Ultra Cloud Core 5G Session Management Function Statistics Reference, Release 2026.02

First Published: 2026-04-23

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

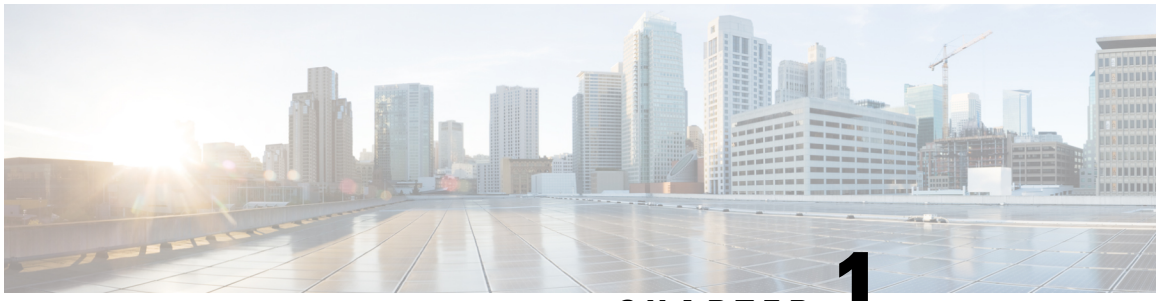
CONTENTS

About this Guide

Note The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. While any existing biased terms are being substituted, exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

This guide describes the metrics supported by 5G Session Management Function (SMF). This guide also provides information on how to gather the statistics or counters from its microservices.

About this Guide



CHAPTER 1

SMF Interface for Metrics

Feature Description

You can a monitor wide range of application and system statistics, and key performance indicators (KPI) within the SMF infrastructure. KPIs are useful to gain insight into the overall health of the SMF environment. Statistics offer a simplified representation of the SMF configurations and utilization-specific data.

The SMF integrates with Prometheus, a third-party monitoring and alerting solution to capture and preserve the performance data. This data is reported as statistics and can be viewed in the web-based dashboard. Grafana provides a graphical or text-based representation of statistics and counters, which the Prometheus database collects. The Grafana dashboard projects a comprehensive set of quantitative and qualitative data that encourages you to analyze SMF metrics in the reporting tool of your choice and take informed decisions.

By default, the monitoring solution is enabled, which indicates that Prometheus continually monitors your SMF environment and the Prometheus data source is associated with Grafana. You must have the administrative privileges to access Grafana. However, to view a specific dashboard, run the Prometheus queries. The queries are available in the built-in and custom format.

The following snapshot is a sample of the Grafana dashboard.

Figure 1: Grafana Dashboard

How it Works

KPIs constitute of metrics, such as statistics and counters. These metrics represent the performance improvement or degradation. By default, Prometheus is enabled on the system where SMF is deployed, and configured with Grafana. Prometheus dynamically starts monitoring the data sources that are available on the system. For new dashboard panels, execute queries in Prometheus.

For more information about Prometheus, consult the Prometheus documentation at https://prometheus.io/docs/introduction/overview/.

Configuring Metrics Collection

The labels of each SMF metrics are classified into the following three categories:

- Production
- Debug
- Granular

All the SMF application metrics are controlled through the CLI command for performance optimization.

To collect the necessary SMF metrics and labels, use the following sample configuration:

```
config
  infra metrics verbose { service | protocol | load-balancer | application
 } [ level { debug | off | production | trace } | metrics metrics_name [
granular-labels label_name| level { debug | off | production | trace } |
pod pod_name | level { debug | off | production | trace }  ] ]
   end
```

NOTES:

- If the metrics verbosity is not configured, then the default verbosity level for pod type is as follows.
  - LoadBalancer = Production
  - Protocol = Trace
  - Service = Trace
  - Application = Debug
- The order of the level for verbose metrics is in the following priority order:
  - **metrics [[metrics_name] level [production|debug|trace|off]**: [Priority 1]
  - **pod [[pod_Name]] level [production | debug | trace | off]]** [Priority 2]
  - **level [production | debug | trace | off]** [Priority 3]
- **infra metrics verbose { service | protocol | load-balancer | application }**: Enable the metric collection. This configuration helps to collect the required application metrics and labels. By default, this command captures the debug labels of metrics.
- **level { debug | off | production | trace }**: Specify the application metrics category to capture the required application metrics and labels.
  - **debug**: Capture all the labels that are classified as production and debug categories. This option is the default configuration.
  - **off**: Disable the application level metrics collection.

    For example, configuring the **infra metrics verbose application smf_service_stats level off** command disables the smf_service_stats application metrics.
  - **production**: Capture the labels that are classified as production category.
  - **trace**: This option is not supported for SMF application metrics. If this option is configured, the SMF treats this option as **debug**.
- If production and debug classification is empty for a metrics, then all the labels except granular-labels (if configured) are classified as debug.
- **metrics** *metrics_name*: Specify the metrics name to capture only the labels that correspond to the given metrics. The metric-level configuration takes precedence over the application-level configuration. If the metrics level is not configured, the labels are captured at the application level.
- **granular-labels**: Capture only the granular labels. By default, this option is disabled.

  If a granular label is required for KPI, then that label must be configured. For example, to capture dnn labels of smf_service_stats metrics, you must configure the following CLI command:

```
infra metrics verbose application metrics smf_service_stats level debug
 granular-labels [ dnn ]
```

Configuration Example

The following is an example configuration to enable only production level for all the application metrics.

```
infra metrics verbose application level production
```

The following is an example configuration to enable production level for smf_service_stats application metrics and debug level for all other application metrics.

```
infra metrics verbose application smf_service_stats level production
```

The following is an example configuration to enable debug level for smf_service_stats application metrics along with granular labels and production level for all other application metrics.

```
infra metrics verbose application level production smf_service_stats level
 debug granular-labels [ dnn ]
```

The following is an example configuration to enable production level for smf_service_stats application metrics along with granular labels and debug level for all other application metrics.

```
infra metrics verbose application smf_service_stats level production
granular-labels [ dnn ]
```

The following is an example configuration to disable smf_service_stats application metrics and debug level for all other application metrics.

```
infra metrics verbose application smf_service_stats level off
```

The following is an example configuration to configure NSSAI labels of smf_service_stats metrics.

```
infra metrics verbose application metrics smf_service_stats level debug
granular-labels [ snssai ]
```

Note The NSSAI statistics are not pegged without configuring the NSSAI label in the granular-labels configuration.

Configuration Verification

To verify the configuration, use the following show command:

```
show running-config infra metrics verbose application
```

The following are example outputs of the **show running-config infra metrics verbose application** command.

```
[smf] smf# show running-config infra metrics verbose application
infra metrics verbose application
 metrics smf_service_stats
  level production
  granular-labels [ dnn ]
 exit
exit
```

The preceding output indicates that the configuration to capture production labels for smf_service_stats application metrics along with granular labels and debug levels of all other application metrics is enabled.

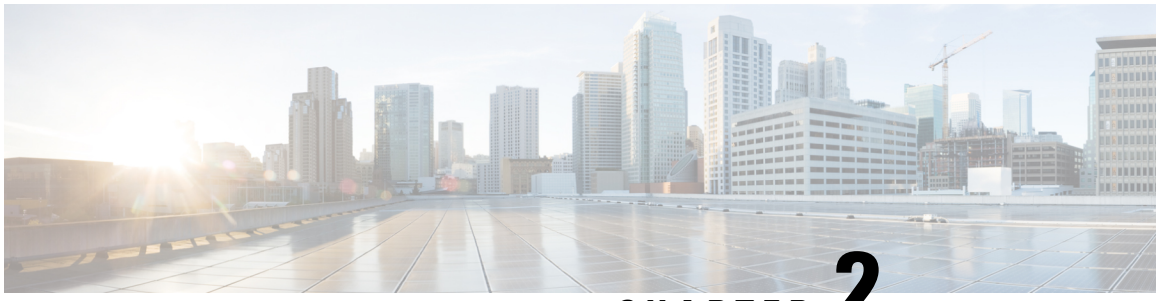
```
[smf] smf# show running-config infra metrics verbose application
 infra metrics verbose application
  level production
  metrics smf_service_stats
```

```
   level debug
   granular-labels [ [dnn] ]
  exit
exit
```

The preceding output indicates that the configuration to capture debug labels for smf_service_stats application metrics along with granular labels and production level of all other application metrics is enabled.

To verify the slice information on procedure and session statistics, use the following show command:

```
show running-config infra metrics verbose application
infra metrics verbose application
 metrics smf_service_stats
  level debug
  granular-labels [ snssai ]
 exit
```



CHAPTER 2

SMF Metrics

chf-service Metrics Reference

CCF Data Consistency Check Category

ccf_datacheck_stats

Description: Total number of sessions checked for consistency

Sample Query: `'ccf_datacheck_stats{rat_type="NR", status="failed"}'`

Labels:

- Label: `procedure_type`

  Label Description: Procedure Name

  Example: N40 Charging Data Request Create, N40 Charging Data Request Release SMF Initiated, N40 Session Modify SMF Initiated, N40 Session Notify - Abort,N40 Session Notify - Reauth, N28 Spending Limit Subscribe, N28 Spending Limit Subscribe Update, N28 Spending Limit UnSubscribe, N28 Spending Limit Notify Status, N28 Spending Limit Terminate Status, N28 Next Reval Notify, N40 Next Reval Notify

- Label: `rat_type`

  Label Description: Type of the radio access associated

  Example: EUTRA, NR, WLAN, VIRTUAL, rat_type_unknown

- Label: `pdu_type`

  Label Description: Type of PDU session

  Example: ipv4, ipv6, ipv4v6, unknown

- Label: `status`

  Label Description: Procedure status after data consistency check

  Example: success, failed

- Label: `reason`

  Label Description: Failure reason of data inconsistency

  Example: String format for failed reason

CCF Procedure Category

ccf_service_stats

Description: CCF call flow procedure counters

Sample Query: `'ccf_service_stats{procedure_type="pdu_sess_create"}'`

Labels:

- Label: `procedure_type`

  Label Description: The procedure type associated with an call flow procedure

  Example: pdu_sess_create, smf_req_pdu_sess_mod, smf_req_pdu_sess_rel, pdu_sess_notify_abort, pdu_sess_notify_reauth, spending_limit_notify_terminate, spending_limit_notify_status, spending_limit_create, spending_limit_update, spending_limit_delete

- Label: `status`

  Label Description: call flow procedure counter

  Example: attempted, success, failures

- Label: `dnn`

  Label Description: Dnn configured in dnn-policy, also can have virtual_dnn if configured, separated by #

  Example: intershat, intershat#cisco.com

- Label: `reason`

  Label Description: Reason for failure status. For sucessess and attempted it will be Empty

  Example: proc_pdu_not_established, proc_pdu_ctx_not_found, internal_error, reason_unknown, pdn_create_over_created_pdn, auth_grpc_failed, maintenance_mode, quota_grpc_failed, auth_failed, cc_relay_failed

- Label: `rat_type`

  Label Description: RAT Type of the Session

  Example: EUTRA, NR, WLAN, rat_type_unknown

- Label: `roaming_status`

  Label Description: Roaming status of the subscriber session

  Example: IN_BOUND, OUT_BOUND, none

- Label: `ccf_current_procedure`

  Label Description: Current Procedure Name for Message Level Stats

  Example:

CCF Procedure Collision Category

ccf_procedure_collision

Description: Total number of procedures collided

Sample Query: `sum(ccf_procedure_collision) by (ccf_current_procedure, ccf_current_state, ccf_new_procedure, ccf_current_procedure_action)`

Labels:

- Label: `ccf_current_procedure`

  Label Description: Current Procedure Name

  Example: PDU Session Release - SMF initiated, PDU 5G to 4G Handover, PDU Session Modify - PCF initiated, PDU UE Sync Procedure, PDU Idle Mode Entry - RAN initiated

- Label: `ccf_current_state`

  Label Description: Current Procedure State

  Example: String format for procedure stat

- Label: `ccf_new_procedure`

  Label Description: New Procedure Name

  Example: String format for new procedure

- Label: `ccf_current_procedure_action`

  Label Description: Current Procedure Action on Collision

  Example: Ignore, Suspend, Resume, Abort, Cleanup, Continue, Ready, INVALID ACTION

CCF Procedure Total Time Statistics Category

ccf_procedure_seconds

Description: Total number of seconds taken to complete the procedure

Sample Query: `'ccf_procedure_seconds{ccf_proc_status="Aborted"}'`

Labels:

- Label: `ccf_proc_type`

  Label Description: Procedure Name

  Example: N40 Charging Data Request Create, N40 Charging Data Request Release SMF Initiated, N40 Session Modify SMF Initiated, N40 Session Notify - Abort,N40 Session Notify - Reauth, N28 Spending Limit Subscribe, N28 Spending Limit Subscribe Update, N28 Spending Limit UnSubscribe, N28 Spending Limit Notify Status, N28 Spending Limit Terminate Status, N28 Next Reval Notify, N40 Next Reval Notify

- Label: `ccf_proc_status`

  Label Description: Procedure Status

  Example: Queued, Running, Aborted, Suspended, Invalid, Cleanedup, RequireSuspend, RequireCleanup, RequireAbort, ProcStatusComplete, Unknown

CCF Start Procedure Statistics Category

ccf_procedure_start

Description: Total number of procedures started

Sample Query: `'ccf_procedure_start{ccf_proc_type="PDN Connect"}'`

Labels:

- Label: `ccf_proc_type`

  Label Description: Procedure Name

  Example: String format for procedure stat

CCF Stop Procedure Statistics Category

ccf_procedure_stop

Description: Total number of procedures stopped

Sample Query: `'ccf_procedure_stop{ccf_proc_type="PDU Session Establishment"}'`

Labels:

- Label: `ccf_proc_type`

  Label Description: Procedure Name

  Example: String format for procedure stat

- Label: `ccf_proc_status`

  Label Description: Procedure Status

Example: Queued, Running, Aborted, Suspended, Invalid, Cleanedup, RequireSuspend, RequireCleanup, RequireAbort, ProcStatusComplete

CCF Total Procedure Count Category

ccf_procedure_total

Description: Total number of procedures executed

Sample Query: `'ccf_procedure_total{ccf_proc_status="Running"}'`

Labels:

- Label: `ccf_proc_type`

  Label Description: Procedure Name

  Example: N40 Charging Data Request Create, N40 Charging Data Request Release SMF Initiated, N40 Session Modify SMF Initiated, N40 Session Notify - Abort,N40 Session Notify - Reauth, N28 Spending Limit Subscribe, N28 Spending Limit Subscribe Update, N28 Spending Limit UnSubscribe, N28 Spending Limit Notify Status, N28 Spending Limit Terminate Status, N28 Next Reval Notify, N40 Next Reval Notify

- Label: `ccf_proc_status`

  Label Description: Procedure Status

  Example: Queued, Running, Aborted, Suspended, Invalid, Cleanedup, RequireSuspend, RequireCleanup, RequireAbort, ProcStatusComplete, Unknown

CCF Total Timedout Procedure Count Category

ccf_procedure_timeout

Description: Total number of procedures executed more than 10sec

Sample Query: `'ccf_procedure_timeout{ccf_proc_status="Running"}'`

Labels:

- Label: `ccf_proc_type`

  Label Description: Procedure Name

  Example: N40 Charging Data Request Create, N40 Charging Data Request Release SMF Initiated, N40 Session Modify SMF Initiated, N40 Session Notify - Abort,N40 Session Notify - Reauth, N28 Spending Limit Subscribe, N28 Spending Limit Subscribe Update, N28 Spending Limit UnSubscribe, N28 Spending Limit Notify Status, N28 Spending Limit Terminate Status, N28 Next Reval Notify, N40 Next Reval Notify

- Label: `ccf_proc_status`

  Label Description: Procedure Status

  Example: Queued, Running, Aborted, Suspended, Invalid, Cleanedup, RequireSuspend, RequireCleanup, RequireAbort, ProcStatusComplete, Unknown

CCF Total Timedout Procedure Time Category

ccf_procedure_timeout_seconds

Description: Total number of seconds taken by procedures executed more than 10sec

Sample Query: `'ccf_procedure_timeout_seconds{ccf_proc_status="Running"}'`

Labels:

- Label: `ccf_proc_type`

  Label Description: Procedure Name

  Example: N40 Charging Data Request Create, N40 Charging Data Request Release SMF Initiated, N40 Session Modify SMF Initiated, N40 Session Notify - Abort,N40 Session Notify - Reauth, N28 Spending Limit Subscribe, N28 Spending Limit Subscribe Update, N28 Spending Limit UnSubscribe, N28 Spending Limit Notify Status, N28 Spending Limit Terminate Status, N28 Next Reval Notify, N40 Next Reval Notify

- Label: `ccf_proc_status`

  Label Description: Procedure Status

  Example: Queued, Running, Aborted, Suspended, Invalid, Cleanedup, RequireSuspend, RequireCleanup, RequireAbort, ProcStatusComplete, Unknown

CCF Total Unhandled Event Statistics Category

ccf_procedure_unhndl_event

Description: Total number of unhandled events per procedure type

Sample Query: `'ccf_procedure_unhndl_event{ccf_proc_type="PDU Session Release - SMF initiated"}'`

Labels:

- Label: `ccf_proc_type`

  Label Description: Procedure Name

  Example: String format for procedure stat

- Label: `message_type`

  Label Description: Type of Request/Response Message associated with Unhandled Event

  Example: String format for event details

- Label: `ccf_current_state`

  Label Description: Current Procedure State

  Example: String format for procedure state

- Label: `guard_timer`

  Label Description: This is a check for Guard Timeout. TRUE if Guard Timer has expired, else FALSE

  Example: TRUE, FALSE

CCF Total Unhandled Transaction Statistics Category

ccf_procedure_unhndl_trans

Description: Total number of unhandled transactions per procedure type

Sample Query: `'xxf_procedure_unhndl_trans{message_type="RadiusCoaDisconnectReq"}'`

Labels:

- Label: `ccf_proc_type`

  Label Description: Procedure Name

  Example: PDU Session Release - SMF initiated, PDU 5G to 4G Handover, PDU Session Modify - PCF initiated, PDU UE Sync Procedure, PDU Idle Mode Entry - RAN initiated

- Label: `message_type`

  Label Description: Type of Request/Response Message associated with Unhandled Transaction

  Example: String format for event details

- Label: `ccf_current_state`

  Label Description: Current Procedure State

  Example: String format for procedure state

- Label: `guard_timer`

  Label Description: This is a check for Guard Timeout. TRUE if Guard Timer has expired, else FALSE

  Example: TRUE, FALSE

CCF Usage Report Stats Category

ccf_usage_trigger_stats

Description: The current count for Used unit Container Recieved from SMF

Sample Query: `'ccf_usage_trigger_stats{ccf_proc_type="N40 Session Modify SMF Initiated"}'`

Labels:

- Label: `rating_group`

  Label Description: Rating Group for which usage is being reported

  Example: Any string

- Label: `service_identifier`

  Label Description: Service Identifier for which usage is being reported

  Example: Any string

- Label: `ccf_proc_type`

  Label Description: Which kind of procedure usage is reported to CCF

Example: Some String

- Label: `trigger`

  Label Description: Trigger associated with UUC

  Example: Triggers defined as per Specs 32.291

SLA Transaction Category

ccf_sla_transaction_stats

Description: Transaction SLA stats

Sample Query: `sum(ccf_sla_transaction_stats) by (ccf_sla_transaction_stats,ccf_proc_type,status, message_type)`

Labels:

- Label: `ccf_proc_type`

  Label Description: Procedure Name

  Example: N40 Charging Data Request Create, N40 Charging Data Request Release SMF Initiated, N40 Session Modify SMF Initiated, N40 Session Notify - Abort,N40 Session Notify - Reauth, N28 Spending Limit Subscribe, N28 Spending Limit Subscribe Update, N28 Spending Limit UnSubscribe, N28 Spending Limit Notify Status, N28 Spending Limit Terminate Status, N28 Next Reval Notify, N40 Next Reval Notify

- Label: `status`

  Label Description: gives status of the procedure

  Example: Queued, Running, Aborted, Suspended, Invalid, Cleanedup, RequireSuspend, RequireCleanup, RequireAbort, Unknown

- Label: `message_type`

  Label Description: gives the message type received during sla transaction

  Example: IntSelfTxnSla

diameter-ep statistics Category

diam_base_msg_seconds_total

Description: Cumulative response time in seconds of diameter base message requests processed by diameter endpoint

Sample Query: `'diam_base_msg_seconds_total{message_name="DPR"}'`

Labels:

- Label: `message_name`

  Label Description: name of interface message

Example: DPR

- Label: `origin_host`

  Label Description: name of the origin host

  Example: 192.168.169.107

- Label: `origin_realm`

  Label Description: name of the origin realm

  Example: xyz.com

- Label: `disconnect_cause`

  Label Description: reason for the disconnection

  Example: REBOOTING, BUSY, DO_NOT_WANT_TO_TALK_TO_YOU

- Label: `result_code`

  Label Description: result_code describes the error that the diameter node encountered in its processing

  Example: 2001, 5012

- Label: `gr_instance`

  Label Description: GR Instance ID

  Example: 1 or 2

diam_base_msg_total

Description: Count of diameter base message requests processed by diameter endpoint

Sample Query: `'diam_base_msg_total{message_name="DPR"}'`

Labels:

- Label: `message_name`

  Label Description: name of interface message

  Example: DPR

- Label: `origin_host`

  Label Description: name of the origin host

  Example: 192.168.169.107

- Label: `origin_realm`

  Label Description: name of the origin realm

  Example: xyz.com

- Label: `disconnect_cause`

  Label Description: reason for the disconnection

  Example: REBOOTING, BUSY, DO_NOT_WANT_TO_TALK_TO_YOU

- Label: `result_code`

Label Description: result_code describes the error that the diameter node encountered in its processing

Example: 2001, 5012

- Label: `gr_instance`

  Label Description: GR Instance ID

  Example: 1 or 2

diameter_decode_message_total

Description: Count of decoding done by diameter endpoint

Sample Query: `'diameter_decode_message_total{interface="gx"}'`

Labels:

- Label: `interface`

  Label Description: name of the interface

  Example: gx, gy

Labels:

- Label: `message_name`

  Label Description: name of interface message

  Example: ccai, ccau, ccat, rar, asr

- Label: `endpoint_name`

  Label Description: name of endpoint profile used during processing

  Example: gxProf1

- Label: `dict_name`

  Label Description: name of the dictionary used

  Example: gx_cust

- Label: `status`

  Label Description: status of the request

  Example: success, failure, partial

- Label: `unknown_avp`

  Label Description: unknown_avp indicates if any unknow AVPs were found during encoding where 0 indicates not found and 1 indicates found

  Example: 0,1

- Label: `gr_instance`

  Label Description: GR Instance ID

  Example: 1 or 2

diameter_encode_message_total

Description: Count of encoding done by diameter endpoint

Sample Query: `'diameter_encode_message_total{interface="gx"}'`

Labels:

- Label: `interface`

  Label Description: name of the interface

  Example: gx, gy

Labels:

- Label: `message_name`

  Label Description: name of interface message

  Example: ccri, ccru, ccrt, raa, asa

Labels:

- Label: `endpoint_name`

  Label Description: name of endpoint profile used during processing

  Example: gxProf1

- Label: `dict_name`

  Label Description: name of the dictionary used

  Example: gx_cust

- Label: `status`

  Label Description: status of the request

  Example: success, failure, partial

- Label: `unknown_avp`

  Label Description: unknown_avp indicates if any unknow AVPs were found during encoding where 0 indicates not found and 1 indicates found

  Example: 0,1

- Label: `gr_instance`

  Label Description: GR Instance ID

  Example: 1 or 2

diameter_pod_status

Description: Pod status as active/standby

Sample Query: `'diameter_pod_status{vip="10.0.0.1"}'`

Labels:

- Label: `vip`

Label Description: any ip

Example: 10.0.0.1

diameter_request_message_total

Description: Count of diameter requests processed by diameter endpoint

Sample Query: `'diameter_request_message_total{interface="gx"}'`

Labels:

- Label: `interface`

  Label Description: name of the interface

  Example: gx, gy

Labels:

- Label: `message_name`

  Label Description: name of interface message

  Example: ccri, ccru, ccrt, rar, asr

Labels:

- Label: `peer_address`

  Label Description: peer_address will be empty for inbound requests, could be empty for outbound requests depending on point of failure

  Example: 10.1.2.110:3868

Labels:

- Label: `status`

  Label Description: status of the request

  Example: attempted, peer_down, err_cfg, err_maxout, timeout_ipc, err_ipc, err_unmarshal

Labels:

- Label: `retry`

  Label Description: retry count

  Example: 0

- Label: `gr_instance`

  Label Description: GR Instance ID

  Example: 1 or 2

- Label: `transaction_type`

  Label Description: transaction type

  Example: origin

- Label: `endpoint_name`

  Label Description: name of endpoint profile used during processing

  Example: gxProf1

- Label: `message_direction`

  Label Description: direction of message from Diameter perspective

  Example: inbound, outbound

diameter_response_message_seconds_total

Description: Cumulative response time in seconds of diameter requests processed by diameter endpoint

Sample Query: `'diameter_response_message_seconds_total{interface="gx"}'`

Labels:

- Label: `interface`

  Label Description: name of the interface

  Example: gx, gy

Labels:

- Label: `message_name`

  Label Description: name of interface message

  Example: ccai, ccau, ccat, raa, asa

Labels:

- Label: `peer_address`

  Label Description: peer_address will be empty for inbound requests, could be empty for outbound requests depending on point of failure

  Example: 10.1.2.110:3868

Labels:

- Label: `status`

  Label Description: status of the request

  Example: success, err_cfg, err_maxout, err_send, timeout_res, timeout_ipc, err_ipc, err_unmarshal, err_rc, err_exp_rc

Labels:

- Label: `result_code`

  Label Description: result_code describes the result-code or experimental-result-code that the diameter node encountered during response processing

  Example: 1001, 2001, 3001, 4001, 5001

Labels:

- Label: `action`

  Label Description: action

  Example: continue, terminate

Labels:

- Label: `sub_action`

  Label Description: sub action

  Example: discard-traffic, local-fallback, retry-server-on-event, send-ccrt-on-call-termination, with-term-req, without-term-req

- Label: `gr_instance`

  Label Description: GR Instance ID

  Example: 1 or 2

- Label: `endpoint_name`

  Label Description: name of endpoint profile used during processing

  Example: gxProf1

- Label: `message_direction`

  Label Description: direction of message from Diameter perspective

  Example: inbound, outbound

diameter_response_message_total

Description: Count of diameter responses processed by diameter endpoint

Sample Query: `'diameter_response_message_total{interface="gx"}'`

Labels:

- Label: `interface`

  Label Description: name of the interface

  Example: gx, gy

Labels:

- Label: `message_name`

  Label Description: name of interface message

  Example: ccai, ccau, ccat, raa, asa

Labels:

- Label: `peer_address`

  Label Description: peer_address will be empty for inbound requests, could be empty for outbound requests depending on point of failure

  Example: 10.1.2.110:3868

Labels:

- Label: `status`

  Label Description: status of the request

  Example: success, err_cfg, err_maxout, err_send, timeout_res, timeout_ipc, err_ipc, err_unmarshal, err_rc, err_exp_rc

Labels:

- Label: `result_code`

  Label Description: result_code describes the result-code or experimental-result-code that the diameter node encountered during response processing

  Example: 1001, 2001, 3001, 4001, 5001

Labels:

- Label: `action`

  Label Description: action

  Example: continue, terminate

Labels:

- Label: `sub_action`

  Label Description: sub action

  Example: discard-traffic, local-fallback, retry-server-on-event, send-ccrt-on-call-termination, with-term-req, without-term-req

Labels:

- Label: `gr_instance`

  Label Description: GR Instance ID

  Example: 1 or 2

Labels:

- Label: `endpoint_name`

  Label Description: name of endpoint profile used during processing

  Example: gxProf1

Labels:

- Label: `message_direction`

  Label Description: direction of message from Diameter perspective

  Example: inbound, outbound

diameter_route_expires_total

Description: Count of all dynamic routes expired

Sample Query: `'diameter_route_expires_total{gr_instance="1"}'`

Labels:

- Label: `route`

  Label Description: route identified by host, realm and peer

  Example: OCS1:OCS.COM:DRA1

Labels:

- Label: `peer_name`

  Label Description: peer name used by the route

  Example: DRA1

Labels:

- Label: `gr_instance`

  Label Description: GR Instance ID

  Example: 1 or 2

diameter_route_hits_total

Description: Count of all route hits for messages processed by diameter endpoint

Sample Query: `'diameter_route_hits_total{route_type="S"}'`

Labels:

- Label: `route`

  Label Description: route identified by host, realm and peer

  Example: OCS1:OCS.COM:DRA1

Labels:

- Label: `route_type`

  Label Description: type of the route

  Example: S or D

Labels:

- Label: `peer_name`

  Label Description: peer name used by the route

  Example: DRA1

Labels:

- Label: `wild_carded_route`

  Label Description: route with wild carded host or realm

  Example: *:OCS.COM:DRA1

Labels:

- Label: `gr_instance`

  Label Description: GR Instance ID

  Example: 1 or 2

diameter_route_misses_total

Description: Count of all route misses for messages processed by diameter endpoint

Sample Query: `'diameter_route_misses_total{route="OCS1:OCS.COM:DRA1"}'`

Labels:

- Label: `route`

  Label Description: route identified by host, realm and peer

  Example: OCS1:OCS.COM:DRA1

Labels:

- Label: `route_type`

  Label Description: type of the route

  Example: S or D

Labels:

- Label: `peer_name`

  Label Description: peer name used by the route

  Example: DRA1

Labels:

- Label: `gr_instance`

  Label Description: GR Instance ID

  Example: 1 or 2

diameter_route_status

Description: Status of a route

Sample Query: `'diameter_route_status{route="OCS1:OCS.COM:DRA1"}'`

Labels:

- Label: `route`

  Label Description: route identified by host, realm and peer

  Example: OCS1:OCS.COM:DRA1

Labels:

- Label: `route_type`

  Label Description: type of the route

  Example: S or D

Labels:

- Label: `peer_name`

  Label Description: peer name used by the route

  Example: DRA1

Labels:

- Label: `gr_instance`

  Label Description: GR Instance ID

  Example: 1 or 2

Labels:

- Label: `route_status`

  Label Description: status of the route

  Example: Pending or Active or Inactive or Failed or Deleted or Expired or Cloned

{{- end}}

diameter_routes_total

Description: Count of all diameter routes added by diameter endpoint

Sample Query: `'diameter_routes_total{gr_instance="1"}'`

Labels:

- Label: `route_type`

  Label Description: type of the route

  Example: S or D

Labels:

- Label: `peer_name`

  Label Description: peer name used by the route

  Example: DRA1

Labels:

- Label: `gr_instance`

  Label Description: GR Instance ID

  Example: 1 or 2

dispatch_error_seconds_total

Description: Cumulative time in seconds spent during dispatching of inbound requests to service that had error or timeout

Sample Query: `'dispatch_error_seconds_total{gr_instance="1"}'`

Labels:

- Label: `application`

  Label Description: name of the application

  Example: diameter

Labels:

- Label: `command_code`

  Label Description: command code

  Example: RAR

Labels:

- Label: `error_code`

  Label Description: error code

  Example: 3002

Labels:

- Label: `gr_instance`

  Label Description: GR Instance ID

  Example: 1 or 2

dispatch_error_total

Description: Count of inbound requests that had error or timeout during dispatch to service

Sample Query: `'dispatch_error_total{gr_instance="1"}'`

Labels:

- Label: `application`

  Label Description: name of the application

  Example: diameter

Labels:

- Label: `command_code`

  Label Description: command code

  Example: RAR

Labels:

- Label: `error_code`

  Label Description: error code

  Example: 3002

Labels:

- Label: `gr_instance`

  Label Description: GR Instance ID

  Example: 1 or 2

policy_engine_message_seconds_total

Description: Cumulative time in seconds spent during processing of message sent to service

Sample Query: `'policy_engine_message_seconds_total{gr_instance="1"}'`

Labels:

- Label: `application`

  Label Description: name of the application

  Example: diameter

Labels:

- Label: `command_code`

  Label Description: command code

  Example: RAR

Labels:

- Label: `gr_instance`

  Label Description: GR Instance ID

  Example: 1 or 2

policy_engine_message_total

Description: Count of messages sent to service for which response is received

Sample Query: `'policy_engine_message_total{gr_instance="1"}'`

Labels:

- Label: `application`

  Label Description: name of the application

  Example: diameter

Labels:

- Label: `command_code`

Label Description: command code

Example: RAR

Labels:

- Label: `gr_instance`

  Label Description: GR Instance ID

  Example: 1 or 2

dns-proxy Metrics Reference

DNS Lookup Request Statistics Category

DNS_Lookup_Requests_Statistics

Description: Total dns packets processed

Sample Query:
`'DNS_Lookup_Requests_Statistics{dnsPacketDir="Rx",dnsProtocol="udp",dnsQueryType="ipv4",dnsResult="Success",dnsSvrIP="172.17.0.2",dnsSvrPort="53",grInstId="1"}'`

Labels:

- Label: `dnsPacketDir`

  Label Description: Direction

  Example: Tx, Rx

- Label: `dnsProtocol`

  Label Description: Protocol

  Example: udp, tcp

- Label: `dnsQueryType`

  Label Description: DNS Lookup Query type

  Example: ipv4, ipv6, ipv4v6

- Label: `dnsResult`

  Label Description: Result

  Example: Success, Failure, Timeout, Failure_No_Response

- Label: `dnsSvrIP`

  Label Description: DNS Server IP Address

  Example: Any string

- Label: `dnsSvrPort`

  Label Description: DNS Server Port

Example: Any string

- Label: `grInstId`

  Label Description: GR Instance Id

  Example: 1 or 2

gtpc-ep Metrics Reference

GTPC BGP Routed Count Stats Category

gtpc_app_bgp_routes_count

Description: BGP routes add counter

Sample Query: `'gtpc_app_bgp_routes_count{status="success"}'`

Labels:

- Label: `status`

  Label Description: status

  Example: success, failed

- Label: `gr_instance_id`

  Label Description: GR instance ID

  Example: 1, 2

GTPC Roaming Peer Path Mgmt Stats Category

gtpc_roaming_peer_path_mgmt

Description: GTPC Roaming Peer Path Mgmt Stats

Sample Query: `'gtpc_roaming_peer_path_mgmt{service_name="gtpc-ep",status="suppressed"}'`

Labels:

- Label: `gtpc_peer_type`

  Label Description: Gtpc Peer type

  Example: ROAMER, HOMER, VISITOR

- Label: `interface_type`

  Label Description: Gtpc Interface type

  Example: S5, S11, S5E, S2B, S8

- Label: `gtpc_msg_type`

  Label Description: Gtpc Message type

Example: NumEchoMsg, NumControlMsg

- Label: `status`

  Label Description: Status

  Example: suppressed

- Label: `gr_instance_id`

  Label Description: GR instance ID

  Example: 1, 2

gtpp-ep Metrics Reference

ASN1 Encoding stats Category

gtpp_asn1field_encoding_failures_total

Description: A counter for total number of fields failed to be ASN1 encoded

Sample Query: `'gtpp_asn1field_encoding_failures_total{gtpp_profile="pf1"}'`

Labels:

- Label: `gtpp_profile`

  Label Description: Gtpp Profile Name

  Example: pf1, pf2

- Label: `dictionary`

  Label Description: Gtpp Dictionary Used

  Example: custom6, custom24

- Label: `asn1_field`

  Label Description: ASN1 Field

  Example: ServedIMSI, ChargingID, ServingNodeAddress, ChargingCharacteristics

- Label: `reason`

  Label Description: Reason for failure

  Example: Constraint Violation

CDR Batch flush duration stats Category

gtpp_batch_flush_duration_histogram_total

Description: Histogram time bins of batch flush time

Sample Query: `'gtpp_batch_flush_duration_histogram_total{gtpp_profile="pf1"}'`

Labels:

- Label: `gtpp_profile`

  Label Description: Gtpp Profile Name

  Example: pf1, pf2

- Label: `dictionary`

  Label Description: Gtpp Dictionary Used

  Example: custom6, custom24

- Label: `status`

  Label Description: Status of the operation

  Example: batch_success,batch_error

CDR Batch flush stats Category

gtpp_batch_flush_millis_total

Description: A Histogram for the time CDRs remain in batch before flushed

Sample Query: `'gtpp_batch_flush_millis_total{gtpp_profile="pf1"}'`

Labels:

- Label: `gtpp_profile`

  Label Description: Gtpp Profile Name

  Example: pf1, pf2

- Label: `dictionary`

  Label Description: Gtpp Dictionary Used

  Example: custom6, custom24

- Label: `status`

  Label Description: Status of the operation

  Example: batch_success,batch_error

CDR Batching Stats Category

gtpp_batched_cdrs_total

Description: No.of CDRs in a batch for a given profile

Sample Query: `'gtpp_batched_cdrs_total{gtpp_profile="pf1"}'`

Labels:

- Label: `gtpp_profile`

  Label Description: Gtpp Profile Name

  Example: pf1, pf2
- Label: `dictionary`

  Label Description: Gtpp Dictionary Used

  Example: custom6, custom24
- Label: `status`

  Label Description: Status of the operation

  Example: batch_success,batch_error

DupReqList buffer gauge Category

gtpp_dup_reqlist_counter

Description: GTPP Dup Req List

Sample Query: `'gtpp_dup_reqlist_counter{gtpp_profile="pf1"}'`

Labels:

- Label: `gtpp_profile`

  Label Description: Gtpp Profile Name

  Example: pf1, pf2
- Label: `cgf_addr`

  Label Description: Cgf Server Address

  Example: 1.2.3.4
- Label: `state`

  Label Description: Request Process State

  Example: empty_cdr_rsp_wait,dup_cdr_rel_rsp_wait,dup_cdr_can_rsp_wait,to_send_empty_rsp_wait

File based CDR Read Category

gtpp_file_record_read

Description: GTPP CDR file based reads

Sample Query: `'gtpp_file_record_read{gtpp_profile="pf1"}'`

Labels:

- Label: `gtpp_profile`

  Label Description: Gtpp Profile Name

Example: pf1, pf2

- Label: `status`

  Label Description: CDR read operation status

  Example: success,failure

- Label: `pod_status`

  Label Description: Active/StandBy

  Example: active,standby

File based CDR Write Category

gtpp_file_record_write

Description: GTPP CDR file based writes

Sample Query: `'gtpp_file_record_write{gtpp_profile="pf1"}'`

Labels:

- Label: `gtpp_profile`

  Label Description: Gtpp Profile Name

  Example: pf1, pf2

- Label: `status`

  Label Description: CDR write operation status

  Example: success,failure

- Label: `pod_status`

  Label Description: Active/StandBy

  Example: active,standby

GTPP Archive List gauge Category

gtpp_archive_list_counter

Description: GTPP Archive List

Sample Query: `'gtpp_archive_list_counter{gtpp_profile="pf1"}'`

Labels:

- Label: `gtpp_profile`

  Label Description: Gtpp Profile Name

  Example: pf1, pf2

- Label: `pod_status`

Label Description: Active/StandBy

Example: active,standby

GTPP Messages Stats Category

gtpp_msg_stats

Description: GTPP MSG Stats

Sample Query: `'gtpp_msg_stats{gtpp_profile="pf1"}'`

Labels:

- Label: `gtpp_profile`

  Label Description: Gtpp Profile Name

  Example: pf1, pf2

- Label: `msg_type`

  Label Description: GTPP Msg Name

  Example: DataRecTransReq, DataRecTransReqPossibleDup, DataRecTransReqCancel, DataRecTransReqRelease, DataRecTransReqEmpty, DataRecTransReqRetried, DataRecTransReqPossibleDuplRetried, DataRecTransReqCancelRetried, DataRecTransReqReleaseRetried, DataRecTransReqEmptyRetried, DataRecTransRsp, DataRecTransPossibleDuplRsp, DataRecTransCancelRsp, DataRecTransReleaseRsp, DataRecTransEmptyRsp, EchoReqSent, EchoReqRcvd, EchoRspSent, EchoRspRcvd, NodeAliveReqSent, NodeAliveReqRcvd, NodeAliveRspSent, NodeAliveRspRcvd, TestEchoReqSent, TestEchoRspRcvd

