrationRequest NFUpdateRequest NFDeregistrationRequest NFCreateSubscriptionRequest NFRemoveSubscriptionRequest NFUpdateSubscriptionRequest** **Note:** Alert rule is based on the total errors & total incoming messages counters calculated till the point To define the rule for error rate calculated over a period use below expression in the alert rules: **Total Incoming Messages Error rate (calculated for last 30s):** "((sum(rate(outgoing_response_total{service_na me=\"nrf-rest-ep\",status=\"error\",protocol=\"http\"}[30s])) / (sum(rate(incoming_request_total{service_nam e=\"nrf-rest-ep\",protocol=\"http\"}[30s]))) * 100 > 10.0" **Incoming Messages (per Type) Error rate (calculated for last 30s):** "((sum(rate(outgoing_response_msg_total{servic e_name=\"nrf-rest-ep\",status=\"error\",msg_type=\"NFRegistratio nRequest\"}[30s])) / (sum(rate(incoming_request_msg_total{service _name=\"nrf-rest-ep\",msg_type=\"NFRegistrationRequest\"}[30s]))) * 100 > 10.0"

Error rate (per Incoming message type) is 25%

Severity	Description
Major	If Error rate (per Incoming message type) is 25%

Alert Rules	Description
Total Errors: alerts rules group INCMSGERR rule INCMSGERR25Perc duration 5m label name value INCMSGERR25PERC ;exit;severity major expression "((sum(outgoing_response_total{service_nam e=\"nrf-rest-ep\",status=\"error\",protocol=\"http\"})) / (sum (incoming_request_total{service_name=\"n rf-rest-ep\",protocol=\"http\"}))) * 100 > 25.0" type Quality\ Of\ Service\ Alarm annotation summary value "Error Response (Incoming Messages) Rate {{ printf \"%f\" $value }} for last 5min" Per Message Type: alerts rules group INCMSGERR rule RegReqERR25Perc duration 5m label name value REGREQERR25PERC ;exit;severity warning expression "((sum(outgoing_response_msg_total{service _name=\"nrf-rest-ep\",status=\"error\",msg_type=\"NFRegistr ationRequest\"})) / (sum (incoming_request_msg_total{service_nam e=\"nrf-rest-ep\",msg_type=\"NFRegistrationRequest\"}))) * 100 > 25.0" type Quality\ Of\ Service\ Alarm annotation summary value "Error Response (NFRegistrationRequest)	Following are the Messages types to be used for per Message Type Error Rate. Make sure rule name & label is different for each rule of message type **NFDiscoveryRequest NFGetRequest NFRegistrationRequest NFUpdateRequest NFDeregistrationRequest NFCreateSubscriptionRequest NFRemoveSubscriptionRequest NFUpdateSubscriptionRequest** **Note:** Alert rule is based on the total errors & total incoming messages counters calculated till the point To define the rule for error rate calculated over a period use below expression in the alert rules: **Total Incoming Messages Error rate (calculated for last 30s):** "((sum(rate(outgoing_response_total{service_na me=\"nrf-rest-ep\",status=\"error\",protocol=\"http\"}[30s])) / (sum(rate(incoming_request_total{service_nam e=\"nrf-rest-ep\",protocol=\"http\"}[30s]))) * 100 > 25.0" **Incoming Messages (per Type) Error rate (calculated for last 30s):** "((sum(rate(outgoing_response_msg_total{servic

Alert Rules	Description
Rate {{ printf \"%f\" $value }} for last 5min"	e_name=\"nrf-rest-ep\",status=\"error\",msg_type=\"NFRegistratio nRequest\"}[30s])) / (sum(rate(incoming_request_msg_total{service _name=\"nrf-rest-ep\",msg_type=\"NFRegistrationRequest\"}[30s]))) * 100 > 25.0"

