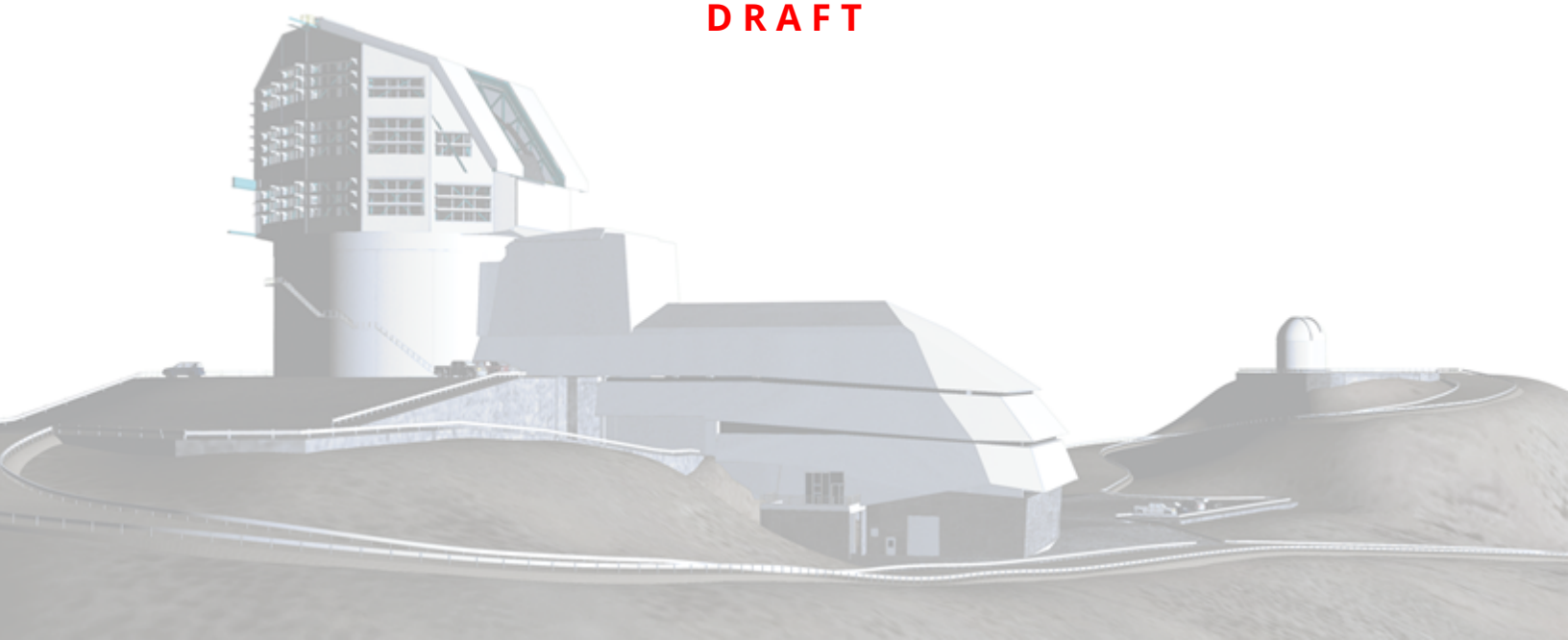

Vera C. Rubin Observatory
Software Test Report

DM-503-EFDa: EFD on Summit for M1/M3 Test Plan and Report

Simon Krughoff

DMTR-271

Latest Revision: 2021-05-11

DRAFT

Abstract

This is the test plan and report for **EFD on Summit for M1/M3** (DM-503-EFDa), an LSST milestone pertaining to the Data Management Subsystem.
This document is based on content automatically extracted from the Jira test database on 2021-05-11 . The most recent change to the document repository was on 2021-05-11.

Change Record

Version	Date	Description	Owner name
	2020-10-30	First draft	Robert Gruendl

Document curator: Robert Gruendl
Document source location: `https://github.com/lsst-dm/DMTR-271`
Version from source repository: 52a026d

Contents

DM-503-EFDa: EFD on Summit for M1/M3 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 Engineering Facility Database (EFD) as implemented with Kafka, InfluxDB and Chronograf. This plan will describe the prerequisites for beginning a test campaign, step by step instructions for each test can 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 XX% of produced messages are recorded.

