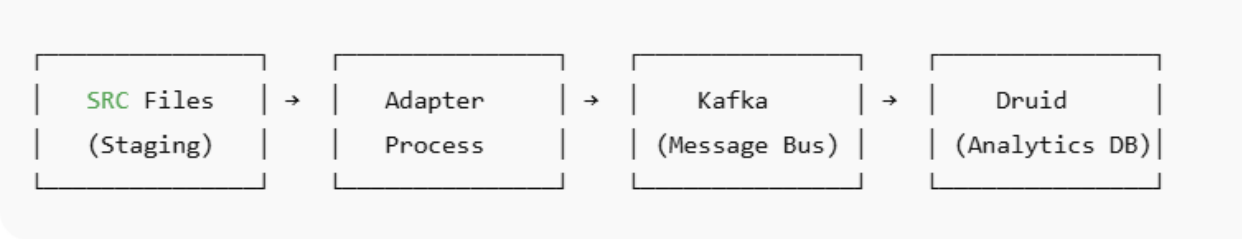
ERICSSON 4G PM SOP

Process Flow:



Adapter Process:

Description: This process reads the landing folder recursively and process the files and push the data into kafka directly. Once the data is pushed into kafka , files are archived into archived folder. UNPARSABLE files will be rejected into REJECT folder.

Server: 10.180.83.252

Script paths:

BASEBAND: /home/Admin/run_pm_processing_kafka.sh

ERBS: /home/Admin/ run_eric_4g_erbs_pm_processing_kafka.sh

How to run the script:

Command: sh <script_name>

For example as shown in below:

1. Execuing the command will give us logs path and process id

```
[Admin@NIFI60 ~]$ sh run_eric_4g_erbs_pm_processing_kafka.sh
Processor started in background with PID 3146653
Logging output to /OSSNode_data/rust_adapter/logs/xml_processor_eric_4g_pm_erbs_local20250821_140222.log
[Admin@NIFI60 ~]$
```

How to stop the process:

1. Run the script again to get the pid as shown in below

```
[Admin@NIFI60 ~]$ sh run_eric_4g_erbs_pm_processing_kafka.sh
Process is already running with 3810060. Ignoring start. Use stop script to kill process.
[Admin@NIFI60 ~]$
```

2. Kill the process id using command: kill <pid>

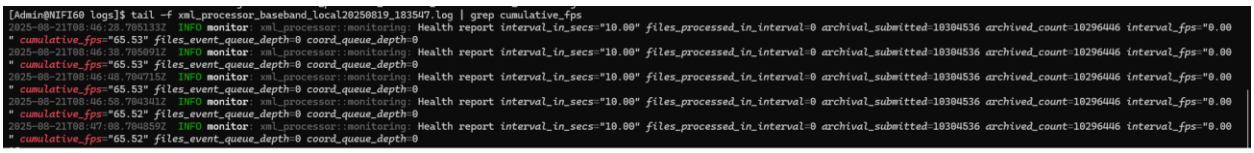
```
[Admin@NIFI60 ~]$ kill 3810060
```

Note: The process is scheduled in crontab in same server, in case of auto restart when network failures happens

Process Throughput:

In logs file cumulative_fps value gives us current throughput of the process as shown in below

Command : tail –f <log_file> | grep cumulative_fps



Kafka to druid process:

1. suspend and resume job: This process suspends and resumes the supervisors in round robin fashion of the supervisors provided.

Server: 10.115.1.85

Script directory: /home/Admin/code/production_code/ingestion/ericson/4g

Script name: pm_druid_suspend_and_resume_job.sh

How to run the script:

1. Go to script dir: cd /home/Admin/code/production_code/ingestion/ericson/4g
2. Run the script: sh pm_druid_suspend_and_resume_job.sh

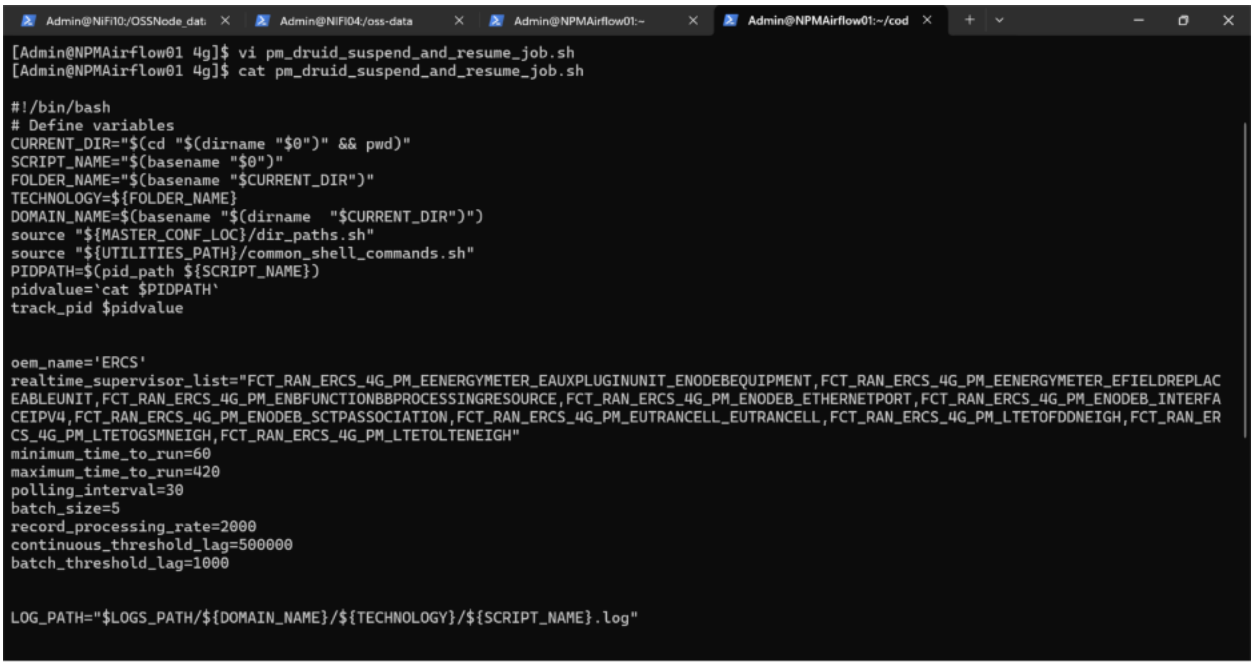
Logs path: /home/Admin/code/production_code/logs/ericson/4g/pm_druid_suspend_and_resume_job.sh.log

Scheduling: Scheduled in crontab for every 15mins

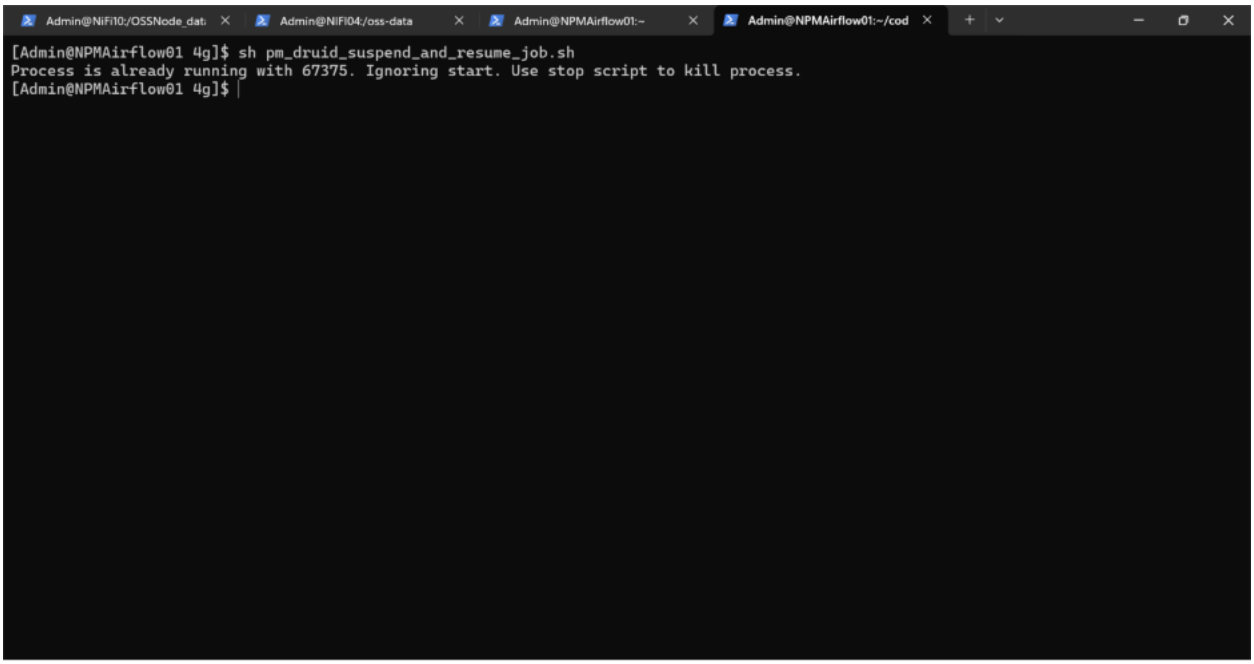
How to know which tables are involved:

1. It is based on 3 types:
2. supervisor_list is provided: Only those topics are involved in the processing
3. if supervisor_list is not provided, but oem and class_type are provided. It will extract the supervisor list and filters out the list which matches this pattern: "%<oem_name>%<TECH>_PM%"
4. If realtime_supervisor_list config. The supervisors listed here will be in resume mode only i.e continuous loading

In this process we have used realtime supervisor list



How to know whether the process is running or not: Run the suspend and resume job script, if it is running then it says already running with some pid.

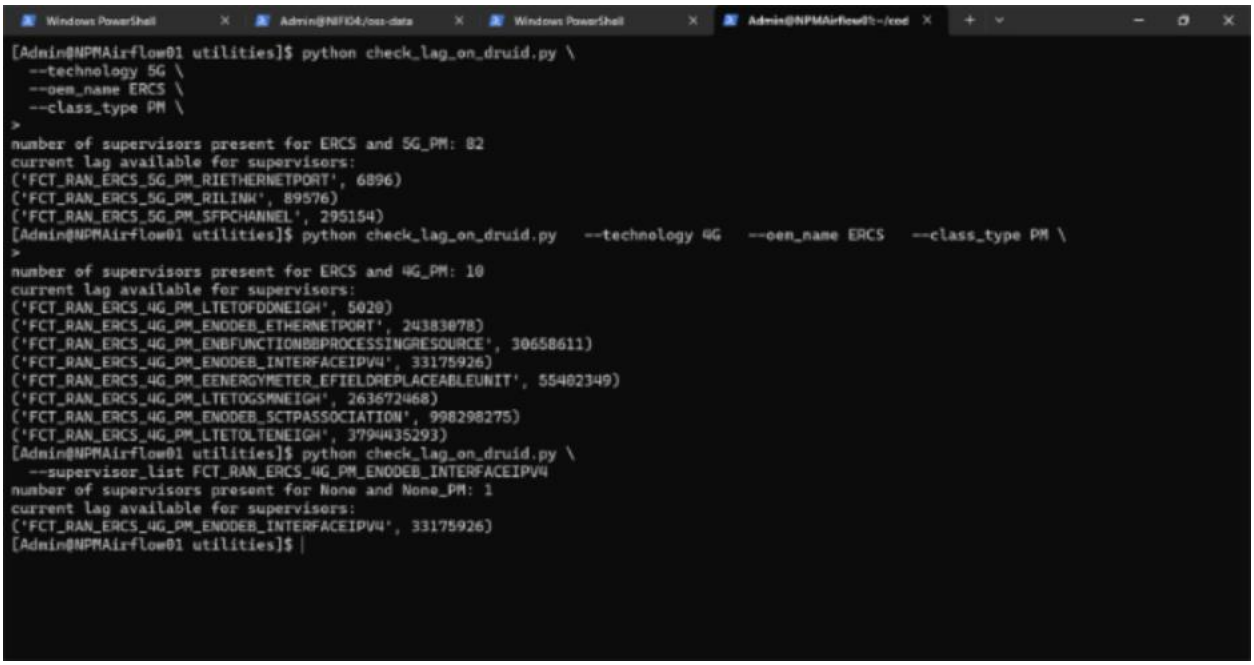


How to check lag between kafka and druid:

We have a script to check lag on server(10.115.1.85)

Path: /home/Admin/code//production_code/utilities/check_lag_on_druid.py

- Login server 10.115.1.85
- Go to directory (command: cd /home/Admin/code//production_code/utilities)
- Run the script(python check_lag_on_druid.py)
- Commad: python check_lag_on_druid.py --oem_name ERCS --technology 4G --class_type PM