Error rate (per Incoming message type) is 50%

Severity	Description
Critical	If Error rate (per Incoming message type) is 50%

Alert Rules	Description
Total Errors: alerts rules group INCMSGERR rule INCMSGERR50Perc duration 5m label name value INCMSGERR50PERC ;exit;severity warning expression "((sum(outgoing_response_total{service_nam e=\"nrf-rest-ep\",status=\"error\",protocol=\"http\"})) / (sum (incoming_request_total{service_name=\"n rf-rest-ep\",protocol=\"http\"}))) * 100 > 50.0" type Quality\ Of\ Service\ Alarm annotation summary value "Error Response (Incoming Messages) Rate {{ printf \"%f\" $value }} for last 5min" Per Message Type: alerts rules group INCMSGERR rule RegReqERR50Perc duration 5m label name value REGREQERR50PERC ;exit;severity warning expression "((sum(outgoing_response_msg_total{service _name=\"nrf-rest-ep\",status=\"error\",msg_type=\"NFRegistr ationRequest\"})) / (sum (incoming_request_msg_total{service_nam e=\"nrf-rest-ep\",msg_type=\"NFRegistrationRequest\"}))) * 100 > 50.0" type Quality\ Of\ Service\ Alarm annotation summary value "Error Response (NFRegistrationRequest) Rate {{ printf \"%f\" $value }} for last 5min"	Following are the Messages types to be used for per Message Type Error Rate. Make sure rule name & label is different for each rule of message type **NFDiscoveryRequest NFGetRequest NFRegistrationRequest NFUpdateRequest NFDeregistrationRequest NFCreateSubscriptionRequest NFRemoveSubscriptionRequest NFUpdateSubscriptionRequest** **Note:** Alert rule is based on the total errors & total incoming messages counters calculated till the point To define the rule for error rate calculated over a period use below expression in the alert rules: **Total Incoming Messages Error rate (calculated for last 30s):** "((sum(rate(outgoing_response_total{service_na me=\"nrf-rest-ep\",status=\"error\",protocol=\"http\"}[30s]))) / (sum(rate(incoming_request_total{service_nam e=\"nrf-rest-ep\",protocol=\"http\"}[30s])))) * 100 > 50.0" **Incoming Messages (per Type) Error rate (calculated for last 30s):** "((sum(rate(outgoing_response_msg_total{servic e_name=\"nrf-rest-ep\",status=\"error\",msg_type=\"NFRegistratio nRequest\"}[30s]))) / (sum(rate(incoming_request_msg_total{service _name=\"nrf-rest-ep\",msg_type=\"NFRegistrationRequest\"}[30s])))) * 100 > 50.0"

Error rate (per outgoing message type) is 1%

Severity	Description
Info	If Error rate (per outgoing message type) is 1%

Alert Rules	Description
Total Errors: alerts rules group OUTMSGERR rule OUTMSGERR1Perc duration 10m label name value OUTMSGERR1PERC ;exit;severity warning expression "(sum(rpc_response_total{service_name=\"nr f-rest-ep\",interface=\"Rest\",status_code!~\"2[0- 9]{2}\"}) / sum(rpc_request_total{service_name=\"nrf-rest-ep\",interface=\"Rest\"})) * 100 > 1.0" type Quality\ Of\ Service\ Alarm annotation summary value "Error Response (Out going Messages) Rate {{ printf \"%f\" $value }} for last 10min" Per Message Type: alerts rules group OUTMSGERR rule StatNotifERR1Perc duration 10m label name value StatNotifERR1Perc ;exit;severity warning expression "(sum(rpc_response_total{service_name=\"nr f-rest-ep\",interface=\"Rest\",msg_type=\"NFStatus NotifyRequest\",status_code!~\"2[0-9]{2}\"}) / sum(rpc_request_total{service_name=\"nrf- rest-ep\",interface=\"Rest\",msg_type=\"NFStatus NotifyRequest\"})) * 100 > 1.0" type Quality\ Of\ Service\ Alarm annotation summary value "Error Response	Currently the supported outgoing message is NFStatusNotifyRequest. If more messages are going be supported, then those Messages types can be used for per Message Type Error Rate. Make sure rule name & label is different for each rule of message type **Note:** Alert rule is based on the total errors & total outgoing messages counters calculated till the point To define the rule for error rate calculated over a period use below expression in the alert rules: **Total Outgoing Messages Error rate (calculated for last 30s):** "(sum(rate(rpc_response_total{service_name=\"nrf-rest- ep\",interface=\"Rest\",status_code!~\"2[0-9]{2}\"}[30s])) / sum(rate(rpc_request_total{service_name=\"nrf-rest-ep\",interface=\"Rest\"}[30s]))) * 100 > 1.0" **Outgoing Messages (per Type) Error rate (calculated for last 30s):** "(sum(rate(rpc_response_total{service_name=\"nrf-rest-

