

UCC 5G PCF Metrics Reference, Release 2026.04

First Published: 2026-11-05

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

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2026 Cisco Systems, Inc. All rights reserved.



CONTENTS

PREFACE

About this Guide v

Conventions Used v

Contacting Customer Support vi

CHAPTER 1

Prometheus and Grafana 1

Feature Summary 1

Summary Data 1

Feature Description 1

How it Works 2

Managing the PCF Statistics 2

Viewing the Statistics 2

Accessing the Grafana Dashboard 2

Viewing the PCF Dashboard 3

Deploy Database Monitoring Dashboard 3

Running a Query in Grafana 4

Configuring Autorefresh 5

Exporting and Importing Dashboards 5

Exporting Dashboards 5

Importing Dashboards 6

Exporting the Graph Data to CSV 6

Filtering the Graphs 6

Bulk Statistics 6

Configuring the Bulk Statistics Collection 7

Sample Queries for Bulk Statistics 8

Sample Configuration 8

Sample Bulk Statistics Configuration 9

CHAPTER 2 **Statistics and KPI Reference** 27

PCF Statistics 27

PCF Service Category 27

PCF Endpoint Requests Category 35

Model-D interface KPIs 49

PCF Diameter Stats Category 51

Interface KPIs 57

System KPIs 60

System Health Monitoring KPIs 60

System Status KPIs 63

System Configuration KPIs 64

CPU Category 64

Disk Category 65

File System Category 68

Load Category 69

Memory Category 70

Network Category 71

CHAPTER 3 **MIB Reference** 73

CISCO-CNEE-MIB 73

CISCO-SMI 73



About this Guide

This preface describes the *Ultra Cloud Core 5G Policy Control Function*, 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.

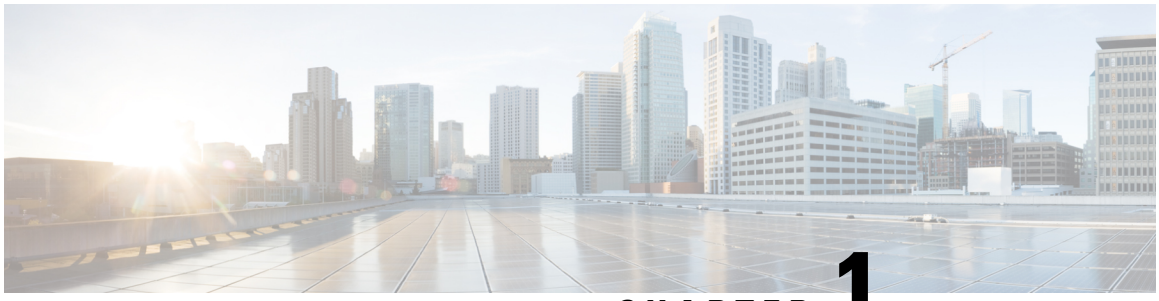
Typeface Conventions	Description
Text represented as a `screen display`	This typeface represents displays that appear on your terminal screen, for example: `Login:`
Text represented as **commands**	This typeface represents commands that you enter, for example: **show ip access-list** This document always 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 applicable 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.



CHAPTER 1

Prometheus and Grafana

- Feature Summary, on page 1
- Feature Description, on page 1
- Managing the PCF Statistics, on page 2
- Bulk Statistics, on page 6

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:

1. Viewing the Statistics, on page 2
2. Accessing the Grafana Dashboard, on page 2
3. Viewing the PCF Dashboard, on page 3
4. Running a Query in Grafana, on page 4
5. Configuring Autorefresh, on page 5
6. Exporting and Importing Dashboards, on page 5

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.

Deploy Database Monitoring Dashboard

Starting with release 2026.01.0, you can optionally configure a new Database (DB) Monitoring dashboard that provides dedicated monitoring capabilities for key PCF data stores, such as MongoDB replica sets.

