Oracle® Financial Services

Compliance Studio Architecture Guide

Release 8.1.2.9.0
G23340-03
June 2025

Oracle Financial Services Compliance Studio Architecture Guide, Release 8.1.2.9.0

G23340-03

Contents

Document Control

The following table lists the document control of this guide.

Table Document Control

Version Number	Revision Date	Change Log
8.1.2.9.0	April 2025	There is no feature update in this version.
8.1.2.8.0	August 2024	There is no feature update in this version.

Preface

This preface provides information on the Oracle Financial Services (OFS) Compliance Studio Architecture Guide.

Audience

Oracle Financial Services Compliance Studio Architecture Guide is intended for implementation consultants and administrators who can view the high-level architecture of the Compliance Studio solution.

Related Documents

This section identifies additional resources to the OFS Compliance Studio. You can access additional documents from the Oracle Help Center.

Abbreviations

The following table lists the abbreviations used in this document.

Table 1 Abbreviations

Abbreviation	Meaning
OFS	Oracle Financial Services
OFSAAI	Oracle Financial Services Analytical Applications Infrastructure
OHC	Oracle Help Center
MOS	My Oracle Support
OFSAA	Oracle Financial Services Analytical Application
BD	Behavior Detection
FCDM	Financial Crime Data Model
MMG	Model Management and Governance
SSO	Single Sign-On
SSH	Secure Shell
OOB	Out of the Box
PGX	Parallel Graph Analytics
AML	Anti-MoneyLaundering
ML	Machine Learning
ML4AML	Machine Learning for AML
ORE	Oracle R Enterprise
SAML	Security Assertion Markup Language
AAI	Advanced Analytics Infrastructure
HTTP	Hypertext Transfer Protocol

Table 1 (Cont.) Abbreviations

Abbreviation	Meaning
HTTPS	HTTPover SSL or HTTP Secure
SSL	SecureSocket Layer
TLS	Transport Layer Security
ETL	Extract, Transform and Load
SSH	Secure Shell Protocol
UI	User Interface
IDP	Identity Provider
REST	Representational State Transfer
GER	Global Entity Resolution
LDAP	Lightweight Directory Access Protocol
SID	System Identifier
REPL	Read Eval Print Loop

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 customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

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.
italic	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.

Comments and Suggestions

Please give us feedback about Oracle Applications Help and guides! You can send an e-mail to: https://support.oracle.com/portal/.

1

OFS Compliance Studio Architecture

This section focuses on the following architecture, components, use cases and when to use this guide.

Overview

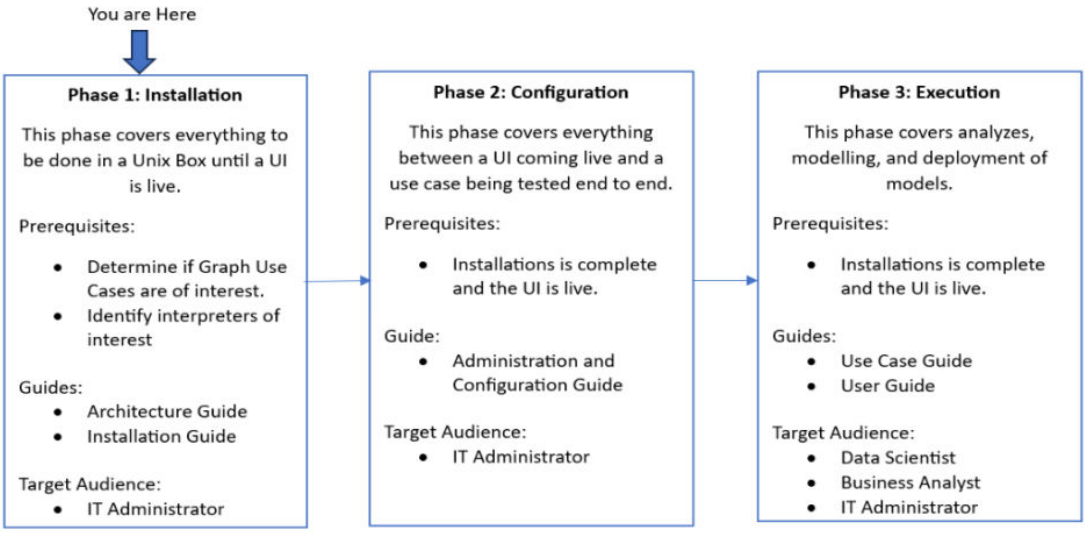
OFS Compliance Studio is an advanced analytics application that supercharges anti-financial crime programs for better customer due diligence, transaction monitoring, and investigations by leveraging the latest innovations in artificial intelligence, open-source technologies, and data management.

It combines Oracle's Parallel Graph Analytics (PGX), Machine Learning for AML, Entity Resolution, notebook-based code development, and enabling Contextual Investigations in one platform with complete and robust model management and governance functionality.

When to Use this Guide

The following illustration demonstrates when this guide should be used.

Figure 1-1 When to Use this Guide



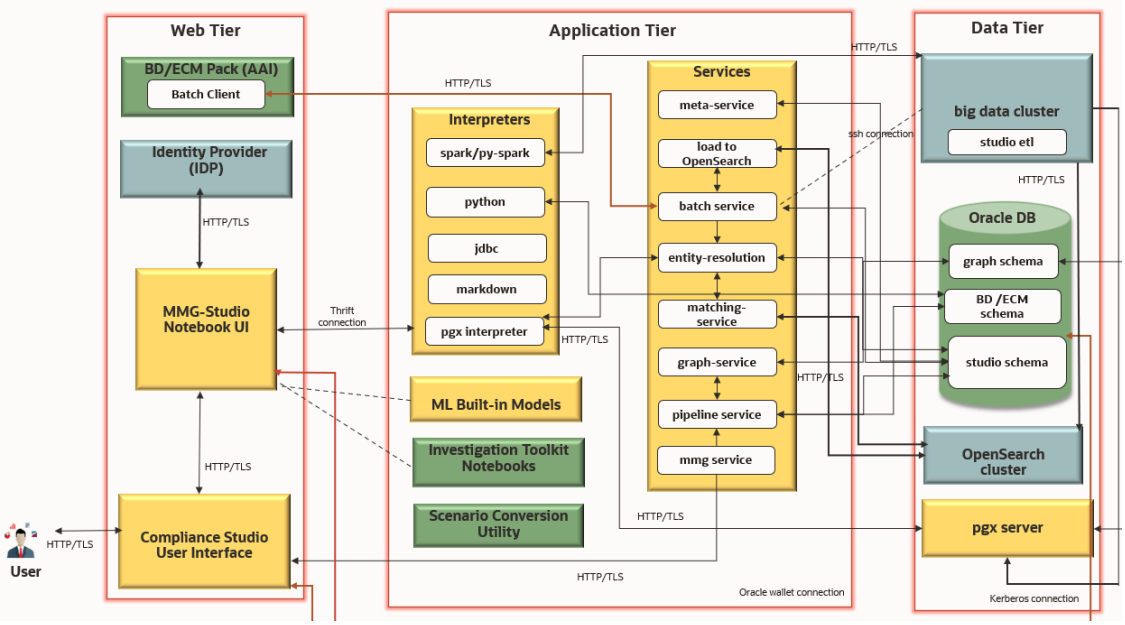
1.1 Architecture Overview

This section provides the architecture details of the Compliance Studio.

Native Architecture

The following diagram exhibits complete architecture of the Compliance Studio.

Figure 1-2 Compliance Studio Architecture

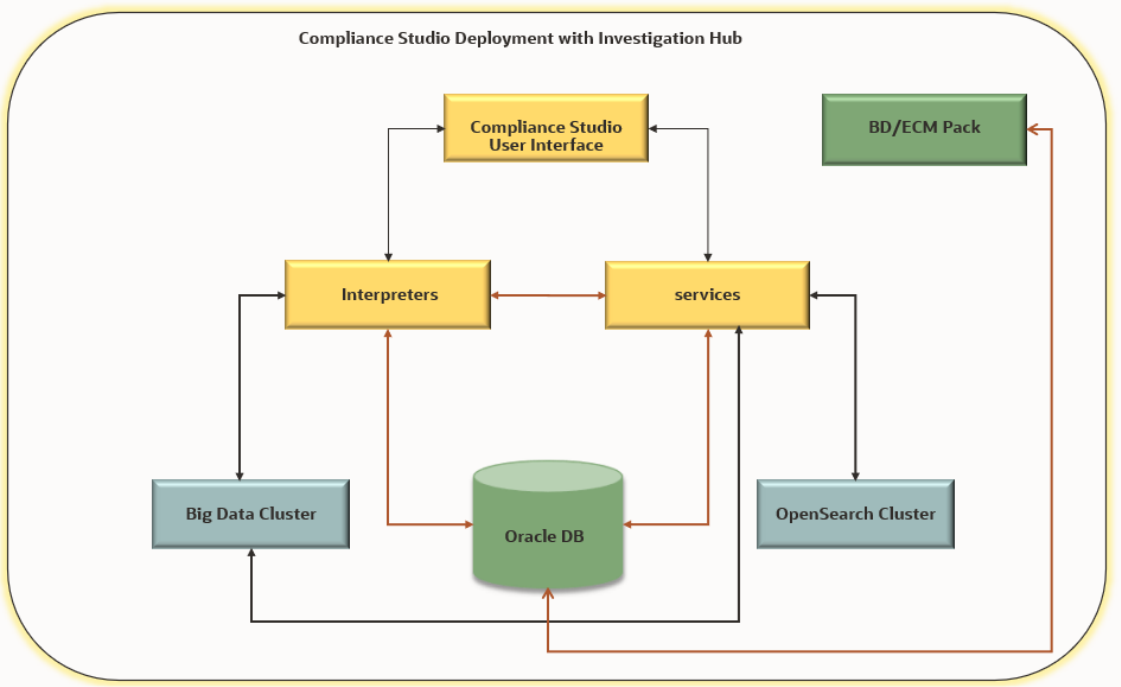


> **Note:**
>
> - Compliance Studio components (indicated in the yellow color) are deployed on the same server.
> - PGX Server can be deployed on the same server or other server based on Graph Sizing requirement.

