Ultra Cloud Core 5G Policy Control Function, Release 2021.02 - Statistics Reference

First Published: 2021-05-03

C O N T E N T S

About this Guide

This preface describes the *Ultra Cloud Core 5G Policy Control Function Statistics Reference*, the document conventions, and the customer support details.

- Conventions Used, on page v
- Contacting Customer Support, on page vi

Conventions Used

The following tables describe the conventions used throughout this documentation.

Notice Type	Description
Information Note	Provides information about important features or instructions.
Caution	Alerts you of potential damage to a program, device, or system.
Warning	Alerts you of potential personal injury or fatality. May also alert you of potential electrical hazards.

Typeface Conventions	Description
Text represented as a `screen display`	This typeface represents displays that appear on your terminal screen, for example: `Login:`
Text represented as **commands**	This typeface represents commands that you enter, for example: **show ip access-list** This document always gives the full form of a command in lowercase letters. Commands are not case sensitive.

Typeface Conventions	Description
Text represented as a **command** *variable*	This typeface represents a variable that is part of a command, for example: **show card** *slot_number* *slot_number* is a variable representing the desired chassis slot number.
Text represented as menu or sub-menu names	This typeface represents menus and sub-menus that you access within a software application, for example: Click the **File** menu, then click **New**

Contacting Customer Support

Use the information in this section to contact customer support.

Refer to the support area of http://www.cisco.com for up-to-date product documentation or to submit a service request. A valid username and password are required to access this site. Please contact your Cisco sales or service representative for additional information.

Prometheus and Grafana

Feature Summary

Summary Data

Table 1: Summary Data

Applicable Product(s) or Functional Area	5G-PCF
Applicable Platform(s)	SMI
Feature Default Setting	Enabled – Always-on
Related Changes in this Release	Not Applicable
Related Documentation	Not Applicable

Feature Description

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

The PCF 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 PCF 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 PCF 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.

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 PCF 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/.

Managing the PCF Statistics

This section describes how to view statistics within PCF.

Managing the PCF statistics involves the following:

Viewing the Statistics

This section describes how to view the statistics information.

1. On the system where PCF is deployed, navigate to the following URL:

   ```
   https://docs.namespace-product-documentation.IP_address.nip.io/
   ```

   All the PCF-specific statistics and other generic statistics such as system-statistics derived from the SMI deployer get displayed on the HTML page.

Accessing the Grafana Dashboard

This section describes how to access Grafana to view the visual representation of KPIs.

1. On the system where PCF is deployed, navigate to the following URL to view the dashboard:

   ```
   https://grafana.smi-cnat-monitoring.IP_address.nip.io
   ```

2. Enter the administrative user's username and password.

For more information on Grafana's capabilities, consult the Grafana documentation available at http://docs.grafana.org.

Viewing the PCF Dashboard

This section how to view the PCF dashboard.

1. On the PCF Application dashboard, in the left pane, click the dashboard icon to open the menu and select **Manage**.

2. In the **Manage** tab, click the *namespace* folder.

   The folder name resembles the namespace in which PCF is installed. The available dashboards are listed.

3. Click **PCF Application**.

   The PCF Application dashboard displays the graphs. You can shuffle the location of the graphs by dragging the panels.

Note Cisco recommends configuring the panel options in the Grafana dashboard. With this option, you can view only the required graphs when the dashboard is loaded.

Running a Query in Grafana

This section describes how to execute a query in Grafana.

The PCF Dashboard creates a panel containing the graph that is based on the query that it ingests. Grafana brings up a panel to visualize data that is retrieved for one or more queries. You can run canned and custom queries from the dashboard. The canned queries are preexisting in the dashboard with the define syntax. Custom queries permit you to formulate queries that return specific information.

1. On the PCF Application Dashboard, in the left pane, click the explore icon to open the menu. On hovering over the icon, the tooltip text appears as **Explore**.

2. In the **Explore** pane, click the drop-down to choose the data source as **Prometheus**.

3. Do one of the following:

   • To execute a built-in query, click the **Metrics** drop-down and choose the query that you want to run.

   • To execute a custom query, enter the query in the corresponding field next to **Metrics**.

4. Click **Run Query**.

   The query retrieves the information from Prometheus and displays it in a graphical representation.

Configuring Autorefresh

This section describes how to configure autorefresh to ensure that you view the recent information on Grafana.

1. On the PCF Application dashboard, click the gear on the top-right corner to open the **Settings**. On hovering over this icon, the tooltip text displays **Dashboard settings**.

2. In the **General** pane, navigate to the **Time Options** section and enter the time range in the **Autorefresh** field. You can specify the range in seconds, minutes, hours, and days format.

Note If you opt not to specify range, then the dashboard gets refreshed at the default interval.

Exporting and Importing Dashboards

This section describes how to export and import Grafana dashboards between environments and share them.

Exporting Dashboards

To export a dashboard configuration to a file:

1. Log in as an administrative user.

2. Open the dashboard that you want to export.

3. Click the gear icon at the top of the page, and then click **Export** to save the dashboard configuration on your local system.

4. If prompted, browse to the location on your local system to save the dashboard template, then click OK.

Importing Dashboards

To import a dashboard from a file:

1. Log in as an administrative user.

2. In the left pane, click the Dashboard icon to open the menu and click **Home**. The home pane opens.

3. Click the Home drop-down and click **Import dashboard**.

4. Specify the Grafana dashboard URL or ID that you want to import, provide the JSON details, or click **Upload.json File** and browse to the JSON file that you want to import.

5. Click **Load**.

   Make sure to save the dashboard to protect the changes that you made to the dashboard.

Exporting the Graph Data to CSV

This section describes how to export a dashboard in a CSV format.

1. On the Grafana dashboard, click the title of the graph to open the graph controls.

2. Click the rows button to open the menu.

3. To view the export option, click **More** and then click **Export CSV**.

   Your web browser downloads the *grafana_data_export.csv* file.

Filtering the Graphs

This section describes how to filter graphs on a dashboard.

You can narrow down the visualizations appearing on a dashboard by filtering them based on the specific time range.

1. On the PCF Application dashboard, in the top-right corner, click the clock icon.

2. Choose the range for which you want to view the graphs. Quick ranges provide the commonly used ranges that retrieve data in the shortest time. For specific range, provide the range under the Custom range heading.

Bulk Statistics

Bulk statistics are the statistics that are collected periodically and written to a set of CSV files. These statistics can be used by external analytic processes and/or network management systems. Bulk stats allows you to combine different KPIs into a unified query that fetches the custom statistical data. For the complete list of PCF KPIs, see Statistics and KPI Reference, on page 25.

The SMI component handles the collection of the bulk stats from the nodes and PCF consumes these stats. The bulk stats are generated for the following components:

- Container: Includes the raw and rate of the change statistics.

- Pod: Includes the raw and rate of the change statistics.

- System: Consists of the system level KPIs.

✎

Note The container and Pod statistics contain the predefined infrastructure outputs such as CPU and memory. You can also customize the query to fetch specific outputs as per your requirement.

There are two types of bulk statistics:

- Gauge - A snapshot value that shows the statistic at that reporting moment (for example, the number of current PDP contexts, simultaneous Active EPS Bearers). Gauge statistics can increment or decrement continuously.

- Counter - A historic value that shows the statistic that accumulated over time (for example, the total number of CSR requests received). Counter values can only increment except in two cases: rollover, where a counter exceeds its maximum value and rolls over to zero, and reset, where a counter is manually reset to zero.

Configuring the Bulk Statistics Collection

This section describes how to configure the bulk statistics collection feature.

You can optimize and control the bulk statistics collection by creating the Prometheus query that you configure on the SMI Ops Center. The cumulative result of the statistics query is available in a CSV file which is created on the node where you run the query.

1. Log in to the SMI Ops Center and run the following:

```
configure
  bulk-stats enable true
  bulk-stats query   kpi_name
  expression "sum(irate(kpi_name
{exported_application=~\".*\",command_code=~\".*\"}[1m])>0) by
(exported_application,command_code)"

  label   operation_name
  exit
  bulk-stats query    kpi_name
  expression "(sum(rate(kpi_name[duration])) by (operation_name))"
  label   operation_name
  exit
```

NOTES:

- **bulk-stats query** *kpi_name*: Specify the statistics name for which you want to generate stats in bulk. For example, inbound_request_total and diameter_request_total.

  See Statistics and KPI Reference, on page 25 for the list of KPIs.

- **expression** *"sum(irate(kpi_name {exported_application=~\".*\",command_code=~\".*\"}[1m])>0) by (exported_application,command_code)"*: Indicates the query format following which SMI collects the stats. For example:
  "sum(irate(diameter_requests_total{exported_application=~\".\",command_code=~\".*\"}[1m])>0) by (exported_application,command_code)"*

  > **Note** Based on the KPI tha you specify, manipulate the query. For instance, in case of the inbound_request_total KPI, add a parameter for specifying the duration as [5m]. This means that the SMI collects the stats for the total inbound requests that are processed in 5 minutes.

- **label** *operation_name*: Specify the operation that processes the KPI.

Sample Queries for Bulk Statistics

This section contains the sample Prometheus queries.

Table 2: PCF Sample Queries

Query	Description
incoming_request_total{interface_name="N7", command="Create"}	Fetches the total incoming create request for the N7 interface.
incoming_request_total{interface_name="N7", command="Update"}	Collects the total incoming update request for the N7 interface.
diameter_requests_total{command_code="AAR"}	Collects the total number of Diameter requests that PCF has processed.

Query	Description
message_total{type=~"ldap_change-res_success"}	Fetches the total number of messages that are triggered for the changes that are successfully completed in the LDAP.
outgoing_request_total{interface_name="NRF", command="DeRegister"}	Collects the total count of the outgoing deregistration requests that NRF processed.

Sample Configuration

The following is a sample bulk statistic:

```
 cee(config)#
bulk-stats enable true
bulk-stats query diameter_request_total
expression "sum(irate(diameter_requests_total{exported_application=~\".*\",command_code=~\
".*\"}[1m])>0) by (exported_application,command_code)"
label command_code
exit
bulk-stats query inbound_request_total
expression "(sum(rate(inbound_request_total[5m])) by (operation_name))"
label operation_name
exit
bulk-stats query outgoing_request_total
expression "(sum(rate(outgoing_request_total[5m])) by (operation_name))"
label operation_name
exit
```

Sample Bulk Statistics Configuration

This section provides sample bulk statistics configurations that are defined in PCF.

LDAP_Max_Avail_Connections

```
bulk-stats query LDAP_Max_Avail_Connections
expression sum(ldap_connections_total{type=~\"MaximumAvailableConnections.*\",
server_set=~\".*\"} OR on () vector (0)) by (namespace)
label (server_set)
exit
```

LDAP_Num_Avail_Connections

```
bulk-stats query LDAP_Num_Avail_Connections
expression sum(ldap_connections_total{type=~\"MaximumAvailableConnections.*\",
server_set=~\".*\"} OR on () vector (0)) by (namespace)
label (server_set)
exit
```

N28_Notify_Failure

```
bulk-stats query N28_Notify_Failure
expression sum(incoming_request_total{interface_name=\"N28\", command=\"Notify\",
result_code!~\"20.*\"} OR on () vector (0)) by (namespace)
```

```
label (interface_name)
exit
```

N28_Notify_Success

```
bulk-stats query  N28_Notify_Success
expression
sum(incoming_request_total{interface_name=\"N28\",command=\"Notify\",result_code=~\"20.*\"}
OR on () vector (0)) by (namespace)
label (interface_name)
exit
```

N28_Notify_Total

```
bulk-stats query  N28_Notify_Total
expression sum(incoming_request_total{interface_name=\"N28\",command=\"Notify\"}
OR on () vector (0)) by (namespace)
label (interface_name)
exit
```

N28_Sessions

```
bulk-stats query N28_Sessions
expression sum(db_records_total{session_type=\"N28_TGPP\"} OR on () vector (0)) by
(namespace)
label (session_type)
exit
```

N28_Subscribe_Failure

```
bulk-stats query N28_Subscribe_Failure
expression sum(outgoing_request_total{interface_name=\"N28\", command=\"Subscribe\",
result_code!~\"20.*\"} OR on () vector (0)) by (namespace)
label (interface_name)
exit
```

N28_Subscribe_Failure

```
bulk-stats query N28_Subscribe_Failure
expression sum(outgoing_request_total{interface_name=\"N28\", command=\"Subscribe\",
result_code!~\"20.*\"} OR on () vector (0)) by (namespace)
label (interface_name)
exit
```

N28_Subscribe_Success

```
bulk-stats query N28_Subscribe_Success
expression sum
(outgoing_request_total{interface_name=\"N28\",command=\"Subscribe\",result_code=~\"20.*\"}
OR on () vector (0)) by (namespace)
label (interface_name)
exit
```

N28_Subscribe_Total

bulk-stats query *N28_Subscribe_Total*
expression *sum(outgoing_request_total{interface_name=\"N28\",command=\"Subscribe\"}*
OR on () vector (0)) by (namespace)
label *(interface_name)*
exit

N28_Terminate_Failure

bulk-stats query *N28_Terminate_Failure*
expression *sum(incoming_request_total{interface_name=\"N28\", command=\"Terminate\",*
result_code!~\"20.\"} OR on () vector (0)) by (namespace)*
label *(interface_name)*
exit

N28_Terminate_Success

bulk-stats query *N28_Terminate_Success*
expression
sum(incoming_request_total{interface_name=\"N28\",command=\"Terminate\",result_code=~\"20.\"}*
OR on () vector (0)) by (namespace)
label *(interface_name)*
exit

N28_Terminate_Total

bulk-stats query *N28_Terminate_Total*
expression *sum(incoming_request_total{interface_name=\"N28\",command=\"Terminate\"}*
OR on () vector (0)) by (namespace)
label *(interface_name)*
exit

N28_Unsubscribe_Failure

bulk-stats query *N28_Unsubscribe_Failure*
expression *sum(outgoing_request_total{interface_name=\"N28\",*
command=\"Unsubscribe\", result_code!~\"20.\"} OR on () vector (0)) by (namespace)*
label *(interface_name)*
exit

N28_Unsubscribe_Success

bulk-stats *N28_Unsubscribe_Success*
expression
sum(outgoing_request_total{interface_name=\"N28\",command=\"Unsubscribe\",result_code=~\"20.\"}*
OR on () vector (0)) by (namespace)
label *(interface_name)*
exit

N28_Unsubscribe_Total

bulk-stats query *N28_Unsubscribe_Total*
expression *sum(outgoing_request_total{interface_name=\"N28\",command=\"Unsubscribe\"}*
OR on () vector (0)) by (namespace)

label *(interface_name)*
exit

N7_Create_Failure

bulk-stats query *N7_Create_Failure*
expression *sum(incoming_request_total{interface_name=\"N7\", command=\"Create\", result_code!~\"20.*\"} OR on () vector (0)) by (namespace)*
label *(interface_name)*
exit