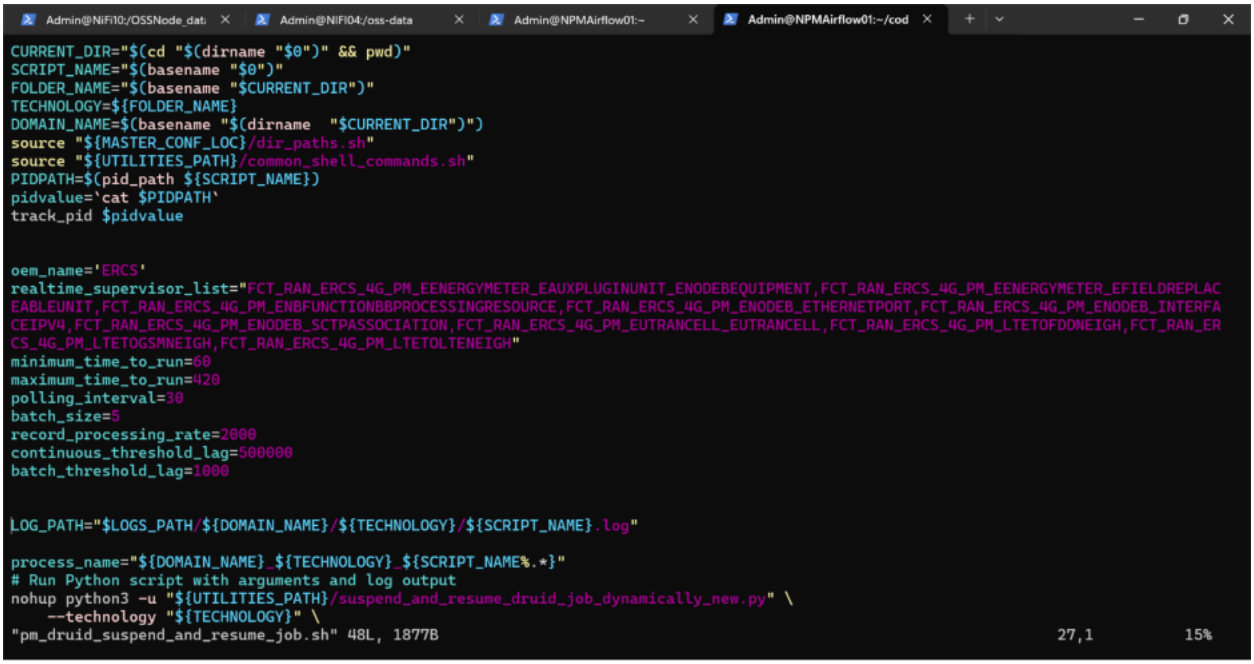


```
[Admin@NPMAirflow01 utilities]$ python check_lag_on_druid.py \
  --technology 5G \
  --oem_name ERCS \
  --class_type PM \
>
number of supervisors present for ERCS and 5G_PM: 82
current lag available for supervisors:
('FCT_RAN_ERCS_5G_PM_RIETHERNETPORT', 6896)
('FCT_RAN_ERCS_5G_PM_RILINK', 89576)
('FCT_RAN_ERCS_5G_PM_SFPCHANNEL', 295154)
[Admin@NPMAirflow01 utilities]$ python check_lag_on_druid.py   --technology 4G   --oem_name ERCS   --class_type PM \
>
number of supervisors present for ERCS and 4G_PM: 10
current lag available for supervisors:
('FCT_RAN_ERCS_4G_PM_LTETOFDDNEIGH', 5020)
('FCT_RAN_ERCS_4G_PM_ENODEB_ETHERNETPORT', 24383078)
('FCT_RAN_ERCS_4G_PM_ENBFUNCTIONBBPROCESSINGRESOURCE', 30658611)
('FCT_RAN_ERCS_4G_PM_ENODEB_INTERFACEIPV4', 33175926)
('FCT_RAN_ERCS_4G_PM_EENERGYMETER_EFIELDREPLACEABLEUNIT', 55402349)
('FCT_RAN_ERCS_4G_PM_LTETOGSMNEIGH', 263672468)
('FCT_RAN_ERCS_4G_PM_ENODEB_SCTPASSOCIATION', 998298275)
('FCT_RAN_ERCS_4G_PM_LTETOLTENEIGH', 3794435293)
[Admin@NPMAirflow01 utilities]$ python check_lag_on_druid.py \
  --supervisor_list FCT_RAN_ERCS_4G_PM_ENODEB_INTERFACEIPV4
number of supervisors present for None and None_PM: 1
current lag available for supervisors:
('FCT_RAN_ERCS_4G_PM_ENODEB_INTERFACEIPV4', 33175926)
[Admin@NPMAirflow01 utilities]$
```

Stop the suspend and resume job:

if you run the suspend and resume script process again then it give as process is running already. And give process id. Or stop using process_id in PIDPATH.

So you can kill process by process id:



```
CURRENT_DIR="$(cd "$(dirname "$0")" && pwd)"
SCRIPT_NAME="$(basename "$0")"
FOLDER_NAME="$(basename "$CURRENT_DIR")"
TECHNOLOGY=${FOLDER_NAME}
DOMAIN_NAME=$(basename "$(dirname  "$CURRENT_DIR")")
source "${MASTER_CONF_LOC}/dir_paths.sh"
source "${UTILITIES_PATH}/common_shell_commands.sh"
PIDPATH=$(pid_path ${SCRIPT_NAME})
pidvalue=`cat $PIDPATH`
track_pid $pidvalue


oem_name='ERCS'
realtime_supervisor_list="FCT_RAN_ERCS_4G_PM_EENERGYMETER_EAUXPLUGINUNIT_ENODEBEQUIPMENT,FCT_RAN_ERCS_4G_PM_EENERGYMETER_EFIELDREPLAC
EABLEUNIT,FCT_RAN_ERCS_4G_PM_ENBFUNCTIONBBPROCESSINGRESOURCE,FCT_RAN_ERCS_4G_PM_ENODEB_ETHERNETPORT,FCT_RAN_ERCS_4G_PM_ENODEB_INTERFA
CEIPV4,FCT_RAN_ERCS_4G_PM_ENODEB_SCTPASSOCIATION,FCT_RAN_ERCS_4G_PM_EUTRANCELL_EUTRANCELL,FCT_RAN_ERCS_4G_PM_LTETOFDDNEIGH,FCT_RAN_ER
CS_4G_PM_LTETOGSMNEIGH,FCT_RAN_ERCS_4G_PM_LTETOLTENEIGH"
minimum_time_to_run=60
maximum_time_to_run=420
polling_interval=30
batch_size=5
record_processing_rate=2000
continuous_threshold_lag=500000
batch_threshold_lag=1000


LOG_PATH="$LOGS_PATH/${DOMAIN_NAME}/${TECHNOLOGY}/${SCRIPT_NAME}.log"

process_name="${DOMAIN_NAME}_${TECHNOLOGY}_${SCRIPT_NAME%.*}"
# Run Python script with arguments and log output
nohup python3 -u "${UTILITIES_PATH}/suspend_and_resume_druid_job_dynamically_new.py" \
    --technology "${TECHNOLOGY}" \
"pm_druid_suspend_and_resume_job.sh" 48L, 1877B                                                    27,1          15%
```

Command: Kill <process_id>

Suspend the Supervisor:

1. Disable the process in crontab and follow the below steps

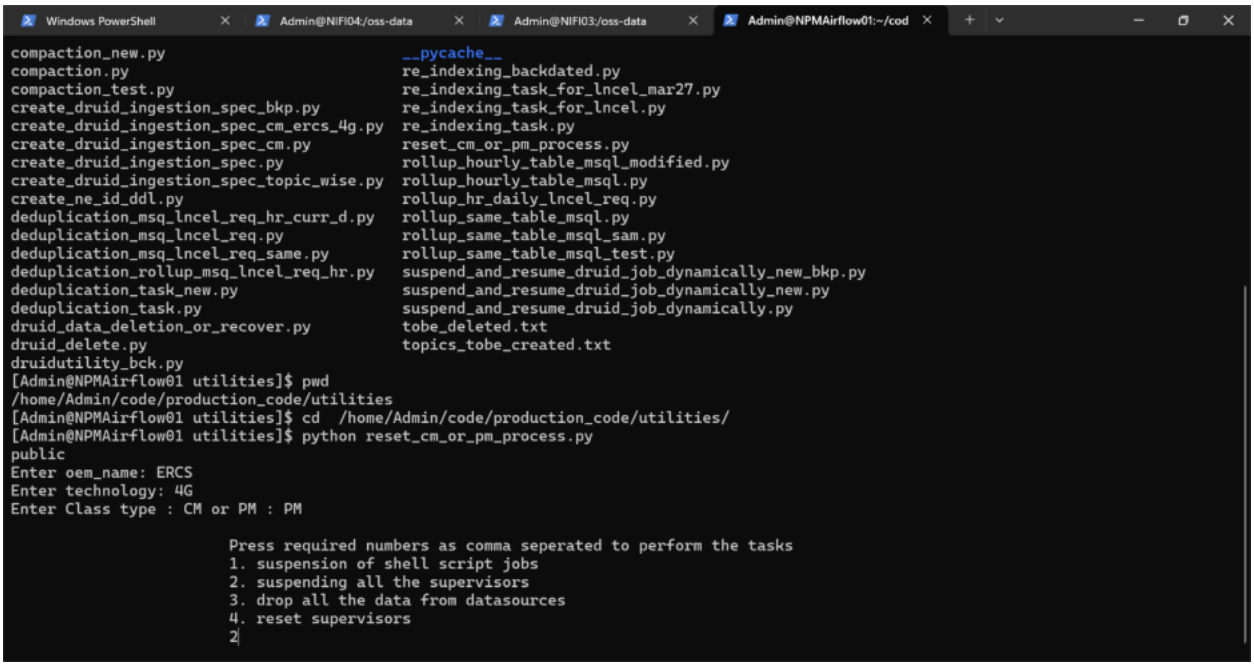
Path: /home/Admin/code/production_code/utilities/**reset_cm_or_pm_process.py**

We have a script **reset_cm_or_pm_process.py.** this help us to suspend the supervisors.

As mention below inputs it can suspend the supervisors respective oem, technology and class_type.

How to run this script:

- Login in 10.115.1.85 server
- Go to directory (command: cd /home/Admin/code/production_code/utilities/)
- **Run the script:** (python **reset_cm_or_pm_process.py)**
- Give inputs as options 1,2 for suspending the process and suspending the supervisors



NOTE: before run the script please check in crontab, the job should be commented so it will not started automatically.

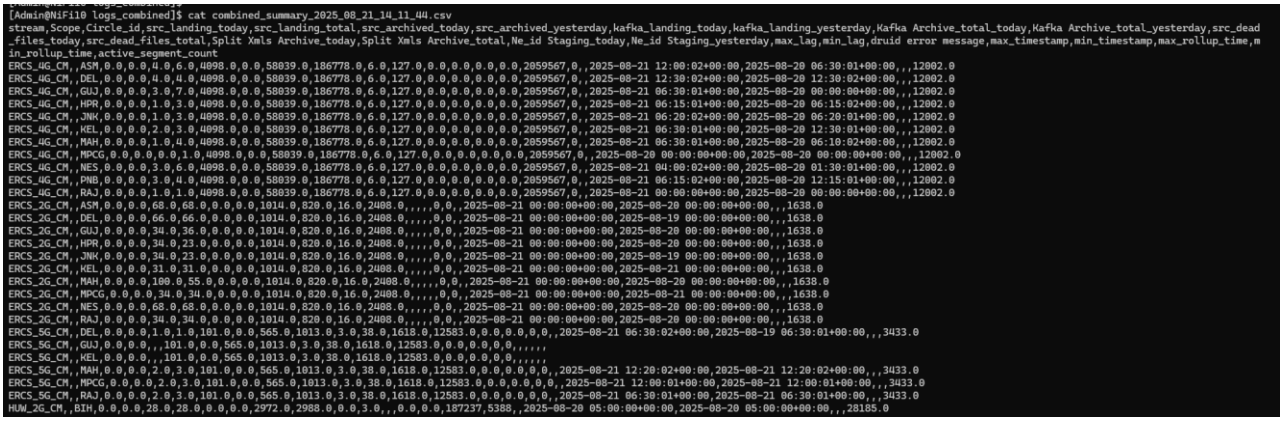
Monitoring Process: Every one hour monitoring stats are generated in the below folder as csv format. We need to take the latest file generated

Server: 10.180.83.161

Monitoring stats folder:

/OSSNode_data/utilities/monitoring/logs_combined

Example: cat combined_summary_2025_08_21_14_11_44.csv

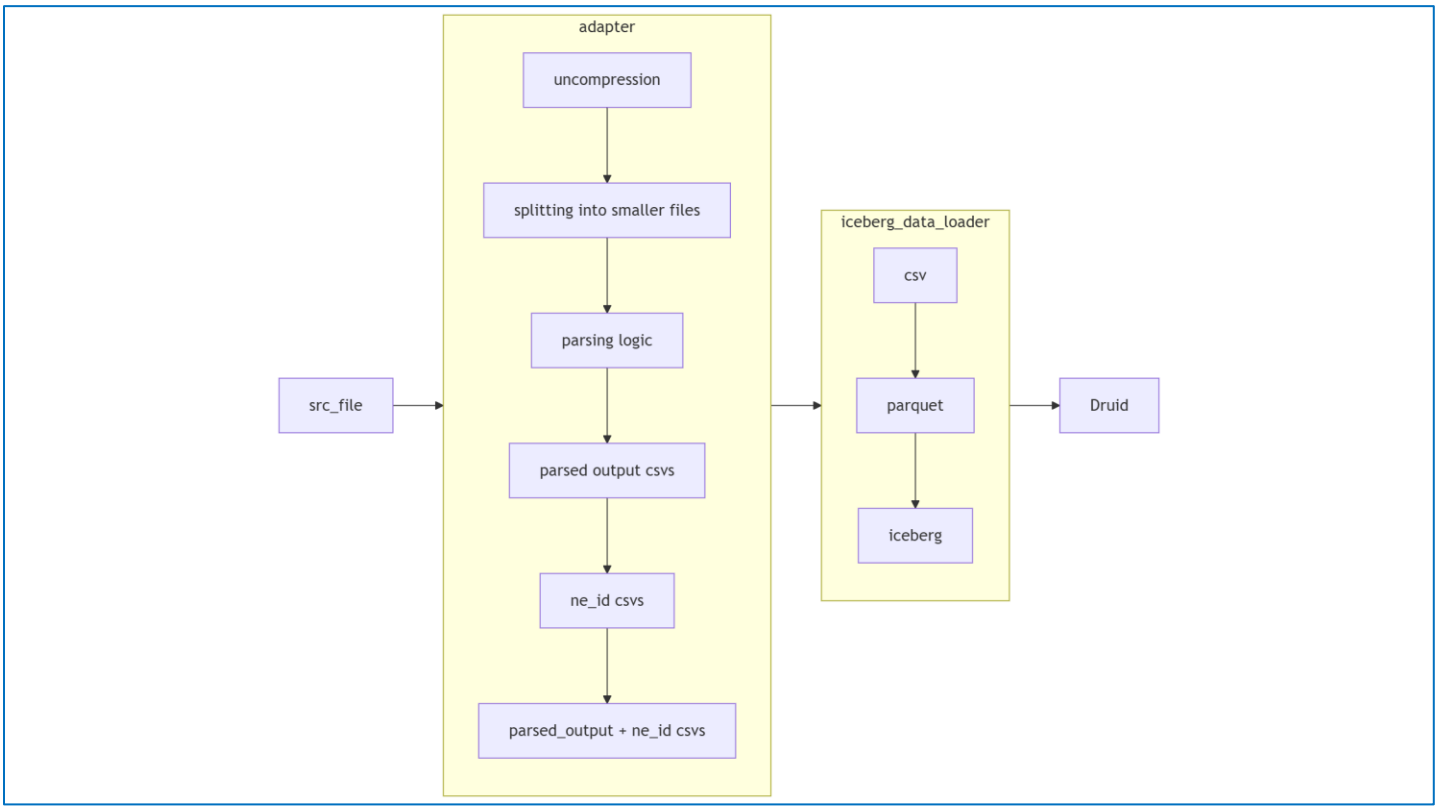


Column Name	Description	circle_wise_results
stream	Combination of OEM, technology, and class_type (CM, PM).	
scope	Baseband or ERBS; valid only for certain streams.	
circle_id	Circle information such as GUJ, DEL, etc.	
src_landing_today	Count of source files landed today in staging.	True
src_landing_total	Total count of source files landed in staging.	True
src_archived_today	Count of source files archived today.	True
src_archived_yesterday	Count of source files archived yesterday.	True
kafka_landing_today	Count of files/messages landed in Kafka today.	False
kafka_landing_yesterday	Count of files/messages landed in Kafka yesterday.	False
kafka_archive_total_today	Total Kafka archived messages/files today.	False
kafka_archive_total_yesterday	Total Kafka archived messages/files yesterday.	False
src_dead_files_today	Count of dead files in source for today.	False
src_dead_files_total	Total count of dead files in source.	False

split_xmls_archive_today	Count of split XML files archived today.	True
split_xmls_archive_total	Total count of split XML files archived.	True
ne_id_staging_today	Count of unique NE IDs in staging today.	False
ne_id_staging_yesterday	Count of unique NE IDs in staging yesterday.	False
max_lag	Maximum observed lag between Kafka/Druid ingestion.	False
min_lag	Minimum observed lag between Kafka/Druid ingestion.	False
druid_error_message	Error messages observed during Druid ingestion/query execution.	False
max_timestamp	Maximum of Maximum event timestamp found in the druid tables	True
min_timestamp	Minimum of maximum event timestamp found in the druid tables.	True
max_rollup_time	Maximum of Maximum event time till when rollup (aggregation) time recorded in Druid tables.	False
min_rollup_time	Minimum of Maximum event time till when rollup (aggregation) time recorded in Druid tables.	False
active_segment_count	Number of active Druid segments for the given stream.	False

ERCS 4G CM Data Loading:

Process Flow:



Server vs adapter:

Stream	parsing Worker Servers (no of workers)	generic workers (no of workers)
eric_4g_cm	10.180.83.161 (10) 10.180.83.253 (10)	10.180.83.161 (10) 10.180.83.253(10)
eric_5g_cm	10.180.83.161 (10) 10.180.83.253 (10)	10.180.83.161 (10) 10.180.83.253(10)
eric_2g_cm	10.180.83.161 (10) 10.180.83.253(10)	10.180.83.161 (10) 10.180.83.253(10)
samsung_4g_cm	10.180.83.161 (10) 10.180.83.253(10)	10.180.83.161 (10) 10.180.83.253(10)

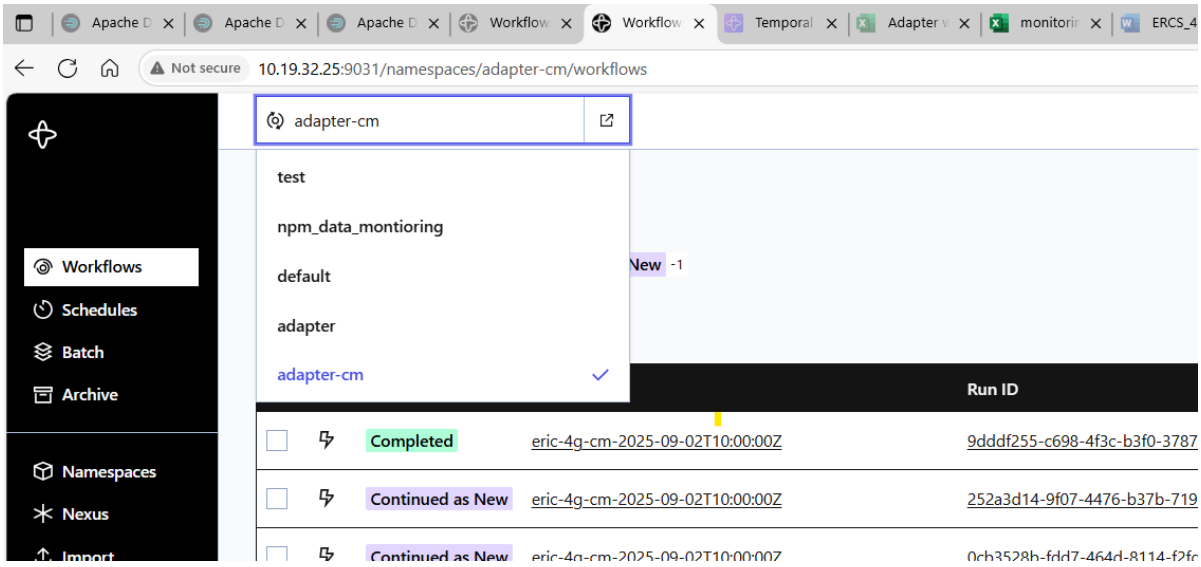
samsung_5g_cm	10.180.83.161 (10) 10.180.83.253(10)	10.180.83.161 (10) 10.180.83.253(10)
huw_2g_cm	10.180.83.161 (10) 10.180.83.253(10)	10.180.83.161 (10) 10.180.83.253(10)
huw_4g_cm	10.180.83.161 (10) 10.180.83.253(10)	10.180.83.161 (10) 10.180.83.253(10)
mav_4g_cm	10.180.83.161 (10) 10.180.83.253(10)	10.180.83.161 (10) 10.180.83.253(10)
mav_5g_cm	10.180.83.161 (10) 10.180.83.253(10)	10.180.83.161 (10) 10.180.83.253(10)

Temporal UI Link:

- Direct link: http://10.19.32.25:9031/namespaces/adapter-cm/workflows

Or

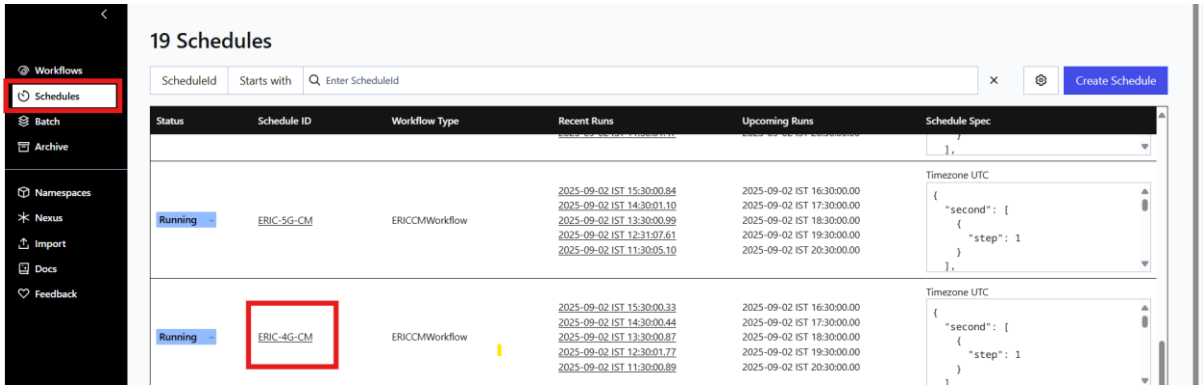
- Open link: http://10.19.32.25:9031/ here in namespace option select adapter-cm to view cm workflows.



Workflow schedules: Here all workflows are scheduled

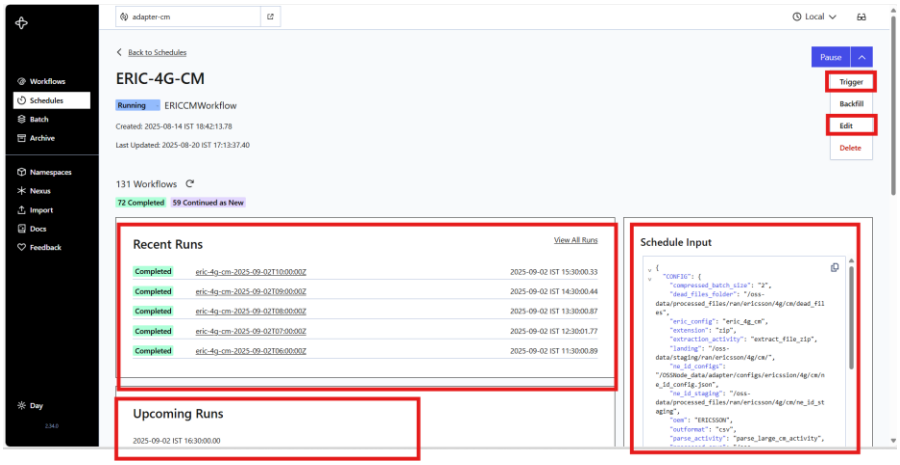
1. There is schedule tab on left sie of Temporal UI.