Simplify Architecture

The following diagram exhibits simplified architecture of the Compliance Studio.

Figure 1-3 Simplify Architecture



1.2 Components

This section provides a list of key components and third-party components.

1.2.1 Key Components

The following components are bundled in the OFS Compliance Studio Installer:

- OFS Compliance Studio Front End Service
  - Compliance Studio UI
  - Notebook Server UI
- OFS Compliance Studio Back End Service
  - Interpreters
  - Services
  - MMG Service
- ML4AML Models
- Python
- Parallel Graph Analytics Server

1.2.2 Other Oracle Components

The Other Oracle Components are:

- Behavior Detection (BD) Pack

- Enterprise Case Management (ECM) Pack
- Investigation Toolkit
- Scenario Conversion Utility
- Oracle DB

1.2.3 Third-party Components

The Third-party components are:

- OpenSearch Cluster
- Identity Provider (IDP)

1.3 Component Details

The following table lists the component details.

Table 1-1 Component Details

Component/Service	Details
Compliance Studio UI	You can access the Compliance Studio UI via browser and enter the login credentials along with the language. For valid login credentials, it navigates to the Workspace Summary page.
Notebook Server UI	You can access Notebook Server UI through the Compliance Studio UI.
Spark Interpreter	You can connect to a big data cluster and create the models to perform analytics on data present in the Big data clusters.
Python Interpreter	You can create/execute the Python models using this Interpreter. Analytics can be done with any python library. By default, python interpreters are configured with predefined Conda environments as follows: • default_8.1.2.9.0 • ml4aml_8.1.2.9.0 • sane_8.1.2.9.0 For more information, see the OFS Compliance Studio Administration andConfiguration Guide.

Table 1-1 (Cont.) Component Details

Component/Service	Details
JDBC Interpreter	You can create/execute the SQL models using this Interpreter. By default, this is connecting to Studio schema. You can connect to any schema by changing the interpreter configuration. For example, BD or ECM schema. **NOTE:** This feature is not recommended approach because it can only be used to connect to a single schema, and all users will have access to that, rather than access being managed per user. In future releases this interpreter will not be enabled by default but instructions will be given to enable if required. **Limitation** • Data source configuration is not dynamic; instead, it is static from the Interpreter Configuration screen. • There is no restriction or secure access of data provided with this interpreter. **Recommendation** Users are recommended to use a python interpreter to get dynamic data source configuration; even data access permission features can also be used with this interpreter.
PGX Interpreter	• **pgx-java**: Java-based Interpreter, you can create/execute Java-based models and interact with the PGX server for graph analytics. • **pgql**:SQL is like an interpreter to query on the graph. • **pgx-python**:python based Interpreter with a PGX python client embedded in it to query on graph present in the PGX server. • **pgx-algorithm**:Graph toolkit that provides a graph query language and optimized analytics algorithms.
Meta Service	This service is responsible for setting up metadata related to Compliance Studio in Studio Schema.
Load to OpenSearch	This service manages OpenSearch indexes used to resolve the entity based on the matching rules.
Batch Service	This service is responsible for executing some of the batch jobs of Compliance Studio. For example, ETL for graph analytics or entity resolution.
Entity Resolution	It is responsible for resolving entities using matching and merging rules. • **Graph ER**: It creates Similarity Edges between nodes by comparing the attributes of the nodes and identifying where the similarity is significant enough to create an edge so the nodes are linked with the graph model and can be analyzed as a single entity. • **Global Party ER**: It creates a Global Party of similar entities by comparing multiple attributes of entities using matching and merging rules. For more information on merging and matching rules, see OFS ComplianceStudio Matching Guide.

Table 1-1 (Cont.) Component Details

Component/Service	Details
Matching Service	It is responsible for scoring in ER based on matching rules. It has the following scoring methods: • Jaro-Winkler • ML-BoostedName • ML-BoostedAddress • Levenshtein • Individual • Name • Entity • Name • Default For more information on merging and matching rules, see the OFS Compliance Studio Matching Guide.
Graph Service	This service is used for managing graphs in Compliance Studio.
Pipeline Service	This service is used for extract transform and load data into target tables.
ML Model Templates	The prepackaged Model templates allow you to perform the following: • Model segmentation (model grouping) • Load and Preview data • User-defined transformation (deriving additional features) • EDA • Feature selections • Model training • Evaluation • Model Agnostics (Explainability) • On-going validations
Python	Python contains all packages required to execute ML4AML models. For example, scikit-learn pandas.
MMG Service	This service is used to manage the following functions: • Work spaces and sandbox • Data sources (external, local file, relational, and distributed) • Model complete life cycle, governance, and execution • Batch and Scheduler services • User roles and accesses • User Provisioning and authentication
Parallel Graph Analytics Server	Graph analysis lets you reveal latent information that is not directly apparent from fields in your data but is encoded as direct and indirect relationships - metadata - between elements of your data. This connectivity-related information is not apparent to the naked eye but can have tremendous value when uncovered. PGX is a toolkit. For graph analysis, It supports both efficient graph algorithms and fast SQL- like graph pattern matching queries. FCGM is loaded into the PGX server for analytics.
BDPACK	In Compliance Studio, the graph model is based on the BD Pack's FCDM model and ML4AML using the same data model. For more information, see the Behavior Detection Application Pack.

Table 1-1 (Cont.) Component Details

Component/Service	Details
ECMPACK	In Compliance Studio, the graph model is based on the ECM Pack's FCDM model.ECM is also used to correlate events generated via Compliance Studio and for case investigation. For more information, see the EnterpriseCase Management ApplicationPack.
Oracle DB	Compliance Studio stores the metadata in the Oracle DB.
Investigation Toolkit	OFS Investigation Toolkit is an application built on Compliance Studio, allowing investigators to view the case and adhoc information within, then creates case narratives and insights, highlight risk factors and red flags meaningful to the investigation, and recommend actions based on graph scoring algorithms. For more information, see the Investigation Toolkit.
Scenario Conversion Utility	This utility converts the Behavior Detection scenario into Compliance Studio scenario.
Identity Provider	Identity Provider (IDP) is required for SSO/SAML authentication.
OpenSearch Cluster	An OpenSearch cluster is a group of nodes that have the same cluster name attribute. As nodes join or leave a cluster, they reorganize to evenly distribute the data across the available nodes. If you are running a single instance of OpenSearch, you have a cluster of one node. It is used for a matching service engine used for Entity Resolution and Similarity Edge for Graph Nodes.

1.4 Communication Details

The following table lists the Communication details.

Table 1-2 Communication Details

Connection/Interface	Details
HTTP	Hypertext Transfer Protocol (HTTP) is a communication protocol in the application.
HTTPS	HTTPS (HTTP over SSL or HTTP Secure) uses a Secure Socket Layer (SSL), a secure protocol that works on top of HTTP to provide security. That means SSL encrypted data will be routed using protocols like HTTP for communication.
TLS	Transport Layer Security (TLS) encrypts data for private and sensitive information such as passwords, credit card numbers, and personal correspondence in the application.
Thrift Connection	Thrift supports clean abstractions and implementations for data transport, data serialization, and application-level processing.
Oracle Wallet Connection	Oracle Wallet is a file that stores database authentication and signing credentials. It allows users to securely access databases without providing credentials to third party software and quickly connects to Oracle products.
SSH Connection	Secure Shell Protocol (SSH) hosts multiple channels simultaneously and transfers data in both directions.

1.5 Application Deployment

A separate installer is provided for the On-premise deployment.

For more installation information, you can see the respective OFS Compliance Studio Installation Guide.

