

Oracle® Fusion Middleware

Troubleshooting GoldenGate Stream Analytics



F17035-07
January 2022



Oracle Fusion Middleware Troubleshooting GoldenGate Stream Analytics,

F17035-07

Copyright © 2019, 2021, Oracle and/or its affiliates.

Primary Author: Oracle Corporation

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

| | |
|-----------------------------|----|
| Audience | v |
| Documentation Accessibility | v |
| Related Documents | v |
| Conventions | vi |

1 Troubleshooting Oracle Stream Analytics

| | |
|----------------------------------------------------------------------------|------|
| 1.1 Pipeline Debug and Monitoring Metrics | 1-1 |
| 1.1.1 Spark Standalone | 1-1 |
| 1.1.2 Spark on YARN | 1-2 |
| 1.1.3 Pipeline Details | 1-3 |
| 1.1.4 Stage Details | 1-4 |
| 1.1.5 Query Details | 1-5 |
| 1.1.6 Execution and HA Statistics | 1-6 |
| 1.1.7 Detailed Query Analysis | 1-8 |
| 1.1.8 Complete CQL Engine Statistics | 1-9 |
| 1.1.9 Internal Kafka Topics | 1-10 |
| 1.2 Common Issues and Remedies | 1-10 |
| 1.2.1 System Settings | 1-10 |
| 1.2.1.1 SQL Queries Configuration | 1-11 |
| 1.2.1.2 VM Memory Issue in Yarn Cluster | 1-11 |
| 1.2.2 Web-tier Logging | 1-11 |
| 1.2.2.1 Change Jetty Log Level | 1-11 |
| 1.2.3 Pipeline | 1-11 |
| 1.2.3.1 Unable to Deploy more than 17 Pipelines | 1-11 |
| 1.2.3.2 Pipelines are not running as expected | 1-12 |
| 1.2.3.3 GGSA Pipeline getting Terminated | 1-13 |
| 1.2.3.4 Live Table Shows Listening Events with No Events in the Table | 1-14 |
| 1.2.3.5 Live Table Still Shows Starting Pipeline | 1-15 |
| 1.2.3.6 Time-out Exception in the Spark Logs when you Unpublish a Pipeline | 1-15 |
| 1.2.3.7 Piling up of Queued Batches in HA mode | 1-16 |
| 1.2.3.8 Null Record from Summary in Query Stage | 1-16 |

| | | |
|----------|--------------------------------------------------------------------------|------|
| 1.2.4 | Stream | 1-16 |
| 1.2.4.1 | Cannot See Any Kafka Topic or a Specific Topic in the List of Topics | 1-16 |
| 1.2.4.2 | Input Kafka Topic is Sending Data but No Events Seen in Live Table | 1-16 |
| 1.2.5 | Connection | 1-17 |
| 1.2.5.1 | Database Connection Failure | 1-17 |
| 1.2.5.2 | Druid Connection Failure | 1-17 |
| 1.2.5.3 | Coherence Connection Failure | 1-17 |
| 1.2.5.4 | JNDI Connection Failure | 1-17 |
| 1.2.6 | Target | 1-18 |
| 1.2.6.1 | Cannot see any Events in Targets | 1-18 |
| 1.2.7 | Geofence | 1-18 |
| 1.2.7.1 | Name and Description Fields are not displayed for the DB-based Geofences | 1-18 |
| 1.2.7.2 | DB-based Geofence is not Working | 1-18 |
| 1.2.8 | Cube | 1-18 |
| 1.2.8.1 | Unable to Explore Cube which was Working Earlier | 1-19 |
| 1.2.8.2 | Cube Displays "Datasource not Ready" | 1-19 |
| 1.2.9 | Dashboard | 1-19 |
| 1.2.9.1 | Visualizations Appearing Earlier are No Longer Available in Dashboard | 1-19 |
| 1.2.9.2 | Dashboard Layout Reset after You Resized/moved the Visualizations | 1-19 |
| 1.2.9.3 | Streaming Visualizations Do not Show Any Data | 1-20 |
| 1.2.10 | Live Output | 1-20 |
| 1.2.10.1 | Issues with Live Output | 1-20 |
| 1.2.10.2 | Missing Events due to Faulty Data | 1-22 |
| 1.2.11 | Pipeline Deployment Failure | 1-22 |

Preface

Troubleshooting Oracle Stream Analytics discusses the common issues with solutions, in Oracle Stream Analytics.

Topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

Audience

This document is intended for users of Oracle Stream Analytics.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

Documentation for Oracle Stream Analytics is available on [Oracle Help Center](#).

Also see the following documents for reference:

- *Understanding Oracle Stream Analytics*
- *Developing Custom Jars and Custom Stages in Oracle Stream Analytics*
- *Quick Installer for Oracle Stream Analytics*
- *Known Issues in Oracle Stream Analytics*
- *Spark Extensibility for CQL in Oracle Stream Analytics*

Conventions

The following text conventions are used in this document:

| Convention | Meaning |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------|
| boldface | Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary. |
| <i>italic</i> | Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values. |
| monospace | Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter. |

1

Troubleshooting Oracle Stream Analytics

Oracle Stream Analytics is used to analyze and monitor a stream of events in real-time. You can create pipelines to model the stream processing solutions. Typically, an Oracle Stream Analytics pipeline runs continuously for long durations until it is stopped explicitly, or it fails. This document provides information needed to troubleshoot various issues that you might encounter while creating, executing, or deploying a pipeline.

1.1 Pipeline Debug and Monitoring Metrics

For every running Oracle Stream Analytics pipeline, there is a corresponding spark application deployed to the Spark Cluster. If Oracle Stream Analytics pipeline is deployed and running in draft mode or published mode, user can monitor and analyze the corresponding Spark application using real-time metrics provided by Oracle Stream Analytics. These metrics provide detailed run-time insights for every pipeline stage and user can drill down to operator level details. Note that these metrics are in addition to metrics provided by Spark. For each application, Oracle Stream Analytics provides detailed monitoring metrics which user can use to analyze if pipeline is working as per expectation.

The following topics explain how to access monitoring and debug metrics.

1.1.1 Spark Standalone

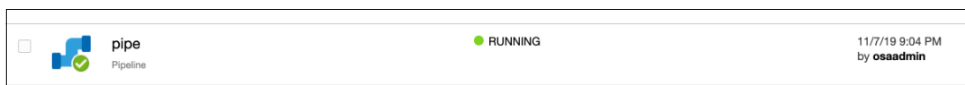
When running on Spark Standalone, the Spark Master URL is same as the host name in **Spark REST URL** field plus the **Spark standalone master console port** field. Spark Master URL is `http://<Spark REST URL>:< Spark standalone master console port>`. You can obtain Spark Master URL from the **System Settings** page.

The Spark Master page also displays a list of running applications. Click on an application to see details such as application ID, name, owner, current status.

For pipelines in 'Published' status, you can navigate to the metrics console from GGSA's catalog page as shown in the screenshot below.



You can check the status of a published application on the catalog page, as shown in the screenshots below:



1.1.2 Spark on YARN

For published pipelines it is easier to navigate to Application Master console from GoldenGate Stream Analytics' catalog page.

