Troubleshoot SMF Call Flow Issues with Custom PromQL Queries

Contents

Introduction

This document describes how to use Grafana/Prometheus in Cisco SMF to create custom queries in order to troubleshoot call flow related issues.

Abbreviations

SMF	Session Management Function
UDM	Unified Data Management
AMF	Access and Mobility Function
PDU	Protocol Data Unit

Why Customize Queries to Troubleshoot SMF Call Flow Issues?

While, the inbuilt dashboards provide great graphs with respect to important KPIs and node health statistics, in order to use the full potential of PromQL queries and grafana to troubleshoot regular problem scenarios, custom queries play an important role. Custom promql queries and graphs add more versatility and convenience to isolate a specific failure.

Benefits of inbuilt dashboards:

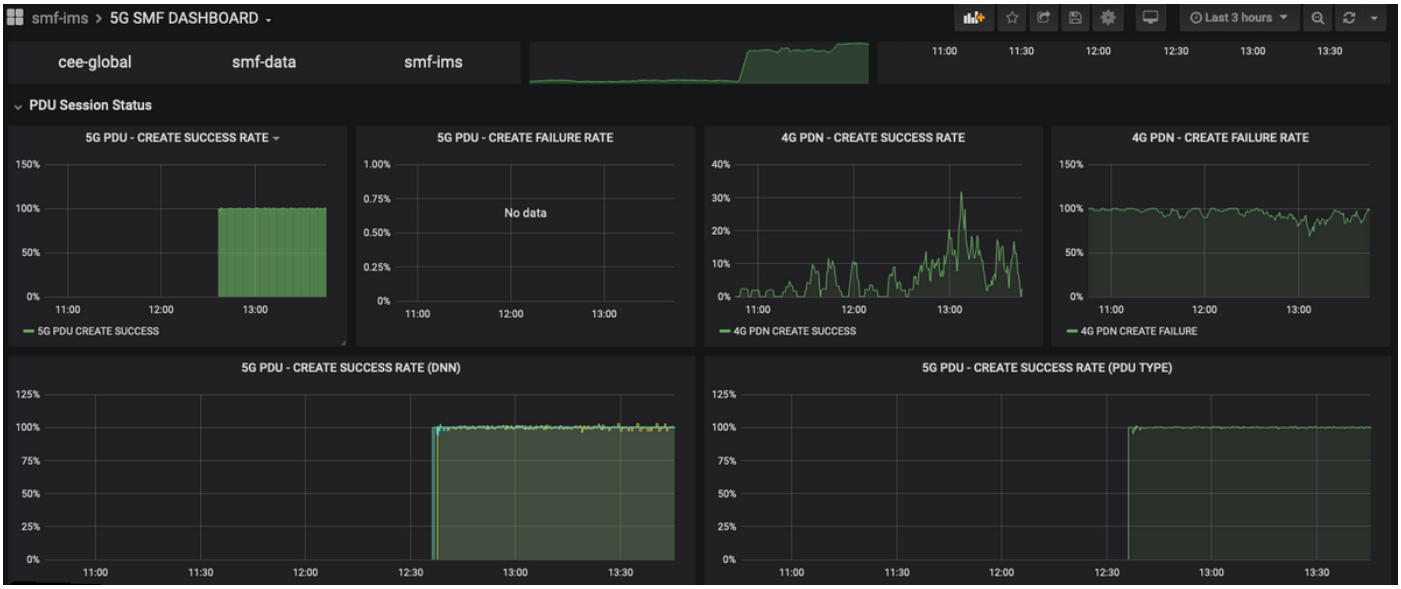
- Grafana provides an easy to use and graphical interface to browse SMF statistics.
- There are inbuilt grafana dashboards available to check most of the KPIs and statistics.

