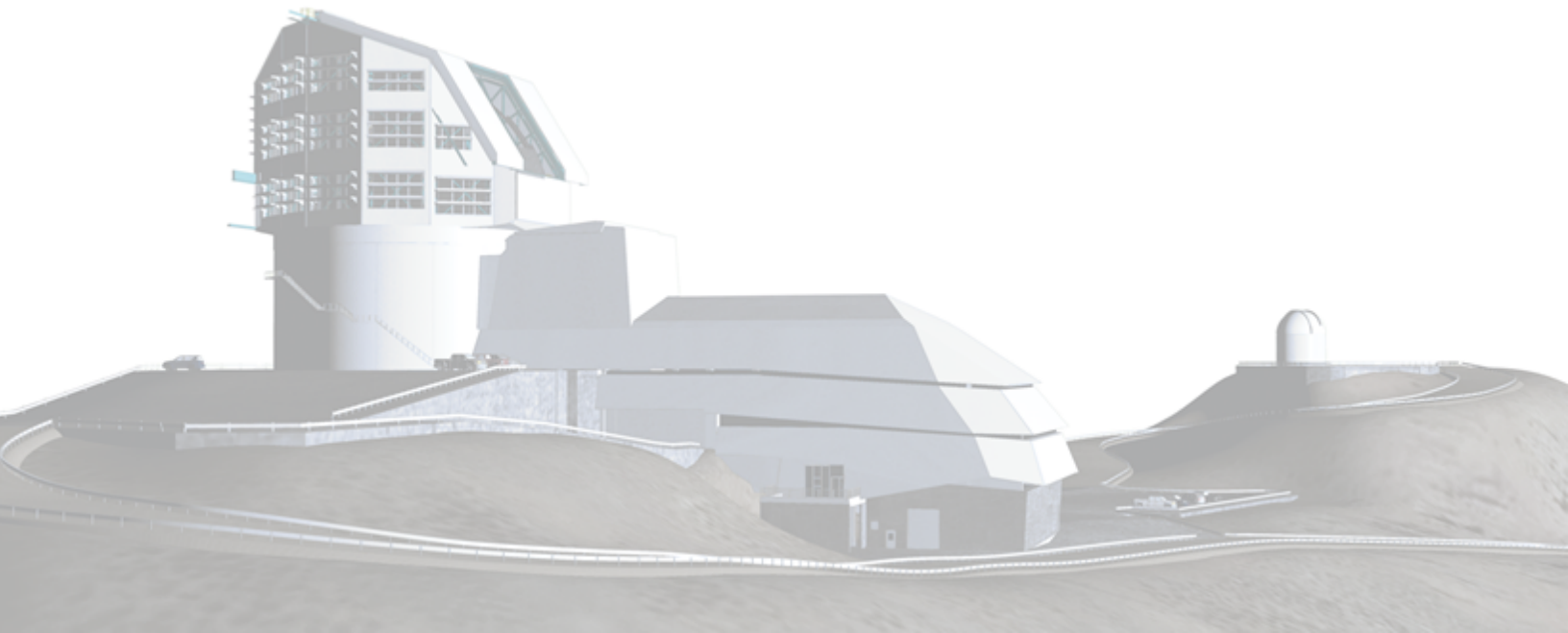


Vera C. Rubin Observatory
Software Test Report

LDM-503-EFDb: Replication of Summit EFD to USDF Test Plan and Report

Wil O'Mullane

DMTR-331

Latest Revision: 2024-02-08

Abstract

This is the test plan and report for **Replication of Summit EFD to USDF** (LDM-503-EFDb), an LSST milestone pertaining to the Data Management Subsystem.
This document is based on content automatically extracted from the Jira test database on 2024-02-08 . The most recent change to the document repository was on 2024-02-09.

Change Record

Version	Date	Description	Owner name
0.1	2021-10-19	First draft	K. Simon Krughoff
0.2	2023-06-16	Update for move to USDF	WOM
1.0	2024-02-08	Surcessfully executed on summit and USDF	WOM

Document curator: William O'Mullane
Document source location: `https://github.com/lsst-dm/DMTR-331`
Version from source repository: 7ba2dca

Contents

LDM-503-EFDb: Replication of Summit EFD to USDF Test Plan and Report

1 Introduction

1.1 Objectives

The purpose of this test plan is to describe all the necessary requirements and infrastructure for successfully testing the replication and archive of the Engineering Facility Database (EFD) as implemented with Kafka, InfluxDB and Chronograf from the summit to the USDF. This plan will describe the prerequisites for beginning a test campaign, step by step instructions for each test case and a description of the expected results and test artifacts.

NB: The use of the term reliability in this document is intended to indicate the number of messages produced relative to the number of messages recorded in the EFD. The system shall be considered reliable if at least 99.9% of produced messages are recorded.

At a high level, this test plan is intended to show that a nominally operating EFD at the summit is able to be replicated to the USDF and archived for future use either directly or via ingest into a secondary database management technology. We assume here that the archive technology will be parquet datasets stored on persistent/redundant disk at the USDF. There are no latency requirements in this test plan, but we will show that the replication and archiving are not falling behind relative to the summit instance in the aggregate. We choose a period of 6 days of continuous nominal operation in order to test the cases in this test plan.
Successful completion of the test campaign will show that:

1. users are able to access the same information at the USDF EFD that was originally ingested in the summit version
2. the reliability of the replication is better than the minimum of 99%
3. archive products are able to be used as a primary source of information for historical examination of EFD topics

1.2 System Overview

The tests will be carried out from within an instance of the notebook aspect of the RSP running at the data facility where the EFD replication is currently happening. An appropriate weekly version of the stack will be chosen.

1.3 Document Overview

This document was generated from Jira, obtaining the relevant information from the LVV-P90 Jira Test Plan and related Test Cycles (LVV-C181).

Section 1 provides an overview of the test campaign, the system under test (Data Management), the applicable documentation, and explains how this document is organized. Section 2 provides additional information about the test plan, like for example the configuration used for this test or related documentation. Section 3 describes the necessary roles and lists the individuals assigned to them.

Section 4 provides a summary of the test results, including an overview in Table 2, an overall assessment statement and suggestions for possible improvements. Section 5 provides detailed results for each step in each test case.