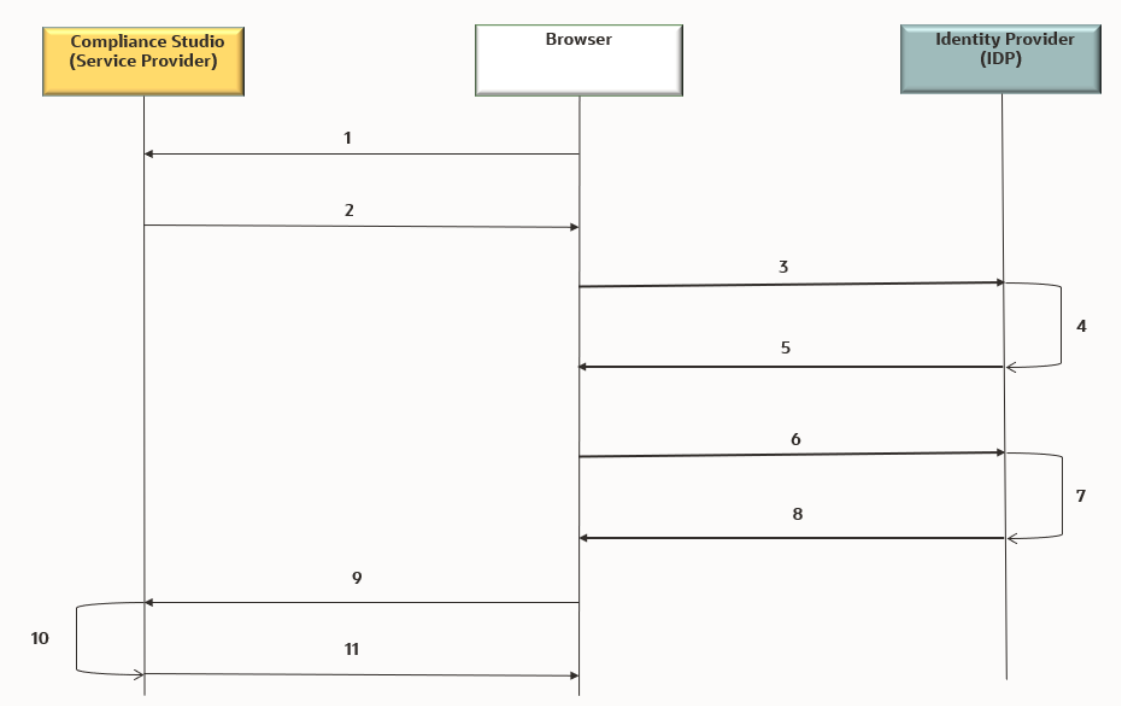
1.6 Application Authentication

This section provides the authentication details.

1.6.1 SSO/SAML

Single Sign-On (SSO)/Security Assertion Markup Language (SAML) is a type of authentication supporting the OFS Compliance Studio. It is an open standard for exchanging authentication and authorization between the user and the Compliance Studio Application, such as login, authentication state, identifiers, and other relevant attributes.

Figure 1-4 SAML Authentication Process



The entities are:

- End-User
- OFS Compliance Studio Application
- SAML

The SAML authentication process is as follows:

1. A user sends a request to access the OFS Compliance Studio Application.
2. The application redirects the request to IDP for authentication with SAML request:
3. The application sends the request to IDP for the SSO login page.
4. IDP validates the SAML request for the login page.
5. IDP sends the response to the user with the SSO login page.
6. The user enters the credentials on the SSO login page.
7. IDP validates the credentials and generates the SAML response.
8. IDP sends the SAML response is as follows:
   - For valid credentials, it sends the response to the application for validating the SAML response.
   - For invalid credentials, it displays an authentication error.
9. It posts SAML response to Assertion Consumer URL for valid credentials.
10. The application verifies the user signature in the SAML response.
11. The application displays the OFS Compliance Studio home page to the user.

1.6.2 OFSAAI

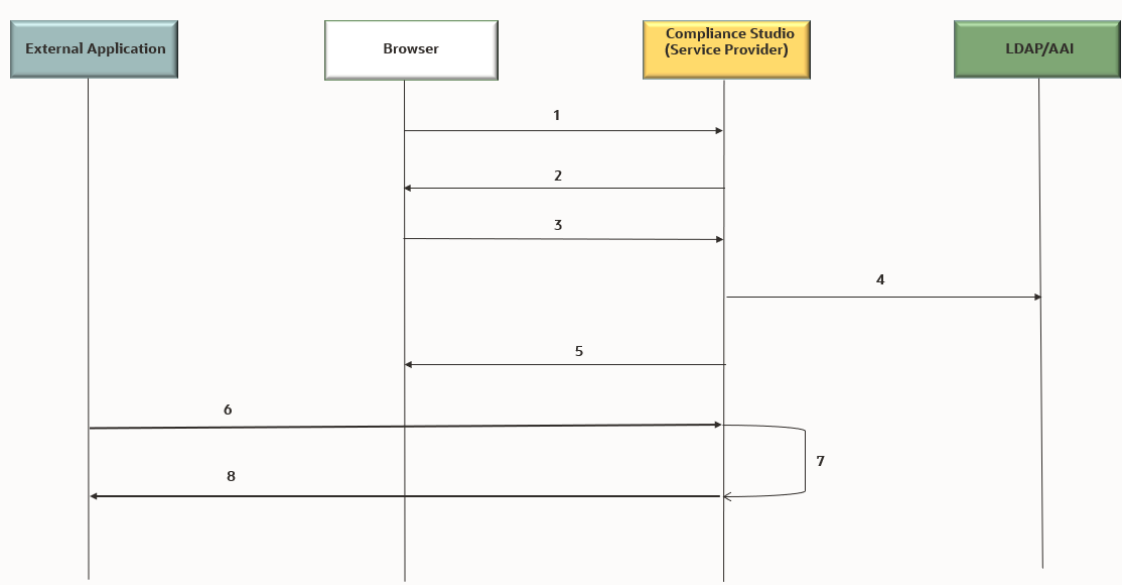
Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) authenticates users using any web browser with a user name/password to login into the application. It is also possible to restrict access to content and services based on user attributes or, conversely, make them accessible internationally.

You can authenticate the OFS Compliance Studio with the following:

- Existing OFSAAI
- Install OFSAAI and authenticate

OFSAAI is available with a pre-installed BD Pack or ECM Pack.

Figure 1-5 OFSAAI Authentication Process



The entities are:

- End-User
- OFS Compliance Studio Application
- AAI
- External Application

The AAI authentication process is as follows:

1. A user sends a request to access the OFS Compliance Studio application.
2. The application displays the OFS Compliance Studio application login page:
3. The user enters the credentials on the login page.
4. The application sends the request to AAI for validation.
5. AAI validates the credentials:
    a. For valid credentials, it displays the OFS Compliance Studio home page to the user.
    b. For invalid credentials, it displays an authentication error.
6. The External Application sends the request with Bearer/Basic token to access the application through REST API.
7. The application validates the Authorization Header using Pre-Filters.
8. The application sends the response to the External Application.
   REST API: Representational State Transfer (REST) is a software architectural style that defines a set of constraints to create Web services. Web services that conform to the REST architectural style, called RESTful Web services, provide interoperability between computer systems on the internet.

1.7 Use Cases

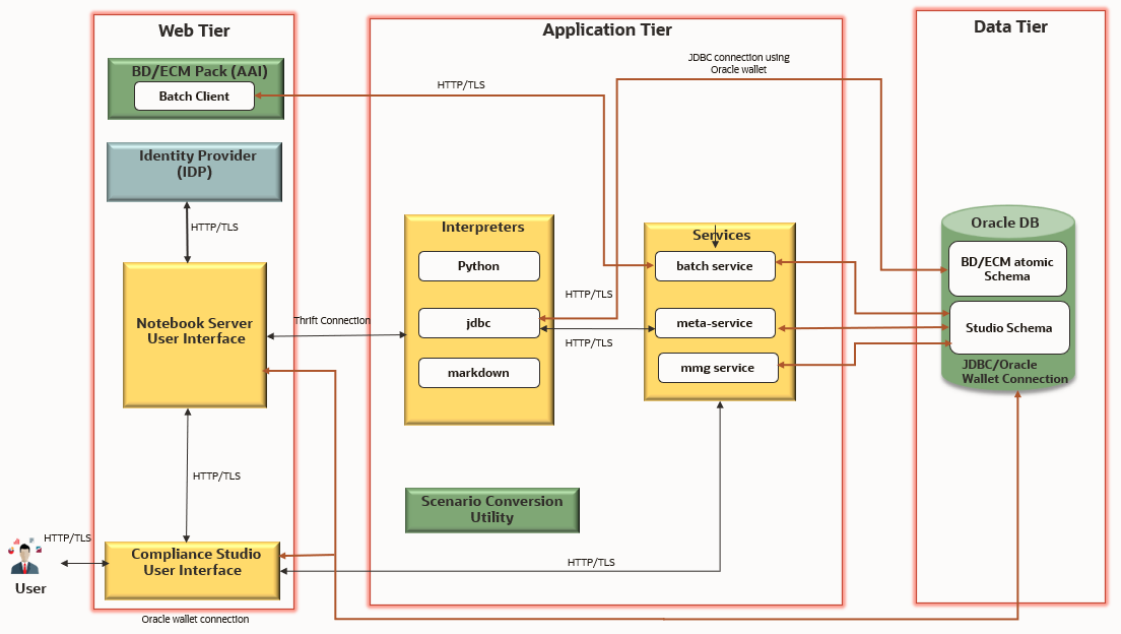
This section provides different types of Use Cases available in the Compliance Studio.

1.7.1 Scenario Authoring

OFS Compliance Studio supports Polyglot Scenario Authoring to author new scenarios in various languages like SQL, Scala, Python, and R.