N7_Create_Success

bulk-stats query *N7_Create_Success*
expression
sum(incoming_request_total{interface_name=\"N7\",command=\"Create\",result_code=~\"20.\"} OR on () vector (0)) by (namespace)*
label *(interface_name)*
exit

N7_Create_Total

bulk-statsquery *N7_Create_Total*
expression *sum(incoming_request_total{interface_name=\"N7\",command=\"Create\"} OR on () vector (0)) by (namespace)*
label *(interface_name)*
exit

N7_Delete_Failure

bulk-stats query *N7_Delete_Failure*
expression *sum(incoming_request_total{interface_name=\"N7\", command=\"Delete\", result_code!~\"20.*\"} OR on () vector (0)) by (namespace)*
label *(interface_name)*
exit

N7_Delete_Success

bulk-stats query *N7_Delete_Success*
expression
sum(incoming_request_total{interface_name=\"N7\",command=\"Delete\",result_code=~\"20.\"} OR on () vector (0)) by (namespace)*
label *(interface_name)*
exit

N7_Delete_Total

bulk-stats query *N7_Delete_Total*
expression *sum(incoming_request_total{interface_name=\"N7\",command=\"Delete\"} OR on () vector (0)) by (namespace)*
label *(interface_name)*
exit

N7_Notify_Failure

bulk-stats query *N7_Notify_Failure*
expression *sum(outgoing_request_total{interface_name=\"N7\", command=\"Notify\", result_code!~\"20.*\"} OR on () vector (0)) by (namespace)*
label *(interface_name)*
exit

N7_Notify_Success

bulk-stats query *N7_Notify_Success*
expression
sum(outgoing_request_total{interface_name=\"N7\",command=\"Notify\",result_code=~\"20.\"} OR on () vector (0)) by (namespace)*
label *(interface_name)*
exit

N7_Notify_Total

bulk-stats query *N7_Notify_Total*
expression *sum(outgoing_request_total{interface_name=\"N7\",command=\"Notify\"} OR on () vector (0)) by (namespace)*
label *(interface_name)*
exit

NAP_Total

bulk-stats query *NAP_Total*
expression *sum(message_total{type=~\"ldap_change-res.+\"} OR on () vector (0)) by (namespace)*
label *(ldap)*
exit

NRF_Deregister_Failure

bulk-stats query *NRF_Deregister_Failure*
expression
sum(outgoing_request_total{interface_name=\"NRF\",command=\"DeRegister\",result_code!~\"20.\"} OR on () vector (0)) by (namespace)*
label *(interface_name)*
exit

NRF_Deregister_Success

bulk-stats query *NRF_Deregister_Success*
expression
sum(outgoing_request_total{interface_name=\"NRF\",command=\"DeRegister\",result_code=~\"20.\"} OR on () vector (0)) by (namespace)*
label *(interface_name)*
exit

NRF_Deregister_Total

bulk-stats query *NRF_Deregister_Total*
expression sum(outgoing_request_total{interface_name=\"NRF\",command=\"DeRegister\"}

```
OR on () vector (0)) by (namespace)
label (interface_name)
exit
```

NRF_Discovery_Failure

```
bulk-stats query NRF_Discovery_Failure
expression
sum(outgoing_request_total{interface_name=\"NRF\",command=\"Discovery\",result_code!~\"20.*\"}
OR on () vector (0)) by (namespace)
label(interface_name)
exit
```

NRF_Discovery_Success

```
bulk-stats query NRF_Discovery_Success
expression
sum(outgoing_request_total{interface_name=\"NRF\",command=\"Discovery\",result_code=~\"20.*\"}
OR on () vector (0)) by (namespace)
label (interface_name)
exit
```

NRF_Discovery_Total

```
bulk-stats query NRF_Discovery_Total
expression sum(outgoing_request_total{interface_name=\"NRF\",command=\"Discovery\"}
OR on () vector (0)) by (namespace)
label (interface_name)
exit
```

NRF_Heartbeat_Failure

```
bulk-stats query NRF_Heartbeat_Failure
expression
sum(outgoing_request_total{interface_name=\"NRF\",command=\"Heartbeat\",result_code!~\"20.*\"}
OR on () vector (0)) by (namespace)
label (interface_name)
exit
```

NRF_Heartbeat_Success

```
bulk-stats query NRF_Heartbeat_Success
expression
sum(outgoing_request_total{interface_name=\"NRF\",command=\"Heartbeat\",result_code=~\"20.*\"}
OR on () vector (0)) by (namespace)
label (interface_name)
exit
```

NRF_Heartbeat_Total

```
bulk-stats query NRF_Heartbeat_Total
expression sum(outgoing_request_total{interface_name=\"NRF\",command=\"Heartbeat\"}
OR on () vector (0)) by (namespace)
label (interface_name)
exit
```

NRF_Notify_Failure

bulk-stats query *NRF_Notify_Failure*
expression
sum(incoming_request_total{interface_name=\"NRF\",command=\"Notify\",result_code!~\"20.\"}*
OR on () vector (0)) by (namespace)
label *(interface_name)*
exit

NRF_Notify_Success

bulk-stats query *NRF_Notify_Success*
expression
sum(incoming_request_total{interface_name=\"NRF\",command=\"Notify\",result_code=~\"20.\"}*
OR on () vector (0)) by (namespace)
label *(interface_name)*
exit

NRF_Notify_Total

bulk-stats query *NRF_Notify_Total*
expression *sum(incoming_request_total{interface_name=\"NRF\",command=\"Notify\"}*
OR on () vector (0)) by (namespace)
label *(interface_name)*
exit

NRF_Register_Failure

bulk-stats query *NRF_Register_Failure*
expression
sum(outgoing_request_total{interface_name=\"NRF\",command=\"Register\",result_code!~\"20.\"}*
OR on () vector (0)) by (namespace)
label *(interface_name)*
exit

NRF_Register_Total

bulk-stats query *NRF_Register_Total*
expression *sum(outgoing_request_total{interface_name=\"NRF\",command=\"Register\"}*
OR on () vector (0)) by (namespace)
label *(interface_name)*
exit

NRF_Register_success

bulk-stats query *NRF_Register_success*
expression
sum(outgoing_request_total{interface_name=\"NRF\",command=\"Register\",result_code=~\"20.\"}*
OR on () vector (0)) by (namespace)
label *(interface_name)*
exit

NRF_Subscribe_Failure

bulk-stats query *NRF_Subscribe_Failure*
expression
sum(outgoing_request_total{interface_name=\"NRF\",command=\"Subscribe\",result_code!~\"20.\"}*
OR on () vector (0)) by (namespace)
label *(interface_name)*
exit

NRF_Subscribe_Success

bulk-stats query *NRF_Subscribe_Success*
expression
sum(outgoing_request_total{interface_name=\"NRF\",command=\"Subscribe\",result_code=~\"20.\"}*
OR on () vector (0)) by (namespace)
label *(interface_name)*
exit

NRF_Subscribe_Total

bulk-stats query *NRF_Subscribe_Total*
expression sum(outgoing_request_total{interface_name=\"NRF\",command=\"Subscribe\"}
OR on () vector (0)) by (namespace)
label (interface_name)
exit

NRF_Unsubscribe_Failure

bulk-stats query *NRF_Unsubscribe_Failure*
expression
sum(outgoing_request_total{interface_name=\"NRF\",command=\"Unubscribe\",result_code!~\"20.\"}*
OR on () vector (0)) by (namespace)
label *(interface_name)*
exit

NRF_Unsubscribe_Success

bulk-stats query *NRF_Unsubscribe_Success*
expression
sum(outgoing_request_total{interface_name=\"NRF\",command=\"Unubscribe\",result_code=~\"20.*\"}
OR on () vector (0)) by (namespace)
label *(interface_name)*
exit

NRF_Unsubscribe_Total

bulk-stats query *NRF_Unsubscribe_Total*
expression sum(outgoing_request_total{interface_name=\"NRF\",command=\"Unsubscribe\"}
OR on () vector (0)) by (namespace)
label *(interface_name)*
exit

PLF_Failure

bulk-stats query *PLF_Failure*
expression *sum(message_total{type!~\"ldap_search-res_norecord|ldap_search-res_success\"} OR on ()*
vector (0)) by (namespace)
label *(ldap)*
exit

PLF_Success

bulk-stats query *PLF_Success*
expression *sum(message_total{type=~\"ldap_search-res_norecord|ldap_search-res_success\"} OR on ()*
vector (0)) by (namespace)
label *(ldap)*
exit

PLF_Total

bulk-stats query *PLF_Total*
expression *sum(message_total{type=~\"ldap_search.+\"} OR on () vector (0)) by (namespace)*
label *(ldap)*
exit

Rx_AAR_Failure

bulk-stats query *Rx_AAR_Failure*
expression *sum(diameter_responses_total{command_code=\"AAA\",result_code!=\"2001\"} OR on () vector (0)) by (namespace)*
label *(command_code)*
exit

Rx_AAR_Success

bulk-stats query *Rx_AAR_Success*
expression *sum(diameter_responses_total{command_code=\"AAA\",result_code=\"2001\"} OR on () vector (0)) by (namespace)*
label *(command_code)*
exit

Rx_AAR_Total

bulk-stats query *Rx_AAR_Total*
expression *sum(diameter_requests_total{command_code=\"AAR\"} OR on () vector (0)) by (namespace)*
label *(command_code)*
exit

Rx_ASR_Failure

bulk-stats query *Rx_ASR_Failure*
expression *sum(diameter_responses_total{command_code=\"ASA\",result_code!=\"2001\"} OR on () vector (0)) by (namespace)*

```
label (command_code)
```
exit

Rx_ASR_Success

bulk-stats query *Rx_ASR_Success*
expression *sum(diameter_responses_total{command_code=\"ASA\",result_code=\"2001\"}*
OR on () vector (0)) by (namespace)
label *(command_code))*
exit

Rx_ASR_Total

bulk-stats query *Rx_ASR_Total*
expression *sum(diameter_requests_total{command_code=\"ASR\"} OR on () vector (0))*
by (namespace)
label *(command_code)*
exit

Rx_RAR_Failure

bulk-stats query*Rx_RAR_Failure*
expression sum(diameter_responses_total{command_code=\"RAA\",result_code!=\"2001\"}
OR on () vector (0)) by (namespace)
label *(command_code)*
exit

Rx_RAR_Success

bulk-stats query *Rx_RAR_Success*
expression *sum(diameter_responses_total{command_code=\"RAA\",result_code=\"2001\"}*
OR on () vector (0)) by (namespace)
label *(command_code)*
exit

Rx_RAR_Total

bulk-stats query *Rx_RAR_Total*
expression sum(diameter_requests_total{command_code=\"RAR\"} OR on () vector (0))
by (namespace)
label *(command_code)*
exit

Rx_STR_Failure

bulk-stats query *Rx_STR_Failure*
expression *sum(diameter_responses_total{command_code=\"STA\",result_code!=\"2001\"}*
OR on () vector (0)) by (namespace)
label *(command_code)*
exit

Rx_STR_Success

bulk-stats query *Rx_STR_Success*
expression *sum(diameter_responses_total{command_code=\"STA\",result_code=\"2001\"}*

OR on () vector (0)) by (namespace)
label *(command_code)*
exit

Rx_STR_Total

bulk-stats query *Rx_STR_Total*
expression *sum(diameter_requests_total{command_code=\"STR\"} OR on () vector (0)) by (namespace)*
label *(command_code)*
exit

Rx_Sessions

bulk-stats query *Rx_Sessions*
expression *sum(db_records_total{session_type=\"RX_5G_TGPP\"} OR on () vector (0)) by (namespace)*
label *(session_type)*
exit

Total_Diameter_Peer

bulk-stats query *Total_Diameter_Peer*
expression *count(diameter_peer_status) OR on() vector(0)*
exit

Total_Diameter_Peer_Connected

bulk-stats query *Total_Diameter_Peer_Connected*
expression *count(diameter_peer_status==0) OR on() vector(0)*
exit

USD_Modify_Failure

bulk-stats query *USD_Modify_Failure*
expression
*sum(message_total{component=\"ldap-ep\",type=~\".*_ldap_Modify\",status!=\"success\"} OR on () vector (0)) by (namespace)*
label *(ldap)*
exit

USD_Modify_Success

bulk-stats query *USD_Modify_Success*
expression
*sum(message_total{component=\"ldap-ep\",type=~\".*_ldap_Modify\",status=\"success\"} OR on () vector (0)) by (namespace)*
label *(ldap)*
exit

USD_Modify_Total

bulk-stats query *USD_Modify_Total*
expression *sum(message_total{component=\"ldap-ep\",type=~\".*_ldap_Modify\"} OR on () vector (0)) by (namespace)*

```
label (ldap)
exit
```

USD_Query_Failure

```
bulk-stats query USD_Query_Failure
expression
sum(message_total{component=\"ldap-ep\",type=~\".*_ldap_query\",status!=\"success\"} OR on
() vector (0)) by (namespace)
label (ldap)
exit
```

USD_Query_Success

```
bulk-stats query USD_Query_Success
expression
sum(message_total{component=\"ldap-ep\",type=~\".*_ldap_query\",status=\"success\"} OR on
() vector (0)) by (namespace)
label (ldap)
exit
```