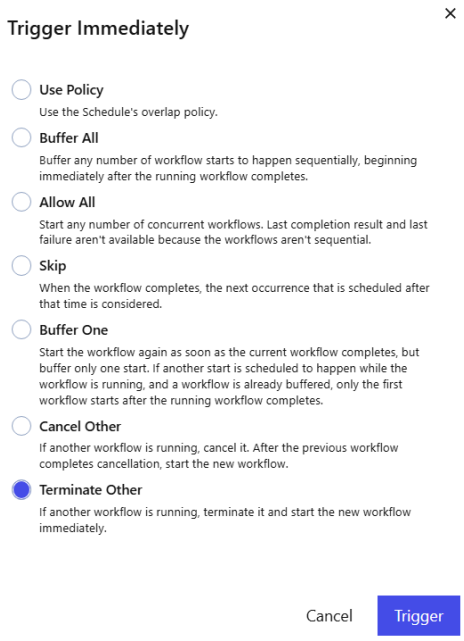
2. Click on ERIC_4G_CM schedule



How to run the workflow instantly:

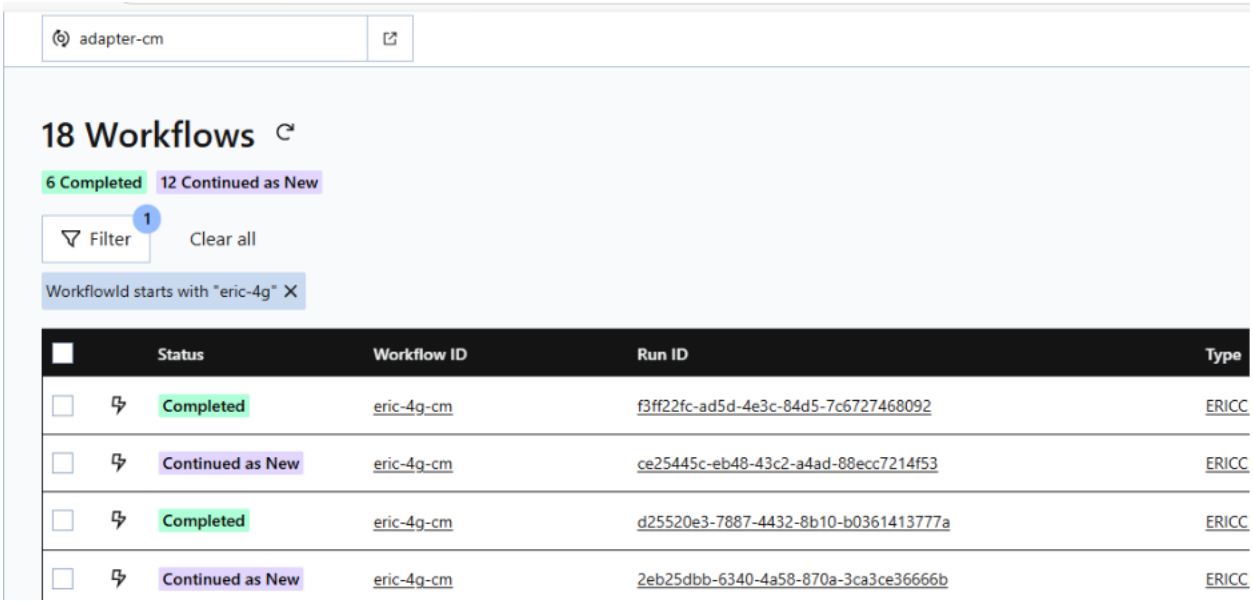
1. There is a trigger option as shown in below to trigger the workflow instantly and chose the option **Terminate Other**





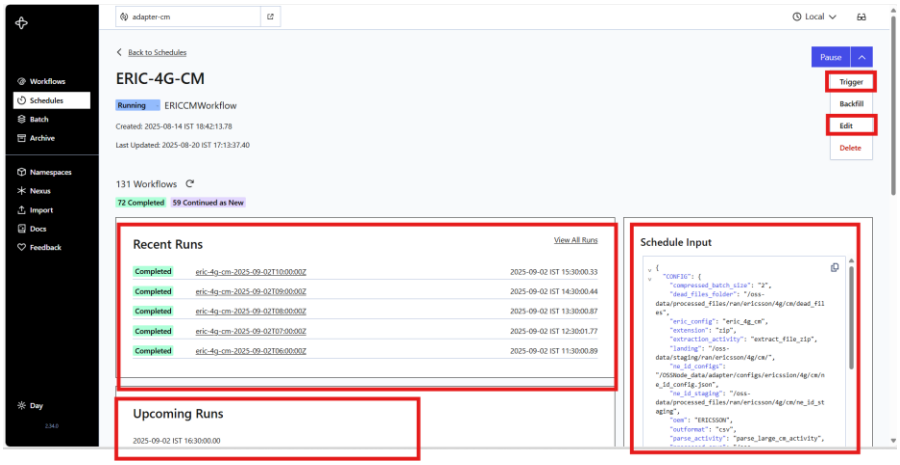
Information about worker:

- In /OSSNode_data/adapter/ in this path we have supervisord.conf file in all servers. That give information about workers
- This server has multiple workers running but for 4g cm we only need workers which are listed below, if any of these are down it might affect the workflow.
- 10.180.83.161, 10.180.83.253: cm_worker and cm-generic-task-worker (generic_worker_cm) and go_worker(eric_4g__worker)
- generic_worker_cm: generic activities extracting zip files,archival of processed files, moving non-parsable files to dead file folder and creating ne_id masters.
- cm_worker : this worker is responsible for splitting of xml file to small xml, initially we get large xml files which are difficult to parse so we split them and parse those.
- go_worker: this worker is responsible for parsing xml to csv. (go_worker which is named as eric_4g__worker in the supervisor is responsible for parsing of ericsson 4g cm and 5g cm files)



Workflow Config can be checked:

1. Json_Input config is specified in the schedule as shown in below



Data loading structure:

- First data is pushed in landing path: landing path can be found from temporal workflow config or above mention config path
- Then data is passed in processed_files path after it is processed successfully

- If there is any file which has issues in parsing or uncompressing it can be found in dead_files_folder
- Then Once CSV are created data will be created in processed_csv path
- As In landing path have zip because we are having source files Extension type with zip (4G CM)

Script to run workers(when workflow is not running):

- Go to the path /OSSNode_data/adapter
- Check if the supervisor/workers are running, command to check -

```
[Admin@NiFi10 oss-data]$ ps -ef | grep supervisord
Admin     482497      1  0 12:41 ?        00:00:04 /usr/bin/python3 /home/Admin/.local/bin/supervisord -c supervisord.conf
Admin     499734  482181  0 13:00 pts/1    00:00:00 grep --color=auto supervisord
```

ps -ef | grep supervisord

ps -ef | grep worker

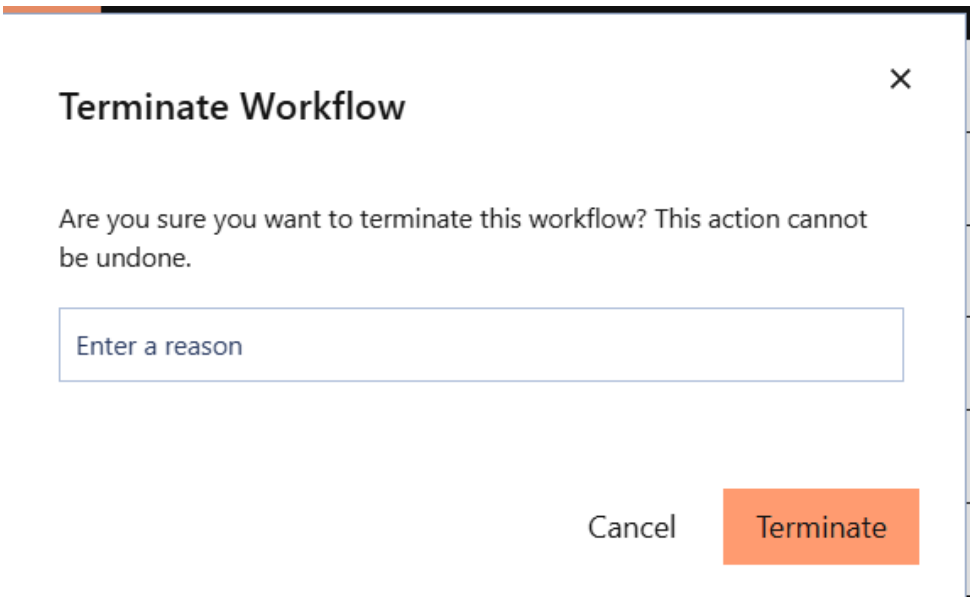
```
Admin    4025905 4025896  1 Jun18 ?        00:17:15 python /OSSNode_data/adapter/parser/temporal/worker_cm.py
Admin    4025906 4025896  1 Jun18 ?        00:16:44 python /OSSNode_data/adapter/parser/temporal/worker_cm.py
Admin    4025907 4025896  1 Jun18 ?        00:13:28 python /OSSNode_data/adapter/parser/temporal/worker_cm.py
Admin    4025908 4025896  1 Jun18 ?        00:13:11 python /OSSNode_data/adapter/parser/temporal/worker_cm.py
Admin    4025909 4025896  1 Jun18 ?        00:13:34 python /OSSNode_data/adapter/parser/temporal/worker_cm.py
Admin    4025910 4025896  0 Jun18 ?        00:12:13 python /OSSNode_data/adapter/parser/temporal/worker_cm.py
Admin    4025911 4025896  1 Jun18 ?        00:17:22 python /OSSNode_data/adapter/parser/temporal/worker_cm.py
Admin    4025912 4025896  1 Jun18 ?        00:15:32 python /OSSNode_data/adapter/parser/temporal/worker_cm.py
Admin    4025913 4025896  1 Jun18 ?        00:13:32 python /OSSNode_data/adapter/parser/temporal/worker_cm.py
Admin    4025914 4025896  1 Jun18 ?        00:14:10 python /OSSNode_data/adapter/parser/temporal/worker_cm.py
Admin    4025915 4025896  0 Jun18 ?        00:00:00 sh /OSSNode_data/adapter/go_worker.sh
Admin    4025916 4025896  0 Jun18 ?        00:00:00 sh /OSSNode_data/adapter/go_worker.sh
Admin    4025917 4025896  0 Jun18 ?        00:00:00 sh /OSSNode_data/adapter/go_worker.sh
Admin    4025921 4025896  0 Jun18 ?        00:00:00 sh /OSSNode_data/adapter/go_worker.sh
Admin    4025927 4025896  0 Jun18 ?        00:00:00 sh /OSSNode_data/adapter/go_worker.sh
Admin    4025931 4025896  0 Jun18 ?        00:00:00 sh /OSSNode_data/adapter/go_worker.sh
Admin    4025933 4025896  0 Jun18 ?        00:00:00 sh /OSSNode_data/adapter/go_worker.sh
Admin    4025940 4025896  0 Jun18 ?        00:00:00 sh /OSSNode_data/adapter/go_worker.sh
Admin    4025943 4025896  0 Jun18 ?        00:00:00 sh /OSSNode_data/adapter/go_worker.sh
Admin    4025944 4025896  0 Jun18 ?        00:00:00 sh /OSSNode_data/adapter/go_worker.sh
Admin    4025982 4025896  0 Jun18 ?        00:00:40 python /OSSNode_data/adapter/parser/temporal/cm-generic-task-worker.py
Admin    4026006 4025896  0 Jun18 ?        00:00:35 python /OSSNode_data/adapter/parser/temporal/cm-generic-task-worker.py
Admin    4026016 4025896  0 Jun18 ?        00:00:35 python /OSSNode_data/adapter/parser/temporal/cm-generic-task-worker.py
Admin    4026017 4025896  0 Jun18 ?        00:00:41 python /OSSNode_data/adapter/parser/temporal/cm-generic-task-worker.py
Admin    4026023 4025896  0 Jun18 ?        00:07:32 python /OSSNode_data/adapter/parser/temporal/cm-generic-task-worker.py
Admin    4026034 4025896  2 Jun18 ?        00:33:32 python /OSSNode_data/adapter/parser/temporal/cm-generic-task-worker.py
Admin    4026041 4025896  2 Jun18 ?        00:27:44 python /OSSNode_data/adapter/parser/temporal/cm-generic-task-worker.py
Admin    4026045 4025896  0 Jun18 ?        00:00:36 python /OSSNode_data/adapter/parser/temporal/cm-generic-task-worker.py
Admin    4026046 4025896  1 Jun18 ?        00:21:50 python /OSSNode_data/adapter/parser/temporal/cm-generic-task-worker.py
Admin    4026067 4025896  0 Jun18 ?        00:00:33 python /OSSNode_data/adapter/parser/temporal/cm-generic-task-worker.py
```

If the supervisord is not running, run it using command
supervisord -c supervisord.conf

And once the workers are up, we can trigger the workflow or wait to workflow to start for next schedule

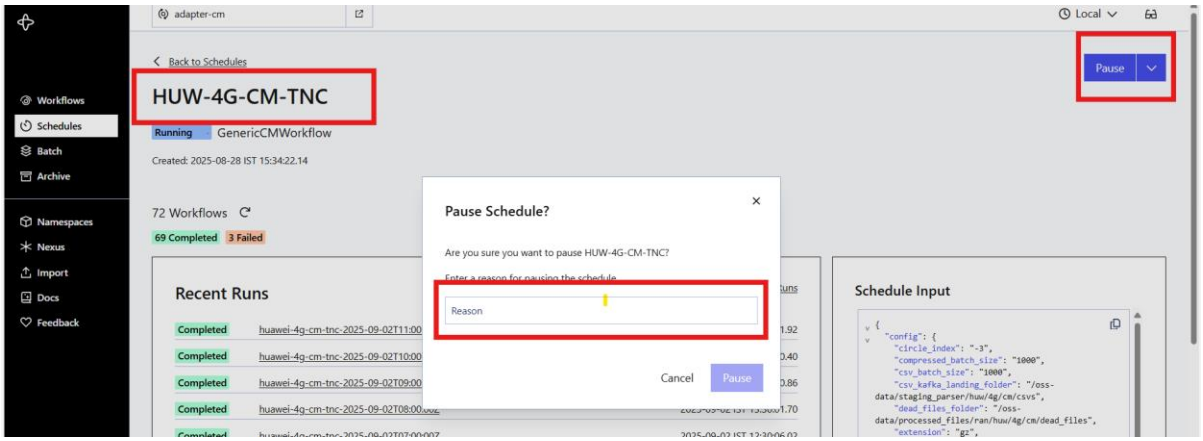
Kill the workflow:

1. Choose the workflow id to stop
2. Click on checkbox of corresponding workflow as shown in below
3. Will see terminate option and click on it to terminate the workflow
4. Enter the reason for termination and click on terminate

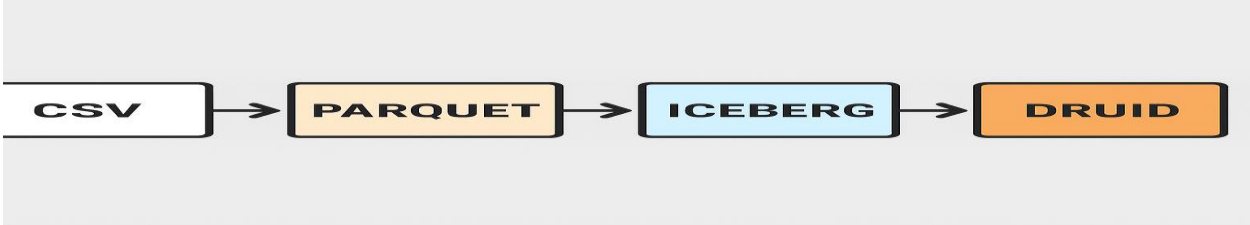


Pausing the Schedule:

1. Go to Schedule tab
2. Click on schedule which needs to be paused
3. Click on pause option
4. Enter the reason to pause
5. Click on Pause



Inserting Data to Druid through Iceberg:



We have two workflows one to insert data into iceberg from csv and other to ingest data from iceberg to druid.