It is used with Oracle's Behavior Detection or other FCC product. There are pre-built integrations for Scenario Authoring and creating events, posting them into our Enterprise Case Management system, and further creating cases for investigation. However, Compliance Studio can be used with any financial crime platform for Scenario Authoring.

Figure 1-6 Scenario Authoring



The following components are involved in this use case:

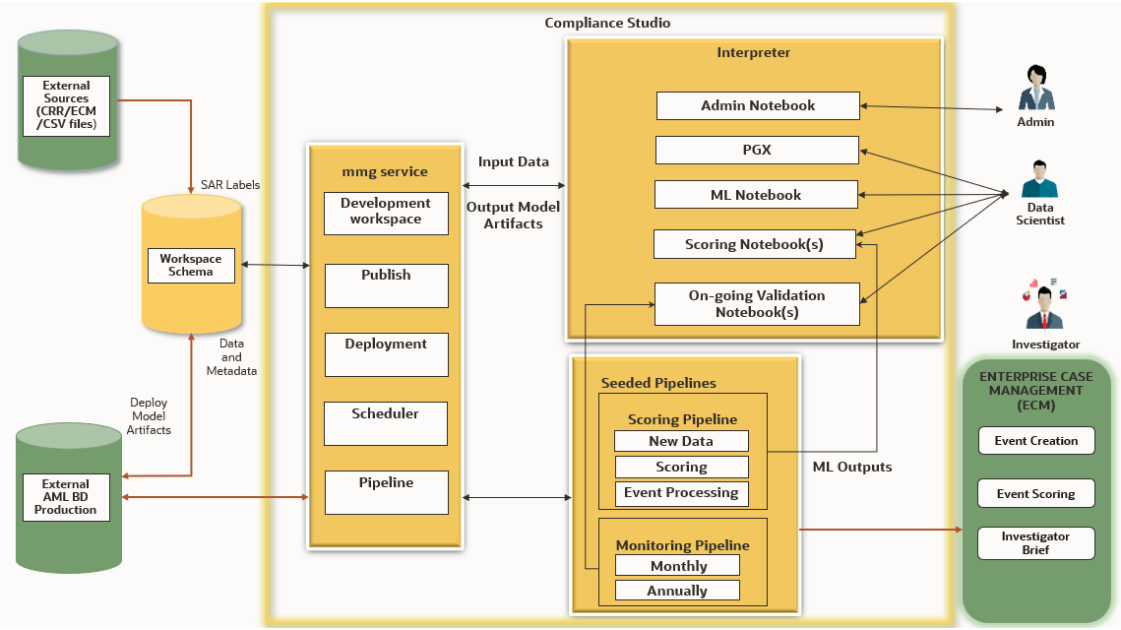
- OFS Compliance Front End Service
- OFS Compliance Back End Service
- IDP
- ECM/BD Pack
- Oracle DB
- Scenario Conversion Utility

For more information on each component, see the Component Details section.

1.7.2 Machine Learning for AML

Compliance Studio supports Machine Learning for AML (ML4AML). It is collection of use cases. For more information about use cases, see the OFS Compliance Studio Use Case Guide.

Figure 1-7 ML4AML



The following components are involved in this use case:

- OFS Compliance Front End Service
- OFS Compliance Back End Service
- Database- External sources (ECM/CRR CSV file)/AML BD production
- ECM

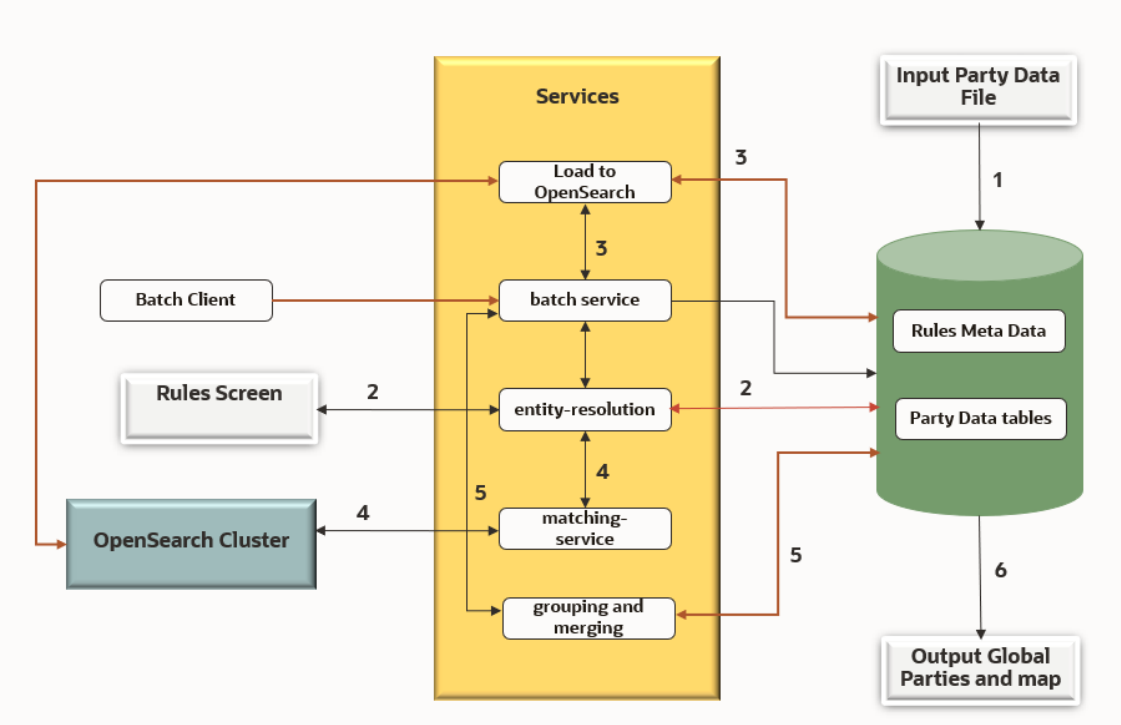
For more information on each component, see the Component Details section.

1.7.3 Entity Resolution

OFS Compliance Studio supports Entity Resolution. It allows firms to break through barriers in their data by gaining single views of their customers and their external entities and have the choice of monitoring them both under one consolidated Global Party.

Entity Resolution leverages ideas and concepts from entity resolution, machine learning, and graph analytics to resolve parties across vast datasets where customers, to avoid detection, may misidentify parties due to segmented business processes or malicious attempts. The new features allow firms to have rich visualization around complex networks and truly gain an entity view across varied datasets. This new clear customer view also can be weaponized within AML detection systems by using this resolved data to drive down false positives and ensure entities are being monitored holistically.

Figure 1-8 Entity Resolution



The following are reference points for the above image:

1. Load Input Data
2. Input Rules
3. Create and Load Index
4. Match and generate similarities
5. Group and merge based on similarities
6. Persist Global parties in the file system

The following components are involved in this use case:

- OFS Compliance Back End Service
- ECM/BD Pack
- Oracle DB
- OpenSearch Cluster

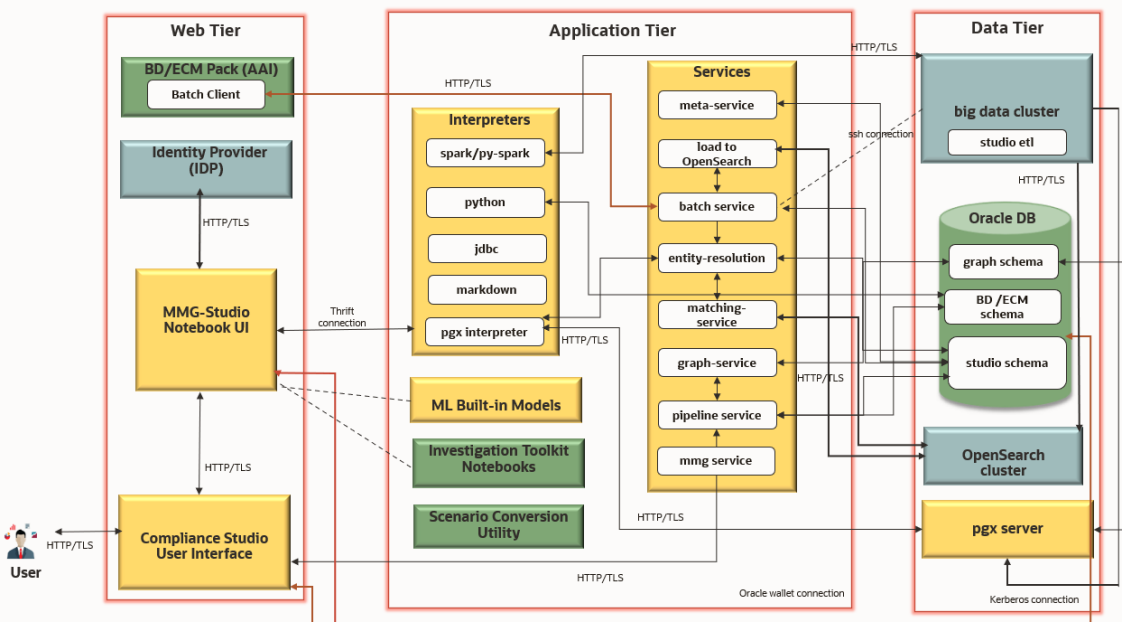
For more information on each component, see the Component Details section.