USD_Query_Total

```
bulk-stats query USD_Query_Total
expression sum(message_total{component=\"ldap-ep\",type=~\".*_ldap_query\"} OR on
() vector (0)) by (namespace)
label (ldap)
exit
```

active-alerts

```
bulk-stats query active-alerts
expression sum(ALERTS{alertstate=\"firing\"})"
label (alertname)
exit
```

config-query-memory-used

```
bulk-stats query config-query-memory-used
expression sum(node_memory_MemTotal_bytes)-sum(node_memory_MemFree_bytes
label (hostname)
exit
```

query cpu-idle

```
bulk-stats query query cpu-idle
expression avg(rate(node_cpu_seconds_total{mode=\"idle\"}[1m]))
label (hostname)
exit
```

cpu-iowait

```
bulk-stats query cpu-iowait
expression avg(rate(node_cpu_seconds_total{mode=\"iowait\"}[1m]))*100.00
```

label *(hostname)*
exit

cpu-softirq

bulk-stats query *cpu-softirq*
expression *avg(rate(node_cpu_seconds_total{mode=\"softirq\"}[1m]))*100.00*
label *(hostname)*
exit

cpu-steal

bulk-stats query *cpu-steal*
expression *avg(rate(node_cpu_seconds_total{mode=\"steal\"}[1m]))*100.00*
label *(hostname)*
exit

cpu-system

bulk-stats query *cpu-system*
expression *avg(rate(node_cpu_seconds_total{mode=\"system\"}[1m]))*100.00*
label *(hostname)*
exit

cpu-user

bulk-stats query *cpu-user*
expression *avg(rate(node_cpu_seconds_total{mode=\"user\"}[1m]))*100.00*
label *(hostname)*
exit

daemonset-ready-percent

bulk-stats query *daemonset-ready-percent*
expression
kube_daemonset_status_number_ready/kube_daemonset_status_desired_number_scheduled*100
label *(daemonset)*
exit

datastore_failures

bulk-stats query *datastore_failures*
expression *sum(datastore_request_total{error_code!~\"0|409\"})*
label *(error_code)*
exit

deployment-ready-percent

bulk-stats query *deployment-ready-percent*
expression *kube_deployment_status_replicas_available/kube_deployment_status_replicas*100*
label *(deployment)*
exit

diameter_peer_status

bulk-stats query *diameter_peer_status*
expression *avg(diameter_peer_status{host=~\".*\"} OR on () vector (0)) by (namespace)*
label *(host)*
exit

entitlement-status

bulk-stats query *entitlement-status*
expression entitlement_status{enforce_mode!=\"InCompliance\"}
label *(hostname)*
exit

filesystem-data-avail-bytes

bulk-stats query *filesystem-data-avail-bytes*
expression *avg(node_filesystem_avail_bytes{device=\"/dev/vda1\"})*
label *(hostname)*
exit

filesystem-root-avail-bytes

bulk-stats query *filesystem-root-avail-bytes*
expression *avg(node_filesystem_avail_bytes{device=\"/dev/sda1\"})*
label *(hostname)*
exit

k8s-pods-status

bulk-stats query *k8s-pods-status*
expression *sum(kube_pod_status_phase)*
label *(phase)*
exit

kubelet-node-status

bulk-stats query *kubelet-node-status*
expression *sum(kube_node_status_condition{status=\"true\"})*
label *(condition)*
exit

kublet-running-pod-count

bulk-stats query *kublet-running-pod-count*
expression *kubelet_running_pod_count*
label *(hostname)*
exit

memory-used

bulk-stats query *query memory-used*
expression *sum(node_memory_MemTotal_bytes)*

label *(hostname)*
exit

network-carrier-bond-changes-total

bulk-stats query *network-carrier-bond-changes-total*
expression *sum(node_network_carrier_changes_total{device=~\"bond[0-9]\"} OR on ()*
vector (0)) by (namespace)
label *(hostname)*
exit

network-carrier-ens-changes-total

bulk-stats query *network-carrier-ens-changes-total*
expression *sum(node_network_carrier_changes_total{device=~\"ens.*\"} OR on () vector*
(0)) by (namespace)
label *(hostname)*
exit

network-errors-total

bulk-stats query *network-errors-total*
expression *sum(node_network_receive_errs_total)*
label *(hostname)*
exit

network-receive-bond-bytes-total

bulk-stats query *network-receive-bond-bytes-total*
expression *sum(node_network_receive_bytes_total{device=~\"bond[0-9]\"})*
label *(hostname)*
exit

network-receive-ens-bytes-total

bulk-stats query *network-receive-ens-bytes-total*
expression *sum(node_network_receive_bytes_total{device=~\"ens.*\"})*
label *(hostname)*
exit

network-transmit-bond-bytes-total

bulk-stats query *network-transmit-bond-bytes-total*
expression *sum(node_network_transmit_bytes_total{device=~\"bond[0-9]\"})*
label *(hostname)*
exit

network-transmit-ens-bytes-total

bulk-stats query *network-transmit-ens-bytes-total*
expression *sum(node_network_transmit_bytes_total{device=~\"ens.*\"})*
label *(hostname)*
exit

node-disk-rate-read-bytes

bulk-stats query *node-disk-rate-read-bytes*
expression *sum(rate(node_disk_read_bytes_total[5m]))*
label *(hostname)*
exit

node-disk-write-read-bytes

bulk-stats query *node-disk-write-read-bytes*
expression *sum(rate(node_disk_written_bytes_total[5m]))*
label *(hostname)*
exit

node-load-15

bulk-stats query *node-load-15*
expression *node_load15*
label *(hostname)*
exit

node-memory-free-bytes

bulk-stats query *node-memory-free-bytes*
expression *sum(node_memory_MemTotal_bytes)*
label *(hostname)*
exit

record_conflict

bulk-stats query *record_conflict*
expression *sum(datastore_notify_total{notification_type=\"RECORD_CONFLICT\"} OR on () vector (0)) by (namespace)*
label *(notification_type)*
exit

statefulset-ready-percent

bulk-stats query *statefulset-ready-percent*
expression *kube_statefulset_status_replicas_ready/kube_statefulset_status_replicas*100*
label *(statefulset)*
exit

timer_expiry

bulk-stats query *timer_expiry*
expression *sum(datastore_notify_total{notification_type=\"TIMER_EPIRED\"} OR on () vector (0)) by (namespace)*
label *(notification_type)*
exit

version_mismatch_retries

```
bulk-stats query version_mismatch_retries
expression sum(datastore_request_total{error_code=\"409\"} OR on () vector (0)) by
(namespace)
label (error_code)
exit
```

Statistics and KPI Reference

PCF Statistics

PCF Service Category

charging_rule_base_total

Description: Total number of charging rule

Sample Query:
```
sum(irate(charging_rule_base{interface=\"N7\",name=~\".*\",action=\"install\"}[1m])) by
(interface,name,action)
```

Labels:

- Label: `interface`

  Label Description: Interface name that connects PCF to other network functions

  Example: N7,Rx etc.

- Label: `name`

  Label Description: Rule name

  Example: cbn#spp-tm etc.

- Label: `action`

  Label Description: Action

  Example: install,remove etc.

charging_rule_total

Description: Total charging rule

Sample Query:
```
sum(irate(charging_rule_total{interface=\"N7\",name=\"AF\",action=\"remove\"}[1m])) by
(interface,name,action)
```
Labels:

- Label: `interface`

  Label Description: Interface name that connects PCF to other network functions

  Example: N7 etc.

- Label: `name`

  Label Description: Rule name

  Example: AF etc.

- Label: `action`

  Label Description: Action

  Example: install,remove etc.

input_queue_buffer_action_total

Description: Total number of input queue buffer

Sample Query:
```
sum(irate(input_queue_buffer_action_total{realm=~\".*\",app_id=\"N7\",message_type=\"NOTIFY-RES\",result=\"bypass\"}[1m]))
by (app_id,message_type,result)
```
Labels:

- Label: `app_id`

  Label Description: App Id

  Example: N7,Rx etc.

Labels:

- Label: `message_type`

  Label Description: Message Type

  Example: NOTIFY-RES,ASA,RAA etc.

- Label: `result`

  Label Description: Result

  Example: pe-submit-error,drop,rate-limit,bypass,sync,send-x etc.

input_queue_result_total

Description: Total number of queue result

Sample Query:
```
sum(irate(input_queue_result_total{realm=~\".*\",app_id=\"N7\",message_type=\"NOTIFY-RES\",result=\"bypass\"}[1m]))
by (realm,app_id,message_type,result)
```

Labels:

- Label: `realm`

    Label Description: realm name

    Example: cisco.com etc.

Labels:

- Label: `app_id`

    Label Description: App Id

    Example: N7,Rx etc.

Labels:

- Label: `message_type`

    Label Description: Message Type

    Example: NOTIFY-RES,ASA,RAA etc.

- Label: `result`

    Label Description: Result

    Example: pe-submit-error,drop,rate-limit,bypass,sync,send-x etc.

input_queue_result_total

Description: Total number of queue result

Sample Query:

```
sum(irate(input_queue_result_total{realm=~\".*\",app_id=\"N7\",message_type=\"NOTIFY-RES\",result=\"bypass\"}[1m]))
by (realm,app_id,message_type,result)
```

Labels:

- Label: `realm`

    Label Description: realm name

    Example: cisco.com etc.

Labels:

- Label: `app_id`

    Label Description: App Id

    Example: N7,Rx etc.

Labels:

- Label: `message_type`

    Label Description: Message Type

    Example: NOTIFY-RES,ASA,RAA etc.

Labels:

- Label: `result`

  Label Description: Result

  Example: pe-submit-error,drop,rate-limit,bypass,sync,send-x etc.

input_queue_total

Description: Total number of Inbound queues

Sample Query: `sum(irate(session_rule{app_id=\"N7\",message_type=~\".*\"}[1m])) by (app_id,message_type)`

Labels:

- Label: `app_id`

  Label Description: Interface name that connects PCF to other network functions

  Example: N7,Rx etc.

Labels:

- Label: `message_type`

  Label Description: Trigger Enum

  Example: CREATE-REQ,DELETE-REQ,UPDATE-REQ,AAR,STR etc.

ldap_policy_request_total

Description: Monitors the total count of the ldap policy requests to evaluate policies that are processed by the PCF Engine.

Sample Query: `sum(irate(ldap_policy_request_total{type=~\".*ldap.*\",result=\"notify error xxx\"}[1m])>0) by (type, result)`

Labels:

- Label: `type`

  Label Description: Message Type as ldap or broadcast

  Example: ldap,broadcast

Labels:

- Label: `result`

  Label Description: The result

  Example: notify_error_xxx, rate_limited

message_total

Description: Monitors the total count of the ldap messages that are processed by the PCF Engine.

Sample Query: `sum(irate(message_total{type=\"ldap-change-res\",status=\"success\"}[1m])) by (type,status)`

Labels:

- Label: `type`

  Label Description: Message Type for ldap change or ldap search

  Example: ldap-change-res,ldap-search-res

Labels:

- Label: `status`

  Label Description: Status of message as error or success

  Example: success,error

pcf_messages_duration_seconds

Description: Aggregates the total amount of time that the PCF Engine took to process the messages

Sample Query: `sum(irate(pcf_messages_duration_seconds[1m])) by (interface,name,message_type)`

Labels:

- Label: `interface`

  Label Description: Interface name that connects PCF to other network functions

  Example: N7, N15, N25, N28, LDAP

Labels:

- Label: `name`

  Label Description: Interface name that connects PCF to other network functions

  Example: N7_CREATE, N7_UPDATE, N7_DELETE, N7_NOTIFY,etc.

Labels:

- Label: `message_type`

  Label Description: Message Type as Request or Response

  Example: REQ,RES

pcf_messages_total

Description: Monitors the total count of the PCF messages that are processed by the PCF Engine.

Sample Query:
```
sum(irate(pcf_messages_total{interface=\"N7\",name=~\".*\",message_type=\"REQ\"}[1m])) by
(interface,name,message_type)
```

Labels:

- Label: `interface`

  Label Description: Interface name that connects PCF to other network functions

  Example: N7, N15, N25, N28, LDAP

Labels:

- Label: `name`

  Label Description: Interface name that connects PCF to other network functions

  Example: N7_CREATE, N7_UPDATE, N7_DELETE, N7_NOTIFY, N15_CREATE, N15_UPDATE, N15_DELETE, N15_NOTIFY, N28_SUBSCRIBE, N28_UNSUBSCRIBE, N28_NOTIFY, N28_TERMINATE, UDR_QUERY_SM, UDR_QUERY_AM

Labels:

- Label: `message_type`

  Label Description: Message Type as Request or Response

  Example: REQ,RES

policy_req_triggers_total

Description: Aggregates the total count of the policy requests triggers toward the PCF Engine

Sample Query: `sum(irate(policy_req_triggers_total{interface=\"N7\",name=~\".*\"}[1m])) by (interface,name)`

Labels:

- Label: `interface`

  Label Description: Interface name that connects PCF to other network functions

  Example: N15,N7

Labels:

- Label: `name`

  Label Description: Trigger Enum

  Example: AN_CH_COR, PLMN_CH, RE_TIMEOUT etc.

ran_nas_total

Description: Total number of rannas cause

Sample Query:
`sum(irate(ran_nas_total{media_type=\"1\",protocol_type=\"1\",cause_type=\"3\"}[1m])) by (media_type, protocol_type,cause_type)`

Labels:

- Label: `media_type`

  Label Description: Media Type

  Example: 1, 2, 3 etc.

Labels:

- Label: `protocol_type`

Label Description: Protocol Type

Example: 1, 2, 3 etc.

Labels:

- Label: `cause_type`

Label Description: Cause Type

Example: 1, 2, 3 etc.

session_rule_total

Description: Captures the total Session Rule requests

Sample Query: `sum(irate(session_rule{interface=\"N7\",name=~\".*\"}[1m])) by (interface,name)`

Labels:

- Label: `interface`

Label Description: Interface name that connects PCF to other network functions

Example: N7

Labels:

- Label: `name`

Label Description: ARP Priority Level

Example: 1,2,3 etc.

PCF Endpoint Requests Category

async_svc_runnable_time

Description: Total Async Service Runnable Count

Sample Query:

`async_svc_runnable_time{service_name=\"npcf-smpolicycontrol\",operation=\"Npcf_SMPolicyControl_UpdateNotify\",command=\"Notify\",status=\"Success\"}`

Labels:

- Label: `service_name`

Label Description: Name of the Service

Example: nnrf-nfm, nnrf-disc, npcf-am-policy-control, npcf-smpolicycontrol, nchf-spendinglimitcontrol

- Label: `operation`

Label Description: Name of the Operation

Example: Npcf_SMPolicyControl_Create, Npcf_SMPolicyControl_Update, Npcf_SMPolicyControl_Delete, Npcf_SMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Create, Npcf_AMPolicyControl_Update, Npcf_AMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Delete, Nchf_SpendingLimitControl_Subscribe, Nchf_SpendingLimitControl_Unsubscribe, Nchf_SpendingLimitControl_Notify, Nchf_SpendingLimitControl_Terminate,

Nudr_Query_PolicyData_SM, Nudr_Query_PolicyData_AM, Nnrf_NfDiscovery, Nnrf_NfDeregister, Nnrf_NfRegister, Nnrf_NfHeartbeat, Nnrf_NfNotify, Nnrf_NfSubsribe, Nnrf_NfUnSubsribe

- Label: `command`

  Label Description: The Command

  Example: Create, Update, Delete, Notify, Subscribe, Unsubscribe, Terminate, SmDataQuery, AmDataQuery, Discovery, DeRegister, Register, Heartbeat, Unsubscribe

- Label: `status`

  Label Description: The Operation was successful or in error state

  Example: Error, Success

async_svc_runnable_total

Description: Total Async Service Runnable Count

Sample Query:

`async_svc_runnable_time{service_name=\"npcf-smpolicycontrol\",operation=\"Npcf_SMPolicyControl_UpdateNotify\",command=\"Notify\",status=\"Success\"}`

Labels:

- Label: `service_name`

  Label Description: Name of the Service

  Example: nnrf-nfm, nnrf-disc, npcf-am-policy-control, npcf-smpolicycontrol, nchf-spendinglimitcontrol