Alert Rules	Description
(NFStatusNotifyRequest) Rate {{ printf \"%f\" $value }} for last 10min"	ep\",interface=\"Rest\",msg_type=\"NFStatus NotifyRequest\",status_code!~\"2[0- 9]{2}\"}[30s])) / sum(rate(rpc_request_total{service_name=\"n rf-rest-ep\",interface=\"Rest\",msg_type=\"NFStatus NotifyRequest\"}[30s]))) * 100 > 1.0"

Error rate (per outgoing message type) is 10%

Severity	Description
Minor	If Error rate (per outgoing message type) is 10%

Alert Rules	Description
Total Errors: alerts rules group OUTMSGERR rule OUTMSGERR10Perc duration 5m label name value OUTMSGERR10PERC ;exit;severity minor expression "((sum(outgoing_response_total{service_name =\"nrf-rest- ep\",status=\"error\",protocol=\"http\"})) / (sum (incoming_request_total{service_name=\"nrf-rest-ep\",protocol=\"http\"}))) * 100 > 10.0" type Quality\ Of\ Service\ Alarm annotation summary value "Error Response (Outgoing Messages) Rate {{ printf \"%f\" $value }} for last 5min" Per Message Type: alerts rules group OUTMSGERR rule StatNotifERR10Perc duration 5m label name value StatNotifERR10Perc ;exit;severity minor expression "(sum(rpc_response_total{service_name=\"nr f-rest-ep\",interface=\"Rest\",msg_type=\"NFStatus NotifyRequest\",status_code!~\"2[0-9]{2}\"}) / sum(rpc_request_total{service_name=\"nrf- rest-ep\",interface=\"Rest\",msg_type=\"NFStatus NotifyRequest\"})) * 100 > 10.0" type Quality\ Of\ Service\ Alarm annotation summary value "Error Response (NFStatusNotifyRequest) Rate {{ printf \"%f\" $value }} for last 5min"	Currently the supported outgoing message is NFStatusNotifyRequest. If more messages are going be supported, then those Messages types can be used for per Message Type Error Rate. Make sure rule name & label is different for each rule of message type **Note:** Alert rule is based on the total errors & total outgoing messages counters calculated till the point To define the rule for error rate calculated over a period use below expression in the alert rules: **Total Outgoing Messages Error rate (calculated for last 30s):** "(sum(rate(rpc_response_total{service_name=\"nrf-rest- ep\",interface=\"Rest\",status_code!~\"2[0-9]{2}\"}[30s])) / sum(rate(rpc_request_total{service_name=\"nrf-rest-ep\",interface=\"Rest\"}[30s]))) * 100 > 10.0" **Outgoing Messages (per Type) Error rate (calculated for last 30s):** "(sum(rate(rpc_response_total{service_name=\"nrf-rest-ep\",interface=\"Rest\",msg_type=\"NFStatus NotifyRequest\",status_code!~\"2[0- 9]{2}\"}[30s])) / sum(rate(rpc_request_total{service_name=\"n rf-rest-ep\",interface=\"Rest\",msg_type=\"NFStatus NotifyRequest\"}[30s]))) * 100 > 10.0"