Example:

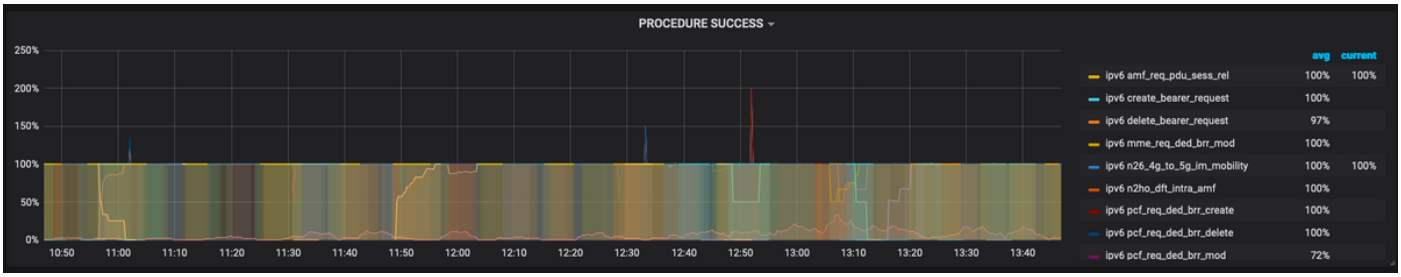
5G SMF Dashboard

- 5G PDU Create Failure/Success Rate

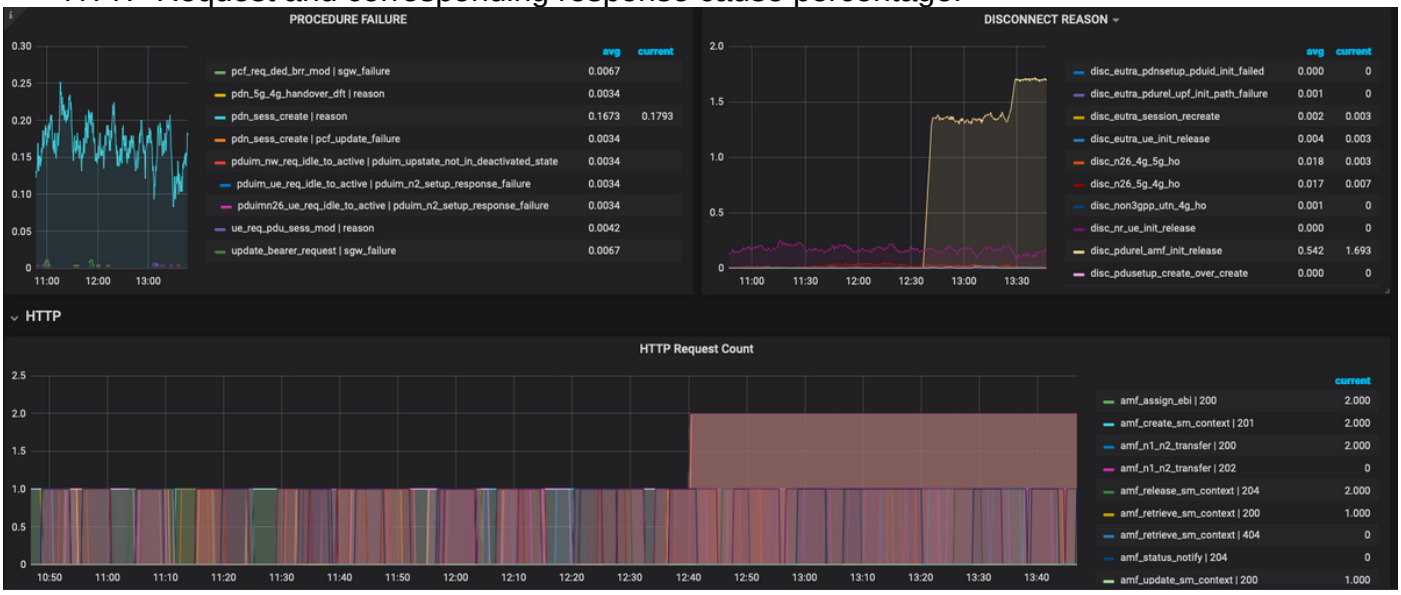
- 4G PDN Create Failure/Success Rate



- Per Procedure Success Rate



- Per Procedure Failure cause percentage.
- Disconnect Reason percentage.
- HTTP Request and corresponding response cause percentage.



To troubleshoot further:

- The available dashboard and panels are mostly about percentages and KPIs. While you investigate further, one might need to look at granular details to identify the particular scenario

and message which triggered this failure.

