E2E - API Audit Logs Test Scenarios

1. Introduction

Purpose: The purpose of this document is to outline the end-to-end test scenarios for the Audit logs application components to ensure that critical functional flows are validated.

Scope:

- The scope of these tests includes NRT (consuming service), audit-logs-srv, Kafka connect gcs sink srv, GCS Bucket and Hive table.
- All NRT endpoints.

2. Test Environment

Environment Setup: Tests will be executed in the staging environment.

1. Audit logs service API -
2. GCS Bucket
3. Hive DB
4. Kafka Lenses Details -

https://lenses.kafka-v2-luminate-core-prod.scus.prod.us.walmart.net:8080/data/

https://lenses.kafka-v2-luminate-core-prod.eus.prod.us.walmart.net:8080/data/

Prerequisites:

1. NRT service should be integrated with api-logging jar and should be running in the stage env.
2. Api-Audit logs service should be up and running in the stage env. API spec should follow API standards- API Spec
3. Kafka connect gcs sink service should be up and running in the stage env.
4. GCS bucket be available.
5. Hive Table should be created as per DDL on API Logs Table

3. Test Scenarios

	Scenario	Objective	Preconditions	Test Steps	Expected Result	Results
1	Verify Traceability for a valid IAC (Inventory Actions Capture) POST API request	Verify that IAC request is audited and stored in Hive DB with all the required details	All the services should be up and running. IAC request should be valid and processed.	1. **Step 1**: Hit a valid IAC endpoint that returns 201 Response Code. 2. **Step 2**: validate that trace_id is returned in the response header. 3. **Step 3**: Use this trace_id to look up in Hive DB- **api_logs.** SE LECT * FROM <schema>.api_logs where trace_id='<trace_Id>'; 4. **Step 4** : check folder structure and file format in GCS bucket. It should be in following format	1. A record should be returned from Hive DB for above request. This record should have all the required details. 2. This record should have	

2	Verify Kafka message schema for a valid IAC (Inventory Actions Capture) POST API request	Verify that IAC request is audited and published on Kafka topic (api_audit_logs_stg) with expected schema - Kafka Header and Body.	All the services should be up and running. IAC request should be valid and processed.	1. **Step 1**: Hit a valid IAC endpoint that returns 201 Response Code. 2. **Step 2**: Validate that trace_id is returned in the response header. 3. **Step 3**: Use this trace_id to look up in Lenses in the respective Kafka cluster.	1. Audit logs API Response should return with 204 Response code. 2. Kafka header and body should be in the expected format as present on <Confluence_link> . This record should have all the mandatory required details.	
3	**NRT App** - Verify audit api logs response for a valid IAC (Inventory Actions Capture) POST API request	Verify that IAC request is audited and audit-logs API returns 204 Response code.	All the services should be up and running. IAC request should be valid and processed.	1. **Step 1**: Hit a valid IAC endpoint that returns 201 Response Code. 2. **Step 2**: using traceId, Validate that request is stored in Hive DB. 3. **Step 3**: Use this trace_id to look up in Lenses ;	• Audit logs API Response should return with 204 Response code. • Response header should contain trace_id • Request_Body and Response_body should be logged. • endpoint name should be saved as IAC.	
4	**IAC App** - Verify audit api logs response for a valid IAC (Inventory Actions Capture) POST API request	Verify that IAC request is audited and audit-logs API returns 204 Response code.	All the services should be up and running. IAC request should be valid and processed.	1. **Step 1**: Hit a valid IAC endpoint that returns 201 Response Code. 2. **Step 2**: using traceId, Validate that request is stored in Hive DB. 3. **Step 3**: Use this trace_id to look up in Lenses ;	• Audit logs API Response should return with 204 Response code. • Response header should contain trace_id • Request_Body and Response_body should be logged. • endpoint name should be saved as IAC. • Headers should log wm_svc.name as "**chhanel performance-iac**"	
5	Verify audit API logs for an Inventory Actions Capture (IAC) POST API request that returns a 4xx response.	Verify that IAC request having GTIN_NOT_MAPPED is audited and stored in Hive DB	All the services should be up and running. IAC request should be valid and processed.	1. **Step 1**: Hit a invalid IAC endpoint that returns 400 Response Code. This request can have all GTINs not mapped in the request payload. 2. **Step 2**: using traceId, Validate that request is stored in Hive DB with relevant details - request_body, response_body. 3. **Step 3**: Use this trace_id to look up in Lenses ;	• Request is stored in Hive DB with relevant details - request_body, response_body etc. • Response_code should be stored as 4xx.	
6	**IAC app** - Verify audit API logs for an Inventory Actions Capture (IAC) POST API request that returns a 4xx response.	Verify that IAC request having GTIN_NOT_MAPPED is audited and stored in Hive DB	All the services should be up and running. IAC request should be valid and processed.	1. **Step 1**: Hit a invalid IAC endpoint that returns 400 Response Code. This request can have all GTINs not mapped in the request payload. 2. **Step 2**: using traceId, Validate that request is stored in Hive DB with relevant details - request_body, response_body. 3. **Step 3**: Use this trace_id to look up in Lenses ;	• Request is stored in Hive DB with relevant details - request_body, response_body etc. • Response_code should be stored as 4xx. • Headers should log wm_svc.name as "**chhanelperformance-iac**"	
7	Verify audit api logs response for a valid TransactionHistory GET API request	Verify that Transaction History request is audited and audit-logs API returns 204 Response code.	All the services should be up and running. Transaction History request should be valid and processed.	1. **Step 1**: Hit a valid Transaction History endpoint that returns 200 Response Code. 2. **Step 2**: using traceId, Validate that request is stored in Hive DB. 3. **Step 3**: Use this trace_id to look up in Lenses ;	• Audit logs API Response should return with 204 Response code. • Response header should contain trace_id • Response_body should be logged. • endpoint name should be saved as transactionHistory.	