- Label: `status`

  Label Description: Request/Response Message Handling Status

  Example: success,failure

- Label: `cause`

  Label Description: GTPP Messages Response Cause

  Example: accepted, mandatory_ie_incorrect, mandatory_ie_missing, invalid_message_format, optional_ie_incorrect, no_resources_available, system_failure, service_not_supported, version_not_supported, not_fulfilled, already_fulfilled, cdr_decode_error, sequence_numbers_incorrect,buffer_full,internal_failure,ipc_err

- Label: `cgf_addr`

  Label Description: Cgf Server Address

  Example: 1.2.3.4

- Label: `trigger_type`

  Label Description: Trigger for this Message

  Example: normal cdr, pod_switchover, peer_restart, cgf_down

GTPP Replication Msg Stats Category

gtpp_replication_stats

Description: GTPP Replication Data Msg

Sample Query: `'gtpp_replication_stats{gtpp_profile="pf1"}'`

Labels:

- Label: `gtpp_profile`

  Label Description: Gtpp Profile Name

  Example: pf1, pf2

- Label: `cgf_addr`

  Label Description: Cgf Server Address

  Example: 1.2.3.4

- Label: `item`

  Label Description: Peer Struct Replicated Item

  Example: SndReqLisfTx,DupReqLisfTx,SeqPoolTx,SndReqListRx,DupReqListRx,SeqPoolRx,ArchiveListTx,ArchiveListRx,FileCdrTx,FileCdrRx

- Label: `op_type`

  Label Description: Operation Type

  Example: Add,Update,Delete

- Label: `status`

  Label Description: Peer Replication Msg Handling Status

  Example: success,failure

Inbound CDR Requests Category

gtpp_received_cdrs_total

Description: A counter for total number of CDRs received in request

Sample Query: `'gtpp_received_cdrs_total{gtpp_profile="pf1"}'`

Labels:

- Label: `gtpp_profile`

  Label Description: Gtpp Profile Name

  Example: pf1, pf2

- Label: `dictionary`

  Label Description: Gtpp Dictionary Used

Example: custom6, custom24

Processed CDR Requests Category

gtpp_processed_cdrs_total

Description: A counter for total number of CDRs processed in transaction

Sample Query: `'gtpp_processed_cdrs_total{gtpp_profile="pf1"}'`

Labels:

- Label: `gtpp_profile`

  Label Description: Gtpp Profile Name

  Example: pf1, pf2

- Label: `dictionary`

  Label Description: Gtpp Dictionary Used

  Example: custom6, custom24

- Label: `status`

  Label Description: Status of processed CDR request

  Example: Success, profile_error, not_leader_pod_error

Read from GTPP Archive List Stats Category

gtpp_archive_list_read

Description: GTPP Archive List Read

Sample Query: `'gtpp_archive_list_read{gtpp_profile="pf1"}'`

Labels:

- Label: `gtpp_profile`

  Label Description: Gtpp Profile Name

  Example: pf1, pf2

- Label: `status`

  Label Description: Status of CDR read from Archive

  Example: success, failure

- Label: `sub_status`

  Label Description: Sub Status of CDR read from Archive

  Example: lookup, send_to_cgf, send_to_hdd, purge

- Label: `pod_status`

Label Description: Active/StandBy

Example: active,standby

SendReqList buffer gauge Category

gtpp_send_reqlist_counter

Description: GTPP Send Req List

Sample Query: `'gtpp_send_reqlist_counter{gtpp_profile="pf1"}'`

Labels:

- Label: `gtpp_profile`

  Label Description: Gtpp Profile Name

  Example: pf1, pf2

- Label: `cgf_addr`

  Label Description: Cgf Server Address

  Example: 1.2.3.4

- Label: `state`

  Label Description: Request Process State

  Example: Normal_CDR_Rsp_Wait,Duplicate_CDR_Rsp_Wait

Write to GTPP Archive List Stats Category

gtpp_archive_list_write

Description: GTPP Archive List Write

Sample Query: `'gtpp_archive_list_write{gtpp_profile="pf1"}'`

Labels:

- Label: `gtpp_profile`

  Label Description: Gtpp Profile Name

  Example: pf1, pf2

- Label: `status`

  Label Description: Status of CDR write to Archive

  Example: success, failure

- Label: `sub_status`

  Label Description: Sub Status of CDR write to Archive

  Example: purge_oldest_cdr

- Label: `pod_status`

  Label Description: Active/StandBy

  Example: active,standby

nodemgr Metrics Reference

Nodemgr diagnostics address validation counter Category

nodemgr_diag_ip_verify

Description: Node manager diagnostics address validation counter

Sample Query: `'nodemgr_diag_ip_verify{status="success"}'`

Labels:

- Label: `status`

  Label Description: diagnostics address validation status

  Example: success, unknown, cdl_ipc_failure, ipv4_alloc_failed, ipv6_alloc_failed, ip_mismatch_ignored, duplicate_record_found

Nodemgr gtpc message statistics Category

nodemgr_gtpc_msg_stats

Description: Node manager gtpc message statistics for updating node status to gtpc peers like SGW, PGW or ePDG

Sample Query: `'nodemgr_gtpc_msg_stats{gtpc_peer_ip="192.168.10.2", gtpc_msg_type="gtpc_echo_res_rx", interface_type="S11"}'`

Labels:

- Label: `gtpc_peer_ip`

  Label Description: IP address of a gtpc peer like SGW, PGW or ePDG

  Example: 192.168.10.2

- Label: `gtpc_msg_type`

  Label Description: GTPC message triggered by the current node, or triggered by peer node

  Example: gtpc_echo_res_rx, gtpc_echo_res_tx, gtpc_echo_req_rx,

- Label: `interface_type`

  Label Description: Interfaces on which the gtpc message is recieved or sent PGW, SGW-Egress, SGW-Ingress etc

Example: S11, S5E, S5, S8, S2B

- Label: `gr_instance_id`

  Label Description: GR instance ID

  Example: 1, 2

- Label: `gtpc_msg_ver`

  Label Description: GTP Message Version

  Example: v1, v2

Nodemgr gtpc peer status statistics Category

nodemgr_gtpc_peer_status

Description: Node manager gtpc peer status statistics for keeping track of gtpc peers like SGW, PGW or ePDG via keep alive or restart counter tracking

Sample Query: `'nodemgr_gtpc_peer_status{gtpc_peer_ip="192.168.10.2", gtpc_msg_type="gtpc_echo_res_rx", interface_type="S11"}'`

Labels:

- Label: `gtpc_peer_ip`

  Label Description: IP address of a gtpc peer like SGW, PGW or ePDG

  Example: 192.168.10.2

- Label: `gtpc_peer_status`

  Label Description: GTPC peer current status as a result of keep alive success/failure or restart counter tracking

  Example: gtpc_peer_path_down, gtpc_peer_path_up, gtpc_peer_restarted

- Label: `interface_type`

  Label Description: Interfaces on which the gtpc message is recieved or sent PGW, SGW-Egress, SGW-Ingress etc

  Example: S11, S5E, S5, S8, S2B

- Label: `restart_time`

  Label Description: Gtpc peer restart time

  Example: 2022-09-30 14:32:52 +0000 UTC

- Label: `gr_instance_id`

  Label Description: GR instance ID

  Example: 1, 2

- Label: `gtpc_msg_ver`

  Label Description: GTP Message Version

  Example: v1, v2

Nodemgr GTPP link status Category

nodemgr_gtpp_stats

Description: Nodemgr gauge counter to keep track of GTPP peer (CGF) link status between CGF and SMF/SGW

Sample Query: `'nodemgr_gtpp_stats{gtpp_cgf_key="192.168.10.2:9001", gtpp_ca_key="192.168.10.3:9001"}'`

Labels:

- Label: `gtpp_cgf_key`

  Label Description: unique key to identify CGF YYY.YYY.YYY.YYY:XXX where YYY.YYY.YYY.YYY is the Ip address of the CGF and XXX is the port number

  Example: 192.168.10.2:9001

- Label: `gtpp_ca_key`

  Label Description: unique key to identify Charging Agent (CA) YYY.YYY.YYY.YYY:XXX where YYY.YYY.YYY.YYY is the Ip address of Charging agent configured on the NF service like SGW / SMF and XXX is the port number

  Example: 192.168.20.3:9001

- Label: `gr_instance_id`

  Label Description: GR instance ID

  Example: 1, 2

Nodemgr gtpp message statistics Category

nodemgr_gtpp_msg_stats

Description: Node manager gtpp message statistics to update stats for msg triggered by NodeMgr

Sample Query: `'nodemgr_gtpp_msg_stats{ca_ip="10.65.45.181",ca_port="2222",cgf_ip="10.78.41.152",cgf_port="3386",gtpp_msg_type="gtpp_echo_req"}'`

Labels:

- Label: `ca_ip`

  Label Description: Charging agent ip address

  Example: 10.65.45.181

- Label: `ca_port`

  Label Description: Charging agent port number

  Example: 2222

- Label: `cgf_ip`

  Label Description: Cgf ip address

  Example: 10.78.41.152

- Label: `cgf_port`

  Label Description: Cgf port number

  Example: 3386

- Label: `gtpp_msg_type`

  Label Description: Msg triggered towards gtpp-ep like echo req/path up/path down/peer restart

  Example: gtpp_echo_req, gtpp_peer_path_up, gtpp_peer_path_down, gtpp_peer_restarted

Nodemgr gtpp peer status statistics Category

nodemgr_gtpp_peer_status

Description: Node manager gtpp peer status statistics for keeping track of cgf nodes via node alive/echo/gtpp control message success/failure or restart counter tracking

Sample Query:

```
'nodemgr_gtpp_peer_status{ca_ip="10.65.45.181",ca_port="2222",cgf_ip="10.78.41.152",cgf_port="3386",gtpp_peer_reason="PEER_ADD",gtpp_peer_status="gtpp_peer_path_up",restart_time="2023-03-29
15:46:27 +0530 IST"}'
```

Labels:

- Label: `ca_ip`

  Label Description: Charging agent ip address

  Example: 10.65.45.181

- Label: `ca_port`

  Label Description: Charging agent port number

  Example: 2222

- Label: `cgf_ip`

  Label Description: Cgf ip address

  Example: 10.78.41.152

- Label: `cgf_port`

  Label Description: Cgf port number

  Example: 3386

- Label: `gtpp_peer_reason`

  Label Description: Cgf update reason (add/update/delete) as a result of node alive/echo/gtpp control message success/failure or restart counter tracking

  Example: PEER_ADD, PEER_UP, PEER_DELETE, NO_ECHO_RESPONSE, NO_CTRL_MSG_RESPONSE

- Label: `gtpp_peer_status`

  Label Description: Cgf status (up/down/restart) as a result of node alive/echo/gtpp control message success/failure or restart counter tracking

Example: gtpp_peer_path_down, gtpp_peer_path_up, gtpp_peer_restarted

- Label: `restart_time`

  Label Description: Cgf restart time

  Example: 2023-03-29 15:46:27 +0530 IST

Nodemgr messages Category

nodemgr_msg_stats

Description: Node Manager Resource management message counters

Sample Query: `'nodemgr_msg_stats{nodemgr_id="0", id_req_type="ID_REQ_ALLOC", ip_req_type="IP_REQ_ALLOC", ip_version="IP_TYPE_V4",sent_to_owner="0", service_user="SERVICE_USER_SMF"}'`

Labels:

- Label: `nodemgr_id`

  Label Description: Node Manager instance for which statistics are to be checked

  Example: 0, 1, 2, 3, 4, 5, 6, 7, 8

- Label: `id_req_type`

  Label Description: Type of request recieved at node manager message

  Example: ID_REQ_NONE, ID_REQ_ALLOC, ID_REQ_REL, ID_REQ_REALLOC

- Label: `ip_req_type`

  Label Description: Type of request recieved at node manager for IP address

  Example: IP_REQ_NONE, IP_REQ_ALLOC, IP_REQ_REL, IP_REQ_REALLOC, IP_REQ_STATIC

- Label: `ip_version`

  Label Description: IP address type for which request was recieved

  Example: IP_TYPE_NONE, IP_TYPE_V4, IP_TYPE_V6, IP_TYPE_V4V6

- Label: `sent_to_owner`

  Label Description: Current Node Manager instance for which statistics are to be checked

  Example: 0, 1, 2, 3, 4, 5, 6, 7, 8

- Label: `service_user`

  Label Description: Node Type which has requested the Node Manager services

  Example: SERVICE_USER_NONE, SERVICE_USER_SMF, SERVICE_USER_SGW

- Label: `gr_instance_id`

  Label Description: GR instance ID

  Example: 1, 2

Nodemgr node report message hanlding from UPF to current node stats Category

nodemgr_node_report_stats

Description: Node Manager handling of node report from UPF about the status of NR's or gNB's having sessions with the UPF

Sample Query: `'nodemgr_node_report_stats{up_ep_key="192.168.10.2:192.168.20.3", node_report_peer_gtpu="192.168.30.4", node_report_no_of_sess="0", status="success", node_report_type="", session_tmr="10", backlog_tmr="1564555678270689300"}'`

Labels:

- Label: `up_ep_key`

  Label Description: unique key to identify UPF XXX.XXX.XXX.XXX:YYY.YYY.YYY.YYY where XXX.XXX.XXX.XXX is Ip address of the NF service like SGW / SMF and YYY.YYY.YYY.YYY is the IP address of UPF

  Example: 192.168.10.2:192.168.20.3

- Label: `node_report_peer_gtpu`

  Label Description: Peer GTPU IP address of gNB or NR to which UPF has established the userplane session

  Example: 192.168.30.4

- Label: `node_report_no_of_sess`

  Label Description: Total number of session established for the Peer GTPU gNB or NR via the UPF