The current status of test plan LVV-P90 in Jira is **Approved** .

1.4 References

[1] **[DMTN-140]**, Comoretto, G., 2021, *Documentation Automation for the Verification and Validation of Rubin Observatory Software*, DMTN-140, URL `https://dmtn-140.lsst.io/`, Vera C. Rubin Observatory Data Management Technical Note

[2] **[DMTN-178]**, Comoretto, G., 2021, *Docsteady Usecases for Rubin Observatory Constructions*, DMTN-178, URL `https://dmtn-178.lsst.io/`, Vera C. Rubin Observatory Data Management Technical Note

[3] **[LSE-160]**, Selvy, B., 2013, *Verification and Validation Process*, LSE-160, URL https://ls.st/LSE-160

2 Test Plan Details

2.1 Data Collection

Observing is not required for this test campaign.

2.2 Verification Environment

The environment will be within notebooks running a modern stack.

2.3 Entry Criteria

1. Before beginning this test, a set of viability tests shall be performed. These will show:
   (a) The system demonstrates reliability (number of recorded messages/number of produced messages) of greater than 99.9%
   (b) The summit data is being replicated to the instance at USDF
   (c) Chronograf is set up and running at both the summit and USDF
2. The summit network and Kubernetes cluster are performing nominally
3. A number of telemetry topics are reliably producing telemetry at both low frequency (1 Hz) and high frequency (> 10 Hz).
4. The notebook aspect of the RSP is deployed in the summit Kubernetes cluster
5. The summit EFD is reliably replicated to an EFD instance running in a data facility
6. The notebook aspect of the RSP is deployed in the same data facility as that running the replicated EFD
7. The most recent version of the EFD client python modules are installed in the various deployed notebook aspects
8. The replication system is also successfully archiving EFD topics to parquet files on persistent disk at the data facility

2.4 Related Documentation

No additional documentation provided.

2.5 PMCS Activity

Primavera milestones related to the test campaign:

- LDM-503-EFDb

3 Personnel

The personnel involved in the test campaign is shown in the following table.

	T. Plan LVV-P90 owner:	**Wil O'Mullane**	
	T. Cycle LVV-C181 owner:	**Wil O'Mullane**	
Test Cases	**Assigned to**	**Executed by**	**Additional Test Personnel**
LVV-T2338	Wil O'Mullane	Wil O'Mullane	William O Mullane
LVV-T2339	Wil O'Mullane	Wil O'Mullane	William O'Mullane

4 Test Campaign Overview

4.1 Summary

T. Plan LVV-P90:	**LDM-503-EFDb: Replication of Summit EFD to USDF**			Approved
T. Cycle LVV-C181:	**LDM-503-EFDb: Replication of Summit EFD to USDF**			Done
Test Cases	**Ver.**	**Status**	**Comment**	**Issues**
LVV-T2338	2	Pass		
LVV-T2339	3	Pass		

Table 2: Test Campaign Summary

4.2 Overall Assessment

After the meltdown we seem to now have a good replica at USDF.

4.3 Recommended Improvements

Non.

5 Detailed Test Results

5.1 Test Cycle LVV-C181

Open test cycle *LDM-503-EFDb: Replication of Summit EFD to USDF* in Jira.

Test Cycle name: LDM-503-EFDb: Replication of Summit EFD to USDF
Status: Done

Engineering Facility Database of summit values queryable at USDF.

5.1.1 Software Version/Baseline

Not provided.

5.1.2 Configuration

Not provided.

5.1.3 Test Cases in LVV-C181 Test Cycle

5.1.3.1 LVV-T2338 - Replicated telemetry data agrees with telemetry produced at the summit

Version **2**. Status **Approved**. Open *LVV-T2338* test case in Jira.

Show that telemetry data can be accessed from the replicated EFD. Further, show that the values in the replicated database agree with the values in the summit EFD over a specified time range and set of topics.

This test case provides partial coverage of the requirement DMS-REQ-0168, Summit Facility Data Communications: ”The DMS shall provide data communications infrastructure to accept science data and associated metadata read-outs, and **the collection of ancillary and engineering data**, for transfer to the base facility.”, as adapted to the current design for EFD

replication (see DMTN-082).

Preconditions:
See prerequisites in the Test Plan LVV-P90

Execution status:

Final comment:

Detailed steps results LVV-C181-LVV-T2338 LVV-E1457-1845:
Note: Steps "Not Executed" and with No Result are not shown in this report.

Step LVV-E1457-1 Step Execution Status: **Pass**

Description
Log in to the USDF notebook aspect: https://usdf-rsp-dev.slac.stanford.edu/
Make sure to choose a recent weekly release and a large instance; record the chosen release and whether or not it is the current "recommended" release.

Expected Result
The JupyterLab interface is displayed in the browser

Actual Result
Logged in choose large with latest weekly (23)

Step LVV-E1457-2 Step Execution Status: **Pass**

Description
Open a notebook:

1. Navigate to the File->New->Notebook
2. When prompted, select the LSST kernel

Expected Result
An empty notebook running in the LSST kernel

Actual Result

Created a notebook LVV-T2338

Step LVV-E1457-3 Step Execution Status: **Pass**

Description

Connect to the USDF EFD.

Use the EFD identifier name from the test case parameter, usdf_efd , unless it is necessary to change at the time the test is executed. Record the value used if different.

Example Code

```
from lsst_efd_client import EfdClient
efd = EfdClient('usdf_efd')
```

Expected Result

A notebook with an instance of the 'EfdClient' configured to talk to the USDF EFD

Actual Result

```
from lsst_efd_client import EfdClient, resample
loc = 'usdf'

client = EfdClient(f'{loc}_efd')
client.output = 'dataframe'
cl=client.influx_client
```

```
Last executed at 2023-06-16 19:28:45 in 829ms
```

Step LVV-E1457-4 Step Execution Status: **Pass**

Description

Since we need to compare results between the summit and the USDF, also create a connection to the summit EFD. Use the EFD identifier name from the test case parameter, summit_efd , unless it is necessary to change at the time the test is executed. Record the value used if different.

Example Code

```
efd_summit = EfdClient('summit_efd')
```

Expected Result

Another cell in the notebook with a second EFD connection to the summit EFD

Actual Result

times out - you can not access efd_summit from USDF.
Connected to https://summit-lsp.lsst.codes/nb this allows connections to both USDF and Summit EFD.