1. For draft pipelines, you can navigate to YARN Applications page using the **YARN Resource Manager URL** and **YARN Master console port** values in **System Settings**.
2. Application master url is `http://<YARN Resource Manager URL>:<Yarn master Console Port>`.

This page displays all the applications running in YARN:

The screenshot shows the Hadoop 'All Applications' page. It features a sidebar with navigation options like 'Cluster', 'About Nodes', and 'Applications'. The main area displays a table of applications with columns for 'App', 'Scheduler', 'Name', 'Application Type', 'Queue', 'Start Time', 'Finish Time', 'State', 'Final Status', 'Progress', and 'Tracking UI'. Two application entries are visible, both in a 'KILLED' state.

3. After identifying the application from the list, click the **ApplicationMaster** link, to open the Spark Application Details page:

The screenshot shows the Spark Application Details page for a job. It includes a navigation bar with tabs for 'Jobs', 'Stages', 'Storage', 'Environment', 'Executors', 'CQLEngine', 'Pipeline', and 'Streaming'. The page title is 'Spark Jobs (?)'. It displays job statistics such as 'User: osa', 'Total Uptime: 53.1 h', and 'Scheduling Mode: FIFO'. There are sections for 'Active Jobs (1)' and 'Completed Jobs (191004, only showing 903)'. A table lists the active job with columns for 'Job Id', 'Description', 'Submitted', 'Duration', 'Stages: Succeeded/Total', and 'Tasks (for all stages): Succeeded/Total'. The active job is a streaming job that has failed.

1.1.3 Pipeline Details

Click the **Pipeline** tab to view the Pipeline details page.

The screenshot shows the Oracle Stream Analytics UI Pipeline Details page. At the top, there is a navigation bar with tabs for Jobs, Stages, Storage, Environment, Executors, CQLEngine, Pipeline (selected), and Streaming. The page title is 'OSA Pipeline'. Below the title, the Pipeline ID is 'application_1558749296171_0002' and the Pipeline Name is 'sx_MY_PIPELINE_8E4CCDD6_471C_4A28_AB15_2BF67A951279_yzkWtshA_public'. The 'Pipeline Stages' section contains a table with the following data:

| Stage Name | Total Number of Transformations | Total Output Partitions | Total Output Events | Avg Output Rate(events/second) | Current Output Rate(events/second) |
|------------|---------------------------------|-------------------------|---------------------|--------------------------------|------------------------------------|
| Q2 | 5 | 2 | 4546 | 0 | 0 |
| GG_STREAM | 5 | 1 | 6098 | 0 | 0 |

The 'Pipeline DAG' section shows a 'DAG Visualization' of the pipeline. The diagram illustrates a single stage named 'Stage:GG_STREAM' which contains two transformations: 'DirectKafkaInputDStream [10]' and 'FlatMappedDStream [9]'. An arrow indicates the flow from the input stream to the output stream.

This page has following information about a pipeline:

- **Pipeline ID:** Unique pipeline id in Spark Cluster
- **Pipeline Name:** Name of Oracle Stream Analytics Pipeline given by user in Oracle Stream Analytics UI.
- **Pipeline Stages Table:** This section displays a table having detailed runtime metrics of each pipeline stage. Each entry in the table is corresponding to a pipeline stage in Oracle Stream Analytics UI pipeline graph.
- **Pipeline DAG:** This is a visual representation of all stages in form of a DAG where it displays the parent-child relation various pipeline stages. This diagram also shows the information about the transformations in each of the pipeline stage.
- The Pipeline Stages table contains the following columns.
- **Total Number of Transformations:** This measurement is number of spark transformations applied to compute each stage.

Oracle Stream Analytics supports various types of stages e.g. Query Stage, Pattern Stage, Custom Stage etc. For each pipeline stage, Oracle Stream Analytics defines a list of transformations which will be applied on the input stream for the stage. The output from final transformation will be the output of stage.

- **Total Output Partitions:** This measurement is total number of partitions in the output stream of each stage.
- Every pipeline stage has its own partitioning requirements which are determined from stage configurations. For example, If a QUERY stage defines a summary function and doesn't define a group-by column, then QUERY stage will have only one partition because there will be no partitioning criteria available.
- **Total Output Events:** This measurement is total number of output events (not micro-batches) emitted by each stage.

- **Average Output Rate:** This measurement is the rate at which each stage has emitted output events so far. The rate is a ratio of total number of output events so far and total application execution time.

If the rate is ZERO, then it doesn't always mean that there is ERROR in stage processing. Sometime stage doesn't output any record at all (e.g. No event passed the Filter in Query stage). This can happen if rate of output events is less than 1 events/second.

- **Current Output Rate:** This measurement is the rate at which each stage is emitting output events. The rate is ratio of total number of output events and total application execution time since last metrics page refresh. To get better picture of current output rate, please refresh the page more frequently.

1.1.4 Stage Details

Click a Name in the **Name** column of the Pipeline Stages table to navigate to stage details page.

The screenshot displays the 'OSA Pipeline Stage Details' page in the Oracle Stream Analytics UI. The page header shows 'spark 2.4.3' and the pipeline name 'sx_MY_PIPELINE_8E4CCDD6_471C_4A2... application UI'. The navigation menu includes 'Jobs', 'Stages', 'Storage', 'Environment', 'Executors', 'CQLEngine', 'Pipeline', and 'Streaming'. The main content area is titled 'OSA Pipeline Stage Details' and provides the following information:

- Pipeline ID:** application_1558749296171_0002
- Pipeline Name:** sx_MY_PIPELINE_8E4CCDD6_471C_4A2B_A815_2BF67A951279_yzkWtshA_public
- Stage ID:** Q2

Below this information is a table of 'Stage Transformations':

| Transformation Name | Transformation Type |
|---------------------|---------------------|
| MonitorDStream | Oracle |
| CQLDStream | Oracle |
| TotalOrderDStream | Oracle |
| TransformedDStream | Spark Native |
| TransformedDStream | Spark Native |

At the bottom, the 'Stage DAG' visualization shows a flow from 'TransformedDStream [5]' to 'TransformedDStream [4]'.

This page provides details about all the transformations in specific stage.

- **Pipeline ID:** Unique pipeline id in Spark Cluster
- **Pipeline Name:** Name of Oracle Stream Analytics Pipeline given by user in Oracle Stream Analytics UI.
- **Stage ID:** Unique stage id in DAG of stages for Oracle Stream Analytics Pipeline.
- **Stage DAG:** This is a visual representation of all transformations in form of a DAG where it displays the parent-child relation various pipeline transformations.

Note Oracle Stream Analytics allows different transformation types including Business Rules but drilldowns are allowed only on CQLDStream transformations. Inside CQLDStream transformation, the input data is transformed using a continuously running query (CQL) in a CQL Engine. Note that there will be one CQL engine associated with one Executor.

- **Stage Transformations Table:** This table displays details about all transformations being performed in a specific stage. Each entry in the table is corresponding to a transformation operation used in computation of the stage. You can observe that final transformation in every stage is MonitorDStream. The

reason is that MonitorDStream pipes output of stage to Oracle Stream Analytics UI Live Table.