1.7.4 Investigation Toolkit

OFS Investigation Toolkit is an application built on Compliance Studio, allowing investigators to view the case and adhoc information within the FCGM rapidly. The in-built scoring, matching, and correlation engines create meaningful investigation units, and pre-configured red flags and risk factors target investigative efforts effectively. The FCGM on which it is built accelerates investigations by bringing relevant information sources together, preventing the need for the manual collation of information from disparate sources for adhoc investigations. OFS IH

automatically generates case narratives and insights, highlights risk factors and red flags meaningful to the investigation, and recommends actions based on graph scoring algorithms.

Figure 1-9 Investigation Toolkit



The following components are involved in this use case:

- OFS Compliance Front End Service
- OFS Compliance Back End Service
- IDP
- ECM/BD Pack
- Oracle DB
- OpenSearch
- PGX
- BIG Data Cluster
- Investigation Toolkit
- Scenario Conversion Utility

For more information on each component, see the Component Details section.

2

High Availability Configuration

The High Availability (HA) architecture is one of the key requirements for any Enterprise Deployment. It refers to the ability of users to access a system without loss of service.

HA preparation is an integral part of contingency planning. This document serves as a reference document for the preparation of specific High Availability (HA) architecture. It explains how a standard Compliance Studio deployment should be architected to protect it from unplanned downtime and minimize planned downtime.

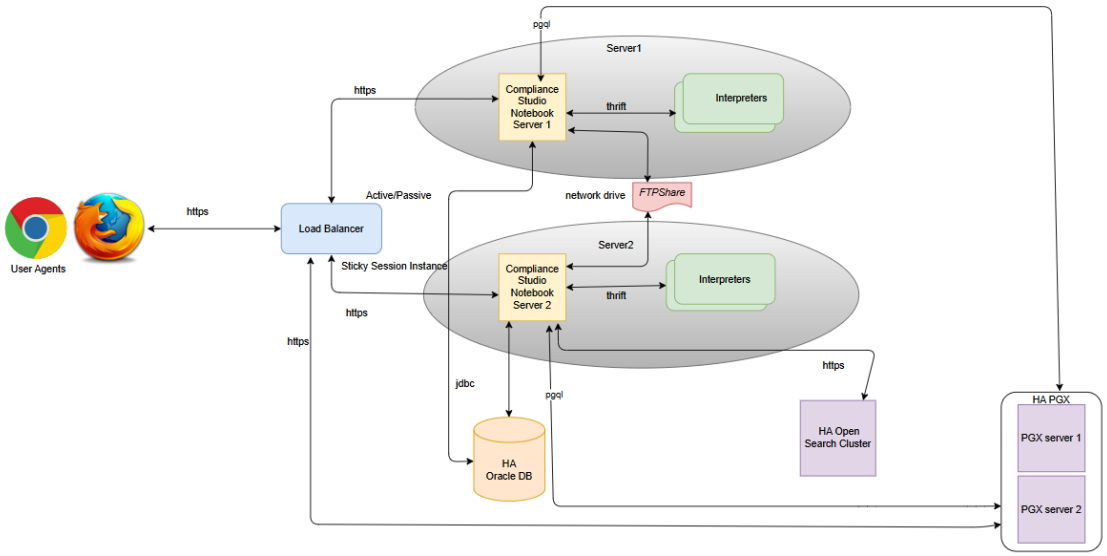
2.1 High Availability Architecture

The high availability of Compliance Studio is currently supported in an ACTIVE-PASSIVE mode where only one of the nodes is up at any given time. The HA configuration will need to hit the primary node at all times. When the primary node fails, it requires the secondary node to be fired up manually (the additional manual steps are required, see the section). The manual steps are minimal, and the node can be made active with minimum application downtime.

> **Note:**
>
> The Architecture was validated with HA Proxy but other Load Balancer could also be used if needed assuming the same capabilities.

The following illustration shows the HA architecture of the Compliance Studio for active-passive mode.

Figure 2-1 High Availability Architecture



2.2 Assumptions

- HAProxy is installed in the load balancer server.
- The load balancer, Primary and Secondary Compliance Studio instances should be on the different servers.
- The same Schema (Studio, ER/FSDF, Graph, and Atomic) should be used between the Primary and Secondary Compliance Studio instances.
- The same OpenSearch cluster is shared between the Primary and Secondary servers. If different OS clusters are used, then data/indices between these servers should be in sync.
- The certificates (studio_server.p12 and studio_server.jks) generated include the IP Addresses/DNS of the load balancer, Primary and Secondary Compliance Studio instances.

2.3 Install Compliance Studio on the Primary Server

To install Compliance Studio on the Primary server:

1. Download the Compliance Studio Installer and its associated patches from the My Oracle Support (MOS).

   > **Note:**
   >
   > If the load balancer is enabled for PGX, then the PGX load balancer URL should be provided in the **PGX_SERVER_URL** parameter of the `config.sh` file which is available in the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/bin` directory. For example, `PGX_SERVER_URL=http:// <PGX_LB_HOSTNAME>:<PGX_LB_PORT>` This configuration has to be done before installing Compliance Studio on the Primary server and this is required only for the Graph pipeline/ PGX server.

2. Install Compliance Studio on the Primary server. For more information on how to install, see the OFS Compliance Studio Installation Guide.

   After successful installation, ensure that all the services are started and the logs are clean.

3. Open the **additional_config.sh** file in the following path and update the parameters as mentioned in the following table.

   Table 2-1 Parameter of additional_config.sh file

File Path	Parameter
`<COMPLIANCE_STUDIO_INSTALLATION_PATH>/bin`	`PGX_INTERPRETER_OPTS="$PGX_INTERPRETER_OPTS -DAPP_BASE_NAME='pgx-interpreter' -Dgraph-service.url=https://<COMPLIANCE_STUDIO_SERVER_IP_ADDRESS>:7059/graph-service`

4. Open the **application.yml** file in the following path and update the parameters as mentioned in the following table.

Table 2-2 Parameter of application.yml file

File Path	Parameter
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/ deployed/mmg-home/mmg- studio/conf	mmgserviceUrl=https:// <LOAD_BALANCER_HOSTNAME>:7002/cs

5. Open the **application.properties** file in the following path and update the parameters as mentioned in the following table.

Table 2-3 Parameter of application.properties file

File Path	Parameter
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/ deployed/mmg-home/mmg-gateway/conf	mmg.gateway.url=https://<LB_URL>/
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/ mmg-home/mmg-ui/conf	BASE_URL=https:// <LOAD_BALANCER_HOSTNAME>:7002/cs
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/ mmg-home/mmg-ui/conf	DP_UI_URL=https:// <LOAD_BALANCER_HOSTNAME>:7063/ pipelineserviceui/pmf/dp/index.jsp
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/ mmg-home/mmg-ui/conf	MATCHSRVC_UI_URL=https:// <LOAD_BALANCER_HOSTNAME>/fcc/ graphmatchruleset.jsp
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/ mmg-home/mmg-ui/conf	LOADINDEX_UI_URL=https:// <LOAD_BALANCER_HOSTNAME>/fcc/ createindex.jsp
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/ deployed/mmg-home/mmg- pipeline/ pipeline/ pipelinegateway/conf	pipelineservice.uri=https:// <LOAD_BALANCER_HOSTNAME>:18005/ pipelineservice/
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/ deployed/mmg-home/mmg- pipeline/ pipeline/ pipelinegateway/conf	datapipelineservice.uri=https:// <LOAD_BALANCER_HOSTNAME>:18006/ datapipelineservice
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/ deployed/mmg-home/mmg- pipeline/ pipeline/data- metadata-job-<version>/ conf	pipeline.url=https:// <LOAD_BALANCER_HOSTNAME>:18005/ pipelineservice
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/ deployed/mmg-home/mmg- pipeline/ pipeline/data- pipeline-service- <version>/conf	mmg.url=https:// <LOAD_BALANCER_HOSTNAME>:7002/cs
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/ deployed/mmg-home/mmg- pipeline/ pipeline/data- pipeline-service- <version>/conf	pipeline.url=https:// <LOAD_BALANCER_HOSTNAME>:18005/ pipelineservice
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/ deployed/mmg-home/mmg- pipeline/ pipeline/data- pipeline-service- <version>/conf	datapipeline.url=https:// <LOAD_BALANCER_HOSTNAME>:18006/ datapipelineservice
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/ deployed/mmg-home/mmg- pipeline/ pipeline/pipeline- service-<version>/ conf	pipeline.url=https:// <LOAD_BALANCER_HOSTNAME>:18005/ pipelineservice

Table 2-3 (Cont.) Parameter of application.properties file

File Path	Parameter
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg- pipeline/pipeline/pipeline- service-<version>/conf	datapipeline.url=https://<LOAD_BALANCER_HOSTNAME>:18006/datapipelineservice
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg- pipeline/pipeline/pipeline- service-<version>/conf	gatewayUrl=https://<LOAD_BALANCER_HOSTNAME>:18006

6. Restart Compliance Studio to reflect the updated configuration. To restart the Compliance Studio on the Primary server, execute the following command.

```
./compliance-studio.sh --restart
```

2.4 Install Compliance Studio on the Secondary Server

To install Compliance Studio on the Secondary server:

> **Note:**
>
> If the load balancer is enabled for PGX, then the PGX load balancer URL should be provided in the **PGX_SERVER_URL** parameter of the `config.sh` file which is available in the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/bin` directory. For example, `PGX_SERVER_URL=http://<PGX_LB_HOSTNAME>:<PGX_LB_PORT>`
>
> This configuration has to be done before installing Compliance Studio on the Secondary server and this is required only for the Graph pipeline/ PGX server.