- The customized queries using specific regular expressions shall help to correlate these statistics and isolate the trigger.
- These queries can be used to plot graphs in SMF grafana or in the offline grafana with the metrics dump from the tac-debug package.
- The range of metrics associated with different services can be used and also can be filtered through label key/value pairs to troubleshoot the specific scenario.

Grafana and Prometheus

Grafana

“Grafana is open source visualization and analytics software. It allows you to query, visualize, alert on, and explore your metrics no matter where they are stored.”

Cisco SMF uses inbuilt grafana to plot the real-time statistics data from application containers.

Prometheus

Prometheus provides a multi-dimensional data model with time series data identified by metric name and key/value pairs and a flexible Query language named PromQL to access these data.

Prometheus is used for gathering statistics/counters from the microservices.

Metrics- They are the identifiers of the Time series statistics.

Labels- Metrics are consists of Labels. Which are basically the key-value pairs? The combinations of labels for a particular metric identifies a particular instance of time-series data

Example:

```
smf_service_stats{app_name="SMF",cluster="Local",data_center="DC",dnn="intershat",emergency_call="false",instance_id="0",pdu_type
="ipv4",procedure_type="pdu_sess_create",qos_5qi="",rat_type="",reason="",service_name="smf-
service",status="attempted",up_state=""} 43
```

The metric “smf_service_stats” highlighted in green, has many labels, which are highlighted in yellow.

Using these label key/value pairs, one can select a particular series of data.

PromQL Query

Prometheus provides a functional query language called PromQL. The inbuilt functions are available in PromQl (eg. Sum(), by(), count() etc) lets us select particular time-series data in a graphical or tabular format.

Example:

```
sum(smf_service_stats{status="success"}) by (procedure_type)
```

This example selects data from smf_service_stats metric by procedure_type where status = “success”

sum (calculate sum over dimensions)

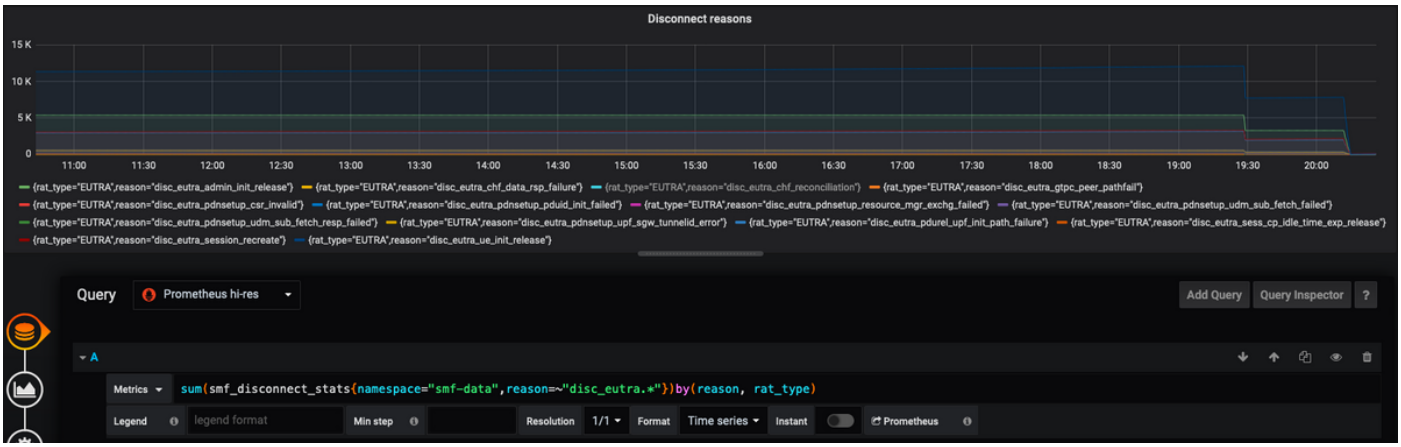
by(Groups the output by labels)

Filters can be used inside sum using Label key/value pairs to further filter the graphs.

Example 1:

```
sum(smf_disconnect_stats{namespace="smf-data",reason=~"disc_eutra.*"})by(reason, rat_type)
```

Here namespace **smf-data** is selected and as reason, all the disconnect reason starting with disc_eutra (i.e. 4G disconnect reasons) shall be considered.