- **Transformation Name:** Name of output DStream for the transformation. Every transformation in spark results into an output dstream.
- **Transformation Type:** This is category information of each transformation being used in the stage execution. If transformation type is "Oracle", it is based on Oracle's proprietary transformation algorithm. If the transformation type is "Native", then transformation is provided by Apache Spark implementation.

CQLDStream transformation allows further drill down as described in the [Query Details](#) section.

1.1.5 Query Details

In the **Stage Transformation** table, click on a CQLDStream transformation to open **CQL Engine Summary** page for query-specific details.

The screenshot displays the 'CQL Engine Summary' page for a query. The page includes the following sections:

- Header:** Spark 2.4.3 navigation bar with tabs for Jobs, Stages, Storage, Environment, Executors, CQL Engine, Pipeline, and Streaming. The current pipeline is 'sx_MY_PIPELINE_8E4CCDD6_471C_4A28... application UI'.
- CQL Engine Summary:**
 - Pipeline ID: application_1558749296171_0002
 - Pipeline Name: sx_MY_PIPELINE_8E4CCDD6_471C_4A28_A815_2BF67A951279_yzkW7shA_public
 - Stage ID: Q2
- Running Queries (1):**

| Query ID | Query Text |
|----------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| sx_MY_PIPELINE_8E4CCDD6_471C_4A28_A815_2BF67A951279_yzkW7shA_public1 | iSTREAM (SELECT table_1 AS table_1, op_type AS op_type, op_ts AS op_ts, current_ts AS current_ts, pos AS pos, before_ORDER_ID AS before_ORDER_ID, before_ORDER_STATUS AS before_ORDER_STATUS, before_PRODUCT_SKU AS before_PRODUCT_SKU, before_QUANTITY AS before_QUANTITY, before_REVENUE AS before_REVENUE, after_ORDER_ID AS after_ORDER_ID, after_ORDER_STATUS AS after_ORDER_STATUS, after_PRODUCT_SKU AS after_PRODUCT_SKU, after_QUANTITY AS after_QUANTITY, after_REVENUE AS after_REVENUE FROM src8130806D_53A1_438D_BE2F_494FA3985150 WHERE (before_ORDER_STATUS is not null)) |
- Registered Sources(Stream/Relation) (2):**

| Source Name | Type | Timestamp Type | Attributes |
|-------------|--------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GG_STREAM | stream | PROCESSING | table_1 char(1024), op_type char(1024), op_ts char(1024), current_ts timestamp, pos char(1024), before_ORDER_ID char(1024), before_ORDER_STATUS char(1024), before_PRODUCT_SKU char(1024), before_QUANTITY int, before_REVENUE int, after_ORDER_ID char(1024), after_ORDER_STATUS char(1024), after_PRODUCT_SKU char(1024), after_QUANTITY int, after_REVENUE int |
- External Sources (0):**

| Source Name | External Source Name | External Source URL | Attributes |
|---------------------------------|----------------------|---------------------|------------|
| No external sources are listed. | | | |
- CQL Engines (1):**

| CQL Engine Id | Executor Id | Executor Host | Status |
|---------------|-------------|------------------------|--------|
| 1 | 1 | src06fdo.us.oracle.com | ACTIVE |

The **CQL Engine Summary** page for the query has details like query text, stream sources feeding the query, external sources if any and all CQL engines with which the query is registered.

1.1.6 Execution and HA Statistics

Click on a Query, in the CQL Engine Summary page.

CQL Engine Summary

Pipeline ID: application_1558749296171_0002
 Pipeline Name: sx_MY_PIPELINE_8E4CCDD6_471C_4A28_A815_2BF67A951279_yzkWtshA_public
 Stage ID: Q2

Running Queries (1)

| Query ID | Query Text |
|----------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| sx_MY_PIPELINE_8E4CCDD6_471C_4A28_A815_2BF67A951279_yzkWtshA_public1 | ISTREAM (SELECT table_1 AS table_1, op_type AS op_type, op_ts AS op_ts, current_ts AS current_ts, pos AS pos, before_ORDER_ID AS before_ORDER_ID, before_ORDER_STATUS AS before_ORDER_STATUS, before_PRODUCT_SKU AS before_PRODUCT_SKU, before_QUANTITY AS before_QUANTITY, before_REVENUE AS before_REVENUE, after_ORDER_ID AS after_ORDER_ID, after_ORDER_STATUS AS after_ORDER_STATUS, after_PRODUCT_SKU AS after_PRODUCT_SKU, after_QUANTITY AS after_QUANTITY, after_REVENUE AS after_REVENUE FROM src8130806D_53A1_439D_BE2F_494FA39B5150 WHERE (before_ORDER_STATUS is not null)) |

Registered Sources(Stream/Relation) (2)

| Source Name | Type | Timestamp Type | Attributes |
|-------------|--------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GG_STREAM | stream | PROCESSING | table_1 char(1024), op_type char(1024), op_ts char(1024), current_ts timestamp, pos char(1024), before_ORDER_ID char(1024), before_ORDER_STATUS char(1024), before_PRODUCT_SKU char(1024), before_QUANTITY int, before_REVENUE int, after_ORDER_ID char(1024), after_ORDER_STATUS char(1024), after_PRODUCT_SKU char(1024), after_QUANTITY int, after_REVENUE int |

External Sources (0)

| Source Name | External Source Name | External Source URL | Attributes |
|-------------|----------------------|---------------------|------------|
|-------------|----------------------|---------------------|------------|

CQL Engines (1)

| CQLEngine Id | Executor Id | Executor Host | Status |
|--------------|-------------|------------------------|--------|
| 1 | 1 | slc066do.us.oracle.com | ACTIVE |

You can view the Execution and HA statistics, in the CQL Engine Query details page that is displayed.

CQL Engine Query Details

Query ID: sx_MY_PIPELINE_8E4CCDD6_471C_4A28_A815_2BF67A951279_yzkWtshA_public1
 Query Text: ISTREAM (SELECT table_1 AS table_1, op_type AS op_type, op_ts AS op_ts, current_ts AS current_ts, pos AS pos, before_ORDER_ID AS before_ORDER_ID, before_ORDER_STATUS AS before_ORDER_STATUS, before_PRODUCT_SKU AS before_PRODUCT_SKU, before_QUANTITY AS before_QUANTITY, before_REVENUE AS before_REVENUE, after_ORDER_ID AS after_ORDER_ID, after_ORDER_STATUS AS after_ORDER_ID, after_PRODUCT_SKU AS after_PRODUCT_SKU, after_QUANTITY AS after_QUANTITY, after_REVENUE AS after_REVENUE FROM src8130806D_53A1_439D_BE2F_494FA39B5150 WHERE (before_ORDER_STATUS is not null))
 Num Partitions: 2

Execution Statistics

| Partition ID | CQLEngine ID | Total Output Events | Total Output Heartbeats | Throughput(events/sec) | Latency(ms) |
|--------------|--------------|---------------------|-------------------------|------------------------|-------------|
| 0 | 1(1) | 4086 | 212107 | 289 | 3 |
| 1 | 1(1) | 460 | 5007 | 32 | 30 |

* Entries in brackets are executor ids.