1. Compliance Studio can be installed on the Secondary server either as a fresh installation or cloned from the Primary server.
   - Install Compliance Studio (fresh installation) on the Secondary server. For more information on how to install, see the OFS Compliance Studio Installation Guide.
   - For cloning files from the Primary server, see the **Setup Compliance Studio Instance for Cloning the Filesystem** section in the OFS Compliance Studio Installation Guide.

     > **Note:**
     >
     > Ensure that the following resources are shared/synced between the Primary and Secondary servers for an HA configuration:
     >
     > – Database (Studio, ER/FSDF, Graph, and Atomic Schemas)
     >
     > – OpenSearch Cluster

     After successful installation/cloning, ensure that all the services are started and the logs are clean.

2. Open the **additional_config.sh** file in the following path and update the parameters as mentioned in the following table.

Table 2-4 Parameter of additional_config.sh file

File Path	Parameter
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/bin	PGX_INTERPRETER_OPTS="$PGX_INTERPRETER_OPTS -DAPP_BASE_NAME='pgx-interpreter' -Dgraph-service.url=https://<COMPLIANCE_STUDIO_SERVER_IP_ADDRESS>:7059/graph-service

3. Open the **application.yml** file in the following path and update the parameters as mentioned in the following table.

Table 2-5 Parameter of application.yml file

File Path	Parameter
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg- studio/conf	mmgserviceUrl=https://<LOAD_BALANCER_HOSTNAME>:7002/cs

4. Open the **application.properties** file in the following path and update the parameters as mentioned in the following table.

Table 2-6 Parameter of application.properties file

File Path	Parameter
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-gateway/conf	mmg.gateway.url=https://<LB_URL>/
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/mmg-home/mmg-ui/conf	BASE_URL=https://<LOAD_BALANCER_HOSTNAME>:7002/cs
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/mmg-home/mmg-ui/conf	DP_UI_URL=https://<LOAD_BALANCER_HOSTNAME>:7063/pipelineserviceui/pmf/dp/index.jsp
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/mmg-home/mmg-ui/conf	MATCHSRVC_UI_URL=https://<LOAD_BALANCER_HOSTNAME>/fcc/graphmatchruleset.jsp
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/mmg-home/mmg-ui/conf	LOADINDEX_UI_URL=https://<LOAD_BALANCER_HOSTNAME>/fcc/createindex.jsp
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-pipeline/pipeline/pipelinegateway/conf	pipelineservice.uri=https://<LOAD_BALANCER_HOSTNAME>:18005/pipelineservice/
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-pipeline/pipeline/pipelinegateway/conf	datapipelineservice.uri=https://<LOAD_BALANCER_HOSTNAME>:18006/datapipelineservice
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-pipeline/pipeline/data-metadata-job-<version>/conf	pipeline.url=https://<LOAD_BALANCER_HOSTNAME>:18005/pipelineservice

Table 2-6 (Cont.) Parameter of application.properties file

File Path	Parameter
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-pipeline/pipeline/pipeline-service-<version>/conf	pipeline.url=https://<LOAD_BALANCER_HOSTNAME>:18005/pipelineservice
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-pipeline/pipeline/pipeline-service-<version>/conf	datapipeline.url=https://<LOAD_BALANCER_HOSTNAME>:18006/datapipelineservice
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-pipeline/pipeline/pipeline-service-<version>/conf	gatewayUrl=https://<LOAD_BALANCER_HOSTNAME>:18006
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-pipeline/pipeline/data-pipeline-service-<version>/ conf	mmg.url=https://<LOAD_BALANCER_HOSTNAME>:7002/cs
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-pipeline/pipeline/data-pipeline-service-<version>/ conf	pipeline.url=https://<LOAD_BALANCER_HOSTNAME>:18005/pipelineservice
<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-pipeline/pipeline/data-pipeline-service-<version>/ conf	datapipeline.url=https://<LOAD_BALANCER_HOSTNAME>:18006/datapipelineservice

5. Restart Compliance Studio to reflect the updated configuration. To restart the Compliance Studio on the Secondary server, execute the following command.

```
./compliance-studio.sh --restart
```

2.5 Studio Schema Configuration

To configure the Studio Schema, follow these steps:

1. Log in to Studio Schema.
2. Open the **NEXTGENEMF_CONFIG** table and update value in the **V_VALUE** column as described in the following table.

Table 2-7 Parameter for NEXTGENEMF_CONFIG

V_NAME	V_VALUE
EMFSTUDIO_SERVICE_URL	https://<LOAD_BALANCER_HOSTNAME>:7002/cs
BASE_URL	https://<LOAD_BALANCER_HOSTNAME>:7002/cs
DATASTUDIO_URL	https://<LOAD_BALANCER_HOSTNAME>:7002/cs

3. Open the **MMG_MENU** table and update value in the **V_MENU_URL** column as described in the following table.

Table 2-8 Parameter for MMG_MENU Table

V_MENU_NAME	V_MENU_URL
Match Rules	`https://<LOAD_BALANCER_HOSTNAME>/fcc/matchrulesetsummary.jsp`
Merge Rules	`https://<LOAD_BALANCER_HOSTNAME>/fcc/mergerulesetsummary.jsp`
Data Survival	`https://<LOAD_BALANCER_HOSTNAME>/fcc/datasurvivalsummary.jsp`
Manual Decisioning	`https://<LOAD_BALANCER_HOSTNAME>/fcc/manualdecisioning.jsp`
Merge and Split Global Entities	`https://<LOAD_BALANCER_HOSTNAME>/fcc/mergeandsplit.jsp`
Data Pipelines	`https://<LOAD_BALANCER_HOSTNAME>:7063/pipelineserviceui/pmf/dp/index.jsp`

4. Open the **AAICL_SS_BATCH_URL** table and update value in the **V_URL** column as described in the following table.

Table 2-9 Parameter for MMG_MENU Table

V_URL_NAME	V_URL
MMG_SERVICE_URL	`https://<LOAD_BALANCER_HOSTNAME>:7002/cs`
WORKSPACE_URL	`https://<LOAD_BALANCER_HOSTNAME>:7002/cs`
CS_SERVICE_URL	`https://<LOAD_BALANCER_HOSTNAME>:7002/cs`

5. Restart Compliance Studio after making changes to the Studio schema. To restart, execute the following command.

```
./compliance-studio.sh --restart
```

2.6 Switching from One Server to another Server

Users can switch from Primary server to Secondary server and vice versa at any time.

> **Note:**
>
> Ensure that all the Compliance Studio services from the other server are down.

To switch from one server to another, follow these steps:

1. Delete entry from the Studio schema using the following query.

```
select * from DATABASECHANGELOG where author = 'Compliance Studio 8.1.2.1'
and id = 'FCC_DATASTUDIO_CONFIG_8121';
```

2. Start Compliance Studio on the other server. To start, execute the following command.

```
./compliance-studio.sh --start
```

2.7 Manual Configurations for Each Use Case

Users need to configure manually for the following use cases based on their requirement.

2.7.1 Entity Resolution

To configure the entity resolution, follow these steps:

1. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/ficdb/conf` directory.
2. Open the `resources.xml` file and update the following ER/FSDF Schema details.

```
<Resource
id="<ER_SCHEMA_ALIAS>"
name="jdbc/erdataschema"
auth="Container"
type="javax.sql.DataSource"
driverClassName="oracle.jdbc.OracleDriver"
url="jdbc:oracle:thin:@<ER_SCHEMA_ALIAS>"
connectionProperties="oracle.net.wallet_location=<STUDIO_WALLET_LOCAT
ION>;oracle.net.tns_admin=<STUDIO_TNS_ADMIN_PATH>;"
maxTotal="20"
maxIdle="0"
maxWaitMillis="-1">
</Resource>
```

2.7.2 Interpreters

Interpreters must be configured in both servers independently to work when switching from one server to another.

To configure Interpreters, see the Configure Interpreters section in the OFS Compliance Studio Administration and Configuration Guide.

2.7.3 PGX Server

Users need to configure both Compliance Studio and PGX server when load balancer is enabled.

2.7.3.1 In Compliance Studio

If the load balancer is enabled for PGX, then the PGX load balancer URL should be provided in the **PGX_SERVER_URL** parameter of the `config.sh` file which is available in the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/bin` directory.

For example, `PGX_SERVER_URL=http://<PGX_LB_HOSTNAME>:<PGX_LB_PORT>`

This configuration has to be done before installing Compliance Studio on the Primary and Secondary servers.