1. **CSV to Iceberg workflow**:
   Workflow Params:
   a. Workflow Type: CsvToIcebergWorkflow
   b. Namespace: default
   c. Url: http://10.19.32.25:9031/namespaces/default/workflows/
   d. Task queue name: cm-file-processing-task-queue
   e. Schedule ID: csv-to-iceberg-ericsson-4g-cm
   f. Workers running on: http://10.180.83.142 and http://10.180.83.143
   g. Input Data:

```
{
  "archieve_folder": "/oss-data/kafka/archive/",
  "csv_folder_path": "/oss-data/processed_files/ran/ericsson/4g/cm/csvs3/",
  "csv_parquet_timeout": 800,
  "iceberg_catalog": "cm_iceberg",
  "oem": "ERCS",
  "parquet_to_iceberg_timeout": 1200,
  "technology": "4G",
  "warehouse_name": "s3testing/iceberg_ercs_prod_test"
}
```

Archieve Folder Path:

a. Csv archieve: /oss-data/kafka/archive/csvs/ERCS_4G_cm/
b. Parquet archieve: /oss-data/kafka/archive/parquet/ERCS/4G

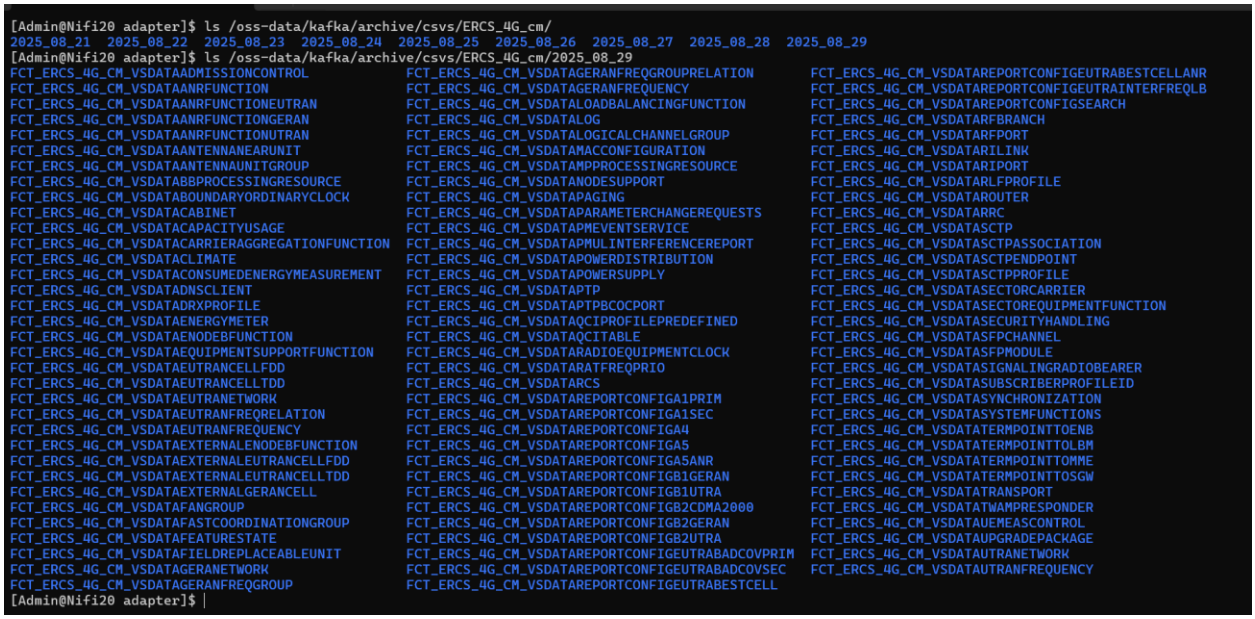
Parquet Folder Path: (here parquet files will land after conversion from csv)

a. Files will be landed in the following folder's sub-folder which are created topic-wise.
b. Folder path: /oss-data/processed_files/ran/ericsson/4g/cm/parquet_output/ERCS/4G/
c. Sub-Folders:

```
[Admin@Nifi20 adapter]$ ls /oss-data/processed_files/ran/ericsson/4g/cm/parquet_output/ERCS/4G/
FCT_ERCS_4G_CM_VSDATAADMISSIONCONTROL            FCT_ERCS_4G_CM_VSDATAGERANFREQGROUPRELATION      FCT_ERCS_4G_CM_VSDATAREPORTCONFIGEUTRABESTCELLANR
FCT_ERCS_4G_CM_VSDATAANRFUNCTION                 FCT_ERCS_4G_CM_VSDATAGERANFREQUENCY              FCT_ERCS_4G_CM_VSDATAREPORTCONFIGEUTRAINTERFREQLB
FCT_ERCS_4G_CM_VSDATAANRFUNCTIONEUTRAN           FCT_ERCS_4G_CM_VSDATALOADBALANCINGFUNCTION       FCT_ERCS_4G_CM_VSDATAREPORTCONFIGSEARCH
FCT_ERCS_4G_CM_VSDATAANRFUNCTIONGERAN            FCT_ERCS_4G_CM_VSDATALOG                         FCT_ERCS_4G_CM_VSDATARFBRANCH
FCT_ERCS_4G_CM_VSDATAANRFUNCTIONUTRAN            FCT_ERCS_4G_CM_VSDATALOGICALCHANNELGROUP         FCT_ERCS_4G_CM_VSDATARFPORT
FCT_ERCS_4G_CM_VSDATAANTENNANEARUNIT             FCT_ERCS_4G_CM_VSDATAMACCONFIGURATION            FCT_ERCS_4G_CM_VSDATARILINK
FCT_ERCS_4G_CM_VSDATAANTENNAUNITGROUP            FCT_ERCS_4G_CM_VSDATAMPPROCESSINGRESOURCE        FCT_ERCS_4G_CM_VSDATARIPORT
FCT_ERCS_4G_CM_VSDATABBPROCESSINGRESOURCE        FCT_ERCS_4G_CM_VSDATANODESUPPORT                 FCT_ERCS_4G_CM_VSDATARLFPROFILE
FCT_ERCS_4G_CM_VSDATABOUNDARYORDINARYCLOCK       FCT_ERCS_4G_CM_VSDATAPAGING                      FCT_ERCS_4G_CM_VSDATAROUTER
FCT_ERCS_4G_CM_VSDATACABINET                     FCT_ERCS_4G_CM_VSDATAPARAMETERCHANGEREQUESTS     FCT_ERCS_4G_CM_VSDATARRC
FCT_ERCS_4G_CM_VSDATACAPACITYUSAGE               FCT_ERCS_4G_CM_VSDATAPMEVENTSERVICE              FCT_ERCS_4G_CM_VSDATASCTP
FCT_ERCS_4G_CM_VSDATACARRIERAGGREGATIONFUNCTION  FCT_ERCS_4G_CM_VSDATAPMULINTERFERENCEREPORT      FCT_ERCS_4G_CM_VSDATASCTPASSOCIATION
FCT_ERCS_4G_CM_VSDATACLIMATE                     FCT_ERCS_4G_CM_VSDATAPOWERDISTRIBUTION           FCT_ERCS_4G_CM_VSDATASCTPENDPOINT
FCT_ERCS_4G_CM_VSDATACONSUMEDENERGYMEASUREMENT   FCT_ERCS_4G_CM_VSDATAPOWERSUPPLY                 FCT_ERCS_4G_CM_VSDATASCTPPROFILE
FCT_ERCS_4G_CM_VSDATADNSCLIENT                   FCT_ERCS_4G_CM_VSDATAPTP                         FCT_ERCS_4G_CM_VSDATASECTORCARRIER
FCT_ERCS_4G_CM_VSDATADRXPROFILE                  FCT_ERCS_4G_CM_VSDATAPTPBCOCPORT                 FCT_ERCS_4G_CM_VSDATASECTOREQUIPMENTFUNCTION
FCT_ERCS_4G_CM_VSDATAENERGYMETER                 FCT_ERCS_4G_CM_VSDATAQCIPROFILEPREDEFINED        FCT_ERCS_4G_CM_VSDATASECURITYHANDLING
FCT_ERCS_4G_CM_VSDATAENODEBFUNCTION              FCT_ERCS_4G_CM_VSDATAQCITABLE                    FCT_ERCS_4G_CM_VSDATASFPCHANNEL
FCT_ERCS_4G_CM_VSDATAEQUIPMENTSUPPORTFUNCTION    FCT_ERCS_4G_CM_VSDATARADIOEQUIPMENTCLOCK         FCT_ERCS_4G_CM_VSDATASFPMODULE
FCT_ERCS_4G_CM_VSDATAEUTRANCELLFDD               FCT_ERCS_4G_CM_VSDATARATFREQPRIO                 FCT_ERCS_4G_CM_VSDATASIGNALINGRADIOBEARER
FCT_ERCS_4G_CM_VSDATAEUTRANCELLTDD               FCT_ERCS_4G_CM_VSDATARCS                         FCT_ERCS_4G_CM_VSDATASUBSCRIBERPROFILEID
FCT_ERCS_4G_CM_VSDATAEUTRANETWORK                FCT_ERCS_4G_CM_VSDATAREPORTCONFIGA1PRIM          FCT_ERCS_4G_CM_VSDATASYNCHRONIZATION
FCT_ERCS_4G_CM_VSDATAEUTRANFREQRELATION          FCT_ERCS_4G_CM_VSDATAREPORTCONFIGA1SEC           FCT_ERCS_4G_CM_VSDATASYSTEMFUNCTIONS
FCT_ERCS_4G_CM_VSDATAEUTRANFREQUENCY             FCT_ERCS_4G_CM_VSDATAREPORTCONFIGA4              FCT_ERCS_4G_CM_VSDATATERMPOINTTOENB
FCT_ERCS_4G_CM_VSDATAEXTERNALENODEBFUNCTION      FCT_ERCS_4G_CM_VSDATAREPORTCONFIGA5              FCT_ERCS_4G_CM_VSDATATERMPOINTTOLBM
FCT_ERCS_4G_CM_VSDATAEXTERNALEUTRANCELLFDD       FCT_ERCS_4G_CM_VSDATAREPORTCONFIGA5ANR           FCT_ERCS_4G_CM_VSDATATERMPOINTTOMME
FCT_ERCS_4G_CM_VSDATAEXTERNALEUTRANCELLTDD       FCT_ERCS_4G_CM_VSDATAREPORTCONFIGB1GERAN         FCT_ERCS_4G_CM_VSDATATERMPOINTTOSGW
FCT_ERCS_4G_CM_VSDATAEXTERNALGERANCELL           FCT_ERCS_4G_CM_VSDATAREPORTCONFIGB1UTRA          FCT_ERCS_4G_CM_VSDATATRANSPORT
FCT_ERCS_4G_CM_VSDATAFANGROUP                    FCT_ERCS_4G_CM_VSDATAREPORTCONFIGB2CDMA2000      FCT_ERCS_4G_CM_VSDATATWAMPRESPONDER
FCT_ERCS_4G_CM_VSDATAFASTCOORDINATIONGROUP       FCT_ERCS_4G_CM_VSDATAREPORTCONFIGB2GERAN         FCT_ERCS_4G_CM_VSDATAUEMEASCONTROL
FCT_ERCS_4G_CM_VSDATAFEATURESTATE                FCT_ERCS_4G_CM_VSDATAREPORTCONFIGB2UTRA          FCT_ERCS_4G_CM_VSDATAUPGRADEPACKAGE
FCT_ERCS_4G_CM_VSDATAFIELDREPLACEABLEUNIT        FCT_ERCS_4G_CM_VSDATAREPORTCONFIGEUTRABADCOVPRIM FCT_ERCS_4G_CM_VSDATAUTRANETWORK
FCT_ERCS_4G_CM_VSDATAGERANETWORK                 FCT_ERCS_4G_CM_VSDATAREPORTCONFIGEUTRABADCOVSEC  FCT_ERCS_4G_CM_VSDATAUTRANFREQUENCY
FCT_ERCS_4G_CM_VSDATAGERANFREQGROUP              FCT_ERCS_4G_CM_VSDATAREPORTCONFIGEUTRABESTCELL
[Admin@Nifi20 adapter]$
```