HA Statistics

| Partition ID | CQLEngine ID | Total FullSnapshots Created | Avg FullSnapshot CreationTime(ms) | Total FullSnapshots Loaded | Avg FullSnapshot LoadTime(ms) | Total JournalSnapshots Created | Avg JournalSnapshot CreationTime(ms) | Total JournalSnapshots Loaded | Avg JournalSnapshot LoadTime(ms) |
|--------------|--------------|-----------------------------|-----------------------------------|----------------------------|-------------------------------|--------------------------------|--------------------------------------|-------------------------------|----------------------------------|
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- Following are the details displayed on this page:
 - **Query ID:** System generated identifier for query
 - **Query Text:** Query String
 - **Number of Partitions:** This fields shows the degree of parallelism of the query. The degree of parallelism is defined by total number of input partitions processed by a query.
 Degree of parallelism depends on the many factors such as query constructs, number of input kafka partitions and number of executors assigned to application.
 - **Execution Statistics Table:** This section shows the detailed execution statistics of each operator.
 - **Partition ID:** Partition Sequence Id

- CQL Engine ID: Sequence ID of CQL Engine on which the partition is being processed.
- Total Output Events: Number of output events emitted by CQL query for each partition.
- Total Output Heartbeats: Number of heartbeat events emitted by CQL query for each partition. Please note that heartbeats are special events which ensures timestamp progression in Oracle Stream Analytics pipeline.
- Throughput: Ratio of total number of events processed and total time spent in processing for each partition.
- Latency: Average turnaround time taken to process a partition of stream.
- **HA Statistics**: This table shows the real-time statistics about query's HA operations. Note that unit of time is in MILLISECONDS.
 - Partition ID: Partition Sequence ID
 - CQL Engine ID: Sequence ID of CQL Engine on which the partition is being processed.
 - Total Full Snapshots Created: Total number of times the full state of query is serialized and saved.
 - Avg Full Snapshot Creation Time: Average time spent in serializing and saving the full state of query.
 - Total Full Snapshots Loaded: Total number of times the full state of query is de-serialized and loaded in query plan.
 - Avg Full Snapshot Load Time: Average time spent in de-serializing and loading the full state of query.
 - Total Journal Snapshots Created: Total number of times the journaled state of query is serialized and saved.
 - Avg Journal Snapshot Creation Time: Average time spent in serializing and saving the journaled state of query.
 - Total Journal Snapshots Loaded: Total number of times the journaled state of query is de-serialized and loaded in query plan.
 - Avg Journal Snapshot Load Time: Average time spent in de-serializing and loading the journaled state of query.

Full Snapshot is the complete state of query. The query state represent the internal data structure and state of each operator in query plan. Journal snapshot is partial and incremental snapshot having a start time and end time. Oracle Stream Analytics optimizes the state preservation by using Journal snapshot if possible.

1.1.7 Detailed Query Analysis

Click on a specific partition in Execution Statistics table, the CQL Engine Detailed Query Analysis page is displayed.

The screenshot displays the 'CQL Engine Detailed Query Analysis' page. At the top, it shows the Spark version (2.4.3) and navigation tabs for Jobs, Stages, Storage, Environment, Executors, CQL Engine, Pipeline, and Streaming. The current page is for a pipeline named 'sx_MY_PIPELINE_BE4CCDD6_471C_4A2... application UI'.

CQL Engine Detailed Query Analysis

Query ID: sx_MY_PIPELINE_BE4CCDD6_471C_4A28_A815_2BF67A951279_yqW7hA_public1
 Query Text: ISTREAM (SELECT table_1 AS table_1, op_type AS op_type, op_ts AS op_ts, current_ts AS current_ts, pos AS pos, before_ORDER_ID AS before_ORDER_ID, before_ORDER_STATUS AS before_ORDER_STATUS, before_ORDER_STREAM AS before_ORDER_STREAM, before_PRODUCT_SKU AS before_PRODUCT_SKU, before_QUANTITY AS before_QUANTITY, before_REVENUE AS before_REVENUE, after_ORDER_ID AS after_ORDER_ID, after_ORDER_STATUS AS after_ORDER_STATUS, after_PRODUCT_SKU AS after_PRODUCT_SKU, after_QUANTITY AS after_QUANTITY, after_REVENUE AS after_REVENUE FROM src81308060_53A1_439D_BE2F_494FA39B5150 WHERE (before_ORDER_STATUS is not null))
 Partition ID: 0

Operator Statistics

| Operator ID: | Total Input Events | Total Output Events | Total Input Heartbeats | Total Output Heartbeats | Throughput(events/sec) | Latency(us) |
|-------------------------------------------|--------------------|---------------------|------------------------|-------------------------|------------------------|-------------|
| src81308060_53A1_439D_BE2F_494FA39B5150#0 | 5467 | 5467 | 211218 | 211218 | 2743 | 364 |
| PO_SELECT#1 | 5467 | 4086 | 211218 | 212599 | 2146 | 465 |

Operator DAG

→ DAG Visualization

```

graph TD
    A[Query sx_MY_PIPELINE_BE4CCDD6_471C_4A28_A815_2BF67A951279_yqW7hA_public1] --> B[src81308060_53A1_439D_BE2F_494FA39B5150#0]
    B --> C[PO_SELECT#1 [2]]
  
```

This page contains details about each execution operator of CQL query for a particular partition of a stage in pipeline.

- **Query ID:** System generated identifier for query
- **Query Text:** Query String
- **Partition ID:** All operator details are corresponding to this partition id.
- **Operator Statistics:**
 - **Operator ID:** System Generated Identifiers
 - **Total Input Events:** Total number of input events received by each operator.
 - **Total Output Events:** Total number of output events generated by each operator.
 - **Total Input Heartbeats:** Total number of heartbeat events received by each operator.
 - **Total Output Heartbeats:** Total number of heartbeat events generated by each operator.
 - **Throughput(events/second):** Ratio of total input events processed and total time spent in processing for each operator.
 - **Latency(ms):** Total turnaround time to process an event for each operator.
- **Operator DAG:** This is visual representation of the query plan. The DAG will show the parent-child details for each operator. You can further drill down the execution statistics of operator. Please click on the operator which will open **CQL Operator Details Page**.

1.1.8 Complete CQL Engine Statistics

Click on the CQL Engine tab, to view the complete CQL engine statistics for all stages and all queries in the Pipeline.

The screenshot shows the 'CQL Engine Summary' page in the Spark UI. It includes the following sections:

- CQL Engine Summary:** Pipeline ID: application_1558749296171_0002, Pipeline Name: ox_MY_PIPELINE_BE4CCDD6_471C_4A28_A815_2BF67A951279_yxW7shA_public, Stage ID: Unrecon.
- Running Queries (1):** A table with columns 'Query ID' and 'Query Text'. The query ID is 'ox_MY_PIPELINE_BE4CCDD6_471C_4A28_A815_2BF67A951279_yxW7shA_public1' and the query text is a complex SQL statement involving a SELECT with multiple AS clauses and a WHERE clause.
- Registered Sources(Stream/Relation) (2):** A table with columns 'Source Name', 'Type', 'Timestamp Type', and 'Attributes'. It lists 'Q2' (stream, PROCESSING) and 'QG_STREAM' (stream, PROCESSING).
- External Sources (0):** A table with columns 'Source Name', 'External Source Name', 'External Source URL', and 'Attributes'. It is currently empty.
- CQL Engines (1):** A table with columns 'CQL Engine Id', 'Executor Id', 'Executor Host', and 'Status'. It shows one engine with ID '1', Executor Id '1', Host 'slc06fdo.us.oracle.com', and Status 'ACTIVE'.