Error rate (per outgoing message type) is 25%

Severity	Description
Major	If Error rate (per outgoing message type) is 25%

Alert Rules	Description
Total Errors: alerts rules group OUTMSGERR rule OUTMSGERR25Perc duration 5m label name value OUTMSGERR25PERC ;exit;severity major expression "((sum(outgoing_response_total{service_name =\"nrf-rest- ep\",status=\"error\",protocol=\"http\"})) / (sum (incoming_request_total{service_name=\"nrf-rest-ep\",protocol=\"http\"}))) * 100 > 25.0" type Quality\ Of\ Service\ Alarm annotation summary value "Error Response (Outgoing Messages) Rate {{ printf \"%f\" $value }} for last 5min" Per Message Type: alerts rules group OUTMSGERR rule StatNotifERR25Perc duration 5m label name value StatNotifERR25Perc ;exit;severity major expression "(sum(rpc_response_total{service_name=\"nr f-rest-ep\",interface=\"Rest\",msg_type=\"NFStatus NotifyRequest\",status_code!~\"2[0-9]{2}\"}) / sum(rpc_request_total{service_name=\"nrf- rest-ep\",interface=\"Rest\",msg_type=\"NFStatus NotifyRequest\"})) * 100 > 25.0" type Quality\ Of\ Service\ Alarm annotation summary value "Error Response	Currently the supported outgoing message is NFStatusNotifyRequest. If more messages are going be supported, then those Messages types can be used for per Message Type Error Rate. Make sure rule name & label is different for each rule of message type **Note:** Alert rule is based on the total errors & total outgoing messages counters calculated till the point To define the rule for error rate calculated over a period use below expression in the alert rules: **Total Outgoing Messages Error rate (calculated for last 30s):** "(sum(rate(rpc_response_total{service_name=\ "nrf-rest- ep\",interface=\"Rest\",status_code!~\"2[0-9]{2}\"}[30s])) / sum(rate(rpc_request_total{service_name=\"n rf-rest-ep\",interface=\"Rest\"}[30s]))) * 100 > 25.0" **Outgoing Messages (per Type) Error rate (calculated for last 30s):** "(sum(rate(rpc_response_total{service_name=\ "nrf-rest- ep\",interface=\"Rest\",msg_type=\"NFStatus

Alert Rules	Description
(NFStatusNotifyRequest) Rate {{ printf \"%f\" $value }} for last 5min"	NotifyRequest\",status_code!~\"2[0- 9]{2}\"}[30s])) / sum(rate(rpc_request_total{service_name=\"n rf-rest-ep\",interface=\"Rest\",msg_type=\"NFStatus NotifyRequest\"}[30s]))) * 100 > 25.0"

Error rate (per outgoing message type) is 50%

Severity	Description
Critical	If Error rate (per outgoing message type) is 50%

Alert Rules	Description
Total Errors: alerts rules group OUTMSGERR rule OUTMSGERR50Perc duration 5m label name value OUTMSGERR50PERC ;exit;severity critical expression "((sum(outgoing_response_total{service_name =\"nrf-rest- ep\",status=\"error\",protocol=\"http\"})) / (sum (incoming_request_total{service_name=\"nrf-rest-ep\",protocol=\"http\"}))) * 100 > 50.0" type Quality\ Of\ Service\ Alarm annotation summary value "Error Response (Outgoing Messages) Rate {{ printf \"%f\" $value }} for last 5min" Per Message Type: alerts rules group OUTMSGERR rule StatNotifERR50Perc duration 5m label name value StatNotifERR50Perc ;exit;severity critical expression "(sum(rpc_response_total{service_name=\"nr f-rest-ep\",interface=\"Rest\",msg_type=\"NFStatus NotifyRequest\",status_code!~\"2[0-9]{2}\"}) / sum(rpc_request_total{service_name=\"nrf- rest-ep\",interface=\"Rest\",msg_type=\"NFStatus NotifyRequest\"})) * 100 > 50.0" type Quality\ Of\ Service\ Alarm annotation summary value "Error Response (NFStatusNotifyRequest) Rate {{ printf \"%f\" $value }} for last 5min"	Currently the supported outgoing message is NFStatusNotifyRequest. If more messages are going be supported, then those Messages types can be used for per Message Type Error Rate. Make sure rule name & label is different for each rule of message type **Note:** Alert rule is based on the total errors & total outgoing messages counters calculated till the point To define the rule for error rate calculated over a period use below expression in the alert rules: **Total Outgoing Messages Error rate (calculated for last 30s):** "(sum(rate(rpc_response_total{service_name=\"nrf-rest- ep\",interface=\"Rest\",status_code!~\"2[0-9]{2}\"}[30s])) / sum(rate(rpc_request_total{service_name=\"nrf-rest-ep\",interface=\"Rest\"}[30s]))) * 100 > 50.0" **Outgoing Messages (per Type) Error rate (calculated for last 30s):** "(sum(rate(rpc_response_total{service_name=\"nrf-rest-ep\",interface=\"Rest\",msg_type=\"NFStatus NotifyRequest\",status_code!~\"2[0- 9]{2}\"}[30s])) / sum(rate(rpc_request_total{service_name=\"n rf-rest-ep\",interface=\"Rest\",msg_type=\"NFStatus NotifyRequest\"}[30s]))) * 100 > 50.0"

