



SON Dashboards – Functional Description
SON Analytics Control Center

Date First Issued	December 18, 2024
Revision Number:	1.0
Revision Date	December 18, 2024

Document Information

The information in these materials is confidential and proprietary to T-Mobile USA, Inc. These materials are authorized for the use of T-Mobile USA service providers and their employees and agents, solely for the purposes of the agreement under which these materials are provided. The rights granted hereunder constitute a limited, nonexclusive, revocable license and not a transfer of title. Authorized T-Mobile USA service providers and their employees and agents may view, copy or print pages of these materials solely for the purposes set forth herein, but may not otherwise use, modify, copy, print, display, reproduce, distribute, transmit, publish, license, sublicense or create derivative works from these materials in whole or in part, or remove any copyright or other proprietary notices set forth herein, without the express written permission of T-Mobile USA.

The information in these materials is subject to change without notice. T-Mobile USA's liability for any errors in these materials is limited to the documentary correction of errors. T-Mobile USA WILL NOT BE RESPONSIBLE IN ANY EVENT FOR ERRORS IN THESE MATERIALS OR FOR ANY DAMAGES, INCIDENTAL OR CONSEQUENTIAL (INCLUDING MONETARY LOSSES), that might arise from the use of these materials or the information in them.

T-MOBILE, the T-MOBILE logo and the WORLD CLASS logo are registered or unregistered trademarks of Deutsche Telekom AG.

Acknowledgements

The following individuals have contributed to this document:

Nilson Peres

Revision Code

All modifications to this document must be made by the work stream Project Manager and recorded in the Document History section below. The version number of the document will reflect the modifications by following the format Revision x, y, where:

X is the first digit, incremented for changes of substance, i.e. technical/procedural issues.

Y is the second digit, incremented when editorial only changes have been incorporated.

Revision History

Revision	Date	Author	Information
1.0	2024-12-18	Nilson Peres	Initial documentation

Table of Contents

1.	Summary	5
1.1.	Short Summary	5
1.2.	Overview.....	5
1.2.1.	Files Processed per Host.....	5
1.2.2.	Total Files Processed per Hour (last 24 hours).....	8
1.2.3.	Last Update per Table.....	10
1.2.4.	Instance Data Processed per Hour (last 24 hours).....	10
1.2.5.	Instance Data Processed per SON Server (last 24 hours).....	12
1.2.6.	Processing Queue per Table	13
1.2.7.	Time to run per Script per Host (in minutes).....	14
1.2.8.	External Tables Refreshed per Day	15
1.2.9.	Managed Tables Refreshed per Day.....	16
1.2.10.	Topology Data per Market and TOPOLOGY_DB_CELL_COUNT	17
1.2.11.	Cell OSS Info per Server and OSS_DB_CELL_COUNT.....	18

1. Summary

1.1. Short Summary

Write in one or just a few sentences what the automation does.

This dashboard is a centralized view of all information required to monitor SON Analytics framework.

1.2. Overview

Write a longer explanation of what the automation does, which systems it uses and how it accomplishes its work.

Using Snowflake Dashboards and multiple queries it is possible to monitor data loading across the main tables in SON Analytics.

Currently, the following monitoring are available:

1.2.1. Files Processed per Host

SON File Processing depends on the following scripts:

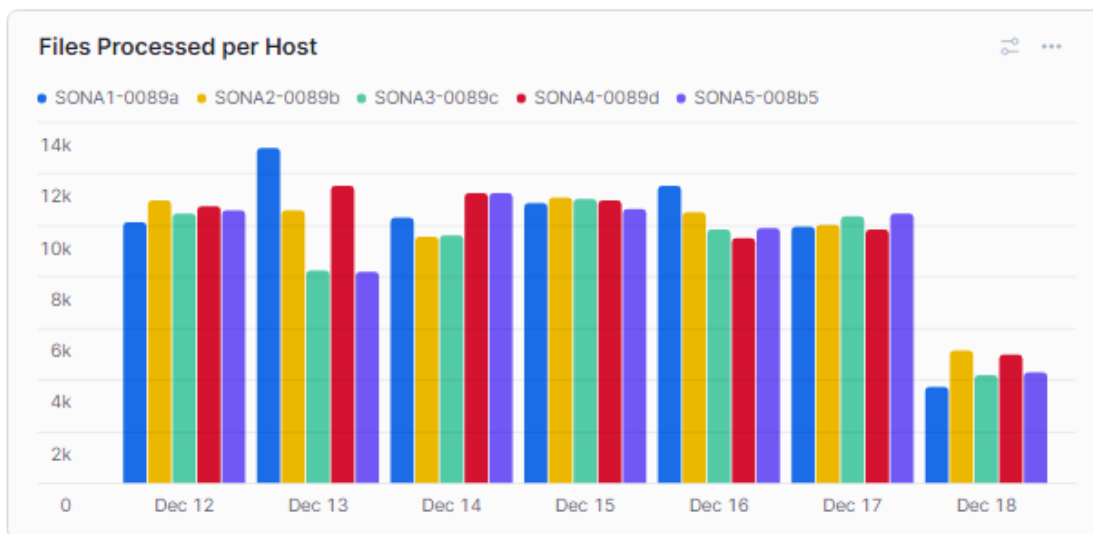
- instance_detector (to detect new instances),
- instance_files_detector (to detect the files related to instances),
- instance_files_downloader (to download the detected files) and
- instance_files_parser (to finally process the files).

The first two – instance_detector and instance_files_detector – run in only 1 host, so any issue with these 2 processing scripts would impact the number of files processed for all hosts...

The other two – instance_files_downloader and instance_files_parser – are split between multiple hosts (virtual machines).