Example 2:

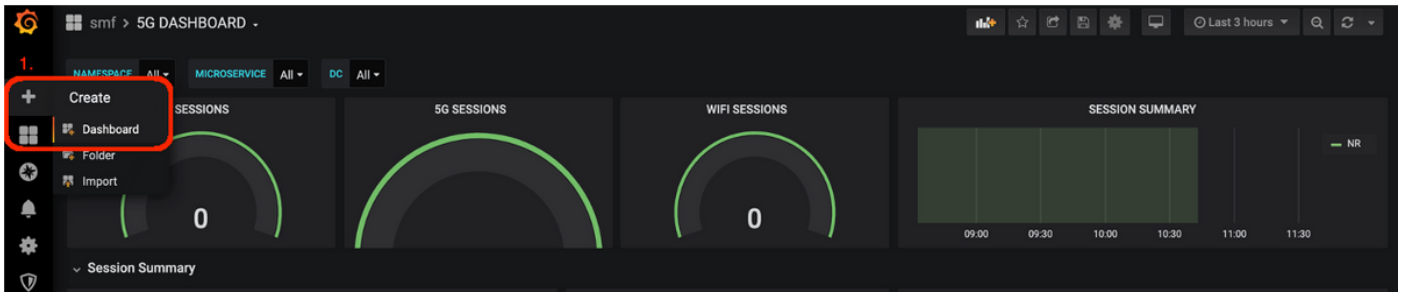
```
sum(smf_restep_http_msg{namespace="smf-data", api_name=~"sdm_.*"})
by(api_name,message_direction,response_status,response_cause)
```

This query shall plot SMF - UDM sdm-subscription messages with the response cause.

How to Create a Dashboard and Panel?

In order to add a **New Dashboard**.

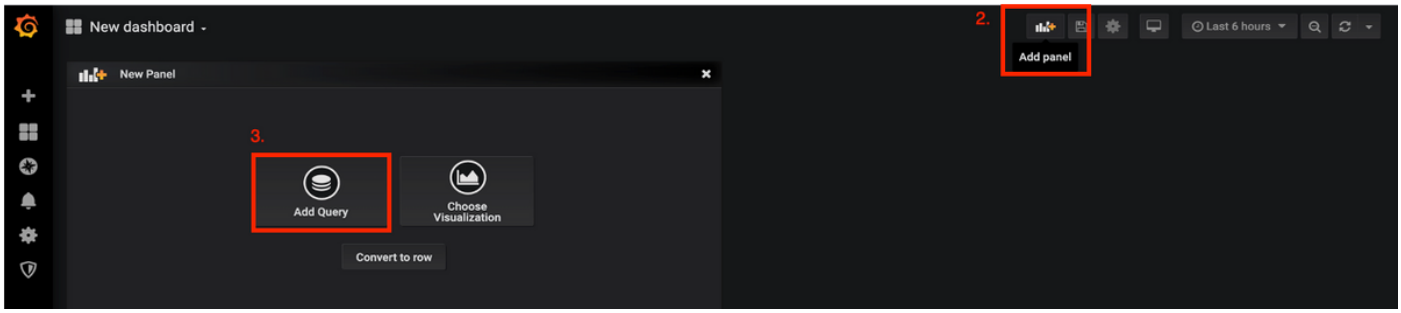
Step 1. Navigate to **Create > Dashboard**, as shown in this image.



In order to add New Panel- Add Query.

Step 2. Navigate to **Add Panel** option on the top to add a new panel.

Step 3. Select the **Add Query** button.