This page contains additional information about each execution operator, apart from the CQL Engine Query Details page, provides all essential metrics for each operator.

Pipeline ID: Unique pipeline id in Spark Cluster

Pipeline Name: Name of Oracle Stream Analytics Pipeline given by user in Oracle Stream Analytics UI.

Stage ID: Unique stage ID in DAG of stages for Oracle Stream Analytics Pipeline.

Running Queries: This section displays list of CQL queries running to compute the CQL transformation for a stage. This table displays a system-generated Query ID and Query Text. Check Oracle Continuous Query Language Reference for CQL Query syntax and semantics. To see more details about query, click on the query id hyperlink in the table entry to open **CQL Engine Query Details** page.

Registered Sources: This section displays internal CQL metadata about all the input sources which the query is based upon. For every input stream of the stage, there will be one entry in this table.

Each entry contains source name, source type, timestamp type and stream attributes. Timestamp type can be PROCESSING or EVENT timestamped. If stream is PROCESSING timestamped, then timestamp of each event will be defined by system. If stream is EVENT timestamped, then timestamp of each event is defined by one of the stream attribute itself. A source can be Stream or Relation.

External Sources: This section displays details about all external sources with which input stream is joined. The external source can be a database table or coherence cache.

CQL Engines: This section displays a table having details about all instances of CQL engines used by the pipeline. Here are details about each field of the table:

- **CQLEngine Id:** System generated id for a CQL engine instance.
- **ExecutorId:** Executor Id with which the CQL engine is associated.
- **Executor Host:** Address of the cluster node on which this CQL engine is running.

- **Status:** Current Status of CQL Engine. Status can be either ACTIVE or INACTIVE. If it is ACTIVE, it means that CQL Engine instance is up and running, Otherwise CQL Engine is stopped explicitly.

1.1.9 Internal Kafka Topics

The internal Kafka topics and Group ID's used by GGSA are standardized to the following naming conventions:

Kafka Topics

| Topic | Resource | Operations |
|--------------------------------------------------------------------|----------|-----------------------------------|
| sx_backend_notification_<UUID> | Topic | CREATE,DELETE,DESCRIBE,READ,WRITE |
| sx_messages_<UUID> | Topic | CREATE,DELETE,DESCRIBE,READ,WRITE |
| sx_<application_name>_<stage_name>_public | Topic | CREATE,DELETE,DESCRIBE,READ,WRITE |
| sx_<application_name>_<stage_name>_draft | Topic | CREATE,DELETE,DESCRIBE,READ,WRITE |
| sx_<application_name>_public_<offset_number>_<stage_name>_<offset> | Topic | CREATE,DELETE,DESCRIBE,READ,WRITE |

Group IDs

| Group ID | Resource | Operations |
|--------------------------------------------------------------------|----------|---------------|
| sx_<UUID>_receiver | Group | DESCRIBE,READ |
| sx_<UUID> | Group | DESCRIBE,READ |
| sx_<application_name>_public_<offset_number>_<stage_name>_<offset> | Group | DESCRIBE,READ |

1.2 Common Issues and Remedies

This section provides a comprehensive list of items to verify if pipelines are not running as expected.

1.2.1 System Settings

Common issues encountered in System Settings are listed in this section.

1.2.1.1 SQL Queries Configuration

When you create a custom query from the SQL queries configuration in System Settings, to access a table to look-up, the pipeline fails, unless you create a synonym with the table name.

For example, to access a table `OSA2.REF_TBL` from user `OSA1`, create a synonym from user `OSA1` using

```
Create synonym REF_TBL for OSA2.REF_TBL
```

1.2.1.2 VM Memory Issue in Yarn Cluster

If the Ignite server is unable to start successfully on yarn cluster, check if there is a VM memory issue in yarn cluster.

To fix this issue, add below property in `yarn-site.xml` and restart the yarn cluster.

```
<property>
    <name>yarn.nodemanager.vmem-check-enabled</name>
    <value>>false</value>
</property>
```

1.2.2 Web-tier Logging

Common web-tier logging issues are listed in this section.

1.2.2.1 Change Jetty Log Level

The jetty log level is set to `Error` by default. To change the log level, change from `Error` to `Info`, in the configuration file: `<osa installation>/osa-base/resources/log4j2.xml`.

1.2.3 Pipeline

Common issues encountered while deploying pipelines are listed in this section.

1.2.3.1 Unable to Deploy more than 17 Pipelines

The default Spark setup is set to a maximum of 16 pipelines.

You can change this limit:

1. Open an SSH terminal on the GGSA instance
2. run `mysql -uroot -p`
3. Enter the password mentioned in `/home/opc/README.txt`
4. `mysql> use OSADB`
5. `mysql> insert into osa_system_property (mkey,value) values('spark.port.maxRetries','40');`

6. Restart GGSA by running `sudo systemctl restart osa`

Ensure that the environment has sufficient core and memory resources in Spark for the required number of pipelines.

1.2.3.2 Pipelines are not running as expected

If a pipeline is not running as expected, verify the following:

Ensure that Pipeline is Deployed Successfully

To verify Pipeline Deployment on Apache Spark Installation based Spark Cluster:

1. Open Spark Master console user interface.
2. If you see the status as `Running`, then the pipeline is currently deployed and running successfully.

Ensure that the Input Stream is Supplying Continuous Stream of Events to the Pipeline

To check for a continuous supply of events from the input stream:

1. Go to the **Catalog**.
2. Locate and click the stream you want to troubleshoot.
3. Check the value of the **topicName** property under the **Source Type Parameters** section.
4. Since this topic is created using Kafka APIs, you cannot consume this topic with REST APIs.

Listen to the Kafka topic hosted on a standard Apache Kafka installation.

You can listen to the Kafka topic using utilities from a Kafka Installation. `kafka-console-consumer.sh` is a utility script available as part of any Kafka installation.