NOTE: Currently there are available 10 VMs for processing, but only 5 VMs are being used, thus, the Files Processed per Host view shows only 5 VMs.

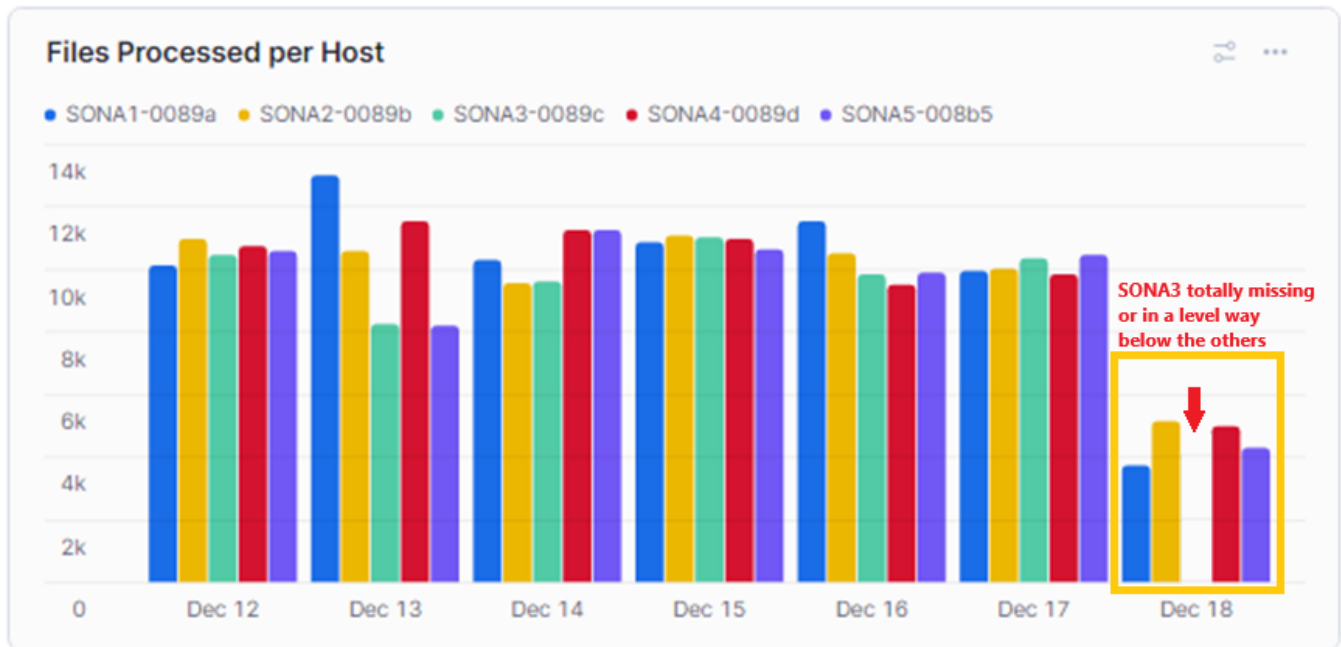
Example:



The bars indicate the number of files processed per day per host.

How this dashboard helps identifying issues?

If for any host the bar is at a level very below the others, such as in the following example, then it is possible that there is a problem with this host file processing.



Possible problems:

- instance_files_downloader OR instance_files_parser may be stuck, meaning that these scripts are not doing any progress on processing

Investigation Suggested:

- Remote connection to host (using MobaXterm, Remote Desktop Manager or another solution)
- Run the following command to check how long the script is running:

```
ps aux | grep instance_files_downloader or
```

```
ps aux | grep instance_files_parser
```
- In the results, check for how long the script is running

```
[svc_prd_edennetssupp@spolled0089a sonadm]$ ps aux | grep instance_files_parser
```

USER	PR	PPID	VSZ	RES	%CPU	%MEM	WCHAN	COMMAND
svc_prd+	4117800	0.3	0.5	1264264	180796	?		0:04 python3 instance_files_parser.py -v
svc_prd+	4129352	0.0	0.3	1264240	126956	?		0:00 python3 instance_files_parser.py -v
svc_prd+	4129360	2.8	0.5	1375036	188448	?		0:12 python3 instance_files_parser.py -v
svc_prd+	4129363	89.4	0.7	1844296	257352	?		6:11 python3 instance_files_parser.py -v
svc_prd+	4129365	1.5	0.5	1364708	186912	?	S1	0:06 python3 instance_files_parser.py -v
svc_prd+	4129368	0.8	0.5	1271220	164156	?	S1	0:03 python3 instance_files_parser.py -v
svc_prd+	4129371	0.6	0.4	1268000	159636	?	S1	0:02 python3 instance_files_parser.py -v
svc_prd+	4129374	1.4	0.5	1362356	184396	?	S1	0:06 python3 instance_files_parser.py -v
svc_prd+	4129377	1.5	0.5	1361996	184092	?	S1	0:06 python3 instance_files_parser.py -v
svc_prd+	4129380	0.5	0.4	1259468	158132	?	S1	0:02 python3 instance_files_parser.py -v
svc_prd+	4129383	1.7	0.5	1366596	183840	?	S1	0:07 python3 instance_files_parser.py -v
svc_prd+	4129386	2.0	0.5	1368484	188848	?	S1	0:08 python3 instance_files_parser.py -v
svc_prd+	4129389	0.8	0.4	1268036	159680	?	S1	0:03 python3 instance_files_parser.py -v
svc_prd+	4129392	89.0	0.8	1856764	289572	?	R1	6:09 python3 instance_files_parser.py -v
svc_prd+	4129395	3.2	0.5	1379440	189568	?	S1	0:13 python3 instance_files_parser.py -v
svc_prd+	4129398	1.1	0.5	1356052	180376	?	S1	0:04 python3 instance_files_parser.py -v

This is the column to check.
Value is in minutes.