Step LVV-E1457-5 Step Execution Status: **Pass**

Description

Choose 5 topics to query and select a 6 day window of data. The window is arbitrary, but must be explicit (not relative to now()) so that it can be reproduced. The topics should be chosen to sample the various topic contexts. I.e. the topics should be chosen to sample both diagnostic topics like heartbeat monitors as well as both high and low cadence telemetry topics to get a broad view on how the system behaves with different kinds of topics.

Expected Result

A list of 5 valid SAL topics to be queried and a time window defined as astropy.Time objects.

Actual Result

```
Random selction of five 'summit' topics which hace data on the start day of the test window


lsst.sal.MTDomeTrajectory.logevent_heartbeat had 30 messages in first 30 min of 2024-01-01
lsst.sal.ATMonochromator.logevent_heartbeat had 29 messages in first 30 min of 2024-01-01
lsst.sal.MTM1M3.logevent_heartbeat had 30 messages in first 30 min of 2024-01-01
lsst.sal.ESS.lightningStrikeStatus had 30 messages in first 30 min of 2024-01-01
lsst.sal.Watcher.logevent_heartbeat had 30 messages in first 30 min of 2024-01-01
Random selction of five 'summit' topics ['lsst.sal.MTDomeTrajectory.logevent_heartbeat', 'lsst.sal.ATMonochromator.logevent
```

Step LVV-E1457-6 Step Execution Status: **Pass**

Description

Issue selections at both the summit and the data facility. These selections should select all fields for the chosen topics.

Expected Result

A total of 10 pandas.DataFrame objects, 5 each for the summit and replicated EFDs. Each topic requires a separate query, so each will get its own DataFrame. All fields in each topic should be selected.

Actual Result

---

combined with next step

Step LVV-E1457-7 Step Execution Status: **Pass**

Description

First compare the index for each topic between the summit and replicated EFD. There should be:

1. The same number of samples in each topic for each location
2. Given 1) each time stamp should represent the same time

Reliability of the replication must be at better than 99%. If there are samples missing from the replicated datasets, confirm that the length of the replicated DataFrame divided by the length of the summit DataFrame is greater than 0.99 for all topics.

Expected Result

A cell in the notebook showing the DataFrames are the same length per topic between the summit and the replicated EFD. A cell showing the times in the index are the same for each topic. This could be done by converting to seconds and showing the difference is zero for every sample.

If there are missing samples, the replication should be better than 99%. If it is not, the deviation must be traced to an intervening event or system other than the replication system itself to explain the discrepancy.

Actual Result

```
The note book check the counts per topic over the day as well as comparing actual values in the topic.

Checking 2024-01-01
Compare lsst.sal.MTDomeTrajectory.logevent_heartbeat
Compare lsst.sal.ATMonochromator.logevent_heartbeat
Compare lsst.sal.MTM1M3.logevent_heartbeat
Compare lsst.sal.ESS.lightningStrikeStatus
Compare lsst.sal.Watcher.logevent_heartbeat
 Checking 2024-01-02
Compare lsst.sal.MTDomeTrajectory.logevent_heartbeat
Compare lsst.sal.ATMonochromator.logevent_heartbeat
Compare lsst.sal.MTM1M3.logevent_heartbeat
Compare lsst.sal.ESS.lightningStrikeStatus
Compare lsst.sal.Watcher.logevent_heartbeat
```

```
 Checking 2024-01-03
Compare lsst.sal.MTDomeTrajectory.logevent_heartbeat
Compare lsst.sal.ATMonochromator.logevent_heartbeat
Compare lsst.sal.MTM1M3.logevent_heartbeat
Compare lsst.sal.ESS.lightningStrikeStatus
Compare lsst.sal.Watcher.logevent_heartbeat
 Checking 2024-01-04
Compare lsst.sal.MTDomeTrajectory.logevent_heartbeat
Compare lsst.sal.ATMonochromator.logevent_heartbeat
Compare lsst.sal.MTM1M3.logevent_heartbeat
Compare lsst.sal.ESS.lightningStrikeStatus
Compare lsst.sal.Watcher.logevent_heartbeat
 Checking 2024-01-05
Compare lsst.sal.MTDomeTrajectory.logevent_heartbeat
Compare lsst.sal.ATMonochromator.logevent_heartbeat
Compare lsst.sal.MTM1M3.logevent_heartbeat
Compare lsst.sal.ESS.lightningStrikeStatus
Compare lsst.sal.Watcher.logevent_heartbeat
 Checking 2024-01-06
Compare lsst.sal.MTDomeTrajectory.logevent_heartbeat
Compare lsst.sal.ATMonochromator.logevent_heartbeat
Compare lsst.sal.MTM1M3.logevent_heartbeat
Compare lsst.sal.ESS.lightningStrikeStatus
Compare lsst.sal.Watcher.logevent_heartbeat
```

Step LVV-E1457-8 Step Execution Status: **Blocked**

Description

Compare the fields for each topic between the summit and the replicated EFDs. They should be equivalent to double precision. This can be done by looping over the topics and fields and showing numpy.all (or similar) evaluates to True.

Expected Result

A cell or cells showing that all fields for all topics evaluate as equivalent given appropriate precision.

Actual Result

The note book check tactual values in the topic in the previous step.

Step LVV-E1457-9 Step Execution Status: **Pass**

Description

Examine the summit messages to confirm that the reliability is better than 99.9% for all topics. Keep in mind that

the private_seqNum is intended to be a sequentially increasing index of the messages, but that it gets reset after every CSC reboot. This must be accounted for by applying an offset when a reset is observed.
Show the reliability is better than 99.9% by showing that the private_seqNum is sequential better than 99.9% of the time (when correcte for resets).

Expected Result

A histogram or similar showing that the difference private_seqNum[1:] - private_seqnum[:-1] is 1 more than 99.9% of the time.

