Oracle Financial Services Crime and Compliance Studio

Deployment and Configuration Guide

Release8.0.7.4.0

May 2020

E91246-01

ORACLE®
Financial Services

OFS Crime and Compliance Studio Deployment and Configuration Guide

Document Control

Version Number	Revision Date	Changes Done
8.0.7.4.0	Updated: May 2020	- FCC Studio can use either the BD or ECM schema as the source of FCDM data for the graph. A new parameter, `FCDM_Source` added to the `studio-env.yml` file. For more information, see Configuring the studio-env.yml File. - IDCS Realm Integration with FCC Studio allows SSO for both FCC Studio and ECM to provide seamless integration and eliminates the requirement to log in separately to FCC Studio. For more information, see Configuring the studio-env.yml File. - FCC Studio supports the installation of any version of Python and also FCC Studio Installer is packaged with Python 3.6 libraries. For more information, see Configuring the fcc-python Interpreter - ES_Hadoop jar files must be placed in all nodes of the Spark cluster. For more information, see Configuring Data Movement and Graph Load. - Elastic Search component has been enhanced to prevent the OutOfMemory error. For more information, see Appendix - Configuring the Elastic Search component. - Configuring the preferred services to be deployed during deployment of FCC Studio. For more information, see Configuring the Preferred Services. - User preferred namespace can be configured for FCC Studio. Configuring the ETL Services.
8.0.7.3.0	Updated: March 2020	- A new component called the Entity Resolution is introduced that enables Entity linking in graphs and Entity Searching in notebooks based on multi-attribute name matching using the Elastic Search service. For more information, see Configuring the Elastic Search Component and Appendix - Configuring the Elastic Search component. - A new script is introduced for the preparation of ICIJ to resolve data quality issues before graph loading. For more information, see Cleaning the ICIJ Data.
8.0.7.2.0	Updated: February 2020	- Deploy Studio application with or without OFSAA on the Kubernetes cluster. For more information, see Deploying FCC Studio on the K8s Cluster and Deploying FCC Studio with Non-OFSAA on the K8s Cluster. - The support for the Data Forward service has been deprecated. - Configuration for the newly introduced interpreters such as Spark and PySpark interpreters. For more information, see Interpreter Settings.

Version Number	Revision Date	Changes Done
8.0.7.1.0	Updated: October 2019	Created the first version of Oracle Financial Services Crime and Compliance Studio Deployment Guide for v8.0.7.1.0 Release.

Table of Contents

1 Preface

This section provides supporting information for the Oracle Financial Services (OFS) Crime and Compliance Studio (FCC Studio) Application Installation Guide.

Topics:

- Summary
- Audience
- Related Documents

1.1 Summary

Before you begin the deployment, ensure that you have access to the Oracle Support Portal with valid login credentials to quickly notify us of any issues at any stage. You can obtain the login credentials by contacting the Oracle Support.

1.2 Audience

This document is intended for System Engineers who are responsible for deploying and configuring or upgrading FCC Studio.

The document assumes that you have experience in installing Enterprise components. and have the basic knowledge of the following:

- UNIX commands
- Database concepts
- Big Data
- Kubernetes
- Docker

1.3 Related Documents

You can access the following additional documents related to the OFS Crime and Compliance Studio application from the *Oracle Help Center (OHC)* Documentation Library:

- *Oracle Financial Services Crime and Compliance Studio Installation Guide*
- *Oracle Financial Services Crime and Compliance Studio Administration Guide*
- *Oracle Financial Services Crime and Compliance Studio User Guide*
- *Oracle Financial Services Crime and Compliance Studio Data Model Guides*
- *Oracle Financial Services Crime and Compliance Studio Release Notes and Readme*

1.4 Abbreviations

The following table lists the abbreviations used in this document.

Table 1: Abbreviations

Abbreviation	Meaning
OFS	Oracle Financial Services
FCC Studio	Financial Crime and Compliance Studio
OFSAA	Oracle Financial Services Analytical Application
K8s	Kubernetes
BD	Behavior Detection
ECM	Enterprise Case Management
FCDM	Financial Crime Data Model
ICIJ	International Consortium of Investigative Journalists
IDCS	Oracle Identity Cloud Service
SSO	Single Sign-On

2 Deployment Overview

This chapter provides the information required to understand the deployment of the Oracle Financial Services (OFS) Crime and Compliance Studio (FCC Studio) application on the Kubernetes (K8s) cluster.

This release (v8.0.7.4.0) of FCC Studio can be used for the following:

- To deploy a new instance FCC Studio on the K8s cluster as follows:
  - FCC Studio with OFSAA (Oracle Financial Services Analytical Application). Here, with OFSAA is with BD (Behavior Detection) or ECM (Enterprise Case Management).
  - FCC Studio without OFSAA
- To upgrade an existing instance of FCC Studio on the K8s cluster as follows:
  - Upgrade FCC Studio from v8.0.7.x onwards to v8.0.7.4.0 with OFSAA.

2.1 Quick Start Steps to Deploy FCC Studio with OFSAA on the K8s Cluster

Table 1: Quick Start Steps to Deploy FCC Studio with OFSAA on the K8s Cluster

Sl. No.	Steps	Reference Links
1.	Prepare for Deployment	1. Prerequisites 2. Hardware and Software Requirements 3. Prerequisite Environmental Settings 4. Performing Common Pre-installation Tasks a. Obtaining the Software b. Extracting the Software 5. Required File Structure 6. Interpreter Settings
2.	Deploy FCC Studio with OFSAA on the K8s Cluster	1. Configuring the Elastic Search Component 2. Configuring Wallet 3. Configuring the studio-env.yml File 4. Configuring the ETL Services 5. Verifying the Resource Allocation for FCC Studio Services 6. Deploying FCC Studio on the K8s Cluster 7. Verifying the FCC Studio Deployment 8. Accessing the FCC Studio Application
3.	Post-deployment configuration	• Configuring the Interpreters • Performing the OFSAA Configuration for Batch Execution • Performing Hive Data Movement • Configuration for Running Published Notebooks

2.2 Quick Start Steps to Deploy FCC Studio with Non-OFSAA on the K8s Cluster

Table 2: Quick Start Steps to Deploy FCC Studio with Non-OFSAA on the K8s Cluster

Sl. No.	Steps	Reference Links
1.	Prepare for Deployment	1. Prerequisites 2. Hardware and Software Requirements 3. Prerequisite Environmental Settings 4. Performing Common Pre-installation Tasks a. Obtaining the Software b. Extracting the Software 5. Required File Structure 6. Interpreter Settings

Table 2: Quick Start Steps to Deploy FCC Studio with Non-OFSAA on the K8s Cluster

Sl. No.	Steps	Reference Links
2.	Deploy FCC Studio with non-OFSAA on the K8s Cluster	1. Configuring the Elastic Search Component 2. Configuring Wallet 3. Configuring the studio-env.yml File 4. Configuring the ETL Services 5. Verifying the Resource Allocation for FCC Studio Services 6. Deploying FCC Studio on the K8s Cluster 7. Verifying the FCC Studio Deployment 8. Accessing the FCC Studio Application
3.	Post-deployment configuration	• Configuring Interpreters • Configuring ICIJ

3 Preparing for Deployment

This chapter provides information about the tasks that must be performed before deploying FCC Studio.

Topics:

- Prerequisites
- Hardware and Software Requirements
- Prerequisite Environmental Settings
- Performing Common Pre-installation Tasks
- Required File Structure
- Interpreter Settings

3.1 Prerequisites

The Linux machine must satisfy the following conditions:

- To deploy FCC Studio with OFSAA, ensure the BD (Behavior Detection) or the ECM (Enterprise Case Management) application pack is installed.
- Kubernetes (k8s) cluster must be installed with the following:
  - Registry to store docker images.
  - Minimum of 8 GB memory (inclusive of all nodes) available for the installation. The actual memory requirement depends on the workload/container size configuration.
- Docker and kubectl must be installed.
- kubectl is configured (that is, connected to cluster where you want to install FCC Studio).
- Docker has push access to a private registry.
- 12GB free space is available to store the FCC Studio Installer zip file in some directory.
- 45GB free space is available in the docker root directory. You can find the docker root directory using the `docker info` command.

3.2 Hardware and Software Requirements

The following hardware and software are required to deploy FCC Studio.

Table 1: Hardware and Software Requirements

Hardware/Software	Component Version
Browser	• Chrome 57.x • Firefox 52.x
Java Version	Java 8
Docker Registry	• Docker registry must be present to store docker images • Min of 45GBspace is required to save docker images

Table 1: Hardware and Software Requirements

Hardware/Software	Component Version
Database Server	● Oracle Database Server 12c Release 2 (12.2.0.1+) Enterprise Edition ● Oracle R Enterprise 1.5.1 with Oracle R Distribution 3.3.0
Elastic Search	elasticsearch-7.3.2
Elastic Search Hadoop Jars	ES - Hadoop v7.3.2 You must download the ZIP file from *https://www.elastic.co/downloads/past-releases/elasticsearch-apache-hadoop-7-3-2*
Kubernetes Cluster	
Processing Server	● RHEL 7.4+ ● SFTP ● Oracle JRE Standard Edition 1.8.x(with JCE) ● Kubernetes(K8s) cluster. For more information, see Prerequisites.
PGX (Graph) Server	● RHEL 7.4+ ● Kubernetes(K8s) cluster. For more information, see Prerequisites. ● Minimum gcc library version 4.8.2
Big Data	
Cloudera Distribution Hadoop 5.12	● CDH Version 5.12 ● Hadoop-2.5.0+cdh5.3.3+844 ● Hive-0.13.1+cdh5.3.3+350 ● Sqoop1 V 1.4.4+cdh5.3.3+67 ● The `.profile` file must be present with the `SPARK_HOME` and `PYTHON_HOME` parameters already set. ● Set `spark2-shell alias` in the `.profile` file as follows: `alias spark2-shell=spark-shell`
Cloudera Hive Connectors	Hive JDBC Connectors V 2.5.15
Hadoop Security Protocol	● Kerberos R release 1.6.1 ● Sentry-1.4.0

3.3 Prerequisite Environmental Settings

The following prerequisite environmental settings must be set before beginning the deployment of FCC Studio.

Table 2: Prerequisite Information

Category	Expected Value
Java Settings	● `PATH` in the `.profile` file must be set to include `kubectl` and the Java Runtime Environment (Java 8) absolute path. **NOTE:** ● Ensure the absolute path to `JRE/bin` is set at the beginning of the `PATH` variable. For example: `PATH=/usr/java/jre1.8/bin:$PATH` ● Ensure no SYMBOLIC links to Java installation are set in the `PATH` variable.
Oracle Database Settings	**NOTE:** This setting is required only if the Wallet must be created on the same server as that of the Studio server. **Oracle Processing Server** ● `ORACLE_HOME` must be set in the `.profile` file pointing to the appropriate Oracle DB Client installation. ● `PATH` in the `.profile` file must be set to include the appropriate `$ORACLE_HOME/bin` path.
Download Directory	Indicates the directory where the product installer zip file is downloaded/copied. The user permission must be set to 755 for this Download directory.
Installation Directory	Indicates the directory where the product installer zip file is extracted and the installation files are placed. The user permission must be set to 755 for this installation directory. **NOTE:** The Installation and the Download Directory can be the same if the product installer zip file is not copied separately to another directory.
OS Locale	Linux: `en_US.utf8` Execute the following command to check the locale installed: `locale -a | grep -i 'en_US.utf'` The installed locale is displayed.
Studio Schema	1. Create a new Oracle Database schema user using the following script: `CREATE USER <Studio Schema User Name> IDENTIFIED BY <Password>;` A new oracle Database schema is created. 2. Grant the permissions that are given in the next row. This newly created schema is referred to as Studio Schema.