Follow these steps to listen to Kafka topic:

- a. Determine the Zookeeper Address from Apache Kafka Installation based Cluster.
- b. Use the following command to listen the Kafka topic:

```
./kafka-console-consumer.sh --zookeeper IPAddress:2181 --  
topicName
```

Ensure that the Output Stream is available in the Monitor Topic

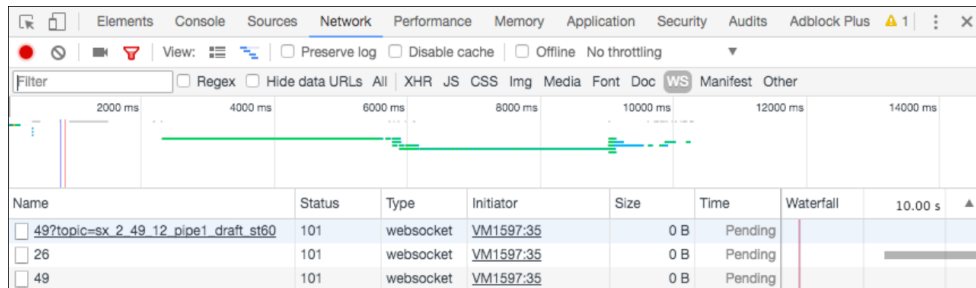
To check if the output stream is available in monitor topic:

1. Navigate to **Catalog**.
2. Open the required pipeline.
3. Ensure that you stay in pipeline editor and do not click **Done**. Otherwise the pipeline gets undeployed.
4. Right-click anywhere in the browser and select **Inspect**.
5. Go to WS tab under the **Network** tab.
6. Refresh the browser.

New websocket connections are created.

7. Locate a websocket whose URL has a parameter with the name `topic`.

The value of `topic param` is the name of Kafka Topic where the output of this stage (query or pattern) is pushed.



| Name | Status | Type | Initiator | Size | Time | Waterfall |
|--------------------------------------|--------|-----------|-----------|------|---------|-----------|
| 49?topic=sx_2_49_12_pipe1_draft_st60 | 101 | websocket | VM1597:35 | 0 B | Pending | |
| 26 | 101 | websocket | VM1597:35 | 0 B | Pending | |
| 49 | 101 | websocket | VM1597:35 | 0 B | Pending | |

The topic name is `AppName_StageId`. The pipeline name can be derived from topic name by removing the `_StageID` from topic name. In the above snapshot, the pipeline name is `sx_2_49_12_pipe1_draft`.

Ensure that Caching is Working if a Pipeline is Correlating a Stream with Reference

1. Go to Spark application master UI.
2. Open Pipeline Summary Page after clicking on Pipeline Tab. The pipeline summary page shows a table of stages with various metrics.
3. Click on the stage id corresponding to the Query stage of the pipeline in which you're correlating stream with reference.
4. In the Pipeline Stage Details page, click the **CQLDStream** stage to open CQL Engine Summary page.
5. In the CQL Engine Summary page, locate the External Sources section. Note the source id for the reference which is used in stage.
6. Open the CQL Engine Detailed Query Analysis page.
7. Click the operator that has the operator id same as source id to open CQL Engine Detailed Operator Analysis page. Click the entry corresponding to the operator.
8. Look for the Cache Statistics section. If there is no such section, then caching is not enabled for the stage. If you see the non-zero entries in Cache Hits and Cache Misses, then caching is enabled and active for the pipeline stage.

1.2.3.3 GGSA Pipeline getting Terminated

A GGSA pipeline can terminate if the targets and references used the pipeline are unreachable or resources are unavailable. You can see following exceptions on the logs:

```
com.tangosol.net.messaging.ConnectionException
```

SQLException in the Spark logs.

In case of a kafka source, republish the pipeline to read records from where it left off before terminating.

1.2.3.4 Live Table Shows `Listening Events` with No Events in the Table

There can be multiple reasons why status of pipeline has not changed to `Listening Events` from `Starting Pipeline`. Following are the steps to troubleshoot this scenario:

1. The live table shows output events of only the currently selected stage. At any time, only one stage is selected. Try switching to a different stage. If you observe output in live table for another stage, the problem can be associated with the stage. To debug further, go to step 5.
If there is no output in any stage, then move to step 2.
2. Ensure that the pipeline is still running on Spark Cluster. See [Ensure that the Pipeline is Deployed Successfully](#)
3. If the Spark application for your pipeline is killed or aborted, then it suggests that the pipeline has crashed. To troubleshoot further, you may need to look into application logs.
4. If application is in `ACCEPTED`, `NEW` or `SUBMITTED` state, then application is waiting for cluster resource and not yet started. If there are not enough resources, check the number of `VCORES` in Big Data Cloud Service Spark Yarn cluster. For a pipeline, Stream Analytics requires minimum 3 `VCORES`.
5. If application is in `RUNNING` state, use the following steps to troubleshoot further:
 - a. Ensure that the input stream is pushing events continuously to the pipeline.
 - b. If the input stream is pushing events, ensure that each of the pipeline stages is processing events and providing outputs.
 - c. If both of the above steps are verified successfully, then ensure that the pipeline is able to push the output events of each stage to its corresponding monitor topic:
 - i. Determine the monitor topic for the stage, where the output of stage is being pushed into. See [Determine the Topic Name where Output of Pipeline Stage is Propagated](#) .
 - ii. Listen to the monitor topic and ensure that the events are continuously being pushed in topic. To listen to the Kafka topic, you must have access to Kafka cluster where topic is created. You can listen to the Kafka topic using utilities from a Kafka Installation. `kafka-console-consumer.sh` is a utility script available as part of any Kafka installation.
 - iii. If you don't see any events in the topic, then this can be an issue related to writing output events from stage to monitor topic. Check the server logs and look for any exception and then report to the administrator.
 - iv. If you can see outputs events in monitor topic, then the issue can be related to reading output events in web browser.

Determine the Topic Name where Output of Pipeline Stage is Propagated

Here are the steps to find the topic name for a stage:

1. Open the Pipeline Summary Page for your pipeline. If you don't know the corresponding application name for this pipeline, see [Ensure that the Output Stream is available in the Monitor Topic](#) for instructions.

2. This page will provide the Pipeline Name and various stage ids. For every pipeline stage, you will see an entry in the table.
3. For every stage, the output topic id will be **PipelineName_StageID**.
4. Click **Done** in the pipeline editor and then go back to Catalog and open the pipeline again.

1.2.3.5 Live Table Still Shows Starting Pipeline

There can be multiple reasons why status of pipeline has not changed to `Listening Events` from `Starting Pipeline`. Following are the steps to troubleshoot this scenario:

1. Ensure that the pipeline has been successfully deployed to Spark Cluster. For more information, see [Ensure that Pipeline is Deployed Successfully](#) . Also ensure that the Spark cluster is not down and is available.
2. If the deployment failed, check the Jetty logs to see the exceptions related to the deployment failure and fix the issues.
3. If the deployment is successful, verify that OSA webtier has received the pipeline deployment from Spark.
4. Click **Done** in the pipeline editor and then go back to Catalog and open the pipeline again.

Ensure that a Pipeline Stage is Still Processing Events

To verify if a particular pipeline stage is still processing events:

1. Go to Spark application master UI.
2. Open Pipeline Summary Page after clicking on Pipeline Tab. The pipeline summary page shows a table of stages with various metrics.
3. Check if Total Output Events is a non-zero value. Refresh the page to see if the value increases or stays the same. If the value remains same and doesn't change for a long time, then drill down into stage details.

1.2.3.6 Time-out Exception in the Spark Logs when you Unpublish a Pipeline

In the Jetty log look for the following message:

```
OsaSparkMessageQueue:182 - received:  
oracle.wlevs.strex.spark.client.spi.messaging.AcknowledgeMessage Undeployment  
Ack: oracle.wlevs.strex.spark.client.spi.messaging.AcknowledgeMessage
```

During an application shutdown, a pipeline may take several minutes to unpublish completely.