- Label: `operation`

  Label Description: Name of the Operation

  Example: Npcf_SMPolicyControl_Create, Npcf_SMPolicyControl_Update, Npcf_SMPolicyControl_Delete, Npcf_SMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Create, Npcf_AMPolicyControl_Update, Npcf_AMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Delete, Nchf_SpendingLimitControl_Subscribe, Nchf_SpendingLimitControl_Unsubscribe, Nchf_SpendingLimitControl_Notify, Nchf_SpendingLimitControl_Terminate, Nudr_Query_PolicyData_SM, Nudr_Query_PolicyData_AM, Nnrf_NfDiscovery, Nnrf_NfDeregister, Nnrf_NfRegister, Nnrf_NfHeartbeat, Nnrf_NfNotify, Nnrf_NfSubsribe, Nnrf_NfUnSubsribe

- Label: `command`

  Label Description: The Command

  Example: Create, Update, Delete, Notify, Subscribe, Unsubscribe, Terminate, SmDataQuery, AmDataQuery, Discovery, DeRegister, Register, Heartbeat, Unsubscribe

- Label: `status`

  Label Description: The Operation was successful or in error state

  Example: Error, Success

async_svc_runnable_total_seconds

Description: Total Async Service Runnable Response Time

Sample Query:

async_svc_runnable_total_seconds{service_name=\"nchf-spendinglimitcontrol\",operation=\"Nchf_SpendingLimitControl_Subscribe\",command=\"Subscribe\",status=\"Success\"}

Labels:

- Label: `service_name`

  Label Description: Name of the Service for a interface

  Example: nnrf-nfm, nnrf-disc, npcf-am-policy-control, npcf-smpolicycontrol, nchf-spendinglimitcontrol

Labels:

- Label: `operation`

  Label Description: Name of the Operation

  Example: Npcf_SMPolicyControl_Create, Npcf_SMPolicyControl_Update, Npcf_SMPolicyControl_Delete, Npcf_AMPolicyControl_Create, Npcf_AMPolicyControl_Update, Npcf_AMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Delete, Nchf_SpendingLimitControl_Subscribe, Nchf_SpendingLimitControl_Unsubscribe, Nchf_SpendingLimitControl_Notify, Nchf_SpendingLimitControl_Terminate, Nudr_Query_PolicyData_SM, Nudr_Query_PolicyData_AM, Nnrf_NfDiscovery, Nnrf_NfDeregister, Nnrf_NfRegister, Nnrf_NfHeartbeat, Nnrf_NfNotify, Nnrf_NfSubsribe, Nnrf_NfUnSubsribe

- Label: `command`

  Label Description: The Command

  Example: Create, Update, Delete, Notify, Subscribe, Unsubscribe, Terminate, SmDataQuery, AmDataQuery, Discovery, DeRegister, Register, Heartbeat, Unsubscribe

- Label: `status`

  Label Description: The Operation was successful or in error state

  Example: Error, Success

inbound_request_total

Description: Total Inbound Request Count

Sample Query:

inbound_request_total{interface_name=\"N7\",service_name=\"npcf-smpolicycontrol\",operation_name=\"Npcf_SMPolicyControl_Delete\",command=\"Delete\"}

Labels:

- Label: `interface_name`

  Label Description: Interface name that connects PCF to other network functions

  Example: N7, N15, N25, N28, NRF

Labels:

- Label: `service_name`

  Label Description: Name of the Service

  Example: nnrf-nfm, nnrf-disc, npcf-am-policy-control, npcf-smpolicycontrol, nchf-spendinglimitcontrol

Labels:

- Label: `command`

  Label Description: The Command

  Example: Create, Update, Delete, Notify, Subscribe, Unsubscribe, Terminate, SmDataQuery, AmDataQuery, Discovery, DeRegister, Register, Heartbeat, Unsubscribe

- Label: `operation_name`

  Label Description: Name of the Operation

  Example: Npcf_SMPolicyControl_Create, Npcf_SMPolicyControl_Update, Npcf_SMPolicyControl_Delete, Npcf_SMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Create, Npcf_AMPolicyControl_Update, Npcf_AMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Delete, Nchf_SpendingLimitControl_Subscribe, Nchf_SpendingLimitControl_Unsubscribe, Nchf_SpendingLimitControl_Notify, Nchf_SpendingLimitControl_Terminate, Nudr_Query_PolicyData_SM, Nudr_Query_PolicyData_AM, Nnrf_NfDiscovery, Nnrf_NfDeregister, Nnrf_NfRegister, Nnrf_NfHeartbeat, Nnrf_NfNotify, Nnrf_NfSubsribe, Nnrf_NfUnSubsribe

incoming_request_time

Description: Total Response Time for Incoming Requests

Sample Query:

```
incoming_request_time{interface_name=\"N7\",service_name=\"npcf-smpolicycontrol\",operation_name=\"Npcf_SMPolicyControl_Delete\",command=\"Delete\",result_code=\"204\"}
```

Labels:

- Label: `interface_name`

  Label Description: Interface name that connects PCF to other network functions

  Example: N7, N15, N25, N28, NRF

Labels:

- Label: `service_name`

  Label Description: Name of the Service

  Example: nnrf-nfm, nnrf-disc, npcf-am-policy-control, npcf-smpolicycontrol, nchf-spendinglimitcontrol

Labels:

- Label: `command`

  Label Description: The Command

  Example: Create, Update, Delete, Notify, Subscribe, Unsubscribe, Terminate, SmDataQuery, AmDataQuery, Discovery, DeRegister, Register, Heartbeat, Unsubscribe

Labels:

- Label: `operation_name`

  Label Description: Name of the Operation

  Example: Npcf_SMPolicyControl_Create, Npcf_SMPolicyControl_Update, Npcf_SMPolicyControl_Delete, Npcf_SMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Create,

Npcf_AMPolicyControl_Update, Npcf_AMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Delete, Nchf_SpendingLimitControl_Subscribe, Nchf_SpendingLimitControl_Unsubscribe, Nchf_SpendingLimitControl_Notify, Nchf_SpendingLimitControl_Terminate, Nudr_Query_PolicyData_SM, Nudr_Query_PolicyData_AM, Nnrf_NfDiscovery, Nnrf_NfDeregister, Nnrf_NfRegister, Nnrf_NfHeartbeat, Nnrf_NfNotify, Nnrf_NfSubsribe, Nnrf_NfUnSubsribe

- Label: `result_code`

  Label Description: Status Code

  Example: 200, 201, 202, 204, 205, 206, 301, 302, 303, 304, 305, 307, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 500, 501, 502, 503, 504, 505

incoming_request_time

Description: Total Response Time for Incoming Requests

Sample Query:

`incoming_request_time{interface_name=\"N7\",service_name=\"npcf-smpolicycontrol\",operation_name=\"Npcf_SMPolicyControl_Delete\",command=\"Delete\",result_code=\"204\"}`

Labels:

- Label: `interface_name`

  Label Description: Interface name that connects PCF to other network functions

  Example: N7, N15, N25, N28, NRF

Labels:

- Label: `service_name`

  Label Description: Name of the Service

  Example: nnrf-nfm, nnrf-disc, npcf-am-policy-control, npcf-smpolicycontrol, nchf-spendinglimitcontrol

Labels:

- Label: `command`

  Label Description: The Command

  Example: Create, Update, Delete, Notify, Subscribe, Unsubscribe, Terminate, SmDataQuery, AmDataQuery, Discovery, DeRegister, Register, Heartbeat, Unsubscribe

Labels:

- Label: `operation_name`

  Label Description: Name of the Operation

  Example: Npcf_SMPolicyControl_Create, Npcf_SMPolicyControl_Update, Npcf_SMPolicyControl_Delete, Npcf_SMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Create, Npcf_AMPolicyControl_Update, Npcf_AMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Delete, Nchf_SpendingLimitControl_Subscribe, Nchf_SpendingLimitControl_Unsubscribe, Nchf_SpendingLimitControl_Notify, Nchf_SpendingLimitControl_Terminate, Nudr_Query_PolicyData_SM, Nudr_Query_PolicyData_AM, Nnrf_NfDiscovery, Nnrf_NfDeregister, Nnrf_NfRegister, Nnrf_NfHeartbeat, Nnrf_NfNotify, Nnrf_NfSubsribe, Nnrf_NfUnSubsribe

Labels:

- Label: `result_code`

  Label Description: Status Code

  Example: 200, 201, 202, 204, 205, 206, 301, 302, 303, 304, 305, 307, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 500, 501, 502, 503, 504, 505

incoming_request_total

Description: Total Incoming Request Count

Sample Query:

`incoming_request_total{interface_name=\"N7\",service_name=\"npcf-smpolicycontrol\",operation_name=\"Npcf_SMPolicyControl_Delete\",command=\"Delete\",result_code=\"204\"}`

Labels:

- Label: `interface_name`

  Label Description: Interface name that connects PCF to other network functions

  Example: N7, N15, N25, N28, NRF

Labels:

- Label: `service_name`

  Label Description: Name of the Service

  Example: nnrf-nfm, nnrf-disc, npcf-am-policy-control, npcf-smpolicycontrol, nchf-spendinglimitcontrol

Labels:

- Label: `command`

  Label Description: The Command

  Example: Create, Update, Delete, Notify, Subscribe, Unsubscribe, Terminate, SmDataQuery, AmDataQuery, Discovery, DeRegister, Register, Heartbeat, Unsubscribe

Labels:

- Label: `operation_name`

  Label Description: Name of the Operation

  Example: Npcf_SMPolicyControl_Create, Npcf_SMPolicyControl_Update, Npcf_SMPolicyControl_Delete, Npcf_SMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Create, Npcf_AMPolicyControl_Update, Npcf_AMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Delete, Nchf_SpendingLimitControl_Subscribe, Nchf_SpendingLimitControl_Unsubscribe, Nchf_SpendingLimitControl_Notify, Nchf_SpendingLimitControl_Terminate, Nudr_Query_PolicyData_SM, Nudr_Query_PolicyData_AM, Nnrf_NfDiscovery, Nnrf_NfDeregister, Nnrf_NfRegister, Nnrf_NfHeartbeat, Nnrf_NfNotify, Nnrf_NfSubsribe, Nnrf_NfUnSubsribe

Labels:

- Label: `result_code`

  Label Description: Status Code

Example: 200, 201, 202, 204, 205, 206, 301, 302, 303, 304, 305, 307, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 500, 501, 502, 503, 504, 505

incoming_rpc_request_time

Description: Total RPC Time for Incoming Requests

Sample Query:

```
incoming_rpc_request_time{interface_name=\"N7\",service_name=\"npcf-smpolicycontrol\",operation_name=\"Npcf_SMPolicyControl_UpdateNotify\",command=\"Notify\",result_code=\"404\"}
```

Labels:

- Label: `interface_name`

  Label Description: Interface name that connects PCF to other network functions

  Example: N7, N15, N25, N28, NRF

Labels:

- Label: `service_name`

  Label Description: Name of the Service

  Example: nnrf-nfm, nnrf-disc, npcf-am-policy-control, npcf-smpolicycontrol, nchf-spendinglimitcontrol

Labels:

- Label: `command`

  Label Description: The Command

  Example: Create, Update, Delete, Notify, Subscribe, Unsubscribe, Terminate, SmDataQuery, AmDataQuery, Discovery, DeRegister, Register, Heartbeat, Unsubscribe

Labels:

- Label: `operation_name`

  Label Description: Name of the Operation

  Example: Npcf_SMPolicyControl_Create, Npcf_SMPolicyControl_Update, Npcf_SMPolicyControl_Delete, Npcf_SMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Create, Npcf_AMPolicyControl_Update, Npcf_AMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Delete, Nchf_SpendingLimitControl_Subscribe, Nchf_SpendingLimitControl_Unsubscribe, Nchf_SpendingLimitControl_Notify, Nchf_SpendingLimitControl_Terminate, Nudr_Query_PolicyData_SM, Nudr_Query_PolicyData_AM, Nnrf_NfDiscovery, Nnrf_NfDeregister, Nnrf_NfRegister, Nnrf_NfHeartbeat, Nnrf_NfNotify, Nnrf_NfSubsribe, Nnrf_NfUnSubsribe

Labels:

- Label: `result_code`

  Label Description: Status Code

  Example: 200, 201, 202, 204, 205, 206, 301, 302, 303, 304, 305, 307, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 500, 501, 502, 503, 504, 505

incoming_rpc_request_total

Description: Total Incoming RPC Request Count

Sample Query:

incoming_rpc_request_total{interface_name=\"N7\",service_name=\"npcf-smpolicycontrol\",operation_name=\"Npcf_SMPolicyControl_UpdateNotify\",command=\"Notify\",result_code=\"200\"}

Labels:

- Label: `interface_name`

  Label Description: Interface name that connects PCF to other network functions

  Example: N7, N15, N25, N28, NRF

Labels:

- Label: `service_name`

  Label Description: Name of the Service

  Example: nnrf-nfm, nnrf-disc, npcf-am-policy-control, npcf-smpolicycontrol, nchf-spendinglimitcontrol

Labels:

- Label: `command`

  Label Description: The Command

  Example: Create, Update, Delete, Notify, Subscribe, Unsubscribe, Terminate, SmDataQuery, AmDataQuery, Discovery, DeRegister, Register, Heartbeat, Unsubscribe

Labels:

- Label: `operation_name`

  Label Description: Name of the Operation

  Example: Npcf_SMPolicyControl_Create, Npcf_SMPolicyControl_Update, Npcf_SMPolicyControl_Delete, Npcf_SMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Create, Npcf_AMPolicyControl_Update, Npcf_AMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Delete, Nchf_SpendingLimitControl_Subscribe, Nchf_SpendingLimitControl_Unsubscribe, Nchf_SpendingLimitControl_Notify, Nchf_SpendingLimitControl_Terminate, Nudr_Query_PolicyData_SM, Nudr_Query_PolicyData_AM, Nnrf_NfDiscovery, Nnrf_NfDeregister, Nnrf_NfRegister, Nnrf_NfHeartbeat, Nnrf_NfNotify, Nnrf_NfSubsribe, Nnrf_NfUnSubsribe

Labels:

- Label: `result_code`

  Label Description: Status Code

  Example: 200, 201, 202, 204, 205, 206, 301, 302, 303, 304, 305, 307, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 500, 501, 502, 503, 504, 505

outgoing_request_time

Description: Total Response Time for Outgoing Requests

Sample Query:

outgoing_request_time{interface_name=\"N7\",service_name=\"npcf-smpolicycontrol\",operation_name=\"Npcf_SMPolicyControl_UpdateNotify\",command=\"Notify\",target_base_url=\"http://192.168.213.98:1/\",result_code=\"404\"}

Labels:

- Label: `interface_name`

  Label Description: Interface name that connects PCF to other network functions

  Example: N7, N15, N25, N28, NRF

Labels:

- Label: `service_name`

  Label Description: Name of the Service

  Example: nnrf-nfm, nnrf-disc, npcf-am-policy-control, npcf-smpolicycontrol, nchf-spendinglimitcontrol

Labels:

- Label: `command`

  Label Description: The Command

  Example: Create, Update, Delete, Notify, Subscribe, Unsubscribe, Terminate, SmDataQuery, AmDataQuery, Discovery, DeRegister, Register, Heartbeat, Unsubscribe

Labels:

- Label: `operation_name`

  Label Description: Name of the Operation

  Example: Npcf_SMPolicyControl_Create, Npcf_SMPolicyControl_Update, Npcf_SMPolicyControl_Delete, Npcf_SMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Create, Npcf_AMPolicyControl_Update, Npcf_AMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Delete, Nchf_SpendingLimitControl_Subscribe, Nchf_SpendingLimitControl_Unsubscribe, Nchf_SpendingLimitControl_Notify, Nchf_SpendingLimitControl_Terminate, Nudr_Query_PolicyData_SM, Nudr_Query_PolicyData_AM, Nnrf_NfDiscovery, Nnrf_NfDeregister, Nnrf_NfRegister, Nnrf_NfHeartbeat, Nnrf_NfNotify, Nnrf_NfSubsribe, Nnrf_NfUnSubsribe

Labels:

- Label: `result_code`

  Label Description: Status Code

  Example: 200, 201, 202, 204, 205, 206, 301, 302, 303, 304, 305, 307, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 500, 501, 502, 503, 504, 505

Labels:

- Label: `target_base_url`

  Label Description: Target Base Url

  Example: http://192.168.2.138:9081/ etc.

outgoing_request_total

Description: Total Outgoing Request Count

Sample Query:

`otging_request_total{interface_name=\"N7\",service_name=\"npcf-smpolicycontrol\",operation_name=\"Npcf_SMPolicyControl_UpdateNotify\",command=\"Notify\",target_base_url=\"http://192.168.2.138:9081/\",result_code=\"404\"}`

Labels:

- Label: `interface_name`

  Label Description: Interface name that connects PCF to other network functions

  Example: N7, N15, N25, N28, NRF

Labels:

- Label: `service_name`

  Label Description: Name of the Service

  Example: nnrf-nfm, nnrf-disc, npcf-am-policy-control, npcf-smpolicycontrol, nchf-spendinglimitcontrol

Labels:

- Label: `command`

  Label Description: The Command

  Example: Create, Update, Delete, Notify, Subscribe, Unsubscribe, Terminate, SmDataQuery, AmDataQuery, Discovery, DeRegister, Register, Heartbeat, Unsubscribe

Labels:

- Label: `operation_name`

  Label Description: Name of the Operation

  Example: Npcf_SMPolicyControl_Create, Npcf_SMPolicyControl_Update, Npcf_SMPolicyControl_Delete, Npcf_SMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Create, Npcf_AMPolicyControl_Update, Npcf_AMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Delete, Nchf_SpendingLimitControl_Subscribe, Nchf_SpendingLimitControl_Unsubscribe, Nchf_SpendingLimitControl_Notify, Nchf_SpendingLimitControl_Terminate, Nudr_Query_PolicyData_SM, Nudr_Query_PolicyData_AM, Nnrf_NfDiscovery, Nnrf_NfDeregister, Nnrf_NfRegister, Nnrf_NfHeartbeat, Nnrf_NfNotify, Nnrf_NfSubsribe, Nnrf_NfUnSubsribe, Nnrf_NFUpdate

Labels:

- Label: `result_code`

  Label Description: Status Code

  Example: 200, 201, 202, 204, 205, 206, 301, 302, 303, 304, 305, 307, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 500, 501, 502, 503, 504, 505

Labels:

- Label: `target_base_url`

  Label Description: Target Base Url

  Example: http://192.168.2.138:9081/

outgoing_rpc_request_time

Description: Total RPC Time for Outgoing Requests

Sample Query:

outgoing_rpc_request_time{interface_name=\"N7\",service_name=\"npcf-smpolicycontrol\",operation_name=\"Npcf_SMPolicyControl_Update\",command=\"Update\",result_code=\"404\"}

Labels:

- Label: `interface_name`

  Label Description: Interface name that connects PCF to other network functions

  Example: N7, N15, N25, N28, NRF

Labels:

- Label: `service_name`

  Label Description: Name of the Service

  Example: nnrf-nfm, nnrf-disc, npcf-am-policy-control, npcf-smpolicycontrol, nchf-spendinglimitcontrol

Labels:

- Label: `command`

  Label Description: The Command

  Example: Create, Update, Delete, Notify, Subscribe, Unsubscribe, Terminate, SmDataQuery, AmDataQuery, Discovery, DeRegister, Register, Heartbeat, Unsubscribe

Labels:

- Label: `operation_name`

  Label Description: Name of the Operation

  Example: Npcf_SMPolicyControl_Create, Npcf_SMPolicyControl_Update, Npcf_SMPolicyControl_Delete, Npcf_SMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Create, Npcf_AMPolicyControl_Update, Npcf_AMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Delete, Nchf_SpendingLimitControl_Subscribe, Nchf_SpendingLimitControl_Unsubscribe, Nchf_SpendingLimitControl_Notify, Nchf_SpendingLimitControl_Terminate, Nudr_Query_PolicyData_SM, Nudr_Query_PolicyData_AM, Nnrf_NfDiscovery, Nnrf_NfDeregister, Nnrf_NfRegister, Nnrf_NfHeartbeat, Nnrf_NfNotify, Nnrf_NfSubsribe, Nnrf_NfUnSubsribe

Labels:

- Label: `result_code`

  Label Description: Status Code

  Example: 200, 201, 202, 204, 205, 206, 301, 302, 303, 304, 305, 307, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 500, 501, 502, 503, 504, 505

outgoing_rpc_request_total

Description: Total Outgoing RPC Request Count

Sample Query:

outgoing_rpc_request_total{interface_name=\"N7\",service_name=\"npcf-smpolicycontrol\",operation_name=\"Npcf_SMPolicyControl_Update\",command=\"Update\",result_code=\"408\"}

Labels:

- Label: `interface_name`

  Label Description: Interface name that connects PCF to other network functions

  Example: N7, N15, N25, N28, NRF

Labels:

- Label: `service_name`

  Label Description: Name of the Service

  Example: nnrf-nfm, nnrf-disc, npcf-am-policy-control, npcf-smpolicycontrol, nchf-spendinglimitcontrol

Labels:

- Label: `command`

  Label Description: The Command

  Example: Create, Update, Delete, Notify, Subscribe, Unsubscribe, Terminate, SmDataQuery, AmDataQuery, Discovery, DeRegister, Register, Heartbeat, Unsubscribe

Labels:

- Label: `operation_name`

  Label Description: Name of the Operation

  Example: Npcf_SMPolicyControl_Create, Npcf_SMPolicyControl_Update, Npcf_SMPolicyControl_Delete, Npcf_SMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Create, Npcf_AMPolicyControl_Update, Npcf_AMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Delete, Nchf_SpendingLimitControl_Subscribe, Nchf_SpendingLimitControl_Unsubscribe, Nchf_SpendingLimitControl_Notify, Nchf_SpendingLimitControl_Terminate, Nudr_Query_PolicyData_SM, Nudr_Query_PolicyData_AM, Nnrf_NfDiscovery, Nnrf_NfDeregister, Nnrf_NfRegister, Nnrf_NfHeartbeat, Nnrf_NfNotify, Nnrf_NfSubsribe, Nnrf_NfUnSubsribe

Labels:

- Label: `result_code`

  Label Description: Status Code

  Example: 200, 201, 202, 204, 205, 206, 301, 302, 303, 304, 305, 307, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 500, 501, 502, 503, 504, 505

route_lookup_requests_duration_seconds

Description: Total Route Lookup Response Time in seconds

Sample Query:

`route_lookup_requests_duration_seconds{interface_name=\"N7\",service_name=\"npcf-smpolicycontrol\",operation_name=\"Npcf_SMPolicyControl_UpdateNotify\",command=\"Notify\",status=\"201\"}`

Labels:

- Label: `interface_name`

  Label Description: Interface name that connects PCF to other network functions

Example: N7, N15, N25, N28, NRF

Labels:

- Label: `service_name`

  Label Description: Name of the Service

  Example: nnrf-nfm, nnrf-disc, npcf-am-policy-control, npcf-smpolicycontrol, nchf-spendinglimitcontrol

Labels:

- Label: `command`

  Label Description: The Command

  Example: Create, Update, Delete, Notify, Subscribe, Unsubscribe, Terminate, SmDataQuery, AmDataQuery, Discovery, DeRegister, Register, Heartbeat, Unsubscribe

Labels:

- Label: `operation_name`

  Label Description: Name of the Operation

  Example: Npcf_SMPolicyControl_Create, Npcf_SMPolicyControl_Update, Npcf_SMPolicyControl_Delete, Npcf_SMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Create, Npcf_AMPolicyControl_Update, Npcf_AMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Delete, Nchf_SpendingLimitControl_Subscribe, Nchf_SpendingLimitControl_Unsubscribe, Nchf_SpendingLimitControl_Notify, Nchf_SpendingLimitControl_Terminate, Nudr_Query_PolicyData_SM, Nudr_Query_PolicyData_AM, Nnrf_NfDiscovery, Nnrf_NfDeregister, Nnrf_NfRegister, Nnrf_NfHeartbeat, Nnrf_NfNotify, Nnrf_NfSubsribe, Nnrf_NfUnSubsribe

Labels:

- Label: `result_code`

  Label Description: Status Code

  Example: 200, 201, 202, 204, 205, 206, 301, 302, 303, 304, 305, 307, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 500, 501, 502, 503, 504, 505

route_lookup_requests_time

Description: Total Route Lookup Response Time

Sample Query:
```
route_lookup_requests_time{interface_name=\"N7\",service_name=\"npcf-smpolicycontrol\",operation_name=\"Npcf_SMPolicyControl_UpdateNotify\",command=\"Notify\",status=\"201\"}
```

Labels:

- Label: `interface_name`

  Label Description: Interface name that connects PCF to other network functions

  Example: N7, N15, N25, N28, NRF

Labels:

- Label: `service_name`

Label Description: Name of the Service

Example: nnrf-nfm, nnrf-disc, npcf-am-policy-control, npcf-smpolicycontrol, nchf-spendinglimitcontrol

Labels:

- Label: `command`

Label Description: The Command

Example: Create, Update, Delete, Notify, Subscribe, Unsubscribe, Terminate, SmDataQuery, AmDataQuery, Discovery, DeRegister, Register, Heartbeat, Unsubscribe

Labels:

- Label: `operation_name`

Label Description: Name of the Operation

Example: Npcf_SMPolicyControl_Create, Npcf_SMPolicyControl_Update, Npcf_SMPolicyControl_Delete, Npcf_SMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Create, Npcf_AMPolicyControl_Update, Npcf_AMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Delete, Nchf_SpendingLimitControl_Subscribe, Nchf_SpendingLimitControl_Unsubscribe, Nchf_SpendingLimitControl_Notify, Nchf_SpendingLimitControl_Terminate, Nudr_Query_PolicyData_SM, Nudr_Query_PolicyData_AM, Nnrf_NfDiscovery, Nnrf_NfDeregister, Nnrf_NfRegister, Nnrf_NfHeartbeat, Nnrf_NfNotify, Nnrf_NfSubsribe, Nnrf_NfUnSubsribe

Labels:

- Label: `result_code`

Label Description: Status Code

Example: 200, 201, 202, 204, 205, 206, 301, 302, 303, 304, 305, 307, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 500, 501, 502, 503, 504, 505

route_lookup_requests_total

Description: Total Route Lookup Request Count

Sample Query:

`route_lookup_requests_total{interface_name=\"N7\",service_name=\"npcf-smpolicycontrol\",operation_name=\"Npcf_SMPolicyControl_UpdateNotify\",command=\"Notify\",status=\"201\"}`

Labels:

- Label: `interface_name`

Label Description: Interface name that connects PCF to other network functions

Example: N7, N15, N25, N28, NRF

Labels:

- Label: `service_name`

Label Description: Name of the Service

Example: nnrf-nfm, nnrf-disc, npcf-am-policy-control, npcf-smpolicycontrol, nchf-spendinglimitcontrol

Labels:

- Label: `command`

  Label Description: The Command

  Example: Create, Update, Delete, Notify, Subscribe, Unsubscribe, Terminate, SmDataQuery, AmDataQuery, Discovery, DeRegister, Register, Heartbeat, Unsubscribe

Labels:

- Label: `operation_name`

  Label Description: Name of the Operation

  Example: Npcf_SMPolicyControl_Create, Npcf_SMPolicyControl_Update, Npcf_SMPolicyControl_Delete, Npcf_SMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Create, Npcf_AMPolicyControl_Update, Npcf_AMPolicyControl_UpdateNotify, Npcf_AMPolicyControl_Delete, Nchf_SpendingLimitControl_Subscribe, Nchf_SpendingLimitControl_Unsubscribe, Nchf_SpendingLimitControl_Notify, Nchf_SpendingLimitControl_Terminate, Nudr_Query_PolicyData_SM, Nudr_Query_PolicyData_AM, Nnrf_NfDiscovery, Nnrf_NfDeregister, Nnrf_NfRegister, Nnrf_NfHeartbeat, Nnrf_NfNotify, Nnrf_NfSubsribe, Nnrf_NfUnSubsribe

Labels:

- Label: `result_code`

  Label Description: Status Code

  Example: 200, 201, 202, 204, 205, 206, 301, 302, 303, 304, 305, 307, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 500, 501, 502, 503, 504, 505

PCF Diameter Stats Category

diameter_late_responses_total

Description: Captures the total count of Diameter responses that the PCF Engine took time to process

Sample Query:
```
diameter_late_responses_total{command_code=\"AAR\",application=\"rx\",result_code=\"2001\"}
```

Labels:

- Label: `command_code`

  Label Description: Command Code

  Example: AAR,ASR,RAR,STR etc

- Label: `application`

  Label Description: Application Name

  Example: rx etc.

- Label: `result_code`

  Label Description: Result Code

  Example: 2001,5012,5065 etc.

diameter_request_total

Description: Collects the total count of Diameter request that the PCF Engine has processed.

Sample Query: `diameter_request_total{command_code=\"AAR\",application=\"rx\"}`

Labels:

- Label: `command_code`

  Label Description: Interface name that connects PCF to other network functions

  Example: AAR,ASR,RAR,STR etc

- Label: `application`

  Label Description: application

  Example: rx etc.

diameter_responses_seconds_total

Description: Captures the total amount of time (in seconds) that the PCF Engine took to process a diameter

Sample Query:
`diameter_responses_seconds_total{command_code=\"AAR\",application=\"rx\",result_code=\"2001\"}`

Labels:

- Label: `command_code`

  Label Description: Command Code

  Example: AAR,ASR,RAR,STR etc

Labels:

- Label: `application`

  Label Description: Application Name

  Example: rx etc.