Actual Result

```
lsst.sal.GIS.logevent_heartbeat private_seqNum increases 100.0% of the sequence
lsst.sal.ATOODS.logevent_heartbeat private_seqNum increases 100.0% of the sequence
lsst.sal.Test.logevent_heartbeat private_seqNum increases 100.0% of the sequence
lsst.sal.ESS.relativeHumidity private_seqNum increases 100.0% of the sequence
lsst.sal.ESS.logevent_heartbeat private_seqNum increases 100.0% of the sequence
```

Step LVV-E1457-10 Step Execution Status: **Pass**

Description

Document the procedure including topics chosen, time window, replication reliability and EFD reliability

Expected Result

- A document describing the process including topics and time window.
- The document shall be in the form of a notebook with saved outputs, or similar

Actual Result

Added LVV-T2334.ipynb to the test report repo

Detailed steps results LVV-C181-LVV-T2338 LVV-E3446-3841:
Note: Steps "Not Executed" and with No Result are not shown in this report.

Step LVV-E3446-1 Step Execution Status: **Pass**

Description

Log in to the USDF notebook aspect: https://usdf-rsp-dev.slac.stanford.edu/

Make sure to choose a recent weekly release and a large instance; record the chosen release and whether or not it is the current ”recommended” release.

Expected Result
The JupyterLab interface is displayed in the browser

Actual Result

Step LVV-E3446-2 Step Execution Status: **Pass**

Description
Open a notebook:

1. Navigate to the File->New->Notebook
2. When prompted, select the LSST kernel

Expected Result
An empty notebook running in the LSST kernel

Actual Result
Loaded existing notebook LVVT-2338 from the github repo already on the RSP.

Step LVV-E3446-3 Step Execution Status: **Pass**

Description
Connect to the USDF EFD.
Use the EFD identifier name from the test case parameter, usdf_efd , unless it is necessary to change at the time the test is executed. Record the value used if different.

Example Code
```
from lsst_efd_client import EfdClient
efd = EfdClient('usdf_efd')
```

Expected Result
A notebook with an instance of the ‘EfdClient‘ configured to talk to the USDF EFD

Actual Result

Last executed at 2024-02-08 10:32:00 in 3.71s

Step LVV-E3446-4 Step Execution Status: **Pass**

Description

Since we need to compare results between the summit and the USDF, also create a connection to the summit EFD. Use the EFD identifier name from the test case parameter, summit_efd , unless it is necessary to change at the time the test is executed. Record the value used if different.

Example Code

efd_summit = EfdClient('summit_efd')

Expected Result

Another cell in the notebook with a second EFD connection to the summit EFD

Actual Result

previous cell also does this

Step LVV-E3446-5 Step Execution Status: **Pass**

Description

Choose 5 topics to query and select a 6 day window of data. The window is arbitrary, but must be explicit (not relative to now()) so that it can be reproduced. The topics should be chosen to sample the various topic contexts. I.e. the topics should be chosen to sample both diagnostic topics like heartbeat monitors as well as both high and low cadence telemetry topics to get a broad view on how the system behaves with different kinds of topics.

Expected Result

A list of 5 valid SAL topics to be queried and a time window defined as astropy.Time objects.

Actual Result

these are selected

```
lsst.sal.MTDomeTrajectory.logevent_heartbeat had 30 messages in first 30 min of 2024-01-01
lsst.sal.ATMonochromator.logevent_heartbeat had 29 messages in first 30 min of 2024-01-01
lsst.sal.MTM1M3.logevent_heartbeat had 30 messages in first 30 min of 2024-01-01
```

```
lsst.sal.ESS.lightningStrikeStatus had 30 messages in first 30 min of 2024-01-01
lsst.sal.Watcher.logevent_heartbeat had 30 messages in first 30 min of 2024-01-01
Random selction of five 'summit' topics ['lsst.sal.MTDomeTrajectory.logevent_heartbeat', 'lsst.sal.ATMonochromator.logevent
```

Step LVV-E3446-6 Step Execution Status: **Pass**

Description

Issue selections at both the summit and the data facility. These selections should select all fields for the chosen topics.

Expected Result

A total of 10 pandas.DataFrame objects, 5 each for the summit and replicated EFDs. Each topic requires a separate query, so each will get its own DataFrame. All fields in each topic should be selected.

Actual Result

previous topis have dtat - there are five - next step will retrieve frames and compare.

Step LVV-E3446-7 Step Execution Status: **Pass**

Description

First compare the index for each topic between the summit and replicated EFD. There should be:

1. The same number of samples in each topic for each location
2. Given 1) each time stamp should represent the same time

Reliability of the replication must be at better than 99%. If there are samples missing from the replicated datasets, confirm that the length of the replicated DataFrame divided by the length of the summit DataFrame is greater than 0.99 for all topics.

Expected Result

A cell in the notebook showing the DataFrames are the same length per topic between the summit and the replicated EFD. A cell showing the times in the index are the same for each topic. This could be done by converting

---

to seconds and showing the difference is zero for every sample.

If there are missing samples, the replication should be better than 99%. If it is not, the deviation must be traced to an intervening event or system other than the replication system itself to explain the discrepancy.

Actual Result

Step LVV-E3446-8 Step Execution Status: **Pass**

Description

Compare the fields for each topic between the summit and the replicated EFDs. They should be equivalent to double precision. This can be done by looping over the topics and fields and showing numpy.all (or similar) evaluates to True.

Expected Result

A cell or cells showing that all fields for all topics evaluate as equivalent given appropriate precision.