So, if you do not see the above message, then you may need to increase the **osa.spark.undeploy.timeout** value accordingly.

Also in the High Availability mode, at the time of unpublishing a pipeline, the snapshot folder is deleted.

In HA mode, if you do not receive the above error message on time, and see the following error:

```
Undeployment couldn't be complete within 60000 the snapshot folder may not be  
completely cleaned.
```

it only means that the processing will not be impacted, but some disk space will be occupied.

1.2.3.7 Piling up of Queued Batches in HA mode

If a GGSA pipeline is deployed in the High Availability mode in a Spark Standalone cluster, every time there is a target or reference failure, Spark spins off new drivers. In case these targets and references are not recoverable, it results in a loop of queued up batches.

To resolve this issue, you have to unpublish the application manually.

1.2.3.8 Null Record from Summary in Query Stage

When you publish a pipeline for the first time, with a summary in one of the Query stages, the first record is null on all columns. This causes the pipeline to fail, if it has targets where key is necessary.

To solve this issue, check if there is a summary stage added before a target stage, and add a query stage with a filter checking for null values for the cache keys.

1.2.4 Stream

Common issues encountered with streams are listed in this section.

1.2.4.1 Cannot See Any Kafka Topic or a Specific Topic in the List of Topics

Use the following steps to troubleshoot:

1. Go to Catalog and select the Kafka connection which you are using to create the stream.
2. Click Next to go to Connection Details tab.
3. Click Test Connection to verify that the connection is still active.
4. Ensure that topic or topics exist in Kafka cluster. You must have access to Kafka cluster where topic is created. You can list all the Kafka topics using utilities from a Kafka Installation. `kafka-console-consumer.sh` is a utility script available as part of any Kafka installation.
5. If you can't see any topic using above command, ensure that you create the topic.
6. If the test connection failed, and you see error message like `OSA-01266 Failed to connect to the ZooKeeper server`, then the Kafka cluster is not reachable. Ensure that Kafka cluster is up and running.

1.2.4.2 Input Kafka Topic is Sending Data but No Events Seen in Live Table

This can happen if the incoming events are not adhered to the expected shape for the Stream. To check if the events are dropped due to shape mismatch, use the following steps to troubleshoot:

1. Verify if lenient parameter under Source Type Properties for the Stream is selected. If it is FALSE, then the event may have been dropped due to shape mismatch. To confirm this, check the application logs for the running application.
2. If the property is set to TRUE, debug further:

- a. Make sure that Kafka Cluster is up and running. Spark cluster should be able to access Kafka cluster.
- b. If Kafka cluster is up and running, obtain the application logs for further debugging.

1.2.5 Connection

Common issues encountered with connections are listed in this section.

1.2.5.1 Database Connection Failure

To test a Database connection:

1. From the **Catalog** page, select the database connection that you want to test.
2. Click **Next**.
3. On the **Connection Details** tab, click **Test Connection**.
 - If the test is successful, it indicates that the connection is active.
 - If the test fails, the following error messages are displayed:
 - OSA-01260 Failed to connect to the database. IO Error: The Network Adapter could not establish the connection: This error message indicates that the DB host is not reachable from GGSA design time.
 - OSA-01260 Failed to connect to the database. ORA-01017: invalid username/password; logon denied: This error message indicates that your login credentials are incorrect.

1.2.5.2 Druid Connection Failure

To test a Druid connection:

1. From the **Catalog** page, select the Druid connection that you want to test.
2. Click **Next**.
3. On the **Connection Details** tab, click **Test Connection**.
 - If the test is successful, it indicates that the connection is active.
 - If the test fails, the following error message is displayed:

OSA-01460 Failed to connect to the druid services at zooKeeper server:
This error indicates that the druid zookeeper is not reachable. Ensure that the druid services and zookeeper cluster are up and running.

1.2.5.3 Coherence Connection Failure

GGSA does not provide a **Test connection** option for a coherence cluster. Refer to Oracle Coherence documentation to find utilities and tools to test a coherence connection.

1.2.5.4 JNDI Connection Failure

To test a JNDI connection:

1. From the **Catalog** page, select the JNDI connection that you want to test.
2. Click **Next**.

3. On the **Connection Details** tab, click **Test Connection**.
 - If the test is successful, it indicates that the connection is active.
 - If the test fails, the following error messages are displayed:
 - OSA-01707 Communication with server failed. Ensure that server is up and server url(s) are specified correctly: **This error indicates that either the server is down or server url(s) is incorrectly specified. Server url should be of the format host1:port1,host2:port2.**
 - OSA-01706 JNDI connection failed. User: weblogic, failed to be authenticated: **This error indicates that the login credentials are incorrect.**

1.2.6 Target

Common issues encountered with targets are listed in this section.

1.2.6.1 Cannot see any Events in Targets

If the pipeline is in the draft mode, it cannot push events to targets. Only published pipelines can push events to targets.

1.2.7 Geofence

Common issues encountered with geofences are listed in this section.

1.2.7.1 Name and Description Fields are not displayed for the DB-based Geofences

If name and description fields are not displayed for database-based geofence, ensure to follow steps mentioned below:

1. Go to Catalog and click Edit for the required database-based geofence.
2. Click Edit for Source Type Properties and then Next.
3. Ensure that the mapping for Name and Description is defined in Shape section.
4. Once these mappings are defined, you can see the name and description for the geofence.

1.2.7.2 DB-based Geofence is not Working

To ensure that a database-based geofence is working:

1. Go to Catalog and open the required database-based geofence.
2. Ensure that the connection used in geofence is active by clicking test button in database connection wizard.
3. Ensure that table used in geofence is still valid and exists in DB.
4. Go to the geofence page and verify that the issue is resolved.

1.2.8 Cube

Common issues encountered with cubes are listed in this section.

1.2.8.1 Unable to Explore Cube which was Working Earlier

If you are unable to explore a cube which was working earlier, follow the steps mentioned below:

1. Check if the druid zookeeper or the associate services for indexer, broker, middle manager or overlord is down.
2. Click the Druid connection and navigate to next page.
3. Test the connection. This step will tell you if the services are down and need to be looked into.

1.2.8.2 Cube Displays "Datasource not Ready"

If you keep seeing "Datasource not Ready" message when you explore a cube, follow the steps mentioned below:

1. Go to the druid indexer logs. Generally, it is `http://DRUID_HOST:3090/console.html`.
2. Look for entry in running tasks `index_kafka_<cube-name>_<somehash>`. If there is no entry in running tasks, look for the same entry in pending tasks or completed tasks.
3. If the entry lies in pending tasks, it means that workers are running out of capacity and datasource will get picked for indexing as soon as it's available.
4. In such cases, either wait OR increase the worker capacity and restart druid services OR kill some existing datasource indexing tasks (they will get started again after sometime).
5. If the entry lies in completed tasks with "FAILED" status, it means that indexing failed either due to incorrect ingestion spec or due to resource issue.
6. You can find the exact reason by clicking "log (all)" link and navigating to the exception.
7. If it is due to ingestion, try changing the timestamp format. (Druid fails to index, if the timestamp is not in JODA timeformat OR if the timeformat specified does not match with format of timestamp value).