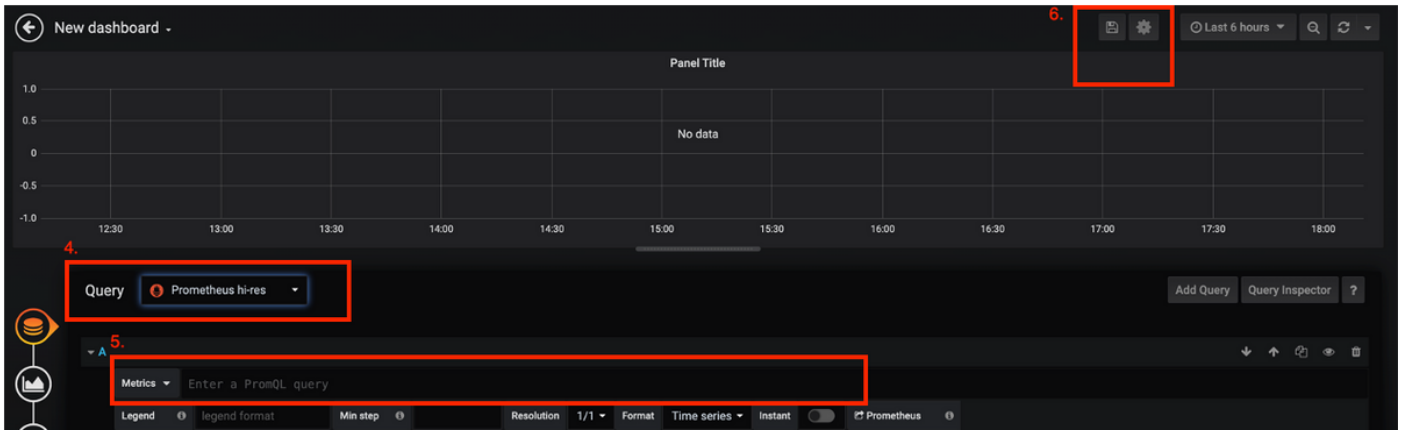


Select Query Type- Prometheus hi-res.

Step 4. Select **Prometheus hi-res** option in Query Drop down list.

Step 5. Then add the promql query in the given box.

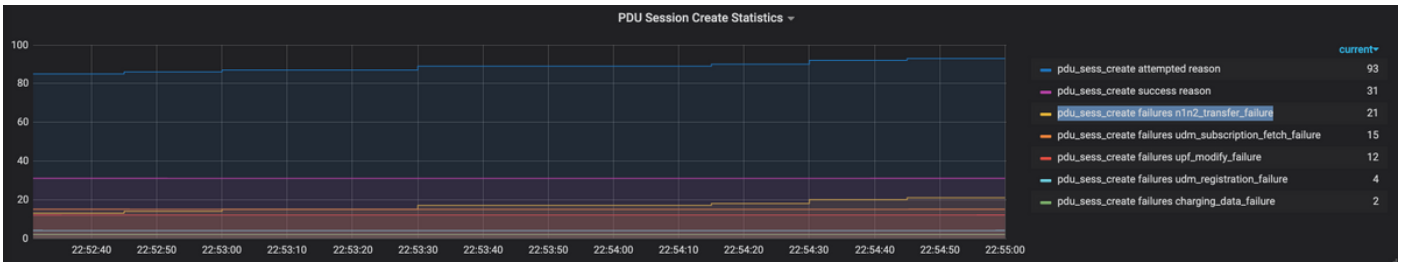
Step 6. Save the panel.



Example: Use the Customized Query and Graphs to Troubleshoot

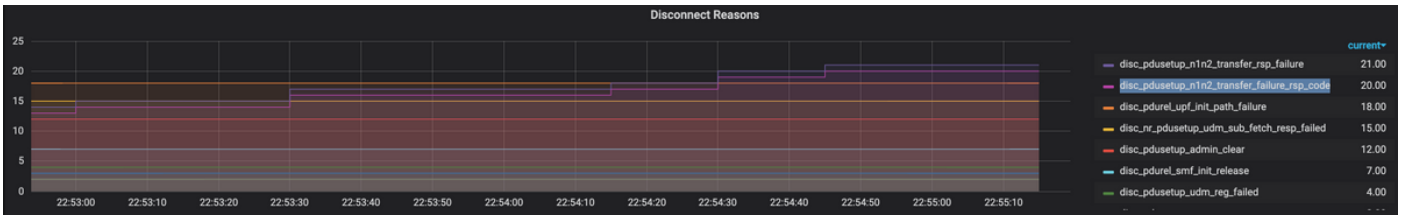
PDU Session Establishment Failure – N1N2 Response Failure