2.7.3.2 In PGX Server

Prerequisites: Load Balancer setup is required for HAProxy.

For configuring and installing PGX Server, see the Compliance Studio with Multiple PGX Servers (Using Load Balancer) section.

2.7.4 Scenario Conversion Utility

Before running the scenario generation notebook, update the **obj_url** and **url** variables in the **Generate Scenario(s)** paragraph to point to the respective Compliance Studio server that is currently running.

For more information, see the **Using Scenario Conversion Utility for ASC** section in the OFS Compliance Studio Administration and Configuration Guide.

2.7.5 Data Pipelines

The pem format of the studio_server certificate is referred to in the frontend configuration in the load balancer configuration.

Example:

```
frontend datapipeline_service
bind *:18006 ssl crt /etc/ssl/certs/haproxy.pem
mode http
default_backend datapipeline_service
```

Here, `haproxy.pem` is the pem format of the `studio_server.p12` file that is available in the `/etc/ ssl/certs/` path.

2.8 HAProxy Configuration

To configure HAProxy, follow these steps:

1. Navigate to `/etc/haproxy` directory and execute following command in the terminal.

   ```
   cd /etc/haproxy
   ```

2. Open the **haproxy.cfg** file and execute following command in the terminal.

   ```
   vi haproxy.cfg
   ```

3. Add the following frontend and backend blocks in the **haproxy.cfg** file.

   # HAPROXY CONFIGURATION STARTS

   Frontend configurations

   > **Note:**
   >
   > For example, use Compliance_Studio_Gateway_Port as 7071.

   ```
   frontend compliance_studio
   bind *:<Compliance_Studio_Gateway_Port>
   mode http
   bind *:443 ssl crt /etc/ssl/certs/haproxy.pem
   ```

```
use_backend fcc_ui if { path /fcc } || { path_beg /fcc/ }
http-request set-header X-Forwarded-For %[src]
reqadd X-Forwarded-Proto:\ https
option http-server-close
default_backend compliance_studio

frontend mmg
bind *:7002 ssl crt /etc/ssl/certs/haproxy.pem mode http
default_backend mmg

frontend data_studio
bind *:7008 ssl crt /etc/ssl/certs/haproxy.pem mode http
default_backend data_studio

frontend dp_ui
bind *:7063 ssl crt /etc/ssl/certs/haproxy.pem mode http
default_backend dp_ui

frontend graph_service
bind *:7059 ssl crt /etc/ssl/certs/haproxy.pem mode http
default_backend graph_service

frontend pipeline_service
bind *:18005 ssl crt /etc/ssl/certs/haproxy.pem
mode http
default_backend pipeline_service

frontend datapipeline_service
bind *:18006 ssl crt /etc/ssl/certs/haproxy.pem
mode http
default_backend datapipeline_service

frontend pgx_server
bind *:7007
mode http
default_backend pgx_server
```

Backend configurations

> **Note:**
>
> For example, use Compliance_Studio_Gateway_Port as 7071.

```
backend compliance_studio
mode http
balance roundrobin
cookie JSESSIONID prefix nocache
server cs_server_1
<server_1_hostname>:<Compliance_Studio_Gateway_Port>/cs/home check ssl
verify none
server cs_server_2
<server_2_hostname>:<Compliance_Studio_Gateway_Port>/cs/home check ssl
verify none
```

```
backend mmg
mode http
balance roundrobin
cookie JSESSIONID prefix nocache
server mmg_1 <server_1_hostname>:7002 check ssl verify none
server mmg_2 <server_2_hostname>:7002 check ssl verify none

backend data_studio
mode http
balance roundrobin
cookie JSESSIONID prefix nocache
server data_studio_1 <server_1_hostname>:7008/cs check ssl verify
none
server data_studio_2 <server_2_hostname>:7008/cs check ssl verify
none

backend dp_ui
mode http
balance roundrobin
cookie JSESSIONID prefix nocache
server dp_ui_1 <server_1_hostname>:7063/pipelineserviceui check ssl
verify none
server dp_ui_2 <server_2_hostname>:7063/pipelineserviceui check ssl
verify none

backend fcc_ui
mode http
balance roundrobin
cookie JSESSIONID prefix nocache
server fcc_ui_1 <server_1_hostname>:7061/fcc check ssl verify none
server fcc_ui_2 <server_2_hostname>: 7061/fcc check ssl verify none

backend graph_service
mode http
balance roundrobin
cookie JSESSIONID prefix nocache
server graph_service_1 <server_1_hostname>:7059 check ssl verify none
server graph_service_2 <server_2_hostname>:7059 check ssl verify none

backend pipeline_service
mode http
balance roundrobin
cookie JSESSIONID prefix nocache
server pipeline_service_1 <server_1_hostname>:18005 check ssl verify
none
server pipeline_service_2 <server_2_hostname>:18005 check ssl verify
none

backend datapipeline_service
mode http
balance roundrobin
cookie JSESSIONID prefix nocache
server datapipeline_service_1 <server_1_hostname>:18006 check ssl
verify none
server datapipeline_service_2 <server_2_hostname>:18006 check ssl
```

```
verify none

backend pgx_server
mode http
cookie PGX_INSTANCE_STICKY_COOKIE insert indirect nocache
server pgx_server_1 <server_1_hostname>:7007 check
server pgx_server_2 <server_1_hostname>:7007 check cookie
pgx_server_2
option httpchk GET /isReady http-check expect string true
```

 # HAPROXY CONFIGURATION ENDS

3 Compliance Studio with Multiple PGX Servers (Using Load Balancer)

> **Note:**
>
> This section describes HAProxy but other load balancers can be used. The Incremental file system changes from the Active PGX server should be synced/ reflected in all the fail over PGX servers as well.

Using a load balancer, users can configure multiple PGX servers in the Compliance Studio.

HAProxy is a high-performance TCP/HTTP load balancer and proxy server that allows multiplexing incoming requests across multiple web servers. You can use HAProxy with multiple instances of the graph server (PGX) for high availability.

Prerequisites

- Load Balancer setup is required for HAProxy.
- Two or more servers should be available for the PGX server.

To configure multiple PGX servers, follow these steps:

1. Configure and Install PGX on both servers. To Configure and Install PGX, see the **Configure the PGX Service**section in the OFS Compliance Studio Installation Guide.
2. Navigate to `<PGX_SERVER_HOME>/pgx-server/conf` directory and place the following files:
   - `studio_server.p12`
   - `public.key and private.key`
   - `graph-keystore.p12`

   > **Note:**
   >
   > - For more information on where to obtain studio_server.p12 file, see the section in the OFS Compliance Studio Installation Guide.
   > - For more information on where to obtain public.key and private.key files, see the section in the OFS Compliance Studio Installation Guide.
   > - For more information on where to obtain graph-keystore.p12 file, see the section in the OFS Compliance Studio Installation Guide.
3. Start the PGX server.
4. Install the Load balancer HAProxy in one of the PGX servers as mentioned in the Using HAProxy for PGX Load Balancing and High Availability section.

5. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/bin` directory.
6. Open the `config.sh` file and provide a Load balancer URL in the **PGX_SERVER_URL** parameter is as follows.
   `PGX_SERVER_URL=http://<PGX_LB_Hostname>:<PGX_LB_PORT>`

   For example, `PGX_SERVER_URL=http://testserver.oracle.com/:1234`

   > **Note:**
   >
   > - The PGX load balancer can be configured in the same server that is used for load balancing the Compliance Studio as well.
   > - Only Subgraph loading is supported and In-memory graph loading is not supported if HA is configured for the PGX server.

7. Start Compliance Studio. To start, execute the following command.
   `./compliance-studio.sh --start`

4

Setup Disaster Recovery (DR) in Compliance Studio

This section describes additional configuration required to setup Disaster Recovery (DR) in the Compliance Studio.

> **Note:**
>
> The Incremental file system changes from the Active Compliance Studio server should be synced/reflected in all the fail over Compliance Studio servers as well.

Prerequisites:

- Create a wallet. To create a wallet, see the **Setup the Password Stores for Database User Accounts** section in the OFS Compliance Studio Installation Guide.
- Add Secondary Database credentials in the wallet.
- ER/FSDF Schema, Atomic Schema, Studio Schema, and Graph Schema names should be the same as the Primary Database.

To setup DR in the Compliance Studio, follow these steps:

1. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/bin` directory.
2. Open `config.sh` file and update the parameters as mentioned in the following table.