Table 2: Feature History

Feature Name	Release Information	Description
Deploy Database Monitoring Dashboard	2026.01.0	You can enable or disable the deployment of a specific DB Monitoring dashboard (Grafana JSON) by setting a configuration value. As a result, the dashboard is deployed only when needed, which optimizes resource usage and provides critical insights into database health and usage. The Database Monitoring Dashboard provides dedicated monitoring capabilities for key PCF data stores, such as MongoDB replica sets. Command Introduced: **testing enable-optional-dashboards [true | false]**.

Information About Deployment of DB Monitoring Dashboard

This feature introduces a new DB Monitoring dashboard, presented as a Grafana JSON, which includes panels to display MongoDB health and usage metrics. This dashboard is designed to be deployed conditionally. The deployment process leverages the `pcf-dashboard` Helm chart, which incorporates conditional logic. The decision to include or exclude the DB dashboard JSON during deployment is driven by a specific configuration value within the `ops-center`.



Note The deployment of existing standard dashboards, such as application, JVM, Diameter, and KPI metrics, is not affected. These dashboards are always deployed, regardless of this new feature's configuration.

Benefits of Deployment of DB Monitoring Dashboard

- **Dedicated DB Monitoring:**

  Provides a specialized dashboard for monitoring key PCF data stores like MongoDB replica sets, offering insights into their health and usage.

  .

- **Resource Optimization:**

  Allows the DB Monitoring dashboard to be deployed only when explicitly enabled, potentially reducing resource consumption when it's not required.

- **Configurability:**

  Enhances the configurability of the deployment process, giving users control over which dashboards are active.

- **Zero Impact:**

  Ensures that the introduction and conditional deployment of this new dashboard have no impact on existing metrics, dashboards, or operational workflows.

Configure Conditional Deployment of DB Monitoring Dashboard

- **Enable the DB Monitoring Dashboard:**

  To deploy the DB Monitoring dashboard alongside the standard dashboards, set the `enable-optional-dashboards` configuration to `true`:

  ```
  testing enable-optional-dashboards true
  ```

- **Disable the DB Monitoring Dashboard:**

  To deploy only the standard dashboards and exclude the DB Monitoring dashboard, set the `enable-optional-dashboards` configuration to `false`:

  ```
  testing enable-optional-dashboards false
  ```

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 27.

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 27 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 3: PCF Sample Queries

Query	Description
incoming_request_total{interface_name="N7""N5", command="Create"}	Fetches the total incoming create request for the N7 and N5 interfaces.
incoming_request_total{interface_name="N7""N5", command="Update"}	Collects the total incoming update request for the N7 and N5 interfaces.
diameter_requests_total{command_code="AAR"}	Collects the total number of Diameter requests that PCF has processed.
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""N5", command="DeRegister"}	Collects the total count of the outgoing deregistration requests that NRF and N5 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 queryRx_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
```

N5_Sessions

```
bulk-stats query N5_Sessions
expression sum(db_records_total{session_type=\"N5_5G_TGPP\"} OR on () vector (0)) by
(namespace)
label (session_type)
exit
```

APN_Sessions

```
bulk-stats query APN_Sessions
expression sum(db_records_total{session_type=~"APN-. *"} OR on () vector (0)) by
(namespace, session_type)
label (session_type)
exit
```

DNN_Sessions

```
bulk-stats query DNN_Sessions
expression sum(db_records_total{session_type=~"DNN-. *"} OR on () vector (0)) by
(namespace, session_type)
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