- If the script is running for hours and hours – e.g. 800 minutes, 350 minutes, etc – there is a considerable possibility for the script to be stuck. Usually, the script runs in iterations of less than 30 minutes.

Fix Suggested:

- Restart the process to reactivate file processing, by running the following:

```
pkill -u svc_prd_edennetssupp -f instance_files_downloader.py
```

```
pkill -u svc_prd_edennetssupp -f instance_files_parser.py
```

- Process will start again in the next minute.
- Check logs for more details on where the script has been stuck.

NOTE: In general, the script should never get stuck, it should be a very rare case. But, if it does occur repeatedly, a deeper investigation on root causes is required.

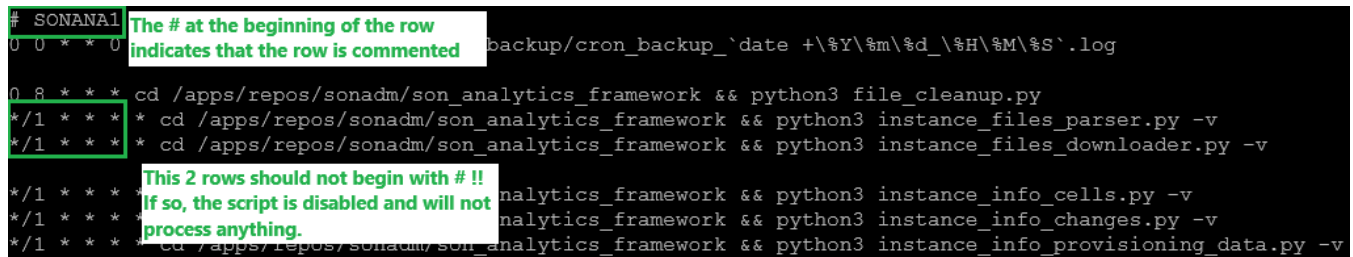
- ii. instance_files_downloader OR instance_files_parser may be disabled

Investigation Suggested:

- Remote connection to host (using MobaXterm, Remote Desktop Manager or another solution)
- Run the following command to check if scripts are enabled:

```
crontab -l
```

- In the results, check if the rows for instance_files_downloader and instance_files_parser are NOT commented:



```
# SONANA1 backup/cron_backup `date +%Y%m%d_%H%M%S`.log
0 0 * * 0
0 8 * * * cd /apps/repos/sonadm/son_analytics_framework && python3 file_cleanup.py
*/1 * * * * cd /apps/repos/sonadm/son_analytics_framework && python3 instance_files_parser.py -v
*/1 * * * * cd /apps/repos/sonadm/son_analytics_framework && python3 instance_files_downloader.py -v
*/1 * * * * cd /apps/repos/sonadm/son_analytics_framework && python3 instance_info_cells.py -v
*/1 * * * * cd /apps/repos/sonadm/son_analytics_framework && python3 instance_info_changes.py -v
*/1 * * * * cd /apps/repos/sonadm/son_analytics_framework && python3 instance_info_provisioning_data.py -v
```

The # at the beginning of the row indicates that the row is commented

This 2 rows should not begin with # !! If so, the script is disabled and will not process anything.

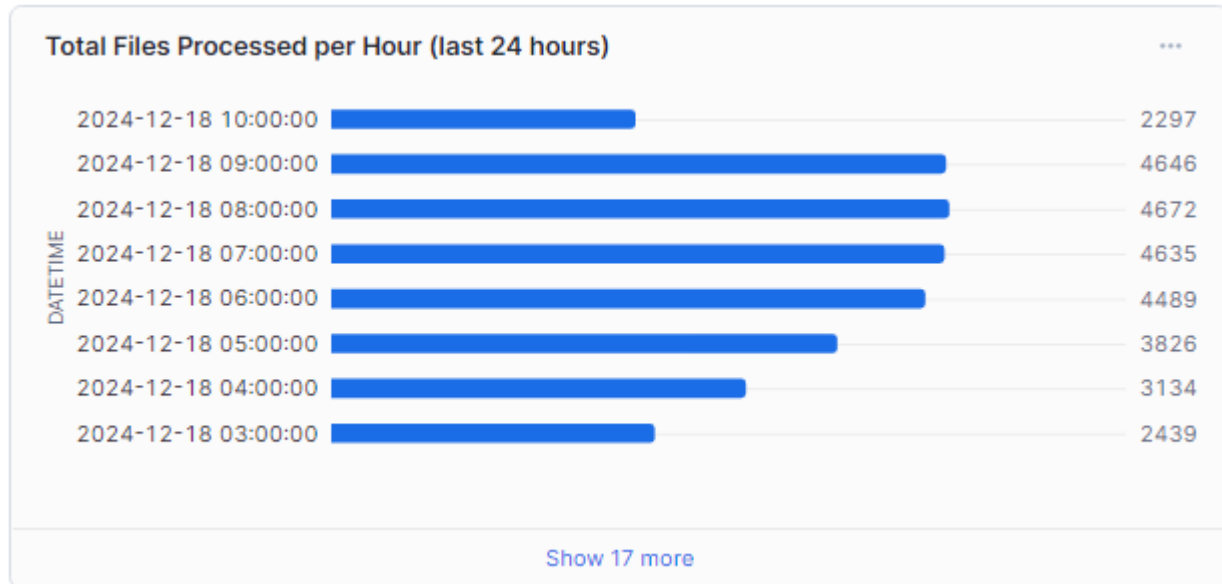
Suggested Fix:

- Reactivate the script by editing the cronjob:

crontab -e; then hit "i" on keyboard to enter insert mode; remove the #; hit "ESC" on keyboard; write :wq and hit "ENTER" to save the changes.