CPU usage is greater than 50%

Severity	Description
Info	If Avg CPU usage for last 10mins is greater than 50%

Alert Rules	Description
alerts rules group CPUUSG rule NrfRestEp0CPUUSG50Perc duration 10m label name value NRFRESTEP0_CPUUSG50PERC ;exit;severity warning expression "cpu_percent{service_name=\"nrf-rest-ep\",instance_id=\"0\"} > 50.0" type Quality\ Of\ Service\ Alarm annotation summary value "CPU Usage {{ printf \"%f\" $value }} for last 10min" **Note**: CPU Usage alert is for each POD i.e. resources are per POD level. for NRF Service pod=> Service name is nrf- service Make sure rule name & labels are different for each pod	**POD Name & Instance Id mapping:** **1.** NRF pods have naming convention of <service-name>-n<node-id>-<replica-id>, for example,: nrf-service-n0-0, nrf-rest-ep-n0-1, nrf-rest-ep-n2-1 etc **1.** InstanceId is sum of <node-id> + <replica-id> e.g: nrf-service-n0-0 => InstanceId = 0 + 0 = 0 nrf-service-n0-1 => InstanceId = 0 + 1 = 1 nrf-service-n2-0 => InstanceId = 2 + 0 = 2 **Note**: Node-Ids increment with a period of number of replicas i.e. if replicas for a deployment is 2 & number of nodes are 2 then nodes-ids are 0, 2

CPU usage is greater than 75%

Severity	Description
Minor	If Avg CPU usage for last 5mins is greater than 75%

Alert Rules	Description
alerts rules group CPUUSG rule NrfRestEp0CPUUSG75Perc duration 5m label name value NRFRESTEP0_CPUUSG75PERC ;exit;severity minor expression "cpu_percent{service_name=\"nrf-rest-ep\",instance_id=\"0\"} > 75.0" type Quality\ Of\ Service\ Alarm annotation summary value "CPU Usage {{ printf \"%f\" $value }} for last 5min" **Note**: CPU Usage alert is for each POD i.e. resources are per POD level. for NRF Service pod=> Service name is nrf- service Make sure rule name & labels are different for each pod	**POD Name & Instance Id mapping:** **1.** NRF pods have naming convention of <service-name>-n<node-id>-<replica-id>, for example,: nrf-service-n0-0, nrf-rest-ep-n0-1, nrf-rest-ep-n2-1 etc **1.** InstanceId is sum of <node-id> + <replica- id> e.g: nrf-service-n0-0 => InstanceId = 0 + 0 = 0 nrf-service-n0-1 => InstanceId = 0 + 1 = 1 nrf-service-n2-0 => InstanceId = 2 + 0 = 2 **Note**: Node-Ids increment with a period of number of replicas i.e. if replicas for a deployment is 2 & number of nodes are 2 then nodes-ids are 0, 2

CPU usage is greater than 90%

Severity	Description
Major	If Avg CPU usage for last 1mins is greater than 90%

Alert Rules	Description
alerts rules group CPUUSG rule NrfRestEp0CPUUSG90Perc duration 1m label name value NRFRESTEP0_CPUUSG90PERC ;exit;severity major expression "cpu_percent{service_name=\"nrf-rest-ep\",instance_id=\"0\"} > 90.0" type Quality\ Of\ Service\ Alarm annotation summary value "CPU Usage {{ printf \"%f\" $value }} for last 1min" **Note**: CPU Usage alert is for each POD i.e. resources are per POD level. for NRF Service pod=> Service name is nrf- service Make sure rule name & labels are different for each pod	**POD Name & Instance Id mapping:** **1.** NRF pods have naming convention of <service-name>-n<node-id>-<replica-id>, for example,: nrf-service-n0-0, nrf-rest-ep-n0-1, nrf-rest-ep-n2-1 etc **1.** InstanceId is sum of <node-id> + <replica- id> e.g: nrf-service-n0-0 => InstanceId = 0 + 0 = 0 nrf-service-n0-1 => InstanceId = 0 + 1 = 1 nrf-service-n2-0 => InstanceId = 2 + 0 = 2 **Note**: Node-Ids increment with a period of number of replicas i.e. if replicas for a deployment is 2 & number of nodes are 2 then nodes-ids are 0, 2

CPU usage is greater than 95%

Severity	Description
Critical	If Avg CPU usage for last 1mins is greater than 95%