Table 2: Prerequisite Information

Category	Expected Value

Oracle Database Schema Settings	Grant the following permissions to the newly created Oracle Database Schema:
	`GRANT create session TO <Studio Schema User>;`
	`GRANT create table TO <Studio Schema User>;`
	`GRANT create view TO <Studio Schema User>;`
	`GRANT create any trigger TO <Studio Schema User>;`
	`GRANT create any procedure TO <Studio Schema User>;`
	`GRANT create sequence TO <Studio Schema User>;`
	`GRANT execute on dbms_rls TO <Studio Schema User>;`
	`GRANT execute on sys.dbms_session TO <Studio Schema User>;`
	`ALTER USER <Studio Schema User> QUOTA 100M ON users;`
	`GRANT create sequence TO <Studio Schema User>;`
	`GRANT create SYNONYM TO <Studio Schema User>;`
	`GRANT create any context TO <BD or ECM Schema User>;`
	`GRANT execute on dbms_rls TO <BD or ECM Schema User>;`
	`GRANT ALL privileges TO <Studio Schema User>;`

Wallet Settings	Set a password store with Oracle Wallet. For more information, see Appendix - Setting Up Password Stores with Oracle Wallet.

3.4 Performing Common Pre-installation Tasks

The common pre-installation tasks that you must perform before installing FCC Studio are:

1. Obtaining the Software
2. Extracting the Software

3.4.1 Obtaining the Software

To download and copy the FCC Studio application installer software, follow these steps:

1. Login to My Oracle Support with a valid Oracle account and search for the Bug ID **31230440** under the *Patches & Updates* tab.
2. Download the following installer archive files to the download directory (in Binary Mode) on the setup identified for the FCC Studio installation:
   - `<FCC_Studio_Installer_Archive_File>_1of2.zip`
   - `<FCC_Studio_Installer_Archive_File>_2of2.zip`

3.4.2 Extracting the Software

To extract the FCC Studio application installer software, follow these steps:

1. Extract the contents of the downloaded zip files using the following command:

   ```
   unzip <FCC_Studio_Installer_Archive_File>_1of2.zip
   ```

```
unzip <FCC_Studio_Installer_Archive_File>_2of2.zip
```

Both installer zip files are extracted to the same directory and the `OFS_FCCM_STUDIO` directory is obtained and is referred to as `<Studio_Installation_Path>`.

> **NOTE** Do not rename the application installer directory name after extraction from the archive.

2. Navigate to the download directory where the installer archive is extracted and assign execute permission to the installer directory using the following command:

```
chmod 0755 OFS_FCCM_STUDIO -R
```

3.5 Required File Structure

Obtain the following configuration files from the Cloudera installation setup:

> **NOTE** These files must be kept ready and provided in the following file structure which is used during FCC Studio installation.

Table 3:

File Category	File Names
Hadoop Cluster	• `core-site.xml` • `hadoop-env.sh` • `hdfs-site.xml` • `log4j.properties` • `ssl-client.xml` • `topology.map` • `topology.py`
Kerberos Files	• `krb5.conf` • `ofsaa.keytab` **NOTE:** Ensure to rename your `.keytab` file to `ofsaa.keytab`.

Table 3:

File Category	File Names
Additional Jars	• `hive-exec-1.1.0-cdh5.13.0.jar` • `HiveJDBC4.jar` • `hive-metastore-1.1.0-cdh5.13.0.jar` • `hive-service-1.1.0-cdh5.13.0.jar` **NOTE:** • The version of the jars is client/user-specific. These jars can be obtained from existing jars of Cloudera installation. • The `HiveJDBC4.jar` file is not available in the Cloudera setup. You must download the same from the Cloudera website.
ES-Hadoop Jars	`elasticsearch-spark-20_2.11-7.3.2.jar` To obtain the `elasticsearch-spark-20_2.11-7.3.2.jar` file, follow these steps: 1. Download the ZIP file from *https://www.elastic.co/downloads/past-releases/elasticsearch-apache-hadoop-7-3-2*. 2. Extract the downloaded file. 3. Navigate to the `dist` directory and obtain the `elasticsearch-spark-20_2.11-7.3.2.jar` file.

3.6 Interpreter Settings

To perform the interpreter settings for the interpreters that you need, see Appendix - Pre-deployment Interpreter Settings.

4 Deploying FCC Studio on the K8s Cluster

To deploy FCC Studio on the K8s cluster, follow these steps:

1. Configuring the Elastic Search Component
2. Enabling Synonym/Stopword with the Elastic Search Service
3. Configuring Wallet
4. Configuring the Preferred Services
5. Configuring the studio-env.yml File
6. Configuring the ETL Services
7. Verifying the Resource Allocation for FCC Studio Services
8. Deploying FCC Studio on the K8s Cluster
9. Verifying the FCC Studio Deployment
10. Accessing the FCC Studio Application

4.1 Configuring the Elastic Search Component

Configure the Elastic Search component as per FCC Studio requirements. For more information, see Appendix - Configuring the Elastic Search component.

4.2 Enabling Synonym/Stopword with the Elastic Search Service

To enable the Synonym and Stopword service with the Elastic Search service, follow these steps:

1. Navigate to the `<Elastic_Search_Installed_Path>/config` directory.
2. Create a directory named `analysis` using the following command:

   ```
   mkdir analysis
   ```

3. Place the following `Stopword` and `Synonym` files in the newly created `analysis` directory:

   NOTE
   - User can decide whether to provide any data or not in the `Stopword` or `Synonym` files.
   - Each stopword must be provided in a separate line.
   - All related synonyms must be provided in the same line separated by a comma.

   - `Synonyms.txt`: Contains name synonyms like bob, bobby, and so on.
   - `Country.txt`: Contains country synonyms like US, United States, America, and so on.
   - `Organisation_suffix.txt`: Contains organization suffices that are used as stopwords.
   - `Title.txt`: Contains title stopwords used as the title for name
   - `Gender.txt`: Contains gender-related synonyms.
   - `Organisation_strip.txt`: Contains organization stopwords.

4.3 Configuring Wallet

To configure wallets, follow these steps:

1. Create a wallet. For information on creating wallets, see Appendix - Setting Up Password Stores with Oracle Wallet.
2. Copy the wallet files, `cwallet.sso, ewallet.p12` and `tnsnames.ora`, and place in the `<Studio_Installation_Path>/configmaps/wallet` directory.

4.4 Configuring the Preferred Services

To configure the preferred services to be deployed during deployment of FCC Studio, follow these steps:

1. Navigate to the `<Studio_Installation_Path>/bin/` directory.
2. Configure the `serviceMapping.sh` file as follows:

   a. Set the following parameters:

      `deployment=All` or `Custom.`

      Where,

      – `All` indicates to deploy all the services .

      – `Custom` indicates to choose selected services to deploy.

   b. If `deployment=Custom`, set `true` for the desired services and `false` for the undesired services.

> **NOTE** Do not set `false` for the following services:
> - agent
> - server

4.5 Configuring the studio-env.yml File

Configure the `studio-env.yml` file to deploy FCC Studio. For more information, see Appendix - Configuring the studio-env.yml File.

4.6 Configuring the ETL Services

To configure the ETL services, follow these steps:

1. Place the Hadoop Cluster files in the `<Studio_Installation_Path>/configmaps/spark` directory. For more information on the file structure, see Required File Structure.
2. Place the Kerberos files in the `<Studio_Installation_Path>/configmaps/batchservice/user/conf/` directory. For more information on the file structure, see Required File Structure.
3. Place the following jars in the `<Studio_Installation_Path>/docker/user/batchservice/lib/` directory:

   - `hive-exec-1.1.0-cdh5.13.0.jar`
   - `HiveJDBC4.jar`

- hive-metastore-1.1.0-cdh5.13.0.jar
- hive-service-1.1.0-cdh5.13.0.jar

> **NOTE**
> - The version of the jars are client/user-specific. These jars can be obtained from existing jars of Cloudera installation.
> - The HiveJDBC4.jar file is not available in the Cloudera setup. You must download the same from the Cloudera website.

4. Configure the config.sh file in <Studio_Installation_Path>/bin directory to replace the placeholder values as described in the following table:

> **NOTE** Do not alter the parameter values that are already set in the config.sh file

Table 1: Configuring config.sh File

Parameter	Description
NAMESPACE	Enter a value to create a namespace with the specified value.
URL_GLOBAL_GRAPH_CONFIG_JSON	Indicates the HDFS URL where the PGX graph configuration .json file is stored at the end of the ETL. The location can be either local or hdfs path. For example: hdfs:///user/fccstudio/graph.json
PGX_SERVER_NUM_REPLICAS	Indicates the number of replicas of the PGX server. For example: 1
PGX_GLOBAL_GRAPH_NAME	Indicates the name that the pre-loaded global graph is published with and the FCC Studio users can use to reference the global graph. For example: GlobalGraphIH

5. Grant Execute permission to the <Studio_Installation_Path>/bin directory using the following command:

```
chmod 755 install.sh config.sh
```

6. Run the following command:

```
./install.sh
```

> **NOTE**
> - Execution of the install.sh command does not generate any log file.
> - The values for the <URL_GLOBAL_GRAPH_CONFIG_JSON> and <PGX_GLOBAL_GRAPH_NAME> parameters in the <Studio_Installation_Path>/configmaps/pgx-server/pgx.conf file are auto-populated with the values that are configured in the <Studio_Installation_Path>/bin/config.sh file.

7. Navigate to the <Studio_Installation_Path>/configmaps/pgx-server/ directory and modify the pgx.conf file as follows:

Comment the following preload graph section:

```
<!--
"preload_graphs": [
  {
    "path": "<URL_GLOBAL_GRAPH_CONFIG_JSON>",
    "name": "<PGX_GLOBAL_GRAPH_NAME>"
  }
 ]
-->
```

4.7 Verifying the Resource Allocation for FCC Studio Services

The required resources must be allocated to the FCC Studio services as per the architecture. For information on resource allocation, see Appendix - Resource Allocation for FCC Studio Services.

4.8 Deploying FCC Studio on the K8s Cluster

To deploy FCC Studio on the K8s cluster, follow these steps:

1. Navigate to the `<Studio_Installation_Path>` directory.
2. Execute the following command:

   ```
   ./fcc-studio.sh --registry <registry URL>:<registry port>
   ```

   NOTE Refer to `./fcc-studio.sh -h` for usage instructions.

   After successful completion of deployment, the script displays a URL that can be used to access the FCC Studio Application. For more information, see Accessing the FCC Studio Application.
3. Verify the FCC Studio deployment. See Verifying the FCC Studio Deployment.

11. If you have added new data sources at this stage, you must redeploy FCC Studio. For more information, see Redeploying the FCC Studio Application.

4.9 Verifying the FCC Studio Deployment

To verify the FCC Studio deployment, follow these steps:

1. Wait for a minimum of 10 minutes, after completing the execution of the `./fcc-studio.sh` command, and run the following command:

   ```
   kubectl get pods -n <Namespace>
   ```

The pod details are displayed to indicate the status of the services. You can also check the logs of the FCC Studio services from the Kubernetes Dashboard. For more information, see Appendix - Checking Logs of FCC Studio Services.

NOTE	
	• For FCC Studio deployed with OFSAA, ensure all the pods are ready before accessing the FCC Studio application. • For FCC Studio deployed with non-OFSAA, ensure the metaservice is up and running before accessing the FCC Studio application..

4.10 Accessing the FCC Studio Application

Access the FCC Studio application. For more information, see Appendix - Accessing the FCC Studio Application.

5 Post-deployment Configuration

On the successful deployment of FCC Studio, you must perform the post-deployment configurations.

- For post-installation configuration for FCC Studio deployed with OFSAA, follow these steps:
  - Configuring the Interpreters
  - Performing the OFSAA Configuration for Batch Execution
  - Performing Hive Data Movement
  - Configuration for Running Published Notebooks
- For post-installation configuration for FCC Studio deployed with non-OFSAA, follow these steps:
  - Configuring the Interpreters
  - Cleaning the ICIJ Data
  - Configuring the FILEPATH for ICIJ

5.1 Configuring the Interpreters

After starting the FCC Studio application, you can perform the interpreter settings. For more information, see Appendix - Post-deployment Interpreter Settings

5.2 Performing the OFSAA Configuration for Batch Execution

To perform OFSAA configuration for batch execution, follow these steps:

1. Copy the files in the `<Studio_Installation_Path>/out/ficdb/bin` directory to the server where the BD or ECM pack is installed and to the `$FIC_DB_HOME/bin` directory of the OFSAA setup.
2. Execute the following command to grant Execute permission to the files:

   ```
   chmod +x <filenames>
   ```
3. Copy all the files in the `<Studio_Installation_Path>/out/ficdb/lib` directory to the `$FIC_DB_HOME/lib` directory.

   For information on running FCC Studio Batches, see *Managing Studio Batches* chapter in the *OFS Crime and Compliance Studio Administration Guide*.