1.2.2. Total Files Processed per Hour (last 24 hours)

This view is very similar to view Files Processed per Host, but instead of having the count of files processed per host, it has the sum of files processed by all hosts in each hour.



How this dashboard helps identifying issues?

NOTE: Dates and times are in PST time zone.

If suddenly the number of files processed drops to a very small number, it is possible that there is something wrong. Nevertheless, it may happen for some hours, specially during the night, where the counts will naturally be smaller as there are fewer module runs and thus fewer files to process.

Now, if it is observed that for last couple of hours no new file has been processed and there is no other reasonable explanation, then a problem may be happening.

Possible problems:

- iii. instance_detector OR instance_files_detector are stuck, meaning that these scripts are not doing any progress on processing

Investigation Suggested:

- Remote connection to host (using MobaXterm, Remote Desktop Manager or another solution)
- Run the following command to check how long the script is running:
ps aux | grep instance_detector or
ps aux | grep instance_files_detector
- In the results, check for how long the script is running


```
(svc_prd_edennetssupp@spolleden0089a sonadm)$ ps aux | grep instance_files_parser
```

svc_prd+	pid	cpu	mem	rss	ss	time	command
svc_prd+	4117800	0.3	0.5	1264264	180796	?	python3 instance_files_parser.py -v
svc_prd+	4129352	0.0	0.3	1264240	126956	?	python3 instance_files_parser.py -v
svc_prd+	4129360	2.8	0.5	1375036	188448	?	python3 instance_files_parser.py -v
svc_prd+	4129363	89.4	0.7	1844296	257352	?	python3 instance_files_parser.py -v
svc_prd+	4129365	1.5	0.5	1364708	186912	?	python3 instance_files_parser.py -v
svc_prd+	4129368	0.8	0.5	1271220	164156	?	python3 instance_files_parser.py -v
svc_prd+	4129371	0.6	0.4	1268000	159636	?	python3 instance_files_parser.py -v
svc_prd+	4129374	1.4	0.5	1362356	184396	?	python3 instance_files_parser.py -v
svc_prd+	4129377	1.5	0.5	1361996	184092	?	python3 instance_files_parser.py -v
svc_prd+	4129380	0.5	0.4	1259468	158132	?	python3 instance_files_parser.py -v
svc_prd+	4129383	1.7	0.5	1366596	183840	?	python3 instance_files_parser.py -v
svc_prd+	4129386	2.0	0.5	1368484	188848	?	python3 instance_files_parser.py -v
svc_prd+	4129389	0.8	0.4	1268036	159680	?	python3 instance_files_parser.py -v
svc_prd+	4129392	89.0	0.8	1856764	289572	?	python3 instance_files_parser.py -v
svc_prd+	4129395	3.2	0.5	1379440	189568	?	python3 instance_files_parser.py -v
svc_prd+	4129398	1.1	0.5	1356052	180376	?	python3 instance_files_parser.py -v

This is the column to check. Value is in minutes.

- If the script is running for hours and hours – e.g. 800 minutes, 350 minutes, etc – there is a considerable possibility for the script to be stuck. Usually, the script runs in iterations of less than 30 minutes.

Fix Suggested:

- Restart the process to reactivate file processing, by running the following:
`kill -u svc_prd_edennetssupp -f instance_detector.py`
`kill -u svc_prd_edennetssupp -f instance_files_detector.py`
- Process will start again in the next minute.
- Check logs for more details on where the script has been stuck.

NOTE: In general, the script should never get stuck, it should be a very rare case. But, if it does occur repeatedly, a deeper investigation on root causes is required.

1.2.3. Last Update per Table

This view helps identifying when was each table last updated.

Last Update per Table		24 rows	...
TABLE_ALIAS	LAST_INSERT		
SON_MOD_USAGE_1	2024-11-26 11:34:53.232		
USAGE_REPORT	2024-12-17 00:00:00.000		
INSTANCE_PUSHES	2024-12-18 00:00:00.000		
CLUSTERS_CELLS	2024-12-18 00:00:00.000		
CLUSTERS_DEFINITION	2024-12-18 00:00:00.000		
EXCLUSION_LIST_DATA	2024-12-18 00:00:00.000		
TOPOLOGY_AUDIT_ISSUES	2024-12-18 00:00:00.000		
INSTANCE_CHANGES	2024-12-18 00:00:00.000		
Show 16 more			

How this dashboard helps identifying issues?

All tables listed should be updated at least once a day. So, if for any table listed, the LAST_INSERT time is longer than 1 day, or even 2 days to be safe, there is probably something wrong.

For example, in the picture attached, the SON_MOD_USAGE_1 table has LAST_INSERT as 2024-11-26 which is already some weeks ago from today (2024-12-18). This indicates that this table is not being daily updated.

Investigation and Fix Suggested:

- Each table has its own population process.
- If table is populated by a backend script: check in which host the script is scheduled, check in the logs if the script is running or if the script is failing
- If table is populated by a Snowflake procedure/task: check if the task is not suspended, if the task is suspended, verify if the task was suspended due to errors in the procedure. Try running the procedure to see if does the update in the table or if it fails. If the procedure fails, fix the procedure and then go to the task and *resume* the runs.

1.2.4. Instance Data Processed per Hour (last 24 hours)

SON Instance Data Processing depends on the following scripts:

instance_info_cells – to process instance cells

instance_info_changes – to process instance changes

instance_info_provisioning_data – to process instance pushes and error logs

These scripts are scheduled to run in only 1 host.

