Data Auditing Framework (DAF)

Technical Design Document

**For**

****

Prepared by:

**Saket Suman (Infosys)**

July 6, 2020

For internal use only

**Document Revision Control**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Rev #** | **Date** | **Author** | **Section #** | **What Changed?** |
| 1. | 15th June 2020 | **Saket Suman** |  | Initial version for review |
|  |  |  |  |  |
|  |  |  |  |  |

**Table of Contents**

1. [INTRODUCTION](#Introduction)

2. [ENVIRONMENT DETAILS](#environment_details)

3. [CODE LOCATION](#code_location)

4. [ARCHITECTURAL DIAGRAM](#_ARCHITECTURAL_DIAGRAM)

5. [AUDIT FLOW](#_AUDIT_FLOW)

# INTRODUCTION

Data audits are another form of an audit activity wherein an organization’s databases are evaluated or inspected for any issues or for regulatory compliance. Most of the time, data audits will look into the kind and amount of data that your business or establishment collects or processes in order to perform or while in the completion of its offered services and products.

# ENVIRONMENT DETAILS

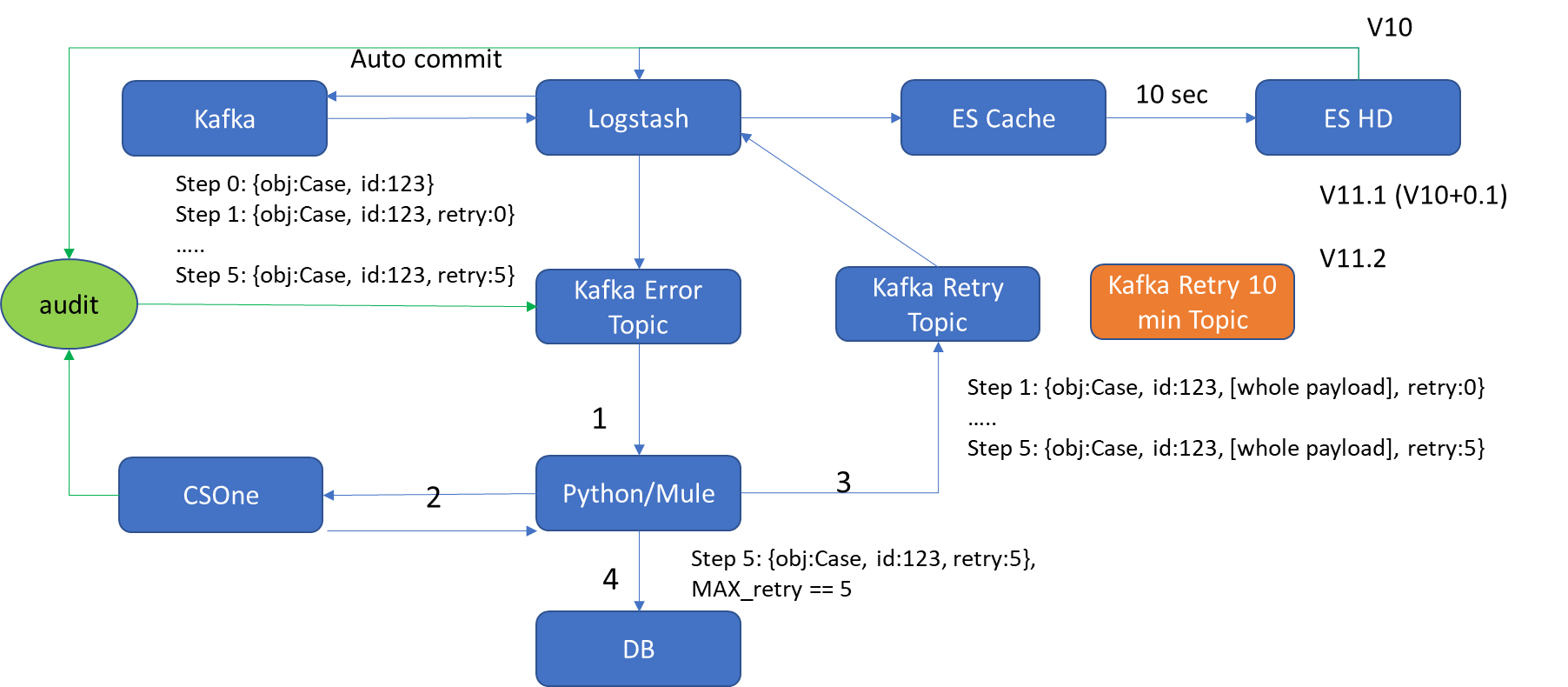
* Python 3.7
* Python Packages Used:
  + - * JayDeBeApi,
      * JPype1 0.6.3
      * snowflake-connector-python
      * hvac