  Example: 0

- Label: `status`

  Label Description: Node report message handling status by Node manager

  Example: attempted, success, failure

- Label: `node_report_type`

  Label Description: Type of node report message being handled

  Example: upd_TS_failed, duplicate, origin

- Label: `session_tmr`

  Label Description: Time duration in minutes during which the node report message has to be handled by the SMF/SGW/PGW node

  Example: 0, 10

- Label: `backlog_tmr`

  Label Description: Current time stamp in unix epoch value for node report message processing

  Example: 1564555678270689300

- Label: `gr_instance_id`

  Label Description: GR instance ID

Example: 1, 2

Nodemgr node report message hanlding timer stats Category

nodemgr_node_rpt_timer_stats

Description: Timer to handle Node Manager handling of node report from UPF about the status of NR's or gNB's having sessions with the UPF

Sample Query: `'nodemgr_node_rpt_timer_stats{up_ep_key="192.168.10.2:192.168.20.3", node_report_peer_gtpu="192.168.30.4", node_report_no_of_sess="0", status="success", node_report_type="", backlog_tmr="1564555678270689300"}'`

Labels:

- Label: `up_ep_key`

  Label Description: unique key to identify UPF XXX.XXX.XXX.XXX:YYY.YYY.YYY.YYY where XXX.XXX.XXX.XXX is Ip address of the NF service like SGW / SMF and YYY.YYY.YYY.YYY is the IP address of UPF

  Example: 192.168.10.2:192.168.20.3

- Label: `node_report_peer_gtpu`

  Label Description: Peer GTPU IP address of gNB or NR to which UPF has established the userplane session

  Example: 192.168.30.4

- Label: `node_report_no_of_sess`

  Label Description: Total number of session established for the Peer GTPU gNB or NR via the UPF

  Example: 0

- Label: `status`

  Label Description: Node report message handling status by Node manager

  Example: attempted, success, failure

- Label: `node_report_type`

  Label Description: Type of node report message being handled

  Example: tmr_start_failed, dbg_tmr, retry_clrBlkSubs

- Label: `backlog_tmr`

  Label Description: Current time stamp in unix epoch value for node report message processing

  Example: 1564555678270689300

- Label: `gr_instance_id`

  Label Description: GR instance ID

  Example: 1, 2

Nodemgr resource management batch reconcilation counter Category

nodemgr_rmgr_batch_reconcile_stats

Description: Node manager resource management batch reconcilation counter

Sample Query: `'nodemgr_rmgr_batch_reconcile_stats{status="success"}'`

Labels:

- Label: `status`

  Label Description: reconcilation status

  Example: success, failed

Nodemgr resource management response statistics Category

nodemgr_resource_mgmt_resp_stats

Description: Node Manager resource management response statistics

Sample Query: `'nodemgr_resource_mgmt_resp_stats{req_type="1", ip_ver_type="1", status="attempted", error=""}'`

Labels:

- Label: `req_type`

  Label Description: The request for which this response is being sent, Request with no operation = 0, Request with IP allocation = 1, Request with IP release = 2, Request with IP reallocation = 3, Request with Static IP allocation = 4

  Example: 0, 1, 2, 3, 4

- Label: `ip_ver_type`

  Label Description: Type of IP addresses requested in the message, IP type requested NONE = 0, IP type requeseted V4 = 1, IP type requeseted V6 = 2, IP type requeseted V4V6 = 3

  Example: 0, 1, 2, 3

- Label: `status`

  Label Description: Status of the request

  Example: attempted, success, failed

- Label: `error`

  Label Description: A non unique error String in case of Status is failure, for other cases use this value as empty string

  Example: Unable to get UpfKey for upf

- Label: `gr_instance_id`

  Label Description: GR instance ID

  Example: 1, 2

Nodemgr UPF ip address threshol hit stats Category

nodemgr_up_threshold_stats

Description: When particular IP address pool threshold hit for usage of ip addresses of a particular address type,this stats will be recorded

Sample Query: `'nodemgr_up_threshold_stats{up_ep_key="192.168.10.2:192.168.20.3", dnn="sampleDNN",threshold_hit="yes",ip_ver_type="IP_TYPE_V4"}'`

Labels:

- Label: `up_ep_key`

  Label Description: unique key to identify UPF XXX.XXX.XXX.XXX:YYY.YYY.YYY.YYY where XXX.XXX.XXX.XXX is Ip address of the NF service like SGW / SMF and YYY.YYY.YYY.YYY is the IP address of UPF

  Example: 192.168.10.2:192.168.20.3

- Label: `dnn`

  Label Description: DNN of which ip pool reached the configured threshold usgae.

  Example: sampleDNN

- Label: `threshold_hit`

  Label Description: Indicates if threhold hit is yes or no.

  Example: yes

- Label: `threshold_clear`

  Label Description: Indicates if threhold hit is cleared or not

  Example: yes

- Label: `nodemgr_id`

  Label Description: Indicates which instance of nodemgr hit the threshold

  Example: 1

- Label: `gr_instance_id`

  Label Description: GR instance ID

  Example: 1, 2

- Label: `ip_ver_type`

  Label Description: Type of IP addresses for which threshold is hit or clear

  Example: IP_TYPE_V4, IP_TYPE_V6

Nodemgr UPF Path failure reasons Category

nodemgr_up_pathfail_reasons

Description: Node manager userplane heart beat message failure reasons stats

Sample Query:

```
'nodemgr_up_pathfail_reasons{app_name="smf",cluster="cn1",data_center="cn1",instance_id="0",service_name="nodemgr",up_pathfail_reason="up_pathfail_ignored_hb_retry"}
1'
```

Labels:

- Label: `up_pathfail_reasons`

  Label Description: Node manager UPF Path Failure reasons due to restransmission failure, RTS change and Sx Release from peer node

  Example:
  up_pathfail_ignored_hb_retry,up_pathfail_reason_hb_retry,up_pathfail_ignored_hb_retry_change,up_pathfail_reason_hb_retry_change,up_pathfail_reason_association_release,up_pathfail_reason_association_update

- Label: `gr_instance_id`

  Label Description: GR instance ID

  Example: 1, 2

Nodemgr UPF Peer status Category

nodemgr_up_peer_status

Description: Node manager userplane heart beat message failure reasons stats

Sample Query:

```
'nodemgr_up_peer_status{app_name="smf",cluster="cn1",data_center="cn1",instance_id="1",interface_type="SX",service_name="nodemgr",up_peer_ip="20.20.20.66:20.20.20.42",up_peer_status="up_peer_path_up"}
1'
```

Labels:

- Label: `up_peer_ip`

  Label Description: unique key to identify UPF YYY.YYY.YYY.YYY:XXX.XXX.XXX.XXX where XXX.XXX.XXX.XXX is Ip address of the NF service like SGW / SMF and YYY.YYY.YYY.YYY is the IP address of UPF

  Example: 20.20.20.66:20.20.20.42

- Label: `up_peer_status`

  Label Description: Node manager UPF Peer status

  Example: up_peer_path_down,up_peer_path_up

- Label: `interface_type`

  Label Description: nterface type between Peer Node (UPF)

  Example: SXA

- Label: `gr_instance_id`

  Label Description: GR instance ID

  Example: 1, 2

Nodemgr userplane heart beat message failure due to retransmission stats Category

nodemgr_up_heartbeat_fail_stats

Description: Node Manager userplane heart beat message failure counters between UPF node and SMF/PGW/SGW node as retransmission requests exhausted to UPF

Sample Query: `'nodemgr_up_heartbeat_fail_stats{up_ep_key="192.168.10.2:192.168.20.3", primary_nodemgr_id="0", current_nodemgr_id="0", up_msg_type="up_heartbeat_req_tx", interface_type="SXB"}'`

Labels:

- Label: `up_ep_key`

  Label Description: unique key to identify UPF XXX.XXX.XXX.XXX:YYY.YYY.YYY.YYY where XXX.XXX.XXX.XXX is Ip address of the NF service like SGW / SMF and YYY.YYY.YYY.YYY is the IP address of UPF

  Example: 192.168.10.2:192.168.20.3

- Label: `interface_type`

  Label Description: Interface type between current node (SMF/SGW) and Peer Node (UPF)

  Example: SXA, SXB, SXAB, SXC, N4

- Label: `gr_instance_id`

  Label Description: GR instance ID

  Example: 1, 2

Nodemgr userplane heart beat message failure stats Category

nodemgr_up_hb_msg_fail_stats

Description: Node Manager userplane heart beat message failure counters between UPF node and SMF/PGW/SGW node as unable to send request to UPF

Sample Query: `'nodemgr_up_hb_msg_fail_stats{up_ep_key="192.168.10.2:192.168.20.3", primary_nodemgr_id="0", current_nodemgr_id="0", up_msg_type="up_heartbeat_req_tx", interface_type="SXB"}'`

Labels:

- Label: `up_ep_key`

  Label Description: unique key to identify UPF XXX.XXX.XXX.XXX:YYY.YYY.YYY.YYY where XXX.XXX.XXX.XXX is Ip address of the NF service like SGW / SMF and YYY.YYY.YYY.YYY is the IP address of UPF

  Example: 192.168.10.2:192.168.20.3

- Label: `primary_nodemgr_id`

  Label Description: Node Manager instance Identifier of SGW/SMF service which originally established interaction with UPF

  Example: 0, 1, 2, 3, 4, 5, 6, 7, 8

- Label: `current_nodemgr_id`

  Label Description: Current Node Manager instance Identifier of SGW/SMF service which is currently established and interacting with UPF

  Example: 0, 1, 2, 3, 4, 5, 6, 7, 8

- Label: `up_msg_type`

  Label Description: Message type which is received or sent for heartbeat messaging

  Example: up_heartbeat_req_tx, up_heartbeat_req_retx, up_heartbeat_rsp_rx

- Label: `interface_type`

  Label Description: Interface type between current node (SMF/SGW) and Peer Node (UPF)

  Example: SXA, SXB, SXAB, SXC, N4

- Label: `gr_instance_id`

  Label Description: GR instance ID

  Example: 1, 2

Nodemgr userplane heart beat message stats Category

nodemgr_up_hb_msg_stats

Description: Node Manager userplane heart beat message counters between UPF node and SMF/PGW/SGW node

Sample Query: `'nodemgr_up_hb_msg_stats{up_ep_key="192.168.10.2:192.168.20.3", primary_nodemgr_id="0", current_nodemgr_id="0", up_msg_type="up_heartbeat_req_tx", interface_type="SXB"}'`

Labels:

- Label: `up_ep_key`

  Label Description: unique key to identify UPF XXX.XXX.XXX.XXX:YYY.YYY.YYY.YYY where XXX.XXX.XXX.XXX is Ip address of the NF service like SGW / SMF and YYY.YYY.YYY.YYY is the IP address of UPF

  Example: 192.168.10.2:192.168.20.3

- Label: `primary_nodemgr_id`

  Label Description: Node Manager instance Identifier of SGW/SMF service which originally established interaction with UPF

  Example: 0, 1, 2, 3, 4, 5, 6, 7, 8

- Label: `current_nodemgr_id`

  Label Description: Current Node Manager instance Identifier of SGW/SMF service which is currently established and interacting with UPF

  Example: 0, 1, 2, 3, 4, 5, 6, 7, 8

- Label: `up_msg_type`

Label Description: Message type which is received or sent for heartbeat messaging

Example: up_heartbeat_req_tx, up_heartbeat_req_retx, up_heartbeat_rsp_rx

- Label: `interface_type`

  Label Description: Interface type between current node (SMF/SGW) and Peer Node (UPF)

  Example: SXA, SXB, SXAB, SXC, N4

- Label: `gr_instance_id`

  Label Description: GR instance ID

  Example: 1, 2

Nodemgr userplane link status stats Category

nodemgr_upf_link_status

Description: Nodemgr gauge counter to keep track of userplane link status between UPF and SMF/SGW

Sample Query: `'nodemgr_upf_link_status{up_ep_key="192.168.10.2:192.168.20.3", primary_nodemgr_id="0", peer_nodemgr_id="0", interface_type="SXB"}'`

Labels:

- Label: `up_ep_key`

  Label Description: unique key to identify UPF XXX.XXX.XXX.XXX:YYY.YYY.YYY.YYY where XXX.XXX.XXX.XXX is Ip address of the NF service like SGW / SMF and YYY.YYY.YYY.YYY is the IP address of UPF

  Example: 192.168.10.2:192.168.20.3

- Label: `primary_nodemgr_id`

  Label Description: Current Node Manager instance Identifier of SGW/SMF service

  Example: 0, 1, 2, 3, 4, 5, 6, 7, 8

- Label: `peer_nodemgr_id`

  Label Description: Peer Node Manager instance Identifier of UPF service

  Example: 0, 1, 2, 3, 4, 5, 6, 7, 8

- Label: `interface_type`

  Label Description: Interface type between current node (SMF/SGW) and Peer Node (UPF)

  Example: SXA, SXB, SXAB, SXC, N4

- Label: `gr_instance_id`

  Label Description: GR instance ID

  Example: 1, 2

Nodemgr userplane routes update or delete counter Category

nodemgr_up_route_stats

Description: Nodemgr userplane routes update or delete counter

Sample Query:
'nodemgr_up_route_stats{app_name="smf",cluster="cl",data_center="cl",instance_id="0",service_name="nodemgr",route_operation="Update",status="Success",up_ep_key="192.168.10.2:192.168.20.3",tx_type="NewTx"}
1'

Labels:

- Label: `route_operation`

  Label Description: Kind of route operation

  Example: Update, Delete

- Label: `status`

  Label Description: Status of the request

  Example: Success, Failed, Dropped

- Label: `up_ep_key`

  Label Description: unique key to identify UPF XXX.XXX.XXX.XXX:YYY.YYY.YYY.YYY where XXX.XXX.XXX.XXX is Ip address of the NF service like SGW / SMF and YYY.YYY.YYY.YYY is the IP address of UPF