DATETIME	CELLS	CHANGES	CONFIG	PUSHES	ERROR_▲
2024-12-18 12:00	null	54	null	56	
2024-12-18 11:00	875	74	102	85	
2024-12-18 10:00	947	77	122	78	
2024-12-18 09:00	1065	108	202	112	
2024-12-18 08:00	1015	111	151	113	
2024-12-18 07:00	1022	119	160	124	
2024-12-18 06:00	1063	106	273	114	
2024-12-18 05:00	1035	122	226	140	

[Show 18 more](#)

The table shows the count of instances processed for each type of data – cells, changes, GUI configuration, pushes and provisioning error logs.

NOTE:

The counts will not necessarily be the same across different types of data – meaning the count for cell instance data and changes instance data will not necessarily be close of each other.

Why?

Reason 1: For cells, every time a new activation is seen, it reprocesses the cells, causing in a higher number. On other hand, for changes, pushes and error logs, only CLOSED LOOP instances that have pushes will be processed, resulting in smaller number. And for GUI configurations, it only updates for each instance 1 time ever, meaning even if there are new activations, it will not process – as configuration does not change from activation to activation.

Reason 2: Each data type is processed by a different script, in a way that the number and the set of instances processed by one script in a timeframe will not necessarily be the same number and set of instances processed by another script.

How this dashboard helps identifying issues?

If for any data type, the number of processed instances is NULL hour after hour, then there is probably an issue with the script data does the processing.

Investigation Suggested:

- Remote connection to host (using MobaXterm, Remote Desktop Manager or another solution)
- Run the following command to check how long the script is running:
ps aux | grep SCRIPT_NAME
- In the results, check for how long the script is running

```
(svc_prd_edennetssupp@spolleden0089a sonadm)$ ps aux | grep instance_files_parser
svc_prd+ 4117800 0.3 0.5 1264264 180796 ? 0:04 python3 instance_files_parser.py -v
svc_prd+ 4129352 0.0 0.3 1264240 126956 ? 0:00 python3 instance_files_parser.py -v
svc_prd+ 4129360 2.8 0.5 1375036 188448 ? 0:12 python3 instance_files_parser.py -v
svc_prd+ 4129363 89.4 0.7 1844296 257352 ? 6:11 python3 instance_files_parser.py -v
svc_prd+ 4129365 1.5 0.5 1364708 186912 ? S1 10:44 0:06 python3 instance_files_parser.py -v
svc_prd+ 4129368 0.8 0.5 1271220 164156 ? S1 10:44 0:03 python3 instance_files_parser.py -v
svc_prd+ 4129371 0.6 0.4 1268000 159636 ? S1 10:44 0:02 python3 instance_files_parser.py -v
svc_prd+ 4129374 1.4 0.5 1362356 184396 ? S1 10:44 0:06 python3 instance_files_parser.py -v
svc_prd+ 4129377 1.5 0.5 1361996 184092 ? S1 10:44 0:06 python3 instance_files_parser.py -v
svc_prd+ 4129380 0.5 0.4 1259468 158132 ? S1 10:44 0:02 python3 instance_files_parser.py -v
svc_prd+ 4129383 1.7 0.5 1366596 183840 ? S1 10:44 0:07 python3 instance_files_parser.py -v
svc_prd+ 4129386 2.0 0.5 1368484 188848 ? S1 10:44 0:08 python3 instance_files_parser.py -v
svc_prd+ 4129389 0.8 0.4 1268036 159680 ? S1 10:44 0:03 python3 instance_files_parser.py -v
svc_prd+ 4129392 89.0 0.8 1856764 289572 ? R1 10:44 6:09 python3 instance_files_parser.py -v
svc_prd+ 4129395 3.2 0.5 1379440 189568 ? S1 10:44 0:13 python3 instance_files_parser.py -v
svc_prd+ 4129398 1.1 0.5 1356052 180376 ? S1 10:44 0:04 python3 instance_files_parser.py -v
```

This is the column to check. Value is in minutes.

- If the script is running for hours and hours – e.g. 800 minutes, 350 minutes, etc – there is a considerable possibility for the script to be stuck. Usually, the script runs in iterations of less than 30 minutes.

Fix Suggested:

- Restart the process to reactivate file processing, by running the following:
`kill -u svc_prd_edennetssupp -f SCRIPT_NAME.py`
- Process will start again in the next minute.
- Check logs for more details on where the script has been stuck.

NOTE: In general, the script should never get stuck, it should be a very rare case. But, if it does occur repeatedly, a deeper investigation on root causes is required.

1.2.5. Instance Data Processed per SON Server (last 24 hours)

This view is very similar to view Instance Data Processed per Hour (last 24 hours), but instead of having the count per hour, it has the sum of processed instances per SON Server.

Instance Data Processed per SON Server (last 24 hours)					29 rows ...
SERVER_NAME	CELLS	CHANGES	CONFIG	PUSHES	ERROR_▲
POLEDENP02	111	38	93	49	
POLEDENP04	266	66	206	79	
POLEDENP05	211	50	135	50	
POLEDENP06	302	109	292	105	
POLEDENP07	126	19	93	20	
POLEDENP09	194	49	171	49	
POLEDENP11	152	27	128	27	
POLEDENP13	242	42	100	42	
					▼
					Show 21 more

How this dashboard helps identifying issues?

For every server (see NOTE) it is expected to have cells, changes, GUI configurations, pushes and provisioning error logs processed in the last 24 hours. If the value is NULL, this is an indication that something may be wrong.