5.3 Performing Hive Data Movement

To perform Hive data movement, follow these steps:

- Configuring Schema Creation
- Creating the Credential Keystore
- Configuring ETL

5.3.1 Configuring Schema Creation

- Configuring Schema Creation from FCC Studio Server
- Configuring Schema Creation from OFSAA Server

5.3.1.1 Configuring Schema Creation from FCC Studio Server

To configure Schema creation from FCC Studio server, follow these steps:

1. Set `FIC_DB_HOME` path to `<Studio_Installation_Path>/out/ficdb`.

   > **NOTE** The `$FIC_DB_HOME` path can be set from the `.profile` file as well.

2. Create a Hive Schema with the name mentioned in the `HIVE_SCHEMA` parameter in the `studio-env.yml` file.

   For information on `studio-env.yml` file, see Configuring the studio-env.yml File.

3. Execute the following shell script in the `<Studio_Installation_Path>/out/ficdb/bin/` directory to create tables in Hive Schema:

   ```
   FCCM_Studio_SchemaCreation.sh HIVE
   ```

   This creates tables in the Hive Schema.

4. Check Batch Service logs for more information.

5.3.1.2 Configuring Schema Creation from OFSAA Server

To configure Schema creation from OFSAA server, follow these steps:

1. Copy all the jar files located in the `<Studio_Installation_Path>/out/ficdb/lib` directory to the `<OFSAA_FIC_HOME_PATH>/ficdb/lib` directory.
2. Copy all the .sh files located in the `<Studio_Installation_Path>/out/ficdb/bin` directory to the `<OFSAA_FIC_HOME_PATH>/ficdb/bin` directory.
3. Create a Hive Schema with the name mentioned in the `HIVE_SCHEMA` parameter in the `studio-env.yml` file.

   For information on `studio-env.yml` file, see Configuring the studio-env.yml File.

4. Execute the following shell script in the `<OFSAA_FIC_HOME_PATH>/ficdb/bin` directory to create tables in the Hive Schema:

   ```
   FCCM_Studio_SchemaCreation.sh HIVE
   ```

   This creates tables in the Hive Schema.

5. Check Batch Service logs for more information.

5.3.2 Creating the Credential Keystore

To create a credential keystore, follow these steps:

1. Login as an HDFS SuperUser.
2. Create a credential keystore on HDFS using the following command:

   ```
   hadoop credential create mydb.password.alias -provider jceks://hdfs/
   user/root/oracle.password.jceks
   ```

3. Verify the credential keystore file using the following command:

   ```
   hadoop credential list -provider jceks://hdfs/user/root/
   oracle.password.jceks
   ```

4. Grant Read permission to the keystore file using the following command:

```
hadoop fs -chmod 744 /user/root/oracle.password.jceks
```

> **NOTE** Ensure the correct values of the credential keystore file path and the alias are provided in the `studio-env.yml` file.

5.3.3 Configuring ETL

Topics:

- Configuring Data Movement and Graph Load
- Cleaning the ICIJ Data
- Configuring the FILEPATH for ICIJ
- Configuring the Pre-load Global Graph for PGX Server

5.3.3.1 Configuring Data Movement and Graph Load

> **NOTE**
> - The Big Data System Administrator must place the `batchservice-8.0.7.4.0.jar` and the ES_Hadoop jar (`elasticsearch-spark-20_2.11-7.3.2.jar)` files in all nodes of the Spark cluster.
> - Ensure that the path of the jar files are present in the Spark `classpath` in the `spark-defaults.conf` file.
> - Ensure to remove the older batchservice jars from the Spark `classpath` in the `spark-defaults.conf` file.

To configure the Data Movement and Graph Load, follow these steps:

1. Copy the required `FCCM_Studio_SqoopJob.sh` files from the `<Studio_Installation_Path>/out/ficdb/bin` directory to the`<FIC_HOME of OFSAA_Installed_Path>/ficdb/bin` directory.

   For information on performing Data Movement and Graph Load, see the *Data Movement and Graph Loading for Big Data Environment* section in the *OFS Crime and Compliance Studio Administration Guide*.

5.3.3.2 Cleaning the ICIJ Data

To clean the ICIJ (International Consortium of Investigative Journalists) data, follow these steps:

1. Download the four dataset directories from *https://offshoreleaks.icij.org/pages/database*.
2. Unzip the four dataset directories and place the unzipped directories in the `<Studio_Installation_Path>/icij_data_cleaning` directory.
3. Navigate to the `<Studio_Installation_Path>/icij_data_cleaning/bin` directory and execute the following command:

   ```
   ./clean.sh
   ```

   > **NOTE** Ensure that Python 3 is available in the machine before executing this command.

   After successful execution of the command:

- The cleaned data is available for the sqoop job to load it in Hive and HDFS.
- A directory named "clean" is created inside each of the datasets where a clean version of each CSV file is created.

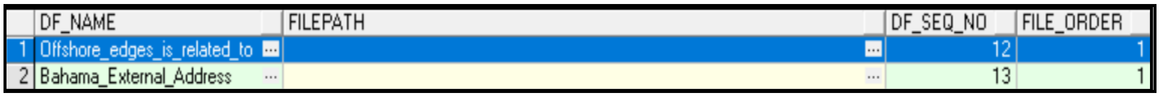
5.3.3.3 Configuring the FILEPATH for ICIJ

NOTE The FCC Studio graph model is configured to include ICIJ watch list files.

To configure the FILEPATH for ICIJ, follow these steps:

1. Place the watch list file in HDFS, which is accessible by the user.
2. Update the `FILEPATH` of the watch list files in the `fcc_studio_etl_files` table.

Figure 1: fcc_studio_etl_files Table



	DF_NAME	FILEPATH	DF_SEQ_NO	FILE_ORDER
1	Offshore_edges_is_related_to ...	...	12	1
2	Bahama_External_Address ...	...	13	1

5.3.3.4 Configuring the Pre-load Global Graph for PGX Server

NOTE

- Ignore this section if you do not want to the pre-load global graph.
- You must run at least one successful ETL batch to start the PGX service with the `graph.json` file located in the `URL_GLOBAL_GRAPH_CONFIG_JSON` path is present. For more information, see the *Data Movement and Graph Loading for Big Data Environment* section in the *OFS Crime and Compliance Studio Administration Guide*.

To configure the pre-load global graph for PGX server, follow these steps:

1. Navigate to the `<Studio_Installation_Path>/configmaps/pgx-server/` directory and modify the `pgx.conf` file as follows:

   Uncomment the following preload graph section in the `pgx.conf` file:

```
"preload_graphs": [

   {

      "path": "<URL_GLOBAL_GRAPH_CONFIG_JSON>",

      "name": "<PGX_GLOBAL_GRAPH_NAME>"

    }

  ]
```

NOTE The values for the `<URL_GLOBAL_GRAPH_CONGFIG_JSON> and <PGX_GLOBAL_GRAPH_NAME>` parameters in the `<Studio_Installation_Path>/configmaps/pgx-server/pgx.conf` file are auto-populated with the values that are configured in the `<Studio_Installation_Path>/bin/config.sh` file.

2. Execute the following command:

   ```
   kubectl -n <Namespace> delete configmap pgx-config
   ```

   ```
   kubectl -n <Namespace> delete -f  deployments/pgx-server.yml
   ```

3. Navigate to the `<Studio_Installation_Path>/out` directory and execute the following command:

   ```
   kubectl -n <Namespace> create configmap pgx-config --from-file=configmaps/pgx-server
   ```

   ```
   kubectl -n <Namespace> apply -f  deployments/pgx-server.yml
   ```

4. Check the Kubernetes Dashboard, Appendix - Checking Logs of FCC Studio Services, to ensure that the pgx-server service is up and running.

5. Execute the Graph_Alive notebook after each time you start/restart the PGX service. For more information, see Appendix - Executing Graph_Alive Notebook.

5.4 Configuration for Running Published Notebooks

To perform the configuration required to run published notebooks, follow these steps:

1. Copy the required `FCCM_Studio_NotebookExecution.sh` file from the `<Studio_Installation_Path>/out/ficdb/bin` directory to the `<FIC_HOME of OFSAA_Installed_Path>/ficdb/bin` directory.

   For information on performing Data Movement and Graph Load, see the *Executing Published Notebook* section in the *OFS Crime and Compliance Studio Administration Guide*.

6 Upgrading FCC Studio with OFSAA on the Kubernetes Cluster

This chapter provides the information required to understand the upgrade of the FCC Studio application with OFSAA on the Kubernetes cluster.

Topics:

- Additional Cleanup for Upgrade
- Upgrade Overview

6.1 Additional Cleanup for Upgrade

To perform the cleanup required for the upgrade, follow these steps:

1. Navigate to the DATABASECHANGELOG table in the Studio Schema.
2. Run the following SQL command in the Studio Schema:

   ```
   select  * from DatabaseChangeLog a where REGEXP_LIKE
   (a.id,'FCC_DATASTUDIO_CONFIG')
   ```

   The entries in the DATABASECHANGELOG table that match the criteria are displayed.
3. Delete all the entries generated as the result of the command.

   For example:

   ```
   FCC_DATASTUDIO_CONFIG_8.0.7.3.0
   ```

   ```
   FCC_DATASTUDIO_CONFIG_8.0.7
   ```

6.2 Upgrade Overview

You can upgrade an existing instance FCC Studio from v8.0.7.x onwards to v8.0.7.4.0 with OFSAA.

> **NOTE** Here, ensure to provide the same BD database, Studio schema, Hive schema, and wallet related information that you used during the installation of the existing instance of FCC Studio.

This section provides quick-start steps to upgrade FCC Studio.

Table 1: Quick Start Steps to Upgrade FCC Studio

Sl. No.	Steps	Reference Links
1.	Prepare for Deployment	**NOTE:** Ensure that the Prerequisites and the Prerequisite Environmental Settings are met before proceeding with the upgrade. 1. Performing Common Pre-installation Tasks a. Obtaining the Software b. Extracting the Software 2. Required File Structure 3. Interpreter Settings

Table 1: Quick Start Steps to Upgrade FCC Studio

Sl. No.	Steps	Reference Links
2.	Deploy FCC Studio with OFSAA on Kubernetes Cluster	1. Configuring the Elastic Search Component 2. Enabling Synonym/Stopword with the Elastic Search Service 3. Configuring Wallet 4. Configuring the Preferred Services 5. Configuring the studio-env.yml File 6. Configuring the ETL Services 7. Verifying the Resource Allocation for FCC Studio Services 8. Deploying FCC Studio on the K8s Cluster 9. Verifying the FCC Studio Deployment 10. Accessing the FCC Studio Application
3.	Post-deployment configuration for FCC Studio deployed with OFSAA	**NOTE:** The post-installation configuration that is not performed as part of the existing installation of FCC Studio must be performed during an upgrade. • Configuring the Interpreters **NOTE:** After upgrade, change the default value of the `zeppelin.python` property in the default fcc-python Interpreter to `python3.6` from the ***Interpreters*** page of the FCC Studio application UI. For more information, see Changing Python Version in the fcc-python Interpreter in the *Oracle Financial Services Crime and Compliance Studio Administration Guide*. • Performing the OFSAA Configuration for Batch Execution • Performing Hive Data Movement **NOTE:** During an upgrade, ensure that you have removed all the older jars and have placed the new jars in all nodes of the Spark cluster. For more information, see Configuring Data Movement and Graph Load. • Configuration for Running Published Notebooks

6.3 Configuring the studio-env.yml File for Upgrade

Configure the `studio-env.yml` file to upgrade FCC Studio with OFSAA. For more information, see Appendix - Configuring the studio-env.yml File.

7 Redeploying the FCC Studio Application

If the deployment of FCC Studio is unsuccessful, you must redeploy the application after performing the required clean up tasks.