Alert Rules	Description
alerts rules group CPUUSG rule NrfRestEp0CPUUSG95Perc duration 1m label name value NRFRESTEP0_CPUUSG95PERC ;exit;severity critical expression "cpu_percent{service_name=\"nrf-rest-ep\",instance_id=\"0\"} > 95.0" type Quality\ Of\ Service\ Alarm annotation summary value "CPU Usage {{ printf \"%f\" $value }} for last 1min" **Note**: CPU Usage alert is for each POD i.e. resources are per POD level. for NRF Service pod=> Service name is nrf- service Make sure rule name & labels are different for each pod	**POD Name & Instance Id mapping:** **1.** NRF pods have naming convention of <service-name>-n<node-id>-<replica-id>, for example,: nrf-service-n0-0, nrf-rest-ep-n0-1, nrf-rest-ep-n2-1 etc **1.** InstanceId is sum of <node-id> + <replica- id> e.g: nrf-service-n0-0 => InstanceId = 0 + 0 = 0 nrf-service-n0-1 => InstanceId = 0 + 1 = 1 nrf-service-n2-0 => InstanceId = 2 + 0 = 2 **Note**: Node-Ids increment with a period of number of replicas, that is, if replicas for a deployment is 2 & number of nodes are 2 then nodes-ids are 0, 2

Memory usage is greater than 50% of Memory Limit

Severity	Description
Info	If Memory usage for last 10mins is greater than 50% of Memory Limit

Alert Rules	Description
alerts rules group MEMUSG rule NrfRestEp0MEMUSG50Perc duration 10m label name value NRFRESTEP0_MEMUSG50PERC ;exit;severity warning expression "((mem_usage_kb{service_name=\"nrf-rest-ep\",instance_id=\"0\"}/1024)/MEMORY_LIMIT_KB) * 100 > 50.0" type Quality\ Of\ Service\ Alarm annotation summary value "Memory Usage {{ printf \"%f\" $value }} for last 10min" **Note**: Memory Usage alert is for each POD, that is, resources are per POD level. for NRF Service pod=> Service name is nrf-service Make sure rule name & labels are different for each pod	At present, no Memory Limit is given for NRF PODs i.e. there is no limit and it depends on the available memory at the worker node at run time. E.g. If worker node has 1GB memory and 20% is used for its own functionality, then 80% of Memory ia available for the PODs deployed on worker node. In case of no memory limt, if 1 POD is deployed, then it can use complete memory. For alerts case, you can provide the MEMORY_LIMIT_KB depends on the environment, that is available memory for the NRF POD on that host

Memory usage is greater than 75% of Memory Limit

Severity	Description
Minor	If Memory usage for last 5mins is greater than 75% of Memory Limit

Alert Rules
alerts rules group MEMUSGUSG rule NrfRestEp0MEMUSG75Perc duration 5m label name value NRFRESTEP0_MEMUSG75PERC ;exit;severity minor expression "((mem_usage_kb{service_name=\"nrf-rest- ep\",instance_id=\"0\"}/1024)/MEMORY_LIMIT_KB) * 100 > 75.0" type Quality\ Of\ Service\ Alarm annotation summary value "Memory Usage {{ printf \"%f\" $value }} for last 5min" **Note**: Memory Usage alert is for each POD, that is, resources are per POD level.

Alert Rules
for NRF Service pod=> Service name is nrf-service Make sure rule name & labels are different for each pod

Memory usage is greater than 90% of provided Memory

Severity	Description
Major	If Memory usage for last 1mins is greater than 90% of provided Memory

Alert Rules
alerts rules group MEMUSG rule NrfRestEp0MEMUSG90Perc duration 1m label name value NRFRESTEP0_MEMUSG90PERC ;exit;severity major expression "((mem_usage_kb{service_name=\"nrf-rest- ep\",instance_id=\"0\"}/1024)/MEMORY_LIMIT_KB) * 100 > 90.0" type Quality\ Of\ Service\ Alarm annotation summary value "Memory Usage {{ printf \"%f\" $value }} for last 1min" **Note**: Memory Usage alert is for each POD, that is, resources are per POD level. for NRF Service pod=> Service name is nrf-service Make sure rule name & labels are different for each pod

Memory usage is greater than 95% of provided Memory

Severity	Description
Critical	If Memory usage for last 1mins is greater than 95% of provided Memory