The highest level description of this test plan is to run the M1/M3 subsystem in an active state for no less than 5 contiguous days. During this time, all telemetry produced by the M1/M3 subsystem will appear in the InfluxDB instance running at the summit with latency less than XX second XX%. The maximum latency shall be less than XX seconds. All telemetry shall also be available for interrogation by Chronograf on similar time scales. Any gap in telemetry or dropped/missing messages will be considered a deviation. Successful completion of the test campaign will show that:

1. users are able to access data ingested in the InfluxDB at the summit in near real time from the Chronograf interface
2. the latency from message production to ingestion in InfluxDB is less than the nominal limit (XX sec) XX% of the time and never more than XX seconds
3. the M1/M3 telemetry is successfully being mirrored to another influxDB instance at the data facility with latency less than XX hour(s)

4. users are able to access and analyze telemetry data from the M1/M3 subsystem from notebooks running in the notebook aspect of the RSP both at the summit and at the data facility

1.2 System Overview

The tests will be carried out from within an instance of the notebook aspect of the RSP running at either the summit or the data facility. 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-P78 Jira Test Plan and related Test Cycles (LVV-C163).

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-P78 in Jira is **Draft** .

1.4 References

[1] **[DMTN-178]**, Comoretto, G., 2021, *Docsteady Usecases for Rubin Observatory Construction*, DMTN-178, URL http://DMTN-178.lsst.io

[2] **[DMTN-140]**, Comoretto, G., Guy, L.P., et al., 2020, *Documentation Automation for the Verification and Validation of Rubin Observatory Software*, DMTN-140, URL https://dmtn-140.lsst.io/

[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 environ

2.3 Entry Criteria

1. Before beginning this test, as 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 XX%
   (b) The typical latency of the system is less than XX sec for a pre defined set of topics
   (c) The summit data is being replicated to the instance at NCSA
   (d) Chronograf is set up and running at both the summit and NCSA
2. The summit network and Kubernetes cluster are performing nominally
3. The EFD is deployed in the summit Kubernetes cluster
4. The M1M3 sub-component is reliably producing telemetry via Kafka producers with correct versions of the schema
5. The notebook aspect of the RSP is deployed in the summit Kubernetes cluster
6. The summit EFD is reliably replicated to an EFD instance running in a data facility
7. The notebook aspect of the RSP is deployed in the same data facility as that running the replicated EFD
8. The most recent version of the EFD client python modules are installed in the various deployed notebook aspects
9. A requirement test that the system demonstrates reliability of greater than XX%

2.4 Related Documentation

No additional documentation provided.

2.5 PMCS Activity

Primavera milestones related to the test campaign:

- DM-503-EFDa

3 Personnel

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

	T. Plan LVV-P78 owner:	**Simon Krughoff**	
	T. Cycle LVV-C163 owner:	**Simon Krughoff**	
Test Cases	**Assigned to**	**Executed by**	**Additional Test Personnel**
LVV-T2111	Simon Krughoff		
LVV-T2112	Simon Krughoff		
LVV-T2117	Simon Krughoff		
LVV-T2116	Simon Krughoff		
LVV-T2115	Simon Krughoff		

4 Test Campaign Overview

4.1 Summary

T. Plan LVV-P78:	**DM-503-EFDa: EFD on Summit for M1/M3**			Draft
T. Cycle LVV-C163:	**DM-503-EFDa: EFD on Summit for M1/M3**			Not Executed
Test Cases	**Ver.**	**Status**	**Comment**	**Issues**
LVV-T2111	1	Not Executed		
LVV-T2112	1	Not Executed		
LVV-T2117	1	Not Executed		
LVV-T2116	1	Not Executed		
LVV-T2115	1	Not Executed		

Table 2: Test Campaign Summary

4.2 Overall Assessment

Not yet available.

4.3 Recommended Improvements

Not yet available.

5 Detailed Test Results

5.1 Test Cycle LVV-C163

Open test cycle *DM-503-EFDa: EFD on Summit for M1/M3* in Jira.

Test Cycle name: DM-503-EFDa: EFD on Summit for M1/M3
Status: Not Executed

5.1.1 Software Version/Baseline

Not provided.

5.1.2 Configuration

Not provided.