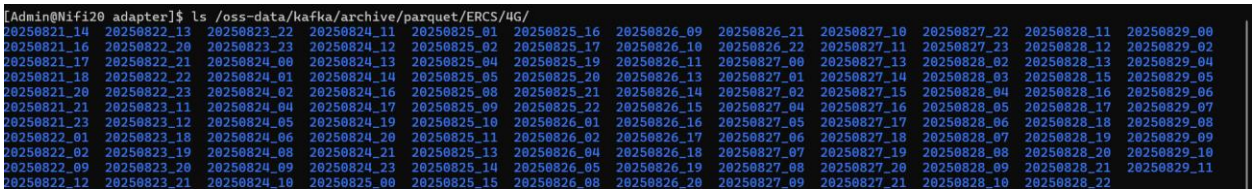
CSV Archieve Path:

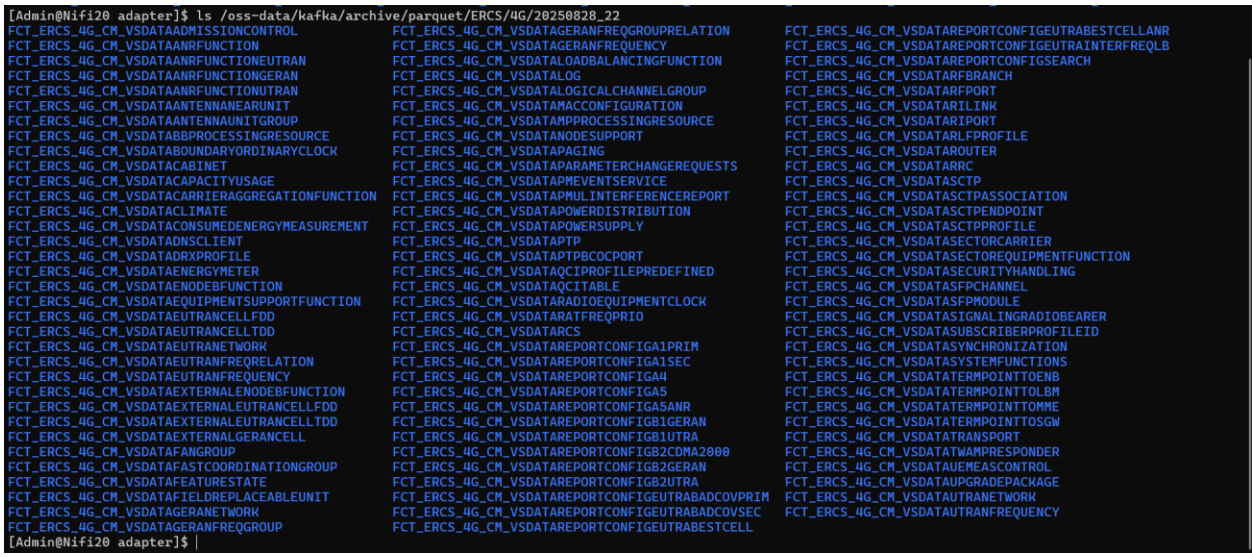
a. In the following folder path sub-folders are created date wise on which file is processed and in-side date folder topic wise sub-folders are created.
b. Folder Path: /oss-data/kafka/archive/csvs/ERCS_4G_cm/
c. Sub-folders:



Parquet Archieve Path:

a. In the following folder path sub-folders are created date_hour wise on which file is processed and in-side date folder topic wise sub-folders are created.
b. Folder Path: /oss-data/kafka/archive/parquet/ERCS/4G/
c. Sub-folders:



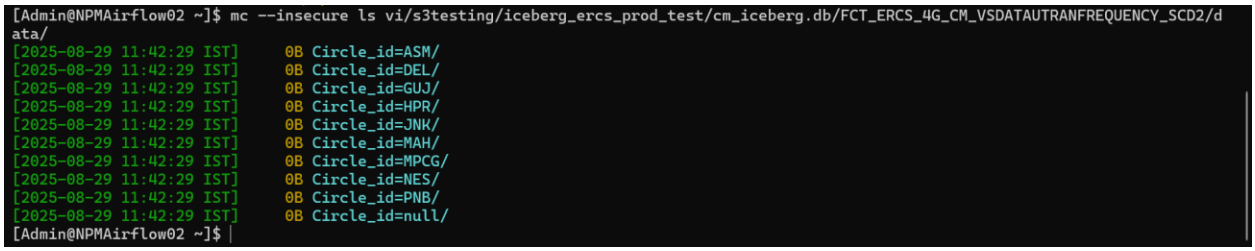


Check Data in Iceberg:

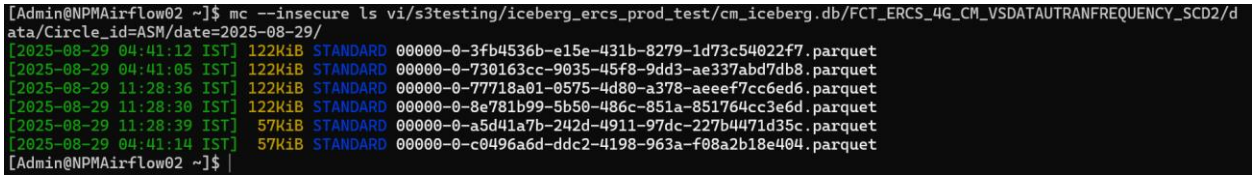
a. Data in iceberg is getting pushed by the name of the tables in druid, they are partitioned by
b. Tables (Folders) are created druid table name wise:

mc --insecure ls vi/s3testing/iceberg_ercs_prod_test/cm_iceberg.db/ | grep FCT_ERCS_4G_CM

c. Tables are partitioned by Circle_id and date:
mc --insecure ls vi/s3testing/iceberg_ercs_prod_test/cm_iceberg.db/FCT_ERCS_4G_CM_VSDATAUTRANFREQUENCY_SCD2/data/



mc --insecure ls vi/s3testing/iceberg_ercs_prod_test/cm_iceberg.db/FCT_ERCS_4G_CM_VSDATAUTRANFREQUENCY_SCD2/data/Circle_id=ASM/



Activities:

1. run_csv_to_parquet_conversion_activity:

Activity convert the csv files present in csv_folder_path which not starts with ne_id i.e. excluding ne_id files to parquet files in Parquet folder path mentioned above path in topic wise sub-folder

2. move_parquet_to_iceberg:
   a. Push parquet files present in the parquet_output folder to iceberg by inserting data into table according to parquet sub-folder name and partitioning data by Circle_id and date as presented above and parquet files gets moved to parquet archieve folder.
   b. Currently we are processing batch of 20 parquet files at a time due to memory issue.

2. Iceberg to Druid:

Workflow Params:

a. Workflow Type: IcebergToDruidWorkflow
b. Namespace: default
c. Url: http://10.19.32.25:9031/namespaces/default/workflows/
d. Task queue name: cm-file-processing-task-queue
e. Schedule ID: iceberg_to_druid_ercs_4g_cm
f. Workers running on: http://10.180.83.142 and http://10.180.83.143
g. Input Data:


```
{
  "batch_size": 5,
  "iceberg_catalog": "cm_iceberg",
  "iceberg_druid_timeout": 600,
  "oem": "ERCS",
  "table_names": [
    "FCT_ERCS_4G_CM_VSDATAADMISSIONCONTROL_SCD2",
    "FCT_ERCS_4G_CM_VSDATAANRFUNCTION_SCD2",
    "FCT_ERCS_4G_CM_VSDATAANRFUNCTIONEUTRAN_SCD2",
    "FCT_ERCS_4G_CM_VSDATAANRFUNCTIONGERAN_SCD2",
    "FCT_ERCS_4G_CM_VSDATAANRFUNCTIONUTRAN_SCD2",
    "FCT_ERCS_4G_CM_VSDATAANTENNANEARUNIT_SCD2",
    "FCT_ERCS_4G_CM_VSDATAANTENNAUNITGROUP_SCD2",
    "FCT_ERCS_4G_CM_VSDATABBPROCESSINGRESOURCE_SCD2",
    "FCT_ERCS_4G_CM_VSDATABOUNDARYORDINARYCLOCK_SCD2",
    "FCT_ERCS_4G_CM_VSDATACABINET_SCD2",
    "FCT_ERCS_4G_CM_VSDATACAPACITYUSAGE_SCD2",
    "FCT_ERCS_4G_CM_VSDATASUBSCRIBERPROFILEID_SCD2",
    "FCT_ERCS_4G_CM_VSDATASYNCHRONIZATION_SCD2",
    "FCT_ERCS_4G_CM_VSDATASYSTEMFUNCTIONS_SCD2",
    "FCT_ERCS_4G_CM_VSDATATERMPOINTTOENB_SCD2",
    "FCT_ERCS_4G_CM_VSDATATERMPOINTTOLBM_SCD2",
    "FCT_ERCS_4G_CM_VSDATATERMPOINTTOMME_SCD2",
    "FCT_ERCS_4G_CM_VSDATATERMPOINTTOSGW_SCD2",
    "FCT_ERCS_4G_CM_VSDATATRANSPORT_SCD2",
    "FCT_ERCS_4G_CM_VSDATATWAMPRESPONDER_SCD2",
    "FCT_ERCS_4G_CM_VSDATAUEMEASCONTROL_SCD2",
    "FCT_ERCS_4G_CM_VSDATAUPGRADEPACKAGE_SCD2",
    "FCT_ERCS_4G_CM_VSDATAUTRANETWORK_SCD2",
    "FCT_ERCS_4G_CM_VSDATAUTRANFREQUENCY_SCD2"
  ],
  "table_type": "CM",
  "technology": "4G",
  "warehouse_name": "s3testing/iceberg_ercs_prod_test"
}
```


Workflow:

1. Separate activities get created for every table mentioned in the table_names list.
2. Intiating the batch_size of activities parallely at a time.
3. The timeout of activity is iceberg_druid_timeout mentioned in the input data of workflow

Activity:

1. We maintain the snapshot of the data being send to druid from iceberg for every table in a postgres table keeping prev_snapshot_id, current_snapshot_id and

task_id represents the task_id being submitted to druid for processing data of between prev and current snapshot_id

2. If the table is new data is been send for the first time to druid via iceberg the there will be no entry and last snapshot id sent will be null, we will process all the snapshots present for that table and submit the task
3. If the table is already processed we will check if the task submitted for the last snapshots with help of task-id and if the task has failed we will take last snapshot id as prev_snapshot_id otherwise as current_snapshot_id and pick the data from last snapshot id to latest snapshot id and submit task to druid
4. We will wait for iceberg_druid_timeout to complete the task which gets mostly completed otherwise move to other table

Postgres Table Used for storing snapshot information for iceberg to druid:

postgres details:

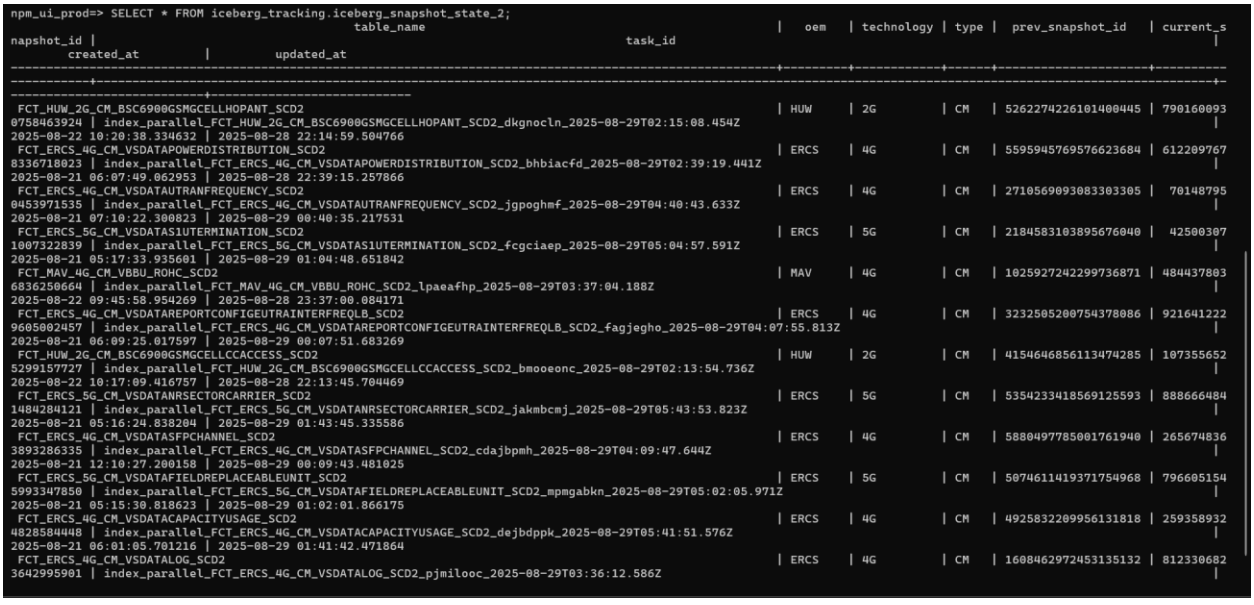
IP:10.115.1.27:5432

Username: npm

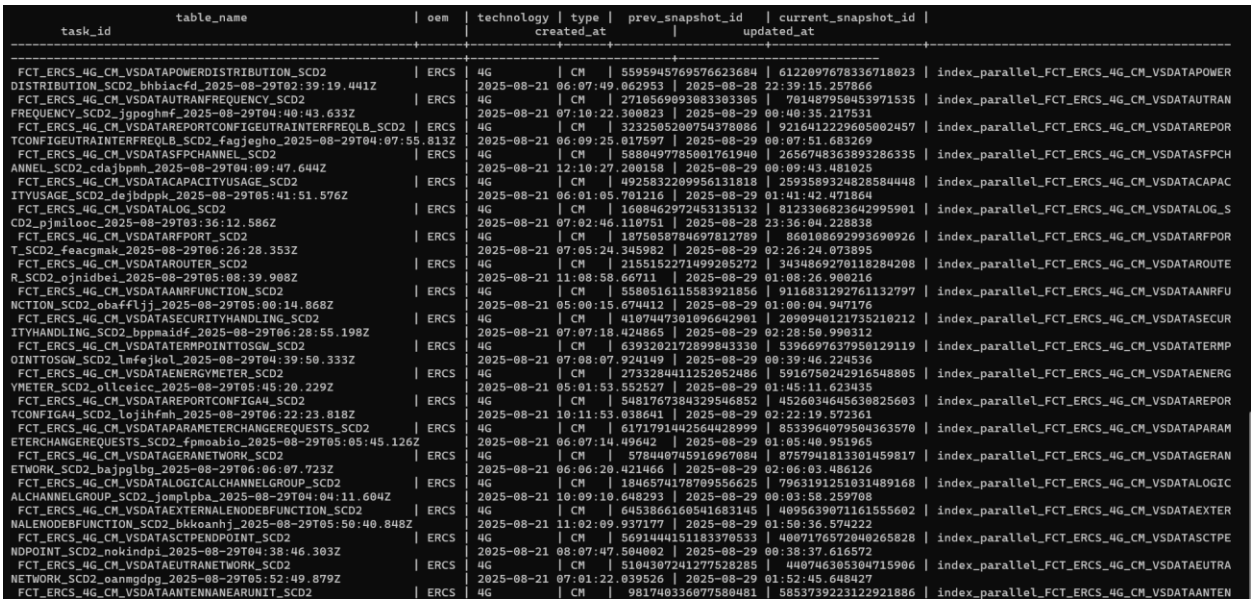
Database: npm_ui_prod

Password: Npm1234

Table_name: iceberg_tracking.iceberg_snapshot_state_2



```
npm_ui_prod=> SELECT * FROM iceberg_tracking.iceberg_snapshot_state_2;
                                    table_name                                    |   oem   | technology | type |  prev_snapshot_id   | current_s
napshot_id |                                                                  task_id
        created_at         |         updated_at
-----------------------------------------------------------------------------------------------------------+---------+------------+------+---------------------+----------
-----------+------------------------------------------------------------------------------------------------------------------------------------------------------------------+-
---------------------------+---------------------------
 FCT_HUW_2G_CM_BSC6900GSMGCELLHOPANT_SCD2                                      | HUW     | 2G         | CM   | 5262274226101400445 | 790160093
0758463924 | index_parallel_FCT_HUW_2G_CM_BSC6900GSMGCELLHOPANT_SCD2_dkgnocln_2025-08-29T02:15:08.454Z
2025-08-22 10:20:38.334632 | 2025-08-28 22:14:59.504766
 FCT_ERCS_4G_CM_VSDATAPOWERDISTRIBUTION_SCD2                                   | ERCS    | 4G         | CM   | 5595945769576623684 | 612209767
8336718023 | index_parallel_FCT_ERCS_4G_CM_VSDATAPOWERDISTRIBUTION_SCD2_bhbiacfd_2025-08-29T02:39:19.441Z
2025-08-21 06:07:49.062953 | 2025-08-28 22:39:15.257866
 FCT_ERCS_4G_CM_VSDATAUTRANFREQUENCY_SCD2                                      | ERCS    | 4G         | CM   | 2710569093083303305 |  70148795
0453971535 | index_parallel_FCT_ERCS_4G_CM_VSDATAUTRANFREQUENCY_SCD2_jgpoghmf_2025-08-29T04:40:43.633Z
2025-08-21 07:10:22.300823 | 2025-08-29 00:40:35.217531
 FCT_ERCS_5G_CM_VSDATAS1UTERMINATION_SCD2                                      | ERCS    | 5G         | CM   | 2184583103895676040 |  42500307
1007322839 | index_parallel_FCT_ERCS_5G_CM_VSDATAS1UTERMINATION_SCD2_fcgciaep_2025-08-29T05:04:57.591Z
2025-08-21 05:17:33.935601 | 2025-08-29 01:04:48.651842
 FCT_MAV_4G_CM_VBBU_ROHC_SCD2                                                  | MAV     | 4G         | CM   | 1025927242299736871 | 484437803
6836250664 | index_parallel_FCT_MAV_4G_CM_VBBU_ROHC_SCD2_lpaeafhp_2025-08-29T03:37:04.188Z
2025-08-22 09:45:58.954269 | 2025-08-28 23:37:00.084171
 FCT_ERCS_4G_CM_VSDATAREPORTCONFIGEUTRAINTERFREQLB_SCD2                        | ERCS    | 4G         | CM   | 3232505200754378086 | 921641222
9605002457 | index_parallel_FCT_ERCS_4G_CM_VSDATAREPORTCONFIGEUTRAINTERFREQLB_SCD2_fagjegho_2025-08-29T04:07:55.813Z
2025-08-21 06:09:25.017597 | 2025-08-29 00:07:51.683269
 FCT_HUW_2G_CM_BSC6900GSMGCELLCCACCESS_SCD2                                    | HUW     | 2G         | CM   | 4154646856113474285 | 107355652
5299157727 | index_parallel_FCT_HUW_2G_CM_BSC6900GSMGCELLCCACCESS_SCD2_bmooeonc_2025-08-29T02:13:54.736Z
2025-08-22 10:17:09.416757 | 2025-08-28 22:13:45.704469
 FCT_ERCS_5G_CM_VSDATANRSECTORCARRIER_SCD2                                     | ERCS    | 5G         | CM   | 5354233418569125593 | 888666484
1484284121 | index_parallel_FCT_ERCS_5G_CM_VSDATANRSECTORCARRIER_SCD2_jakmbcmj_2025-08-29T05:43:53.823Z
2025-08-21 05:16:24.838204 | 2025-08-29 01:43:45.335586
 FCT_ERCS_4G_CM_VSDATASFPCHANNEL_SCD2                                          | ERCS    | 4G         | CM   | 5880497785001761940 | 265674836
3893286335 | index_parallel_FCT_ERCS_4G_CM_VSDATASFPCHANNEL_SCD2_cdajbpmh_2025-08-29T04:09:47.644Z
2025-08-21 12:10:27.200158 | 2025-08-29 00:09:43.481025
 FCT_ERCS_5G_CM_VSDATAFIELDREPLACEABLEUNIT_SCD2                                | ERCS    | 5G         | CM   | 5074611419371754968 | 796605154
5993347850 | index_parallel_FCT_ERCS_5G_CM_VSDATAFIELDREPLACEABLEUNIT_SCD2_mpmgabkn_2025-08-29T05:02:05.971Z
2025-08-21 05:15:30.818623 | 2025-08-29 01:02:01.866175
 FCT_ERCS_4G_CM_VSDATACAPACITYUSAGE_SCD2                                       | ERCS    | 4G         | CM   | 4925832209956131818 | 259358932
48285844448 | index_parallel_FCT_ERCS_4G_CM_VSDATACAPACITYUSAGE_SCD2_dejbdppk_2025-08-29T05:41:51.576Z
2025-08-21 06:01:05.701216 | 2025-08-29 01:41:42.471864
 FCT_ERCS_4G_CM_VSDATALOG_SCD2                                                 | ERCS    | 4G         | CM   | 1608462972453135132 | 812330682
3642995901 | index_parallel_FCT_ERCS_4G_CM_VSDATALOG_SCD2_pjmilooc_2025-08-29T03:36:12.586Z
```