  Example: 192.168.10.2:192.168.20.3

- Label: `tx_type`

  Label Description: Type of the transanction for the route operation

  Example: NewTx, RetTx

Nodemgr userplane stats Category

nodemgr_up_stats

Description: Node Manager to userplane (UPF) link status up guage counters

Sample Query: `'nodemgr_up_stats{up_ep_key="192.168.10.2:192.168.20.3", primary_nodemgr_id="0", peer_nodemgr_id="0", interface_type="SXB"}'`

Labels:

- Label: `up_ep_key`

  Label Description: unique key to identify UPF XXX.XXX.XXX.XXX:YYY.YYY.YYY.YYY where XXX.XXX.XXX.XXX is Ip address of the NF service like SGW / SMF and YYY.YYY.YYY.YYY is the IP address of UPF

  Example: 192.168.10.2:192.168.20.3

- Label: `primary_nodemgr_id`

  Label Description: Current Node Manager instance Identifier of SGW/SMF service

Example: 0, 1, 2, 3, 4, 5, 6, 7, 8

- Label: `peer_nodemgr_id`

  Label Description: Peer Node Manager instance Identifier of UPF service

  Example: 0, 1, 2, 3, 4, 5, 6, 7, 8

- Label: `interface_type`

  Label Description: Interface type between current node (SMF/SGW) and Peer Node (UPF)

  Example: SXA, SXB, SXAB, SXC, N4

- Label: `gr_instance_id`

  Label Description: GR instance ID

  Example: 1, 2

SMF Recovery Value stats Category

smf_recovery_value

Description: SMF Recovery Value stats

Sample Query: `'smf_recovery_value{smf_ip="192.168.10.2"}'`

Labels:

- Label: `smf_ip`

  Label Description: smf ip address

  Example: 192.168.10.2

oam Metrics Reference

Node level LCI metric Category

node_lci_metric

Description: Total node level LCI metric

Sample Query: `node_lci_metric{}`

Node level OCI metric Category

node_oci_metric

Description: Total node level OCI reduction metric

Sample Query: `node_oci_metric{}`

Node level overload state Category

node_overload_status

Description: Total node level overload state

Sample Query: `node_overload_status{}`

protocol Metrics Reference

PFCP Decoded Messages Category

proto_pfcp_decode_msg_total

Description: Total number of pfcp decode by type,size

Sample Query: `'proto_pfcp_decode_msg_total{message_name="session_modification_res"}'`

Labels:

- Label: `message_name`

  Label Description: PFCP Message name

  Example: session_modification_res, session_report_req, session_deletion_res, heartbeat_res, heartbeat_req

- Label: `optimised`

  Label Description: PFCP Message decode optimised

  Example: true, false

- Label: `status`

  Label Description: PFCP Message status - accepted/denied/discarded

  Example: accepted, denied, discarded

- Label: `gr_instance_id`

  Label Description: GR instance ID

  Example: 1, 2

PFCP Encoded Messages Category

proto_pfcp_encode_msg_total

Description: Total number of pfcp encode by type,size

Sample Query: `'proto_pfcp_encode_msg_total{message_name="session_modification_req"}'`

Labels:

- Label: `message_name`

Label Description: PFCP Message name

Example: session_establishment_req, session_modification_req, session_report_req, session_deletion_req, heartbeat_req, heartbeat_res, session_report_res

- Label: `msgbufsize`

  Label Description: PFCP Message buffer size

  Example: little, jumbo, optimized

- Label: `status`

  Label Description: PFCP Message status - accepted/denied/discarded

  Example: accepted, denied, discarded

- Label: `gr_instance_id`

  Label Description: GR instance ID

  Example: 1, 2

radius-ep Metrics Reference

Radius COA and DM packet statistics Category

Radius_CoaDM_Requests_Current

Description: Current outstanding radius COA/DM requests

Sample Query:
`'Radius_CoaDM_Requests_Current{radSvrIp="1.1.1.1",radMsgCode="CoaReq",grInstId="1"}'`

Labels:

- Label: `radSvrIp`

  Label Description: Radius Server IP address

  Example: Any string

- Label: `radMsgCode`

  Label Description: Message type

  Example: DisconnectRequest, CoARequest

- Label: `grInstId`

  Label Description: GR Instance Id

  Example: 1 or 2

Radius_CoaDM_Requests_Statistics

Description: Total number of radius COA and DM packets sent and received

Sample Query:
`'Radius_CoaDM_Requests_Statistics{radSvrIp="1.1.1.1",radMsgCode="CoaRequest",grInstId="1"}'`

Labels:

- Label: `radSvrIp`

  Label Description: Radius Server IP address

  Example: Any string

- Label: `radMsgCode`

  Label Description: Message type

  Example: DisconnectRequest, DisconnectACK, DisconnectNAK, CoARequest, CoaDMReq, CoAACK

- Label: `radPacketType`

  Label Description: Direction

  Example: Tx, Rx

- Label: `radResult`

  Label Description: Result

  Example: Success, Failure_Invalid_Request

- Label: `grInstId`

  Label Description: GR Instance Id

  Example: 1 or 2

Radius Server status Category

Radius_Server_Status

Description: Display active/inactive status of radius-server

Sample Query:
`'Radius_Server_Status{radSvrIp="1.1.1.1",radSvrPort="1812",radSvrPortType="Auth"}'`

Labels:

- Label: `radSvrIP`

  Label Description: Radius Server IP address

  Example: Any string

- Label: `radSvrPort`

  Label Description: Radius Server Port

  Example: Any string

- Label: `radSvrPortType`

  Label Description: Type of server

  Example: Auth, Acct

Radius packet statistics Category

Radius_requests_current

Description: Current outstanding radius requests

Sample Query:
`'Radius_requests_current{radSvrIp="1.1.1.1",radSvrPort="1812",radSvrPortType="Auth",grInstId="1"}'`

Labels:

- Label: `radSvrIp`

  Label Description: Radius Server IP address

  Example: Any string

- Label: `radSvrPort`

  Label Description: Radius Server Port

  Example: Any string

- Label: `radSvrPortType`

  Label Description: Type of server

  Example: Auth, Acct

- Label: `radMsgCode`

  Label Description: Message type

  Example: SecondaryAuthenReq, RadiusAcctReq, TestAuth, TestAcct

- Label: `radPacketType`

  Label Description: Direction

  Example: Tx, Rx

- Label: `dnn`

  Label Description: DNN of session

  Example: Any string

- Label: `procType`

  Label Description: Procedure type

  Example: Any string

- Label: `ratType`

  Label Description: RAT Type

  Example: Any string

- Label: `sessType`

  Label Description: Session type

  Example: Any string

- Label: `grInstId`

  Label Description: GR Instance Id

  Example: 1 or 2

Radius_requests_statistics

Description: Total number of radius packets sent and received

Sample Query:

```
'Radius_requests_statistics{radSvrIp="1.1.1.1",radSvrPort="1812",radSvrPortType="Auth",grInstId="1"}'
```

Labels:

- Label: `radSvrIp`

  Label Description: Radius Server IP address

  Example: Any string

- Label: `radSvrPort`

  Label Description: Radius Server Port

  Example: Any string

- Label: `radSvrPortType`

  Label Description: Type of server

  Example: Auth, Acct

- Label: `radMsgCode`

  Label Description: Message type

  Example: SecondaryAuthenReq, RadiusAcctReq, TestAuth, TestAcct

- Label: `radPacketType`

  Label Description: Direction

  Example: Tx, Retry_Tx, Rx

- Label: `radResult`

  Label Description: Result

  Example: Success, Timeout, Failure_Reject, Failure_NoServer

- Label: `dnn`

  Label Description: DNN of session

  Example: Any string

- Label: `procType`

  Label Description: Procedure type

  Example: Any string

- Label: `ratType`

Label Description: RAT Type

Example: Any string

- Label: `sessType`

  Label Description: Session type

  Example: Any string

- Label: `grInstId`

  Label Description: GR Instance Id

  Example: 1 or 2

rest-ep Metrics Reference

Discover Messages statistics Category

nf_discover_messages_total

Description: Discover Messages statistics

Sample Query: `nf_discover_messages_total{nf_type=\"udm\", host=\"10.105.227.109:8082\", svc_name=\"nudm-sdm\", version=\"v1\", result=\"timeouOrRPCError\"}`

Labels:

- Label: `nf_type`

  Label Description: Network Function type

  Example: nrf, udm, amf, pcf, chf, ciscocontrol

- Label: `host`

  Label Description: End Point address

  Example: 10.105.227.109:8082

- Label: `svc_name`

  Label Description: Network function service name

  Example: nudm-sdm, namf-comm

- Label: `version`

  Label Description: Api version info

  Example: v1, v2,

- Label: `result`

  Label Description: result of discover message

  Example: 200, 201, 204, success, timeout_rpc_error, response_parse_failure, empty_response

Discover Messages Time statistics Category

nf_discover_total_time

Description: Discover Messages Total time statistics

Sample Query: `nf_discover_total_time{nf_type=\"udm\", host=\"10.105.227.109:8082\", svc_name=\"nudm-sdm\", version=\"v1\", result=\"timeouOrRPCError\"}`

Labels:

- Label: `nf_type`

  Label Description: Network Function type

  Example: nrf, udm, amf, pcf, chf, ciscocontrol

- Label: `host`

  Label Description: End Point address

  Example: 10.105.227.109:8082

- Label: `svc_name`

  Label Description: Network function service name

  Example: nudm-sdm, namf-comm

- Label: `version`

  Label Description: Api version info

  Example: v1, v2,

- Label: `result`

  Label Description: result of discover message

  Example: 200, 201, 204, success, timeout_rpc_error, response_parse_failure

FQDN resolution statistics Category

fqdn_dns_msg_total

Description: Number of FQDN attempted/resolved/failed

Sample Query: `fqdn_dns_msg_total{fqdn=\"host.example.com\", nf_type=\"udm\", svc_name=\"nudm-sdm\", status=\"attempted\", req=\"resolution\", error=\"8314\"}`

Labels:

- Label: `fqdn`

  Label Description: End Point hostname

  Example: host.example.com

- Label: `nf_type`

  Label Description: Network Function type

Example: nrf, udm, amf, pcf, chf, ciscocontrol

- Label: `svc_name`

  Label Description: Network function service name

  Example: nudm-sdm, namf-comm

- Label: `status`

  Label Description: resolution status

  Example: attempted, success, failure

- Label: `req`

  Label Description: whether resolved from dns-server or read from cache

  Example: resolution, cache

- Label: `error`

  Label Description: result of fqdn

  Example: 8311, 8312, 8314,

NF End point selections Category

nf_endpoint_selections_total

Description: NF End Point Selection Statistics

Sample Query: `nf_endpoint_selections_total{nf_type=\"udm\", host=\"10.105.227.109:8097\", svc_name=\"nudm-sdm\", version=\"v1\", req=\"initial\", retransmit=\"retransmit_1\", peer_input=\"yes\", gr_instance_id=\"1\"}`

Labels:

- Label: `nf_type`

  Label Description: Network Function type

  Example: nrf, udm, amf, pcf, chf, ciscocontrol

- Label: `host`

  Label Description: End Point address

  Example: 10.105.227.109:8097

- Label: `svc_name`

  Label Description: Network function service name

  Example: nudm-sdm, namf-comm

- Label: `version`

  Label Description: Api version info

  Example: v1, v2,

- Label: `req`

  Label Description: req type

  Example: initial, fallback,

- Label: `retransmit`

  Label Description: retransmit count

  Example: retransmit_1, retransmit_2,

- Label: `peer_input`

  Label Description: peer input

  Example: yes, no,

- Label: `gr_instance_id`

  Label Description: GRInstanceId

  Example: 0, 1

NF failure handling stats Category

nf_failure_handling_stats_total

Description: NF Failure handling stats

Sample Query: `nf_failure_handling_stats_total{nf_type=\"udm\", host=\"10.105.227.109:8097\", svc_name=\"nudm-sdm\", version=\"v1\", message_type=\"UdmUecmRegisterSMF\",req=\"initial\", response=\"202\", status=\"final\", peer_input=\"yes\", failure_source=\"SCP\"}`

Labels:

- Label: `nf_type`

  Label Description: Network Function type

  Example: nrf, udm, amf, pcf, chf, ciscocontrol

- Label: `host`

  Label Description: End Point address

  Example: 10.105.227.109:8097

- Label: `svc_name`

  Label Description: Network function service name

  Example: nudm-sdm, namf-comm, nnrf-disc

- Label: `version`

  Label Description: Api version info

  Example: v1, v2,

- Label: `message_type`

  Label Description: Message Type

Example: UdmUecmRegisterSMF, UdmSdmGetUESMSubscriptionData, DiscoverNF

- Label: `req`

  Label Description: Request type

  Example: initial, fallback,

- Label: `response`

  Label Description: Response from the server

  Example: 200, 201, 204, timeout_rpc_error,

- Label: `status`

  Label Description: Status from the server

  Example: retry, final, retry-next, blocklist

- Label: `peer_input`

  Label Description: peer input

  Example: yes, no

- Label: `failure_source`

  Label Description: Source Network Function type from where failure originated

  Example: SCP, SEPP, UDM, PCF, CHF, AMF

NF management messages statistics Category

nf_management_stats_total

Description: NF management messages statistics

Sample Query: `nf_management_stats_total{host=\"10.105.227.109:8082\", svc_name=\"nnrf-nfm\", version=\"v1\",direction=\"outbound\", message_type=\"registration\",result=\"timeouOrRPCError\" }`