5.1.3 Test Cases in LVV-C163 Test Cycle

5.1.3.1 LVV-T2111 - Access to M1/M3 telemetry data in near real time via the Chronograf interface

Version **1**. Open *LVV-T2111* test case in Jira.

Show that users can get access to visualizations in Chronograf with telemetry arriving in less than XX seconds from a command run on the DDS network.

Preconditions:
See prerequisites in the Test Plan LVV-P78

Execution status: **Not Executed**

Final comment:

Detailed steps results:

Step 1 Step Execution Status: **Not Executed**

Description

Log in to whatever VPNs are necessary to both see Chronograf at the summit and the control network necessary for commanding components fo the M1/M3 subsystem

Expected Result

VPN connects are live

Actual Result

Step 2 Step Execution Status: **Not Executed**

Description

Log in to chronograf running at the summit. The endpoint is currently https://chronograf-summit-efd.lsst.codes, though https://sqr-034.lsst.io shall be considered the primary source of truth for service endpoints relating to the EFD

Expected Result

The browser showing the front page of the chronograf interface

Actual Result

Step 3 Step Execution Status: **Not Executed**

Description

Using the "Explore" tab in chronograph enter a query that populates the graph window for one value: e.g. lsst.sal.MTM1M3.forceActu
This may change depending on what command is given to the subsystem

Expected Result

There should be a graph showing a trace of the chosen field over the default time window (now() - 1hr)

Actual Result

Step 4 Step Execution Status: **Not Executed**

Description

Set the refresh in chronograph to the minimum value (currently 5 seconds) and adjust the time window to show the past 15 minutes of data. Both of these operations are done using buttons immediately above the graph window on the right side

Expected Result

- The trace is now the last 15 minutes and the view refreshes every 5 seconds
- The query used to produce the trace plot will be captured as an artifact

Actual Result

Step 5 Step Execution Status: **Not Executed**

Description

Issue a command to the M1/M3 subsystem to e.g. modify the force actuators and confirm the message was successfully sent and acted upon

Expected Result

- Confirmation that the command was acted upon
- This will also include a qualitative description of whether the command produced the expected change in telemetry.

Actual Result

Step 6 Step Execution Status: **Not Executed**

Description

Verify new values are showing up in the summit instance of InfluxDB by observing that values with newer timestamps are appearing in the Chronograf visualization

Expected Result
Screenshots or similar documenting dashboards before and after the command to M1/M3 is made.

Actual Result

Step 7 Step Execution Status: **Not Executed**

Description
Produce a report for the test

Expected Result

- A document explaining the procedure including the topics monitored, the command issued and the query used to do the visualization

Actual Result

5.1.3.2 LVV-T2112 - Latency from production to ingestion and telemetry can be analyzed via notebooks: High and Low Cadence

Version **1**. Open *LVV-T2112* test case in Jira.

Measure the latency of production to ingestion in the EFD and show it is less than XX seconds XX% of the time.
Show that this analysis can be completed via a notebook running in an instance of the notebook aspect of the RSP.

Preconditions:
See prerequisites in the Test Plan LVV-P78

Execution status: **Not Executed**

Final comment:

Detailed steps results:

Step 1 Step Execution Status: **Not Executed**

Description

Log in to whatever VPNs are necessary to access to the summit notebook aspect of the RSP

Expected Result

VPN connection is active

Actual Result

Step 2 Step Execution Status: **Not Executed**

Description

Log in to the summit notebook aspect: https://summit-lsp.lsst.codes/nb

Make sure to choose a recent weekly and a large instance

Expected Result

The JupyterLab interface is displayed in the browser

Actual Result

Step 3 Step Execution Status: **Not Executed**

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

Step 4 Step Execution Status: **Not Executed**

Description

Connect to the summit EFD

Example Code

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

Expected Result

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

Actual Result

Step 5 Step Execution Status: **Not Executed**

Description

Choose a topic to query and select a 5 day window of data. The topic and window are arbitrary, but it shall be explicit (not relative to now()) so that it can reproduced. The topic shall be one of high enough cadence that there are many measurements for the topic, i.e. not a command or log topic that could potentially be very sparsely populated over the 5 day window. A high cadence, ideally greater than XX Hz, topic will give the most precise measurement of the distribution of the latency.