table_name	oem	technology	type	prev_snapshot_id	current_snapshot_id	task_id	created_at	updated_at
FCT_ERCS_4G_CM_VSDATAPOWERDISTRIBUTION_SCD2	ERCS	4G	CM	5595945769576623684	6122097678336718023	index_parallel_FCT_ERCS_4G_CM_VSDATAPOWERDISTRIBUTION_SCD2_bhbiacfd_2025-08-29T02:39:19.441Z	2025-08-21 06:07:49.062953	2025-08-28 22:39:15.257866
FCT_ERCS_4G_CM_VSDATAUTRANFREQUENCY_SCD2	ERCS	4G	CM	2710569093083303305	7014879504539715 35	index_parallel_FCT_ERCS_4G_CM_VSDATAUTRANFREQUENCY_SCD2_jgpoghmf_2025-08-29T04:40:43.633Z	2025-08-21 07:10:22.300823	2025-08-29 00:40:35.217531
FCT_ERCS_4G_CM_VSDATAREPORTCONFIGEUTRAINTERFREQLB_SCD2	ERCS	4G	CM	3232505200754378086	9216412229605002457	index_parallel_FCT_ERCS_4G_CM_VSDATAREPORTCONFIGEUTRAINTERFREQLB_SCD2_fagjegho_2025-08-29T04:07:55.813Z	2025-08-21 06:09:25.017597	2025-08-29 00:07:51.683269
FCT_ERCS_4G_CM_VSDATASFPCHANNEL_SCD2	ERCS	4G	CM	5880497785001761940	2656748363893286335	index_parallel_FCT_ERCS_4G_CM_VSDATASFPCHANNEL_SCD2_cdajbpmh_2025-08-29T04:09:47.644Z	2025-08-21 12:10:27.200158	2025-08-29 00:09:43.481025
FCT_ERCS_4G_CM_VSDATACAPACITYUSAGE_SCD2	ERCS	4G	CM	4925832209956131818	2593589324828584448	index_parallel_FCT_ERCS_4G_CM_VSDATACAPACITYUSAGE_SCD2_dejbdppk_2025-08-29T05:41:51.576Z	2025-08-21 06:01:05.701216	2025-08-29 01:41:42.471864
FCT_ERCS_4G_CM_VSDATALOG_SCD2	ERCS	4G	CM	1608462972453135132	8123306823642995901	index_parallel_FCT_ERCS_4G_CM_VSDATALOG_SCD2_pjmilooc_2025-08-29T03:36:12.586Z	2025-08-21 07:02:46.110751	2025-08-28 23:36:04.228838
FCT_ERCS_4G_CM_VSDATARFPORT_SCD2	ERCS	4G	CM	1875058784697812789	860108692993690926	index_parallel_FCT_ERCS_4G_CM_VSDATARFPORT_SCD2_feacgmak_2025-08-29T06:26:28.353Z	2025-08-21 07:05:24.345982	2025-08-29 02:26:24.073895
FCT_ERCS_4G_CM_VSDATAROUTER_SCD2	ERCS	4G	CM	2155152271499205272	3434869270118284208	index_parallel_FCT_ERCS_4G_CM_VSDATAROUTER_SCD2_ojnidbei_2025-08-29T05:08:39.908Z	2025-08-21 11:08:58.66711	2025-08-29 01:08:26.900216
FCT_ERCS_4G_CM_VSDATAANRFUNCTION_SCD2	ERCS	4G	CM	5580516115583921856	9116831292761132797	index_parallel_FCT_ERCS_4G_CM_VSDATAANRFUNCTION_SCD2_obaffljj_2025-08-29T05:00:14.868Z	2025-08-21 05:00:15.674412	2025-08-29 01:00:04.947176
FCT_ERCS_4G_CM_VSDATASECURITYHANDLING_SCD2	ERCS	4G	CM	4107447301096642901	2090940121735210212	index_parallel_FCT_ERCS_4G_CM_VSDATASECURITYHANDLING_SCD2_bppmaidf_2025-08-29T06:28:55.198Z	2025-08-21 07:07:18.424865	2025-08-29 02:28:50.990312
FCT_ERCS_4G_CM_VSDATATERMPOINTTOSGW_SCD2	ERCS	4G	CM	6393202172899843330	5396697637950129119	index_parallel_FCT_ERCS_4G_CM_VSDATATERMPOINTTOSGW_SCD2_lmfejkol_2025-08-29T04:39:50.333Z	2025-08-21 07:08:07.924149	2025-08-29 00:39:46.224536
FCT_ERCS_4G_CM_VSDATAENERGYMETER_SCD2	ERCS	4G	CM	2733284411252052486	5916750242916548805	index_parallel_FCT_ERCS_4G_CM_VSDATAENERGYMETER_SCD2_ollceicc_2025-08-29T05:45:20.229Z	2025-08-21 05:01:53.552527	2025-08-29 01:45:11.623435
FCT_ERCS_4G_CM_VSDATAREPORTCONFIG4_SCD2	ERCS	4G	CM	5481767384329546852	4526034645630825603	index_parallel_FCT_ERCS_4G_CM_VSDATAREPORTCONFIG4_SCD2_lojihfmh_2025-08-29T06:22:23.818Z	2025-08-21 10:11:53.038641	2025-08-29 02:22:19.572361
FCT_ERCS_4G_CM_VSDATAPARAMETERCHANGEREQUESTS_SCD2	ERCS	4G	CM	6171791442564428999	8533964079504363570	index_parallel_FCT_ERCS_4G_CM_VSDATAPARAMETERCHANGEREQUESTS_SCD2_fpmoabio_2025-08-29T05:05:45.126Z	2025-08-21 06:07:14.49642	2025-08-29 01:05:40.951965
FCT_ERCS_4G_CM_VSDATAGERANETWORK_SCD2	ERCS	4G	CM	5784407459169670 84	8757941813301459817	index_parallel_FCT_ERCS_4G_CM_VSDATAGERANETWORK_SCD2_bajpglbg_2025-08-29T06:06:07.723Z	2025-08-21 06:06:20.421466	2025-08-29 02:06:03.486126
FCT_ERCS_4G_CM_VSDATALOGICALCHANNELGROUP_SCD2	ERCS	4G	CM	1846574178709556625	7963191251031489168	index_parallel_FCT_ERCS_4G_CM_VSDATALOGICALCHANNELGROUP_SCD2_jomplpba_2025-08-29T04:04:11.604Z	2025-08-21 10:09:10.648293	2025-08-29 00:03:58.259708
FCT_ERCS_4G_CM_VSDATAEXTERNALENODEBFUNCTION_SCD2	ERCS	4G	CM	6453866160541683145	4095639071161555602	index_parallel_FCT_ERCS_4G_CM_VSDATAEXTERNALENODEBFUNCTION_SCD2_bkkoanhj_2025-08-29T05:50:40.848Z	2025-08-21 11:02:09.937177	2025-08-29 01:50:36.574222
FCT_ERCS_4G_CM_VSDATASCTPENDPOINT_SCD2	ERCS	4G	CM	5691444151183370533	4007176572040265828	index_parallel_FCT_ERCS_4G_CM_VSDATASCTPENDPOINT_SCD2_nokindpi_2025-08-29T04:38:46.303Z	2025-08-21 08:07:47.504002	2025-08-29 00:38:37.616572
FCT_ERCS_4G_CM_VSDATAEUTRANETWORK_SCD2	ERCS	4G	CM	5104307241277528285	4407463053047159 06	index_parallel_FCT_ERCS_4G_CM_VSDATAEUTRANETWORK_SCD2_oanmgdpg_2025-08-29T05:52:49.879Z	2025-08-21 07:01:22.039526	2025-08-29 01:52:45.648427
FCT_ERCS_4G_CM_VSDATAANTENNANEARUNIT_SCD2	ERCS	4G	CM	9817403360775804 81	5853739223122921886	index_parallel_FCT_ERCS_4G_CM_VSDATAANTEN		

NE-ID Movement to DRUID

We are using two workflows to move ne_id to druid from csv:

1. Move csv data to iceberg
2. Iceberg to druid

CSV to Iceberg

Workers required: worker-ingestion-activity.py, worker_ingestion_workflow.py

Workflow Name: NEIDCsvToIcebergWorkflow

Schedule Name: csv-iceberg-ne-id

Queue: cm-file-processing-task-queue-1

Url: http://10.19.32.25:9031/namespaces/default/workflows/

Namespace: default

Worker Running on: 10.115.1.78

Params:


```
{
  "batch_size": 1,
  "CSVToIcebergNEID": [
    {
      "csv_folder_path": "/oss-data/processed_files/ran/ericsson/4g/cm/csvs3/",
      "oem": "ERICSSON",
      "technology": "4G",
      "archieve_folder": "/oss-data/kafka/archive/",
      "iceberg_catalog": "cm_neid_master",
      "warehouse_name": "s3testing/iceberg_ne_id_prod_test",
      "iceberg_table_name": "NEID_MASTER",
      "csv_parquet_timeout": 1800,
      "parquet_to_iceberg_timeout": 1800
    },
    {
      "csv_folder_path": "/oss-data/processed_files/ran/ericsson/2g/cm/csvs/",
      "oem": "ERICSSON",
      "technology": "2G",
      "archieve_folder": "/oss-data/kafka/archive/",
      "iceberg_catalog": "cm_neid_master",
      "warehouse_name": "s3testing/iceberg_ne_id_prod_test",
      "iceberg_table_name": "NEID_MASTER",
      "csv_parquet_timeout": 1800,
      "parquet_to_iceberg_timeout": 1800
    },
    {
      "csv_folder_path": "/oss-data/processed_files/ran/ericsson/5g/cm/csvs/",
      "oem": "ERICSSON",
      "technology": "5G",
      "archieve_folder": "/oss-data/kafka/archive/",
      "iceberg_catalog": "cm_neid_master",
      "warehouse_name": "s3testing/iceberg_ne_id_prod_test",
      "iceberg_table_name": "NEID_MASTER",
      "csv_parquet_timeout": 1800,
      "parquet_to_iceberg_timeout": 1800
    },
    {
      "csv_folder_path": "/oss-data/processed_files/ran/samsung/4g/cm/csvs/",
      "oem": "SAMSUNG",
      "technology": "4G",
      "archieve_folder": "/oss-data/kafka/archive/",
      "iceberg_catalog": "cm_neid_master",
```

```
    "warehouse_name": "s3testing/iceberg_ne_id_prod_test",
    "iceberg_table_name": "NEID_MASTER",
    "csv_parquet_timeout": 1800,
    "parquet_to_iceberg_timeout": 1800
  },
  {
    "csv_folder_path": "/oss-data/processed_files/ran/samsung/2g/cm/csvs/",
    "oem": "SAMSUNG",
    "technology": "2G",
    "archieve_folder": "/oss-data/kafka/archive/",
    "iceberg_catalog": "cm_neid_master",
    "warehouse_name": "s3testing/iceberg_ne_id_prod_test",
    "iceberg_table_name": "NEID_MASTER",
    "csv_parquet_timeout": 1800,
    "parquet_to_iceberg_timeout": 1800
  },
  {
    "csv_folder_path": "/oss-data/processed_files/ran/samsung/5g/cm/csvs/",
    "oem": "SAMSUNG",
    "technology": "5G",
    "archieve_folder": "/oss-data/kafka/archive/",
    "iceberg_catalog": "cm_neid_master",
    "warehouse_name": "s3testing/iceberg_ne_id_prod_test",
    "iceberg_table_name": "NEID_MASTER",
    "csv_parquet_timeout": 1800,
    "parquet_to_iceberg_timeout": 1800
  },
  {
    "csv_folder_path": "/oss-data/staging_parser/huw/4g/cm/csvs/",
    "oem": "HUW",
    "technology": "4G",
    "archieve_folder": "/oss-data/kafka/archive/",
    "iceberg_catalog": "cm_neid_master",
    "warehouse_name": "s3testing/iceberg_ne_id_prod_test",
    "iceberg_table_name": "NEID_MASTER",
    "csv_parquet_timeout": 1800,
    "parquet_to_iceberg_timeout": 1800
  },
  {
    "csv_folder_path": "/oss-data/staging_parser/huw/2g/cm/csvs/",
    "oem": "HUW",
    "technology": "2G",
    "archieve_folder": "/oss-data/kafka/archive/",
    "iceberg_catalog": "cm_neid_master",
    "warehouse_name": "s3testing/iceberg_ne_id_prod_test",
    "iceberg_table_name": "NEID_MASTER",
```

```
      "csv_parquet_timeout": 1800,
      "parquet_to_iceberg_timeout": 1800
    },
    {
      "csv_folder_path": "/oss-data/processed_files/ran/mav/4g/cm/csvs/",
      "oem": "MAV",
      "technology": "4G",
      "archieve_folder": "/oss-data/kafka/archive/",
      "iceberg_catalog": "cm_neid_master",
      "warehouse_name": "s3testing/iceberg_ne_id_prod_test",
      "iceberg_table_name": "NEID_MASTER",
      "csv_parquet_timeout": 1800,
      "parquet_to_iceberg_timeout": 1800
    },
    {
      "csv_folder_path": "/oss-data/processed_files/ran/mav/5g/cm/csvs/",
      "oem": "MAV",
      "technology": "5G",
      "archieve_folder": "/oss-data/kafka/archive/",
      "iceberg_catalog": "cm_neid_master",
      "warehouse_name": "s3testing/iceberg_ne_id_prod_test",
      "iceberg_table_name": "NEID_MASTER",
      "csv_parquet_timeout": 1800,
      "parquet_to_iceberg_timeout": 1800
    }
  ]
}
```


Activities Involved:

1. CSV to Parquet Conversion (run_csv_to_parquet_conversion_ne_id_activity)
2. Parquet to iceberg (move_parquet_to_iceberg_ne_id)

CSV to Parquet Conversion:

1. Csv landing folder: /oss-data/processed_files/ran/ericsson/4g/cm/csvs3
2. Parquet folder: /oss-data/processed_files/ran/ericsson/4g/cm/parquet_output/ERICSSON/4G/NEID_MASTER/
3. Csv archieve folder: /oss-data/kafka/archive/csvs/ERICSSON_4G_cm/2025_09_01/NEID_MASTER/

4. It picks the csv files from the csv landing folder and creates parquet files in NEID_MASTER folder.

Parquet to Iceberg Conversion:

1. Parquet files get picked from the parquet NEID_MASTER folder and gets ingested to iceberg in the batch 20 files at a time.
2. We partition iceberg table by Name (i.e. Topic Name), Circle_id and Date.
3. We can see the iceberg tables been created at the mini io client as shown below

Iceberg to Druid Workflow:

Workers required: worker-ingestion-activity.py, worker_ingestion_workflow.py

Schedule Name: iceberg-to-druid-ne-id

Schedule Frequency: 15 min

Workflow Name: IcebergDruidNEIDWorkflow

Namespace: default

Url: http://10.19.32.25:9031/namespaces/default/workflows/

Task Queue: cm-file-processing-task-queue-1

Worker Running on: 10.115.1.78

Params:

```
{
  "table_name": "NEID_MASTER",
  "catalog_name": "cm_neid_master",
  "warehouse_name": "s3testing/iceberg_ne_id_prod_test",
  "max_snapshots": 10,
  "batch_size": 5
}
```

Activities:

1. iceberg_to_druid_ingestion_ne_id_activity

2. process_iceberg_to_druid_ne_id_activity
3. update_last_loaded_snapshot_id_activity

Workflow Description:

1. The activity **iceberg_to_druid_ingestion_ne_id_activity,** fetches the snapshot till we have to load and the list of the topic wise ne-id and there corresponding iceberg parquet files.
2. Creates batch of batch_size provided and fire activities parallely to fire the task to druid using activity **process_iceberg_to_druid_ne_id_activity**.
3. After all batch gets processed we update the snapshot for the table (NEID_MASTER) we processed using activity **update_last_loaded_snapshot_id_activity**

Activity Description:

1. **iceberg_to_druid_ingestion_ne_id_activity:**
    a. For the given iceberg table (NEID_MASTER) it will check the snapshot till which it has processed and takes maximum snapshots (max_snapshots)
    b. It returns the topic wise dict containing the list of corresponding iceberg parquet files for that topic and last_snapshot_id and current_snapshot_id.
2. **process_iceberg_to_druid_ne_id_activit**y:
    a. It takes the topic name and corresponding parquet files and fires task to druid
    b. Will wait for the task to be successfully or gets timeout after the activity timeout
    c. If the activity gets timeout the workflow will fail
3. **update_last_loaded_snapshot_id_activity:**
    a. After all the topics get processed will update the snapshot id in postgres using these function