Alert Rules
alerts rules group MEMUSG rule NrfRestEp0MEMUSG95Perc duration 1m label name value NRFRESTEP0_MEMUSG95PERC ;exit;severity critical expression

Alert Rules
"((mem_usage_kb{service_name=\"nrf-rest- ep\",instance_id=\"0\"}/1024)/MEMORY_LIMIT_KB) * 100 > 95.0" type Quality\ Of\ Service\ Alarm annotation summary value "Memory Usage {{ printf \"%f\" $value }} for last 1min" **Note**: Memory Usage alert is for each POD, that is, resources are per POD level. for NRF Service pod=> Service name is nrf-service Make sure rule name & labels are different for each pod

NF Profiles count reach 50% of CDL capacity

Severity	Description
Info	If NF profiles is greater than 50%, it is an indication of growing number of profiles

Alert Rules
alerts rules group NFPROFCNT rule NFPROFCNT50Perc duration 10m label name value NFPROFCNT50PERC ;exit;severity warning expression "sum(avg(nrf_profiles_total{service_name=\"nrf-service\"}) by (nf_type)) >= (0.5 * MAX_NF_PROF_CNT)" type Quality\ Of\ Service\ Alarm annotation summary value "Number of NF Profiles {{ printf \"%f\" $value }} for last 10min" **Note**: MAX_NF_PROF_CNT depends on environment, that is, Maximum CDL capacity

NF Profiles count reach 85% of CDL capacity

Severity	Description
Minor	If NF profiles is greater than 85%, it is a minor fault of growing number of profiles.

Alert Rules
alerts rules group NFPROFCNT rule NFPROFCNT85Perc duration 5m label name value NFPROFCNT85PERC ;exit;severity minor expression "sum(avg(nrf_profiles_total{service_name=\"nrf-service\"}) by (nf_type)) >= (0.85 * MAX_NF_PROF_CNT)" type Quality\ Of\ Service\ Alarm annotation summary value "Number of NF Profiles {{ printf \"%f\" $value }} for last 5min" **Note**: MAX_NF_PROF_CNT depends on environment, that is, Maximum CDL capacity

NF Profiles count reach 90% of CDL capacity

Severity	Description
Major	If NF profiles is greater than 90%, a major fault is required to look into the deployment for further actions, for example, scaling

Alert Rules

alerts rules group NFPROFCNT rule NFPROFCNT90Perc duration 3m label name value NFPROFCNT90PERC ;exit;severity major expression "sum(avg(nrf_profiles_total{service_name=\"nrf-service\"}) by (nf_type)) >= (0.9 * MAX_NF_PROF_CNT)" type Quality\ Of\ Service\ Alarm annotation summary value "Number of NF Profiles {{ printf \"%f\" $value }} for last 3min" **Note**: MAX_NF_PROF_CNT depends on environment, that is, Maximum CDL capacity

NF Profiles count reach 95% of CDL capacity

Severity	Description
Critical	If NF profiles is greater than 95%, a critical fault is required to look into the deployment for further actions, for example, scaling

Alert Rules
alerts rules group NFPROFCNT rule NFPROFCNT95Perc duration 1m label name value NFPROFCNT95PERC ;exit;severity critical expression "sum(avg(nrf_profiles_total{service_name=\"nrf-service\"}) by (nf_type)) >= (0.95 * MAX_NF_PROF_CNT)" type Quality\ Of\ Service\ Alarm annotation summary value "Number of NF Profiles {{ printf \"%f\" $value }} for last 1min" **Note**: MAX_NF_PROF_CNT depends on environment, that is, Maximum CDL capacity

POD connectivity or Status failure

Severity	Description
Critical	If any inter POD connectivity is failed, for example, rest-ep to service, service to cdl etc.

Alert Rules
alerts rules group SERVICE_DOWN rule XXXSERVICE_DOWN1 duration 1s label name value "SERVICE_DOWN" ;exit;severity critical expression "sum(endpoint_status{ep_name=~\"internal-ipc.*ep\"}) by (service_name) == 0" type Quality\ Of\ Service\ Alarm annotation summary value "{{ $labels.service_name }} is DOWN" **Note**: Alert will be raised if any service is down and unable to serve any requests from other PODS. It is based on the POD internal rpc status. Alert will be raised for each service required for NRF, that is, **nrf-service, nrf-rest-ep, cache-pod, datastore-ep, datastore-index, datastore-slot, oam-pod**