Actual Result

```
 Checking 2024-01-01
Compare lsst.sal.MTDomeTrajectory.logevent_heartbeat
Compare lsst.sal.ATMonochromator.logevent_heartbeat
Compare lsst.sal.MTM1M3.logevent_heartbeat
Compare lsst.sal.ESS.lightningStrikeStatus
Compare lsst.sal.Watcher.logevent_heartbeat
 Checking 2024-01-02
Compare lsst.sal.MTDomeTrajectory.logevent_heartbeat
Compare lsst.sal.ATMonochromator.logevent_heartbeat
Compare lsst.sal.MTM1M3.logevent_heartbeat
Compare lsst.sal.ESS.lightningStrikeStatus
Compare lsst.sal.Watcher.logevent_heartbeat
 Checking 2024-01-03
Compare lsst.sal.MTDomeTrajectory.logevent_heartbeat
Compare lsst.sal.ATMonochromator.logevent_heartbeat
Compare lsst.sal.MTM1M3.logevent_heartbeat
Compare lsst.sal.ESS.lightningStrikeStatus
Compare lsst.sal.Watcher.logevent_heartbeat
 Checking 2024-01-04
Compare lsst.sal.MTDomeTrajectory.logevent_heartbeat
```

```
Compare lsst.sal.ATMonochromator.logevent_heartbeat
Compare lsst.sal.MTM1M3.logevent_heartbeat
Compare lsst.sal.ESS.lightningStrikeStatus
Compare lsst.sal.Watcher.logevent_heartbeat
 Checking 2024-01-05
Compare lsst.sal.MTDomeTrajectory.logevent_heartbeat
Compare lsst.sal.ATMonochromator.logevent_heartbeat
Compare lsst.sal.MTM1M3.logevent_heartbeat
Compare lsst.sal.ESS.lightningStrikeStatus
Compare lsst.sal.Watcher.logevent_heartbeat
 Checking 2024-01-06
Compare lsst.sal.MTDomeTrajectory.logevent_heartbeat
Compare lsst.sal.ATMonochromator.logevent_heartbeat
Compare lsst.sal.MTM1M3.logevent_heartbeat
Compare lsst.sal.ESS.lightningStrikeStatus
Compare lsst.sal.Watcher.logevent_heartbeat
```

Step LVV-E3446-9 Step Execution Status: **Pass**

Description

Examine the summit messages to confirm that the reliability is better than 99.9% for all topics. Keep in mind that the private_seqNum is intended to be a sequentially increasing index of the messages, but that it gets reset after every CSC reboot. This must be accounted for by applying an offset when a reset is observed.
Show the reliability is better than 99.9% by showing that the private_seqNum is sequential better than 99.9% of the time (when correcte for resets).

Expected Result

A histogram or similar showing that the difference private_seqNum[1:] - private_seqnum[:-1] is 1 more than 99.9% of the time.

Actual Result

```
lsst.sal.MTDomeTrajectory.logevent_heartbeat private_seqNum increases 100.0% of the sequence
lsst.sal.ATMonochromator.logevent_heartbeat private_seqNum increases 100.0% of the sequence
lsst.sal.MTM1M3.logevent_heartbeat private_seqNum increases 100.0% of the sequence
lsst.sal.ESS.lightningStrikeStatus private_seqNum increases 100.0% of the sequence
lsst.sal.Watcher.logevent_heartbeat private_seqNum increases 100.0% of the sequence
```

Step LVV-E3446-10 Step Execution Status: **Pass**

Description

Document the procedure including topics chosen, time window, replication reliability and EFD reliability

Expected Result

- A document describing the process including topics and time window.
- The document shall be in the form of a notebook with saved outputs, or similar

Actual Result

The notebook LVV-T2338 does this.

5.1.3.2 LVV-T2339 - Archival EFD products can be used for historical analysis

Version **3**. Status **Approved**. Open *LVV-T2339* test case in Jira.

Show that the archival replicated EFD products (nominally Parquet files) can be used to query for topics post facto and that analysis in the Notebook Aspect of the RSP is not only possible but straightforward.

This test case provides partial coverage of the requirement DMS-NB-REQ-0023, Access to All Data Products: ”An authorized user of the Notebook Aspect shall be able to access the Transformed Engineering and Facilities Database (EFD) and and all other LSST released data products.”, as adapted to the current design for EFD replication and access (see DMTN-082). Note that it is still also anticipated that TAP access to the EFD will be provided in the RSP as an alternative to the Python-API access covered by this test case. Note in particular that this test case does not cover the feature of the ”Transformed” EFD from the original design in which EFD data is pre-processed with associations to exposure/visit IDs.

Preconditions:

Execution status:

Final comment:

Detailed steps results LVV-C181-LVV-T2339 LVV-E1552-1940:
Note: Steps "Not Executed" and with No Result are not shown in this report.

Step LVV-E1552-1 Step Execution Status: **Pass**

Description
Log in to the USDF notebook aspect: https://usdf-rsp-dev.slac.stanford.edu/
Make sure to choose a recent weekly release and a large instance; record the chosen release and whether or not it is the current "recommended" release.

Expected Result
The JupyterLab interface is displayed in the browser

Actual Result
logged in have RSP at USDF

Step LVV-E1552-2 Step Execution Status: **Pass**

Description
Open a notebook:

1. Navigate to the File->New->Notebook
2. When prompted, select the LSST kernel

Expected Result
An empty notebook running in the LSST kernel

Actual Result
Copied LVV-T2338.ipynb to LVV-TT339.ipynb

Step LVV-E1552-3 Step Execution Status: **Pass**

Description
Connect to the USDF EFD.
Use the EFD identifier name from the test case parameter, usdf_efd , unless it is necessary to change at the time

the test is executed. Record the value used if different.

Example Code

from lsst_efd_client import EfdClient
efd = EfdClient('{EFD_ID}')

Expected Result

A notebook with an instance of the 'EfdClient' configured to talk to the USDF EFD

Actual Result

Last executed at 2023-06-16 19:50:41 in 3.71s

Step LVV-E1552-4 Step Execution Status: **Not Executed**

Description

Choose 3 topics to query and select a 6 day window of data. The window is arbitrary, but must be explicit (not relative to now()) so that it can be reproduced. The topics are also arbitrary but will sample multiple subsystems.

Expected Result

A cell in a notebook specifying topic names and start and end times for a 6 day window as astropy.Time objects.

Actual Result

```
lsst.sal.MTAirCompressor.logevent_summaryState had 22 messages between 2023-06-13 and 2023-06-20
lsst.sal.MTM1M3.logevent_raisingLoweringInfo had 6582 messages between 2023-06-13 and 2023-06-20
Random selction of 3 'usdf' topics ['lsst.sal.ATPneumatics.logevent_heartbeat', 'lsst.sal.MTAirCompressor.logevent_summaryS
```

Step LVV-E1552-5 Step Execution Status: **Pass**

Description

Use the defined topics and the time window to read in fields from the chosen topics. These are expected to be pandas.DataFrames.

Expected Result

A cell in a notebook with 1 DataFrame per topic. This shall be explicit about all the necessary steps and any hand tuning necessary to make it possible to make selections from the archival data.