To redeploy FCC Studio, follow these steps:

1. Execute the following command to delete the namespace of FCC Studio:

   ```
   kubectl delete namespace <Namespace>
   ```

2. Navigate to the `<Studio_Installation_Path>` path and correct the parameters or files as suggested by the error.
3. Navigate to `<Studio_Installation_Path>/secrets` directory and re-enter the values of sensitive information in the `studio-env.yml` file. For example, `STUDIO_DB_PASSWORD`, `HADOOP_PASSWORD-ALIAS`, and so on.
4. Perform database clean up by performing the following:

Schema	Applicable for OFSAA	Applicable for non-OFSAA
Cleanup for Studio Schema	Yes	Yes
Cleanup for BD or ECM Atomic Schema	Yes	No
Cleanup for BD or ECM Config Schema	Yes	No

5. Redeploy FCC Studio. For more information, see Deploying FCC Studio on the K8s Cluster.

7.0.1 Cleanup for Studio Schema

To clean up the Studio schema, follow these steps:

1. Drop the existing Studio schema and create a new Studio schema.

   > **NOTE** The username and password credentials of the Studio Schema in the wallet files must be updated accordingly. (If applicable)

2. Grant the following permissions to the newly created Oracle Database Schema:
   - `GRANT create session TO <Schema User>;`
   - `GRANT create table TO <Schema User>;`
   - `GRANT create view TO <Schema User>;`
   - `GRANT create any trigger TO <Schema User>;`
   - `GRANT create any procedure TO <Schema User>;`
   - `GRANT create sequence TO <Schema User>;`
   - `GRANT execute on dbms_rls TO <Schema User>;`
   - `GRANT execute on sys.dbms_session TO <Schema User>;`

- `ALTER USER <Schema User> QUOTA 100M ON users;`
- `GRANT create sequence TO <Schema User>;`
- `GRANT create SYNONYM TO <Schema User>;`
- `GRANT ALL privileges TO <Studio Schema User>;`

NOTE If dropping the schema is not an option, drop the tables and sequences as mentioned in Appendix - Tables and Sequences.

7.0.2 Cleanup for BD or ECM Atomic Schema

To clean up the BD or ECM Atomic Schema, follow these steps:

1. Log in to the BD or ECM Atomic Schema.
2. Truncate the DATABASECHANGELOG and DATABASECHANGELOGLOCK tables using the following command:

```
TRUNCATE TABLE DATABASECHANGELOGLOCK;

TRUNCATE TABLE DATABASECHANGELOG;
```

7.0.3 Cleanup for BD or ECM Config Schema

To clean up the BD or ECM Config schema, follow these steps:

1. Log in to the BD or ECM Config Schema.
2. Truncate the DATABASECHANGELOG and DATABASECHANGELOGLOCK tables using the following command:

```
TRUNCATE TABLE DATABASECHANGELOGLOCK;

TRUNCATE TABLE DATABASECHANGELOG;
```

8 Appendix - Pre-deployment Interpreter Settings

You must perform the interpreter settings for the required interpreters before deploying FCC Studio.

> **NOTE** Ensure to perform the following pre-installation settings only for the interpreters that you need.

Table 1: Pre-installation Interpreter Settings

Interpreter	Prerequisite Settings
fcc-jdbc	No additional configuration is required. **NOTE:** The FCC Studio application installed with non-OFSAA can use the Vanilla jdbc interpreter instead of the fcc-jdbc interpreter to connect to the Studio schema.
fcc-ore	For the required configuration, see Configuring the fcc-ore Interpreter.
fcc-pyspark	● Install the py4j package in the Spark cluster. ● Install the Livy server (0.5.0) on the master node of the Big Data cluster.
fcc-python.	For the required configuration, see Configuring the fcc-python Interpreter.
fcc-spark-scala	Install the Livy server (0.5.0) on the master node of the Big Data cluster.
fcc-spark-sql	Install the Livy server (0.5.0) on the master node of the Big Data cluster.
jdbc	No additional configuration is required.
md	No additional configuration is required.
pgql	No additional configuration is required.
pgx-algorithm	No additional configuration is required.
pgx-java	No additional configuration is required.
spark	For the required configuration, see Configuring Spark Interpreter.
pyspark	For the required configuration, see Configuring PySpark Interpreter.

Topics:

- Configuring the fcc-ore Interpreter
- Configuring the fcc-python Interpreter
- Configuring Spark Interpreter
- Configuring PySpark Interpreter

8.1 Configuring the fcc-ore Interpreter

8.1.1 Installing the Oracle R Distribution

To install the Oracle R Distribution (ORD), enable the **addons** and **optional_latest** channels in yum as shown in the following code:

~~~
```bash

(root)# yum-config-manager --enable ol7_addons

(root)# yum-config-manager --enable ol7_optional_latest

```
~~~

After completing the previous step, pull ORD from the yum repository using the following command:

~~~
```bash

(root)# yum install R.x86_64 R-core-extra

```
~~~

To install ORD, see *https://docs.oracle.com/cd/E83411_01/OREAD/installing-R-for-ORE.htm#OREAD129*.

8.2 Configuring the fcc-python Interpreter

The fcc-python interpreter image includes the following:

- The support for python2 has been deprecated.
- By default, the Python interpreter points to the Python 3.6 version.
- The following Python libraries are part of the fcc-python interpreter images for Python 3.6 version:
  - `pandas 0.25.3`
  - `numpy 1.17.4`
  - `scipy 1.3.2`
  - `scikit-learn 0.21.3`
  - `matplot-lib 3.1.1`
  - `seaborn 0.9.0`
  - `cx-oracle 7.2.2`
  - `sqlalchemy 1.3.11`

To modify the Python packages in the Python 3.6 or to add a different version of Python to the Python interpreter, see Modifying the Python Images for the Python Interpreter.

8.3 Configuring Spark Interpreter

- Prerequisites
- Configuration

8.3.1 Prerequisites

8.3.1.1 Setting Spark Master

To operate the Spark interpreter in local mode or Yarn mode, the `spark.master` property must be set accordingly.

You can update the Spark Master by changing the default value of the `spark.master` property in the Interpreter Settings of the Spark interpreter (spark.json) or the ***Interpreters*** page of the FCC Studio application UI after startup.

- The default value of `spark.master` property is `local[*]`, which means that the interpreter will run in local mode.
- For a Yarn cluster, you must change the default value of `spark.master` property to yarn-client.

> **NOTE** The Hadoop client-configuration files are required to connect to a Yarn cluster.

8.3.1.2 Local Mode

No additional configuration is required to operate the Spark interpreter in local mode.

8.3.1.3 Yarn Mode

To operate the Spark interpreter in yarn mode, follow these steps:

- **Provide Custom Spark libraries**

  To provide your own Spark libraries and/or Hadoop client-configuration files to connect to a Yarn cluster, follow these steps to create a new Init Container/Configmap that copies your Spark libraries directory and/or Hadoop client-configuration directory into the right location, where the Spark interpreter can access it.

  To provide Spark libraries, follow these steps:

  a. Download the desired Spark libraries from the *Spark Official Website*.

  b. Prepare Spark libraries.

     i. Place your libraries in the `<Studio_Installation_Path>/docker/user/spark-interpreter-libraries/` directory.

> **NOTE** Ensure that the directory name of the Spark library is prefixed with the term ’spark’

     ii. To use separate Hadoop libraries, download them as well and place the Hadoop libraries directory in the same directory.

     iii. Be cautious when linking the two libraries, since the path where they are located is in the K8s pod, and the location is as follows:

     `/var/olds-spark-interpreter/interpreter/spark/libs/`

  c. Change the image of the Spark interpreter Init Container in the `<Studio_Installation_Path>/deployments/spark.yml` file to `{{imageRepository}}/fcc-studio/3rd-party:init.`

  d. (Optional) Place your Hadoop Client Configuration files in the `<Studio_Installation_Path>/configmaps/spark-interpreter-conf/` directory.

- The cluster's Hadoop client-side configuration files that include XML files such as `yarn-site.xml` are required and must be supplied with the Spark libraries. These files are available in the Hadoop configuration directory (`HADOOP_CONF_DIR)` of the cluster or can be downloaded from the cluster manager's UI if you are using a Cloudera cluster.

8.3.2 Configuration

Spark interpreter configuration can be divided into the following categories:

- Configuration related to deployment

  These properties can be set either in the Spark libraries, for example, the `spark-defaults.conf` file, or through the system environment variable, `SPARK_CONF`, for example, `SPARK_CONF="--conf spark.driver.memory=2g"`.

  > **NOTE** These properties cannot be changed when the Spark interpreter is running.

- Configuration related to Spark runtime control

  These properties can be set from the ***Interpreters*** page of the FCC Studio application UI, this includes properties such as `spark.executor.memory`.

  > **NOTE** The properties related to the driver cannot be set during runtime and are considered deployment configuration. The properties related to the executors can be set during runtime. Hence, the latter option of runtime control configuration is preferred.

A list of possible properties can be found in the *Spark Official Documentation*. All the properties prefixed with the term "zeppelin" that are listed in the *Zeppelin Spark Configuration Document*, can also be set through the ***Interpreters*** page of the FCC Studio application UI.

8.4 Configuring PySpark Interpreter

- Prerequisites
- Configuration

8.4.1 Prerequisites

The PySpark interpreter has the same prerequisites as that of the Spark interpreter. For more information, see Configuring Spark Interpreter. Also, all Spark components must be configured to use the same Python version.

8.4.2 Configuration

The PySpark interpreter can be configured through the Spark interpreter with the only exception being the Python version used. By default, the Python version is set to 3, which can be changed either in the interpreter JSON files before the startup or from the ***Interpreters*** page of the FCC Studio application UI during runtime by changing the following properties:

- In the ***Spark Interpreter Settings*** page of the FCC Studio application UI (or `spark.json` file), change the value of the `spark.pyspark.python` property to the path of the Python executable that is to be used by the Spark executors.
- In the ***PySpark Interpreter Settings*** page of the FCC Studio application UI (or `pyspark.json` file), change the value of the `zeppelin.pyspark.python` property to the path of the Python executable that is to be used by the Spark driver.

To ensure that the two Python versions match in the case where your components run on different machines, you can use the Appendix - Using Python Virtual Environments with PySpark. This step is not required if different machines have matching Python versions available.

9 Appendix - Setting Up Password Stores with Oracle Wallet

Topics:

- Overview
- Setting Up Password Stores for Database User Accounts
- Verifying the Connectivity of the Wallet

9.1 Overview

As part of an application installation, administrators must set up password stores for database user accounts using Oracle Wallet. These password stores must be installed on the application database side. The installer handles much of this process, the administrators must perform some additional steps.

A password store for the application and application server user accounts must also be installed; however, the installer takes care of this entire process.

9.2 Setting Up Password Stores for Database User Accounts

After the database is installed and the default database user accounts are set up, administrators must set up a password store using the Oracle Wallet. This involves assigning an alias for the username and associated password for each database user account. The alias is used later during the application installation. This password store must be created on the system where the application server and database client are installed.

This section describes the steps to set up a wallet and the aliases for the database user accounts. For more information on configuring authentication and password stores, see the *Oracle Database Security Guide*.

> **NOTE** In this section, `<wallet_location>` is a placeholder text for illustration purposes. Before running the command, ensure that you have already created the `<wallet_location>` directory where you want to create and store the wallet.

To set up a password store for the database user accounts, follow these steps:

1. Login to the server as a Linux user.
2. Create a wallet in the `<wallet_location>` using the following command:

   ```
   mkstore -wrl  -create
   ```

   After you run the command, a prompt appears. Enter a password for the Oracle Wallet in the prompt.

   > **NOTE** The mkstore utility is included in the Oracle Database Client installation.

   The wallet is created with the auto-login feature enabled. This feature enables the database client to access the wallet contents without using the password. For more information, see the *Oracle Database Advanced Security Administration Guide*.
3. Create the database connection credentials in the wallet using the following command:

```
mkstore -wrl <wallet_location> -createCredential <alias-name> <database-user-name>
```

Run this command for the following <alias-name>:

Table 1:

Schema	Applicable for OFSAA	Applicable for non-OFSAA
BD_Config_Schema	Yes	No
BD_Atomic_Schema	Yes	No
Studio_Schema	Yes	Yes

After you run the command, a prompt appears. Enter the password associated with the database user account in the prompt. You are prompted to re-enter the password. Then you are prompted for the wallet password used in Step 1.

4. Repeat step 2 for all the database user accounts.
5. Update the `tnsnames.ora` file to include the following entry for each alias name to be set up.

```
<alias-name> =

(DESCRIPTION =

(ADDRESS_LIST =

(ADDRESS = (PROTOCOL = TCP) (HOST = <host>) (PORT = <port>))

)

(CONNECT_DATA =

(SERVICE_NAME = <service>)

)

)
```

NOTE

- You can either update the existing `tnsnames.ora` file with these details or create a new `tnsnames.ora` file and make required entries.
- `<alias-name>` is a user-defined value.