- Label: `result_code`

  Label Description: Result Code

  Example: 2001,5012,5065 etc.

diameter_responses_total

Description: Records the total count of the Diameter responses that the Policy Engine has processed

Sample Query:
`diameter_responses_total{command_code=\"AAR\",application=\"rx\",result_code=\"2001\"}`

Labels:

- Label: `command_code`

  Label Description: Command Code

  Example: AAR,ASR,RAR,STR etc

Labels:

- Label: `application`

  Label Description: Application Name

  Example: rx etc.

Labels:

- Label: `result_code`

  Label Description: Result Code

  Example: 2001,5012,5065 etc.

dispatch_error_seconds_total

Description: Captures the total amount of time (in seconds) used to detect messages that encountered the dispatch error

Sample Query:
`dispatch_error_seconds_total{command_code=\"AAR\",application=\"rx\",error_code=\"5002\"}`

Labels:

- Label: `command_code`

  Label Description: Command Code

  Example: AAR,ASR,RAR,STR etc

Labels:

- Label: `application`

  Label Description: Application Name

  Example: rx etc.

Labels:

- Label: `error_code`

  Label Description: Error Code

  Example: 5002,5065,5012 etc.

dispatch_error_total

Description: Captures the total count of messages that encountered dispatch error through the GRPC Interface

Sample Query: `dispatch_error_total{command_code=\"AAR\",application=\"rx\",error_code=\"5002\"}`

Labels:

- Label: `command_code`

  Label Description: Command Code

  Example: AAR,ASR,RAR,STR etc

Labels:

- Label: `application`

  Label Description: Application Name

  Example: rx etc.

Labels:

- Label: `error_code`

  Label Description: Error Code

  Example: 5002,5065,5012 etc.

grpc_message_send_total

Description: Collects the total count of messages that are sent over the gRPC toward the PCF Engine

Sample Query:
```
grpc_message_send_total{command_code=\"AAR\",application=\"rx\",remote_service=\"diameter-engine\"}
```

Labels:

- Label: `command_code`

  Label Description: Command Code

  Example: AAR,ASR,RAR,STR etc

Labels:

- Label: `remote_service`

  Label Description: Remote Service

  Example: diameter-engine etc.

policy_engine_message_seconds_total

Description: Projects the total amount of time (in seconds) taken to process the round-trip responses for the PCF Engine GRPC messages

Sample Query: `policy_engine_message_seconds_total{command_code=\"AAR\",application=\"rx\"}`

Labels:

- Label: `command_code`

  Label Description: Command Code

  Example: AAR,ASR,RAR,STR etc

Labels:

- Label: `application`

  Label Description: Application Name

  Example: rx etc.

policy_engine_message_seconds_total

Description: Round trip response timeout total for Policy Engine GRPC message

Sample Query: `policy_engine_message_seconds_total{command_code=\"AAR\",application=\"rx\"}`

Labels:

- Label: `command_code`

  Label Description: Command Code

  Example: AAR,ASR,RAR,STR etc

Labels:

- Label: `application`

  Label Description: Application Name

  Example: rx etc.

policy_engine_message_total

Description: Round trip response total for Policy Engine GRPC message

Sample Query: `policy_engine_message_total{command_code=\"AAR\",application=\"rx\"}`

Labels:

- Label: `command_code`

  Label Description: Command Code

  Example: AAR,ASR,RAR,STR etc

Labels:

- Label: `application`

  Label Description: Application Name

  Example: rx etc.

process_message_seconds_total

Description: Projects the total amount of time (in seconds) used to process the messages through the Diameter interface

Sample Query: `process_message_seconds_total{command_code=\"AAR\",application=\"rx\"}`

Labels:

- Label: `command_code`

  Label Description: Command Code

  Example: AAR,ASR,RAR,STR etc

Labels:

- Label: `application`

Label Description: Application Name

Example: rx etc.

process_message_total

Description: Projects the total count of messages that were processed through the Diameter interface

Sample Query: `process_message_total{command_code=\"AAR\",application=\"rx\"}`

Labels:

- Label: `command_code`

  Label Description: Command Code

  Example: AAR,ASR,RAR,STR etc

Labels:

- Label: `application`

  Label Description: Application Name

  Example: rx etc.

route_avp_seconds_total

Description: Indicates the total time (in seconds) taken to route the AVP requests

Sample Query: `route_avp_seconds_total{command_code=\"AAR\",application=\"rx\",status=\"2001\"}`

Labels:

- Label: `command_code`

  Label Description: Command Code

  Example: AAR,ASR,RAR,STR etc

Labels:

- Label: `application`

  Label Description: Application Name

  Example: rx etc.

Labels:

- Label: `status`

  Label Description: Status

  Example: error,success

route_avp_total

Description: Captures the total number of the route AVP requests that are built

Sample Query: `route_avp_total{command_code=\"AAR\",application=\"rx\",status=\"success\"}`

Labels:

- Label: `command_code`

  Label Description: Command Code

  Example: AAR,ASR,RAR,STR etc

Labels:

- Label: `application`

  Label Description: Application Name

  Example: rx etc.

Labels:

- Label: `status`

  Label Description: Status

  Example: error,success

session_binding_lookup_seconds_total

Description: Captures the total time (in seconds) taken to complete the Session Binding Lookup Requests

Sample Query:
```
session_binding_lookup_seconds_total{command_code=\"AAR\",application=\"rx\",status=\"success\"}
```

Labels:

- Label: `command_code`

  Label Description: Command Code

  Example: AAR,ASR,RAR,STR etc

Labels:

- Label: `application`

  Label Description: Application Name

  Example: rx etc.

Labels:

- Label: `status`

  Label Description: Status

  Example: error,success

session_binding_lookup_total

Description: Projects the total number of the Session Binding Lookup Requests that were built

Sample Query:
```
session_binding_lookup_total{command_code=\"AAR\",application=\"rx\",status=\"success\"}
```

Labels:

- Label: `command_code`

  Label Description: Command Code

  Example: AAR,ASR,RAR,STR etc

Labels:

- Label: `application`

  Label Description: Application Name

  Example: rx etc.

Labels:

- Label: `status`

  Label Description: Status

  Example: error,success

Interface KPIs

N7 Data Create Success Rate without timeouts

Description: Counts the N7 data create success rate without timeouts.

Formula:

sum(irate(incoming_request_total{interface_name="N7", command="Create", result_code=~"2.*"}[1m])) * 100 / sum(irate(incoming_request_total{interface_name="N7", command="Create", result_code!="408"}[1m]))

N7 Data Create Success Rate with timeouts

Description: Counts the N7 data create success rate with timeouts.

Formula:

sum(irate(incoming_request_total{interface_name="N7", command="Create", result_code=~"2.*"}[1m])) * 100 / sum(irate(incoming_request_total{interface_name="N7", command="Create"}[1m]))

N7 Data Update Success Rate

Description: Counts the N7 data update success rate.

Formula:

sum(irate(incoming_request_total{interface_name="N7", command="Update", result_code=~"2.*"}[1m])) * 100 / sum(irate(incoming_request_total{interface_name="N7", command="Update"}[1m]))

N7 Data Delete Success Rate

Description: Counts the N7 data delete success rate.

Formula:

sum(irate(incoming_request_total{interface_name="N7", command="Delete", result_code=~"2.*"}[1m])) * 100 / sum(irate(incoming_request_total{interface_name="N7", command="Delete"}[1m]))

N7 Notify Success Rate

Description: Counts the N7 notification success rate.

Formula:

sum(irate(outgoing_request_total{interface_name="N7", command="Notify", result_code=~"2.*"}[1m])) * 100 / sum(irate(outgoing_request_total{interface_name="N7", command="Notify"}[1m]))

AAR Success Rate

Description: Counts the AAR success rate.

Formula:

sum(irate(diameter_responses_total{exported_application="Rx",command_code="AAA",result_code="2001"}[1m])*100) /sum(irate(diameter_responses_total{exported_application="Rx",command_code="AAA"}[1m]))

ASR Success Rate

Description: Counts the ASR success rate.

Formula:

sum(irate(diameter_responses_total{exported_application="Rx",command_code="ASA",result_code="2001"}[1m])*100) / sum(irate(diameter_responses_total{exported_application="Rx",command_code="ASA"}[1m]))

STR Success Rate

Description: Counts the STR success rate.

Formula:

sum(irate(diameter_responses_total{exported_application="Rx",command_code="STA",result_code="2001"}[1m])*100) / sum(irate(diameter_responses_total{exported_application="Rx",command_code="STA"}[1m]))

Rx RAR Success Rate

Description: Counts the Rx RAR success rate.

Formula:

sum(irate(diameter_responses_total{exported_application="Rx",command_code="RAA",result_code="2001"}[1m])*100) / sum(irate(diameter_responses_total{exported_application="Rx",command_code="RAA"}[1m]))

N28 Subscribe Success Rate without timeouts

Description: Counts the N28 subscribe success rate without timeouts.

Formula:

sum(irate(outgoing_request_total{interface_name="N28", command="Subscribe", result_code=~"2.*"}[1m])) * 100 / sum(irate(outgoing_request_total{interface_name="N28", command="Subscribe", result_code!="408"}[1m]))

N28 Subscribe Success Rate with timeouts

Description: Counts the N28 subscribe success rate with timeouts.

Formula:

sum(irate(outgoing_request_total{interface_name="N28", command="Subscribe", result_code=~"2.*"}[1m])) * 100 / sum(irate(outgoing_request_total{interface_name="N28", command="Subscribe"}[1m]))

N28 Notify Success Rate

Description: Counts the N28 notification success rate.

Formula:

sum(irate(incoming_request_total{interface_name="N28", command="Notify"}[1m])) * 100 / sum(irate(incoming_request_total{interface_name="N28", command="Notify", result_code!="408"}[1m]))

USD Query Success Rate

Description: Counts the interface success rate.

Formula:

sum(irate(message_total{type=~".*_ldap_query", status="success"}[1m])) *100 / sum(irate(message_total{type=~".*_ldap_query"}[1m]))

Note   Failed authorization on LDAP query will still count as USD_ldap*.success.

NAP Notification Success Rate

Description:

Formula: Counts the NAP notification success rate.

sum(irate(action_total{type="ldap_notify", status="success"}[1m])) *100 / sum(irate(action_total{type="ldap_notify"}[1m]))

GPASS Count

Description: Counts the number of GPASS triggered from the policy.

Formula:

sum(irate(action_total{type=resolve-virtual-service-gpass, status="success"}[1m]))

EPASS Count

Description: Counts the number of EPASS triggered from the policy.

Formula:

sum(irate(charging_rule_total{action="install", name="goodwill"}[1m]))

System KPIs

System Health Monitoring KPIs

The following table lists the KPIs and thresholds to track the overall performance of the PCF deployment, including information about the underlying hardware.

CPU Utilization

Description: CPU is a critical system resource. When the demand increases and CPU utilization exceeds 80% utilization, the efficiency of the CPU is reduced. When CPU utilization exceeds 80%, the application processing time will increase, message response will increase, and drops and timeouts will be seen.

Statistics/Formula: (avg without (cpu,mode)(irate(node_cpu_seconds_total{component="node-exporter",mode!="idle"}[1m])))

Warning Threshold: > 60% utilization over 60 second period (assuming that idle is less than 40%)

Major Threshold: > 80% utilization over 60 second period (assuming idle is less than 20%)

CPU Steal

Description: If multiple VMs on the same hypervisor and same hardware have concurrent CPU demands, the hypervisor will "steal" CPU from one VM to satisfy another VM CPU needs. If the CPU Steal statistic is non-zero, there is not enough CPU allocated for the VMs.

Statistics/Formula: (avg without (cpu,mode)(irate(node_cpu_seconds_total{component="node-exporter",mode="steal"}[1m])))

Warning Threshold: NA

Major Threshold: > 2% over 60 second period

CPU I/O Wait

Description: This monitors CPU I/O wait time. High CPU wait times may indicate CPUs waiting on disk access.

Statistics/Formula: (avg without (cpu,mode)(irate(node_cpu_seconds_total{component="node-exporter",mode="wait"}[1m])))

Warning Threshold: > 30 for more than 5 min

Major Threshold: > 50 for more than 10 min

Memory utilization

Description: Memory is a system resource, which needs to be less than 80%. The swap threshold has been reduced, and swapping should occur when the system resources are exhausted and memory utilization hits 99%.

Statistics/Formula: 100 - ((node_memory_MemAvailable_bytes * 100) / node_memory_MemTotal_bytes)

Warning Threshold: > 70% utilization over 60 second period

Major Threshold: > 80% utilization over 60 second period

Disk Utilization

Description: Disk storage is a critical system resource, and when file system utilization exceeds 90% utilization the system can become less efficient. When the file system utilization hits 100%, then application can stop functioning.

Statistics/Formula:

100 - ((node_filesystem_avail_bytes{mountpoint="/",fstype!="rootfs"} * 100) / node_filesystem_size_bytes{mountpoint="/",fstype!="rootfs"})

Warning Threshold: > 80% utilization

Major Threshold: > 90% utilization

In Queue

Description: These statistics monitors how long a message waits in the application queue, waiting to be serviced. The value should be 10ms all the time. higher values indicate the application is too slow, short of resources, or overwhelmed.

Statistics/Formula: sum(irate(input_queue_duration_seconds[1m])) / sum(irate(input_queue_total[1m]))

Warning Threshold: NA

Major Threshold: More than 10 ms over 60 seconds

Diameter 3xxx errors

Description: Diameter Too Busy 3xxx message indicate that the PCF is overwhelmed, or responding too slowly. This can be related to In Queue issues, system resources, database problems, network latency, or issues with other external nodes in the call flow.

Statistics/Formula: sum(irate(diameter_responses_total{result_code=~"3.*"}[1m])) *100/sum(irate(diameter_responses_total{result_code=~"2001"}[1m]))

Warning Threshold: > 0.5% Over 30 minute period

Major Threshold: > 1% Over 30 minute period

Diameter 5xxx errors

Description: Session Not Found and other Diameter 5xxx errors indicate a critical problem with the ability to process the incoming diameter message. This can be related to exhausted system resources, invalid session id or bad message structure, length, or content, or even database corruption.

Statistics/Formula: sum(irate(diameter_responses_total{result_code=~"3.*"}[1m])) *100/sum(irate(diameter_responses_total{result_code=~"2001"}[1m]))

Warning Threshold: > 0.5% Over 5 minute period

Major Threshold: > 1% Over 5 minute period

N7 5xx Errors

Description: N7 errors indicate that the PCF is unable to process N7 requests. This can be related to service timeout or service unavailable or an internal error.

Statistics/Formula: sum(irate(incoming_request_total{interface_name="N7", result_code=~"5.*"}[1m])) * 100 / sum(irate(incoming_request_total{interface_name="N7", result_code=~"2.*"}[1m]))

Warning Threshold: > 0.5% over 30 minute period

Major Threshold: > 1% over 30 minute period

N28 5xx Errors

Description: N28 errors indicate that the PCF is unable to process N28 requests. This can be related to service timeout or service unavailable or an internal error.