Sample Query: `nf_management_stats_total{host=\"10.105.227.109:8082\", svc_name=\"nnrf-nfm\", version=\"v1\",direction=\"inbound\", message_type=\"notification\", result=\"success\", notification_event_type=\"NF_REGISTERED\", condition_event_type=\"NF_ADDED\", gr_instance_id=\"0\" }`

Labels:

- Label: `host`

  Label Description: End Point address

  Example: 10.105.227.109:8082

- Label: `svc_name`

  Label Description: Network function service name

  Example: nudm-sdm, namf-comm

- Label: `version`

Label Description: Api version info

Example: v1, v2,

- Label: `direction`

  Label Description: Direction indicates about the message going out or coming in

  Example: inbound, outbound

- Label: `message_type`

  Label Description: Type of Message

  Example: registration, hearbeat, subscription, notification

- Label: `notification_event_type`

  Label Description: Notification Event Type

  Example: NF_REGISTERED, NF_DEREGISTERED, NF_PROFILE_CHANGED

- Label: `condition_event_type`

  Label Description: Condition Event Type

  Example: NF_ADDED, NF_REMOVED

- Label: `result`

  Label Description: result of registration/subscription/notification message

  Example: 200, 201, 204, success, timeout_rpc_error, response_parse_failure

- Label: `gr_instance_id`

  Label Description: GRInstanceId

  Example: 0, 1

NF management message time statistics Category

nf_management_total_time

Description: NF management messages total time taken

Sample Query: `nf_management_total_time{host=\"10.105.227.109:8082\", svc_name=\"nudm-sdm\", version=\"v1\",direction=\"outbound\", message_type=\"registration\",result=\"timeouOrRPCError\" }`

Labels:

- Label: `host`

  Label Description: End Point address

  Example: 10.105.227.109:8082

- Label: `svc_name`

  Label Description: Network function service name

  Example: nudm-sdm, namf-comm

- Label: `version`

  Label Description: Api version info

  Example: v1, v2,

- Label: `direction`

  Label Description: Direction indicates about the message going out or coming in

  Example: inbound, outbound

- Label: `message_type`

  Label Description: Type of Message

  Example: registration, hearbeat, subscription, notification

- Label: `result`

  Label Description: result of discover message

  Example: 200, 201, 204, success, timeout_rpc_error, response_parse_failure, request_parse_failure, invalid_notify_event, invalid_nf_instance_uri, internal_error

NF Send messages statistics Category

nf_req_recieved_messages_total

Description: NF recieved messages to NRF client library

Sample Query: `nf_req_recieved_messages_total{nf_type=\"udm\", svc_name=\"nudm-sdm\", message_type=\"UdmUecmRegisterSMF\"}`

Labels:

- Label: `nf_type`

  Label Description: Network Function type

  Example: udm, amf, pcf, chf, ciscocontrol

- Label: `svc_name`

  Label Description: Network function service name

  Example: nudm-sdm, namf-comm

- Label: `message_type`

  Label Description: Message Type

  Example: UdmUecmRegisterSMF, UdmSdmGetUESMSubscriptionData

nf_resp_sent_messages_total

Description: NF message responses sent from NRF client library

Sample Query: `nf_resp_sent_messages_total{nf_type=\"udm\", svc_name=\"nudm-sdm\", message_type=\"UdmUecmRegisterSMF\",result=\"SendSuccess\", status_code=\"200\"}`

Labels:

- Label: `nf_type`

  Label Description: Network Function type

  Example: nrf, udm, amf, pcf, chf, ciscocontrol

- Label: `svc_name`

  Label Description: Network function service name

  Example: nudm-sdm, namf-comm

- Label: `message_type`

  Label Description: Message Type

  Example: UdmUecmRegisterSMF, UdmSdmGetUESMSubscriptionData

- Label: `result`

  Label Description: result of discover message

  Example: SendSuccess, SendFailure

- Label: `status_code`

  Label Description: result of NF send message

  Example: 200, 201, 204,

nf_send_message_total_time

Description: NF send message total time taken

Sample Query: `nf_send_message_total_time{nf_type=\"udm\", svc_name=\"nudm-sdm\", message_type=\"UdmUecmRegisterSMF\",result=\"SendSuccess\", status_code=\"200\"}`

Labels:

- Label: `nf_type`

  Label Description: Network Function type

  Example: nrf, udm, amf, pcf, chf, ciscocontrol

Labels:

- Label: `svc_name`

  Label Description: Network function service name

  Example: nudm-sdm, namf-comm

- Label: `message_type`

  Label Description: Message Type

  Example: UdmUecmRegisterSMF, UdmSdmGetUESMSubscriptionData

- Label: `result`

  Label Description: result of discover message

Example: SendSuccess, SendFailure

- Label: `status_code`

  Label Description: result of NF send message

  Example: 200, 201, 204,

NRF Discovery Category

nf_discover_events_total

Description: NF Discover Stats

Sample Query: `nf_discover_events_total{nf_type=\"pcf\", response_type=\"local\"}`

Labels:

- Label: `nf_type`

  Label Description: Network Function type

  Example: nrf, udm, amf, pcf, chf, ciscocontrol

- Label: `response_type`

  Label Description: Discovery response choosen from

  Example: local, cache, expired-cache, intervaltree-cache

NRF subscription messages statistics Category

nrf_subscription_send_messages_total

Description: NRF Subscription send messages total

Sample Query: `nrf_subscription_send_messages_total{host=\"10.105.227.109:8082\", message_type=\"subscription\", req=\"initial\", gr_instance_id=\"1\", subscrCond=\"NFSetId\"}`

Labels:

- Label: `host`

  Label Description: End Point address

  Example: 10.105.227.109:8082

- Label: `message_type`

  Label Description: subscription message typwe

  Example: unsubscription,subscription,updateSubscription

- Label: `req`

  Label Description: req type

  Example: resourceUri, initial,retry_2

- Label: `gr_instance_id`

  Label Description: GRInstanceId

  Example: 0, 1

- Label: `subscrCond`

  Label Description: sunscription condition

  Example: NFInstanceId, NFSetId

Range based discovery statistics Category

nf_rangebased_discovery_total

Description: Range based discovery operation on IntervalTree cache stats

Sample Query: `nf_rangebased_discovery_total{range_type=\"supi\", nf_type=\"UDM\", operation=\"add\", message_type=\"discovery\", status=\"success\"}`

Labels:

- Label: `range_type`

  Label Description: range type used in discovery-filter

  Example: supi, gpsi

- Label: `nf_type`

  Label Description: Network Function type

  Example: udm, pcf, chf, ausf, udr

- Label: `operation`

  Label Description: operation performed in IntervalTree cache

  Example: add, update, remove

- Label: `message_type`

  Label Description: which triggered operation on IntervalTree cache

  Example: discovery, cache-refresh, cache-expiry, profile-removal, notification

- Label: `status`

  Label Description: status of the operation

  Example: success, failure

REST EP message Exchange Time Category

smf_restep_http_msg_seconds

Description: SMF REST time between request and response messages

Sample Query: `'smf_restep_http_msg_seconds{message_direction="inbound",nf_type="amf"}'`

Labels:

- Label: `nf_type`

  Label Description: Network Function type

  Example: nrf, udm, amf, pcf, chf, ciscocontrol

- Label: `message_direction`

  Label Description: direction of message from SMF perspective

  Example: inbound, outbound

- Label: `api_name`

  Label Description: API name

  Example: register_ue, deregister_ue, subscription_req, sdm_subscription_req, sdm_data_change_notify, nf_registration, nf_discovery, slice_selection, amf_create_sm_context, amf_update_sm_context, amf_release_sm_context, amf_n1_n2_transfer, amf_n1_n2_transfer_notify_failure, amf_assign_ebi, amf_status_notify, pcf_sm_policy_control_create, chf_charging_data_request, pcf_sm_policy_control_update, pcf_sm_policy_control_delete, pcf_sm_policy_control_update_notify, cisco_control_clear_subscriber, cisco_control_show_subscriber, pcf_sm_policy_control_terminate_notify, chf_abort_notify

- Label: `nf_uri`

  Label Description: Network Function URI

  Example: actual HTTP URI of the message

- Label: `response_status`

  Label Description: HTTP response status code

  Example: 200, 201, 204

- Label: `response_cause`

  Label Description: HTTP response cause code

  Example: cause string as received from peer nf

REST EP messages Category

smf_restep_http_msg_total

Description: SMF REST message counter

Sample Query: `'smf_restep_http_msg_total{message_direction="inbound",nf_type="amf"}'`

Labels:

- Label: `nf_type`

  Label Description: Network Function type

  Example: nrf, udm, amf, pcf, chf, ciscocontrol

- Label: `message_direction`

  Label Description: direction of message from SMF perspective

  Example: inbound, outbound

- Label: `api_name`

  Label Description: API name

  Example: register_ue, deregister_ue, subscription_req, sdm_subscription_req, sdm_data_change_notify, nf_registration, nf_discovery, slice_selection, amf_create_sm_context, amf_update_sm_context, amf_release_sm_context, amf_n1_n2_transfer, amf_n1_n2_transfer_notify_failure, amf_assign_ebi, amf_status_notify, pcf_sm_policy_control_create, chf_charging_data_request, pcf_sm_policy_control_update, pcf_sm_policy_control_delete, pcf_sm_policy_control_update_notify, cisco_control_clear_subscriber, cisco_control_show_subscriber, pcf_sm_policy_control_terminate_notify, chf_abort_notify

- Label: `nf_uri`

  Label Description: Network Function URI

  Example: actual HTTP URI of the message

- Label: `response_status`

  Label Description: HTTP response status code

  Example: 200, 201, 204

REST EP messages Decode Status Category

smf_restep_http_msg_decode

Description: SMF REST number of decoding failures

Sample Query:
`'smf_restep_http_msg_decode{nf_type="amf",api_name="register_ue",decoding_status="decoding_failure"}'`

Labels:

- Label: `nf_type`

  Label Description: Network Function type

  Example: nrf, udm, amf, pcf, chf, ciscocontrol

- Label: `api_name`

  Label Description: API name

  Example: register_ue, deregister_ue, subscription_req, sdm_subscription_req, sdm_data_change_notify, nf_registration, nf_discovery, slice_selection, amf_create_sm_context, amf_update_sm_context, amf_release_sm_context, amf_n1_n2_transfer, amf_n1_n2_transfer_notify_failure, amf_assign_ebi, amf_status_notify, pcf_sm_policy_control_create, chf_charging_data_request, pcf_sm_policy_control_update, pcf_sm_policy_control_delete, pcf_sm_policy_control_update_notify, cisco_control_clear_subscriber, cisco_control_show_subscriber, pcf_sm_policy_control_terminate_notify, chf_abort_notify

- Label: `decoding_status`

  Label Description: Decoding status

  Example: decoding_failure

- Label: `interface_type`

  Label Description: Iterface Type

  Example: N11, N1, N2

- Label: `response_status`

  Label Description: HTTP response status code

  Example: 200, 201, 204

- Label: `application_error`

  Label Description: Application error

Charging final unit indication statistics Category

chf_recieved_fui_stats

Description: Statistics for final unit indication with final unit action

Sample Query: `'sum (chf_recieved_fui_stats{interface_type="Gy"})'`

Labels:

- Label: `chf_type`

  Label Description: Type of CHF with which message is exchanged

  Example: online, offline

- Label: `interface_type`

  Label Description: Type of Interface communicate with PGW

  Example: N40, Gy

- Label: `fua_type`

  Label Description: Type of final unit action

  Example: FinalUnitActionType_TERMINATE, FinalUnitActionType_REDIRECT, FinalUnitActionType_RESTRICT_ACCESS

CHF Notification Statistics Category

smf_chf_notification_stats

Description: SMF Charging CHF Notification stats

Sample Query: `'smf_chf_notification_stats{notification_type="reauthorization"}'`

Labels:

- Label: `notification_type`

  Label Description: Type of notification request

  Example: reauthorization, abort_charging

- Label: `dnn`

  Label Description: DNN for which the flow is created

  Example: cisco.com

- Label: `status`

  Label Description: Status of notify message processing

  Example: attempted, success, failures

- Label: `rat_type`

  Label Description: RAT type on which the flow is created

  Example: EUTRA, NR, WLAN, VIRTUAL, rat_type_unknown

- Label: `reason`

  Label Description: Reason for notify message failure

  Example: pdu_session_not_established, charging_failed, offline_converted

udp-proxy Metrics Reference

UDP-Proxy BGP Routes Count Category

upd_proxy_bgp_routes_count

Description: UDP Proxy BGP routes added count

Sample Query: `'upd_proxy_bgp_routes_count{service_name="udp-proxy", status="success"}'`

Labels:

- Label: `status`

  Label Description: Status of message while sending or receiving

  Example: success, failed

UDP-Proxy messages Category

udp_proxy_msg_total

Description: UDP Proxy message counters being recieved or sent

Sample Query: `'udp_proxy_msg_total{message_name="radius_request", message_direction="inbound", status="success"}'`

Labels:

- Label: `message_name`

  Label Description: UDP messages coming via udp-proxy service

  Example: radius_request, radius_response, heartbeat_request, heartbeat_response

- Label: `message_direction`

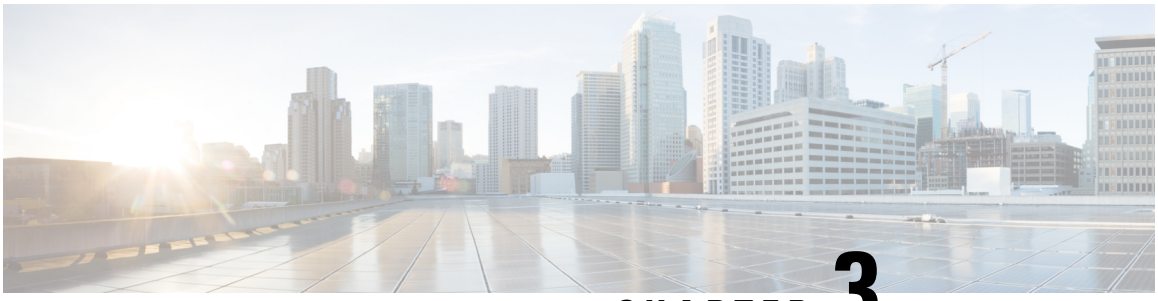
  Label Description: Message being sent or being received

  Example: inbound, outbound

- Label: `status`

  Label Description: Status of message while sending or receiving

  Example: success, failed



CHAPTER 3

Failure Disconnect Reasons Reference

SMF Disconnect Reasons

This section describes the procedure failure disconnect reasons supported on SMF.