9.3 Verifying the Connectivity of the Wallet

To verify the connectivity of the wallet, follow these steps:

1. Create a `sqlnet.ora` in the wallet directory using the following content:

```
WALLET_LOCATION = (SOURCE = (METHOD = FILE) (METHOD_DATA = (DIRECTORY =
<Wallet_Location>)) )

SQLNET.WALLET_OVERRIDE=TRUE

SSL_CLIENT_AUTHENTICATION=FALSE
```

2. Test the connectivity using the following command:

NOTE	The `ORACLE_HOME` used with the wallet must be the same version or higher than what the wallet was created with.

   `$ export WALLET_LOCATION=<wallet_location>`

   `$ export TNS_ADMIN=<tnsnames.ora_location>,` Here ensure to use the wallet to point to the alternate `tnsnames.ora` as created above.

   `$ sqlplus /@<alias_name>`

   The output is similar to:

   `SQL*Plus: Release 11`

   Connected to:

   Oracle Database 12c

   To verify if you are connected to the correct user:

   `SQL> show user`

   The output is similar to:

   `USER is "<database-user-name>"`

10 Appendix - Configuring the Elastic Search component

To configure the Elastic Search component, follow these steps:

> - Ensure that a minimum of 4GB free RAM space is available for elastic search. If RAM is low, the shards of the elastic search fails and correct result is not fetched.
> - You must manually clean the cache if facing performance issue.

1. Navigate to the `elasticsearch-7.3.2/config` directory.
2. Configure the `elasticsearch.yml` file as follows:

Table 1: elasticsearch.yml File

InteractionVariable Name	Significance
cluster.name	Indicates the name of the cluster.
node.name	Indicates the name given for the node.
node.master	Indicates whether the node is a master.
node.data	Indicates the node data.
path.data	Indicates the directory where you want to store the data.
path.logs	Indicates the directory where you want to store the logs.
network.host	Indicates the hostname of the machine where you want to install elastic search.
http.port	Indicates the port number where the elastic search is installed.
discovery.seed_hosts	(Optional) Indicates the hostnames of the nodes of the cluster.
cluster.initial_master_nodes	(Optional) Indicates the number given to the nodes of the cluster.
indices.breaker.total.use_real_memory	• Indicates the static setting to determine whether the parent breaker must consider the real memory usage into account or only consider the amount that is reserved by the child circuit breakers. • This setting is used to prevent the `OutOfMemory` error.

3. Configure the `jvm.options` file as follows:

Table 2: elasticsearch.yml File

InteractionVariable Name	Significance
-Xms1g	● Indicates the maximum and minimum heap memory size (mainly used for storing graphs' string properties) for the Java process of PGX. ● Set the value for these parameters. ● The maximum value set can be up to 50% of the ram size of the machine. ● Recommended Value: Less than 32GB.
-Xmx1g	