NOTE: For servers TTNEDENP16 (Sprint cells), TTNEDENP06 and TTNEDENP13 (GSM / UMTS cells), the NULLS are expected as these servers process very few instances and most of those does not even run in closed loop – these servers are not listed in the report.

It has happened some cases where due to a new MantaRay SON release that the endpoints used by SON Analytics backend were updated from one version to the new one. Then, when the new version of MantaRay was deployed to the FOA servers, the count of instances, activations and files processed in those servers dropped to zero.

It would not be possible to see this in the Instance Data Processed per Hour table, but in this report, if for any server there is a processing problem, the server will be listed to the top and NULLS should be seen in any of the columns.

Investigation Suggested:

- Remote connection to host (using MobaXterm, Remote Desktop Manager or another solution)
- Check the logs to identify why the script is failing to process for the specific server

NOTE: Remind that the problem may at the beginning of the flow – instance/activation/file is not being detected by instance_detector or instance_files_detector - or at the end of the flow – processing script is failing for instances of such server.

1.2.6. Processing Queue per Table

This view shows how many Instances, Activations or Files are waiting in the queue to be processed.

NOTE: It also includes the instances, activations and files that are under processing.

Processing Queue per Table			6 rows	***
TABLE_ALIAS	LEVEL	PENDING		
PARSED FILES	FILES	7212		
FILES	INSTANCE	498		
CELLS	INSTANCE	401		
PUSHES	ACTIVATIONS	1		
CHANGES	ACTIVATIONS	1		
ERROR_LOGS	ACTIVATIONS	1		

How this dashboard helps identifying issues?

It helps monitoring the queues and proactively acting if the queues starting to grow too much.

Expected queue size per table and level:

PARSED FILES	FILES	< 10.000	PENDING	scripts	instance_files_parser instance_files_downloader
FILES	INSTANCE	< 1.000	PENDING	scripts	instance_files_parser instance_files_downloader
CELLS	INSTANCE	< 1.000	PENDING	script	instance_info_cells
CHANGES	INSTANCE	< 1.000	PENDING	script	instance_info_changes
PUSHES	INSTANCE	< 1.000	PENDING	script	instance_info_provisioning_data
ERROR_LOGS	INSTANCE	< 1.000	PENDING	script	instance_info_provisioning_data

This is not a hard threshold, meaning the even if the numbers seen are slightly above the limits here and matching to a time of the day when there is a higher demand from SON, it is unlikely that there is a problem, and the Queue will be zeroed out soon.

But, if the Queue is higher than the counts above, a deeper check might be needed.

Investigation Suggested:

- Remote connection to host (using MobaXterm, Remote Desktop Manager or another solution)
- Check the logs to identify why the script is failing to process

NOTE: If the Queue keeps growing, the problem is not in the detection scripts – instance_detector or instance_files_detector.

1.2.7. Time to run per Script per Host (in minutes)

This view allows to monitor how long each script took to complete in each host in the last run.

When the value for one script-host combination is null, means it is not running in that host.

Time to run per Script per Host (in minutes)				7 rows	...
SCRIPT	SONA1-0089a	SONA2-0089b	SONA3-0089c		
instance_detector	null	19.00	null		
instance_files_detector	null	0.00	null		
instance_files_downloader	2.00	0.00	8.00		
instance_files_parser	3.00	22.00	1.20		
instance_info_cells	3.00	null	null		
instance_info_changes	1.20	null	null		
instance_info_provisioning_data	0.00	null	null		

The values are in minutes, and for most scripts' iterations should have a duration of less than one hour.

How this dashboard helps identifying issues?

It helps identifying if any script is stuck – when the duration of run in minutes is too high (such as over 300 minutes, 500 minutes, etc).

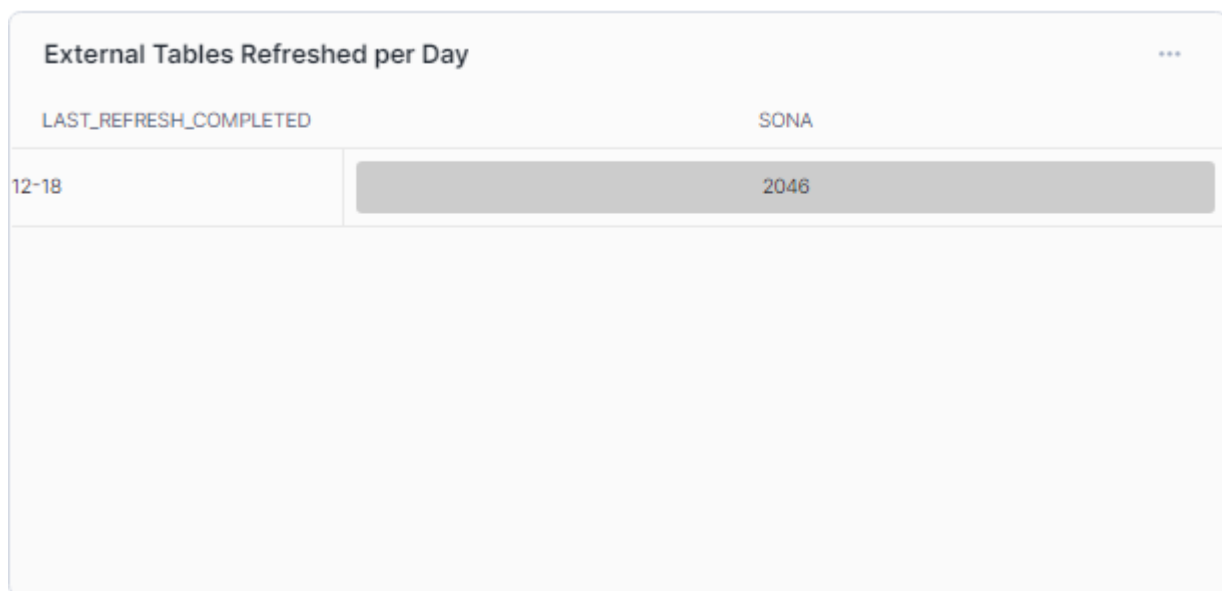
Investigation and Fix Suggested:

Please check 1.2.1. Files Processed per Host.