Statistics/Formula: sum(irate(outgoing_request_total{interface_name="N28", result_code=~"5.*"}[1m])) * 100 / sum(irate(outgoing_request_total{interface_name="N28", result_code=~"2.*"}[1m]))

Warning Threshold: > 0.5% over 30 minute period

Major Threshold: > 1% over 30 minute period

Active Session Count

Description: Number of total sessions currently active.

Statistics/Formula: avg(db_records_total{session_type="total"})

Warning Threshold:

>80% of the lessor of the dimensioned or licensed capacity for more than 1 hour

or

= 0 for more than 5 minutes

Major Threshold:

>80% of the lessor of the dimensioned or licensed capacity for more than 10 minutes

or

= 0 for more than 10 minutes

% of Messages dropped due to SLA timeout

Description: Messages dropped due to SLA timeouts indicate that the PCF is overwhelmed, or responding too slowly. This can be related to In Queue issues, system resources, database problems, network latency, or issues with other external nodes in the call flow.

Statistics/Formula:
sum(irate(input_queue_result_total{result="drop"}[1m]))*100/(sum(irate(incoming_request_total{result_code=~"2.*"}[1m])) + sum(irate(diameter_responses_total{result_code="2001"}[1m])))

Warning Threshold: > 0.5%

Major Threshold: > 1%

System Status KPIs

system_mode

Description: Indicates the current mode the system is running on.

Statistics/Formula: system_mode

Labels:

- Label: `0`

  Label Description: The system is in shutdown mode.

- Label: `1`

  Label Description: The system is running.

- Label: `2`

  Label Description: The system is under maintenance.

- Label: `-1`

  Label Description: The system mode is unknown.

system_synch_running

Description: Specifies whether the system configuration synch process is running or not.

Statistics/Formula: system_synch_running

Labels:

- Label: `1`

  Label Description: The system configuration sync process is running.

- Label: `0`

  Label Description: The system configuration sync process is not running.

system_running_percent

Description: Captures the percentage of the system currently in use.

Statistics/Formula: system_running_percent

System Configuration KPIs

system_configuration_bac kup_total

Description: Captures the total number of system configuration backups that are executed.

Statistics/Formula: irate(system_configuration_backup_total [1m])

Labels:

- Label: `status`

  Label Description: The status of the executed backups. For example, success or error.

configuration_change_total

Description: Captures the total number of configuration changes that are executed.

Statistics/Formula: sum(irate(configuration_change_total[1m]))

CPU Category

node_cpu_seconds_total

Description: Seconds the cpus spent in each mode

Metric Type:

Data Type:

Sample Query: `avg(irate(node_cpu_seconds_total{mode=~\"irq|softirq\"}[1m])) by (instance) * 100`

Labels:

- Label: `instance`

  Label Description: the virtual machine/instance

  Example: master-0, control-0, dra-director-1, etc

- Label: `job`

  Label Description: the name of job

  Example: node_exporter

- Label: `cpu`

  Label Description: the cpu number

  Example: cpu0, cpu1, etc

- Label: `mode`

  Label Description: the cpu mode

  Example: system, user, sotirq, irq, idle, iowait, etc

CPU Utilization

Description: CPU is a critical system resource. When the demand increases and CPU utilization exceeds 80% utilization, the efficiency of the CPU is reduced. When CPU utilization exceeds 80%, the application processing time will increase, message response will increase, and drops and timeouts will be seen.

Metric Type:

Data Type:

Sample Query: `100 - cpu.<cpuid>.idle`

Warning Threshold:

- > 60% utilization over 60 second period (assuming that idle is less than 40%).

Major Threshold:

- > 80% utilization over 60 second period (assuming idle is less than 20%).

CPU Steal

Description: If multiple VMs on the same hypervisor and same hardware have concurrent CPU demands, the hypervisor will "steal" CPU from one VM to satisfy another VM CPU needs. If the CPU Steal statistic is non-zero, there is not enough CPU allocated for the VMs.

Metric Type:

Data Type:

Sample Query: `cpu.<cpuid>.steal`

Major Threshold:

- \> 2% over 60 second period.

CPU I/O Wait

Description: This monitors CPU I/O wait time. High CPU wait times may indicate CPUs waiting on disk access.

Metric Type:

Data Type:

Sample Query: `cpu.<cpuid>.wait`

Warning Threshold:

- \> 30 for more than 5 min.

Major Threshold:

- \> 50 for more than 10 min.

Disk Category

node_disk_bytes_read

Description: This metrics gives the total number of bytes read successfully.

Metric Type:

Data Type:

Sample Query: `sum(irate(node_disk_bytes_read[1m])) by (instance)`

Labels:

- Label: `instance`

  Label Description: the virtual machine/instance

  Example: master-0, control-0, dra-director-1, etc

- Label: `job`

  Label Description: the name of job

  Example: node_exporter

- Label: `device`

  Label Description: the name of the disk device

  Example: vdb, vdd, sr0

node_disk_read_time_seconds_total

Description: This metrics gives the total number of seconds spent by all reads

Metric Type:

Data Type:

Sample Query: `sum(irate(node_disk_read_time_seconds_total[1m])) by (instance) / sum(irate(node_disk_reads_completed_total[1m])) by (instance)`

Labels:

- Label: `instance`

  Label Description: the virtual machine/instance

  Example: master-0, control-0, dra-director-1, etc

- Label: `job`

  Label Description: the name of job

  Example: node_exporter

- Label: `device`

  Label Description: the name of the disk device

  Example: vdb, vdd, sr0

node_disk_reads_completed_total

Description: This metrics gives the total number of reads completed successfully.

Metric Type:

Data Type:

Sample Query: `sum(irate(node_disk_reads_completed_total[1m])) by (instance)`

Labels:

- Label: `instance`

  Label Description: the virtual machine/instance

  Example: master-0, control-0, dra-director-1, etc

Labels:

- Label: `job`

  Label Description: the name of job

  Example: node_exporter

- Label: `device`

  Label Description: the name of the disk device

  Example: vdb, vdd, sr0

node_disk_write_time_seconds_total

Description: This metrics gives the total number of seconds spent by all writes

Metric Type:

Data Type:

Sample Query: `sum(irate(node_disk_write_time_seconds_total[1m])) by (instance) / sum(irate(node_disk_writes_completed_total[1m])) by (instance)`

Labels:

- Label: `instance`

  Label Description: the virtual machine/instance

  Example: master-0, control-0, dra-director-1, etc

Labels:

- Label: `job`

  Label Description: the name of job

  Example: node_exporter

Labels:

- Label: `device`

  Label Description: the name of the disk device

  Example: vdb, vdd, sr0

node_disk_writes_completed_total

Description: This metrics gives the total number of writes completed successfully.

Metric Type:

Data Type:

Sample Query: `sum(irate(node_disk_writes_completed[1m])) by (instance)`

Labels:

- Label: `instance`

  Label Description: the virtual machine/instance

  Example: master-0, control-0, dra-director-1, etc

Labels:

- Label: `job`

Label Description: the name of job

Example: node_exporter

Labels:

- Label: `device`

  Label Description: the name of the disk device

  Example: vdb, vdd, sr0

node_disk_written_bytes_total

Description: This metrics gives the total number of bytes written successfully.

Metric Type:

Data Type:

Sample Query: `sum(irate(node_disk_written_bytes_total[1m])) by (instance)`

Labels:

- Label: `instance`

  Label Description: the virtual machine/instance

  Example: master-0, control-0, dra-director-1, etc

Labels:

- Label: `job`

  Label Description: the name of job

  Example: node_exporter

Labels:

- Label: `device`

  Label Description: the name of the disk device

  Example: vdb, vdd, sr0

Disk Utilization

Description: Disk storage is a critical system resource, and when file system utilization exceeds 90% utilization the system can become less efficient. When the file system utilization hits 100%, then application can stop functioning.

Metric Type:

Data Type:

Sample Query: `df.<fs>.df_complex.free - df.<fs>.df_complex.used`

Warning Threshold:

- > 80% utilization.

Major Threshold:

- \> 90% utilization

File System Category

node_filesystem_free_bytes

Description: This metrics gives the total number of bytes of the free disk space available on the instance

Metric Type:

Data Type:

Sample Query: `sum(node_filesystem_free_bytes{mountpoint=\"/data\"}) by (device, instance)`

Labels:

- Label: `instance`

  Label Description: the virtual machine/instance

  Example: master-0, control-0, dra-director-1, etc

- Label: `job`

  Label Description: the name of job

  Example: node_exporter

- Label: `device`

  Label Description: the name of the disk device

  Example: /dev/vda3, /dev/vdb

- Label: `fstype`

  Label Description: the file system type

  Example: ext4

- Label: `mountpoint`

  Label Description: the file system mount directory

  Example: /data, /tootfs

node_filesystem_size_bytes

Description: This metrics gives the total number of bytes of the total disk space provisioned on the instance

Metric Type:

Data Type:

Sample Query: `sum(node_filesystem_size_bytes{mountpoint=\"/data\"}) by (device, instance)`

Labels:

- Label: `instance`

Label Description: the virtual machine/instance

Example: master-0, control-0, dra-director-1, etc

- Label: `job`

  Label Description: the name of job

  Example: node_exporter

- Label: `device`

  Label Description: the name of the disk device

  Example: /dev/vda3, /dev/vdb

- Label: `fstype`

  Label Description: the file system type

  Example: ext4

- Label: `mountpoint`

  Label Description: the file system mount directory

  Example: /data, /tootfs

Load Category

node_load1

Description: This metrics gives the 1m load average.

Metric Type: Gauge

Data Type: Float

Sample Query: `avg(irate(node_load1[1m])) by (instance)`

Labels:

- Label: `instance`

  Label Description: the virtual machine/instance

  Example: master-0, control-0, dra-director-1, etc

- Label: `job`

  Label Description: the name of job

  Example: node_exporter

node_load15

Description: This metrics gives the 15m load average.

Metric Type: Gauge

Data Type: Float

Sample Query: `avg(irate(node_load15[1m])) by (instance)`

Labels:

- Label: `instance`

  Label Description: the virtual machine/instance

  Example: master-0, control-0, dra-director-1, etc

- Label: `job`

  Label Description: the name of job

  Example: node_exporter

node_load5

Description: This metrics gives the 5m load average.

Metric Type: Gauge

Data Type: Float

Sample Query: `avg(irate(node_load5[1m])) by (instance)`

Labels:

- Label: `instance`

  Label Description: the virtual machine/instance

  Example: master-0, control-0, dra-director-1, etc

Labels:

- Label: `job`

  Label Description: the name of job

  Example: node_exporter

Memory Category

node_memory_MemFree_bytes

Description: This metrics gives the total number of bytes of the free memory available on the node

Metric Type:

Data Type:

Sample Query: `sum(node_memory_MemFree_bytes) by (instance)`

Labels:

- Label: `instance`

  Label Description: the virtual machine/instance

  Example: master-0, control-0, dra-director-1, etc

• Label: `job`

Label Description: the name of job

Example: node_exporter

node_memory_MemTotal_bytes

Description: This metrics gives the total number of bytes of the total memory provisioned on the node

Metric Type:

Data Type:

Sample Query: `sum(node_memory_MemTotal_bytes) by (instance)`

Labels:

• Label: `instance`

Label Description: the virtual machine/instance

Example: master-0, control-0, dra-director-1, etc

• Label: `job`

Label Description: the name of job

Example: node_exporter

Memory Utilization

Description: Memory is a system resource, which needs to be less than 80%. The swap threshold has been reduced for CPS, and swapping should occur when the system resources are exhausted and memory utilization hits 99%.

Metric Type:

Data Type:

Sample Query: `memory.free – memory.used`

Warning Threshold:

• > 70% utilization over 60 second period.

Major Threshold:

• > 80% utilization over 60 second period.

Network Category

node_network_receive_bytes_total

Description: This metrics gives the total number of bytes received over the network device

Metric Type:

Data Type:

Sample Query: `sum(irate(node_network_receive_bytes_total[1m])) by (device)`

Labels:

- Label: `instance`

  Label Description: the virtual machine/instance

  Example: master-0, control-0, dra-director-1, etc

- Label: `job`

  Label Description: the name of job

  Example: node_exporter

- Label: `device`

  Label Description: the name of the network device/interface

  Example: ens3, ens4

node_network_transmit_bytes_total

Description: This metrics gives the total number of bytes sent over the network device

Metric Type:

Data Type:

Sample Query: `sum(irate(node_network_transmit_bytes_total[1m])) by (device)`

Labels:

- Label: `instance`

  Label Description: the virtual machine/instance

  Example: master-0, control-0, dra-director-1, etc

- Label: `job`

  Label Description: the name of job

  Example: node_exporter

- Label: `device`

  Label Description: the name of the network device/interface

  Example: ens3, ens4

MIB Reference

CISCO-CNEE-MIB

ciscoCneeMIB Module Identity

Last Updated: "201910120000Z"

Organization: "Cisco Systems, Inc."

Contact Info: "Cisco Systems Customer Service Postal: 170 W Tasman Drive San Jose, CA 95134 USA Tel: +1 800 553-NETS"

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

{ ciscoMgmt 999 }

ciscoCneeMIBNotifs Object ID

{ ciscoCneeMIB 0 }

ciscoCneeMIBFaults Object ID

{ ciscoCneeMIB 1 }

ciscoCneeMIBConform Object ID

{ ciscoCneeMIB 2 }

cneeFaultId Object Type

Syntax: Octet string of 1-64 characters.

Max Access: not-accessible

Status: current

Description: Uniquely identify the fault within a monitored entity.

{ ciscoCneeMIBFaults 1 }

cneeFaultSource Object Type

Syntax: Octet string of 1-128 characters.

Max Access: not-accessible

Status: current

Description: Uniquely identify the monitored entity It can be a hostname or IP Address or human readable identity.

{ ciscoCneeMIBFaults 2 }

cneeFaultSeverity Object Type

Syntax: Octet string of 1-16 characters.

Max Access: not-accessible

Status: current

Description: Indicates the level of urgency for operator attention Refer 3GPP TS32.111-5 v9.0.0 section 4.3.

{ ciscoCneeMIBFaults 3 }

cneeFaultTime Object Type

Syntax: DateAndTime

Max Access: not-accessible

Status: current

Description: The date and time when the fault is detected.

{ ciscoCneeMIBFaults 4 }

cneeFaultType Object Type

Syntax: Octet string of 1-64 characters.

Max Access: not-accessible

Status: current

Description: Indicates the type of fault Refer 3GPP TS32.111-5 v9.0.0 section 4.3.

{ ciscoCneeMIBFaults 5 }

cneeFaultAdditionalInfo Object Type

Syntax: Octet string of 1-2048 characters.