4. Enter the URL in the following format into the browser:

   `http://<network.host>:<http.port>`

   The following output is displayed to indicate the successful installation of Elastic Search.

```
{
  "name" : "node-1",
  "cluster_name" : "my-application",
  "cluster_uuid" : "_D-2qEGGSgKQPd3i-UtgWQ",
  "version" : {
    "number" : "7.3.2",
    "build_flavor" : "oss",
    "build_type" : "tar",
    "build_hash" : "1c1faf1",
    "build_date" : "2019-09-06T14:40:30.409026Z",
    "build_snapshot" : false,
    "lucene_version" : "8.1.0",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

11 Appendix - Resource Allocation for FCC Studio Services

Topics:

- Resource Limits
- Resource Types
- Resource Parameters in FCC Studio

11.1 Resource Limits

For FCC Studio to run reliably, the available resources of the Kubernetes cluster have to be allocated accordingly. The components are mainly memory intensive and therefore we recommend setting memory constraints for each component.

11.2 Resource Types

Each container requires a memory request and memory limit size as defined by the Kubernetes API. In short, containers specify a request, which is the amount of that resource that the system will guarantee to the container and a limit which is the maximum amount that the system will allow the container to use. For more information on troubleshooting tips, see Managing Compute Resources for Containers.

Some components require additional resource limits which are set as environment variables.

11.3 Resource Parameters in FCC Studio

After extracting the FCC Studio application installer software, the resource limits have to be adjusted for each component. The configuration files can be found in the `<Studio_Installation_Path>` directory.

NOTE

- The sizing recommendations are preliminary. In the case of deployment failures, a manual configuration of the sizing parameters is required.
- Depending on the use case, the recommended value changes.
- The default value in the following table is the value that is already set in the file.

Table 1: Resource Parameters in FCC Studio

Configuration File/Container	Parameter type	Parameter Name	Description	Recommendation
server.yml / server	k8	spec.containers[].resources.requests.memory	Memory request size for the FCC server (web application) component.	default
	k8	spec.containers[].resources.requests.memory	Memory limit size for the FCC server (web application) component.	default

Table 1: Resource Parameters in FCC Studio

Configuration File/Container	Parameter type	Parameter Name	Description	Recommendation
agent.yml / agent	k8	spec.containers[].resources.requests.memory	Memory request size for the Agent (manages all interpreters) component.	default
	k8	spec.containers[].resources.limits.memory	Memory limit size for the Agent (manages all interpreters) component.	default
pgx-server.yml / pgx-server	k8	spec.containers[].resources.requests.memory	Memory request size for the PGX server (manages graph processing) component.	Slightly less than the memory of the PGX server as calculated in the sizing guide.
	k8	spec.containers[].resources.requests.memory	Memory limit size for the PGX server (manages graph processing) component.	The same as the request size above.
	ENV VAR (JAVA_OPTS)	-Xmx -Xms	The maximum and minimum heap memory size (mainly used for storing graphs' string properties) for the Java process of PGX.	58% of the container's memory limit size above. For a better understanding of this sizing parameter, please consult the PGX 20.0.2 Memory Consumption documentation.
	ENV VAR (JAVA_OPTS)	-Dpgx.max_off_heap_size	The maximum off-heap memory size in megabytes (mainly used for storing graphs except for their string properties) that PGX tries to respect.	42% of the container's memory limit size above. For a better understanding of this sizing parameter, please consult the PGX 20.0.2 Memory Consumption documentation.
fcc-pgx.yml / pgx-interpreter	k8	spec.containers[].resources.requests.memory	Memory request size for the PGX interpreter.	4Gi
	k8	spec.containers[].resources.limits.memory	Memory limit size for the PGX interpreter.	16Gi Sizing should depend on the number and behavior (memory requirements of sessions) of concurrent users

Table 1: Resource Parameters in FCC Studio

Configuration File/Container	Parameter type	Parameter Name	Description	Recommendation
authservice.yml / authservice	k8	spec.containers[].resources.requests.memory	Memory request size for the authservice (used for getting roles of a user from DB) component.	default
	k8	spec.containers[].resources.limits.memory	Memory limit size for the authservice (used for getting roles of a user from DB) component.	default
metaservice.yml / metaservice	k8	spec.containers[].resources.requests.memory	Memory request size for the metaservice (used for custom interpreter api's like loaddataset, listdataset in scala interpreter etc.) component.	default
	k8	spec.containers[].resources.limits.memory	Memory limit size for the metaservice (used for custom interpreter api's like loaddataset, listdataset in scala interpreter etc.) component.	default
sessionservice.yml / sessionservice	k8	spec.containers[].resources.requests.memory	Memory request size for the sessionservice (used for managing session between pgx and scala interpreter) component.	default
	k8	spec.containers[].resources.limits.memory	Memory limit size for the sessionservice (used for managing session between pgx and scala interpreter) component.	default
batchservice.yml / batchservice	k8	spec.containers[].resources.requests.memory	Memory request size for the batchservice (used for managing batches like sqoopjob, graph load, notebook execution etc) component.	default
	k8	spec.containers[].resources.limits.memory	Memory limit size for the batchservice (used for managing batches like sqoopjob, graph load, notebook execution etc) component.	default

Table 1: Resource Parameters in FCC Studio

Configuration File/Container	Parameter type	Parameter Name	Description	Recommendation
entity-resolution.yml/ entity resolution	k8	spec.containers[].resources.requests.memory	Memory request size for the Entity Resolution component.	default
	k8	spec.containers[].resources.limits.memory	Memory limit size for the Entity Resolution component.	default
matching-service.yml/ matching service	k8	spec.containers[].resources.requests.memory	Memory request size for the Matching Service component.	default
	k8	spec.containers[].resources.limits.memory	Memory limit size for the Matching Service component.	default
spark.yml/spark and pyspark Interpreter	k8	spec.containers[].resources.requests.memory	Memory request size for the Spark interpreter.	default
	k8	spec.containers[].resources.limits.memory	Memory limit size for the Spark interpreter.	default
fcc-jdbc.yml / fcc-jdbc	k8	spec.containers[].resources.requests.memory	Memory request size for the jdbc connection.	default
	k8	spec.containers[].resources.limits.memory	Memory limit size for the jdbc connection.	default
fcc-livy.yml / fcc-spark-scala, fcc-spark-sql, and fcc-pyspark interpreters	k8	spec.containers[].resources.requests.memory	Memory request size for the livy connection to big data Spark cluster.	default
	k8	spec.containers[].resources.limits.memory	Memory limit size for the livy connection to big data Spark cluster.	default
fcc-markdown.yml / markdown-interpreter	k8	spec.containers[].resources.requests.memory	Memory request size for the Markdown interpreter.	default
	k8	spec.containers[].resources.limits.memory	Memory limit size for the Markdown interpreter.	default

Table 1: Resource Parameters in FCC Studio

Configuration File/Container	Parameter type	Parameter Name	Description	Recommendation
fcc-ore.yml / ore-interpreter	k8	spec.containers[].resources.requests.memory	Memory request size for the ore connection.	default
	k8	spec.containers[].resources.limits.memory	Memory limit size for the ore connection.	default
fcc-python.yml / python-interpreter	k8	spec.containers[].resources.requests.memory	Memory request size for the Python interpreter.	depending on use case
	k8	spec.containers[].resources.limits.memory	Memory limit size for the Python interpreter.	depending on use case

12 Appendix - Configuring the studio-env.yml File

To configure the `studio-env.yml` file for installing FCC Studio, follow these steps:

1. Login to the server as a non-root user.
2. Navigate to the `<Studio_Installation_Path>/bin/` directory.
3. Configure the `studio-env.yml` file as follows:

NOTE

- Do not alter the parameter values that are already set in the `studio-env.yml` file
- Retain the existing placeholder values for the non-mandatory parameters.
- You must manually set the parameter value in the `studio-env.yml` file. If a value is not applicable, enter NA and ensure that the value is not entered as NULL.
- Depending on the installation architecture, ensure to provide the correct hostname for URL of PGX service in the `studio-env.yml` file.
- During upgrading FCC Studio with OFSAA, ensure to provide the same BD database, Studio schema, Hive schema, wallet related information that you used during the installation of the existing instance FCC Studio.
- During upgrading FCC Studio with non-OFSAA, ensure to provide the same Studio schema and wallet related information that you used during the installation of the existing instance of FCC Studio.

Table 1: studio-env.yml Parameters

Parameter	Significance	Installing with OFSAA (Mandatory)	Upgrading with OFSAA (Mandatory)	Installing with non-OFSAA (Mandatory)
apiVersion	For example: v1	Yes	Yes	Yes
kind	For example: Secret	Yes	Yes	Yes
metadata				
name	For example: studio-env	Yes	Yes	Yes

Table 1: studio-env.yml Parameters

Parameter	Significance	Installing with OFSAA (Mandatory)	Upgrading with OFSAA (Mandatory)	Installing with non-OFSAA (Mandatory)
stringData				
NON_OFSAA	• To install FCC Studio with OFSAA on the K8s cluster, enter “false”. • To install FCC Studio with non-OFSAA on the K8s cluster, enter “true”.	Enter false	Enter false	Enter true
REALM	Realm indicates functional grouping of database schemas and roles that must be secured for an application. Realms protect data from access through system privileges; realms do not give additional privileges to its owner or participants. FCC Studio uses realm based authorization and authentication for its users. For more information, see the *Realm Based Authorization for FCC Studio* section in the *OFS Crime and Compliance Studio Administration Guide*. The FCC Studio application can be accessed using the following realms: • FCCMRealm Value=`com.oracle.ofss.fccm.studio.datastudio.auth.FCCMRealm` • IdcsRealm Value=`oracle.datastudio.realm.idcs.IdcsRealm` • DemoRealm Value=`com.oracle.ofss.fccm.studio.datastudio.auth.DemoRealm` **NOTE:** The DemoRealm is used only for demo purpose and is not recommended for usage.	Yes	Yes	Yes

Table 1: studio-env.yml Parameters

Parameter	Significance	Installing with OFSAA (Mandatory)	Upgrading with OFSAA (Mandatory)	Installing with non-OFSAA (Mandatory)
OFSAA_SERVICE_URL	Indicates the URL of the OFSAA instance. Do not enter '/' at the end of the URL. **NOTE:** For OFSAAAI, the value must be in the following format: `https://<HostName>:<PortNo>/<ContextName>/rest-api`	Yes	Yes	No
LIVY_HOST_URL	Indicates the URL of the Livy application. Example: `http://<HostName>:<PortNo>` **NOTE:** This parameter is applicable only if fcc-spark-sql, fcc-spark-scala, and/or fcc-pyspark interpreters are to be used.	No	No	No
DB Details for Studio Schema				
STUDIO_DB_HOSTNAME	Indicates the hostname of the database where Studio schema is created.	Yes	Yes	Yes
STUDIO_DB_PORT	Indicates the port number where Studio schema is created.	Yes	Yes	Yes
STUDIO_DB_SERVICE_NAME	Indicates the service name of the database where Studio schema is created.	Yes	Yes	Yes
STUDIO_DB_SID	Indicates the SID of the database where Studio schema is created.	Yes	Yes	Yes
STUDIO_DB_USERNAME	Indicates the username of the Studio Schema (newly created Oracle Schema).	Yes	Yes	Yes
STUDIO_DB_PASSWORD	Indicates the password for the newly created schema.	Yes	Yes	Yes

Table 1: studio-env.yml Parameters

Parameter	Significance	Installing with OFSAA (Mandatory)	Upgrading with OFSAA (Mandatory)	Installing with non-OFSAA (Mandatory)
Studio DB Wallet Details For more information on creating wallet, see Appendix - Setting Up Password Stores with Oracle Wallet.				
STUDIO_ALIAS_NAME	Indicates the Studio alias name. **NOTE:** Enter the alias name that was created during wallet creation.	Yes	Yes	Yes
STUDIO_WALLET_LOCATION	Indicates the Studio wallet location.	Yes	Yes	Yes
STUDIO_TNS_ADMIN_PATH	Indicates the path of the `tnsnames.ora` file where an entry for the `STUDIO_ALIAS_NAME` is present.	Yes	Yes	Yes
DB Details for BD Config Schema				
BD_CONFIG_HOSTNAME	Indicates the hostname of the database where BD or ECM config schema is installed.	Yes	Yes	Enter NA
BD_CONFIG_PORT	Indicates the port of the database where BD or ECM config schema is installed.	Yes	Yes	Enter NA
BD_CONFIG_SERVICE_NAME	Indicates the service name of the database where BD or ECM config schema is installed.	Yes	Yes	Enter NA
BD_CONFIG_SID	Indicates the SID of the database where BD or ECM config schema is installed.	Yes	Yes	Enter NA
BD_CONFIG_USERNAME	Indicates the username for the BD or ECM config schema.	Yes	Yes	Enter NA