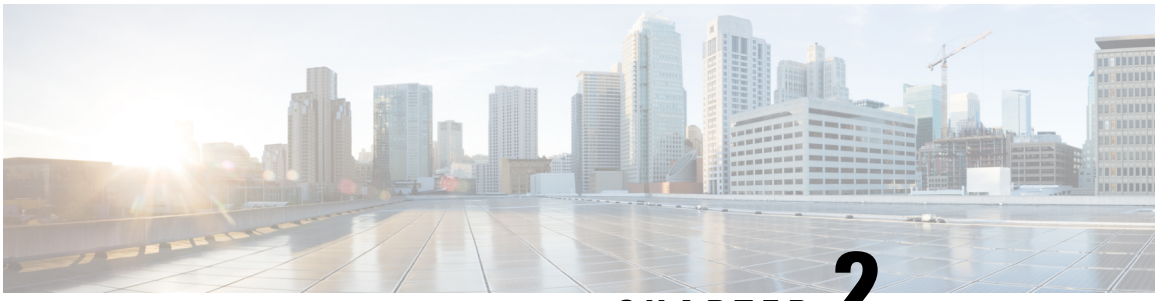
```
bulk-stats query statefulset-ready-percent
expression kube_statefulset_status_replicas_ready/kube_statefulset_status_replicas*100
label (statefulset)
exit
```

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
```



CHAPTER 2

Statistics and KPI Reference

- PCF Statistics, on page 27
- Interface KPIs, on page 57
- System KPIs, on page 60

PCF Statistics

PCF Service Category

charging_rule_base_total

Description: Total number of charging rule

Sample Query:

```
sum(irate(charging_rule_base{interface=\"N7\"\"N5\",name=~\".*\",action=\"install\"}[1m]))
by (interface,name,action)
```

Labels:

- Label: `interface`

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

  Example: N7, N5, 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\"\"N5\",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, N5, 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\"\"N5\",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,N5, 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.

unifiedapi_total

Description: Captures total count of API request received.

Sample Query: `sum(rate(unifiedapi_total{type=~".*request"}[1m])>0) by (type)`

Labels:

- Label: `method`

  Label Description: HTTP methods

  Example: POST

Labels:

- Label: `interface`

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

  Example: soap

Labels:

- Label: `type`

  Label Description: API request type

  Example: CreateSubscriberRequest.request

unifiedapi_response_total

Description: Captures success message count of given API type and request type.

Sample Query:
`sum(rate(unifiedapi_response_total{interface=~".",type=~".",status="success"}[1m])>0) by (interface,type,status)`

Labels:

- Label: `method`

  Label Description: HTTP methods

  Example: POST

Labels:

- Label: `interface`

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

  Example: soap

- Label: `type`

  Label Description: API type

  Example: CreateSubscriberRequest

- Label: `status`

  Label Description: Status of message as error or success

  Example: success,error

unifiedapi_response_duration_seconds

Description: Captures total milliseconds of successful API request processed by API type and request type.

Sample Query: `sort(sum(rate(unifiedapi_response_duration_seconds[1m])) by (type) / sum(rate(unifiedapi_response_total[1m])) by (type))`

Labels:

- Label: `method`

  Label Description: HTTP methods

  Example: POST

Labels:

- Label: `interface`

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

  Example: soap

Labels:

- Label: `type`

Label Description: API request type

Example: CreateSubscriberRequest

api_router_total

Description: Captures total count of getEntry/setKeys/updateKeys/ removeKeys/isDuplicateKey Request received and result type for ApiRouter calls.

Sample Query: `sum(rate(api_router_total{method=~".",status=~"."} [1m])>0) by (method,status)`

Labels:

- Label: `method`

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

  Example: updateKeys

Labels:

- Label: `status`

  Label Description: Status of message as error or success

  Example: success,error

PCF Endpoint Requests Category

"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"

- 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"

all_registration_failure_total

Description: "Registration failure to all NRFs"

Sample Query: `"all_registration_failure_total{pod_namespace=\"pcf*\", pod_name=\"pcf-rest-ep*\",all_registration_failure=\"*\"}"`

labels:

- Label: `"pod_namespace"`

  Label Description: "Pod namespace that connects PCF to other network functions"

  Example: "PCF, PCF-CLUSTER1"

- Label: `"pod_name"`

  Label Description: "Name of the Pod"

  Example: "pcf-rest-ep-6475f5b8b9-fgx47"

- Label: `"all_registration_failure"`

  Label Description: "registration failure count for all NRFs"

  Example: "34,64,100"

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"

inbound_request_threshold_exceeded

Description: "Total Inbound Threshold Requests due to Overload"

Sample Query:
"inbound_request_threshold_exceeded{interface_name=\"N7\",service_name=\"npcf-smpolicycontrol\",operation_name=\"Npcf_SMPolicyControl_Delete\",command=\"Delete\",action=\"REJECT\"}"

labels:

- Label: "interface_name"

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

  Example: "N7, N15"

- Label: "service_name"

  Label Description: "Name of the Service"

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

- 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"

- Label: "action"

  Label Description: "Discard behaviour"

  Example: "REJECT, DROP"

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\",userAgent=\"PCF\"}"

labels:

- Label: "interface_name"

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

  Example: "N7, N15, N25, N28, NRF"

- Label: "service_name"

  Label Description: "Name of the Service"

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

- 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"

- Label: `"userAgent"`

  Label Description: "Name of the user agent - NF"

  Example: "SMF, NRF, AF"

incoming_request_time

Description: "Total Response Time for Incoming Requests"

Sample Query:

"incoming_request_time{interface_name=\"N7\",service_name=\"pcf-policycntl\",operation_name=\"Npcf_SMPolicyControl_Delete\",command=\"Delete\",result_code=\"204\",peer_host_port=\"192.168.120.562\",dir=\"ss\"}"

labels:

- Label: `"interface_name"`

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

  Example: "N7, N15, N25, N28, NRF"

- Label: `"service_name"`

  Label Description: "Name of the Service"

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

- 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"

- 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"

- Label: " peer_host_port"

  Label Description: "peer host and port"

  Example: "192.168.72.81:9003"

- Label: "dnn"

  Label Description: "Data Network Name"

  Example: "sos"

incoming_request_total

Description: "Total Incoming Request Count"

Sample Query:
"incoming_request_total{interface_name=\"N7\",service_name=\"pcf-smpolicycontrol\",operation_name=\"Npcf_SMPolicyControl_Delete\",command=\"Delete\",result_code=\"204\",peer_host_port=\"192.168.72.81:9003\",dnn=\"sos\"
}"

labels:

- Label: "interface_name"

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

  Example: "N7, N15, N25, N28, NRF"

- Label: "service_name"

  Label Description: "Name of the Service"

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

- 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"

- 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"

- Label: `" peer_host_port"`

  Label Description: "peer host and port"

  Example: "192.168.72.81:9003"

- Label: `"dnn"`

  Label Description: "Data Network Name"

  Example: "sos"

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"

- Label: `"service_name"`

  Label Description: "Name of the Service"

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

- 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"

- 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=\"pcf-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"

- Label: "service_name"

  Label Description: "Name of the Service"

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

- 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"

- 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=\"pcf-smpolicycontrol\",operation_name=\"Npcf_SMPolicyControl_UpdateNotify\",command=\"Notify\",result_code=\"404\",peer_host_ip=\"192.168.1.20:5682\",le=\"+Inf\"}"

labels:

- Label: `"interface_name"`

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

  Example: "N7, N15, N25, N28, NRF"

- Label: `"service_name"`

  Label Description: "Name of the Service"

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

- 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"

- 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"

- Label: `" peer_host_port"`

  Label Description: "peer host and port"

  Example: "192.168.72.81:9003"

- Label: `"dnn"`

  Label Description: "Data Network Name"

  Example: "sos"

outgoing_request_total

Description: "Total Outgoing Request Count"

Sample Query:

"outgoing_request_total{interface_name=\"N7\",service_name=\"npcf-smpolicycontrol\",operation_name=\"Npcf_SMPolicyControl_UpdateNotify\",command=\"Notify\",result_code=\"404\",peer_host_port=\"192.168.1.20:5062\",dnn=\"sos\"}"

labels:

- Label: `"interface_name"`

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

Example: "N7, N15, N25, N28, NRF"

- Label: `"service_name"`

  Label Description: "Name of the Service"

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

- 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, Nnrf_NFUpdate"

- 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"

- Label: `" peer_host_port"`

  Label Description: "peer host and port"

  Example: "192.168.72.81:9003"

- Label: `"dnn"`

  Label Description: "Data Network Name"

  Example: "sos"

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"

- Label: `"service_name"`

  Label Description: "Name of the Service"

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

- 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"

- 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"

- Label: `"service_name"`

  Label Description: "Name of the Service"

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

- 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"

- 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"

restep_leadership_change_total

Description: "Leadership change count for a pod"

Sample Query: `"restep_leadership_change_total{pod_namespace=\"pcf*\", pod_name=\"pcf-rest-ep*\",leadership_change_count=\"*\"}"`

labels:

- Label: `"pod_namespace"`

  Label Description: "Pod namespace that connects PCF to other network functions"

  Example: "PCF, PCF-CLUSTER1"

- Label: `"pod_name"`

  Label Description: "Name of the Pod"

  Example: "pcf-rest-ep-6475f5b8b9-fgx47"

- Label: `"leadership_change_count"`

  Label Description: "Leadership change count for a pod"

  Example: 1,2,3,4

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=\"pcf-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"

- Label: "service_name"

  Label Description: "Name of the Service"

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

- 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"

- 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"

- Label: "service_name"

  Label Description: "Name of the Service"

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

- 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"

- 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"

- Label: `"service_name"`

  Label Description: "Name of the Service"

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

- 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"

- 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"

Model-D interface KPIs

outgoing_request_routing_total

Description: "Total Request Count over Model-D interface."

Sample Query:

```
"sum(cnf_request_total{interface_name=\"Model-D\",service_name=\"npcf-policycontrol\",operation_name=\"Npcf_SMPolicyControl_Create\",command=\"Create\",result_code=\"200\",peer_host_port=\"192.168.2.81:9103\",dir=\"ss\"})"
```

labels:

- Label: `"interface_name"`

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

  Example: "Model-D"

- Label:`"service_name"`

  Label Description: Name of the Service

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

- 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"

- Label:`"command"`

  Label Description: "The Command"

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

- Label: `"routing_mode"`

  Label Description: "Routing mode selected for the interface"

  Example: "scp, direct, direct_fallback"

- Label: `"result_code"`

  Label Description: "Status Code"

  Example: "200, 201, 202, 204, 400, 401, 403, 404, 500, 502, 503"

outgoing_request_routing_time

Description: "Total Response Time for Model-D Requests"

Sample Query:

```
"sum(cnf_request_time{interface_name=\"Model-D\",service_name=\"npcf-policycontrol\",operation_name=\"Npcf_SMPolicyControl_Update\",command=\"Update\",result_code=\"200\",peer_host_port=\"192.168.2.81:9103\",dir=\"ss\"})
```

labels:

- Label: `"interface_name"`

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

  Example: "Model-D"

- Label: `"service_name"`

  Label Description: "Name of the Service"

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

- 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"

- Label: `"command"`

  Label Description: "The Command"

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

- Label: `"routing_mode"`

  Label Description: "Routing mode selected for the interface"

  Example: "scp, direct, direct_fallback"

- Label: `"result_code"`

  Label Description: "Status Code"

  Example: "200, 201, 202, 204, 400, 401, 403, 404, 500, 502, 503"

model_d_fallback_total

Description: "Total Failed Request Count over Model-D interface"

Sample Query:
"modeld_request_failure_total{interface_name=\"Model-D\",service_name=\"pcf-smpolicycontrol\",operation_name=\"Npcf_SMPolicyControl_Delete\",command=\"Delete\",result_code=\"500\"}"

labels:

- Label: `"interface_name"`

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

  Example: "Model-D"

- Label: `"service_name"`

  Label Description: "Name of the Service"

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

- 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"

- Label: `"reason"`

  Label description: "fallback mode reason"

  Example: "scp_exhausted", "scp_timeout", "scp_5xx", "scp_unavailable"

PCF Diameter Stats Category

diameter_requests_total

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

Sample Query: `'diameter_requests_total{command_code="AAR",application="rx",dnn="sos"}'`

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.

- Label: `dnn`

  Label Description: Data Network Name

  Example: sos

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",dnn="sos"}'`

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.

- Label: `dnn`

  Label Description: Data Network Name

  Example: sos

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_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",dnn="sos"}'
```

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.