1.2.8. External Tables Refreshed per Day

This view allows to know if all external tables are being regularly updated.

The updates in external tables are very important as it is the process that links the stored files in Azure Storage (ADLS) with respective tables in Snowflake.



How this dashboard helps identifying issues?

If any table is not being regularly refreshed, the view will show the number of tables that are not refreshed and the day it was last refreshed.

When the REFRESH does not work for a table, the table will not have latest data available.

NOTE: Refreshing the external table DOES NOT ensure that the table will have the latest data available. Why? To have latest data available in an external table, 2 parts of the process should occur: 1) output report file needs to be loaded to Azure Storage (ADLS) – this is managed by instance_files_parser.py script and 2) the external table in Snowflake needs to be refreshed to link the files with the table – this is managed by SP_REFRESH_EXTERNAL_TABLES procedure in Snowflake.

Investigation and Fix Suggested:

- Check in Snowflake if the tasks TASK_SP_REFRESH_EXTERNAL_TABLES_DAILY, TASK_SP_REFRESH_EXTERNAL_TABLES_WEEKLY and TASK_SP_REFRESH_EXTERNAL_TABLES_MONTHLY in **suspended** status.
- If that is the case, it is possible that the procedure SP_REFRESH_EXTERNAL_TABLES is failing.

- Try running the procedure SP_REFRESH_EXTERNAL_TABLES.
- If the procedure fails, identify and fix the issue.
- Remind to resume the tasks TASK_SP_REFRESH_EXTERNAL_TABLES_DAILY, TASK_SP_REFRESH_EXTERNAL_TABLES_WEEKLY and TASK_SP_REFRESH_EXTERNAL_TABLES_MONTHLY.

1.2.9. Managed Tables Refreshed per Day

This view allows to know if all managed tables are being regularly updated.

Managed Tables Refreshed per Day		76 rows	...
TABLE_NAME	LAST_UPDATED		
TB_AF_FACT_TMO_5GN_NETWORK_SLICING_CLEANUP_RESULTS	null		
TB_AF_FACT_PARAMCONSISTCHK_EXCLUSIONLISTWARNINGSE	2024-06-27		
TB_AF_FACT_PARAMCONSISTCHK_COMBINED_PARAMETER_DETAILS	2024-06-27		
TB_AF_FACT_PARAMCONSISTCHK_COMBINED_CONFIGURATION	2024-06-27		
TB_AF_FACT_NOTGA_TMO_45GN_VALIDATIONERRORREPORT_LOG_DATA	2024-11-15		
TB_AF_FACT_PCI_OPTIMIZATION_5G_PCI_CONFUSION_ISSUES	2024-12-18		
TB_AF_FACT_TMO_5GE_NETWORK_SLICING_UPDATE_RESULTS	2024-12-18		
TB_AF_FACT_PARAMCONSISTCHK_SUMMARY	2024-12-18		
		Show 68 more	

How this dashboard helps identifying issues?

If any table is not being populated it will be possible to identify through the view by checking LAST_UPDATED date. As it is expected that most of these tables get data daily, if for any table there is a date difference of more than 2 days for example, it could be alarming.

There are 2 requirements for an output report file be populated to a managed table.

- 1) The sheet from the output report must be listed in the TB_AF_CONTROL_TB_MANAGED_TABLES_LIST with CREATED as TRUE;
- 2) The sheet needs to be correctly processed by instance_files_parser.py script;

This view reports only the managed tables that already satisfy 1), in a way that if a managed table is not being reported here, it is probably not converted from external to managed table yet.

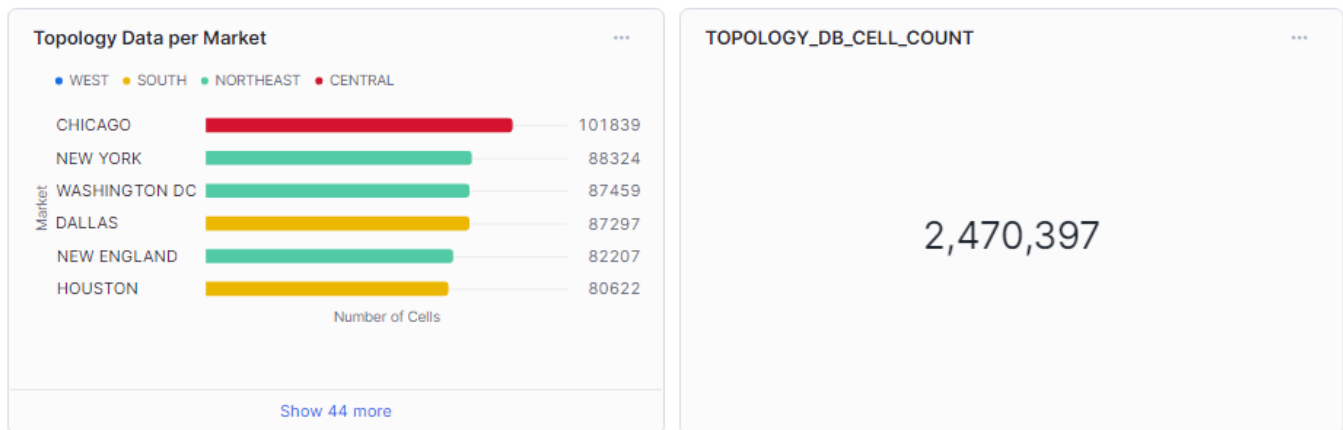
Investigation and Fix Suggested:

- Check if there are instance runs for this module in the period – a quick query on TB_AF_CONTROL_TB_SON_INSTANCES filtering by module name can help.