Table 4-1 Parameter Values in Config.sh File

Parameter	Placeholder Value
STUDIO_DB_HOSTNAME	##SECONDARY_STUDIO_DB_HOSTNAME ##
STUDIO_DB_PORT	##SECONDARY _STUDIO_DB_PORT##
STUDIO_DB_SERVICE_NAME	##SECONDARY _STUDIO_DB_SERVICE_NAME ##
STUDIO_DB_SID	##SECONDARY _STUDIO_DB_SERVICE_NAME ##
STUDIO_DB_USERNAME	For example, CS81250_DR_2477
ATOMIC_DB_HOSTNAME	##SECONDARY _ATOMIC_DB_HOSTNAME ##
ATOMIC_DB_PORT	##SECONDARY _STUDIO_DB_PORT##
ATOMIC_DB_SERVICE_NAME	##SECONDARY _ATOMIC_DB_SERVICE_NAME ##
ATOMIC_DB_SID	##SECONDARY _ATOMIC_DB_SERVICE_NAME ##
ATOMIC_DB_USERNAME	For example, STD_ATOM8125
GRAPH_DB_SERVER_NAME	##SECONDARY _GRAPH_DB_SERVER_NAME ##
GRAPH_DB_PORT	## SECONDARY_GRAPH_DB_PORT ##

Table 4-1 (Cont.) Parameter Values in Config.sh File

Parameter	Placeholder Value
GRAPH_DB_SERVICE_NAME	## SECONDARY _GRAPH_DB_SERVICE_NAME ##
GRAPH_KEYSTORE_PASSWORD	For example, password123
GRAPH_SCHEMA_DB_SCHEMA_NAME	For example, GS81250_DR_2477
GRAPH_SCHEMA_WALLET_ALIAS	For example, GS81250_DR_2477_alias
GRAPH_SCHEMA_WALLET_LOCATION	##POINTING_TO_DR_DB##
GRAPH_SCHEMA_TNS_ADMIN_PATH	##POINTING_TO_DR_DB##
WALLET_LOCATION	##POINTING_TO_DR_DB##
TNS_ADMIN_PATH	##POINTING_TO_DR_DB##

Note:

For the parameter description, see the **Configure the config.sh File** section in the OFS Compliance Studio Installation Guide.

3. Stop the Compliance Studio by executing the following command.
   `./compliance-studio.sh --stop`
4. Reinstall the Compliance Studio by executing the following command.
   `./compliance-studio.sh --reinstall`
5. Start the Compliance Studio by executing the following command.
   `./compliance-studio.sh --start`
6. Navigate to the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/ficdb/conf` directory.
7. Open the **resources.xml** file and update the following details for ER/FSDF Schema. Example:

```
<Resource
id="###ER_SCHEMA#"
name="jdbc/erdataschema"
auth="Container"
type="javax.sql.DataSource"
driverClassName="oracle.jdbc.OracleDriver"
url="jdbc:oracle:thin:@ "###ER_SCHEMA#""
connectionProperties= "oracle.net.wallet_location
=<WALLET_PATH/ABCD>;
oracle.net.tns_admin=<WALLET_PATH/ABCD>;”
maxTotal="5"
maxIdle="0"
maxWaitMillis="-1" >
</Resource>
```

5
Frequently Asked Questions (FAQs)

This section consists of resolutions to the frequently asked questions noticed during the Compliance Studio Installation.

1. What should I do if the data pipelines are failing with the following error?

```
09/Jun/2022 10:48:32,978- ImportDAOImpl: Exception in import processor
org.springframework.web.client.ResourceAccessException: I/O error on
POST request for "https://<LOAD_BALANCER_HOSTNAME>:18004/
datapipelineservice/MAP/IMPORT": PKIX path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to
find valid certification path to requested target; nested exception is
javax.net.ssl.SSLHandshakeException: PKIX path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to
find valid certification path to requested target
at
org.springframework.web.client.RestTemplate.doExecute(RestTemplate.java:
748) ~[spring-web-5.2.5.RELEASE.jar!/:5.2.5.RELEASE]
at
org.springframework.web.client.RestTemplate.execute(RestTemplate.java:67
4) ~[spring-web-5.2.5.RELEASE.jar!/:5.2.5.RELEASE]
at
org.springframework.web.client.RestTemplate.postForObject(RestTemplate.j
ava:418) ~[spring-web-5.2.5.RELEASE.jar!/:5.2.5.RELEASE]
at
com.oracle.fccm.amlxe.pipelineService.client.RESTClient.callRESTService(
RESTClient.java:107) ~[class]
```

   To resolve this issue, ensure that the pem format of the studio_server certificate is referred to in the frontend configuration in the load balancer configuration.

   For example,

```
frontend datapipeline_service
bind *:18006 ssl crt /etc/ssl/certs/haproxy.pem
mode http
default_backend datapipeline_service
```

   Here, `haproxy.pem` is the pem format of the `studio_server.p12` file that is available in the `/ etc/ssl/certs/`path.

2. What should I do if Compliance Studio server fails with the below error in metaservice.log when switching from one server to another?

```
26-03-2024 08:35:26.942 [ main] ERROR
ofss.fccm.applicationserver.server.GrizzlyServer - Server failed to
start liquibase.exception.ValidationFailedException: Validation Failed:
1 change sets check sum model/datamodel/
fccstudio_Atomic_Schema_8121.xml::FCC_DATASTUDIO_CONFIG_8121::Compliance
```

```
Studio 8.1.2.1 was: 8:34bcb65a5125e53bd31acbc46a504d04 but is now:
8:c52d8dba1a04bcd378a9744484ab01be at
liquibase.changelog.DatabaseChangeLog.validate(DatabaseChangeLog.java:29
6) ~[liquibase-core-4.8.0.jar:?]
```

To resolve this issue, before switching the Compliance Studio instance from one server to another delete the record returned by the following query:

```
select * from DATABASECHANGELOG where author = 'Compliance Studio 8.1.2.1'
and id = 'FCC_DATASTUDIO_CONFIG_8121';
```

3. What should I do if PGX fails to start/restart and displays the below error in the pgx-server.log?

```
22-05-2024 08:48:33 ERROR o.p.r.PgxContextListener - Exception while
initializing PGX webapp
java.util.concurrent.ExecutionException:
java.lang.IllegalArgumentException: javax.net.ssl.SSLHandshakeException:
No name matching <PGX_SERVER_HOSTNAME> found
at java.base/
java.util.concurrent.CompletableFuture.reportGet(CompletableFuture.java:
395)
at java.base/
java.util.concurrent.CompletableFuture.get(CompletableFuture.java:1999)
at oracle.pgx.api.PgxFuture.get(PgxFuture.java:113)
at
oracle.pgx.rest.PgxContextListener.contextInitialized(PgxContextListener
.java:64)
at
org.apache.catalina.core.StandardContext.listenerStart(StandardContext.j
ava:4462)
at
org.apache.catalina.core.StandardContext.startInternal(StandardContext.j
ava:4914)
at
org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:171)
at
org.apache.catalina.core.ContainerBase$StartChild.call(ContainerBase.jav
a:1332)
at
org.apache.catalina.core.ContainerBase$StartChild.call(ContainerBase.jav
a:1322)
at java.base/
java.util.concurrent.FutureTask.run(FutureTask.java:264)

.. .. ..
Caused by: java.lang.IllegalArgumentException:
javax.net.ssl.SSLHandshakeException: No name matching
<PGX_SERVER_HOSTNAME> found
at oracle.pgx.engine.admin.Ctrl.preloadGraphs(Ctrl.java:355)
at oracle.pgx.engine.admin.Ctrl.access$1600(Ctrl.java:97)
at oracle.pgx.engine.admin.Ctrl$1.call(Ctrl.java:248)
at oracle.pgx.engine.admin.Ctrl$1.call(Ctrl.java:186)
at
oracle.pgx.api.admin.internal.AbstractControl.runOnCallerThread(Abstract
```

```
Control.java:45)
at oracle.pgx.engine.admin.Ctrl.start(Ctrl.java:186)
at
oracle.pgx.api.admin.internal.AbstractControl.start(AbstractControl.java
:115)
at
oracle.pgx.api.admin.internal.AbstractControl.lambda$start$5(AbstractCon
trol.java:101)
at java.base/
java.util.function.Function.lambda$andThen$1(Function.java:88)
at java.base/
java.util.concurrent.CompletableFuture.uniComposeStage(CompletableFuture
.java:1106)
at java.base/
java.util.concurrent.CompletableFuture.thenCompose(CompletableFuture.jav
a:2235)
at oracle.pgx.api.PgxFuture.thenCompose(PgxFuture.java:178)
at
oracle.pgx.api.admin.internal.AbstractControl.start(AbstractControl.java
:101)
at
oracle.pgx.api.admin.internal.AbstractControl.lambda$start$1(AbstractCon
trol.java:79)
at java.base/
java.util.function.Function.lambda$andThen$1(Function.java:88)
at java.base/
java.util.concurrent.CompletableFuture.uniComposeStage(CompletableFuture
.java:1106)
at java.base/
java.util.concurrent.CompletableFuture.thenCompose(CompletableFuture.jav
a:2235)
at oracle.pgx.api.PgxFuture.thenCompose(PgxFuture.java:178)
at
oracle.pgx.api.admin.internal.AbstractControl.start(AbstractControl.java
:67)
at
oracle.pgx.rest.PgxContextListener.contextInitialized(PgxContextListener
.java:62)
... 26 common frames omitted
```

This happens due to graph was loaded with IN-MEMORY mode which is not supported in a HA PGX configuration.

To clean up the IN-MEMORY graph and reload it in OFFLOADED mode, follow these steps:

a. Truncate the records from **FCC_PGX_M_CONFIG, FCC_GRAPH_M_DATA_SOURCES** and **FCC_GRAPH_M_CONFIG_JSON** tables.

b. Start the PGX server.

c. Run the **Refresh Graph** task with graph type as **OFFLOADED**.