Actual Result

```
lsst.sal.ATPneumatics.logevent_heartbeat has fields:Index(['heartbeat', 'private_efdStamp', 'private_identity',
       'private_kafkaStamp', 'private_origin', 'private_rcvStamp',
       'private_revCode', 'private_seqNum', 'private_sndStamp'],
      dtype='object')
lsst.sal.MTAirCompressor.logevent_summaryState has fields:Index(['private_efdStamp', 'private_identity', 'private_kafkaSta
       'private_origin', 'private_rcvStamp', 'private_revCode',
       'private_seqNum', 'private_sndStamp', 'salIndex', 'summaryState'],
      dtype='object')
lsst.sal.MTM1M3.logevent_raisingLoweringInfo has fields:Index(['private_efdStamp', 'private_identity', 'private_kafkaStamp
       'private_origin', 'private_rcvStamp', 'private_revCode',
       'private_seqNum', 'private_sndStamp', 'waitAirPressure',
       'waitHardpoint0',
       ...
       'waitZForceActuator91', 'waitZForceActuator92', 'waitZForceActuator93',
       'waitZForceActuator94', 'waitZForceActuator95', 'waitZForceActuator96',
       'waitZForceActuator97', 'waitZForceActuator98', 'waitZForceActuator99',
       'weightSupportedPercent'],
      dtype='object', length=284)
```

Step LVV-E1552-6 Step Execution Status: **Pass**

Description

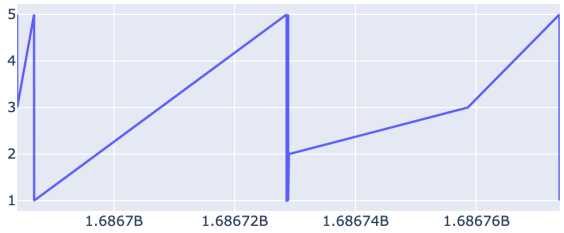
Execute various analysis tasks, e.g. plotting various quantities, to show that it is practical to do so.

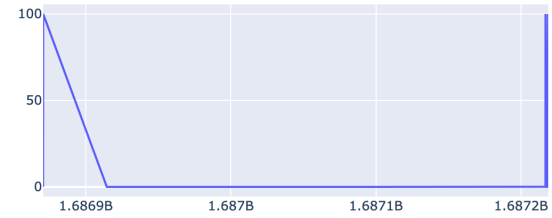
Expected Result

Plots of quantities including filtering and other common activities. This shall specifically identify cases where there is an obvious difference between how the interaction would be done with the running EFD vs the archival version.

Actual Result

not so interesting plots like this





Step LVV-E1552-7 Step Execution Status: **Pass**

Description

Document the procedure including topics and time window. Particular attention shall be paid to differences in interactions between the running EFD and the archival version. If any topics chosen for this test were downsampled in the archiving process, the document shall report on the self consistency of the down/re-sampling of those topics.

Expected Result

- A document describing the process including the topics chosen and the time window.
- The document shall be in the form of a notebook with saved outputs.

Actual Result

LVV-T2339.ipynb added to the DMTR-331 repo - also renderd at the end fo the test report (\secref{sec:LVV-T2339nb})

A Documentation

The verification process is defined in LSE-160. The use of Docsteady to format Jira information in various test and planing documents is described in DMTN-140 and practical commands are given in DMTN-178.

B Acronyms used in this document

Acronym	Description
API	Application Programming Interface
CSC	Commandable SAL Component
DMS	Data Management Subsystem
DMS-REQ	Data Management System Requirements prefix
DMTN	DM Technical Note
DMTR	DM Test Report
EFD	Engineering and Facility Database
GIS	Global Interlock System
LDM	LSST Data Management (Document Handle)
LSE	LSST Systems Engineering (Document Handle)
LSST	Legacy Survey of Space and Time (formerly Large Synoptic Survey Telescope)
LVV	LSST Verification and Validation
MTM1M3	Main Telescope M1M3
PMCS	Project Management Controls System
RSP	Rubin Science Platform
SAL	Service Abstraction Layer
TAP	Table Access Protocol (IVOA standard)
USDF	United States Data Facility
VE	vendor estimate

C Traceability

Test Case	VE Key	VE Summary
LVV-T2338	LVV-71	DMS-REQ-0168-V-01: Summit Facility Data Communications
LVV-T2339	LVV-9979	DMS-NB-REQ-0023-V-01: Access to All Data Products_1

D LVV-T2338 test Notebook

LVV-T2338

February 9, 2024

1 For LDM-503-EFDb

Initialize EFDs Summit and USDF

[26]:
```
from lsst_efd_client import EfdClient, resample

client = EfdClient('summit_efd')
client.output = 'dataframe'
cl=client.influx_client



#usdf_client = EfdClient('summit_efd_copy')
usdf_client = EfdClient('usdf_efd')
usdf_client.output = 'dataframe'
usdf_cl=usdf_client.influx_client
```

1.1 Pick five topics

[27]:
```
import random
async def selectTopics(pick):
    topics = await client.get_topics()

    selected_topics = []
    loc = "summit"

    day = '2024-01-01'


    # want to select topics randomly but with messages so randomize all indexes
    randoms = random.sample(range(0,len(topics)),len(topics))
    for r in randoms:
       topic = topics[r]
       if ('Test' in topic):
           next
       result = []
       if len(selected_topics) < 5:
```

```
            query=f'''SELECT * FROM "{topic}" WHERE time >  '{day}T00:00:00.
↪000Z' and time < '{day}T00:00:30.000Z' '''
            result = await cl.query(query)

        if len(result) > 20:
            print (f"{topic} had {len(result)} messages in first 30 min of␣
↪{day}")
            selected_topics.append(topic)
            if len(selected_topics) > 4:
              break
    print(f"Random selction of five '{loc}' topics {selected_topics} with␣
↪messages on {day}")
    return selected_topics
```

1.2 Utility function to compare reusults

[33]:
```
def cmp(topic, result, sresult):
    print (f"Compare {topic}")
    problems = 0
    if len(result) != len(sresult):
        print (f"\033[91m  {topic} does not have the same number of results␣
↪{len(result)}  - summit had {len(sresult)}\033[0m")
        problems = problems + 1
    else:
        for k in sresult.keys():
            if result[k].all() != sresult[k].all():
                print (f"\033[91m {topic} {key} does not match \033[0m")
                problems = problems + 1
```