1.2.9 Dashboard

Common issues encountered with dashboards are listed in this section.

1.2.9.1 Visualizations Appearing Earlier are No Longer Available in Dashboard

Use the following steps to troubleshoot:

1. For missing streaming visualization, it might be due to the following reasons:
 - a. Corresponding pipeline/stage for the missing visualizations no longer exists
 - b. The visualization itself is removed from catalog or the pipeline editor
2. For missing exploration visualization (created from cube), it might happen as cube or visualization might have been deleted already.

1.2.9.2 Dashboard Layout Reset after You Resized/moved the Visualizations

Use the following steps to troubleshoot:

1. This might happen, if you forget to save the dashboard after movement/ resizing of visualizations.
2. Make sure to click Save after changing the layout.

1.2.9.3 Streaming Visualizations Do not Show Any Data

Use the following steps to troubleshoot:

1. Go to visualization in pipeline editor and make sure that live output table is displaying data.
2. If there is no output, ensure that the pipeline is deployed and running on cluster. Once you have the live output table displaying data, it shows up on the streaming visualization.

1.2.10 Live Output

Common issues encountered with live output are listed in this section.

1.2.10.1 Issues with Live Output

For every pipeline, there will be one Spark streaming pipeline running on Spark Cluster. If a Stream Analytics pipeline uses one or more Query Stage or Pattern Stage, then the pipeline will run one or more continuous query for each of these stages.

For more information about continuous query, see [Understanding Oracle CQL](#).

If there are no output events in Live Output Table for Query Stage or Pattern Stage, use the following steps to determine or narrow down the problem:

- [Ensure that the Pipeline is Deployed Successfully](#)
- [Ensure that the input Stream is Supplying Continuous Stream of Events to the Pipeline](#)
- [Ensure that CQL Queries for each query stage emit output](#)
- [Ensure that the output of stage is available](#)

Ensure that CQL Queries for Each Query Stage Emit Output

To check if the CQL queries are emitting output events to monitor CQL Queries using CQL Engine Metrics:

1. Open CQL Engine Query Details page. For more information, see [Access CQL Engine Metrics](#).
2. Check that at least one partition has **Total Output Events** greater than zero under the **Execution Statistics** section.

Spark 1.6.0 CQL Engine Query Details

Query ID: sx_1_38_5_a1_draft1
 Query Text: ISTREAM (SELECT client_age AS client_age, count(purchase_sum) AS COUNT_of_purchase_sum FROM sx_1_38_5_a1_draft_2 [range 1 HOURS slide 1 NANOSECONDS] AS S1 GROUP BY client_age)
 Num Partitions: 7

Execution Statistics

| Partition ID | CQL Engine ID | Total Output Events | Total Output Heartbeats | Throughput(events/sec) | Latency(ms) |
|--------------|---------------|---------------------|-------------------------|------------------------|-------------|
| 0 | 3 | 262 | 2046 | 351 | 2 |
| 1 | 1 | 325 | 2108 | 704 | 1 |
| 2 | 3 | 271 | 2055 | 363 | 2 |
| 3 | 4 | 228 | 2013 | 373 | 2 |
| 4 | 2 | 226 | 2011 | 328 | 3 |
| 5 | 2 | 199 | 1984 | 289 | 3 |
| 6 | 4 | 283 | 2086 | 463 | 2 |

HA Statistics

| Partition ID | CQL Engine ID | Total FullSnapshots Created | Avg FullSnapshot CreationTime(ms) | Total FullSnapshots Loaded | Avg FullSnapshot LoadTime(ms) | Total JournalSnapshots Created | Avg JournalSnapshot CreationTime(ms) | Total JournalSnapshots Loaded | Avg JournalSnapshot LoadTime(ms) |
|--------------|---------------|-----------------------------|-----------------------------------|----------------------------|-------------------------------|--------------------------------|--------------------------------------|-------------------------------|----------------------------------|
| 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

If your query is running without any error and input data is continuously coming, then the **Total Output Events** will keep rising.

Ensure that the Output of Stage is Available

1. Ensure that you stay in the Pipeline Editor and do not click **Done**. Else, the pipeline gets undeployed.
2. Right-click anywhere in the browser and click **Inspect**.
3. Select **Network** from the top tab and then select **WS**.
4. Refresh the browser.

New websocket connections are created.

5. Locate a websocket whose URL has a parameter with name `topic`.

The value of the topic param is the name of the Kafka topic where the output of this stage is pushed.

| Name | Status | Type | Initiator | Size | Time | Waterfall |
|--------------------------------------|--------|-----------|-----------|------|---------|-----------|
| 49?topic=sx_2_49_12_pipe1_draft_st60 | 101 | websocket | VM1597:35 | 0 B | Pending | |
| 26 | 101 | websocket | VM1597:35 | 0 B | Pending | |
| 49 | 101 | websocket | VM1597:35 | 0 B | Pending | |

6. Listen to the Kafka topic where output of the stage is being pushed.

Since this topic is created using Kafka APIs, you cannot consume this topic with REST APIs. Follow these steps to listen to the Kafka topic:

- a. Listen to the Kafka topic hosted on a standard Apache Kafka installation.

You can listen to the Kafka topic using utilities from a Kafka Installation. `kafka-console-consumer.sh` is a utility script available as part of any Kafka installation.

To listen to Kafka topic:

- i. Determine the Zookeeper Address from Apache Kafka Installation based Cluster.

- ii. Use following command to listen the Kafka topic:

```
./kafka-console-consumer.sh --zookeeper IPAddress:2181 --  
topic sx_2_49_12_pipeline_draft_st60
```

1.2.10.2 Missing Events due to Faulty Data

If the CQLEngine encounters faulty data in user defined functions, the exceptions of the events with faulty data are logged in executor logs, and the processing continues uninterrupted.

Sample logs of the dropped events and exceptions:

```
20/04/02 14:41:42 ERROR spark: Fault in CQL query processing.  
Detailed Fault Information [Exception=user defined  
function(oracle.cep.extensibility.functions.builtin.math.Sqrt@74a5e306)
```

```
runtime error while execution,  
Service-Name=SparkCQLProcessor, Context=phyOpt:1;  
queries:sx_SquareRootPipeline_osaadmin_draft1
```

```
20/04/02 14:41:42 ERROR spark: Continue on exception by dropping  
faulty event.
```

```
20/04/02 14:41:42 ERROR spark: Dropped event details  
<TupleValue><ObjectName>sx_SquareRootPipeline_osaadmin_draft_1</  
ObjectName><Timestamp>1585838502000000000</Timestamp>  
<TupleKind>PLUS</TupleKind><IntAttribute  
name="squareNumber"><Value>-2</Value></  
IntAttribute><IsTotalOrderGuarantee>true</IsTotalOrderGuarantee></  
TupleValue>:
```

1.2.11 Pipeline Deployment Failure

Sometimes pipeline deployment fails with the following exception:

```
Spark pipeline did not start successfully after 60000 ms.
```

This exception usually occurs when you do not have free resources on your cluster.

Workaround:

Use external Spark cluster or get better machine and configure the cluster with more resources.