- Label: `dnn`

  Label Description: Data Network Name

  Example: sos

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

- Label: `application`

  Label Description: Application Name

  Example: rx etc.

- Label: `status`

  Label Description: Status

  Example: error,success

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

- Label: `application`

  Label Description: Application Name

  Example: rx etc.

- 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

- Label: `application`

  Label Description: Application Name

  Example: rx etc.

- 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

- Label: `application`

  Label Description: Application Name

  Example: rx etc.

- Label: `status`

  Label Description: Status

  Example: error,success

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

- 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

- Label: `application`

  Label Description: Application Name

  Example: rx 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

- Label: `application`

  Label Description: Application Name

  Example: rx etc.

- Label: `error_code`

  Label Description: Error Code

  Example: 5002,5065,5012 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

- Label: `application`

  Label Description: Application Name

  Example: rx etc.

- Label: `error_code`

  Label Description: Error Code

  Example: 5002,5065,5012 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

- Label: `application`

  Label Description: Application Name

  Example: rx 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

- 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

- Label: `application`

  Label Description: Application Name

  Example: rx 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

- Label: `remote_service`

  Label Description: Remote Service

  Example: diameter-engine etc.

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]))

N5 Notify Event Success Rate

Description: Counts the N5 notification event success rate.

Formula:

sum(irate(outgoing_request_total{interface_name="N5", command="Notify_Event", result_code=~"2.*"}[1m])) * 100 / sum(irate(outgoing_request_total{interface_name="N5", command="Notify_Event"}[1m]))

N5 Notify Terminate Success Rate

Description: Counts the N5 notifiy terminate success rate.

Formula:

sum(irate(outgoing_request_total{interface_name="N5", command="Notify_Terminate", result_code=~"2.*"}[1m])) * 100 / sum(irate(outgoing_request_total{interface_name="N5", command="Notify_Terminate"}[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

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

Disk Category

node_disk_bytes_read

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

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

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.

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

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.

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.

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

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

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

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

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

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

Network Category

node_network_receive_bytes_total

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

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

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`

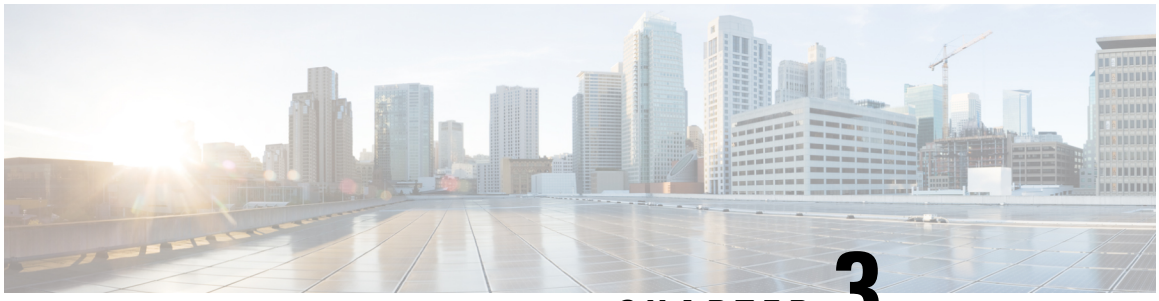
  Label Description: the name of job

  Example: node_exporter

- Label: `device`

  Label Description: the name of the network device/interface

Example: ens3, ens4



CHAPTER 3

MIB Reference

- CISCO-CNEE-MIB, on page 73
- CISCO-SMI, on page 73

CISCO-CNEE-MIB

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

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

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

CISCO-SMI

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

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

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