BD_CONFIG_PASSWORD	Indicates the password for the BD or ECM config schema.	Yes	Yes	Enter NA

Table 1: studio-env.yml Parameters

Parameter	Significance	Installing with OFSAA (Mandatory)	Upgrading with OFSAA (Mandatory)	Installing with non-OFSAA (Mandatory)
BD Config Wallet Details For more information on creating wallet, see Appendix - Setting Up Password Stores with Oracle Wallet.				
BD_CONFIG_ALIAS_NAME	Indicates the BD or ECM config alias name. **NOTE:** Enter the alias name that was created during wallet creation.	Yes	Yes	Enter NA
BD_CONFIG_WALLET_LOCATION	Indicates the BD or ECM config wallet location.	Yes	Yes	Enter NA
BD_CONFIG_TNS_ADMIN_PATH	Indicates the path of the `tnsnames.ora` file where an entry for the `BD_CONFIG_ALIAS_NAME` is present.	Yes	Yes	Enter NA
DB Details for BD Atomic Schema				
BD_ATOMIC_HOSTNAME	Indicates the BD or ECM atomic schema hostname.	Yes	Yes	Enter NA
BD_ATOMIC_PORT	Indicates the BD or ECM atomic schema port number.	Yes	Yes	Enter NA
BD_ATOMIC_SERVICE_NAME	Indicates the BD or ECM atomic schema service name.	Yes	Yes	Enter NA
BD_ATOMIC_SID	Indicates the BD or ECM atomic schema SID.	Yes	Yes	Enter NA
BD_ATOMIC_USERNAME	Indicates the username of the BD or ECM atomic schema.	Yes	Yes	Enter NA
BD_ATOMIC_PASSWORD	Indicates the password of the BD or ECM atomic schema.	Yes	Yes	Enter NA

Table 1: studio-env.yml Parameters

Parameter	Significance	Installing with OFSAA (Mandatory)	Upgrading with OFSAA (Mandatory)	Installing with non-OFSAA (Mandatory)
BD Atomic Wallet Details. For more information on creating wallet, see Appendix - Setting Up Password Stores with Oracle Wallet.				
BD_ATOMIC_ALIAS_NAME	Indicates the BD or ECM atomic alias name. **NOTE:** Enter the alias name that was created during wallet creation.	Yes	Yes	Enter NA
BD_ATOMIC_WALLET_LOCATION	Indicates the BD or ECM atomic wallet location.	Yes	Yes	Enter NA
BD_ATOMIC_TNS_ADMIN_PATH	Indicates the path of the tnsnames.ora file where an entry for the BD_ATOMIC_ALIAS_NAME is present.	Yes	Yes	Enter NA
SQL Scripts				
FSINFODOM	Indicates the name of the BD or ECM Infodom.	Yes	Yes	Enter NA
FSSEGMENT	Indicates the name of the BD or ECM segment.	Yes	Yes	Enter NA
DATAMOVEMENT_LINK_NAME	● If the newly created schema is in a different database host, you must create a DB link and provide the same link in this parameter. Alternatively, you can provide the source schema name. If no DB link is present, provide the <SCHEMA_NAME> in this parameter. ● If the newly created schema is in the same database host, the value for this parameter is the user name of the BD or ECM atomic schema.	Yes	Yes	Enter NA
DATAMOVEMENT_LINK_TYPE	If the DB link is used, enter DBLINK in this field. If the DB link is not used, enter SCHEMA in this field.	Yes	Yes	Enter NA

Table 1: studio-env.yml Parameters

Parameter	Significance	Installing with OFSAA (Mandatory)	Upgrading with OFSAA (Mandatory)	Installing with non-OFSAA (Mandatory)
Cloudera Setup Details Contact System Administrator to obtain the required details for these parameters.				
HADOOP_CREDENTIAL_PROVIDER_PATH	Indicates the path where Hadoop credential is stored.	Yes	Yes	Enter NA
HADOOP_PASSWORD_ALIAS	Indicates the Hadoop alias given when creating the Hadoop credentials. **NOTE:** Enter the alias name that was created during wallet creation. For information on how to create a credential keystore, see Creating the Credential Keystore	Yes	Yes	Enter NA
Hive_Host_Name	Indicates the Hive hostname.	Yes	Yes	Enter NA
Hive_Port_number	Indicates the Hive port number. Contact System Administrator to obtain the port number.	Yes	Yes	Enter NA
HIVE_PRINCIPAL	Indicates the Hive Principal. Contact System Administrator to obtain HIVE_PRINCIPAL.	Yes	Yes	Enter NA
HIVE_SCHEMA	Indicates the new Hive schema name.	Yes	Yes	Enter NA
JAAS_CONF_FILE_PATH	Created for future use.	No	No	No
Krb_Host_FQDN_Name	Indicates the Kerberos host FQDN name.	Yes	Yes	Enter NA
Krb_Realm_Name	Indicates the Kerberos realm name.	Yes	Yes	Enter NA
Krb_Service_Name	Indicates the Kerberos service name. Example: Hive	Yes	Yes	Enter NA

Table 1: studio-env.yml Parameters

Parameter	Significance	Installing with OFSAA (Mandatory)	Upgrading with OFSAA (Mandatory)	Installing with non-OFSAA (Mandatory)
KRB5_CONF_FILE_PATH	Created for future use.	No	No	No
security_krb5_kdc_server	Created for future use.	No	No	No
security_krb5_realm	Created for future use.	No	No	No
server_kerberos_keytab_file	Created for future use.	Yes	Yes	Enter NA
server_kerberos_principal	Created for future use.	Yes	Yes	Enter NA
SQOOP_HOSTMACHINE_USER_NAME	Indicates the user name of the Big Data server where SQOOP will run.	Yes	Yes	Enter NA
SQOOP_PARAMFILE_PATH	1. Create a file with the name `sqoop.properties` in the Big Data server and add the following entry to the same: `oracle.jdbc.mapDateToTimestamp=false` 2. Enter the location of the `sqoop.properties` file as the value for this parameter. Example: `/scratch/ofsaa/` **NOTE:** Ensure that the location name ends with a '/'.	Yes	Yes	Enter NA
SQOOP_PARTITION_COL	Indicates the column in which the HIVE table is partitioned. The value must be `SNAPSHOT_DT`	Yes	Yes	Enter NA
SQOOP_TRG_HOSTNAME	Indicates the hostname of the Big Data server where SQOOP will run. Example: `<HostName>`	Yes	Yes	Enter NA
SQOOP_TRG_PASSWORD	Indicates the password of the user of the Big Data server where SQOOP will run.	Yes	Yes	Enter NA
SQOOP_WORKDIR_HDFS	Indicates the SQOOP working directory in HDFS. Example: `/user/ofsaa`	Yes	Yes	Enter NA

Table 1: studio-env.yml Parameters

Parameter	Significance	Installing with OFSAA (Mandatory)	Upgrading with OFSAA (Mandatory)	Installing with non-OFSAA (Mandatory)
Internal Services				
AUTH_SERVICE_URL	Indicates the AUTH service URL that gets activated after the `fcc-studio.sh` file runs. Example: `http://<HostName>:7041/authservice`	Yes	Yes	No
BATCH_SERVICE_URL	Indicates the Batch service URL that gets activated after the `fcc-studio.sh` file runs. Example: `http://<HostName>:7043/batchservice`	Yes	Yes	Yes
META_SERVICE_URL	Indicates the META service URL that gets activated after the `fcc-studio.sh` file runs. Example: `http://<HostName>:7045/metaservice`	Yes	Yes	Yes
SESSION_SERVICE_URL	Indicates the Session service URL that gets activated after the `fcc-studio.sh` file runs. Example: `http://<HostName>:7047/sessionservice`	Yes	Yes	Yes
PGX_SERVER_URL	Indicates the URL of the PGX server. Example: http://<HostName>:<PortNo> The value for PortNo must be 7007.	Yes	Yes	Yes
ORE Interpreter Settings **NOTE:** This section is applicable only if ORE interpreter is to be used.				

Table 1: studio-env.yml Parameters

Parameter	Significance	Installing with OFSAA (Mandatory)	Upgrading with OFSAA (Mandatory)	Installing with non-OFSAA (Mandatory)
RSERVE_USERNAME	Indicates the RServe username. Value: oml	No	No	No
RSERVE_PASSWORD	Indicates the RServe password. Value: password	No	No	No
HTTP_PROXY	Indicates the proxy for the host where FCC Studio is deployed.	No	No	No
HTTPS_PROXY	Indicates the proxy for the host where FCC Studio is deployed.	No	No	No
REPO_CRAN_URL	Indicates the URL from where the R packages are obtained. The format for the REPO_CRAN_URL is as follows: *https://cran.r-project.org/*	No	No	No
USERS_LIB_PATH	Indicates the path where the R packages are installed. Default value: `/usr/lib64/R/library`	No	No	No
RSERVE_CONF_PATH	Indicates the path where the `Rserve.conf` file is present. Default value: `/var/ore-interpreter/rserve`	No	No	No
ElasticSearch Cluster details				
ELASTIC_SEARCH_HOSTNAME	Indicates the hostname of the database where the elastic search service is installed.	Yes	Yes	Yes
ELASTIC_SEARCH_PORT	Indicates the port number where the elastic search service is installed.	Yes	Yes	Yes
Matching Service				
EXECUTOR_THREADS	Indicates the number of threads to run in parallel during one scroll. For example: 10	Yes	Yes	Yes
SCROLL_TIME	Indicates the duration for which the scroll_size output is active. For example: 5	Yes	Yes	Yes
SCROLL_SIZE	Indicates the amount of data that must be obtained in one attempt when a query is fired on an index in the elastic search service. For example: 1000	Yes	Yes	Yes

Table 1: studio-env.yml Parameters

Parameter	Significance	Installing with OFSAA (Mandatory)	Upgrading with OFSAA (Mandatory)	Installing with non-OFSAA (Mandatory)
ELASTICRESPONSE_BUFFERLIMIT_BYTE	Indicates the buffer size of the response obtained from the elastic search service. For example: 200	Yes	Yes	Yes
MATCHING_SERVICE_HOSTNAME	Indicates the hostname of the database where matching service is installed.	Yes	Yes	Yes
MATCHING_SERVICE_PORT	Indicates the port number where matching service is installed.	Yes	Yes	Yes
ER_SERVICE_URL	Indicates the URL of the entity resolution service. For example: http://<HostName>:7051	Yes	Yes	Yes
ER_SERVICE_PORT	Indicates the port number where the entity resolution service is installed. Default value: 7051	Yes	Yes	Yes
Graphs				
HDFS_GRAPH_FILES_PATH	Indicates the filepath in the HDFS where the `graph.json` file is formed.	Yes	Yes	Yes
GRAPH_FILES_PATH	Indicates the directory in the Big Data server for graph files.	Yes	Yes	Yes
GRAPH_NAME	Indicates the name you want to assign to the global graph at the end of ETL.	Yes	Yes	Yes
IDCS **NOTE:** The IDCS related parameters are applicable only if IdcsRealm is used in the Realm parameter.				
IDCS_HOST	Indicates the server of the Oracle Identity Cloud Service (IDCS) instance.	Yes	Yes	Yes
IDCS_PORT	Indicates the port number of the IDCS instance.	Yes	Yes	Yes

Table 1: studio-env.yml Parameters

Parameter	Significance	Installing with OFSAA (Mandatory)	Upgrading with OFSAA (Mandatory)	Installing with non-OFSAA (Mandatory)
IDCS_SSL_ENABLED	Indicates if SSL is enabled for the IDCS application. Default value: true	Yes	Yes	Yes
LOGOUT_URL	Indicates the URL to redirect after logout from FCC Studio.	Yes	Yes	Yes
IDCS_TENANT	Indicates the IDCS tenant provided by the IDCS Administrator while creating the IDCS application for FCC Studio.	Yes	Yes	Yes
IDCS_CLIENT_ID	Indicates the IDCS client identifier provided by the IDCS Administrator while creating the IDCS application for FCC Studio.	Yes	Yes	Yes
IDCS_CLIENT_SECRET	Indicates the IDCS client secret provided by the IDCS Administrator while creating the IDCS application for FCC Studio.	Yes	Yes	Yes
FCDM_SOURCE	Indicates the source database for FCC Studio. The available options are ECM and BD. **NOTE:** ● FCC Studio can use either the BD or ECM schema as the source of FCDM data for the graph. ● Ensure to enter the value as ECM whenever ECM integration is required with Investigation Hub. Here, ECM schema is used as the source of the FCDM data to load the case information into the graph.	Enter BD or ECM	Enter BD or ECM	Enter NA

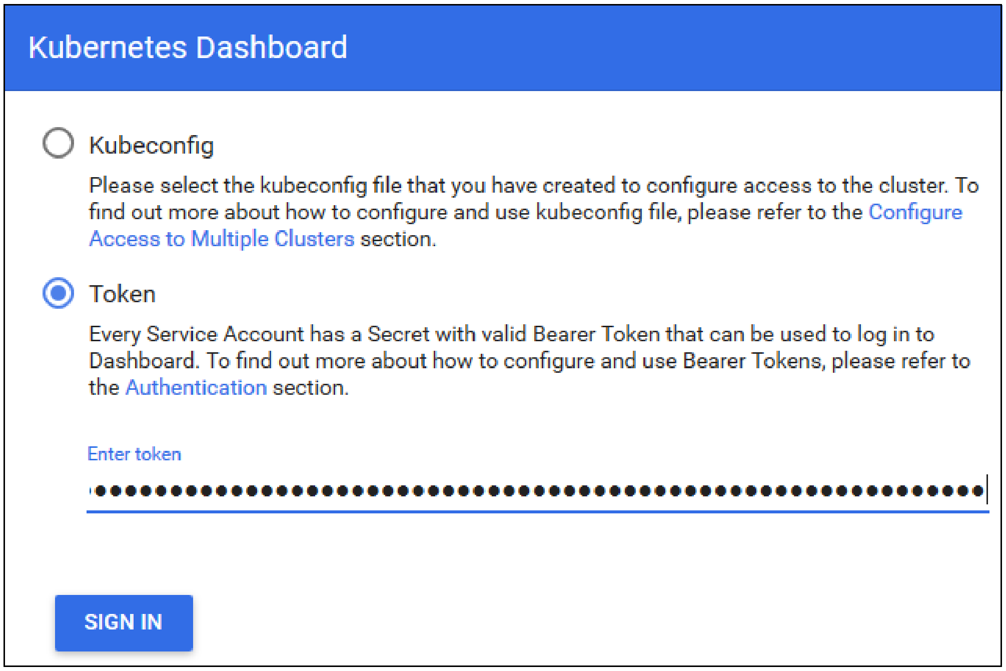
13 Appendix - Checking Logs of FCC Studio Services

The Kubernetes UI enables you to view the logs of the services installed as part of FCC Studio.

To check the logs of the FCC Studio services from the Kubernetes Dashboard, follow these steps:

1. Access the Kubernetes Dashboard.

   The **Kubernetes Dashboard** login page is displayed.



Kubernetes Dashboard

○ Kubeconfig

Please select the kubeconfig file that you have created to configure access to the cluster. To find out more about how to configure and use kubeconfig file, please refer to the Configure Access to Multiple Clusters section.

● Token

Every Service Account has a Secret with valid Bearer Token that can be used to log in to Dashboard. To find out more about how to configure and use Bearer Tokens, please refer to the Authentication section.

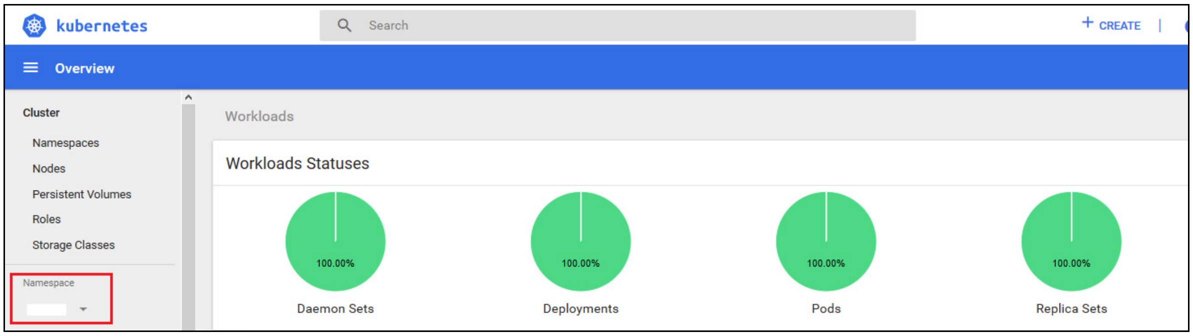
Enter token

●●

SIGN IN

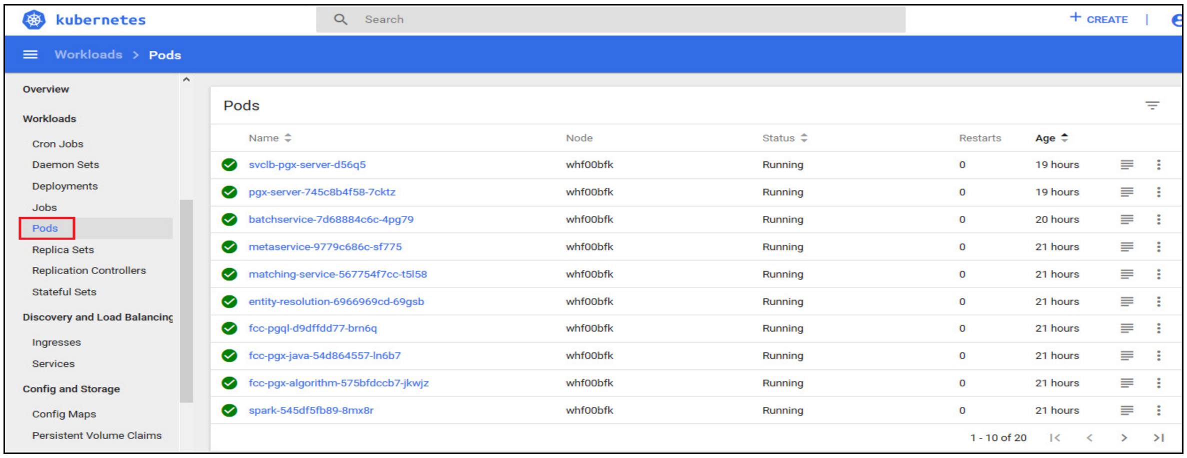
2. Select **Token** and enter the Admin user secret token in the **Enter Token** field.
3. Click **SIGN IN**.

   The Kubernetes Dashboard page is displayed.
4. Select <Namespace> from the **Namespace** drop-down list on the menu items displayed on the LHS.



5. Navigate to **Workloads** > **Pods** from the menu items displayed on the LHS.

   The **Pods** page is displayed with the details of all the services installed as part of the FCC Studio installation.



Pods

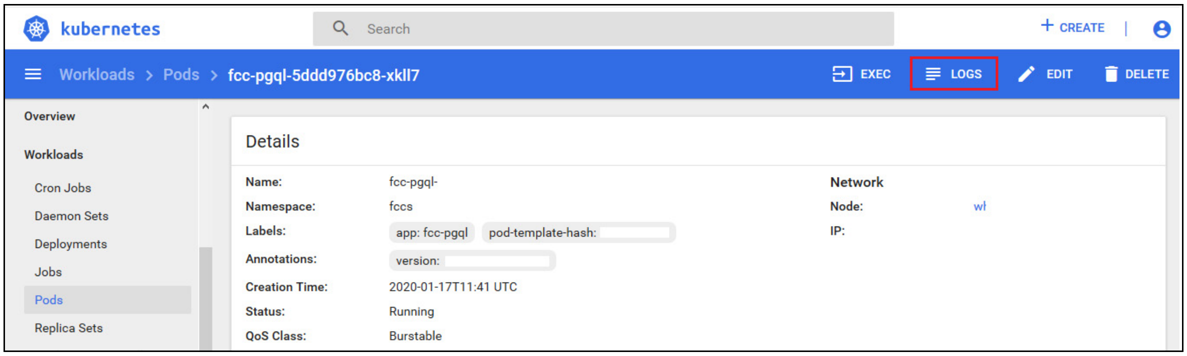
Name	Node	Status	Restarts	Age
svclb-pgx-server-d56q5	whf00bfk	Running	0	19 hours
pgx-server-745c8b4f58-7cktz	whf00bfk	Running	0	19 hours
batchservice-7d68884c6c-4pg79	whf00bfk	Running	0	20 hours
metaservice-9779c686c-sf775	whf00bfk	Running	0	21 hours
matching-service-567754f7cc-t5l58	whf00bfk	Running	0	21 hours
entity-resolution-6966969cd-69gsb	whf00bfk	Running	0	21 hours
fcc-pgql-d9dffdd77-brn6q	whf00bfk	Running	0	21 hours
fcc-pgx-java-54d864557-ln6b7	whf00bfk	Running	0	21 hours
fcc-pgx-algorithm-575bfdccb7-jkwjz	whf00bfk	Running	0	21 hours
spark-545df5fb89-8mx8r	whf00bfk	Running	0	21 hours

1 - 10 of 20

6. Click the service name from the **Name** column.

   The service details are displayed.

7. Click the **Logs** tab.

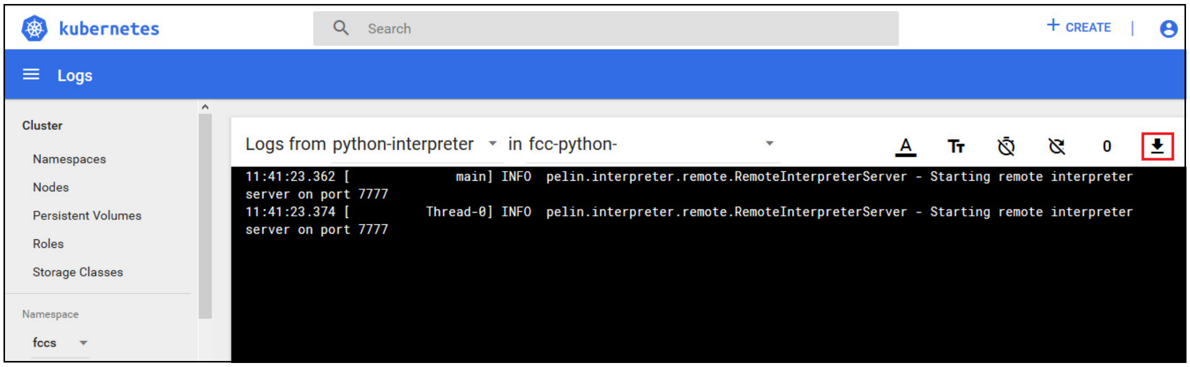


Details

Name: fcc-pgql-
Namespace: fccs
Labels: app: fcc-pgql pod-template-hash:
Annotations: version:
Creation Time: 2020-01-17T11:41 UTC
Status: Running
QoS Class: Burstable

Network
Node: wl
IP:

The service logs are displayed and you can download the logs by clicking **Download Logs** icon.