The following table provides the descriptions for the key failure disconnect reasons.

Table 1: Failure Disconnect Reasons

Disconnect Reason	Description
disc_chf_reconciliation	The total number of sessions released by the SMF due to CHF reconciliation.
disc_sess_report_erir_pdn_sess_rel	The total number of 4G or Wi-Fi sessions released by the SMF due to N4 Session Report Request from UPF with ERIR report type. If the ERIR delay timer is configured under access profile, the configured value delays the N4 Session Report Request handling.
disc_pdusetup_create_over_create	The total number of ongoing 5G sessions rejected by the SMF when 5G session establishment is received while handling N11SmContextCreateRequest for 5G session establishment (Create over Create case).
disc_pdurel_amf_init_release_404	The total number of 5G sessions released by the SMF due to 404 response from AMF for N1N2Transfer Request during 5G session modification.
disc_sess_report_erir_pdn_sess_rel	The total number of 4G or Wi-Fi sessions released by the SMF due to N4 Session Report Request from UPF with ERIR report type. If the ERIR delay timer is configured under access profile, the configured value delays the N4 Session Report Request handling.

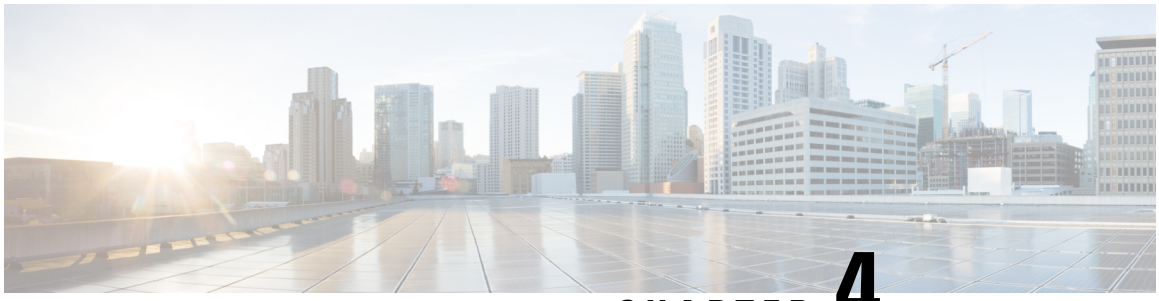
Disconnect Reason	Description
disc_pdusetup_create_over_create	The total number of ongoing 5G sessions rejected by the SMF when 5G session establishment is received while handling N11SmContextCreateRequest for 5G session establishment (Create over Create case).
disc_pdurel_amf_init_release_404	The total number of 5G sessions released by the SMF due to 404 response from AMF for N1N2Transfer Request during 5G session modification.
disc_pduim_context_not_found	The total number of 5G sessions released by the SMF due to 404 response from AMF for N1N2Transfer Request during idle to active mobility and vice versa.
disc_pdnsetup_smf_mop_offline	The total number of 4G or Wi-Fi sessions rejected by the SMF due to Session Create received when SMF is in maintenance mode and when the offline mode configuration is set in the SMF profile or specifically for a DNN in the DNN profile.
disc_pdusetup_n2_setup_failed	The total number of 5G sessions rejected by the SMF when N2_PDU_SESSION_RESOURCE_SETUP_ UNSUCCESS_TRANSFER is received from AMF indicating the N2 failure during 5G session establishment.
disc_pdusetup_n1n2_transfer_ rsp_failure	The total number of 5G sessions rejected by the SMF due to N11N1N2MessageTransferFailure response from AMF during 5G session setup.
disc_pdnsetup_non5gcapableue_ not_allowed	The total number of 4G or Wi-Fi sessions rejected by the SMF due to Session Create received without 5G InterWorking (IWK_5GS) indication and when the DNN profile is configured to support only NR capable UE by setting **only-nr-capable-ue** to true.
disc_pdnsetup_udm_sub_fetch_failed	The total number of sessions rejected by the SMF due to failure in fetching the session management subscription data (sm-data) from UDM during 4G or Wi-Fi session establishment time. This disconnect reason is pegged in the following scenarios: • SMF request to UDM for fetching the session management subscription data (sm-data) fails. • SMF receives failure response from UDM for SM data request. • Validation of request from UE (SSC mode, PDU session type and Snssai) fails against the subscription allowed based on UDM response.

Disconnect Reason	Description
disc_pdnsetup_udm_sub_fetch_ resp_failed	The total number of sessions rejected by the SMF due to failure in fetching the session management subscription data (sm-data) from UDM during 4G or Wi-Fi session establishment time. This disconnect reason is pegged in the following scenarios: • SMF receives failure response from UDM for session management subscription data (sm-data) request. • Validation of request from UE (SSC mode, PDU session type, and Snssai) fails against the subscription allowed based on UDM response.
disc_pdusetup_release_over_create	The total number of 5G sessions rejected by the SMF due to 5G session release event during ongoing 5G session establishment.
disc_pdusetup_pdu_sess_ does_not_exist	The total number of 5G sessions rejected by the SMF when SmContextCreateRequest is received with RequestType as EXISTING_PDU_SESSION during Wi-Fi to 5G handover, but the session doesn't exist with SMF.
disc_sess_cp_idle_time_exp_release	The total number of 4G or Wi-Fi sessions released by the SMF due to Control Plane (CP) idle timeout that started on successful session establishment. The idle timeout is configured in the DNN profile.
disc_sgw_ctx_failure	The total number of 4G or Wi-Fi sessions rejected by the SMF due to default flow failure caused by S-GW.
disc_pdnsetup_pcf_create_ resp_failed	The total number of 4G or Wi-Fi sessions rejected by the SMF due to PCF Create Failure during 4G or Wi-Fi session establishment.
disc_gtpc_peer_pathfail	The total number of 4G or Wi-Fi sessions released by the SMF due to GTPC path failure in the network.
disc_pdusetup_rm_ exchg_failure	The total number of 5G sessions rejected by the SMF due to IP allocation failure for the PDU session during 5G session setup.
disc_rel_chf_err	The total number of sessions released by the SMF due to CHF-initiated session release.
disc_pdnsetup_udm_reg_ resp_failed	The total number of sessions rejected by the SMF due to SMF registration failure with UDM during 4G or Wi-Fi session establishment time. The SMF sends registration request to UDM for storing UE context management information. This disconnect reason is pegged in the following scenarios: • SMF registration request to UDM fails. • SMF receives failure response from UDM for registration request.

Disconnect Reason	Description
disc_pdumodify_context_not_found	The total number of 5G sessions released by the SMF due to 404 response from AMF for N1N2Transfer Request during 5G session modification.
disc_pdusetup_sm_cxt_sess_id_err	The total number of sessions rejected by the SMF when pduSessionId in 5G PDU Session Establishment Request (N11SmContextCreate Request) is either zero or not in the expected format. This disconnect reason is also pegged when there is no subscriber ID (SUPI or PEI) but the ueEpsPdnConnection parameter is present in the request.
disc_pdusetup_upf_setup_rsp_failure	The total number of sessions rejected by the SMF when N4 session establishment with UPF fails during 5G session establishment time.
disc_pdusetup_sess_cp_idle_timeout	The total number of PDN sessions released by the SMF due to Control Plane (CP) idle timer expiry. The CP idle timer expires when there is no control plane activity within the CP idle timeout.
disc_pdusetup_ip_alloc_failed	The total number of sessions rejected by the SMF or PGW-C when IP address allocation fails. This disconnect reason is pegged in the following scenarios: • SMF service (node manager) which handles IP address allocation is down. • SMF service (node manager) couldn't allocate the IP address of the requested PDU session type.
disc_pdusetup_n1n2_transfer_ exchg_failure	The total number of sessions rejected by the SMF when there is failure in N1N2 Transfer Request with AMF during 5G PDU session establishment.
disc_pdnsetup_resource_mgr_ exchg_failed	The total number of sessions rejected by the SMF or PGW-C when resource manager exchange fails due to IP address allocation failure during 4G or Wi-Fi PDN connection time.
disc_pdusetup_pcf_create_rsp_failure	The total number of sessions rejected by the SMF due to Policy Create Failure. This disconnect reason is pegged in the following scenarios: • SMF receives failure from PCF for Policy Create Request during 5G session establishment. • No response from PCF for Policy Create Failure.

Disconnect Reason	Description
disc_pdnsetup_csr_invalid	The total number of Create Session Requests rejected by the SMF when Create Session Request includes invalid parameters. This disconnect reason is pegged in the following scenarios: • Create Session Request with invalid parameters for new PDN connection (4G or Wi-Fi). • Create Session Request with invalid parameters in handover requests—5G to Wi-Fi HO, 4G to Wi-Fi HO, and Wi-Fi to 4G HO.
disc_n26_4g_5g_ho_n4_modify_failed	The total number of sessions released by the SMF or PGW-C when N4 modification with UPF fails in the execution phase of 4G to 5G N26 handover. This disconnect reason is pegged in the following scenarios: • N4 Modification Request failure in the execution phase of N26 HO. • SMF receives failure response from UPF for N4 modification in the execution phase of N26 HO. • SLA timeout at SMF during N4 modification in the execution phase of N26 HO.
disc_sess_cp_idle_time_exp_release	The total number of PDN sessions released by the SMF or PGW-C due to Control Plane (CP) idle timer expiry. The CP idle timer expires when there is no control plane activity within the CP idle timeout.
disc_pdusetup_dnn_not_supported_ in_slice	The total number of sessions rejected by the SMF where the 5G PDU Session Establishment Request (N11smContextCreate) received from AMF contains DNN which is not supported in the requested network slice.
disc_pdusetup_udm_reg_failed	The total number of sessions rejected by the SMF due to SMF registration failure with UDM during 5G session establishment time. The SMF sends registration request to UDM for storing UE context management information. This disconnect reason is pegged in the following scenarios: • SMF registration request to UDM fails. • SMF receives failure response from UDM for registration request.
disc_pdurel_db_conflict	The total number of sessions released by the SMF due to internal issue related to the database conflict.

Disconnect Reason	Description
disc_pdusetup_udm_sub_fetch_ resp_failed	The total number of sessions rejected by the SMF due to failure in fetching the session management subscription data (sm-data) from UDM during 5G session establishment time. This disconnect reason is pegged in the following scenarios: • SMF receives failure response from UDM for session management subscription data (sm-data) request. • Validation of request from UE (SSC mode, PDU session type and Snssai) fails against the subscription allowed based on UDM response.
disc_pdusetup_udm_sub_fetch_failure	The total number of sessions rejected by the SMF due to failure in fetching the session management subscription data (sm-data) from UDM during 5G session establishment time. This disconnect reason is pegged in the following scenarios: • SMF request to UDM for fetching the session management subscription data (sm-data) fails. • SMF receives failure response from UDM for SM data request. • Validation of request from UE (SSC mode, PDU session type, and Snssai) fails against the subscription allowed based on UDM response.
disc_pdnsetup_secondary_auth_ip_addr_conflict	The total number of 4G sessions released by the SMF when the static UE IP address received from access side does not match the Framed IP address received in RADIUS Authentication response.
disc_pdnsetup_pcrf_create_failed	The total number of 4G sessions released by the SMF when the initial request to the PCRF server cannot be sent during session setup.
disc_pdnsetup_pcrf_create_resp_failed	The total number of 4G sessions released by the SMF when the processing of the response of the initial request to the PCRF server fails, resulting in session termination.
disc_pdnsetup_charging_create_resp_failed	The total number of 4G sessions released by the SMF when fetching the response to the initial charging request fails, resulting in session termination.
disc_pdusetup_static_duplicate_ip	The total number of 5G sessions that are rejected by SMF due to detection of duplicate static IP.
disc_pdnsetup_static_duplicate_ip	The total number of 4G or Wi-Fi sessions that are rejected by SMF due to detection of duplicate static IP.



CHAPTER 4

MIB Reference

CISCO-CNEE-MIB

This is the MIB module for the Cisco Cloud Native Execution Environment (CNEE) platform. This MIB only handles notifications from the CNEE.

Note The Cisco Cloud Native Execution Environment (CNEE) MIB (CISCO-CNEE-MIB.my) uses definitions that are defined in the Cisco Enterprise Structure of Management Information (SMI) MIB (CISCO-SMI.my).

For more information, see the "*UCC Subscriber Microservice Infrastructure - Operations Guide*" > *SMI MIB Reference* chapter.

CISCO-SMI

This is the Structure of Management Information for the Cisco Enterprise.

Note The Cisco Cloud Native Execution Environment (CNEE) MIB (CISCO-CNEE-MIB.my) uses definitions that are defined in the Cisco Enterprise Structure of Management Information (SMI) MIB (CISCO-SMI.my).

For more information, see the "*UCC Subscriber Microservice Infrastructure - Operations Guide*" > *SMI MIB Reference* chapter.