[30]:
```
selected_topics = await selectTopics(5) # do this once at least
```

```
lsst.sal.MTDomeTrajectory.logevent_heartbeat had 30 messages in first 30 min of
2024-01-01
lsst.sal.ATMonochromator.logevent_heartbeat had 29 messages in first 30 min of
2024-01-01
lsst.sal.MTM1M3.logevent_heartbeat had 30 messages in first 30 min of 2024-01-01
lsst.sal.ESS.lightningStrikeStatus had 30 messages in first 30 min of 2024-01-01
lsst.sal.Watcher.logevent_heartbeat had 30 messages in first 30 min of
2024-01-01
Random selction of five 'summit' topics
['lsst.sal.MTDomeTrajectory.logevent_heartbeat',
'lsst.sal.ATMonochromator.logevent_heartbeat',
'lsst.sal.MTM1M3.logevent_heartbeat', 'lsst.sal.ESS.lightningStrikeStatus',
'lsst.sal.Watcher.logevent_heartbeat'] with messages on 2024-01-01
```

1.3 Get dataframes and compare summit and USDF

Doing this day by day

[34]:
```
# test gives large window 6 days

# May topics

data = {}
for d in range(1,7):
    if d < 10:
       day = f"2024-01-0{d}"
    else:
       day = f"2024-01-{d}"

    day2 = day
    print(f" Checking {day}")
    for topic in selected_topics:
        query=f'''SELECT * FROM "{topic}" WHERE time >  '{day}T00:00:00.000Z'␣
 ↪and time < '{day2}T23:59:59.000Z' limit 1300000  '''
        # could GROUP BY time(1h) maybe ??
        result = await cl.query(query)
        usdf_result = await usdf_cl.query(query)
        cmp(topic, usdf_result, result)
        data[topic] = usdf_result
```

```
 Checking 2024-01-01
Compare lsst.sal.MTDomeTrajectory.logevent_heartbeat
Compare lsst.sal.ATMonochromator.logevent_heartbeat
Compare lsst.sal.MTM1M3.logevent_heartbeat
Compare lsst.sal.ESS.lightningStrikeStatus
Compare lsst.sal.Watcher.logevent_heartbeat
 Checking 2024-01-02
Compare lsst.sal.MTDomeTrajectory.logevent_heartbeat
Compare lsst.sal.ATMonochromator.logevent_heartbeat
Compare lsst.sal.MTM1M3.logevent_heartbeat
Compare lsst.sal.ESS.lightningStrikeStatus
Compare lsst.sal.Watcher.logevent_heartbeat
 Checking 2024-01-03
Compare lsst.sal.MTDomeTrajectory.logevent_heartbeat
Compare lsst.sal.ATMonochromator.logevent_heartbeat
Compare lsst.sal.MTM1M3.logevent_heartbeat
Compare lsst.sal.ESS.lightningStrikeStatus
Compare lsst.sal.Watcher.logevent_heartbeat
 Checking 2024-01-04
Compare lsst.sal.MTDomeTrajectory.logevent_heartbeat
Compare lsst.sal.ATMonochromator.logevent_heartbeat
Compare lsst.sal.MTM1M3.logevent_heartbeat
```

```
Compare lsst.sal.ESS.lightningStrikeStatus
Compare lsst.sal.Watcher.logevent_heartbeat
 Checking 2024-01-05
Compare lsst.sal.MTDomeTrajectory.logevent_heartbeat
Compare lsst.sal.ATMonochromator.logevent_heartbeat
Compare lsst.sal.MTM1M3.logevent_heartbeat
Compare lsst.sal.ESS.lightningStrikeStatus
Compare lsst.sal.Watcher.logevent_heartbeat
 Checking 2024-01-06
Compare lsst.sal.MTDomeTrajectory.logevent_heartbeat
Compare lsst.sal.ATMonochromator.logevent_heartbeat
Compare lsst.sal.MTM1M3.logevent_heartbeat
Compare lsst.sal.ESS.lightningStrikeStatus
Compare lsst.sal.Watcher.logevent_heartbeat
```

[32]:
```
## Reliability
```

[35]:
```
for topic in selected_topics:
    seqnum = data[topic]['private_seqNum']
    count = 0
    i = 1 # see if the sequence increases mostly
    prev = seqnum[0]
    while i < len(seqnum):
        if (seqnum[i] < prev):
            #print (f"Reset at {i} seqnum : {prev}, {seqnum[i]}")
            count = count + 1
        i = i + 1
    percent = 100 * ((len(seqnum) - count) / len(seqnum))
    print(f"{topic} private_seqNum increases {percent}% of the sequence")
```

```
/tmp/ipykernel_713/336143790.py:5: FutureWarning: Series.__getitem__ treating
keys as positions is deprecated. In a future version, integer keys will always
be treated as labels (consistent with DataFrame behavior). To access a value by
position, use `ser.iloc[pos]`
  prev = seqnum[0]
/tmp/ipykernel_713/336143790.py:7: FutureWarning: Series.__getitem__ treating
keys as positions is deprecated. In a future version, integer keys will always
be treated as labels (consistent with DataFrame behavior). To access a value by
position, use `ser.iloc[pos]`
  if (seqnum[i] < prev):

lsst.sal.MTDomeTrajectory.logevent_heartbeat private_seqNum increases 100.0% of
the sequence
lsst.sal.ATMonochromator.logevent_heartbeat private_seqNum increases 100.0% of
the sequence
lsst.sal.MTM1M3.logevent_heartbeat private_seqNum increases 100.0% of the
sequence
lsst.sal.ESS.lightningStrikeStatus private_seqNum increases 100.0% of the
```

```
sequence
lsst.sal.Watcher.logevent_heartbeat private_seqNum increases 100.0% of the
sequence
```

E LVV-T2339 test Notebook

LVV-T2339

February 9, 2024

1 For LDM-503-EFDb

Initialize EFD at USDF

[6]:
```
from lsst_efd_client import EfdClient, resample

client = EfdClient('usdf_efd')
client.output = 'dataframe'
cl=usdf_client.influx_client
```