```
11:41:23.362 [             main] INFO  pelin.interpreter.remote.RemoteInterpreterServer - Starting remote interpreter
server on port 7777
11:41:23.374 [         Thread-0] INFO  pelin.interpreter.remote.RemoteInterpreterServer - Starting remote interpreter
server on port 7777
```

14 Appendix - Accessing the FCC Studio Application

To access the FCC Studio application, follow these steps:

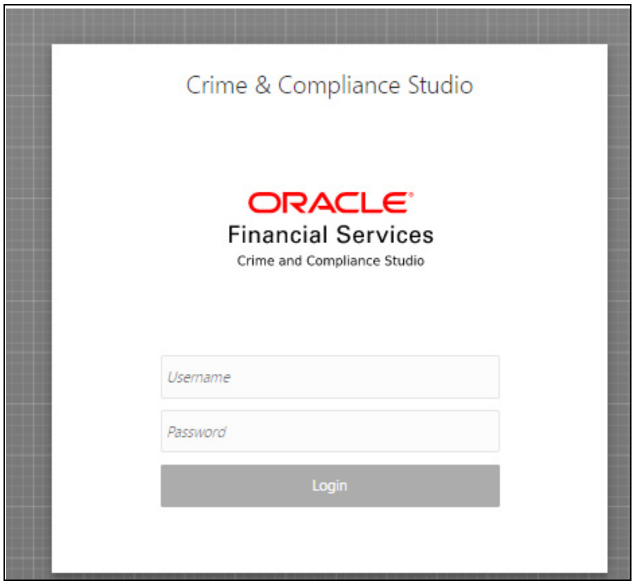
1. Enter the URL in the following format in the web browser:

   `https://<Host_Name>:<Port_Number>`

   Here `<Port_Number>` is 30078 for the FCC Studio application deployed on the K8s cluster.

   The FCC Studio application login page is displayed.

   Figure 1: Crime and Compliance Studio Login Page

   Crime & Compliance Studio

   ORACLE
   Financial Services
   Crime and Compliance Studio

2. Log in with the **Username** and **Password**.
3. Click **Login**.

   The Crime and Compliance Studio application's landing page is displayed with the list of all the out-of-the-box notebooks packaged with FCC Studio.

   You can check the logs of the FCC Studio services from the Kubernetes Dashboard. For more information, see Appendix - Checking Logs of FCC Studio Services.

15 Appendix - Starting/Stopping FCC Studio Services

This section describes how to start and stop the services required for FCC Studio.

Topics:

- Starting a Service
- Stopping a Service
- Starting the FCC Studio Application

15.1 Starting a Service

To start a service, follow these steps:

1. Open the `<Studio_Installation_Path>/out/deploy.sh` file

   The `deploy.sh` file contains details of the commands to start the FCC Studio services.

2. Navigate to the `<Studio_Installation_Path>/out` directory.
3. Run the start command in the console to start a service:

   For example:

   ```
   kubectl -n <Namespace> apply -f  deployments/authservice.yml
   ```

   You can check the logs of FCC Studio from the Kubernetes Dashboard. For more information, see Appendix - Checking Logs of FCC Studio Services.

15.2 Stopping a Service

To stop a service, follow these steps:

1. Navigate to the `<Studio_Installation_Path>/out` directory.
2. Run the stop command in the console to stop service:

   For example:

   ```
   kubectl -n <Namespace> delete -f  deployments/authservice.yml
   ```

15.3 Starting the FCC Studio Application

To start the FCC Studio application, follow these steps:

1. Navigate to the `<Studio_Installation_Path>/bin/` directory.
2. Run the following command:

   ```
   ./fcc-studio.sh --registry <registry URL>:<registry port>
   ```

   The FCC Studio application is restarted.

   Check the logs of the FCC Studio services from the Kubernetes Dashboard. For more information, see Appendix - Checking Logs of FCC Studio Services.

   Once all the services are up and running, the FCC Studio application can be accessed using the following URL:

   ```
   http://<HostName>:30078
   ```

16 Appendix - Post-deployment Interpreter Settings

After starting FCC Studio, the interpreter settings can be performed from the ***Interpreters*** page of the FCC Studio application UI. For information on configuring interpreters, see the *Configuring Interpreters* chapter in the *OFS Crime and Compliance Studio Administration Guide.*

16.1 Configuring the fcc-python Interpreter

16.1.1 Modifying the Python Images for the Python Interpreter

To modify Python packages in Python 3.6 or to add different versions of Python to the Python interpreter, you must modify the Python image. For information on modifying Python images, see *Modifying the Python Images for the Python Interpreter* section in the *OFS Crime and Compliance Studio Administration Guide* .

17 Appendix - Tables and Sequences

The list of tables and sequences that are to be dropped during redeployment of FCC Studio are as follows:

- Studio Schema Tables
- Studio Schema Sequences

17.1 Studio Schema Tables

The following table includes the details of the Studio Schema tables that must be dropped during the redeployment of the Studio application:

Table 1: Studio Schema Tables

DS_PARAGRAPH	DS_NOTEBOOK_TAGS	DS_TASK_RESULTS
DS_ENTITY_PERMISSIONS	DS_ROLE	DS_PERMISSION_ACTIONS
DS_GROUP	DS_IS_PERMITTED	DS_PERMISSION_MAPPING
DS_USER_PERMS_MAP	DS_USER_ROLES	DS_NOTEBOOK
DS_INTERPRETER_RESULT_MSGS	DS_USER	DS_PERMS_MAP_ACTIONS
DS_ENTITY_PERMS_MAP	DS_TASK	DS_GRAPH
DS_INTERPRETER_RESULT	DS_GROUP_PERMS_MAP	DS_NOTEBOOK_RELATIONS
DS_INTERPRETER_PROPS	DS_JOB	DS_PERMISSION
DS_ROLE_PERMS_MAP	DS_VISUALIZATION_TEMPLATE	DS_RESULT_MESSAGE
DS_INTERPRETER_ABILITIES	DATABASECHANGELOG	DATABASECHANGELOGLOCK
DS_USER_GROUPS	DS_INTERPRETER_VARIANT	DS_COMMENT
DS_PARAGRAPH_RELATIONS		

17.2 Studio Schema Sequences

The following table includes the details of the Studio Schema sequences that must be dropped during the redeployment of FCC Studio:

Table 2: Studio Schema Sequences

SEQ_COMMENT	SEQ_ENTITY_PERMISSIONS	SEQ_GRAPH
SEQ_GROUP	SEQ_INTERPRETER_RESULT	SEQ_INTERPRETER_VARIANT

Table 2: Studio Schema Sequences

SEQ_JOB	SEQ_NOTEBOOK	SEQ_PARAGRAPH
SEQ_PERMISSION	SEQ_PERMISSION_MAPPING	SEQ_RESULT_MESSAGE
SEQ_ROLE	SEQ_TASK	SEQ_USER
SEQ_VISUALIZATION_TEMPLATE		

18 Appendix - Using Python Virtual Environments with PySpark

To use Python Virtual Environments with PySpark, follow these steps:

1. Creating a Virtual Environment with Conda
2. Including Virtual Environment in the Init Container
3. Updating Interpreter Properties

18.1 Creating a Virtual Environment with Conda

> **NOTE** You can also use `virtualenv` to create your virtual environment instead of `conda`.

To create a virtual environment with Conda, follow these steps:

1. Ensure that you have `conda` and `conda-pack` installed.
2. Create your virtual environment using the following command:

   ```
   conda create -y -n <environment-name> python=<python-version>
   <additional-packages>
   ```

   > **NOTE** The `<environment-name>` can be chosen freely and subsequently must be used in further commands.

3. Activate your virtual environment using the following command:

   ```
   conda activate <environment-name>
   ```

4. Execute the following command to obtain the path to your virtual environment:

   ```
   which python
   ```

   The obtained result is referred to as `<environment-abs-path>`

5. Compress your virtual environment using the following command:

   ```
   conda pack -n <environment-name> -o <environment-abs-path>/<environment-
   name>.tar.gz
   ```

18.2 Including Virtual Environment in the Init Container

To include the virtual environment in the Init container, you must place the Virtual Environment in the same path as the Spark libraries. For more information, see Provide Custom Spark libraries.

18.3 Updating Interpreter Properties

All the properties can either be configured in the interpreter JSON files or from the ***Interpreters*** page of the FCC Studio application UI after starting the FCC Studio application.

- In the ***Spark Interpreter Settings*** page of the FCC Studio application UI (or `spark.json` file), change the following values:

  - Change the value of the `spark.yarn.dist.archives` property to `/var/olds-spark-interpreter/interpreter/spark/libs/<environment-name>/<environment-name>.tar.gz#<environment-name>`
  - Change the value of the `spark.pyspark.python` property to `./<environment-name>/bin/python`
- In the ***PySpark Interpreter Settings*** page of the FCC Studio application UI (or `pyspark.json` file), change the value of the `zeppelin.pyspark.python` property to `/var/olds-spark-interpreter/interpreter/spark/libs/<environment-name>/bin/python.`

19 Appendix - Executing Graph_Alive Notebook

In an Investigation Hub notebook, the graph is lost whenever a session is reset and this occurs as part of the session clean-up. You must execute the Graph_Alive notebook to retain the link to the graph even when a session is reset.

To execute the Graph_Alive notebook, follow these steps:

1. Access the FCC Studio application. For more information, see Appendix - Accessing the FCC Studio Application.
2. Click the Graph_Alive notebook.

   The Graph_Alive notebook is displayed.
3. Click **Run Paragraphs** ▷ to execute all the paragraphs in the notebook in sequential order.

20 Appendix - Uninstalling FCC Studio

To uninstall FCC Studio, follow these steps:

> **NOTE** Uninstalling the FCC Studio application deletes all the data from FCC Studio namespace.

1. Delete the FCC Studio namespace using the following command:

   ```
   kubectl delete namespace <Namespace>
   ```

2. Manually delete the FCC studio images for each Kubernetes node using the following command:

   ```
   docker rmi <Image ID>
   ```

   You can get the list of image IDs by running the docker images.

OFSAA Support Contact Details

Raise a Service Request (SR) in My Oracle Support (MOS) for queries related to the OFSAA applications.

Send Us Your Comments

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, indicate the title and part number of the documentation along with the chapter/section/page number (if available) and contact the Oracle Support.

Before sending us your comments, you might like to ensure that you have the latest version of the document wherein any of your concerns have already been addressed. You can access My Oracle Support site which has all the revised/recently released documents.

ORACLE®