# CODE LOCATION

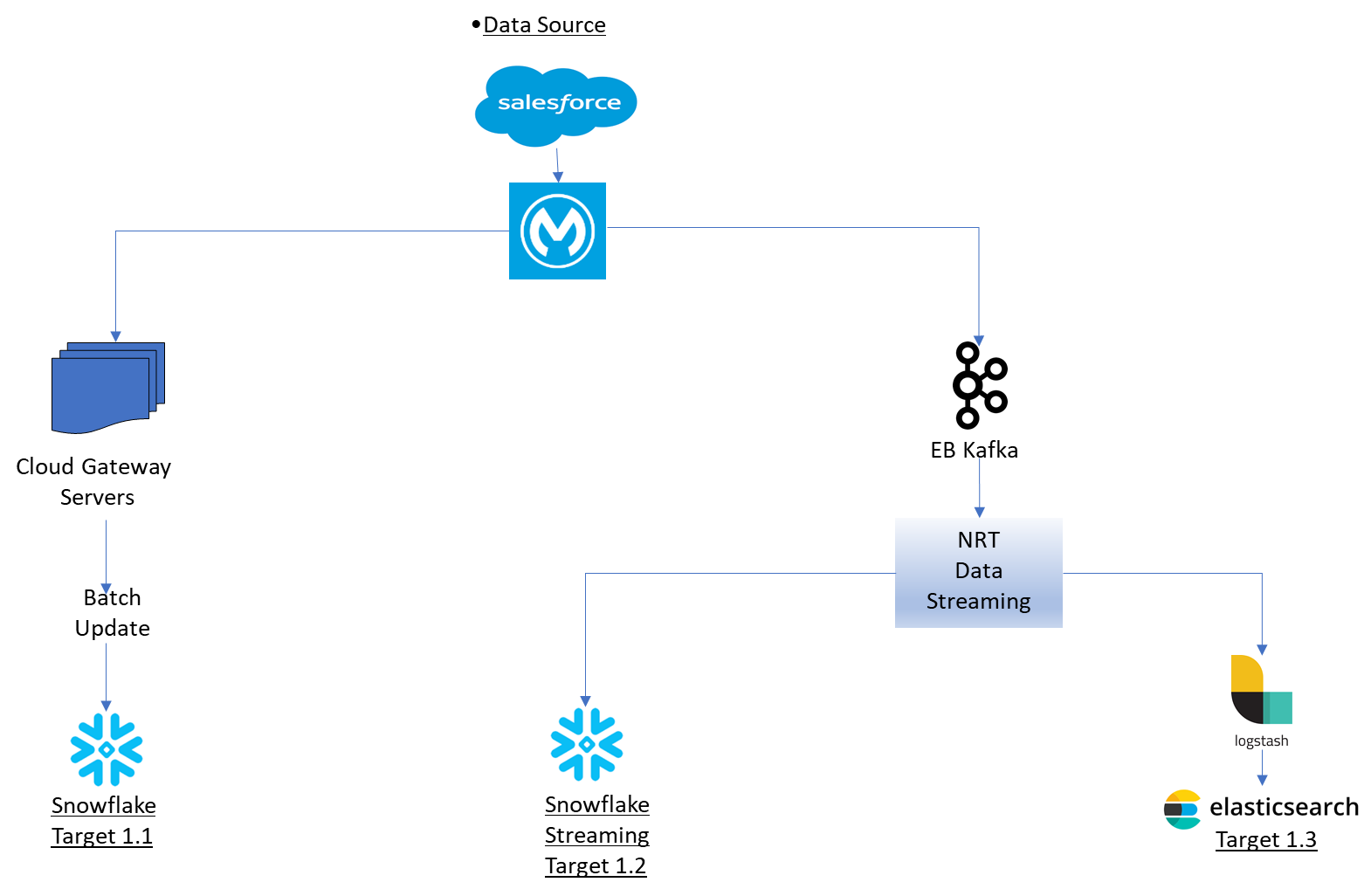
* Code is running on Cloud via Stage and Prod environments
* Once logged in into stg/prod environment, go to /users/cgwcsd/PROD/custom-framework-code-base/data-retention/cm-audit-framework for codebase of all the scripts
* The git location for the scripts is mentioned below:

<https://gitscm.cisco.com/projects/IT-CITS-CUSTOMERCARE-CUSTCARECMSTACK/repos/cm-audit-framework/browse>

# ARCHITECTURAL DIAGRAM



# AUDIT FLOW



Steps of Data Audit:

**1.1:** To query SFDC (source database)

*SELECT ID, SFDC\_Counter\_\_c, Type*

*FROM CASE*

*WHERE LASTMODIFIEDDATE >= {start time} AND LASTMODIFIEDDATE < {end time}*

**2.1:** To query Snowflake Batch load (Target 1.1)

*SELECT ID*

*FROM EDW\_CSD\_ETL\_DB.SS.CS1\_CASE*

*WHERE LASTMODIFIEDDATE >= {start time} AND LASTMODIFIEDDATE < {end time}*

**2.2:** To query Snowflake Kafka streaming (Target 1.2)

*SELECT ID, SFDC\_COUNTER\_\_C*

*FROM EDW\_CSD\_ETL\_DB.SS.CS1\_CASE\_PT*

*WHERE LASTMODIFIEDDATE >= {start time} AND LASTMODIFIEDDATE < {end time}*

**2.3:** To query Elasticsearch (Target 1.3)

*GET cm-case-case-otm/\_search{'\_source': ['caseID', 'sfdcCounter', 'lastModifiedDate'], 'query': {'bool': {'filter': [{'range': {'lastModifiedDate': {'gte': '{start time}', 'lte': '{end time}'}}}], 'must': [{'exists': {'field': 'caseNumber'}}]}}, 'size': 10000}*

**3.1:** To take subset from step 1 and filter using Type in ('TAC', 'CIN', 'ANC', 'SMB', 'IC')

**3.2:** To compare Snowflake SFDC\_result\_set compared with snowflake\_result\_set

**3.3:** To compare Snowflake-streaming SFDC\_result\_set compared with snowflake\_streaming\_result\_set

**3.4:** To compare Elasticsearch SFDC result\_set case\_type in ('TAC', 'CIN', 'ANC', 'SMB', 'IC') compared with ES\_result\_set

**4.1:** To match ES\_result\_set with full ES and segregate the unmatched one

**5:** Putting the unmatched one on New Kafka error Topic