- If there are instance runs, check if the instances produced output report files – user can go to the server and download the output report
If there are no instance runs, no issue is happening.
- If there are output reports available, check if the output reports were detected by instance_files_detector – user can query TB_AF_CONTROL_TB_SON_INSTANCE_FILES;
If there are no output report available, no issue is happening on SON Analytics, but module may be failing to produce output reports and this may need investigation from Nokia (if vendor module) or SON Dev (if TMO custom module).
- If the output reports were detected by instance_files_detector, check if there were failures during parsing process – user can query TB_AF_CONTROL_TB_PARSED_FILES;
- If there are failures during the parsing, check the logs – user can go to the host where instance_files_parser is running and download the logs to investigate.

1.2.10. Topology Data per Market and TOPOLOGY_DB_CELL_COUNT

This view allows to check current count of cells in the network. This is an estimation, since not all cells from topology data are populated into SON. Also, allows to have an estimation of how many cells there are per market.



How this dashboard helps identifying issues?

Usually, the count of cells in topology data should be > 2.4 million cells.

If a number way below the expectation is seen, then probably there is an issue.

Investigation and Fix:

- Check topology_data_export automation logs – user can login to the host where the automation is running (NSSA1) and download the logs;
- It is possible that data is being incorrectly reported by NED API.

1.2.11. Cell OSS Info per Server and OSS_DB_CELL_COUNT

This view is similar to the one from topology, but it actually returns the number of cells in SON.

Instead of having the information per market, here the information available is per sever.

<div>OSS_DB_CELL_COUNT</div> <div>2,415,949</div>	<div>Cell OSS Info per Server32 rows</div> <table><tr><th>SERVER_NAME</th><th>OSS_CELL_INFO_COUNT</th></tr><tr><td>POLEDENP02</td><td>49307</td></tr><tr><td>POLEDENP04</td><td>76429</td></tr><tr><td>POLEDENP05</td><td>60971</td></tr><tr><td>POLEDENP06</td><td>87378</td></tr><tr><td>POLEDENP07</td><td>80095</td></tr><tr><td>POLEDENP09</td><td>85101</td></tr><tr><td>POLEDENP11</td><td>112935</td></tr><tr><td>POLEDENP12</td><td>133135</td></tr></table> <div>Show 24 more</div>	SERVER_NAME	OSS_CELL_INFO_COUNT	POLEDENP02	49307	POLEDENP04	76429	POLEDENP05	60971	POLEDENP06	87378	POLEDENP07	80095	POLEDENP09	85101	POLEDENP11	112935	POLEDENP12	133135
SERVER_NAME	OSS_CELL_INFO_COUNT																		
POLEDENP02	49307																		
POLEDENP04	76429																		
POLEDENP05	60971																		
POLEDENP06	87378																		
POLEDENP07	80095																		
POLEDENP09	85101																		
POLEDENP11	112935																		
POLEDENP12	133135																		

How this dashboard helps identifying issues?

Usually, the count of cells in OSS table should be > 2.4 million cells.

If a number way below the expectation is seen, then probably there is an issue.

Investigation and Fix:

- Check topology_data_export automation logs – user can login to the host where the automation is running (NSSA1) and download the logs;
- It is possible that data is being incorrectly reported by NED API.

2. SON Analytics Control Center – Audit Checklist

- ☐ [Files Processed per Host] Is for any host the number of processed files very less than the other hosts?
- ☐ [Files Processed per Host] Is for any host the number of processed files empty?
- ☐ [Total Files Processed per Hour] Is for any processing hour the number of processed files too small compared to the demand expected for this period?
- ☐ [Total Files Processed per Hour] Is for any processing hour the number of processed files empty when there are many files waiting in the queue to be processed?
- ☐ [Last Update per Table] Is any of the tables update date older than 2 days?
- ☐ [Instance Data Processed per Hour] Is for any processing hour the number of processed instances empty when there are many instances waiting in the queue to be processed?
- ☐ [Instance Data Processed per SON Server] Is for any SON Server the number of processed instances empty for last day?
- ☐ [Processing Queue per Table] Is the number of files/instances/activations in the queue too high (e.g. > 10.000)?
- ☐ [Time to run per script per Host] Is the time to run for any script in any host too high (e.g. > 500 minutes)?
- ☐ [External Tables Refreshed per Day] Is there any external table that was not updated in the last 24 hours?
- ☐ [Managed Tables Refreshed per day] Is there any managed table which last update time is older than 2 days? – NOTE: some managed tables are updated monthly or weekly, so in certain cases it may be reasonable to have last update time older than 2 days, but for modules such as SleepingCell, COC, ParameterChanges, Ducting, it should have data for last 2 days.
- ☐ [Topology Data per Market] Is there any market with a number of cells that does not match the expectation (too few cells for the market size)?
- ☐ [Topology DB Cell Count] Is the number of cells in topology data smaller than expected (e.g. < 2.4 million)?
- ☐ [OSS DB Cell Count] Is the number of cells in OSS data smaller than expected (e.g. < 2.4 million)?
- ☐ [Cell OSS Info per Server] Is there any server with a number of cells that does not match the expectation (too few cells for the server size)?