Expected Result

- A table-like object in memory containing data from the chosen topic and time window.
- The window and topic are artifacts to be preserved

Actual Result

Step 6 Step Execution Status: **Not Executed**

Description

1. The total latency is the time from the message being published, private_sndStamp, and when it is ingested in the influxDB database. Currently the index timestamp is private_sndStamp There will be an additional field added to the measurement to reflect the ingest timestamp. This may be technically difficult, in which case a subset of topics will have an extra column added by hand for the purpose of this analysis. Care shall also be taken to ensure the messages are in the same (TAI) time system. In the past some CSCs have been reporting TAI and some report UTC. Currently, the difference is 37 seconds.
2. Compute the total latency by taking the difference of the two columns, arr['timestamp'] - arr['private_sndStamp'] (this is where correction for TAI/UTC would be included if necessary). The result shall be in seconds.

Expected Result

An array-like object in memory containing the latency in seconds for every message in the window

Actual Result

Step 7 Step Execution Status: **Not Executed**

Description

Count the number of entries less than XX second(s) and divide that by the total number of entries. This value shall be greater than or equal to XX.

Expected Result

- A plot showing a histogram of the latency values indicating the XX% value.
- If the XX% latency is greater than XX sec, an explanation shall be supplied describing why the latency is higher than expected more often than expected
- The plot shall also indicate the maximum latency observed
- If the maximum latency is above the nominal maximum (XX sec), an explanation shall be provided

Actual Result

Step 8 Step Execution Status: **Not Executed**

Description

Choose a topic to query and select a 5 day window of data. The topic and window are arbitrary, but it shall be explicit (not relative to now()) so that it can reproduced. The topic shall be low latency in order to to measure the impact of publishing low and high latency topics simultaneously

Expected Result

- A table-like object in memory containing data from the chosen topic and time window.
- The window and topic are artifacts to be preserved

Actual Result

Step 9 Step Execution Status: **Not Executed**

Description

1. The total latency is the time from the message being published, private_sndStamp, and when it is ingested in the influxDB database. Currently the index timestamp is private_sndStamp There will be an additional field added to the measurement to reflect the ingest timestamp. This may be technically difficult, in which case a subset of topics will have an extra column added by hand for the purpose of this analysis. Care shall also be taken to ensure the messages are in the same (TAI) time system. In the past some CSCs have been reporting TAI and some report UTC. Currently, the difference is 37 seconds.
2. Compute the total latency by taking the difference of the two columns, arr['timestamp'] - arr['private_sndStamp'] (this is where correction for TAI/UTC would be included if necessary). The result shall be in seconds.

Expected Result

An array-like object in memory containing the latency in seconds for every message in the window

Actual Result

Step 10 Step Execution Status: **Not Executed**

Description

Count the number of entries less than XX second(s) and divide that by the total number of entries. This value shall be greater than or equal to XX. This is intended to show that even when publishing simultaneously with high and low cadence topics we still meet the latency goal for both topics

Expected Result

- A plot showing a histogram of the latency values indicating the XX% value.
- If the XX% latency is greater than XX sec, an explanation shall be supplied describing why the latency is higher than expected more often than expected
- The plot shall also indicate the maximum latency observed
- If the maximum latency is above the nominal maximum (XX sec), an explanation shall be provided

Actual Result

Step 11 Step Execution Status: **Not Executed**

Description

Document the procedure including latency distributions, time window, and both high and low cadence topics

Expected Result

- A document describing the process including the topic chosen and the time window.
- The document shall be in the form on a notebook with saved outputs, or an instance of an nbreport.

Actual Result

5.1.3.3 LVV-T2117 - Latency from production to ingestion and telemetry can be analyzed via notebooks: Low Cadence

Version **1**. Open *LVV-T2117* test case in Jira.

Measure the latency of production to ingestion in the EFD and show it is less than XX seconds XX% of the time.
Show that this analysis can be completed via a notebook running in an instance of the note-

book aspect of the RSP.