8	Verify audit api logs response for a TransactionHistory API request that returns a 4xx response.	Verify that Transaction History request is audited and audit-logs API returns 400 Response code.	All the services should be up and running. GTIN present in Transaction History request should be not mapped with supplier .	1. **Step 1**: Hit a Transaction History endpoint that returns 400 Response Code. 2. **Step 2**: using traceId, Validate that request is stored in Hive DB. 3. **Step 3**: Use this trace_id to look up in Lenses ;	• Audit logs API Response should return with 204 Response code. • Response header should contain trace_id • Response_body should be logged. • endpoint name should be saved as transactionHistory.	
9	Verify audit api logs response for a IAC API request that returns a 5xx response.					
10	Verify audit api logs response for a TransactionHistory API request that returns a 5xx response.					
11						
12	Verify that for NRT, you are able to get all the API hits so far.					
13	Verify that for NRT, you are able to get all the API hits in a month/year.					
14	Verify that for a supplier (consumer_id), you are able to get all the API hits grouped by endpoint name and response code.					
15	Verify that you are able to get all the API hits grouped by supplier name, endpoint name and response code.					
16	Verify that you are able to get successful API hits (non 5xx) percentage for IAC.					
17	Verify that you are able to get successful API hits(non 5xx) percentage for Transaction History.					
18	Verify that you are able to get successful API hits(non 5xx) percentage for all the APIs for a supplier.					
19	Verify that you are able to get successful API hits(non 5xx) percentage for all the APIs for all the suppliers.					
20	Verify that for a supplier (consumer_id), you are able to obtain the requested GTINs and storeNbr that resulted in a 400 error with the message, "The request store-GTIN is not mapped to the supplier"."					
21	Verify that for a supplier (consumer_id), you are able to obtain the requested GTINs and storeNbr that resulted in a 400 error with the technical reasons.					
22	Verify that you are able to compare the API usage month by month and year by year for a supplier (consumer_id).					
23	Verify that you are able to obtain the store numbers for which the IAC API is used.					
24	Verify that you are able to get the **new** store numbers for which the IAC API is used in this month.					
25	Verify that you can identify the new suppliers who began using the IAC API this month.					
26	Verify that you can identify the new suppliers who began using the IAC API this year.					
27	Verify that you can identify the new suppliers who began using the Transaction History API this month.					
28	Verify that you can identify the new suppliers who began using the Transaction History API this year.					
29	Verify that for IAC, you are able to obtain API hits based on event types.					
30	Verify that for IAC, you are able to obtain all the used GTINs, grouped by month and year.					

31	Verify that for IAC, you are able to get List of consumerids / number of supplier IDs adopted					
32	Verify that for IAC and for a supplier, you are able to get the number/percentage of requests that took more than 200 ms in a day.					
33	Verify that for TransactionHistory and for a supplier, you are able to get the number /percentage of requests that took more than 200 ms in a day.					
34	Verify that for IAC , you are able to get the number/percentage of requests that took more than 200 ms in a day.					
35	Verify that for TransactionHistory , you are able to get the number/percentage of requests that took more than 200 ms in a day.					
36	Verify that for NRT , you are able to get the number/percentage of requests that took more than 200 ms in a month.					

Load Testing

1. Transaction History
2. IAC
3. Mix of transaction History and IAC