[ ]:
```
## Pick 3  topics
```

[11]:
```
import random
topics = await client.get_topics()

selected_topics = []
results = {}
loc = "usdf"
pick = 3
day = '2023-06-13'
day2 = '2023-06-20'

# want to selct 5 topics randomly but with messages so randomize all indexes
randoms = random.sample(range(0,len(topics)),len(topics))
for r in randoms:
   topic = topics[r]
   result = []
   if len(selected_topics) < pick:
      query=f'''SELECT * FROM "{topic}" WHERE time >  '{day}T00:00:00.000Z' and␣
 ↪time < '{day2}T00:00:00.000Z' '''
      result = await cl.query(query)

   if len(result) > 20:
        print (f"{topic} had {len(result)} messages between {day} and {day2} ")
        selected_topics.append(topic)
        results[topic] = result
        if len(selected_topics) > (pick -1):
            break
```

```
print(f"Random selction of {pick} '{loc}' topics {selected_topics} with␣
  ↪messages between {day} and {day2}")
```

```
lsst.sal.ATPneumatics.logevent_heartbeat had 529266 messages between 2023-06-13
and 2023-06-20
lsst.sal.MTAirCompressor.logevent_summaryState had 22 messages between
2023-06-13 and 2023-06-20
lsst.sal.MTM1M3.logevent_raisingLoweringInfo had 6582 messages between
2023-06-13 and 2023-06-20
Random selction of 3 'usdf' topics ['lsst.sal.ATPneumatics.logevent_heartbeat',
'lsst.sal.MTAirCompressor.logevent_summaryState',
'lsst.sal.MTM1M3.logevent_raisingLoweringInfo'] with messages between 2023-06-13
and 2023-06-20
```

[ ]:
```
## read fields ..
```

[13]:
```
for topic in selected_topics:
   result = results[topic]
   print (f"{topic} has fields:{result.columns}")
```

```
lsst.sal.ATPneumatics.logevent_heartbeat has fields:Index(['heartbeat',
'private_efdStamp', 'private_identity',
       'private_kafkaStamp', 'private_origin', 'private_rcvStamp',
       'private_revCode', 'private_seqNum', 'private_sndStamp'],
      dtype='object')
lsst.sal.MTAirCompressor.logevent_summaryState has
fields:Index(['private_efdStamp', 'private_identity', 'private_kafkaStamp',
       'private_origin', 'private_rcvStamp', 'private_revCode',
       'private_seqNum', 'private_sndStamp', 'salIndex', 'summaryState'],
      dtype='object')
lsst.sal.MTM1M3.logevent_raisingLoweringInfo has
fields:Index(['private_efdStamp', 'private_identity', 'private_kafkaStamp',
       'private_origin', 'private_rcvStamp', 'private_revCode',
       'private_seqNum', 'private_sndStamp', 'waitAirPressure',
       'waitHardpoint0',
       …
       'waitZForceActuator91', 'waitZForceActuator92', 'waitZForceActuator93',
       'waitZForceActuator94', 'waitZForceActuator95', 'waitZForceActuator96',
       'waitZForceActuator97', 'waitZForceActuator98', 'waitZForceActuator99',
       'weightSupportedPercent'],
      dtype='object', length=284)
```

[ ]:
```
## plots ..
```

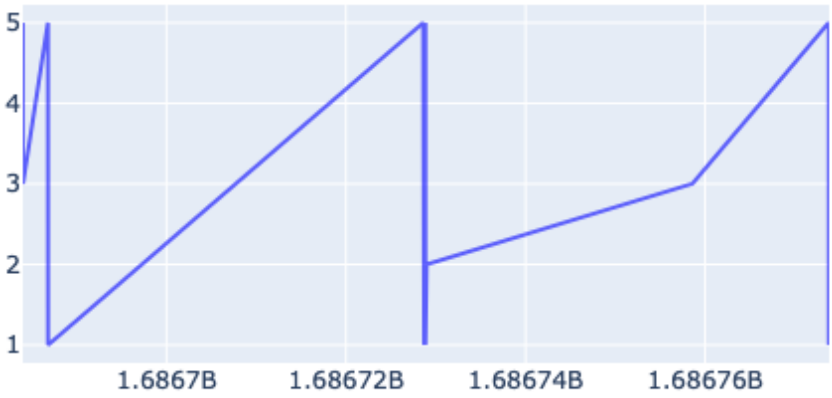
[29]:
```
def plot(topic, results, field):
    df = results[topic]
    bins = (6*24)
    timestep = np.arange(0, bins, 1)
    dates = pd.date_range(start=day, end=day2)
    fig[topic] = go.Figure([go.Scatter(x=df['private_efdStamp'], y=df[field],
                            marker_color='blue',
                            opacity=0.6,
                            name=field)])

    fig[topic].show()
```

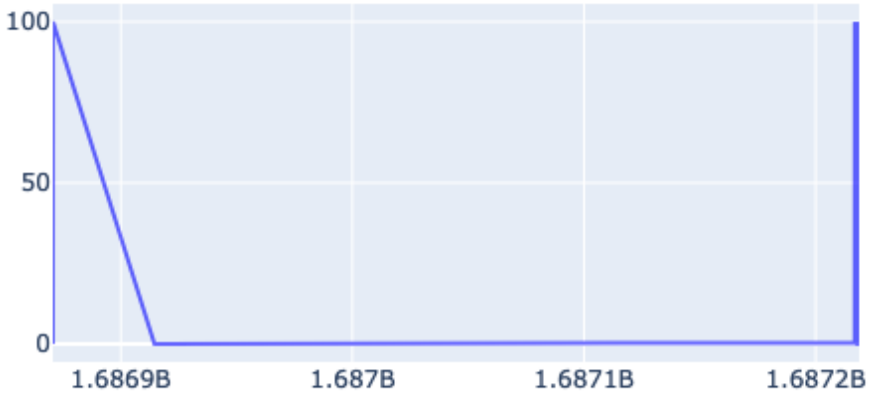
[ ]:
```
import plotly.graph_objects as go
import pandas as pd
import numpy as np

plot('lsst.sal.ATPneumatics.logevent_heartbeat', results, 'heartbeat')
```

[31]:
```
plot('lsst.sal.MTAirCompressor.logevent_summaryState', results, 'summaryState')
```



[30]:
```
plot('lsst.sal.MTM1M3.logevent_raisingLoweringInfo', results,␣
↪'weightSupportedPercent')
```


100
50
0
1.6869B
1.687B
1.6871B
1.6872B