Preconditions:
See prerequisites in the Test Plan LVV-P78

Execution status: **Not Executed**

Final comment:

Detailed steps results:

Step 1 Step Execution Status: **Not Executed**

Description
Log in to whatever VPNs are necessary to access to the summit notebook aspect of the RSP

Expected Result
VPN connection is active

Actual Result

Step 2 Step Execution Status: **Not Executed**

Description
Log in to the summit notebook aspect: https://summit-lsp.lsst.codes/nb
Make sure to choose a recent weekly and a large instance

Expected Result
The JupyterLab interface is displayed in the browser

Actual Result

Step 3 Step Execution Status: **Not Executed**

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

Step 4 Step Execution Status: **Not Executed**

Description

Connect to the summit EFD

Example Code

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

Expected Result

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

Actual Result

Step 5 Step Execution Status: **Not Executed**

Description

Choose a topic to query and select a 5 day window of data. The topic and window are somewhat arbitrary, but it shall be explicit (not relative to now()) so that it can reproduced. The topic shall be a low cadence topic with reasonably even sampling. I.e. not a command or event topic that could be very sparse and unevenly sampled. The window should be chosen specifically to be during a period where high cadence topics are not publishing, so as to measure the latency of the low cadence topics on a quiet network

Expected Result

- A table-like object in memory containing data from the chosen topic and time window.
- The window and topic are artifacts to be preserved

Actual Result

Step 6 Step Execution Status: **Not Executed**

Description

1. The total latency is the time from the message being published, private_sndStamp, and when it is ingested in the influxDB database. Currently the index timestamp is private_sndStamp There will be an additional field added to the measurement to reflect the ingest timestamp. This may be technically difficult, in which case a subset of topics will have an extra column added by hand for the purpose of this analysis. Care shall also be taken to ensure the messages are in the same (TAI) time system. In the past some CSCs have been reporting TAI and some report UTC. Currently, the difference is 37 seconds.
2. Compute the total latency by taking the difference of the two columns, arr['timestamp'] - arr['private_sndStamp'] (this is where correction for TAI/UTC would be included if necessary). The result shall be in seconds.

Expected Result

An array-like object in memory containing the latency in seconds for every message in the window

Actual Result

Step 7 Step Execution Status: **Not Executed**

Description

Count the number of entries less than XX second(s) and divide that by the total number of entries. This value shall be greater than or equal to XX.

Expected Result

- A plot showing a histogram of the latency values indicating the XX% value.
- If the XX% latency is greater than XX sec, an explanation shall be supplied describing why the latency is higher than expected more often than expected
- The plot shall also indicate the maximum latency observed

- If the maximum latency is above the nominal maximum (XX sec), an explanation shall be provided

Actual Result

Step 8 Step Execution Status: **Not Executed**

Description

Document the procedure including latency distributions, time window, and topics

Expected Result

- A document describing the process including the topic chosen and the time window.
- The document shall be in the form on a notebook with saved outputs, or an instance of an nbreport.

Actual Result

5.1.3.4 LVV-T2116 - Verify telemetry is uninterrupted for 5 days and can be analyzed via a notebook at NCSA

Version **1**. Open *LVV-T2116* test case in Jira.

Test that telemetry is being recorded without missing messages for 5 days. This analysis is to be carried out using NCSA infrastructure.

Preconditions:

Execution status: **Not Executed**

Final comment:

Detailed steps results:

Step 1 Step Execution Status: **Not Executed**

Description

Log in to whatever VPNs are necessary to access to the NCSA notebook aspect of the RSP

Expected Result

VPN connection is active

Actual Result

Step 2 Step Execution Status: **Not Executed**

Description

Log in to the NCSA notebook aspect: https://lsst-lsp-stable.ncsa.illinois.edu/nb/

Make sure to choose a recent weekly and a large instance

Expected Result

The JupyterLab interface is displayed in the browser

Actual Result

Step 3 Step Execution Status: **Not Executed**

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

Step 4 Step Execution Status: **Not Executed**

Description

Connect to the NCSA EFD

Note:The efd identifier is yet to be determined, but shall be ldf_efd or similar.

Example Code

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

Expected Result

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

Actual Result

Step 5 Step Execution Status: **Not Executed**

Description

Choose a topic to query and select a 5 day window of data. The window is arbitrary, but must be explicit (not relative to now()) so that it can be reproduced. Note that command topics shall be avoided for this purpose since the semantics of private_seqNum are different in the context of commands than for other topic types

Expected Result

A time window and topic name that will be queried for the sequence number

Actual Result

Step 6 Step Execution Status: **Not Executed**

Description

Care shall be taken to fix up the id column, private_seqNum. It must be monotonically increasing, but gets reset when the CSC is restarted. When this happens an offset must be applied to the rest of the message ids in order to produce a monotonically increasing sequence

Expected Result

A table-like object in memory with monotonically increasing message numbers

Actual Result

Step 7 Step Execution Status: **Not Executed**

Description

Compute the difference between message ids via diff = arr['private_seqNum'][1:] - arr['private_seqNum'][:-1]

Expected Result

A histogram normalized to the total number of samples from 1 to the maximum number of the array diff.