Max Access: not-accessible

Status: current

Description: Additional Information about the fault.

{ ciscoCneeMIBFaults 6 }

cneeFaultClusterName Object Type

Syntax: Octet string of 1-128 characters.

Max Access: not-accessible

Status: current

Description: The cluster name associated to the fault.

{ ciscoCneeMIBFaults 7 }

cneeFaultNamespace Object Type

Syntax: Octet string of 1-128 characters.

Max Access: not-accessible

Status: current

Description: Identifies the namespace associated to the fault. This field is not always available for every fault.

{ ciscoCneeMIBFaults 8 }

cneeFaultHostname Object Type

Syntax: Octet string of 1-128 characters.

Max Access: not-accessible

Status: current

Description: Identifies the hostname or ip address associated with the fault. This field is not always available for every fault.

{ ciscoCneeMIBFaults 9 }

cneeFaultInstance Object Type

Syntax: Octet string of 1-128 characters.

Max Access: not-accessible

Status: current

Description: Identifies the instance associated to the fault. The instance is set by the alert rule creator and may not reference a host but could reference a process or KPI that is associated to the fault. This field is not always available for every fault

{ ciscoCneeMIBFaults 10 }

cneeVnfAlias Object Type

Syntax: Octet string of 1-128 characters.

Max Access: not-accessible

Status: current

Description: Alias for the monitored entity

{ ciscoCneeMIBFaults 11 }

cneeFaultActiveNotif Notification Type

Objects: cneeFaultId, cneeFaultSource, cneeFaultSeverity, cneeFaultTime, cneeFaultType, cneeFaultAdditionalInfo, cneeFaultClusterName, cneeFaultNamespace, cneeFaultHostname, cneeFaultInstance, cneeVnfAlias

Status: current

Description: This notification is generated by CNEE whenever a fault gets triggered.

{ ciscoCneeMIBNotifs 1 }

cneeFaultClearNotif Notification Type

Objects: cneeFaultId, cneeFaultSource, cneeFaultSeverity, cneeFaultTime, cneeFaultType, cneeFaultAdditionalInfo, cneeFaultClusterName, cneeFaultNamespace, cneeFaultHostname, cneeFaultInstance, cneeVnfAlias

Status: current

Description: This notification is generated by CNEE whenever a fault gets cleared.

{ ciscoCneeMIBNotifs 2 }

ciscoCneeMIBCompliances Object ID

{ ciscoCneeMIBConform 1 }

ciscoCneeMIBGroups Object ID

{ ciscoCneeMIBConform 2 }

cneeMIBCompliance Module Compliance

Status: current

Description: The compliance statement for entities that support the Cisco CNEE Managed Objects

Module: -- this module

Mandatory Groups: cneeMIBFaultGroup, cneeMIBNotificationGroup

{ ciscoCneeMIBCompliances 1 }

cneeMIBFaultGroup Object Group

Objects: cneeFaultId, cneeFaultSource, cneeFaultSeverity, cneeFaultTime, cneeFaultType, cneeFaultAdditionalInfo, cneeFaultClusterName, cneeFaultNamespace, cneeFaultHostname, cneeFaultInstance, cneeVnfAlias

Status: current

Description: The set of CNEE Fault groups defined by this MIB

{ ciscoCneeMIBGroups 1 }

cneeMIBNotificationGroup Notification Group

Notifications: cneeFaultActiveNotif, cneeFaultClearNotif

Status: current

Description: The set of CNEE notifications defined by this MIB

{ ciscoCneeMIBGroups 2 }

CISCO-SMI

ciscoProducts Object ID

Status: current

Description: ciscoProducts is the root OBJECT IDENTIFIER from which sysObjectID values are assigned. Actual values are defined in CISCO-PRODUCTS-MIB.

{ cisco 1 }

local Object ID

Status: current

Description: Subtree beneath which pre-10.2 MIBS were built.

{ cisco 2 }

temporary Object ID

Status: current

Description: Subtree beneath which pre-10.2 experiments were placed.

{ cisco 3 }

pakmon Object ID

Status: current

Description: reserved for pakmon

{ cisco 4 }

workgroup Object ID

Status: current

Description: subtree reserved for use by the Workgroup Business Unit

{ cisco 5 }

otherEnterprises Object ID

Status: current

Description: otherEnterprises provides a root object identifier from which mibs produced by other companies may be placed. mibs produced by other enterprises are typically implemented with the object identifiers as defined in the mib, but if the mib is deemed to be uncontrolled, we may reroot the mib at this subtree in order to have a controlled version.

{ cisco 6 }

ciscoSB Object ID

Status: current

Description: ciscoSB provides root Object Identifier for Management Information Base for products of Cisco Small Business. This includes products rebranded from linksys aquisition. MIB numbers under this root are managed and controlled by ciscosb_mib@cisco.com.

{ otherEnterprises 1 }

ciscoSMB Object ID

Status: current

Description: ciscoSMB provides root Object Identifier for Management Information Base for products of Cisco built for Small and Medium Business market.The MIB numbers under this root are managed and controlled by ciscosmb_mib@cisco.com

{ otherEnterprises 2 }

ciscoAgentCapability Object ID

Status: current

Description: ciscoAgentCapability provides a root object identifier from which AGENT-CAPABILITIES values may be assigned.

{ cisco 7 }

ciscoConfig Object ID

Status: current

Description: ciscoConfig is the main subtree for configuration mibs.

{ cisco 8 }

ciscoMgmt Object ID

Status: current

Description: ciscoMgmt is the main subtree for new mib development.

{ cisco 9 }

ciscoExperiment Object ID

Status: current

Description: ciscoExperiment provides a root object identifier from which experimental mibs may be temporarily based. mibs are typiically based here if they fall in one of two categories 1) are IETF work-in-process mibs which have not been assigned a permanent object identifier by the IANA. 2) are cisco work-in-process which has not been assigned a permanent object identifier by the cisco assigned number authority, typically because the mib is not ready for deployment. NOTE WELL: support for mibs in the ciscoExperiment subtree will be deleted when a permanent object identifier assignment is made.

{ cisco 10 }

ciscoAdmin Object ID

Status: current

Description: ciscoAdmin is reserved for administratively assigned OBJECT IDENTIFIERS, i.e. those not associated with MIB objects

{ cisco 11 }

ciscoModules Object ID

Status: current

Description: ciscoModules provides a root object identifier from which MODULE-IDENTITY values may be assigned.

{ cisco 12 }

lightstream Object ID

Status: current

Description: subtree reserved for use by Lightstream

{ cisco 13 }

ciscoworks Object ID

Status: current

Description: ciscoworks provides a root object identifier beneath which mibs applicable to the CiscoWorks family of network management products are defined.

{ cisco 14 }

newport Object ID

Status: current

Description: subtree reserved for use by the former Newport Systems Solutions, now a portion of the Access Business Unit.

{ cisco 15 }

ciscoPartnerProducts Object ID

Status: current

Description: ciscoPartnerProducts is the root OBJECT IDENTIFIER from which partner sysObjectID values may be assigned. Such sysObjectID values are composed of the ciscoPartnerProducts prefix, followed by a single identifier that is unique for each partner, followed by the value of sysObjectID of the Cisco product from which partner product is derived. Note that the chassisPartner MIB object defines the value of the identifier assigned to each partner.

{ cisco 16 }

ciscoPolicy Object ID

Status: current

Description: ciscoPolicy is the root of the Cisco-assigned OID subtree for use with Policy Management.

{ cisco 17 }

ciscoPIB Object ID

Status: current

Description: ciscoPIB is the root of the Cisco-assigned OID subtree for assignment to PIB (Policy Information Base) modules.

{ ciscoPolicy 2 }

ciscoPolicyAuto Object ID

Status: current

Description: ciscoPolicyAuto is the root of the Cisco-assigned OID subtree for OIDs which are automatically assigned for use in Policy Management.

{ cisco 18 }

ciscoPibToMib Object ID

Status: current

Description: ciscoPibToMib is the root of the Cisco-assigned OID subtree for MIBs which are algorithmically generated/translated from Cisco PIBs with OIDs assigned under the ciscoPIB subtree. These generated MIBs allow management entities (other the current Policy Server) to read the downloaded policy. By convention, for PIB 'ciscoPIB.x', the generated MIB shall have the name 'ciscoPibToMib.x'.

{ ciscoPolicyAuto 2 }

ciscoDomains Object ID

Status: current

Description: ciscoDomains provides a root object identifier from which different transport mapping values may be assigned.

{ cisco 19 }

ciscoCIB Object ID

Status: current

Description: ciscoCIB is the root of the Cisco-assigned OID subtree for assignment to MIB modules describing managed objects that part of the CPE automatic configuration framework.

{ cisco 20 }

ciscoCibMmiGroup Object ID

Status: current

Description: ciscoCibMmiGroup is the root of the Cisco-assigned OID subtree for assignment to MIB modules describing managed objects supporting the Modem Management Interface (MMI), the interface that facilitates CPE automatic configuration.

{ ciscoCIB 1 }

ciscoCibProvGroup Object ID

Status: current

Description: ciscoCibStoreGroup is the root of the Cisco-assigned OID subtree for assignment to MIB modules describing managed objects contributing to the Configuration Information Base (CIB).

{ ciscoCIB 2 }

ciscoPKI Object ID

Status: current

Description: ciscoPKI is the root of cisco-assigned OID subtree for PKI Certificate Policies and Certificate Extensions.

{ cisco 21 }

ciscoLDAP Object ID

Status: current

Description: ciscoLDAP is the root of the Cisco-assigned OID subtree for assignment to LDAP (Lightweight Directory Access Protocol) modules.

{ cisco 22 }

ciscoProxy Object ID

Status: current

Description: ciscoProxy OBJECT IDENTIFIERS are used to uniquely name party mib records created to proxy for SNMPv1.

{ ciscoAdmin 1 }

ciscoPartyProxy Object ID

{ ciscoProxy 1 }

ciscoContextProxy Object ID

{ ciscoProxy 2 }

ciscoRptrGroupObjectID Object ID

Status: current

Description: ciscoRptrGroupObjectID OBJECT IDENTIFIERS are used to uniquely identify groups of repeater ports for use by the SNMP-REPEATER-MIB (RFC 1516) rptrGroupObjectID object.

{ ciscoAdmin 2 }

ciscoUnknownRptrGroup Object ID

Status: current

Description: The identity of an unknown repeater port group.

{ ciscoRptrGroupObjectID 1 }

cisco2505RptrGroup Object ID

Status: current

Description: The authoritative identity of the Cisco 2505 repeater port group.

{ ciscoRptrGroupObjectID 2 }

cisco2507RptrGroup Object ID

Status: current

Description: The authoritative identity of the Cisco 2507 repeater port group.

{ ciscoRptrGroupObjectID 3 }

cisco2516RptrGroup Object ID

Status: current

Description: The authoritative identity of the Cisco 2516 repeater port group.

{ ciscoRptrGroupObjectID 4 }

ciscoWsx5020RptrGroup Object ID

Status: current

Description: The authoritative identity of the wsx5020 repeater port group.

{ ciscoRptrGroupObjectID 5 }

ciscoChipSets Object ID

Status: current

Description: Numerous media-specific MIBS have an object, defined as an OBJECT IDENTIFIER, which is the identity of the chipset realizing the interface. Cisco-specific chipsets have their OBJECT IDENTIFIERS assigned under this subtree.

{ ciscoAdmin 3 }

ciscoChipSetSaint1 Object ID

Status: current

Description: The identity of the Rev 1 SAINT ethernet chipset manufactured for cisco by LSI Logic.

{ ciscoChipSets 1 }

ciscoChipSetSaint2 Object ID

Status: current

Description: The identity of the Rev 2 SAINT ethernet chipset manufactured for cisco by LSI Logic.

{ ciscoChipSets 2 }

ciscoChipSetSaint3 Object ID

Status: current

Description: The identity of the Rev 3 SAINT ethernet chipset manufactured for cisco by Plessey.

{ ciscoChipSets 3 }

ciscoChipSetSaint4 Object ID

Status: current

Description: The identity of the Rev 4 SAINT ethernet chipset manufactured for cisco by Mitsubishi.

{ ciscoChipSets 4 }

ciscoTDomains Object ID

{ ciscoDomains 99999 }

ciscoTDomainUdpIpv4 Object ID

Status: current

Description: The UDP over IPv4 transport domain. The corresponding transport address is of type CiscoTAddressIPv4.

{ ciscoTDomains 1 }

ciscoTDomainUdpIpv6 Object ID

Status: current

Description: The UDP over IPv6 transport domain. The corresponding transport address is of type CiscoTAddressIPv6 for global IPv6 addresses and CiscoTAddressIPv6s for scoped IPv6 addresses.

{ ciscoTDomains 2 }

ciscoTDomainTcpIpv4 Object ID

Status: current

Description: The TCP over IPv4 transport domain. The corresponding transport address is of type CiscoTAddressIPv4.

{ ciscoTDomains 3 }

ciscoTDomainTcpIpv6 Object ID

Status: current

Description: The TCP over IPv6 transport domain. The corresponding transport address is of type CiscoTAddressIPv6 for global IPv6 addresses and CiscoTAddressIPv6s for scoped IPv6 addresses.

{ ciscoTDomains 4 }

ciscoTDomainLocal Object ID

Status: current

Description: The Posix Local IPC transport domain. The corresponding transport address is of type CiscoTAddressLocal. The Posix Local IPC transport domain incorporates the well known UNIX domain sockets.

{ ciscoTDomains 5 }

ciscoTDomainClns Object ID

Status: current

Description: The CLNS transport domain. The corresponding transport address is of type CiscoTAddressOSI.

{ ciscoTDomains 6 }

ciscoTDomainCons Object ID

Status: current

Description: The CONS transport domain. The corresponding transport address is of type CiscoTAddressOSI.

{ ciscoTDomains 7 }

ciscoTDomainDdp Object ID

Status: current

Description: The DDP transport domain. The corresponding transport address is of type CiscoTAddressNBP.

{ ciscoTDomains 8 }

ciscoTDomainIpx Object ID

Status: current

Description: The IPX transport domain. The corresponding transport address is of type CiscoTAddressIPX.

{ ciscoTDomains 9 }

ciscoTDomainSctpIpv4 Object ID

Status: current

Description: The SCTP over IPv4 transport domain. The corresponding transport address is of type CiscoTAddressIPv4.

Reference: RFC 2960 - Stream Control Transmission Protocol. R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, V. Paxson. October 2000.

{ ciscoTDomains 10 }

ciscoTDomainSctpIpv6 Object ID

Status: current

Description: The SCTP over IPv6 transport domain. The corresponding transport address is of type CiscoTAddressIPv6 for global IPv6 addresses and CiscoTAddressIPv6s for scoped IPv6 addresses.

Reference: RFC 2960 - Stream Control Transmission Protocol. R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, V. Paxson. October 2000.

{ ciscoTDomains 11 }