Step 1. KPI Dip Observation and identify the PDU session create failure.




```
Query: sum by (procedure_type, pdu_type, status, reason)
(smf_service_stats{namespace="smf",procedure_type="pdu_sess_create"})
```

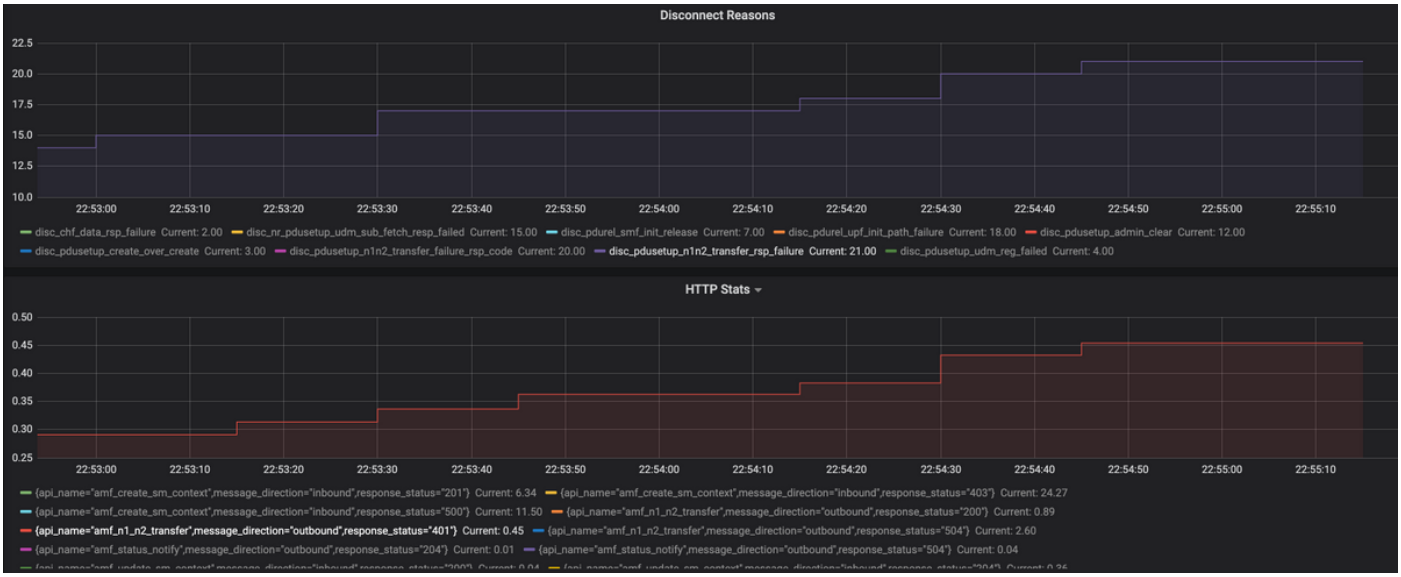
Step 2. The Failure Cause is “n1n2_transfer_failure_rsp_code”. Let’s look at the Disconnect Reasons:




```
Query: sum(smf_disconnect_stats{namespace=”smf"}) by (reason)
```

Step 3. The Disconnect reason “disc_pdusetup_n1n2_transfer_rsp_failure” indicates negative response from AMF peer. Since SMF-AMF interaction is over HTTP service-based interface, HTTP Stats need to be looked at further (metric: smf_restep_http_msg)

The HTTP statistics indicate that during the Failure SMF has received an HTTP status code 401 – Unauthorized from AMF



`Query: sum(smf_restep_http_msg{namespace="smf"}) by(api_name,message_direction,response_status)`

Important metrics to troubleshoot:

```
smf_disconnect_stats

smf_proto_pfcp_msg_total

smf_service_stats

smf_restep_http_msg

smf_n1_message_stats

smf_proto_pfcp_msg_total

nodemgr_msg_stats

nodemgr_gtpc_msg_stats

chf_message_stats

policy_msg_processing_status

procedure_protocol_total

procedure_service_total
```

Further information about SMF Metrics:

As demonstrated in these examples, one can plot their own custom graphs as and when required for the specific failure scenario to correlate different messages and isolate the failure. Such queries can be run in local systems as well after the metric data from Tac_debug_pkg is mounted on local grafana.