Actual Result

Step 8 Step Execution Status: **Not Executed**

Description

Confirm that all values are one. Any larger numbers would indicate a gap in messages

Expected Result

The result is that all differences in the histogram should be 1. Any values larger than 1, which indicates a gap in messages, must be traced to a problem with the CSC and not an issue with the EFD.

Actual Result

Step 9 Step Execution Status: **Not Executed**

Description

Produce a report outlining this test

Expected Result

- A document describing the process including which topics/fields where used and what time window was

selected.
- The document shall be in the form of a notebook with saved outputs, or an instance of an nbreport

Actual Result

5.1.3.5 LVV-T2115 - Verify telemetry is uninterrupted for 5 days and can be analyzed via a notebook at the summit

Version **1**. Open *LVV-T2115* test case in Jira.

Test that telemetry is being recorded without missing messages for 5 days. This analysis is to be carried out using summit infrastructure.

Preconditions:

Execution status: **Not Executed**

Final comment:

Detailed steps results:

Step 1 Step Execution Status: **Not Executed**

Description
Log in to whatever VPNs are necessary to access to the summit notebook aspect of the RSP

Expected Result
VPN connection is active

Actual Result

Step 2 Step Execution Status: **Not Executed**

Description

Log in to the summit notebook aspect: https://summit-lsp.lsst.codes/nb
Make sure to choose a recent weekly and a large instance

Expected Result

The JupyterLab interface is displayed in the browser

Actual Result

Step 3 Step Execution Status: **Not Executed**

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

Step 4 Step Execution Status: **Not Executed**

Description

Connect to the summit EFD

Example Code

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

Expected Result

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

Actual Result

Step 5 Step Execution Status: **Not Executed**

Description

Choose a topic to query and select a 5 day window of data. The window is arbitrary, but must be explicit (not relative to now()) so that it can be reproduced. Note that command topics shall be avoided for this purpose since the semantics of private_seqNum are different in the context of commands than for other topic types

Expected Result

A time window and topic name that will be queried for the sequence number

Actual Result

Step 6 Step Execution Status: **Not Executed**

Description

Care shall be taken to fix up the id column, private_seqNum. It must be monotonically increasing, but gets reset when the CSC is restarted. When this happens an offset must be applied to the rest of the message ids in order to produce a monotonically increasing sequence

Expected Result

A table-like object in memory with monotonically increasing message numbers

Actual Result

Step 7 Step Execution Status: **Not Executed**

Description

Compute the difference between message ids via diff = arr['private_seqNum'][1:] - arr['private_seqNum'][:-1]

Expected Result

A histogram normalized to the total number of samples from 1 to the maximum number of the array diff.

Actual Result

Step 8 Step Execution Status: **Not Executed**

Description

Confirm that all values are one. Any larger numbers would indicate a gap in messages

Expected Result

The result is that all differences in the histogram should be 1. Any values larger than 1, which indicates a gap in messages, must be traced to a problem with the CSC and not an issue with the EFD.

Actual Result

Step 9 Step Execution Status: **Not Executed**

Description

Produce a report outlining this test

Expected Result

- A document describing the process including which topics/fields where used and what time window was selected.
- The document shall be in the form of a notebook with saved outputs, or an instance of an nbreport

Actual Result

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
CSC	Commandable SAL Component
DDS	Data Distribution System
DM	Data Management
DMTN	DM Technical Note
EFD	Engineering and Facility Database
LSE	LSST Systems Engineering (Document Handle)
LSST	Legacy Survey of Space and Time (formerly Large Synoptic Survey Telescope)
LVV	LSST Verification and Validation
M1	primary mirror
M1M3	Primary Mirror Tertiary Mirror
M3	tertiary mirror
MTM1M3	Main Telescope M1M3
NCSA	National Center for Supercomputing Applications
PMCS	Project Management Controls System
RSP	Rubin Science Platform
TAI	International Atomic Time
UTC	Coordinated Universal Time
VPN	virtual private network